

抽出計算コード(MIXSET-X)

A Computer Code for Simulating the PUREX Solvent Extraction Process

(研究報告)

1999年3月

核燃料サイクル開発機構
東海事業所

複製又はこの資料の入手については、下記にお問い合わせ下さい。

〒319-11 茨城県那珂郡東海村大字村松4-33

動力炉・核燃料開発事業団

東海事業所 技術開発推進部・技術管理室

Enquires about copyright and reproduction should be addressed to: Technology Management Section, Tokai Works, Power Reactor and Nuclear Fuel Development Corporation, 4-33 O-aza-Muramatsu, Tokai-mura, Naka, Ibaraki-ken, 319-11, Japan

動力炉・核燃料開発事業団 (Power Reactor and Nuclear Fuel Development Corporation)

抽出計算コード (M I X S E T - X)

- A Computer Code for Simulating the PUREX Solvent Extraction Process -
(研究報告)

内藤大志*, 須藤俊幸*, 麻川和裕**, 柏木栄介**

要旨

ミキサ・セトラ型の連続抽出器を用いた使用済核燃料再処理の溶媒抽出工程を対象とした定常状態及び過渡変化のシミュレーションを行う計算コードである Revised M I X S E T (PNCT841-79-26)を改良し, M I X S E T - X (ミックスセット・テン)を開発した。

Revised M I X S E Tは8成分, 6化学反応の計算を対象とした大型計算機上のコードであったが, M I X S E T - XではF P, T R Uを含めた31成分の分配係数式及び, 合計45の化学反応速度式を内蔵し, 将来の拡張も比較的容易である。コーディングについては全面改訂を実施し, 計算の安定性を改善するため従来の数値解法を見直して陰解法スキームに基づく数値計算手法を採用すると共に, 機種依存のコードは極力排除した。また, 単純な貯槽モデルを新たに導入してプロセス液のフィードバックを模擬した計算を可能とし, 簡易的な溶媒劣化計算の機能も組み込んだ。

M I X S E T - Xでは, 各成分毎に複数の分配係数を内蔵し, 計算インプット上でそれらを選択することが可能である。また, 内蔵した化学反応速度式の速度を計算インプット上で調節することが可能であり, 化学反応速度を0とすることによって反応を抑制することも可能である。本計算コードによって, 東海再処理工場の抽出工程全体の同時計算が可能となった。

今後, 内蔵した種々の分配係数及び化学反応式について, 工場の運転データ等との比較・検討を予定している。

* 再処理センター 技術部

** 住友金属鉱山(株)

March, 1999

MIXSET-X :

A Computer Code for Simulating the PUREX Solvent Extraction Process

Motoyuki NAITO*, Toshiyuki SUTO*,
Kazuhiro ASAKAWA** and Eisuke KASHIWAGI**

Abstract

MIXSET is a FORTRAN code developed in about 1980 to simulate the PUREX solvent extraction process using mixer-settler type extractors. A rebuilt version of it, MIXSET-X, has been developed to give new features listed below.

- 1) Multiple distribution coefficients of 31 constituents including actinides and FPs and 45 chemical reactions are built in.
- 2) Users can select any distribution coefficients and chemical reactions to be handled and adjust each reaction rate constants by the input data.
- 3) Recycle flow through buffer tank and TBP degradation calculation can be treated.

MIXSET-X can simulate the whole solvent extraction processes of Tokai reprocessing plant at one calculation run. Program structure and mathematical modeling of the system have been entirely changed to get calculational stability and program readability. This report describes detailed features and program list of the code. For further study, validation for calculation results of MIXSET-X will be performed.

* Tokai Reprocessing Center Technology Co-ordination Division.

** Sumitomo Metal Mining Co. LTD.

1	はじめに	1
2	抽出工程のモデル化	3
2. 1	抽出器のモデル化	3
2. 2	貯槽のモデル化	5
2. 2. 1	フィードバック計算機能	5
2. 2. 2	水相貯槽のモデル化	6
2. 2. 3	有機相貯槽のモデル化	6
2. 3	抽出プロセスフローのモデル化	7
2. 4	物質収支の基礎式	9
2. 4. 1	ミキサセトラでの物質収支式	9
2. 4. 2	貯槽での物質収支式	10
2. 5	流量計算	11
2. 6	分配平衡	13
2. 7	段効率	14
2. 7. 1	Murphree型段効率	14
2. 7. 2	回収率型段効率	16
2. 8	溶媒劣化	18
2. 8. 1	概要	18
2. 8. 2	TBPの化学分解	18
2. 8. 3	TBPの放射線分解	18
2. 8. 4	生成DBP量 Q_{DBP}	20
3	数値解法	23
3. 1	基本方針	23
3. 1. 1	概要	23
3. 1. 2	分配係数の計算	23
3. 1. 3	化学反応計算	23
3. 2	過渡計算手法	25
3. 2. 1	抽出器の物質収支基本式	25
3. 2. 2	分配平衡式及び段効率式	26
3. 2. 3	差分化手法	27
3. 2. 4	ニュートン法による繰り返し計算手順	27
3. 2. 5	行列計算手法	29
3. 3	定常計算手法	30
3. 3. 1	現状のMIXSET-Xによる定常値の計算	30
3. 3. 2	定常計算を組み込む場合の方針	30
4	MIXSET-X入力説明	31
4. 1	計算コードの制限	31
4. 2	入力データおよびフォーマット	32
4. 2. 1	フリーフォーマット入力	32
4. 3	入力仕様	33
4. 3. 1	入力キーワードの説明	33
4. 3. 2	入力カード説明	34
4. 4	エラーメッセージ	49
4. 4. 1	入力のエラーチェック	49
4. 4. 2	計算開始後のメッセージ	51
5	総合解析計算	52
5. 1	実験解析	52
5. 2	再処理工場の N_p 、 T_c の分配の計算	66
5. 3	N_p 、 T_c 、 Z_r の存在によるフリーTBP減少の抽出特性への影響	70
	参考文献	82

付録A	MIXSET-Xの取り扱い成分のまとめ	付 - 1
付録B	MIXSET-Xの組み込み化学反応	付 - 3
付録C	MIXSET-Xの組み込み分配係数	付 - 6
付録D	MIXSET-X組み込み化学反応の速度定数	付 - 21
付録E	ルテニウム錯体の存在比計算式	付 - 34
付録F	エラーメッセージ一覧	付 - 34
付録G	MIXSET-XとEXTRA, Mの化学反応式のちがい	付 - 39
付録H	容量マトリックス法	付 - 45
付録I	Solute-Free濃度と通常濃度の変換式	付 - 47
付録J	抽出器の定常計算のまとめ	付 - 48
付録K	再処理工場のNp, Tcの分配計算の計算体系図と入力データリスト	付 - 60
付録L	MIXSET-Xのプログラム構造	付 - 69
付録M	MIXSET-Xのプログラムの変数説明	付 - 71
付録N	MIXSET-Xソースプログラムリスト	付 - 77

1 はじめに

MIXSET⁽¹⁻²⁾は、ミキサセトラ型の連続抽出器を用いた使用済核燃料再処理の溶媒抽出工程を対象とした定常状態及び過渡変化のシミュレーションを行う計算コードとして広く利用されてきた。

旧動燃事業団再処理工場では、抽出工程の技術的検討を行うためにMIXSETコードに対して、取り扱い成分、分配係数、化学反応の追加などの機能拡張を行うとともにパーソナルコンピュータへの移植作業を行い、改訂を重ねながら使用してきた（以下パーソナルコンピュータ版のコードをMIXSET98という）。これらの機能拡張では、初期のMIXSETコード開発当時のプログラミング手法とプログラム構成を継承しつつ、その制約の中でコードの改訂作業を行ってきた。このことは計算インプット上の計算オプションを複雑かつ多様化させることとなり、同時に、プログラム内では条件分岐が多用されることなどにより処理の流れが分かりづらくなっていた。さらに、機能の追加・修正を行う際にプログラム全体の整合性を考慮することが困難となっていた。そのため、例えば、計算オプションの組合せによっては正しく計算処理されないという不整合が生じることがあった。

一方、近年では軽水炉の高燃焼度燃料やMOX燃料の再処理などで、抽出工程におけるNp, Tcの成分挙動が着目されており、このような技術的検討に対応していくためにMIXSETの機能をさらに拡張する必要があった。今回、MIXSETを抽出工程の多面的な技術検討を行うツールとして機能拡張を行うに当たっては、これまでのようなプログラム構成の継承を行わず、コーディングの全面改訂を実施することとした。

今回の改訂では、従来マイクロ成分として取り扱ってきたNp, Tc等をマクロ成分とし、新たにPu(V), Pu(VI)等をマクロ成分として取り扱うとともに、成分間の化学反応を大幅に考慮した計算ができるように改訂した。また、単純な貯槽モデルを新たに導入してプロセス液のフィードバックを模擬した計算を可能とし、簡易的な溶媒劣化計算の機能も組み込んだ。コーディングの全面改訂に際しては、計算の安定性を改善するため、従来の数値解法を見直して陰解法スキームに基づく数値計算手法を採用した。

今回改訂したMIXSET（以下、MIXSET-X：「ミックスセット・テン」という）は、各成分毎に複数の分配係数を内蔵し、計算インプット上でそれらを選択することが可能である。また、内蔵した化学反応速度式の速度を計算インプット上で調節することが可能であり、化学反応速度を0とすることによって反応を抑制することも可能である。従って、MIXSET-Xは、完成された抽出計算コードというよりは、むしろ、各種の取り扱い成分の抽出挙動に対して種々の分配係数や化学反応の影響も解析しうる計算ツールとしての性格が強い。

現段階のMIXSET-Xは過渡計算のみが可能であり、定常計算結果は長時間の過渡計算によって得られる最終状態の結果で代用することになる。

以下に今回の改訂の主要部分を示す。

・マクロ成分の追加。

次の成分をマクロ成分として追加した。

Pu ⁺⁶	6価のプルトニウム
Pu ⁺⁵	5価のプルトニウム
Np(N)	4価のネプツニウム
Np(V)	5価のネプツニウム
Np(VI)	6価のネプツニウム
Tc ⁺⁴	4価のテクネチウム
Tc ⁺⁵	5価のテクネチウム
Tc ⁺⁶	6価のテクネチウム
Tc ⁺⁷	7価のテクネチウム
UO ₂ (NO ₃)(TcO ₄)	
Pu(NO ₃) ₃ (TcO ₄)	
Zr(NO ₃) ₃ (TcO ₄)	
Zr	ジルコニウム

- ・ N p , T c の化学反応を取り入れた。
N p と T c に関する化学反応式は J A E R I 1 3 3 1 ⁽³⁾に記載されているものを使用した。
これらの化学反応を考慮しない従来の M I X S E T 互換モードも利用可能である。
- ・ 段効率定義式の変更
新しい数値計算手法ではプログラムの作成が困難な Murphree 型段効率を削除し、抽出の回収率に効率を持ち込んだ段効率に変更した。これは、 J A E R I 1 3 3 1 ⁽³⁾で用いている段効率と同様の定義である。
- ・ 溶媒劣化計算機能を組み込んだ
- ・ 数値計算手法を陰解法に変更し、複雑な化学反応、段効率等を統一して扱えるようにした。
- ・ 不要機能等の削除
従来の M I X S E T ⁽¹⁻²⁾にある次の計算機能を削除した。
 - 物質移動計算
 - 電解還元反応
 - 最適化機能
 - 液面高さの時間変化
 - 還元剤選択機能
 - 供給液の温度
 - 平衡定数のイオン強度の3次式の係数入力
 - 分配係数の表入力
 - 一次化学反応
 - 温度計算機能
 - 実験値データプロット
 - Mixing-Plug型段効率
 - Murphree型段効率
 - 複数計算ケース順次計算機能

M I X S E T - X を使用するために必要なメモリ及び概略の計算時間は次のとおりである。

プログラム実行時に必要な計算機メモリ

プログラムテキスト+格納データ+変数領域の合計 約60 MB

定常状態に達するまでの過渡計算に必要な概略計算時間

Pentium 100MHz , RAM 80MB, OSにWindows95を用いた計算機を使用し、

M I X S E T - X による計算を単独で行った場合、おおよそ次のようになる。

25段程度の抽出器1つで構成される抽出プロセスの計算	約10分
3つの抽出器と2つの貯槽を組み合わせた抽出プロセスの計算	約1時間30分
9つの抽出器と5つの貯槽を組み合わせた抽出プロセスの計算	約9時間

一般的に、機器を組み合わせると急速に計算時間が長くなるため、抽出プロセスのフローシートを分割して計算できる場合には、分割して計算するほうが得策である。分割して計算した場合と一括して計算した場合で計算結果が数値計算の誤差内で一致することは確認済みである。

なお、M I X S E T - X のソースコードは標準的なFORTRAN77で記述し、機種依存が少なくなるように配慮した。付録Nに示したソースリストは、SunOSを用いたunixWSで動作確認を行ったが、他の計算機に移植することも比較的容易であると思われる。

2 抽出工程のモデル化

2.1 抽出器のモデル化

実際のミキサセトラの構造は非常に複雑であるため、これを忠実に実現するのは困難であり通常性能の計算機での動作を考えると実用的でない。MIXSET-Xでは、一つの抽出器として最大25段のステージで構成される工程を取り扱うこととした。また、図2-1に示すように、有機相はステージの番号の小さい方から大きい方へステージ間を流れ、水相はその逆の方向に流れ、全体として水相と有機相は反対方向へ流れる。ステージ1には有機相フィード液、ステージnには水相フィード液が必要である。このフィード液は他の抽出器からの出口液でも良い。

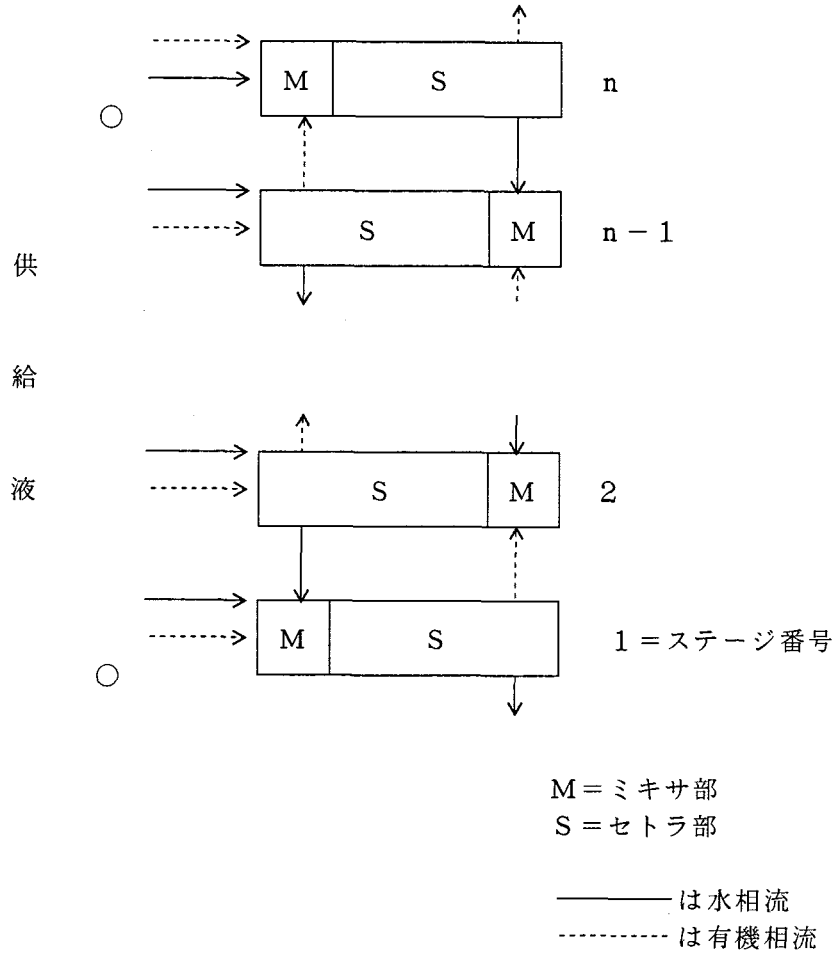


図2-1 ミキサセトラ内のモデル化

ミキサセトラの各ステージは図2-2に示すようにミキサ部とセトラ部の二つの部分から構成される。ミキサ部とセトラ部の各相の中の成分濃度は一様であるとしてモデル化し、ミキサ部では有機相と水相は完全混合しており、各相の容積比は流量比に等しいと考える。従って、 i 段のミキサ部での各相の液量割合 h_a, h_o は次の式で計算される。 W, B はそれぞれ水相、有機相の流量である。

$$h_a^i = h^i \frac{W^{i+1} + W_f^i + W_R^i}{W^{i+1} + B^{i-1} + W_f^i + B_f^i + W_R^i}$$

$$h_o^i = h^i - h_a^i$$

セトラ部においては、両相の容積比は界面レベルにより決定される。通常では、界面レベルは 0.5 に設定してある。MIXSET-X ではセトラ部の界面レベルの変更は考慮しない。

$$H_a^i = H^i \cdot L^i(t)$$

$$H_o^i = H^i - H_a^i$$

流れは押し出し式と仮定するので、ステージ間の流量は供給液入量だけに依存する。

$$W^i(t) = W^{i+1}(t) + W_f^i(t) \quad (i = 1 \sim N-1)$$

$$B^i(t) = B^{i-1}(t) + B_f^i(t) \quad (i = 2 \sim N)$$

$$W^N(t) = W_f^N(t)$$

$$B^1(t) = B_f^1(t)$$

ここで、 $W^i(t), B^i(t)$ とは、ステージ i のセトラ部より流出する液流量であり、 $W_f^i(t), B_f^i(t)$ は入力される供給量なので既知である。

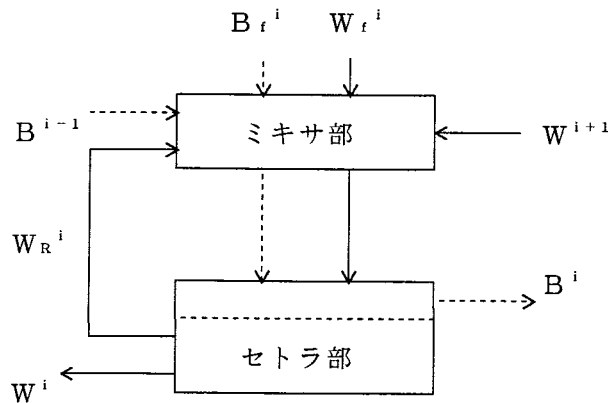


図2-2 ミキサセトラのステージのモデル化

2. 2 貯槽のモデル化

2. 2. 1 フィードバック計算機能

MIXSET-Xにフィードバック計算機能を組み込むに当たって、貯槽モデルとしては、「貯槽での時間遅れを考慮した計算モデル」を採用することとした。

具体的には、MIXSET-Xの基本プログラムに対し以下の様な組み込み作業を行った。

- ・一定の液量を持った貯槽(液量変化のない貯槽)のモデルを導入し、抽出器の流出液を貯槽に入れ、貯槽より抽出器に液をフィードできるようにする。
- ・抽出器の流出液を貯槽に入れるときに酸濃度調整とプルトニウム酸化を行うことができるようにする。抽出器と貯槽の組み合わせは任意にできるものとする。
- ・目標とする酸濃度、流量、プルトニウム酸化率は入力とする。硝酸以外の濃度は、貯槽に流入する段階で物質収支が合うように自動調整されるものとする。
- ・水相の貯槽にはプルトニウムの酸化機能を持たせる。

この方法の短所・長所は次のようになる。

短所：

貯槽モデルにより濃度変化の時間遅れがあるため、貯槽での液滞留時間によっては計算結果が定常に達するまでかなりの時間がかかる。貯槽モデルがあるため、プログラムが複雑になる。ただし、液滞留時間による定常状態への時間は貯槽の液量を小さくする事により回避できる。

長所：

貯槽による濃度変化の時間遅れが模擬できるため、時間変化の計算結果にある程度の信頼性がある(貯槽での液量変化を考慮していないため、現実を模擬しているとは言えない)。貯槽モデルがあるものの、貯槽での液量変化を考えないため、入力が比較的容易である。

なお、フィードバックを考慮した場合には従来のMIXSETの行列計算手法を用いる事ができないため、MIXSET-Xでは「容量マトリックス法⁴⁾」と呼ばれる計算手法を採用した。詳細については後述する。

2. 2. 2 水相貯槽のモデル化

MIXSET-Xで考慮する水相貯槽は液量変化の無い単純な完全混合の貯槽であり、液の出口は2つである。水相貯槽からの総出口流量は、酸調整を行う場合以外、入り口流量と同じである。2つの各出口流量は、入力で与える「総出口流量に対する比」により計算される。水相貯槽内の濃度は完全混合モデルで計算され、濃度変化についての時間遅れが考慮される。

水相貯槽においては、プルトニウム酸化工程を模擬できるようにプルトニウム酸化率の入力を行うことができる。酸化率として0以外の値を入力したときは、この貯槽に入力された液のプルトニウムは酸化率に応じて3価から4価になる。このとき、ウランはすべて6価に、HANとヒドラジンは完全に分解されるものとしている。さらに、酸調整槽を模擬するために、酸調整後の貯槽の硝酸濃度を入力で指定することができる。酸調整を行う場合、「貯槽出口の目標酸濃度」と「貯槽の出口流量」を入力として与える必要がある。このとき「調整用酸の酸濃度」を入力として与えると、目標酸濃度及び流量を達成するために必要な調整酸の酸濃度との差を出力する（「調整用酸の酸濃度」は貯槽出口の酸濃度の計算の際に用いられない）。

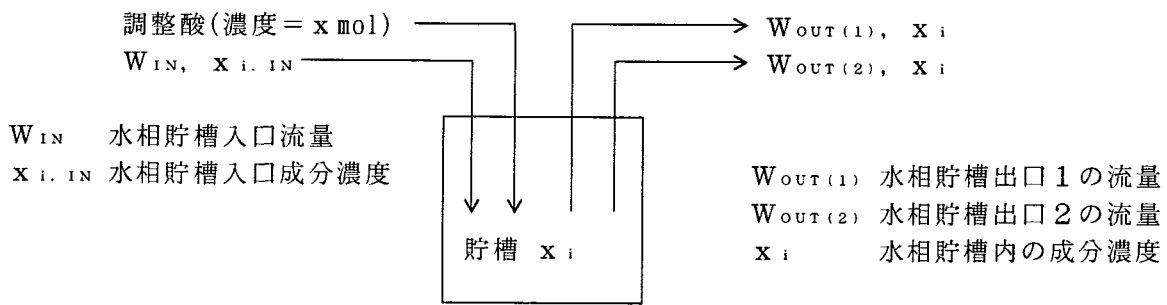


図2-3 水相貯槽のモデル化

2. 2. 3 有機相貯槽のモデル化

MIXSET-Xで考慮する有機相貯槽は液量変化の無い単純な完全混合の貯槽とする。液の出口は2つであり、有機相貯槽からの総出口流量は入口流量と同じである。2つの各出口流量は入力で与える「総出口流量に対する比」により計算される。有機相貯槽内の濃度は完全混合モデルで計算され、濃度変化についての時間遅れが考慮される。

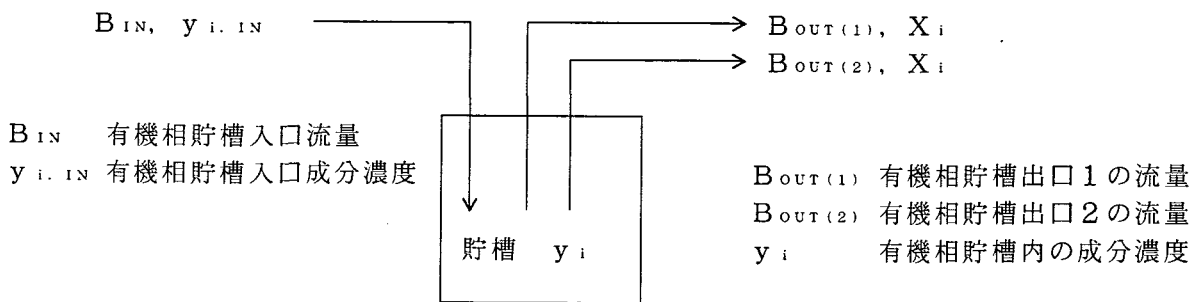


図2-4 有機相貯槽のモデル化

2.3 抽出プロセスフローのモデル化

MIXSET-Xでは、ミキサセトラと貯槽を合わせて19までの機器を組み合わせたプロセスを取り扱うことが可能である。溶解液フィード等のフィード液は、ミキサセトラのセトラ部を除いた任意の場所に水相、有機相それぞれ30まで設定することができる。貯槽の出口が2つあり、フィードバック計算が可能であるため、図2-5に示すような水相に循環流がある抽出プロセスのモデル化が可能である。また、複数の貯槽を組み合わせることがにより、例えば図2-6に示すような複雑な抽出プロセスのモデル化も可能であり、抽出プロセスをモデル化する際の自由度は高い。抽出器から流出する液出口は水相と有機相でそれぞれ一つであり貯槽からの液出口は2つに制限されているものの、実用上は、ミキサセトラと貯槽から構成される任意の抽出プロセスのモデル化が可能である。

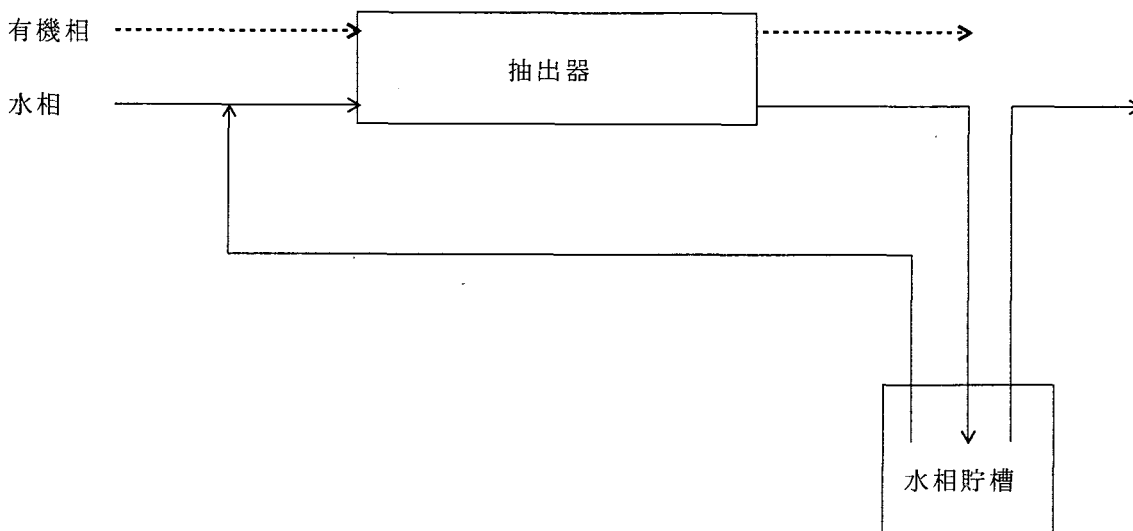


図2-5 水相に循環流のある抽出プロセスのモデル化

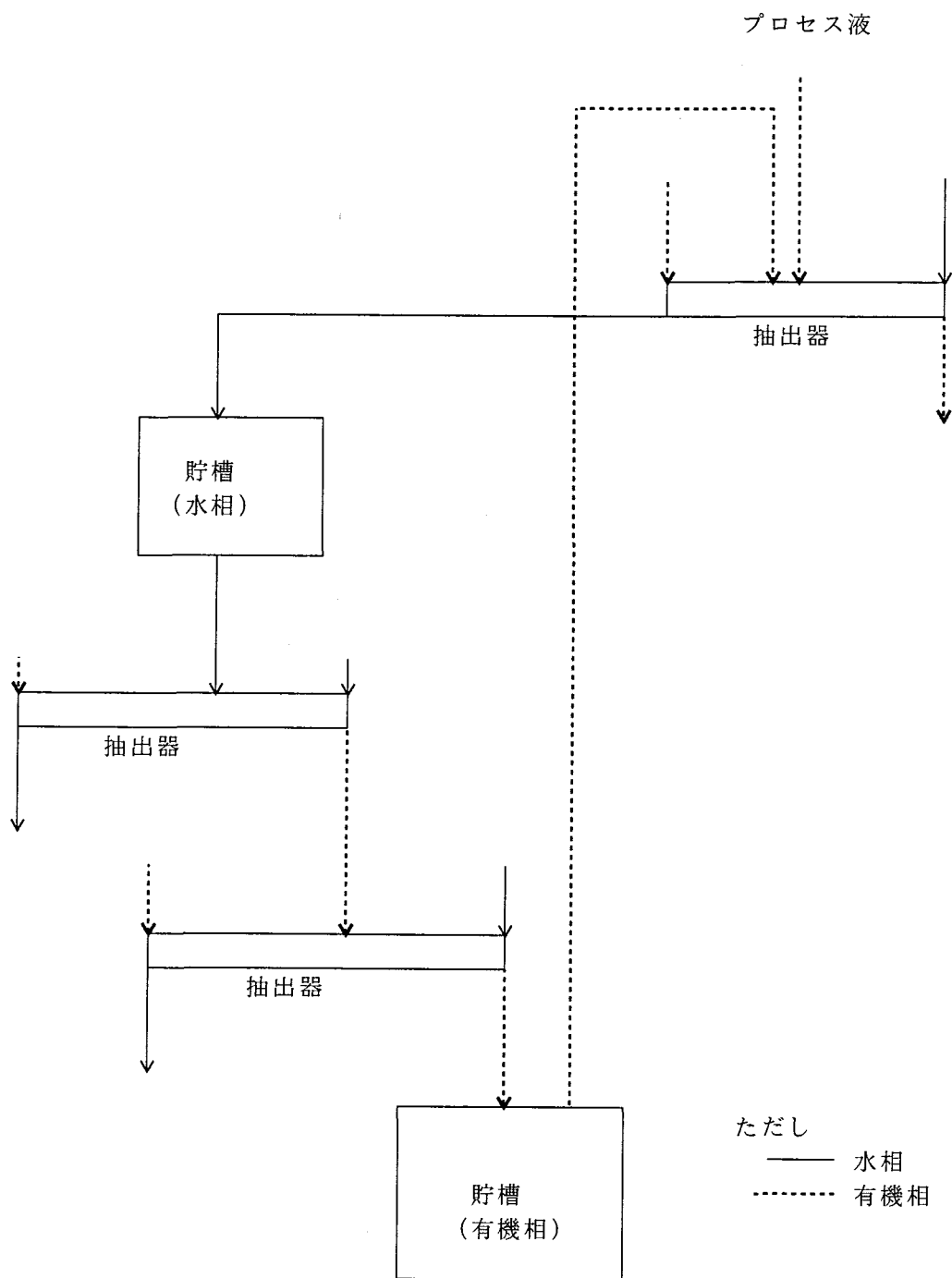


図 2 - 6 抽出工程全体のモデル化

2. 4 物質収支の基礎式

2. 4. 1 ミキサセトラでの物質収支式

MIXSET-Xコードにおいて、ある1段内の濃度分布はミキサ部の水相 (x_m)、有機相 (y_m)、セトラ部の水相 (x_s)、有機相 (y_s) の4点で代表される。i 段目の物質収支は図2-7のようになる。

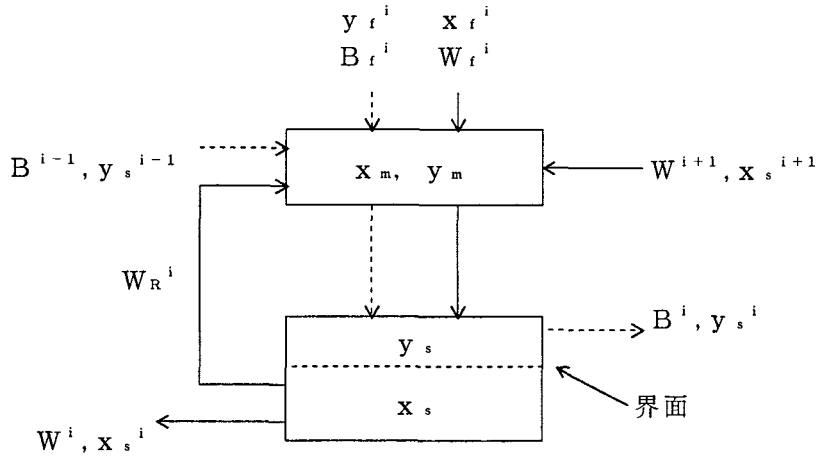


図2-7 ミキサセトラのステージ i での物質収支

ここで、化学反応項の増分を P (mol/hr)、減分を K (mol/hr) とする。その他の記号は2. 1節のミキサセトラのモデル化で使用したものと同一である。

ミキサ部では、

$$h_a^i \frac{d x_m^i}{d t} + h_o^i \frac{d y_m^i}{d t} = W_m^i \cdot x_{in}^i + B_m^i \cdot y_{in}^i + W_R^i \cdot x_s^i - (W_m^i + W_R^i) \cdot x_m^i - B_m^i \cdot y_m^i + P_m^i - K_m^i \dots (1)$$

ここで、

$$W_m^i = W^{i+1} + W_f^i$$

$$B_m^i = B^{i-1} + B_f^i$$

$$x_{in}^i = \frac{W^{i+1} \cdot x_s^{i+1} + W_f^i \cdot x_f^i}{W_m^i}$$

$$y_{in}^i = \frac{B^{i-1} \cdot y_s^{i-1} + B_f^i \cdot y_f^i}{B_m^i}$$

である。

セトラ部水相では、

$$H_a^i \frac{d x_s^i}{d t} = (W_m^i + W_R^i) \cdot x_m^i - (W^i + W_R^i) \cdot x_s^i + P_{sa}^i - K_{sa}^i \dots (2)$$

セトラ部有機相では、

$$H_o^i \frac{d y_s^i}{d t} = B_m^i \cdot y_m^i - B^i \cdot y_s^i + P_{so}^i - K_{so}^i \dots (3)$$

と表わされる。(1)、(2)、(3)式の微分方程式が段1~ N_s について成立する。ミキサ部については、 x_m と y_m の関係式が分配係数と段効率によって与えられる。段数 N_s 、成分数 N とおくと、 $N \times N_s \times 4$ 個の未知数の非線型連立常微分方程式が成立する。ただし、非線型性は分配係数と反応項の関数型によるので、これらが線型の場合、方程式も線型となる。

2. 4. 2 貯槽での物質収支式

(1) 水相貯槽の物質収支式

MIXSET-Xでは、水相貯槽の濃度は完全混合モデルにより計算される。物質収支式は次のようになる。総出口流量 ($W_{OUT(1)}$ と $W_{OUT(2)}$ の和)は、通常、水相貯槽入口流量に等しい。酸調整を行う貯槽では、総出口流量及び出口酸濃度は入力で与えられる値である。また、プルトニウム酸化率が入力された場合には、この貯槽に入力された液のプルトニウムは酸化率に応じて3価から4価になる。このとき、ウランはすべて6価に、HANとヒドラジンは完全に分解されるものとして計算する。

水相貯槽の物質収支式

$$V \frac{d x_i}{d t} = W_{IN} \cdot x_{i, IN} - W_{OUT(1)} \cdot x_i - W_{OUT(2)} \cdot x_i \quad \dots\dots\dots(4)$$

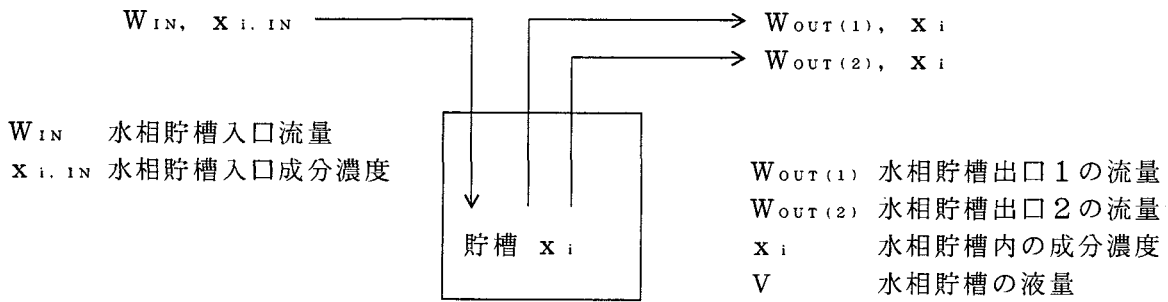


図2-8 水相貯槽の物質収支

(2) 有機相貯槽の物質収支式

MIXSET-Xでは、有機相貯槽の濃度は完全混合モデルにより計算される。物質収支式は次のようになる。総出口流量 ($B_{OUT(1)}$ と $B_{OUT(2)}$ の和)は、有機相貯槽入口流量に等しい。

有機相貯槽の物質収支式

$$V \frac{d y_i}{d t} = B_{IN} \cdot y_{i, IN} - B_{OUT(1)} \cdot y_i - B_{OUT(2)} \cdot y_i \quad \dots\dots\dots(5)$$

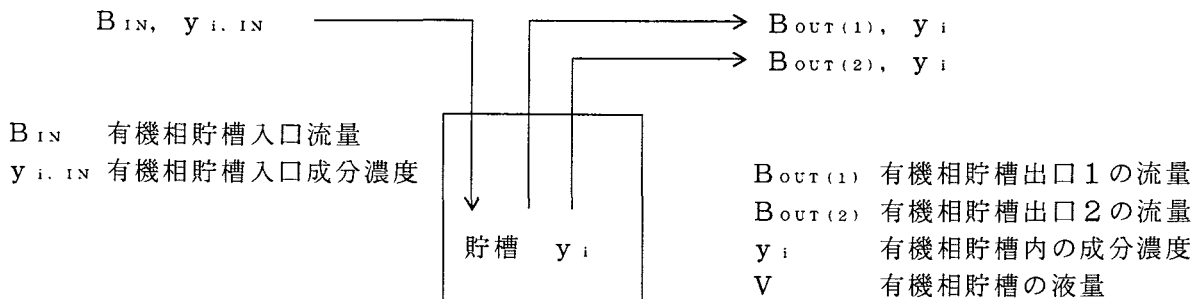


図2-9 有機相貯槽の物質収支

2.5 流量計算

従来のMIXSETでは、フィードバックが無い場合フィード流量をプロセスの下流側に向かって積算していくことにより体系内の流量を順次決定することができた。フィードバックがある場合には、これが多少複雑になってくる。MIXSET-Xでは次の2ステップに分けて流量の計算を行う。

(1) 機器(抽出器または貯槽)からの総流出流量を計算する。

これは、次の一次連立方程式(流量方程式)を解くことによって計算できる。まず、水相の流量方程式について考える。機器*i*からの水相の総流出流量を x_i とし、水相フィード番号 α の機器*i*へのフィード流量を $F_{\alpha, i}$ とする。また、機器*i*に機器*j*から流入する水相の流量を $x_{j, i}$ とする。このとき、各機器において流入する流量の和は機器の出口流量に等しいことから機器毎に次の式が成立する。

$$x_i = \sum_j x_{j, i} + \sum_{\alpha} F_{\alpha, i}$$

ただし、

$$x_{j, i} = R_{j, i} x_j$$

$$R_{j, i} = \text{流出流量比(機器 } j \text{ がミキサセトラの場合は } 1)$$

この方程式は x_i に関する一次連立方程式であり、式の数(機器の数)だけある。また、 $F_{\alpha, i}$ は入力と与えられる量であるから、一次連立方程式は、不定にならなければ必ず解が存在し、解を求めることができる。解が不定になる場合は、循環的な流れがあり必要な流量の入力がない場合に相当する。従って、意味のあるプロセス体系の場合には必ず解が存在する。

上で述べた一次連立方程式で構成される行列を流量決定行列と呼ぶことにする。流量決定行列をA、機器より流出する総流出流量をx、機器へ流入する外部からのフィード液流量ベクトルをBとする。機器がN個ある場合のAおよびB要素の設定方法は以下になる。

- ・ Aを単位行列に設定する。
- ・ 機器*i*について次の設定を1からNまで繰り返す。

機器*i*がミキサセトラのとき

$$A_{j, i} \leftarrow A_{j, i} - 1$$

$$B_i \leftarrow (\text{機器 } i \text{ にフィードする液の総流量})$$

ただし j はミキサセトラ水相流出液の流出先機器番号である。

機器*i*が貯槽のときすべての*k*について次の処理を行う。

$$A_{j(k), i} \leftarrow A_{j(k), i} - R_k$$

$$B_i \leftarrow (\text{機器 } i \text{ にフィードする液の総流量})$$

ただし R_k 流出口*k*より流出する液の流出流量比
 $j(k)$ 貯槽の流出口*k*の流出液の流出先機器番号

- ・ 酸調整を行う貯槽の機器番号を*i*とするととき次の修正を行う。

$$A_{i, j} \leftarrow 0 \quad (i \neq j)$$

$$A_{i, i} \leftarrow 1$$

$$B_i \leftarrow (\text{酸調整槽 } i \text{ より流出する液の総流量})$$

連立一次方程式 $Ax = B$ を解くことにより機器ごとの流出流量が得られる。ここで、流量方程式を解くために行列計算を必要とするが、LU分解法により一回LU分解しておけば後は後退代入だけで簡単に流量を計算することができるので計算のCPU負荷は問題にはならない。

(2) 機器毎に機器内部の流れの流量を計算する。

上記の方法により機器の出口流量は既にわかっている。このとき、各機器の流入流量のうち他の機器からの流量は、上流側機器の出口流量として既に計算されていることになる。また、その他の流入流量はフィード流量として与えられることになっている。従って、各機器での流入流量はすべて計算されていることになる。そこで、押し出し流れを仮定すれば機器内部の流量は下流側に向かって積算することにより簡単に計算できる。

以上、水相の場合について述べたが、有機相の場合も全く同様である。

2.6 分配平衡

溶媒抽出工程は各抽出成分の分配係数の差によって、各成分を分配するので、分配平衡データの扱いは重要な因子となる。ここではMIXSETコード内における分配平衡の定義を明らかにしておく。コード中では分配係数Dが抽出成分の水相濃度と有機相濃度とを関連させる。もちろん実際の抽出工程において、各成分ともに水相と有機相との間に分配平衡が完全に成立していることは希であるが、本報告で述べる分配係数は平衡状態における抽出成分の水相と有機相の濃度比を表わしている。分配係数は共存する成分の水相濃度によって表わされるものとする。つまり、

$$D = \frac{y}{x}$$

$$D = f(x_1, x_2, \dots, x_N) \quad (1 \sim N \text{は共存する成分番号})$$

分配係数Dの表現法によって成分のタイプが決定される。つまり、マクロ成分とマイクロ成分に分類される。

ここで、マクロ成分とは成分濃度がお互いに他の成分の分配係数へ影響を及ぼしている成分（Interactive成分）を示す。お互いに影響を及ぼしあう原因としては、硝酸根濃度やフリーTBPの計算に考慮されている成分で分配係数式で直接影響しあう場合と化学反応を通して間接に影響しあう場合が考えられる。一方、マイクロ成分とは、他の成分と化学反応を行わず、自分自身の濃度が他の成分の分配係数に影響を及ぼさない成分を示す。

具体例として、ここで4種の成分A, B, C, Dを仮定し、それらの分配係数 D_A , D_B , D_C , D_D , D_E が

$$D_A = f(x_A, x_B, x_C)$$

$$D_B = f(x_B)$$

$$D_C = f(x_A, x_B)$$

$$D_D = f(x_A)$$

$$D_E = f(x_A, x_E)$$

のように表わされるとすると、成分A, B, Cはマクロ成分、成分D, Eはマイクロ成分となる。一般に分配係数式および化学反応速度式は濃度の非線型の関数になるので、分配係数を求める際、マクロ成分については、関係する成分濃度を同時に反復計算して求める必要がある。マイクロ成分については、分配係数が自分自身の濃度に依存することがない場合には、反復計算することなく分配係数を計算することができる。マイクロ成分でも分配係数が自分自身の濃度に依存する場合には、反復計算が必要である。ただし、この場合には1変数のみの反復計算となる。

2. 7 段効率

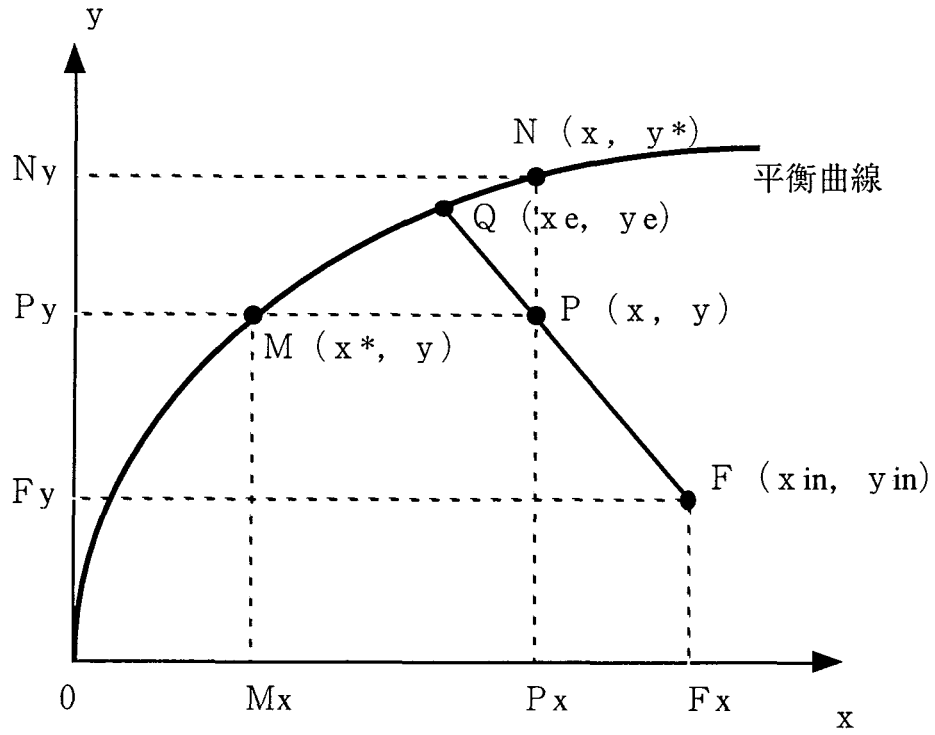
定常計算及び過渡計算において、各ミキサ部での水相と有機相の濃度比は分配平衡の状態からのずれ（段効率）によって表わされる。そしてこのような分配平衡への到達速度は非常に遅いと仮定される。つまり、定常状態であれ、過渡状態であれ、あらゆる時間のミキサ部における水相と有機相の濃度比は分配係数と段効率によって計算されることを意味する。

2. 7. 1 Murphree型段効率

Murphree型段効率 E_i は、水相または有機相のどちらかの濃度によって以下のように定義される。

$$\begin{aligned} \text{水相: } E_i &= \frac{x^i - x_{in}}{x^* - x_{in}} \\ x^* &= \frac{y^i}{D^i} \\ x_{in} &= \frac{W^{i+1} \cdot x^{i+1} + W_{f^i} \cdot x_{f^i}}{W^i} \\ \text{有機相: } E_i &= \frac{y^i - y_{in}}{y^* - y_{in}} \\ y^* &= D^i \cdot x^i \\ y_{in} &= \frac{B^{i-1} \cdot y^{i-1} + B_{f^i} \cdot y_{f^i}}{B^i} \end{aligned}$$

どちらの相濃度について段効率を定義するかにより、濃度分布が異なるので注意を要する。以上の関係を図2-10に示す。このMurphree型段効率がよく使われる概念であるが、MIXSET-Xでは陰解法を用いているため、Murphree型段効率ではなく、次項で示す回収率に段効率をいれた別の段効率を導入している。



水相：
$$E_f = \frac{F_x P_x}{F_x M_x}$$

有機相：
$$E_f = \frac{F_y P_y}{F_y M_y}$$

直線 QPF：
$$y = -\frac{W^i}{B^i} \cdot x^i + \frac{W^i \cdot x_{in} + B^i \cdot y_{in}}{B^i}$$

図2-10 Murphree型段効率の図示

2. 7. 2 回収率型段効率

MIXSET-Xでは、Murphree型段効率の代わりに、回収率（抽出率）に効率の考え方をもち込んだ段効率を使用する。これを「回収率型段効率」と呼ぶことにする。理想的に抽出されるときの回収率 η^* の定義は次のとおりである。

$$\begin{aligned}\eta^* &= \frac{F_o Y^*}{F_a X + F_o Y^*} \\ &= \frac{F_o D X}{F_a X + F_o D X} \\ &= \frac{F_o D}{F_a + F_o D}\end{aligned}$$

- η^* 理想的回収率
 X 水相の成分濃度
 Y^* 有機相の分配平衡成分濃度
 F_a ミキサ部水相から流出する液流量
 F_o ミキサ部有機相から流出する液流量
 D 分配係数

これに段効率 E をもち込んで実際の回収率を η^*E とし、そのときの見かけの分配係数を D' とする。

$$\begin{aligned}\eta^*E &= \frac{F_o Y}{F_a X + F_o Y} \\ &= \frac{F_o D' X}{F_a X + F_o D' X} \\ &= \frac{F_o D'}{F_a + F_o D'}\end{aligned}$$

- η^*E 実際の回収率
 E 段効率
 X 水相の成分濃度
 Y 有機相の実際の成分濃度
 F_a ミキサ部水相から流出する液流量
 F_o ミキサ部有機相から流出する液流量
 D' 見かけの分配係数

このとき次の式が成立する。

$$\frac{F_o D E}{F_a + F_o D} = \frac{F_o D'}{F_a + F_o D'}$$

これを見かけの分配係数 D' について解くと次の式が得られる。

$$D' = \frac{F_a DE}{F_a + F_o D(1-E)}$$

従って、分配係数を見かけの分配係数 D' にする事によって段効率を導入することができる。この見かけの分配係数 D' はミキサ部の濃度のみから計算できるのでプログラム上扱いが容易である。以上は、有機相に抽出されることを想定しているが、水相へ逆抽出されることを想定した段効率の定義も可能であり、次式で逆抽出の場合の見かけの分配係数が計算できる。

$$D' = \frac{F_a(1-E) + F_o D}{F_o E}$$

2. 8 溶媒劣化

2. 8. 1 概要

溶媒劣化物としては、DBP、MBP等が考えられるが、MIXSET-Xにおいては、劣化生成物としてTBPの分解物であるDBPのみを考え、DBPの生成量を計算する機能を組み込んでいる。この計算は、通常の抽出計算が終了し濃度分布が決定された後で行っている。TBPからDBPへの分解の経路としては、化学反応による分解と放射線による分解の二つの経路を考慮する。その際、二つの経路の間に相互作用は無いものとする。また、有機相-水相間の溶解平衡によるTBPの移行は考慮しない。

2. 8. 2 TBPの化学分解

TBPからDBPへの分解は次の化学反応によるものとする。



表2-1に化学反応速度の基礎データ¹⁵を示す。この表より、TBP劣化化学反応の反応速度についての温度依存性のみを考慮し、次式で求める事とした。

$$k = 2.64 \times 10^{-8} \exp(0.0946 \cdot T)$$

k 反応速度定数[1/hr]
T 抽出器の温度[°C]

従って、滞留時間 t [hr]が経過した後の化学分解によるTBP劣化生成物DBPの量 Q₁ [g]は次式で計算される。

$$Q_1 = C_{\text{TBP}} k t V \cdot M_{\text{DBP}}$$

Q₁ 化学分解によるDBP生成量[g]
t 滞留時間[hr]
C_{TBP} TBPモル濃度[mol/l]
V ミキサセトラ各段有機相液量[l]
M_{DBP} DBPの分子量

2. 8. 3 TBPの放射線分解

有機溶媒中のTBPは放射性物質からの放射線によりDBPへ分解される。分解に寄与する放射性物質としては溶媒中にあるU、Pu、Np、Tc、Sr(II)、Zr(V)、Cs(I)、Ce(III)、Gd(III)、Am、Cm、Ruを考慮する。水相中にある上記の核種及びFP核種からの放射線によりTBPが分解されることも考えられるが、MIXSET-Xでは考慮しない。各成分の単位重量あたりの発熱量は入力値として与える必要がある。放射線分解によるDBP生成量は次の手順で計算される。

吸収線量Dの計算

TBPの放射線分解反応G値の計算

分解するTBPのモル分率 f_{TBP}の計算

放射線分解による生成DBP量 Q₂の計算

以下、溶媒の放射線分解についての計算式を個別に記載する。

(1) 吸収線量Dの計算

$$D = f \times F \times \sum_i (q_i C_i) \times t$$

D 吸収線量[w·hr/ℓ]

f 放射線分解反応に寄与する熱出力の補正係数(デフォルトでは1.0)

F 有機相溶媒中のTBPの体積割合(約0.3)

q_i 核種 i の単位重量あたりの発熱量[w/g]

C_i 核種 i の濃度[g/ℓ]

t 滞留時間[hr]

i 核種の添字

(2) TBPの放射線分解反応G値の計算

G値の基礎データを表2-2, 図2-8に示す。データにはDry条件のものとWet条件のものがある。これらの基礎データを処理して

$$G = A \cdot C_{TBP}^4 + B \cdot C_{TBP}^3 + C \cdot C_{TBP}^2 + D \cdot C_{TBP}$$

の形の近似式で表現したときの係数を表2-3に示す。溶媒抽出状態では有機相と水相が接触しているの、溶解平衡により有機相中には飽和量の水が存在している。従って、MIXSET-Xでは、Wet条件のデータに基づく次の近似式を使用する。

$$\begin{aligned} G = & 1.263 \times 10^{-2} C_{TBP}^4 \\ & - 6.972 \times 10^{-2} C_{TBP}^3 \\ & - 1.525 \times 10^{-2} C_{TBP}^2 \\ & + 8.631 \times 10^{-1} C_{TBP} \end{aligned}$$

G TBPの放射線分解反応G値[1/100eV]

C_{TBP} TBP濃度[mol/ℓ]

(3) 分解するTBPのモル分率 f_{TBP} の計算

$$f_{TBP} = 3.73 \times 10^{-4} D G M_{TBP} / \rho$$

f_{TBP} 分解するTBPの初期TBPに対するモル分率

D 吸収線量[w·hr/ℓ]

G TBPの放射線分解G値[1/100eV]

M_{TBP} TBP分子量[266.3g/mol]

ρ TBPの密度[973g/ℓ]

(4) 放射線分解による生成DBP量 Q_2 の計算

$$Q_2 = f_{TBP} C_{TBP} V M_{DBP}$$

Q_2 放射線分解によるDBP生成量[g]

f_{TBP} 分解するTBPの初期TBPに対するモル分率

C_{TBP} TBP濃度[mol/ℓ]

V ミキサセトラ各段有機相液量[ℓ]

M_{DBP} DBP分子量[210.2g/mol]

2. 8. 4 生成DBP量 Q_{DBP}

有機溶媒中のTBPが化学反応分解または放射線分解により分解されて生成されるDBPは次式で計算される。

$$Q_{DBP} = (Q_1 + Q_2) / t$$

Q_{TBP} DBP生成量[g/hr]
 Q_1 化学分解によるDBP生成量[g]
 Q_2 放射線分解によるDBP生成量[g]
 t 滞留時間[hr]

図2-11にTBP放射線分解G値のプロットを示す。表2-1に化学溶媒劣化の基礎データを表2-2にTBP放射線分解G値の基礎データを示す。G値とは、吸収線量100ev当りに分解反応を起こす分子の数である。

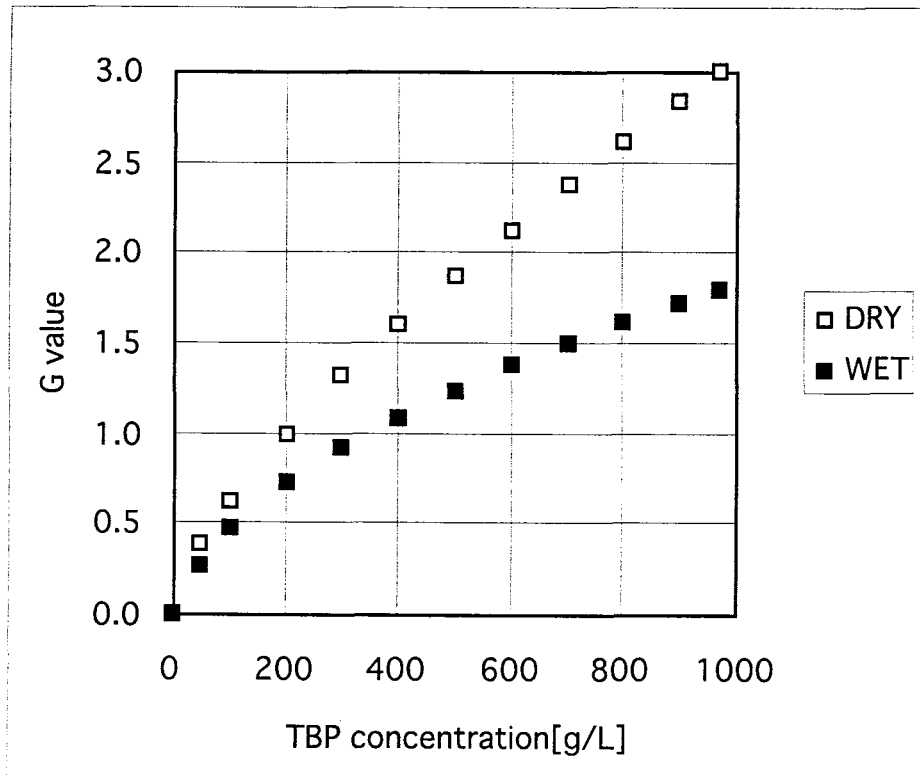


図2-11 TBP放射線分解G値

表 2 - 1 TBP化学分解反応速度定数⁽⁵⁾

水相	有機相		温度	速度定数
HN03濃度 [M]	%TBP [%]	HN03濃度 [M]	[°C]	[/h]
0.5	30	0.066	100	2E-04
1	30	0.185	23	3E-07
		0.18	26	1E-07
			40	7E-07
		0.17	70	2E-05
		0.185	100	3E-04
2.5	30	0.5	100	7E-04
3	30	0.62	70	3E-05
5	30	0.86	26	1E-07
			40	7E-07
			70	4E-05
		0.9	100	0.001
6	30	1.1	70	0.001
7.2	30	1.2	70	2E-05

出典：CRC-TBP-Hand Book p.142, Table 3

有機相中のTBP濃度が30v/o条件での測定結果をまとめなおしたものである。

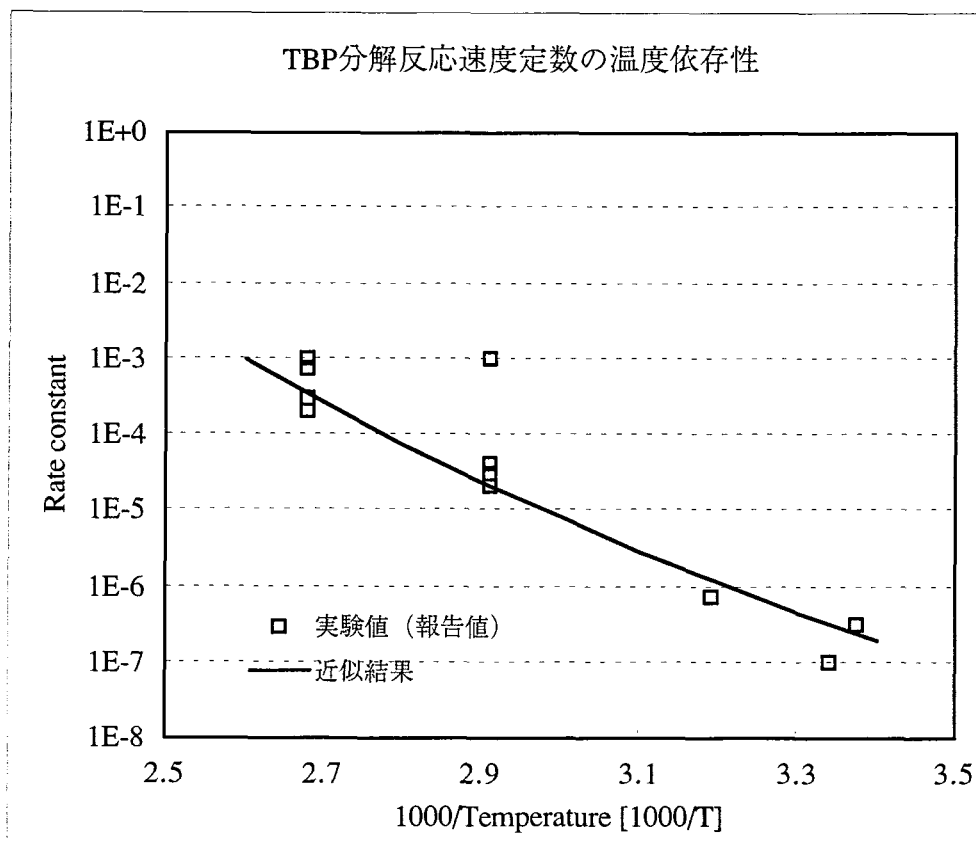


図 2 - 1 2 TBP化学分解反応速度定数近似式

表 2 - 2 TBP放射線分解反応G値¹⁵

TBP concentration [g/l]	G(total acid)	
	Dry	Wet
0	0	0
50	0.38	0.27
100	0.62	0.47
200	1	0.73
300	1.32	0.92
400	1.61	1.09
500	1.87	1.24
600	2.13	1.38
700	2.37	1.5
800	2.61	1.62
900	2.83	1.73
973	3	1.8

出典：CRC TBP-Book p.249, Table 9

表 2 - 3 TBP放射線分解反応G値のTBP濃度依存性の近似式の係数

	DRY	WET
A	-2.3E-02	1.263E-02
B	2.1E-01	-6.972E-02
C	-7.3E-01	-1.525E-02
D	1.8E+00	8.631E-01

$$G = A \cdot C_{\text{TBP}}^4 + B \cdot C_{\text{TBP}}^3 + C \cdot C_{\text{TBP}}^2 + D \cdot C_{\text{TBP}}$$

C_{TBP} TBP濃度 [mol/l]

3 数値解法

3.1 基本方針

3.1.1 概要

MIXSET-Xにおいては、計算条件によらず統一した段効率で取り扱う事を念頭においた。そのためには、化学反応の有る成分についても統一した段効率が適用できるように数値解法を変更する必要があった。従来のMIXSETで化学反応の有る成分についてMurphree型段効率を使用することはできないのは、化学反応の有る場合と化学反応の無い場合で、数値解法が全く別であり、それぞれの場合に応じてサブルーチンが全く独立していることが大きな原因であった。段効率の方式を統一して計算を行うには、化学反応の有無にかかわらず同一の数値解法で計算することが重要である。また、従来のMIXSETの数値計算手法では逐次代入法あるいは緩和法による繰り返し計算を行っているが、コーディング方法が従来のMIXSETで対象としているマクロ成分の化学反応に密接に依存したものにもなっている。このままでは、新たに化学反応を加えて段効率を全ての成分で統一的に扱うことが不可能であり、プログラムを全面的に変更する必要があることがわかった。プログラムの全面改訂に際しては従来のMIXSETで問題となっていた計算の収束性の改善とプログラムのメンテナンス性の改善に重点を置くこととした。そこで、MIXSET-Xでは、方程式の差分スキームに陰解法を採用し、差分化された方程式をニュートン法を用いて解く方法を採用し、段効率についても従来のMurphree型段効率よりもプログラム上扱いやすい回収率段効率を採用した。以下に、MIXSET-Xの数値解法を説明する。

3.1.2 分配係数の計算

マクロ成分に Pu^{6+} 、 $\text{Np}(V)$ 、 $\text{Np}(IV)$ 、 $\text{Np}(III)$ 、 Tc^{7+} 、 Zr を追加して分配係数を計算するためには、分配係数の計算手法を変更する必要がある。

従来のMIXSETのマクロ成分の分配係数は成分濃度の他に変数として T_f （フリーTBP）と X_{NO_3} （硝酸根濃度）のみを持ち、この二つの変数の両方ともに水相の成分濃度から計算できる量で、 X_{NO_3} は水相成分濃度の重み付きの和として簡単に計算でき、 T_f も水相成分濃度から計算される係数を持つ二次関数を解くことにより簡単に計算できた。マクロ成分の分配係数は水相成分濃度から簡単に求めることができるため、収束計算の必要はなかった。

ところが、新たに追加したマクロ成分のうち、 Tc^{7+} は分配係数が有機相のウラン濃度（間接的に T_f ）の対数関数になっており、また、 Pu^{6+} 、 $\text{Np}(V)$ の分配係数の計算式は表を内挿する形になっており、その中に有機相のウラン濃度（間接的に T_f ）が使用されているため、もはや T_f は単純な二次関数を解くことでは計算することができない。そこで、MIXSET-Xでは、分配係数の計算において収束計算を行う必要があるが、この計算手法としては安定性を考慮しニュートン法を使用する。その際の初期値としては、従来のマクロ成分のみを考えたときの T_f を採用する。従ってこの初期値計算部分については従来のプログラムを利用することになる。 T_f を与える関数を $f(T_f)$ とすると、ニュートン法による収束計算手順は次の式で与えられる。ただし、 n は計算繰り返し回数であり、 f' は f の T_f による一次微分である。

$$T_f^{n+1} = T_f^n - f'(T_f^n)^{-1} \cdot f(T_f^n)$$

3.1.3 化学反応計算

従来のMIXSETでは化学反応計算の積分を陰的台形公式で行っている。この方法は安定に計算でき問題も少ないが、化学反応速度定数が非常に大きい場合には、過渡計算の計算刻み時間を非常に小さくしないと負の濃度が計算されるという問題がある。そこで、今回の改訂では、やや計算精度が劣るが負の濃度が計算されることのない後退オイラー法を採用する。後退オイラー法の計算手順により、プログラムのメンテナンス性も改善される。ただし、陰的台形公式の場合と同様に一次の化学反応でない場合には（ほとんどの化学反応は一次の化学反応ではない）非線形方程式を解くことになりニュートン法で収束計算をすることになる。

陰的台形公式と後退オイラー法の違いを簡単な一次の反応の場合を例にとって示す。一次の反応の場合に解くべき微分方程式は次のようになる。ただし、 X は成分濃度、 X' はその時間微分であり、 k は正の反応速度定数とする。

$$X' = -kX$$

陰的台形公式による時間積分公式は次の式で与えられる。ただし、 n は計算繰り返し回数であり、 Δt は計算刻み時間である。

$$X^{n+1} = X^n - k (X^{n+1} + X^n) \Delta t / 2$$

従って計算手順は次の式で与えられる。

$$X^{n+1} = \left(\frac{1 - k \Delta t / 2}{1 + k \Delta t / 2} \right) \cdot X^n$$

この式でわかるように、 X^n は等比数列となり、比の分子の絶対値は分母の絶対値より常に小さいため数列は発散することなく安定に計算される。しかし、計算刻み時間幅 Δt または化学反応速度 k が大きく分子が負になる場合には安定に計算できるものの計算される成分濃度は負の値と正の値を繰り返すことになる。計算される成分濃度が負にならないようにするためには次の条件が必要である。

$$k \Delta t < 2$$

化学反応が速く反応速度 k が非常に大きい場合にこの条件を満足しようとする、計算刻み時間 Δt を非常に小さくとらなければならない、過渡計算の計算回数が著しく増加し、計算時間が増大することになる。

一方、後退オイラー法による時間積分公式は次の式で与えられる。ただし、 n は計算繰り返し回数であり、 Δt は計算刻み時間である。

$$X^{n+1} = X^n - k X^{n+1} \Delta t$$

従って計算手順は次の式で与えられる。

$$X^{n+1} = \left(1 / (1 + k \Delta t) \right) \cdot X^n$$

この式からわかるように、陰的台形公式の場合と同様に X^n は等比数列となり、比の分子の絶対値は分母の絶対値より常に小さいため数列は発散することなく安定に計算される。また、比は Δt の大きさによらず常に正であるため負の成分濃度が計算されることもない。従って、計算精度をさほど必要としない場合には、非常に大きな計算刻み時間で過渡計算の回数を少なくして短時間で計算することも可能である。

以上、流れのない非常に簡単な例を取り上げて説明した。実際の抽出器においては当然の事ながら流れが存在し、化学反応も一次の反応ではないため、後退オイラー法を使用することだけでは、必ずしも負の成分濃度が計算されること無く安定に計算されるという保証はないが、他の方法に比べて安定的な計算が期待できる。従って、今回の改訂に当たっては、化学反応計算に後退オイラー法を採用することにした。後退オイラー法は陰的解法であり、化学反応計算は非線形方程式を解くことになる。MIXSET-Xにおいてはこれをニュートン法で解いている。

3. 2 過渡計算手法

3. 2. 1 抽出器の物質収支基本式

ミキサセトラ型抽出器における物質収支基本式は、各段各成分毎に次のようになる。簡単のため成分の添え字を省略する。

ミキサ部

$$\begin{aligned}
 V_{mai} \frac{dX_{mi}}{dt} + V_{moi} \frac{dY_{mi}}{dt} &= W_{i+1}X_{si+1} + B_{i-1}Y_{si-1} + W_{Ri}X_{si} \\
 &\quad - (W_{i+1} + W_{Ri} + W_{fi})X_{mi} - (B_{i-1} + B_{fi})Y_{mi} \\
 &\quad + W_{fi}X_{fi} + B_{fi}Y_{fi} \\
 &\quad + V_{mai}R_{mai} + V_{moi}R_{moi}
 \end{aligned}$$

セトラ部（水相）

$$V_{sai} \frac{dX_{si}}{dt} = (W_{i+1} + W_{Ri} + W_{fi})X_{mi} - (W_i + W_{Ri})X_{si} + V_{sai}R_{sai}$$

セトラ部（有機相）

$$V_{soi} \frac{dY_{si}}{dt} = (B_{i-1} + B_{fi})Y_{mi} - B_i Y_{si} + V_{soi}R_{soi}$$

液量収支（水相）

$$W_i = W_{i+1} + W_{fi}$$

液量収支（有機相）

$$B_i = B_{i-1} + B_{fi}$$

t	時間[h]
i	段を示す添え字
X _{mi}	i 段ミキサ部水相成分濃度[mol/ℓ]
Y _{mi}	i 段ミキサ部有機相成分濃度[mol/ℓ]
X _{si}	i 段セトラ部水相成分濃度[mol/ℓ]
Y _{si}	i 段セトラ部有機相成分濃度[mol/ℓ]
V _{mai}	i 段ミキサ部水相液量[ℓ]
V _{moi}	i 段セトラ部有機相液量[ℓ]
V _{sai}	i 段セトラ部水相液量[ℓ]
V _{soi}	i 段セトラ部有機相液量[ℓ]
R _{mai}	i 段ミキサ部水相化学反応による成分生成速度[mol/ℓh]
R _{moi}	i 段ミキサ部有機相化学反応による成分生成速度[mol/ℓh]
R _{sai}	i 段セトラ部水相化学反応による成分生成速度[mol/ℓh]
R _{soi}	i 段セトラ部有機相化学反応による成分生成速度[mol/ℓh]
W _i	i 段ミキサ部より流出する水相液流量[ℓ/h]
W _{fi}	i 段ミキサ部に外部より供給される水相液流量[ℓ/h]
W _{Ri}	i 段ミキサ部にセトラ部より循環する水相液流量[ℓ/h]
B _i	i 段ミキサ部より流出する有機相液流量[ℓ/h]
B _{fi}	i 段ミキサ部に外部より供給される有機相液流量[ℓ/h]

ここで、液量は成分濃度に依存しないものとしている。また、化学反応による成分生成速度は各成分の関数であり、成分が生成されるときは正の値、消滅するときには負の値をとるものとする。

また、貯槽における物質収支式は次のようになる。

貯槽水相の物質収支式

$$V \frac{d x}{d t} = W_{IN} \cdot x_{IN} - W_{OUT} \cdot x$$

貯槽有機相の物質収支式

$$V \frac{d y}{d t} = B_{IN} \cdot y_{IN} - B_{OUT} \cdot y$$

x_{IN}	水相フィード液成分濃度[mol/ℓ]
y_{OUT}	有機相フィード液成分濃度[mol/ℓ]
x	貯槽での水相成分濃度[mol/ℓ]
y	貯槽での有機相成分濃度[mol/ℓ]
W_{IN}	水相フィード側水相液総流量[ℓ/h]
W_{OUT}	水相貯槽出口総流量[ℓ/h]
B_{IN}	有機相フィード側有機相液総流量[ℓ/h]
B_{OUT}	有機相貯槽出口総流量[ℓ/h]

3. 2. 2 分配平衡式及び段効率式

ミキサセトラ型抽出器においては物質収支基本式だけでは、方程式系が不完全であり解くことができない。そのために分配平衡式が必要になる。段効率を考慮する場合には、さらに段効率定義式が必要である。物質収支式を解くために必要な分配平衡式および段効率定義式は、各段各成分毎に次のようになる。簡単のため成分の添え字を省略する。

分配平衡式

分配係数をDとして次のようになる。

$$Y_{mi} = Y_m^* i = D X_{mi} \quad (A)$$

または

$$\frac{Y_{mi}}{D} = X_m^* i = X_{mi} \quad (B)$$

段効率を入れる場合には、分配係数の代わりに、ミキサ部の回収率型段効率をEとしたときに次の式で計算される見かけの分配係数D'を使用する。

有機相抽出での定義式

$$D' = \frac{F_a D E}{F_a + F \cdot D (1 - E)} \quad (C)$$

または、

水相逆抽出での定義式

$$D' = \frac{F_a (1 - E) + F \cdot D}{F \cdot E} \quad (D)$$

E	段効率
X	水相の成分濃度
Y	有機相の実際の成分濃度
F _a	ミキサ部水相から流出する液流量
F _o	ミキサ部有機相から流出する液流量
D'	見かけの分配係数

方程式系を完全にするためには (A) と (B) のうちどれかひとつを物質収支基本式と組み合わせればよい。

3. 2. 3 差分手法

ミキサセトラ型抽出器における物質収支基本式と分配平衡式または段効率定義式を連立させて解くことにより、過渡状態の各段成分濃度を計算することができる。差分手法は、後退オイラー法を化学反応項および移流項部分に採用する。この解法は陰解法であるため、ミキサ部またはセトラ部での液滞留時間より計算刻み時間を大きくとって計算することが可であり、結果として少ない計算時間で計算できる可能性がある。移流項部分について陽解法を採用することもできるが、その場合には計算刻み時間はミキサ部またはセトラ部での液滞留時間の最小値よりも大きくすることはできない。従来の MIXSET は陽解法に近いものである。MIXSET-X の差分式は、物質収支基本式と分配平衡式または段効率定義式の組み合わせにおいて、時間微分項を次のように離散化近似したものとなる。ただし、各式においてその他の項は $t + \Delta t$ での時間の物理量とする。

$$\frac{d X_{mi}}{d t} = \frac{X_{mi}(t + \Delta t) - X_{mi}(t)}{\Delta t}$$

$$\frac{d Y_{mi}}{d t} = \frac{Y_{mi}(t + \Delta t) - Y_{mi}(t)}{\Delta t}$$

$$\frac{d X_{si}}{d t} = \frac{X_{si}(t + \Delta t) - X_{si}(t)}{\Delta t}$$

$$\frac{d Y_{si}}{d t} = \frac{Y_{si}(t + \Delta t) - Y_{si}(t)}{\Delta t}$$

定常状態の方程式は以上の時間微分項を 0 にしたものとなる。

以上の作業で差分方程式の完全な組み合わせが得られた。分配係数 D 及び化学反応による成分生成速度が、各成分の非線形関数であるためこの差分方程式を解くためには収束計算が必要になってくる。MIXSET-X においては差分式をニュートン法により解くことにする。

3. 2. 4 ニュートン法による繰り返し計算手順

一般に n 個の変数 x_i と n 個の関数 $f_i(x_1, x_2, \dots, x_n)$ があるとき、

$$f_i(x_1, x_2, \dots, x_n) = 0 \quad (i = 1, 2, \dots, n)$$

を満足させるような x_i は、次の反復計算を行うことにより数値計算で解を求めることが可能である。ただし、初期値は適切に選ぶものとする。

$$J(x^k) \delta x^k = f(x^k) \quad (E)$$

$$x_i^{k+1} = x_i^k - \delta x_i^k$$

ここで、 k は反復回数、 δx^k は未知の列ベクトル、 x^k は x_i を成分とする列ベクトル、 $f(x^k)$ は関数値 $f_i(x^k)$ を成分とする列ベクトル、 J は行列成分が次の式で定義されるヤコビアンである。

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

関数 f_i を抽出器の物質収支基本式及び分配平衡式または段効率定義式とすると、式(D)に対応する次の式が得られる。ただし、 i は段を示す添え字である。

ミキサ部の物質収支基本式より

$$a \delta Y_{s_{i-1}}^k + b \delta Y_{m_i}^k + c \delta X_{m_i}^k + d \delta X_{s_i}^k + e \delta X_{s_{i+1}}^k = g$$

セトラ部の水相物質収支基本式より

$$\delta X_{m_i}^k = a' \delta X_{s_i}^k + b'$$

セトラ部の有機相物質収支基本式より

$$\delta Y_{m_i}^k = c' \delta Y_{s_i}^k + d'$$

分配平衡式より

$$e' \delta Y_{m_i}^k + g' \delta X_{m_i}^k = 0 \quad (A')$$

水相で定義された段効率式より

$$a_1 \delta X_{m_i}^k + a_2 \delta Y_{m_i}^k + a_3 \delta X_{s_i}^k + a_4 \delta X_{s_{i+1}}^k = b_1 \quad (B')$$

有機相で定義された段効率式より

$$a_5 \delta X_{m_i}^k + a_6 \delta Y_{m_i}^k + a_7 \delta Y_{s_i}^k + a_8 \delta Y_{s_{i-1}}^k = b_2 \quad (C')$$

ここで、式(A'), (B'), (C')は各成分についてどれか一つが使用されるものとする。抽出計算のニュートン法による数値解法は最終的には $\delta Y_{m_i}^k$, $\delta X_{m_i}^k$, $\delta Y_{s_i}^k$, $\delta X_{s_i}^k$ についての上記の連立一次方程式を解くことに帰着する。ここで、 $\delta x_i^k = (\delta Y_{m_i}^k, \delta X_{m_i}^k)$ を一つの組と考え、さらに $\delta x_{i+1}^k = (\delta Y_{s_i}^k, \delta X_{s_{i+1}}^k)$ を一つの組と考える。このように考えて、上記の連立一次方程式を δx_i^k に関する一次連立方程式としてみると、式の中に同時に現れる段の添え字は、 $i-1$, i , $i+1$ の隣り合ったもののみであることがわかる。このことは、 δx_i^k に対するヤコビアンがブロック三重対角行列であることを意味している。従って、成分数を N 、段数を M とした抽出計算の行列計算を行う場合にフル行列の計算を行う必要は無く、 $2N \times 2N$ の小行列から構成されるブロック三重対角行列用の計算方法を使用して比較的短い計算時間で計算することができる。

3. 3 定常計算手法

3. 3. 1 現状のMIXSET-Xによる定常値の計算

現状では定常計算の機能は組み込まれていないので、後述する試計算では5000時間という長時間の過渡計算を行うことによって定常状態の計算結果を得ている。MIXSET-Xでは、きわめて長い計算時間を指定した場合でも、同じ計算対象ではCPU時間の大きな差は生じない。これは、MIXSET-Xでは方程式の差分スキームに陰解法を採用するとともに、定常状態に近づき少ない回数で計算が収束するような状態ではタイムステップを非常に大きくとる事ができるようにプログラムしているためである。このタイムステップの自動調節については4章の入力仕様で、TAULIMカードの説明を参照されたい。

3. 3. 2 定常計算を組み込む場合の方針

(a) 基本的な考え方

定常計算を組み込む場合の基組み込みの方針を記述しておく。定常計算の物質収支基本式は、過渡計算の物質収支基本式の濃度の時間微分項を0としたものである。その他の分配平衡式や段効率式は過渡計算のものと同一である。化学反応項のために、方程式は非線形の連立方程式となり、過渡計算の場合と同様にしてニュートン法により解くことができる。必要な行列要素は、過渡計算の場合の計算式で、濃度の時間微分から派生する部分を0にすることにより計算することができる。その後、ブロック三重対角行列にして解く部分は過渡計算の場合と全く同じである。

(b) 初期値の設定方法

定常計算の場合は、ニュートン法の初期値をどのように設定するかが非常に重要である。初期値が適切でない場合には解が収束しないか、数値解が真の解と全く異なる場所へ収束してしまう可能性がある。そこで、定常計算の場合には次の手順を踏んで計算することを考える。

- (1) ウラン・プルトニウム等の成分を入れない酸-溶媒の定常状態を計算する。この場合の初期値は、硝酸濃度一定の分布とする。
- (2) ウラニルを加えて定常状態を計算する。
- (3) 4価のプルトニウムを加えて定常状態を計算する
- (4) ウラナス, 3価のプルトニウム, HAN, ヒドラジン, 亜硝酸を加えて定常状態を計算する。
- (5) 全ての成分を加えて定常状態を計算する。

この方法で計算可能かどうかは未確認であり、この方法以外にも、上記の各段階で過渡計算を何ステップか行い、そこを初期値としてニュートン法の計算を行うことや、減速ニュートン法にすることなどが考えられる。

4 MIXSET-X入力説明

4.1 計算コードの制限

入力データのサイズに関する制限は次の通りである。

NUNIT	≦	19	(抽出器の数+貯槽の数, 抽出器の数の最大は9)
MSTG	≦	25	(各バンクのステージ数)
MICCOMP	≦	9	(追加マイクロ成分の数)
NFA	≦	30	(水相供給液の数)
NFO	≦	30	(水相供給液の数)
NXTABL	≦	20	(供給液の時間変化の数)
NPRT	≦	99	(PRINT時刻指定の数)
NPLT	≦	99	(PLOT時刻指定の数)
NTAUC	≦	10	(過渡計算刻み時間の一時的制限指定の数)

入力データは、表4-1に示すカードで入力するが、プログラムのデータ構造のため、入力するカードの関係において以下のような制約がある。

- (1) TITLEカードを第1行、CONTROLカードを2行目に入力し、各バンクのデータ入力の最初にSTAGEカードを入力しなければならない。他のほとんどのカードは、CONTROLカード及び、直前のSTAGEカードで入力したステージ数、または成分数に相当する数のデータを入力する必要がある。
- (2) 分配係数の計算方法は、各バンク、各成分ごとに指定できる。その指定はSDISTカードによって行う。これらのカードで計算方法の指定を行わないバンク、成分については、プログラム内蔵の計算式に従って計算される。ただし、SDISTカードにより定数の平衡式または定数の分配係数を使用するよう指定した成分では、指定されている方法によって計算される。
- (3) 追加マイクロ成分の取扱いにおいては化学反応は考慮できない。
- (4) MIXSET-Xで標準的に取り扱うことのできる成分は次の31成分である。
 $\text{HNO}_3, \text{U(VI)}, \text{Pu(IV)}, \text{Pu(III)}, \text{U(IV)}, \text{HNO}_2, \text{HYD}, \text{HAN}, \text{Pu(V)}, \text{Pu(VI)}, \text{Np(IV)}, \text{Np(V)}, \text{Np(VI)},$
 $\text{Zr(IV)}, \text{Tc(IV)}, \text{Tc(V)}, \text{Tc(VI)}, \text{Tc(VII)}, \text{UO}_2(\text{NO}_3)(\text{TcO}_4), \text{Pu}(\text{NO}_3)_3(\text{TcO}_4), \text{Zr}(\text{NO}_3)_3(\text{TcO}_4),$
 $\text{Sr(II)}, \text{Ru(DI)}, \text{Ru(TR)}, \text{Ru(NI)}, \text{Ru(MO)}, \text{Cs(I)}, \text{Ce(III)}, \text{Gd(III)}, \text{Am(III)}, \text{Cm(III)}$
 以上の内、 $\text{Sr(II)}, \text{Ru(DI)}, \text{Ru(TR)}, \text{Ru(NI)}, \text{Ru(MO)}, \text{Cs(I)}, \text{Ce(III)}, \text{Gd(III)}, \text{Am(III)}, \text{Cm(III)}$ はマイクロ成分である。
- (5) 同一のバンク内においては、定義された段効率は全て、水相への回収率で定義されるか、有機相への回収率で定義されるかのどちらか一方に制限されている。成分毎に定義方法を指定することはできない。
- (6) 過渡計算の計算刻み時間は全てのバンクで共通に使用され、従来のMIXSETのようにバンク毎に異なった計算刻み時間とすることはできない。これは、フィードバック計算を可能にするために設けられた制限である。

4. 2 入力データおよびフォーマット

4. 2. 1 フリーフォーマット入力

M I X S E Tコードの入力はすべて、FORTRANの標準的なリスト形式READ文（READ(5,*)...の形式のREAD文）によるフリーフォーマット入力である。ただし、入力カード上にコメントを記述できるようにするため、入力データファイルの前処理ルーチン（RDCARD）を用いている。

入力カードはT I T L Eカードを除いて、キーワード、データ、区切り記号と終端記号の4つから成っている。

- キーワード： 入力カードの種類を表わし、各カードの最初に入力されていなければならない。キーワードが5文字以上の時、空白も含み最初の4文字のみが意味をもつ。また、キーワードの次にくるデータとの区切りは、必ず空白文字' 'で区切らなければならない。
- データ： 入力データ。区切り記号によって区切られ、0~9, E+, E-, E, +, -, .の文字からなる数値及び*。
*は、例えば「5*2.0」のように入力し、2.0を5回繰り返し入力した意味となる（従来のM I X S E Tの「2.0R5」という入力に相当する）。
- 区切り記号： データとデータの区切り。空白' ', カンマ', 'のどちらかである。空白の区切り記号は一つ以上続いて、一つだけの空白があるものと見なされる。また、カンマの前、又は後に続く空白は無視される。
（注：カンマが二つ以上続くと、カンマとカンマに挟まれた部分に対応するデータは入力されなかったものとされ、プログラム内部での値が保たれる）
- 終端記号： あるキーワードについてのデータの入力を打ち切るための記号'/'であり、この記号が読み込まれると、入力したデータの数がプログラムで要求される数より少なくともデータの読み込みは打ち切られ、残りのデータは入力されなかったものとされ、プログラム内部での値が保たれる。

カラム1~72までがキーワード、データ、区切り記号、終端記号の領域であり、73カラム以降はコメント領域である。また、次に示す様にいくつかの文字は特別な意味を持つ。T I T L Eにこれらの文字を使った場合の動作は確認していない。

第一カラムに'*'または'#'のあるカードはコメントカードであり入力しない場合と同じである。第一カラムに'@'のあるカードは次のような特別な意味を持つ。カード自体は入力されなかったものと見なされる。

@LISTOFF 入力データのプリントを抑制する。

@LISTON 抑制されている入力データのプリントを再開する。

入力データの途中にある!*...!*の形式の記述はコメントと見なされ、ブランクを入力したものと見なされる。このコメントは複数行にまたがることができる。

4. 3 入力仕様

4. 3. 1 入力キーワードの説明

表4-1に、入力カードの項目を示す。この中でTITLEカードは1行目に入力し、CONTROLカード、ADDMICカードは他のカードより先に入力されなければならない。また、STAGEカードはそのカードで定義される抽出器についてのFEEDカード、INITIALカード、VOLUMEカード等その抽出器についてのデータを入力するカードより前に入力される必要がある。ENDカード以降のカードは無視される。フリーフォーマットになっているので、必要な項目についてのみ入力すればよく、入力の順序も自由である。重複するカードについてはFEEDカードでは合計され、その他では後に入力されたものが有効となる。このことは分配係数入力の際に特に注意を要する。

表4-1 入力カード一覧

No.	カード名 (Keyword)	内 容	入力の 必要性	デフォルト値 の有無	ページ
1	TITLE	計算ケースの表題	◎		34
2	TBP	TBP分率		有	34
3	DBP	DBP量		有	34
4	DEGRADATION	溶媒の放射線劣化割合の強度		有	34
5	ADDMIC	追加マイクロ成分の名前等			34
6	CHARGE	電荷		有	35
7	DHEAT	成分の崩壊熱		有	36
8	CONTROL	計算制御(計算終了時その他)	◎		36
9	STAGE	抽出器の定義	◎		37
10	VOLUME	ミキサ部とセトラ部の容積	◎		37
11	HEIGHT	セトラ部の界面レベル		有	37
12	RECYCLE	水相リサイクル流量		有	38
13	TOFSTG	各段の温度分布の指定		有	38
14	TANK	貯槽の定義			38
15	INITIAL	初期濃度	◎		39
16	FEED	供給液流量と成分濃度	◎		40
17	SDIST	分配係数の計算式の選択		有	42
18	EQUILIBRIUM	平衡定数	○		43
19	CDIST	分配係数(一定値)	○		43
20	EFFICIENCY	段効率			43
21	REACTION	化学反応速度の調整		有	43
22	EPSILON	反復計算の収束判定値		有	46
23	ITERATION	反復計算の回数の制御		有	46
24	TAULIM	過渡計算刻み時間		有	46
25	TAUCONST	過渡計算刻み時間の一時的制限		有	46
26	PRINT	プリント出力の時刻指定		有	47
27	PLOT	プロット用出力の時刻指定		有	47
28	LDIST	分配係数のプリント			47
29	FILE	計算出力のファイル名の指定			47
30	END	入力を終了して計算を開始	◎		47

注) 入力の必要性 ◎……………必ず入力すること

○……………SDISTカードでの指定内容によっては必ず入力すること

4. 3. 2 入力カード説明

Card No. 1 T I T L Eカード (必ず1行目に入力すること)

```

TITLE
    
```

TITLE = 計算ケースの表題。72文字まで入力可能
カードの先頭の文字が '*' や '#' でない最初のカードはTITLEカードになる。

Card No. 2 T B Pカード (省略可能)

```

TBP          FRC
    
```

FRC = TBPの体積分率 (デフォルト=0.3)
TBPのモル分率 CTBP=3.653774×FRC

Card No. 3 D B Pカード (省略可能)

```

DBP          CDBP
    
```

CDBP = 分配係数の計算に使われるDBPの濃度[g/l] (デフォルト=0.0[g/l])

Card No. 4 D E G R A D A T I O Nカード (省略可能)

```

DEGRADATION  FDEG
    
```

FDEG = 有機溶媒中のTBPの放射線劣化強度の調節因子 (デフォルト=1.0)
このカードは2.8節で述べた有機溶媒中の吸収線量Dの計算式でfに相当するものである。

$$D = f \times F \times \sum_i (q_i C_i) \times t$$

D 吸収線量[w·hr/l], F 有機相溶媒中のTBPの体積割合(約0.3)
q_i 核種iの単位重量あたりの発熱量[w/g], C_i 核種iの濃度[g/l]
t 滞留時間[hr], i 核種の添字

Card No. 5 A D D M I Cカード (追加マイクロ成分の計算のときに入力)

```

ADDMIC      J,CNAME(J),CHRG(J),CNTBP(J),CVUNIT(J)
    
```

J = 追加マイクロ番号 (J=32~40)
CNAME(J) = 追加マイクロ成分Jの名前 (8文字以内)
CHRG(J) = 追加マイクロ成分Jの水相中電荷
CNTBP(J) = 追加マイクロ成分Jの有機相中TBP配位数
CVUNIT(J) = 濃度単位を[g/l]から[mol/l]へ変換するための変換係数[g/mol]
このカードはFEEDSカードおよびINITIALカードの前に入力する必要がある。

Card No. 6 CHARGEカード (省略可能)

CHARGE	J, CHRG(J), CNTBP(J)
--------	----------------------

J = 成分番号(1から31までの数)
 CHRG(J) = 成分Jの水相中電荷
 CNTBP(J) = 成分Jの有機相中TBP配位数

表4-2 CHARGEカード デフォルト値

J	成分	CHRG	CNTBP
1	HNO ₃	1.0	1.0
2	UO ₂ (NO ₃) ₂	2.0	2.0
3	Pu(NO ₃) ₄	4.0	2.0
4	Pu(NO ₃) ₃	3.0	2.0
5	U(NO ₃) ₄	4.0	2.0
6	HNO ₂	0.0	0.0
7	N ₂ H ₅ NO ₃	1.0	0.0
8	NH ₃ OHNO ₃	1.0	0.0
9	Pu(V)	1.0	1.0
10	Pu(VI)	2.0	2.0
11	Np(IV)	4.0	2.0
12	Np(V)	1.0	1.0
13	Np(VI)	2.0	2.0
14	Zr(IV)	4.0	2.0
15	Tc(IV)	0.0	0.0
16	Tc(V)	0.0	0.0
17	Tc(VI)	0.0	0.0
18	Tc(VII)	0.0	2.0
19	Tc-U(VI)	0.0	2.0
20	Tc-Pu(IV)	0.0	2.0
21	Tc-Zr(IV)	0.0	2.0
22	Sr(II)	2.0	2.0
23	Ru(DI)	2.0	2.0
24	Ru(TR)	3.0	2.0
25	Ru(NI)	0.0	2.0
26	Ru(MO)	0.0	0.0
27	Cs(I)	1.0	1.0
28	Ce(III)	3.0	3.0
29	Gd(III)	3.0	3.0
30	Am(III)	3.0	3.0
31	Cm(III)	3.0	3.0

HAN : NH₃OH⁺
 HYD : N₂H₅⁺
 Ru(DI) : Ruジニトラ錯体
 Ru(TR) : Ruトリニトラ錯体
 Ru(NI) : Ruジニトロ錯体
 Ru(MO) : Ruモノニトラ錯体

分配係数計算に使用される全硝酸根濃度 [NO₃⁻] は、次の式で計算される。
 (Tc共抽出物以降の成分は、表4-2の電荷数を使用しない)

$$[\text{NO}_3^-] = \sum_{j=1}^{18} \text{CHRG}(J) \cdot [X_j]$$

- 注) ・TBP配位数の指定は、プログラム内蔵の分配係数式を使用する場合は無視される。
 ・SDISTカードでIDIST=2(分配係数を入力)とした場合、TBP配位数の指定は無視されるが、マクロ成分については、フリーTBP濃度の計算を介し、分配係数値に影響する。

Card No. 7 D H E A Tカード (省略可能)

DHEAT

J, DECAYH(J)

J = 成分番号(1から40までの数)

DECAYH(J) = 成分Jの崩壊熱[w/g] (デフォルト=0.0 [w/g])

このカードで崩壊熱を与えないと化学反応のみによるDBP生成量が計算されることになる。

また、DHEATカードを使用する場合、成分番号19～21のTcの共抽出物の崩壊熱の値も入力できるが、その値は無視され、共抽出物を生成する各々の成分の崩壊熱の入力値から算出される。

Card No. 8 C O N T R O Lカード (必ず入力すること)

CONTROL

ICALC, ISF, ENDTIM, DTO

ICALC = 計算オプション

0 過渡計算のみ

>0 定常計算のみ

<0 定常計算後、過渡計算を実行

現在のところ計算できるのは過渡計算のみである。

ISF = 溶液濃度オプション

0 : Solute free濃度 (中性基準濃度: 付録I参照)。

1 : 通常濃度近似

ENDTIM = 過渡計算時の計算終了時間(Hour)

DTO = 過渡計算で使用される計算刻み時間の初期値(Hour)

注) ・M I X S E T - Xで標準的に取り扱うことできる成分は以下の通りである。

HNO₃, U(VI), Pu(IV), Pu(III), U(IV), HNO₂, HYD, HAN, Pu(V), Pu(VI), Np(IV), Np(V), Np(VI),
Zr(IV), Tc(IV), Tc(V), Tc(VI), Tc(VII),UO₂(NO₃)(TcO₄), Pu(NO₃)₃(TcO₄), Zr(NO₃)₃(TcO₄),

Sr(II), Ru(DI), Ru(TR), Ru(NI), Ru(MO), Cs(I), Ce(III), Gd(III), Am(III), Cm(III)

(但し、ADDMICカードによりマイクロ成分を追加し、CDISTまたはEQUILIBRIUMカードで追加マイクロの分配係数を入力することによりすることにより、これら以外の成分の計算が可能である。)

・DTOの値は計算精度に影響する。必要に応じて、セトラ部の液滞留時間の最小値以下の値を目安としてDTOの値を変えた計算を行い、DTOを適切な値に設定する必要がある。

Card No. 9 S T A G Eカード (必ず入力すること)

STAGE N, MSTG(N), IUDSTA(1,N), IUDSTA(2,N), IUDSTO(1,N), IUDSTO(2,N), IREAC(N)

- N = 抽出器の番号 (1~999)。機器を区別するために使用される。
貯槽を含めて機器についての一意的な番号とする必要がある。
- MSTG (N) = 抽出器Nの段数 (≤ 25)
- ICDSTA(1,N) = 抽出器Nの水相流が入る抽出器番号。
- ICDSTA(2,N) = 抽出器Nの水相流が入る抽出器内のステージ番号
- ICDSTO(1,N) = 抽出器Nの有機相流が入る抽出器番号。
- ICDSTO(2,N) = 抽出器Nの有機相流が入る抽出器内のステージ番号
- IREAC (N) = 抽出器Nの化学反応計算の指定
- 0 : M I X S E T 9 8相当の化学反応のみで計算する。
 このとき, 表4-5 (水相反応) の1~5, 27, 41, 42,
 および, 表4-6 (有機相反応) の1, 4, 41, 42のみが考慮され,
 これ以外の番号の反応に対するREACTIONカードの指定は無視される。
- 1 : 全ての化学反応を考慮する。

- 注) ・ S T A G Eカードは抽出器の数だけ必要であり, 必ずその抽出器に関するデータを入力するカードより前に入力しなければならない。
- ・ 抽出器の水相出口液の接続先が無い場合はICDSTA(1,N), ICDSTA(2,N)で0を指定し, 有機相出口液の接続先が無い場合はICDSTO(1,N), ICDSTO(2,N)で0を指定すること。
 - ・ 化学反応計算の指定のオプションは, 例えば, T c, N pが存在しない場合にウラン・プルトニウムの濃度分布を短時間で計算したい場合, M I X S E T 9 8相当の化学反応のみとして計算する場合を想定して設けてある。尚, 従来のM I X S E Tと異なり, REACTIONカードは単に化学反応速度の調整を行うために使用されており, REACTIONカードの有無では, 化学反応の有無を判断できない。

Card No.10 V O L U M Eカード (必ず入力すること)

VOLUME N, (V(I), I=1, MSTG(N)), (VS(I), I=1, MSTG(N))

- N = 抽出器番号
- V = 抽出器NステージIのミキサ容積 (ℓ)
- VS = 抽出器NステージIのセトラ容積 (ℓ)
- 抽出器の数だけ入力する。

Card No.11 H E I G H Tカード (省略可能)

HEIGHT N, (HEIGHT(I), I=1, MSTG(N))

- N = 抽出器番号
- HEIGHT = 抽出器NのステージIのセトラ部の界面レベル (0. 0~1. 0)
- HEIGHT(I) = 0. 5がデフォルト値

Card No.12 R E C Y C L Eカード (省略可能)

RECYCLE	N, (FR(I), I=1, MSTG(N))
---------	--------------------------

- N = 水相リサイクル流の存在する抽出器番号
 FR = 抽出器 N のステージ I の水相リサイクル流量[ℓ/Hour]
 デフォルト値は0である。

Card No.13 T O F S T Gカード (省略可能)

TOFSTG	N, (TSTG(I), I=1, MSTG(N))
--------	----------------------------

- N = 抽出器番号
 TSTG = 抽出器NのステージIの温度

注) このカードが入力されない場合は、温度は25°Cに設定される。

Card No.14 T A N Kカード (省略可能)

TANK	N, IPHX(N), VTOT(N), ACID(N), ACIDIN(N), OXID(N), FOUT(N), IUDST1(1, N), IUDST1(2, N), FFOUT2(N), IUDST2(1, N), IUDST2(2, N)
------	---

- N = 貯槽の番号(1~999)。機器を区別するために使用される。
 抽出器を含めて機器についての一意的な番号とする。
 IPHX(N) = 貯槽Nの相を指定する。(1:水相, -1:有機相)
 VTOT(N) = 貯槽Nの容量[ℓ]
 ACID(N) = 水相貯槽の酸調整の目標酸濃度(N)。0ならば酸調整をしない通常の貯槽。
 ACIDIN(N) = 水相貯槽の酸調整の調整用酸濃度(N)。(通常の貯槽では0を入力)
 OXID(N) = 水相貯槽のプルトニウム酸化率。0ならば酸化のない貯槽である。
 FOUT(N) = 貯槽Nの出口総流出流量[ℓ/hr](0ならば、総流出流量=総流入流量)
 IUDST1(1, N) = 貯槽Nの第1出口流が入る機器の番号
 IUDST1(2, N) = 貯槽Nの第1出口流が入る機器内のステージ番号(行き先が貯槽の時は1)
 FFOUT2(N) = 貯槽Nの第2出口流量[ℓ/hr](0~総流出流量の10倍までの指定が可能)
 IUDST2(1, N) = 貯槽Nの第2出口流が入る機器の番号
 IUDST2(2, N) = 貯槽Nの第2出口流が入る機器内のステージ番号(行き先が貯槽の時は1)

- 注) ・貯槽Nの第1出口流量は(総流出流量-貯槽Nの第2出口流量)で、負の場合0となる。
 ・貯槽の出口液の接続先が無い場合はIUDST1(1, N), IUDST1(2, N)等で0を指定すること。
 ・ACIDIN(N)は水相貯槽出口濃度の計算の際に用いられないが、0以外の値が入力されると酸調整の有無に関わらず(有機相貯槽を含む)、出口総流量がFOUT(N)に固定される。この場合でも、物質収支式で出口濃度が計算されるので、濃縮(又は、希釈)の簡易的な模擬が可能である。また、酸調整を行う場合(ACID(N)≠0)では、必要な調整酸濃度との違いを出力するので、入力ミスを確認するために利用できる。
 ・FOUT(N)≠0では、ACIDIN(N)=0の場合に、総流出流量の初期値がFOUT(N)となる。
 ・ACID(N)≠0では、貯槽出口の酸濃度は目標酸濃度となる。他の物質と同様に酸に対しても物質収支式で求めた出口濃度を使用する場合は、ACID(N)=0とし、FEEDカードで貯槽に調整用酸を供給する構成の入力データとすること。

Card No.15 INITIALカード(必ず入力すること)

成分濃度初期値に入力方法には2種類の方法がある。

- ・成分濃度初期値をカード入力する場合

```

INITIAL      N,J,M,(X(I),I=1~MSTG(N))

N            = 初期値を入力する機器番号(貯槽でもよい)
J            = 成分番号
M            = 0 [  $x_m(J, I) = x_s(J, I)$ , I=1~MSTG(N) ]
              = 1 [  $x_m(J, I)$ , I=1~MSTG(N) ]
              = 2 [  $y_m(J, I)$ , I=1~MSTG(N) ]
              = 3 [  $x_s(J, I)$ , I=1~MSTG(N) ]
              = 4 [  $y_s(J, I)$ , I=1~MSTG(N) ] を入力する。
              入力する濃度の単位はFEEDカードと同じである。
X            = 成分濃度初期値

```

- ・成分濃度初期値をファイルより読み込む場合

```

INITIAL      N,INFILE

N            = 0 (必ず0とする。)
INFILE       = 成分濃度を読み込むファイルの名前を拡張子を含めて指定する。
              必ず。クォーテーションで囲むこと。
              以前MIXSETを計算したときのプロット用出力ファイルの名前を指定す
              る。この入力方法を指定する場合には、プロット用ファイルを出力したとき
              の計算ケースと計算体系が同じであることが必要である。また、抽出器毎に
              入力することはできず、入力しても無効である。

```

Card No.16 FEEDカード (必ず入力すること)

各供給液の最初のFEEDデータの形式

FEED N, I, NSET, [t, FF, (XF(J), J=1~40), RUTOT]

N = 供給液の流入する機器番号(貯槽でもよい)
 I = 供給液の流入するステージ番号 (I > 0 : 水相流)
 (I < 0 : 有機相流)
 NSET = 入力する時間依存データのセット数。時間依存がない場合には1とする。
 t = 時間 (Hour)
 FF = tにおける供給液流量 (ℓ / Hour)
 XF(J) = tにおける供給液成分濃度
 (成分1から追加マイクロまでを含む成分濃度)
 RUTOT = tにおける全R u 指定での供給液中ルテニウム濃度。
 この値が0.0でないときは、この値に次の式で計算されるルテニウム錯体成分比を乗算したものが対応するR u の各成分濃度の値に加算されて計算で使用されるルテニウムのフィード濃度となる。
 ルテニウム錯体成分比の計算式。XHは硝酸濃度である。

$$RU(DI) = 1.615110 * (-DEXP(-0.1120660 * (XH + 0.610490)) + DEXP(-0.00690 * (XH * XH + 0.853990)))$$

$$RU(TR) = 0.07997080 * (XH + 2.202260) * DEXP(2.693150 / (XH + 2.202260)) - 0.5867920$$

$$RU(NI) = 0.4894390 * DEXP(0.2341360 / (XH + 0.4937680)) - 0.4887880$$

$$RU(MO) = 1.000 - RU(DI) - RU(TR) - RU(NI)$$

各供給液の2セット目以降のFEEDデータの形式

[t, FF, (XF(J), J=1~40), RUTOT]

t = 時間 (Hour)
 FF = tにおける供給液流量 (ℓ / Hour)
 XF(J) = tにおける供給液成分濃度
 (成分1から追加マイクロまでを含む成分濃度)
 RUTOT = tにおける供給液成全R u 濃度。

- 注) ・追加マイクロ成分を指定しない場合でも該当の変数に0.0を入力すること。
 ・追加マイクロ成分の濃度は32~40番目になるようにすること。その場合、必要でない部分には0.0を入力すること。
 ・ [t, FF, XF, RUTOT] がセットとなり、最大20個まで時間依存供給液を入力可能。
 ・供給液が一定の場合、t=0とし1セットだけで良い。
 ・FP, TRU成分, 追加マイクロ成分の供給液成分濃度も、本カードにて指定する。
 ・成分番号Jと対応する各成分名を表4-3に示す。
 ・追加マイクロ成分の濃度の単位はg/ℓとする。
 ・各供給液のFEEDデータの最初のセットには、FEEDキーワードをつけること。
 ・このカードは供給液流量と供給液成分濃度を1つのデータのセットとして、複数セットのデータを入力する場合には、2セット目以降ではFEEDキーワードをつけずに入力すること。2セット目以降のキーワード部分に空白以外を入力すると誤りである。
 ・このデータはかなり長いものになる。従って、成分濃度のデータの前に!*...!*形式のコメントで成分の名前を入れるとわかりやすくなる。
 ・各セットの時間の間のデータは内挿される。ステップ状に変化させる入力を行う時は、同時刻で値の異なるデータを2セット入力する。(実際には時刻t以降のステップで読み込まれるので、TAUCONSTコマンドで適切な過渡計算刻み時間を指定する必要がある)

表4-3 FEEDカードの成分番号と成分名の対応表(1)

J	CNAME (J)	CUNIT	CVUNIT
1	HNO ₃	mol/l	1.0
2	U (V)	g/l	238.03
3	P u (N)	g/l	239.0
4	P u (III)	g/l	239.0
5	U (M)	g/l	238.03
6	HNO ₂	mol/l	1.0
7	HYD	g/l	32.0
8	HAN	g/l	33.0
9	P u (V)	g/l	239.0
10	P u (M)	g/l	239.0
11	N p (M)	g/l	237.0
12	N p (V)	g/l	237.0
13	N p (M)	g/l	237.0
14	Z r (M)	g/l	95.0
15	T c (M)	g/l	99.0
16	T c (V)	g/l	99.0
17	T c (M)	g/l	99.0
18	T c (VII)	g/l	99.0
19	T c - U (M)	g/l	99.0
20	T c - P u (M)	g/l	99.0
21	T c - Z r (M)	g/l	99.0
22	S r (II)	g/l	90.0
23	R u (DI)	g/l	106.0
24	R u (TR)	g/l	106.0
25	R u (NI)	g/l	106.0
26	R u (MO)	g/l	106.0
27	C s (I)	g/l	137.0
28	C e (III)	g/l	144.0
29	G d (III)	g/l	153.0
30	A m (III)	g/l	242.0
31	C m (III)	g/l	243.0

HAN : NH₃OH⁺HYD : N₂H₅⁺

R u (DI) : R u ジニトラ錯体

R u (TR) : R u トリニトラ錯体

R u (NI) : R u ジニトロ錯体

R u (MO) : R u モノニトラ錯体

CVUNIT : 入力値からモル濃度への変換係数[g/mol]

Card No.17 S D I S Tカード (省略可能)

SDIST N, (IDIST(J), J=1, 40)

- N = 抽出器番号
 J = 成分番号
 IDIST = 抽出器Nの成分Jの分配係数の計算式の選択
 0, 1, 4 : プログラム内蔵の計算式
 2 : 定数の分配係数 (CDISTカードで入力した値)
 3 : 定数の平衡定数 (EQUILIBRIUMカードで入力した値)

注) デフォルト値として、追加マイクロ成分についてはIDIST=2が、その他の成分についてはIDIST=0が設定されている。各成分に対するIDISTの値とプログラム内蔵の分配係数計算式の間を関係を表4-4に示す。詳細については付録Cを参照のこと。
 なお、表4-4に記載の無い成分の分配係数内蔵値は0としている。

表4-4 S D I S Tカードによる内蔵分配係数計算式の選択対応表

J	CNAME (J)	IDIST=0	IDIST=1	IDIST=4
1	HNO ₃	KfK	Hanford	KfK*
2	U (V)	KfK	Hanford	KfK*
3	P u (N)	KfK	Hanford	KfK*
4	P u (III)	**		
5	U (M)	KfK	R.MIXSET	KfK***
6	HNO ₂	JAERI	JAERI	A.MIXSET
10	P u (V)	**		
11	N p (N)	A.MIXSET*	A.MIXSET*	A.MIXSET
12	N p (V)	A.MIXSET*	A.MIXSET*	A.MIXSET
13	N p (VI)	KfK	A.MIXSET*	A.MIXSET
14	Z r (N)	**		
18	T c (VII)	CEA	ORNL	CEA
19	T c - U (VI)	CEA	ORNL	CEA*
20	T c - P u (N)	CEA	ORNL	CEA*
21	T c - Z r (N)	CEA	ORNL	CEA*
22	S r (II)	**		
23	R u (DI)	**		
24	R u (TR)	**		
25	R u (NI)	**		
27	C s (I)	**		
28	C e (III)	**		
29	G d (III)	**		
30	A m (III)	**		
31	C m (III)	**		

*, *** : 付録C参照 **: 内蔵の分配係数式は1種類

R u (DI) : R u ジニトラ錯体
 R u (TR) : R u トリニトラ錯体
 R u (NI) : R u ジニトロ錯体

Card No.18 EQUILIBRIUMカード (省略可能)

```

EQUILIBRIUM  N,J,(CKEQU(I),I=1,MSTG(N))

```

- N = 抽出器番号
 J = 成分番号
 CKEQU = 抽出器Nの成分JのステージI毎に準備される定数の平衡定数

注) ・平衡定数が一定の分配係数を使用したいときにこのカードを用いる。実際の計算に使用するためにはSDISTの分配係数の計算式の選択で定数の平衡定数を指定する必要がある。

Card No.19 CDISTカード (省略可能)

```

CDIST        N,J,(CDIS(I),I=1,MSTG(N))

```

- N = 抽出器番号
 J = 成分番号
 CDIS = 抽出器Nの成分JのステージI毎に準備される一定の分配係数

注) ・定数の分配係数を使用したいときにこのカードを用いる。実際の計算に使用するためにはSDISTの分配係数の計算式の選択で平衡定数を指定する必要がある。

Card No.20 EFFICIENCYカード (省略可能)

```

EFFICIENCY   N,J,IEFF(J;N),(EFF(J,I),I=1~MSTG(N))

```

- N = 回収率型段効率を入力する抽出器番号
 J = 成分番号
 IEF(J,N) = 抽出器Nの成分Jの回収率型段効率の定義方法
 - 1 : 有機相への回収率で定義
 1 : 水相への回収率で定義
 EFF(J,I) = 抽出器NのステージIの成分Jの回収率型段効率で、デフォルト値は1.0
 分配係数が内蔵計算式の値より大きい場合を考慮し、1.0以上の値を入力可能である。
 また、現在の所、Tc共抽出物については水相定義の段効率を入力できない。

Card No.21 REACTIONカード (省略可能)

```

REACTION     N,J,FCHEM(J)

```

- N = 抽出器番号
 J = 化学反応番号。化学反応番号と化学反応の対応を表4-5, 表4-6に示す。
 値が正のとき水相反応
 値が負のとき有機相反応
 FCHEM = 化学反応速度定数に乗算する反応速度調整因子。
 プログラム内で設定されている化学反応速度にFCHEMを乗算した値が数値計算上の反応速度として取り扱われる。この値を0.0に設定することにより化学反応を抑制することができる。デフォルト値は1.0であり、計算対象成分に関係する全ての化学反応が考慮される。

表4-5 化学反応番号と化学反応の対応(水相反応)(1/2)

反応番号	化学反応式
1*	$2\text{Pu(V)} + \text{U(IV)} + 2\text{H}_2\text{O} \rightarrow 2\text{Pu(III)} + \text{U(VI)} + 4\text{H}^+$
2*	$2\text{Pu(V)} + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Pu(III)} + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
3**	$2\text{Pu(V)} + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Pu(III)} + 2\text{NH}_4^+ + \text{N}_2 + 2\text{H}^+$
4***	$2\text{Pu(III)} + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Pu(V)} + \text{HNO}_2 + \text{H}_2\text{O}$
5***	$\text{HNO}_2 + \text{N}_2\text{H}_5^+ \rightarrow \text{H}^+ + 2\text{H}_2\text{O} + \text{NH}_3$
6	$\text{Np(VI)} + \text{Np(IV)} + 2\text{H}_2\text{O} \rightarrow 2\text{Np(V)} + 4\text{H}^+$
7	$\text{Pu(V)} + \text{Np(IV)} + 2\text{H}_2\text{O} \rightarrow \text{Pu(III)} + \text{Np(V)} + 4\text{H}^+$
8	$2\text{Np(VI)} + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Np(V)} + \text{N}_2 + 2\text{H}^+ + 2\text{NH}_4^+$
9	$2\text{Np(VI)} + \text{U(IV)} + 2\text{H}_2\text{O} \rightarrow 2\text{Np(V)} + \text{U(VI)} + 4\text{H}^+$
10	$2\text{Np(V)} + \text{U(IV)} + 4\text{H}^+ \rightarrow 2\text{Np(IV)} + \text{U(VI)} + 2\text{H}_2\text{O}$
11	$2\text{Np(VI)} + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Np(V)} + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
12	$\text{Pu(V)} + \text{Pu(IV)} \rightarrow \text{Pu(VI)} + \text{Pu(III)}$
13	$\text{Pu(V)} + \text{Pu(III)} + 4\text{H}^+ \rightarrow 2\text{Pu(IV)} + 2\text{H}_2\text{O}$
14	$2\text{Pu(VI)} + \text{HNO}_2 + \text{H}_2\text{O} \rightarrow 2\text{Pu(V)} + 3\text{H}^+ + \text{NO}_3^-$
15	$\text{Pu(VI)} + \text{Pu(III)} \rightarrow \text{Pu(V)} + \text{Pu(IV)}$
16	$2\text{Pu(V)} + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Pu(VI)} + \text{HNO}_2 + \text{H}_2\text{O}$
17	$\text{NO}_3^- \rightarrow \text{HNO}_2$
18	$\text{HNO}_2 \rightarrow \text{H}^+$
19	$2\text{Np(V)} + 4\text{H}^+ \rightarrow \text{Np(IV)} + \text{Np(VI)} + 2\text{H}_2\text{O}$
20	$2\text{Np(V)} + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Np(VI)} + \text{HNO}_2 + \text{H}_2\text{O}$
21	$2\text{Np(IV)} + \text{NO}_3^- + 3\text{H}_2\text{O} \rightarrow 2\text{Np(V)} + \text{HNO}_2 + 5\text{H}^+$
22	$\text{Pu(III)} + \text{Np(VI)} \rightarrow \text{Pu(IV)} + \text{Np(V)}$
23	$\text{Pu(III)} + \text{Np(V)} + 4\text{H}^+ \rightarrow \text{Pu(IV)} + \text{Np(IV)} + 2\text{H}_2\text{O}$
24	$2\text{Pu(VI)} + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Pu(V)} + 2\text{NH}_4^+ + \text{N}_2 + 2\text{H}^+$
25	$2\text{Pu(IV)} + 2\text{H}_2\text{O} \rightarrow \text{Pu(V)} + \text{Pu(III)} + 4\text{H}^+$
26	$2\text{Pu(V)} + 4\text{H}^+ \rightarrow \text{Pu(VI)} + \text{Pu(IV)} + 2\text{H}_2\text{O}$
27*	$\text{HNO}_2 + \text{NH}_3\text{OH}^+ \rightarrow \text{N}_2\text{O} + 2\text{H}_2\text{O} + \text{H}^+$
28	$2\text{Pu(VI)} + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Pu(V)} + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
29	$2\text{Pu(V)} + 2\text{NH}_3\text{OH}^+ + 2\text{H}^+ \rightarrow 2\text{Pu(III)} + \text{N}_2\text{O} + 5\text{H}_2\text{O}$
30	$2\text{Pu(VI)} + \text{U(IV)} + 2\text{H}_2\text{O} \rightarrow 2\text{Pu(V)} + \text{U(VI)} + 4\text{H}^+$

注) * 従来よりMIXSETで考慮されていた化学反応である。

** EXTRA, Mで考慮されており, 従来よりMIXSET98でも考慮されていた反応であるが, 従来のMIXSETの反応速度定数は推定値にすぎないため, 実験に基づくEXTRA, Mの反応速度定数をMIXSET-Xでは使用している。

*** EXTRA, Mでは考慮されていないが, 従来よりMIXSETで考慮されていた化学反応である。これらの反応に対応するEXTRA, Mの反応式は次のようになっている。

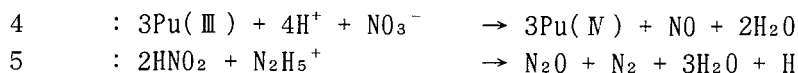


表4-5 化学反応番号と化学反応の対応(水相反応)(2/2)

31	: Tc(VI) + N ₂ H ₅ ⁺	→ Tc(V) + products
32	: Tc(V) + N ₂ H ₅ ⁺	→ Tc(IV) + products
33	: Tc(V) + N ₂ H ₅ ⁺	→ Tc(IV) + products
34	: Tc(IV) + NO ₃ ⁻	→ Tc(V) + NO ₂ ⁻
35	: Tc(V) + NO ₃ ⁻	→ Tc(VI) + NO ₂ ⁻
36	: Tc(V) + NO ₃ ⁻	→ Tc(VI) + NO ₂ ⁻
37	: Tc(IV) + Tc(VI)	→ Tc(V) + Tc(V)
38	: Tc(VI) + 3Pu(III) + 4H ⁺	→ Tc(IV) + 3Pu(IV) + 2H ₂
39	: 2Tc(VI) + 3U(IV)	→ 2Tc(IV) + 3U(V)
40	: Tc(VI) + NH ₃ OH ⁺	→ Tc(V) + products
41**	: U(IV) + NO ₃ ⁻ + H ₂ O	→ U(V) + H ⁺ + HNO ₂
42**	: 2U(IV) + O ₂ + 2H ₂ O	→ 2U(V) + 4H ⁺
43****	: 2Np(V) + 2N ₂ H ₄ + 8H ⁺	→ 2Np(IV) + 2NH ₄ ⁺ + N ₂ + 4H ₂ O

注)** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。

****Advanced MIXSETに記載されている反応で、EXTRA. Mでは考慮されていない化学反応である。

表4-6 化学反応番号と化学反応の対応(有機相反応)

反応番号	化学反応式
1*	: 2Pu(IV) + U(IV) + 2H ₂ O → 2Pu(III) + U(V) + 4H ⁺
4****	: 2Pu(III) + 3H ⁺ + NO ₃ ⁻ → 2Pu(IV) + HNO ₂ + H ₂ O
5	: 2HNO ₂ + (N ₂ H ₅ ⁺) _{aq} → N ₂ O + N ₂ + 3H ₂ O + H ⁺
	(ミキサ部のみ)
6	: Np(V) + Np(IV) + 2H ₂ O → 2Np(V) + 4H ⁺
9	: 2Np(V) + U(IV) + 2H ₂ O → 2Np(V) + U(V) + 4H ⁺
14	: 2Pu(V) + HNO ₂ + H ₂ O → 2Pu(IV) + 3H ⁺ + NO ₃ ⁻
19	: 2Np(V) + 4H ⁺ → Np(V) + Np(IV) + 2H ₂ O
20	: 2Np(V) + 3H ⁺ + NO ₃ ⁻ → 2Np(V) + HNO ₂ + H ₂ O
41**	: U(IV) + NO ₃ ⁻ + H ₂ O → U(V) + H ⁺ + HNO ₂
42**	: 2U(IV) + O ₂ + 2H ₂ O → 2U(V) + 4H ⁺

注)* 従来よりMIXSETで考慮されていた化学反応である。

** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。

*** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。これらの反応に対応するEXTRA. Mの反応式は次のようになっている。



Card No.22 EPSILONカード (省略可能)

EPSILON	EPS, EPSLIM, RXLIM
---------	--------------------

- EPS = 濃度計算の許容相対誤差目標値。デフォルト値は 10^{-4}
 EPSLIM = 濃度計算の許容相対誤差限界値。デフォルト値は 10^{-3}
 RXLIM = 濃度計算の誤差判定を絶対誤差判定に変更する限界濃度値。
 この値より濃度の計算値が小さい場合には計算された濃度をRXLIMで割った値が誤差評価に使われる。デフォルト値は 10^{-13}

注) ここで、EPSはニュートン法の繰り返し計算の途中で相対誤差が、EPSより小さくなった場合に収束 (完全収束) と判定する値であり、EPSLIM はニュートン法の繰り返し計算で相対誤差が EPSより小さくならずには繰り返し計算の最大回数になった場合でも EPSLIMより相対誤差が小さい場合には収束 (弱い収束) と判定する値である。

Card No.23 ITERATIONカード (省略可能)

ITERATION	ITMIN, ITMAX
-----------	--------------

- ITMAX = 濃度計算の収束計算反復上限数。デフォルト値は 20
 ITMIN = 濃度計算の収束計算反復回数がこの値以下の場合、TAULIMカードのTRATEを使用中の計算ステップ幅に乘算したものを新しいタイムステップ幅とする。ただし、それがタイムステップ幅の上限以上であれば、その上限値になる。
 デフォルト値は 5

Card No.24 TAULIMカード (省略可能)

TAULIM	DTMIN, DTMAX, TRATE
--------	---------------------

- DTMIN = 過渡計算で使用される計算刻み時間の最小値(Hour)。デフォルト値は 10^{-4}
 DTMAX = 過渡計算で使用される計算刻み時間の最大値(Hour)。デフォルト値は 240
 TRATE = 少ない計算繰り返し数で収束するとき計算刻み時間をTRATE倍する。
 TRATEのデフォルト値は 1.2

Card No.25 T A U C O N S Tカード (省略可能)

TAUCONST NTAUC, TMCONSB, TMCONSE, DTCONS (1セット目)

TMCONSB, TMCONSE, DTCONS (2セット目以降)

- NTAUC = 過渡計算において過渡変化の直後はタイムステップ幅をTAULIMカードのDTMAXよりも制限したほうが良い場合がある。そのようなときに入力するカードで、入力する制限時間のセット数を入力する。
デフォルト値は0。(すなわち制限しない)
- TMCONSB = タイムステップ制限を開始する時刻を指定する。(Hour)
- TMCONSE = タイムステップ制限を終了する時刻を指定する。(Hour)
- DTCONS = 指定時刻の間、過渡計算で使用される計算刻み時間の最大値(Hour)

注) DTCONSの値は計算精度に影響する。精密な過渡応答を計算する必要がある場合は、DTCONSの値をセトラ部の液滞留時間の最小値以下に設定する必要がある。

Card No.26 P R I N Tカード (省略可能)

PRINT NPRT (DTPRT(I), I=1, NPRT)

- NPRT = 印刷出力用データの出力時刻指定個数
- DTPRT = 印刷出力用データの出力時刻指定 (Hour)

注) このカードで指定されない場合でも、CONTROLカードで指定したENDTIMでのPRINT出力は行われる。

Card No.27 P L O Tカード (省略可能)

PLOT DTPLT

- NPLT = プロット用ファイル出力の出力個数
- DTPLT = プロット用データファイルの出力時刻 (Hour)

注) ・このプロット用データは従来のSPLOT, TPLLOTを統合した図形処理用のファイルであり、プロット用データの出力間隔毎に、その時刻の段毎のミキサ部成分濃度、セトラ部成分濃度を全て出力したファイルである。作図用のデータは、このプロット用出力データファイルから適当なものを読み込んで時間依存のデータまたは濃度分布の形に加工したものを使用する。なお、本プログラムでは作図機能はサポートしていないので作図には市販のソフトウェア等を使用する。

・このカードで指定されない場合でも、CONTROLカードで指定したENDTIMでのPLOT出力は行われる。

Card No.28 L D I S Tカード (省略可能)

LDIST

入力されると、分配係数のプリントを行う。

Card No.29 F I L Eカード (省略可能)

FILE CTFILE

CTFILE = 計算結果を出力するファイルの名前を指定する。
 必ず。クォーテーションで囲むこと。
 指定した名前に'.outlist'の修飾子をつけたファイルに計算結果が
 出力される。
 このファイルを指定しない場合には標準出力に出力される。

Card No.30 E N Dカード (必ず入力すること)

END

全てのカードの最後に入力し計算を開始する。(ENDカード以降の入力は無視される)

4. 4 エラーメッセージ

4. 4. 1 入力のエラーチェック

入力に誤りがある場合には、エラーメッセージを出して計算を行わない。以下に示すエラーチェックが行われる。

入力のエラーメッセージ一覧については付録の表F-1に示す。

(1) 入力カードの枚数及び順番に関するもの

- ・最初のカードがCONTROLカードであること
- ・CONTROLカードはSTAGEカードより先に入力されていること
- ・STAGEカードはバンクの数だけ入力されていること
- ・VOLUMEカードがバンクの数だけ入力されていること
- ・FEEDカードが各バンクに入力されていること
- ・ENDカードが入力されていること
- ・ADDMICカードがFEEDカードより先に入力されていること
- ・ADDMICカードがINITIALカードより先に入力されていること

(2) 各カードで入力された入力値について

- ・CONTROLカードで入力されたバンク数がプログラムで許される最大数以下であること
- ・CONTROLカードの溶液濃度オプションが0または1であること
- ・CONTROLカードの計算刻み時間が負でないこと
- ・STAGEカードのバンク番号が1～999であること
- ・STAGEカードのバンク番号に重複がないこと
- ・STAGEカードのステージ数がプログラムで許される最大数以下であること
- ・STAGEカードのステージ数の合計がプログラムで許される最大数以下であること
- ・ADDMICカードの追加マイクロ番号がプログラムで許される値であること
- ・ADDMICカードの追加マイクロ番号に重複がないこと
- ・ADDMICカードの電荷、TBP配位数が負でないこと
- ・ADDMICカードの濃度単位変換係数が正であること
- ・CHARGEカードの成分番号がプログラムで許される値であること
- ・CHARGEカードの電荷、TBP配位数が負でないこと
- ・TBPカードのTBP体積分率が0から1までの範囲にあること
- ・DBPカードのDBP濃度が負でないこと
- ・VOLUMEカードのバンク番号がSTAGEカードで定義されていること
- ・VOLUMEカードのバンク番号に重複がないこと
- ・VOLUMEカードのミキサ容量、セトラ容量が正であること
- ・HEIGHTカードのバンク番号がSTAGEカードで定義されていること
- ・HEIGHTカードのバンク番号に重複がないこと
- ・HEIGHTカードの界面レベルが0から1までの範囲にあること
- ・FEEDカードのバンク番号がSTAGEカードで定義されていること
- ・FEEDカードのバンク番号に重複がないこと
- ・FEEDカードのステージ番号が適切な範囲にあること
- ・FEEDカードのデータのセット数がプログラム許される最大数以下であること
- ・FEEDカードの時間がデータのセット内で昇順になっていること
- ・FEEDカードの時間が負でないこと
- ・FEEDカードの供給液流量、濃度が負でないこと
- ・FEEDカードの供給液本数がプログラム許される最大数以下であること
- ・FEEDカードで各バンクのステージ1に有機相が供給されていること
- ・FEEDカードで各バンクの最終ステージに水相が供給されていること
- ・RECYCLEカードのバンク番号がSTAGEカードで定義されていること
- ・RECYCLEカードのバンク番号に重複がないこと

- ・ RECYCLEカードの水相リサイクル流量が負でないこと
- ・ EQUILIBRIUMカードのバンク番号がSTAGEカードで定義されていること
- ・ EQUILIBRIUMカードの成分番号が適切であること
- ・ EQUILIBRIUMカードの平衡定数が負でないこと
- ・ CDISTカードのバンク番号がSTAGEカードで定義されていること
- ・ CDISTカードの成分番号が適切であること
- ・ CDISTカードの分配係数が負でないこと
- ・ SDISTカードのバンク番号が適切な範囲にあること
- ・ SDISTカードの成分番号が適切であること
- ・ SDISTカードの分配係数の計算式選択オプションが0, 1, 2, 3, 4のいずれかであること
- ・ REACTIONカードのバンク番号がSTAGEカードで定義されていること
- ・ REACTIONカードの化学反応番号が適切であること
- ・ REACTIONカードの反応速度調整因子が負でないこと
- ・ EFFICIENCYカードのバンク番号がSTAGEカードで定義されていること
- ・ EFFICIENCYカードの成分番号が適切であること
- ・ EFFICIENCYカードの段効率が負でないこと
- ・ INITIALカードのバンク番号がSTAGEカードで定義されていること
- ・ INITIALカードの成分番号が適切であること
- ・ INITIALカードの濃度が負でないこと
- ・ PRINTカードの印刷用データ出力指定時刻が正であること
- ・ PLOTカードの作図用データ出力指定時刻が正であること
- ・ PRINT及びPLOTカードの出力時刻指定セット数がプログラム許される最大数以下であること
- ・ EPSILONカードの反復上限数が正であること
- ・ EPSILONカードの許容相対誤差が正であること
- ・ TAULIMカードの計算刻み時間の最大値と最小値が正であり最大値のほうが大きいこと
- ・ TAULIMカードの出力時刻指定セット数がプログラム許される最大数以下であること
- ・ TOFSTGカードのバンク番号が適切な範囲にあること
- ・ TOFSTGカードの温度が正であること

4. 4. 2 計算開始後のメッセージ

計算開始後は以下のメッセージを出力する。途中、計算エラーが生じた場合以下に示すメッセージは出して計算を終了する。

計算開始後のエラーメッセージ一覧については付録の表F-2に示す。

(1) 通常メッセージ

- ・ 計算対象の時刻及びタイムステップ
- ・ 各反復計算の回数
- ・ フリーTBP濃度 (T_f) の計算でエラーが生じ、再度計算を行うとき

(2) 計算を途中で終了した時のメッセージ

- ・ システムマトリックスの次元が最大値 ($MXSYS = 11$) を越えたとき
- ・ 行列がsingularとなり、方程式を解くことができないとき
- ・ タイムステップを最小値DTMINとしても計算が指定反復回数以下で計算が収束しないとき

5 総合解析計算

5.1 実験解析

MIXSET-Xを用いて5ケースの実験解析を行った。選んだ実験はいずれもJAERI 1331⁽³⁾に記載されているものでEXTRA, M⁽³⁾の検証解析用に使用されたものである。

(1) Pu(IV)抽出実験

実験のフローシートを図5-1に示す。レポートにはミキサセトラについての容積の記述が無いので、ミキサ部容量を10mLセトラ部容積を50mLとして解析を行った。MIXSET-X用の入力データを図5-2に示す。MIXSET-Xによる計算結果を図5-3に示す。計算結果はよく一致している。

(2) Pu(IV)-Pu(VI)抽出実験

実験のフローシートを図5-4に示す。レポートにはミキサセトラについての容積の記述が無いので、ミキサ部容量を10mLセトラ部容積を50mLとして解析を行った。MIXSET-X用の入力データを図5-5に示す。MIXSET-Xによる計算結果を図5-6に示す。計算結果はよく一致している。

(3) HAN還元Pu逆抽出実験

実験のフローシートを図5-7に示す。レポートにはミキサセトラについての容積の記述が無いので、ミキサ部容量を10mLセトラ部容積を50mLとして解析を行った。MIXSET-X用の入力データを図5-8に示す。MIXSET-Xによる計算結果を図5-9に示す。計算結果は実験値とかなり異なっている。HANとPu(IV)の反応速度はウラナスとPu(IV)の反応速度よりもかなり小さく、HANによるPu(IV)の還元反応の場合は滞留時間が大きく結果に影響する。その影響を見るためにミキサセトラの容積を1/10にした計算を行った結果を図5-10に示す。また、反応速度が大きすぎる可能性もあるので反応速度を1/10にした計算も行い、その結果を図5-11に示す。図5-10と図5-11を比較するとわかるように、ミキサセトラの容積を小さくした場合と反応速度を小さくした場合の計算結果はほとんど同じである。これは、反応速度が比較的小さい場合には、化学反応量は滞留時間と反応速度の積に概略比例することで説明できる。その意味でMIXSET-Xの計算は正しく行われているものと考えられるが、滞留時間を1/10にしても計算結果と実験値はあまり一致しない。この原因はよくわからないが、段平衡モデルではなく物質移動モデルに基づくEXTRA, Mによる解析では、水相のPu(IV)以外の成分については比較的計算値と実験値が一致しているので、この実験の場合には、分配平衡について物質移動速度を考慮しなければならない可能性も考えられる。

(4) Np抽出実験

実験のフローシートを図5-12に示す。Eurochemic Plantのデータ⁽³⁾である。このレポートにはミキサセトラについての容積の記述が無いので、ミキサ部容量を30mLセトラ部容積を150mLとして解析を行った。MIXSET-X用の入力データを図5-13に示す。

MIXSET-Xによる計算結果を図5-14に示す。この計算結果はフィード開始から8時間後の状態である。データの方も定常状態でのデータでは無いので直接の比較は困難であるが、Npがプロセスフローシートの中段で蓄積される状況が計算でも再現されている。

(5) Tc抽出実験

実験のフローシートを図5-15に示す。KfKのKolaricらのデータ¹³である。レポートにはミキサセトラについての容積の記述が無いため、ミキサ部容量を30mLセトラ部容積を150 mLとして解析を行った。MIXSET-X用の入力データを図5-16に示す。MIXSET-Xによる計算結果を図5-17に示す。この計算結果はフィード開始から20時間後の状態である。データの方も定常状態でのデータでは無いため直接の比較は困難であるが、MIXSET-Xのコードでは有機相にTcが蓄積する状況が再現されていない。

図からわかるように、MIXSET-Xの計算ではTc-UまたはTc-Puの形でウランとプルトニウムに共抽出されるTcが非常に多い。このために実験と異なる分布となった可能性が高い。還元環境にないため化学反応の影響は少ないと考えると、MIXSET-Xに組み込まれている分配係数式があまり適切でない可能性がある。

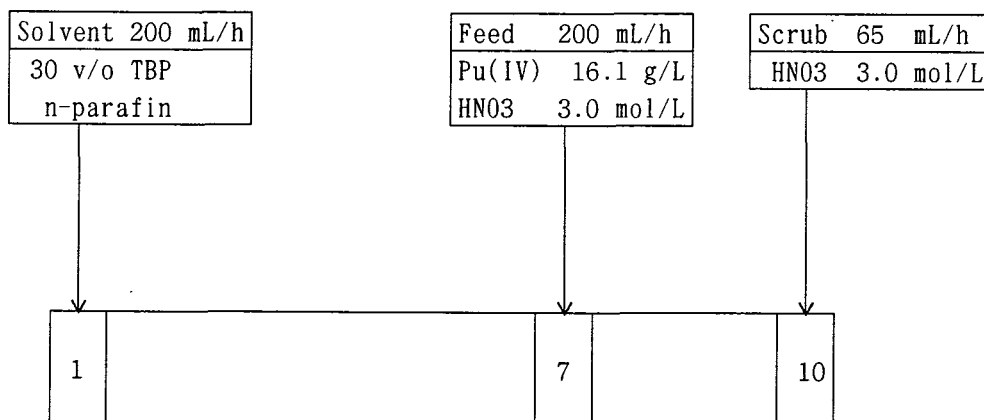


図 5 - 1 P u (IV)抽出実験のフローシート

```

file fig53a.inp  JAERI 1331  Fig 5-3 Experiment(a)
CONTR 0 0 40.0 0.01
STAGE 1 10 0 0 0 0 1
VOLUM 1 10*0.01 10*0.05
RECYC 1 10*0.0
*-BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD-----
FEED 1 -1 1 0.0 0.20 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 40*0.0
FEED 1 7 1 0.0 0.20 3.0 0.0 16.1 0.0 0.0 0.0 0.0 0.0 0.0
          0.0 0.0 0.0 0.0 0.0 28*0.0
FEED 1 10 1 0.0 0.065 3.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 40*0.0
*-----
INITI 1, 1, 0, 10*3.0
*-----
TAUCONST 1 0,40,1.0
FILE 'fig53a'
END
    
```

図 5 - 2 P u (IV)抽出実験解析の入力データ

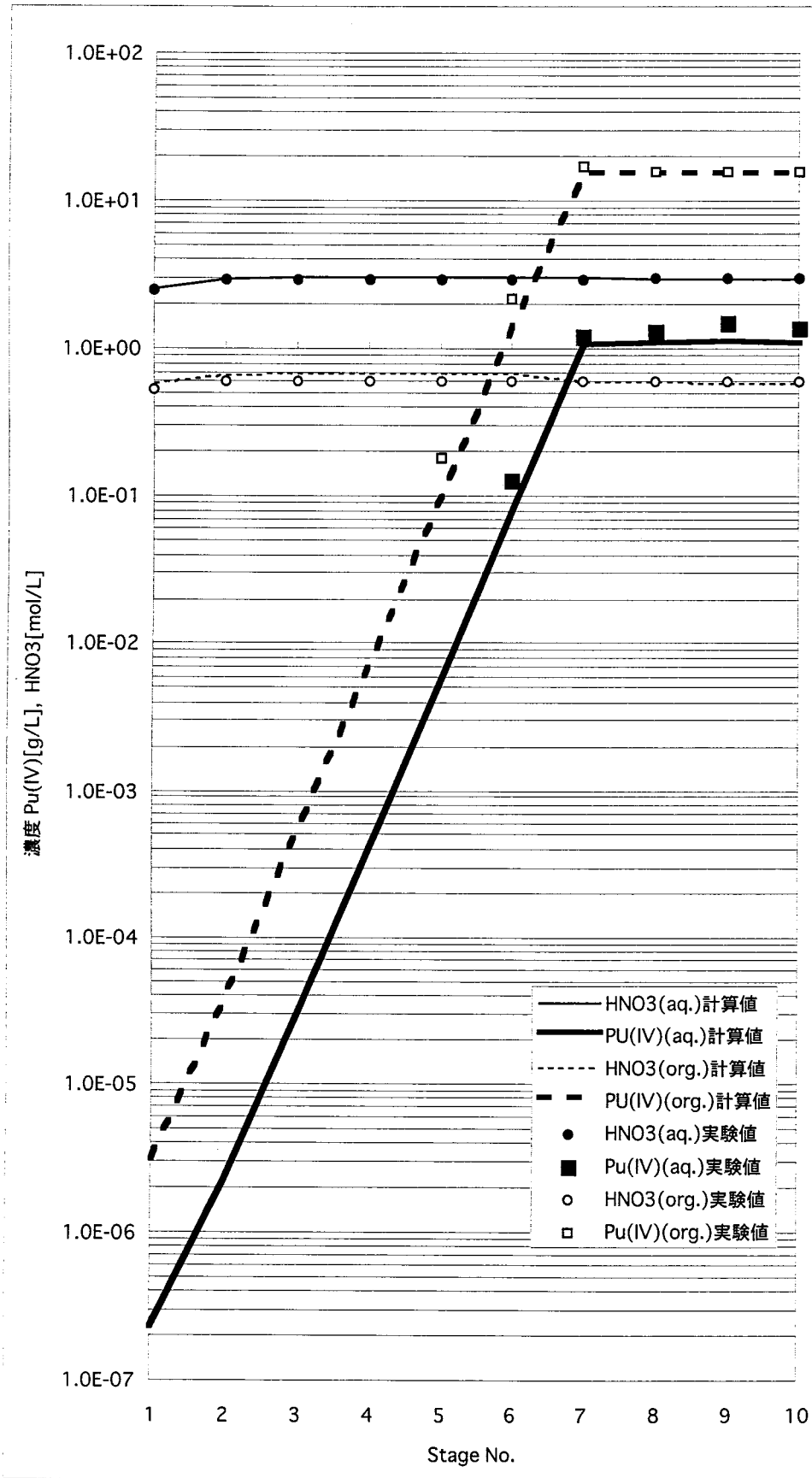


図 5 - 3 Pu(IV)抽出実験解析 (MIXSET)

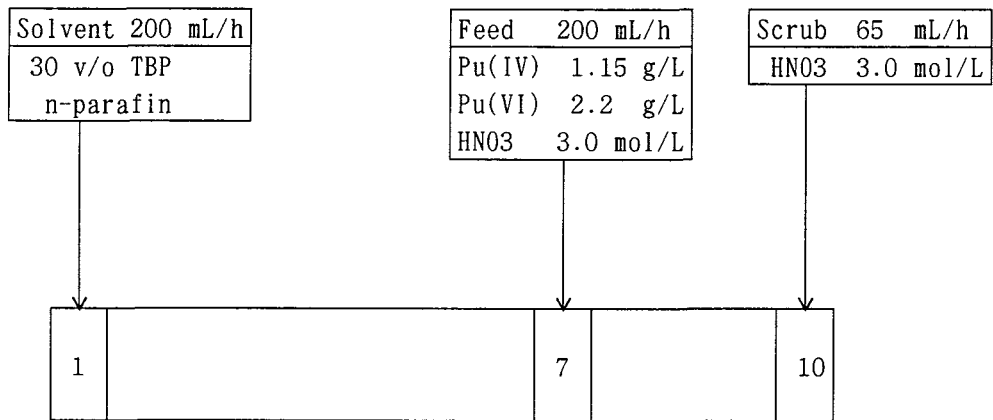


図 5-4 Pu(IV)-Pu(VI)抽出実験のフローシート

```

file fig53b.inp  JAERI 1331  Fig 5-3 Experiment(b)
CONTR 0 0 40.0  0.01
STAGE 1 10 0 0 0 0 1
VOLUM 1 10*0.01  10*0.05
RECYC 1 10*0.0
*----BANK-STAGE-NSET-TIME--FLOW---H+----UO2--PU4--PU3--U4---HN02--HYD--
FEED  1 -1 1 0.0 0.20  0.0                                40*0.0
FEED  1  7 1 0.0 0.20  3.0 0.0   1.15   0.0   0.0  0.0  0.0  0.0
      0.0 2.2   0.0   0.0   0.0                                28*0.0
FEED  1 10 1 0.0 0.065 3.0                                40*0.0
*-----
INITI  1, 1, 0, 10*3.0
*-----
TAUCONST  1 0,40,1.0
FILE  'fig53b'
END
    
```

図 5-5 Pu(IV)-Pu(VI)抽出実験解析の入力データ

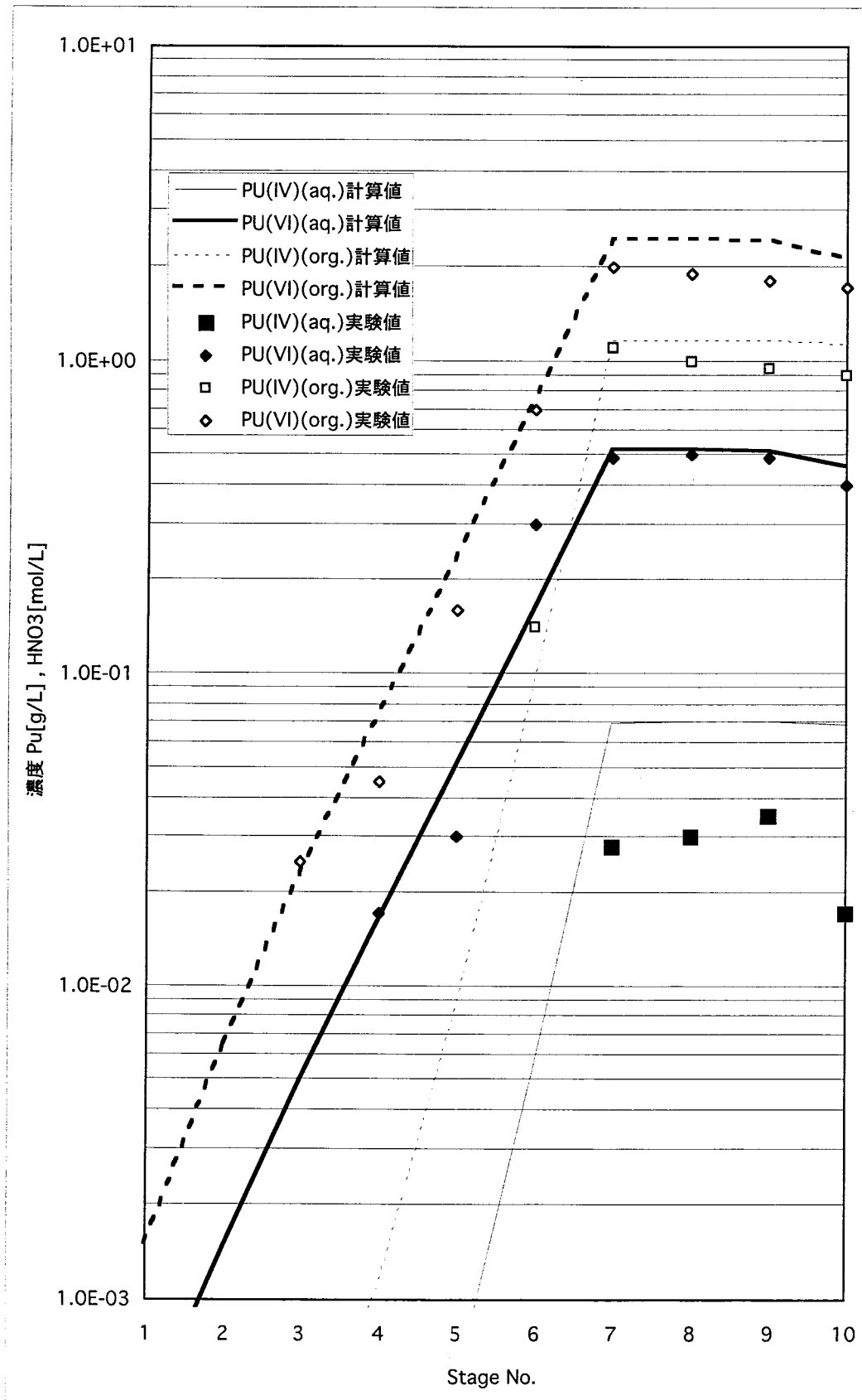


図5-6 Pu(IV)-Pu(VI)抽出実験解析(MIXSET)

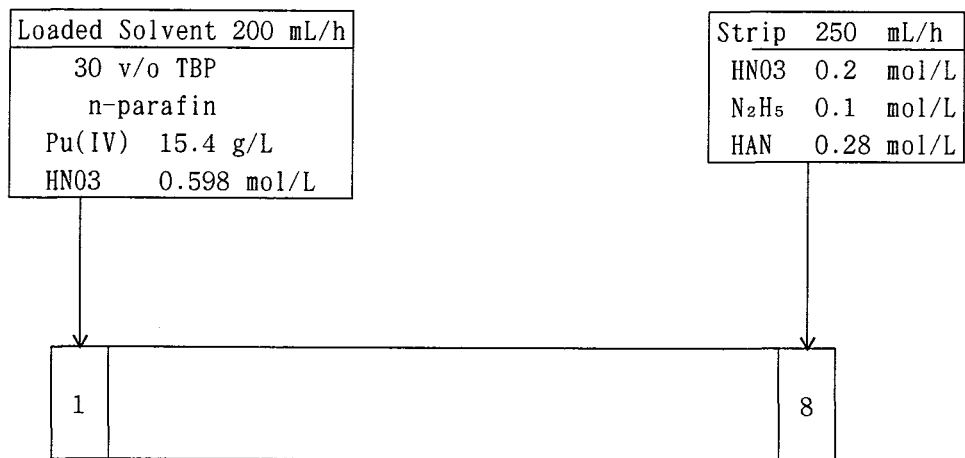


図5-7 HAN還元Pu逆抽出実験のフローシート

```

file fig56.inp  JAERI 1331  Fig 5-6 Experiment
CONTR 0 0 40.0 0.001
STAGE 1 8 0 0 0 0 1
VOLUM 1 8*0.01 8*0.05
RECYC 1 8*0.0
*---BANK-STAGE-NSET-TIME--FLOW--H+---U02--PU4--PU3--U4---HN02--HYD---
FEED 1 -1 1 0.0 0.20 0.598 0.0 15.4 38*0.0
FEED 1 8 1 0.0 0.25 0.2 0.0 0.0 0.0 0.0 0.0 3.2 9.24 33*0.0
*-----
INITI 1, 1, 0, 8*0.2
INITI 1, 7, 0, 8*9.24
*-----
TAUCONST 1 0,40,1.0
FILE 'fig56'
END
    
```

図5-8 HAN還元Pu逆抽出実験解析の入力データ

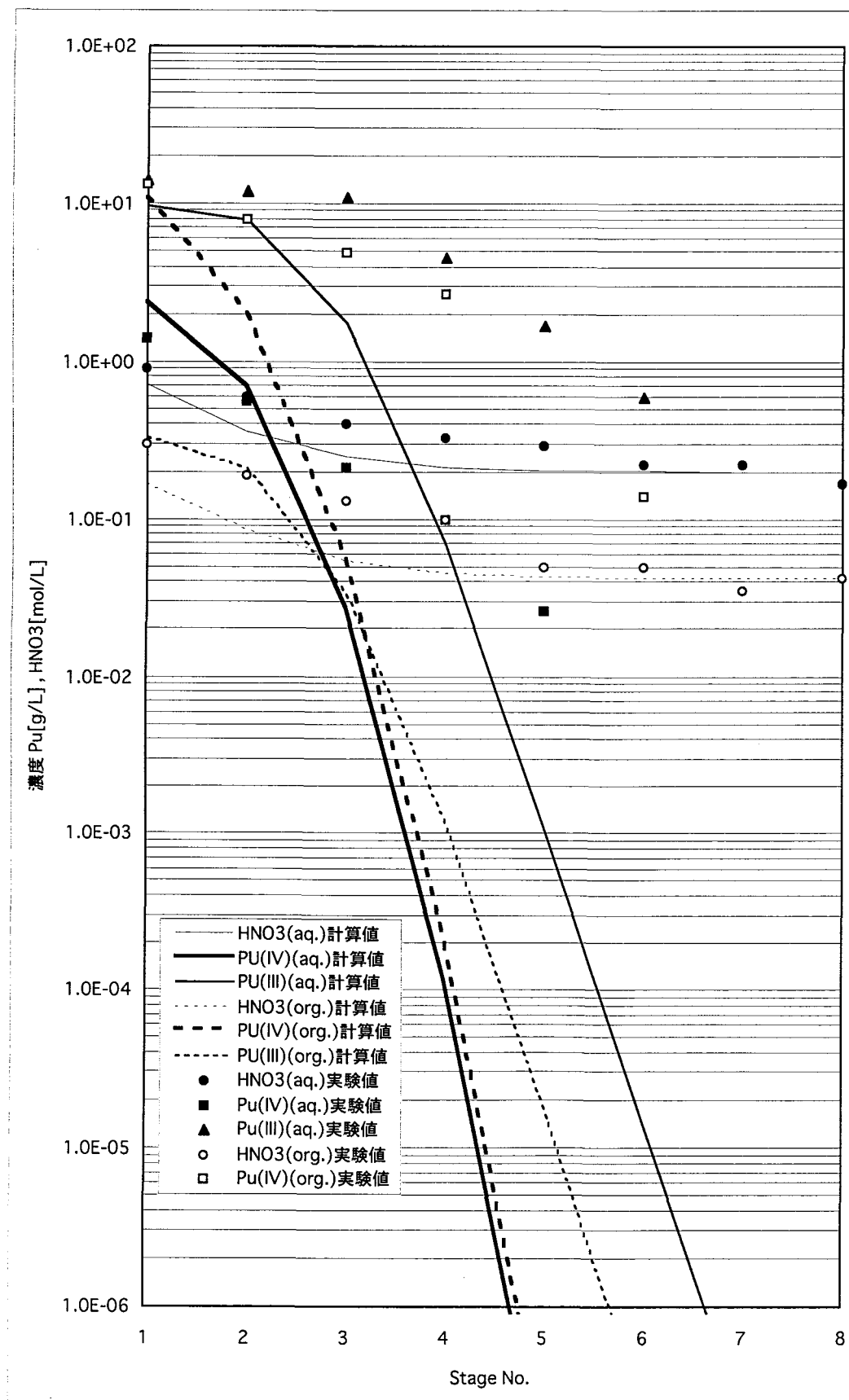


図5-9 HAN還元Pu逆抽出実験解析(MIXSET)

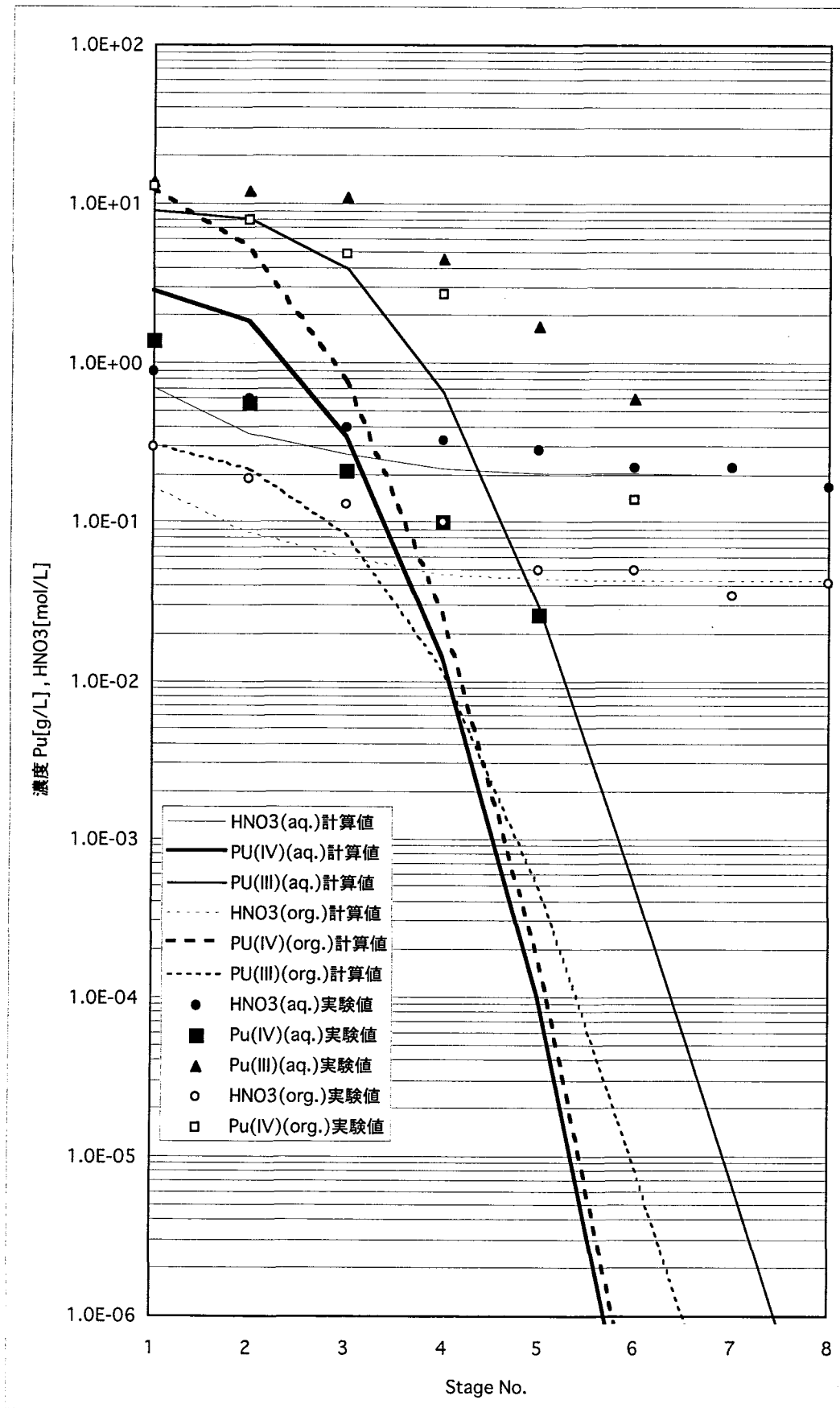


図5-10 HAN還元Pu逆抽出実験解析(MIXSET)(容積1/10)

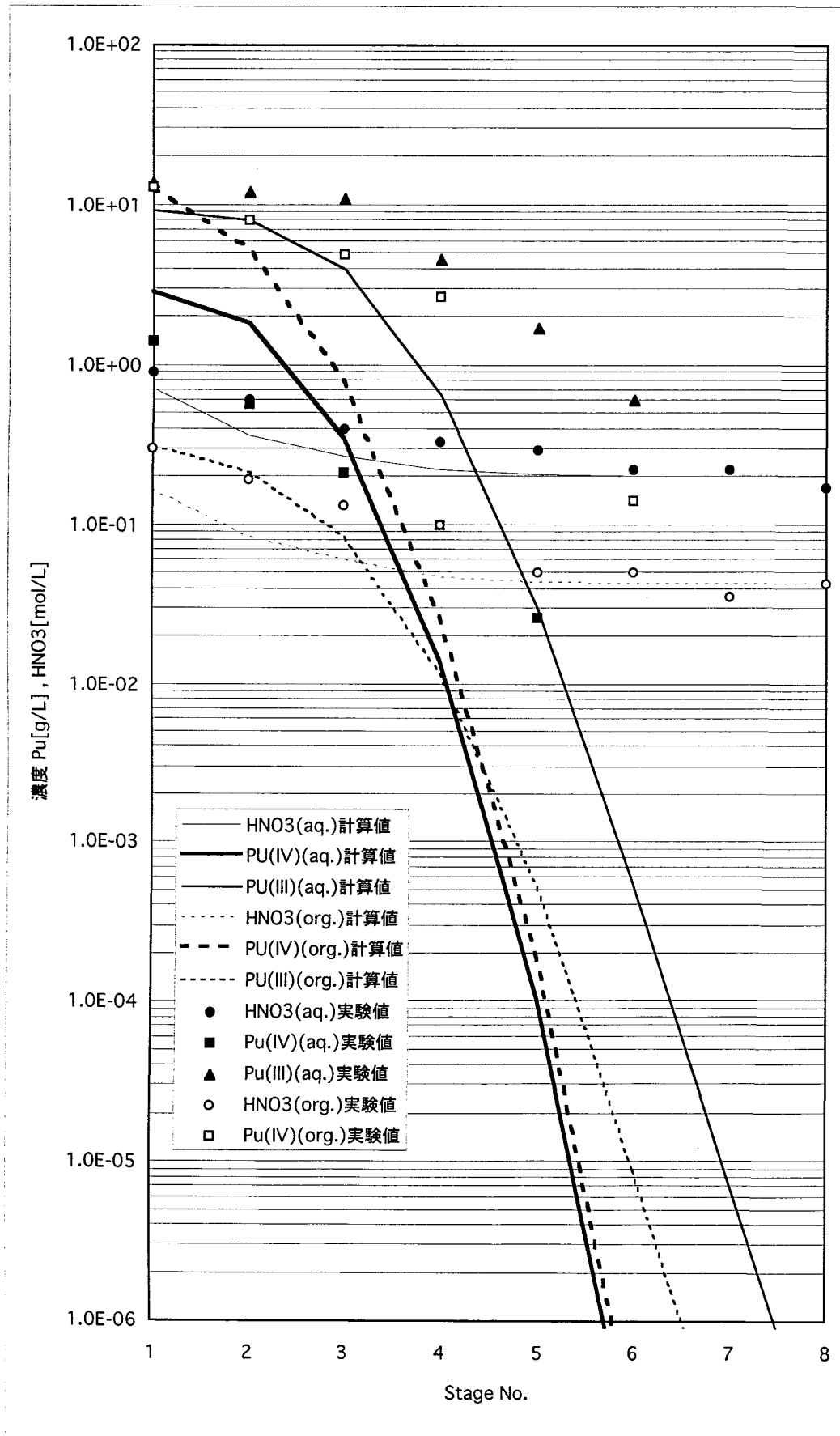


図5-11 HAN還元Pu逆抽出実験解析(MIXSET)(反応速度1/10)

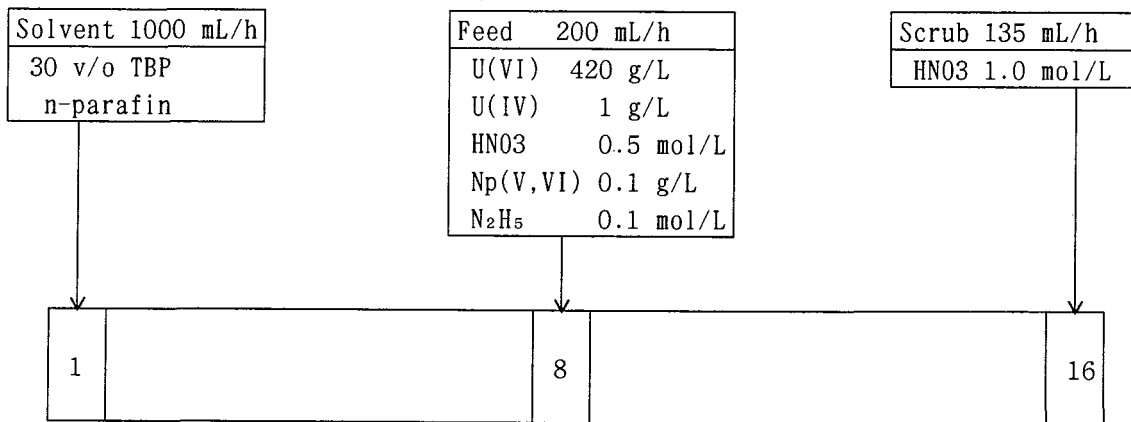


図 5 - 1 2 N p 抽出実験のフローシート

```

file fig58.inp  JAERI 1331  Fig 5-8 Experiment(Eurochemic Plant)
CONTR 0 0 8.0 0.01
STAGE 1 16 0 0 0 0 1
VOLUM 1 16*0.03 16*0.15
RECYC 1 16*0.0
*-----BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HNO2--HYD--
FEED 1 -1 1 0.0 1.00 0.0 40*0.0
FEED 1 8 1 0.0 0.20 3.0 420.0 0.0 0.0 1.0 0.0 3.2 0.0
      0.0 0.0 0.0 0.05 0.05 28*0.0
FEED 1 16 1 0.0 0.135 1.0 40*0.0
*-----
INITI 1, 1, 0, 16*1.0
*-----
TAULIM 0.01 0.04 1.2
FILE 'fig58'
END
    
```

図 5 - 1 3 N p 抽出実験解析の入力データ

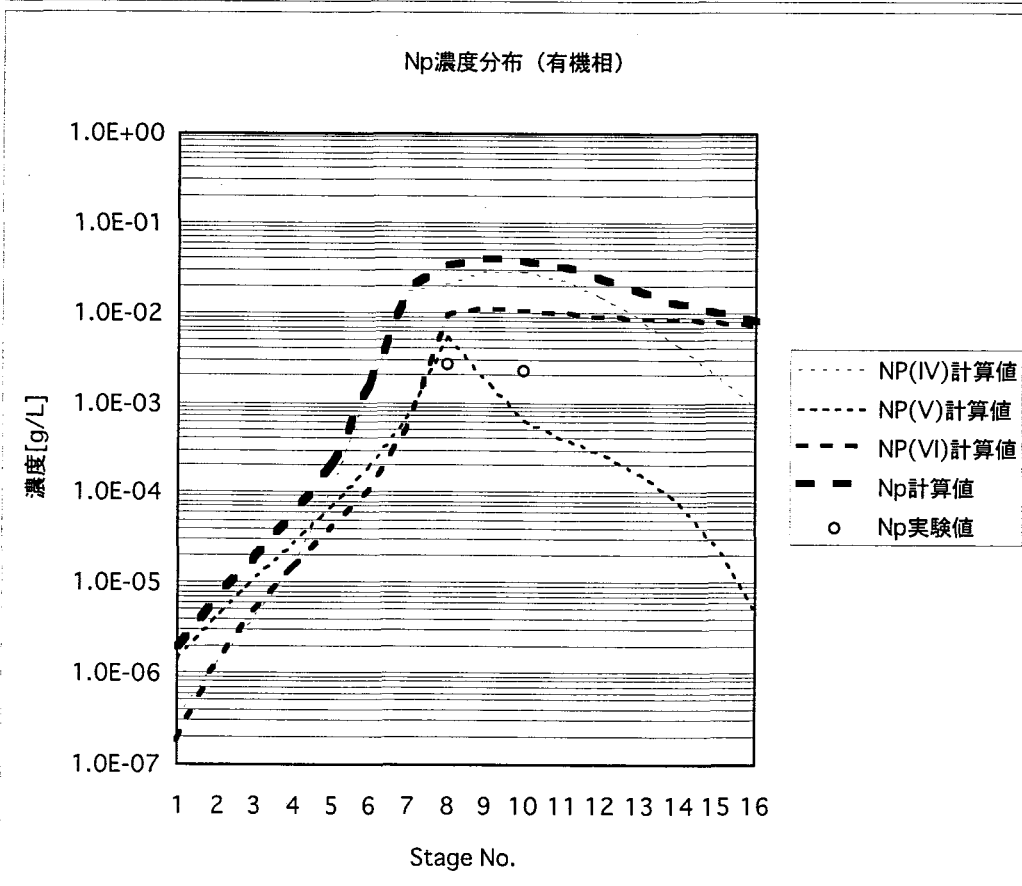
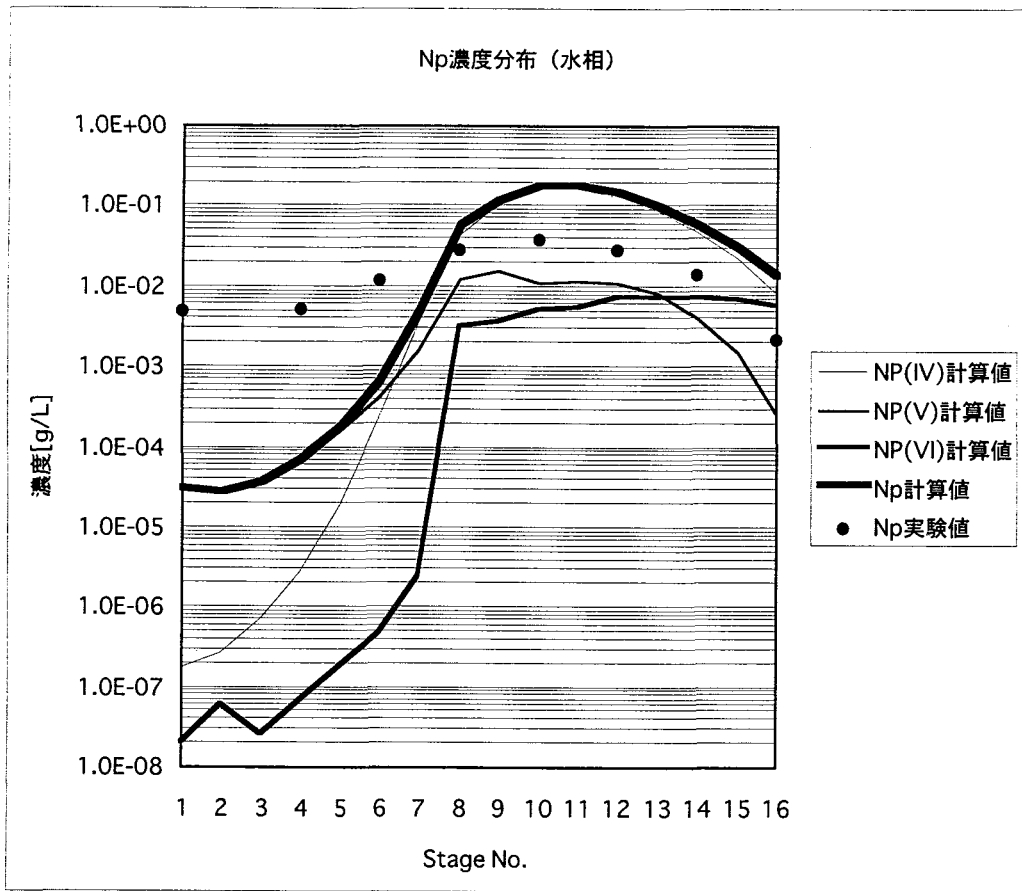


図 5 - 1 4 Np抽出実験解析 (MIXSET)

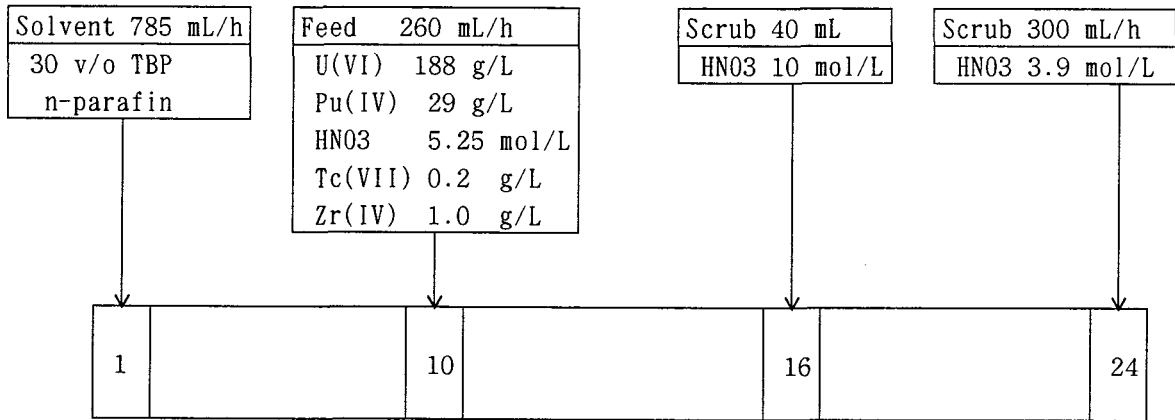


図5-15 Tc抽出実験のフローシート

```

file fig510.inp  JAERI 1331  Fig 5-10 Experiment(KfK)
CONTR 0 0 20.0  0.01
STAGE 1 24 0 0 0 0 1
VOLUM 1 24*0.03  24*0.15
RECYC 1 24*0.0
*----BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--
FEED  1 -1 1 0.0 0.785 0.0                                     40*0.0
FEED  1 10 1 0.0 0.260 5.25 188.0 29.0      0.0    0.0  0.0 0.0 0.0
          0.0 0.0  0.0  0.0    0.0  1.0
          0.0 0.0  0.0  0.2                                     23*0.0
FEED  1 16 1 0.0 0.040 10.0                                   40*0.0
FEED  1 24 1 0.0 0.300 3.9                                    40*0.0
*-----
INITI  1, 1, 0, 24*1.0
*-----
TAUCONST 1 0,40,1.0
FILE  'fig510'
END
    
```

図5-16 Tc抽出実験解析の入力データ

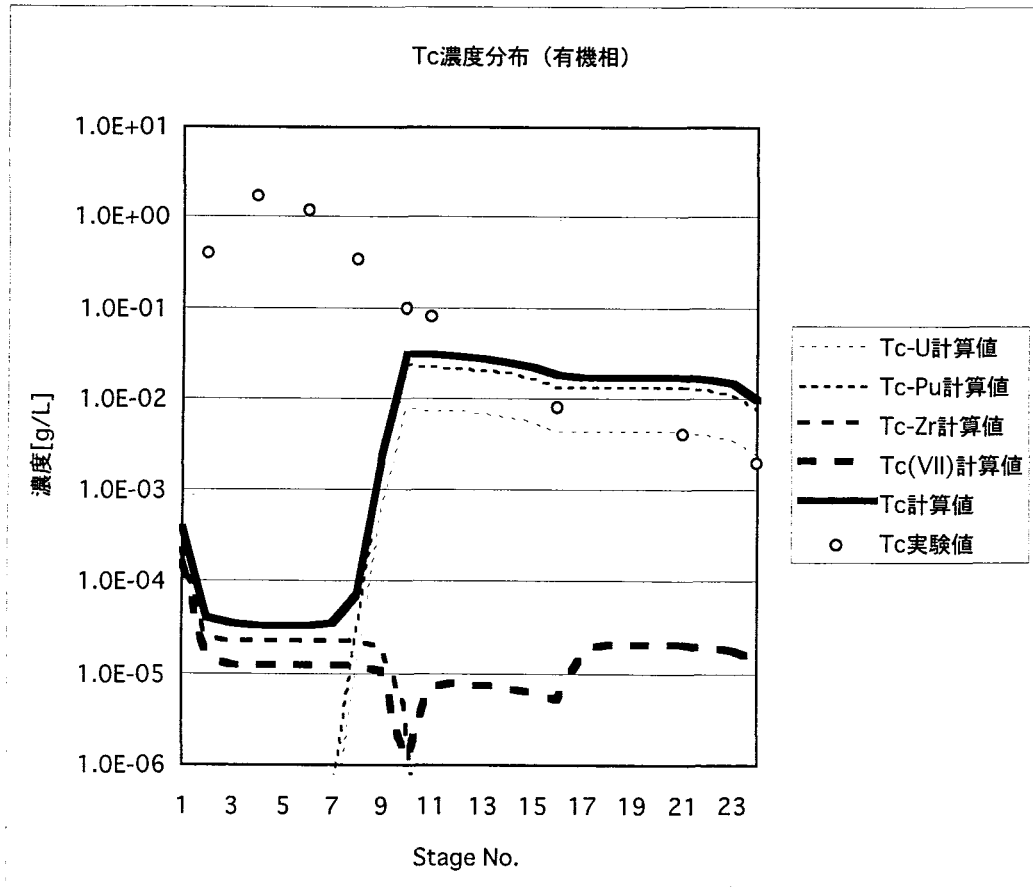
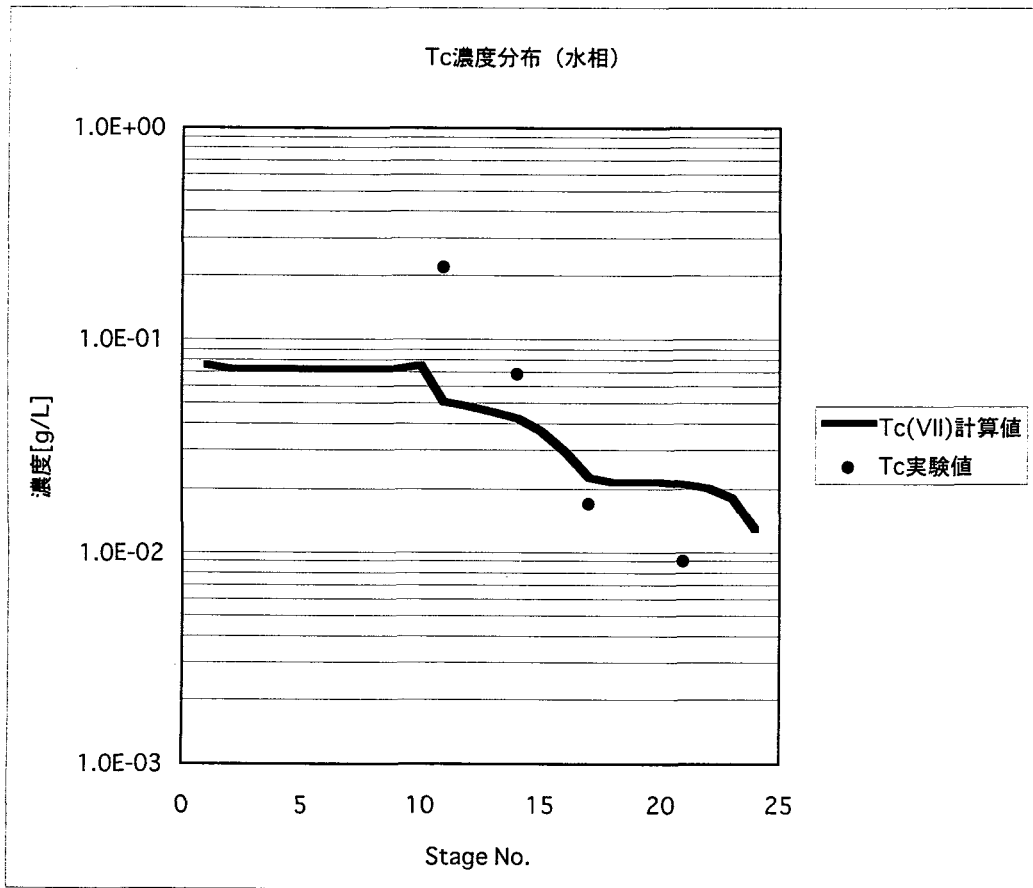


図5-17 Tc抽出実験解析 (MIXSET)

5. 2 再処理工場のNp, Tcの分配の計算

MIXSET-Xを用いて、燃料溶解液フィードにTRU, FPを加えて第1抽出器から第9抽出器にいたる再処理工場のプロセスの解析計算を行い、再処理工場においてNpとTcがどのように分配していくかを計算した。付録Kに再処理工場のプロセスの模式図と計算用の入力データリストを示す。ただし、第1抽出器のフィード条件は表5-1に示す。計算結果を表5-2, 表5-3に示す。表において左端の数字は抽出器の番号を表し、記号Aは水相、記号Oは有機相を表す。濃度は各相の出口濃度である。今回の計算条件ではNpはほぼ全量がプルトニウム製品側に移行し、Tcはほとんど除去される結果となった。計算されたNpとTcの挙動を以下にまとめる。

(1) ネプツニウム

Np(V)として供給されたネプツニウムは、第1抽出器でほぼ全量が酸化されNp(VI)として有機相に抽出される。第2抽出器では、硝酸濃度が低いところで起こるPu(IV)がPu(III)とPu(V)になる不均化反応によって生じるPu(III)によって還元されてNp(V)として水相に逆抽出される。有機相には供給量の0.2%がNp(VI)として溶媒再生工程に同伴する。第3抽出器でもほぼ全量がNp(VI)として有機相に抽出される。第4抽出器では、Pu(IV)の還元で使用されるウラナスによりNpも還元され、Np(IV)となる。Np(IV)は本来抽出性を持ち有機相に抽出される可能性があるが、計算ではPu(III)と共に水相に分配されている。第4抽出器のNp量は、第9抽出器からの溶媒に同伴してNpがリサイクルされるので、水相からの供給量よりも増加している。第4抽出器の有機相は第5抽出器に供給されウランが逆抽出される。そのさい約0.74%のNpがNp(IV)として同伴する。このウランに同伴したNpは第6抽出器で洗浄され、廃液に同伴し、ウラン精製工程製品に同伴する量は約0.01%である。第8抽出器では廃液にNp(IV)として、抽出溶媒にはNp(VI)として同伴し、第9抽出器ではウラナスによりPuが還元されるに伴い、Npも還元され、Np(V)およびNp(IV)としてプルトニウム製品にNp供給量の約91%が同伴する。第9抽出器から第4抽出器にリサイクルされる溶媒にはNp(IV)として約4%がリサイクルされ、第4抽出器で処理するNp量が若干増加する。以上を整理すると、ネプツニウム供給量の約0.01%がウラン製品に、約91%がプルトニウム製品に同伴し、9%が廃液に移行する。

(2) テクネチウム

Tc(VII)として供給されたテクネチウムは、第1抽出器で約69%が有機相に抽出され、約31%が廃液に移行する。第2抽出器ではTc(VII)として水相に逆抽出される。第3抽出器ではTcの約4%が有機相に抽出され、約26%が廃液に移行する。この段階で約95%のTcが除去されている。残りのTcは、第4抽出器で、ウラナスによりPuが還元されるに伴い、Tcも一部Tc(IV)まで還元される。そして、水相および有機相（主にTc-Uの錯体として）にほぼ同じ量のTcが分配される。第4抽出器の有機相は第5抽出器に供給されウランが逆抽出される。その際、約2.7%のTcが同伴する。このウランに同伴したTcは第6抽出器で洗浄され、約1.7%が廃液に同伴し、ウラン精製工程製品に同伴する量は約1%である。第8抽出器では供給されたTcのほとんどが廃液に同伴し、抽出溶媒には約0.04%が同伴する。第9抽出器ではプルトニウム製品にTc供給量の約0.02%が同伴する。第9抽出器から第4抽出器にリサイクルされる溶媒にはTcの約0.02%がリサイクルされる。以上を整理すると、テクネチウム供給量の約1%がウラン製品に、約0.02%がプルトニウム製品に同伴し、ほとんどのTcは廃液に移行する。

表5-1 第1抽出器の計算条件

段数	17			
容積	ミキサ部	12.0	L	
	セトラ部	1~9段	74.1	L
		10~17段	93.8	L
循環流量		10~17段	198.0	L/h
温度		1~9段	25.0	°C
		10~17段	30.0	°C
フィード条件				
1段	有機相	300.0	L/h	有効成分なし
9段	水相	155.0	L/h	成分
				硝酸 3.0 mol/L
				U(VI) 180.0 g/L
				Pu(IV) 1.4 g/L
				Np(IV) 0.0 g/L
				Np(V) 2.38×10^{-2} g/L
				Np(VI) 0.0 g/L
				Zr(IV) 7.24×10^{-2} g/L
				Tc(VII) 1.00×10^{-2} g/L
				Sr(II) 3.50×10^{-2} g/L
				Cs(I) 7.29×10^{-2} g/L
				Ce(III) 6.19×10^{-7} g/L
				Gd(III) 0.0 g/L
				Am(III) 2.75×10^{-2} g/L
				Cm(III) 7.81×10^{-5} g/L
				Ru(tot) 2.27×10^{-6} g/L
17段	水相	66.5	L/h	成分
				硝酸 3.0 mol/L

表5-2 再処理工場のNpの分配計算結果

	Flow (L/Hr)	Np(IV) (g/L)	Np(V) (g/L)	Np(VI) (g/L)	Np(total) (g/Hr)	Ratio
FEED	1.55E+02	0.00E+00	2.38E-02	0.00E+00	3.69E+00	100.00%
EXIT						
1A	2.22E+02	3.07E-14	1.05E-07	6.94E-08	3.86E-05	0.00%
10	3.00E+02	3.37E-13	6.38E-08	1.23E-02	3.69E+00	100.00%
2A	3.73E+02	4.29E-09	8.98E-03	8.84E-04	3.68E+00	99.80%
20	3.00E+02	4.05E-24	4.06E-18	2.42E-05	7.27E-03	0.20%
3A	5.51E+02	5.78E-13	1.54E-08	1.57E-08	1.71E-05	0.00%
30	3.70E+02	6.70E-08	3.45E-06	9.95E-03	3.68E+00	99.80%
4A	1.31E+02	2.89E-02	1.66E-04	3.12E-10	3.80E+00	103.01%
40	4.53E+02	6.02E-05	9.39E-10	7.07E-09	2.72E-02	0.74%
5A	5.60E+02	4.86E-05	7.77E-10	4.83E-09	2.72E-02	0.74%
50	4.53E+02	6.51E-08	1.27E-17	1.08E-09	2.99E-05	0.00%
6A	7.33E+02	3.67E-05	3.92E-10	4.87E-11	2.69E-02	0.73%
60	4.00E+02	7.52E-07	2.91E-11	3.74E-09	3.02E-04	0.01%
7A	5.60E+02	5.36E-07	1.75E-12	2.35E-09	3.02E-04	0.01%
70	4.00E+02	1.51E-09	3.87E-20	4.28E-10	7.77E-07	0.00%
8A	1.92E+02	1.52E-03	3.87E-05	3.13E-05	3.05E-01	8.26%
80	4.10E+01	3.93E-04	4.44E-05	8.48E-02	3.50E+00	94.75%
9A	1.70E+01	1.76E-03	1.95E-01	7.68E-10	3.35E+00	90.80%
90	5.25E+01	2.77E-03	2.42E-12	5.44E-12	1.46E-01	3.95%

表5-3 再処理工場のTcの分配計算結果

	Flow (L/Hr)	Tc(IV) Tc-U (g/L)	Tc(V) Tc-Pu (g/L)	Tc(VI) Tc-Zr (g/L)	Tc(VII) Tc(VII) (g/L)	Tc(total) (g/Hr)	Ratio
FEED	1.55E+02	0.00E+00	0.00E+00	0.00E+00	1.00E-02	1.55E+00	100.00%
EXIT							
1A	2.22E+02	1.31E-13	1.08E-12	1.28E-13	4.80E-03	1.06E+00	68.55%
10	3.00E+02	1.40E-03	2.17E-04	9.02E-25	6.70E-06	4.87E-01	31.45%
2A	3.73E+02	4.02E-12	5.72E-12	1.53E-13	1.31E-03	4.87E-01	31.45%
20	3.00E+02	2.72E-18	6.87E-20	2.44E-86	2.31E-11	6.94E-09	0.00%
3A	5.51E+02	5.48E-11	2.88E-10	6.33E-12	7.31E-04	4.03E-01	26.00%
30	3.70E+02	1.95E-04	2.97E-05	3.91E-40	3.25E-06	8.45E-02	5.45%
4A	1.31E+02	2.31E-04	6.64E-05	6.76E-09	3.44E-05	4.35E-02	2.80%
40	4.53E+02	9.06E-05	5.61E-17	3.00E-54	8.49E-07	4.14E-02	2.67%
5A	5.60E+02	5.56E-13	1.43E-13	1.31E-16	7.39E-05	4.14E-02	2.67%
50	4.53E+02	3.60E-21	0.00E+00	0.00E+00	4.35E-14	1.97E-11	0.00%
6A	7.33E+02	8.94E-08	3.86E-07	2.10E-10	3.45E-05	2.57E-02	1.66%
60	4.00E+02	3.92E-05	4.96E-22	3.81E-39	1.44E-07	1.57E-02	1.02%
7A	5.60E+02	5.80E-11	6.08E-11	1.66E-14	2.81E-05	1.57E-02	1.02%
70	4.00E+02	3.43E-17	4.68E-45	2.23E-86	1.63E-12	6.54E-10	0.00%
8A	1.92E+02	3.28E-10	9.71E-07	4.79E-09	2.22E-04	4.29E-02	2.77%
80	4.10E+01	1.73E-06	1.09E-05	1.13E-48	1.45E-06	5.77E-04	0.04%
9A	1.70E+01	7.71E-06	2.03E-06	1.34E-10	4.09E-06	2.35E-04	0.02%
90	5.25E+01	5.46E-06	2.43E-15	4.60E-61	1.06E-06	3.42E-04	0.02%

5.3 Np、Tc、Zrの存在によるフリーTBP減少の抽出特性への影響

分配係数の計算のときに使用されるフリーTBPは、全TBPから各溶質成分に配位されたTBPを除くことによって計算される。従来の計算方法では、 HNO_3 、 U(VI) 、 U(IV) 、 Pu(IV) 、 Pu(III) と HNO_2 の6成分に配位されたTBPのみを全TBPより除いてフリーTBPが計算されていた。MIXSET-Xでは以上の6成分に配位されるTBPに加えて、Np、Tc、Zrに配位されるTBPも全TBPより除いてフリーTBPが計算される。

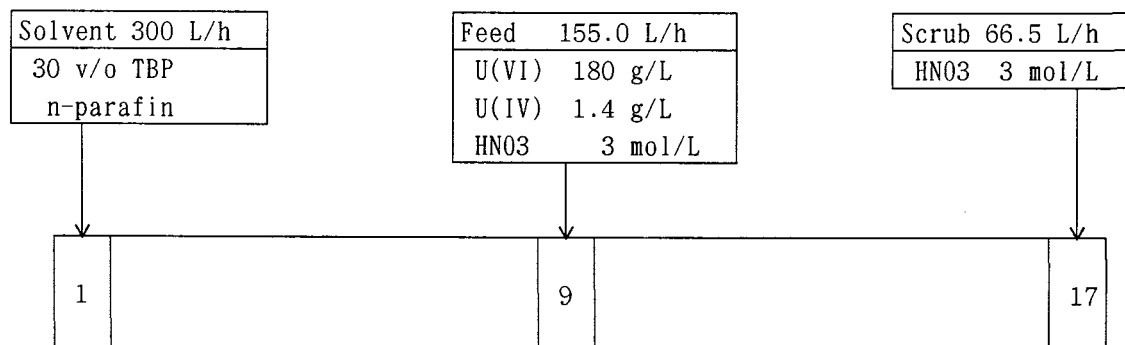
そこで、MIXSET-Xを用いてパラメータサーベイ計算を行うことにより、燃料溶解液中に含まれているNp、Tc、Zrを原因とするフリーTBPの減少が抽出特性にどの程度影響を与えるかを評価した。

まず、動燃東海再処理工場の共除染工程を想定して、図5-18に示す抽出プロセスフローシートを基準のプロセス条件に設定した。このときのMIXSET-Xの入力データを図5-19に示す。このプロセス条件は、フィード流量または有機溶媒流量の1%以内の変動でウランおよびプルトニウムが廃液にリークする厳しい設定になっている。フィード流量を増加させた場合と有機溶媒流量を低下させたときのウランとプルトニウムの濃度分布を図5-20～図5-23に示す。抽出廃液中のウラン濃度とプルトニウム濃度を表5-4と表5-5に示す。いずれの場合にも、流量の1%以内の変動でウランおよびプルトニウムが廃液にリークしている。

このプロセス条件のもとで、フィードにNp、Tc、Zrを添加してウランとプルトニウムの濃度分布がどのように変化するかを計算した。フィードに添加したNp、Tc、Zrの量を表5-6に示す。最大の添加量はウラン燃料の燃焼度にして60GWD/t程度に相当する。計算結果を図5-24、図5-25と表5-7に示す。計算結果では、抽出能力の低下が見られるものの、最大の添加量でもウランまたはプルトニウムが廃液にリークする状態にはなっていない。この計算で興味深いのは、Np、Tc、Zrが少ない場合には、廃液にリークするプルトニウムは3価のプルトニウムであり、この3価のプルトニウムはNp、Tc、Zrの添加とともにしばらく減少し、それに伴いリークする全プルトニウムも減少することである。Np、Tc、Zrの添加量が多くなると3価のプルトニウムは増加に転じ、リークする全プルトニウムも増加に転じる。

その次に、ウラン燃料の燃焼度にして45GWD/t程度に相当するNp、Tc、Zrを添加したとき(ケース6)に有機溶媒流量をどの程度まで減少させるとウランまたはプルトニウムが廃液にリークするかを調べた。図5-26、図5-27と表5-8にNp、Tc、Zrを添加した条件で有機溶媒を減少させた場合の計算結果を示す。計算では有機溶媒流量を0.4L/h低下したところでウランとプルトニウムが廃液にリークする状態になった。Np、Tc、Zrの添加がない場合には有機溶媒流量が0.8L/h低下したところでウランとプルトニウムが廃液にリークする状態になっているので、燃焼度45GWD/tに相当するNp、Tc、Zrの添加は有機溶媒流量の0.4L/h(約0.14%)の低下に相当することになる。

以上の結果をまとめると、燃焼度45GWD/t相当のNp、Tc、Zrを添加しても、それらの成分の存在によるフリーTBPの減少を原因とする抽出能力の低下はわずかであり、フィード流量または有機溶媒流量の変動の影響に比べるとNp、Tc、Zrの存在は抽出能力にはわずかな影響しか与えない。従って、通常の運転条件の範囲内では、Np、Tc、Zrの存在によるフリーTBPの減少は考慮する必要はないと推定される。



ミキサ部容積 12.0 L
 セトラ部容積 74.1 L for Stage 1-9 , 93.8 L for Stage 10-17
 内部循環流量 0.0 L/h for Stage 1-9 , 198.0 L/h for Stage 10-17
 温度 25.0 °C for Stage 1-9 , 30.0 °C for Stage 10-17

図 5 - 1 8 基準抽出プロセスのフローシート

```

EXT01TRU0 --- file ext01tru0.inp
CONTR 0 1 5000.0 1.00
STAGE 1 17 0 0 0 0 1
VOLUM 1 17*12.0 9*74.1 8*93.8
RECYC 1 9*0.0 8*198.0
*---BANK-STAGE-NSET-TIME--FLOW--H+----U02--PU4--PU3--U4---HN02--HYD--
FEED 1 -1 1 0.0 300.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0
FEED 1 9 1 0.0 155.0 3.0 180.0 1.4 0.0 0.0 0.0 0.0 0.0
!* PU(V), PU(VI) *! 0.0 0.0
!* NP(IV),NP(V),NP(VI) *! 0.0 0.0 0.0
!* ZR(IV) *! 0.0
!* TC(IV,V,VI,VII) *! 0.0 0.0 0.0 0.0
!* TC-U,TC-PU,TC-ZR *! 3*0.0
!* SR(II) *! 0.0
!* RU(DI),RU(TR) *! 0.0 0.0
!* RU(NI),RU(MO) *! 0.0 0.0
!* CS(I),CE(III),GD(III) *! 0.0 0.0 0.0
!* AM(III),CM(III) *! 0.0 0.0
!* ADD-MIC(1-9) *! 9*0.0
!* RU(TOTAL) *! 0.0
FEED 1 17 1 0.0 66.5 3.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0
*-----
INITI 1, 1, 0, 17*3.0
*-----
TOFSTG 1 9*25.0 8*30.0
SDIST 1 40*1
*-----
FILE 'ext01tru0'
END
    
```

図 5 - 1 9 基準抽出プロセスの入力データ

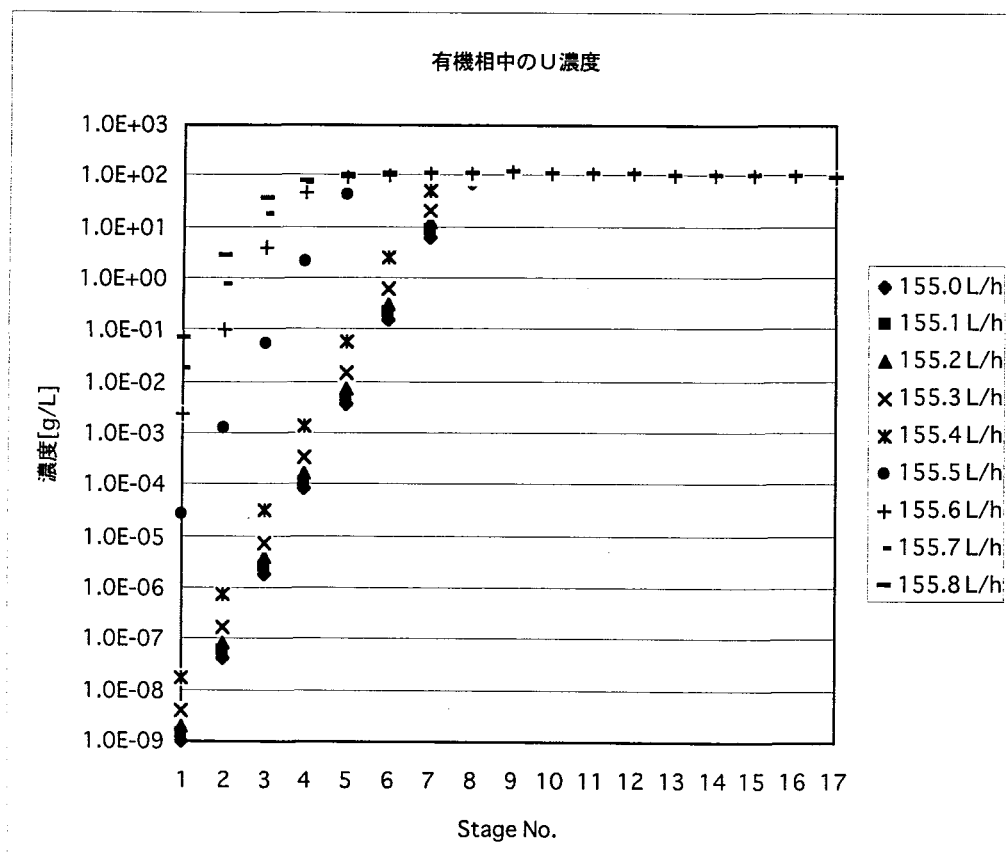
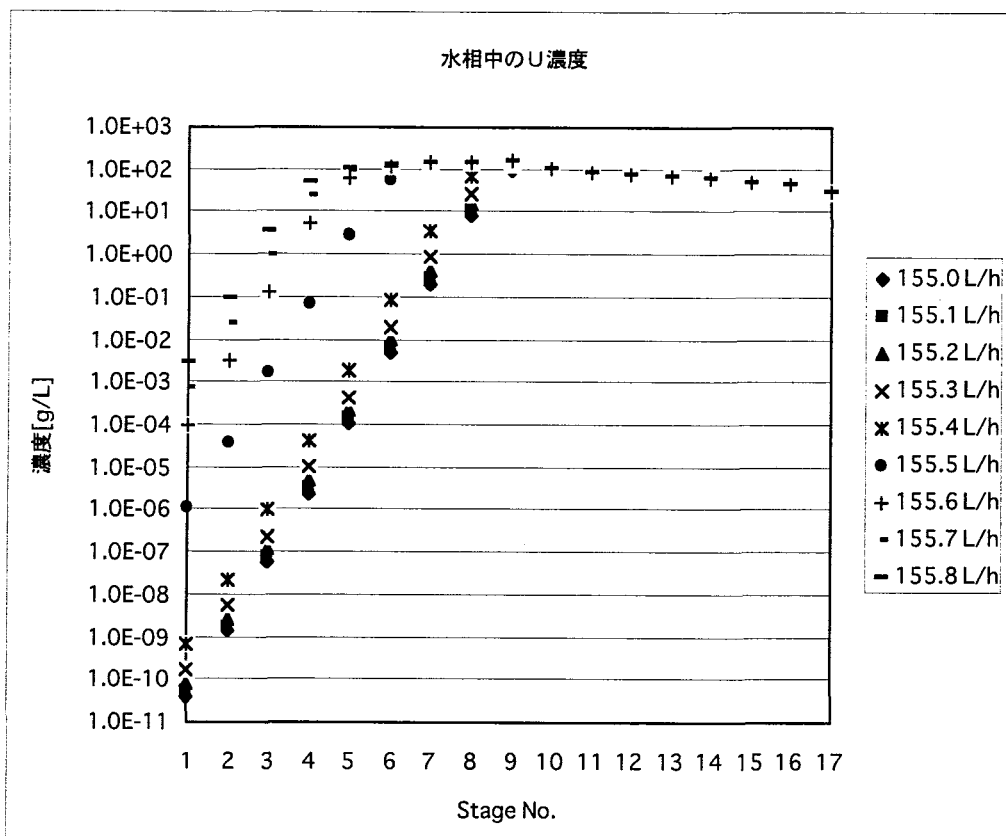


図5-20 フィード流量を増加させたときのU濃度分布変化

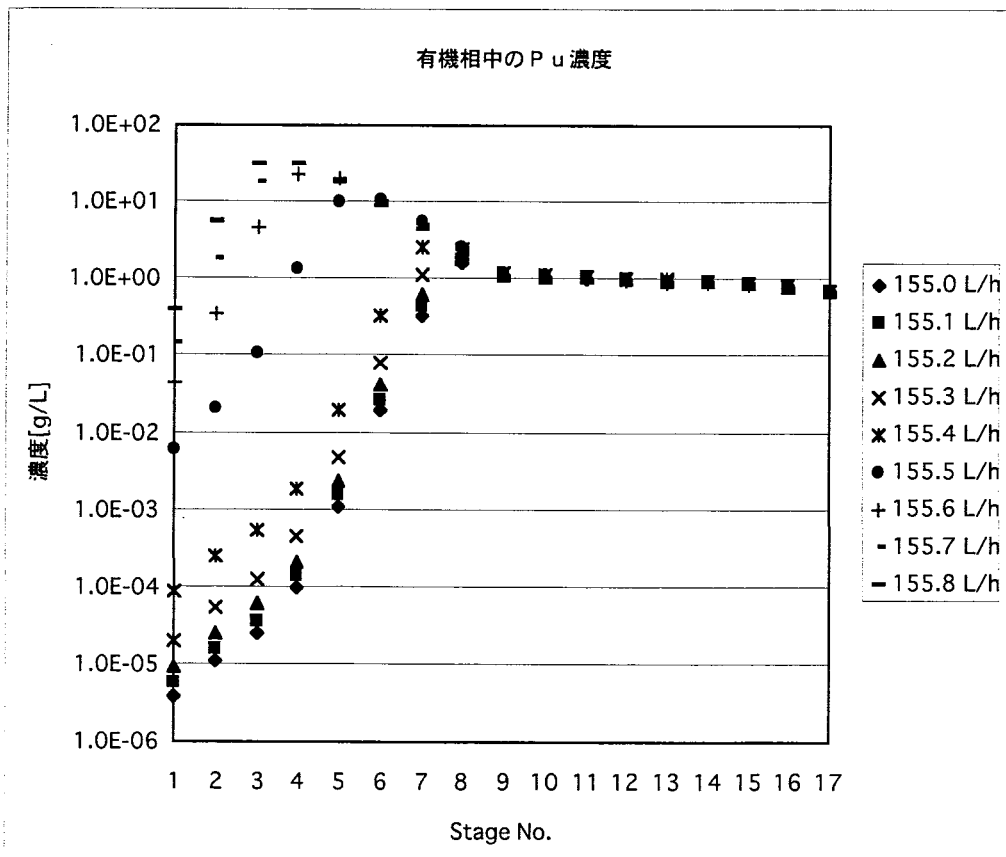
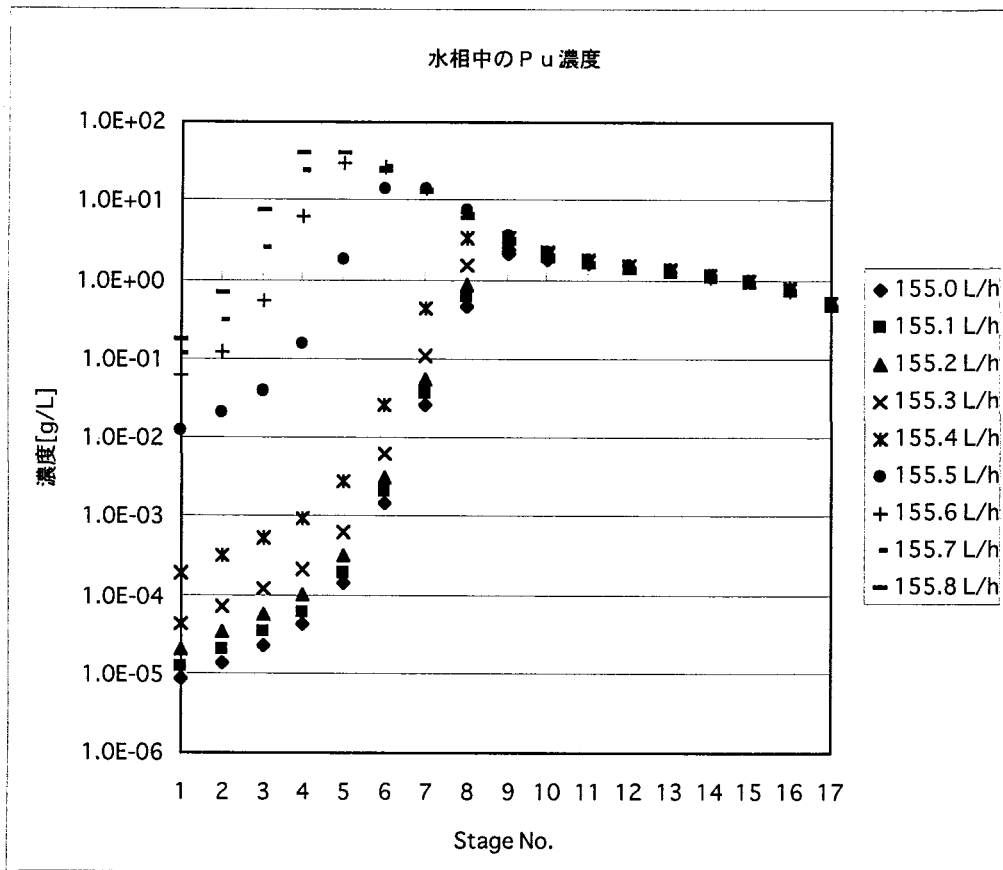


図 5-21 フィード流量を増加させたときの P u 濃度分布変化

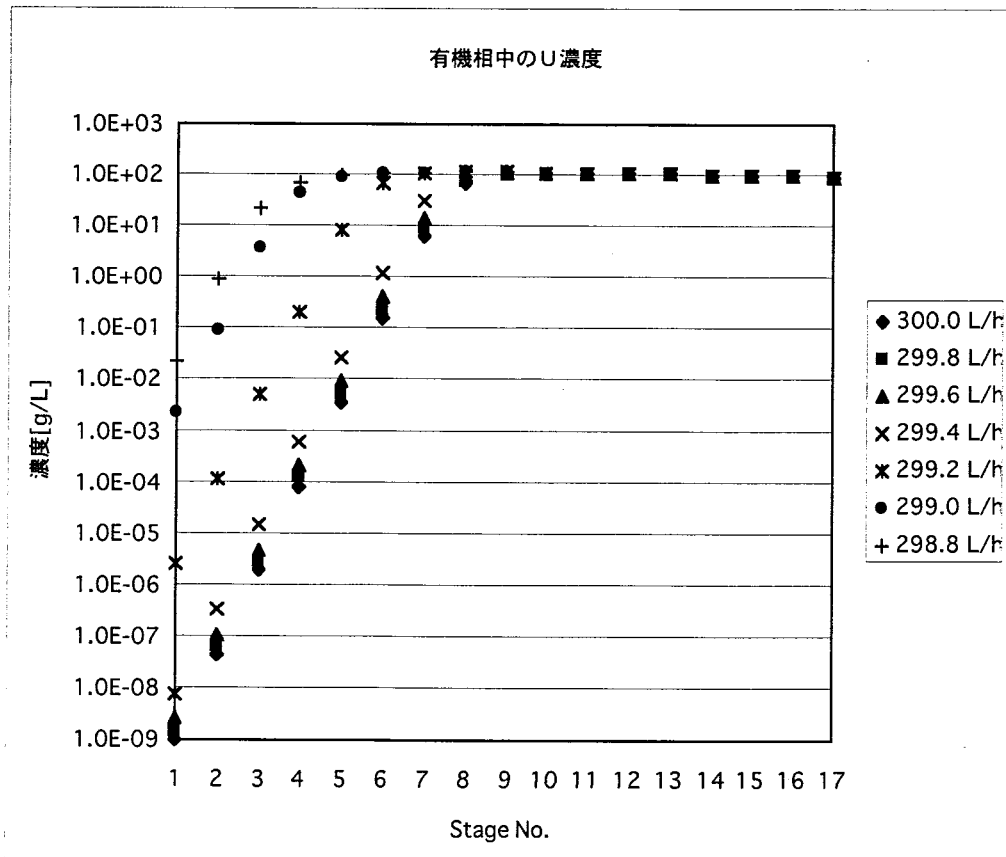
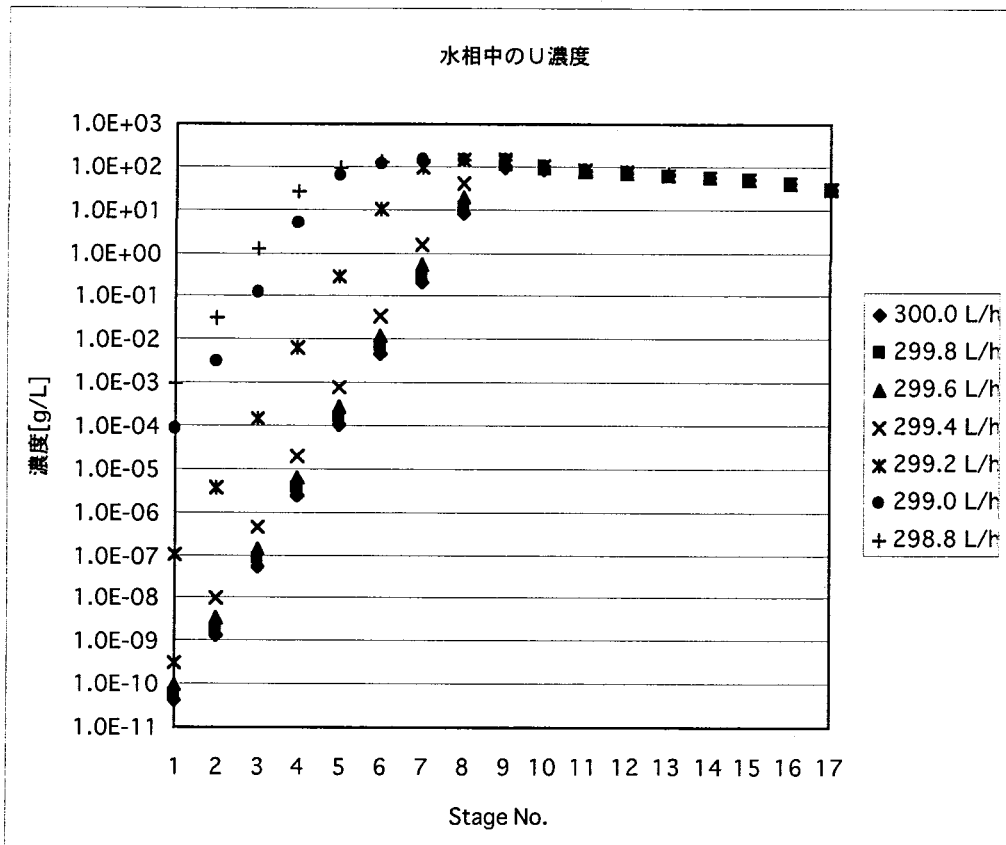


図5-22 有機溶媒流量を減少を増加させたときのU濃度分布変化

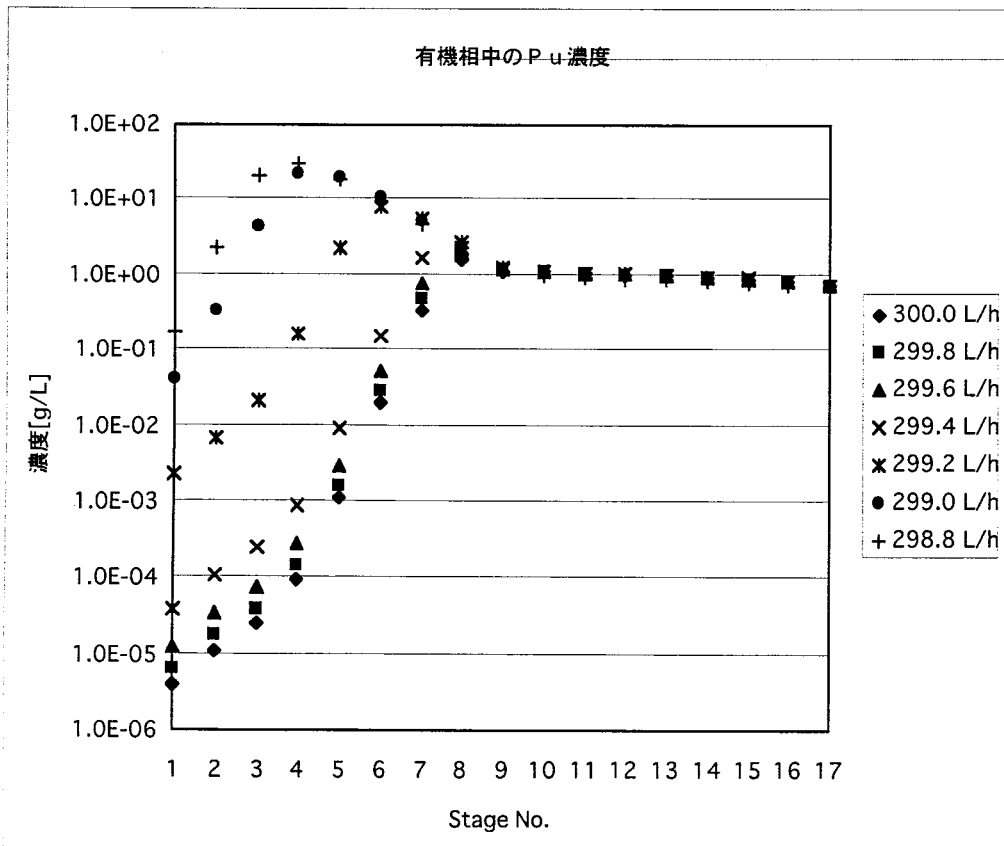
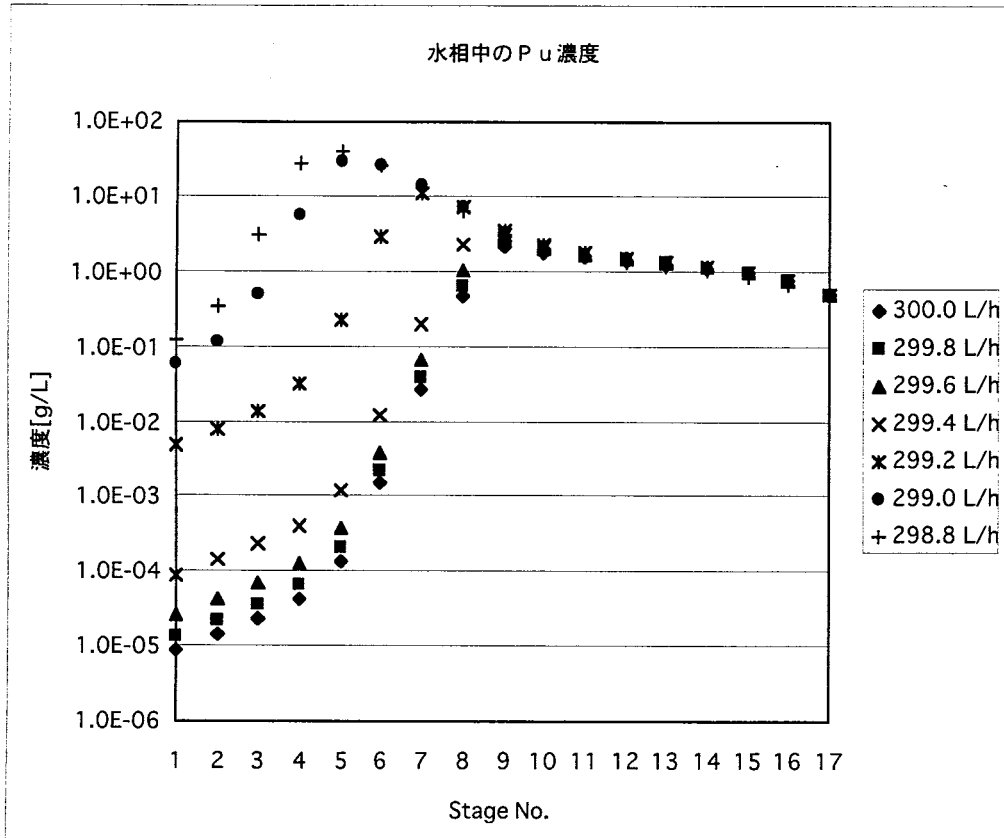


図 5 - 2 3 有機溶媒流量を減少を増加させたときの P u 濃度分布変化

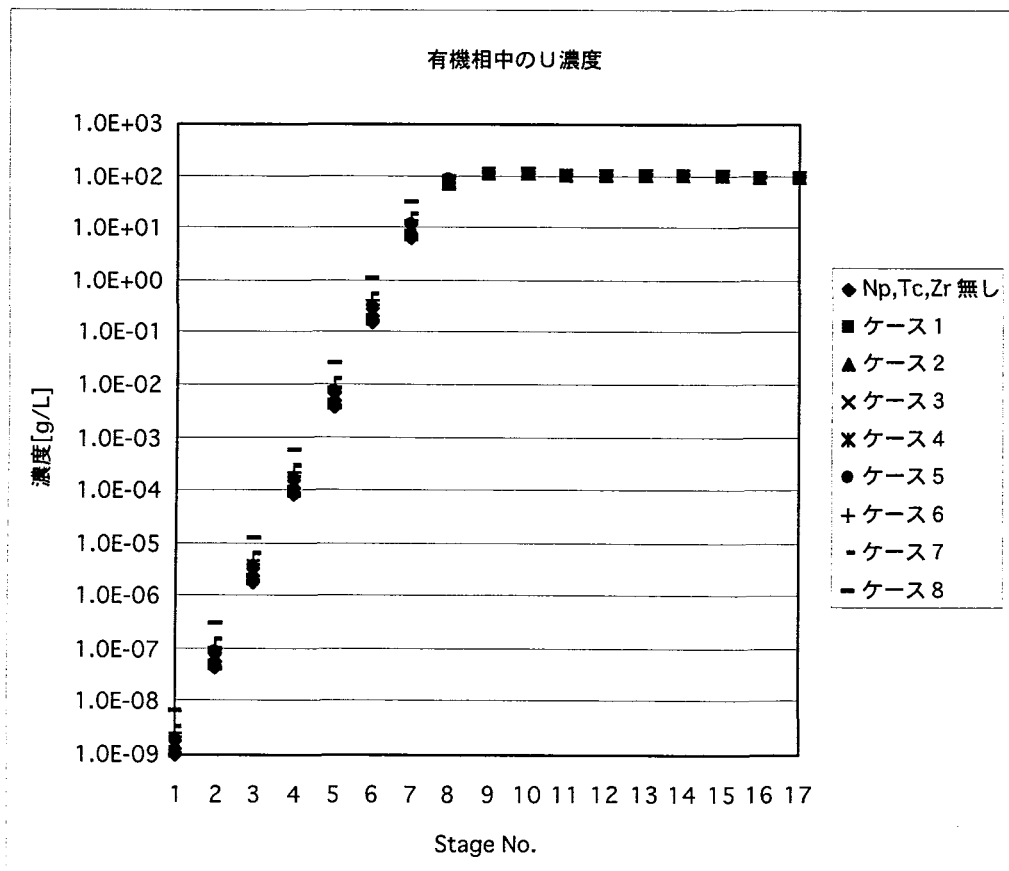
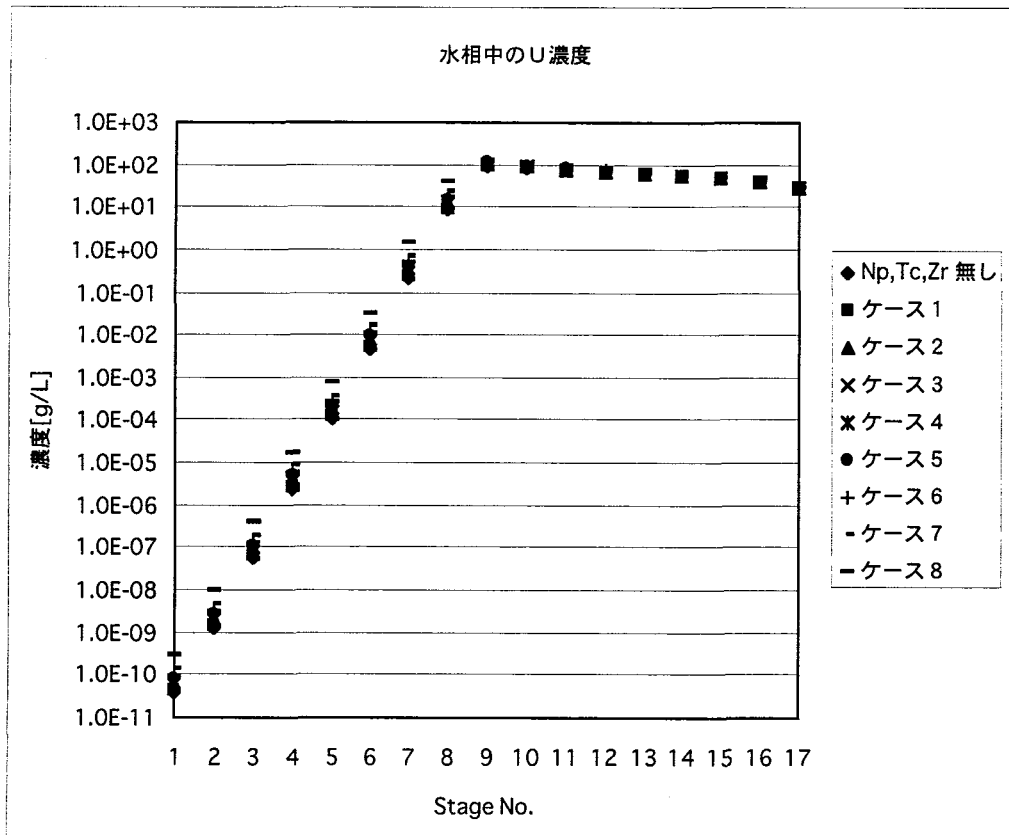


図5-24 Np, Tc, Zrを添加したときのU濃度分布変化

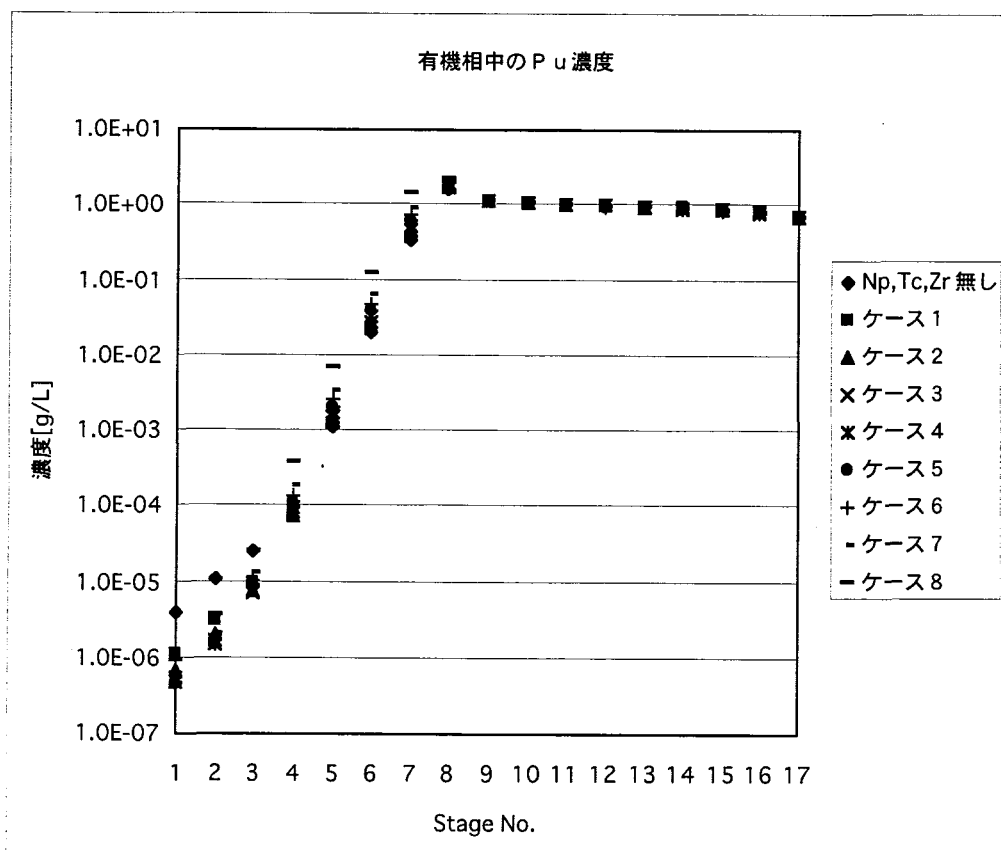
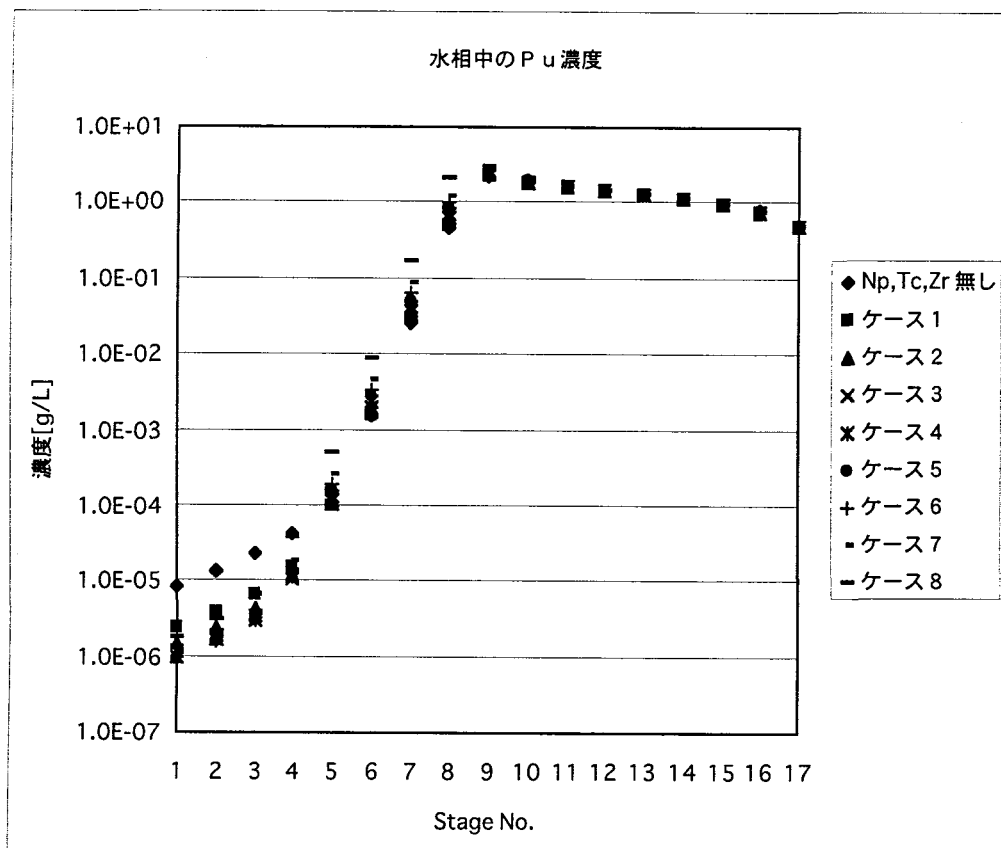


図 5-25 Np, Tc, Zr を添加したときの P u 濃度分布変化

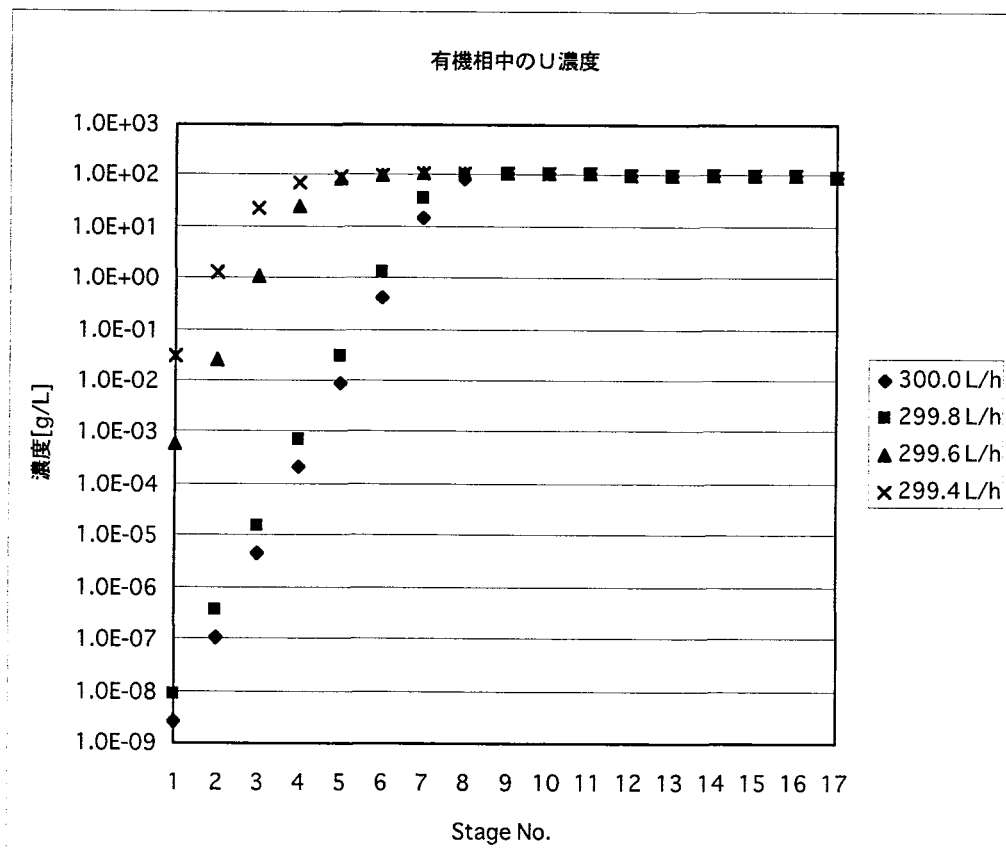
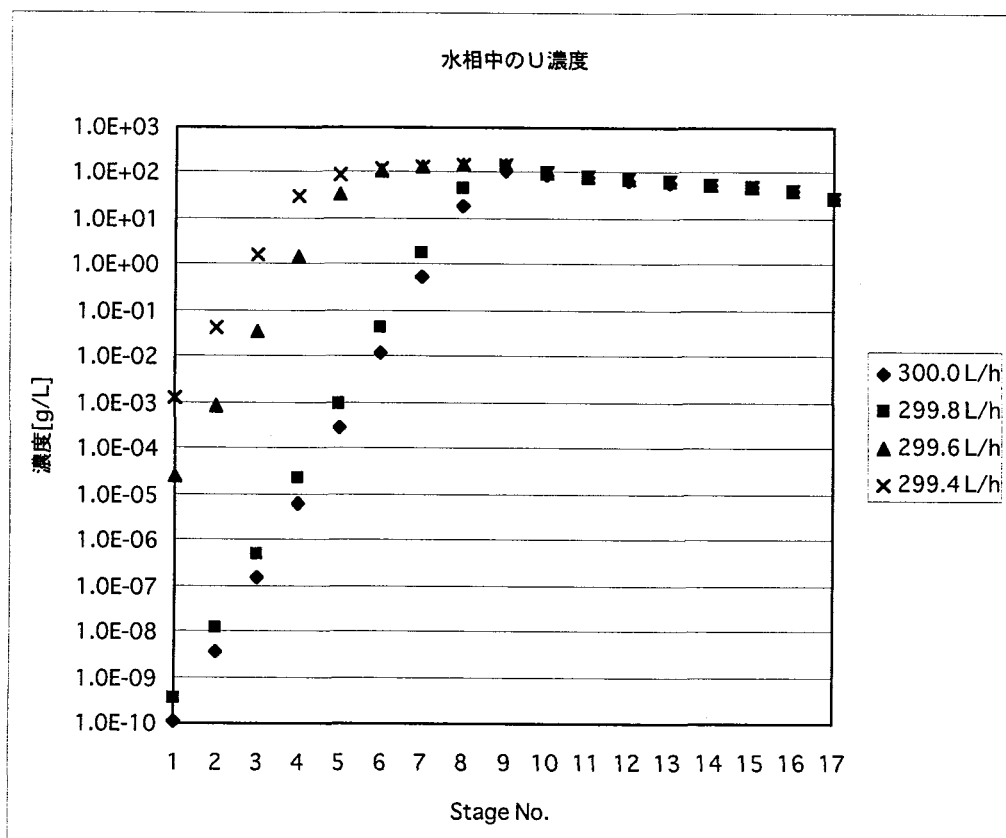


図5-26 有機溶媒流量を減少を増加させたときのU濃度分布変化
(ケース6, 45 GWD/t相当のN_p, T_c, Z_rを添加した場合)

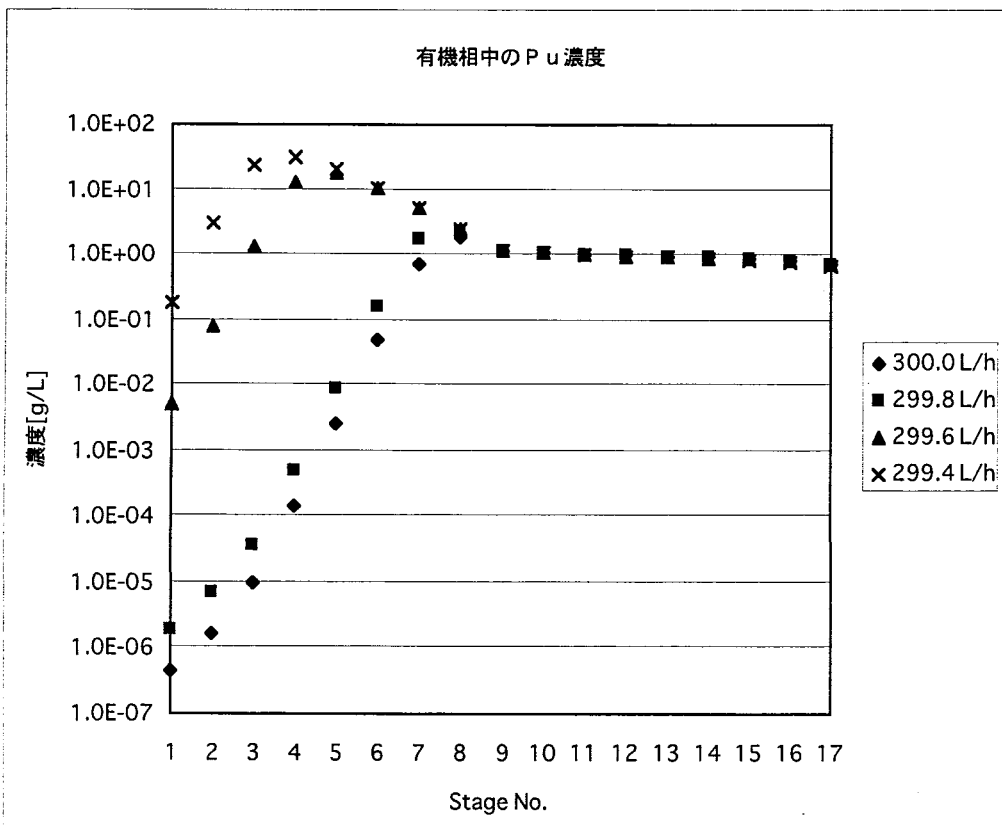
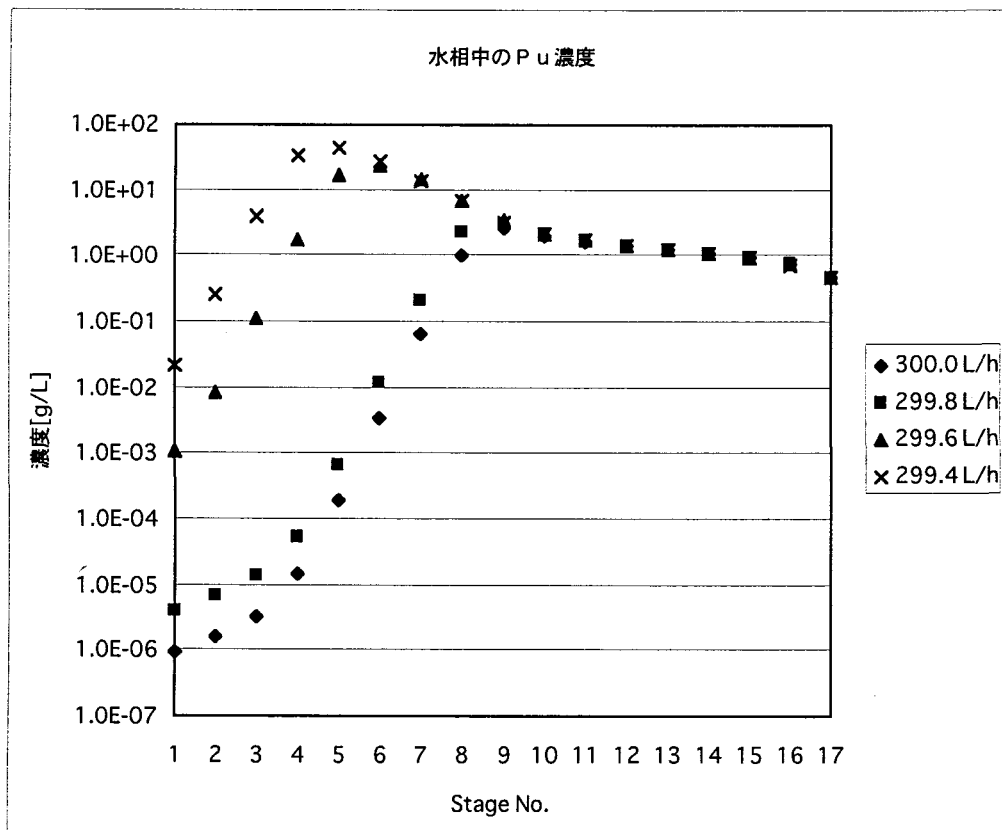


図5-27 有機溶媒流量を減少を増加させたときのPu濃度分布変化
(ケース6, 45 GWD/t相当のNp, Tc, Zrを添加した場合)

表5-4 フィード流量を増加させたときの抽出廃液中のU、Pu濃度

フィード流量 (L/h)	HN03 (MOL/L)	U(total) (G/L)	Pu(total) (G/L)	PU(IV) (G/L)	PU(III) (G/L)	PU(V) (G/L)	PU(VI) (G/L)
155.0	2.74	4.16E-11	8.72E-06	4.02E-07	8.32E-06	1.17E-10	1.66E-09
155.1	2.74	5.64E-11	1.27E-05	5.87E-07	1.21E-05	1.88E-10	2.66E-09
155.2	2.75	8.63E-11	2.10E-05	9.67E-07	2.00E-05	3.49E-10	4.92E-09
155.3	2.75	1.68E-10	4.44E-05	2.04E-06	4.23E-05	8.66E-10	1.21E-08
155.4	2.75	7.14E-10	1.96E-04	9.04E-06	1.87E-04	5.39E-09	7.28E-08
155.5	2.75	1.20E-06	1.28E-02	6.17E-04	1.22E-02	4.27E-06	2.21E-05
155.6	2.75	1.02E-04	6.17E-02	4.53E-03	5.69E-02	6.63E-05	2.01E-04
155.7	2.75	7.58E-04	1.14E-01	1.56E-02	9.81E-02	2.00E-04	5.00E-04
155.8	2.75	3.01E-03	1.75E-01	4.21E-02	1.32E-01	4.12E-04	9.54E-04

表5-5 有機溶媒流量を減少させたときの抽出廃液中のU、Pu濃度

有機溶媒流量 (L/h)	HN03 (MOL/L)	U(total) (G/L)	Pu(total) (G/L)	PU(IV) (G/L)	PU(III) (G/L)	PU(V) (G/L)	PU(VI) (G/L)
300.0	2.74	4.16E-11	8.72E-06	4.02E-07	8.32E-06	1.17E-10	1.66E-09
299.8	2.74	6.05E-11	1.38E-05	6.37E-07	1.32E-05	2.09E-10	2.95E-09
299.6	2.75	1.07E-10	2.68E-05	1.23E-06	2.55E-05	4.71E-10	6.63E-09
299.4	2.75	3.21E-10	8.76E-05	4.04E-06	8.36E-05	1.98E-09	2.76E-08
299.2	2.75	1.10E-07	4.91E-03	2.29E-04	4.67E-03	7.82E-07	5.79E-06
299.0	2.75	9.28E-05	5.97E-02	4.32E-03	5.51E-02	6.29E-05	1.93E-04
298.8	2.75	9.74E-04	1.23E-01	1.86E-02	1.04E-01	2.29E-04	5.62E-04

表5-6 添加するNp, Tc, Zrの量
(ウラン180g/Lに対する添加量)

ケース名	Np (g/L)	Tc (g/L)	Zr (g/L)
ケース1	0.02	0.04	0.004
ケース2	0.04	0.08	0.008
ケース3	0.06	0.12	0.012
ケース4	0.08	0.16	0.016
ケース5	0.10	0.20	0.020
ケース6	0.12	0.24	0.024
ケース7	0.14	0.28	0.028
ケース8	0.16	0.32	0.032

今回の計算ではNpはすべて5価, Tcはすべて7価,
Zrは4価とした。

表5-7 Np, Tc, Zrを添加したときの抽出廃液中のU, Pu濃度

計算ケース	HN03 (MOL/L)	U(total) (G/L)	Pu(total) (G/L)	PU(IV) (G/L)	PU(III) (G/L)	PU(V) (G/L)	PU(VI) (G/L)
標準(添加無し)	2.74	4.16E-11	8.72E-06	4.02E-07	8.32E-06	1.17E-10	1.66E-09
ケース1	2.74	4.63E-11	2.41E-06	1.12E-07	2.30E-06	1.36E-10	1.93E-09
ケース2	2.74	5.22E-11	1.48E-06	6.89E-08	1.41E-06	1.60E-10	2.27E-09
ケース3	2.74	5.99E-11	1.14E-06	5.31E-08	1.08E-06	1.93E-10	2.73E-09
ケース4	2.74	7.02E-11	9.87E-07	4.62E-08	9.37E-07	2.38E-10	3.37E-09
ケース5	2.75	8.47E-11	9.34E-07	4.39E-08	8.86E-07	3.04E-10	4.31E-09
ケース6	2.75	1.07E-10	9.58E-07	4.53E-08	9.07E-07	4.06E-10	5.76E-09
ケース7	2.75	1.42E-10	1.08E-06	5.14E-08	1.02E-06	5.81E-10	8.25E-09
ケース8	2.75	2.87E-10	1.82E-06	8.68E-08	1.71E-06	1.36E-09	1.93E-08

表5-8 有機溶媒流量を減少させたときの抽出廃液中のU, Pu濃度
(ケース6, 45GWD/t相当のNp, Tc, Zrを添加した場合)

有機溶媒流量 (L/h)	HN03 (MOL/L)	U(total) (G/L)	Pu(total) (G/L)	PU(IV) (G/L)	PU(III) (G/L)	PU(V) (G/L)	PU(VI) (G/L)
300.0	2.75	1.07E-10	9.58E-07	4.53E-08	9.07E-07	4.06E-10	5.76E-09
299.8	2.75	3.62E-10	4.03E-06	1.89E-07	3.81E-06	1.91E-09	2.71E-08
298.6	2.75	2.58E-05	1.11E-03	4.72E-04	3.44E-04	3.03E-05	2.63E-04
298.4	2.75	1.34E-03	2.20E-02	1.92E-02	8.44E-04	3.80E-04	1.58E-03

参考文献

1. 権田浩三, 岡紘一郎, 福田章二: "Purexプロセス計算コードRevised MIXSET", PNCT 841-79-26(1979)
2. 根本剛, 宮地茂彦, 権田浩三: "Purexプロセスにおける超ウラン元素の分配計算コードAdvanced MIXSET", PNCT 841-81-35(1981)
3. 館盛勝一: "EXTRAM・M:再処理ピュレックス工程(ミキサセトラ)解析用過渡計算コードシステム", JAERI 1331(1993)
4. R.G.Steinke, J.F.Dearing: "Capacitance Matrix Method in TRAC and MELPROG", LA-UR 89-264(1989)
5. W.W.Schulz, J.D.Navatil: Science and Technology of TBP, CRC press(1984)

分配係数及び化学反応速度定数の参考文献については付録C及び付録Dに記載した。

付録 A MIXSET-Xの取り扱い成分のまとめ

表A-1 MIXSET-Xの取り扱い成分

J	成分	入力単位	単位変換係数[g/mol]
1	HNO ₃	mol/l	1.0
2	U (Ⅴ)	g/l	238.03
3	Pu (Ⅳ)	g/l	239.0
4	Pu (Ⅲ)	g/l	239.0
5	U (Ⅳ)	g/l	238.03
6	HNO ₂	mol/l	1.0
7	HYD	g/l	32.0
8	HAN	g/l	33.0
9	Pu (Ⅵ)	g/l	239.0
10	Pu (Ⅴ)	g/l	239.0
11	Np (Ⅳ)	g/l	237.0
12	Np (Ⅴ)	g/l	237.0
13	Np (Ⅵ)	g/l	237.0
14	Zr (Ⅳ)	g/l	95.0
15	Tc (Ⅳ)	g/l	99.0
16	Tc (Ⅴ)	g/l	99.0
17	Tc (Ⅵ)	g/l	99.0
18	Tc (Ⅶ)	g/l	99.0
19	Tc-U (Ⅴ)	g/l	99.0
20	Tc-Pu (Ⅳ)	g/l	99.0
21	Tc-Zr (Ⅳ)	g/l	99.0
22	Sr (Ⅱ)	g/l	90.0
23	Ru (DI)	g/l	106.0
24	Ru (TR)	g/l	106.0
25	Ru (NI)	g/l	106.0
26	Ru (MO)	g/l	106.0
27	Cs (I)	g/l	137.0
28	Ce (Ⅲ)	g/l	144.0
29	Gd (Ⅲ)	g/l	153.0
30	Am (Ⅲ)	g/l	242.0
31	Cm (Ⅲ)	g/l	243.0

HAN : NH₃OH⁺HYD : N₂H₅⁺

Ru (DI) : Ruジニトラ錯体

Ru (TR) : Ruトリニトラ錯体

Ru (NI) : Ruジニトロ錯体

Ru (MO) : Ruモノニトラ錯体

表A-2 取り扱い成分の電荷、TBP配位数、及び崩壊熱の初期値

J	成分	電荷	TBP配位数	崩壊熱[w/g]
1	HNO ₃	1.0	1.0 , 2.0	0.0
2	UO ₂ (NO ₃) ₂	2.0	2.0	0.0
3	Pu(NO ₃) ₄	2.0	2.0	0.0
4	Pu(NO ₃) ₃	3.0	2.0	0.0
5	U(NO ₃) ₄	4.0	2.0	0.0
6	HNO ₂	0.0	0.0	0.0
7	N ₂ H ₅ NO ₃	1.0	0.0	0.0
8	NH ₃ OHNO ₃	1.0	0.0	0.0
9	Pu(V)	1.0	1.0	0.0
10	Pu(VI)	2.0	2.0	0.0
11	Np(V)	4.0	2.0	0.0
12	Np(VI)	1.0	1.0	0.0
13	Np(VI)	2.0	2.0	0.0
14	Zr(V)	4.0	2.0	0.0
15	Tc(V)	0.0	0.0	0.0
16	Tc(VI)	0.0	0.0	0.0
17	Tc(VI)	0.0	0.0	0.0
18	Tc(VII)	0.0	2.0	0.0
19	Tc-U(VI)	0.0	2.0	---
20	Tc-Pu(V)	0.0	2.0	---
21	Tc-Zr(V)	0.0	2.0	---
22	Sr(II)	2.0	2.0	0.0
23	Ru(DI)	2.0	2.0	0.0
24	Ru(TR)	3.0	2.0	0.0
25	Ru(NI)	0.0	2.0	0.0
26	Ru(MO)	0.0	0.0	0.0
27	Cs(I)	1.0	1.0	0.0
28	Ce(III)	3.0	3.0	0.0
29	Gd(III)	3.0	3.0	0.0
30	Am(III)	3.0	3.0	0.0
31	Cm(III)	3.0	3.0	0.0

HAN : NH₃OH⁺HYD : N₂H₅⁺

Ru(DI) : Ruジニトラ錯体

Ru(TR) : Ruトリニトラ錯体

Ru(NI) : Ruジニトロ錯体

Ru(MO) : Ruモノニトラ錯体

注) Tc共抽出物の崩壊熱については、構成成分の値から計算される。

付録 B MIXSET-Xの組み込み化学反応

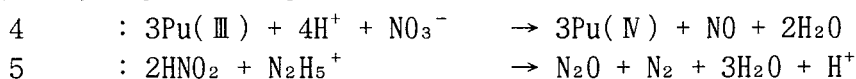
表B-1 化学反応番号と化学反応の対応(水相反応)(1/2)

反応番号	化学反応式
1*	$2\text{Pu}(\text{IV}) + \text{U}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Pu}(\text{III}) + \text{U}(\text{VI}) + 4\text{H}^+$
2*	$2\text{Pu}(\text{IV}) + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Pu}(\text{III}) + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
3**	$2\text{Pu}(\text{IV}) + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Pu}(\text{III}) + 2\text{NH}_4^+ + \text{N}_2 + 2\text{H}^+$
4***	$2\text{Pu}(\text{III}) + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Pu}(\text{IV}) + \text{HNO}_2 + \text{H}_2\text{O}$
5***	$\text{HNO}_2 + \text{N}_2\text{H}_5^+ \rightarrow \text{H}^+ + 2\text{H}_2\text{O} + \text{NH}_3$
6	$\text{Np}(\text{VI}) + \text{Np}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Np}(\text{V}) + 4\text{H}^+$
7	$\text{Pu}(\text{IV}) + \text{Np}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow \text{Pu}(\text{III}) + \text{Np}(\text{V}) + 4\text{H}^+$
8	$2\text{Np}(\text{VI}) + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Np}(\text{V}) + \text{N}_2 + 2\text{H}^+ + 2\text{NH}_4^+$
9	$2\text{Np}(\text{VI}) + \text{U}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Np}(\text{V}) + \text{U}(\text{VI}) + 4\text{H}^+$
10	$2\text{Np}(\text{V}) + \text{U}(\text{IV}) + 4\text{H}^+ \rightarrow 2\text{Np}(\text{IV}) + \text{U}(\text{VI}) + 2\text{H}_2\text{O}$
11	$2\text{Np}(\text{VI}) + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Np}(\text{V}) + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
12	$\text{Pu}(\text{V}) + \text{Pu}(\text{IV}) \rightarrow \text{Pu}(\text{VI}) + \text{Pu}(\text{III})$
13	$\text{Pu}(\text{V}) + \text{Pu}(\text{III}) + 4\text{H}^+ \rightarrow 2\text{Pu}(\text{IV}) + 2\text{H}_2\text{O}$
14	$2\text{Pu}(\text{VI}) + \text{HNO}_2 + \text{H}_2\text{O} \rightarrow 2\text{Pu}(\text{V}) + 3\text{H}^+ + \text{NO}_3^-$
15	$\text{Pu}(\text{VI}) + \text{Pu}(\text{III}) \rightarrow \text{Pu}(\text{V}) + \text{Pu}(\text{IV})$
16	$2\text{Pu}(\text{V}) + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Pu}(\text{VI}) + \text{HNO}_2 + \text{H}_2\text{O}$
17	$\text{NO}_3^- \rightarrow \text{HNO}_2$
18	$\text{HNO}_2 \rightarrow \text{H}^+$
19	$2\text{Np}(\text{V}) + 4\text{H}^+ \rightarrow \text{Np}(\text{IV}) + \text{Np}(\text{VI}) + 2\text{H}_2\text{O}$
20	$2\text{Np}(\text{V}) + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Np}(\text{VI}) + \text{HNO}_2 + \text{H}_2\text{O}$
21	$2\text{Np}(\text{IV}) + \text{NO}_3^- + 3\text{H}_2\text{O} \rightarrow 2\text{Np}(\text{V}) + \text{HNO}_2 + 5\text{H}^+$
22	$\text{Pu}(\text{III}) + \text{Np}(\text{VI}) \rightarrow \text{Pu}(\text{IV}) + \text{Np}(\text{V})$
23	$\text{Pu}(\text{III}) + \text{Np}(\text{V}) + 4\text{H}^+ \rightarrow \text{Pu}(\text{IV}) + \text{Np}(\text{IV}) + 2\text{H}_2\text{O}$
24	$2\text{Pu}(\text{VI}) + 2\text{N}_2\text{H}_5^+ \rightarrow 2\text{Pu}(\text{V}) + 2\text{NH}_4^+ + \text{N}_2 + 2\text{H}^+$
25	$2\text{Pu}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow \text{Pu}(\text{V}) + \text{Pu}(\text{III}) + 4\text{H}^+$
26	$2\text{Pu}(\text{V}) + 4\text{H}^+ \rightarrow \text{Pu}(\text{VI}) + \text{Pu}(\text{IV}) + 2\text{H}_2\text{O}$
27*	$\text{HNO}_2 + \text{NH}_3\text{OH}^+ \rightarrow \text{N}_2\text{O} + 2\text{H}_2\text{O} + \text{H}^+$
28	$2\text{Pu}(\text{VI}) + 2\text{NH}_3\text{OH}^+ \rightarrow 2\text{Pu}(\text{V}) + 4\text{H}^+ + \text{N}_2 + 2\text{H}_2\text{O}$
29	$2\text{Pu}(\text{V}) + 2\text{NH}_3\text{OH}^+ + 2\text{H}^+ \rightarrow 2\text{Pu}(\text{III}) + \text{N}_2\text{O} + 5\text{H}_2\text{O}$
30	$2\text{Pu}(\text{VI}) + \text{U}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Pu}(\text{V}) + \text{U}(\text{VI}) + 4\text{H}^+$

注) * 従来よりMIXSETで考慮されていた化学反応である。

** EXTRA. Mで考慮されており、従来よりMIXSET98でも考慮されている反応であるが、MIXSET98の反応速度定数は推定値にすぎないため、MIXSET-Xでは実験に基づくEXTRA. Mの反応速度定数を使用している。

*** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。これらの反応に対応するEXTRA. Mの反応式は次のようになっている。



表B-1 化学反応番号と化学反応の対応（水相反応）(2/2)

31	: Tc(VII) + N ₂ H ₅ ⁺	→ Tc(V) + products
32	: Tc(VI) + N ₂ H ₅ ⁺	→ Tc(IV) + products
33	: Tc(V) + N ₂ H ₅ ⁺	→ Tc(IV) + products
34	: Tc(IV) + NO ₃ ⁻	→ Tc(V) + NO ₂ ⁻
35	: Tc(V) + NO ₃ ⁻	→ Tc(VII) + NO ₂ ⁻
36	: Tc(VI) + NO ₃ ⁻	→ Tc(VII) + NO ₂ ⁻
37	: Tc(IV) + Tc(VII)	→ Tc(V) + Tc(V)
38	: Tc(VII) + 3Pu(III) + 4H ⁺	→ Tc(IV) + 3Pu(IV) + 2H ₂
39	: 2Tc(VII) + 3U(IV)	→ 2Tc(IV) + 3U(V)
40	: Tc(VII) + NH ₃ OH ⁺	→ Tc(V) + products
41*	: U(IV) + NO ₃ ⁻ + H ₂ O	→ U(V) + H ⁺ + HNO ₂
42*	: 2U(IV) + O ₂ + 2H ₂ O	→ 2U(V) + 4H ⁺
43**	: 2Np(V) + 2N ₂ H ₄ + 8H ⁺	→ 2Np(IV) + 2NH ₄ ⁺ + N ₂ + 4H ₂ O
44***	: 2Np(V) + HNO ₂ + H ₂ O	→ 2Np(V) + 3H ⁺ + NO ₃ ⁻
45****	: H ⁺ + Pu(III) + HNO ₂	→ Pu(IV) + NO + H ₂ O

注) * EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。

** Advanced MIXSETに記載されている反応で、EXTRA. Mでは考慮されていない化学反応である。

*** REACT-Mod (JAERI-Data/Code 96-030) で考慮されている反応で、表B-1の20番の逆反応であるが、反応速度定数はAdvanced MIXSET相当の値を使用している。

**** 12番、16番の反応の逆反応に相当する反応で、暫定的に導入。

表B-2 化学反応番号と化学反応の対応（有機相反応）

反応番号	化学反応式
1*	$2\text{Pu}(\text{IV}) + \text{U}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Pu}(\text{III}) + \text{U}(\text{VI}) + 4\text{H}^+$
4***	$2\text{Pu}(\text{III}) + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Pu}(\text{IV}) + \text{HNO}_2 + \text{H}_2\text{O}$
5	$2\text{HNO}_2 + (\text{N}_2\text{H}_5^+)_{\text{aq}} \rightarrow \text{N}_2\text{O} + \text{N}_2 + 3\text{H}_2\text{O} + \text{H}^+$ (ミキサ部のみ)
6	$\text{Np}(\text{VI}) + \text{Np}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Np}(\text{V}) + 4\text{H}^+$
9	$2\text{Np}(\text{VI}) + \text{U}(\text{IV}) + 2\text{H}_2\text{O} \rightarrow 2\text{Np}(\text{V}) + \text{U}(\text{VI}) + 4\text{H}^+$
14	$2\text{Pu}(\text{VI}) + \text{HNO}_2 + \text{H}_2\text{O} \rightarrow 2\text{Pu}(\text{V}) + 3\text{H}^+ + \text{NO}_3^-$
19	$2\text{Np}(\text{V}) + 4\text{H}^+ \rightarrow \text{Np}(\text{VI}) + \text{Np}(\text{IV}) + 2\text{H}_2\text{O}$
20	$2\text{Np}(\text{V}) + 3\text{H}^+ + \text{NO}_3^- \rightarrow 2\text{Np}(\text{VI}) + \text{HNO}_2 + \text{H}_2\text{O}$
41**	$\text{U}(\text{IV}) + \text{NO}_3^- + \text{H}_2\text{O} \rightarrow \text{U}(\text{VI}) + \text{H}^+ + \text{HNO}_2$
42**	$2\text{U}(\text{IV}) + \text{O}_2 + 2\text{H}_2\text{O} \rightarrow 2\text{U}(\text{VI}) + 4\text{H}^+$

注) * 従来よりMIXSETで考慮されていた化学反応である。

** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。

*** EXTRA. Mでは考慮されていないが、従来よりMIXSETで考慮されていた化学反応である。これらの反応に対応するEXTRA. Mの反応式は次のようになっている。



付録 C MIXSET-Xの組み込み分配係数

表C-1 SDISTカードによる内蔵分配係数計算式の選択対応表

J	CNAME (J)	IDIST=0	IDIST=1	IDIST=4	文献
1	HNO ₃	KfK	Hanford	KfK*	1,2,3
2	U (V)	KfK	Hanford	KfK*	1,2,3
3	Pu (N)	KfK	Hanford	KfK*	1,2,3
4	Pu (III)	**			1
5	U (N)	KfK	R.MIXSET	KfK***	3,4,5
6	HNO ₂	JAERI	JAERI	A.MIXSET	6,7
10	Pu (V)	**			8,9
11	Np (N)	A.MIXSET*	A.MIXSET*	A.MIXSET	7
12	Np (V)	A.MIXSET*	A.MIXSET*	A.MIXSET	7
13	Np (V)	KfK	A.MIXSET*	A.MIXSET	10,7
14	Zr (N)	**			11
18	Tc (VI)	CEA	ORNL	CEA	12,13
19	Tc - U (V)	CEA	ORNL	CEA*	12,13
20	Tc - Pu (N)	CEA	ORNL	CEA*	12,13
21	Tc - Zr (N)	CEA	ORNL	CEA*	12,13
22	Sr (II)	**			14
23	Ru (DI)	**			15
24	Ru (TR)	**			15
25	Ru (NI)	**			15
27	Cs (I)	**			14
28	Ce (III)	**			14
29	Gd (III)	**			16
30	Am (III)	**			7
31	Cm (III)	**			7

Ru (DI) : Ru ジニトラ錯体

Ru (TR) : Ru トリニトラ錯体

Ru (NI) : Ru ジニトロ錯体

- ** : プログラム内蔵の分配係数計算式が1種類であり, それを使用する
 KfK : KfKのデータを基にした分配係数計算式を使用 (*:温度係数が異なる)
 KfK*** : KfK3637のデータを基にした分配係数計算式を使用
 Hanford : HEDL-TME-75-31の分配係数計算式を使用
 JAERI : JAERI-M 93-095の分配係数計算式を使用
 R.MIXSET : Revised MIXSET (PNCT841-79-26)の分配係数計算式を使用
 A.MIXSET : Advanced MIXSET (PNCT841-81-35)の分配係数計算式を使用
 A.MIXSET* : Advanced MIXSET (PNCT841-81-35)の分配係数表を使用する
 CEA : CEA/ISEC'86 I-37の分配係数計算式を使用 (*:物質収支が異なる)
 ORNL : ORNL/TM-8668の分配係数計算式を使用

注) この表に記載のない成分についてはプログラム内蔵の分配係数値を0としている。

MIXSET-Xでは、Richardsonの式を元にした以下に示す抽出平衡式を内蔵している。
成分jのみかけの平衡定数 K_j を

$$K_j = y_j / (x_j \cdot T_f^{n_j})$$

ここで、 x_j : 成分jの水相濃度 T_j : Free TBP 濃度
 y_j : 成分jの有機相濃度 n_j : 成分jの TBP 配位数

とおくと、みかけの平衡定数は、水相の硝酸根濃度 X_{NO3} の関数として以下の様に表現される。

$$X_{NO3} = X_H + 2X_{UO2} + 2X_{Pu4} + 3X_{Pu3} + 4X_{U4} + X_{HAN} + X_{HYD} + X_{Pu5} + 2X_{Pu6} + 4X_{NP4} + X_{NP5} + 2X_{NP6} + 4X_{Zr}$$

$$K_{H1} = (a_1 X_{NO3}^{b1} + a_2 X_{NO3}^{b2})(1 - 0.54e^{-1.5F})$$

$$K_{H2} = K_{H1}$$

$$K_{UO2} = (a_3 X_{NO3}^{b3} + a_4 X_{NO3}^{b4} + a_5 X_{NO3}^{b5})(4F^{-0.17} - 3)$$

$$K_{Pu4} = K_{U6}(a_6 + b_6 X_{NO3}^{b7})$$

$$K_{Pu3} = 0.04 X_{NO3}^{1.8} + 0.000156F \cdot X_{NO3}^7$$

$$K_{U4} = K_{U6}(a_8 + b_8 X_{NO3}^{b9})$$

$$K_{HNO2} = 24.8 X_H^{(-0.28)} - 3.209 X_H^{(-0.653)}$$

ここで、 K_j : 成分jのみかけの平衡定数
 X_{NO3} : (水相の)硝酸根濃度 (mol/l)
 F : TBP体積分率

また、Free TBP濃度は次式で求められる。

$$T_f = T_0 - 2(K_{H2} X_H + K_{UO2} X_{UO2} + K_{Pu4} X_{Pu4} + K_{Pu3} X_{Pu3}) \cdot T_f^2 - K_{H1} X_H T_f - K_{HNO2} X_{HNO2} T_f - K_{Pu5} X_{Pu5} T_f - K_{NP5} X_{NP5} T_f - 2(K_{Pu6} X_{Pu6} + K_{NP4} X_{NP4} + K_{NP6} X_{NP6} + K_{Zr} X_{Zr} + K_{Tc7} X_{Tc7}) \cdot T_f^2$$

$$T_0 = 3.6538 \cdot X \cdot F$$

以上から、各成分jの分配係数 D_j は次式で求められる。

$$D_H = K_{H1} \cdot T_f + K_{H2} \cdot T_f^2$$

$$D_{UO2} = K_{UO2} \cdot T_f^2$$

$$D_{Pu4} = K_{Pu4} \cdot T_f^2$$

$$D_{Pu3} = K_{Pu3} \cdot T_f^2$$

$$D_{U4} = K_{U4} \cdot T_f^2$$

$$D_{HNO2} = K_{HNO2} \cdot T_f$$

各パラメータを次に示す。

表C-2 Richardsonの分配係数式の係数

係数	KfKデータからの係数	Hanfordの係数
a1	0.1416	0.135
b1	0.6724	0.82
a2	0.006058	0.0052
b2	3.418	3.44
a3	5.284	3.7
b3	1.582	1.57
a4	1.557	1.4
b4	3.843	3.9
a5	0.01267	0.011
b5	7.447	7.3
a6	0.3429	0.322114
b6	0.009552	0.0074
b7	2.154	2.0
a8	0.03909	---
b8	0.01205	---
b9	-0.5728	---

なお、U (IV) では、SDISTカードで IDIST=1 (R.MIXSET) を選択すると、 K_{U4} は次式で計算される。

$$\ln K_{U4} = -3.3360 + 1.9331 \times [X_{NO3}]$$

SDISTカードで IDIST=4 (KfK**) を選択すると、 K_{U4} は次式で計算される。

$$K_{U4} = 0.3556 \times \exp \{ 1.487 \times [X_{NO3}] \}$$

さらに、亜硝酸 HNO2では、SDISTカードで IDIST=4 (A.MIXSET) を選択すると表C-3の分配係数で計算される。

各成分 j の分配係数 D_j は次式

$$D_H = K_{H1} \cdot T_f + K_{H2} \cdot T_f^2$$

$$D_{UO2} = K_{UO2} \cdot T_f^2$$

$$D_{Pu4} = K_{Pu4} \cdot T_f^2$$

$$D_{Pu3} = K_{Pu3} \cdot T_f^2$$

$$D_{U4} = K_{U4} \cdot T_f^2$$

で求められるが、この式により求められる平衡定数は、25℃の標準値であり、段の温度が25℃以外の時には次の補正が施される。

$$K_j \leftarrow K_j \cdot \exp \left[E_j \left(\frac{1000}{25+273.15} - \frac{1000}{T+273.15} \right) \right]$$

ここで E_j は活性化エネルギーをガス定数で割った値であり、上記の内3成分のみについて以下の値が与えられている。MIXSET-Xでは、SDISTカードで IDIST=0 (KfK) または、IDIST=1 (Hanford) を指定したときにこの値を用いる。

$$E_H = -0.34(K)$$

$$E_{UO2} = -2.50(K)$$

$$E_{Pu4} = 0.20(K)$$

なお、SDISTカードで IDIST=4 (KfK*) を選択すると、表C-2の「KfKデータからの係数」の値を用い、 E_j として KfK データから求めた以下の値を用いる。

$$E_H = 0.60(K)$$

$$E_{UO2} = -0.60(K)$$

$$E_{Pu4} = 1.50(K)$$

上記以外の各成分については以下に示す分配係数式が内蔵されている。

・プルトニウム Pu(VI)

表C-4の式で計算される。(表C-6, 表C-7及び図C-1を参照)

表C-3 亜硝酸の分配係数

[HNO3]	D_{HNO2}
0.1	1.5
0.5	15.0
1.0	16.0
2.0	15.0
3.2	6.0

表C-4 Pu⁶⁺の分配係数の近似式

[HNO3]	D
2.0	$D = 2.85 - 4.91 \times C^{(8.13e-1)}$
3.0	$D = 4.02 - 6.95 \times C^{(8.15e-1)}$
4.0	$D = 5.13 - 8.48 \times C^{(7.62e-1)}$
4.5	$D = 5.39 - 8.82 \times C^{(7.47e-1)}$

C:有機相中のウラン濃度[mol/L]

・ネプツニウム

S D I S TカードでIDIST=4 (A.MIXSET) を選択すると、
Npの分配係数は次の式で計算される。

$$D_{Np(IV)} = K_{Np(IV)} \cdot T_f^2$$

$$D_{Np(V)} = K_{Np(V)} \cdot T_f$$

$$D_{Np(VI)} = K_{Np(VI)} \cdot T_f^2$$

ここで、

$$K_{Np(IV)} = b \times \exp \{ a \times X_{NO_3}^2 \}$$

$$a = 0.437 - 0.24 \times \log \{ X_U \} \quad (X_U : 0.035 \text{ 以上 })$$

$$= 0.78 \quad (X_U : 0.035 \text{ 未満 })$$

$$b = -0.548 + 0.175 \times \log \{ X_U \} \quad (X_U : 0.05 \text{ 以上 })$$

$$= -0.76 \quad (X_U : 0.05 \text{ 未満 })$$

$$K_{Np(V)} = 0.0234 \times X_{NO_3}^{1.41}$$

$$K_{Np(VI)} = 5.00 \times X_{NO_3}^{3.055} \quad (X_{NO_3} : 1 \text{ 以上 })$$

$$K_{Np(VI)} = 4.67 \times X_{NO_3}^{1.433} \quad (X_{NO_3} : 1 \text{ 未満 })$$

X_U : 水相中のウラン濃度(mol/l)

X_{NO_3} : 水相中の硝酸根濃度(mol/l)

・ネプツニウム Np (N)

S D I S TカードでIDIST=0 または 1 (A.MIXSET*) を選択すると次式で計算される。
 $Np^{4+} + 4NO_3^- + 2TBP = Np(NO_3)_4 \cdot 2TBP$

$$D_{Np(IV)} = K_1(H) \times \exp \{ -K_2(H) \times [U]_{org}^2 + K_3(H) \times [U]_{org} \}$$

$$= K_1(H) \times \exp \{ -K_2(H) \times Y_U^2 + K_3(H) \times Y_U \}$$

[H]: 水相硝酸濃度 X_H (N or mol/l), $[U]_{org}$: 有機相 U濃度 Y_U (g/l)

ここで、 $K_1(H) \sim K_3(H)$ は硝酸濃度: [H] (X_H) の関数であり1.0, 2.0, 3.0, 3.5 M
の各濃度に対して、次のような離散値で与えられている。

$$K_1(3.5) = 5 \quad K_2(3.5) = 3.6032 \times 10^{-4} \quad K_3(3.5) = 2.9736 \times 10^{-3}$$

$$K_1(3.0) = 3.2 \quad K_2(3.0) = 3.4688 \times 10^{-4} \quad K_3(3.0) = 2.9758 \times 10^{-3}$$

$$K_1(2.0) = 2 \quad K_2(2.0) = 2.7992 \times 10^{-4} \quad K_3(2.0) = -3.2225 \times 10^{-3}$$

$$K_1(1.0) = 0.722845 \quad K_2(1.0) = 2.1633 \times 10^{-4} \quad K_3(1.0) = -3.2236 \times 10^{-3}$$

従って、有機相 U濃度 Y_U を mol/lで表わすと、

$$K_1(3.5) = 5 \quad K_2(3.5) = 20.410 \quad K_3(3.5) = 0.70772$$

$$K_1(3.0) = 3.2 \quad K_2(3.0) = 19.649 \quad K_3(3.0) = 0.70824$$

$$K_1(2.0) = 2 \quad K_2(2.0) = 15.856 \quad K_3(2.0) = -0.76696$$

$$K_1(1.0) = 0.722845 \quad K_2(1.0) = 12.254 \quad K_3(1.0) = -0.76722$$

・ネプツニウム Np (V)

S D I S TカードでIDIST=0 または 1 (A.MIXSET*) を選択すると次式で計算される。
 $NpO_2^+ + NO_3^- + TBP = NpO_2NO_3 \cdot TBP$

$$D_{Np(V)} = \exp \left[-5.53012 \times \exp \{ -3.25919 \times \exp(\ln[H] - 2.33874) \} \right]$$

$$= \exp \left[-5.53012 \times \exp \{ -3.25919 \times \exp(\ln X_H - 2.33874) \} \right]$$

[H]: 水相硝酸濃度 X_H (N or mol/l)

・ネプツニウムNp (M)



S D I S TカードでIDIST=0 (K f K) を選択すると、表C-5の式で計算される。
(分配係数データについては表C-8を参照)

表C-5 Np (M)分配係数の近似式

[HNO3][mol/L]	U (M) [g/L]	KNp (M)
0.2058	70未満	$(-4.08295e-8)x^4+(9.75667e-6)x^3+(8.12588e-4)x^2+(0.0239163)x+(0.577235)$
	70以上	$(-1.10758e1)+(1.31967e1)x^(-2.812e-2)$
0.5291	15未満	$(1.98507)+(-0.082342)x^(0.749606)$
	15以上	$(-4.98318)+(7.85637)x^(-7.6909e-2)$
1.00	110未満	$(-59.0245)+(63.1353)x^(-0.012409)$
	110以上	$(5.62821)+(-3.09702)x^(1.05751e-1)$
1.67	40未満	$(-1.3877e2)+(1.44908e2)x^(-8.54918e-3)$
	40以上	$(-9.06522e-1)+(8.49352)x^(-3.78244e-1)$
2.07	40未満	$(-7.32508e1)+(8.05908e1)x^(-2.13834e-2)$
	40以上	$(-7.83706e-1)+(8.21658)x^(-3.95856e-1)$
3.00	40未満	$(-3.14568e2)+(3.24482e2)x^(-7.67715e-3)$
	40以上	$(-4.7101)+(9.63305)x^(-1.35568e-1)$
3.47	40未満	$(-3.47764)+(1.60775e1)x^(-3.34389e-1)$
	40以上	$(5.7535)+(-2.9546)x^(1.22887e-1)$
4.62		$(0.46222)+(39.0202)x^(-1.0898)$

x:水相のウラン濃度[g/L]

S D I S TカードでIDIST=1 (A.MIXSET*) を選択すると、Np (VI)の分配係数は次の式で計算される。

$$D_{Np(VI)} = K_1(H) \times \exp \{-K_2(H) \times [U]_{org}^2 + K_3(H) \times [U]_{org}\}$$

$$= K_1(H) \times \exp \{-K_2(H) \times Y_U^2 + K_3(H) \times Y_U\}$$

ここで、 $K_1(H) \sim K_3(H)$ は硝酸濃度： $[H]$ (X_H) の関数であり、1.0,2.0,3.0,3.5 Mの各濃度に対して、次のような離散値で与えられている。

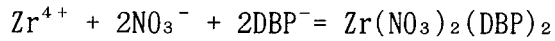
$$\begin{array}{lll} K_1(3.5) = 15.622 & K_2(3.5) = 1.8995 \times 10^{-4} & K_3(3.5) = -3.3539 \times 10^{-3} \\ K_1(3.0) = 12.2062 & K_2(3.0) = 1.9264 \times 10^{-4} & K_3(3.0) = -3.0115 \times 10^{-3} \\ K_1(2.0) = 9.29807 & K_2(2.0) = 2.0411 \times 10^{-4} & K_3(2.0) = -2.8993 \times 10^{-3} \\ K_1(1.0) = 4.57168 & K_2(1.0) = 8.9435 \times 10^{-4} & K_3(1.0) = -1.1422 \times 10^{-2} \end{array}$$

[H]: 水相硝酸濃度 X_H (N or mol/l), $[U]_{org}$: 有機相 U濃度 Y_U (g/l)

従って、有機相 U濃度 Y_U を mol/lで表わすと、

$$\begin{array}{lll} K_1(3.5) = 15.622 & K_2(3.5) = 10.760 & K_3(3.5) = -0.79823 \\ K_1(3.0) = 12.2062 & K_2(3.0) = 10.912 & K_3(3.0) = -0.71674 \\ K_1(2.0) = 9.29807 & K_2(2.0) = 11.562 & K_3(2.0) = -0.69003 \\ K_1(1.0) = 4.57168 & K_2(1.0) = 50.660 & K_3(1.0) = -2.7184 \end{array}$$

・ジルコニウム(+4)



$$D_{\text{Zr}} = K_1 \times X_{\text{NO}_3}^2 \times P_f^2 + K_2 \times X_{\text{NO}_3}^4 \times T_f'^2$$

$$K_1 = 7.5427 \times 10^7$$

$$P_f = [(1 + P_0 \times K_1 \times X_{\text{Zr}} \times X_{\text{NO}_3}^2)^{1/2} - 1] / (K_1 \times X_{\text{Zr}} \times X_{\text{NO}_3}^2)$$

$$K_2 = \exp(11.156 - 28.424 \times I^{1/2} + 17.655 \times I - 4.483 \times I^{3/2} + 0.433 \times I^2)$$

$$T_f' = T_f / [1 + 0.189 X_{\text{H}}(X_{\text{NO}_3}) + 4.05 \times 10^{-4} \times X_{\text{H}}^2 (X_{\text{NO}_3})^2]$$

ここで

$$I = X_{\text{H}} + 3X_{\text{U}}$$

K_1 : Zr-HDBP反応の見掛けの平衡定数

K_2 : Zr-TBP反応の見掛けの平衡定数

X_{H} : 水相中の硝酸濃度(N or mol/l)

X_{U} : 水相中のウラン濃度(mol/l)

P_0 : 初期HDBP濃度(mol/l)

P_f : 未反応のHDBP濃度(mol/l)

T_0 : 初期TBP濃度(mol/l)

T_f : 未反応のTBP濃度(mol/l)

Y_{U} : 有機相中のウラン濃度(mol/l)

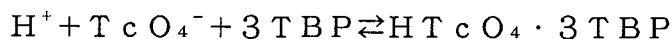
X_{NO_3} : 水相中の硝酸根濃度(mol/l)

X_{Zr} : 水相中のZr濃度(mol/l)

・テクネチウム(+7)

・SDISTカードでIDIST=0 (CEA) を選択すると共抽出を考えた次の式で計算される。

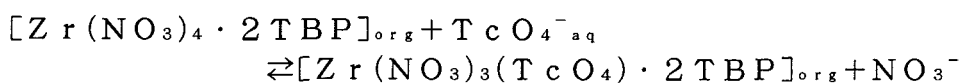
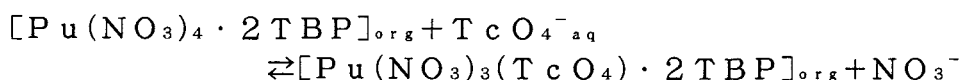
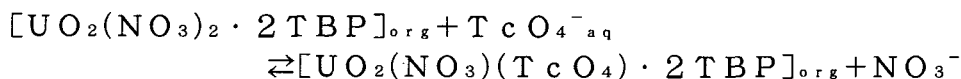
Tc 単独抽出の化学式



Tc 単独抽出の分配係数

$$D_{\text{Tc}} = \exp \left\{ (-20.13) + \frac{6013}{T} + (0.815) \ln[\text{H}^+] + (2.91) \ln[\text{TBP}] \right\}$$

Tc 共抽出の化学式



Tc 共抽出の分配係数

$$D_{\text{M-Tc}} = K_{\text{M}} \frac{K_{\text{A}}}{K_{\text{A}} + [\text{H}^+]_{\text{aq}}} \cdot \frac{[\text{M}_{\text{org}}]}{[\text{NO}_3^-]_{\text{aq}}}$$

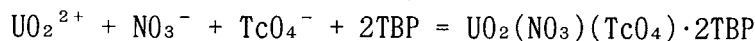
$$K_{\text{A}} = (2.63) \exp \left\{ (-4000) \left[\frac{1}{T} - \frac{1}{298} \right] \right\}$$

$$K_{\text{U-Tc}} = 8 \quad K_{\text{Pu-Tc}} = 160 \quad K_{\text{Zr-Tc}} = 2500$$

注) SDISTカードでIDIST1 (CEA*) を選択すると、分配係数計算式自体は上記のIDIST=0 (CEA) の式を用いるが、物質収支の取り方が異なる。IDIST1 (CEA*) ではM-TcとしてTc共抽出物の形態となったM (U, Pu, Zr) も単独抽出のMに含まれるとして計算される。

・SDISTカードでIDIST=1 (ORNL) を選択するとUとの共抽出のみを考えた次の式でTc(VII)の分配係数が計算される。

Tcの単独抽出



$$D_{\text{Tc}} = \exp(0.341 + 0.573 \times \ln[\text{U}]_{\text{org}} - 0.832 \times \ln[\text{NO}_3])$$

$$= \exp(0.341 + 0.573 \times \ln Y_{\text{U}} - 0.832 \times \ln X_{\text{NO}_3})$$

[NO₃]: 水相硝酸根濃度 X_{NO₃} (mol/l)

[U]_{org}: 有機相 U濃度 Y_U (mol/l), Y_U = MAX(2.95x10⁻³, [U]_{org}) (mol/l)

Tcの共抽出

$$D_{\text{M-Tc}} = 0$$

注) Tc(VII), Tc-U, Tc-Pu, Tc-Zrの各分配係数オプションを同時にIDIST=1 (ORNL) とすることにより、Tcの分配係数はMIXSET98相当の取り扱いとなる。

・ストロンチウム(+2)



$$D_{\text{Sr}} = \exp K_1 \times [\text{TBP}]_{\text{free}}^2 \\ = \exp K_1 \times T_f^2$$

$$K_1 = 1.0226 \times [\text{NO}_3] - 9.052 + (2.3303/[\text{NO}_3]) - (1.2659/[\text{NO}_3]^2) \\ = 1.0226 \times X_{\text{NO}_3} - 9.052 + (2.3303/X_{\text{NO}_3}) - (1.2659/X_{\text{NO}_3}^2)$$

$[\text{NO}_3]$: 水相硝酸根濃度 X_{NO_3} (mol/l), $[\text{TBP}]_{\text{free}}$: Free TBP 濃度 T_f (mol/l)

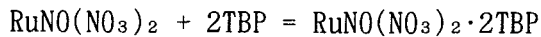
・ルテニウム

[分配係数]

X_{NO_3} : 水相中の硝酸根濃度(mol/l), T_f : 未反応のTBP濃度(mol/l)とすると

$$D_{\text{Ru}} = K_{\text{Ru}} \times T_f^2$$

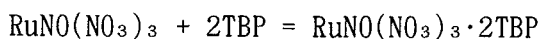
①ジニトラト錯体



$$K_{\text{Ru}}(\text{DI}) = \exp(0.3044 X_{\text{NO}_3}^3 - 1.4419 X_{\text{NO}_3}^2 + 1.8536 X_{\text{NO}_3} - 1.7247) \\ (0 < X_{\text{NO}_3} < 2.1 \text{ M})$$

$$K_{\text{Ru}}(\text{DI}) = \exp(-0.1549 X_{\text{NO}_3} - 1.0514) \quad (2.1 < X_{\text{NO}_3} \text{ M})$$

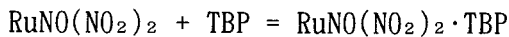
②トリニトラト錯体



$$K_{\text{Ru}}(\text{TR}) = \exp(-13.911 X_{\text{NO}_3}^2 + 8.524 X_{\text{NO}_3} + 3.447) \quad (0 < X_{\text{NO}_3} \leq 0.4 \text{ M})$$

$$K_{\text{Ru}}(\text{TR}) = \exp(0.1152 X_{\text{NO}_3}^2 - 1.002 X_{\text{NO}_3} + 5.015) \quad (0.4 < X_{\text{NO}_3} \leq 6 \text{ M})$$

③ジニトロ錯体

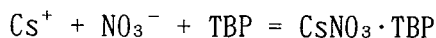


$$K_{\text{Ru}}(\text{NI}) = \exp\{0.3703 X_{\text{NO}_3} + 0.6187 - (0.02096/X_{\text{NO}_3})\}$$

④モノニトラト錯体

他の錯体に比べ十分小さく, 分配係数は 0 と仮定する。

・セシウム (Cs) (+1)



$$D_{\text{Cs}} = \exp\{0.33114[\text{NO}_3] - 7.1608 - (0.444/[\text{NO}_3])\} \times T_f \\ = \exp\{0.33114 X_{\text{NO}_3} - 7.1608 - (0.444/X_{\text{NO}_3})\} \times T_f$$

$[\text{NO}_3]$: 水相硝酸根濃度 X_{NO_3} (mol/l), T_f : Free TBP 濃度 (mol/l)

・セリウム (Ce) (+3)



$$D_{\text{Ce}} = \exp\{1.3771[\text{NO}_3] - 4.3411 - (0.41314/[\text{NO}_3])\} \times T_f^3 \\ = \exp\{1.3771 X_{\text{NO}_3} - 4.3411 - (0.41314/X_{\text{NO}_3})\} \times T_f^3$$

$[\text{NO}_3]$: 水相硝酸根濃度 X_{NO_3} (mol/l), T_f : Free TBP 濃度 (mol/l)

・ガドリニウム(+3)



$$D_{\text{Gd}} = \exp(2.303 \times (\exp K_1 + K_4))$$

$$K_1 = K_{11} \times (\exp K_{12} - \exp K_{13})$$

$$K_{11} = 1.2 + 0.001 \times [\text{U}]_{\text{org}}$$

$$= 1.2 + 0.001 \times Y_U$$

$$(= 1.2 + 0.238 \times Y_U \quad (Y_U : \text{mol/l}))$$

$$K_{12} = -[\text{H}^+] / [-0.000075 \times ([\text{U}]_{\text{org}} + 40)^{2.467} + 21]$$

$$= -X_{\text{H}^+} / [-0.000075 \times (Y_U + 40)^{2.467} + 21]$$

$$(= -X_{\text{H}^+} / [-54.71 \times (Y_U + 0.1681)^{2.467} + 21] \quad (Y_U : \text{mol/l}))$$

$$K_{13} = -[\text{H}^+]$$

$$= -X_{\text{H}^+}$$

$$K_4 = -4 - 0.01 \times [\text{U}]_{\text{org}}$$

$$= -4 - 0.01 \times Y_U$$

$$(= -4 - 2.38 \times Y_U \quad (Y_U : \text{mol/l}))$$

$[\text{H}^+]$: 水相硝酸濃度 X_{H^+} (mol/l), $[\text{U}]_{\text{org}}$: 有機相 U濃度 Y_U (g/l)

・アメリシウム (Am) (+3)

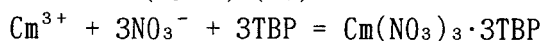


$$D_{\text{Am}} = 0.0174 \times X_{\text{NO}_3}^{2.37} \times T_f^3 \quad (X_{\text{NO}_3} < 2.0 \text{ M})$$

$$D_{\text{Am}} = 0.00151 \times X_{\text{NO}_3}^{5.69} \times T_f^3 \quad (2 \leq X_{\text{NO}_3} < 6.0 \text{ M})$$

X_{NO_3} : 水相硝酸根濃度(mol/l), T_f : Free TBP 濃度(mol/l)

・キュリウム (Cm) (+3)



$$D_{\text{Cm}} = 0.0115 \times X_{\text{NO}_3}^{2.31} \times T_f^3 \quad (X_{\text{NO}_3} < 2.0 \text{ M})$$

$$D_{\text{Cm}} = 0.000617 \times X_{\text{NO}_3}^{6.45} \times T_f^3 \quad (2 \leq X_{\text{NO}_3} < 6.2 \text{ M})$$

X_{NO_3} : 水相硝酸根濃度(mol/l), T_f : Free TBP 濃度(mol/l)

有機相中のウラン濃度はU(VI)とU(IV)の和で計算している。

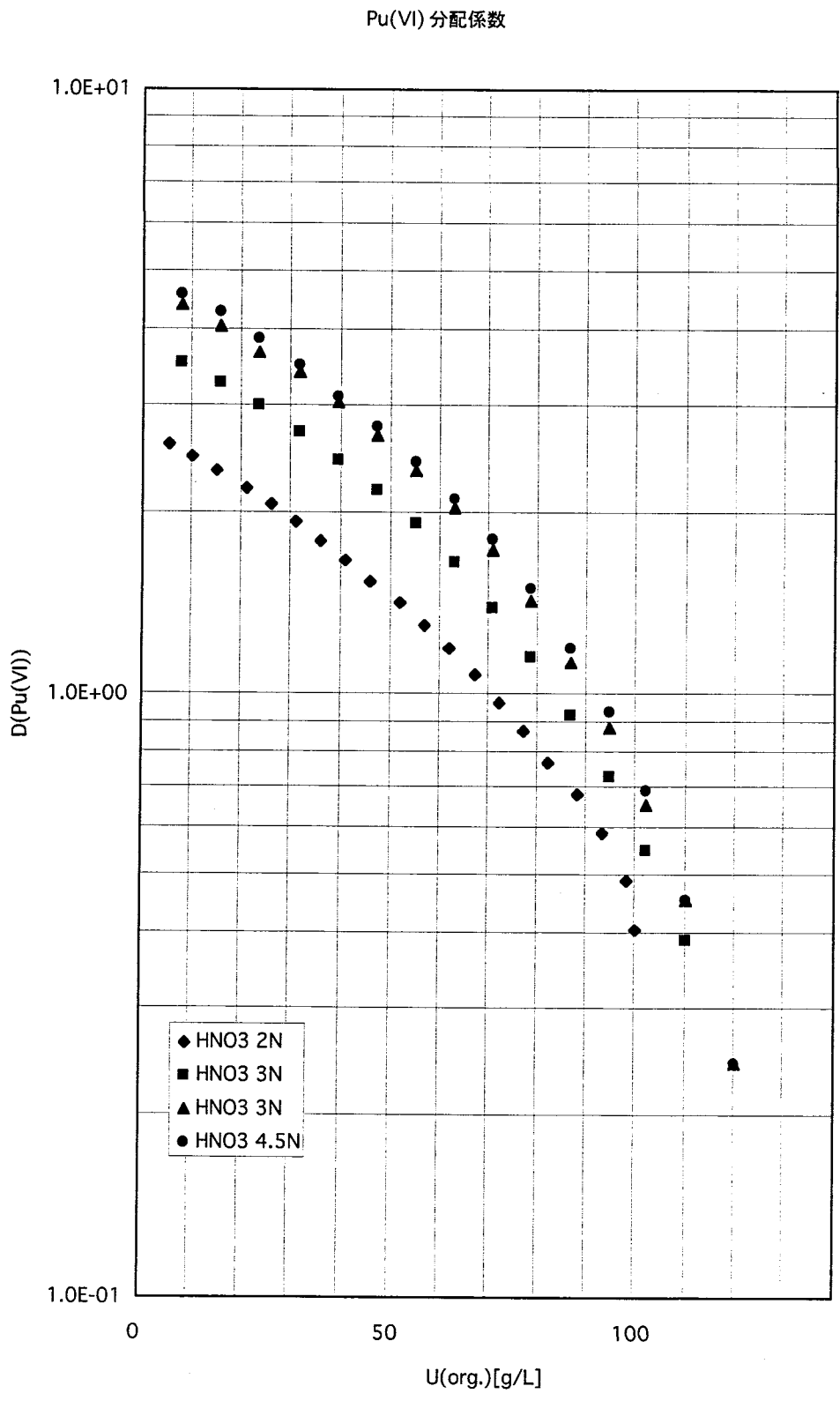


図 C - 1 TBP30v/o, Dodecane-UO₂(NO₃)₂-HNO₃-H₂O 系での PuO₂(NO₃)₂分配比のU濃度による影響

表C-6 TBP30v/o, Dodecane-UO₂(NO₃)₂-HNO₃-H₂O 系での
PuO₂(NO₃)₂ 分配比のU濃度による影響(温度 22°C)

[HNO ₃][N]	2.0		3.0	4.0	4.5
[U][g/L]	D(Pu)	[U][g/L]	D(Pu)	D(Pu)	D(Pu)
5.20E+00	2.60E+00	7.90E+00	3.54E+00	4.41E+00	4.58E+00
1.00E+01	2.47E+00	1.57E+01	3.28E+00	4.08E+00	4.30E+00
1.50E+01	2.34E+00	2.36E+01	3.00E+00	3.68E+00	3.88E+00
2.10E+01	2.20E+00	3.15E+01	2.71E+00	3.41E+00	3.50E+00
2.60E+01	2.06E+00	3.93E+01	2.44E+00	3.04E+00	3.12E+00
3.10E+01	1.93E+00	4.72E+01	2.18E+00	2.68E+00	2.78E+00
3.60E+01	1.79E+00	5.51E+01	1.92E+00	2.35E+00	2.42E+00
4.10E+01	1.67E+00	6.30E+01	1.66E+00	2.04E+00	2.10E+00
4.60E+01	1.54E+00	7.08E+01	1.39E+00	1.73E+00	1.80E+00
5.20E+01	1.42E+00	7.87E+01	1.15E+00	1.43E+00	1.50E+00
5.70E+01	1.30E+00	8.66E+01	9.20E-01	1.13E+00	1.19E+00
6.20E+01	1.19E+00	9.44E+01	7.30E-01	8.79E-01	9.37E-01
6.70E+01	1.08E+00	1.02E+02	5.50E-01	6.54E-01	6.89E-01
7.20E+01	9.70E-01	1.10E+02	3.90E-01	4.57E-01	4.57E-01
7.70E+01	8.70E-01	1.20E+02		2.44E-01	2.44E-01
8.20E+01	7.71E-01				
8.80E+01	6.84E-01				
9.30E+01	5.87E-01				
9.80E+01	4.92E-01				
1.00E+02	4.06E-01				

表C-7 Pu(V)分配係数の硝酸濃度依存性
 Pu(V); HN03; 19-20 % TBP - Kerosene
 T:25C
 Tracer[Pu]

Aqueous Phase HN03, M	Organic Phase TBP, vol%	Ea
0.5	19	0.286
1.1	19	0.75
2.1	19	1.77
2.8	19	2.22
4.07	19	2.87
5.03	19	3.16
5.77	19	3.38
6.01	19	3.57
6.76	19	3.33
7.03	19	3.27
8.08	19	2.9
9.28	19	2.27
10.6	19	1.99
11.1	19	1.86
12.6	19	1.71
13.7	19	1.86
0.3	20	0.13
0.5	20	0.27
0.7	20	0.6
1	20	0.72
2	20	1.54
3	20	2.26
5	20	2.7
6	20	2.86
7	20	3.05
8	20	2.56
10	20	2.27
11.5	20	1.84
13	20	1.92

Reference:

1:Tri-n-Butyl Phosphate as an Extracting Solvent for Inorganic Nitrates -V. Further Results for the Tetra- and Hexavalent Actinide Nitrates.

Alcock, K., G.F. Best, E. Hesford, and H.A.C. McKay.

J. Inorg. and Nucl. Chem., 6(1958)328-333.

2: Collected Partition Data for Tri-Butyl Phosphate.

Geary, N.R., Research and Development Report RDB(W)-8142, 54pp.

表C-8 Np(N)とNp(M)の分配係数基礎データ(1/2)

Distribution ratios of Np(N) and Np(M) at varied concentrations of nitric acid and U(V) in the equilibrium aqueous phase (25-60C)

Np(N, M)分配係数

HN03 mol/L	U(V) g/L	Temp C	Np(N)	Np(M)
0.200	1.35	25		0.61
0.200	2.5	25		0.623
0.201	6.1	25		0.703
0.202	10.8	25		0.757
0.204	21.4	25		0.796
0.208	41.7	25		0.749
0.211	69.2	25		0.637
0.212	99.2	25		0.524
0.214	137.00	25		0.414
0.500	0.69	25		1.7
0.501	1.32	25	0.135	
0.502	1.32	25		1.876
0.506	3.11	25	0.0625	
0.506	3.11	25		1.81
0.51	6.44	25	0.059	
0.51	6.44	25		1.64
0.52	14.7	25	0.033	
0.52	14.7	25		1.37
0.535	34.0	25	0.043	
0.535	34.0	25		1.09
0.548	63.7	25	0.035	
0.548	63.7	25		0.666
0.548	63.7	25		0.732
0.555	96.7	25	0.0335	
0.555	96.7	25		0.542
0.562	136.0	25	0.0295	
0.562	136.0	25		0.4
1.0	0.34	25		3.88
1.0	0.53	25		4.69
1.02	1.43	25		4.6
1.03	3.0	25		3.94
1.0	4.04	25	0.48	
1.0	4.67	25		3.21
1.06	8.12	25		2.84
1.0	8.37	25	-0.376	
1.0	8.41	25		2.62
0.9	8.5	25	0.19	2.45
0.9	8.5	25	0.19	2.41
1.11	24.0	25		1.7
0.93	25.5	25	0.1	1.6
0.93	25.5	25	0.094	1.48
1.0	29.0	25	-0.2	
1.0	30.5	25		1.26
1.0	42.8	25	-0.15	
1.0	48.3	25		0.94
0.96	47.0	25	0.058	1.07
0.96	47.0	25	0.057	1.05
1.14	51.5	25		0.884
0.97	74.0	25	0.046	0.71
0.97	74.0	25	0.047	0.71
1.16	85.6	25		0.611
1.0	88.0	25	0.065	
1.0	96.8	25		0.54
0.98	108.0	25	0.038	0.51
0.98	108.0	25	0.038	0.5
1.0	121.2	25	0.035	
1.17	126.0	25		0.448
0.99	180.0	25	0.035	0.27
0.99	180.0	25	0.032	0.3
1.0	210.6	25		0.24
1.5	0.16	25		8.04
1.5	0.35	25	1.193	
1.5	0.35	25		7.41
1.53	0.94	25	1.08	
1.63	0.94	25		6.72
1.55	2.0	25	0.933	
1.55	2.0	25		5.83
1.78	4.2	25	0.56	4.26
1.78	4.2	25	0.55	3.81
1.6	5.16	25	0.617	
1.6	5.16	25		4.46

HN03 mol/L	U(V) g/L	Temp C	Np(N)	Np(M)
1.68	16.8	25	0.225	
1.68	16.8	25		2.444
1.75	42.2	25	0.106	
1.75	42.2	25	0.112	
1.75	42.2	25		1.27
1.75	42.2	25		1.045
1.78	76.6	25	0.067	
1.78	76.6	25		0.723
1.79	118.0	25	0.048	
1.79	118.0	25		0.495
2.02	0.17	25		8.94
2.04	0.29	25	2.174	
2.04	0.29	25		9.0
2.04	0.77	25	1.805	
2.04	0.77	25		8.66
2.07	1.49	25	1.48	
2.07	1.49	25		7.71
2.0	1.83	25		7.03
2.0	1.88	25	1.36	
2.0	3.13	25	1.24	
2.0	3.62	25		5.9
2.15	4.17	25	0.938	
2.15	4.17	25		5.46
2.0	12.12	25	0.453	
2.0	12.27	25		3.21
1.9	15.7	25	0.22	2.49
1.9	15.7	25	0.22	2.22
2.26	16.8	25	0.316	
2.26	16.8	25		2.41
2.0	23.4	25	0.193	
2.0	28.2	25		1.51
1.96	33.7	25	0.11	1.28
1.96	33.7	25	1.0	1.2
2.31	43.8	25	0.121	
2.31	43.8	25		1.08
1.98	45.0	25	0.088	1.07
1.98	45.0	25	0.09	0.94
1.91	45.6	25		1.13
2.0	59.5	25		0.796
2.0	60.0	25	0.073	0.83
2.0	60.0	25	0.076	0.73
2.0	82.8	25	0.063	
2.35	83.9	25	0.071	
2.35	83.9	25		0.605
2.02	115.0	25	0.058	0.56
2.02	115.0	25	0.055	0.49
2.37	129.0	25	0.05	
2.37	129.0	25		0.418
2.0	178.3	25	0.034	
3.01	0.1	25		13.4
3.02	0.18	25	2.92	
3.02	0.16	25		13.34
3.02	0.18	25		13.75
3.05	0.43	25	2.7	
3.05	0.48	25		12.25
3.05	0.43	25		13.25
3.1	0.94	25	2.23	
3.1	0.94	25		11.5
3.1	1.1	25		10.7
3.0	1.19	25		11.6
3.0	1.19	25	2.995	
3.0	2.36	25		8.03
3.0	2.85	25	-2.33	
2.7	3.4	25	1.05	6.67
2.7	3.4	25	1.1	7.14
3.0	4.41	25	1.405	
3.0	7.92	25		4.09
3.0	8.51	25	0.963	
2.87	10.0	25	0.39	3.39
2.87	10.0	25	0.44	3.68
3.27	11.9	25		3.36

HN03 mol/L	U(V) g/L	Temp C	Np(N)	Np(M)
3.0	16.4	25	-0.677	
3.0	17.7	25		2.48
2.98	29.2	25	0.17	1.52
2.98	29.2	25	0.19	1.32
3.01	40.5	25	0.13	1.18
3.01	40.5	25	0.14	1.08
3.0	46.4	25	-0.373	
3.0	48.4	25		0.977
3.03	62.0	25	0.1	0.83
3.03	62.0	25	0.1	0.73
3.0	72.7	25	0.138	
3.05	101.2	25	0.076	0.48
3.05	101.2	25	0.088	0.42
3.0	182.0	25	0.039	
3.5	0.94	25	3.93	
3.5	1.04	25		12.1
3.52	2.08	25	2.01	9.47
3.5	2.14	25		9.08
3.5	2.21	25	-3.41	
3.21	2.54	25	1.47	
3.21	2.54	25		8.2
3.16	2.55	25		8.64
3.21	2.69	25		7.65
3.38	4.0	25	1.74	
3.38	4.0	25	1.74	
3.75	4.5	25	1.33	6.63
3.5	7.21	25		5.24
3.5	7.92	25	1.2	
3.38	11.2	25		3.37
3.38	11.2	25	0.559	
3.38	13.4	25		2.82
3.5	13.8	25		2.9
3.5	14.5	25	0.568	
3.5	35.9	25	0.22	
3.5	38.0	25	0.198	
3.5	38.0	25	0.186	
3.5	38.0	25		1.22
3.5	38.0	25		1.136
3.5	41.5	25		1.09
3.5	50.3	25		1.05
3.5	70.2	25	0.083	
3.52	78.5	25		0.616
3.52	78.5	25	0.114	
3.52	81.6	25		0.58
3.75	102.0	25	0.103	
3.7	104.0	25	0.107	
3.54	111.0	25	0.091	
3.54	111.0	25		0.4
3.54	123.0	25		0.4
3.36	124.0	25		0.41
3.5	139.0	25		-0.193
3.39	191.0	25		0.235
5.28	1.71	25	3.22	
3.88	8.5	25	0.84	4.24
5.24	9.88	25	1.0175	
3.9	16.0	25	0.51	2.42
3.95	26.5	25	0.32	1.48
4.87	34.3	25	0.386	
3.99	53.0	25	0.23	1.01
5.07	71.2	25	0.236	
5.19	143.0	25	0.156	
4.79	189.0	25	0.1294	
6.54	9.91	25	2.03	
6.7	35.2	25	0.888	
6.54	76.8	25	0.515	
6.41	122.0	25	0.342	

表C-8 Np(V)とNp(VI)の分配係数基礎データ(2/2)

Distribution ratios of Np(V) and Np(VI) at varied concentrations of nitric acid and U(VI) in the equilibrium aqueous phase (25-60C)

Np(V, VI)分配係数

HN03 mol/L	U(VI) g/L	Temp C	Np(V)	Np(VI)
0.2	3.6	60		0.261
0.2	7.84	60		0.287
0.202	14.3	60		0.329
0.204	27.7	60		0.331
0.208	50.8	60		0.327
0.211	78.3	60		0.367
0.212	109.0	60		0.288
0.214	143.0	60		0.285
0.502	2.18	60		0.951
0.505	5.17	60		0.857
0.508	9.06	60		0.845
0.518	19.7	60		0.829
0.532	41.1	60		0.675
0.544	69.9	60		0.535
0.553	99.3	60		0.378
0.56	138.0	60		0.299
1.0	3.97	60		2.82
1.04	6.0	60	0.51	2.18
1.02	10.7	60		-2.82
1.07	14.0	60	0.4	1.68
1.03	18.1	60		-2.49
1.06	37.0	60		-1.85
1.11	37.8	60	0.26	1.05
1.1	65.3	60		-1.16
1.14	73.5	60	0.19	0.69
1.13	84.0	60		0.692
1.15	98.4	60		0.517
1.16	107.0	60		0.383
1.16	113.0	60	0.16	0.5
1.16	160.5	60	0.15	0.39
1.51	0.54	60		4.79
1.53	1.26	60		4.24
1.55	2.9	60		3.97
1.6	7.6	60		2.565
1.67	22.9	60		1.746
1.74	49.8	60		0.915
1.77	86.7	60		0.57
1.78	125.0	60		0.414
2.0	0.43	60		5.66
2.04	1.11	60		5.09
2.07	2.41	60		4.585
2.05	3.6	60	1.87	4.41
2.14	7.0	60		3.22
2.13	9.6	60	1.18	2.89
2.23	22.8	60		1.44
2.24	36.3	60	0.52	1.24
2.29	52.0	60		0.777
2.29	80.8	60	0.29	0.68
2.33	90.0	60		0.545
2.37	133.0	60		0.318
2.3	134.5	60	0.22	0.45
2.33	186.0	60	0.2	0.33
3.02	0.34	60		7.9
3.05	0.79	60		7.28
3.1	1.66	60		5.51
3.09	2.5	60	3.27	5.62
3.21	4.68	60		4.41
3.22	7.42	60	2.04	3.49
3.36	18.3	60		2.06
3.42	32.9	60	0.81	1.27
3.44	47.5	60		0.905
3.51	81.0	60	0.42	0.65
3.49	85.1	60		0.515
3.53	130.0	60		0.354
3.54	139.7	60	0.31	0.41
3.56	183.5	60	0.28	0.31
4.15	2.49	60	4.71	5.87
4.22	7.35	60	2.96	3.34
4.51	35.2	60	1.12	1.16
4.59	82.0	60	0.62	0.58
4.63	135.5	60	0.48	0.38
4.65	192.0	60	0.41	0.3

HN03 mol/L	U(VI) g/L	Temp C	Np(V)	Np(VI)
1.05	4.4	45	0.54	2.81
1.08	10.6	45	0.36	2.1
1.13	32.9	45	0.19	1.2
1.14	66.1	45	0.12	0.76
1.17	104.0	45	0.087	0.52
1.18	162.0	45	0.071	0.38
2.08	2.64	45	1.52	5.64
2.16	7.65	45	0.89	3.64
2.27	32.0	45	0.34	1.42
2.34	77.9	45	0.17	0.7
2.35	130.0	45	0.13	0.45
2.39	184.5	45	0.11	0.33
3.13	1.83	45	2.65	7.52
3.27	6.08	45	1.61	4.64
3.48	17.5	45	0.55	4.51
3.54	73.4	45	0.25	0.67
3.59	128.0	45	0.19	0.43
3.6	185.0	45	0.16	0.3
4.09	1.64	45	4.12	8.38
4.26	5.21	45	2.48	4.99
4.53	30.6	45	0.79	1.36
4.7	78.6	45	0.4	0.64
4.73	129.0	45	0.29	0.4
4.74	193.0	45	0.26	0.29

HN03 mol/L	U(VI) g/L	Temp C	Np(V)	Np(VI)
3.29	3.54	50	2.3	
3.16	3.92	50		4.67
3.31	16.3	50		2.17
3.51	92.6	50	0.292	
3.33	124.0	50		0.406
3.42	191.0	50		0.246
4.99	3.51	50	3.56	
4.95	16.1	50	1.417	
4.83	43.3	50	0.697	
4.95	78.2	50	0.488	
5.07	154.0	50	0.381	
4.75	203.0	50	0.329	
6.62	19.3	50	2.22	
6.54	49.7	50	1.113	
6.49	88.7	50	0.761	
6.33	136.0	50	0.52	

分配係数の参考文献

- C1) G.L.Richardson, J.L.Swanson: "Plutonium Partitioning in the PUREX Process with Hydrazine-Stabilized Hydroxylamine nitrate", HEDL-TME 75-31(1975)
- C2) S.B.Watson, R.H.Rainey: "Modification of the SEPHIS Computer Code for Calculating the PUREX Solvent Extraction System", ORNL-TM-5123(1975)
- C3) G.Petrich, Z.Kolarik: "The 1981 Purex Distribution Data Index", KfK 3080(1981)
- C4) 権田浩三, 岡紘一郎, 福田章二: "Purexプロセス計算コード Revised MIXSET", PNCT 841-79-26(1979)
- C5) S.Tachimori, T.Nakashima, Z.Kolarik, R.Schuler: "Distribution of U(IV), U(VI), Pu(III) and Nitric Acid between 30 vol.% Tributyl Phosphate in Dodecane and Aqueous Nitrate Solutions", KfK 3637 (1983)
- C6) 宝徳忍, 木原武広, 内山軍蔵, 藤根幸雄, 前田充: "亜硝酸の再処理溶液中での分配係数", JAERI-M 93-095, (1993)
- C7) 根本剛, 宮地茂彦, 権田浩三: "Purexプロセスにおける超ウラン元素の分配計算コード Advanced MIXSET", PNCT 841-81-35(1981)
- C8) Alcock, K., G.F.Best, E.Hesford, and H.A.C.McKay: "Tri-n-Butyl Phosphate as an Extracting Solvent for Inorganic Nitrates -V. Further Results for the Tetra- and Hexavalent Actinide Nitrates.", J. Inorg. and Nucl. Chem., 6(1958)
- C9) Geary, N.R.: "Collected Partition Data for Tri-Butyl Phosphate.", Research and Development Report RDB(W)-8142
- C10) Z.Kolarik, P.Dressler: "Purex Process Related Distribution Data on Neptunium(IV, VI)", KfK 4667 (1990)
- C11) Hardy, C.J et al.: "Studies on mono- and di-n-butylphosphoric acids -III: The extraction of zirconium from nitrate solution by di-n-butylphosphoric acid", J. Inorg. Nucl. Chem. 17 (1961)
- C12) E.Vialad, M.Germain: "Technetium behaviour control in the Purex Process", CEA/ISEC'86 I-37 (1986)
- C13) D.J.Pruett: "The Extraction of Heptavalent Technetium and Rhenium by Tributyl Phosphate", ORNL/TM-8668 (1984)
- C14) Svantesson, I. et al.: "Distribution Ratios and Empirical Equations for the Extraction of Elements in the Purex High level Waste Solution", J. Inorg. Nucl. Chem. 41 (1979)
- C15) 村瀬武男: "湿式再処理に関するルテニウムの問題点", PNCT-6011 (1969)
- C16) "溶媒抽出プロセスに関する共同調査報告書", PNC ZN-8410 86-47 (1986)

付録D MIXSET-X組み込み化学反応の速度定数

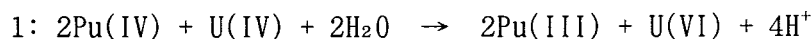
MIXSET-X組み込み化学反応の速度定数を以下に示す。速度式中の C_N は硝酸根濃度[mol/l]であり、 k 及び k' は化学反応の速度定数[mol/l・min]である。各々の値を表D-1、表D-2に示す。これらの反応速度定数はほとんどがEXTRA-Mに記載の値である。この表には、反応速度の温度依存性を与える活性化エネルギーの各々の値も記載しており、MIXSET-Xでは、反応1、反応4、反応5、反応41、反応42、反応43を除き、表中の値を使用している。

反応1の反応速度定数の温度係数を組み込まなかったのは、従来のMIXSETの速度定数が決定されたときの温度が不明であり、25°Cの値に換算できなかったためである。EXTRA-M自体には反応1に対応する活性化エネルギーの記載がある。また、反応4、反応5、反応41、反応42、反応43にはEXTRA-Mに活性化エネルギーの記載がなく、MIXSET-Xでも温度係数を組み込まなかった。

以下に示す化学反応速度式で左辺の記述のない式は、本来、例えば成分 y については

$$d[y]/dt = (yの化学量論数) \cdot (反応速度式) \quad (\text{mol/l} \cdot \text{min})$$
と書くべき所の(反応速度式)に相当する部分のみを記述したものである。

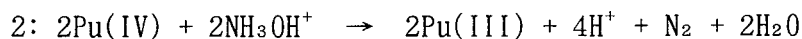
(1) 水相化学反応速度式



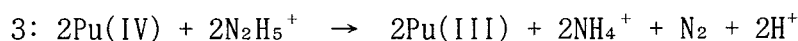
$$- \frac{d[\text{Pu(IV)}]}{dt} = 1.5 \times 10^2 \frac{[\text{U(IV)}][\text{Pu(IV)}]}{[\text{H}^+]^2} \quad (\text{mol/l} \cdot \text{min})$$

上記の反応速度式は従来のMIXSETで使用されているものである。EXTRA-Mでは少し異なる次の式を使用している。

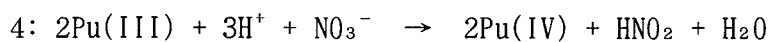
$$k_1 \frac{[\text{U(IV)}][\text{Pu(IV)}]}{[\text{H}^+ + 0.05]^2}$$



$$- \frac{d[\text{Pu(IV)}]}{dt} = 12.9 \frac{[\text{Pu(IV)}]^2 [\text{NH}_3\text{OH}^+]^2}{[\text{Pu(III)}]^2 [\text{H}^+]^4 \{1 + 4.7C_N\}^2} \quad (\text{mol/l} \cdot \text{min})$$



$$k_3 \frac{[\text{Pu(IV)}][\text{N}_2\text{H}_5^+]}{[\text{HNO}_3] + 0.35}$$



$$- \frac{d[\text{Pu(III)}]}{dt} = 5.5 \times 10^{-2} [\text{Pu(III)}] \quad (\text{mol/l}\cdot\text{min})$$

$$([\text{HNO}_2] \geq 2.3 \times 10^{-2} \text{M})$$

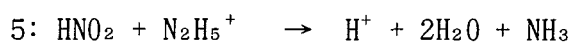
$$- \frac{d[\text{Pu(III)}]}{dt} = 10^{-(1.3 \log_{10}[\text{H}^+] + 0.54)} \times [\text{Pu(III)}]$$

$$\times [\text{HNO}_2]^{(0.44 - 0.76 \cdot \log_{10}[\text{H}^+])} \quad (\text{mol/l}\cdot\text{min})$$

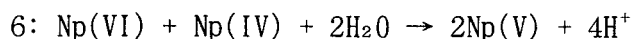
$$(2.3 \times 10^{-2} \text{M} > [\text{HNO}_2] \geq 10^{-4} \text{M})$$

$$- \frac{d[\text{Pu(III)}]}{dt} = 5.1 \times 10^{-3} [\text{Pu(III)}][\text{H}^+]^{1.8} \quad (\text{mol/l}\cdot\text{min})$$

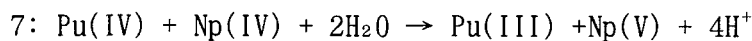
$$(10^{-4} \text{M} > [\text{HNO}_2])$$



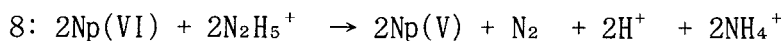
$$- \frac{d[\text{HNO}_2]}{dt} = 3.7 \times 10^4 [\text{H}^+][\text{HNO}_2][\text{N}_2\text{H}_5^+] \quad (\text{mol/l}\cdot\text{min})$$



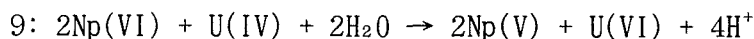
$$k_6[\text{Np(IV)}][\text{Np(VI)}](2.16 + 12.5C_N)$$



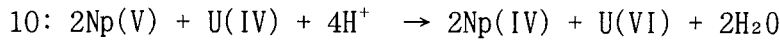
$$k_7[\text{Np(IV)}][\text{Pu(IV)}][\text{H}^+]^{-4}$$



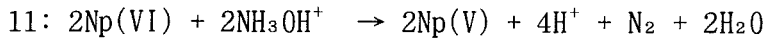
$$k_8[\text{Np(VI)}][\text{N}_2\text{H}_5^+][\text{H}^+]^{-1.3}$$



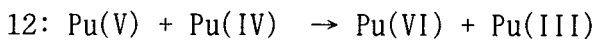
$$k_9[\text{Np(VI)}][\text{U(IV)}]$$



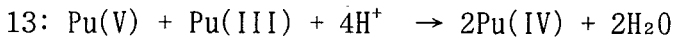
$$k_{10}[\text{Np(V)}][\text{U(IV)}]\{1.6[\text{H}^+]^{-2} + 1.42[\text{H}^+]\}$$



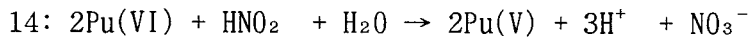
$$k_{11}[\text{Np(VI)}][\text{NH}_3\text{OH}^+][\text{H}^+]^{-1}$$



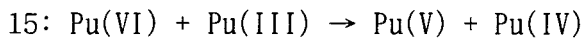
$$k_{12}[\text{Pu(V)}][\text{Pu(IV)}][\text{H}^+]\{[\text{H}^+] + 0.054\}^{-1}$$



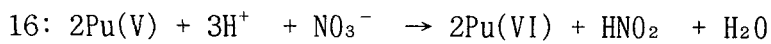
$$k_{13}[\text{Pu(V)}][\text{Pu(III)}][\text{H}^+]$$



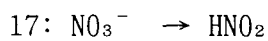
$$k_{14}[\text{Pu(VI)}][\text{HNO}_2][\text{H}^+]^{-1}$$



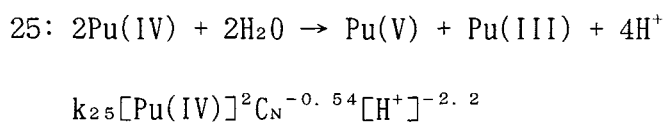
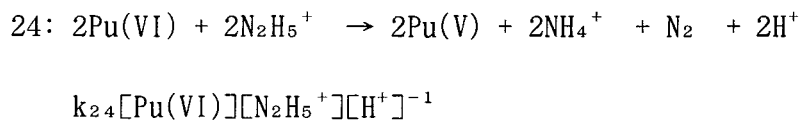
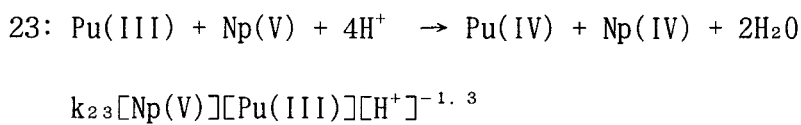
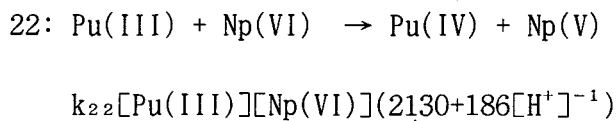
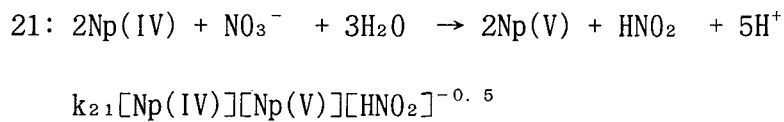
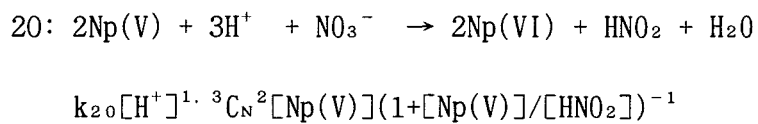
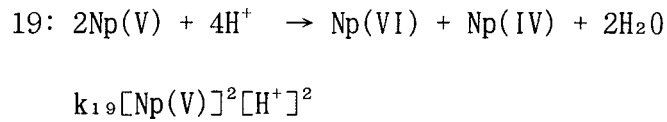
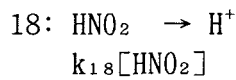
$$k_{15}[\text{Pu(VI)}][\text{Pu(III)}]$$

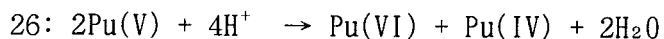


$$k_{16}[\text{Pu(V)}][\text{HNO}_2]^{0.5} \text{C}_\text{N}^{0.4} [\text{H}^+]^{0.6}$$

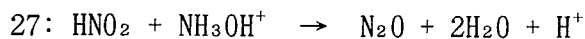


$$k_{17}$$

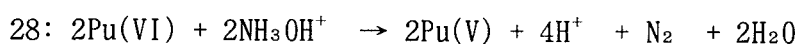




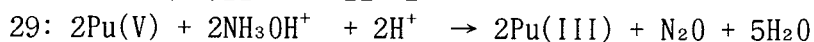
$$k_{26}[\text{Pu(V)}]^2[\text{H}^+]$$



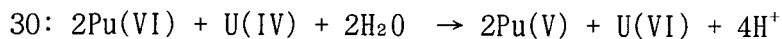
$$- \frac{d[\text{HNO}_2]}{dt} = 4.38 \times 10^2 [\text{H}^+][\text{HNO}_2][\text{NH}_3\text{OH}^+] \quad (\text{mol/l}\cdot\text{min})$$



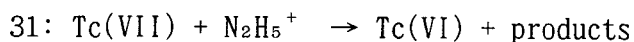
$$k_{28}[\text{Pu(VI)}][\text{NH}_3\text{OH}^+][\text{H}^+]^{-1}$$



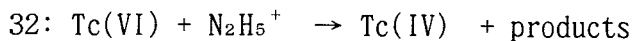
$$k_{29}[\text{Pu(V)}][\text{NH}_3\text{OH}^+]$$



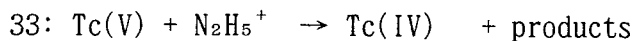
$$k_{30}[\text{Pu(VI)}][\text{U(IV)}][\text{H}^+]^{-1.2}$$



$$k_{31}[\text{Tc(VII)}][\text{N}_2\text{H}_5^+]$$



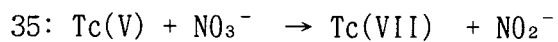
$$k_{32}[\text{Tc(VI)}][\text{N}_2\text{H}_5^+]$$



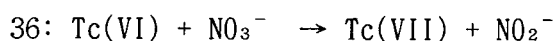
$$k_{33}[\text{Tc(V)}][\text{N}_2\text{H}_5^+]$$



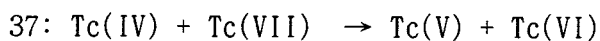
$$k_{34}[\text{Tc(IV)}]C_N$$



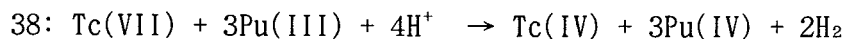
$$k_{35}[\text{Tc(V)}]C_N$$



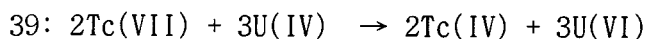
$$k_{36}[\text{Tc(VI)}]C_N$$



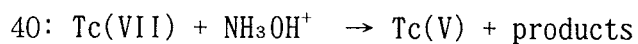
$$k_{37}[\text{Tc(IV)}][\text{Tc(VII)}]$$



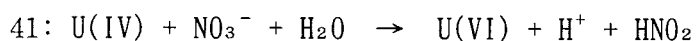
$$k_{38}[\text{Pu(III)}][\text{Tc(VII)}][\text{H}^+]^4$$



$$k_{39}[\text{U(IV)}][\text{Tc(VII)}][\text{H}^+]^2$$



$$k_{40}=10^{-7}$$

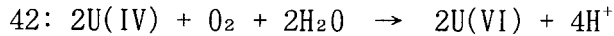


$$-\frac{d[\text{U(IV)}]}{dt} = 2.5 \times 10^{-2} [\text{U(IV)}][\text{HNO}_2]^{0.38}[\text{H}^+]^{2.7} \text{ (mol/l}\cdot\text{min)}$$

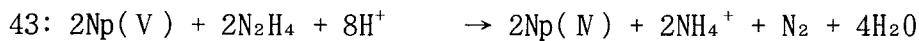
(0.8M > [H⁺])

$$-\frac{d[U(IV)]}{dt} = 1.3 \times 10^{-2} [U(IV)][HNO_2]^{0.38} \quad (\text{mol/l}\cdot\text{min})$$

$$([H^+] \geq 0.8M)$$



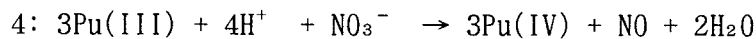
$$-\frac{d[U(IV)]}{dt} = 2.5 \times 10^{-4} \frac{[U(IV)]}{[H^+]} \quad (\text{mol/l}\cdot\text{min})$$



$$-\frac{d[Np(V)]}{dt} = 5.8 \times 10^{-4} [Np(V)][N_2H_5^+][H^+]^3 \quad (\text{mol/l}\cdot\text{min})$$

注) 亜硝酸を介してのプルトニウムの酸化(反応4)および亜硝酸のヒドラジンによる分解(反応5)は, MIXSETとEXTRA, Mで反応式が異なる。

MIXSET-Xには組み込んではいないが, EXTRA, Mの反応式を示す。

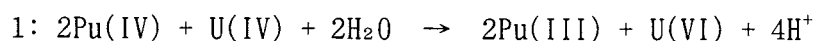


$$k_4[Pu(III)][HNO_2]^{0.5}[H^+]^{0.5}C_N^{0.4}$$



$$k_5[HNO_2][N_2H_5^+][H^+]$$

(2) 有機相化学反応速度式

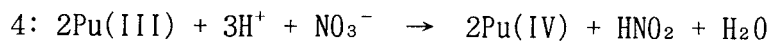


$$-\frac{d[Pu(IV)]}{dt} = 6.5 \frac{[U(IV)][Pu(IV)]}{[H^+]^2} \quad (\text{mol/l}\cdot\text{min})$$

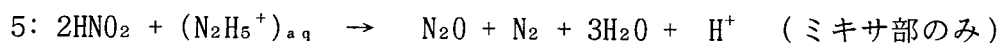
上記の反応速度式は従来のMIXSETで使用されているものである。

EXTRA. Mでは少し異なる次の式を使用している。

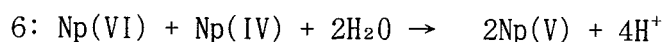
$$k_1 \frac{[U(M)][Pu(M)]}{[H^+ + 0.05]^2}$$



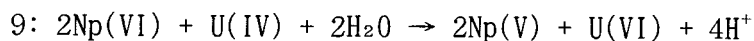
$$- \frac{d[Pu(III)]}{dt} = 0.15[Pu(III)][HNO_2][H^+]^{3.1} \quad (\text{mol/l}\cdot\text{min})$$



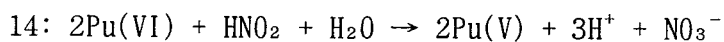
$$k_5 [HNO_2][N_2H_5^+]_{aq}^{0.2} [H^+]^{1.3}$$



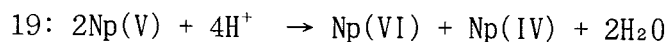
$$k_6 [Np(IV)][Np(VI)](2.16+12.5C_N)$$



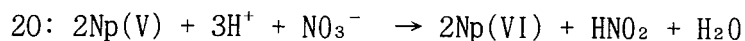
$$k_9 [Np(VI)][U(IV)]$$



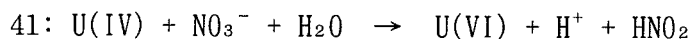
$$k_{14} \frac{[Pu(VI)][HNO_2]}{[H^+]}$$



$$k_{19} [Np(V)]^2 [H^+]^2$$



$$k_{20} \frac{[\text{HNO}_2][\text{Np(V)}]}{[\text{HNO}_2] + [\text{Np(V)}]} [\text{H}^+]^{1.3} \text{C}_N^2$$

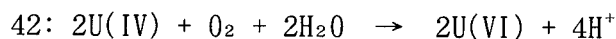


$$- \frac{d[\text{U(IV)}]}{dt} = 1.6 \times 10^{-2} [\text{U(IV)}][\text{HNO}_2]^{0.49} \quad (\text{mol/l}\cdot\text{min})$$

$$(0.34\text{M} \geq [\text{H}^+])$$

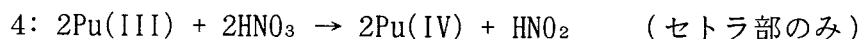
$$- \frac{d[\text{U(IV)}]}{dt} = 4.0 \times 10^{-2} [\text{U(IV)}][\text{H}^+]^{0.63} [\text{HNO}_2]^{0.49} \quad (\text{mol/l}\cdot\text{min})$$

$$([\text{H}^+] > 0.34\text{M})$$



$$- \frac{d[\text{U(IV)}]}{dt} = 0.0032 \frac{[\text{U(IV)}]}{[\text{H}^+]^{0.86}} \quad (\text{mol/l}\cdot\text{min})$$

注) 亜硝酸を介してのプルトニウムの酸化(反応4)は、従来のMIXSETとEXTRA、Mで反応式が異なる。MIXSET-Xには組み込んでいないが、EXTRA、Mの反応式を示す。



$$k_{4} \exp\{7.28[\text{H}^+] + 5.22\} [\text{Pu(III)}][\text{HNO}_2]$$

表D-1 MIXSET-Xで使用している化学量論数, 反応速度定数(k_i :水相 25°C) と活性化エネルギー(E_i) (1/2)
濃度変数の成分番号はMIXSET-Xで設定されているものである

反応式 j	HNO ₃	U(W)	Pu(F)	Pu(I)	U(F)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(F)	Tc(V)	Tc(W)	Tc(W)	k_i mol/min	$E_i/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
1	+2	+0.5	-1	+1	-0.5													150		
2	+2		-1	+1				-1										12.9	184	2
3			-1	+1			-1											0.038	93	3
4	-1.5		+1	-1		+0.5												0.0051 (0.0001 > [HNO ₂]) 0.055 (0.023 ≤ [HNO ₂]) (中間範囲は付録D詳細を参照)		
5	+1					-1	-1											37000		
6	+4										-1	+2	-1					2.5	103	8
7	+4		-1	+1							-1	+1						70	145	9
8							-1					+1	-1					8.3	70.3	10
9	+4	+1			-1							+2	-2					7		11
10	-4	+1			-1						+2	-2						2.25	40.2	12
11	+2							-1				+1	-1					92.1	82	13
12			-1	+1					-1	+1								250	59	14
13	-4		+2	-1					-1									37	43	15
14	+3					-1			+2	-2								7.9	111	16,17
15			+1	-1					+1	-1								276	23	*18,19
16	-3					+1			-2	+2								7.15	92	17
17	-1					+1												1×10^{-5}		
18	+1					-1												2×10^{-4}		
19	-4										+1	-2	+1					8×10^{-4}	68.7	20
20	-3					+1						-2	+2					9×10^{-4}	62.8	21
21	+5					+1					-2	+2						4.8×10^{-4}	105	21
22			+1	-1								+1	-1					1.0	17.2	*22

* HClO₄ system

表D-1 MIXSET-Xで使用している化学量論数、反応速度定数(k_i ;水相 25°C)と活性化エネルギー(E_i) (2/2)
濃度変数の成分番号はMIXSET-Xで設定されているものである

反応式 J	HNO ₃	U(W)	Pu(W)	Pu(I)	U(W)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(W)	Tc(V)	Tc(W)	Tc(W)	k_i mol/min	$E_i/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
23	-4		+1	-1							+1	-1						0.0598	85	23
24	+1						-1		+1	-1								0.314	67.4	24
25	+4		-2	+1					+1									1.65×10^{-3}	105	25
26	-4		+1						-2	+1								0.345	82*	*26
27	+1					-1		-1										437	51.9	27
28	+4							-2	+2	-2								0.153	93	*28,29
29	-2			+2				-2	-2									0.0664	73	*29
30	+4	+1			-1				+2	-2								372	80	*30
31							-1									+1	-1	6.6×10^{-5}	50	31
32							-1							+1		-1		2.5	50	31
33							-1							+1	-1			2.5	50	31
34	-1					+1								-1	+1			0.1	50	31
35	-1					+1									-1		+1	3.3×10^{-2}	50	31
36	-1					+1										-1	+1	6.6×10^{-3}	50	31
37														-1	+1	+1	-1	3.3×10^{-4}	50	31
38	-4		+3	-3										+1			-1	7.82×10^{-4}	29.7	32
39		+3			-3									+2			-2	1.39 ± 0.061	76.6	33
40								-1							+1		-1	1.0×10^{-7}	50	34
41	+1	+1			-1	+1												0.025 ($0.8 > [H^+]$) 0.013 ($0.8 \leq [H^+]$)		
42	+2	+1			-1													2.5×10^{-4}		
43	-4						-1				+1	-1						5.8×10^{-4}		
44	+3					-1						+2	-2					18		38
45	-1		+1	-1		-1												100		39

* HClO₄ system

表D-2 MIXSET-Xで使用している化学量論数, 反応速度定数(k_j :有機相 25°C)と活性化エネルギー(E_j)
濃度変数の成分番号はMIXSET-Xで設定されているものである

反応式 J	HNO ₃	U(W)	Pu(W)	Pu(I)	U(W)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(W)	Tc(V)	Tc(W)	Tc(W)	k_j' mol/min	$E_j/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
1	+2	+0.5	-1	+1	-0.5													6.5		
4	-1.5		+1	-1		+0.5												0.15		
5	+1					-2	-1 (水相)											250 (ミキサ部のみ)	79.5	36
6	+4										-1	+2	-1					2×10^{-4}	96	37
9	+4	+1			-1							+2	-2					7		
14	+3					-1			+2	-2								7.5		
19	-4										+1	-2	+1					0		
20	-3					+1						-2	+2					1	63	37
41	+1	+1			-1	+1												0.016 ($0.34 \geq [H^+]$)		
																		0.04 ($0.34 < [H^+]$)		
42	+2	+1			-1													0.0032		

化学反応速度定数の参考文献

- D1) Biddle P., Miles J. H., Waterman M. J.: J. inorg. nucl. Chem., 28, 1736 (1966).
D2) Barney G. S.: J. inorg. nucl. Chem., 38, 1677 (1976).
D3) Koltunov V. S., Zhuravleva G. I.: Radiokhimiya, 16, 84 (1974).
D4) Marchenko V. I. et al.: Radiokhimiya, 24, 331 (1982).
D5) Koltunov V. S., et al.: Radiokhimiya, 15, 777 (1973).
D6) Koltunov V. S., Marchenko V. I.: Kient. Katal., 7, 224 (1966).
D7) Biddle P., Miles J. H.: J. inorg. nucl. Chem., 30, 1291 (1968).
D8) Rykov. A. G., et al.: Radiokhimiya, 14, 128 (1972), and 8, 27 (1966).
D9) Koltunov V. S., et al.: Radiokhimiya, 18, 65 (1976).
D10) Koltunov V. S., Tikhnov. M. F.: Radiokhimiya, 15, 194 (1973).
D11) Silver G. L., Buzzelli O. R.: J. inorg. nucl. Chem., 30, 1317 (1968).
D12) Shilin I. V., Rumyantseva T. A.: Radiokhimiya, 20, 536 (1978).
D13) Koltunov V. S., Tikhonov M. F.: Radiokhimiya, 19, 611 (1977).
D14) Ryabova A. A., et al.: Radiokhimiya, 22, 254 (1980).
D15) Koltunov V. S., Mikhailova N. A.: Radiokhimiya, 19 (3), 342 (1977).
D16) Koltunov V. S., Zhuravleva G. I.: Radiokhimiya, 10, 662 (1968).
D17) Koltunov V. S., Ryabova A. A.: Radiokhimiya, 22, 635 (1980).
D18) Rabideau Sh. W., Kline R. J.: J. Phys. Chem., 62 (5), 617 (1958).
D19) Artyukhin P. I., et al.: Zh. Neorgan. Khim., 4, 1324 (1959).
D20) Escure H.: CEA-R-4574 (1974).
D21) Moulin J. P.: CEA-R-4912 (1978).
D22) Fulton R. B., Newton T. W.: J. Phys. Chem., 74, 1661 (1970).
D23) Koltunov V. S., et al.: Radiokhimiya, 23, 559 (1981), and 24, 607 (1987).
D24) Koltunov V. S., Zhuravleva G. I.: Radiokhimiya, 15, 74 (1973).
D25) Egorov G. F., et al.: Radiokhimiya, 23, 369 (1981).
D26) Koltunov V. S., et al.: Radiokhimiya, 22, 65 (1980).
D27) Barney G. S.: ARH-SA-97, (1971).
D28) Koltunov V. S., et al.: Radiokhimiya, 23 (4), 545 (1981).
D29) Koltunov V. S., et al.: Radiokhimiya, 23 (4), 552 (1981).
D30) Newton T. W.: J. Phys. Chem., 62, 943 (1958).
D31) Garraway J., Wilson P. D.: J. Less-Common Metals, 97, 191 (1984).
D32) Borovinskii V. A. et al.: Radiokhimiya, 23, 220 (1981).
D33) Koltunov V. S. et al.: Atomnaya Energiya, 60 (1), 35 (1986).
D34) Zelverte A.: CEA-R-5443 (1988).
D35) Petrich G., et al.: KfK 3740, p.227 (1984).
D36) Revyakin V. V., et al.: Radiokhimiya, 25, 617 (1983).
D37) Wehrey F., Guillaume B.: Radiochimica Acta, 46, 95 (1989).
D38) REACT-mod:JAERI-Data/Code 96-030 (1996).
D39) PLUTONIUM HANDBOOK ,ANS (1980).

付録E ルテニウム錯体の存在比計算式

ルテニウム錯体の存在比は、次の式で計算される。

①ジニトラ錯体

$$D I (H) = 1.61511 \times \left[-\exp \{-0.112066 \times ([H] + 0.61049)\} + \exp \{-0.0069 \times ([H]^2 + 0.85399)\} \right]$$

②トリニトラト錯体

$$T R (H) = 0.0799708 \times ([H] + 2.20226) \times \exp \{2.69315 / ([H] + 2.20226)\} - 0.586792$$

③ジニトロ錯体

$$N I (H) = 0.489439 \times \exp \{0.234136 / ([H] + 0.493768)\} - 0.488788$$

④モノニトラト錯体

$$M O (H) = 1.00 - D I (H) - T R (H) - N I (H)$$

[H]：水相中の硝酸濃度 X_H (N or mol/l)

(文献：Kjeller Report KR52 (1963))

付録F エラーメッセージ一覧

MIXSET-Xのエラーメッセージを表F-1及び表F-2に示す。表F-1は入力のエラーに関する物、表F-2は計算開始後のメッセージである。

表F-1 エラーメッセージ一覧 (1/3)

No.	KEY WORD	エラーメッセージ	サブルーチン	説明
1	TBP	FRC VALUE SPECIFIED IS INVALID.	input.f	TBP体積分率が0~10.以外。
2	DBP	CDBP MUST HAVE POSITIVE VALUE.	input.f	DBP濃度で負の値を指定した。
3	CONT	CALCULATION OPTION NOT AVAILABEL.	input.f	計算オプションICALCで0以外を指定した。
4	CONT	ISFIN IS INVALID.	input.f	溶液濃度オプションISFで0,1以外を指定した。
5	CONT	ENDTIM MUST BE POSITIVE.	input.f	過渡計算の計算終了時刻ENDTIMで0以下の値を指定した。
6	CONT	DT0 MUST BE POSITIVE.	input.f	過渡計算のタイムステップ初期値DT0で0以下の値を指定した。
7	STAG	STAGE ID MUST BE BETWEEN 1 TO 999.	input.f	ステージIDが0~999以外。
8	STAG	TOO MANY STAGES IN UNIT NO. **	input.f	抽出器の段数MSTGで25段を越える値を指定した。
9	STAG	TOO MANY EXTRACTION UNITS.	input.f	抽出器を最大数9基を越えて入力した。
10	STAG	TOO MANY TOTAL STAGES IN SYSTEM	input.f	ステージ数の合計が最大数(内部行列の最大ブロック数*300)以上。 *ミキサセトラのステージ数をNとすると内部ブロック数は2*N+1
11	TANK	TANK ID MUST BE BETWEEN 1 TO 999.	input.f	タンクのIDが0~999以外。
12	TANK	TANK PHASE SPECIFIER IS INVALID.	input.f	貯槽の相指定IPHXで1,-1以外を指定した。
13	TANK	TANK VOLUME IS INVALID.	input.f	貯槽の容量VTOTで0以下の値を指定した。
14	TANK	TANK FLOW RATE IS INVALID.	input.f	酸濃度調整時の水相貯槽の出口総流量FOUTで0以下の値を指定した。
15	TANK	ACID CONCENTRATION IS INVALID.	input.f	酸濃度調整時の目標酸濃度または調整酸の濃度範囲が0~13.5以外。
16	TANK	OXIDATION RATIO IS INVALID.	input.f	水相貯槽の酸化率OXIDが0~1.0以外。
17	ADDM	ADDMIC MUST BE DEFINED BEFORE FEED CARD.	input.f	ADDMICカードがFEEDカード以降に入力された。
18	ADDM	ADDMIC MUST BE DEFINED BEFORE INIT CARD.	input.f	ADDMICカードがINITIALカード以降に入力された。
19	ADDM	J VALUE SPECIFIED IS INVALID.	input.f	追加マイクロ番号Jが32~40以外。
20	ADDM	J VALUE IS ALREADY DEFINED.	input.f	追加マイクロ成分を重複して指定した。
21	ADDM	CHRGX MUST HAVE POSITIVE VALUE.	input.f	追加マイクロ成分の電荷で負の値を指定した。
22	ADDM	CNTBPX MUST HAVE POSITIVE VALUE.	input.f	追加マイクロ成分のTBP配位数で負の値を指定した。
23	ADDM	CVUNIT MUST HAVE POSITIVE VALUE.	input.f	追加マイクロ成分の単位変換係数で負の値を指定した。
24	DHEA	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~40以外。
25	CHAR	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~31以外。
26	CHAR	CHRG MUST HAVE POSITIVE VALUE.	input.f	成分の電荷として負の値を指定した。
27	CHAR	CNTBP MUST HAVE POSITIVE VALUE.	input.f	成分のTBP配位数として負の値を指定した。
28	VOLU	MIXER VOLUME MUST BE POSITIVE.	input.f	抽出器のミキサ容量Vで0以下の値を指定した。
29	VOLU	SETTLER VOLUME MUST BE POSITIVE.	input.f	抽出器のセトラ容量VSで0以下の値を指定した。
30	HEIG	HEIGHT MUST BE POSITIVE AND LESS THAN UNITY.	input.f	抽出器のセトラ部の界面レベルHEIGHTが0~1.0以外。
31	FEED	TOO LONG FEED TABLE.	input.f	時間依存データのセット数NSETで20を越える値を指定した。
32	FEED	TOO MANY FEED POINTS(AQUEOUS).	input.f	水相供給液の最大数30を越えた。
33	FEED	TOO MANY FEED POINTS(ORGANIC).	input.f	有機相供給液の最大数30を越えた。
34	FEED	I VALUE SPECIFIED IS INVALID.	input.f	供給液の流入するステージ番号Iが1~最終段の範囲外。
35	FEED	TIM IS INVALID.	input.f	供給液の時間tで0又は直前のt以下の値を指定した。
36	FEED	FLOW RATE IS INVALID.	input.f	時間tでの供給液流量FFで負の値を指定した。
37	FEED	NEGATIVE CONCENTRATION.	input.f	時間tでの供給液成分濃度XFで負の値を指定した。

表F-1 エラーメッセージ一覧 (2/3)

No.	KEY WORD	エラーメッセージ	サブルーチン	説明
38	RECY	RECYCLE FLOW MUST NOT BE NEGATIVE.	input.f	水相リサイクル流量FRで負の値を指定した。
39	EQUI	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~40以外。
40	EQUI	EQUILIBRIUM CONS. MUST BE POSITIVE.	input.f	平衡定数CKEQUで負の値を指定した。
41	CDIS	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~40以外。
42	CDIS	DISTRIBUTION COEFF. MUST BE POSITIVE.	input.f	分配係数CDISTで負の値を指定した。
43	SDIS	SDIST VALUE IS INVALID.	input.f	分配係数の計算式の選択が0,1,2,3,4以外。
44	REAC	J VALUE SPECIFIED IS INVALID.	input.f	化学反応番号Jが0または、-44~44以外。
45	REAC	FCHEM VALUE SPECIFIED IS INVALID.	input.f	反応速度調整因子FCHEMで負の値を指定した。
46	EFFI	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~40以外。
47	EFFI	IEFF IS INVALID.	input.f	回収率型段効率の定義方法IEFが0,1以外。
48	EFFI	EFF VALUE SPECIFIED IS NOT POSITIVE.	input.f	段効率EFFで負の値を指定した。
49	EFFI	EFFI FOR AQU. PHASE IS NOT PERMITTED.	input.f	Tc共抽出成分に対する水相定義の段効率は指定できない。
50	INIT	J VALUE SPECIFIED IS INVALID.	input.f	成分番号Jが1~40以外。
51	INIT	M VALUE SPECIFIED IS INVALID.	input.f	抽出器の成分濃度初期値の指定方法Mが0~4以外。
52	INIT	M VALUE SPECIFIED IS INVALID.	input.f	貯槽の成分濃度初期値の指定方法Mが0~2以外。
53	INIT	NEGATIVE CONCENTRATION.	input.f	成分濃度初期値Xで負の値を指定した。
54	PRIN	PRINTOUT TIMES MUST BE BETWEEN 1 TO 99.	input.f	出力時刻指定セット数NPRTが1~99以外。
55	PRIN	DTPRT MUST HAVE POSITIVE VALUE.	input.f	出力時刻DTPRTで負の値を指定した。
56	PLOT	PLOTOUT TIMES MUST BE BETWEEN 1 TO 99.	input.f	出力時刻指定セット数NPLTが1~99以外。
57	PLOT	DTPLT MUST HAVE POSITIVE VALUE.	input.f	出力時刻DTPLTで負の値を指定した。
58	EPSI	EPS MUST HAVE POSITIVE VALUE.	input.f	許容相対誤差目標値EPSで0以下の値を指定した。
59	EPSI	EPSLIM MUST HAVE POSITIVE VALUE.	input.f	許容相対誤差限界値EPSLIMで0以下の値を指定した。
60	EPSI	RXLIM MUST HAVE POSITIVE VALUE.	input.f	絶対誤差判定の限界濃度値RXLIMで0以下の値を指定した。
61	ITER	ITMIN MUST HAVE POSITIVE INTEGER VALUE.	input.f	濃度計算の収束計算反復上限数ITMAXで0以下の値を指定した。
62	ITER	ITMAX MUST HAVE POSITIVE INTEGER VALUE.	input.f	タイムステップTRATE倍するかどうかの限界反復回数ITMIXが0以下。
63	TAUL	DTMIN MUST HAVE POSITIVE VALUE.	input.f	タイムステップの最小値DTMINで0以下の値を指定した。
64	TAUL	DTMAX MUST HAVE POSITIVE VALUE.	input.f	タイムステップの最大値DTMAXで1以下の値を指定した。
65	TAUL	TRATE MUST BE GREATER THAN UNITY.	input.f	ITMAX以下の反復回数時のタイムステップ増大係数TRATEが1.0以下。
66	TAUC	TOO LONG TAUCONST TABLE.	input.f	タイムステップの一時的制限の指定セット数NSETが10以上。
67	TAUC	TMCONB MUST NOT HAVE NEGATIVE VALUE.	input.f	タイムステップの一時的制限開始時刻TMCONSBが負の値。
68	TAUC	TMCON.BIGIN LESS THAN TMCON.END.	input.f	タイムステップの一時的制限の開始時刻が終了時刻以前。
69	TAUC	DTCONS MUST HAVE POSITIVE VALUE.	input.f	タイムステップの一時的制限終了時刻TMCONSEが負の値。
70	-	KEY WORD IS NOT AVAILABLE.	input.f	入力されたキーワードは存在しない。
71	END	KEY WORD IS MISSING.	input.f	ENDカードが入力されていない。
72	-	IUNT VALUE SPECIFIED IS INVALID AT ISEXTR.	isextr.f	入力された機器番号は存在しない。
73	-	UNIT IS NOT EXTRACTOR.	isextr.f	入力された機器番号は抽出器でない。
74	FEED	UNIT ID OR STAGE OF AQUEOUS FEED NOT FOUND.	idchek.f	水相供給液の供給先が存在しない。
75	FEED	UNIT ID OR STAGE OF ORGANIC FEED NOT FOUND.	idchek.f	有機相供給液の供給先が存在しない。

表F-1 エラーメッセージ一覧 (3/3)

No.	KEY WORD	エラーメッセージ	サブルーチン	説明
76	FEED	UNIT ID OR STAGE(AQUEOUS) NOT FOUND.	idchek.f	機器の水相出口液の行き先で指定された供給先が存在しない。
77	FEED	UNIT ID OR STAGE(ORGANIC) NOT FOUND.	idchek.f	機器の有機相出口液の行き先で指定された供給先が存在しない。
78	FEED	ORGANIC FLOW RATE IS ZERO.	inpprt.f	ステージ1に有機相が供給されていないバンクがある。
79	FEED	AQUEOUS FLOW RATE IS ZERO.	inpprt.f	最終ステージに水相が供給されていないバンクがある。
80	-	ID NUMBER IS ALREADY DEFINED.	idconv.f	機器番号が重複している。
81	-	TOO MANY UNITS.	idconv.f	ミキサセトラと貯槽を合わせた機器の数が最大数19を越えた。
82	-	UNIT IS NOT DEFINED.	idconv.f	定義されていない機器IDに関する条件を指定した。
83	INITIAL	PROCESS SYSTEM IN PLOTFILE IS MISMATCH.	readpl.f	initial 0 で指定された初期濃度ファイル計算時と現在の体系が異なる。

表F-2 計算開始後のメッセージ一覧

No.	メッセージ	サブルーチン	説明
1	TOO MANY BLOCKS OF SYSTEM MATRIX	capmak.f	システムマトリックスの次元が最大値 (MXSYS=11) を越えた
2	TWO SUB-MATRICES OCUPY SAME POSITION.	capmak.f	同一の接続が入力された。2つある貯槽の出口を同じ機器の同じ場所に接続した
3	TOO MANY SOURCE TERM(AQUEOUS PHASE).	setrmp.f	水相の質量保存・濃度行列中の小行列が最大数 (MXRM=20) を越えた
4	TOO MANY SOURCE TERM(ORGANIC PHASE).	setrmp.f	有機相の質量保存・濃度行列中の小行列が最大数 (MXRM=20) を越えた
5	FLOW RATE EQUATION IS ILLPOSED(AQUEOUS).	calflw.f	水相の流量方程式を解くことができない。出口流量の配分方法が間違っている
6	FLOW RATE EQUATION IS ILLPOSED(ORGANIC).	calflw.f	有機相の流量方程式を解くことができない。出口流量の配分方法が間違っている
7	ERRMAC,XNMAX,DXMAX,J,K,I =	convck.f	ERRMAC 誤差判定に使用する濃度の相対誤差の最大値 XNMAX 誤差判定に使用する濃度の相対誤差の最大値が現れた場所での濃度値 DXMAX 誤差判定に使用する濃度の相対誤差の最大値が現れた場所での濃度変化値 J 誤差判定に使用する濃度の相対誤差の最大値が現れた成分番号 K 誤差判定に使用する濃度の相対誤差の最大値が現れた相 I 誤差判定に使用する濃度の相対誤差の最大値が現れた場所
8	(SUBR. LUDCOM) MATRIX IS SINGULAR AT	ludcom.f	行列がsingularとなって計算できない (後に続く数字は処理中の行を示す)
9	MATRIX SINGULAR.	tran.f	tran.fの行列がsingularとなって計算できない
10	***** DT BECAME LESS THAN DTMIN *****	tran.f	タイムステップを最小値DTMINとしても計算が指定反復回数以下で収束しない
11	TIME= ,STEP= ,NO.= ,ITER = ,SUM=,JMIC= ERROR DETECTED AND TIME STEP HAS BEEN CHANGED. 本メッセージは以下のいずれかの場合に出力される ・ニュートン法で収束しない場合 ・どれかの成分濃度の値が10M/Lを越えた場合 ・マトリックスがsingularの場合	tran.f	TIME 計算対象プロセスの時刻 DT タイムステップ NO ステップ数 ITER 現ステップの収束に必要としたニュートン法の反復回数 SUM 計算開始後の反復回数の積算数 JMIC ニュートン法で収束しなかった場合の成分番号N 0 の時はマクロ成分の計算でエラーとなった (マクロ成分のうちどの成分かについては特定できない) N エラーとなったマイクロ成分の番号 (N>21の場合は分配係数が異常である)
12	TIME = ,STEP = ,NO. =, ITER = ,ITERM= , SUM=	tran.f	ITERMの他は、No.11のメッセージと同じ No.11のITERはマクロ成分及び各マイクロ成分毎に行うニュートン法の収束に要した反復計算回数の最大値であるが、ITERMは弱い収束 (収束条件の緩やかな収束) に達した時の反復計算回数の最大値

付録G MIXSET-XとEXTRA. Mの化学反応式のちがい

MIXSET-Xには、EXTRA. Mで記載されているTcとNpに関する化学反応式の全てが組み込まれている。しかし、従来よりマクロ成分として計算している主な8成分に関するいくつかの反応については、反応機構の考え方が異なるなどのちがいがあるため、MIXSET-XとEXTRA. Mの化学反応は完全に一致してはいない。ここでは、2つのコードの化学反応のちがいを説明する。表G-1及び表G-2にEXTRA. Mで使用している化学量論数及び反応速度定数を示す。なお、これらの表はMIXSET-X組み込み化学反応の速度定数について述べた付録Dの表D-1及び行D-2に対応する。

(1) 反応機構の考え方にちがいがある反応

一つはプルトニウムの再酸化の反応である。

R. MIXSETの反応式



水相の反応速度式

$$\frac{d[\text{Pu(III)}]}{dt} = 5.5 \times 10^{-2} [\text{Pu(III)}] \quad (\text{mol/l}\cdot\text{min})$$

([HNO₂] ≥ 2.3×10⁻²M)

$$\frac{d[\text{Pu(III)}]}{dt} = 10^{-1.3 \log_{10}[\text{H}^+] + 0.54} \times [\text{Pu(III)}]$$

(0.44 - 0.76 · log₁₀[H⁺]) (mol/l·min)
(2.3×10⁻²M > [HNO₂] ≥ 10⁻⁴M)

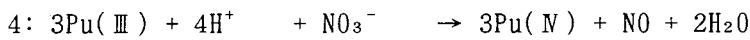
$$\frac{d[\text{Pu(III)}]}{dt} = 5.1 \times 10^{-3} [\text{Pu(III)}][\text{H}^+] 1.8 \quad (\text{mol/l}\cdot\text{min})$$

(10⁻⁴M > [HNO₂])

有機相の反応速度式

$$\frac{d[\text{Pu(III)}]}{dt} = 0.15[\text{Pu(III)}][\text{HNO}_2][\text{H}^+]^{3.1} \quad (\text{mol/l}\cdot\text{min})$$

EXTRA. Mの反応式



$$k_4[\text{Pu(III)}][\text{HNO}_2]^{0.5}[\text{H}^+]^{0.5}C_N^{0.4}$$

水相の速度定数k₄ 6.25(mixer)

70(settler)

有機相の速度定数k₄ 0.0(mixer)

0.5(settler)

MIXSET-XにはR. MIXSETの反応速度式が組み込まれている。

もう一つは、亜硝酸のヒドラジンによる分解反応である。

R. MIXSETの反応式



水相の反応速度式

$$k_5[\text{HNO}_2][\text{N}_2\text{H}_5^+][\text{H}^+]$$

$$k_5 = 37000$$

有機相の反応速度式

反応は起こらないとしている。

EXTRA. Mの反応式



水相の反応式

$$k_5[\text{HNO}_2][\text{N}_2\text{H}_5^+][\text{H}^+]$$

$$k_5 = 30000$$

有機相の反応速度式

ミキサ部界面での反応が起こるとしている。

$$k_5[\text{HNO}_2][\text{N}_2\text{H}_5^+]_{\text{aq}}^{0.2}[\text{H}^+]^{1.3}$$

$$k_5 = 250$$

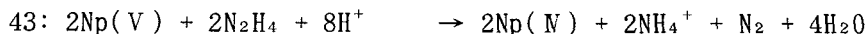
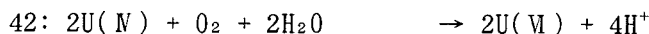
ただし、化学量論数がR. MIXSETとEXTRA. Mで異なる。

反応速度定数の単位は[mol/l·min]である。

MIXSET-Xでは水相の反応速度式としてR. MIXSETのものを使用し、有機相の反応を取り入れている。但し、STAGEカードのIREACオプションで1以外を指定すると有機相反応は考慮されない。

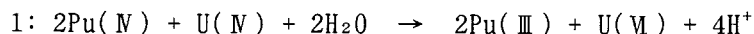
(2) EXTRA. Mに無くMIXSET-Xにある反応

次の3つの反応はEXTRA. Mでは考慮されていない反応である。



(3) 反応速度定数および反応速度式が異なる反応

次の反応は、R. MIXSETとEXTRA. Mで反応速度式及び反応速度定数が異なる反応である。



R. MIXSETの反応速度式

$$[\text{U}(\text{N})][\text{Pu}(\text{N})]$$

$$k_1 \frac{[\text{U}(\text{N})][\text{Pu}(\text{N})]}{[\text{H}^+]^2}$$

EXTRA. Mの反応速度式

$$[\text{U}(\text{N})][\text{Pu}(\text{N})]$$

$$k_1 \frac{[\text{U}(\text{N})][\text{Pu}(\text{N})]}{[\text{H}^+ + 0.05]^2}$$

反応速度定数は2つのコードで次のようになっている。

水相反応

R. MIXSETの値 75(温度不明)

EXTRA. Mの値 5000(25°C)

有機相反応

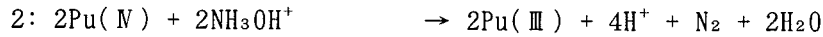
R. MIXSET の値 3.25(温度不明)

EXTRA. M の値 50(25°C)

ただし、化学量論数を EXTRA. M のものにあわせた値[mol/l·min]である。

(4) 反応速度定数のみが異なる反応

次の2つの化学反応については、2つのコードで反応速度式は同じで、反応速度定数のみが異なる。



この2つの反応については、R. MIXSET に20°Cの反応速度定数であるとの記述があり、これを25°Cに換算した反応速度定数を組み込んで同時に温度依存性の計算式を導入した。2と27の化学反応のR. MIXSETの化学反応速度定数を25°Cに換算した値は、EXTRA. Mの値と少し異なるがほとんど同じである。MIXSET-XではR. MIXSETで引用している文献(HEDL-TME7531及びARH-SA-97)に記載されている反応速度定数および温度依存性を組み込んでいる。2つのコードの反応速度式の違いは次の通りである。

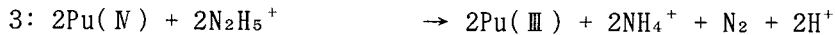
2の化学反応速度定数のR. MIXSETの値 0.584(25°Cに換算した値)

2の化学反応速度定数のEXTRA. Mの値 0.732

27の化学反応速度定数のR. MIXSETの値 438(25°Cに換算した値)

27の化学反応速度定数のEXTRA. Mの値 436

また、次の化学反応の化学反応速度定数は、MIXSET 98では計算による感度解析からの推定値が記載されているが、EXTRA. Mでは実験に基づく値が記載されており、EXTRA. Mの値を採用することにした。



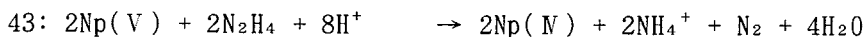
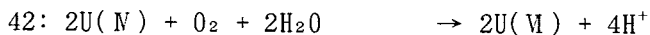
3の化学反応速度定数のR. MIXSETの値 1/60(温度不明)

3の化学反応速度定数のEXTRA. Mの値 0.038

試みに、第4抽出器の計算を3の反応速度定数を0として計算を行ってみたところ、計算結果にはほとんど影響が無かった。標準のプロセスではあまり重要でない化学反応と思われる。ただし、反応速度定数の単位は[mol/l·min]である。

(5) 反応速度の温度依存性が組み込めなかった反応

MIXSET-XにはEXTRA. Mに記載されているほとんどの反応速度定数の温度依存性が組み込まれている。ただし、次の5つの化学反応の反応速度定数の温度依存性は導入していない。



反応1については、最初にR. MIXSETを作成したときに決定された反応速度定数について温度の記述がないため、その定数が何度のときの反応速度定数かが不明であり、温度依存性を入れることは適当でないと判断したためである。4, 5, 41, 42および43の反応については、1と同様にR. MIXSETを作成したときに決定された反応速度定数について温度の記述がなく、さらに、EXTRA. Mには無い反応であり、温度依存性計算に用いる活性化エネルギーが不明なためである。

表G-1 EXTRA-Mで使用している化学量論数、反応速度定数(k_j :水相 25°C)と活性化エネルギー(E_j) (1/2)
濃度変数の成分番号はM I X S E T - Xで設定されているものである

反応式 j	U(W)	Pu(W)	HNO ₃	U(W)	Pu(W)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(W)	Tc(V)	Tc(W)	Tc(W)	k_j mol/min	$E_j/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
1	+1	-2	+4	-1	+2													5000	104	1
2		-1	+2		+1			-1										0.732	130	2
3		-1			+1		-1											0.038	93	3
4		+2	-2		-2	+1												M:6.25 S:0.70	35 60.7	4 5
5			+1			-2	-1											30000	36	6,7
6			+4								-1	+2	-1					2.5	103	8
7		-1	+4		+1						-1	+1						70	145	9
8							-1					+1	-1					8.3	70.3	10
9	+1		+4	-1								+2	-2					7		11
10	+1		-4	-1							+2	-2						2.25	40.2	12
11			+2					-1				+1	-1					92.1	82	13
12		-1			+1				-1	+1								250	59	14
13		+2	-4		-1				-1									37	43	15
14			+3			-1			+2	-2								7.9	111	16,17
15		+1			-1				+1	-1								276	23	*18,19
16			-3			+1			-2	+2								7.15	92	17
17			-1			+1												M:0 S:1×10 ⁻⁵		
18			+1			-1												M:0 S:2×10 ⁻⁴		
19			-4								+1	-2	+1					8×10 ⁻⁴	68.7	20
20			-3			+1						-2	+2					9×10 ⁻⁴	62.8	21

* HClO₄ system

(M:Mixing part, S:Settling part)

表G-1 EXTRA-Mで使用する化学量論数, 反応速度定数(k_j :水相 25°C)と活性化エネルギー(E_j) (2/2)
濃度変数の成分番号はM I X S E T - Xで設定されているものである

反応式 J	U(W)	Pu(W)	HNO ₃	U(W)	Pu(W)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(W)	Tc(V)	Tc(W)	Tc(W)	k_j mol/min	$E_j/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
21			+5			+1					-2	+2						4.8×10^{-4}	105	21
22		+1			-1							+1	-1					1.0	17.2	*22
23		+1	-4		-1						+1	-1						0.0598	85	23
24			+1				-1		+1	-1								0.314	67.4	24
25		-2	+4		+1				+1									1.65×10^{-3}	105	25
26		+1	-4						-2	+1								0.345	82*	*26
27			+1			-1		-1										436	48.1	27
28			+4					-2	+2	-2								0.153	93	*28,29
29			-2		+2			-2	-2									0.0664	73	*29
30	+1		+4	-1					+2	-2								372	80	*30
31							-1									+1	-1	6.6×10^{-5}	50	31
32							-1							+1		-1		2.5	50	31
33							-1							+1	-1			2.5	50	31
34			-1			+1								-1	+1			0.1	50	31
35			-1			+1									-1		+1	3.3×10^{-2}	50	31
36			-1			+1										-1	+1	6.6×10^{-3}	50	31
37														-1	+1	+1	-1	3.3×10	50	31
38		+3	-4		-3									+1			-1	7.82×10^{-4}	29.7	32
39	+3			-3										+2			-2	1.39 ± 0.061	76.6	33
40								-1							+1		-1	1.0×10^{-7}	50	34

表G-2 EXTRA-Mで使用している化学量論数, 反応速度定数(k_i :有機相 25°C) と活性化エネルギー(E_i)
濃度変数の成分番号はM I X S E T - Xで設定されているものである

反応式 j	U(W)	Pu(W)	HNO ₃	U(W)	Pu(W)	HNO ₂	Hyd	HAN	Pu(V)	Pu(W)	Np(W)	Np(V)	Np(W)	Tc(W)	Tc(V)	Tc(W)	Tc(W)	k_i' mol/min	$E_i/10^3$ kJ/mol	文献
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₅	X ₁₆	X ₁₇	X ₁₈			
1	+1	-2	+4	-1	+2													50		35
4		+2	-2		-2	+1												M:0 S:0.5		35
5			+1			-2	-1 (水相)											M:250 S:0	79.5	36
6			+4								-1	+2	-1					2×10^{-4}	96	37
9	+1		+4	-1								+2	-2					7		
14			+3			-1			+2	-2								7.5		
19			-4								+1	-2	+1							
20			-3			+1						-2	+2					1	63	37

(M:Mixing part, S:Settling part)

付録H 容量マトリックス法

容量マトリックス法は、3重対角行列の帯状部分からはずれたところいくつかのゼロでない行列要素がある疎行列の解を求めるのに適した行列解法である。解法は次のような手順となる。次の一次連立方程式を考える。

$$A x = b \quad (H-1)$$

行列Aは3重対角に近いものとする。そこで、行列Aを3重対角行列部分Bとそれ以外の部分E・Rに分解する。以下に6行6列の例を挙げる。

$$A = \begin{pmatrix} XX0000 \\ XXX0T0 \\ OXXX00 \\ OOXXX0 \\ ROSXXX \\ 0000XX \end{pmatrix}$$

$$B = \begin{pmatrix} XX0000 \\ XXX000 \\ OXXX00 \\ OOXXX0 \\ 000XXX \\ 0000XX \end{pmatrix} \quad E \cdot R = \begin{pmatrix} 000000 \\ 0000T0 \\ 000000 \\ 000000 \\ ROS000 \\ 000000 \end{pmatrix}$$

$$E = \begin{pmatrix} 00 \\ 01 \\ 00 \\ 00 \\ 10 \\ 00 \end{pmatrix} \quad R = \begin{pmatrix} ROS000 \\ 0000T0 \end{pmatrix}$$

$$A = B + E \cdot R \quad (H-2)$$

$$(B + E \cdot R) x = b \quad (H-3)$$

式H-3の両辺に $R \cdot B^{-1}$ をかけると次の式を得る。

$$(R + R \cdot B^{-1} E \cdot R) x = R \cdot B^{-1} b \quad (H-4)$$

ここで、

$$y = R x$$

とおくと、yに関する次の方程式を得る。

$$(1 + R \cdot B^{-1} E) y = R \cdot B^{-1} b \quad (H-5)$$

yが求まると、xは式(H-3)より、次の式で計算される。

$$x = B^{-1} b - B^{-1} \cdot E y \quad (H-6)$$

この計算方法の利点は、 y に関する方程式の次数が元の方程式の次数より少なくなっていることである。例であげた行列で言えば次数は6から2になっている。計算量は次数の3乗に比例するため、3重対角の帯状部分以外のゼロでない要素が少なければ、この解法により大幅に計算時間を短縮することが可能である。再処理プロセスを計算モデル化して得られる行列はちょうどこのような形の行列となり、容量マトリックス法は非常に有効である。

(文献, R. G. Steinke, J. F. Dearing: "Capacitance Matrix Method in TRAC and MELPROG", LA-UR 89-264 (1989))

付録 I Solute-Free濃度と通常濃度の変換式

Solute-Free濃度（従来のMIXSETでは中性基準濃度と呼んでいる）は溶媒中に有効成分が全くないときの溶媒体積で溶媒中の溶質の量を割ったものであり、単位は[mol/L]である。Richardsonの分配係数式により分配係数を計算する際には、Solute-Free濃度に変換してから計算することが望ましい。以下に、通常濃度とSolute-Free濃度の変換式を示す。

本式はHEDL-TME-75 31 のものを多少修正したものになっている。斜体文字が通常濃度である。Tは温度[K]、 C_{TBP} はTBPのモル濃度、FはTBPの体積割合をあらわす。

（文献 G.L.Richardson: "Effect of High Solvent Irradiation Exposure on TBP Processing of Spent LMFBR Fuels", HEDL-TME 73-51(1973)）

(1) 通常濃度（モル濃度）からSolute-Free濃度への変換

水相

$$X_i = \frac{X_i}{1 - 0.0309X_{H^+} - 0.00724(X_{U6} + X_{U4}) - 0.13X_{Pu4} - 0.1X_{Pu3} - 0.03X_{HYD} - 0.04X_{HAN}}$$

有機相

$$Y_i = \frac{Y_i}{1 - 0.043Y_{H^+} - 0.092(Y_{U6} + Y_{U4}) - 0.139(Y_{Pu4} + Y_{Pu3}) - 0.0174W_{CO}}$$

ただし、

$$W_{LT} = 3.95 - 0.0144(T - 273.15)$$

$$U_{os} = 0.5 \frac{C_{TBP}}{1 + 0.046C_{TBP}}$$

$$P_{os} = 0.5 \frac{C_{TBP}}{1 + 0.070C_{TBP}}$$

$$H_{os} = \frac{C_{TBP}}{1 + 0.043C_{TBP}} (1 - 0.00609W_{LT}F^{1.65})$$

$$W_{CO} = W_{LT}F^{1.65} \left[1 - \frac{(Y_{U6} + Y_{U4})}{U_{os}} - \frac{(Y_{Pu4} + Y_{Pu3})}{P_{os}} - 0.65 \frac{Y_{H^+}}{H_{os}} \right]$$

(2) Solute-Free濃度から通常濃度（モル濃度）への変換

水相

$$X_i = \frac{X_i}{1 + 0.0309X_{H^+} + 0.00724(X_{U6} + X_{U4}) + 0.13X_{Pu4} + 0.1X_{Pu3} + 0.03X_{HYD} + 0.04X_{HAN}}$$

有機相

$$Y_i = \frac{Y_i}{1 + 0.043Y_{H^+} + 0.092(Y_{U6} + Y_{U4}) + 0.139(Y_{Pu4} + Y_{Pu3}) + 0.0174W_{CO}}$$

ただし、

$$W_{ST} = 4.2 - 0.015(T - 273.15)$$

$$W_{CO} = W_{ST}F^{1.69} \left[1 - \frac{2(Y_{U6} + Y_{U4} + Y_{Pu4} + Y_{Pu3}) + 0.6Y_{H^+}}{C_{TBP}} \right]$$

付録 J 抽出器の定常計算のまとめ

MIXSET-Xの定常状態の物質収支式が正しいかどうかを確認することを目的として、再処理工場の第1抽出器から第9抽出器を計算対象として定常計算を行った。計算条件を表J-1～表J-9に、計算結果を表J-10～表J-18に示す。ただし、現状のMIXSET-Xには定常計算機能がないため、長時間(5000時間)の過渡計算を行い、最終結果を定常計算結果としている。この計算では、化学反応はMIXSET98相当のもののみを制限し、通常濃度近似で計算している。計算された結果の物質収支はいずれの場合にもよく合っていることが確認された。また、ウランの蓄積効果がある第1抽出器の濃度が蓄積の起こる段で違うもの、概ね計算結果は従来のMIXSETとMIXSET-Xの計算結果は数値計算誤差の範囲内で一致している。

表 J-1 第1抽出器の計算条件

段数	17			
容積	ミキサ部	12.0 L		
	セトラ部	1~9段	74.1 L	
		10~17段	93.8 L	
循環流量		10~17段	198.0 L/h	
温度		1~9段	25.0 °C	
		10~17段	30.0 °C	
フィード条件				
1段	有機相	300.0 L/h	有効成分なし	
9段	水相	155.7 L/h	成分	
			硝酸	3.0 mol/L
			U(VI)	180.0 g/L
			Pu(IV)	1.4 g/L
17段	水相	66.5 L/h	成分	
			硝酸	3.0 mol/L

表 J-2 第2抽出器の計算条件

段数	12			
容積	ミキサ部	12.0 L		
	セトラ部	99.2 L		
循環流量		1~12段	0.0 L/h	
温度		30.0, 31.7, 33.3, 35.0, 37.5, 40.0, 41.7, 43.3, 45.0, 46.7, 48.3, 50.0 °C		
フィード条件				
1段	有機相	300.0 L/h	成分	
			硝酸	0.184636 mol/L
			U(VI)	93.9457 g/L
			Pu(IV)	0.73058 g/L
4段	水相	3.2 L/h	成分	
			硝酸	11.0 mol/L
12段	水相	370.0 L/h	成分	
			硝酸	0.02 mol/L

表 J-3 第3抽出器の計算条件

段数	21			
容積	ミキサ部	1~14段	21.9 L	
		15~21段	18.0 L	
	セトラ部	1~14段	147.3 L	
		15~21段	107.4 L	
循環流量		15~21段	300.0 L/h	
温度		1~21段	30.0 °C	
フィード条件				
1段	有機相	370.0 L/h	有効成分なし	
14段	水相	500.9 L/h	成分	
			硝酸	3.0 mol/L
			U(VI)	56.26 g/L
			Pu(IV)	0.4375 g/L
21段	水相	50.0 L/h	成分	
			硝酸	3.0 mol/L

表 J - 4 第4抽出器の計算条件

段数	1 1			
容積	ミキサ部	1 ~ 4段	5. 4 4 L	
		5 ~ 11段	1 8. 0 L	
	セトラ部	1 ~ 4段	3 6. 1 L	
		5 ~ 11段	1 4 2. 5 L	
循環流量	5 ~ 11段	3 0 0. 0 L/h		
温度	1 ~ 2 1段	3 0. 0 ° C		
亜硝酸の分配係数	1 0. 0			
フィード条件				
1段	有機相	3 0. 0 L/h	有効成分なし	
4段	有機相	5 2. 5 L/h	成分	
			硝酸	0. 0 2 4 mol/L
			U(VI)	2 5. 0 g /L
			Pu(IV)	0. 0 g /L
			Pu(III)	0. 0 g /L
			U(IV)	0. 2 4 g /L
5段	有機相	3 7 0. 0 L/h	成分	
			硝酸	0. 2 7 4 5 N
			U(VI)	7 6. 1 7 g /L
			Pu(IV)	0. 5 9 2 3 g /L
			Pu(III)	0. 0 g /L
			U(IV)	0. 0 g /L
5段	水相	7. 6 L/h	成分	
			硝酸	1. 5 mol/L
			U(VI)	8 0. 0 g /L
			U(IV)	1 2 0. 0 g /L
			HYD	6. 0 g /L
7段	水相	2. 2 L/h	成分	
			硝酸	1. 5 mol/L
			U(VI)	8 0. 0 g /L
			U(IV)	1 2 0. 0 g /L
			HYD	6. 0 g /L
8段	水相	1. 0 7 L/h	成分	
			硝酸	1. 5 mol/L
			U(VI)	8 0. 0 g /L
			U(IV)	1 2 0. 0 g /L
			HYD	6. 0 g /L
1 0段	水相	0. 0 L/h	成分	
			硝酸	1. 5 mol/L
			U(VI)	8 0. 0 g /L
			U(IV)	1 2 0. 0 g /L
			HYD	6. 0 g /L
1 1段	水相	1 2 0. 0 L/h	成分	
			硝酸	0. 2 mol/L
			HYD	6. 0 g /L

表 J-5 第5抽出器のデータ

段数	9						
容積	ミキサ部	21.9	L				
	セトラ部	157.3	L				
循環流量		5~9段	300.0	L/h			
温度		35.0, 36.3, 37.5, 38.8, 40.0, 42.5,					
		45.0, 47.5, 50.0					°C
亜硝酸の分配係数		10.0					
フィード条件							
1段	有機相	452.5	L/h	成分			
				硝酸	0.0161	mol/L	
				U(VI)	67.757	g/L	
9段	水相	560.0	L/h	成分			
				Pu(IV)	0.0	g/L	
				硝酸	0.02	mol/L	

表 J-6 第6抽出器のデータ

段数	12						
容積	ミキサ部	1~6段	27.0	L			
		7~12段	18.0	L			
	セトラ部	1~6段	174.5	L			
		7~12段	122.0	L			
循環流量		7~12段	300.0	L/h			
温度		30.0		°C			
亜硝酸の分配係数		10.0					
フィード条件							
1段	有機相	400.0	L/h	有効成分なし			
6段	水相	646.5	L/h	成分			
				硝酸	1.5	mol/L	
				U(VI)	46.64	g/L	
				Pu(IV)	0.0	g/L	
10段	水相	6.0	L/h	成分			
				硝酸	13.5	N	
12段	水相	80.5	L/h	成分			
				硝酸	0.029	mol/L	
				U(VI)	0.4969	g/L	
				U(IV)	0.7453	g/L	
				HYD	0.03727	g/L	

表 J-7 第7抽出器のデータ

段数	7						
容積	ミキサ部	18.0	L				
	セトラ部	134.0	L				
循環流量		1~7段	0.0	L/h			
温度			35.0, 36.7, 38.3, 40.0, 43.3, 46.7, 50.0				°C
フィード条件							
1段	有機相	402.0	L/h	成分			
				硝酸	0.009		mol/L
				U(VI)	75.54		g /L
				U(IV)	0.014		g /L
7段	水相	560.0	L/h	成分			
				硝酸	0.02		mol/L

表 J-8 第8抽出器のデータ

段数	15						
容積	ミキサ部	1段~9段	5.4	L			
		10段~15段	3.08	L			
	セトラ部	1段~9段	30.73	L			
		10段~15段	12.47	L			
循環流量		1段~15段	0.0	L/h			
温度			30.0	°C			
フィード条件							
1段	有機相	41.0	L/h	有効成分なし			
9段	水相	171.2	L/h	成分			
				硝酸	4.0		mol/L
				U(VI)	6.587		g /L
				Pu(IV)	1.28		g /L
15段	水相	21.0	L/h	成分			
				硝酸	1.0		mol/L

表 J-9 第9抽出器のデータ

段数	13				
容積	ミキサ部	1~5段	1.3	L	
		6~13段	3.08	L	
	セトラ部	1~5段	5.09	L	
		6~13段	12.47	L	
循環流量		6~13段	50.0	L/h	
温度		1~21段	30.0	°C	
ウラサスの分配係数	0.333				
亜硝酸の分配係数	10.0				
フィード条件					
1段	有機相	11.5	L/h	有効成分なし	
6段	有機相	41.0	L/h	成分	
				硝酸	0.148 mol/L
				U(VI)	27.5 g/L
				Pu(IV)	5.343 g/L
7段	水相	1.5	L/h	成分	
				硝酸	1.5 mol/L
				U(VI)	80.0 g/L
				U(IV)	120.0 g/L
				HYD	6.0 g/L
9段	水相	0.0	L/h	成分	
				硝酸	1.5 mol/L
				U(VI)	80.0 g/L
				U(IV)	120.0 g/L
				HYD	6.0 g/L
11段	水相	0.5	L/h	成分	
				硝酸	1.5 mol/L
				U(VI)	80.0 g/L
				U(IV)	120.0 g/L
				HYD	6.0 g/L
13段	水相	15.0	L/h	成分	
				硝酸	0.2 mol/L
				HYD	6.0 g/L

表 J-10 第1抽出器の計算結果
(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	2.74794E+00	3.40567E-10	7.47022E-09	6.00247E-01	8.23361E-09	6.82660E-08
2	3.55872E+00	1.14621E-08	9.96800E-08	7.44594E-01	3.55394E-07	1.28520E-06
3	3.75370E+00	4.80386E-07	1.74345E-06	7.74821E-01	1.54938E-05	2.39758E-05
4	3.79452E+00	2.09284E-05	3.23927E-05	7.80911E-01	6.80005E-04	4.51167E-04
5	3.80275E+00	9.18512E-04	6.09418E-04	7.81920E-01	2.98741E-02	8.50569E-03
6	3.80411E+00	4.03523E-02	1.14890E-02	7.74190E-01	1.29006E+00	1.57657E-01
7	3.79367E+00	1.74253E+00	2.12955E-01	5.79149E-01	3.37388E+01	1.77149E+00
8	3.53022E+00	4.55724E+01	2.39283E+00	1.65744E-01	1.02188E+02	2.34879E+00
9	2.97182E+00	1.38030E+02	3.17262E+00	8.25435E-02	1.16381E+02	1.20831E+00
10	2.53053E+00	1.03856E+02	2.17521E+00	9.02727E-02	1.12465E+02	1.12042E+00
11	2.56540E+00	8.61900E+01	1.77874E+00	1.01132E-01	1.10058E+02	1.06451E+00
12	2.61439E+00	7.53292E+01	1.52650E+00	1.10299E-01	1.08249E+02	1.02230E+00
13	2.65575E+00	6.71676E+01	1.33609E+00	1.18572E-01	1.06648E+02	9.85068E-01
14	2.69307E+00	5.99481E+01	1.16813E+00	1.27104E-01	1.05000E+02	9.47060E-01
15	2.73156E+00	5.25098E+01	9.96664E-01	1.37441E-01	1.02996E+02	9.02014E-01
16	2.77819E+00	4.34699E+01	7.93445E-01	1.52946E-01	9.99756E+01	8.38733E-01
17	2.84814E+00	2.98446E+01	5.07966E-01	1.86608E-01	9.33600E+01	7.26133E-01

表 J-11 第2抽出器の計算結果
(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	2.62389E-01	7.55191E+01	5.87283E-01	1.97617E-02	7.53570E+01	2.96782E-01
2	1.29854E-01	6.05764E+01	2.38571E-01	1.11682E-02	5.67849E+01	1.16922E-01
3	1.22946E-01	4.56470E+01	9.39891E-02	1.14574E-02	4.06472E+01	4.52917E-02
4	1.23178E-01	3.26746E+01	3.64081E-02	1.17310E-02	2.54570E+01	1.59462E-02
5	2.93301E-02	2.06408E+01	1.29294E-02	2.06992E-03	7.39912E+00	2.69730E-03
6	2.14968E-02	5.99928E+00	2.18700E-03	7.02597E-04	4.52115E-01	9.97124E-05
7	2.03881E-02	3.66580E-01	8.08479E-05	2.66353E-04	4.63451E-03	6.41393E-07
8	2.00344E-02	3.75771E-03	5.20049E-07	2.29081E-04	3.58382E-05	3.23584E-09
9	2.00042E-02	2.90580E-05	2.62366E-09	2.27129E-04	2.66760E-07	1.62898E-11
10	2.00026E-02	2.16292E-07	1.32080E-11	2.26004E-04	1.91630E-09	8.22269E-14
11	2.00017E-02	1.55368E-09	6.66687E-14	2.24982E-04	1.33210E-11	4.16110E-16
12	2.00009E-02	1.07288E-11	3.35683E-16	2.23904E-04	8.88371E-14	2.10085E-18

表 J-12 第3抽出器の計算結果
(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	2.81666E+00	1.55220E-15	5.15993E-13	6.09900E-01	3.42793E-14	5.03868E-12
2	3.22629E+00	2.45751E-14	3.90012E-12	6.85373E-01	6.24007E-13	4.58948E-11
3	3.27698E+00	4.20653E-13	3.13403E-11	6.94214E-01	1.08376E-11	3.76486E-10
4	3.28292E+00	7.28036E-12	2.53374E-10	6.95242E-01	1.87882E-10	3.05101E-09
5	3.28361E+00	1.26188E-10	2.04966E-09	6.95361E-01	3.25712E-09	2.46879E-08
6	3.28369E+00	2.18758E-09	1.65816E-08	6.95375E-01	5.64661E-08	1.99729E-07
7	3.28370E+00	3.79243E-08	1.34144E-07	6.95377E-01	9.78911E-07	1.61581E-06
8	3.28370E+00	6.57464E-07	1.08522E-06	6.95377E-01	1.69706E-05	1.30718E-05
9	3.28370E+00	1.13980E-05	8.77941E-06	6.95375E-01	2.94206E-04	1.05750E-04
10	3.28369E+00	1.97597E-04	7.10247E-05	6.95347E-01	5.10007E-03	8.55453E-04
11	3.28368E+00	3.42535E-03	5.74546E-04	6.94893E-01	8.83154E-02	6.91271E-03
12	3.28337E+00	5.93151E-02	4.64277E-03	6.87447E-01	1.50256E+00	5.48844E-02
13	3.27837E+00	1.00916E+00	3.68619E-02	5.91577E-01	1.99524E+01	3.40252E-01
14	3.21398E+00	1.34006E+01	2.28523E-01	2.86343E-01	7.80212E+01	6.24413E-01
15	3.09891E+00	1.37440E+01	2.37784E-01	2.77153E-01	7.80521E+01	6.25224E-01
16	3.03091E+00	1.39727E+01	2.43782E-01	2.71566E-01	7.80722E+01	6.25744E-01
17	2.98956E+00	1.41218E+01	2.47634E-01	2.68115E-01	7.80851E+01	6.26074E-01
18	2.96403E+00	1.42173E+01	2.50076E-01	2.65968E-01	7.80924E+01	6.26255E-01
19	2.94814E+00	1.42710E+01	2.51413E-01	2.64680E-01	7.80854E+01	6.26054E-01
20	2.93861E+00	1.42191E+01	2.49925E-01	2.64561E-01	7.79444E+01	6.22776E-01
21	2.93773E+00	1.31761E+01	2.25664E-01	2.72976E-01	7.61639E+01	5.92280E-01

表 J - 1 3 第 4 抽出器の計算結果(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	1.06963E+00	1.94229E+00	2.02643E-03	2.17343E-01	1.59522E+01	6.50951E-03
2	1.11880E+00	5.52022E+00	2.27881E-03	1.85409E-01	3.61370E+01	5.86710E-03
3	1.10989E+00	9.95818E+00	2.22915E-03	1.53756E-01	5.09873E+01	4.49808E-03
4	1.10111E+00	1.31816E+01	1.85354E-03	1.37727E-01	5.85890E+01	3.25127E-03
5	1.14105E+00	2.81844E+01	5.37177E-02	1.00594E-01	8.09370E+01	6.16354E-02
6	5.56065E-01	5.30926E+01	3.52587E-04	4.15352E-02	8.50064E+01	2.18907E-04
7	3.32764E-01	6.73682E+01	6.56028E-07	2.30363E-02	8.70105E+01	3.26365E-07
8	2.39440E-01	7.42754E+01	4.70683E-09	1.61227E-02	8.70769E+01	2.11813E-09
9	2.00052E-01	7.42274E+01	6.76086E-10	1.36745E-02	8.52286E+01	2.97170E-10
10	1.90159E-01	6.71793E+01	2.10557E-10	1.38447E-02	8.05977E+01	9.64051E-11
11	1.90763E-01	4.97123E+01	9.72246E-11	1.62948E-02	6.74144E+01	5.00285E-11

STAGE	水相		有機相	
	PU(III) (G/L)	U(IV) (G/L)	PU(III) (G/L)	U(IV) (G/L)
1	1.67254E+00	7.05129E+00	5.22017E-02	1.60032E+00
2	1.68575E+00	7.49702E+00	3.97455E-02	1.36215E+00
3	1.68280E+00	7.63152E+00	3.04365E-02	1.08722E+00
4	1.68072E+00	7.74928E+00	2.60490E-02	9.60215E-01
5	1.63932E+00	8.22356E+00	1.46636E-02	6.74447E-01
6	2.79726E-01	4.32737E+00	1.85249E-03	1.91909E-01
7	7.60304E-03	3.21847E+00	4.34049E-05	1.16693E-01
8	1.63441E-04	1.16889E+00	8.75817E-07	3.89690E-02
9	3.30964E-06	6.34971E-02	1.78236E-08	2.10313E-03
10	6.78920E-08	3.13443E-03	3.92851E-10	1.10887E-04
11	1.51980E-09	1.55599E-04	1.04037E-11	6.65298E-06

表 J - 1 4 第 5 抽出器の計算結果(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	3.28307E-02	5.47501E+01	0.00000E+00	2.91943E-03	4.30222E+01	0.00000E+00
2	2.21803E-02	3.47635E+01	0.00000E+00	1.90687E-03	1.97997E+01	0.00000E+00
3	2.13621E-02	1.59989E+01	0.00000E+00	1.25819E-03	3.72583E+00	0.00000E+00
4	2.08380E-02	3.01061E+00	0.00000E+00	4.70148E-04	1.04881E-01	0.00000E+00
5	2.02012E-02	8.47476E-02	0.00000E+00	2.37182E-04	8.16623E-04	0.00000E+00
6	2.00130E-02	6.59860E-04	0.00000E+00	2.25335E-04	5.67817E-06	0.00000E+00
7	2.00034E-02	4.58817E-06	0.00000E+00	2.23815E-04	3.78241E-08	0.00000E+00
8	2.00022E-02	3.05620E-08	0.00000E+00	2.22469E-04	2.41191E-10	0.00000E+00
9	2.00011E-02	1.93707E-10	0.00000E+00	2.21150E-04	1.46419E-12	0.00000E+00

表 J - 1 5 第 6 抽出器の計算結果(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	1.43211E+00	1.50898E-02	7.91567E-35	3.07748E-01	1.36745E-01	2.80937E-34
2	1.60004E+00	8.95957E-02	1.54256E-34	3.45744E-01	9.39176E-01	6.40392E-34
3	1.62077E+00	5.26852E-01	2.75133E-34	3.37885E-01	5.34341E+00	1.10703E-33
4	1.61648E+00	2.92950E+00	4.72573E-34	2.85334E-01	2.35271E+01	1.50736E-33
5	1.58779E+00	1.28516E+01	6.61975E-34	1.87935E-01	5.81548E+01	1.19442E-33
6	1.53464E+00	3.17474E+01	4.67777E-34	1.23612E-01	8.24295E+01	4.88140E-34
7	1.49603E+00	3.26878E+01	6.67044E-34	1.19700E-01	8.26942E+01	6.76946E-34
8	1.47786E+00	3.39029E+01	6.40568E-34	1.16546E-01	8.34201E+01	6.31987E-34
9	1.46306E+00	3.72333E+01	5.77758E-34	1.10324E-01	8.57493E+01	5.34100E-34
10	1.43401E+00	4.79717E+01	5.94660E-34	9.50231E-02	9.20708E+01	4.60329E-34
11	4.58316E-01	8.29182E+01	2.20637E-34	2.82574E-02	9.05869E+01	9.29700E-35
12	1.26029E-01	7.54817E+01	1.14276E-36	8.92281E-03	7.55192E+01	4.33462E-37

表 J - 1 6 第 7 抽出器の計算結果(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	2.63082E-02	5.42276E+01	0.00000E+00	2.37307E-03	4.04698E+01	0.00000E+00
2	2.15482E-02	2.90518E+01	0.00000E+00	1.74243E-03	1.30768E+01	0.00000E+00
3	2.10930E-02	9.38731E+00	0.00000E+00	8.80841E-04	1.03317E+00	0.00000E+00
4	2.04745E-02	7.41666E-01	0.00000E+00	2.91370E-04	9.89260E-03	0.00000E+00
5	2.00514E-02	7.10148E-03	0.00000E+00	2.24568E-04	5.75654E-05	0.00000E+00
6	2.00034E-02	4.13225E-05	0.00000E+00	2.21403E-04	3.14533E-07	0.00000E+00
7	2.00011E-02	2.24624E-07	0.00000E+00	2.19809E-04	1.62354E-09	0.00000E+00

表 J-17 第8抽出器の計算結果(ミキサ部, 通常濃度近似)

STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	3.64039E+00	2.93399E-07	1.03596E-04	7.57472E-01	9.25620E-06	1.37326E-03
2	3.80197E+00	2.26793E-06	3.96539E-04	7.81989E-01	7.37828E-05	5.53543E-03
3	3.80720E+00	1.60327E-05	1.28441E-03	7.82691E-01	5.21995E-04	1.79557E-02
4	3.80735E+00	1.11645E-04	3.93389E-03	7.82491E-01	3.63332E-03	5.49713E-02
5	3.80731E+00	7.75352E-04	1.18300E-02	7.81755E-01	2.51930E-02	1.65052E-01
6	3.80715E+00	5.37444E-03	3.53124E-02	7.79101E-01	1.73632E-01	4.89886E-01
7	3.80659E+00	3.70395E-02	1.04606E-01	7.68276E-01	1.16866E+00	1.41742E+00
8	3.80428E+00	2.49298E-01	3.02466E-01	7.21960E-01	7.07492E+00	3.68772E+00
9	3.79440E+00	1.50922E+00	7.86766E-01	5.78446E-01	2.92152E+01	6.55064E+00
10	1.83805E+00	3.33950E+00	2.35524E+00	2.89323E-01	3.02539E+01	7.44629E+00
11	1.27357E+00	5.36737E+00	4.10389E+00	1.90482E-01	3.08285E+01	7.92840E+00
12	1.08059E+00	6.48918E+00	5.04516E+00	1.56848E-01	3.10493E+01	8.04150E+00
13	1.01493E+00	6.92030E+00	5.26596E+00	1.45862E-01	3.10527E+01	7.84711E+00
14	9.93477E-01	6.92693E+00	4.88645E+00	1.43562E-01	3.05926E+01	7.15667E+00
15	9.88986E-01	6.02867E+00	3.53844E+00	1.49203E-01	2.75047E+01	5.34429E+00

表 J - 1 8 第 9 抽出器の計算結果(ミキサ部, 通常濃度近似)

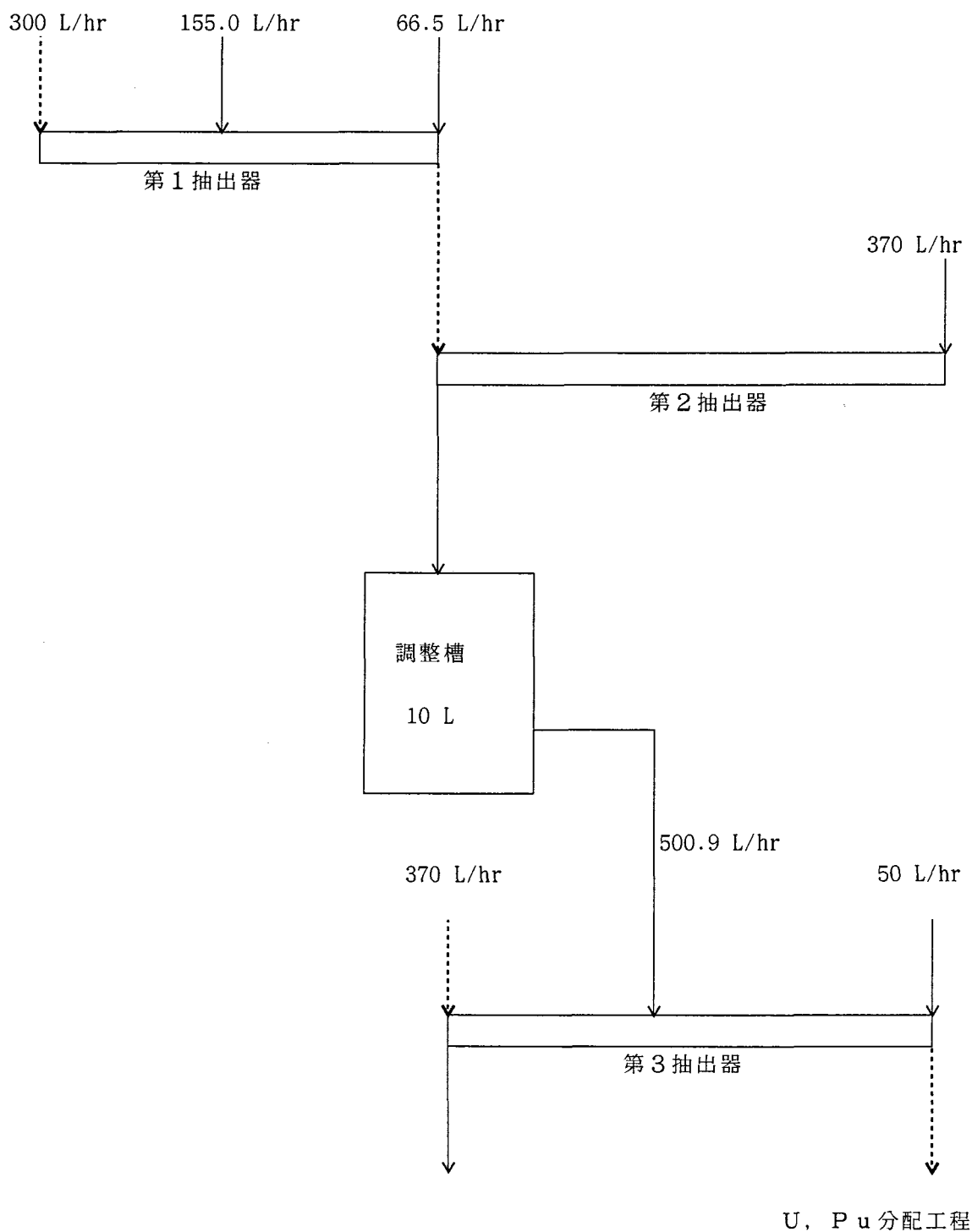
STAGE	水相			有機相		
	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)	HN03 (MOLE)	U(VI) (G/L)	PU(IV) (G/L)
1	7.82396E-01	2.94187E-04	4.64519E-01	1.80559E-01	2.49098E-03	1.30335E+00
2	9.07055E-01	1.96860E-03	8.10325E-01	2.12679E-01	1.94174E-02	2.66853E+00
3	9.31684E-01	1.32435E-02	1.12585E+00	2.18047E-01	1.35028E-01	3.84121E+00
4	9.38360E-01	8.78480E-02	1.28251E+00	2.18329E-01	9.04374E-01	4.42331E+00
5	9.40667E-01	5.39320E-01	1.12252E+00	2.12025E-01	5.37387E+00	3.75022E+00
6	9.30844E-01	3.09016E+00	1.08551E+00	1.75081E-01	2.40921E+01	2.84015E+00
7	9.16601E-01	3.83719E+00	1.54540E-01	1.66545E-01	2.82433E+01	3.81529E-01
8	7.56903E-01	5.74164E+00	3.62231E-03	1.25069E-01	2.92035E+01	6.09186E-03
9	6.00039E-01	7.67538E+00	6.62132E-05	9.18791E-02	2.99029E+01	8.46449E-05
10	4.82378E-01	9.43050E+00	1.18199E-06	6.96170E-02	3.06509E+01	1.25560E-06
11	4.01161E-01	1.12724E+01	2.05223E-08	5.49934E-02	3.24048E+01	1.92466E-08
12	3.05313E-01	1.41635E+01	6.72514E-10	3.86114E-02	3.27595E+01	5.05793E-10
13	2.38389E-01	1.42638E+01	2.57487E-11	2.92305E-02	2.88730E+01	1.69086E-11

STAGE	水相		有機相	
	PU(III) (G/L)	U(IV) (G/L)	PU(III) (G/L)	U(IV) (G/L)
1	1.24215E+01	1.01605E-06	4.09307E-01	3.38346E-07
2	1.32343E+01	1.19125E-05	4.71600E-01	3.96687E-06
3	1.38844E+01	1.89711E-04	5.00060E-01	6.31738E-05
4	1.45403E+01	3.83246E-03	5.22677E-01	1.27621E-03
5	1.51094E+01	7.36208E-02	5.21518E-01	2.45157E-02
6	1.46903E+01	5.61989E-01	3.93211E-01	1.87142E-01
7	9.83084E+00	4.81953E+00	2.49721E-01	1.60490E+00
8	2.13448E+00	3.26067E+00	4.31492E-02	1.08580E+00
9	1.66718E-01	2.82089E+00	2.76793E-03	9.39355E-01
10	9.66073E-03	2.93879E+00	1.37938E-04	9.78617E-01
11	4.71418E-04	3.76351E+00	6.02251E-06	1.25325E+00
12	2.11215E-05	1.80672E+00	2.20216E-07	6.01638E-01
13	7.48505E-07	6.67052E-01	6.83954E-09	2.22128E-01

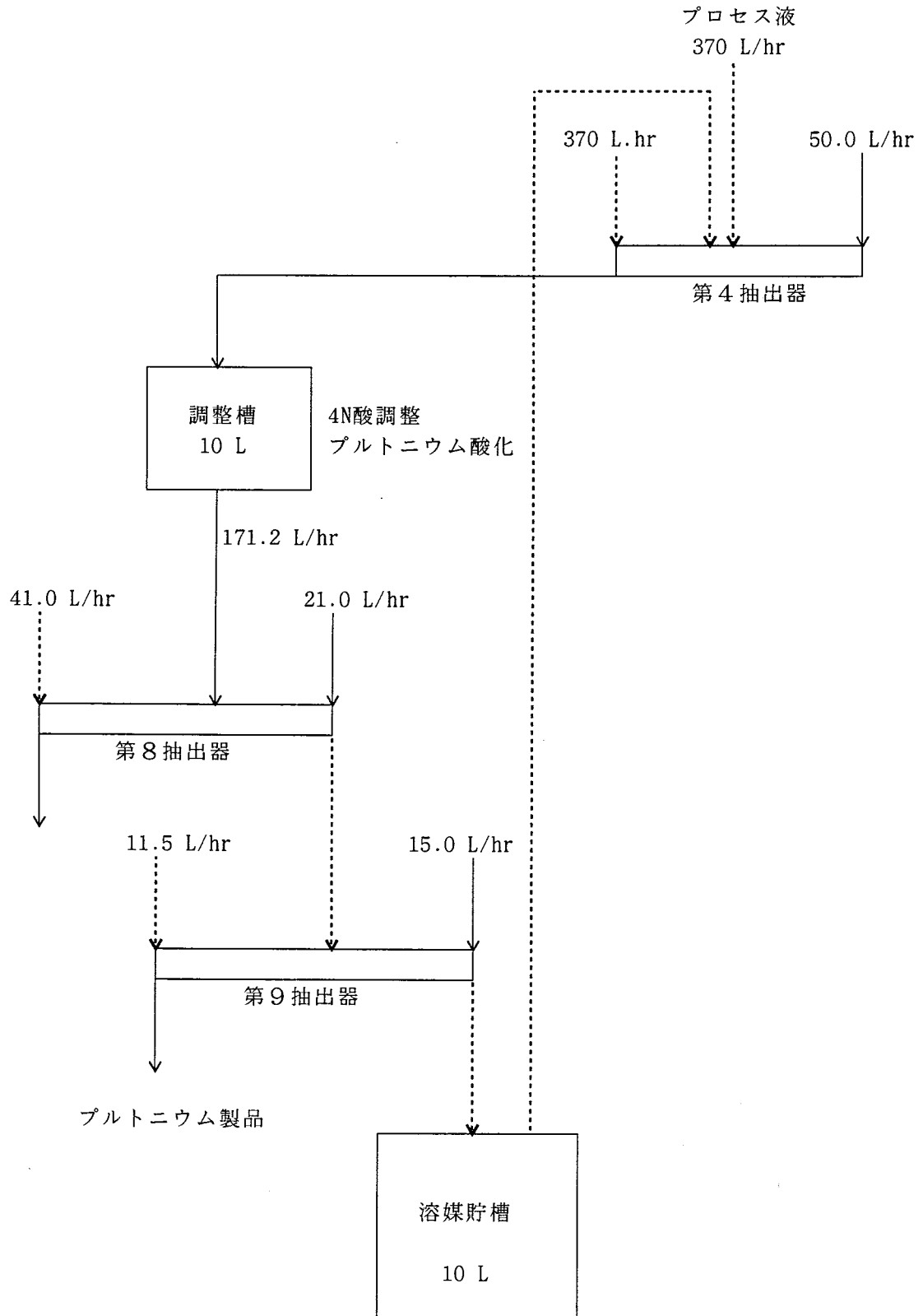
付録K 再処理工場のNp, Tcの分配計算の計算体系図と入力データリスト

(1) 計算体系図

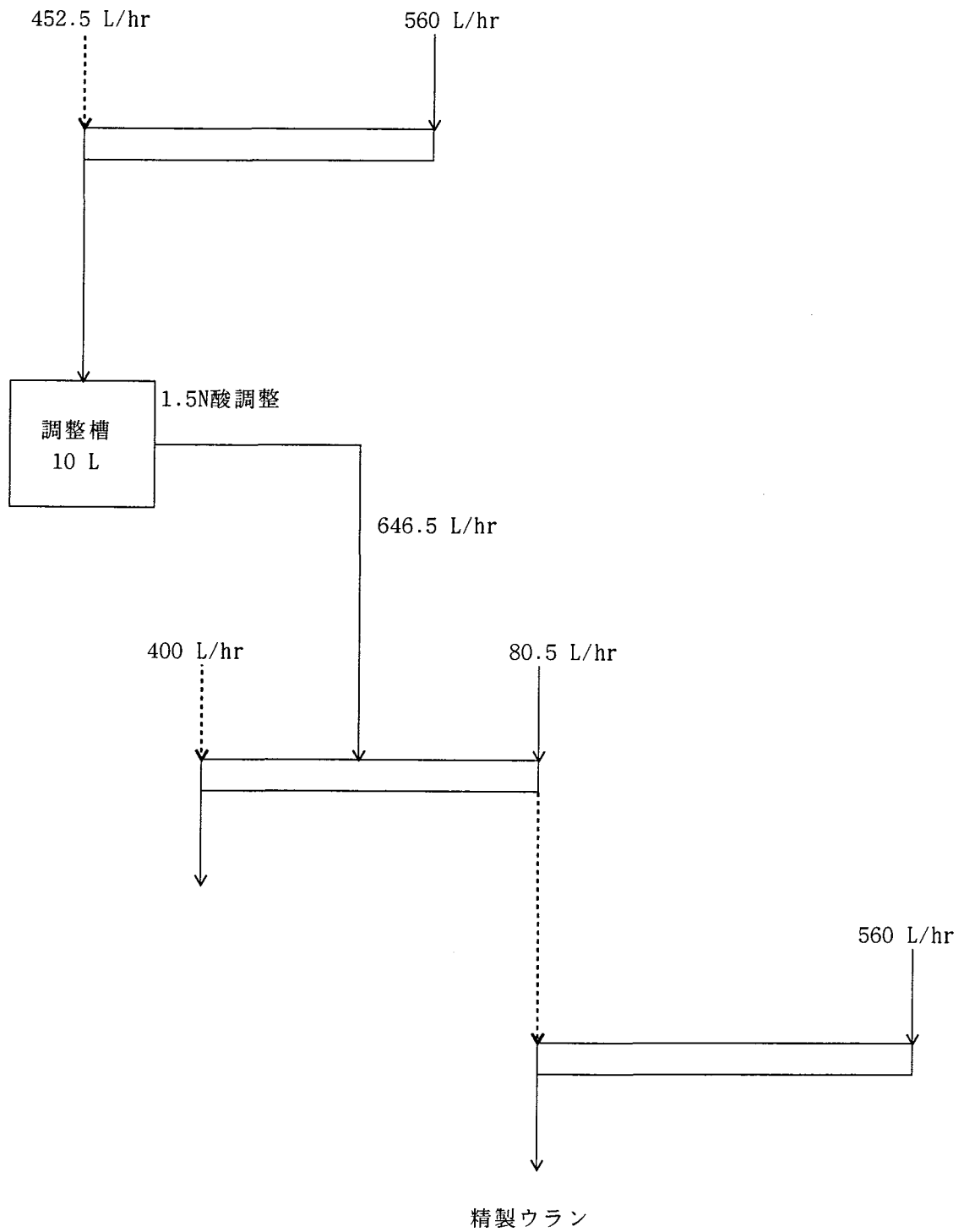
計算体系図を図K-1, 図K-2及び図K-3に示す。



図K-1 Np, Tc 分配計算体系図 (第1~第3抽出器)



図K-2 Np, Tc分配計算体系図(第4, 第8, 第9抽出器)



図K-3 Np, Tc分配計算体系図(第5, 第6, 第7抽出器)

(2) 入力データリスト

(a) 第1抽出器から第3抽出器までの計算の入力データリスト

EXTO1-EXTO2-EXTO3

CONTR 0 1 5000.0 1.00

STAGE 1 17 0 0 2 1 1

VOLUM 1 17*12.0 9*74.1 8*93.8

RECYC 1 9*0.0 8*198.0

*---BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--

FEED 1 -1 1 0.0 300.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

FEED 1 9 1 0.0 155.0 3.0 180.0 1.4 0.0 0.0 0.0 0.0 0.0

!* PU(V), PU(VI) *! 0.0 0.0

!* NP(IV),NP(V),NP(VI) *! 0.0 2.38E-2 0.0

!* ZR(IV) *! 7.24E-2

!* TC(IV,V,VI,VII) *! 0.0 0.0 0.0 1.0E-2

!* TC-U,TC-PU,TC-ZR *! 3*0.0

!* SR(II) *! 3.5E-2

!* RU(DI),RU(TR) *! 0.0 0.0

!* RU(NI),RU(MO) *! 0.0 0.0

!* CS(I),CE(III),GD(III) *! 7.29E-2 6.19E-7 0.0

!* AM(III),CM(III) *! 2.75E-2 7.81E-5

!* ADD-MIC(1-9) *! 9*0.0

!* RU(TOTAL) *! 2.27E-6

FEED 1 17 1 0.0 66.5 3.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 1, 1, 0, 17*3.0

*-----

TOFSTG 1 9*25.0 8*30.0

*-----

STAGE 2 12 301 1 0 0 1

VOLUM 2 12*12.0 12*99.2

RECYC 2 12*0.0

*----BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--

*EED 2 -1 1 0.0 300.0 0.184636 93.9457 0.73058 4*0.0 34*0.0

FEED 2 4 1 0.0 3.2 11.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

FEED 2 12 1 0.0 370.0 0.02 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 2 1 0 12*0.02

*-----

TOFSTG 2 40.0, 41.5, 42.9, 44.4, 45.8, 47.3, 48.7, 50.2, 51.6, 53.1,
54.5, 56.0

*-----

STAGE 3 21 0 0 0 0 1

VOLUM 3 14*21.9 7*18.0 14*147.3 7*107.4

RECYC 3 14*0.0 7*300.0

*---BANK-STAGE-NSET--TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--

FEED 3 -1 1 0.0 370.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*EED 3 14 1 0.0 500.9 3.0 56.26 0.4375 4*0.0 34*0.0

FEED 3 21 1 0.0 50.0 3.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 3 1 0 21*3.0

```
*-----  
TOFSTG 3 21*30.0  
*-----  
TANK 301 0 10.0 3.0 11.0 0.0 500.9 3 14 0.0 0 0  
INITI 301 1, 0, 3.0  
*-----  
DBP 1.0E-6  
EFFI 1 23 1 17*0.2  
EFFI 1 24 1 17*0.3  
FILE 'test1tru'  
END
```


(b) 第4, 第8, 第9抽出器の計算の入力データリスト

EXT4-EXT8-EXT9

CONTR 0 1 5000.0 0.10

STAGE 4 11 801 1 0 0 1

VOLUM 4 4*5.44 7*18.0 4*36.1 7*142.5

RECYC 4 4*0.0 7*300.0

*---BANK-STAGE-NSET--TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--

FEED 4 -1 1 0.0 30.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*EED 4 -4 1 0.0 52.5 0.024 25.0 0.0 0.0 0.24 2*0.0 34*0.0

FEED 4 -5 1 0.0 370.0

2.74277E-01 7.54049E+01 5.76209E-01

1.10454E-05 0.00000E+00 2.75752E-03

0.00000E+00 0.00000E+00

!* PU(V), PU(VI) *! 5.68135E-03 4.32620E-03

!* NP(IV),NP(V),NP(VI) *! 6.70126E-08 3.45150E-06 9.94696E-03

!* ZR(IV) *! 1.92927E-37

!* TC(IV,V,VI,VII) *! 3*0.0 3.25111E-06

!* TC-U,TC-PU,TC-ZR *! 1.95391E-04 2.97406E-05 3.91432E-40

!* SR(II) *! 1.23894E-63

!* RU(DI),RU(TR) *! 1.97299E-20 4.74385E-09

!* RU(NI),RU(MO) *! 2.74690E-08 0.00000E+00

!* CS(I),CE(III),GD(III) *! 4.12576E-54 9.69514E-48 0.00000E+00

!* AM(III),CM(III) *! 8.45594E-42 1.91396E-44

!* ADD-MIC(1-9) *! 9*0.0

!* RU(TOTAL) *! 0.0

FEED 4 5 1 0.0 7.6 1.5 80.0 0.0 0.0 120.0 0.0 6.0 34*0.0

FEED 4 7 1 0.0 2.2 1.5 80.0 0.0 0.0 120.0 0.0 6.0 34*0.0

FEED 4 8 1 0.0 1.07 1.5 80.0 0.0 0.0 120.0 0.0 6.0 34*0.0

FEED 4 10 1 0.0 0.0 1.5 80.0 0.0 0.0 120.0 0.0 6.0 34*0.0

FEED 4 11 1 0.0 120.0 0.2 0.0 0.0 0.0 0.0 0.0 6.0 34*0.0

*-----

INITI 4, 1, 0, 11*0.02

INITI 4, 7, 0, 11*6.0

*-----

TOFSTG 4 11*30.0

*-----

SDIST 4 5*0 2 34*0

CDIST 4 6 11*10.0

*-----

STAGE 8 15 0 0 9 6 1

VOLUM 8 9*5.4 6*3.08 9*30.73 6*12.47

RECYC 8 15*0.0

*---BANK-STAGE-NSET-TIME--FLOW---H+----U02--PU4--PU3--U4---HN02--HYD--

FEED 8 -1 1 0.0 41.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*EED 8 9 1 0.0 171.2 4.0 6.587 1.280 4*0.0 34*0.0

FEED 8 15 1 0.0 21.0 1.0 0.0, 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 8, 1, 0, 15*1.0

*-----

TOFSTG 8 15*25.0

```

*-----
STAGE  9 13  0  0 901 1  1
VOLUM  9 5*1.3  8*3.08  5*5.09  8*12.47
RECYC  9 5*0.0  8*50.0
*---BANK-NSET-STAGE-TIME--FLOW---H+---U02--PU4--PU3--U4---HN02--HYD--
FEED   9 -1 1  0.0  11.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  34*0.0
*BED   9 -6 1  0.0  41.0  0.148  27.50  5.343  4*0.0  34*0.0
FEED   9  7 1  0.0   1.5  1.5  80.0  0.0  0.0 120.0 0.0  6.0  34*0.0
FEED   9  9 1  0.0   0.0  1.5  80.0  0.0  0.0 120.0 0.0  6.0  34*0.0
FEED   9 11 1  0.0   0.5  1.5  80.0  0.0  0.0 120.0 0.0  6.0  34*0.0
FEED   9 13 1  0.0  15.0  0.2  0.0  0.0  0.0  0.0 0.0  6.0  34*0.0
*-----
INITI  9, 1, 0, 13*0.2
INITI  9, 7, 0, 13*6.0
*-----
TOPSTG 9  13*25.0
*-----
SDIST  9  4*0  0  2  34*0
CDIST  9  6 13*10.0
*-----
TANK   801 0 10.0  4.0 11.0 1.0 171.2  8 9 0.0  0 0
INITI  801 1, 0, 4.0
*-----
TANK   901 1 10.0  0.0  0.0 0.0  0.0  4 4 0.0  0 0
INITI  901 1, 0, 0.0
*-----
TAUCONST 1  0,40,1.0
FILE    'test2tru'
END

```

(c) 第5, 第6, 第7抽出器の計算の入力データリスト

EXT05-EXT06-EXT07

CONTR 0 1 5000.0 1.00

*-----

TANK 601 0 10.0 1.5 11.0 0.0 646.5 6 6 0.0 0 0

INITI 601 1, 0, 1.5

*-----

STAGE 5 9 601 1 0 0 1

VOLUM 5 9*21.9 9*157.3

RECYC 5 4*0.0 5*300.0

*---BANK-STAGE-NSET-TIME--FLOW---H+---U02--PU4--PU3--U4---HNO2--HYD--

FEED 5 -1 1 0.0 452.5

1.67592E-02 6.73455E+01 1.90108E-12

7.33676E-12 1.09783E-04 7.36053E-08

0.00000E+00 0.00000E+00

!* PU(V), PU(VI) *! 8.82151E-16 1.41735E-12

!* NP(IV), NP(V), NP(VI) *! 6.01850E-05 9.38947E-10 7.07066E-09

!* ZR(IV) *! 2.84606E-51

!* TC(IV, V, VI, VII) *! 3*0.0 8.48753E-07

!* TC-U, TC-PU, TC-ZR *! 9.06285E-05 5.60977E-17 2.99888E-54

!* SR(II) *! 8.50144E-87

!* RU(DI), RU(TR) *! 1.10561E-25 3.87833E-09

!* RU(NI), RU(MO) *! 1.99344E-08 0.00000E+00

!* CS(I), CE(III), GD(III) *! 3.22508E-75 1.16577E-62 0.00000E+00

!* AM(III), CM(III) *! 7.55877E-59 9.37322E-63

!* ADD-MIC(1-9) *! 9*0.0

!* RU(TOTAL) *! 0.0

FEED 5 9 1 0.0 560.0 0.02 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 5, 1, 0, 9*0.02

*-----

TOFSTG 5 45.0, 46.9, 48.8, 50.6, 52.5, 54.4, 56.2, 58.1, 60.0

*-----

SDIST 5 5*0 2 34*0

CDIST 5 6 9*10.0

*-----

STAGE 6 12 0 0 7 1 1

VOLUM 6 6*27.0 6*18.0 6*174.5 6*122.0

RECYC 6 6*0.0 6*300.0

*---BANK-STAGE-NSET-TIME--FLOW---H+---U02--PU4--PU3--U4---HNO2--HYD--

FEED 6 -1 1 0.0 400.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*EED 6 6 1 0.0 646.5 1.5 46.64 0.0 0.0 0.0 0.0 0.0 34*0.0

FEED 6 10 1 0.0 6.0 13.5 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

FEED 6 12 1 0.0 80.5 0.029 0.4969 0.0 0.0 0.7453 0.0 0.03727 34*0.0

*-----

INITI 6, 1, 0, 12*1.5

INITI 6, 7, 0, 12*0.03727

*-----

TOFSTG 6 12*30.0

*-----

SDIST 6 5*0 2 34*0

CDIST 6 6 12*10.0

*-----

STAGE 7 7 0 0 0 0 1

VOLUM 7 7*18.0 7*134.0

RECYC 7 7*0.0

*---BANK-STAGE-NSET-TIME--FLOW---H+-----U02--PU4--PU3--U4---HNO2--HYD--

*EED 7 -1 1 0.0 402.0 0.009 75.54 0.0 0.0 0.014 0.0 0.0 34*0.0

FEED 7 7 1 0.0 560.0 0.02 0.0 0.0 0.0 0.0 0.0 0.0 34*0.0

*-----

INITI 7, 1, 0, 7*1.02

*-----

TOFSTG 7 48.0, 50.3, 52.7, 55.0, 57.3, 59.7, 62.0

SDIST 7 40*0

*-----

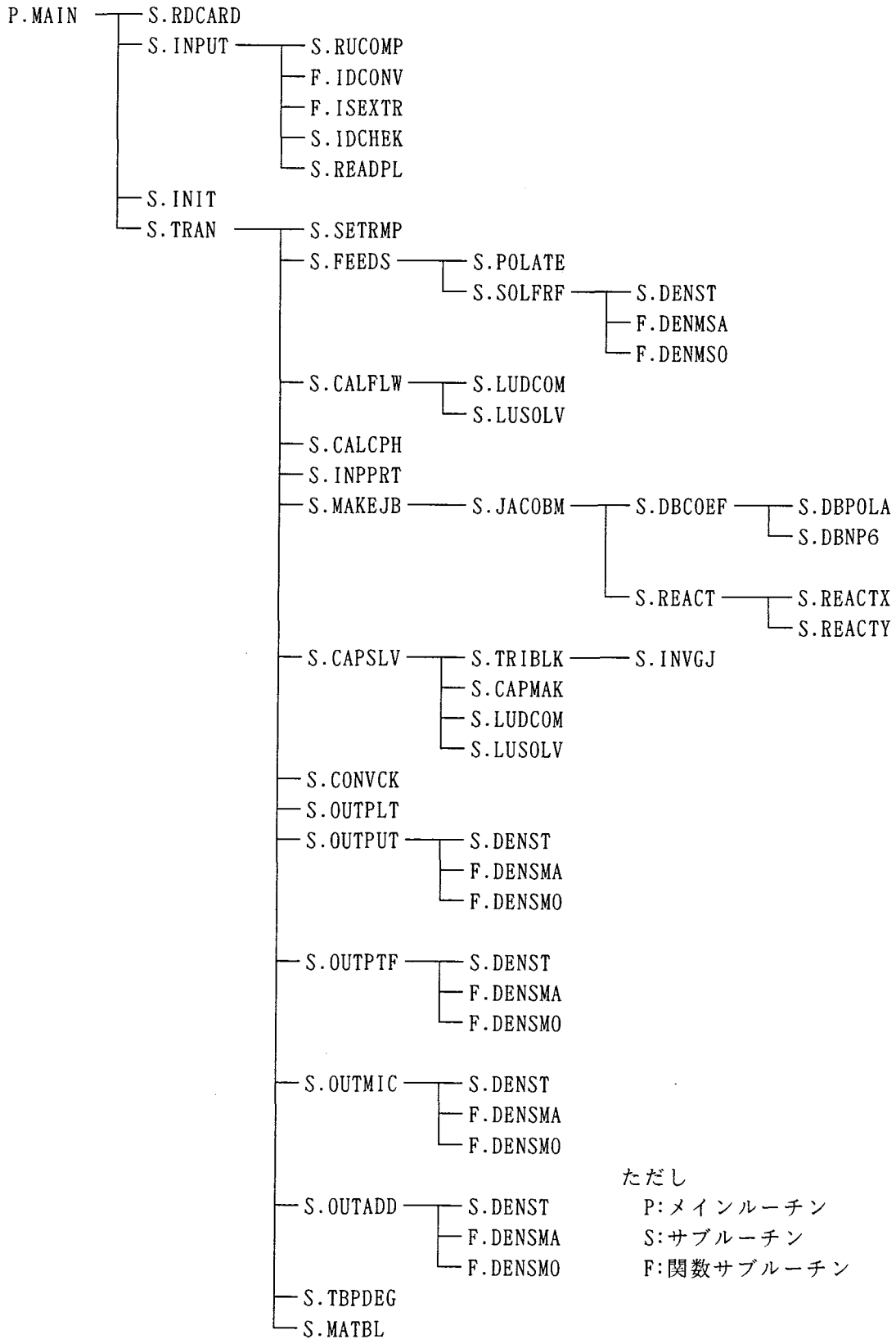
FILE 'test3tru'

END

付録L MIXSET-Xのプログラム構造

MIXSET-Xコードを構成するサブルーチンの簡単な説明とプログラムツリー構造を示す。

(MAIN)	MIXSET-Xのメインルーチン。
CALCPH	ミキサ部の水相体積割合を計算する。
CALFLW	ミキサセトラ内部の流量を計算する。
CAPMAK	容量マトリックス法の前処理を行う。
CAPSLV	容量マトリックス法の行列計算を行う。3重ブロック対角行列計算を含む。
CONST	成分添え字に使用する索引定数を設定する。
CONVCK	ニュートン法の収束判定を行う。
DBCDEF	分配係数とそのヤコビアンを計算する。
DBNP6	KfK-4667のデータに基づく N_p (VI)の分配係数を計算する。
DBPOLA	分配係数の計算に必要な表データ内挿計算を行う。
DENFC	濃度変換ルーチンの初期化を行う。
DENSMA	水相濃度を中性基準濃度から通常濃度に変換するための換算係数を計算する。
DENSMO	有機相濃度を中性基準濃度から通常濃度に変換するための換算係数を計算する。
DENMSA	水相濃度を通常濃度から中性基準濃度に変換するための換算係数を計算する。
DENMSO	有機相濃度を通常濃度から中性基準濃度に変換するための換算係数を計算する。
DENST	濃度変換ルーチンの初期化を行う。
FEEDS	フィードテーブルを内挿してフィード流量と濃度を計算する。
IDCHEK	ミキサセトラまたは貯槽のIDが正しいかどうかを検査する。
IDCONV	ミキサセトラまたは貯槽のIDをプログラム内で使用する機器の順序番号に変ずる。
INIT	過渡計算の初期設定を行う。
INPPRT	入力データを出力する。
INPUT	入力データを読み込む。
INVGJ	行列の逆行列を計算する。
ISEXTR	機器がミキサセトラかどうかを検査する。
JACOBM	ミキサセトラの基本方程式のヤコビアンを計算する。
LUDCOM	行列のLU分解を行う。(フルマトリックス用)
LUSOLV	LUDCOMでLU分解された行列を使用して連立一次方程式の解を計算する。
MAKEJB	計算体系のヤコビアンを計算する。
MATBL	物質収支を計算する。
OUTADD	入力で追加したマイクロ成分の計算結果を出力する。
OUTMIC	プログラム内蔵のマイクロ成分の計算結果を出力する。
OUTPLT	全成分の濃度の計算結果を出力する。
OUTPTF	N_p , T_c の計算結果を出力する。
OUTPUT	主成分8成分の計算結果を出力する。
POLATE	フィードテーブル内挿用の内挿ルーチン。
RDCARD	入力データの前処理および印刷を行う。
REACT	化学反応速度を計算する。REACTX, REACTYを呼び出す。
REACTX	水相の化学反応速度とヤコビアンを計算する。
REACTY	有機相の化学反応速度とヤコビアンを計算する。
READPL	INITIAL 0 カードの指定により初期濃度を読み込む。
RUCOMP	ルテニウム錯体の成分比を計算する。
SETRMP	容量マトリックス法のバンド部分以外の成分の位置を設定する。
SOLFRF	フィード流量と濃度を中性基準濃度に変換する。
TBPDEG	溶媒劣化計算を行い出力する。
TRAN	ミキサセトラの過渡計算を行う。
TRIBLK	3重ブロック対角行列の計算を行う。



図L-1 MIXSET-Xのプログラムツリー構造

付録M MIXSET-Xのプログラムの変数説明

ソースプログラムの可読性を考慮し、MIXSET-Xでは同一の変数名はすべてのサブルーチンでできる限りで同じ意味を持つように命名した。ループ変数についても同様の方針で命名した。例えば、成分でループする場合はループ変数にJを使用し、ミキサセトラの段でループする場合は変数名にIを使用している。例外は、汎用的に使用できる行列計算用サブルーチン INVGJ, LUDCOM, LUSOLV, CAPSLV, CAPMAK および TRIBLK で使用した変数名であるが、この場合に付いても、引数の意味は容易に理解できる様に各サブルーチンのソースプログラムの始めの行に引数のコメントを入れある。

MIXSET-Xプログラムの定数名と主な変数名の簡単な説明を以下に示す。

(1) 定数名

括弧の中はプログラムで定義されている値である。配列の大きさを定義している。

MXCOMP (= 40)	マクロ成分とマイクロ成分を合わせた成分の総数。
NCHEM (= 21)	マクロ成分の数。
MXREAC (= 45)	各相あたりの化学反応の種類数の総数。
MXMCMP (= MXCOMP - 18)	物質収支を計算する成分の総数。
MXEXTR (= 9)	計算可能なミキサセトラの最大数
MXSTG (= 25)	バンクあたりのステージの最大数。
MXTANK (= 10)	計算可能な貯槽の数。
MXFEED (= 30)	フィードの数の最大値。
MXTABL (= 20)	フィードテーブルの時間区分の最大数。
MXSTG2 (= 2*MXSTG+1)	1バンクあたりに使用する配列IPOSの大きさ
MXDEST (= 10)	流出先の最大数
MXUNIT (= MXEXTR+MXTANK)	ミキサセトラと貯槽を含めた機器の最大数
MXN (= 2*NCHEM)	3重ブロック対角行列の小行列の次元。
MXBLK (= 300)	3重ブロック対角行列のブロックの最大数。
MXSYS (= 11)	システム行列の次元の最大値。
MXRM (= 20)	3重ブロック対角からはずれる小行列の最大数。

(2) 共通変数

(a) ラベルつき共通変数領域 IXCOMP

以下の形式で定義される。成分を示す添え字に使用する変数名である。変数の形式をとっているがブロックデータで定義され変更されることはない。特定の成分について特別の処理を行う場合に、添え字に直接成分番号を記述する代わりにこの変数を使用して成分を指定する。

```
COMMON /IXCOMP/NCHEM,
*          IOHN03, IOU6 , IOPU4 , IOPU3 , IOU4 ,
*          IOHN02, IOHYD , IOHAN , IOPU5 , IOPU6 ,
*          IONP4 , IONP5 , IONP6 , IOZR , IOTC4 ,
*          IOTC5 , IOTC6 , IOTC7 , IOTCU6, IOTCP4,
*          IOTCZR, IOSR , IORUDI, IORUTR, IORUNI,
*          IORUMO, IOCS , IOCE , IOGD , IOAM ,
*          IOCM
```

NCHEM 硝酸根濃度の計算に使用する成分の数。

IOHN03	HNO_3	IONP4	$\text{Np}(\text{N})$
IOU6	$\text{UO}_2(\text{NO}_3)_2$	IONP5	$\text{Np}(\text{V})$

IOPU4	$\text{Pu}(\text{NO}_3)_4$	IONP6	$\text{Np}(\text{VI})$
IOPU3	$\text{Pu}(\text{NO}_3)_3$	IOZR	$\text{Zr}(\text{IV})$
IOU4	$\text{U}(\text{NO}_3)_4$	IOTC4	$\text{Tc}(\text{IV})$
IOHN02	HNO_2	IOTC5	$\text{Tc}(\text{V})$
IOHYD	$\text{N}_2\text{H}_5\text{NO}_3$	IOTC6	$\text{Tc}(\text{VI})$
IOHAN	NH_3OHNO_3	IOTC7	$\text{Tc}(\text{VII})$
IOPU5	$\text{Pu}(\text{V})$	IOTCU6	$\text{Tc-U}(\text{VI})$
IOPU6	$\text{Pu}(\text{VI})$	IOTCP4	$\text{Tc-Pu}(\text{IV})$
		IOTCZR	$\text{Tc-Zr}(\text{IV})$
IOSR	$\text{Sr}(\text{II})$	IOCS	$\text{Cs}(\text{I})$
IORUDI	$\text{Ru}(\text{DI})$	IOCE	$\text{Ce}(\text{III})$
IORUTR	$\text{Ru}(\text{TR})$	IOGD	$\text{Gd}(\text{III})$
IORUNI	$\text{Ru}(\text{NI})$	IOAM	$\text{Am}(\text{III})$
IORUMO	$\text{Ru}(\text{MO})$	IOCM	$\text{Cm}(\text{III})$

(b) ラベルつき共通変数領域 IICOMP

以下の形式で定義される。成分を示す添え字に使用する変数名である。変数の形式をとっているがブロックデータで定義され変更されることはない。物質収支の計算のときに使用される。成分は価数を統合したものである。特定の成分について特別の処理を行う場合に、添え字に直接成分番号を記述する代わりにこの変数を使用して成分を指定する。

```
COMMON /IICOMP/I1HNO3,I1U ,I1PU ,I1NP ,I1ZR ,
*          I1TC ,I1SR ,I1RU ,I1CS ,I1CE ,
*          I1GD ,I1AM ,I1CM
```

I1HNO3	硝酸	I1RU	ルテニウム
I1U	ウラン	I1CS	セシウム
I1PU	プルトニウム	I1CE	セリウム
I1NP	ネプツニウム	I1GD	ガドリニウム
I1ZR	ジルコニウム	I1AM	アメリシウム
I1TC	テクネチウム	I1CM	キュリウム
I1SR	ストロンチウム		

(c) ラベルつき共通変数領域 IDISOP

以下の形式で定義される。分配係数のオプションの値を比較するのに使用される。変数の形式をとっているがメインルーチンで定義され変更されることはない。分配係数計算サブルーチンDBCDEFで参照される。

```
COMMON /IDISOP/IDCONS, IDEQUI, IDHANF, IDKFK, IDAMIX
```

IDCONS	定数の分配係数を使用する。
IDEQUI	定数の平衡定数を使用する。
IDHANF	プログラム内蔵の分配係数計算式を使用する。
IDKFK	プログラム内蔵の分配係数計算式を使用する。
IDAMIX	プログラム内蔵の分配係数計算式を使用する。

(3) 主要変数名

(a) 総合的なもの

TITLE 計算ケースのタイトル
 PROGNM プログラムの名前[MIXSET]
 CTFILE 計算結果出力ファイルの名前
 CO TBPの濃度を体積割合からモル濃度へ変換する変換係数(= 3.653774)
 TO 0°Cの絶対温度(= 273.15)
 RCON ガス定数(= 8.3144)
 ISYSIN 入力データのユニット番号(= 5)
 IN 作業用のユニット番号(= 3)
 IRST 計算結果出力用のユニット番号(= 20)
 IUPRT 計算の途中経過を出力するユニット番号(= 6)
 TFEED フィード液の温度(= 25.000 + TO)
 ICALC 計算オプション
 NBLK 水相濃度と有機相濃度の組で1ブロックにしたときの成分濃度配列 XN の最大使用ブロック数。
 NUNIT 計算する抽出器と貯槽の和(機器の数)
 NCOMP 計算対象の成分濃度数。マクロ成分計算中はマクロ成分数(=NCHEM)となり、マイクロ成分の計算中は1である。
 MICCMP 計算する追加マイクロの数
 ISFIN 中性基準濃度の計算オプション
 ENDTIM 計算の終了時刻[hr]
 DTO 計算ステップ幅の初期値[hr]
 TMCONS 初期の計算ステップ幅の制限を終了する時刻[hr]
 DTCONS 初期の計算ステップ幅の最大値[hr]
 DTMIN 計算ステップ幅の最小値[hr]
 DTMAX 計算ステップ幅の最大値[hr]
 DTPRT 計算結果印刷間隔[hr]
 DTPLT 計算結果プロット出力間隔[hr]
 FRC TBPの体積割合
 CDBP DBPの濃度 [g/L]
 CTBP TBPのモル濃度[mol/L]
 ITMAX ニュートン法の繰り返し計算の最大反復数
 EPS ニュートン法の目標収束判定値
 TRATE 計算ステップ幅逡増の増加割合
 ITMIN ニュートン法の計算の収束の速さを決める判定値
 EPSLIM ニュートン法の収束判定値限界。これより大きい誤差はエラーとなる。
 RXLIM 濃度の収束を判定するときの限界値。これより小さい濃度値の場合絶対誤差判定となる。
 FDEG 有機溶媒中のTBPの劣化の度合いを調整するための調節因子
 FMTIN 物質収支計算で使用される物質流入量[g/L]
 FMTOUT 物質収支計算で使用される物質流出量[g/L]
 RMTBL 物質流出量を物質流入量で割ったもの。

(b) 成分の基本情報

CNAME 成分の名前
 FMASS 成分の原子量
 DECAYH 成分の崩壊熱[w/g]
 CVUNIT 入力の成分濃度をモル濃度に変換するときの変換係数。
 CUNIT 入出力に使用する濃度の単位を示す文字

CHRG 成分イオンの電荷
CNTBP 成分の T B P 配位数

(c) フィードデータの情報

FF フィード流量現在値
NFFTAB フィード流量表のデータ個数入力値
FFTAB フィード流量入力値
XF フィード濃度現在値
NXFTAB フィード濃度表のデータ個数入力値
XFTAB フィード濃度入力値
IFDSTA フィード先(水相の場合)
IFDSTO フィード先(有機相の場合)

(d) 抽出器または貯槽の機器毎の情報

ID 機器の I D 番号
IREACT 機器の化学反応の種類を示す符号。0のとき R. M I X S E T 互換。
FCHEM 化学反応の強度を調整する調整因子
ECHEM 化学反応の温度依存性の計算に使用する変数。
TYPE 機器の種類を示す文字。MSならばミキサセトラ。TAならば水相貯槽。
T0ならば有機相貯槽。
MSTG 機器がミキサセトラの場合そのステージ数
IUDSTA 機器の水相流出液の行き先。
IUDSTO 機器の有機相流出液の行き先。
MDEST 機器の流出液の行き先の数。
MEXTRA 機器がミキサセトラの場合ミキサセトラのプログラムないでの順序番号。
TAV 平均温度(予約)
VTOT 機器の液量[L]
ACID 酸調整の目標酸濃度[mol/L]
ACIDIN 調整用酸の濃度[mol/L]
OXID 酸化率
FIN 総流入流量[L/hr]
FOUT 総流出流量[L/hr]
RFOUT 流出流量の行き先毎の割合。
FMSIN 物質流入量[g/hr]
FMSOUT 物質流出量[g/hr]
NSTG 機器がミキサセトラの場合そのステージ数(単純変数)

(e) 抽出器の情報

IDIST 分配係数計算オプション
IEFF 段効率計算オプション
CDIS 定数の分配係数
CKEQU 定数の平衡定数
EFF 段効率
DIS 分配係数
TSTG ステージの温度[K]
V ステージミキサ部の体積[L]
VS ステージセトラ部の体積[L]
HEIGHT ステージセトラ部の水相高さ[-]
FL ステージより他のステージへ流出する流量[L/hr]
FR ステージの内部循環流量[L/hr]

PHIOLD	ステージミキサ部の前のステップの水相液量割合
PHI	ステージミキサ部の現在のステップの水相液量割合
PHIS	ステージセトラ部の前の水相液量割合
WST	濃度変換に使用する変数。
WLT	濃度変換に使用する変数。
HOS	濃度変換に使用する変数。
TF	フリー T B P 濃度[mol/L]
FLS	流出液量総量[L/hr]
FLSMT	流出液量総量の計算に使用される行列用の配列。
IPFLS	流出液量総量の計算に使用される配列。
IPOS	濃度値を保存している配列 XN 内の位置を示すアドレス。

(f) 成分濃度の変数

XN	計算中の成分濃度を保存する基本変数[mol/L]
X	前の計算ステップの成分濃度を保存する基本変数[mol/L]
DX	ニュートン法での濃度値の変化分[mol/L]

どの配列変数も最初の次元の添え字が 1 の場合が水相, 2 の場合有機相である。

(g) ニュートン法で使用する物理量とその微分値

REAC	化学反応による成分の増加分計算値
DRDX	REAC の微分値
DDDX	分配係数の成分による微分値
FKEQ1	T B P 配位数 1 の平衡定数の計算値
FKEQ2	T B P 配位数 2 の平衡定数の計算値
FKEQ3	T B P 配位数 3 の平衡定数の計算値
FKEQC	定数形式の平衡定数の値
DK1	T B P 配位数 1 の平衡定数の硝酸根濃度による微分値
DK2	T B P 配位数 2 の平衡定数の硝酸根濃度による微分値
DK3	T B P 配位数 3 の平衡定数の硝酸根濃度による微分値
DKC	定数形式の平衡定数の硝酸根濃度による微分値
DK1DX	T B P 配位数 1 の平衡定数の成分濃度による微分値
DK2DX	T B P 配位数 2 の平衡定数の成分濃度による微分値
DK3DX	T B P 配位数 3 の平衡定数の成分濃度による微分値
DKCDX	定数形式の平衡定数の成分濃度による微分値
DADX	フリー T B P の計算に使用する 2 次多項式の 2 次の係数の微分値
DBDX	フリー T B P の計算に使用する 2 次多項式の 1 次の係数の微分値
DTFDX	フリー T B P の成分濃度での微分値
DISTC	定数形式の分配係数
DDCDX	定数形式の分配係数の濃度での微分値
DDCDTF	定数形式の分配係数のフリー T B P での微分値

(h) ニュートン法で使用する行列計算用配列

FX	ニュートン法の関数値 (物質収支式または分配平衡式より計算される。)
DFDXD	FX を成分濃度で微分した微分値。
DFDXL	FX を有機相基準で一つ上流の成分濃度で微分した微分値。
DFDXU	FX を水相基準で一つ上流の成分濃度で微分した微分値。
IP	小行列の L U 分解に使用される配列。
WK	小行列の L U 分解に使用される配列。
RM	フィードバックがある時に現れる小行列。微分値が保存されている。
IROWRM	RM の行位置。

ICOLRM	RMの桁位置。
IPOSRM	トータルマトリックス上でRMが存在しているかどうかを示す符号。存在している場合はRMの参照順序番号
EM	容量マトリックス法に現れる行列E
IROWEM	容量マトリックス法のシステム行列を構成するときに使用されるEの参照位置。
SYSM	容量マトリックス法のシステム行列。
Y	容量マトリックス法のシステムベクトル。
IPSYS	容量マトリックス法のシステム行列をLU分解するときに使用される配列。

(4) ループ変数および配列添え字用変数

IUNT	貯槽とミキサセトラを通算した機器順序番号。機器でループする場合のループ変数。
IEXTRA	抽出器の順序番号。データで定義された抽出器の順序に従う。
I	ミキサセトラの段または成分濃度配列内の位置を示すループ変数。
IND	ミキサセトラのミキサ部の成分濃度の配列 XN 上での位置を示す。
INDSA	ミキサセトラのセトラ部の水相成分濃度の配列 XN 上での位置を示す。
INDSO	ミキサセトラのセトラ部の有機相成分濃度の配列 XN 上での位置を示す。
JBASE	計算中の成分濃度の成分番号のオフセット。マクロ成分計算中は0であり、ミクロ成分の計算中は計算対象の成分番号より1を引いた数である。
J	成分濃度でループする場合に使用されるループ変数。多くの場合で成分番号そのものであるが、JACOBM等のヤコビアン計算ルーチンではJJ(=J+JBASE)が成分番号になる。
JJ	成分番号。(=J+JBASE)
K	成分番号ヤコビアン計算の微分変数となる成分の成分番号。

※ 上記以外の変数は、作業用に使用される単純変数であり、概ね次のように名前が付けられている。

F····	流量またはニュートン法の対象になる方程式の関数値。
V····	液量。
CK···	化学反応速度式の数値定数。
EK···	化学反応速度の温度係数。
RR	単位時間当たりの化学反応量。
DX···	水相濃度による微分値。
DY···	有機相濃度による微分値。
Z····	化学反応速度を計算するときの成分濃度下限値(必要な成分について)。
D····	第2文字がXでもYでもない場合には分配係数。
X····	水相成分濃度。
Y····	有機相成分濃度。
TIM··	時間。
····A	水相に関する変数。
····0	有機相に関する変数。

付録N MIXSET-Xソースプログラムリスト

(main)	付- 78
CALCPH	付- 83
CALFLW	付- 83
CAPMAK	付- 85
CAPSLV	付- 86
CONST(BLOCK DATA)	付- 88
CONVCK	付- 88
DBCDEF	付- 89
DBNP6	付- 99
DBPOLA	付-101
DENFC	付-102
DENST	付-102
DENMSA	付-103
DENMSO	付-103
DENSMA	付-104
DENSMO	付-104
FEEDS	付-105
IDCHEK	付-105
IDCONV	付-106
INIT	付-107
INPPRT	付-107
INPUT	付-115
INVGJ	付-124
ISEXTR	付-125
JACOBM	付-126
LUDCOM	付-129
LUSOLV	付-130
MAKEJB	付-130
MATBL	付-135
OUTADD	付-147
OUTMIC	付-150
OUTPLT	付-153
OUTPTF	付-155
OUTPUT	付-158
POLATE	付-161
RDCARD	付-161
REACT	付-165
REACTX	付-166
REACTY	付-178
READPL	付-180
RUCOMP	付-182
SETRMP	付-183
SOLFRF	付-184
TBPDEG	付-185
TRAN	付-186
TRIBLK	付-191

```

C(main)
      IMPLICIT REAL*8 ( A-H , 0-Z )
C
C
C      PARAMETER ( MXCOMP = 40 )
C      PARAMETER ( NCHEM = 21 )
C      PARAMETER ( MXREAC = 45 )
C      PARAMETER ( MXMCOMP = MXCOMP - 18 )
C
C      PARAMETER ( MXEXTR = 9 , MXSTG = 25 , MXTANK = 10 )
C      PARAMETER ( MXFEED = 30 , MXTABL = 20 , MXSTG2 = 2*MXSTG+1 )
C      PARAMETER ( MXDEST = 10 , MXUNIT = MXEXTR+MXTANK )
C
C      PARAMETER ( MXN = 2*NCHEM , MXBLK = 300 , MXSYS=11 , MXRM = 20 )
C
C      PARAMETER ( MXTAUC = 10 )
C
C
C      CHARACTER*72 TITLE
C      CHARACTER*8  PROGNM
C
C      CHARACTER*30 CTFILE
C
C      INDICES FOR CHEMICAL COMPONENTS
C
C      COMMON /IXCOMP/NCHEM,
C      *      IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
C      *      IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
C      *      IONP4 , IONP5 , IONP6 , IOZR , IOZC4 ,
C      *      IOZC5 , IOZC6 , IOZC7 , IOZC6 , IOZCP4 ,
C      *      IOZCZR, IOSR , IOFUDI, IOFUTR, IOFUNI,
C      *      IOFUMO, IOCS , IOCE , IOGD , IOAM ,
C      *      IOCM
C
C      COMMON /IICOMP/I1HNO3, I1U , I1PU , I1NP , I1ZR ,
C      *      I1TC , I1SR , I1FU , I1CS , I1CE ,
C      *      I1GD , I1AM , I1CM
C
C      INDICES FOR DISTRIBUTION CALCULATION OPTION
C
C      COMMON /IDISOP/IDCONS, IDEQUI, IDHANF, IDKFK , IDAMIX
C
C      CHARACTER*8 CNAME (MXCOMP)
C      DIMENSION FMASS (MXCOMP)
C      DIMENSION DECAH (MXCOMP)
C      DIMENSION CVUNIT (MXCOMP)
C      CHARACTER*8 CUNIT (MXCOMP)
C      DIMENSION CHRIG (MXCOMP),
C      *      CNTBP (MXCOMP)
C
C      CHARACTER*8 CPHASE (MXCOMP)
C
C
C      DIMENSION FF ( , MXFEED, 2),
C      *      NFFTAB( , MXFEED, 2),
C      *      FFTAB (2 , MXTABL, MXFEED, 2),
C      *      XF (MXCOMP , MXFEED, 2),
C      *      NXFTAB( , MXFEED, 2),
C      *      XFTAB (MXCOMP+1, MXTABL, MXFEED, 2),
C      *      IFDSTA(2 , MXFEED ),
C      *      IFDSTO(2 , MXFEED )
C
C
C      DIMENSION ID ( , MXUNIT)
C      DIMENSION IREACT( , MXUNIT)
C      DIMENSION FCHEM (MXREAC, 2 , MXUNIT),
C      *      ECHEM (MXREAC, 2 , MXUNIT)
C
C      CHARACTER*2 TYPE ( , MXUNIT)
C
C      DIMENSION MSTG ( , MXUNIT),
C      *      IUDSTA(3 , MXDEST, MXUNIT),
C      *      IUDSTO(3 , MXDEST, MXUNIT)
C      DIMENSION MDEST ( 2 , MXUNIT),
C      *      MEXTRA( , MXUNIT)
C
C      DIMENSION TAV ( , MXUNIT),
C      *      VTOT ( , MXUNIT),
C      *      ACID ( , MXUNIT),
C      *      ACIDIN( , MXUNIT),
C      *      OXID ( , MXUNIT),
C      *      FIN ( 2 , MXUNIT),
C      *      FOUT ( 2 , MXUNIT),
C      *      RFOUT (MXDEST, 2 , MXUNIT),
C      *      FMSIN (MXMCOMP, 2 , MXUNIT),
C
C      *      FMSOUT (MXMCOMP, 2 , MXUNIT)
C
C      DIMENSION FMTIN (MXMCOMP, 2),
C      *      FMTOUT (MXMCOMP, 2),
C      *      FMTBL (MXMCOMP, 3)
C
C      DIMENSION IOEXTR( , MXEXTR)
C      CHARACTER*8 CHFDIS (MXCOMP , MXEXTR)
C
C      DIMENSION IDIST (MXCOMP , MXEXTR)
C      DIMENSION IEFF (MXCOMP , MXEXTR)
C
C      DIMENSION CDIS (MXCOMP, MXSTG , MXEXTR),
C      *      CKEQU (MXCOMP, MXSTG , MXEXTR),
C      *      EFF (MXCOMP, MXSTG , MXEXTR),
C      *      DIS (MXCOMP, MXSTG , MXEXTR)
C
C      DIMENSION TSTG ( , MXSTG , MXEXTR),
C      *      V ( , MXSTG , MXEXTR),
C      *      VS ( , MXSTG , MXEXTR),
C      *      HEIGHT( , MXSTG , MXEXTR),
C      *      FL (2 , MXSTG , MXEXTR),
C      *      FR ( , MXSTG , MXEXTR),
C      *      PHIOLD( , MXSTG , MXEXTR),
C      *      PHI ( , MXSTG , MXEXTR),
C      *      PHIS ( , MXSTG , MXEXTR),
C      *      WST ( , MXSTG , MXEXTR),
C      *      WLT ( , MXSTG , MXEXTR),
C      *      HOS ( , MXSTG , MXEXTR),
C      *      TF ( , MXSTG , MXEXTR)
C
C      DIMENSION FLS ( , MXUNIT, 2),
C      *      FLSMT ( , MXUNIT, MXUNIT, 2),
C      *      IFFLS ( , MXUNIT, 2)
C
C      DIMENSION IPOS (2 , MXSTG2, MXUNIT)
C
C      DIMENSION DTPRT (100)
C      DIMENSION DTPLT (100)
C
C      DIMENSION XN (MXCOMP, 2, MXBLK),
C      *      X (MXCOMP, 2, MXBLK),
C      *      DX (MXCOMP, 2, MXBLK)
C
C      DIMENSION REAC (NCHEM, 2),
C      *      DRDX (NCHEM, NCHEM, 2)
C
C      DIMENSION DDDX (NCHEM, NCHEM)
C
C      DIMENSION FKEQ1 (NCHEM),
C      *      FKEQ2 (NCHEM),
C      *      FKEQ3 (NCHEM),
C      *      FKEQC (NCHEM)
C      DIMENSION DK1 (NCHEM),
C      *      DK2 (NCHEM),
C      *      DK3 (NCHEM),
C      *      DKC (NCHEM)
C      DIMENSION DK1DX (NCHEM, NCHEM),
C      *      DK2DX (NCHEM, NCHEM),
C      *      DK3DX (NCHEM, NCHEM),
C      *      DKCDX (NCHEM, NCHEM)
C      DIMENSION DADX (NCHEM),
C      *      DBDX (NCHEM),
C      *      DTFDX (NCHEM)
C      DIMENSION DISTC (NCHEM),
C      *      DDCDX (NCHEM, NCHEM),
C      *      DDCDTF (NCHEM)
C
C      DIMENSION FX (MXN , MXBLK),
C      *      DFDXD (MXN , MXN , MXBLK),
C      *      DFDXL (MXN , MXN , MXBLK),
C      *      DFDXJ (MXN , MXN , MXBLK),
C      *      IP (MXN , MXBLK),
C      *      WK (MXN )
C      DIMENSION FRM (MXN , MXN , MXRM ),
C      *      IROWRM( , MXRM ),
C      *      ICOLRM( , MXRM )
C      DIMENSION IPOSRM (MXBLK , MXSYS)
C
C      DIMENSION EM (MXN, MXN, MXBLK, MXSYS)
C      DIMENSION IROWEM (MXSYS)
C
C      DIMENSION SYSM (MXN*MXSYS, MXN*MXSYS),
C      *      Y (MXN*MXSYS),
C      *      IPSYS (MXN*MXSYS)
C
C      DIMENSION TMCONE (MXTAUC), TMCONE (MXTAUC), DTCONS (MXTAUC)

```

```

DIMENSION FFOUT(2,2,MXUNIT)
C
C
C
C
C --- ABSOLUTE ZERO TEMPERATURE
C AND FACTOR FROM VOLUME RATIO TO MOL FOR TBP ---
C
C CO = 3.65277400 (VALUE OF MIXSET98 ORIGINAL PROGRAM)
C CO = 0.973*1000.0/266.3 = 3.653773939 (R. MIXSET MANUAL)
C CO = 3.65377400
C TO = 273.1500
C RCON = 8.314400
C
C IDKFK = 0
C IDHANF = 1
C IDCONS = 2
C IDEQUI = 3
C IDAMIX = 4
C
C
C CTFILE = ' '
C
C ISYSIN = 5
C IN = 3
C IFRST = 20
C IUPLT = 30
C IUPRT = 6
C
C
C TFEED = 25.000 + TO
C
C ICALC = 0
C NUNIT = 1
C MICOMP = 0
C ISFIN = 1
C
C ENDTIM = 240.000
C DTO = 0.0100
C
C DO I = 1, MXTAUC
C TMCONB(I) = 0.000
C TMCONC(I) = 0.000
C DTCONS(I) = 1.000
C END DO
C
C DTMIN = 1.0D-4
C DTMAX = 240.000
C
C DO I=1,100
C DTPRT(I) = 0.000
C ENDDO
C
C DO I=1,100
C DTPLT(I) = 0.000
C ENDDO
C
C FRC = 0.300
C CDBP = 0.000
C
C ITMAX = 20
C EPS = 1.0D-4
C TRATE = 1.200
C
C ITMIN = 5
C EPSLIM = 1.0D-3
C RXLIM = 1.0D-13
C
C FDEG = 1.000
C IPRDIS = 0
C
C DO 130 NF = 1, MXFEED
C DO 120 ITBL = 1, MXTABL
C
C FFTAB (1, ITBL, NF, 1) = 0.000
C FFTAB (2, ITBL, NF, 1) = 0.000
C FFTAB (1, ITBL, NF, 2) = 0.000
C FFTAB (2, ITBL, NF, 2) = 0.000
C
C DO 110 J = 1, MXCOMP + 1
C XFTAB (J, ITBL, NF, 1) = 0.000
C XFTAB (J, ITBL, NF, 2) = 0.000
110 CONTINUE
C
120 CONTINUE
130 CONTINUE

```

```

C
C
C DO 150 I = 1, MXBLK
C DO 140 J = 1, MXCOMP
C X(J, 1, I) = 0.000
C X(J, 2, I) = 0.000
140 CONTINUE
150 CONTINUE
C
C
C DO 220 IUNT = 1, MXUNIT
C
C DO 210 L = 1, MXREAC
C FCHEM(L, 1, IUNT) = 1.000
C FCHEM(L, 2, IUNT) = 1.000
210 CONTINUE
C
C IREACT( IUNT) = 0
C MDEST (1, IUNT) = 0
C MDEST (2, IUNT) = 0
C MEXTRA( IUNT) = 0
C
C TAV ( IUNT) = 25.0 + TO
C VTOT ( IUNT) = 0.000
C ACID ( IUNT) = 0.000
C OXID ( IUNT) = 0.000
C FOUT (1, IUNT) = 0.000
C FOUT (2, IUNT) = 0.000
C
C 220 CONTINUE
C
C DO 350 IEXTRA = 1, MXEXTR
C
C DO 310 J = 1, MXCOMP
C IDIST (J, IEXTRA) = IDKFK
C IEFF (J, IEXTRA) = -1
310 CONTINUE
C
C DO 320 ISTG = 1, MXSTG
C TSTG (ISTG, IEXTRA) = 25.0 + TO
C V (ISTG, IEXTRA) = 0.0
C VS (ISTG, IEXTRA) = 0.0
C HEIGHT(ISTG, IEXTRA) = 0.500
C FR (ISTG, IEXTRA) = 0.0
320 CONTINUE
C
C DO 340 ISTG = 1, MXSTG
C DO 330 J = 1, MXCOMP
C CDIS (J, ISTG, IEXTRA) = 0.000
C CKEQU(J, ISTG, IEXTRA) = 0.000
C EFF (J, ISTG, IEXTRA) = 1.000
330 CONTINUE
340 CONTINUE
C
C 350 CONTINUE
C
C
C DO 410 J = 1, MXCOMP
C CNAME (J) = ' '
C FMASS (J) = 0.000
C DECAYH(J) = 0.000
C CVUNIT(J) = 1.000
C CUNIT (J) = ' (MOL/L) '
C CHRIG (J) = 0.000
C CNTEP (J) = 0.000
410 CONTINUE
C
C
C CNAME (10HNO3) = 'HNO3 '
C FMASS (10HNO3) = 0.000
C CVUNIT(10HNO3) = 1.000
C CUNIT (10HNO3) = ' (MOL/L) '
C CHRIG (10HNO3) = 1.000
C CNTEP (10HNO3) = 2.000
C
C CNAME (10U6 ) = 'U(VI) '
C FMASS (10U6 ) = 238.0300
C CVUNIT(10U6 ) = 238.0300
C CUNIT (10U6 ) = ' (G/L) '
C CHRIG (10U6 ) = 2.000
C CNTEP (10U6 ) = 2.000
C
C CNAME (10PU4 ) = 'PU(IV) '

```

```

FMASS (10PU4 ) = 239.0000
CVUNIT(10PU4 ) = 239.0000
CUNIT (10PU4 ) = ' (G/L) '
CHRG (10PU4 ) = 2.000
CNTBP (10PU4 ) = 2.000
C
CNAME (10PU3 ) = 'PU(111) '
FMASS (10PU3 ) = 239.0000
CVUNIT(10PU3 ) = 239.0000
CUNIT (10PU3 ) = ' (G/L) '
CHRG (10PU3 ) = 3.000
CNTBP (10PU3 ) = 2.000
C
CNAME (10U4 ) = 'U(IV) '
FMASS (10U4 ) = 238.0300
CVUNIT(10U4 ) = 238.0300
CUNIT (10U4 ) = ' (G/L) '
CHRG (10U4 ) = 4.000
CNTBP (10U4 ) = 2.000
C
CNAME (10HNO2) = 'HNO2 '
FMASS (10HNO2) = 0.000
CVUNIT(10HNO2) = 1.000
CUNIT (10HNO2) = ' (MOL/L) '
CHRG (10HNO2) = 0.000
CNTBP (10HNO2) = 0.000
C
CNAME (10HYD ) = 'HYD '
FMASS (10HYD ) = 0.000
CVUNIT(10HYD ) = 32.000
CUNIT (10HYD ) = ' (G/L) '
CHRG (10HYD ) = 1.000
CNTBP (10HYD ) = 0.000
C
CNAME (10HAN ) = 'HAN '
FMASS (10HAN ) = 0.000
CVUNIT(10HAN ) = 33.000
CUNIT (10HAN ) = ' (G/L) '
CHRG (10HAN ) = 1.000
CNTBP (10HAN ) = 0.000
C
CNAME (10PU5 ) = 'PU(V) '
FMASS (10PU5 ) = 239.0000
CVUNIT(10PU5 ) = 239.0000
CUNIT (10PU5 ) = ' (G/L) '
CHRG (10PU5 ) = 1.000
CNTBP (10PU5 ) = 1.000
C
CNAME (10PU6 ) = 'PU(V1) '
FMASS (10PU6 ) = 239.0000
CVUNIT(10PU6 ) = 239.0000
CUNIT (10PU6 ) = ' (G/L) '
CHRG (10PU6 ) = 2.000
CNTBP (10PU6 ) = 2.000
C
CNAME (10NP4 ) = 'NP(IV) '
FMASS (10NP4 ) = 237.0000
CVUNIT(10NP4 ) = 237.000
CUNIT (10NP4 ) = ' (G/L) '
CHRG (10NP4 ) = 4.000
CNTBP (10NP4 ) = 2.000
C
CNAME (10NP5 ) = 'NP(V) '
FMASS (10NP5 ) = 237.0000
CVUNIT(10NP5 ) = 237.000
CUNIT (10NP5 ) = ' (G/L) '
CHRG (10NP5 ) = 1.000
CNTBP (10NP5 ) = 1.000
C
CNAME (10NP6 ) = 'NP(V1) '
FMASS (10NP6 ) = 237.0000
CVUNIT(10NP6 ) = 237.000
CUNIT (10NP6 ) = ' (G/L) '
CHRG (10NP6 ) = 2.000
CNTBP (10NP6 ) = 2.000
C
CNAME (10ZR ) = 'ZR(IV) '
FMASS (10ZR ) = 95.000
CVUNIT(10ZR ) = 95.000
CUNIT (10ZR ) = ' (G/L) '
CHRG (10ZR ) = 4.000
CNTBP (10ZR ) = 2.000
C
CNAME (10TC4 ) = 'TC(IV) '
FMASS (10TC4 ) = 99.000
CVUNIT(10TC4 ) = 99.000
CUNIT (10TC4 ) = ' (G/L) '
CHRG (10TC4 ) = 0.000
CNTBP (10TC4 ) = 0.000
C
CNAME (10TC5 ) = 'TC(V) '
FMASS (10TC5 ) = 99.000
CVUNIT(10TC5 ) = 99.000
CUNIT (10TC5 ) = ' (G/L) '
CHRG (10TC5 ) = 0.000
CNTBP (10TC5 ) = 0.000
C
CNAME (10TC6 ) = 'TC(V1) '
FMASS (10TC6 ) = 99.000
CVUNIT(10TC6 ) = 99.000
CUNIT (10TC6 ) = ' (G/L) '
CHRG (10TC6 ) = 0.000
CNTBP (10TC6 ) = 0.000
C
CNAME (10TC7 ) = 'TC(V11) '
FMASS (10TC7 ) = 99.000
CVUNIT(10TC7 ) = 99.000
CUNIT (10TC7 ) = ' (G/L) '
CHRG (10TC7 ) = 0.000
CNTBP (10TC7 ) = 3.000
C
CNAME (10TCU6) = 'TC-U '
FMASS (10TCU6) = FMASS (10TC7 ) + FMASS (10U6 )
CVUNIT(10TCU6) = 99.000
CUNIT (10TCU6) = ' (G/L) '
CHRG (10TCU6) = 0.000
CNTBP (10TCU6) = 2.000
C
CNAME (10TCP4) = 'TC-PU '
FMASS (10TCP4) = FMASS (10TC7 ) + FMASS (10PU4 )
CVUNIT(10TCP4) = 99.000
CUNIT (10TCP4) = ' (G/L) '
CHRG (10TCP4) = 0.000
CNTBP (10TCP4) = 2.000
C
CNAME (10TCZR) = 'TC-ZR '
FMASS (10TCZR) = FMASS (10TC7 ) + FMASS (10ZR )
CVUNIT(10TCZR) = 99.000
CUNIT (10TCZR) = ' (G/L) '
CHRG (10TCZR) = 0.000
CNTBP (10TCZR) = 2.000
C
CNAME (10SR ) = 'SR(11) '
FMASS (10SR ) = 90.000
CVUNIT(10SR ) = 90.000
CUNIT (10SR ) = ' (G/L) '
CHRG (10SR ) = 2.000
CNTBP (10SR ) = 2.000
C
CNAME (10RUDI) = 'RU(D1) '
FMASS (10RUDI) = 106.000
CVUNIT(10RUDI) = 106.000
CUNIT (10RUDI) = ' (G/L) '
CHRG (10RUDI) = 2.000
CNTBP (10RUDI) = 2.000
C
CNAME (10RUJR) = 'RU(JR) '
FMASS (10RUJR) = 106.000
CVUNIT(10RUJR) = 106.000
CUNIT (10RUJR) = ' (G/L) '
CHRG (10RUJR) = 3.000
CNTBP (10RUJR) = 2.000
C
CNAME (10RUNI) = 'RU(N1) '
FMASS (10RUNI) = 106.000
CVUNIT(10RUNI) = 106.000
CUNIT (10RUNI) = ' (G/L) '
CHRG (10RUNI) = 0.000
CNTBP (10RUNI) = 2.000
C
CNAME (10RUMD) = 'RU(MD) '
FMASS (10RUMD) = 106.000
CVUNIT(10RUMD) = 106.000
CUNIT (10RUMD) = ' (G/L) '
CHRG (10RUMD) = 0.000
CNTBP (10RUMD) = 0.000
C
CNAME (10CS ) = 'CS(1) '
FMASS (10CS ) = 137.000
CVUNIT(10CS ) = 137.000
CUNIT (10CS ) = ' (G/L) '
CHRG (10CS ) = 1.000
CNTBP (10CS ) = 1.000
C
CNAME (10CE ) = 'CE(111) '
FMASS (10CE ) = 144.000
CVUNIT(10CE ) = 144.000
CUNIT (10CE ) = ' (G/L) '

```



```

CHRG (10CE ) = 3.000
CNTBP (10CE ) = 3.000
C
CNAME (10GD ) = 'GD(111)'
FMASS (10GD ) = 153.000
CVUNIT(10GD ) = 153.000
CUNIT (10GD ) = '(G/L)'
CHRG (10GD ) = 3.000
CNTBP (10GD ) = 3.000
C
CNAME (10AM ) = 'AM(111)'
FMASS (10AM ) = 242.000
CVUNIT(10AM ) = 242.000
CUNIT (10AM ) = '(G/L)'
CHRG (10AM ) = 3.000
CNTBP (10AM ) = 3.000
C
CNAME (10CM ) = 'CM(111)'
FMASS (10CM ) = 243.000
CVUNIT(10CM ) = 243.000
CUNIT (10CM ) = '(G/L)'
CHRG (10CM ) = 3.000
CNTBP (10CM ) = 3.000
C
CNAME ( 32 ) = 'ADD-MIC1'
CVUNIT( 32 ) = 1.000
CUNIT ( 32 ) = '(G/L)'
C
CNAME ( 33 ) = 'ADD-MIC2'
CVUNIT( 33 ) = 1.000
CUNIT ( 33 ) = '(G/L)'
C
CNAME ( 34 ) = 'ADD-MIC3'
CVUNIT( 34 ) = 1.000
CUNIT ( 34 ) = '(G/L)'
C
CNAME ( 35 ) = 'ADD-MIC4'
CVUNIT( 35 ) = 1.000
CUNIT ( 35 ) = '(G/L)'
C
CNAME ( 36 ) = 'ADD-MIC5'
CVUNIT( 36 ) = 1.000
CUNIT ( 36 ) = '(G/L)'
C
CNAME ( 37 ) = 'ADD-MIC6'
CVUNIT( 37 ) = 1.000
CUNIT ( 37 ) = '(G/L)'
C
CNAME ( 38 ) = 'ADD-MIC7'
CVUNIT( 38 ) = 1.000
CUNIT ( 38 ) = '(G/L)'
C
CNAME ( 39 ) = 'ADD-MIC8'
CVUNIT( 39 ) = 1.000
CUNIT ( 39 ) = '(G/L)'
C
CNAME ( 40 ) = 'ADD-MIC9'
CVUNIT( 40 ) = 1.000
CUNIT ( 40 ) = '(G/L)'
C
C
C
C
C
... PUT INPUT DATA INTO WORK FILE(IN).
C
PROGRAM = 'MIXSET'
C
OPEN (UNIT=IN)
C
CALL FDCARD(1SYSIN,IN ,0 ,PROGRAM)
C
... READ INPUT DATA FROM WORK FILE INTO VARIABLES
C
CALL INPUT(IN ,IUPRT ,IUPLT ,IPRDIS ,TITLE ,CTFILE,RXLIM ,
*
* MXTAUC ,TMCONB ,TMCONC ,DTCONS ,
*
* MXEXTR ,MXDEST ,MXCOMP ,MXUNIT ,MXSTG ,MXSTG2 ,NCHEM ,
*
* MXREAC ,MXBLK ,NUNIT ,NBLK ,NEXTRA ,
*
* ENDTIM ,DIO ,DTMAX ,DTMIN ,DTPRT ,DTPLT ,TRATE ,
*
* ICALC ,ITMAX ,EPS ,ITMIN ,EPSLIM ,
*
* ISFIN ,FRC ,CO ,TO ,CDBP ,FDEG ,
*
* CNAME ,FMASS ,DECAYH ,CVUNIT ,CUNIT ,CHRG ,CNTBP ,
*
* TFEED ,
*
* NFA ,NFO ,
*
* MXFEED ,MXTABL ,NFFTAB ,FFTAB ,
*
* MXFTAB ,XFTAB ,IFDSTA ,IFDSTO ,
*
* ID ,IREACT ,FCHEM ,ECHEM ,TYPE ,MSTG ,
*
* IUDSTA ,IUDSTO ,MDEST ,MEXTRA ,TAV ,VTOT ,
*
* ACID ,ACIDIN ,OXID ,FOUT ,RFOUT ,FFOUT ,
*
* IDIST ,IEFF ,CDIS ,CKEQU ,EFF ,

```

```

*
* TSTG ,V ,VS ,HEIGHT,FR ,
*
* IPOS ,
*
* X )
C
C
CLOSE(UNIT=IN)
C
C
--- INITIALIZATION ---
C
DO 520 IUNT = 1 , NUNIT
C
DEFAULT IREACT=1 FOR ALL REACTION CONSIDERD
IF ( IREACT(IUNT) .EQ. 0 ) THEN
C
IREACT=0 FOR MIXSET98'S REACTION CONSIDERD
FCHEM( 5,2,IUNT) = 0.000
DO L = 6 , 26
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
FCHEM(27,2,IUNT) = 0.000
DO L = 28 , 40
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
DO L = 43 , MXREAC
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
END IF
C
IF ( IREACT(IUNT) .EQ. 2 ) THEN
C
IREACT=2 FOR NO REACTION CONSIDERD
DO L = 1 , MXREAC
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
END IF
C
IF ( IREACT(IUNT) .EQ. 3 ) THEN
C
IREACT=3 FOR REVISED MIXSET'S HAN REACTION CONSIDERD
FCHEM( 1,1,IUNT) = 0.000
FCHEM( 1,2,IUNT) = 0.000
FCHEM( 3,1,IUNT) = 0.000
FCHEM( 3,2,IUNT) = 0.000
FCHEM( 5,2,IUNT) = 0.000
DO L = 6 , 26
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
FCHEM(27,2,IUNT) = 0.000
DO L = 28 , MXREAC
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
END IF
C
IF ( IREACT(IUNT) .EQ. 4 ) THEN
C
IREACT=4 FOR REVISED MIXSET'S U(IV) REACTION CONSIDERD
FCHEM( 2,1,IUNT) = 0.000
FCHEM( 2,2,IUNT) = 0.000
FCHEM( 3,1,IUNT) = 0.000
FCHEM( 3,2,IUNT) = 0.000
FCHEM( 5,2,IUNT) = 0.000
DO L = 6 , 40
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
DO L = 43 , MXREAC
FCHEM(L,1,IUNT) = 0.000
FCHEM(L,2,IUNT) = 0.000
ENDDO
END IF
C
520 CONTINUE
C
C
FMASS (10TCU6) = FMASS (10TC7 ) + FMASS (10U6 )
DECAYH(10TCU6) = ( DECAYH(10TC7 )*FMASS (10TC7 )
*
* + DECAYH(10U6 )*FMASS (10U6 ) )
*
* /( FMASS (10TC7 ) + FMASS (10U6 ) )
C
FMASS (10TCP4) = FMASS (10TC7 ) + FMASS (10PU4 )
DECAYH(10TCP4) = ( DECAYH(10TC7 )*FMASS (10TC7 )
*
* + DECAYH(10PU4 )*FMASS (10PU4 ) )
*
* /( FMASS (10TC7 ) + FMASS (10PU4 ) )
C
FMASS (10TCZR) = FMASS (10TC7 ) + FMASS (10ZR )
DECAYH(10TCZR) = ( DECAYH(10TC7 )*FMASS (10TC7 )
*
* + DECAYH(10ZR )*FMASS (10ZR ) )

```

```

*          /( FMASS (10TC7) + FMASS (10ZR) )
C
C
CALL INIT(MXCOMP,MXUNIT,MXSTG,NCHEM,MXBLK,MXEXTR,
*        NUNIT,NBLK,
*        MSTG,MEXTRA,
*        PHIOLD,PHI,PHIS,DIS,
*        X,XN)
C
C
C
DO 540 IUNT = 1, NUNIT
C
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C
IEXTRA = MEXTRA(IUNT)
C
C
IDEXTR(IEXTRA) = ID(IUNT)
C
C
DO 530 J = 1, MXCOMP
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = '***'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = '***'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDCONS ) THEN
C
C
CHRDIS(J,IEXTRA) = 'D-CONST'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDEQUI ) THEN
C
C
CHRDIS(J,IEXTRA) = 'K-CONST'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = '***'
C
C
ELSE
C
C
CHRDIS(J,IEXTRA) = '---'
C
C
END IF
C
C
530 CONTINUE
C
C
DO J = IOHNO3, IOPU4
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'KFK'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'HANFORD'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'Kfk*'
C
C
END IF
C
C
END DO
C
C
J = IOU4
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'KFK'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'R.MIXSET'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'Kfk***'
C
C
END IF
C
C
J = IOHNO2
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'JAERI'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'JAERI'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'A.MIXSET'
C
C
END IF
C
C
DO J = IOHP4, IOHP5
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'Kfk'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'Kfk'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'A.MIXSET'
C
C
END IF
C
C
END DO
C
C
J = IOHP6
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'Kfk'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'MIXSET98'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'A.MIXSET'
C
C
END IF
C
C
J = 10TC7
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN
C
C
CHRDIS(J,IEXTRA) = 'CEA'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'ORNL'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'CEA'
C
C
END IF
C
C
DO J = 10TCU6, 10TCZR
C
C
IF ( IDIST(J,IEXTRA) .EQ. IDKFK ) THEN

```

```

C
C
CHRDIS(J,IEXTRA) = 'CEA'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDHANF ) THEN
C
C
CHRDIS(J,IEXTRA) = 'ORNL'
C
C
ELSE IF ( IDIST(J,IEXTRA) .EQ. IDAMIX ) THEN
C
C
CHRDIS(J,IEXTRA) = 'CEA*'
C
C
END IF
C
C
END DO
C
C
END IF
C
C
540 CONTINUE
C
C
... CONTROL TRANSIENT CALCULATION
C
C
C
CALL TRAN
C
C
* (IUPRT,IUPLT,IRST,IPRDIS,TITLE,CTFILE,RKLLIM,
*
*
MXTAUC,TMCONB,TMCONC,DTCONS,
*
MXCOMP,MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,MXREAC,MXDEST,
*
MXEXTR,NUNIT,NBLK,NEXTRA,ENDTIM,DTO,DTMAX,DTMIN,DTPRT,
*
DTPLT,TRATE,ITMAX,EPS,ITMIN,EPSLIM,ISFIN,FRC,CO,
*
TO,CDBP,FDEG,RCON,CNAME,FMASS,DECAYH,CVUNIT,CUNIT,
*
CHRG,CNTBP,CPHASE,MXFEED,MXTABL,NFA,NFO,TFEED,FF,
*
NFFTAB,FFTAB,XF,NXFTAB,XTAB,IFDSTA,IFDSTO,
*
ID,IReact,FCHEM,ECHEM,TYPE,MSTG,IUDSTA,IUDSTO,MDEST,
*
MEXTRA,TAV,VTOT,ACID,ACIDIN,OXID,FIN,FOUT,FFOUT,
*
FFOUT,FMSIN,FMSOUT,FMTIN,FMTOUT,FMTBL,
*
IDEXTR,CHRDIS,IDIST,IEFF,CDIS,CKEQU,EFF,DIS,TSTG,
*
V,VS,HEIGHT,FL,FR,PHIOLD,PHI,PHIS,WLT,
*
WST,HOS,TF,FLS,FLSMT,IPFLS,IPOS,XN,X,
*
DX,REAC,DFDX,DDDX,FKEQ1,FKEQ2,FKEQ3,FKEQ4,
*
DK1,DK2,DK3,DKC,DK1DX,DK2DX,DK3DX,DKCDX,DADX,
*
DBDX,DTFDX,DISTC,DDCDX,DDCOTF,
*
MXN,MXSYS,MXRM,NXRM,FX,DFDXD,DFDXL,DFDXU,
*
IP,WK,FM,IROWFM,ICOLFM,
*
IPOSFM,EM,IROWEM,SYSM,Y,IPSYS)
C
C
STOP
C
C
END

```

```

SUBROUTINE CALCPH(MXUNIT, NUNIT, MXSTG, MEXEXTR,
*             TYPE, MSTG, MEXTRA,
*             HEIGHT, PHIOLD, PHI, PHIS,
*             FL, FR )
C
C*****
C
C MAKE HOLDUP RATIO
C
C
C
C*****
C
C IMPLICIT REAL*8 ( A-H, O-Z )
C
C CHARACTER*2 TYPE ( MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION MEXTRA( MXUNIT)
C
C DIMENSION HEIGHT( MXSTG, MEXEXTR),
*             PHIOLD( MXSTG, MEXEXTR),
*             PHI ( MXSTG, MEXEXTR),
*             PHIS ( MXSTG, MEXEXTR),
*             FL (2, MXSTG, MEXEXTR),
*             FR ( MXSTG, MEXEXTR)
C
C DATA PHIMIN/ 0.05D0 /
C DATA PHIMAX/ 0.95D0 /
C
C DO 300 IUNT = 1, NUNIT
C
C     NSTG = MSTG (IUNT)
C     IEXTRA = MEXTRA(IUNT)
C
C     IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C         DO 200 i = 1, NSTG
C
C             FRA = FR( i, IEXTRA)
C             FA = FL(1, i, IEXTRA)
C             FO = FL(2, i, IEXTRA)
C
C             FMA = FA + FRA
C             FMO = FO
C
C             PHIX = FMA / ( FMA + FMO )
C
C             IF ( PHIX .LE. PHIMIN ) THEN
C                 PHIX = PHIMIN
C             END IF
C
C             IF ( PHIX .GE. PHIMAX ) THEN
C                 PHIX = PHIMAX
C             END IF
C
C             PHIOLD(i, IEXTRA) = PHI ( i, IEXTRA)
C             PHI ( i, IEXTRA) = PHIX
C             PHIS (i, IEXTRA) = HEIGHT(i, IEXTRA)
C
C 200 CONTINUE
C
C     END IF
C
C 300 CONTINUE
C
C RETURN
C END

```

```

SUBROUTINE CALFLW(MXUNIT, NUNIT, MXSTG, MXDEST, MEXEXTR, NEXTRA,
*             MXFEED, NFA, NFO,
*             IFDSTA, IFDSTO, IUDSTA, IUDSTO, MDEST, MEXTRA,
*             FOUT, RFOUT, ACIDIN, FFOUT,
*             FF, FL,
*             FLS, FLSMT, IPFLS,
*             MSTG )
C
C*****
C
C CALCULATE FLOW RATE FOR EVERY DIVISION.
C
C*****
C
C IMPLICIT REAL*8 ( A-H, O-Z )
C
C DIMENSION FF( MXFEED, 2),
*             IFDSTA(2, MXFEED),
*             IFDSTO(2, MXFEED)
C
C DIMENSION MSTG ( MXUNIT)
C
C DIMENSION IUDSTA(3, MXDEST, MXUNIT),
*             IUDSTO(3, MXDEST, MXUNIT)
C DIMENSION MDEST ( 2, MXUNIT),
*             MEXTRA( MXUNIT)
C DIMENSION FOUT ( 2, MXUNIT),
*             RFOUT (MXDEST, 2, MXUNIT),
*             ACIDIN( MXUNIT)
C DIMENSION FFOUT(2, 2, MXUNIT)
C
C DIMENSION FL (2, MXSTG, MEXEXTR)
C
C DIMENSION FLS ( MXUNIT, 2)
C DIMENSION FLSMT ( MXUNIT, MXUNIT, 2)
C DIMENSION IPFLS ( MXUNIT, 2)
C
C DO 120 IUNT = 1, NUNIT
C DO 110 IUNTIX = 1, NUNIT
C     FLSMT(IUNTIX, IUNT, 1) = 0.0D0
C     FLSMT(IUNTIX, IUNT, 2) = 0.0D0
110 CONTINUE
C     FLSMT(IUNT, IUNT, 1) = 1.0D0
C     FLSMT(IUNT, IUNT, 2) = 1.0D0
120 CONTINUE
C
C DO 230 IUNT = 1, NUNIT
C
C     NDEST = MDEST(1, IUNT)
C
C     IF ( NDEST .GT. 0 ) THEN
C
C         IF ( NDEST .EQ. 2 ) THEN
C             KDEST = 2
C             IF ( FOUT(1, IUNT) .GT. 0.0D0 ) THEN
C                 RFTMP = FFOUT(KDEST, 1, IUNT) / FOUT(1, IUNT)
C                 IF ( RFTMP .GT. 1.0D1 ) RFTMP = 1.0D0
C             ELSE
C                 RFTMP = 0.0D0
C             END IF
C             RFOUT(KDEST, 1, IUNT) = RFTMP
C             RFTMP = 1.0D0 - RFOUT(KDEST, 1, IUNT)
C             KDEST = 1
C             IF ( RFTMP .GT. 0.0D0 ) RFOUT(KDEST, 1, IUNT) = RFTMP
C         END IF
C
C         DO 210 KDEST = 1, NDEST
C             IUNTIX = IUDSTA(1, KDEST, IUNT)
C             IF ( IUNTIX .NE. 0 ) THEN
C                 FLSMT(IUNTIX, IUNT, 1) = FLSMT(IUNTIX, IUNT, 1)
C                 - RFOUT(KDEST, 1, IUNT)
C             END IF
C         CONTINUE
210 CONTINUE
C
C     END IF
C
C     NDEST = MDEST(2, IUNT)
C
C     IF ( NDEST .GT. 0 ) THEN
C
C         IF ( NDEST .EQ. 2 ) THEN
C             KDEST = 2
C             IF ( FOUT(2, IUNT) .GT. 0.0D0 ) THEN
C                 RFTMP = FFOUT(KDEST, 2, IUNT) / FOUT(2, IUNT)
C                 IF ( RFTMP .GT. 1.0D1 ) RFTMP = 1.0D0
C             END IF

```

```

ELSE
  RFTMP = 0.000
END IF
RFOUT(KDEST,2,IUNT) = RFTMP
RFTMP = 1.000 - RFOUT(KDEST,2,IUNT)
KDEST = 1
IF ( RFTMP .GE. 0.000 ) RFOUT(KDEST,2,IUNT) = RFTMP
END IF
C
DO 220 KDEST = 1 , NDEST
  IUNT = IUDSTO(1,KDEST,IUNT)
  IF ( IUNT .NE. 0 ) THEN
    FLSMT(IUNTX,IUNT,2) = FLSMT(IUNTX,IUNT,2)
    *
    - RFOUT(KDEST,2,IUNT)
  END IF
C
220 CONTINUE
C
END IF
C
230 CONTINUE
C
DO 250 IUNT = 1 , NUNIT
  IF ( ACIDIN(IUNT) .GT. 0.000 ) THEN
    DO 240 IUNT = 1 , NUNIT
      FLSMT(IUNT,IUNT,1) = 0.000
    240 CONTINUE
      FLSMT(IUNT,IUNT,1) = 1.000
    END IF
  250 CONTINUE
C
DO 270 IUNT = 1 , NUNIT
  IF ( ACIDIN(IUNT) .GT. 0.000 ) THEN
    DO 260 IUNT = 1 , NUNIT
      FLSMT(IUNT,IUNT,2) = 0.000
    260 CONTINUE
      FLSMT(IUNT,IUNT,2) = 1.000
    END IF
  270 CONTINUE
C
C
C
LU-DECOMPOSITION OF FLOW RATE EQUATION(AQUEOUS)
CALL LUDCOM(FLSMT(1,1,1) ,NUNIT ,MXUNIT ,
*
EPSX ,FLS(1,1) , IPFLS(1,1) ,IER )
C
IF ( IER .NE. 0 ) THEN
  WRITE(6,*) ' FLOW RATE EQUATION IS ILLPOSED(AQUEOUS).'
  STOP
END IF
C
DO 310 IUNT = 1 , NUNIT
  FLS(IUNT,1) = 0.000
310 CONTINUE
C
DO 320 IFA = 1 , NFA
  IUNT = IFDSTA(1,IFA)
  IF ( IUNT .NE. 0 ) THEN
    FLS(IUNT,1) = FLS(IUNT,1) + FF(IFA,1)
  END IF
C
320 CONTINUE
C
DO 330 IUNT = 1 , NUNIT
  IF ( ACIDIN(IUNT) .GT. 0.000 ) THEN
    FLS(IUNT,1) = FOUT(1,IUNT)
  END IF
330 CONTINUE
C
CALL LUSOLV(FLSMT(1,1,1) ,NUNIT ,MXUNIT ,FLS(1,1) ,IPFLS(1,1) )
C
C
C
LU-DECOMPOSITION OF FLOW RATE EQUATION(ORGANIC)
CALL LUDCOM(FLSMT(1,1,2) ,NUNIT ,MXUNIT ,
*
EPSX ,FLS(1,2) , IPFLS(1,2) ,IER )
C
IF ( IER .NE. 0 ) THEN
  WRITE(6,*) ' FLOW RATE EQUATION IS ILLPOSED(ORGANIC).'
  STOP
END IF

```

```

C
DO 340 IUNT = 1 , NUNIT
  FLS(IUNT,2) = 0.000
340 CONTINUE
C
DO 350 IFO = 1 , NFO
  IUNT = IFDSTO(1,IFO)
  IF ( IUNT .NE. 0 ) THEN
    FLS(IUNT,2) = FLS(IUNT,2) + FF(IFO,2)
  END IF
C
350 CONTINUE
C
DO 360 IUNT = 1 , NUNIT
  IF ( ACIDIN(IUNT) .GT. 0.000 ) THEN
    FLS(IUNT,2) = FOUT(2,IUNT)
  END IF
360 CONTINUE
C
CALL LUSOLV(FLSMT(1,1,2) ,NUNIT ,MXUNIT ,FLS(1,2) ,IPFLS(1,2) )
C
DO 400 IUNT = 1 , NUNIT
  FOUT(1,IUNT) = FLS(IUNT,1)
  FOUT(2,IUNT) = FLS(IUNT,2)
400 CONTINUE
C
IF ( NEXTRA .LE. 0 ) THEN
  RETURN
END IF
C
C
C
SET FLOW RATES FOR EACH STAGE
C
DO 420 IUNT = 1 , NUNIT
  NSTG = MSTG (IUNT)
  IEXTRA = MEXTRA(IUNT)
  IF ( IEXTRA .GT. 0 ) THEN
    DO 410 I = 1 , NSTG
      FL(1,I,IEXTRA) = 0.000
      FL(2,I,IEXTRA) = 0.000
    410 CONTINUE
  END IF
C
420 CONTINUE
C
DO 520 IFA = 1 , NFA
  IUNT = IFDSTA(1,IFA)
  KSTGX = IFDSTA(2,IFA)
  FFA = FF(IFA,1)
  IF ( IUNT .NE. 0 ) THEN
    IEXTRA = MEXTRA(IUNT)
    IF ( IEXTRA .GT. 0 ) THEN
      DO 510 I = 1 , KSTGX
        FL(1,I,IEXTRA) = FL(1,I,IEXTRA) + FFA
      510 CONTINUE
    END IF
  END IF
C
520 CONTINUE
C
DO 540 IFO = 1 , NFO
  IUNT = IFDSTO(1,IFO)
  KSTGX = IFDSTO(2,IFO)
  FFO = FF(IFO,2)

```

```

C
C      IF ( IUNTX .NE. 0 ) THEN
C
C      NSTGX = MSTG (IUNTX)
C      IEXTRA = MEXTRA(IUNTX)
C
C      IF ( IEXTRA .GT. 0 ) THEN
C
C      DO 530 I = KSTGX , NSTGX
C      FL(2,I,IEXTRA) = FL(2,I,IEXTRA) + FFO
530      CONTINUE
C
C      END IF
C
C      END IF
C
C      540 CONTINUE
C
C
C      DO 650 IUNT = 1 , NUNIT
C
C      AQUEOUS PHASE FLOW RATE
C
C      NDEST = MDEST(1,IUNT)
C
C      IF ( NDEST .GT. 0 ) THEN
C
C      DO 620 KDEST = 1 , NDEST
C
C      IUNTX = IUDSTA(1,KDEST,IUNT)
C      KSTGX = IUDSTA(2,KDEST,IUNT)
C      FLSA = FOUT(1,IUNT)*RFOUT(KDEST,1,IUNT)
C
C      IF ( IUNTX .NE. 0 ) THEN
C
C      IEXTRA = MEXTRA(IUNTX)
C
C      IF ( IEXTRA .GT. 0 ) THEN
C
C      DO 610 I = 1 , KSTGX
C      FL(1,I,IEXTRA) = FL(1,I,IEXTRA) + FLSA
610      CONTINUE
C
C      END IF
C
C      END IF
C
C      620 CONTINUE
C
C      END IF
C
C      ORGANIC PHASE FLOW RATE
C
C      NDEST = MDEST(2,IUNT)
C
C      IF ( NDEST .GT. 0 ) THEN
C
C      DO 640 KDEST = 1 , NDEST
C
C      IUNTX = IUDSTO(1,KDEST,IUNT)
C      KSTGX = IUDSTO(2,KDEST,IUNT)
C      FLSO = FOUT(2,IUNT)*RFOUT(KDEST,2,IUNT)
C
C      IF ( IUNTX .NE. 0 ) THEN
C
C      NSTGX = MSTG (IUNTX)
C      IEXTRA = MEXTRA(IUNTX)
C
C      IF ( IEXTRA .GT. 0 ) THEN
C
C      DO 630 I = KSTGX , NSTGX
C      FL(2,I,IEXTRA) = FL(2,I,IEXTRA) + FLSO
630      CONTINUE
C
C      END IF
C
C      END IF
C
C      640 CONTINUE
C
C      END IF
C
C      650 CONTINUE
C
C      RETURN
C      END

```

```

SUBROUTINE CAPMAK(M ,NXRM ,MXBLK ,MXPRM ,MXSYS ,
*          NM ,
*          IROWPRM,ICOLRM,IPOSRM,IROWEM,
*          MSYS ,MSYSN )
C
C
C      IMPLICIT REAL*8(A-H,O-Z)
C
C      DIMENSION IROWPRM(MXPRM ) ,
*          ICOLRM(MXPRM )
C
C      DIMENSION IPOSRM(MXBLK ,MXSYS)
C      DIMENSION IROWEM(      MXSYS)
C
C
C      MSYS = 0
C
C      DO 400 IXRM = 1 , NXRM
C
C      IF ( MSYS .GT. 0 ) THEN
C      DO 100 MMM = 1 , MSYS
C      MM = MMM
C      IF ( IROWEM(MMM) .EQ. IROWPRM(IXRM) ) GO TO 300
100      CONTINUE
C      END IF
C
C      IF ( MSYS .EQ. MXSYS ) THEN
C      WRITE(6,*) ' TOO MANY BLOCKS OF SYSTEM MATRIX'
C      WRITE(6,*) ' MAXIMUM NUMBER OF BLOCKS IS ',MXSYS
C      END IF
C
C      MSYS = MSYS + 1
C      MM = MSYS
C
C      DO 200 I = 1 , M
C      IPOSRM(I,MM) = 0
200      CONTINUE
C      IROWEM(MM) = IROWPRM(IXRM)
C
C      300 CONTINUE
C      IF ( IPOSRM(ICOLRM(IXRM),MM) .NE. 0 ) THEN
C      WRITE(6,*) ' ERROR AT CAPMAK.'
C      WRITE(6,*) ' TWO SUB-MATRICES OCCUPY SAME POSITION.'
C      WRITE(6,*) ' SUBMATRIX NO. ',IXRM
C      WRITE(6,*) ' POSITION (ROW,COLUMN) IS ',
*          IROWPRM(IXRM),ICOLRM(IXRM)
C
C      STOP
C      END IF
C
C      IPOSRM(ICOLRM(IXRM),MM) = IXRM
C      400 CONTINUE
C
C
C      MSYSN = 0
C      DO 500 MM = 1 , MSYS
C      MSYSN = MSYSN + NM
500 CONTINUE
C
C
C      RETURN
C      END

```

```

SUBROUTINE CAPSLV(NXFM ,MXFM ,MXSYS ,
* M ,MXN ,MXBLK ,
* NM ,D ,E ,F ,
* B ,IP ,WK ,
* FM ,IROWFM ,ICOLFM ,IPOSFM ,
* EM ,IROWEM ,
* SYSM ,Y ,IPSYS ,EPSX ,IER )
C
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C
C DIMENSION D (MXN ,MXN ,MXBLK),
* E (MXN ,MXN ,MXBLK),
* F (MXN ,MXN ,MXBLK),
* B (MXN ,MXBLK),
* IP (MXN ,MXBLK),
* WK (MXN )
C DIMENSION FM (MXN ,MXN ,MXFM ),
* IROWFM ( MXFM ),
* ICOLFM ( MXFM )
C DIMENSION IPOSFM (MXBLK ,MXSYS)
C
C DIMENSION EM (MXN ,MXN ,MXBLK ,MXSYS)
C DIMENSION IROWEM (MXSYS)
C
C DIMENSION SYSM (MXN *MXSYS ,MXN *MXSYS),
* Y (MXN *MXSYS),
* IPSYS (MXN *MXSYS)
C
C DATA ONE /1.000/
C DATA ZERO /0.000/
C
C TX = ( E + D + F ) ** (-1) * B
C
C CALL TRIBLK (MXN ,M ,NM ,D ,E ,F ,B ,IP ,WK ,EPSX ,IER)
C IF ( IER .NE. 0 ) THEN
C RETURN
C END IF
C
C IF NXFM = 0 , SOLVING SYSTEM MATRIX IS NOT NECESSARY.
C IF ( NXFM .LE. 0 ) RETURN
C
C CALL CAPMAK (M ,NXFM ,MXBLK ,MXFM ,MXSYS ,
* NM ,
* IROWFM ,ICOLFM ,IPOSFM ,IROWEM ,
* MSYS ,MSYSN )
C
C TE = B ** (-1) * E
C
C DO 1000 MM = 1 , MSYS
C IX = IROWEM (MM)
C
C DO 130 I = 1 , M
C DO 120 J = 1 , NM
C DO 110 K = 1 , NM
C EM (K ,J ,I ,MM) = ZERO
110 CONTINUE
120 CONTINUE
130 CONTINUE
C
C I = IX
C
C DO 220 J = 1 , NM
C DO 210 K = 1 , NM
C EM (K ,J ,I ,MM) = D (K ,J ,I)
210 CONTINUE
220 CONTINUE
C
C --- FORWARD SUBSTITUTION ---
C
C DO 400 I = IX + 1 , M
C
C DO 350 J = 1 , NM
C DO 340 K = 1 , NM
C S = 0.000
C DO 330 L = 1 , NM
C S = S + E (K ,L ,I) * EM (L ,J ,I -1 ,MM)
330 CONTINUE

```

```

EM (K ,J ,I ,MM) = - S
340 CONTINUE
350 CONTINUE
C
C DO 390 J = 1 , NM
C
C DO 360 K = 1 , NM
C WK (K) = EM (K ,J ,I ,MM)
360 CONTINUE
C
C DO 380 K = 1 , NM
C S = 0.000
C DO 370 L = 1 , NM
C S = S + D (K ,L ,I) * WK (L)
370 CONTINUE
C EM (K ,J ,I ,MM) = S
C
C 380 CONTINUE
C
C 390 CONTINUE
C
C 400 CONTINUE
C
C BACKWARD SUBSTITUTION
C
C DO 440 I = M -1 , 1 , -1
C
C DO 430 J = 1 , NM
C DO 420 K = 1 , NM
C S = 0.000
C DO 410 L = 1 , NM
C S = S + F (K ,L ,I) * EM (L ,J ,I +1 ,MM)
410 CONTINUE
C EM (K ,J ,I ,MM) = EM (K ,J ,I ,MM) - S
420 CONTINUE
430 CONTINUE
C
C 440 CONTINUE
C
C 1000 CONTINUE
C
C
C
C SYSM = 1 + FM * TE
C
C JSYS = 0
C
C DO 1160 MMM = 1 , MSYS
C IXX = IROWEM (MMM)
C DO 1150 KK = 1 , NM
C
C JSYS = JSYS + 1
C
C ISYS = 0
C
C DO 1140 MM = 1 , MSYS
C IX = IROWEM (MM)
C DO 1130 K = 1 , NM
C
C ISYS = ISYS + 1
C S = ZERO
C
C DO 1120 I = 1 , M
C IXFM = IPOSFM (I ,MM)
C IF ( IXFM .GT. 0 ) THEN
C
C DO 1110 L = 1 , NM
C S = S + FM (K ,L ,IXFM) * EM (L ,KK ,I ,MMM)
1110 CONTINUE
C
C END IF
C
C 1120 CONTINUE
C
C SYSM (ISYS ,JSYS) = S
C
C 1130 CONTINUE
C 1140 CONTINUE
C
C 1150 CONTINUE
C 1160 CONTINUE
C
C DO 1170 J = 1 , MSYSN
C SYSM (J ,J) = SYSM (J ,J) + ONE
1170 CONTINUE
C

```

```

C
C
C LU-DECOMPOSITION OF SYSTEM MATRIX.
C
CALL LUDCOM(SYSM ,MSYSN ,MXN*MXXSYS ,EPSX ,Y ,IPSYS ,IER )
IF ( IER .NE. 0 ) THEN
  RETURN
END IF

C
C
C
C SB = RM * ( E + D + F )**(-1) * B   ( = RM * TX )
C
ISYS = 0

C
DO 1240 MM = 1 , MSYS
  IX = IROWEM(MM)
  DO 1230 K = 1 , NM

C
  ISYS = ISYS + 1
  S = ZERO

C
  DO 1220 I = 1 , M
    IXRM = IPOSRM(I,MM)
    IF ( IXRM .GT. 0 ) THEN

C
      DO 1210 L = 1 , NM
        S = S + RM(K,L,IXRM) * B(L,I)
1210      CONTINUE

C
      END IF

C
1220    CONTINUE

C
  Y(ISYS) = S

C
1230  CONTINUE
1240 CONTINUE

C
C
C
C Y = ( SYSM )**(-1) * SB
C
CALL LUSOLV(SYSM ,MSYSN ,MXN*MXXSYS ,Y ,IPSYS )

C
C
C
C X = TX - TE * Y

C
DO 1340 I = 1 , M
  DO 1330 K = 1 , NM

C
  S = ZERO
  ISYS = 0
  DO 1320 MM = 1 , MSYS
    IX = IROWEM(MM)
    DO 1310 L = 1 , NM
      ISYS = ISYS + 1
      S = S + EM(K,L,I,MM)*Y(ISYS)
1310    CONTINUE
1320  CONTINUE

C
  B(K,I) = B(K,I) - S

C
1330 CONTINUE
1340 CONTINUE

C
C
C
C RETURN
END

```

```

BLOCK DATA CONST
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C INDICES FOR CHEMICAL COMPONENTS
C
COMMON /IXCOMP/NCHEMO,
*      IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
*      IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
*      IONP4 , IONP5 , IONP6 , IOZR , IOZC4 ,
*      IOTC5 , IOTC6 , IOTC7 , IOTCU6, IOTCP4,
*      IOTCZR, IOSR , IOFUDI, IOFUTR, IOFUNI,
*      IOFUMO, IOCS , IOCE , IOGD , IOAM ,
*      IOCM
C
COMMON /IICOMP/I1HNO3, I1U , I1PU , I1NP , I1ZR ,
*      I1TC , I1SR , I1FU , I1CS , I1CE ,
*      I1GD , I1AM , I1CM
C
DATA NCHEMO/18/
DATA IOHNO3/ 1/, IOU6 / 2/, IOPU4 / 3/, IOPU3 / 4/, IOU4 / 5/,
* IOHNO2/ 6/, IOHYD / 7/, IOHAN / 8/, IOPU5 / 9/, IOPU6 /10/,
* IONP4 /11/, IONP5 /12/, IONP6 /13/, IOZR /14/, IOZC4 /15/,
* IOTC5 /16/, IOTC6 /17/, IOTC7 /18/, IOTCU6/19/, IOTCP4/20/,
* IOTCZR/21/, IOSR /22/, IOFUDI/23/, IOFUTR/24/, IOFUNI/25/,
* IOFUMO/26/, IOCS /27/, IOCE /28/, IOGD /29/, IOAM /30/,
* IOCM /31/
C
DATA I1HNO3/ 1/, I1U / 2/, I1PU / 3/, I1NP / 4/, I1ZR / 5/,
* I1TC / 6/, I1SR / 7/, I1FU / 8/, I1CS / 9/, I1CE /10/,
* I1GD /11/, I1AM /12/, I1CM /13/
END
    
```

```

SUBROUTINE CONVCK(ICONV , EPS , RXLIM , EPSLIM, ITMAX , ITERL ,
*                MXCOMP, MXUNIT, MXSTG2,
*                JBASE , NCOMP , MXBLK , NUNIT ,
*                NBLK , TYPE , MSTG ,
*                IPOS ,
*                XN , DX )
C
C*****
C
C CALCULATE RELATIVE ERROR AND JUDGE THE CONVERGENCE.
C*****
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER*2 TYPE( MXUNIT)
C DIMENSION MSTG( MXUNIT)
C
C DIMENSION IPOS(2, MXSTG2, MXUNIT)
C
C DIMENSION XN(MXCOMP, 2, MXBLK ) ,
*           DX(MXCOMP, 2, MXBLK )
C
C INDICES FOR CHEMICAL COMPONENTS
C
COMMON /IXCOMP/NCHEMO,
*      IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
*      IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
*      IONP4 , IONP5 , IONP6 , IOZR , IOZC4 ,
*      IOTC5 , IOTC6 , IOTC7 , IOTCU6, IOTCP4,
*      IOTCZR, IOSR , IOFUDI, IOFUTR, IOFUNI,
*      IOFUMO, IOCS , IOCE , IOGD , IOAM ,
*      IOCM
C
C
C
C
C EFRMAX = 0.000
C EFRMAC = 0.000
C XNMAX = 0.000
C DXMAX = 0.000
C
C JERR = 0
C KERR = 0
C IERR = 0
C
C DO 200 I = 1 , NBLK
C DO 100 J = 1 , NCOMP
C
C JJ = J + JBASE
C
C IF ( DABS(XN(JJ, 1, I)) .LE. RXLIM ) THEN
C EFR = DABS( DX(JJ, 1, I) / RXLIM )
C ELSE
C EFR = DABS( DX(JJ, 1, I) / XN(JJ, 1, I) )
C END IF
C
C IF ( ERR .GT. EFRMAX ) THEN
C KERR = 1
C JERR = JJ
C IERR = I
C EFRMAX = ERR
C XNMAX = XN(JJ, 1, I)
C DXMAX = DX(JJ, 1, I)
C END IF
C
C IF ( ERR .GT. EFRMAC ) THEN
C EFRMAC = ERR
C END IF
C
C IF ( DABS(XN(JJ, 2, I)) .LE. RXLIM ) THEN
C EFR = DABS( DX(JJ, 2, I) / RXLIM )
C ELSE
C EFR = DABS( DX(JJ, 2, I) / XN(JJ, 2, I) )
C END IF
C
C IF ( ERR .GT. EFRMAX ) THEN
C KERR = 2
C JERR = JJ
C IERR = I
C EFRMAX = ERR
C XNMAX = XN(JJ, 2, I)
C DXMAX = DX(JJ, 2, I)
C END IF
C
C IF ( ERR .GT. EFRMAC ) THEN
C EFRMAC = ERR
C END IF
    
```



```

C
C
100 CONTINUE
200 CONTINUE
C
C
C
CC IF ( JMIC .LE. 0 ) THEN
CC WRITE(6,'(A,1P3E12.3,3I3)') ' ERRMAC, XNMAX, DXMAX, J,K, I = ',
CC * ERRMAC, XNMAX, DXMAX,
CC * JERR, KEPR, IERR
CC END IF
C
C
ERR = ERRMAC
IF ( ERR .LT. EPS .AND. ERR .LT. EPSLIM ) THEN
ICONV = 2
ELSE IF ( ERR .LT. EPSLIM ) THEN
ICONV = 1
ELSE
ICONV = 0
C
C
IF ( ITERL .GE. ITMAX .AND. JMIC .GT. 0 ) THEN
WRITE(6,'(A,1P3E12.3,3I3)') ' ERRMAC, XNMAX, DXMAX, J,K, I = ',
* ERRMAC, XNMAX, DXMAX,
* JERR, KEPR, IERR
END IF
C
END IF
C
C
C
RETURN
END

```

```

SUBROUTINE DBCOEF( MXCOMP, JMIC, JBASE, NCOMP, NCHEM,
* T, FRC, X,
* CO, TO, CHRG, CNTBP, XTF, CDBP,
* IDIST,
* CDIS,
* CKEOU,
* DIS,
* DDDX,
* FKEQ1, FKEQ2, FKEQ3, FKEQC,
* DK1, DK2, DK3, DKC,
* DK1DX, DK2DX, DK3DX, DKCDX,
* DADX, DBDX, DTFDX,
* DISTC, DDCDX, DDCDTF)
C
C
C
DIS : DISTRIBUTION COEFFICIENTS
FKEQ : EQUILIBRIUM CONSTANTS
DK : DERIVATIVE OF DISTRIBUTION COEFFICIENT
T SHOULD BE GIVEN IN KELVIN.
C
C
C
IMPLICIT REAL*8 ( A-H, O-Z )
C
C
DIMENSION CHRG (MXCOMP),
* CNTBP (MXCOMP)
DIMENSION X (MXCOMP)
C
DIMENSION IDIST (MXCOMP)
C
DIMENSION CDIS (MXCOMP),
* CKEOU (MXCOMP),
* DIS (MXCOMP)
C
DIMENSION DDDX (NCHEM, NCHEM)
C
DIMENSION FKEQ1 (NCHEM),
* FKEQ2 (NCHEM),
* FKEQ3 (NCHEM),
* FKEQC (NCHEM)
DIMENSION DK1 (NCHEM),
* DK2 (NCHEM),
* DK3 (NCHEM),
* DKC (NCHEM)
DIMENSION DK1DX (NCHEM, NCHEM),
* DK2DX (NCHEM, NCHEM),
* DK3DX (NCHEM, NCHEM),
* DKCDX (NCHEM, NCHEM)
DIMENSION DADX (NCHEM),
* DBDX (NCHEM),
* DTFDX (NCHEM)
DIMENSION DISTC (NCHEM),
* DDCDX (NCHEM, NCHEM),
* DDCDTF (NCHEM)
C
COMMON /IXCOMP/NCHEM0,
* IOHNO3, IOU6, IOPU4, IOPU3, IOU4,
* IOHNO2, IOHYD, IOHAN, IOPU5, IOPU6,
* IONP4, IONP5, IONP6, IOZR, IOTC4,
* IOTC5, IOTC6, IOTC7, IOTCU6, IOTCP4,
* IOTCZR, IOSR, IORUDI, IORUTR, IORUNI,
* IORUMO, IOCS, IOCE, IOGD, IOAM,
* IOCM
C
C
C
INDICES FOR DISTRIBUTION CALCULATION OPTION
COMMON /IDISOP/IDCONS, IDEQUI, IDHANF, IDKFK, IDAMIX
C
C
C
C
PARAMETER( MXNYY = 10 )
DIMENSION YY(MXNYY), DYY(MXNYY)
C
PARAMETER( NYYPUG = 3, LENPUG = 4 )
DIMENSION TABPUG(NYYPUG+1, LENPUG)
DATA (TABPUG(1,J), J=1, LENPUG) / 2.00D0, 3.00D0, 4.00D0, 4.50D0 /
DATA (TABPUG(2,J), J=1, LENPUG) / 2.85D0, 4.02D0, 5.13D0, 5.39D0 /
DATA (TABPUG(3,J), J=1, LENPUG) / -4.91D0, -6.95D0, -8.48D0, -8.82D0 /
DATA (TABPUG(4,J), J=1, LENPUG) / 8.13D-1, 8.15D-1, 7.62D-1, 7.47D-1 /
C
PARAMETER( NYYNP4 = 3, LENNP4 = 4 )
DIMENSION TABNP4(NYYPUG+1, LENNP4)
DATA (TABNP4(1,J), J=1, LENNP4) / 1.00D0, 2.00D0, 3.00D0, 3.50D0 /
DATA (TABNP4(2,J), J=1, LENNP4) / 0.722845D0, 2.0D0, 3.20D0, 5.00D0 /
DATA (TABNP4(3,J), J=1, LENNP4) / 12.254D0, 15.856D0, 19.649D0, 20.410D0 /
DATA (TABNP4(4,J), J=1, LENNP4) / -0.76722D0, -0.76696D0,
* 0.70824D0, 0.70772D0 /

```

```

C
PARAMETER( NYYP6 = 3 , LENNP6 = 4 )
DIMENSION TABNP6(NYYNP6+1,LENNP6)
DATA (TABNP6(1,J),J=1,LENNP6)/ 1.00D0 , 2.00D0 , 3.00D0 , 3.50D0 /
DATA (TABNP6(2,J),J=1,LENNP6)/ 4.57168D0, 9.29807D0,
* 12.2062D0 ,15.622D0 /
DATA (TABNP6(3,J),J=1,LENNP6)/50.660D0,11.562D0,10.912D0,10.760D0/
DATA (TABNP6(4,J),J=1,LENNP6)/-2.71840D0,-0.69003D0,
* -0.71674D0,-0.79823D0/
-----
C
D(HNO2) FORM A.MIXSET
C
PARAMETER( NYYN02 = 1 , LENNO2 = 5 )
DIMENSION TABNO2(NYYNO2+1,LENN02)
DATA (TABNO2(1,J),J=1,LENN02)/ 0.1D00 , 0.50D0 , 1.00D0 , 2.00D0 ,
* 3.2D0 /
DATA (TABNO2(2,J),J=1,LENN02)/ 1.5D0 , 15.0D0 , 16.0D0 , 15.0D0 ,
* 6.0D0 /
-----
C
MXTFLP = 20
EPSTF = 1.0D-5
TFMIN = 1.0D-10
C
C
F = FRC
TAU = 1.0D0/T - 1.0D0/( T0 + 25.0D0 )
C
C
IF ( JMIC .GT. 0 ) GO TO 6000
C
MMACRO COMPONENTS
C
DO 10 J = 1 , NCHEM
C
FKE01(J) = 0.0D0
DK1 (J) = 0.0D0
C
FKE02(J) = 0.0D0
DK2 (J) = 0.0D0
C
FKE03(J) = 0.0D0
DK3 (J) = 0.0D0
C
FKE0C(J) = 0.0D0
DKC (J) = 0.0D0
C
10 CONTINUE
C
DO 20 J = 1 , NCHEM
DISTC(J) = 0.0D0
DDCCTF(J) = 0.0D0
20 CONTINUE
C
DO 40 K = 1 , NCHEM
DO 30 J = 1 , NCHEM
DDCDX(J,K) = 0.0D0
30 CONTINUE
40 CONTINUE
C
DO 60 K = 1 , NCOMP
DO 50 J = 1 , NCOMP
DK1DX(J,K) = 0.0D0
DK2DX(J,K) = 0.0D0
DK3DX(J,K) = 0.0D0
DKCDX(J,K) = 0.0D0
DDCDX(J,K) = 0.0D0
50 CONTINUE
60 CONTINUE
C
C
XNO3 = 0.0D0
C
DO 70 J=1,NCHEMO
XNO3 = XNO3 + CHR(J)*X(J)
70 CONTINUE
C
----- HNO3 -----
C
IF ( IDIST(10HNO3) .EQ. 1DHANF ) THEN
C
...HANFORD...
A1 = 0.1350D0*(XNO3**0.82D0)
A2 = 0.0052D0*(XNO3**3.44D0)
B1 =( 1.0D0 - 0.54D0*DEXP(-15.0D0*F) )

```

```

FTAU = DEXP(340.0D0*TAU)
C
----- TF**1 -----
C
FKE01 (10HNO3) = ( A1 + A2 ) *B1*FTAU
DK1 (10HNO3) = ( 0.8200D0*A1 + 3.440D0*A2 )/XNO3 *B1*FTAU
C
----- TF**2 -----
C
FKE02 (10HNO3) = FKE01 (10HNO3)
DK2 (10HNO3) = DK1 (10HNO3)
C
ELSE
...KFK...
A1 = 0.14160D0*(XNO3**0.6724D0)
A2 = 0.006058D0*(XNO3**3.4180D0)
B1 =( 1.0D0 - 0.54D0*DEXP(-15.0D0*F) )
IF ( IDIST(10HNO3) .EQ. 1DAMIX ) THEN
FTAU = DEXP(-600.0D0*TAU)
ELSE
FTAU = DEXP( 340.0D0*TAU)
END IF
C
----- TF**1 -----
C
FKE01 (10HNO3) = ( A1 + A2 ) *B1*FTAU
DK1 (10HNO3) = ( 0.6724D0*A1 + 3.4180D0*A2 )/XNO3 *B1*FTAU
C
----- TF**2 -----
C
FKE02 (10HNO3) = FKE01 (10HNO3)
DK2 (10HNO3) = DK1 (10HNO3)
C
END IF
C
----- U(VI) -----
C
IF ( IDIST(10U6) ) .EQ. 1DHANF ) THEN
C
...HANFORD...
A1 = 3.700D0*(XNO3**1.57D0)
A2 = 1.400D0*(XNO3**3.90D0)
A3 = 0.011D0*(XNO3**7.30D0)
B1 = ( 4.0D0*(F**(-0.17D0)) - 3.0D0 )
FKU6 = ( A1 + A2 + A3 ) *B1
DFKU6 = ( 1.57D0 *A1 + 3.90D0*A2 + 7.30D0*A3 )/XNO3 *B1
FTAU = DEXP(2500.0D0*TAU)
FKE02 (10U6 ) = FKU6 *FTAU
DK2 (10U6 ) = DFKU6 *FTAU
C
ELSE
...KFK...
A1 = 5.2840D0*(XNO3**1.582D0)
A2 = 1.5570D0*(XNO3**3.843D0)
A3 = 0.01267D0*(XNO3**7.447D0)
B1 = ( 4.0D0*(F**(-0.17D0)) - 3.0D0 )
FKU6 = ( A1 + A2 + A3 ) *B1
DFKU6 = ( 1.582D0 *A1 + 3.843D0*A2 + 7.447D0*A3 )/XNO3 *B1
IF ( IDIST(10U6) ) .EQ. 1DAMIX ) THEN
FTAU = DEXP( 600.0D0*TAU)
ELSE
FTAU = DEXP(2500.0D0*TAU)
END IF
FKE02 (10U6 ) = FKU6 *FTAU
DK2 (10U6 ) = DFKU6 *FTAU
C
END IF
C
----- PU(IV) -----
C
IF ( IDIST(10PU4) ) .EQ. 1DHANF ) THEN
C
...HANFORD...
A1 = 3.700D0*(XNO3**1.57D0)
A2 = 1.400D0*(XNO3**3.90D0)
A3 = 0.011D0*(XNO3**7.30D0)
B1 = ( 4.0D0*(F**(-0.17D0)) - 3.0D0 )
FKU6 = ( A1 + A2 + A3 ) *B1
DFKU6 = ( 1.57D0 *A1 + 3.90D0*A2 + 7.30D0*A3 )/XNO3 *B1
A1 = 0.200D0 + 0.55D0*(F**1.25D0)
A2 = 0.0074D0*XNO3*XNO3
FTAU = DEXP(-200.0D0*TAU)
FKE02 (10PU4) = FKU6*(A1 + A2)*FTAU
DK2 (10PU4) = ( FKU6*2.0D0*A2/XNO3 + DFKU6*(A1 + A2) )*FTAU

```

```

C
ELSE
C
...KFK...
A1 = 5.28400D0*(XN03**1.582D0)
A2 = 1.55700D0*(XN03**3.843D0)
A3 = 0.01267D0*(XN03**7.447D0)
B1 = ( 4.0D0*(F**(-0.17D0)) - 3.0D0 )
FKU6 = ( A1 + A2 + A3 ) *B1
DFKU6 = ( 1.582D0 *A1 + 3.843D0*A2 + 7.447D0*A3 )/XN03 *B1
C
A1 = 0.3429D0
A2 = 0.009552D0*(XN03**2.154D0)
IF ( IDIST(10U6 ) .EQ. IDAMIX ) THEN
    FTAU = DEXP(-1500.0D0*TAU)
ELSE
    FTAU = DEXP( -200.0D0*TAU)
END IF
FKEQ2 (10PU4 ) = FKU6*(A1+A2)*FTAU
DK2 (10PU4 ) = ( FKU6*2.154D0*A2/XN03+DFKU6*(A1+A2) )*FTAU
C
END IF
C
----- PU(111)-----
C
IF ( IDIST(10PU3 ) .EQ. IDCONS
* .OR. IDIST(10PU3 ) .EQ. IDEQU1 ) THEN
ELSE
C
...HANFORD...
A1 = 0.04000D0*(XN03**1.8D0)
A2 = 0.000156D0*(XN03**7.0D0)
FKEQ2 (10PU3 ) = A1 + A2
DK2 (10PU3 ) = ( 1.80D0*A1 + 7.0D0*A2 )/XN03
C
END IF
C
----- U(IV)-----
C
IF ( IDIST(10U4 ) .EQ. IDHANF ) THEN
C
...HANFORD...
DU4MAX = 600.0
FF = -3.336D0 + 1.9331D0*XN03
C
IF ( FF .LT. DLOG(DU4MAX) ) THEN
    FKEQ2 (10U4 ) = DEXP( -3.336D0 + 1.9331D0*XN03 )
    DK2 (10U4 ) = 1.9331D0 *FKEQ2(10U4 )
ELSE
    FKEQ2 (10U4 ) = DU4MAX
    DK2 (10U4 ) = 1.9331D0*DU4MAX
END IF
C
ELSE IF ( IDIST(10U4 ) .EQ. IDAMIX ) THEN
C
...KFK-3637...
DU4MAX = 600.0
A1 = 1.4870D0*XN03
B1 = 0.3556D0
C
IF ( FF .LT. DLOG(DU4MAX) ) THEN
    FKEQ2 (10U4 ) = DEXP( A1 )
    DK2 (10U4 ) = B1 *FKEQ2(10U4 )
ELSE
    FKEQ2 (10U4 ) = DU4MAX
    DK2 (10U4 ) = B1 *DU4MAX
END IF
C
ELSE
C
...KFK...
A1 = 5.28400D0*(XN03**1.582D0)
A2 = 1.55700D0*(XN03**3.843D0)
A3 = 0.01267D0*(XN03**7.447D0)
B1 = ( 4.0D0*(F**(-0.17D0)) - 3.0D0 )
FKU6 = ( A1 + A2 + A3 ) *B1
DFKU6 = ( 1.582D0 *A1 + 3.843D0*A2 + 7.447D0*A3 )/XN03 *B1
FTAU = 1.0D0
C
A1 = 0.03909D0
A2 = 0.01205D0*(XN03**(-0.5728D0))
IF ( FKU6*(A1+A2)*FTAU .LT. 600.0D0 ) THEN
    FKEQ2 (10U4 ) = FKU6*(A1+A2)*FTAU
    DK2 (10U4 ) = ( - FKU6*0.5728D0*A2/XN03 + DFKU6*(A1+A2)
* *FTAU
ELSE

```

```

FKEQ2 (10U4 ) = 600.0D0
DK2 (10U4 ) = 0.0D0
END IF
C
END IF
C
----- NP(V1)-----
C
IF ( IDIST(10NP6 ) .EQ. IDAMIX ) THEN
C
... A.MIXSET
C
IF ( XN03 .LT. 1.0D0 ) THEN
    FKEQ2(10NP6 ) = 4.67D0 *(XN03**1.433D0)
    DK2 (10NP6 ) = 4.67D0*1.433D0*(XN03**0.433D0)
ELSE
    FKEQ2(10NP6 ) = 5.00D0 *(XN03**3.055D0)
    DK2 (10NP6 ) = 5.00D0*3.055D0*(XN03**2.055D0)
END IF
C
END IF
C
----- TF**1 -----
C
----- NP(V)-----
C
IF ( IDIST(10NP5 ) .EQ. IDAMIX ) THEN
C
... A.MIXSET
C
FKEQ1 (10NP5 ) = 0.0234D0 *(XN03**1.14D0)
DK1 (10NP5 ) = 0.0234D0*1.14D0*(XN03**0.14D0)
C
END IF
C
----- TF**3 -----
C
DO 130 J = 1 , NCHEM
    IF ( IDIST(J) .EQ. IDCONS ) THEN
C
        FKEQ1 (J) = 0.0D0
        DK1 (J) = 0.0D0
C
        FKEQ2 (J) = 0.0D0
        DK2 (J) = 0.0D0
C
        FKEQ3 (J) = 0.0D0
        DK3 (J) = 0.0D0
C
        FKEQC (J) = 0.0D0
        DKC (J) = 0.0D0
C
        DISTC (J) = CDIS(J)
        DDCDTF (J) = 0.0D0
C
        DO 110 K = 1 , NCHEM
            DK1DX(J,K) = 0.0D0
            DK2DX(J,K) = 0.0D0
            DK3DX(J,K) = 0.0D0
            DKCDX(J,K) = 0.0D0
            DDCDX(J,K) = 0.0D0
110        CONTINUE
C
ELSE IF ( IDIST(J) .EQ. IDEQU1 ) THEN
C
        FKEQ1 (J) = 0.0D0
        DK1 (J) = 0.0D0
C
        FKEQ2 (J) = 0.0D0
        DK2 (J) = 0.0D0
C
        FKEQ3 (J) = 0.0D0
        DK3 (J) = 0.0D0
C
        FKEQC (J) = CK EQU(J)
        DKC (J) = 0.0D0
C
        DISTC (J) = 0.0D0
        DDCDTF (J) = 0.0D0
C

```

```

DO 120 K = 1 , NCHEM
  DK1DX(J,K) = 0.000
  DK2DX(J,K) = 0.000
  DK3DX(J,K) = 0.000
  DKCDX(J,K) = 0.000
  DDCCDX(J,K) = 0.000
120 CONTINUE
C
  END IF
C
130 CONTINUE
C
  CTBP = CO * F
C
  AA = 0.000
  A = 0.000
  B = 1.000
  C = CTBP
DO 210 J = 1 , NCHEMO
  AA = AA + 3.000*FKEQ3(J)*X(J)
  A = A + 2.000*FKEQ2(J)*X(J)
  B = B + FKEQ1(J)*X(J)
  C = C - CNTBP(J)*DISTC(J)*X(J)
C
C
  NTBP = INT( CNTBP(J) + 0.500 )
C
  IF ( NTBP .LE. 1 ) THEN
    B = B + CNTBP(J)*FKEQC(J)*X(J)
C
  ELSE IF ( NTBP .EQ. 2 ) THEN
    A = A + CNTBP(J)*FKEQC(J)*X(J)
C
  ELSE IF ( NTBP .GE. 3 ) THEN
    AA = AA + CNTBP(J)*FKEQC(J)*X(J)
C
  END IF
C
210 CONTINUE
C
  IF ( C .LE. 0.000 ) THEN
    TFN = 0.000
    TFO = 0.000
C
  ELSE
    DENOM1 = DSQRT( B**2 + 4.000*A*C )
    DENOM2 = B + DENOM1
    TFN = 2.000 * C / DENOM2
C
    DO 220 L = 1 , MXTFLP
      TFO = TFN
      FX = AA*TFO*TFO*TFO + A*TFO*TFO + B*TFO - C
      DFX = 3.000*AA*TFO*TFO + 2.000*A*TFO + B
      TFN = TFO - FX/DFX
C
      IF ( TFN .GE. TFMIN ) THEN
        DTFN = DABS( ( TFN - TFO ) / TFN )
      ELSE
        DTFN = DABS( ( TFN - TFO ) / TFMIN )
      END IF
C
      IF ( DTFN .LE. EPSTF ) GO TO 230
C
      TFO = TFN
C
220 CONTINUE
C
230 CONTINUE
C
  IF ( TFN .LT. 0.000 ) TFN = 0.000
C
  END IF

```

```

TF = TFN
DFX = 0.000
C
C
C
  IF ( TF .GT. 0.000 ) THEN
    TFO = TF
C
C
  AA = 0.000
  A = 0.000
  B = 1.000
  C = CTBP
DO 310 J = 1 , NCHEMO
  AA = AA + 3.000*FKEQ3(J)*X(J)
  A = A + 2.000*FKEQ2(J)*X(J)
  B = B + FKEQ1(J)*X(J)
  C = C - CNTBP(J)*DISTC(J)*X(J)
310 CONTINUE
C
  IF ( C .LE. 0.000 ) THEN
    TFO = 0.000
    DFX = 0.000
C
  ELSE
    TFN = TF
C
DO 700 L = 1 , MXTFLP
  TFO = TFN
C
  DU6 = FKEQ1(10U6 )*TFO
  + FKEQ2(10U6 )*TFO*TFO
  + FKEQ3(10U6 )*TFO*TFO*TFO
  + FKEQC(10U6 )*(TFO**CNTBP(10U6 ) )
  + DISTC(10U6 )
  DU6TF = FKEQ1(10U6 )
  + FKEQ2(10U6 )*2.000*TFO
  + FKEQ3(10U6 )*3.000*TFO*TFO
  + FKEQC(10U6 )*CNTBP(10U6 )
  *(TFO**CNTBP(10U6 )-1.000 )
  + DDCTF(10U6 )
C
  DU4 = FKEQ1(10U4 )*TFO
  + FKEQ2(10U4 )*TFO*TFO
  + FKEQ3(10U4 )*TFO*TFO*TFO
  + FKEQC(10U4 )*(TFO**CNTBP(10U4 ) )
  + DISTC(10U4 )
  DU4TF = FKEQ1(10U4 )
  + FKEQ2(10U4 )*2.000*TFO
  + FKEQ3(10U4 )*3.000*TFO*TFO
  + FKEQC(10U4 )*CNTBP(10U4 )
  *(TFO**CNTBP(10U4 )-1.000 )
  + DDCTF(10U4 )
C
  DPU4 = FKEQ1(10PU4 )*TFO
  + FKEQ2(10PU4 )*TFO*TFO
  + FKEQ3(10PU4 )*TFO*TFO*TFO
  + FKEQC(10PU4 )*(TFO**CNTBP(10PU4 ) )
  + DISTC(10PU4 )
  DPU4TF = FKEQ1(10PU4 )
  + FKEQ2(10PU4 )*2.000*TFO
  + FKEQ3(10PU4 )*3.000*TFO*TFO
  + FKEQC(10PU4 )*CNTBP(10PU4 )
  *(TFO**CNTBP(10PU4 )-1.000 )
  + DDCTF(10PU4 )
C
  DZR = FKEQ1(10ZR )*TFO
  + FKEQ2(10ZR )*TFO*TFO
  + FKEQ3(10ZR )*TFO*TFO*TFO
  + FKEQC(10ZR )*(TFO**CNTBP(10ZR ) )
  + DISTC(10ZR )
  DZRTF = FKEQ1(10ZR )
  + FKEQ2(10ZR )*2.000*TFO
  + FKEQ3(10ZR )*3.000*TFO*TFO
  + FKEQC(10ZR )*CNTBP(10ZR )
  *(TFO**CNTBP(10ZR )-1.000 )
  + DDCTF(10ZR )
C
----- HN02 -----
C
  IF ( IDIST(10HN02) .EQ. 1DCONS
  .OR. IDIST(10HN02) .EQ. 1DEQU1 ) THEN
C
  ELSE IF ( IDIST(10HN02) .EQ. 1DAMIX ) THEN

```

```

C
      XH = X(10HN03)
      CALL DBPOLA(XH ,NYYN02,
*          LENN02,
*          TABN02,YY ,DYY )
      CNTBP (10HN02) = 1.000
      DISTC (10HN02) = YY(1)
      DDCDTF(10HN02) = 0.000
      DDCDX (10HN02,10HN03) = DYY(1)

C
      ELSE

C
      ... JAERI-M 93-095 ...
      XHL = 1.00-6
      XH = X(10HN03)

C
      IF ( XH .LE. XHL ) THEN
          A1 = 24.8000*XHL**(-0.28000)
          A2 = 3.20900*XHL**(-0.65300)
          CNTBP (10HN02) = 1.000
          DISTC (10HN02) = ( A1 - A2 )*TFO
          DDCDTF(10HN02) = ( A1 - A2 )
          DDCDX (10HN02,10HN03) = 0.000
      ELSE
          A1 = 24.8000*XH**(-0.28000)
          A2 = 3.20900*XH**(-0.65300)
          CNTBP (10HN02) = 1.000
          DISTC (10HN02) = ( A1 - A2 )*TFO
          DDCDTF(10HN02) = ( A1 - A2 )
          DDCDX (10HN02,10HN03)
*          = ( -0.28000*A1 + 0.65300*A2 )/XH*TFO
      END IF

C
      END IF

C
      ----- PU(V1)-----
      IF ( IDIST(10PU6 ) .NE. IDCONS
*          .AND. IDIST(10PU6 ) .NE. IDEQUI ) THEN

C
          XH = X(10HN03)
          YU = DU6*X(10U6 ) + DU4*X(10U4 )
          YUL = 1.000-3
          CALL DBPOLA(XH ,NYYPU6,
*              LENPU6,
*              TABPU6,YY ,DYY )

C
          IF ( YU .LE. YUL ) THEN

C
              DPU6 = YY(1) + YY(2)*YUL**YY(3)
              DISTC (10PU6 ) = DPU6
              CNTBP (10PU6 ) = 2.000
              DDCDTF(10PU6 ) = 0.000
              DDCDX (10PU6 ,10HN03)
*                  = DYY(1) + DYY(2)*YUL**YY(3)
*                    + YY(2)*DLOG(YUL)*DYY(3)
*                      *YUL**YY(3)
          ELSE

C
              DPU6 = YY(1) + YY(2)*YU**YY(3)
              DISTC (10PU6 ) = DPU6
              CNTBP (10PU6 ) = 2.000
              DDCDTF(10PU6 ) = YY(2)*YY(3)
*                  *(YU**YY(3))/YU
*                  *( DU6TF*X(10U6 )
*                    + DU4TF*X(10U4 ) )

C
              DO 410 K = 1 , NCHEMO

C
                  DU6DX = CHR(K)*DK1(10U6)*TFO
*                  + CHR(K)*DK2(10U6)*TFO*TFO
*                  + CHR(K)*DK3(10U6)*TFO*TFO*TFO
*                  + CHR(K)*DKC(10U6)
*                    *(TFO**CNTBP(10U6))
*                  + DDCDX(10U6,K)

C
                  DU4DX = CHR(K)*DK1(10U4)*TFO
*                  + CHR(K)*DK2(10U4)*TFO*TFO
*                  + CHR(K)*DK3(10U4)*TFO*TFO*TFO
*                  + CHR(K)*DKC(10U4)
*                    *(TFO**CNTBP(10U4))
*                  + DDCDX(10U4,K)

C
                  DDCDX(10PU6 ,K) = YY(2)*YY(3)
*                  *(YU**YY(3))/YU
*                  *( DU6DX*X(10U6 )
*                    + DU4DX*X(10U4 ) )
          END IF
      END IF
    
```

```

410      CONTINUE
          DDCDX(10PU6 ,10U6 ) = DDCDX(10PU6 ,10U6 )
*              + YY(2)*YY(3)
*                *(YU**YY(3))/YU*DU6
          DDCDX(10PU6 ,10U4 ) = DDCDX(10PU6 ,10U4 )
*              + YY(2)*YY(3)
*                *(YU**YY(3))/YU*DU4
          DDCDX(10PU6 ,10HN03)
*          = DDCDX(10PU6 ,10HN03)
*            + DYY(1) + DYY(2)*YU**YY(3)
*              + YY(2)*DLOG(YU)*DYY(3)
*                *YU**YY(3)

C
          END IF

C
          END IF

C
          ----- NP(IV)-----
          IF ( IDIST(10NP4 ) .EQ. IDCONS
*              .OR. IDIST(10NP4 ) .EQ. IDEQUI ) THEN

C
          ELSE IF ( IDIST(10NP4 ) .EQ. IDAMIX ) THEN
              ...A.MIXSET...

C
              XU = X(10U6 )

C
              IF ( XU .LT. 0.03500 ) THEN
                  A1 = 0.7800
                  DA1 = 0.000
              ELSE
                  A1 = 0.43700 - 0.2400*DLOG( XU )
                  DA1 = - 0.2400/XU
              END IF

C
              IF ( XU .LT. 0.0500 ) THEN
                  A2 = -0.7600
                  DA2 = 0.000
              ELSE
                  A2 = -0.54800 + 0.17500*DLOG( XU )
                  DA2 = 0.17500/XU
              END IF

C
              DNP4 = DEXP( A1*XN03 + A2 )*TFO*TFO
              DISTC (10NP4 ) = DNP4
              CNTBP (10NP4 ) = 2.000
              DDCDTF(10NP4 ) = 2.000*DEXP( A1*XN03 + A2 )*TFO

C
              DO 412 K = 1 , NCHEM
                  DDCDX(10NP4 ,K) = CHR(K)*A1*DNP4
              CONTINUE
              DDCDX(10NP4 ,10U6 ) = DDCDX(10NP4 ,10U6 )
*                  + DNP4*( DA1*XN03 + DA2 )

C
          ELSE

C
              XH = X(10HN03)
              YU = DU6*X(10U6 ) + DU4*X(10U4 )
              CALL DBPOLA(XH ,NYYNP4,
*                  LENNP4,
*                  TABNP4,YY ,DYY )

C
              DNP4 = YY(1)*DEXP( - YY(2)*YU*YU + YY(3)*YU )
              DISTC (10NP4 ) = DNP4
              CNTBP (10NP4 ) = 2.000
              DDCDTF(10NP4 ) = DNP4*(-2.000*YY(2)*YU + YY(3) )
*                  *( DU6TF*X(10U6 )
*                    + DU4TF*X(10U4 ) )

C
              DO 420 K = 1 , NCHEMO

C
                  DU6DX = CHR(K)*DK1(10U6)*TFO
*                  + CHR(K)*DK2(10U6)*TFO*TFO
*                  + CHR(K)*DK3(10U6)*TFO*TFO*TFO
*                  + CHR(K)*DKC(10U6)
*                    *(TFO**CNTBP(10U6))
*                  + DDCDX(10U6,K)

C
                  DU4DX = CHR(K)*DK1(10U4)*TFO
*                  + CHR(K)*DK2(10U4)*TFO*TFO
*                  + CHR(K)*DK3(10U4)*TFO*TFO*TFO
*                  + CHR(K)*DKC(10U4)
*                    *(TFO**CNTBP(10U4))
*                  + DDCDX(10U4,K)

C
                  DDCDX(10NP4 ,K) = DNP4*(-2.000*YY(2)*YU + YY(3) )
*                  *( DU6DX*X(10U6 )
*                    + DU4DX*X(10U4 ) )
          END IF
      END IF
    
```

```

      DDCDX(10NP4 ,10U6 ) = DDCDX(10NP4 ,10U6 )
      + DNP4
      * ( - 2.000*YY(2)*YU+YY(3) )
      *DU6
      DDCDX(10NP4 ,10U4 ) = DDCDX(10NP4 ,10U4 )
      + DNP4
      * ( - 2.000*YY(2)*YU+YY(3) )
      *DU4
      DDCDX(10NP4 ,10HN03)
      = DDCDX(10NP4 ,10HN03)
      + DYY(1)*DEXP( - YY(2)*YU*YU + YY(3)*YU )
      + DNP4*( - DYY(2)*YU*YU + DYY(3)*YU )
C
      END IF
C
C
C
C
      NP(V)-----
C
      IF ( IDIST(10NP5 ) .EQ. IDCONS
      .OR. IDIST(10NP5 ) .EQ. IDEQUI
      .OR. IDIST(10NP5 ) .EQ. IDAMIX ) THEN
C
      ELSE
C
      XH = X(10HN03)
      XHL = 1.00-4
C
      IF ( XH .LE. XHL ) THEN
      DNP5 = DEXP( - 5.5301200
      *DEXP( - 3.2591900
      *DEXP(DLOG(XHL) - 2.3387400))
      DISTC (10NP5 ) = DNP5
      CNTBP (10NP5 ) = 1.000
      DDCDTF(10NP5 ) = 0.000
      DDCDX (10NP5 ,10HN03) = 0.000
      ELSE
      DNP5 = DEXP( - 5.5301200
      *DEXP( - 3.2591900
      *DEXP(DLOG(XH) - 2.3387400))
      DISTC (10NP5 ) = DNP5
      CNTBP (10NP5 ) = 1.000
      DDCDTF(10NP5 ) = 0.000
      DDCDX (10NP5 ,10HN03)
      = DNP5*(-5.5301200)
      *DEXP( - 3.2591900
      *DEXP(DLOG(XH) - 2.3387400)
      *(-3.2591900)
      *DEXP(DLOG(XH) - 2.3387400)/XH
      END IF
C
      END IF
C
C
C
C
      NP(VI)-----
C
      IF ( IDIST(10NP6 ) .EQ. IDCONS
      .OR. IDIST(10NP6 ) .EQ. IDEQUI
      .OR. IDIST(10NP6 ) .EQ. IDAMIX ) THEN
C
      ELSE IF ( IDIST(10NP6 ) .EQ. IDHANF ) THEN
C
      XH = X(10HN03)
      YU = DU6*X(10U6 ) + DU4*X(10U4 )
      CALL DBPOLA(XH ,NYYNP6,
      * LNNP6,
      * TABNP6,YY ,DYY )
C
      DNP6 = YY(1)*DEXP( - YY(2)*YU*YU + YY(3)*YU )
      DISTC (10NP6 ) = DNP6
      CNTBP (10NP6 ) = 2.000
      DDCDTF(10NP6 ) = DNP6*( -2.000*YY(2)*YU + YY(3) )
      * ( DU6TF*X(10U6 )
      + DU4TF*X(10U4 ) )
C
      DO 430 K = 1 , NCEM0
C
      DU6DX = CHRQ(K)*DK1(10U6)*TFO
      + CHRQ(K)*DK2(10U6)*TFO*TFO
      + CHRQ(K)*DK3(10U6)*TFO*TFO*TFO
      + CHRQ(K)*DKC(10U6)
      *(TFO**CNTBP(10U6))
      + DDCDX(10U6,K)
C
      DU4DX = CHRQ(K)*DK1(10U4)*TFO
      + CHRQ(K)*DK2(10U4)*TFO*TFO
      + CHRQ(K)*DK3(10U4)*TFO*TFO*TFO
      + CHRQ(K)*DKC(10U4)
      *(TFO**CNTBP(10U4))
      + DDCDX(10U4,K)

```

```

C
      DDCDX(10NP6 ,K) = DNP6*(-2.000*YY(2)*YU+YY(3)
      * ( DU6DX*X(10U6 )
      + DU4DX*X(10U4 ) ) )
430
      CONTINUE
      DDCDX(10NP6 ,10U6 )
      = DDCDX(10NP6 ,10U6 )
      + DNP6*( - 2.000*YY(2)*YU+YY(3) )*DU6
      DDCDX(10NP6 ,10U4 )
      = DDCDX(10NP6 ,10U4 )
      + DNP6*( - 2.000*YY(2)*YU+YY(3) )*DU4
      DDCDX(10NP6 ,10HN03)
      = DDCDX(10NP6 ,10HN03)
      + DYY(1)*DEXP( - YY(2)*YU*YU + YY(3)*YU )
      + DNP6*( - DYY(2)*YU*YU + DYY(3)*YU )
C
      ELSE
C
      XH = X(10HN03)
      XU6 = X(10U6 )
      CALL DBNP6(XH ,XU6 ,
      * YNPN6 ,DYH ,DYYU )
C
      DNP6 = YNPN6
      DISTC (10NP6 ) = DNP6
      CNTBP (10NP6 ) = 2.000
      DDCDTF(10NP6 ) = 0.000
C
      DDCDX(10NP6 ,10U6 ) = DYYU
      DDCDX(10NP6 ,10HN03) = DYYH
C
      END IF
C
      ZR(IV)-----
C
      IF ( IDIST(10ZR ) .NE. IDCONS
      .AND. IDIST(10ZR ) .NE. IDEQUI ) THEN
C
      XH = X(10HN03)
      AZR1 = 7.542707
      PF = CDBP
      /* ( SORT( 1.000+CDBP*AZR1*X(10ZR)*XN03*XN03
      + 1.000 )
      DPFZR = - PF
      /* ( SORT( 1.000+CDBP*AZR1*X(10ZR)*XN03*XN03
      + 1.000 )
      *0.500*CDBP*AZR1*XN03*XN03
      /*SORT( 1.000+CDBP*AZR1*X(10ZR)*XN03*XN03
      DPFDX = - PF
      /* ( SORT( 1.000+CDBP*AZR1*X(10ZR)*XN03*XN03
      + 1.000 )
      *0.500*CDBP*AZR1*X(10ZR)*2.000*XN03
      /*SORT( 1.000+CDBP*AZR1*X(10ZR)*XN03*XN03
C
      ST1 = X(10HN03) + 3.000*X(10U6 )
      STIMIN = 1.000
      STIMAX = 5.000
      IF ( ST1 .LE. STIMIN ) THEN
      DSOST1 = 0.500/SQRT(STIMIN)
      ST1 = STIMIN
      ELSE IF ( ST1 .GE. STIMAX ) THEN
      DSOST1 = 0.500/SQRT(STIMAX)
      ST1 = STIMAX
      ELSE
      DSOST1 = 0.500/SQRT(ST1)
      END IF
C
      AZR2 = DEXP( 11.15600 - 28.42400*SQRT(ST1)
      + 17.65500*ST1
      - 4.48300*ST1*SQRT(ST1)
      + 0.43300*ST1*ST1 )
      DAZR2H = AZR2*(
      + 28.42400*DSOST1
      + 17.65500
      - 4.48300*1.500*SQRT(ST1)
      + 0.43300*2.000*ST1 )
      DAZR2U = DAZR2H*3.000
C
      FFXH = 1.000 + 0.18900*XH*XN03
      + 4.050-4*XH*H*XN03*XN03
      TFP = TFO/FFXH
      DTFPH = - TFO/FFXH/FFXH
      * ( 0.18900 *XN03
      + 4.050-4*2.000*XH*XN03*XN03 )
      DTFPDX = - TFO/FFXH/FFXH
      * ( 0.18900*XH
      + 4.050-4*XH*H*2.000*XN03 )
C
      DZR = AZR1*XN03*XN03*PF*PF

```



```

      DDCDX(10TCU6,10U6 )
*      = DDCDX(10TCU6,10U6 )
*      + FKM*FKA/( FKA + XH ) *DU6/XN03
      DDCDX(10TCU6,10HN03)
*      = DDCDX(10TCU6,10HN03)
*      - DTCU6/( FKA + XH )
C
      END IF
C
C
      IF (      IDIST(10TCP4) .NE. IDCONS
*      .AND. IDIST(10TCP4) .NE. IDEOU1
*      .AND. IDIST(10TCP4) .NE. IDHANF ) THEN
C
      FKA = 2.63D0*DEXP( -4000.0D0
*      *( 1.0D0/T
*      - 1.0D0/(T0 + 25.0D0) ) )
      FKM = 160.0D0
      XH = X(10HN03)
      YPU4 = DPU4*X(10PU4 )
      DTCPU4 = FKM*FKA/( FKA + XH ) *YPU4/XN03
      DISTC (10TCP4) = DTCPU4
      CNTBP (10TCP4) = 2.0D0
      DDCDTF(10TCP4) = FKM*FKA/( FKA + XH )
*      *DPU4TF*X(10PU4 )/XN03
C
      DO 570 K = 1 , NCHEMO
C
      DPU4DX = CHR( K ) *DK1(10PU4) *TFO
*      + CHR( K ) *DK2(10PU4) *TFO *TFO
*      + CHR( K ) *DK3(10PU4) *TFO *TFO *TFO
*      + CHR( K ) *DKC(10PU4)
*      *(TFO**CNTBP(10PU4))
*      + DDCDX(10PU4, K)
C
      DDCDX(10TCP4, K) = FKM*FKA/( FKA + XH )
*      *DPU4DX*X(10PU4 )/XN03
*      - DTCPU4/XN03*CHR( K)
570      CONTINUE
      DDCDX(10TCP4,10PU4 )
*      = DDCDX(10TCP4,10PU4 )
*      + FKM*FKA/( FKA + XH ) *DPU4/XN03
      DDCDX(10TCP4,10HN03)
*      = DDCDX(10TCP4,10HN03)
*      - DTCPU4/( FKA + XH )
C
      END IF
C
C
      IF (      IDIST(10TCZR) .NE. IDCONS
*      .AND. IDIST(10TCZR) .NE. IDEOU1
*      .AND. IDIST(10TCZR) .NE. IDHANF ) THEN
C
      FKA = 2.63D0*DEXP( -4000.0D0
*      *( 1.0D0/T
*      - 1.0D0/(T0 + 25.0D0) ) )
      FKM = 2500.0D0
      XH = X(10HN03)
      YZR = DZR*X(10ZR )
      DTCZR = FKM*FKA/( FKA + XH ) *YZR/XN03
      DISTC (10TCZR) = DTCZR
      CNTBP (10TCZR) = 2.0D0
      DDCDTF(10TCZR) = FKM*FKA/( FKA + XH )
*      *DZRTF*X(10ZR )/XN03
C
      DO 580 K = 1 , NCHEMO
C
      DZFDX = CHR( K ) *DK1(10ZR) *TFO
*      + CHR( K ) *DK2(10ZR) *TFO *TFO
*      + CHR( K ) *DK3(10ZR) *TFO *TFO *TFO
*      + CHR( K ) *DKC(10ZR)
*      *(TFO**CNTBP(10ZR))
*      + DDCDX(10ZR, K)
C
      DDCDX(10TCZR, K) = FKM*FKA/( FKA + XH )
*      *DZFDX*X(10ZR )/XN03
*      - DTCZR/XN03*CHR( K)
580      CONTINUE
      DDCDX(10TCZR,10ZR )
*      = DDCDX(10TCZR,10ZR )
*      + FKM*FKA/( FKA + XH ) *DZR/XN03
      DDCDX(10TCZR,10HN03)
*      = DDCDX(10TCZR,10HN03)
*      - DTCZR/( FKA + XH )
C
      END IF
C
      FX = AA*TFO*TFO*TFO + A*TFO*TFO + B*TFO - CTBP

```



```

        DDDX(J,K) = DDCDX(J,K)
1200    CONTINUE
1300    CONTINUE
C
C
C    ELSE
C
        DO 2100 J = 1 , NCHEM
            DIS(J) = FKE01(J)*TF
            *           + FKE02(J)*TF*TF
            *           + FKE03(J)*TF*TF*TF
            *           + FKE0C(J)*(TF**CNTBP(J) )
            *           + DISTC(J)
2100    CONTINUE
C
C
        DO 2300 J = 1 , NCHEM
            DO 2200 K = 1 , NCHEM
                DK1DX(J,K) = CHRQ(K)*DK1(J)
                DK2DX(J,K) = CHRQ(K)*DK2(J)
                DK3DX(J,K) = CHRQ(K)*DK3(J)
                DKCDX(J,K) = CHRQ(K)*DKC(J)
2200    CONTINUE
2300    CONTINUE
C
C
        DO 2500 J = 1 , NCHEM
            DADX(J) = 2.000*FKE02(J)
            DBDX(J) = FKE01(J)
            DO 2400 K = 1 , NCHEM
                DADX(J) = DADX(J) + 2.000*DK2DX(K,J)*X(K)
                DBDX(J) = DBDX(J) + DK1DX(K,J)*X(K)
2400    CONTINUE
2500    CONTINUE
C
C
C
C
        DO 2620 K = 1 , NCHEM
            IF ( K .LE. NCHEM0 ) THEN
                DAA = 3.000*FKE03(K)
                DA = 2.000*FKE02(K)
                DB = FKE01(K)
                DC = CNTBP(K)*DISTC(K)
            *           + CNTBP(K)*FKE0C(K)*(TF**CNTBP(K) )
            ELSE
                DAA = 0.000
                DA = 0.000
                DB = 0.000
                DC = 0.000
            END IF
C
        DO 2610 J = 1 , NCHEM0
            DAA = DAA + 3.000*DK3DX(J,K)*X(J)
            DA = DA + 2.000*DK2DX(J,K)*X(J)
            DB = DB + DK1DX(J,K)*X(J)
            DC = DC + CNTBP(J)*DDCDX(J,K)*X(J)
            *           + CNTBP(J)*DKCDX(J,K)*(TF**CNTBP(J) ) *X(J)
2610    CONTINUE
C
        DFDX = DAA*TF*TF*TF + DA*TF*TF + DB*TF + DC
C
C
        XTC7 = X(10TC7 )
C
        J = 10TCU6
        DFDX = DFDX + CNTBP(J)*DDCDX(J,K)*XTC7
            *           + CNTBP(J)*DKCDX(J,K)*(TF**CNTBP(J) ) *XTC7
        IF ( K .EQ. 10TC7 ) THEN
            DFDX = DFDX + CNTBP(J)*DISTC(J)
            *           + CNTBP(J)*FKE0C(J)*(TF**CNTBP(J) )
        END IF
C
        J = 10TCP4
        DFDX = DFDX + CNTBP(J)*DDCDX(J,K)*XTC7
            *           + CNTBP(J)*DKCDX(J,K)*(TF**CNTBP(J) ) *XTC7
        IF ( K .EQ. 10TC7 ) THEN
            DFDX = DFDX + CNTBP(J)*DISTC(J)
            *           + CNTBP(J)*FKE0C(J)*(TF**CNTBP(J) )
        END IF
C
        J = 10TCZR
        DFDX = DFDX + CNTBP(J)*DDCDX(J,K)*XTC7
            *           + CNTBP(J)*DKCDX(J,K)*(TF**CNTBP(J) ) *XTC7
        IF ( K .EQ. 10TC7 ) THEN
            DFDX = DFDX + CNTBP(J)*DISTC(J)
            *           + CNTBP(J)*FKE0C(J)*(TF**CNTBP(J) )
        END IF
C
C
C

```

```

C
        IF ( DFDTF .NE. 0.000 ) THEN
            DTFDX(K) = - DFDX/DFDTF
        ELSE
            DTFDX(K) = 0.000
        END IF
C
C
C
2620    CONTINUE
C
C
        DO 2800 J = 1 , NCHEM
            DDTF = FKE01(J)
            *           + 2.000 *FKE02(J)*TF
            *           + 3.000 *FKE03(J)*TF*TF
            *           + CNTBP(J)*FKE0C(J)*( TF**CNTBP(J) - 1.000 )
            *           + DDCDTF(J)
C
            DO 2700 K = 1 , NCHEM
                DDX = DK1DX(J,K)*TF
                *           + DK2DX(J,K)*TF*TF
                *           + DK3DX(J,K)*TF*TF*TF
                *           + DKCDX(J,K)*( TF**CNTBP(J) )
                *           + DDCDX(J,K)
C
                DDDX(J,K) = DDX + DDTF*DTFDX(K)
2700    CONTINUE
2800    CONTINUE
            END IF
C
C
C
        RETURN
C
C
C
C-----
C
6000    CONTINUE
C
C    MICRO COMPONENTS
C
        TF = XTF
C
        DO 6110 J = 1 , NCOMP
            FKE01(J) = 0.000
            FKE02(J) = 0.000
            FKE03(J) = 0.000
            FKE0C(J) = 0.000
C
        6110    CONTINUE
C
        DO 6120 J = 1 , NCOMP
            DISTC(J) = 0.000
        6120    CONTINUE
C
        DO 6140 K = 1 , NCOMP
            DO 6130 J = 1 , NCOMP
                DK1DX(J,K) = 0.000
                DK2DX(J,K) = 0.000
                DK3DX(J,K) = 0.000
                DKCDX(J,K) = 0.000
                DDCDX(J,K) = 0.000
            6130    CONTINUE
        6140    CONTINUE
C
        XN03 = 0.000
C
        DO 6150 J = 1 , NCHEM0
            XN03 = XN03 + CHRQ(J)*X(J)
        6150    CONTINUE
C
C
C
        DO 8000 J = 1 , NCOMP
            JJ = J + JBASE
            IF ( IDIST(JJ) .NE. 1DCONS
            *           .AND. IDIST(JJ) .NE. 1DEQUI ) THEN
C
                SR(2+)
C
                SR(N03)2 + 2TBP = SR(N03)2*2TBP
C
                IF ( JJ .EQ. 10SR ) THEN

```

```

      FKEQ2(J) = DEXP (  1.0226D0*XN03
*                   -  9.052D0
*                   +  2.3303D0/XN03
*                   -  1.2659D0/(XN03*XN03) )
      CNTBP(10SR ) = 2.0D0
      END IF
C-----
C      FU(DI)
C-----
      RUNO(N03)2 + 2TBP = RUNO(N03)2*2TBP
C
      IF ( JJ .EQ. 10RUDI ) THEN
      IF ( XN03 .LE. 2.1D0 ) THEN
*         FKEQ2(J) = DEXP (  0.3044D0*XN03*XN03*XN03
*                   -  1.4419D0*XN03*XN03
*                   +  1.8536D0*XN03
*                   -  1.7247D0 )
      ELSE
      FKEQ2(J) = DEXP ( - 0.1549D0*XN03 - 1.0514D0 )
      END IF
      CNTBP(10RUDI) = 2.0D0
      END IF
C-----
C      FU(TR)
C-----
      RUNO(N03)3 + 2TBP = RUNO(N03)3*2TBP
C
      IF ( JJ .EQ. 10RUTR ) THEN
      IF ( XN03 .LE. 0.4D0 ) THEN
*         FKEQ2(J) = DEXP ( - 13.911D0*XN03*XN03
*                   +  8.524D0*XN03
*                   +  3.447D0 )
      ELSE
*         FKEQ2(J) = DEXP (  0.1152D0*XN03*XN03
*                   -  1.0020D0*XN03
*                   +  5.0150D0 )
      END IF
      CNTBP(10RUTR) = 2.0D0
      END IF
C-----
C      FU(NI)
C-----
      RUNO(N02)2 + TBP = RUNO(N02)2*TBP
C
      IF ( JJ .EQ. 10RUNI ) THEN
*         FKEQ2(J) = DEXP (  0.3707D0*XN03
*                   +  0.6187D0
*                   -  0.02096D0/XN03 )
      CNTBP(10RUNI) = 2.0D0
      END IF
C-----
C      FU(MO)
C-----
      RUNO(N02)
C
      IF ( JJ .EQ. 10RUMO ) THEN
      DISTC(J) = 0.0D0
      CNTBP(10RUMO) = 0.0D0
      END IF
C-----
C      CS(1+)
C-----
      CS(N03) + TBP = CS(N03)*TBP
C
      IF ( JJ .EQ. 10CS ) THEN
*         FKEQ1(J) = DEXP (  0.33114D0*XN03
*                   -  7.1608D0
*                   -  0.444D0/XN03 )
      CNTBP(10CS ) = 1.0D0
      END IF
C-----
C      CE(3+)
C-----
      CE(N03)3 + 3TBP = CE(N03)3*3TBP
C
      IF ( JJ .EQ. 10CE ) THEN
*         FKEQ3(J) = DEXP (  1.3771D0*XN03
*                   -  4.3411D0
*                   -  0.41314D0/XN03 )
      CNTBP(10CE ) = 3.0D0
      END IF
C-----
C      GD(3+)
C-----
      GD(N03)3 + 3TBP = GD(N03)3*3TBP
C
      IF ( JJ .EQ. 10GD ) THEN
      YU = DIS(10U6 ) * X(10U6 ) + DIS(10U4 ) * X(10U4 )

```

```

C
      A11 = 1.2D0 + 0.238D0*YU
      A12 = - X(10HN03)
*         / ( -54.71D0*(YU+0.1681D0)**2.467D0 + 21.0D0 )
      A13 = - X(10HN03)
      A1 = A11*( DEXP(A12) - DEXP(A13) )
      A4 = - 4.0D0 - 2.38*YU
C-----
C      DISTC(J) = DEXP( 2.303D0*( DEXP(A1) + A4 ) )
      CNTBP(10GD ) = 3.0D0
      END IF
C-----
C      AM(3+)
C-----
      AM(N03)3 + 3TBP = AM(N03)3*3TBP
C
      IF ( JJ .EQ. 10AM ) THEN
      IF ( XN03 .LT. 2.0D0 ) THEN
      FKEQ3(J) = 0.0174D0 *XN03**2.37D0
      ELSE
      FKEQ3(J) = 0.00151D0*XN03**5.69D0
      END IF
      CNTBP(10AM ) = 3.0D0
      END IF
C-----
C      CM(3+)
C-----
      CM(N03)3 + 3TBP = CM(N03)3*3TBP
C
      IF ( JJ .EQ. 10CM ) THEN
      IF ( XN03 .LT. 2.0D0 ) THEN
      FKEQ3(J) = 0.0115D0 *XN03**2.31D0
      ELSE
      FKEQ3(J) = 0.000617D0*XN03**6.45D0
      END IF
      CNTBP(10CM ) = 3.0D0
      END IF
C-----
      END IF
C
      8000 CONTINUE
C
      DO 8130 J = 1 , NCOMP
      JJ = J + JBASE
      IF ( IDIST(JJ) .EQ. IDCONS ) THEN
      FKEQ1 (J) = 0.0D0
      FKEQ2 (J) = 0.0D0
      FKEQ3 (J) = 0.0D0
      FKEQC (J) = 0.0D0
      DISTC (J) = CDIS(JJ)
C
      DO 8110 K = 1 , NCOMP
      DK1DX(J,K) = 0.0D0
      DK2DX(J,K) = 0.0D0
      DK3DX(J,K) = 0.0D0
      DKCDX(J,K) = 0.0D0
      DDCDX(J,K) = 0.0D0
      8110 CONTINUE
      ELSE IF ( IDIST(JJ) .EQ. IDEOU1 ) THEN
      FKEQ1 (J) = 0.0D0
      FKEQ2 (J) = 0.0D0
      FKEQ3 (J) = 0.0D0
      FKEQC (J) = CKEQU(JJ)
      DISTC (J) = 0.0D0
C
      DO 8120 K = 1 , NCOMP
      DK1DX(J,K) = 0.0D0
      DK2DX(J,K) = 0.0D0
      DK3DX(J,K) = 0.0D0
      DKCDX(J,K) = 0.0D0
      DDCDX(J,K) = 0.0D0
      8120 CONTINUE
      END IF
      8130 CONTINUE
C

```

```

      IF ( TF .LE. 0.000 ) THEN
C
      DO 9100 J = 1 , NCOMP
        JJ = J + JBASE
        DIS(JJ) = DISTC(J)
9100 CONTINUE
C
      DO 9300 J = 1 , NCOMP
        DO 9200 K = 1 , NCOMP
          DDDX(J,K) = DDCDX(J,K)
9200 CONTINUE
9300 CONTINUE
C
      ELSE
C
      DO 9400 J = 1 , NCOMP
        JJ = J + JBASE
        DIS(JJ) = FKE01(J)*TF
          + FKE02(J)*TF*TF
          + FKE03(J)*TF*TF*TF
          + FKE0C(J)*(TF**CNTBP(JJ) )
          + DISTC(J)
9400 CONTINUE
C
      DO 9600 J = 1 , NCOMP
        DO 9500 K = 1 , NCOMP
          DDDX(J,K) = DK1DX(J,K)*TF
            + DK2DX(J,K)*TF*TF
            + DK3DX(J,K)*TF*TF*TF
            + DKCDX(J,K)*( TF**CNTBP(JJ) )
            + DDCDX(J,K)
9500 CONTINUE
9600 CONTINUE
C
      END IF
C
      RETURN
C
      END

```

```

SUBROUTINE DBNP6(XX ,XU6 ,
*               YYNP6 ,DYH ,DYYU )
C
C   XX           : HNO3 CONCENTRATION IN AQUEOUS PHASE
C   XU6          : URANIUM CONCENTRATION IN AQUEOUS PHASE
C   YYNP6        : CALCULATED DISTRIBUTION COEFFICIENT OF NP(VI)
C   DYH          : DERIVATIVE OF DISTRIBUTION COEFFICIENT BY HNO3
C   DYYU         : DERIVATIVE OF DISTRIBUTION COEFFICIENT BY XU6
C
      IMPLICIT REAL*8 ( A-H , O-Z )
C
      UMASS = 238.000
      XU6L = 1.00-3
C
      XU6GL = XU6L*UMASS
      XU6G = XU6 *UMASS
C
      IF ( XX .LT. 0.2058D0 ) THEN
C
      X1 = 0.000
      X2 = 0.2058D0
C
      IF ( XU6G .LE. XU6GL ) THEN
        F1 = - 4.08295D-8 *XU6GL*XU6GL*XU6GL*XU6GL
          + 9.75667D-6 *XU6GL*XU6GL*XU6GL
          + 8.12588D-4 *XU6GL*XU6GL
          + 0.0239163D0*XU6GL
          + 0.577235D0
        DF1U = 0.000
C
      ELSE
C
      IF ( XU6G .LT. 70.000 ) THEN
        F1 = - 4.08295D-8 *XU6G*XU6G*XU6G*XU6G
          + 9.75667D-6 *XU6G*XU6G*XU6G
          + 8.12588D-4 *XU6G*XU6G
          + 0.0239163D0*XU6G
          + 0.577235D0
        DF1U = - 4.08295D-8*4.000*XU6G*XU6G*XU6G
          + 9.75667D-6*3.000*XU6G*XU6G
          + 8.12588D-4*2.000*XU6G
          + 0.0239163D0
C
      ELSE
        F1 = - 1.10758D1 + 1.31967D1*(XU6G**(-2.812D-2))
        DF1U = 1.31967D1*(-2.812D-2)*(XU6G**(-2.812D-2))
          /XU6G
C
      END IF
C
      END IF
C
      F2 = F1
      DF2U = F2
C
      ELSE IF ( XX .LE. 0.5291D0 ) THEN
C
      X1 = 0.2058D0
      X2 = 0.5291D0
C
      IF ( XU6G .LE. XU6GL ) THEN
        F1 = - 4.08295D-8 *XU6GL*XU6GL*XU6GL*XU6GL
          + 9.75667D-6 *XU6GL*XU6GL*XU6GL
          + 8.12588D-4 *XU6GL*XU6GL
          + 0.0239163D0*XU6GL
          + 0.577235D0
        DF1U = 0.000
C
        F2 = 1.98507D0 - 0.082342D0*(XU6GL**0.749606D0)
        DF2U = 0.000
C
      ELSE
C
      IF ( XU6G .LT. 70.000 ) THEN
        F1 = - 4.08295D-8 *XU6G*XU6G*XU6G*XU6G
          + 9.75667D-6 *XU6G*XU6G*XU6G
          + 8.12588D-4 *XU6G*XU6G
          + 0.0239163D0*XU6G
          + 0.577235D0
        DF1U = - 4.08295D-8*4.000*XU6G*XU6G*XU6G
          + 9.75667D-6*3.000*XU6G*XU6G
          + 8.12588D-4*2.000*XU6G
          + 0.0239163D0
C
      ELSE
        F1 = - 1.10758D1 + 1.31967D1*(XU6G**(-2.812D-2))
        DF1U = 1.31967D1*(-2.812D-2)*(XU6G**(-2.812D-2))
          /XU6G
C

```

```

C      END IF
C      IF ( XU6G .LT. 15.000 ) THEN
C          F2 = 1.98507D0 - 0.082342D0*(XU6G**0.749606D0)
C          DF2U = - 0.082342D0*(0.749606D0)*(XU6G**0.749606D0)
C          *
C          /XU6G
C      ELSE
C          F2 = - 4.98318D0 + 7.85637D0*(XU6G**(-7.6909D-2))
C          DF2U = 7.85637D0*(-7.6909D-2)*(XU6G**(-7.6909D-2))
C          *
C          /XU6G
C      END IF
C      END IF
C      ELSE IF ( XX .LE. 1.0000 ) THEN
C      C
C      X1 = 0.5291D0
C      X2 = 1.000D0
C      IF ( XU6G .LE. XU6GL ) THEN
C      C
C      F1 = 1.98507D0 - 0.082342D0*(XU6G**0.749606D0)
C      DF1U = 0.0D0
C      C
C      F2 = - 59.0245D0 + 63.1353D0*(XU6GL**(-0.012409D0))
C      DF2U = 0.0D0
C      ELSE
C      C
C      IF ( XU6G .LT. 15.000 ) THEN
C          F1 = 1.98507D0 - 0.082342D0*(XU6G**0.749606D0)
C          DF1U = - 0.082342D0*(0.749606D0)*(XU6G**0.749606D0)
C          *
C          /XU6G
C      ELSE
C          F1 = - 4.98318D0 + 7.85637D0*(XU6G**(-7.6909D-2))
C          DF1U = 7.85637D0*(-7.6909D-2)*(XU6G**(-7.6909D-2))
C          *
C          /XU6G
C      END IF
C      END IF
C      IF ( XU6G .LT. 110.000 ) THEN
C          F2 = - 59.0245D0 + 63.1353D0*(XU6G**(-0.012409D0))
C          DF2U = 63.1353D0*(-0.012409D0)*(XU6G**(-0.012409D0))
C          *
C          /XU6G
C      ELSE
C          F2 = 5.62821D0 - 3.09702D0*(XU6G**(1.05751D-1))
C          DF2U = - 3.09702D0*(1.05751D-1)*(XU6G**(1.05751D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      ELSE IF ( XX .LE. 1.67D0 ) THEN
C      C
C      X1 = 1.000D0
C      X2 = 1.67D0
C      IF ( XU6G .LE. XU6GL ) THEN
C      C
C      F1 = - 59.0245D0 + 63.1353D0*(XU6GL**(-0.012409D0))
C      DF1U = 0.0D0
C      C
C      F2 = - 1.3877D2 + 1.44908D2*(XU6GL**(-8.54918D-3))
C      DF2U = 0.0D0
C      ELSE
C      C
C      IF ( XU6G .LT. 110.000 ) THEN
C          F1 = - 59.0245D0 + 63.1353D0*(XU6G**(-0.012409D0))
C          DF1U = 63.1353D0*(-0.012409D0)*(XU6G**(-0.012409D0))
C          *
C          /XU6G
C      ELSE
C          F1 = 5.62821D0 - 3.09702D0*(XU6G**(1.05751D-1))
C          DF1U = - 3.09702D0*(1.05751D-1)*(XU6G**(1.05751D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      IF ( XU6G .LT. 40.000 ) THEN
C          F2 = - 1.3877D2 + 1.44908D2*(XU6G**(-8.54918D-3))
C          DF2U = 1.44908D2*(-8.54918D-3)*(XU6G**(-8.54918D-3))
C          *
C          /XU6G
C      ELSE
C          F2 = - 9.06522D-1 + 8.49352D0*(XU6G**(-3.78244D-1))
C          DF2U = 8.49352D0*(-3.78244D-1)*(XU6G**(-3.78244D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      END IF

```

```

C      ELSE IF ( XX .LE. 2.07D0 ) THEN
C      C
C      X1 = 1.67D0
C      X2 = 2.07D0
C      IF ( XU6G .LE. XU6GL ) THEN
C      C
C      F1 = - 1.3877D2 + 1.44908D2*(XU6GL**(-8.54918D-3))
C      DF1U = 0.0D0
C      C
C      F2 = - 7.32508D1 + 8.05908D1*(XU6GL**(-2.13834D-2))
C      DF2U = 0.0D0
C      ELSE
C      C
C      IF ( XU6G .LT. 40.000 ) THEN
C          F1 = - 1.3877D2 + 1.44908D2*(XU6G**(-8.54918D-3))
C          DF1U = 1.44908D2*(-8.54918D-3)*(XU6G**(-8.54918D-3))
C          *
C          /XU6G
C      ELSE
C          F1 = - 9.06522D-1 + 8.49352D0*(XU6G**(-3.78244D-1))
C          DF1U = 8.49352D0*(-3.78244D-1)*(XU6G**(-3.78244D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      IF ( XU6G .LT. 40.000 ) THEN
C          F2 = - 7.32508D1 + 8.05908D1*(XU6G**(-2.13834D-2))
C          DF2U = 8.05908D1*(-2.13834D-2)*(XU6G**(-2.13834D-2))
C          *
C          /XU6G
C      ELSE
C          F2 = - 7.83706D-1 + 8.21658D0*(XU6G**(-3.98856D-1))
C          DF2U = 8.21658D0*(-3.98856D-1)*(XU6G**(-3.98856D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      ELSE IF ( XX .LE. 3.0000 ) THEN
C      C
C      X1 = 2.07D0
C      X2 = 3.000D0
C      IF ( XU6G .LE. XU6GL ) THEN
C      C
C      F1 = - 7.32508D1 + 8.05908D1*(XU6GL**(-2.13834D-2))
C      DF1U = 0.0D0
C      C
C      F2 = - 3.14568D2 + 3.24482D2*(XU6GL**(-7.67715D-3))
C      DF2U = 0.0D0
C      ELSE
C      C
C      IF ( XU6G .LT. 40.000 ) THEN
C          F1 = - 7.32508D1 + 8.05908D1*(XU6G**(-2.13834D-2))
C          DF1U = 8.05908D1*(-2.13834D-2)*(XU6G**(-2.13834D-2))
C          *
C          /XU6G
C      ELSE
C          F1 = - 7.83706D-1 + 8.21658D0*(XU6G**(-3.98856D-1))
C          DF1U = 8.21658D0*(-3.98856D-1)*(XU6G**(-3.98856D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      IF ( XU6G .LT. 40.000 ) THEN
C          F2 = - 3.14568D2 + 3.24482D2*(XU6G**(-7.67715D-3))
C          DF2U = 3.24482D2*(-7.67715D-3)*(XU6G**(-7.67715D-3))
C          *
C          /XU6G
C      ELSE
C          F2 = - 4.7101D0 + 9.63305D0*(XU6G**(-1.35568D-1))
C          DF2U = 9.63305D0*(-1.35568D-1)*(XU6G**(-1.35568D-1))
C          *
C          /XU6G
C      END IF
C      END IF
C      ELSE IF ( XX .LE. 3.47D0 ) THEN
C      C
C      X1 = 3.000D0
C      X2 = 3.47D0
C      IF ( XU6G .LE. XU6GL ) THEN
C      C
C      F1 = - 3.14568D2 + 3.24482D2*(XU6GL**(-7.67715D-3))
C      DF1U = 0.0D0
C      C
C      F2 = - 3.47764D0 + 1.60775D1*(XU6GL**(-3.34389D-1))
C      DF2U = 0.0D0

```

```

C
C     ELSE
C
C       IF ( XU6G .LT. 40.000 ) THEN
C         F1 = - 3.14568D2 + 3.24482D2*(XU6G**(-7.67715D-3))
C         DF1U = 3.24482D2*(-7.67715D-3)*(XU6G**(-7.67715D-3))
C         *
C         ELSE
C         F1 = - 4.7101D0 + 9.63305D0*(XU6G**(-1.35568D-1))
C         DF1U = 9.63305D0*(-1.35568D-1)*(XU6G**(-1.35568D-1))
C         *
C         END IF
C
C       IF ( XU6G .LT. 40.000 ) THEN
C         F2 = - 3.47764D0 + 1.60775D1*(XU6G**(-3.34389D-1))
C         DF2U = 1.60775D1*(-3.34389D-1)*(XU6G**(-3.34389D-1))
C         *
C         ELSE
C         F2 = 5.7535D0 - 2.9546D0*(XU6G**1.22887D-1)
C         DF2U = - 2.9546D0*(1.22887D-1)*(XU6G**1.22887D-1)
C         *
C         END IF
C
C     END IF
C
C     ELSE IF ( XX .LE. 4.62D0 ) THEN
C
C       X1 = 3.47D0
C       X2 = 4.62D0
C
C       IF ( XU6G .LE. XU6GL ) THEN
C
C         F1 = - 3.47764D0 + 1.60775D1*(XU6GL**(-3.34389D-1))
C         DF1U = 0.0D0
C
C         F2 = 0.46222D0 + 39.0202D0*(XU6GL**(-1.0898D0))
C         DF2U = 0.0D0
C
C       ELSE
C
C         IF ( XU6G .LT. 40.000 ) THEN
C           F1 = - 3.47764D0 + 1.60775D1*(XU6G**(-3.34389D-1))
C           DF1U = 1.60775D1*(-3.34389D-1)*(XU6G**(-3.34389D-1))
C           *
C           ELSE
C           F1 = 5.7535D0 - 2.9546D0*(XU6G**1.22887D-1)
C           DF1U = - 2.9546D0*(1.22887D-1)*(XU6G**1.22887D-1)
C           *
C           END IF
C
C         F2 = 0.46222D0 + 39.0202D0*(XU6G**(-1.0898D0))
C         DF2U = 39.0202D0*(-1.0898D0)*(XU6G**(-1.0898D0))
C         *
C         END IF
C
C       ELSE
C
C         X1 = 4.62D0
C         X2 = 10.0D0
C
C         IF ( XU6G .LE. XU6GL ) THEN
C
C           F1 = 0.46222D0 + 39.0202D0*(XU6GL**(-1.0898D0))
C           DF1U = 0.0D0
C
C         ELSE
C
C           F1 = 0.46222D0 + 39.0202D0*(XU6G**(-1.0898D0))
C           DF1U = 39.0202D0*(-1.0898D0)*(XU6G**(-1.0898D0))
C           *
C           END IF
C
C         F2 = F1
C         DF2U = DF1U
C
C       END IF
C
C     YYNP6 = F1 * ( X2 - XX ) / ( X2 - X1 )
C     *
C     + F2 * ( XX - X1 ) / ( X2 - X1 )
C
C     DYYH = ( F2 - F1 ) / ( X2 - X1 )
C     DYYU = ( DF1U * ( X2 - XX ) / ( X2 - X1 )
C     *
C     + DF2U * ( XX - X1 ) / ( X2 - X1 ) ) * UMASS
C
C     RETURN
C     END

```

```

SUBROUTINE DBPOLA(XX , NYY ,
*          LENGTH,
*          TAB , YY , DYY )
C
C     LINEARLY INTERPOLATE TABLE TAB.
C     TAB(I,1) : INDEPENDENT VARIABLES
C     TAB(J,1) (J>1) : DEPENDENT VARIABLES
C     XX : THE POINT OF INTERPLATION
C     YY : INTERPOLATED VALUES
C     DYY : DERIVATIVE AT INTERPOLATED VALUES
C     NYY : NUMBER OF DEPENDENT VARIABLES
C     LENGTH : TABLE LENGTH
C
C     IMPLICIT REAL*8 ( A-H , O-Z )
C
C     DIMENSION TAB(NYY+1,LENGTH) , YY(NYY) , DYY(NYY)
C
C     IF ( LENGTH .LE. 1 ) THEN
C
C       DO 10 I = 1 , NYY
C         YY (I) = TAB(I+1,1)
C         DYY(I) = 0.0D0
C       10 CONTINUE
C       RETURN
C     END IF
C
C     IP = ( LENGTH - 1 ) / 2 + 1
C
C     IF ( XX .GE. TAB(1,1) .AND. XX .LE. TAB(1,LENGTH) ) THEN
C
C       20 CONTINUE
C
C       IF ( XX .LT. TAB(1,IP) ) THEN
C         IP = IP - 1
C         GO TO 20
C       ELSE IF( XX .GT. TAB(1,IP+1) ) THEN
C         IP = IP + 1
C         GO TO 20
C       ELSE
C
C         DENOM = TAB(1,IP+1) - TAB(1,IP)
C         F1 = ( TAB(1,IP+1) - XX ) / DENOM
C         F2 = ( XX - TAB(1,IP) ) / DENOM
C         DO 30 I = 1 , NYY
C           YY (I) = F1 * TAB(I+1,IP) + F2 * TAB(I+1,IP+1)
C           DYY(I) = ( TAB(I+1,IP+1) - TAB(I+1,IP) ) / DENOM
C         30 CONTINUE
C       END IF
C
C     ELSE IF( XX .LT. TAB(1,1) ) THEN
C
C       DO 100 I = 1 , NYY
C         YY (I) = TAB(I+1,1)
C         DYY(I) = 0.0D0
C       100 CONTINUE
C
C     ELSE
C
C       DO 200 I = 1 , NYY
C         YY (I) = TAB(I+1,LENGTH)
C         DYY(I) = 0.0D0
C       200 CONTINUE
C
C     END IF
C
C     RETURN
C     END

```

```

SUBROUTINE DENFC( MO ,NO ,
*             ISFIN ,FRC ,
*             CCO ,TTO ,
*             TMPO ,WLTO ,
*             WSTO ,HOSO )
C
C*****
C DENFC PROVIDES THE DENSITY FACTOR FROM MOLARITY TO SOLUTE-FREE
C UNIT OR FROM SOLUTE-FREE TO MOLARITY
C X(1) : H , X(2) : U6 , X(3) : PU4 , X(4) : PU3 ,
C X(5) : U4 , X(6) : HAN , X(7) : HYD , X(8) : NO2
C
C*****
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C DIMENSION TMPO(MO) ,WLTO(MO) ,WSTO(MO) ,HOSO(MO)
C
C COMMON /DEN000/ CO ,TO ,
*             CTBP,F65,F69,UOS,POS
C COMMON /DEN001/ ISF,ISFREE
C
C COMMON /IXCOMP/NCHEMO,
*             IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*             IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*             IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*             IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
*             IOTCZR,IOSR ,IORUDI,IORUTR,IORUNI,
*             IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
*             IOCM
C
C
C... INITIALIZATION
C
CO = CCO
TO = TTO
ISFREE = 0
C
ISF = ISFIN
CTBP = CO * FRC
F65 = FRC**1.65D0
F69 = FRC**1.69D0
UOS = 0.5D0 * CTBP / ( 1.0D0 + 0.046D0*CTBP )
POS = 0.5D0 * CTBP / ( 1.0D0 + 0.070D0*CTBP )
C
DO 10 I = 1 , NO
WLTO(I) = ( 3.95D0 - 0.0144D0 * ( TMPO(I) - TO ) ) * F65
WSTO(I) = ( 4.20D0 - 0.0150D0 * ( TMPO(I) - TO ) ) * F69
HOSO(I) = CTBP * ( 1.0D0 - 0.00609D0 * WLTO(I) )
* / ( 1.0D0 + 0.043D0 * CTBP )
10 CONTINUE
C
C
RETURN
END

```

```

SUBROUTINE DENST( MO ,NO ,
*             ISFIN ,FRC ,
*             CCO ,TTO ,
*             TMPO ,WLTO ,
*             WSTO ,HOSO )
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C DIMENSION TMPO(MO) ,WLTO(MO) ,WSTO(MO) ,HOSO(MO)
C
C COMMON /DEN000/ CO ,TO ,
*             CTBP,F65,F69,UOS,POS
C COMMON /DEN001/ ISF,ISFREE
C
C COMMON /IXCOMP/NCHEMO,
*             IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*             IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*             IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*             IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
*             IOTCZR,IOSR ,IORUDI,IORUTR,IORUNI,
*             IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
*             IOCM
C
CO = CCO
TO = TTO
ISFREE = 0
C
ISF = ISFIN
CTBP = CO * FRC
F65 = FRC**1.65D0
F69 = FRC**1.69D0
UOS = 0.5D0 * CTBP / ( 1.0D0 + 0.046D0*CTBP )
POS = 0.5D0 * CTBP / ( 1.0D0 + 0.070D0*CTBP )
C
DO 20 I = 1 , NO
WLTO(I) = ( 3.95D0 - 0.0144D0 * ( TMPO(I) - TO ) ) * F65
WSTO(I) = ( 4.20D0 - 0.0150D0 * ( TMPO(I) - TO ) ) * F69
HOSO(I) = CTBP * ( 1.0D0 - 0.00609D0 * WLTO(I) )
* / ( 1.0D0 + 0.043D0 * CTBP )
20 CONTINUE
C
C
RETURN
END

```

```

DOUBLE PRECISION FUNCTION DENMSA( MXCOMP, X )
C
C MOLARITY TO SLUTE-FREE ( AQUEOUS )
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C DIMENSION X(MXCOMP)
C
COMMON /DEN000/ CO ,TO ,
* CTBP,F65,F69,UOS,POS
COMMON /DEN001/ ISF,ISFREE
C
COMMON /IXCOMP/NCHEMO,
* IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
* IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
* IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
* IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
* IOTCZR,IOSR ,IORUDI,IORUTR,IORUNI,
* IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
* IOCM
C
C IF ( ISF .EQ. ISFREE ) THEN
DENMSA = 1.000
* - 0.030900*X(IOHNO3)
* - 0.072400*X(IOU6 )
* - 0.130000*X(IOPU4 )
* - 0.100000*X(IOPU3 )
* - 0.072400*X(IOU4 )
* - 0.030000*X(IOHYD )
* - 0.040000*X(IOHAN )
C
ELSE
DENMSA = 1.000
END IF
C
RETURN
END

```

```

DOUBLE PRECISION FUNCTION DENMSO( MXCOMP,
* WLT , HOS ,
* X )
C
C MOLARITY TO SOLUTE-FREE ( ORGANIC )
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C DIMENSION X(MXCOMP)
C
COMMON /DEN000/ CO ,TO ,
* CTBP,F65,F69,UOS,POS
COMMON /DEN001/ ISF,ISFREE
C
COMMON /IXCOMP/NCHEMO,
* IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
* IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
* IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
* IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
* IOTCZR,IOSR ,IORUDI,IORUTR,IORUNI,
* IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
* IOCM
C
C IF ( ISF .EQ. ISFREE ) THEN
WCO = WLT * ( 1.000 - (X(IOU6 )+X(IOU4 ))/UOS
* - (X(IOPU4 )+X(IOPU3 ))/POS
* - 0.6500*X(IOHNO3) /HOS )
C
DENMSO = 1.000
* - 0.04300 * X(IOHNO3)
* - 0.09200 * X(IOU6 )
* - 0.13900 * X(IOPU4 )
* - 0.13900 * X(IOPU3 )
* - 0.09200 * X(IOU4 )
* - 0.017400* WCO
C
ELSE
DENMSO = 1.000
END IF
C
RETURN
END

```

```

      DOUBLE PRECISION FUNCTION DENSMA( MXCOMP, X )
C
C   SOLUTE-FREE TO MOLARITY ( AQUEOUS )
C
C   IMPLICIT REAL*8 ( A-H , 0-Z )
C
C   DIMENSION X(MXCOMP)
C
C   COMMON /DEN000/ CO ,TO ,
*   CTBP,F65,F69,U0S,POS
C   COMMON /DEN001/ ISF,ISFREE
C
C   COMMON /IXCOMP/NCHEMO,
*   IOHN03,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*   IOHN02,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*   IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*   IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
*   IOTCZR,IOSR ,IORJDI,IORJTR,IORUN1,
*   IORJMO,IOCS ,IOCE ,IOGD ,IOAM ,
*   IOCM
C
C
C   IF ( ISF .EQ. ISFREE ) THEN
      DENSMA = 1.000
*       + 0.030900*(IOHN03)
*       + 0.072400*(IOU6 )
*       + 0.130000*(IOPU4 )
*       + 0.100000*(IOPU3 )
*       + 0.072400*(IOU4 )
*       + 0.030000*(IOHYD )
*       + 0.040000*(IOHAN )
      ELSE
      DENSMA = 1.000
      END IF
C
      RETURN
      END

```

```

      DOUBLE PRECISION FUNCTION DENSMO( MXCOMP,
*   WST ,
*   X )
C
C   SOLUTE-FREE TO MOLARITY ( ORGANIC )
C
C   IMPLICIT REAL*8 ( A-H , 0-Z )
C
C   DIMENSION X(MXCOMP)
C
C   COMMON /DEN000/ CO ,TO ,
*   CTBP,F65,F69,U0S,POS
C   COMMON /DEN001/ ISF,ISFREE
C
C   COMMON /IXCOMP/NCHEMO,
*   IOHN03,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*   IOHN02,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*   IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*   IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
*   IOTCZR,IOSR ,IORJDI,IORJTR,IORUN1,
*   IORJMO,IOCS ,IOCE ,IOGD ,IOAM ,
*   IOCM
C
C
C   IF ( ISF .EQ. ISFREE ) THEN
      WCO = WST*( 1.000 - ( 2.000*( X(IOU6 ) + X(IOPU4 )
*   + X(IOPU3 ) + X(IOU4 )
*   )
*   + 0.600*(IOHN03)
*   )/CTBP
C
      DENSMO = 1.000
*       + 0.04300 * X(IOHN03)
*       + 0.09200 * X(IOU6 )
*       + 0.13900 * X(IOPU4 )
*       + 0.13900 * X(IOPU3 )
*       + 0.09200 * X(IOU4 )
*       + 0.017400* WCO
C
      ELSE
      DENSMO = 1.000
      END IF
C
      RETURN
C
C
      END

```



```

SUBROUTINE FEEDS(TIME ,
*          ISFIN ,FRC ,
*          CO ,TO ,
*          MXCOMP,
*          MXFEED,MXTABL,NFA ,NFO ,
*          TFEED ,
*          FF ,NFFTAB,FFTAB ,
*          XF ,NXFTAB,XFTAB )
C
C
C INTERPOLATE FEED FLOW RATES:FF AND FEED CONCENTRATIONS:XF.
C
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C
C DIMENSION FF (          MXFEED,2),
*          NFFTAB(          MXFEED,2),
*          FFTAB (2          ,MXTABL,MXFEED,2),
*          XF (MXCOMP          ,MXFEED,2),
*          NXFTAB(          MXFEED,2),
*          XFTAB (MXCOMP+1,MXTABL,MXFEED,2)
C
C
C... AQUEOUS PHASE
C
C DO 100 I = 1 , NFA
C
C CALL POLATE(TIME ,MXCOMP,
*          NXFTAB(I,1) ,XFTAB(1,1,1,1),XF(1,1,1) )
C
C CALL POLATE(TIME ,1 ,
*          NFFTAB(I,1) ,FFTAB(1,1,1,1),FF( 1,1) )
C
100 CONTINUE
C
C
C... ORGANIC PHASE
C
C DO 200 I = 1 , NFO
C
C CALL POLATE(TIME ,MXCOMP,
*          NXFTAB(I,2) ,XFTAB(1,1,1,2),XF(1,1,2) )
C
C CALL POLATE(TIME ,1 ,
*          NFFTAB(I,2) ,FFTAB(1,1,1,2),FF( 1,2) )
C
200 CONTINUE
C
C
C... TRANSFORM XF AND FF TO SOLUTE FREE UNIT
C
C CALL SOLFRF(ISFIN ,FRC ,
*          CO ,TO ,
*          MXCOMP,
*          MXFEED,NFA ,NFO ,
*          TFEED ,
*          XF ,FF )
C
C
C RETURN
C END

```

```

SUBROUTINE IDCHEK(IUPRT ,
*          MXDEST,MXUNIT,NUNIT ,
*          TYPE ,MSTG ,
*          ID ,IUDSTA,IUDSTO,MDEST ,
*          MXFEED,NFA ,NFO ,IFDSTA,IFDSTO)
C
C*****
C
C CHANGE THE UNIT ID'S TO DESTINATION UNIT NO.
C*****
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER*2 TYPE (          MXUNIT)
C
C DIMENSION MSTG (          MXUNIT),
*          ID (          MXUNIT),
*          IUDSTA(3          ,MXDEST,MXUNIT),
*          IUDSTO(3          ,MXDEST,MXUNIT)
C DIMENSION MDEST ( 2          ,MXUNIT)
C
C DIMENSION IFDSTA(2,MXFEED),
*          IFDSTO(2,MXFEED)
C
C
C --- AQUEOUS FEED POSITION ---
C
C DO 1300 IFA = 1 , NFA
C IJUNT = IFDSTA(1,IFA)
C KSTG = IFDSTA(2,IFA)
C DO 1100 IUNTS = 1 , NUNIT
C IUNT = IUNTS
C IF ( TYPE(IUNTS) .NE. 'TO' ) THEN
C IF( IJUNT .EQ. ID (IUNTS)
* .AND. KSTG .GT. 0
* .AND. KSTG .LE. MSTG(IUNTS) ) THEN
C GO TO 1200
C END IF
C END IF
1100 CONTINUE
C
C WRITE(IUPRT,*) ' UNIT ID OR STAGE OF AQUEOUS FEED NOT FOUND.'
C WRITE(IUPRT,*) ' FEED NO. ',IFA
C WRITE(IUPRT,*) ' FEED UNIT. ',IFDSTA(1,IFA)
C WRITE(IUPRT,*) ' FEED POSITION. ',IFDSTA(2,IFA)
C STOP
C
1200 CONTINUE
C IFDSTA(1,IFA) = IUNT
1300 CONTINUE
C
C
C --- ORGANIC FEED POSITION ---
C
C DO 2300 IFO = 1 , NFO
C IJUNT = IFDSTO(1,IFO)
C KSTG = IFDSTO(2,IFO)
C DO 2100 IUNTS = 1 , NUNIT
C IUNT = IUNTS
C IF ( TYPE(IUNTS) .NE. 'TA' ) THEN
C IF( IJUNT .EQ. ID (IUNTS)
* .AND. KSTG .GT. 0
* .AND. KSTG .LE. MSTG(IUNTS) ) THEN
C GO TO 2200
C END IF
C END IF
2100 CONTINUE
C
C WRITE(IUPRT,*) ' UNIT ID OR STAGE OF ORGANIC FEED NOT FOUND.'
C WRITE(IUPRT,*) ' FEED NO. ',IFO
C WRITE(IUPRT,*) ' FEED UNIT. ',IFDSTO(1,IFO)
C WRITE(IUPRT,*) ' FEED POSITION. ',IFDSTO(2,IFO)
C STOP
C
2200 CONTINUE
C IFDSTO(1,IFO) = IUNT
2300 CONTINUE
C
C
C --- AQUEOUS FEED POSITION ---
C
C DO 3400 IUNT = 1 , NUNIT
C
C NDEST = MDEST(1,IUNT)
C

```

```

IF ( NDEST .GT. 0 ) THEN
C
DO 3300 KDEST = 1 , NDEST
C
IDUNT = IUDSTA(1,KDEST,IUNT)
C
IF ( IDUNT .NE. 0 ) THEN
C
DO 3100 IUNTS = 1 , NUNIT
IUNTX = IUNTS
IF ( TYPE(IUNTS) .NE. 'TO' ) THEN
IF( IDUNT .EQ. ID (IUNTS)
* .AND. KSTG .GT. 0
* .AND. KSTG .LE. MSTG(IUNTS) ) THEN
GO TO 3200
END IF
END IF
3100 CONTINUE
C
WRITE(IUPRT,*) ' UNIT ID OR STAGE(AQUEOUS) NOT FOUND.'
WRITE(IUPRT,*) ' SOURCE UNIT ID.',ID(IUNT)
WRITE(IUPRT,*) ' DEST. UNIT. ',IUDSTA(1,KDEST,IUNT)
WRITE(IUPRT,*) ' DEST. STAGE. ',IUDSTA(2,KDEST,IUNT)
STOP
C
C
3200 CONTINUE
IUDSTA(1,KDEST,IUNT) = IUNTX
C
C
END IF
C
3300 CONTINUE
C
END IF
C
3400 CONTINUE
C
--- ORGANIC FEED POSITION ---
C
DO 4400 IUNT = 1 , NUNIT
C
NDEST = MDEST(2,IUNT)
C
IF ( NDEST .GT. 0 ) THEN
C
DO 4300 KDEST = 1 , NDEST
C
IDUNT = IUDSTO(1,KDEST,IUNT)
C
IF ( IDUNT .NE. 0 ) THEN
C
DO 4100 IUNTS = 1 , NUNIT
IUNTX = IUNTS
IF ( TYPE(IUNTS) .NE. 'TA' ) THEN
IF( IDUNT .EQ. ID (IUNTS)
* .AND. KSTG .GT. 0
* .AND. KSTG .LE. MSTG(IUNTS) ) THEN
GO TO 4200
END IF
END IF
4100 CONTINUE
C
WRITE(IUPRT,*) ' UNIT ID OR STAGE(ORGANIC) NOT FOUND.'
WRITE(IUPRT,*) ' SOURCE UNIT ID.',ID(IUNT)
WRITE(IUPRT,*) ' DEST. UNIT. ',IUDSTO(1,KDEST,IUNT)
WRITE(IUPRT,*) ' DEST. STAGE. ',IUDSTO(2,KDEST,IUNT)
STOP
C
C
4200 CONTINUE
IUDSTO(1,KDEST,IUNT) = IUNTX
C
C
END IF
C
C
4300 CONTINUE
C
END IF
C
4400 CONTINUE
C
RETURN
END

```

```

FUNCTION IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
C
CONVERT ID NUMBER TO UNIT NO.
C
CHARACTER*8 KEYWD
C
C
DIMENSION ID (MXUNIT)
C
IF ( NUNIT .GT. 0 ) THEN
C
DO 100 I = 1 , NUNIT
IUNT = I
IF ( N .EQ. ID(I) ) THEN
IF ( IDEF .EQ. 1 ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' ID NUMBER IS ALREADY DEFINED.'
WRITE(IUPRT,*) ' ID NUMBER = ',N
STOP
END IF
GO TO 200
END IF
C
100 CONTINUE
C
END IF
C
IF ( IDEF .EQ. 1 ) THEN
C
NUNIT = NUNIT + 1
IUNT = NUNIT
C
IF ( NUNIT .GT. MXUNIT ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' TOO MANY UNITS.'
WRITE(IUPRT,*) ' NUMBER OF UNITS(INPUT) IS ',NUNIT
WRITE(IUPRT,*) ' NUMBER OF UNITS(LIMIT) IS ',MXUNIT
STOP
END IF
C
ELSE
C
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' UNIT IS NOT DEFINED.'
WRITE(IUPRT,*) ' ID NUMBER IS ',N
STOP
C
END IF
C
200 CONTINUE
C
IDCONV = IUNT
RETURN
END

```

```

SUBROUTINE INIT(MXCOMP,MXUNIT,MXSTG ,NCHEM ,MXBLK ,MXEXTR,
*           NUNIT ,NBLK ,
*           MSTG ,MEXTRA,
*           PHIOLD,PHI ,PHIS ,DIS ,
*           X ,XN )
C
C*****
C
C   INITIALIZATION
C*****
C
C   IMPLICIT REAL*8 ( A-H , 0-Z )
C
C
C   DIMENSION MSTG (           MXUNIT)
C   DIMENSION MEXTRA (           MXUNIT)
C
C   DIMENSION PHIOLD (           MXSTG ,MXEXTR),
*   PHI (           MXSTG ,MXEXTR),
*   PHIS (           MXSTG ,MXEXTR),
*   DIS (MXCOMP ,MXSTG ,MXEXTR)
C
C   DIMENSION X (MXCOMP,2,MXBLK ),
*   XN (MXCOMP,2,MXBLK )
C
C
C   DO 130 IUNT = 1 , NUNIT
C
C     NSTG = MSTG (IUNT)
C     IEXTRA = MEXTRA (IUNT)
C
C     IF ( IEXTRA .GT. 0 ) THEN
C
C       DO 120 I = 1 , NSTG
C
C         PHIOLD(I,IEXTRA) = 0.5D0
C         PHI (I,IEXTRA) = 0.5D0
C         PHIS (I,IEXTRA) = 0.5D0
C
C         DO 110 J = 1 , MXCOMP
C           DIS (J,I,IEXTRA) = 0.0D0
C110        CONTINUE
C
C120        CONTINUE
C
C       END IF
C130        CONTINUE
C
C... SUBSTITUTE X TO XN
C
C   DO 220 I = 1 , NBLK
C     DO 210 J = 1 , MXCOMP
C       XN(J,1,I) = X(J,1,I)
C       XN(J,2,I) = X(J,2,I)
C210        CONTINUE
C220        CONTINUE
C
C
C   RETURN
C   END

```

```

SUBROUTINE INPPRT(IFOUT ,
*           TITLE ,TIME ,ENDTIM,DT ,
*           CNAME ,FMASS ,DECAYH,CVUNIT,
*           CUNIT ,CHRG ,CNTBP ,FDEG ,
*           MXCOMP ,MXUNIT,MXSTG ,MXSTG2 ,NCHEM ,MXBLK ,
*           MXREAC ,MXDEST ,MXFEED ,MXTABL,
*           MXEXTR,NUNIT ,FRC ,CO ,TO ,CDBP ,
*           NFA ,NFO ,NEXTRA,
*           FF ,NFFTAB,FFTAB ,XF ,NXFTAB ,XF-TAB ,
*           IFDSTA ,IFDSTO,
*           ID ,IREACT ,FCHEM ,ECHEM ,TYPE ,MSTG ,
*           IUDSTA ,IUDSTO ,MDEST ,MEXTRA ,TAV ,VTOT ,
*           ACID ,ACIDIN ,OXID ,FOUT ,FFOUT ,FFOUT ,
*           IDEXTR,CHRDIS,
*           FL ,FR ,TSTG ,V ,VS ,
*           PHI ,PHIS ,
*           IPOS ,
*           XN )
C
C
C   IMPLICIT REAL*8 ( A-H , 0-Z )
C
C   PARAMETER ( MXFROM = 10 )
C   CHARACTER*8 FROMID (MXFROM)
C   CHARACTER*2 TYPEU
C
C   CHARACTER*72 TITLE
C
C   CHARACTER*8 CNAME (MXCOMP)
C   DIMENSION FMASS (MXCOMP)
C   DIMENSION DECAYH (MXCOMP)
C   DIMENSION CVUNIT (MXCOMP)
C   CHARACTER*8 CUNIT (MXCOMP)
C   DIMENSION CHRG (MXCOMP),
*   CNTBP (MXCOMP)
C
C
C   DIMENSION FF (           MXFEED,2),
*   NFFTAB (           MXFEED,2),
*   FFTAB (2 ,           ,MXTABL ,MXFEED,2),
*   XF (MXCOMP ,           MXFEED,2),
*   NXFTAB (           MXFEED,2),
*   XF-TAB (MXCOMP+1,MXTABL ,MXFEED,2),
*   IFDSTA(2 ,           ,MXFEED ),
*   IFDSTO(2 ,           ,MXFEED )
C
C   DIMENSION ID (           MXUNIT)
C   DIMENSION IREACT (           MXUNIT)
C   DIMENSION FCHEM (MXFEAC,2 ,           MXUNIT),
*   ECHEM (MXFEAC,2 ,           MXUNIT)
C
C   CHARACTER*2 TYPE (           MXUNIT)
C
C   DIMENSION MSTG (           MXUNIT),
*   IUDSTA(3 ,           ,MXDEST ,MXUNIT),
*   IUDSTO(3 ,           ,MXDEST ,MXUNIT)
C   DIMENSION MDEST ( 2 ,           MXUNIT),
*   MEXTRA (           MXUNIT)
C
C   DIMENSION TAV (           MXUNIT),
*   VTOT (           MXUNIT),
*   ACID (           MXUNIT),
*   ACIDIN (           MXUNIT),
*   OXID (           MXUNIT),
*   FOUT ( 2 ,           MXUNIT),
*   FFOUT (MXDEST,2 ,           MXUNIT)
C
C   DIMENSION IDEXTR (           MXEXTR),
*   DCHEM ( 10)
C   CHARACTER*8 CHRDIS (MXCOMP ,MXEXTR)
C
C   DIMENSION FL (2,MXSTG ,MXEXTR),
*   FR ( MXSTG ,MXEXTR),
*   TSTG ( MXSTG ,MXEXTR),
*   V ( MXSTG ,MXEXTR),
*   VS ( MXSTG ,MXEXTR),
*   PHI ( MXSTG ,MXEXTR),
*   PHIS ( MXSTG ,MXEXTR)
C
C   DIMENSION IPOS (2,MXSTG2 ,MXUNIT)
C
C   DIMENSION XN (MXCOMP,2,MXBLK )
C
C   COMMON /IXCOMP/NCHEM,
*   IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*   IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,

```

```
*         IONP4 , IONP5 , IONP6 , IOZR , IOTC4 ,
*         IOTC5 , IOTC6 , IOTC7 , IOTCU6 , IOTCP4 ,
*         IOTCZR , IOSR , IOFLDI , IOFLTR , IOFLNI ,
*         IOFLMO , IOCS , IOCE , IOGD , IOAM ,
*         IOCM
C
COMMON /DAY/CDATE
CHARACTER CDATE*9
DIMENSION FFOUT(2,2, MXUNIT)
C
DATA ONE/1.000/
C
IUERR = 0
C
C
MIC1 = 32
MIC2 = 33
MIC3 = 34
MIC4 = 35
MIC5 = 36
MIC6 = 37
MIC7 = 38
MIC8 = 39
MIC9 = 40
C
C
FMFBP = 266.3D0
FMDBP = 210.2D0
EV = 1.6021892D-19
ANA = 6.022045D23
CTBP = C0 * FRC
GVAL = 0.73D0
C
C
WRITE(IFOUT,4010) TITLE,TIME
WRITE(IFOUT,4020) FRC
WRITE(IFOUT,4030) CDBP
WRITE(IFOUT,4040) FDEG
C
C
WRITE(IFOUT,4100) 'CHEMICAL COMPONENTS'
WRITE(IFOUT,4200) 'CHEMICAL ',
*              'ATOMI C ',
*              'DECAY-HEAT',
*              'INPUT-UNIT',
*              'CHARGE ',
*              'N-TBP ',
WRITE(IFOUT,4210) 'COMPONENTS',
*              'WEI GHT ',
*              ', ',
*              ', ',
*              ', ',
*              ', ',
*              ', ',
*              ', ',
WRITE(IFOUT,4210) ', ',
*              '(G/MOL) ',
*              '(W/G) ',
*              ', ',
*              ', ',
*              ', ',
*              ', ',
C
DO 100 J = 1 , MXCOMP
  WRITE(IFOUT,4220) J ,
*              CNAME (J),
*              FMASS (J),
*              DECAYH(J),
*              CUNIT (J),
*              CHR G (J),
*              CNTBP (J)
100 CONTINUE
WRITE(IFOUT,4210) ' ', ' ',
C
C
C
WRITE(IFOUT,4010) TITLE,TIME
WRITE(IFOUT,4100) 'DISTRIBUTION COEFFICIENT OPTION' //
*              ' AT MIXER-SETTLER'
WRITE(IFOUT,4230) 'CHEMICAL ',
*              (IDEXTR(IEXTRA), IEXTRA=1, NEXTRA)
WRITE(IFOUT,4240) 'COMPONENTS'
C
DO 200 J = 1 , MXCOMP
  WRITE(IFOUT,4250) J ,
*              CNAME (J),
*              (CHFDIS(J, IEXTRA), IEXTRA=1, NEXTRA)
200 CONTINUE
WRITE(IFOUT,4210) ' ', ' ',
C
```

```

C
C
WRITE(IFOUT,4010) TITLE,TIME
WRITE(IFOUT,4100) 'MULTIPLIER TO RATE CONSTANT' //
*              ' IN MIXER-SETTLER(AQUEOUS)'
WRITE(IFOUT,4260) (IDEXTR(IEXTRA), IEXTRA=1, NEXTRA)
WRITE(IFOUT,4270)
C
DO 300 J = 1 , MXREAC
  IEXTRA = 1
  DO IUNT = 1, NUNIT
    IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
      DCHEM(IEXTRA) = FCHEM(J,1, IUNT)
      IEXTRA = IEXTRA + 1
    END IF
  ENDDO
  WRITE(IFOUT,4280) J , (DCHEM(IEXTRA), IEXTRA=1, NEXTRA)
300 CONTINUE
WRITE(IFOUT,4210) ' ', ' ',
C
WRITE(IFOUT,4010) TITLE,TIME
WRITE(IFOUT,4100) 'MULTIPLIER TO RATE CONSTANT' //
*              ' IN MIXER-SETTLER(ORGANIC)'
WRITE(IFOUT,4260) (IDEXTR(IEXTRA), IEXTRA=1, NEXTRA)
WRITE(IFOUT,4270)
C
DO J = 1 , 42
  IF (
* J .EQ. 1 .OR.
* J .EQ. 4 .OR.
* J .EQ. 5 .OR.
* J .EQ. 6 .OR.
* J .EQ. 9 .OR.
* J .EQ. 14 .OR.
* J .EQ. 19 .OR.
* J .EQ. 20 .OR.
* J .EQ. 41 .OR.
* J .EQ. 42 ) THEN
    IEXTRA = 1
    DO IUNT = 1, NUNIT
      IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
        DCHEM(IEXTRA) = FCHEM(J,2, IUNT)
        IEXTRA = IEXTRA + 1
      END IF
    ENDDO
    WRITE(IFOUT,4280) J , (DCHEM(IEXTRA), IEXTRA=1, NEXTRA)
  END IF
ENDDO
WRITE(IFOUT,4210) ' ', ' ',
C
C
C
WRITE(IFOUT,5000) NUNIT , CDATE
C
C
WRITE(IFOUT,4300) 'PROCESS UNIT(S) INFORMATION'
WRITE(IFOUT,4310) 'NO ID TYP ',
*              'REACT',
*              'STG ',
*              'V(TOT.) ',
*              'ACID ',
*              'ACID-IN ',
*              'OXID ',
*              'OUTFLW(A) ',
*              'OUTFLW(O) ',
*              'DEST (A/1) OUTFLW(2) ',
*              'DEST (O/2) '
WRITE(IFOUT,4310) ' ', ' ', ' ',
*              ' ', ' ', ' ',
*              ' ', ' ', ' ',
*              '(L) ',
*              '(MOL/L) ',
*              '(MOL/L) ',
*              '(-) ',
*              '(L/HR) ',
*              '(L/HR) ',
*              '(L/HR) ',
*              '(L/HR) ',
C
DO 1100 IUNT = 1 , NUNIT
C
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C
... MIXER-SETTLER ...
C
IDEST1 = IUDSTA(1,1, IUNT)
IF ( IDEST1 .GT. 0 ) THEN
  IDEST1 = ID(IDEST1)
END IF
```

```

C
  IDEST2 = IUDSTO(1,1,IUNT)
  IF ( IDEST2 .GT. 0 ) THEN
    IDEST2 = ID(IDEST2)
  END IF
C
  WRITE(IFOUT,4315) IUNT
  *
  *      ID ( IUNT),
  *      TYPE ( IUNT),
  *      IREACT( IUNT),
  *      MSTG ( IUNT),
  *      FOUT (1 ,IUNT),
  *      FOUT (2 ,IUNT),
  *      IDEST1
  *      IUDSTA(2,1,IUNT),
  *      IDEST2
  *      IUDSTO(2,1,IUNT)
C
  ELSE IF ( TYPE(IUNT) .EQ. 'TA' ) THEN
C
  ... AQUEOUS TANK ...
C
  IDEST1 = IUDSTA(1,1,IUNT)
  IF ( IDEST1 .GT. 0 ) THEN
    IDEST1 = ID(IDEST1)
  END IF
C
  IDEST2 = IUDSTA(1,2,IUNT)
  IF ( IDEST2 .GT. 0 ) THEN
    IDEST2 = ID(IDEST2)
  END IF
C
  WRITE(IFOUT,4320) IUNT
  *
  *      ID ( IUNT),
  *      TYPE ( IUNT),
  *      IREACT( IUNT),
  *      MSTG ( IUNT),
  *      VTOT ( IUNT),
  *      ACID ( IUNT),
  *      ACIDIN( IUNT),
  *      OXID ( IUNT),
  *      FOUT (1 ,IUNT),
  *      FOUT (2 ,IUNT),
  *      IDEST1
  *      IUDSTA(2,1,IUNT),
  *      FFOUT (2,1,IUNT),
  *      IDEST2
  *      IUDSTA(2,2,IUNT)
C
  ELSE
C
  ... ORGANIC TANK ...
C
  IDEST1 = IUDSTO(1,1,IUNT)
  IF ( IDEST1 .GT. 0 ) THEN
    IDEST1 = ID(IDEST1)
  END IF
C
  IDEST2 = IUDSTO(1,2,IUNT)
  IF ( IDEST2 .GT. 0 ) THEN
    IDEST2 = ID(IDEST2)
  END IF
C
  WRITE(IFOUT,4320) IUNT
  *
  *      ID ( IUNT),
  *      TYPE ( IUNT),
  *      IREACT( IUNT),
  *      MSTG ( IUNT),
  *      VTOT ( IUNT),
  *      ACID ( IUNT),
  *      ACIDIN( IUNT),
  *      OXID ( IUNT),
  *      FOUT (1 ,IUNT),
  *      FOUT (2 ,IUNT),
  *      IDEST1
  *      IUDSTO(2,1,IUNT),
  *      FFOUT (2,2,IUNT),
  *      IDEST2
  *      IUDSTO(2,2,IUNT)
C
  END IF
C
1100 CONTINUE
C
  WRITE(IFOUT,4400) ' AQUEOUS FEED(S) INFORMATION'
  WRITE(IFOUT,4411) ' NO. DEST.(ID,TYPE STAGE) ',
  *      ' TIME '

```

```

*      'FLOW_RATE '
*      CNAME(10HNO3),
*      CNAME(10U6 ),
*      CNAME(10PU4 )
  WRITE(IFOUT,4411) '
  *      (HR)
  *      (L/HR)
  *      CUNIT(10HNO3),
  *      CUNIT(10U6 ),
  *      CUNIT(10PU4 )
C
DO IFA = 1 , NFA
C
  NFT = NFFTAB(IFA,1)
C
  DO IFT = 1 , NFT
C
    TIM = FFTAB (1,IFT,IFA,1)
    FRATE = FFTAB (2,IFT,IFA,1)
    XH = XFTAB (10HNO3+1,IFT,IFA,1) * CVUNIT(10HNO3)
    XU6 = XFTAB (10U6 +1,IFT,IFA,1) * CVUNIT(10U6 )
    XPU4 = XFTAB (10PU4 +1,IFT,IFA,1) * CVUNIT(10PU4 )
C
    IF ( IFT .EQ. 1 ) THEN
C
      WRITE(IFOUT,4431) IFA,
      *      ID (IFDSTA(1,IFA)),
      *      TYPE(IFDSTA(1,IFA)),
      *      IFDSTA(2,IFA),
      *      TIM,
      *      FRATE,
      *      XH,
      *      XU6,
      *      XPU4
C
    ELSE
C
      WRITE(IFOUT,4441) TIM,
      *      FRATE,
      *      XH,
      *      XU6,
      *      XPU4
C
    END IF
C
  ENDDO
C
  ENDDO
C
  WRITE(IFOUT,4400) ' AQUEOUS FEED(S) INFORMATION(1)'
  WRITE(IFOUT,4412) ' NO. ',
  *      CNAME(10PU3 ),
  *      CNAME(10U4 ),
  *      CNAME(10HNO2),
  *      CNAME(10HYD ),
  *      CNAME(10HAN ),
  *      CNAME(10PU5 ),
  *      CNAME(10PU6 ),
  *      CNAME(10NP4 ),
  *      CNAME(10NP5 ),
  *      CNAME(10NP6 )
  WRITE(IFOUT,4412) '
  *      CUNIT(10PU3 ),
  *      CUNIT(10U4 ),
  *      CUNIT(10HNO2),
  *      CUNIT(10HYD ),
  *      CUNIT(10HAN ),
  *      CUNIT(10PU5 ),
  *      CUNIT(10PU6 ),
  *      CUNIT(10NP4 ),
  *      CUNIT(10NP5 ),
  *      CUNIT(10NP6 )
C
DO 2120 IFA = 1 , NFA
C
  NFT = NFFTAB(IFA,1)
C
  DO 2110 IFT = 1 , NFT
C
    XPU3 = XFTAB (10PU3 +1,IFT,IFA,1) * CVUNIT(10PU3 )
    XU4 = XFTAB (10U4 +1,IFT,IFA,1) * CVUNIT(10U4 )
    XNO2 = XFTAB (10HNO2+1,IFT,IFA,1) * CVUNIT(10HNO2)
    XHYD = XFTAB (10HYD +1,IFT,IFA,1) * CVUNIT(10HYD )
    XHAN = XFTAB (10HAN +1,IFT,IFA,1) * CVUNIT(10HAN )
    XPU5 = XFTAB (10PU5 +1,IFT,IFA,1) * CVUNIT(10PU5 )
    XPU6 = XFTAB (10PU6 +1,IFT,IFA,1) * CVUNIT(10PU6 )
    XNP4 = XFTAB (10NP4 +1,IFT,IFA,1) * CVUNIT(10NP4 )
    XNP5 = XFTAB (10NP5 +1,IFT,IFA,1) * CVUNIT(10NP5 )

```

```

XNP6 = XFTAB (10NP6 +1, IFT, IFA, 1) * CVUNIT(10NP6 )
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT, 4432) IFA,
*
* XPU3,
* XU4,
* XNO2,
* XHYD,
* XHAN,
* XPUS,
* XPU6,
* XNP4,
* XNP5,
* XNP6
C
ELSE
C
WRITE(IFOUT, 4442)
*
* XPU3,
* XU4,
* XNO2,
* XHYD,
* XHAN,
* XPUS,
* XPU6,
* XNP4,
* XNP5,
* XNP6
C
END IF
C
2110 CONTINUE
2120 CONTINUE
C
WRITE (IFOUT, 4400) ' AQUEOUS FEED(S) INFORMATION(2) '
WRITE (IFOUT, 4412) ' NO. ',
*
* CNAME(10ZR ),
* CNAME(10TC4 ),
* CNAME(10TC5 ),
* CNAME(10TC6 ),
* CNAME(10TC7 )
WRITE (IFOUT, 4412) ' . ',
*
* CUNIT(10ZR ),
* CUNIT(10TC4 ),
* CUNIT(10TC5 ),
* CUNIT(10TC6 ),
* CUNIT(10TC7 )
C
DO 2140 IFA = 1 , NFA
C
NFT = NFFTAB(IFA,1)
C
DO 2130 IFT = 1 , NFT
C
XZR = XFTAB (10ZR +1, IFT, IFA, 1) * CVUNIT(10ZR )
XTC4 = XFTAB (10TC4 +1, IFT, IFA, 1) * CVUNIT(10TC4 )
XTC5 = XFTAB (10TC5 +1, IFT, IFA, 1) * CVUNIT(10TC5 )
XTC6 = XFTAB (10TC6 +1, IFT, IFA, 1) * CVUNIT(10TC6 )
XTC7 = XFTAB (10TC7 +1, IFT, IFA, 1) * CVUNIT(10TC7 )
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT, 4432) IFA,
*
* XZR,
* XTC4,
* XTC5,
* XTC6,
* XTC7
C
ELSE
C
WRITE(IFOUT, 4442)
*
* XZR,
* XTC4,
* XTC5,
* XTC6,
* XTC7
C
END IF
C
2130 CONTINUE
2140 CONTINUE
C
WRITE (IFOUT, 4400) ' AQUEOUS FEED(S) INFORMATION(3) '

```

```

WRITE (IFOUT, 4412) ' NO. ',
*
* CNAME(10SR ),
* CNAME(10RUDI ),
* CNAME(10RUTR ),
* CNAME(10RUNI ),
* CNAME(10RUMO ),
* CNAME(10CS ),
* CNAME(10CE ),
* CNAME(10GD ),
* CNAME(10AM ),
* CNAME(10CM )
WRITE (IFOUT, 4412) ' . ',
*
* CUNIT(10SR ),
* CUNIT(10RUDI ),
* CUNIT(10RUTR ),
* CUNIT(10RUNI ),
* CUNIT(10RUMO ),
* CUNIT(10CS ),
* CUNIT(10CE ),
* CUNIT(10GD ),
* CUNIT(10AM ),
* CUNIT(10CM )
C
DO 2160 IFA = 1 , NFA
C
NFT = NFFTAB(IFA,1)
C
DO 2150 IFT = 1 , NFT
C
XSR = XFTAB (10SR +1, IFT, IFA, 1) * CVUNIT(10SR )
XRUDI = XFTAB (10RUDI+1, IFT, IFA, 1) * CVUNIT(10RUDI)
XRUTR = XFTAB (10RUTR+1, IFT, IFA, 1) * CVUNIT(10RUTR)
XRUNI = XFTAB (10RUNI+1, IFT, IFA, 1) * CVUNIT(10RUNI)
XRUMO = XFTAB (10RUMO+1, IFT, IFA, 1) * CVUNIT(10RUMO)
XCS = XFTAB (10CS +1, IFT, IFA, 1) * CVUNIT(10CS )
XCE = XFTAB (10CE +1, IFT, IFA, 1) * CVUNIT(10CE )
XGD = XFTAB (10GD +1, IFT, IFA, 1) * CVUNIT(10GD )
XAM = XFTAB (10AM +1, IFT, IFA, 1) * CVUNIT(10AM )
XCM = XFTAB (10CM +1, IFT, IFA, 1) * CVUNIT(10CM )
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT, 4432) IFA,
*
* XSR ,
* XRUDI ,
* XRUTR ,
* XRUNI ,
* XRUMO ,
* XCS ,
* XCE ,
* XGD ,
* XAM ,
* XCM
C
ELSE
C
WRITE(IFOUT, 4442)
*
* XSR ,
* XRUDI ,
* XRUTR ,
* XRUNI ,
* XRUMO ,
* XCS ,
* XCE ,
* XGD ,
* XAM ,
* XCM
C
END IF
C
2150 CONTINUE
2160 CONTINUE
C
WRITE (IFOUT, 4400) ' AQUEOUS FEED(S) INFORMATION(4) '
WRITE (IFOUT, 4412) ' NO. ',
*
* CNAME(MI C1 ),
* CNAME(MI C2 ),
* CNAME(MI C3 ),
* CNAME(MI C4 ),
* CNAME(MI C5 ),
* CNAME(MI C6 ),
* CNAME(MI C7 ),
* CNAME(MI C8 ),
* CNAME(MI C9 )
WRITE (IFOUT, 4412) ' . ',
*
* CUNIT(MI C1 ),
* CUNIT(MI C2 ),

```

```

*          CUNIT(MIC3),
*          CUNIT(MIC4),
*          CUNIT(MIC5),
*          CUNIT(MIC6),
*          CUNIT(MIC7),
*          CUNIT(MIC8),
*          CUNIT(MIC9)
C
DO 2180 IFA = 1 , NFA
C
NFT = NFFTAB(IFA,1)
C
DO 2170 IFT = 1 , NFT
C
XMI C1 = XFTAB (MIC1+1, IFT, IFA, 1) * CVUNIT(MIC1)
XMI C2 = XFTAB (MIC2+1, IFT, IFA, 1) * CVUNIT(MIC2)
XMI C3 = XFTAB (MIC3+1, IFT, IFA, 1) * CVUNIT(MIC3)
XMI C4 = XFTAB (MIC4+1, IFT, IFA, 1) * CVUNIT(MIC4)
XMI C5 = XFTAB (MIC5+1, IFT, IFA, 1) * CVUNIT(MIC5)
XMI C6 = XFTAB (MIC6+1, IFT, IFA, 1) * CVUNIT(MIC6)
XMI C7 = XFTAB (MIC7+1, IFT, IFA, 1) * CVUNIT(MIC7)
XMI C8 = XFTAB (MIC8+1, IFT, IFA, 1) * CVUNIT(MIC8)
XMI C9 = XFTAB (MIC9+1, IFT, IFA, 1) * CVUNIT(MIC9)
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT,4432) IFA,
*          XMI C1,
*          XMI C2,
*          XMI C3,
*          XMI C4,
*          XMI C5,
*          XMI C6,
*          XMI C7,
*          XMI C8,
*          XMI C9
C
ELSE
C
WRITE(IFOUT,4442)
*          XMI C1,
*          XMI C2,
*          XMI C3,
*          XMI C4,
*          XMI C5,
*          XMI C6,
*          XMI C7,
*          XMI C8,
*          XMI C9
C
END IF
C
2170 CONTINUE
2180 CONTINUE
C
WRITE(IFOUT,4400) ' ORGANIC FEED(S) INFORMATION'
WRITE(IFOUT,4411) ' NO. DEST. (ID. TYPE STAGE) ',
*          ' TIME ',
*          ' FLOW RATE ',
*          CNAME(10HNO3),
*          CNAME(10U6 ),
*          CNAME(10PU4 )
WRITE(IFOUT,4411) ' (HR) ',
*          ' (L/HR) ',
*          CUNIT(10HNO3),
*          CUNIT(10U6 ),
*          CUNIT(10PU4 )
C
DO IFO = 1 , NFO
C
NFT = NFFTAB(IFO,2)
C
DO IFT = 1 , NFT
C
TIM = FFTAB (1, IFT, IFO, 2)
FRATE = FFTAB (2, IFT, IFO, 2)
YH = XFTAB (10HNO3+1, IFT, IFO, 2)
YU6 = XFTAB (10U6 +1, IFT, IFO, 2)
YPU4 = XFTAB (10PU4 +1, IFT, IFO, 2)
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT,4431) IFO,
*          ID (IFDSTO(1, IFO)),
*          TYPE(IFDSTO(1, IFO)),

```

```

*          IFDSTO(2, IFO),
*          TIM,
*          FRATE,
*          YH,
*          YU6,
*          YPU4
C
ELSE
C
WRITE(IFOUT,4441) TIM,
*          FRATE,
*          YH,
*          YU6,
*          YPU4
C
END IF
C
ENDDO
C
ENDDO
C
WRITE(IFOUT,4400) ' ORGANIC FEED(S) INFORMATION(1)'
WRITE(IFOUT,4412) ' NO. ',
*          CNAME(10PU3 ),
*          CNAME(10U4 ),
*          CNAME(10HNO2),
*          CNAME(10HYD ),
*          CNAME(10HAN ),
*          CNAME(10PU5 ),
*          CNAME(10NP4 ),
*          CNAME(10NPS ),
*          CNAME(10NP6 )
WRITE(IFOUT,4412) ' ',
*          CUNIT(10PU3 ),
*          CUNIT(10U4 ),
*          CUNIT(10HNO2),
*          CUNIT(10HYD ),
*          CUNIT(10HAN ),
*          CUNIT(10PU5 ),
*          CUNIT(10PU6 ),
*          CUNIT(10NP4 ),
*          CUNIT(10NPS ),
*          CUNIT(10NP6 )
C
DO 2220 IFO = 1 , NFO
C
NFT = NFFTAB(IFO,2)
C
DO 2210 IFT = 1 , NFT
C
YPU3 = XFTAB (10PU3 +1, IFT, IFO, 2) * CVUNIT(10PU3 )
YU4 = XFTAB (10U4 +1, IFT, IFO, 2) * CVUNIT(10U4 )
YNO2 = XFTAB (10HNO2+1, IFT, IFO, 2) * CVUNIT(10HNO2)
YHYD = XFTAB (10HYD +1, IFT, IFO, 2) * CVUNIT(10HYD )
YHAN = XFTAB (10HAN +1, IFT, IFO, 2) * CVUNIT(10HAN )
YPU5 = XFTAB (10PU5 +1, IFT, IFO, 2) * CVUNIT(10PU5 )
YPU6 = XFTAB (10PU6 +1, IFT, IFO, 2) * CVUNIT(10PU6 )
YNP4 = XFTAB (10NP4 +1, IFT, IFO, 2) * CVUNIT(10NP4 )
YNP5 = XFTAB (10NP5 +1, IFT, IFO, 2) * CVUNIT(10NP5 )
YNP6 = XFTAB (10NP6 +1, IFT, IFO, 2) * CVUNIT(10NP6 )
C
IF ( IFT .EQ. 1 ) THEN
C
WRITE(IFOUT,4432) IFO,
*          YPU3,
*          YU4,
*          YNO2,
*          YHYD,
*          YHAN,
*          YPU5,
*          YPU6,
*          YNP4,
*          YNP5,
*          YNP6
C
ELSE
C
WRITE(IFOUT,4442)
*          YPU3,
*          YU4,
*          YNO2,
*          YHYD,
*          YHAN,
*          YPU5,
*          YPU6,
*          YNP4,
*          YNP5,

```

```

*
C          YNP6
          END IF
C
C
C 2210 CONTINUE
C 2220 CONTINUE
C
C      WRITE (IFOUT, 4400) ' ORGANIC FEED(S) INFORMATION(2)'
C      WRITE (IFOUT, 4412) ' NO. ',
*          CNAME(10ZR ),
*          CNAME(10TC4 ),
*          CNAME(10TC5 ),
*          CNAME(10TC6 ),
*          CNAME(10TC7 ),
*          CNAME(10TCU6),
*          CNAME(10TCP4),
*          CNAME(10TCZR)
C      WRITE (IFOUT, 4412) ' . ',
*          CUNIT(10ZR ),
*          CUNIT(10TC4 ),
*          CUNIT(10TC5 ),
*          CUNIT(10TC6 ),
*          CUNIT(10TC7 ),
*          CUNIT(10TCU6),
*          CUNIT(10TCP4),
*          CUNIT(10TCZR)
C
C      DO 2240 IFO = 1 , NFO
C
C          NFT = NFFTAB(IFO,2)
C
C          DO 2230 IFT = 1 , NFT
C
C              YZR = XFTAB (10ZR +1,IFT, IFO,2) * CVUNIT(10ZR )
C              YTC4 = XFTAB (10TC4 +1,IFT, IFO,2) * CVUNIT(10TC4 )
C              YTC5 = XFTAB (10TC5 +1,IFT, IFO,2) * CVUNIT(10TC5 )
C              YTC6 = XFTAB (10TC6 +1,IFT, IFO,2) * CVUNIT(10TC6 )
C              YTC7 = XFTAB (10TC7 +1,IFT, IFO,2) * CVUNIT(10TC7 )
C              YTCU6 = XFTAB (10TCU6+1, IFT, IFO,2) * CVUNIT(10TCU6)
C              YTCP4 = XFTAB (10TCP4+1, IFT, IFO,2) * CVUNIT(10TCP4)
C              YTCZR = XFTAB (10TCZR+1, IFT, IFO,2) * CVUNIT(10TCZR)
C
C          IF ( IFT .EQ. 1 ) THEN
C
C              WRITE (IFOUT, 4432) IFO,
*                  YZR,
*                  YTC4,
*                  YTC5,
*                  YTC6,
*                  YTC7,
*                  YTCU6,
*                  YTCP4,
*                  YTCZR
C
C          ELSE
C
C              WRITE (IFOUT, 4442)
*                  YZR,
*                  YTC4,
*                  YTC5,
*                  YTC6,
*                  YTC7,
*                  YTCU6,
*                  YTCP4,
*                  YTCZR
C
C          END IF
C
C 2230 CONTINUE
C 2240 CONTINUE
C
C      WRITE (IFOUT, 4400) ' ORGANIC FEED(S) INFORMATION(3)'
C      WRITE (IFOUT, 4412) ' NO. ',
*          CNAME(10SR ),
*          CNAME(10RUD1),
*          CNAME(10RUTR),
*          CNAME(10RUN1),
*          CNAME(10RUMO),
*          CNAME(10CS ),
*          CNAME(10CE ),
*          CNAME(10GD ),
*          CNAME(10AM ),
*          CNAME(10CM )
C      WRITE (IFOUT, 4412) ' . ',
*          CUNIT(10SR ),

```

```

*          CUNIT(10RUD1),
*          CUNIT(10RUTR),
*          CUNIT(10RUN1),
*          CUNIT(10RUMO),
*          CUNIT(10CS ),
*          CUNIT(10CE ),
*          CUNIT(10GD ),
*          CUNIT(10AM ),
*          CUNIT(10CM )
C
C      DO 2260 IFO = 1 , NFO
C
C          NFT = NFFTAB(IFO,2)
C
C          DO 2250 IFT = 1 , NFT
C
C              YSR = XFTAB (10SR +1,IFT, IFO,2) * CVUNIT(10SR )
C              YRUD1 = XFTAB (10RUD1+1, IFT, IFO,2) * CVUNIT(10RUD1)
C              YRUTR = XFTAB (10RUTR+1, IFT, IFO,2) * CVUNIT(10RUTR)
C              YRUN1 = XFTAB (10RUN1+1, IFT, IFO,2) * CVUNIT(10RUN1)
C              YRUMO = XFTAB (10RUMO+1, IFT, IFO,2) * CVUNIT(10RUMO)
C              YCS = XFTAB (10CS +1,IFT, IFO,2) * CVUNIT(10CS )
C              YCE = XFTAB (10CE +1,IFT, IFO,2) * CVUNIT(10CE )
C              YGD = XFTAB (10GD +1, IFT, IFO,2) * CVUNIT(10GD )
C              YAM = XFTAB (10AM +1, IFT, IFO,2) * CVUNIT(10AM )
C              YCM = XFTAB (10CM +1, IFT, IFO,2) * CVUNIT(10CM )
C
C          IF ( IFT .EQ. 1 ) THEN
C
C              WRITE (IFOUT, 4432) IFO,
*                  YSR,
*                  YRUD1,
*                  YRUTR,
*                  YRUN1,
*                  YRUMO,
*                  YCS,
*                  YCE,
*                  YGD,
*                  YAM,
*                  YCM
C
C          ELSE
C
C              WRITE (IFOUT, 4442)
*                  YSR,
*                  YRUD1,
*                  YRUTR,
*                  YRUN1,
*                  YRUMO,
*                  YCS,
*                  YCE,
*                  YGD,
*                  YAM,
*                  YCM
C
C          END IF
C
C 2250 CONTINUE
C 2260 CONTINUE
C
C      WRITE (IFOUT, 4400) ' ORGANIC FEED(S) INFORMATION(4)'
C      WRITE (IFOUT, 4412) ' NO. ',
*          CNAME(MI C1),
*          CNAME(MI C2),
*          CNAME(MI C3),
*          CNAME(MI C4),
*          CNAME(MI C5),
*          CNAME(MI C6),
*          CNAME(MI C7),
*          CNAME(MI C8),
*          CNAME(MI C9)
C      WRITE (IFOUT, 4412) ' . ',
*          CUNIT(MI C1),
*          CUNIT(MI C2),
*          CUNIT(MI C3),
*          CUNIT(MI C4),
*          CUNIT(MI C5),
*          CUNIT(MI C6),
*          CUNIT(MI C7),
*          CUNIT(MI C8),
*          CUNIT(MI C9)
C
C      DO 2280 IFO = 1 , NFO
C
C          NFT = NFFTAB(IFO,2)
C
C          DO 2270 IFT = 1 , NFT

```



```

C
YMIC1 = XFTAB (MIC1+1, IFT, IFO, 2) * CVUNIT(MIC1)
YMIC2 = XFTAB (MIC2+1, IFT, IFO, 2) * CVUNIT(MIC2)
YMIC3 = XFTAB (MIC3+1, IFT, IFO, 2) * CVUNIT(MIC3)
YMIC4 = XFTAB (MIC4+1, IFT, IFO, 2) * CVUNIT(MIC4)
YMIC5 = XFTAB (MIC5+1, IFT, IFO, 2) * CVUNIT(MIC5)
YMIC6 = XFTAB (MIC6+1, IFT, IFO, 2) * CVUNIT(MIC6)
YMIC7 = XFTAB (MIC7+1, IFT, IFO, 2) * CVUNIT(MIC7)
YMIC8 = XFTAB (MIC8+1, IFT, IFO, 2) * CVUNIT(MIC8)
YMIC9 = XFTAB (MIC9+1, IFT, IFO, 2) * CVUNIT(MIC9)

C
IF ( IFT .EQ. 1 ) THEN

C
WRITE(IFO, 4432) IFO,
*
* YMIC1,
* YMIC2,
* YMIC3,
* YMIC4,
* YMIC5,
* YMIC6,
* YMIC7,
* YMIC8,
* YMIC9

C
ELSE

C
WRITE(IFO, 4442)
*
* YMIC1,
* YMIC2,
* YMIC3,
* YMIC4,
* YMIC5,
* YMIC6,
* YMIC7,
* YMIC8,
* YMIC9

C
END IF

C
2270 CONTINUE
2280 CONTINUE

C
DO 3400 IUNT = 1 , NUNIT

C
NSTG = MSTG (IUNT)
IEXTRA = MEXTRA(IUNT)

C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN

C
... MIXER-SETTLER CONCENTRATIONS ...

C
WRITE(IFO, 5010) TITLE, TIME
WRITE(IFO, 5100) ID(IUNT), TYPE(IUNT),
* 'OPERATION CONDITIONS'
WRITE(IFO, 5200) 'TMP',
* 'VOL-MIXER',
* 'VOL-SETTL',
* 'FLOW(AQ.)',
* 'FLOW(ORG)',
* 'FLOW(RCY)',
* 'FRACT-AQ',
* 'FRACT-ORG',
* 'HGT',
* 'AQ-FED-TOT',
* 'ORG-FED-TOT',
* 'FEED_STREAM(FROM)'
WRITE(IFO, 5210) '(C)',
* '(L)',
* '(L)',
* '(L/HR)',
* '(L/HR)',
* '(L/HR)',
* '(L/HR)',
* '(-)',
* '(-)',
* '(-)',
* '(L/HR)',
* '(L/HR)'

C
DO 3300 I = 1 , NSTG

C
IND = IPOS(2, 2*I-1, IUNT)
INDSO = IPOS(2, 2*I , IUNT)

C
NFROM = 0
AQTOT = 0.0D0
ORGTOT = 0.0D0

```

```

C
DO 3010 IFA = 1 , NFA
*
* IF ( IFDSTA(1, IFA) .EQ. IUNT
* .AND. IFDSTA(2, IFA) .EQ. I ) THEN
* NFROM = NFROM + 1
* AQTOT = AQTOT + FF(IFA, 1)
* END IF
3010 CONTINUE

C
DO 3020 IFO = 1 , NFO
*
* IF ( IFDSTA(1, IFO) .EQ. IUNT
* .AND. IFDSTA(2, IFO) .EQ. I ) THEN
* NFROM = NFROM + 1
* ORGTOT = ORGTOT + FF(IFO, 2)
* END IF
3020 CONTINUE

C
DO 3040 IUA = 1 , NUNIT
*
* NDESTA = MDEST(1, IUA)

C
DO 3030 KDEST = 1 , NDESTA
*
* IF ( IUDSTA(1, KDEST, IUA) .EQ. IUNT
* .AND. IUDSTA(2, KDEST, IUA) .EQ. I ) THEN
* RFX = RFOOT(KDEST, 1, IUA)
* NFROM = NFROM + 1
* AQTOT = AQTOT + FOUT(1, IUA)*RFX
* END IF
3030 CONTINUE
3040 CONTINUE

C
DO 3060 IUO = 1 , NUNIT
*
* NDESTO = MDEST(2, IUO)

C
DO 3050 KDEST = KDESOW , NDESTO
*
* IF ( IUDSTA(1, KDEST, IUO) .EQ. IUNT
* .AND. IUDSTA(2, KDEST, IUO) .EQ. I ) THEN
* RFX = RFOOT(KDEST, 2, IUO)
* NFROM = NFROM + 1
* ORGTOT = ORGTOT + FOUT(2, IUO)*RFX
* END IF
3050 CONTINUE
3060 CONTINUE

C
IFROM = 0
NLINE = 0

C
IFAW = 0
IFOW = 0
IUAW = 0
IUOW = 0
KDESAW = 0
KDESOW = 0

C
3110 CONTINUE

C
DO 3120 KFROM = 1 , MXFROM
*
* FROMID(KFROM) = ' '
3120 CONTINUE

C
NLINE = NLINE + 1
KFROM = 0

C
IF ( NFROM .LE. 0 ) GO TO 3190

C
IF ( IFAW .LE. NFA ) THEN

C
DO 3130 IFA = IFAW , NFA
*
* IFAW = IFA + 1
* IF ( IFDSTA(1, IFA) .EQ. IUNT
* .AND. IFDSTA(2, IFA) .EQ. I ) THEN
* IFROM = IFROM + 1
* KFROM = KFROM + 1
* WRITE(FROMID(KFROM), 5240) IFA, 'FA'
* IF ( IFROM .EQ. NFROM
* .OR. KFROM .EQ. MXFROM ) THEN
* GO TO 3190
* END IF
* END IF
3130 CONTINUE

C
END IF

C
IF ( IFOW .LE. NFO ) THEN

C
DO 3140 IFO = IFOW , NFO
*
* IFOW = IFO + 1

```

```

*       IF (      IFDSTO(1,IFO) .EQ. IUNT
        .AND. IFDSTO(2,IFO) .EQ. I ) THEN
        IFROM = IFROM + 1
        KFROM = KFROM + 1
        WRITE (FROMID(KFROM),5240) IFO,'FO'
*       IF (      IFROM .EQ. NFROM
        .OR. KFROM .EQ. MXFROM ) THEN
        GO TO 3190
        END IF
3140    CONTINUE
C
C       END IF
C
C       IF ( IUAW .LE. NUNIT ) THEN
C
C       DO 3160 IUA = IUAW , NUNIT
        NDESTA = NDEST(1, IUA)
C
C       IF ( KDESAW .LE. NDESTA ) THEN
        DO 3150 KDEST = KDESAW , NDESTA
        KDESAW = KDEST + 1
        IF ( KDESAW .GT. NDESTA ) THEN
            IUAW = IUA + 1
            KDESAW = 1
        END IF
*       IF (      IUDSTA(1,KDEST,IUA) .EQ. IUNT
        .AND. IUDSTA(2,KDEST,IUA) .EQ. I ) THEN
        IFROM = IFROM + 1
        KFROM = KFROM + 1
        IF ( TYPE(IUA) .EQ. 'MS' ) THEN
            TYPEU = 'MA'
        ELSE IF ( TYPE(IUA) .EQ. 'TA' ) THEN
            TYPEU = 'TA'
        ELSE
            TYPEU = 'UA'
        END IF
        WRITE (FROMID(KFROM),5240) ID(IUA),TYPEU
        IF (      IFROM .EQ. NFROM
        .OR. KFROM .EQ. MXFROM ) THEN
            GO TO 3190
        END IF
        END IF
3150    CONTINUE
        END IF
C
C       CONTINUE
3160
C
C       END IF
C
C       IF ( IUOW .LE. NUNIT ) THEN
C
C       DO 3180 IUO = IUOW , NUNIT
        NDESTO = NDEST(2, IUO)
        IF ( KDESOW .LE. NDESTO ) THEN
C
C       DO 3170 KDEST = KDESOW , NDESTO
        KDESOW = KDEST + 1
        IF ( KDESOW .GT. NDESTO ) THEN
            IUOW = IUO + 1
            KDESOW = 1
        END IF
*       IF (      IUDSTO(1,KDEST,IUO) .EQ. IUNT
        .AND. IUDSTO(2,KDEST,IUO) .EQ. I ) THEN
        IFROM = IFROM + 1
        KFROM = KFROM + 1
        IF ( TYPE(IUO) .EQ. 'MS' ) THEN
            TYPEU = 'MO'
        ELSE IF ( TYPE(IUO) .EQ. 'TO' ) THEN
            TYPEU = 'TO'
        ELSE
            TYPEU = 'UO'
        END IF
        WRITE (FROMID(KFROM),5240) ID(IUO),TYPEU
        IF (      IFROM .EQ. NFROM
        .OR. KFROM .EQ. MXFROM ) THEN
            GO TO 3190
        END IF
        END IF
3170    CONTINUE
        END IF
C
C       CONTINUE
3180
C
C       END IF
C
C
C
3190    CONTINUE

```

```

C
C
C       IF ( NLINE .EQ. 1 ) THEN
C
C       T      = TSTG(1, IEXTRA) - TO
        VMIX   = V  (1, IEXTRA)
        VSTL   = VS (1, IEXTRA)
        FA     = FL(1,1, IEXTRA)
        FO     = FL(2,1, IEXTRA)
        FCYC   = FR (1, IEXTRA)
        PHIA   = PHI (1, IEXTRA)
        PHIO   = ( ONE - PHI (1, IEXTRA) )
        PHISX  = PHIS(1, IEXTRA)
C
C
C       WRITE(IFOUT,5220) I ,
*       T ,
*       VMIX,
*       VSTL,
*       FA ,
*       FO ,
*       FCYC,
*       PHIA,
*       PHIO,
*       PHISX,
*       AOTOT,
*       ORGTOT,
*       (FROMID(KFX),KFX = 1 , KFROM )
        IF ( FA .LT. 1.0D-100 ) THEN
            WRITE(IUEERR,*) ' ERROR DETECTED.'
            WRITE(IUEERR,*) ' KEY WORD IS FEED '
            WRITE(IUEERR,*) ' AQUEOUS FLOW RATE IS ZERO.'
            WRITE(IUEERR,*) ' UNIT IS ', ID(IUNT)
            STOP
        END IF
        IF ( FO .LT. 1.0D-100 ) THEN
            WRITE(IUEERR,*) ' ERROR DETECTED.'
            WRITE(IUEERR,*) ' KEY WORD IS FEED '
            WRITE(IUEERR,*) ' ORGANIC FLOW RATE IS ZERO.'
            WRITE(IUEERR,*) ' UNIT IS ', ID(IUNT)
            STOP
        END IF
        ELSE
C
C       WRITE(IFOUT,5230) (FROMID(KFX),KFX = 1 , KFROM )
C
C       END IF
C
C       IF ( IFROM .LT. NFROM ) GO TO 3110
C
C
C
3300    CONTINUE
C
C       END IF
C
C
C
3400 CONTINUE
C
C
C       RETURN
C
C
4010 FORMAT(1H /1H /1H ,A,5X,'TIME:',1P,E12.5,' (HR)')
4020 FORMAT(1H /
*       1H ,'TBP_VOLUME_FRACTION  :',1P,E12.5,' (-)')
4030 FORMAT(1H ,'DBP_CONCENTRATION  :',1P,E12.5,' (G/L)')
4040 FORMAT(1H ,'DEGRADATION_MULTPLIER :',1P,E12.5,' (-)')
C
4100 FORMAT(1H /1H ,5X,A)
4200 FORMAT(1H ,' NO. ',6(3X,A10))
4210 FORMAT(1H ,' ',6(3X,A10))
4220 FORMAT(1H ,17,3X,3X,A8,2X,1P2E13.5,3X,A8,2X,1P2E13.5)
4230 FORMAT(1H ,' NO. ',3X,A10,20(15,'MS',5X))
4240 FORMAT(1H ,' ',3X,A10)
4250 FORMAT(1H ,17,3X,3X,A8,3X,20(2X,A8,2X))
4260 FORMAT(1H ,' REACT ',3X,20(2X,15,'MS',3X))
4270 FORMAT(1H ,' NO. ')
4280 FORMAT(1H ,17,3X,3X,20(1P,E11.3,1X))
C
4300 FORMAT(1H /1H ,A/1H )
4310 FORMAT(1H ,11A)
4315 FORMAT(1H ,13,1X,14,2X,A2,5X,12,1X,13,1X,4(' ',1X,
*       1P2E11.3,0P,3X,13,2X,13,6X,' ',9X,13,2X,13)
4320 FORMAT(1H ,13,1X,14,2X,A2,5X,12,1X,13,1X,1P4E11.3,1X,1P2E11.3,
*       0P,3X,13,2X,13,2X,1PE11.3,3X,13,2X,13)
C
4400 FORMAT(1H /1H /1H ,2A)
C
4411 FORMAT(1H ,A ,2A,2(A8,5X),3(A8,5X))

```

```

4431 FORMAT(1H , I6, 6X, I5, 2X, A2, 1X, I4 , 6X, 1P2E13.5, 3E13.5)
4441 FORMAT(1H , ' . . . . . ' , 6X, 1P2E13.5, 3E13.5)
C
4412 FORMAT(1H , A, 1X, 10(A8, 5X))
4432 FORMAT(1H , I2, 1P10E13.5)
4442 FORMAT(1H , ' . ' , 1P10E13.5)
C
5000 FORMAT(1H1/1H /1H , 'NUMBER_OF_UNIT(S):' , I3, T100, 'DATE ' , A9)
5010 FORMAT(1H /1H /1H , A, 5X, 'TIME:' , 1P, E12.5, ' (HR)')
5100 FORMAT(1H /1H , 'UNIT ID.' I5, 1X, A2, 5X, A)
5200 FORMAT(1H , 'STG' , I2A)
5210 FORMAT(1H , ' ' , I2A)
5220 FORMAT(1H , I2, 1X, F4.1, 1P, 7E11.4, 0PF4.1, 1P2E11.4, 10A6)
5230 FORMAT(1H , 2X, 1X, 4X, 7(11X), 4X, 2(11X), 10A6)
5240 FORMAT(1X, I3, A2)
C
END

```

```

SUBROUTINE INPUT(IN , IUPRT , IUPLT , IPPDIS, TITLE , CTFILE, RXLIM ,
*
* MXTAUC, TMCONB, TMCONC, DTCONS,
*
* MXEXTR, MXDEST, MXCOMP, MXUNIT, MXSTG , MXSTG2, NCHEM ,
*
* MXREAC, MXBLK , NUNIT , NBLK , NEXTRA,
*
* ENDTIM, DTO , DTMAX , DTMIN , DTPRT , DTPLT , TRATE ,
*
* ICALC , ITMAX , EPS , ITMIN , EPSLIM,
*
* ISFIN , FRC , CO , TO , CDBP , FDEG ,
*
* CNAME , FMASS , DECAYH, CVUNIT, CUNIT , CHRIG , CNTBP ,
*
* TFEED ,
*
* NFA , NFO ,
*
* MXFEED, MXTABL, NFFTAB, FFTAB ,
*
* NXFTAB, XFTAB , IFDSTA, IFDSTO,
*
* ID , IREACT, FCHEM , ECHEM , TYPE , MSTG ,
*
* IUDSTA, IUDSTO, MDEST , MEXTRA, TAV , VTOT ,
*
* ACID , ACIDIN, OXID , FOUT , RFOUT , FFOUT ,
*
* IDIST , IEFF , CDIS , CKEQU , EFF ,
*
* TSTG , V , VS , HEIGHT, FR ,
*
* IPOS ,
*
* X )
C
C
C
C*****
C
C READ INPUT DATA FROM WORK FILE
C
C T : T IS INPUT IN CENTIGRADE.
C*****
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER* 72 TITLE
C CHARACTER* 30 CTFILE
C CHARACTER* 33 INFILE
C
C
C INDICES FOR DISTRIBUTION CALCULATION OPTION
C
COMMON /IDISOP/IDCONS, IDEQUI, IDHANF, IDKFK , IDAMIX
C
C
C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAYH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRIG (MXCOMP),
* CNTBP (MXCOMP)
C
C
C DIMENSION NFFTAB( MXFEED, 2),
* FFTAB ( 2 , MXTABL, MXFEED, 2),
* NXFTAB( MXFEED, 2),
* XFTAB (MXCOMP+1, MXTABL, MXFEED, 2),
* IFDSTA(2 , MXFEED ) ,
* IFDSTO(2 , MXFEED )
C
C DIMENSION ID ( MXUNIT)
C DIMENSION IREACT( MXUNIT)
C DIMENSION FCHEM (MXFEAC, 2 , MXUNIT),
* ECHEM (MXFEAC, 2 , MXUNIT)
C
C CHARACTER*2 TYPE ( MXUNIT)
C
C DIMENSION MSTG ( MXUNIT),
* IUDSTA(3 , MXDEST, MXUNIT),
* IUDSTO(3 , MXDEST, MXUNIT)
C DIMENSION MDEST ( 2 , MXUNIT),
* MEXTRA( MXUNIT)
C
C DIMENSION TAV ( MXUNIT),
* VTOT ( MXUNIT),
* ACID ( MXUNIT),
* ACIDIN( MXUNIT),
* OXID ( MXUNIT),
* FOUT ( 2 , MXUNIT),
* RFOUT (MXDEST, 2 , MXUNIT)
C
C DIMENSION IDIST (MXCOMP , MXEXTR)
C DIMENSION IEFF (MXCOMP , MXEXTR)
C
C DIMENSION CDIS (MXCOMP, MXSTG , MXEXTR),
* CKEQU (MXCOMP, MXSTG , MXEXTR),
* EFF (MXCOMP, MXSTG , MXEXTR)
C
C DIMENSION TSTG ( MXSTG , MXEXTR),

```

```

*          V (      MXSTG ,MXEXTR),
*          VS (      MXSTG ,MXEXTR),
*          HEIGHT( MXSTG ,MXEXTR),
*          FR (      MXSTG ,MXEXTR)
C
C DIMENSION IPOS (2 ,MXSTG2,MXUNIT)
C
C DIMENSION DTPRT(100)
C DIMENSION DTPLT(100)
C
C DIMENSION X (MXCOMP,2,MXBLK)
C
C DIMENSION TMCONB(MXTAUC),TMCONC(MXTAUC),DTCONS(MXTAUC)
C DIMENSION FFOUT(2,2,MXUNIT)
C
C COMMON /IXCOMP/NCHEMO,
*          IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*          IOHNO2,IOHYD ,IOHAN ,IOPIUS ,IOPIU6 ,
*          IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*          IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6,IOTCP4,
*          IOTCZR,IOSR ,IORUDI,IORUTR,IORUNI,
*          IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
*          IOCM
C
C PARAMETER (MXWORK=2000)
C
C DIMENSION DATWRK(MXWORK)
C DIMENSION IDTWRK(MXWORK)
C CHARACTER*8 CNAM
C CHARACTER*8 CUNT
C CHARACTER*8 KEYWD
C CHARACTER*12 NAMEOFCARD
C INTEGER NUMBOFCARD
C DIMENSION ADDMC(10)
C
C DATA ZERO / 0.000 /
C DATA ONE / 1.000 /
C DO J = 1 , 10
C   ADDMC(J) = 0.
C ENDDO
C
C INFILE = ' '
C
C NUNIT = 0
C NEXTRA = 0
C NBLK = 0
C NFA = 0
C NFO = 0
C
C ICRDFD = 0
C ICRDIN = 0
C
C --- READ INPUT DATA CARDS ---
C
C TITLE CARD (FIRST CARD).
C
C READ(IN,'(A)') TITLE
C NUMBOFCARD = 1
C WRITE(IUPRT,'(1H ,A)') TITLE
C NAMEOFCARD = 'TITLE'
C
C 200 CONTINUE
C WRITE(0,'(8X,13,5X,2A)')
C > NUMBOFCARD,NAMEOFCARD,'CARD WAS READ FROM THE INPUT NORMALLY.'
C
C DO 201 I = 1 , MXWORK
C   DATWRK(I) = 0.000
C   IDTWRK(I) = 0
C 201 CONTINUE
C RUTOT = 0.000
C
C READ(IN,'(A4)',END=900) KEYWD
C READ(IN,'(I)') NUMBOFCARD
C WRITE(IUPRT,'(1H ,A)') KEYWD
C
C CONTROL CARD.
C
C IF ( KEYWD .EQ. 'CONT' ) THEN
C   READ(IN,*) ICALC,ISFIN,ENDTIM,DTO

```

```

IF ( ICALC .NE. 0 ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' CALCULATION OPTION NOT AVAILABEL.'
  WRITE(IUPRT,*) ' ICALC IS ',ICALC
  STOP
END IF
C
IF ( ISFIN .NE. 0 .AND. ISFIN .NE. 1 ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' ISFIN IS INVALID.'
  WRITE(IUPRT,*) ' ISFIN(INPUT) IS ',ISFIN
  STOP
END IF
C
IF ( ENDTIM .LE. ZERO ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' ENDTIM MUST BE POSITIVE.'
  WRITE(IUPRT,*) ' ENDTIM(INPUT) IS ',ENDTIM
  STOP
END IF
C
IF ( DTO .LE. ZERO ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' DTO MUST BE POSITIVE.'
  WRITE(IUPRT,*) ' DTO(INPUT) IS ',DTO
  STOP
END IF
C
DTPRT(1) = ENDTIM
DTPLT(1) = ENDTIM
C
NAMEOFCARD = 'CONTROL'
GO TO 200
END IF
C
-----
C STAGE CARD.
C
IF ( KEYWD .EQ. 'STAG' ) THEN
  READ(IN,*) N
  IF ( N .GT. 999 .OR. N .LT. 1 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' STAGE ID MUST BE BETWEEN 1 TO 999.'
    WRITE(IUPRT,*) ' INPUT IS ',N
    STOP
  END IF
C
  READ(IN,*) MSTGX,ICDA1,ICDA2,ICD01,ICD02,IRE
C
  IDEF = 1
  IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
C
  NEXTRA = NEXTRA + 1
C
  IF ( MSTGX .GT. MXSTG ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TOO MANY STAGES IN UNIT NO.',N
    WRITE(IUPRT,*) ' INPUT IS ',MSTGX
    WRITE(IUPRT,*) ' LIMIT IS ',MXSTG
    STOP
  END IF
C
  IF ( NEXTRA .GT. MXEXTR ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TOO MANY EXTRACTION UNITS.'
    WRITE(IUPRT,*) ' NUMBER OF UNITS(INPUT) IS ',NEXTRA
    WRITE(IUPRT,*) ' NUMBER OF UNITS(LIMIT) IS ',MXEXTR
    STOP
  END IF
C
  IF ( IRE .GT. 4 .OR. IRE .LT. 0 ) THEN
    IRE = 0
  END IF
C
  KDEST = 1
C
  TYPE ( IUNT) = 'MS'
  ID ( IUNT) = N
  IREACT( IUNT) = IRE
C
  MSTG ( IUNT) = MSTGX

```

```

IUDSTA(1,KDEST,IUNT) = ICD11
IUDSTA(2,KDEST,IUNT) = ICD12
IUDSTO(1,KDEST,IUNT) = ICD01
IUDSTO(2,KDEST,IUNT) = ICD02
MDEST (1 ,IUNT) = 1
MDEST (2 ,IUNT) = 1
MEXTRA( IUNT) = NEXTRA
TAV ( IUNT) = 25.000 + TO
VTOT ( IUNT) = 0.000
ACID ( IUNT) = 0.000
OXID ( IUNT) = 0.000
FOUT (1 ,IUNT) = 0.000
RFOUT (KDEST,1,IUNT) = 1.000
FOUT (2 ,IUNT) = 0.000
RFOUT (KDEST,2,IUNT) = 1.000

C
C
NSTG = MSTG(IUNT)

C
C
--- MIXER-SETTLER ---

DO 110 I = 1 , NSTG
  IPOS(1,2*I-1,IUNT) = NBLK + 2*I
  IPOS(1,2*I ,IUNT) = NBLK + 2*I - 1
  IPOS(2,2*I-1,IUNT) = NBLK + 2*I
  IPOS(2,2*I ,IUNT) = NBLK + 2*I + 1
110 CONTINUE
C
NBLK = NBLK + 2*NSTG + 1

C
IF ( NBLK .GT. MXBLK ) THEN
  WRITE(IUPRT,*) ' TOO MANY TOTAL STAGES IN SYSTEM'
  WRITE(IUPRT,*) ' UNIT NO. ',IUNT
  WRITE(IUPRT,*) ' INPUT IS ',NBLK
  WRITE(IUPRT,*) ' LIMIT IS ',MXBLK
  STOP
END IF

C
C
NAMEOFCARD = 'STAGE'
GO TO 200
END IF

C
C
-----
C
TANK CARD.

C
IF ( KEYWD .EQ. 'TANK' ) THEN
  READ(IN,*) N
  IF ( N .GT. 999 .OR. N .LT. 1 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TANK ID MUST BE BETWEEN 1 TO 999.'
    WRITE(IUPRT,*) ' INPUT IS ',N
    STOP
  END IF

C
  READ(IN,*) IPHX,VTX,ACIDX,ACIDIX,OXIDX,FOUTX,
*          ICD11,ICD12,
*          FF2,ICD21,ICD22

C
  IDEF = 1
  IUNT = IDCONV(IUPRT,MXUNIT,NUNIT,ID,N,IDEF,KEYWD)

C
  IF ( IPHX .NE. -1 .AND. IPHX .NE. 1 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TANK PHASE SPECIFIER IS INVALID.'
    WRITE(IUPRT,*) ' IPHX ',IPHX
    STOP
  END IF

C
  IF ( VTX .LE. ZERO ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TANK VOLUME IS INVALID.'
    WRITE(IUPRT,*) ' VTX ',VTX
    STOP
  END IF

C
  IF ( ACIDX .GT. ZERO .AND. FOUTX .LE. ZERO ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' TANK FLOW RATE IS INVALID.'
    WRITE(IUPRT,*) ' FOUTX ',FOUTX
    STOP
  END IF

C

```

```

IF ( IPHX .EQ. 1 ) THEN
  IF ( ACIDX .LT. 0.000 .OR. ACIDX .GT. 13.500 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' ACID CONCENTRATION IS INVALID.'
    WRITE(IUPRT,*) ' ACIDX ',ACIDX
    STOP
  END IF

C
  IF ( ACIDIX .EQ. 100.00 ) THEN

C
  ELSE IF ( ACIDIX .LT. 0.000 .OR. ACIDIX .GT. 13.500 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' ACID CONCENTRATION IS INVALID.'
    WRITE(IUPRT,*) ' ACIDIX ',ACIDIX
    STOP
  END IF

C
  IF ( OXIDX .LT. 0.000 .OR. OXIDX .GT. ONE ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' OXIDATION RATIO IS INVALID.'
    WRITE(IUPRT,*) ' OXIDX ',OXIDX
    STOP
  END IF

C
  ELSE

C
  ACIDX = ZERO

C
  END IF

C
  IF ( FF2 .LT. ZERO ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' FLOW RATE RATIO IS INVALID.'
    WRITE(IUPRT,*) ' FF2 ',FF2
    STOP
  END IF

C
  MSTGX = 1

C
  ID ( IUNT) = N
  IREACT( IUNT) = 0

C
  MSTG ( IUNT) = MSTGX
  MEXTRA( IUNT) = 0
  TAV ( IUNT) = 25.000 + TO
  VTOT ( IUNT) = VTX
  ACID ( IUNT) = ACIDX
  ACIDIX( IUNT) = ACIDIX
  OXID ( IUNT) = OXIDX

C
  IF ( IPHX .EQ. 1 ) THEN

C
  TYPE ( IUNT) = 'TA'

C
  MDEST (1 ,IUNT) = 2
  FOUT (1 ,IUNT) = FOUTX
  KDEST = 1
  IUDSTA(1,KDEST,IUNT) = ICD11
  IUDSTA(2,KDEST,IUNT) = ICD12
  FFOUT (KDEST,1,IUNT) = ZERO
  RFOUT (KDEST,1,IUNT) = ZERO
  KDEST = 2
  IUDSTA(1,KDEST,IUNT) = ICD21
  IUDSTA(2,KDEST,IUNT) = ICD22
  FFOUT (KDEST,1,IUNT) = FF2
  RFOUT (KDEST,1,IUNT) = ZERO

C
  MDEST (2 ,IUNT) = 0
  FOUT (2 ,IUNT) = ZERO
  KDEST = 1
  IUDSTO(1,KDEST,IUNT) = 0
  IUDSTO(2,KDEST,IUNT) = 0
  FFOUT (KDEST,2,IUNT) = ZERO
  RFOUT (KDEST,2,IUNT) = ZERO
  KDEST = 2
  IUDSTO(1,KDEST,IUNT) = 0
  IUDSTO(2,KDEST,IUNT) = 0
  FFOUT (KDEST,2,IUNT) = ZERO
  RFOUT (KDEST,2,IUNT) = ZERO

C
  ELSE

C

```

```

C      TYPE (      IUNT) = 'TO'
C      MDEST (1      ,IUNT) = 0
C      FOUT (1      ,IUNT) = ZERO
C      KDEST      = 1
C      IUDSTA(1,KDEST,IUNT) = 0
C      IUDSTA(2,KDEST,IUNT) = 0
C      FFOUT (KDEST,1,IUNT) = ZERO
C      RFOUT (KDEST,1,IUNT) = ZERO
C      KDEST      = 2
C      IUDSTA(1,KDEST,IUNT) = 0
C      IUDSTA(2,KDEST,IUNT) = 0
C      FFOUT (KDEST,1,IUNT) = ZERO
C      RFOUT (KDEST,1,IUNT) = ZERO
C
C      MDEST (2      ,IUNT) = 2
C      FOUT (2      ,IUNT) = FOUTX
C      KDEST      = 1
C      IUDSTO(1,KDEST,IUNT) = 1CD11
C      IUDSTO(2,KDEST,IUNT) = 1CD12
C      FFOUT (KDEST,2,IUNT) = ZERO
C      RFOUT (KDEST,2,IUNT) = ZERO
C      KDEST      = 2
C      IUDSTO(1,KDEST,IUNT) = 1CD21
C      IUDSTO(2,KDEST,IUNT) = 1CD22
C      FFOUT (KDEST,2,IUNT) = FF2
C      RFOUT (KDEST,2,IUNT) = ZERO
C
C      END IF
C
C      NSTG = MSTG(IUNT)
C
C      --- TANK ---
C
C      DO 130 I = 1 , NSTG
C        IPOS(1,I,IUNT) = NBLK + I
C        IPOS(2,I,IUNT) = NBLK + I
130    CONTINUE
C
C      NBLK = NBLK + NSTG
C
C      IF ( NBLK .GT. MXBLK ) THEN
C        WRITE(IUPRT,*) ' TOO MANY TOTAL STAGES IN SYSTEM'
C        WRITE(IUPRT,*) ' UNIT NO. ',IUNT
C        WRITE(IUPRT,*) ' INPUT IS ',NBLK
C        WRITE(IUPRT,*) ' LIMIT IS ',MXBLK
C        STOP
C      END IF
C
C      NAMEOFCARD = 'TANK'
C      GO TO 200
C    END IF
C
C    -----
C    ADDMIC CARD.
C
C    IF ( KEYWD .EQ. 'ADDM' ) THEN
C
C      IF ( 1CRDFD .NE. 0 ) THEN
C        WRITE(IUPRT,*) ' ERROR DETECTED.'
C        WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C        WRITE(IUPRT,*) ' ADDMIC MUST BE DEFINED'//
C        *      ' BEFORE FEEDS CARD.'
C        STOP
C      END IF
C
C      IF ( 1CRDIN .NE. 0 ) THEN
C        WRITE(IUPRT,*) ' ERROR DETECTED.'
C        WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C        WRITE(IUPRT,*) ' ADDMIC MUST BE DEFINED'//
C        *      ' BEFORE INITIAL CARD.'
C        STOP
C      END IF
C
C      READ(IN,*) J,CNAM,CHRGX,CNTBPX,CVUNT,CUNT
C
C      IF ( J .LT. 32 .OR. J .GT. MXCOMP ) THEN
C        WRITE(IUPRT,*) ' ERROR DETECTED.'
C        WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C        WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
C        STOP
C      END IF
C
C      IF ( ADDMC(J-31) .NE. 0. ) THEN
C        WRITE(IUPRT,*) ' ERROR DETECTED.'
C        WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD

```

```

      WRITE(IUPRT,*) ' J VALUE IS ALREADY DEFINED.'
      STOP
    END IF
    ADDMC (J-31) = 1.
C
C    IF ( CHRGX .LT. 0. ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C      WRITE(IUPRT,*) ' CHRGX MUST HAVE POSITIVE VALUE.'
C      STOP
    END IF
C
C    IF ( CNTBPX .LT. 0. ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C      WRITE(IUPRT,*) ' CNTBPX MUST HAVE POSITIVE VALUE.'
C      STOP
    END IF
C
C    IF ( CVUNT .LT. 0. ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C      WRITE(IUPRT,*) ' CVUNIT MUST HAVE POSITIVE VALUE.'
C      STOP
    END IF
C
C    CNAM (J) = CNAM
C    CHRG (J) = CHRGX
C    CNTBP (J) = CNTBPX
C
C    CVUNT(J) = CVUNT
C    FMASS (J) = CVUNT
C    CUNT (J) = CUNT
C
C    NAMEOFCARD = 'ADDMIC'
C    GO TO 200
C  END IF
C
C  -----
C  TBP CARD.
C
C  IF ( KEYWD .EQ. 'TBP' ) THEN
C    READ(IN,*) FRC
C
C    IF ( FRC .LT. 0. .OR. FRC .GT. 1.0 ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C      WRITE(IUPRT,*) ' FRC VALUE SPECIFIED IS INVALID.'
C      STOP
C    END IF
C
C    NAMEOFCARD = 'TBP'
C    GO TO 200
C  END IF
C
C  -----
C  DBP CARD.
C
C  IF ( KEYWD .EQ. 'DBP' ) THEN
C    READ(IN,*) CDBP
C    IF ( CDBP .LT. 0. ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C      WRITE(IUPRT,*) ' CDBP MUST HAVE POSITIVE VALUE.'
C      STOP
C    END IF
C
C    NAMEOFCARD = 'DBP'
C    GO TO 200
C  END IF
C
C  -----
C  DEGRADATION CARD.
C
C  IF ( KEYWD .EQ. 'DEGR' ) THEN
C    READ(IN,*) FDEG
C    NAMEOFCARD = 'DEGRADATION'
C    GO TO 200
C  END IF
C
C  -----
C  DHEAT CARD.
C
C  IF ( KEYWD .EQ. 'DHEA' ) THEN
C    READ(IN,*) J
C
C    IF ( J .LT. 1 .OR. J .GT. MXCOMP ) THEN
C      WRITE(IUPRT,*) ' ERROR DETECTED.'
C      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD

```

```

        WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
        STOP
    END IF
C
    READ(IN,*) DECAYH(J)
    NAMEOFCARD = 'DHEAT'
    GO TO 200
    END IF
C
-----
C
    CHARGE CARD.
C
    IF ( KEYWD .EQ. 'CHAR' ) THEN
        READ(IN,*) J
C
        IF ( J .LT. 1 .OR. J .GT. 31 ) THEN
            WRITE(IUPRT,*) ' ERROR DETECTED.'
            WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
            WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
            STOP
        END IF
C
        READ(IN,*) CHR(J) , CNTBP(J)
C
        IF ( CHR(J) .LT. 0. ) THEN
            WRITE(IUPRT,*) ' ERROR DETECTED.'
            WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
            WRITE(IUPRT,*) ' CHR MUST HAVE POSITIVE VALUE.'
            STOP
        END IF
C
        IF ( CNTBP(J) .LT. 0. ) THEN
            WRITE(IUPRT,*) ' ERROR DETECTED.'
            WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
            WRITE(IUPRT,*) ' CNTBP MUST HAVE POSITIVE VALUE.'
            STOP
        END IF
C
        NAMEOFCARD = 'CHARGE'
        GO TO 200
    END IF
C
-----
C
    VOLUME CARD.
C
    IF ( KEYWD .EQ. 'VOLLU' ) THEN
        READ(IN,*) N
C
        IDEF = 0
        IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
        IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
*           ID , IUNT ,KEYWD ,MEXTRA)
        NSTG = MSTG(IUNT)
        READ(IN,*) (DATWRK(I),I=1,2*NSTG )
C
        DO 210 I = 1 , NSTG
C
            IF ( DATWRK(I) .LE. ZERO ) THEN
                WRITE(IUPRT,*) ' ERROR DETECTED.'
                WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
                WRITE(IUPRT,*) ' MIXER VOLUME MUST BE POSITIVE.'
                WRITE(IUPRT,*) ' BANK, STAGE =', ID(IUNT),I
                STOP
            END IF
C
            IF ( DATWRK(NSTG+1) .LE. ZERO ) THEN
                WRITE(IUPRT,*) ' ERROR DETECTED.'
                WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
                WRITE(IUPRT,*) ' SETTLER VOLUME MUST BE POSITIVE.'
                WRITE(IUPRT,*) ' BANK, STAGE =', ID(IUNT),I
                STOP
            END IF
C
            V (I,IEXTRA) = DATWRK( I)
            VS(I,IEXTRA) = DATWRK(NSTG+I)
210    CONTINUE
C
        NAMEOFCARD = 'VOLUME'
        GO TO 200
    END IF
C
-----
C
    HEIGHT CARD.
C
    IF ( KEYWD .EQ. 'HEIG' ) THEN
        READ(IN,*) N
C
        IDEF = 0

```

```

        IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
        IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
*           ID , IUNT ,KEYWD ,MEXTRA)
        NSTG = MSTG(IUNT)
        READ(IN,*) (DATWRK(I),I=1,NSTG )
C
        DO 220 I = 1 , NSTG
C
            IF ( DATWRK(I) .LE. ZERO .OR. DATWRK(I) .GE. 1.000 ) THEN
                WRITE(IUPRT,*) ' ERROR DETECTED.'
                WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
                WRITE(IUPRT,*) ' HEIGHT MUST BE POSITIVE' //
*           ' AND LESS THAN UNITY.'
                WRITE(IUPRT,*) ' BANK, STAGE =', ID(IUNT),I
                STOP
            END IF
            HEIGHT(I,IEXTRA) = DATWRK(I)
220    CONTINUE
C
        NAMEOFCARD = 'HEIGHT'
        GO TO 200
    END IF
C
-----
C
    FEEDS CARD.
C
    IF ( KEYWD .EQ. 'FEED' ) THEN
        ICRDFD = 1
        READ(IN,*) N
C
        IDEF = 0
        IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
        READ(IN,*) I ,NSET, TIM, FF, (DATWRK(J),J=1,MXCOMP) ,RUTOT
C
        IF ( NSET .GT. MXTABL ) THEN
            WRITE(IUPRT,*) ' TOO LONG FEED TABLE.'
            WRITE(IUPRT,*) ' LENGTH(INPUT) IS ',NSET
            WRITE(IUPRT,*) ' LENGTH(LIMIT) IS ',MXTABL
            STOP
        END IF
C
        IF ( I .GT. 0 ) THEN
            IPH = 1
            NFA = NFA + 1
            NFX = NFA
C
            IF ( NFA .GT. MXFEED ) THEN
                WRITE(IUPRT,*) ' TOO MANY FEED POINTS(AQUEOUS).'
                WRITE(IUPRT,*) ' MAXIMUM IS ',MXFEED
                STOP
            END IF
            IFDSTA(1,NFA) = N
            IFDSTA(2,NFA) = IABS(1)
C
        ELSE
            IPH = 2
            NFO = NFO + 1
            NFX = NFO
C
            IF ( NFO .GT. MXFEED ) THEN
                WRITE(IUPRT,*) ' TOO MANY FEED POINTS(ORGANIC).'
                WRITE(IUPRT,*) ' MAXIMUM IS ',MXFEED
                STOP
            END IF
            IFDSTO(1,NFO) = N
            IFDSTO(2,NFO) = IABS(1)
C
        END IF
C
        NSTG = MSTG(IUNT)
        I = IABS(I)
C
        IF ( I .LT. 0 .OR. I .GT. NSTG ) THEN
            WRITE(IUPRT,*) ' ERROR DETECTED.'
            WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
            WRITE(IUPRT,*) ' I VALUE SPECIFIED IS INVALID.'
            WRITE(IUPRT,*) ' BANK, I =', ID(IUNT),I
            STOP
        END IF
C
        NFFTAB(NFX,IPH) = NSET
        NXFTAB(NFX,IPH) = NSET
C
        LSET = 0

```

```

TIMO = 0.000
C
C
230 CONTINUE
LSET = LSET + 1
IF ( TIM .LT. TIMO ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' TIM IS INVALID.'
WRITE(IUPRT,*) ' BANK, I, TIM =', ID(IUNT), I, TIM
STOP
END IF
C
FFTAB(1,LSET,NFX,IPH) = TIM
XFTAB(1,LSET,NFX,IPH) = TIM
C
IF ( FF .LT. 0.000 ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' FLOW RATE IS INVALID.'
WRITE(IUPRT,*) ' BANK, I, FLOW RATE =', ID(IUNT), I, FF
STOP
END IF
FFTAB(2,LSET,NFX,IPH) = FF
C
DO 240 J = 1 , MXCOMP
DATAJN = DATWRK(J)
IF ( DATAJN .LT. ZERO ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' NEGATIVE CONCENTRATION.'
WRITE(IUPRT,*) ' BANK, STAGE, J =', ID(IUNT), I, J
STOP
END IF
XFTAB(J+1,LSET,NFX,IPH) = DATAJN / CVUNIT(J)
240 CONTINUE
RUTOT = RUTOT/CVUNIT(IORUMO)
XH = XFTAB(1+IORNO3,LSET,NFX,IPH)
IF ( XH .LT. 1.0D-3 ) THEN
XH = 1.0D-3
END IF
CALL RUCOMP(XH ,RRUDI ,RRUTR ,RRUNI ,RRUMO )
XFTAB(1+IORUDI,LSET,NFX,IPH)
* = XFTAB(1+IORUDI,LSET,NFX,IPH)+RRUDI*RUTOT
XFTAB(1+IORUTR,LSET,NFX,IPH)
* = XFTAB(1+IORUTR,LSET,NFX,IPH)+RRUTR*RUTOT
XFTAB(1+IORUNI,LSET,NFX,IPH)
* = XFTAB(1+IORUNI,LSET,NFX,IPH)+RRUNI*RUTOT
XFTAB(1+IORUMO,LSET,NFX,IPH)
* = XFTAB(1+IORUMO,LSET,NFX,IPH)+RRUMO*RUTOT
C
IF ( LSET .LT. NSET ) THEN
TIMO = TIM
READ(IN,*) TIM,FF, (DATWRK(J),J=1,MXCOMP), RUTOT
GO TO 230
END IF
C
NAMEOFCARD = 'FEEDS'
GO TO 200
END IF
C
-----
C RECYCLE CARD.
IF ( KEYWD .EQ. 'RECY' ) THEN
READ(IN,*) N
C
IDEF = 0
IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
* ID ,IUNT ,KEYWD ,MEXTRA)
NSTG = MSTG(IUNT)
READ(IN,*) (DATWRK(I),I=1,NSTG )
C
DO 250 I = 1 , NSTG
C
IF ( DATWRK(I) .LT. ZERO ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' DISTRIBUTION COEFF. MUST BE POSITIVE.'
WRITE(IUPRT,*) ' BANK, STAGE, J =', ID(IUNT), I, J
STOP
END IF
310 CONTINUE
C
NAMEOFCARD = 'CDIST'
GO TO 200
END IF
C
-----

```

```

WRITE(IUPRT,*) ' RECYCLE FLOW MUST NOT BE NEGATIVE.'
WRITE(IUPRT,*) ' BANK, STAGE =', ID(IUNT), I
STOP
END IF
C
FR(I,IEXTRA) = DATWRK(I)
250 CONTINUE
C
NAMEOFCARD = 'RECYCLE'
GO TO 200
END IF
C
-----
C EQUILIBRIUM CARD.
IF ( KEYWD .EQ. 'EQUI' ) THEN
READ(IN,*) N
C
IDEF = 0
IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
* ID ,IUNT ,KEYWD ,MEXTRA)
NSTG = MSTG(IUNT)
READ(IN,*) J, (DATWRK(I),I=1,NSTG )
C
IF ( J .LT. 0 .OR. J .GT. MXCOMP ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
STOP
END IF
C
DO 260 I = 1 , NSTG
C
IF ( DATWRK(I) .LT. ZERO ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' EQUILIBRIUM CONS. MUST BE POSITIVE.'
WRITE(IUPRT,*) ' BANK, STAGE, J =', ID(IUNT), I, J
STOP
END IF
CKEQU(J,I,IEXTRA) = DATWRK(I)
260 CONTINUE
C
NAMEOFCARD = 'EQUILIBRIUM'
GO TO 200
END IF
C
-----
C CDIST CARD.
IF ( KEYWD .EQ. 'CDIS' ) THEN
READ(IN,*) N
C
IDEF = 0
IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
* ID ,IUNT ,KEYWD ,MEXTRA)
NSTG = MSTG(IUNT)
READ(IN,*) J, (DATWRK(I),I=1,NSTG )
C
IF ( J .LT. 0 .OR. J .GT. MXCOMP ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
STOP
END IF
C
DO 310 I = 1 , NSTG
C
IF ( DATWRK(I) .LT. ZERO ) THEN
WRITE(IUPRT,*) ' ERROR DETECTED.'
WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' DISTRIBUTION COEFF. MUST BE POSITIVE.'
WRITE(IUPRT,*) ' BANK, STAGE, J =', ID(IUNT), I, J
STOP
END IF
310 CONTINUE
C
NAMEOFCARD = 'CDIST'
GO TO 200
END IF
C
-----

```



```

C SDIST CARD.
C
IF ( KEYWD .EQ. 'SDIS' ) THEN
  READ(IN,*) N
C
  IDEF = 0
  IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
  IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
*           ID ,IUNT ,KEYWD ,MEXTRA)
  NSTG = MSTG(IUNT)
  READ(IN,*) (IDTWK(J),J=1,MXCOMP )
C
DO 320 J = 1 , MXCOMP
C
  IF (
    IDTWK(J) .NE. IDKFK
*   .AND. IDTWK(J) .NE. IDHANF
*   .AND. IDTWK(J) .NE. IDCONS
*   .AND. IDTWK(J) .NE. IDEQU1
*   .AND. IDTWK(J) .NE. IDAMIX
*   ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' SDIST VALUE IS INVALID.'
    WRITE(IUPRT,*) ' BANK,J =',ID(IUNT),J
    STOP
  END IF
  IDIST (J,IEXTRA) = IDTWK(J)
320 CONTINUE
C
NAMEOFCARD = 'SDIST'
GO TO 200
END IF
C
-----
C REACT CARD.
C
IF ( KEYWD .EQ. 'REAC' ) THEN
  READ(IN,*) N
C
  IDEF = 0
  IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
  READ(IN,*) J,FCEMXX
C
IF ( IABS(J) .EQ. 0 .OR. IABS(J) .GT. MXREAC ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
  STOP
END IF
C
IF ( FCEMXX .LT. ZERO ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' FCEM VALUE SPECIFIED IS INVALID.'
  STOP
END IF
C
IF ( J .GT. 0 ) THEN
  FCEM( J,1,IUNT) = FCEMXX
ELSE
  FCEM(-J,2,IUNT) = FCEMXX
END IF
C
NAMEOFCARD = 'REACT'
GO TO 200
END IF
C
-----
C EFFICIENCY CARD.
C
IF ( KEYWD .EQ. 'EFFI' ) THEN
  READ(IN,*) N
C
  IDEF = 0
  IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
  IEXTRA = ISEXTR(IUPRT ,MXUNIT,NUNIT ,
*           ID ,IUNT ,KEYWD ,MEXTRA)
  NSTG = MSTG(IUNT)
  READ(IN,*) J,IEFF,(DATWRK(1),I=1,NSTG )
C
IF ( J .LT. 0 .OR. J .GT. MXCOMP ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
  STOP
END IF

```

```

C
IF ( IEF .NE. -1 .AND. IEF .NE. 1 ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' IEFF IS INVALID.'
  WRITE(IUPRT,*) ' IEFF(INPUT) IS ',IEF
  STOP
END IF
C
IF ( IEF .EQ. 1 ) THEN
  IF ( J .EQ. IOTCU6 .OR.
*     J .EQ. IOTCP4 .OR.
*     J .EQ. IOTCZR ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' EFFI FOR AQU. PHASE IS NOT PERMITTED.'
    STOP
  END IF
END IF
C
IEFF(J,IEXTRA) = IEF
C
DO 330 I = 1 , NSTG
C
  IF ( DATWRK(I) .LE. 0.0D0 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' EFF VALUE SPECIFIED IS NOT POSITIVE.'
    STOP
  END IF
C
  EFF(J,I,IEXTRA) = DATWRK(I)
330 CONTINUE
C
NAMEOFCARD = 'EFFICIENCY'
GO TO 200
END IF
C
-----
C INITIAL CARD.
C
IF ( KEYWD .EQ. 'INIT' ) THEN
  ICRDIN = 1
  READ(IN,*) N
C
IF ( N .EQ. 0 ) THEN
  READ(IN,*) INFILE
  NAMEOFCARD = 'INITIAL'
  GO TO 200
END IF
C
IDEF = 0
IUNT = IDCONV(IUPRT ,MXUNIT,NUNIT ,ID ,N ,IDEF ,KEYWD )
NSTG = MSTG(IUNT)
READ(IN,*) J,M,(DATWRK(1),I=1,NSTG )
C
IF ( J .LT. 0 .OR. J .GT. MXCOMP+1 ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' J VALUE SPECIFIED IS INVALID.'
  STOP
END IF
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
  IF ( M .LT. 0 .OR. M .GT. 4 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' M VALUE SPECIFIED IS INVALID.'
    STOP
  END IF
END IF
C
IF ( TYPE(IUNT) .EQ. 'TA' .OR. TYPE(IUNT) .EQ. 'TO' ) THEN
  IF ( M .LT. 0 .OR. M .GT. 2 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' M VALUE SPECIFIED IS INVALID.'
    STOP
  END IF
END IF
C
DO 340 I = 1 , NSTG

```

```

IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
  IM = 2*I - 1
  IS = 2*I
  IND = IPOS(1,IM ,IUNT)
  INDSA = IPOS(1,IS ,IUNT)
  INDSO = IPOS(2,IS ,IUNT)
ELSE
  IM = I
  IS = I
  IND = IPOS(1,IM ,IUNT)
  INDSA = IPOS(1,IS ,IUNT)
  INDSO = IPOS(2,IS ,IUNT)
END IF
C
IF ( M .EQ. 0 ) THEN
  IPH = 1
  INDX = IND
ELSE IF ( M .EQ. 1 ) THEN
  IPH = 1
  INDX = IND
ELSE IF ( M .EQ. 2 ) THEN
  IPH = 2
  INDX = IND
ELSE IF ( M .EQ. 3 ) THEN
  IPH = 1
  INDX = INDSA
ELSE
  IPH = 2
  INDX = INDSO
END IF
C
DATAAXN = DATWRK(1)
C
IF ( DATAAXN .LT. ZERO ) THEN
  WRITE(IUPRT,*) ' ERROR DETECTED.'
  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
  WRITE(IUPRT,*) ' NEGATIVE CONCENTRATION.'
  WRITE(IUPRT,*) ' BANK, STAGE, J =', ID(IUNT), I, J
  STOP
END IF
C
IF ( J .LE. MXCOMP ) THEN
C
  X(J, IPH, INDX) = DATAAXN / CVUNIT(J)
C
ELSE
  RUTOT = DATAAXN/CVUNIT(IORUMO)
C
  XH = X(10HNO3, IPH, INDX)
  IF ( XH .LT. 1.0D-3 ) THEN
    XH = 1.0D-3
  END IF
C
  CALL RUCOMP(XH ,RRUDI ,RRUTR ,RRUNI ,RRUMO )
C
  X(IORUDI, IPH, INDX) = X(IORUDI, IPH, INDX) + RRUDI*RUTOT
  X(IORUTR, IPH, INDX) = X(IORUTR, IPH, INDX) + RRUTR*RUTOT
  X(IORUNI, IPH, INDX) = X(IORUNI, IPH, INDX) + RRUNI*RUTOT
  X(IORUMO, IPH, INDX) = X(IORUMO, IPH, INDX) + RRUMO*RUTOT
C
END IF
C
IF ( M .EQ. 0 ) THEN
  IPH = 1
  INDX = INDSA
  IF ( J .LE. MXCOMP ) THEN
C
    X(J, IPH, INDX) = DATAAXN / CVUNIT(J)
C
  ELSE
    RUTOT = DATAAXN/CVUNIT(IORUMO)
C
    XH = X(10HNO3, IPH, INDX)
    IF ( XH .LT. 1.0D-3 ) THEN
      XH = 1.0D-3
    END IF
C
    CALL RUCOMP(XH ,RRUDI ,RRUTR ,RRUNI ,RRUMO )
C
    X(IORUDI, IPH, INDX) = X(IORUDI, IPH, INDX) + RRUDI*RUTOT
    X(IORUTR, IPH, INDX) = X(IORUTR, IPH, INDX) + RRUTR*RUTOT
    X(IORUNI, IPH, INDX) = X(IORUNI, IPH, INDX) + RRUNI*RUTOT
    X(IORUMO, IPH, INDX) = X(IORUMO, IPH, INDX) + RRUMO*RUTOT
C
  END IF
END IF
C
340 CONTINUE
C

```

```

C
  NAMEOFCARD = ' INITIAL'
  GO TO 200
END IF
C
C-----
C PRINT CARD.
C
IF ( KEYWD .EQ. 'PRIN' ) THEN
  READ(IN,*) N, (DTPRT(I), I=1,N)
  IF ( N .GT. 99 .OR. N .LT. 1 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' PRINTOUT TIMES MUST BE BETWEEN 1 TO 99.'
    WRITE(IUPRT,*) ' N(INPUT) IS ',N
  END IF
  DO I=1,N
    IF ( DTPRT(I) .GE. ENDTIM ) DTPRT(I) = ENDTIM
    IF ( DTPRT(I) .LE. 0.0D0 ) THEN
      WRITE(IUPRT,*) ' ERROR DETECTED.'
      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
      WRITE(IUPRT,*) ' DTPRT MUST HAVE POSITIVE VALUE.'
      WRITE(IUPRT,*) ' DTPRT(INPUT) IS ',DTPRT(I)
      STOP
    END IF
  ENDDO
  DTPRT(N+1) = ENDTIM
  NAMEOFCARD = 'PRINT'
  GO TO 200
END IF
C
C-----
C PLOT CARD.
C
IF ( KEYWD .EQ. 'PLOT' ) THEN
  READ(IN,*) N, (DTPLT(I), I=1,N)
  IF ( N .GT. 99 .OR. N .LT. 1 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' PLOTOUT TIMES MUST BE BETWEEN 1 TO 99.'
    WRITE(IUPRT,*) ' N(INPUT) IS ',N
  END IF
  DO I=1,N
    IF ( DTPLT(I) .GE. ENDTIM ) DTPLT(I) = ENDTIM
    IF ( DTPLT(I) .LT. 0.0D0 ) THEN
      WRITE(IUPRT,*) ' ERROR DETECTED.'
      WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
      WRITE(IUPRT,*) ' DTPLT MUST HAVE POSITIVE VALUE.'
      WRITE(IUPRT,*) ' DTPLT(INPUT) IS ',DTPLT(I)
      STOP
    END IF
  ENDDO
  DTPLT(N+1) = ENDTIM
  NAMEOFCARD = 'PLOT'
  GO TO 200
END IF
C
C-----
C EPSILON CARD.
C
IF ( KEYWD .EQ. 'EPSI' ) THEN
  READ(IN,*) EPS,EPSLIM,RXLIM
C
  IF ( EPS .LE. 0.0D0 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' EPS MUST HAVE POSITIVE VALUE.'
    WRITE(IUPRT,*) ' EPS(INPUT) IS ',EPS
    STOP
  END IF
C
  IF ( EPSLIM .LE. 0.0D0 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' EPSLIM MUST HAVE POSITIVE VALUE.'
    WRITE(IUPRT,*) ' EPSLIM(INPUT) IS ',EPSLIM
    STOP
  END IF
C
  IF ( RXLIM .LE. 0.0D0 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' RXLIM MUST HAVE POSITIVE VALUE.'
    WRITE(IUPRT,*) ' RXLIM(INPUT) IS ',RXLIM
    STOP
  END IF
C
  NAMEOFCARD = 'EPSILON'
  GO TO 200

```

```

C      END IF
C-----
C      ITERATION CARD.
C
C      IF ( KEYWD .EQ. 'ITER' ) THEN
C          READ(IN,*) ITMIN,ITMAX
C
C          IF ( ITMIN .LT. 1 ) THEN
C              WRITE(IUPRT,*) ' ERROR DETECTED.'
C              WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C              WRITE(IUPRT,*) ' ITMIN MUST HAVE POSITIVE INTEGER VALUE.'
C              WRITE(IUPRT,*) ' ITMIN(INPUT) IS ',ITMIN
C              STOP
C          END IF
C
C          IF ( ITMAX .LT. 1 ) THEN
C              WRITE(IUPRT,*) ' ERROR DETECTED.'
C              WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C              WRITE(IUPRT,*) ' ITMAX MUST HAVE POSITIVE INTEGER VALUE.'
C              WRITE(IUPRT,*) ' ITMAX(INPUT) IS ',ITMAX
C              STOP
C          END IF
C
C          NAMEOFCARD = 'ITERATION'
C          GO TO 200
C      END IF
C-----
C      TAULIM CARD.
C
C      IF ( KEYWD .EQ. 'TAUL' ) THEN
C          READ(IN,*) DTMIN,DTMAX,TRATE
C
C          IF ( DTMIN .LE. 0.000 ) THEN
C              WRITE(IUPRT,*) ' ERROR DETECTED.'
C              WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C              WRITE(IUPRT,*) ' DTMIN MUST HAVE POSITIVE VALUE.'
C              WRITE(IUPRT,*) ' DTMIN(INPUT) IS ',DTMIN
C              STOP
C          END IF
C
C          IF ( DTMAX .LE. 0.000 ) THEN
C              WRITE(IUPRT,*) ' ERROR DETECTED.'
C              WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C              WRITE(IUPRT,*) ' DTMAX MUST HAVE POSITIVE VALUE.'
C              WRITE(IUPRT,*) ' DTMAX(INPUT) IS ',DTMAX
C              STOP
C          END IF
C
C          IF ( TRATE .LT. 1.000 ) THEN
C              WRITE(IUPRT,*) ' ERROR DETECTED.'
C              WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C              WRITE(IUPRT,*) ' TRATE MUST BE GREATER THAN UNITY.'
C              WRITE(IUPRT,*) ' TRATE(INPUT) IS ',TRATE
C              STOP
C          END IF
C
C          NAMEOFCARD = 'TAULIM'
C          GO TO 200
C      END IF
C-----
C      TAUCONST CARD.
C
C      IF ( KEYWD .EQ. 'TAUC' ) THEN
C          ICRDTC = 1
C          READ(IN,*) NSET
C
C          IF ( NSET .GT. MXTAUC ) THEN
C              WRITE(IUPRT,*) ' TOO LONG TAUCONST TABLE.'
C              WRITE(IUPRT,*) ' LENGTH(INPUT) IS ',NSET
C              WRITE(IUPRT,*) ' LENGTH(LIMIT) IS ',MXTAUC
C              STOP
C          END IF
C
C          DO I = 1 , NSET
C              READ(IN,*) TMCONB(I), TMCONC(I), DTCONS(I)
C
C              IF ( I .EQ. 1 .AND. TMCONB(I) .LT. 0.000 ) THEN
C                  WRITE(IUPRT,*) ' ERROR DETECTED.'
C                  WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C                  WRITE(IUPRT,*) ' TMCONB MUST NOT HAVE NEGATIVE VALUE.'
C                  WRITE(IUPRT,*) ' TMCONB(INPUT) IS ',TMCONB(I)
C                  STOP
C              END IF
C
C              IF ( TMCONC(I) .LT. TMCONB(I) ) THEN
C                  WRITE(IUPRT,*) ' ERROR DETECTED.'

```

```

WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
WRITE(IUPRT,*) ' TMCON.BIGIN LESS THAN TMCON.END.'
WRITE(IUPRT,*) ' TMCONC(INPUT) IS ',TMCONC(I)
STOP
END IF
C
C      IF ( DTCONS(I) .LE. 0.000 ) THEN
C          WRITE(IUPRT,*) ' ERROR DETECTED.'
C          WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
C          WRITE(IUPRT,*) ' DTCONS MUST HAVE POSITIVE VALUE.'
C          WRITE(IUPRT,*) ' DTCONS(INPUT) IS ',DTCONS(I)
C          STOP
C      END IF
C      END DO
C
C      NAMEOFCARD = 'TAUCONST'
C      GO TO 200
C      END IF
C-----
C      LDIST CARD.
C
C      IF ( KEYWD .EQ. 'LDIS' ) THEN
C          IPRDIS = 1
C          NAMEOFCARD = 'LDIST'
C          GO TO 200
C      END IF
C-----
C      TOFSTG CARD.
C
C      IF ( KEYWD .EQ. 'TOFS' ) THEN
C          READ(IN,*) N
C
C          IDEF = 0
C          IUNT = IDCONV(IUPRT,MXUNIT,NUNIT,ID,N,IDEF,KEYWD)
C          IEXTRA = ISEXTR(IUPRT,MXUNIT,NUNIT,
C              *          ID ,IUNT ,KEYWD ,MEXTRA)
C          NSTG = MSTG(IUNT)
C          READ(IN,*) (DATWRK(I),I=1,NSTG)
C
C          DO 350 I = 1 , NSTG
C              TSTG(I,IEXTRA) = DATWRK(I) + TO
C          350 CONTINUE
C
C          NAMEOFCARD = 'TOFSTG'
C          GO TO 200
C      END IF
C-----
C      FILE CARD.
C
C      IF ( KEYWD .EQ. 'FILE' ) THEN
C          READ(IN,*) CTFILE
C          NAMEOFCARD = 'FILE'
C          GO TO 200
C      END IF
C-----
C      END CARD.
C
C      IF ( KEYWD .EQ. 'END' ) GO TO 1000
C-----
C      BLANK CARD.
C
C      IF ( KEYWD .EQ. ' ' ) THEN
C          NAMEOFCARD = 'BLANK'
C          GO TO 200
C      END IF
C-----
C          WRITE(IUPRT,*) ' ERROR DETECTED.'
C          WRITE(IUPRT,*) ' KEY WORD IS NOT AVAILABLE. KEY WORD IS ',KEYWD
C          STOP
C
C
C
C          900 KEYWD = 'END'
C          WRITE(IUPRT,*) ' ERROR DETECTED.'
C          WRITE(IUPRT,*) ' KEY WORD IS MISSING. KEY WORD IS ',KEYWD
C          STOP
C
C
C          1000 CONTINUE
C
C      --- CHANGE UNIT ID'S TO DESTINATION UNIT NUMBERS. ---

```

```

C
  CALL IDCHEK(IUPRT ,
*      MXDEST, MXUNIT, NUNIT ,
*      TYPE , MSTG ,
*      ID , IUDSTA, IUDSTO, MDEST ,
*      MXFEED, NFA , NFO , IFDSTA, IFDSTO)
C
C
C
C
  IF ( INFILE .NE. ' ' ) THEN
C
  CALL READPL(IUPRT , IUPLT , INFILE,
*      CVUNIT,
*      MXCOMP, MXUNIT, MXSTG , MXSTG2, MXBLK ,
*      NUNIT , NBLK ,
*      ID , TYPE , MSTG ,
*      IPOS ,
*      X )
C
  END IF
C
C
C
  RETURN
C
  END

```

```

SUBROUTINE INVGJ( MXN , N , A , IP , EPSX , IER )
C
  IMPLICIT REAL*8 ( A-H , O-Z )
C
C*****
C
  MATRIX INVERSION BY GAUSS-JORDAN METHOD
  A(MXN , MXN ) = INPUT MATRIX AND INVERSE MATRIX IS RETURNED.
  N              = DIMENSION OF MATRIX
  IP             = WORK AREA STORED STATUS OF ROW EXCHANGE.
  IER           = ERROR FLAG
                IER = 0 NORMAL END
                IER = 1 NO INVERSE MATRIX
C*****
C
  DIMENSION IP(MXN) , A(MXN, MXN)
C
  DATA ZERO / 0.0D0 /
C
  IF ( EPSX .LE. ZERO ) THEN
    EPS = 1.0D-50
  ELSE
    EPS = EPSX
  END IF
C
C... TREATMENT FOR THE N=1 CASE.
C
  IF ( N .LE. 0 ) GO TO 9000
C
C
C
  IF ( N .EQ. 1 ) THEN
    NN = 1
    P = A(1,1)
C
    IF ( DABS(A(1,1)) .EQ. ZERO ) GO TO 9000
C
    IER = 0
    A(1,1) = 1.0D0/A(1,1)
    RETURN
C
  ELSE
C
C
C
    MATRIX INVERSION PROCESS
C
    DO 110 NN = 1 , N
      IP(NN) = NN
110    CONTINUE
C
    DO 170 NN = 1 , N
      ADIAG = ZERO
C
      DO 120 I = NN , N
        IF ( ADIAG.GE. DABS(A(1,1)) ) GO TO 120
        ADIAG = DABS(A(1,1))
        IROW = I
120      CONTINUE
C
        IF( ADIAG .LT. EPS ) GO TO 9000
C
        NW = IP(IROW)
        IP(IROW) = IP(NN )
        IP(NN ) = NW
C
        DO 130 J = 1 , N
          W = A(IROW, J)
          A(IROW, J) = A(NN , J)
          A(NN , J) = W
130        CONTINUE
C
          W = A(NN,1)
          DO 140 J = 2 , N
            A(NN, J-1) = A(NN, J)/W
140          CONTINUE
          A(NN,N) = 1.0D0/W
C
          DO 160 I = 1 , N
            IF ( I .EQ. NN ) GO TO 160
            W = A(1,1)
            DO 150 J = 2 , N
              A(1, J-1) = A(1, J) - W*A(NN, J-1)
150            CONTINUE

```

```

      A(I,N) = -W*A(NN,N)
160  CONTINUE
C
C 170  CONTINUE
C
C
C RECOVER THE COLUMN ORDER OF INVERSE MATRIX.
C
      DO 240 NN = 1 , N
C
          DO 210 J = NN , N
              ICOL = J
              IF ( IP(J) .EQ. NN ) GO TO 220
210  CONTINUE
C
          IP(ICOL) = IP(NN)
          DO 230 I = 1 , N
              W = A(I,ICOL)
              A(I,ICOL) = A(I,NN)
              A(I,NN) = W
230  CONTINUE
C
240  CONTINUE
C
      END IF
C
      IER = 0
      RETURN
C
9000 CONTINUE
C
C.... MATRIX IS SINGULAR.
C
      IER = 1
      RETURN
C
      END

```

```

FUNCTION ISEXTR(IUPRT ,MXUNIT,NUNIT ,
*           ID , IUNT ,KEYWD ,MEXTRA)
C
C CONVERT ID NUMBER TO UNIT NO.
C
C CHARACTER*8 KEYWD
C
C
C DIMENSION ID (MXUNIT)
C DIMENSION MEXTRA(MXUNIT)
C
C
C IF ( IUNT .LT. 0 .OR. IUNT .GT. NUNIT ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' IUNT IS ',IUNT
    WRITE(IUPRT,*) ' IUNT VALUE SPECIFIED IS INVALID AT ISEXTR.'
    STOP
END IF
C
C IEXTRA = MEXTRA(IUNT)
C
C
C IF ( IEXTRA .LE. 0 ) THEN
    WRITE(IUPRT,*) ' ERROR DETECTED.'
    WRITE(IUPRT,*) ' KEY WORD IS ',KEYWD
    WRITE(IUPRT,*) ' ID NUMBER IS ',ID(IUNT)
    WRITE(IUPRT,*) ' UNIT IS NOT EXTRACTOR.'
    STOP
END IF
C
ISEXTR = IEXTRA
RETURN
END

```



```

      TF(1, IEXTRA) = XTF
      END IF
C
      FMA = - ( FA + FRA )
      FMR =      FRA
      FMO = -  F0
C
      --- MIXER MIXTURE ---
C
      DO 200 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. 10TCU6
      *      .AND. JJ .NE. 10TCP4
      *      .AND. JJ .NE. 10TCZR ) THEN
C
          FFFA = DTINV*( XN(JJ, 1, IND)*VMA - X(JJ, 1, IND)*VMAOLD )
          *      - FMA*XN(JJ, 1, IND)
          *      - FMR*XN(JJ, 1, INDSA)
          *      - REAC ( J , 1)*VMA
          FFF0 = DTINV*( XN(JJ, 2, IND)*VMO - X(JJ, 2, IND)*VMOOLD )
          *      - FMO*XN(JJ, 2, IND)
          *      - REAC ( J , 2)*VMO
          FX(J, IND) = FFFA + FFF0
          DO 110 K = 1 , NCOMP
          DFDXD(J, K , IND) = DRDX(J, K, 1)*VMA
          DFDXD(J, K+NCOMP, IND) = DRDX(J, K, 2)*VMO
110      CONTINUE
          DFDXD(J, J , IND) = DFDXD(J, J , IND)
          *      - DTINV*VMA + FMA
          DFDXD(J, J+NCOMP, IND) = DFDXD(J, J+NCOMP, IND)
          *      - DTINV*VMO + FMO
C
          ELSE
C
              FFFA = 0.000
              FFF0 = DTINV*( XN(JJ, 2, IND)*VMO - X(JJ, 2, IND)*VMOOLD )
              *      - FMO*XN(JJ, 2, IND)
              *      - REAC ( J , 2)*VMO
              FX(J, IND) = FFFA + FFF0
              DO 120 K = 1 , NCOMP
              DFDXD(J, K+NCOMP, IND) = DRDX(J, K, 2)*VMO
120      CONTINUE
              DFDXD(J, J+NCOMP, IND) = DFDXD(J, J+NCOMP, IND)
              *      - DTINV*VMO + FMO
C
          END IF
200      CONTINUE
C
      IF ( JMIC .LE. 0 ) THEN
C
          DISINTEGRATION OF HNO2 BY HYDRAZINE
          FROM EXTRA, M-CODE EQ-(2, 5-43)
          ( INTERFACIAL REACTION )
C
          FX(10HNO3, IND) = FX(10HNO3, IND) - RXHYD *VMO
          DFDXD(10HNO3, 10HNO3+NCOMP, IND)
          *      = DFDXD(10HNO3, 10HNO3+NCOMP, IND) + DRYH *VMO
          DFDXD(10HNO3, 10HYD , IND)
          *      = DFDXD(10HNO3, 10HYD , IND) + DRXHYD*VMO
          DFDXD(10HNO3, 10HNO2+NCOMP, IND)
          *      = DFDXD(10HNO3, 10HNO2+NCOMP, IND) + DRYNO2*VMO
C
          FX(10HYD , IND) = FX(10HYD , IND) + RXHYD *VMO
          DFDXD(10HYD , 10HNO3+NCOMP, IND)
          *      = DFDXD(10HYD , 10HNO3+NCOMP, IND) - DRYH *VMO
          DFDXD(10HYD , 10HYD , IND)
          *      = DFDXD(10HYD , 10HYD , IND) - DRXHYD*VMO
          DFDXD(10HYD , 10HNO2+NCOMP, IND)
          *      = DFDXD(10HYD , 10HNO2+NCOMP, IND) - DRYNO2*VMO
C
          FX(10HNO2, IND) = FX(10HNO2, IND) + 2.000*RXHYD *VMO
          DFDXD(10HNO2, 10HNO3+NCOMP, IND)
          *      = DFDXD(10HNO2, 10HNO3+NCOMP, IND) - 2.000*DRYH *VMO
          DFDXD(10HNO2, 10HYD , IND)
          *      = DFDXD(10HNO2, 10HYD , IND) - 2.000*DRXHYD*VMO
          DFDXD(10HNO2, 10HNO2+NCOMP, IND)
          *      = DFDXD(10HNO2, 10HNO2+NCOMP, IND) - 2.000*DRYNO2*VMO
C
          .. TECHNETIUM-U(VI) CO-EXTRACTION...
C
          FX(10TC7 , IND)
          *      = FX(10TC7 , IND) + XN(10TCU6, 1, IND)*VMA
          DFDXD(10TC7 , 10TCU6, IND)
          *      = DFDXD(10TC7 , 10TCU6, IND) - VMA
          IF ( !D1ST(10TCU6, IEXTRA) .NE. !DAMI X ) THEN
          *      FX(10U6 , IND)
          *      = FX(10U6 , IND) + XN(10TCU6, 1, IND)*VMA
      
```

```

      *      DFDXD(10U6 , 10TCU6, IND)
      *      = DFDXD(10U6 , 10TCU6, IND) - VMA
      END IF
      *      FX(10TCU6, IND)
      *      = FX(10TCU6, IND) - XN(10TCU6, 1, IND)*VMA
      DFDXD(10TCU6, 10TCU6, IND)
      *      = DFDXD(10TCU6, 10TCU6, IND) + VMA
C
      .. TECHNETIUM-PU(IV) CO-EXTRACTION...
C
      FX(10TC7 , IND)
      *      = FX(10TC7 , IND) + XN(10TCP4, 1, IND)*VMA
      DFDXD(10TC7 , 10TCP4, IND)
      *      = DFDXD(10TC7 , 10TCP4, IND) - VMA
      IF ( !D1ST(10TCP4, IEXTRA) .NE. !DAMI X ) THEN
      *      FX(10PU4 , IND)
      *      = FX(10PU4 , IND) + XN(10TCP4, 1, IND)*VMA
      DFDXD(10PU4 , 10TCP4, IND)
      *      = DFDXD(10PU4 , 10TCP4, IND) - VMA
      END IF
      *      FX(10TCP4, IND)
      *      = FX(10TCP4, IND) - XN(10TCP4, 1, IND)*VMA
      DFDXD(10TCP4, 10TCP4, IND)
      *      = DFDXD(10TCP4, 10TCP4, IND) + VMA
C
      .. TECHNETIUM-ZR(IV) CO-EXTRACTION...
C
      FX(10TC7 , IND)
      *      = FX(10TC7 , IND) + XN(10TCZR, 1, IND)*VMA
      DFDXD(10TC7 , 10TCZR, IND)
      *      = DFDXD(10TC7 , 10TCZR, IND) - VMA
      IF ( !D1ST(10TCZR, IEXTRA) .NE. !DAMI X ) THEN
      *      FX(10ZR , IND)
      *      = FX(10ZR , IND) + XN(10TCZR, 1, IND)*VMA
      DFDXD(10ZR , 10TCZR, IND)
      *      = DFDXD(10ZR , 10TCZR, IND) - VMA
      END IF
      *      FX(10TCZR, IND)
      *      = FX(10TCZR, IND) - XN(10TCZR, 1, IND)*VMA
      DFDXD(10TCZR, 10TCZR, IND)
      *      = DFDXD(10TCZR, 10TCZR, IND) + VMA
C
      END IF
C
      DO 300 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. 10TCU6
      *      .AND. JJ .NE. 10TCP4
      *      .AND. JJ .NE. 10TCZR ) THEN
          DFDXL(J, J, IND) = FMR
      END IF
300      CONTINUE
C
      --- MIXER ORGANIC ---
C
      DO 500 J = 1 , NCOMP
      JJ = J + JBASE
C
      EF = EFF(JJ, 1, IEXTRA)
      DISCL = DIS(JJ, 1, IEXTRA)
C
      IF ( !EFF(JJ, IEXTRA) .EQ. 1 ) THEN
          DISEF = ( ( FA + FRA)*(ONE - EF) + F0*DISCL )/(F0*EF)
          DDISEF = ONE/EF
      ELSE
          DENOM = ( FA + FRA ) + F0*DISCL*( ONE - EF )
          DISEF = ( FA + FRA ) *DISCL*EF/DENOM
          DDISEF = ( FA + FRA ) *EF/DENOM
          *      - ( FA + FRA ) *DISCL*EF*F0*( ONE - EF )
          *      / (DENOM*DENOM)
      END IF
C
      IF (      JJ .NE. 10TCU6
      *      .AND. JJ .NE. 10TCP4
      *      .AND. JJ .NE. 10TCZR ) THEN
          FX(J+NCOMP, IND) = DISEF*XN(JJ, 1, IND) - XN(JJ, 2, IND)
          DO 410 K = 1 , NCOMP
      
```

```

      DFDXD(J+NCOMP,K,IND) = - DDISEF*DDDX(J,K)*XN(JJ,1,IND)
410  CONTINUE
      DFDXD(J+NCOMP,J,IND) = DFDXD(J+NCOMP,J,IND) - DISEF
      DFDXD(J+NCOMP,J+NCOMP,IND) = ONE
C
      ELSE
C
      FX(J+NCOMP,IND) = DISEF*XN(10TC7,1,IND) - XN(JJ,2,IND)
      DO 420 K = 1, NCOMP
      DFDXD(J+NCOMP,K,IND) = - DDISEF*DDDX(J,K)
      *                               *XN(10TC7,1,IND)
420  CONTINUE
      DFDXD(J+NCOMP,10TC7,IND)
      * = DFDXD(J+NCOMP,10TC7,IND) - DISEF
      DFDXD(J+NCOMP,J+NCOMP,IND) = ONE
C
      END IF
C
500  CONTINUE
C
600  CONTINUE
C
      ---- MIXER AQUEOUS ----
C
      DO 1200 I = 1, NSTG - 1
C
      IM = 2*I - 1
      IS = 2*I
      IND = IPOS(1,IM,IUNT)
      INDSAP = IPOS(1,IS+2,IUNT)
C
      FMAP = FL(1,I+1,IEXTRA)
C
      DO 1100 J = 1, NCOMP
      JJ = J + JBASE
      IF ( JJ.NE.10TCU6
      * .AND. JJ.NE.10TCP4
      * .AND. JJ.NE.10TCZR ) THEN
C
      FX(J,IND) = FX(J,IND) - FMAP*XN(JJ,1,INDSAP)
      DFDXU(J,J,IND) = FMAP
C
      END IF
1100  CONTINUE
C
1200  CONTINUE
C
      ---- MIXER ORGANIC ----
C
      DO 1500 I = 2, NSTG
C
      IM = 2*I - 1
      IS = 2*I
      IND = IPOS(2,IM,IUNT)
      INDSOM = IPOS(2,IS-2,IUNT)
C
      FMOM = FL(2,I-1,IEXTRA)
      DO 1400 J = 1, NCOMP
      JJ = J + JBASE
      FX(J,IND) = FX(J,IND) - FMOM*XN(JJ,2,INDSOM)
      DFDXL(J,J+NCOMP,IND) = FMOM
1400  CONTINUE
C
1500  CONTINUE
C
      DO 2600 I = 1, NSTG
C
      VSA = PHIS(1,IEXTRA) * VS(1,IEXTRA)
      VSO = (1.000 - PHIS(1,IEXTRA)) * VS(1,IEXTRA)
C
      FRA = FR(1,IEXTRA)
      FA = FL(1,1,IEXTRA)
      FO = FL(2,1,IEXTRA)
C
      IM = 2*I - 1
      IS = 2*I
      IND = IPOS(1,IM,IUNT)
      INDSA = IPOS(1,IS,IUNT)
      INDSO = IPOS(2,IS,IUNT)
C
      FSA = (FA + FRA)
      FSO = FO

```

```

C
      ---- SETTLER AQUEOUS ----
C
      T = TSTG(1,IEXTRA)
C
      IREACT = 1
C
      CALL REACT(IREACT,MXCOMP,JM1C,NCOMP,NCHEM,MXREAC,
      * T,TO,RCON,
      * X(1,1,INDSA),XN(1,1,INDSA),
      * FCHEM(1,1,IUNT),
      * ECHEM(1,1,IUNT),
      * REAC,DFDX,
      * RXHYD,DRYH,DRXHYD,DRYN02)
C
      DO 2200 J = 1, NCOMP
C
      JJ = J + JBASE
      IF ( JJ.NE.10TCU6
      * .AND. JJ.NE.10TCP4
      * .AND. JJ.NE.10TCZR ) THEN
C
      FX(J,INDSA) = DTINV*(XN(JJ,1,INDSA) - X(JJ,1,INDSA))*VSA
      * - FSA*XN(JJ,1,IND)
      * + FSA*XN(JJ,1,INDSA)
      * - REAC(J,1)*VSA
      DO 2110 K = 1, NCOMP
      DFDXD(J,K,INDSA) = DFDX(J,K,1)*VSA
2110  CONTINUE
C
      DFDXD(J,J,INDSA) = DFDXD(J,J,INDSA) - DTINV*VSA - FSA
      DFDXU(J,J,INDSA) = FSA
C
      ELSE
C
      FX(J,INDSA) = ZERO - XN(JJ,1,INDSA)
      DO 2120 K = 1, NCOMP
      DFDXD(J,K,INDSA) = 0.000
2120  CONTINUE
C
      DFDXD(J,J,INDSA) = ONE
C
      END IF
C
2200  CONTINUE
C
      ---- SETTLER ORGANIC ----
C
      IREACT = 2
C
      CALL REACT(IREACT,MXCOMP,JM1C,NCOMP,NCHEM,MXREAC,
      * T,TO,RCON,
      * X(1,1,INDSO),XN(1,1,INDSO),
      * FCHEM(1,1,IUNT),
      * ECHEM(1,1,IUNT),
      * REAC,DFDX,
      * RXHYD,DRYH,DRXHYD,DRYN02)
C
      DO 2300 J = 1, NCOMP
      DFDXL(J+NCOMP,J+NCOMP,INDSO) = FSO
2300  CONTINUE
C
      DO 2500 J = 1, NCOMP
      JJ = J + JBASE
      FX(J+NCOMP,INDSO) = DTINV*(XN(JJ,2,INDSO) - X(JJ,2,INDSO))*VSO
      * - FSO*XN(JJ,2,IND)
      * + FSO*XN(JJ,2,INDSO)
      * - REAC(J,2)*VSO
      DO 2400 K = 1, NCOMP
      DFDXD(J+NCOMP,K+NCOMP,INDSO) = DRDX(J,K,2)*VSO
2400  CONTINUE
      DFDXD(J+NCOMP,J+NCOMP,INDSO) = DFDXD(J+NCOMP,J+NCOMP,INDSO)
      * - DTINV*VSO - FSO
C
2500  CONTINUE
C
2600  CONTINUE
C
      ... MODIFY LEFT MOST STAGE
C
      I = 1
      IS = 2*I
      INDSA = IPOS(1,IS,IUNT)
      DO 3100 J = 1, NCOMP

```



```

        FX (J+NCOMP , INDSA) = 0.000
        DFDXD (J+NCOMP, J+NCOMP, INDSA) = 1.000
3100 CONTINUE
C
C
C... MODIFY RIGHT MOST STAGE
C
        I = NSTG
        IS = 2*I
        INDSO = IPOS(2, IS, IUNT)
        DO 3200 J = 1 , NCOMP
            FX (J , INDSO) = 0.000
            DFDXD (J, J, INDSO) = 1.000
3200 CONTINUE
C
C
C
        RETURN
        END
    
```

```

SUBROUTINE LUDCOM( A, N, MXN, EPSX, WK, IP, IER )
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      SOLVES SIMULTANEOUS LINEAR EQUATIONS
C      BY GAUSSIAN ELIMINATION METHOD.
C
C      A      2-DIM. ARRAY CONTAINING THE COEFFICIENTS
C             OR RESULT OF GAUSSIAN ELIMINATION.
C      N      ORDER OF MATRIX.
C      MXN    SIZE OF ARRAY A.
C      EPSX   PARAMETER TO CHECK SINGULARITY OF THE MATRIX.
C             ( STANDARD VALUE 1.0D-20 )
C      WK     1-DIM. ARRAY FOR WORKING.
C      IP     PIVOT NUMBERS.
C      IER    = 0, FOR NORMAL EXECUTION.
C             = 1, FOR SINGULAR MATRIX.
C
C      DIMENSION A(MXN,N), IP(N), WK(N)
C
C
C      LEFT HAND SIDE
C
C      IER = 0
C      EPS = EPSX
C      IF ( EPS .LE. 0.000 ) EPS = 1.0D-20
C
C
C      DO 500 K = 1 , N
C
C          FIND MAXIMUM ELEMENT IN THE K-TH COLUMN.
C
C          IROW = K
C          OMAX = 0.000
C
C          DO 100 I = K , N
C              O = DABS(A(I,K))
C              IF ( OMAX .LT. O ) THEN
C                  IROW = I
C                  OMAX = O
C              END IF
100          CONTINUE
C          IP(K) = IROW
C
C          IF ( OMAX .LE. EPS ) THEN
C              IER = 1
C              IP(K) = K
C              WRITE(6,*)
C              *      (SUBR. LUDCOM) MATRIX IS SINGULAR AT K =', K
C              RETURN
C          END IF
C
C          IF ( IROW .NE. K ) THEN
C              W = A(IROW,K)
C              A(IROW,K) = A(K ,K)
C              A(K ,K) = W
C          END IF
C
C          DO 200 I = K+1 , N
C              A (I,K) = - A(I,K)/A(K,K)
C              WK(I) = A(I,K)
200          CONTINUE
C
C          DO 400 J = K+1 , N
C              IF ( IROW .NE. K ) THEN
C                  W = A(IROW,J)
C                  A(IROW,J) = A(K ,J)
C                  A(K ,J) = W
C              END IF
C
C              GAUSSIAN ELIMINATION
C
C              T = A(K,J)
C              DO 300 I = K+1 , N
C                  A(I,J) = A(I,J) + WK(I)*T
300              CONTINUE
400              CONTINUE
C
C          500 CONTINUE
C              RETURN
C              END
C
    
```

```

SUBROUTINE LUSOLV( A, N, MXN, B, IP )
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      SOLVES SIMULTANEOUS LINEAR EQUATIONS
C      BY GAUSSIAN ELIMINATION METHOD.
C
C      A      RESULT OF GAUSSIAN ELIMINATION.
C      N      ORDER OF MATRIX.
C      MXN    SIZE OF ARRAY A.
C      B      1-DIM. ARRAY CONTAINING THE RIGHT HAND SIDE VECTOR
C             OR SOLUTION VECTOR.
C      IP     PIVOT NUMBERS.
C
C      A AND IP MUST BE COMPUTED BY LUDCOM.
C
C      DIMENSION A(MXN,N), B(N), IP(N)
C
C      FORWARD ELIMINATION PROCESS
C
C      DO 200 K = 1, N
C
C         IF ( IP(K) .NE. K ) THEN
C            W = B(IP(K))
C            B(IP(K)) = B(K)
C            B(K) = W
C         END IF
C
C      GAUSSIAN ELIMINATION
C
C         T = B(K)
C         DO 100 I = K+1, N
C            B(I) = B(I) + A(I,K)*T
C 100    CONTINUE
C 200    CONTINUE
C
C      BACKWARD SUBSTITUTION PROCESS
C
C      B(N) = B(N)/A(N,N)
C      DO 1200 K = N-1, 1, -1
C         T = B(K+1)
C         DO 1100 I = 1, K
C            B(I) = B(I) - A(I,K+1)*T
C 1100    CONTINUE
C         B(K) = B(K)/A(K,K)
C 1200    CONTINUE
C
C      RETURN
C      END
    
```

```

SUBROUTINE MAKEJB(MXDEST,MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,
*
*      MXBLK, JMTC, JBASE, NCOMP, MXREAC, MEXEXTR,
*
*      NUNIT, NBLK, ISTDY, DT,
*
*      FRC, CO, TO, FCON, CHRG, CNTBP,
*
*      MXFEED, NFA, NFO,
*
*      FF, XF, IFDSTA, IFDSTO,
*
*      TYPE, MSTG, IUDSTA, IUDSTO, MDEST, MEXTRA,
*
*      TAV, VTOT, ACID, OXID, FOUT, RFOUT,
*
*      IDIST, FCHEM, ECHEM, CDIS, CKEOU,
*
*      IEFF, EFF,
*
*      DIS, TSTG, V, VS, FL, FR,
*
*      PHIOLD, PHI, PHIS, TF, CDBP,
*
*      IPOS, XN, X,
*
*      REAC, DFDX, DDDX,
*
*      FKEQ1, FKEQ2, FKEQ3, FKEQC,
*
*      DK1, DK2, DK3, DKC,
*
*      DK1DX, DK2DX, DK3DX, DKCDX,
*
*      DADX, DEDX, DTFDX, DISTC, DDCDX, DDCDTF,
*
*      MXN, MXRM, NXPM,
*
*      FX, DFDXD, DFDXL, DFDXU, RM )
C
C      MAKE JACOBIAN
C
C      IMPLICIT REAL*8 ( A-H, O-Z )
C
C      COMMON /IXCOMP/NCHEM0,
*
*      IOHNO3, IOU6, IOPU4, IOPU3, IOU4,
*
*      IOHNO2, IOHYD, IOHAN, IOPU5, IOPU6,
*
*      IONP4, IONP5, IONP6, IOZR, IOTC4,
*
*      IOTC5, IOTC6, IOTC7, IOTCJ6, IOTCP4,
*
*      IOTCZR, IOSR, IOFUD1, IOFUTR, IOFURN1,
*
*      IOFUMO, IOCS, IOCE, IOGD, IOAM,
*
*      IOCM
C
C      DIMENSION CHRG (MXCOMP),
*
*      CNTBP (MXCOMP)
C
C      DIMENSION FF ( MXFEED, 2),
*
*      XF (MXCOMP, MXFEED, 2),
*
*      IFDSTA (2, MXFEED),
*
*      IFDSTO (2, MXFEED)
C
C      DIMENSION FCHEM (MXREAC, 2, MXUNIT),
*
*      ECHEM (MXREAC, 2, MXUNIT)
C
C      CHARACTER*2 TYPE ( MXUNIT)
C
C      DIMENSION MSTG ( MXUNIT),
*
*      IUDSTA (3, MXDEST, MXUNIT),
*
*      IUDSTO (3, MXDEST, MXUNIT)
C
C      DIMENSION MDEST ( 2, MXUNIT),
*
*      MEXTRA ( MXUNIT)
C
C      DIMENSION TAV ( MXUNIT),
*
*      VTOT ( MXUNIT),
*
*      ACID ( MXUNIT),
*
*      OXID ( MXUNIT),
*
*      FOUT ( 2, MXUNIT),
*
*      RFOUT (MXDEST, 2, MXUNIT)
C
C      DIMENSION IDIST (MXCOMP, MXEXTR)
C
C      DIMENSION IEFF (MXCOMP, MXEXTR)
C
C      DIMENSION CDIS (MXCOMP, MXSTG, MXEXTR),
*
*      CKEOU (MXCOMP, MXSTG, MXEXTR),
*
*      EFF (MXCOMP, MXSTG, MXEXTR),
*
*      DIS (MXCOMP, MXSTG, MXEXTR)
C
C      DIMENSION TSTG ( MXSTG, MXEXTR),
*
*      V ( MXSTG, MXEXTR),
*
*      VS ( MXSTG, MXEXTR),
*
*      FL (2, MXSTG, MXEXTR),
*
*      FR ( MXSTG, MXEXTR),
*
*      PHIOLD ( MXSTG, MXEXTR),
*
*      PHI ( MXSTG, MXEXTR),
*
*      PHIS ( MXSTG, MXEXTR),
*
*      TF ( MXSTG, MXEXTR)
C
C      DIMENSION IPOS (2, MXSTG2, MXUNIT)
C
C      DIMENSION XN (MXCOMP, 2, MXBLK ),
*
*      X (MXCOMP, 2, MXBLK )
C
    
```

```

C
C
  DIMENSION REAC(NCHEM, 2) ,
*           DRDX(NCHEM, NCHEM, 2)
C
  DIMENSION FKE01 (NCHEM),
*           FKE02 (NCHEM),
*           FKE03 (NCHEM),
*           FKE0C (NCHEM)
  DIMENSION DK1 (NCHEM),
*           DK2 (NCHEM),
*           DK3 (NCHEM),
*           DKC (NCHEM)
  DIMENSION DK1DX (NCHEM, NCHEM),
*           DK2DX (NCHEM, NCHEM),
*           DK3DX (NCHEM, NCHEM),
*           DKCDX (NCHEM, NCHEM)
  DIMENSION DADX (NCHEM),
*           DBDX (NCHEM),
*           DTFDX (NCHEM)
  DIMENSION DISTC (NCHEM),
*           DDCDX (NCHEM, NCHEM),
*           DDCDTF(NCHEM)
C
  DIMENSION FX (MXN ,MXBLK),
*           DFDXD (MXN ,MXN ,MXBLK),
*           DFDXL (MXN ,MXN ,MXBLK),
*           DFDXU (MXN ,MXN ,MXBLK)
  DIMENSION FM (MXN ,MXN ,MXRM )
C
C
C
  DATA ZERO/0.000/
  DATA ONE /1.000/
C
C
  NCOMP2 = 2*NCOMP
C
C
  --- SET ZERO TO JACOBIAN ---
C
  DO 130 I = 1 , NBLK
C
  DO 120 JJ = 1 , NCOMP2
    DO 110 J = 1 , NCOMP2
      DFDXL(J, JJ, I) = ZERO
      DFDXD(J, JJ, I) = ZERO
      DFDXU(J, JJ, I) = ZERO
    110 CONTINUE
    120 CONTINUE
  130 CONTINUE
C
C
  DO 230 IXRM = 1 , NXRM
C
  DO 220 JJ = 1 , NCOMP2
    DO 210 J = 1 , NCOMP2
      FM(J, JJ, IXRM) = ZERO
    210 CONTINUE
    220 CONTINUE
  230 CONTINUE
C
C
  IF ( ISTDY .GE. 1 ) THEN
    DTINV = ZERO
  ELSE
    DTINV = ONE / DT
  END IF
C
C
  --- MAKE JACOBIAN ---
C
  DO 350 IUNT = 1 , NUNIT
C
C
  NSTG = MSTG (IUNT)
  IEXTRA = MEXTRA(IUNT)
C
  T = TAV ( IUNT)
  VTA = VTOT( IUNT)
  VTO = VTOT( IUNT)
  FTA = FOUT(1, IUNT)
  FTO = FOUT(2, IUNT)
C
C

```

```

C
IF( TYPE(IUNT) .EQ. 'MS' ) THEN
C
  CALL JACOBI(MXCOMP, MXUNIT, MEXTR, MXSTG , MXSTG2,
*           JM1C , JBASE , NCOMP , NCHEM , MXREAC, MXBLK ,
*           ISTDY , DT ,
*           FRC , CO , TO , RCON , CHRGR , CNTBP ,
*           NSTG , IUNT , IEXTRA,
*           IDIST , FCHEM , ECHEM , CDIS , CKEOU ,
*           IEFF , EFF ,
*           DIS , TSTG , V , VS , FL , FR ,
*           PHIOLD, PHI , PHIS , TF , CDBP ,
*           IPOS , XN , X ,
*           REAC , DRDX ,
*           DDDX ,
*           FKE01 , FKE02 , FKE03 , FKE0C ,
*           DK1 , DK2 , DK3 , DKC ,
*           DK1DX , DK2DX , DK3DX , DKCDX ,
*           DADX , DBDX , DTFDX ,
*           DISTC , DDCDX , DDCDTF,
*           MXN ,
*           FX , DFDXD , DFDXL , DFDXU )
C
C
ELSE IF( TYPE(IUNT) .EQ. 'TA' ) THEN
C
C
  --- TANK AQUEOUS ---
C
  I = 1
  IND = IPOS(1, I, IUNT)
C
  DO 310 J = 1 , NCOMP
C
  JJ = J + JBASE
  IF ( JJ .NE. IOTCU6
*     .AND. JJ .NE. IOTCP4
*     .AND. JJ .NE. IOTCZR ) THEN
C
  FX(J, IND) = DTINV*( XN(JJ, 1, IND)-X(JJ, 1, IND) ) *VTA
*           + FTA*XN(JJ, 1, IND)
C
  DFDXD(J, J, IND) = - DTINV*VTA - FTA
C
  ELSE
C
  FX (J , IND) = ZERO - XN(JJ, 1, IND)
  DFDXD(J, J, IND) = ONE
C
  END IF
C
  310 CONTINUE
C
  DO 320 J = 1 , NCOMP
  FX (J+NCOMP , IND) = ZERO - XN(JJ, 2, IND)
  DFDXD(J+NCOMP, J+NCOMP, IND) = ONE
  320 CONTINUE
C
C
ELSE IF( TYPE(IUNT) .EQ. 'TO' ) THEN
C
C
  --- TANK ORGANIC ---
C
  I = 1
  IND = IPOS(2, I, IUNT)
C
  DO 330 J = 1 , NCOMP
  JJ = J + JBASE
  FX (J , IND) = ZERO - XN(JJ, 1, IND)
  DFDXD(J, J, IND) = ONE
  330 CONTINUE
C
  DO 340 J = 1 , NCOMP
  JJ = J + JBASE
  FX(J+NCOMP, IND) = DTINV*(XN(JJ, 2, IND)-X(JJ, 2, IND) ) *VTO
*           + FTO*XN(JJ, 2, IND)
  DFDXD(J+NCOMP, J+NCOMP, IND) = - DTINV*VTO - FTO
  340 CONTINUE
C
  END IF
C

```

```

C
350 CONTINUE
C
C
DO 1500 IUNTS = 1 , NUNIT
C
NDEST = MDEST(1, IUNTS)
C
IF ( NDEST .GT. 0 ) THEN
C
DO 1400 KDEST = 1 , NDEST
C
NSTGS = MSTG ( IUNTS)
IUNT = IUDSTA(1, KDEST, IUNTS)
I = IUDSTA(2, KDEST, IUNTS)
IS = 1
C
IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
IMS = 2*IS
ELSE
IMS = IS
END IF
C
INDAS = IPOS(1, IMS, IUNTS)
INDOS = IPOS(2, IMS, IUNTS)
C
IF ( IUNT .NE. 0 ) THEN
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
IM = 2*I - 1
ELSE
IM = I
END IF
C
INDA = IPOS(1, IM, IUNT)
INDO = IPOS(2, IM, IUNT)
C
FAS = FOUT(1, IUNTS)*RFOUT(KDEST, 1, IUNTS)
C
IF ( TYPE(IUNT) .EQ. 'TA'
.AND. OXID(IUNT) .GT. ZERO ) THEN
C
OXIDX = OXID(IUNT)
OXID1 = ONE - OXIDX
C
DO 1110 J = 1 , NCOMP
JJ = J + JBASE
IF ( JJ .NE. 10TCU6
.AND. JJ .NE. 10TCP4
.AND. JJ .NE. 10TCZR ) THEN
IF ( JJ .EQ. 10U6 ) THEN
FX(J, INDA)
= FX(J, INDA)
- FAS*XN(10U6 ,1, INDAS)
- FAS*XN(10U4 ,1, INDAS)
ELSE IF ( JJ .EQ. 10PU4 ) THEN
FX(J, INDA)
= FX(J, INDA)
- OXIDX*FAS*XN(10PU4 ,1, INDAS)
- OXIDX*FAS*XN(10PU3 ,1, INDAS)
ELSE IF ( JJ .EQ. 10PU3 ) THEN
FX(J, INDA)
= FX(J, INDA)
- OXID1*FAS*XN(10PU4 ,1, INDAS)
- OXID1*FAS*XN(10PU3 ,1, INDAS)
ELSE IF ( JJ .EQ. 10U4 ) THEN
ELSE IF ( JJ .EQ. 10HYD ) THEN
ELSE IF ( JJ .EQ. 10HAN ) THEN
ELSE
FX(J, INDA) =FX(J, INDA)-FAS*XN(JJ, 1, INDAS)
END IF
END IF
CONTINUE
1110
C
IF ( INDAS .EQ. (INDA-1) ) THEN
C
DO 1120 J = 1 , NCOMP
JJ = J + JBASE
IF ( JJ .NE. 10TCU6
.AND. JJ .NE. 10TCP4
.AND. JJ .NE. 10TCZR ) THEN
IF ( JJ .EQ. 10U6 ) THEN
JJJ = 10U4
DFDXL(J, J , INDA) = DFDXL(J, J , INDA)
+ FAS
DFDXL(J, JJJ, INDA) = DFDXL(J, JJJ, INDA)
+ FAS
ELSE IF ( JJ .EQ. 10PU4 ) THEN
JJJ = 10PU3
DFDXL(J, J , INDA) = DFDXL(J, J , INDA)
+ OXIDX*FAS
DFDXL(J, JJJ, INDA) = DFDXL(J, JJJ, INDA)
+ OXIDX*FAS
ELSE IF ( JJ .EQ. 10PU3 ) THEN
JJJ = 10PU4
DFDXL(J, J , INDA) = DFDXL(J, J , INDA)
+ OXID1*FAS
DFDXL(J, JJJ, INDA) = DFDXL(J, JJJ, INDA)
+ OXID1*FAS
ELSE IF ( JJ .EQ. 10U4 ) THEN
ELSE IF ( JJ .EQ. 10HYD ) THEN
ELSE IF ( JJ .EQ. 10HAN ) THEN
ELSE
DFDXL(J, J, INDA) = DFDXL(J, J, INDA) + FAS
END IF
END IF
CONTINUE
ELSE IF( INDAS .EQ. (INDA+1) ) THEN
DO 1130 J = 1 , NCOMP
JJ = J + JBASE
IF ( JJ .NE. 10TCU6
.AND. JJ .NE. 10TCP4
.AND. JJ .NE. 10TCZR ) THEN
IF ( JJ .EQ. 10U6 ) THEN
JJJ = 10U4
DFDXU(J, J , INDA) = DFDXU(J, J , INDA)
+ FAS
DFDXU(J, JJJ, INDA) = DFDXU(J, JJJ, INDA)
+ FAS
ELSE IF ( JJ .EQ. 10PU4 ) THEN
JJJ = 10PU3
DFDXU(J, J , INDA) = DFDXU(J, J , INDA)
+ OXIDX*FAS
DFDXU(J, JJJ, INDA) = DFDXU(J, JJJ, INDA)
+ OXIDX*FAS
ELSE IF ( JJ .EQ. 10PU3 ) THEN
JJJ = 10PU4
DFDXU(J, JJJ, INDA) = DFDXU(J, JJJ, INDA)
+ OXID1*FAS
DFDXU(J, J , INDA) = DFDXU(J, J , INDA)
+ OXID1*FAS
ELSE IF ( JJ .EQ. 10U4 ) THEN
ELSE IF ( JJ .EQ. 10HYD ) THEN
ELSE IF ( JJ .EQ. 10HAN ) THEN
ELSE
DFDXU(J, J, INDA) = DFDXU(J, J, INDA) + FAS
END IF
END IF
CONTINUE
ELSE
IXPM = IUDSTA(3, KDEST, IUNTS)
DO 1140 J = 1 , NCOMP
JJ = J + JBASE
IF ( JJ .NE. 10TCU6
.AND. JJ .NE. 10TCP4
.AND. JJ .NE. 10TCZR ) THEN
IF ( JJ .EQ. 10U6 ) THEN
JJJ = 10U4
FM(J, J , IXPM) = FM(J, J , IXPM)
+ FAS
FM(J, JJJ, IXPM) = FM(J, JJJ, IXPM)
+ FAS
ELSE IF ( JJ .EQ. 10PU4 ) THEN
JJJ = 10PU3
FM(J, J , IXPM) = FM(J, J , IXPM)
+ OXIDX*FAS
FM(J, JJJ, IXPM) = FM(J, JJJ, IXPM)
+ OXIDX*FAS
ELSE IF ( JJ .EQ. 10PU3 ) THEN
JJJ = 10PU4
FM(J, JJJ, IXPM) = FM(J, JJJ, IXPM)
+ OXID1*FAS
FM(J, J , IXPM) = FM(J, J , IXPM)
+ OXID1*FAS
ELSE IF ( JJ .EQ. 10U4 ) THEN
ELSE IF ( JJ .EQ. 10HYD ) THEN
ELSE IF ( JJ .EQ. 10HAN ) THEN
ELSE
FM(J, J, IXPM) = FM(J, J, IXPM) + FAS
END IF
END IF
CONTINUE
1140

```

```

C
      END IF
C
C
      ELSE
C
      DO 1210 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. IOTCU6
      *      .AND. JJ .NE. IOTCP4
      *      .AND. JJ .NE. IOTCZR ) THEN
      FX(J, INDA) = FX(J, INDA) - FAS*XN(JJ, 1, INDAS)
      END IF
1210      CONTINUE
C
      IF ( INDAS .EQ. (INDA-1) ) THEN
C
      DO 1220 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. IOTCU6
      *      .AND. JJ .NE. IOTCP4
      *      .AND. JJ .NE. IOTCZR ) THEN
      DFDXL(J, J, INDA) = DFDXL(J, J, INDA) + FAS
      END IF
1220      CONTINUE
C
      ELSE IF( INDAS .EQ. (INDA+1) ) THEN
C
      DO 1230 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. IOTCU6
      *      .AND. JJ .NE. IOTCP4
      *      .AND. JJ .NE. IOTCZR ) THEN
      DFDXU(J, J, INDA) = DFDXU(J, J, INDA) + FAS
      END IF
1230      CONTINUE
C
      ELSE
C
      IXFM = IUDSTA(3, KDEST, IUNTS)
      DO 1240 J = 1 , NCOMP
      JJ = J + JBASE
      IF (      JJ .NE. IOTCU6
      *      .AND. JJ .NE. IOTCP4
      *      .AND. JJ .NE. IOTCZR ) THEN
      FM(J, J, IXFM) = FM(J, J, IXFM) + FAS
      END IF
1240      CONTINUE
C
      END IF
C
      END IF
C
      IF (      IOHNO3 .GT. JBASE
      *      .AND. IOHNO3 .LE. JBASE + NCOMP ) THEN
C
      IF (      TYPE(IUNT) .EQ. 'TA'
      *      .AND. ACID(IUNT) .GT. ZERO ) THEN
C
      IF ( INDAS .EQ. (INDA-1) ) THEN
C
      J = IOHNO3
      DO 1320 K = 1 , NCOMP2
      DFDXL(J, K, INDA) = ZERO
      CONTINUE
1320      CONTINUE
C
      ELSE IF( INDAS .EQ. (INDA+1) ) THEN
C
      J = IOHNO3
      DO 1330 K = 1 , NCOMP2
      DFDXU(J, K, INDA) = ZERO
      CONTINUE
1330      CONTINUE
C
      ELSE
C
      IXFM = IUDSTA(3, KDEST, IUNTS)
      J = IOHNO3
      DO 1340 K = 1 , NCOMP2
      FM(J, K, IXFM) = ZERO
      CONTINUE
1340      CONTINUE
C
      END IF
C
      END IF
      END IF
C
      END IF
C
1400      CONTINUE
C
      END IF
C
      1500 CONTINUE
C
      DO 2200 IUNTS = 1 , NUNIT
C
      NDEST = MDEST(2, IUNTS)
C
      IF ( NDEST .GT. 0 ) THEN
C
      DO 2190 KDEST = 1 , NDEST
C
      NSTGS = MSTG (      IUNTS)
      IUNT = IUDSTO(1, KDEST, IUNTS)
      I = IUDSTO(2, KDEST, IUNTS)
      IS = NSTGS
C
      IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
      IMS = 2*IS
      ELSE
      IMS = IS
      END IF
C
      INDAS = IPOS(1, IMS, IUNTS)
      INDOS = IPOS(2, IMS, IUNTS)
C
      IF ( IUNT .NE. 0 ) THEN
C
      IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
      IM = 2*I - 1
      ELSE
      IM = I
      END IF
C
      INDA = IPOS(1, IM, IUNT)
      INDO = IPOS(2, IM, IUNT)
C
      FOS = FOUT(2, IUNTS)*RFOUT(KDEST, 2, IUNTS)
C
      IF ( TYPE(IUNT) .NE. 'MS' ) THEN
C
      DO 2110 J = 1 , NCOMP
      JJ = J + JBASE
      FX(J+NCOMP, INDO) = FX(J+NCOMP, INDO)
      *      - FOS*XN(JJ, 2, INDOS)
2110      CONTINUE
C
      IF ( INDOS .EQ. (INDO-1) ) THEN
C
      DO 2120 J = 1 , NCOMP
      DFDXL(J+NCOMP, J+NCOMP, INDO)
      *      = DFDXL(J+NCOMP, J+NCOMP, INDO) + FOS
2120      CONTINUE
C
      ELSE IF( INDOS .EQ. (INDO+1) ) THEN
C
      DO 2130 J = 1 , NCOMP
      DFDXU(J+NCOMP, J+NCOMP, INDO)
      *      = DFDXU(J+NCOMP, J+NCOMP, INDO) + FOS
2130      CONTINUE
C
      ELSE
C
      IXFM = IUDSTO(3, KDEST, IUNTS)
      DO 2140 J = 1 , NCOMP
      FM(J+NCOMP, J+NCOMP, IXFM)
      *      = FM(J+NCOMP, J+NCOMP, IXFM) + FOS
2140      CONTINUE
C
      END IF
C
      END IF
C
      IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
      DO 2150 J = 1 , NCOMP
      JJ = J + JBASE
      FX(J, INDO) = FX(J, INDO) - FOS*XN(JJ, 2, INDOS)
      CONTINUE
2150      CONTINUE
C

```

```

C          IF ( INDO .EQ. (INDO-1) ) THEN
C
C          DO 2160 J = 1 , NCOMP
C            DFDXL(J,J+NCOMP,INDO)
C            = DFDXL(J,J+NCOMP,INDO) + FOS
C          CONTINUE
C
C          ELSE IF( INDO .EQ. (INDO+1) ) THEN
C
C          DO 2170 J = 1 , NCOMP
C            DFDXU(J,J+NCOMP,INDO)
C            = DFDXU(J,J+NCOMP,INDO) + FOS
C          CONTINUE
C
C          ELSE
C
C            IXRM = IUDSTO(3,KDEST,IUNT)
C            DO 2180 J = 1 , NCOMP
C              RM(J,J+NCOMP,IXRM)
C              = RM(J,J+NCOMP,IXRM) + FOS
C            CONTINUE
C
C          END IF
C
C          END IF
C
C          END IF
C
C          CONTINUE
C
C          END IF
C
C          2190 CONTINUE
C
C          END IF
C
C          2200 CONTINUE
C
C
C          DO 3200 IFA = 1 , NFA
C
C            IUNT = IFDSTA(1,IFA)
C            I   = IFDSTA(2,IFA)
C
C            IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C              IM = 2*I - 1
C            ELSE
C              IM = I
C            END IF
C
C            INDA = IPOS(1,IM,IUNT)
C
C            FFA = FF(IFA,1)
C
C            DO 3100 J = 1 , NCOMP
C              JJ = J + JBASE
C              IF ( JJ .NE. IOTCU6
C                * .AND. JJ .NE. IOTCP4
C                * .AND. JJ .NE. IOTCZR ) THEN
C                FX(J,INDA) = FX(J,INDA) - FFA*XF(JJ,IFA,1)
C              END IF
C            CONTINUE
C
C          3100 CONTINUE
C
C          3200 CONTINUE
C
C
C          DO 3500 IFO = 1 , NFO
C
C            IUNT = IFDSTO(1,IFO)
C            I   = IFDSTO(2,IFO)
C
C            IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C              IM = 2*I - 1
C            ELSE
C              IM = I
C            END IF
C
C            INDO = IPOS(2,IM,IUNT)
C
C            FFO = FF(IFO,2)
C
C            IF ( TYPE(IUNT) .NE. 'MS' ) THEN
C
C              DO 3300 J = 1 , NCOMP
C                JJ = J + JBASE
C                FX(J+NCOMP,INDO) = FX(J+NCOMP,INDO) - FFO*XF(JJ,IFO,2)
C              CONTINUE
C
C            3300 CONTINUE
C

```

```

C          END IF
C
C          IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C          DO 3400 J = 1 , NCOMP
C            JJ = J + JBASE
C            FX(J,INDO) = FX(J,INDO) - FFO*XF(JJ,IFO,2)
C          3400 CONTINUE
C
C          END IF
C
C          3500 CONTINUE
C
C
C
C          IF ( IOHNO3 .GT. JBASE
C            * .AND. IOHNO3 .LE. JBASE + NCOMP ) THEN
C
C          DO 4200 IUNT = 1 , NUNIT
C
C            IF ( TYPE(IUNT) .EQ. 'TA'
C              * .AND. ACID(IUNT) .GT. ZERO ) THEN
C
C              I = 1
C              INDA = IPOS(1,I,IUNT)
C
C              J = IOHNO3
C              FX(J,INDA) = ACID(IUNT) - XN(J,I,INDA)
C
C              DO 4100 K = 1 , NCOMP2
C                DFDXL(J,K,INDA) = ZERO
C                DFDXD(J,K,INDA) = ZERO
C                DFDXU(J,K,INDA) = ZERO
C              4100 CONTINUE
C                DFDXD(J,J,INDA) = ONE
C
C            END IF
C
C          4200 CONTINUE
C
C          END IF
C
C
C          RETURN
C          END

```

```

SUBROUTINE MATBL(IFOUT ,
*      TITLE , TIME , ENDTIM , DT ,
*      MXMCOMP , MXDEST , MXCOMP , MXUNIT ,
*      MXSTG2 , MXBLK , NUNIT ,
*      FMASS ,
*      RXLIM , EPS ,
*      MXFEED , NFA , NFO ,
*      FF , XF , IFDSTA , IFDSTO ,
*      ID , TYPE , MSTG , IUDSTA , IUDSTO ,
*      MDEST , MEXTRA , TAV , VTOT , ACID ,
*      ACIDIN , OXID , FIN , FOUT , RFOUT ,
*      FMSIN , FMSOUT , FMTIN , FMTOUT , RMTBL ,
*      IPOS , XN )
C
C MATERIAL BALANCE
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C
C COMMON /IXCOMP/NCHEMO,
*      IOHNO3 , IOU6 , IOPU4 , IOPU3 , IOU4 ,
*      IOHNO2 , IOHYD , IOHAN , IOPU5 , IOPU6 ,
*      IONP4 , IONP5 , IONP6 , IOZR , IOZC4 ,
*      IOTC5 , IOTC6 , IOTC7 , IOTC6 , IOTCP4 ,
*      IOTCZR , IOSR , IORUDI , IORUTR , IORUNI ,
*      IORUMO , IOCS , IOCE , IOGD , IOAM ,
*      IOCM
C
C COMMON /IICOMP/I1HNO3 , I1U , I1PU , I1NP , I1ZR ,
*      I1TC , I1SR , I1RU , I1CS , I1CE ,
*      I1GD , I1AM , I1CM
C
C CHARACTER*72 TITLE
C
C DIMENSION FMASS (MXCOMP)
C
C DIMENSION FF( MXFEED , 2 ) ,
*      XF (MXCOMP , MXFEED , 2 ) ,
*      IFDSTA ( 2 , MXFEED ) ,
*      IFDSTO ( 2 , MXFEED )
C
C DIMENSION ID ( MXUNIT )
C
C CHARACTER*2 TYPE ( MXUNIT )
C
C DIMENSION MSTG ( MXUNIT ) ,
*      IUDSTA ( 3 , MXDEST , MXUNIT ) ,
*      IUDSTO ( 3 , MXDEST , MXUNIT )
C DIMENSION MDEST ( 2 , MXUNIT ) ,
*      MEXTRA ( MXUNIT )
C
C DIMENSION TAV ( MXUNIT ) ,
*      VTOT ( MXUNIT ) ,
*      ACID ( MXUNIT ) ,
*      ACIDIN ( MXUNIT ) ,
*      OXID ( MXUNIT ) ,
*      FIN ( 2 , MXUNIT ) ,
*      FOUT ( 2 , MXUNIT ) ,
*      RFOUT ( MXDEST , 2 , MXUNIT ) ,
*      FMSIN ( MXMCOMP , 2 , MXUNIT ) ,
*      FMSOUT ( MXMCOMP , 2 , MXUNIT )
C
C DIMENSION FMTIN ( MXMCOMP , 2 ) ,
*      FMTOUT ( MXMCOMP , 2 ) ,
*      RMTBL ( MXMCOMP , 3 )
C
C DIMENSION IPOS ( 2 , MXSTG2 , MXUNIT )
C
C DIMENSION XN ( MXCOMP , 2 , MXBLK )
C
C DATA ZERO / 0.000 /
C DATA ONE / 1.000 /
C
C FMSEPS = RXLIM * EPS
C FFEPS = 1.0D-100
C
C WRITE ( IFOUT , 9000 ) TITLE , TIME

```

```

... MATERIAL BALANCE OF EACH UNIT ...
C
C
C DO 200 IUNT = 1 , NUNIT
C
C     FIN ( 1 , IUNT ) = ZERO
C     FIN ( 2 , IUNT ) = ZERO
C
C     DO 100 J = 1 , MXMCOMP
C
C         FMSIN ( J , 1 , IUNT ) = ZERO
C         FMSIN ( J , 2 , IUNT ) = ZERO
C
C         FMSOUT ( J , 1 , IUNT ) = ZERO
C         FMSOUT ( J , 2 , IUNT ) = ZERO
C
C     100 CONTINUE
C     200 CONTINUE
C
C
C DO 1400 IUNTS = 1 , NUNIT
C
C     NSTGS = MSTG ( IUNTS )
C     NDEST = MDEST ( 1 , IUNTS )
C     IS = 1
C
C     IF ( TYPE ( IUNTS ) .EQ. 'MS' ) THEN
C         IMS = 2 * IS
C     ELSE
C         IMS = IS
C     END IF
C
C     INDAS = IPOS ( 1 , IMS , IUNTS )
C     INDOS = IPOS ( 2 , IMS , IUNTS )
C
C     IF ( NDEST .GT. 0 ) THEN
C
C         DO 1200 KDEST = 1 , NDEST
C
C             IUNT = IUDSTA ( 1 , KDEST , IUNTS )
C             I = IUDSTA ( 2 , KDEST , IUNTS )
C
C             FAS = FOUT ( 1 , IUNTS ) * RFOUT ( KDEST , 1 , IUNTS )
C
C             IF ( IUNT .NE. 0 ) THEN
C
C                 FIN ( I , IUNT ) = FIN ( I , IUNT ) + FAS
C
C                 FMSIN ( IOHNO3 , 1 , IUNT ) = FMSIN ( IOHNO3 , 1 , IUNT )
C                 *      + FAS * XN ( IOHNO3 , 1 , INDAS )
C
C                 FMSIN ( I1U , 1 , IUNT ) = FMSIN ( I1U , 1 , IUNT )
C                 *      + FAS * XN ( IOU6 , 1 , INDAS )
C                 *      + FAS * XN ( IOU4 , 1 , INDAS )
C
C                 FMSIN ( I1PU , 1 , IUNT ) = FMSIN ( I1PU , 1 , IUNT )
C                 *      + FAS * XN ( IOPU4 , 1 , INDAS )
C                 *      + FAS * XN ( IOPU3 , 1 , INDAS )
C                 *      + FAS * XN ( IOPU5 , 1 , INDAS )
C                 *      + FAS * XN ( IOPU6 , 1 , INDAS )
C
C                 FMSIN ( I1NP , 1 , IUNT ) = FMSIN ( I1NP , 1 , IUNT )
C                 *      + FAS * XN ( IONP4 , 1 , INDAS )
C                 *      + FAS * XN ( IONP5 , 1 , INDAS )
C                 *      + FAS * XN ( IONP6 , 1 , INDAS )
C
C                 FMSIN ( I1ZR , 1 , IUNT ) = FMSIN ( I1ZR , 1 , IUNT )
C                 *      + FAS * XN ( IOZR , 1 , INDAS )
C
C                 FMSIN ( I1TC , 1 , IUNT ) = FMSIN ( I1TC , 1 , IUNT )
C                 *      + FAS * XN ( IOTC4 , 1 , INDAS )
C                 *      + FAS * XN ( IOTC5 , 1 , INDAS )
C                 *      + FAS * XN ( IOTC6 , 1 , INDAS )
C                 *      + FAS * XN ( IOTC7 , 1 , INDAS )
C
C                 FMSIN ( I1SR , 1 , IUNT ) = FMSIN ( I1SR , 1 , IUNT )
C                 *      + FAS * XN ( IOSR , 1 , INDAS )
C
C                 FMSIN ( I1RU , 1 , IUNT ) = FMSIN ( I1RU , 1 , IUNT )
C                 *      + FAS * XN ( IORUDI , 1 , INDAS )
C                 *      + FAS * XN ( IORUTR , 1 , INDAS )
C                 *      + FAS * XN ( IORUNI , 1 , INDAS )
C                 *      + FAS * XN ( IORUMO , 1 , INDAS )
C
C                 FMSIN ( I1CS , 1 , IUNT ) = FMSIN ( I1CS , 1 , IUNT )
C                 *      + FAS * XN ( IOCS , 1 , INDAS )

```

```

*       FMSIN(11CE ,1, IUNT) = FMSIN(11CE ,1, IUNT)
      + FAS*XN(10CE ,1, INDAS)
C
*       FMSIN(11GD ,1, IUNT) = FMSIN(11GD ,1, IUNT)
      + FAS*XN(10GD ,1, INDAS)
C
*       FMSIN(11AM ,1, IUNT) = FMSIN(11AM ,1, IUNT)
      + FAS*XN(10AM ,1, INDAS)
C
*       FMSIN(11CM ,1, IUNT) = FMSIN(11CM ,1, IUNT)
      + FAS*XN(10CM ,1, INDAS)
C
DO 1100 J = 14 , MAXCMP
      FMSIN(J,1, IUNT) = FMSIN(J,1, IUNT)
      + FAS*XN(J+18 ,1, INDAS)
*
1100 CONTINUE
C
      END IF
C
C
1200 CONTINUE
C
      END IF
C
      FAS = FOUT(1, IUNTS)
C
      FMSOUT(11HN03,1, IUNTS) = FMSOUT(11HN03,1, IUNTS)
      + FAS*XN(10HN03,1, INDAS)
C
*       FMSOUT(11U ,1, IUNTS) = FMSOUT(11U ,1, IUNTS)
      + FAS*XN(10U6 ,1, INDAS)
*       + FAS*XN(10U4 ,1, INDAS)
C
*       FMSOUT(11PU ,1, IUNTS) = FMSOUT(11PU ,1, IUNTS)
      + FAS*XN(10PU4 ,1, INDAS)
*       + FAS*XN(10PU3 ,1, INDAS)
*       + FAS*XN(10PU5 ,1, INDAS)
*       + FAS*XN(10PU6 ,1, INDAS)
C
*       FMSOUT(11NP ,1, IUNTS) = FMSOUT(11NP ,1, IUNTS)
      + FAS*XN(10NP4 ,1, INDAS)
*       + FAS*XN(10NP5 ,1, INDAS)
*       + FAS*XN(10NP6 ,1, INDAS)
C
*       FMSOUT(11ZR ,1, IUNTS) = FMSOUT(11ZR ,1, IUNTS)
      + FAS*XN(10ZR ,1, INDAS)
C
*       FMSOUT(11TC ,1, IUNTS) = FMSOUT(11TC ,1, IUNTS)
      + FAS*XN(10TC4 ,1, INDAS)
*       + FAS*XN(10TC5 ,1, INDAS)
*       + FAS*XN(10TC6 ,1, INDAS)
*       + FAS*XN(10TC7 ,1, INDAS)
C
*       FMSOUT(11SR ,1, IUNTS) = FMSOUT(11SR ,1, IUNTS)
      + FAS*XN(10SR ,1, INDAS)
C
*       FMSOUT(11RU ,1, IUNTS) = FMSOUT(11RU ,1, IUNTS)
      + FAS*XN(10RUD1 ,1, INDAS)
*       + FAS*XN(10RUTR ,1, INDAS)
*       + FAS*XN(10RUN1 ,1, INDAS)
*       + FAS*XN(10RUM0 ,1, INDAS)
C
*       FMSOUT(11CS ,1, IUNTS) = FMSOUT(11CS ,1, IUNTS)
      + FAS*XN(10CS ,1, INDAS)
C
*       FMSOUT(11CE ,1, IUNTS) = FMSOUT(11CE ,1, IUNTS)
      + FAS*XN(10CE ,1, INDAS)
C
*       FMSOUT(11GD ,1, IUNTS) = FMSOUT(11GD ,1, IUNTS)
      + FAS*XN(10GD ,1, INDAS)
C
*       FMSOUT(11AM ,1, IUNTS) = FMSOUT(11AM ,1, IUNTS)
      + FAS*XN(10AM ,1, INDAS)
C
*       FMSOUT(11CM ,1, IUNTS) = FMSOUT(11CM ,1, IUNTS)
      + FAS*XN(10CM ,1, INDAS)
C
DO 1300 J = 14 , MAXCMP
      FMSOUT(J,1, IUNTS) = FMSOUT(J,1, IUNTS)
      + FAS*XN(J+18 ,1, INDAS)
*
1300 CONTINUE
C
1400 CONTINUE
C
C
C
DO 2400 IUNTS = 1 , NUNIT

```

```

NSTGS = MSTG ( IUNTS)
NDEST = MDEST(2, IUNTS)
IS = NSTGS
C
IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
      IMS = 2*IS
ELSE
      IMS = IS
END IF
C
INDAS = IPOS(1, IMS, IUNTS)
INDOS = IPOS(2, IMS, IUNTS)
C
IF ( NDEST .GT. 0 ) THEN
C
DO 2200 KDEST = 1 , NDEST
C
      IUNT = IUDSTO(1, KDEST, IUNTS)
      I = IUDSTO(2, KDEST, IUNTS)
C
      FOS = FOUT(2, IUNTS)*RFOUT(KDEST, 2, IUNTS)
C
      IF ( IUNT .NE. 0 ) THEN
C
          FIN ( , 2, IUNT) = FIN ( , 2, IUNT) + FOS
C
          FMSIN(11HN03,2, IUNT) = FMSIN(11HN03,2, IUNT)
          + FOS*XN(10HN03,2, INDOS)
C
          *       FMSIN(11U ,2, IUNT) = FMSIN(11U ,2, IUNT)
          *       + FOS*XN(10U6 ,2, INDOS)
          *       + FOS*XN(10U4 ,2, INDOS)
          *       + FOS*XN(10UCU6,2, INDOS)
C
          FMSIN(11PU ,2, IUNT) = FMSIN(11PU ,2, IUNT)
          + FOS*XN(10PU4 ,2, INDOS)
          + FOS*XN(10PU3 ,2, INDOS)
          + FOS*XN(10PU5 ,2, INDOS)
          + FOS*XN(10PU6 ,2, INDOS)
          + FOS*XN(10TCP4,2, INDOS)
C
          FMSIN(11NP ,2, IUNT) = FMSIN(11NP ,1, IUNT)
          + FOS*XN(10NP4 ,2, INDOS)
          + FOS*XN(10NP5 ,2, INDOS)
          + FOS*XN(10NP6 ,2, INDOS)
C
          FMSIN(11ZR ,2, IUNT) = FMSIN(11ZR ,2, IUNT)
          + FOS*XN(10ZR ,2, INDOS)
          + FOS*XN(10TCZR,2, INDOS)
C
          FMSIN(11TC ,2, IUNT) = FMSIN(11TC ,2, IUNT)
          + FOS*XN(10TC4 ,2, INDOS)
          + FOS*XN(10TC5 ,2, INDOS)
          + FOS*XN(10TC6 ,2, INDOS)
          + FOS*XN(10TC7 ,2, INDOS)
          + FOS*XN(10TCU6,2, INDOS)
          + FOS*XN(10TCP4,2, INDOS)
          + FOS*XN(10TCZR,2, INDOS)
C
          FMSIN(11SR ,2, IUNT) = FMSIN(11SR ,2, IUNT)
          + FOS*XN(10SR ,2, INDOS)
C
          FMSIN(11RU ,2, IUNT) = FMSIN(11RU ,2, IUNT)
          + FOS*XN(10RUD1,2, INDOS)
          + FOS*XN(10RUTR,2, INDOS)
          + FOS*XN(10RUN1,2, INDOS)
          + FOS*XN(10RUM0,2, INDOS)
C
          FMSIN(11CS ,2, IUNT) = FMSIN(11CS ,2, IUNT)
          + FOS*XN(10CS ,2, INDOS)
C
          FMSIN(11CE ,2, IUNT) = FMSIN(11CE ,2, IUNT)
          + FOS*XN(10CE ,2, INDOS)
C
          FMSIN(11GD ,2, IUNT) = FMSIN(11GD ,2, IUNT)
          + FOS*XN(10GD ,2, INDOS)
C
          FMSIN(11AM ,2, IUNT) = FMSIN(11AM ,2, IUNT)
          + FOS*XN(10AM ,2, INDOS)
C
          FMSIN(11CM ,2, IUNT) = FMSIN(11CM ,2, IUNT)
          + FOS*XN(10CM ,2, INDOS)
C
          DO 2100 J = 14 , MAXCMP
          FMSIN(J,2, IUNT) = FMSIN(J,2, IUNT)
          + FOS*XN(J+18 ,2, INDOS)
          *
          CONTINUE
          C
          2100
      C

```



```

END IF
C
C
2200 CONTINUE
C
END IF
C
FOS = FOUT(2, IUNTS)
C
FMSOUT(11HN03, 2, IUNTS) = FMSOUT(11HN03, 2, IUNTS)
* + FOS*XN(10HN03, 2, INDOS)
C
FMSOUT(11U , 2, IUNTS) = FMSOUT(11U , 2, IUNTS)
* + FOS*XN(10U6 , 2, INDOS)
* + FOS*XN(10U4 , 2, INDOS)
* + FOS*XN(10TCU6, 2, INDOS)
C
FMSOUT(11PU , 2, IUNTS) = FMSOUT(11PU , 2, IUNTS)
* + FOS*XN(10PU4 , 2, INDOS)
* + FOS*XN(10PU3 , 2, INDOS)
* + FOS*XN(10PU5 , 2, INDOS)
* + FOS*XN(10PU6 , 2, INDOS)
* + FOS*XN(10TCP4, 2, INDOS)
C
FMSOUT(11NP , 2, IUNTS) = FMSOUT(11NP , 2, IUNTS)
* + FOS*XN(10NP4 , 2, INDOS)
* + FOS*XN(10NP5 , 2, INDOS)
* + FOS*XN(10NP6 , 2, INDOS)
C
FMSOUT(11ZR , 2, IUNTS) = FMSOUT(11ZR , 2, IUNTS)
* + FOS*XN(10ZR , 2, INDOS)
* + FOS*XN(10TCZR, 2, INDOS)
C
FMSOUT(11TC , 2, IUNTS) = FMSOUT(11TC , 2, IUNTS)
* + FOS*XN(10TC4 , 2, INDOS)
* + FOS*XN(10TC5 , 2, INDOS)
* + FOS*XN(10TC6 , 2, INDOS)
* + FOS*XN(10TC7 , 2, INDOS)
* + FOS*XN(10TCU6, 2, INDOS)
* + FOS*XN(10TCP4, 2, INDOS)
* + FOS*XN(10TCZR, 2, INDOS)
C
FMSOUT(11SR , 2, IUNTS) = FMSOUT(11SR , 2, IUNTS)
* + FOS*XN(10SR , 2, INDOS)
C
FMSOUT(11RU , 2, IUNTS) = FMSOUT(11RU , 2, IUNTS)
* + FOS*XN(10RUD1, 2, INDOS)
* + FOS*XN(10RUTR, 2, INDOS)
* + FOS*XN(10RUN1, 2, INDOS)
* + FOS*XN(10RUM0, 2, INDOS)
C
FMSOUT(11CS , 2, IUNTS) = FMSOUT(11CS , 2, IUNTS)
* + FOS*XN(10CS , 2, INDOS)
C
FMSOUT(11CE , 2, IUNTS) = FMSOUT(11CE , 2, IUNTS)
* + FOS*XN(10CE , 2, INDOS)
C
FMSOUT(11GD , 2, IUNTS) = FMSOUT(11GD , 2, IUNTS)
* + FOS*XN(10GD , 2, INDOS)
C
FMSOUT(11AM , 2, IUNTS) = FMSOUT(11AM , 2, IUNTS)
* + FOS*XN(10AM , 2, INDOS)
C
FMSOUT(11CM , 2, IUNTS) = FMSOUT(11CM , 2, IUNTS)
* + FOS*XN(10CM , 2, INDOS)
C
DO 2300 J = 14 , MXMCMF
* FMSOUT(J, 2, IUNTS) = FMSOUT(J, 2, IUNTS)
* + FOS*XN(J+18 , 2, INDOS)
2300 CONTINUE
C
2400 CONTINUE
C
C
DO 3120 IFA = 1 , NFA
C
IUNT = IFDSTA(1, IFA)
I = IFDSTA(2, IFA)
C
FFA = FF(IFA, 1)
C
FIN ( 1, IUNT) = FIN ( 1, IUNT) + FFA
C
FMSIN(11HN03, 1, IUNT) = FMSIN(11HN03, 1, IUNT)
* + FFA*XF(10HN03, IFA, 1)
C
FMSIN(11U , 1, IUNT) = FMSIN(11U , 1, IUNT)

```

```

* + FFA*XF(10U6 , IFA, 1)
* + FFA*XF(10U4 , IFA, 1)
C
FMSIN(11PU , 1, IUNT) = FMSIN(11PU , 1, IUNT)
* + FFA*XF(10PU4 , IFA, 1)
* + FFA*XF(10PU3 , IFA, 1)
* + FFA*XF(10PU5 , IFA, 1)
* + FFA*XF(10PU6 , IFA, 1)
C
FMSIN(11NP , 1, IUNT) = FMSIN(11NP , 1, IUNT)
* + FFA*XF(10NP4 , IFA, 1)
* + FFA*XF(10NP5 , IFA, 1)
* + FFA*XF(10NP6 , IFA, 1)
C
FMSIN(11ZR , 1, IUNT) = FMSIN(11ZR , 1, IUNT)
* + FFA*XF(10ZR , IFA, 1)
C
FMSIN(11TC , 1, IUNT) = FMSIN(11TC , 1, IUNT)
* + FFA*XF(10TC4 , IFA, 1)
* + FFA*XF(10TC5 , IFA, 1)
* + FFA*XF(10TC6 , IFA, 1)
* + FFA*XF(10TC7 , IFA, 1)
C
FMSIN(11SR , 1, IUNT) = FMSIN(11SR , 1, IUNT)
* + FFA*XF(10SR , IFA, 1)
C
FMSIN(11RU , 1, IUNT) = FMSIN(11RU , 1, IUNT)
* + FFA*XF(10RUD1, IFA, 1)
* + FFA*XF(10RUTR, IFA, 1)
* + FFA*XF(10RUN1, IFA, 1)
* + FFA*XF(10RUM0, IFA, 1)
C
FMSIN(11CS , 1, IUNT) = FMSIN(11CS , 1, IUNT)
* + FFA*XF(10CS , IFA, 1)
C
FMSIN(11CE , 1, IUNT) = FMSIN(11CE , 1, IUNT)
* + FFA*XF(10CE , IFA, 1)
C
FMSIN(11GD , 1, IUNT) = FMSIN(11GD , 1, IUNT)
* + FFA*XF(10GD , IFA, 1)
C
FMSIN(11AM , 1, IUNT) = FMSIN(11AM , 1, IUNT)
* + FFA*XF(10AM , IFA, 1)
C
FMSIN(11CM , 1, IUNT) = FMSIN(11CM , 1, IUNT)
* + FFA*XF(10CM , IFA, 1)
C
DO 3110 J = 14 , MXMCMF
* FMSIN(J, 1, IUNT) = FMSIN(J, 1, IUNT)
* + FFA*XF(J+18 , IFA, 1)
3110 CONTINUE
C
3120 CONTINUE
C
C
DO 3220 IFO = 1 , NFO
C
IUNT = IFDSTO(1, IFO)
I = IFDSTO(2, IFO)
C
FFO = FF(IFO, 2)
C
FIN ( 2, IUNT) = FIN ( 2, IUNT) + FFO
C
FMSIN(11HN03, 2, IUNT) = FMSIN(11HN03, 2, IUNT)
* + FFO*XF(10HN03, IFO, 2)
C
FMSIN(11U , 2, IUNT) = FMSIN(11U , 2, IUNT)
* + FFO*XF(10U6 , IFO, 2)
* + FFO*XF(10U4 , IFO, 2)
* + FFO*XF(10TCU6, IFO, 2)
C
FMSIN(11PU , 2, IUNT) = FMSIN(11PU , 2, IUNT)
* + FFO*XF(10PU4 , IFO, 2)
* + FFO*XF(10PU3 , IFO, 2)
* + FFO*XF(10PU5 , IFO, 2)
* + FFO*XF(10PU6 , IFO, 2)
* + FFO*XF(10TCP4, IFO, 2)
C
FMSIN(11NP , 2, IUNT) = FMSIN(11NP , 2, IUNT)
* + FFO*XF(10NP4 , IFO, 2)
* + FFO*XF(10NP5 , IFO, 2)
* + FFO*XF(10NP6 , IFO, 2)
C
FMSIN(11ZR , 2, IUNT) = FMSIN(11ZR , 2, IUNT)
* + FFO*XF(10ZR , IFO, 2)
* + FFO*XF(10TCZR, IFO, 2)
C

```

```

*   FMSIN(11TC ,2, IUNT) = FMSIN(11TC ,2, IUNT)
*   + FFO*XF(10TC4 ,1FO,2)
*   + FFO*XF(10TC5 ,1FO,2)
*   + FFO*XF(10TC6 ,1FO,2)
*   + FFO*XF(10TC7 ,1FO,2)
*   + FFO*XF(10TCU6, 1FO,2)
*   + FFO*XF(10TCP4, 1FO,2)
*   + FFO*XF(10TCZR, 1FO,2)
C
*   FMSIN(11SR ,2, IUNT) = FMSIN(11SR ,2, IUNT)
*   + FFO*XF(10SR ,1FO,2)
C
*   FMSIN(11RU ,2, IUNT) = FMSIN(11RU ,2, IUNT)
*   + FFO*XF(10RUD1, 1FO,2)
*   + FFO*XF(10RUTR, 1FO,2)
*   + FFO*XF(10RUN1, 1FO,2)
*   + FFO*XF(10RUM0, 1FO,2)
C
*   FMSIN(11CS ,2, IUNT) = FMSIN(11CS ,2, IUNT)
*   + FFO*XF(10CS ,1FO,2)
C
*   FMSIN(11CE ,2, IUNT) = FMSIN(11CE ,2, IUNT)
*   + FFO*XF(10CE ,1FO,2)
C
*   FMSIN(11GD ,2, IUNT) = FMSIN(11GD ,2, IUNT)
*   + FFO*XF(10GD ,1FO,2)
C
*   FMSIN(11AM ,2, IUNT) = FMSIN(11AM ,2, IUNT)
*   + FFO*XF(10AM ,1FO,2)
C
*   FMSIN(11CM ,2, IUNT) = FMSIN(11CM ,2, IUNT)
*   + FFO*XF(10CM ,1FO,2)
C
DO 3210 J = 14 , MAXCMP
*   FMSIN(J,2, IUNT) = FMSIN(J,2, IUNT)
*   + FFO*XF(J+18 ,1FO,2)
3210 CONTINUE
C
3220 CONTINUE
C
DO 3230 IUNT = 1 , NUNIT
C
*   FMSIN (11U ,1, IUNT) = FMSIN (11U ,1, IUNT)*FMASS(10U6 )
*   FMSIN (11PU ,1, IUNT) = FMSIN (11PU ,1, IUNT)*FMASS(10PU4 )
*   FMSIN (11NP ,1, IUNT) = FMSIN (11NP ,1, IUNT)*FMASS(10NP6 )
*   FMSIN (11ZR ,1, IUNT) = FMSIN (11ZR ,1, IUNT)*FMASS(10ZR )
*   FMSIN (11TC ,1, IUNT) = FMSIN (11TC ,1, IUNT)*FMASS(10TC7 )
*   FMSIN (11SR ,1, IUNT) = FMSIN (11SR ,1, IUNT)*FMASS(10SR )
*   FMSIN (11RU ,1, IUNT) = FMSIN (11RU ,1, IUNT)*FMASS(10RUD1)
*   FMSIN (11CS ,1, IUNT) = FMSIN (11CS ,1, IUNT)*FMASS(10CS )
*   FMSIN (11CE ,1, IUNT) = FMSIN (11CE ,1, IUNT)*FMASS(10CE )
*   FMSIN (11GD ,1, IUNT) = FMSIN (11GD ,1, IUNT)*FMASS(10GD )
*   FMSIN (11AM ,1, IUNT) = FMSIN (11AM ,1, IUNT)*FMASS(10AM )
*   FMSIN (11CM ,1, IUNT) = FMSIN (11CM ,1, IUNT)*FMASS(10CM )
C
*   FMSIN (11U ,2, IUNT) = FMSIN (11U ,2, IUNT)*FMASS(10U6 )
*   FMSIN (11PU ,2, IUNT) = FMSIN (11PU ,2, IUNT)*FMASS(10PU4 )
*   FMSIN (11NP ,2, IUNT) = FMSIN (11NP ,2, IUNT)*FMASS(10NP6 )
*   FMSIN (11ZR ,2, IUNT) = FMSIN (11ZR ,2, IUNT)*FMASS(10ZR )
*   FMSIN (11TC ,2, IUNT) = FMSIN (11TC ,2, IUNT)*FMASS(10TC7 )
*   FMSIN (11SR ,2, IUNT) = FMSIN (11SR ,2, IUNT)*FMASS(10SR )
*   FMSIN (11RU ,2, IUNT) = FMSIN (11RU ,2, IUNT)*FMASS(10RUD1)
*   FMSIN (11CS ,2, IUNT) = FMSIN (11CS ,2, IUNT)*FMASS(10CS )
*   FMSIN (11CE ,2, IUNT) = FMSIN (11CE ,2, IUNT)*FMASS(10CE )
*   FMSIN (11GD ,2, IUNT) = FMSIN (11GD ,2, IUNT)*FMASS(10GD )
*   FMSIN (11AM ,2, IUNT) = FMSIN (11AM ,2, IUNT)*FMASS(10AM )
*   FMSIN (11CM ,2, IUNT) = FMSIN (11CM ,2, IUNT)*FMASS(10CM )
C
*   FMSOUT(11U ,1, IUNT) = FMSOUT(11U ,1, IUNT)*FMASS(10U6 )
*   FMSOUT(11PU ,1, IUNT) = FMSOUT(11PU ,1, IUNT)*FMASS(10PU4 )
*   FMSOUT(11NP ,1, IUNT) = FMSOUT(11NP ,1, IUNT)*FMASS(10NP6 )
*   FMSOUT(11ZR ,1, IUNT) = FMSOUT(11ZR ,1, IUNT)*FMASS(10ZR )
*   FMSOUT(11TC ,1, IUNT) = FMSOUT(11TC ,1, IUNT)*FMASS(10TC7 )
*   FMSOUT(11SR ,1, IUNT) = FMSOUT(11SR ,1, IUNT)*FMASS(10SR )
*   FMSOUT(11RU ,1, IUNT) = FMSOUT(11RU ,1, IUNT)*FMASS(10RUD1)
*   FMSOUT(11CS ,1, IUNT) = FMSOUT(11CS ,1, IUNT)*FMASS(10CS )
*   FMSOUT(11CE ,1, IUNT) = FMSOUT(11CE ,1, IUNT)*FMASS(10CE )
*   FMSOUT(11GD ,1, IUNT) = FMSOUT(11GD ,1, IUNT)*FMASS(10GD )
*   FMSOUT(11AM ,1, IUNT) = FMSOUT(11AM ,1, IUNT)*FMASS(10AM )
*   FMSOUT(11CM ,1, IUNT) = FMSOUT(11CM ,1, IUNT)*FMASS(10CM )
C
*   FMSOUT(11U ,2, IUNT) = FMSOUT(11U ,2, IUNT)*FMASS(10U6 )
*   FMSOUT(11PU ,2, IUNT) = FMSOUT(11PU ,2, IUNT)*FMASS(10PU4 )
*   FMSOUT(11NP ,2, IUNT) = FMSOUT(11NP ,2, IUNT)*FMASS(10NP6 )
*   FMSOUT(11ZR ,2, IUNT) = FMSOUT(11ZR ,2, IUNT)*FMASS(10ZR )
*   FMSOUT(11TC ,2, IUNT) = FMSOUT(11TC ,2, IUNT)*FMASS(10TC7 )
*   FMSOUT(11SR ,2, IUNT) = FMSOUT(11SR ,2, IUNT)*FMASS(10SR )

```

```

FMSOUT(11RU ,2, IUNT) = FMSOUT(11RU ,2, IUNT)*FMASS(10RUD1)
FMSOUT(11CS ,2, IUNT) = FMSOUT(11CS ,2, IUNT)*FMASS(10CS )
FMSOUT(11CE ,2, IUNT) = FMSOUT(11CE ,2, IUNT)*FMASS(10CE )
FMSOUT(11GD ,2, IUNT) = FMSOUT(11GD ,2, IUNT)*FMASS(10GD )
FMSOUT(11AM ,2, IUNT) = FMSOUT(11AM ,2, IUNT)*FMASS(10AM )
FMSOUT(11CM ,2, IUNT) = FMSOUT(11CM ,2, IUNT)*FMASS(10CM )
C
3230 CONTINUE
C
WRITE(1FOUT,9010)
C
DO 3240 IUNT = 1 , NUNIT
C
IF ( TYPE(IUNT) .EQ. 'TA' .AND. ACID(IUNT) .GT. 0.000 ) THEN
C
I = 1
INDAS = IPOS(1,1, IUNT)
INDOS = IPOS(2,1, IUNT)
ACIDXN = XN(10HN03,1,INDAS)
FACID = FOUT(1, IUNT) - FIN(1, IUNT)
C
IF ( ACIDIN(IUNT) .GT. 0.000 .AND. FACID .GT. 0.000 ) THEN
*   HPRECL = ( FMSOUT(11HN03,1, IUNT) - FMSIN(11HN03,1, IUNT) )
*   /FACID
ELSE
HPRECL = - 1.00+10
END IF
C
IF ( ACIDIN(IUNT) .GT. 0.000
.AND. HPRECL .GT. 0.000 ) THEN
*   HDIFF = 1.000 - ACIDIN(IUNT)/HPRECL
ELSE
HDIFF = - 1.00+10
END IF
C
WRITE(1FOUT,9020) IUNT, ID(IUNT), TYPE(IUNT),
*   ACIDXN
*   ACIDIN( IUNT),
*   FMSIN (11HN03,1, IUNT),
*   FMSOUT(11HN03,1, IUNT),
*   FACID
*   HPRECL
*   HDIFF
C
ELSE IF ( TYPE(IUNT) .EQ. 'TA' ) THEN
C
I = 1
INDAS = IPOS(1,1, IUNT)
INDOS = IPOS(2,1, IUNT)
ACIDXN = XN(10HN03,1,INDAS)
C
WRITE(1FOUT,9020) IUNT, ID(IUNT), TYPE(IUNT),
*   ACIDXN
*   ACIDIN( IUNT),
*   FMSIN (11HN03,1, IUNT),
*   FMSOUT(11HN03,1, IUNT)
C
ELSE IF ( TYPE(IUNT) .EQ. 'TO' ) THEN
C
I = 1
INDAS = IPOS(1,1, IUNT)
INDOS = IPOS(2,1, IUNT)
ACIDXN = XN(10HN03,2,INDOS)
C
WRITE(1FOUT,9020) IUNT, ID(IUNT), TYPE(IUNT),
*   ACIDXN
*   ACIDIN( IUNT),
*   FMSIN (11HN03,2, IUNT),
*   FMSOUT(11HN03,2, IUNT)
C
ELSE
C
WRITE(1FOUT,9030) IUNT, ID(IUNT), TYPE(IUNT),
*   FMSIN (11HN03,1, IUNT)
*   + FMSIN (11HN03,2, IUNT),
*   FMSOUT(11HN03,1, IUNT)
*   + FMSOUT(11HN03,2, IUNT)
C
END IF
C
3240 CONTINUE
C
WRITE(1FOUT,9110)
C
DO 3320 IUNT = 1 , NUNIT

```

```

C
DO 3310 J = 1 , MXMCP
C
TOTIN = FMSIN(J,1,IUNT) + FMSIN(J,2,IUNT)
TOTFIN = FIN ( 1,IUNT) + FIN ( 2,IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
3310 CONTINUE
C
WRITE(IFOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSIN (J,1,IUNT),
*
FMSIN (J,2,IUNT),
*
FMSIN (J,1,IUNT) + FMSIN (J,2,IUNT),
*
J = 2 , 4 )
C
WRITE(IFOUT,9220) ( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 2 , 4 )
C
WRITE(IFOUT,9230) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 2 , 4 )
C
3320 CONTINUE
C
WRITE(IFOUT,9115)
C
DO IUNT = 1 , NUNIT
C
DO J = 1 , MXMCP
C
TOTIN = FMSIN(J,1,IUNT) + FMSIN(J,2,IUNT)
TOTFIN = FIN ( 1,IUNT) + FIN ( 2,IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
ENDDO
C
WRITE(IFOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSIN (J,1,IUNT),
*
FMSIN (J,2,IUNT),
*
FMSIN (J,1,IUNT) + FMSIN (J,2,IUNT),
*
J = 5 , 7 )
C
WRITE(IFOUT,9220) ( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 5 , 7 )
C
WRITE(IFOUT,9230) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 5 , 7 )
C
ENDDO
C
WRITE(IFOUT,9120)
C
DO 3340 IUNT = 1 , NUNIT
C
DO 3330 J = 1 , MXMCP
C
TOTIN = FMSIN(J,1,IUNT) + FMSIN(J,2,IUNT)
TOTFIN = FIN ( 1,IUNT) + FIN ( 2,IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
END IF
C
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
3330 CONTINUE
C
WRITE(IFOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSIN (J,1,IUNT),
*
FMSIN (J,2,IUNT),
*
FMSIN (J,1,IUNT) + FMSIN (J,2,IUNT),
*
J = 8 , 10 )
C
WRITE(IFOUT,9220) ( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 8 , 10 )
C
WRITE(IFOUT,9230) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 8 , 10 )
C
3340 CONTINUE
C
WRITE(IFOUT,9130)
C
DO 3360 IUNT = 1 , NUNIT
C
DO 3350 J = 1 , MXMCP
C
TOTIN = FMSIN(J,1,IUNT) + FMSIN(J,2,IUNT)
TOTFIN = FIN ( 1,IUNT) + FIN ( 2,IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
3350 CONTINUE
C
WRITE(IFOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSIN (J,1,IUNT),
*
FMSIN (J,2,IUNT),
*
FMSIN (J,1,IUNT) + FMSIN (J,2,IUNT),
*
J = 11 , 13 )
C
WRITE(IFOUT,9220) ( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 11 , 13 )
C
WRITE(IFOUT,9230) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 11 , 13 )
C
3360 CONTINUE
C
WRITE(IFOUT,9140)
C
DO 3380 IUNT = 1 , NUNIT
C
DO 3370 J = 1 , MXMCP
C
TOTIN = FMSIN(J,1,IUNT) + FMSIN(J,2,IUNT)
TOTFIN = FIN ( 1,IUNT) + FIN ( 2,IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
END IF

```

```

RMTBL(J,3) = -1.000
END IF
C
3370 CONTINUE
C
WRITE(1FOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
* ( FMSIN (J,1, IUNT),
* FMSIN (J,2, IUNT),
* FMSIN (J,1, IUNT) + FMSIN (J,2, IUNT),
* J = 14 , 16 )
C
WRITE(1FOUT,9220) ( FMSOUT(J,1, IUNT),
* FMSOUT(J,2, IUNT),
* FMSOUT(J,1, IUNT) + FMSOUT(J,2, IUNT),
* J = 14 , 16 )
C
WRITE(1FOUT,9230) ( RMTBL (J,1),
* RMTBL (J,2),
* RMTBL (J,3),
* J = 14 , 16 )
C
3380 CONTINUE
C
C
WRITE(1FOUT,9145)
C
DO IUNT = 1 , NUNIT
C
DO J = 1 , MXMCMF
C
TOTIN = FMSIN(J,1, IUNT) + FMSIN(J,2, IUNT)
TOTFIN = FIN ( 1, IUNT) + FIN ( 2, IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1, IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2, IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1, IUNT)/TOTIN
* + FMSOUT(J,2, IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
ENDDO
C
WRITE(1FOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
* ( FMSIN (J,1, IUNT),
* FMSIN (J,2, IUNT),
* FMSIN (J,1, IUNT) + FMSIN (J,2, IUNT),
* J = 17 , 19 )
C
WRITE(1FOUT,9220) ( FMSOUT(J,1, IUNT),
* FMSOUT(J,2, IUNT),
* FMSOUT(J,1, IUNT) + FMSOUT(J,2, IUNT),
* J = 17 , 19 )
C
WRITE(1FOUT,9230) ( RMTBL (J,1),
* RMTBL (J,2),
* RMTBL (J,3),
* J = 17 , 19 )
C
ENDDO
C
C
WRITE(1FOUT,9150)
C
DO 3400 IUNT = 1 , NUNIT
C
DO 3390 J = 1 , MXMCMF
C
TOTIN = FMSIN(J,1, IUNT) + FMSIN(J,2, IUNT)
TOTFIN = FIN ( 1, IUNT) + FIN ( 2, IUNT) + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1, IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2, IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1, IUNT)/TOTIN
* + FMSOUT(J,2, IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
3390 CONTINUE
C
WRITE(1FOUT,9210) IUNT, ID(IUNT), TYPE(IUNT),
* ( FMSIN (J,1, IUNT),

```

```

* FMSIN (J,2, IUNT),
* FMSIN (J,1, IUNT) + FMSIN (J,2, IUNT),
* J = 20 , 22 )
C
WRITE(1FOUT,9220) ( FMSOUT(J,1, IUNT),
* FMSOUT(J,2, IUNT),
* FMSOUT(J,1, IUNT) + FMSOUT(J,2, IUNT),
* J = 20 , 22 )
C
WRITE(1FOUT,9230) ( RMTBL (J,1),
* RMTBL (J,2),
* RMTBL (J,3),
* J = 20 , 22 )
C
3400 CONTINUE
C
C
... MAETERIAL BALANCE OF SYSTEM ...
C
DO 4120 J = 1 , MXMCMF
C
FMTIN (J,1) = ZERO
FMTIN (J,2) = ZERO
C
FMTOUT(J,1) = ZERO
FMTOUT(J,2) = ZERO
C
DO 4110 IUNT = 1 , NUNIT
C
FMSOUT(J,1, IUNT) = ZERO
FMSOUT(J,2, IUNT) = ZERO
C
4110 CONTINUE
C
4120 CONTINUE
C
C
DO 4240 IUNTS = 1 , NUNIT
C
NSTGS = MSTG ( IUNTS)
NDEST = MDEST(1, IUNTS)
IS = 1
C
IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
IMS = 2*IS
ELSE
IMS = IS
END IF
C
INDAS = IPOS(1, IMS, IUNTS)
INDOS = IPOS(2, IMS, IUNTS)
C
IF ( NDEST .GT. 0 ) THEN
C
DO 4220 KDEST = 1 , NDEST
C
IUNT = IUDESTA(1, KDEST, IUNTS)
I = IUDESTA(2, KDEST, IUNTS)
C
FAS = FOUT(1, IUNTS)*RFOUT(KDEST, 1, IUNTS)
C
C
IF ( IUNT .EQ. 0 ) THEN
C
FMSOUT(11HN03, 1, IUNTS) = FMSOUT(11HN03, 1, IUNTS)
* + FAS*XN(10HN03, 1, INDAS)
C
FMSOUT(11U , 1, IUNTS) = FMSOUT(11U , 1, IUNTS)
* + FAS*XN(10U6 , 1, INDAS)
* + FAS*XN(10U4 , 1, INDAS)
C
FMSOUT(11PU , 1, IUNTS) = FMSOUT(11PU , 1, IUNTS)
* + FAS*XN(10PU4 , 1, INDAS)
* + FAS*XN(10PU3 , 1, INDAS)
* + FAS*XN(10PU5 , 1, INDAS)
* + FAS*XN(10PU6 , 1, INDAS)
C
FMSOUT(11NP , 1, IUNTS) = FMSOUT(11NP , 1, IUNTS)
* + FAS*XN(10NP4 , 1, INDAS)
* + FAS*XN(10NP5 , 1, INDAS)
* + FAS*XN(10NP6 , 1, INDAS)
C
FMSOUT(11ZR , 1, IUNTS) = FMSOUT(11ZR , 1, IUNTS)
* + FAS*XN(10ZR , 1, INDAS)
C
FMSOUT(11TC , 1, IUNTS) = FMSOUT(11TC , 1, IUNTS)
* + FAS*XN(10TC4 , 1, INDAS)

```

```

*          + FAS*%N(10TC5 ,1,INDAS)
*          + FAS*%N(10TC6 ,1,INDAS)
*          + FAS*%N(10TC7 ,1,INDAS)
C
FMSOUT(11SR ,1,IUNTS) = FMSOUT(11SR ,1,IUNTS)
*          + FAS*%N(10SR ,1,INDAS)
C
FMSOUT(11RU ,1,IUNTS) = FMSOUT(11RU ,1,IUNTS)
*          + FAS*%N(10RUD1,1,INDAS)
*          + FAS*%N(10RUTR,1,INDAS)
*          + FAS*%N(10RUN1,1,INDAS)
*          + FAS*%N(10RUM0,1,INDAS)
C
FMSOUT(11CS ,1,IUNTS) = FMSOUT(11CS ,1,IUNTS)
*          + FAS*%N(10CS ,1,INDAS)
C
FMSOUT(11CE ,1,IUNTS) = FMSOUT(11CE ,1,IUNTS)
*          + FAS*%N(10CE ,1,INDAS)
C
FMSOUT(11GD ,1,IUNTS) = FMSOUT(11GD ,1,IUNTS)
*          + FAS*%N(10GD ,1,INDAS)
C
FMSOUT(11AM ,1,IUNTS) = FMSOUT(11AM ,1,IUNTS)
*          + FAS*%N(10AM ,1,INDAS)
C
FMSOUT(11CM ,1,IUNTS) = FMSOUT(11CM ,1,IUNTS)
*          + FAS*%N(10CM ,1,INDAS)
C
DO 4210 J = 14 , MXMCMF
*          FMSOUT(J,1,IUNTS) = FMSOUT(J,1,IUNTS)
*          + FAS*%N(J+18 ,1,INDAS)
4210      CONTINUE
C
      END IF
C
4220      CONTINUE
C
      ELSE
C
      FAS = FOUT(1,IUNTS)
C
      FMSOUT(11HN03,1,IUNTS) = FMSOUT(11HN03,1,IUNTS)
*          + FAS*%N(10HN03,1,INDAS)
C
      FMSOUT(11U ,1,IUNTS) = FMSOUT(11U ,1,IUNTS)
*          + FAS*%N(10U6 ,1,INDAS)
*          + FAS*%N(10U4 ,1,INDAS)
C
      FMSOUT(11PU ,1,IUNTS) = FMSOUT(11PU ,1,IUNTS)
*          + FAS*%N(10PU4 ,1,INDAS)
*          + FAS*%N(10PU3 ,1,INDAS)
*          + FAS*%N(10PU5 ,1,INDAS)
*          + FAS*%N(10PU6 ,1,INDAS)
C
      FMSOUT(11NP ,1,IUNTS) = FMSOUT(11NP ,1,IUNTS)
*          + FAS*%N(10NP4 ,1,INDAS)
*          + FAS*%N(10NP5 ,1,INDAS)
*          + FAS*%N(10NP6 ,1,INDAS)
C
      FMSOUT(11ZR ,1,IUNTS) = FMSOUT(11ZR ,1,IUNTS)
*          + FAS*%N(10ZR ,1,INDAS)
C
      FMSOUT(11TC ,1,IUNTS) = FMSOUT(11TC ,1,IUNTS)
*          + FAS*%N(10TC4 ,1,INDAS)
*          + FAS*%N(10TC5 ,1,INDAS)
*          + FAS*%N(10TC6 ,1,INDAS)
*          + FAS*%N(10TC7 ,1,INDAS)
C
      FMSOUT(11SR ,1,IUNTS) = FMSOUT(11SR ,1,IUNTS)
*          + FAS*%N(10SR ,1,INDAS)
C
      FMSOUT(11RU ,1,IUNTS) = FMSOUT(11RU ,1,IUNTS)
*          + FAS*%N(10RUD1,1,INDAS)
*          + FAS*%N(10RUTR,1,INDAS)
*          + FAS*%N(10RUN1,1,INDAS)
*          + FAS*%N(10RUM0,1,INDAS)
C
      FMSOUT(11CS ,1,IUNTS) = FMSOUT(11CS ,1,IUNTS)
*          + FAS*%N(10CS ,1,INDAS)
C
      FMSOUT(11CE ,1,IUNTS) = FMSOUT(11CE ,1,IUNTS)
*          + FAS*%N(10CE ,1,INDAS)
C
      FMSOUT(11GD ,1,IUNTS) = FMSOUT(11GD ,1,IUNTS)
*          + FAS*%N(10GD ,1,INDAS)
C
      FMSOUT(11AM ,1,IUNTS) = FMSOUT(11AM ,1,IUNTS)
*          + FAS*%N(10AM ,1,INDAS)

```

```

C          FMSOUT(11CM ,1,IUNTS) = FMSOUT(11CM ,1,IUNTS)
*          + FAS*%N(10CM ,1,INDAS)
C
DO 4230 J = 14 , MXMCMF
*          FMSOUT(J,1,IUNTS) = FMSOUT(J,1,IUNTS)
*          + FAS*%N(J+18 ,1,INDAS)
4230      CONTINUE
C
      END IF
C
4240      CONTINUE
C
DO 4340 IUNTS = 1 , NUNIT
C
      NSTGS = MSTG ( IUNTS)
      NDEST = MDEST(2,IUNTS)
      IS = NSTGS
C
      IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
          IMS = 2*IS
      ELSE
          IMS = IS
      END IF
C
      IINDAS = IPOS(1,IMS,IUNTS)
      IINDOS = IPOS(2,IMS,IUNTS)
C
      IF ( NDEST .GT. 0 ) THEN
C
          DO 4320 KDEST = 1 , NDEST
C
              IUNT = IUDSTO(1,KDEST,IUNTS)
              I = IUDSTO(2,KDEST,IUNTS)
C
              FOS = FOUT(2,IUNTS)*FOUT(KDEST,2,IUNTS)
C
              IF ( IUNT .EQ. 0 ) THEN
C
                  FMSOUT(11HN03,2,IUNTS) = FMSOUT(11HN03,2,IUNTS)
*                  + FOS*%N(10HN03,2,INDOS)
C
                  FMSOUT(11U ,2,IUNTS) = FMSOUT(11U ,2,IUNTS)
*                  + FOS*%N(10U6 ,2,INDOS)
*                  + FOS*%N(10U4 ,2,INDOS)
*                  + FOS*%N(10UCU6,2,INDOS)
C
                  FMSOUT(11PU ,2,IUNTS) = FMSOUT(11PU ,2,IUNTS)
*                  + FOS*%N(10PU4 ,2,INDOS)
*                  + FOS*%N(10PU3 ,2,INDOS)
*                  + FOS*%N(10PU5 ,2,INDOS)
*                  + FOS*%N(10PU6 ,2,INDOS)
*                  + FOS*%N(10TCP4,2,INDOS)
C
                  FMSOUT(11NP ,2,IUNTS) = FMSOUT(11NP ,2,IUNTS)
*                  + FOS*%N(10NP4 ,2,INDOS)
*                  + FOS*%N(10NP5 ,2,INDOS)
*                  + FOS*%N(10NP6 ,2,INDOS)
C
                  FMSOUT(11ZR ,2,IUNTS) = FMSOUT(11ZR ,2,IUNTS)
*                  + FOS*%N(10ZR ,2,INDOS)
*                  + FOS*%N(10TCZR,2,INDOS)
C
                  FMSOUT(11TC ,2,IUNTS) = FMSOUT(11TC ,2,IUNTS)
*                  + FOS*%N(10TC4 ,2,INDOS)
*                  + FOS*%N(10TC5 ,2,INDOS)
*                  + FOS*%N(10TC6 ,2,INDOS)
*                  + FOS*%N(10TC7 ,2,INDOS)
*                  + FOS*%N(10TCU6,2,INDOS)
*                  + FOS*%N(10TCP4,2,INDOS)
*                  + FOS*%N(10TCZR,2,INDOS)
C
                  FMSOUT(11SR ,2,IUNTS) = FMSOUT(11SR ,2,IUNTS)
*                  + FOS*%N(10SR ,2,INDOS)
C
                  FMSOUT(11RU ,2,IUNTS) = FMSOUT(11RU ,2,IUNTS)
*                  + FOS*%N(10RUD1,2,INDOS)
*                  + FOS*%N(10RUTR,2,INDOS)
*                  + FOS*%N(10RUN1,2,INDOS)
*                  + FOS*%N(10RUM0,2,INDOS)
C
                  FMSOUT(11CS ,2,IUNTS) = FMSOUT(11CS ,2,IUNTS)
*                  + FOS*%N(10CS ,2,INDOS)

```

```

*      FMSOUT(11CE ,2,1UNTS) = FMSOUT(11CE ,2,1UNTS)
      + FOS*%N(10CE ,2,INDOS)
C
*      FMSOUT(11GD ,2,1UNTS) = FMSOUT(11GD ,2,1UNTS)
      + FOS*%N(10GD ,2,INDOS)
C
*      FMSOUT(11AM ,2,1UNTS) = FMSOUT(11AM ,2,1UNTS)
      + FOS*%N(10AM ,2,INDOS)
C
*      FMSOUT(11CM ,2,1UNTS) = FMSOUT(11CM ,2,1UNTS)
      + FOS*%N(10CM ,2,INDOS)
C
DO 4310 J = 14 , MXMCMF
      FMSOUT(J,2,1UNTS) = FMSOUT(J,2,1UNTS)
      + FOS*%N(J+18 ,2,INDOS)
4310 CONTINUE
C
      END IF
C
4320 CONTINUE
C
ELSE
      FOS = FOUT(2,1UNTS)
C
      FMSOUT(11HNO3,2,1UNTS) = FMSOUT(11HNO3,2,1UNTS)
      + FOS*%N(10HNO3,2,INDOS)
C
      FMSOUT(11U ,2,1UNTS) = FMSOUT(11U ,2,1UNTS)
      + FOS*%N(10U6 ,2,INDOS)
      + FOS*%N(10U4 ,2,INDOS)
      + FOS*%N(10TCU6,2,INDOS)
C
      FMSOUT(11PU ,2,1UNTS) = FMSOUT(11PU ,2,1UNTS)
      + FOS*%N(10PU4 ,2,INDOS)
      + FOS*%N(10PU3 ,2,INDOS)
      + FOS*%N(10PU5 ,2,INDOS)
      + FOS*%N(10PU6 ,2,INDOS)
      + FOS*%N(10TCP4,2,INDOS)
C
      FMSOUT(11NP ,2,1UNTS) = FMSOUT(11NP ,2,1UNTS)
      + FOS*%N(10NP4 ,2,INDOS)
      + FOS*%N(10NP5 ,2,INDOS)
      + FOS*%N(10NP6 ,2,INDOS)
C
      FMSOUT(11ZR ,2,1UNTS) = FMSOUT(11ZR ,2,1UNTS)
      + FOS*%N(10ZR ,2,INDOS)
      + FOS*%N(10TCZR,2,INDOS)
C
      FMSOUT(11TC ,2,1UNTS) = FMSOUT(11TC ,2,1UNTS)
      + FOS*%N(10TC4 ,2,INDOS)
      + FOS*%N(10TC5 ,2,INDOS)
      + FOS*%N(10TC6 ,2,INDOS)
      + FOS*%N(10TC7 ,2,INDOS)
      + FOS*%N(10TCU6,2,INDOS)
      + FOS*%N(10TCP4,2,INDOS)
      + FOS*%N(10TCZR,2,INDOS)
C
      FMSOUT(11SR ,2,1UNTS) = FMSOUT(11SR ,2,1UNTS)
      + FOS*%N(10SR ,2,INDOS)
C
      FMSOUT(11RU ,2,1UNTS) = FMSOUT(11RU ,2,1UNTS)
      + FOS*%N(10RU01,2,INDOS)
      + FOS*%N(10RUTR,2,INDOS)
      + FOS*%N(10RUM1,2,INDOS)
      + FOS*%N(10RUM0,2,INDOS)
C
      FMSOUT(11CS ,2,1UNTS) = FMSOUT(11CS ,2,1UNTS)
      + FOS*%N(10CS ,2,INDOS)
C
      FMSOUT(11CE ,2,1UNTS) = FMSOUT(11CE ,2,1UNTS)
      + FOS*%N(10CE ,2,INDOS)
C
      FMSOUT(11GD ,2,1UNTS) = FMSOUT(11GD ,2,1UNTS)
      + FOS*%N(10GD ,2,INDOS)
C
      FMSOUT(11AM ,2,1UNTS) = FMSOUT(11AM ,2,1UNTS)
      + FOS*%N(10AM ,2,INDOS)
C
      FMSOUT(11CM ,2,1UNTS) = FMSOUT(11CM ,2,1UNTS)
      + FOS*%N(10CM ,2,INDOS)
C
DO 4330 J = 14 , MXMCMF
      FMSOUT(J,2,1UNTS) = FMSOUT(J,2,1UNTS)
      + FOS*%N(J+18 ,2,INDOS)
4330 CONTINUE

```

```

C
      END IF
C
4340 CONTINUE
C
      FINA = ZERO
      FINO = ZERO
C
DO 5120 IFA = 1 , NFA
C
      IUNT = IFDSTA(1,IFA)
      I = IFDSTA(2,IFA)
C
      FFA = FF(IFA,1)
      FINA = FINA + FF(IF0,2)
C
      FMTIN(11HNO3,1) = FMTIN(11HNO3,1)
      + FFA*%X(10HNO3,IFA,1)
C
      FMTIN(11U ,1) = FMTIN(11U ,1)
      + FFA*%X(10U6 ,IFA,1)
      + FFA*%X(10U4 ,IFA,1)
C
      FMTIN(11PU ,1) = FMTIN(11PU ,1)
      + FFA*%X(10PU4 ,IFA,1)
      + FFA*%X(10PU3 ,IFA,1)
      + FFA*%X(10PU5 ,IFA,1)
      + FFA*%X(10PU6 ,IFA,1)
C
      FMTIN(11NP ,1) = FMTIN(11NP ,1)
      + FFA*%X(10NP4 ,IFA,1)
      + FFA*%X(10NP5 ,IFA,1)
      + FFA*%X(10NP6 ,IFA,1)
C
      FMTIN(11ZR ,1) = FMTIN(11ZR ,1)
      + FFA*%X(10ZR ,IFA,1)
C
      FMTIN(11TC ,1) = FMTIN(11TC ,1)
      + FFA*%X(10TC4 ,IFA,1)
      + FFA*%X(10TC5 ,IFA,1)
      + FFA*%X(10TC6 ,IFA,1)
      + FFA*%X(10TC7 ,IFA,1)
C
      FMTIN(11SR ,1) = FMTIN(11SR ,1)
      + FFA*%X(10SR ,IFA,1)
C
      FMTIN(11RU ,1) = FMTIN(11RU ,1)
      + FFA*%X(10RU01,IFA,1)
      + FFA*%X(10RUTR,IFA,1)
      + FFA*%X(10RUM1,IFA,1)
      + FFA*%X(10RUM0,IFA,1)
C
      FMTIN(11CS ,1) = FMTIN(11CS ,1)
      + FFA*%X(10CS ,IFA,1)
C
      FMTIN(11CE ,1) = FMTIN(11CE ,1)
      + FFA*%X(10CE ,IFA,1)
C
      FMTIN(11GD ,1) = FMTIN(11GD ,1)
      + FFA*%X(10GD ,IFA,1)
C
      FMTIN(11AM ,1) = FMTIN(11AM ,1)
      + FFA*%X(10AM ,IFA,1)
C
      FMTIN(11CM ,1) = FMTIN(11CM ,1)
      + FFA*%X(10CM ,IFA,1)
C
DO 5110 J = 14 , MXMCMF
      FMTIN(J ,1) = FMTIN(J ,1)
      + FFA*%X(J+18 ,IFA,1)
5110 CONTINUE
C
5120 CONTINUE
C
DO 5220 IFO = 1 , NFO
C
      IUNT = IFDSTO(1,IFO)
      I = IFDSTO(2,IFO)
C
      FFO = FF(IFO,2)
      FINO = FINO + FF(IFO,2)
C

```

```

*   FMTIN(11HN03,2) = FMTIN(11HN03,2)
*   + FFO*XF(10HN03,1F0,2)
C
*   FMTIN(11U ,2) = FMTIN(11U ,2)
*   + FFO*XF(10U6 ,1F0,2)
*   + FFO*XF(10U4 ,1F0,2)
*   + FFO*XF(10TCU6,1F0,2)
C
*   FMTIN(11PU ,2) = FMTIN(11PU ,2)
*   + FFO*XF(10PU4 ,1F0,2)
*   + FFO*XF(10PU3 ,1F0,2)
*   + FFO*XF(10PU5 ,1F0,2)
*   + FFO*XF(10PU6 ,1F0,2)
*   + FFO*XF(10TCP4,1F0,2)
C
*   FMTIN(11NP ,2) = FMTIN(11NP ,2)
*   + FFO*XF(10NP4 ,1F0,2)
*   + FFO*XF(10NP5 ,1F0,2)
*   + FFO*XF(10NP6 ,1F0,2)
C
*   FMTIN(11ZR ,2) = FMTIN(11ZR ,2)
*   + FFO*XF(10ZR ,1F0,2)
*   + FFO*XF(10TCZR,1F0,2)
C
*   FMTIN(11TC ,2) = FMTIN(11TC ,2)
*   + FFO*XF(10TC4 ,1F0,2)
*   + FFO*XF(10TC5 ,1F0,2)
*   + FFO*XF(10TC6 ,1F0,2)
*   + FFO*XF(10TC7 ,1F0,2)
*   + FFO*XF(10TCU6,1F0,2)
*   + FFO*XF(10TCP4,1F0,2)
*   + FFO*XF(10TCZR,1F0,2)
C
*   FMTIN(11SR ,2) = FMTIN(11SR ,2)
*   + FFO*XF(10SR ,1F0,2)
C
*   FMTIN(11RU ,2) = FMTIN(11RU ,2)
*   + FFO*XF(10RUDI,1F0,2)
*   + FFO*XF(10RUR,1F0,2)
*   + FFO*XF(10RUNI,1F0,2)
*   + FFO*XF(10RUMO,1F0,2)
C
*   FMTIN(11CS ,2) = FMTIN(11CS ,2)
*   + FFO*XF(10CS ,1F0,2)
C
*   FMTIN(11CE ,2) = FMTIN(11CE ,2)
*   + FFO*XF(10CE ,1F0,2)
C
*   FMTIN(11GD ,2) = FMTIN(11GD ,2)
*   + FFO*XF(10GD ,1F0,2)
C
*   FMTIN(11AM ,2) = FMTIN(11AM ,2)
*   + FFO*XF(10AM ,1F0,2)
C
*   FMTIN(11CM ,2) = FMTIN(11CM ,2)
*   + FFO*XF(10CM ,1F0,2)
C
DO 5210 J = 14 , MXMCM
*   FMTIN(J ,2) = FMTIN(J ,2)
*   + FFO*XF(J+18 ,1F0,2)
5210 CONTINUE
C
5220 CONTINUE
C
C
C
*   FMTIN(11U ,1) = FMTIN(11U ,1)*FMASS(10U6 )
*   FMTIN(11PU ,1) = FMTIN(11PU ,1)*FMASS(10PU4 )
*   FMTIN(11NP ,1) = FMTIN(11NP ,1)*FMASS(10NP6 )
*   FMTIN(11ZR ,1) = FMTIN(11ZR ,1)*FMASS(10ZR )
*   FMTIN(11TC ,1) = FMTIN(11TC ,1)*FMASS(10TC7 )
*   FMTIN(11SR ,1) = FMTIN(11SR ,1)*FMASS(10SR )
*   FMTIN(11RU ,1) = FMTIN(11RU ,1)*FMASS(10RUDI)
*   FMTIN(11CS ,1) = FMTIN(11CS ,1)*FMASS(10CS )
*   FMTIN(11CE ,1) = FMTIN(11CE ,1)*FMASS(10CE )
*   FMTIN(11GD ,1) = FMTIN(11GD ,1)*FMASS(10GD )
*   FMTIN(11AM ,1) = FMTIN(11AM ,1)*FMASS(10AM )
*   FMTIN(11CM ,1) = FMTIN(11CM ,1)*FMASS(10CM )
C
*   FMTIN(11U ,2) = FMTIN(11U ,2)*FMASS(10U6 )
*   FMTIN(11PU ,2) = FMTIN(11PU ,2)*FMASS(10PU4 )
*   FMTIN(11NP ,2) = FMTIN(11NP ,2)*FMASS(10NP6 )
*   FMTIN(11ZR ,2) = FMTIN(11ZR ,2)*FMASS(10ZR )
*   FMTIN(11TC ,2) = FMTIN(11TC ,2)*FMASS(10TC7 )
*   FMTIN(11SR ,2) = FMTIN(11SR ,2)*FMASS(10SR )
*   FMTIN(11RU ,2) = FMTIN(11RU ,2)*FMASS(10RUDI)
*   FMTIN(11CS ,2) = FMTIN(11CS ,2)*FMASS(10CS )
*   FMTIN(11CE ,2) = FMTIN(11CE ,2)*FMASS(10CE )
*   FMTIN(11GD ,2) = FMTIN(11GD ,2)*FMASS(10GD )

```

```

*   FMTIN(11AM ,2) = FMTIN(11AM ,2)*FMASS(10AM )
*   FMTIN(11CM ,2) = FMTIN(11CM ,2)*FMASS(10CM )
C
DO 5230 IUNT = 1 , NJUNIT
C
*   FMSOUT(11U ,1,IUNT) = FMSOUT(11U ,1,IUNT)*FMASS(10U6 )
*   FMSOUT(11PU ,1,IUNT) = FMSOUT(11PU ,1,IUNT)*FMASS(10PU4 )
*   FMSOUT(11NP ,1,IUNT) = FMSOUT(11NP ,1,IUNT)*FMASS(10NP6 )
*   FMSOUT(11ZR ,1,IUNT) = FMSOUT(11ZR ,1,IUNT)*FMASS(10ZR )
*   FMSOUT(11TC ,1,IUNT) = FMSOUT(11TC ,1,IUNT)*FMASS(10TC7 )
*   FMSOUT(11SR ,1,IUNT) = FMSOUT(11SR ,1,IUNT)*FMASS(10SR )
*   FMSOUT(11RU ,1,IUNT) = FMSOUT(11RU ,1,IUNT)*FMASS(10RUDI)
*   FMSOUT(11CS ,1,IUNT) = FMSOUT(11CS ,1,IUNT)*FMASS(10CS )
*   FMSOUT(11CE ,1,IUNT) = FMSOUT(11CE ,1,IUNT)*FMASS(10CE )
*   FMSOUT(11GD ,1,IUNT) = FMSOUT(11GD ,1,IUNT)*FMASS(10GD )
*   FMSOUT(11AM ,1,IUNT) = FMSOUT(11AM ,1,IUNT)*FMASS(10AM )
*   FMSOUT(11CM ,1,IUNT) = FMSOUT(11CM ,1,IUNT)*FMASS(10CM )
C
*   FMSOUT(11U ,2,IUNT) = FMSOUT(11U ,2,IUNT)*FMASS(10U6 )
*   FMSOUT(11PU ,2,IUNT) = FMSOUT(11PU ,2,IUNT)*FMASS(10PU4 )
*   FMSOUT(11NP ,2,IUNT) = FMSOUT(11NP ,2,IUNT)*FMASS(10NP6 )
*   FMSOUT(11ZR ,2,IUNT) = FMSOUT(11ZR ,2,IUNT)*FMASS(10ZR )
*   FMSOUT(11TC ,2,IUNT) = FMSOUT(11TC ,2,IUNT)*FMASS(10TC7 )
*   FMSOUT(11SR ,2,IUNT) = FMSOUT(11SR ,2,IUNT)*FMASS(10SR )
*   FMSOUT(11RU ,2,IUNT) = FMSOUT(11RU ,2,IUNT)*FMASS(10RUDI)
*   FMSOUT(11CS ,2,IUNT) = FMSOUT(11CS ,2,IUNT)*FMASS(10CS )
*   FMSOUT(11CE ,2,IUNT) = FMSOUT(11CE ,2,IUNT)*FMASS(10CE )
*   FMSOUT(11GD ,2,IUNT) = FMSOUT(11GD ,2,IUNT)*FMASS(10GD )
*   FMSOUT(11AM ,2,IUNT) = FMSOUT(11AM ,2,IUNT)*FMASS(10AM )
*   FMSOUT(11CM ,2,IUNT) = FMSOUT(11CM ,2,IUNT)*FMASS(10CM )
C
5230 CONTINUE
C
C
C
DO 5320 J = 1 , MXMCM
C
*   FMTOUT(J,1) = ZERO
*   FMTOUT(J,2) = ZERO
C
DO 5310 IUNT = 1 , NJUNIT
*   FMTOUT(J,1) = FMTOUT(J,1) + FMSOUT(J,1,IUNT)
*   FMTOUT(J,2) = FMTOUT(J,2) + FMSOUT(J,2,IUNT)
5310 CONTINUE
C
5320 CONTINUE
C
C
C
WRITE(1FOUT,9310)
C
*   WRITE(1FOUT,9410) ( FMTIN (J,1),
*   FMTIN (J,2),
*   FMTIN (J,1) + FMTIN (J,2),
*   J = 2 , 4 )
C
DO 5420 IUNT = 1 , NJUNIT
C
DO 5410 J = 1 , MXMCM
C
*   TOTIN = FMTIN(J,1) + FMTIN(J,2)
*   TOTFIN = FINA + FINO + FFEPS
C
C
C
IF ( TOTIN .GE. FMASS*TOTFIN ) THEN
*   RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
*   RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
*   RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*   + FMSOUT(J,2,IUNT)/TOTIN
*
ELSE
*   RMTBL(J,1) = -1.000
*   RMTBL(J,2) = -1.000
*   RMTBL(J,3) = -1.000
*
END IF
C
5410 CONTINUE
C
*   WRITE(1FOUT,9420) IUNT,1D(IUNT),TYPE(IUNT),
*   ( FMSOUT(J,1,IUNT),
*   FMSOUT(J,2,IUNT),
*   FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*   J = 2 , 4 )
C
*   WRITE(1FOUT,9430) ( RMTBL (J,1),
*   RMTBL (J,2),
*   RMTBL (J,3),
*   J = 2 , 4 )
C

```

```

5420 CONTINUE
C
C
DO 5430 J = 1 , MXMCMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMTOUT(J,1)/TOTIN
RMTBL(J,2) = FMTOUT(J,2)/TOTIN
RMTBL(J,3) = FMTOUT(J,1)/TOTIN
*
+ FMTOUT(J,2)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
5430 CONTINUE
C
WRITE(1FOUT,9440) ( FMTOUT(J,1),
*
FMTOUT(J,2),
*
FMTOUT(J,1) + FMTOUT(J,2),
*
J = 2 , 4 )
C
WRITE(1FOUT,9430) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 2 , 4 )
C
C
WRITE(1FOUT,9315)
C
WRITE(1FOUT,9410) ( FMTIN (J,1),
*
FMTIN (J,2),
*
FMTIN (J,1) + FMTIN (J,2),
*
J = 5 , 7 )
C
DO IUNT = 1 , NUNIT
C
DO J = 1 , MXMCMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
ENDDO
C
WRITE(1FOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 5 , 7 )
C
WRITE(1FOUT,9430) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 5 , 7 )
C
ENDDO
C
DO J = 1 , MXMCMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMTOUT(J,1)/TOTIN
RMTBL(J,2) = FMTOUT(J,2)/TOTIN
RMTBL(J,3) = FMTOUT(J,1)/TOTIN
*
+ FMTOUT(J,2)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF

```

```

C
ENDDO
C
WRITE(1FOUT,9440) ( FMTOUT(J,1),
*
FMTOUT(J,2),
*
FMTOUT(J,1) + FMTOUT(J,2),
*
J = 5 , 7 )
C
WRITE(1FOUT,9430) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 5 , 7 )
C
C
WRITE(1FOUT,9320)
C
WRITE(1FOUT,9410) ( FMTIN (J,1),
*
FMTIN (J,2),
*
FMTIN (J,1) + FMTIN (J,2),
*
J = 8 , 10 )
C
DO 5520 IUNT = 1 , NUNIT
C
DO 5510 J = 1 , MXMCMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*
+ FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
5510 CONTINUE
C
WRITE(1FOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
*
( FMSOUT(J,1,IUNT),
*
FMSOUT(J,2,IUNT),
*
FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*
J = 8 , 10 )
C
WRITE(1FOUT,9430) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 8 , 10 )
C
5520 CONTINUE
C
DO 5530 J = 1 , MXMCMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMTOUT(J,1)/TOTIN
RMTBL(J,2) = FMTOUT(J,2)/TOTIN
RMTBL(J,3) = FMTOUT(J,1)/TOTIN
*
+ FMTOUT(J,2)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
5530 CONTINUE
C
WRITE(1FOUT,9440) ( FMTOUT(J,1),
*
FMTOUT(J,2),
*
FMTOUT(J,1) + FMTOUT(J,2),
*
J = 8 , 10 )
C
WRITE(1FOUT,9430) ( RMTBL (J,1),
*
RMTBL (J,2),
*
RMTBL (J,3),
*
J = 8 , 10 )
C
C
WRITE(1FOUT,9330)
C
WRITE(1FOUT,9410) ( FMTIN (J,1),
*
FMTIN (J,2),
*
FMTIN (J,1) + FMTIN (J,2),
*
J = 11 , 13 )

```



```

C      DO 5620 IUNT = 1 , NUNIT
C
C      DO 5610 J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
C              RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
C              RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
C              RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
C              *
C              + FMSOUT(J,2,IUNT)/TOTIN
C          ELSE
C              RMTBL(J,1) = -1.000
C              RMTBL(J,2) = -1.000
C              RMTBL(J,3) = -1.000
C          END IF
C
C      5610 CONTINUE
C
C      WRITE(IFOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
C      *
C      *      ( FMSOUT(J,1,IUNT),
C      *      FMSOUT(J,2,IUNT),
C      *      FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
C      *      J = 11 , 13 )
C
C      WRITE(IFOUT,9430) ( RMTBL (J,1),
C      *
C      *      RMTBL (J,2),
C      *      RMTBL (J,3),
C      *      J = 11 , 13 )
C
C      5620 CONTINUE
C
C      DO 5630 J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
C              RMTBL(J,1) = FMTOUT(J,1)/TOTIN
C              RMTBL(J,2) = FMTOUT(J,2)/TOTIN
C              RMTBL(J,3) = FMTOUT(J,1)/TOTIN
C              *
C              + FMTOUT(J,2)/TOTIN
C          ELSE
C              RMTBL(J,1) = -1.000
C              RMTBL(J,2) = -1.000
C              RMTBL(J,3) = -1.000
C          END IF
C
C      5630 CONTINUE
C
C      WRITE(IFOUT,9440) ( FMTOUT(J,1),
C      *
C      *      FMTOUT(J,2),
C      *      FMTOUT(J,1) + FMTOUT(J,2),
C      *      J = 11 , 13 )
C
C      WRITE(IFOUT,9430) ( RMTBL (J,1),
C      *
C      *      RMTBL (J,2),
C      *      RMTBL (J,3),
C      *      J = 11 , 13 )
C
C      WRITE(IFOUT,9340)
C
C      WRITE(IFOUT,9410) ( FMTIN (J,1),
C      *
C      *      FMTIN (J,2),
C      *      FMTIN (J,1) + FMTIN (J,2),
C      *      J = 14 , 16 )
C
C      DO 5720 IUNT = 1 , NUNIT
C
C      DO 5710 J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
C              RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
C              RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
C              RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
C              *
C              + FMSOUT(J,2,IUNT)/TOTIN
C          ELSE
C              RMTBL(J,1) = -1.000
C              RMTBL(J,2) = -1.000
C              RMTBL(J,3) = -1.000
C          END IF
C
C      5710 CONTINUE

```

```

C
C      WRITE(IFOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
C      *
C      *      ( FMSOUT(J,1,IUNT),
C      *      FMSOUT(J,2,IUNT),
C      *      FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
C      *      J = 14 , 16 )
C
C      WRITE(IFOUT,9430) ( RMTBL (J,1),
C      *
C      *      RMTBL (J,2),
C      *      RMTBL (J,3),
C      *      J = 14 , 16 )
C
C      5720 CONTINUE
C
C      DO 5730 J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
C              RMTBL(J,1) = FMTOUT(J,1)/TOTIN
C              RMTBL(J,2) = FMTOUT(J,2)/TOTIN
C              RMTBL(J,3) = FMTOUT(J,1)/TOTIN
C              *
C              + FMTOUT(J,2)/TOTIN
C          ELSE
C              RMTBL(J,1) = -1.000
C              RMTBL(J,2) = -1.000
C              RMTBL(J,3) = -1.000
C          END IF
C
C      5730 CONTINUE
C
C      WRITE(IFOUT,9440) ( FMTOUT(J,1),
C      *
C      *      FMTOUT(J,2),
C      *      FMTOUT(J,1) + FMTOUT(J,2),
C      *      J = 14 , 16 )
C
C      WRITE(IFOUT,9430) ( RMTBL (J,1),
C      *
C      *      RMTBL (J,2),
C      *      RMTBL (J,3),
C      *      J = 14 , 16 )
C
C      WRITE(IFOUT,9345)
C
C      WRITE(IFOUT,9410) ( FMTIN (J,1),
C      *
C      *      FMTIN (J,2),
C      *      FMTIN (J,1) + FMTIN (J,2),
C      *      J = 17 , 19 )
C
C      DO IUNT = 1 , NUNIT
C
C      DO J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
C              RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
C              RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
C              RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
C              *
C              + FMSOUT(J,2,IUNT)/TOTIN
C          ELSE
C              RMTBL(J,1) = -1.000
C              RMTBL(J,2) = -1.000
C              RMTBL(J,3) = -1.000
C          END IF
C
C      ENDDO
C
C      WRITE(IFOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
C      *
C      *      ( FMSOUT(J,1,IUNT),
C      *      FMSOUT(J,2,IUNT),
C      *      FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT);
C      *      J = 17 , 19 )
C
C      WRITE(IFOUT,9430) ( RMTBL (J,1),
C      *
C      *      RMTBL (J,2),
C      *      RMTBL (J,3),
C      *      J = 17 , 19 )
C
C      ENDDO
C
C      DO J = 1 , MXMCMP
C
C          TOTIN = FMTIN(J,1) + FMTIN(J,2)
C          TOTFIN = FINA      + FINO + FFEPS
C
C          IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN

```

```

RMTBL(J,1) = FMTOUT(J,1)/TOTIN
RMTBL(J,2) = FMTOUT(J,2)/TOTIN
RMTBL(J,3) = FMTOUT(J,1)/TOTIN
*             + FMTOUT(J,2)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
ENDDO
C
WRITE(IFOUT,9440) ( FMTOUT(J,1),
*                 FMTOUT(J,2),
*                 FMTOUT(J,1) + FMTOUT(J,2),
*                 J = 17 , 19 )
C
WRITE(IFOUT,9430) ( RMTBL (J,1),
*                 RMTBL (J,2),
*                 RMTBL (J,3),
*                 J = 17 , 19 )
C
WRITE(IFOUT,9350)
C
WRITE(IFOUT,9410) ( FMTIN (J,1),
*                 FMTIN (J,2),
*                 FMTIN (J,1) + FMTIN (J,2),
*                 J = 20 , 22 )
C
DO 5820 IUNT = 1 , NUNIT
C
DO 5810 J = 1 , MXMCOMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMSOUT(J,1,IUNT)/TOTIN
RMTBL(J,2) = FMSOUT(J,2,IUNT)/TOTIN
RMTBL(J,3) = FMSOUT(J,1,IUNT)/TOTIN
*             + FMSOUT(J,2,IUNT)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
5810 CONTINUE
C
WRITE(IFOUT,9420) IUNT, ID(IUNT), TYPE(IUNT),
*                 ( FMSOUT(J,1,IUNT),
*                 FMSOUT(J,2,IUNT),
*                 FMSOUT(J,1,IUNT) + FMSOUT(J,2,IUNT),
*                 J = 20 , 22 )
C
WRITE(IFOUT,9430) ( RMTBL (J,1),
*                 RMTBL (J,2),
*                 RMTBL (J,3),
*                 J = 20 , 22 )
C
5820 CONTINUE
C
DO 5830 J = 1 , MXMCOMP
C
TOTIN = FMTIN(J,1) + FMTIN(J,2)
TOTFIN = FINA + FINO + FFEPS
C
IF ( TOTIN .GE. FMSEPS*TOTFIN ) THEN
RMTBL(J,1) = FMTOUT(J,1)/TOTIN
RMTBL(J,2) = FMTOUT(J,2)/TOTIN
RMTBL(J,3) = FMTOUT(J,1)/TOTIN
*             + FMTOUT(J,2)/TOTIN
ELSE
RMTBL(J,1) = -1.000
RMTBL(J,2) = -1.000
RMTBL(J,3) = -1.000
END IF
C
5830 CONTINUE
C
WRITE(IFOUT,9440) ( FMTOUT(J,1),
*                 FMTOUT(J,2),
*                 FMTOUT(J,1) + FMTOUT(J,2),
*                 J = 20 , 22 )
C
WRITE(IFOUT,9430) ( RMTBL (J,1),
*                 RMTBL (J,2),
*                 RMTBL (J,3),

```

```

*             J = 20 , 22 )
C
C
C
RETURN
C
9000 FORMAT(1H /1H /1H , A,5X, 'TIME:', 1P, E12.5, ' (HR)')
9010 FORMAT(1H /1H /1H , 'NITRIC ACID BALANCE IN EACH UNIT.'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' HNO3 ' , ' HNO3 (PRE.) ' ,
* ' HNO3 (IN) ' , ' HNO3 (OUT) ' , ' FLOW (PRE.) ' ,
* ' H_PRE. (CAL.) ' , ' H_PRE. (ERR.) ' , 2X/
* 1H ,
* (MOL/L) , (MOL/L) ,
* (MOL/HR) , (MOL/HR) , (L/HR) ,
* (MOL/L) , (-)')
9020 FORMAT(1H , 14, 1X, 15, 1X, A2, 2X, 2(2X, 1P2E12.3) , 1P3(3X, E12.3))
9030 FORMAT(1H , 14, 1X, 15, 1X, A2, 2X, 2(8X, ' , 3X), 4X, 1P2E12.3, 2X)
C
9110 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' U(AQ.) ' , ' U(ORG.) ' , ' U(TOT.) ' , 1X,
* ' PU(AQ.) ' , ' PU(ORG.) ' , ' PU(TOT.) ' , 1X,
* ' NP(AQ.) ' , ' NP(ORG.) ' , ' NP(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9115 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' ZR(AQ.) ' , ' ZR(ORG.) ' , ' ZR(TOT.) ' , 1X,
* ' TC(AQ.) ' , ' TC(ORG.) ' , ' TC(TOT.) ' , 1X,
* ' SR(AQ.) ' , ' SR(ORG.) ' , ' SR(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9120 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' RU(AQ.) ' , ' RU(ORG.) ' , ' RU(TOT.) ' , 1X,
* ' CS(AQ.) ' , ' CS(ORG.) ' , ' CS(TOT.) ' , 1X,
* ' CE(AQ.) ' , ' CE(ORG.) ' , ' CE(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9130 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' GD(AQ.) ' , ' GD(ORG.) ' , ' GD(TOT.) ' , 1X,
* ' AM(AQ.) ' , ' AM(ORG.) ' , ' AM(TOT.) ' , 1X,
* ' CM(AQ.) ' , ' CM(ORG.) ' , ' CM(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9140 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' MIC1(AQ.) ' , ' MIC1(ORG.) ' , ' MIC1(TOT.) ' , 1X,
* ' MIC2(AQ.) ' , ' MIC2(ORG.) ' , ' MIC2(TOT.) ' , 1X,
* ' MIC3(AQ.) ' , ' MIC3(ORG.) ' , ' MIC3(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9145 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' MIC4(AQ.) ' , ' MIC4(ORG.) ' , ' MIC4(TOT.) ' , 1X,
* ' MIC5(AQ.) ' , ' MIC5(ORG.) ' , ' MIC5(TOT.) ' , 1X,
* ' MIC6(AQ.) ' , ' MIC6(ORG.) ' , ' MIC6(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9150 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN EACH UNIT'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' MIC7(AQ.) ' , ' MIC7(ORG.) ' , ' MIC7(TOT.) ' , 1X,
* ' MIC8(AQ.) ' , ' MIC8(ORG.) ' , ' MIC8(TOT.) ' , 1X,
* ' MIC9(AQ.) ' , ' MIC9(ORG.) ' , ' MIC9(TOT.) ' /
* 1H ,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) , 1X,
* (G/HR) , (G/HR) , (G/HR) )
9210 FORMAT(1H /1H , 14, 1X, 15, 1X, A2, 2X, ' IN ' , 3(1P3E12.3, 1X))
9220 FORMAT( 1H , 9X, ' , ' , ' OUT ' , 3(1P3E12.3, 1X))
9230 FORMAT( 1H , 9X, ' , ' , ' RAT10 ' , 3( 3F12.4, 1X))
C
9310 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM'/
* 1H , ' NO. ' , ' ID. TYPE' ,
* ' U(AQ.) ' , ' U(ORG.) ' , ' U(TOT.) ' , 1X,

```

```

*      PU(AQ.)      PU(ORG.)      PU(TOT.)      ,1X,
*      NP(AQ.)      NP(ORG.)      NP(TOT.)      ,./
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9315 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      ZR(AQ.)      ZR(ORG.)      ZR(TOT.)      ,1X,
*      TC(AQ.)      TC(ORG.)      TC(TOT.)      ,1X,
*      SR(AQ.)      SR(ORG.)      SR(TOT.)      ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9320 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      RU(AQ.)      RU(ORG.)      RU(TOT.)      ,1X,
*      CS(AQ.)      CS(ORG.)      CS(TOT.)      ,1X,
*      CE(AQ.)      CE(ORG.)      CE(TOT.)      ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9330 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      GD(AQ.)      GD(ORG.)      GD(TOT.)      ,1X,
*      AM(AQ.)      AM(ORG.)      AM(TOT.)      ,1X,
*      CM(AQ.)      CM(ORG.)      CM(TOT.)      ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9340 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      MIC1(AQ.)    MIC1(ORG.)    MIC1(TOT.) ,1X,
*      MIC2(AQ.)    MIC2(ORG.)    MIC2(TOT.) ,1X,
*      MIC3(AQ.)    MIC3(ORG.)    MIC3(TOT.) ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9345 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      MIC4(AQ.)    MIC4(ORG.)    MIC4(TOT.) ,1X,
*      MIC5(AQ.)    MIC5(ORG.)    MIC5(TOT.) ,1X,
*      MIC6(AQ.)    MIC6(ORG.)    MIC6(TOT.) ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9350 FORMAT(1H /1H /1H , 'MATERIAL BALANCE IN SYSTEM' /
*      1H , NO. , ID. TYPE' ,
*      MIC7(AQ.)    MIC7(ORG.)    MIC7(TOT.) ,1X,
*      MIC8(AQ.)    MIC8(ORG.)    MIC8(TOT.) ,1X,
*      MIC9(AQ.)    MIC9(ORG.)    MIC9(TOT.) ) /
*      1H
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      ,1X,
*      (G/HR)      (G/HR)      (G/HR)      )
9410 FORMAT(1H /1H , 'TOTAL MASS FLOW ' , 'IN ' ,3(1P3E12.3,1X) /)
9420 FORMAT(1H /1H ,14,1X,15,1X,A2,4X , 'OUT ' ,3(1P3E12.3,1X) /)
9430 FORMAT( 1H ,11X, , , , , 'RATIO' ,3( 3F12.4,1X) /)
9440 FORMAT(1H /1H , 'TOTAL MASS FLOW ' , 'OUT ' ,3(1P3E12.3,1X) /)
9450 FORMAT( 1H ,11X, , , , , 'RATIO' ,3( 3F12.4,1X) /)
END

```

```

SUBROUTINE OUTADD(IFOUT , IPRDIS,
*      TITLE , TIME , ENDTIM,DT
*      CNAME , FMASS , DECAYH,CVUNIT,
*      CUNIT , CHRG , CNTBP , CPHASE,
*      MXCOMP , MXUNIT , MXSTG , MXSTG2 , NCHEM , MXBLK ,
*      MEXTR , MUNIT , ISF1N , FRC , CO , TO
*      ID , TYPE , FOUT , MSTG , MEXTRA,
*      FL , TSTG , WLT , WST , HOS , TF
*      IEFF , EFF , DIS
*      XN )
C
C
C PRINT TRANSIENT DATA
C
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER*72 TITLE
C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAYH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRG (MXCOMP)
* CNTBP (MXCOMP)
C
C CHARACTER*8 CPHASE (MXCOMP)
C
C DIMENSION ID ( MXUNIT)
C CHARACTER*2 TYPE ( MXUNIT)
C DIMENSION FOUT ( 2 ,MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION MEXTRA ( MXUNIT)
C
C DIMENSION FL (2 ,MXSTG ,MEXTR),
* TSTG ( MXSTG ,MEXTR),
* WLT ( MXSTG ,MEXTR),
* WST ( MXSTG ,MEXTR),
* HOS ( MXSTG ,MEXTR),
* TF ( MXSTG ,MEXTR)
C
C DIMENSION IEFF (MXCOMP ,MXEXTR)
C
C DIMENSION EFF (MXCOMP ,MXSTG ,MEXTR),
* DIS (MXCOMP ,MXSTG ,MEXTR)
C
C DIMENSION IPOS (2 ,MXSTG2 ,MXUNIT)
C
C DIMENSION XN (MXCOMP ,2 ,MXBLK )
C
COMMON /IXCOMP/NCHEM,
* IOHNO3 , IOU6 , IOPU4 , IOPU3 , IOU4 ,
* IOHNO2 , IOHYD , IOHAN , IOPU5 , IOPU6 ,
* IOHP4 , IOHP5 , IOHP6 , IOZR , IO TC4 ,
* IO TC5 , IO TC6 , IO TC7 , IO TC6 , IO TC4 ,
* IO TCZR , IO SR , IO RUD1 , IO RJTR , IO RUI1 ,
* IO RUM0 , IO CS , IO CE , IO GD , IO AM ,
* IO CM
C
C MIC1 = 32
C MIC2 = 33
C MIC3 = 34
C MIC4 = 35
C MIC5 = 36
C MIC6 = 37
C MIC7 = 38
C MIC8 = 39
C MIC9 = 40
C
C
C DO 100 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA(IUNT)
C CO = CO
C TIO = TO
C
C CALL DENST( MXSTG , NSTG

```

```

*          ISFIN ,FRC ,
*          CCO ,TTO ,
*          TSTG(1, IEXTRA) ,WLT(1, IEXTRA) ,
*          WST (1, IEXTRA) ,HOS(1, IEXTRA) )
100 CONTINUE
C
C
C
DO 1000 IUNT = 1 , NUNIT
C
NSTG = MSTG (IUNT)
IEXTRA = MEXTRA(IUNT)
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
... MIXER-SETTLER CONCENTRATIONS ...
C
WRITE(IFOUT,5010) TITLE, TIME
WRITE(IFOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS AT MIXERS (AQUEOUS)'
WRITE(IFOUT,5200) ( CNAME(J), J=32, MXCOMP)
WRITE(IFOUT,5210) ( CUNIT(J), J=32, MXCOMP)
DO 120 I = 1 , NSTG
IND = IPOS(1, 2*I-1, IUNT)
IND = IPOS(2, 2*I-1, IUNT)
DENFCA = DEN SMA( MXCOMP ,XN(1,1, IND) )
XMIC1 = XN(MIC1, 1, IND) / DENFCA * CVUNIT(MIC1)
XMIC2 = XN(MIC2, 1, IND) / DENFCA * CVUNIT(MIC2)
XMIC3 = XN(MIC3, 1, IND) / DENFCA * CVUNIT(MIC3)
XMIC4 = XN(MIC4, 1, IND) / DENFCA * CVUNIT(MIC4)
XMIC5 = XN(MIC5, 1, IND) / DENFCA * CVUNIT(MIC5)
XMIC6 = XN(MIC6, 1, IND) / DENFCA * CVUNIT(MIC6)
XMIC7 = XN(MIC7, 1, IND) / DENFCA * CVUNIT(MIC7)
XMIC8 = XN(MIC8, 1, IND) / DENFCA * CVUNIT(MIC8)
XMIC9 = XN(MIC9, 1, IND) / DENFCA * CVUNIT(MIC9)
XTF = TF(1, IEXTRA)
C
XNO3 = 0.000
DO 110 J = 1 , NCHEMO
XNO3 = XNO3 + CHR G(J)*XN(J, 1, IND)
110 CONTINUE
C
FA = FL(1, 1, IEXTRA) * DENFCA
WRITE(IFOUT,5220) I ,
*          XMIC1,
*          XMIC2,
*          XMIC3,
*          XMIC4,
*          XMIC5,
*          XMIC6,
*          XMIC7,
*          XMIC8,
*          XMIC9
120 CONTINUE
C
WRITE(IFOUT,5010) TITLE, TIME
WRITE(IFOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS AT MIXERS (ORGANIC)'
WRITE(IFOUT,5200) ( CNAME(J), J=32, MXCOMP)
WRITE(IFOUT,5210) ( CUNIT(J), J=32, MXCOMP)
DO 140 I = 1 , NSTG
IND = IPOS(1, 2*I-1, IUNT)
IND = IPOS(2, 2*I-1, IUNT)
DENFCO = DEN SMO( MXCOMP ,WST(1, IEXTRA) ,XN(1,2, IND) )
YMIC1 = XN(MIC1, 2, IND) / DENFCO * CVUNIT(MIC1)
YMIC2 = XN(MIC2, 2, IND) / DENFCO * CVUNIT(MIC2)
YMIC3 = XN(MIC3, 2, IND) / DENFCO * CVUNIT(MIC3)
YMIC4 = XN(MIC4, 2, IND) / DENFCO * CVUNIT(MIC4)
YMIC5 = XN(MIC5, 2, IND) / DENFCO * CVUNIT(MIC5)
YMIC6 = XN(MIC6, 2, IND) / DENFCO * CVUNIT(MIC6)
YMIC7 = XN(MIC7, 2, IND) / DENFCO * CVUNIT(MIC7)
YMIC8 = XN(MIC8, 2, IND) / DENFCO * CVUNIT(MIC8)
YMIC9 = XN(MIC9, 2, IND) / DENFCO * CVUNIT(MIC9)
YTF = TF(1, IEXTRA)
C
YNO3 = 0.000
DO 130 J = 1 , NCHEMO
YNO3 = YNO3 + CHR G(J)*XN(J, 1, IND)
130 CONTINUE
C
FO = FL(2, 1, IEXTRA) * DENFCO
WRITE(IFOUT,5220) I ,
*          YMIC1,
*          YMIC2,
*          YMIC3,
*          YMIC4,
*          YMIC5,
*          YMIC6,
*          YMIC7,

```

```

*          YMIC8,
*          YMIC9
140 CONTINUE
C
WRITE(IFOUT,5010) TITLE, TIME
WRITE(IFOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS AT SETTLERS (AQUEOUS)'
WRITE(IFOUT,5200) ( CNAME(J), J=32, MXCOMP)
WRITE(IFOUT,5210) ( CUNIT(J), J=32, MXCOMP)
DO 160 I = 1 , NSTG
INDSA = IPOS(1, 2*I, IUNT)
INDSO = IPOS(2, 2*I, IUNT)
DENFCA = DEN SMA( MXCOMP ,XN(1,1, INDSA) )
XMIC1 = XN(MIC1, 1, INDSA) / DENFCA * CVUNIT(MIC1)
XMIC2 = XN(MIC2, 1, INDSA) / DENFCA * CVUNIT(MIC2)
XMIC3 = XN(MIC3, 1, INDSA) / DENFCA * CVUNIT(MIC3)
XMIC4 = XN(MIC4, 1, INDSA) / DENFCA * CVUNIT(MIC4)
XMIC5 = XN(MIC5, 1, INDSA) / DENFCA * CVUNIT(MIC5)
XMIC6 = XN(MIC6, 1, INDSA) / DENFCA * CVUNIT(MIC6)
XMIC7 = XN(MIC7, 1, INDSA) / DENFCA * CVUNIT(MIC7)
XMIC8 = XN(MIC8, 1, INDSA) / DENFCA * CVUNIT(MIC8)
XMIC9 = XN(MIC9, 1, INDSA) / DENFCA * CVUNIT(MIC9)
XTF = TF(1, IEXTRA)
C
XNO3 = 0.000
DO 150 J = 1 , NCHEMO
XNO3 = XNO3 + CHR G(J)*XN(J, 1, INDSA)
150 CONTINUE
C
FA = FL(1, 1, IEXTRA) * DENFCA
WRITE(IFOUT,5220) I ,
*          XMIC1,
*          XMIC2,
*          XMIC3,
*          XMIC4,
*          XMIC5,
*          XMIC6,
*          XMIC7,
*          XMIC8,
*          XMIC9
160 CONTINUE
C
WRITE(IFOUT,5010) TITLE, TIME
WRITE(IFOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS AT SETTLERS (ORGANIC)'
WRITE(IFOUT,5200) ( CNAME(J), J=32, MXCOMP)
WRITE(IFOUT,5210) ( CUNIT(J), J=32, MXCOMP)
DO 180 I = 1 , NSTG
INDSA = IPOS(1, 2*I, IUNT)
INDSO = IPOS(2, 2*I, IUNT)
DENFCO = DEN SMO( MXCOMP ,WST(1, IEXTRA) ,XN(1,2, INDSO) )
YMIC1 = XN(MIC1, 2, INDSO) / DENFCO * CVUNIT(MIC1)
YMIC2 = XN(MIC2, 2, INDSO) / DENFCO * CVUNIT(MIC2)
YMIC3 = XN(MIC3, 2, INDSO) / DENFCO * CVUNIT(MIC3)
YMIC4 = XN(MIC4, 2, INDSO) / DENFCO * CVUNIT(MIC4)
YMIC5 = XN(MIC5, 2, INDSO) / DENFCO * CVUNIT(MIC5)
YMIC6 = XN(MIC6, 2, INDSO) / DENFCO * CVUNIT(MIC6)
YMIC7 = XN(MIC7, 2, INDSO) / DENFCO * CVUNIT(MIC7)
YMIC8 = XN(MIC8, 2, INDSO) / DENFCO * CVUNIT(MIC8)
YMIC9 = XN(MIC9, 2, INDSO) / DENFCO * CVUNIT(MIC9)
YTF = TF(1, IEXTRA)
C
YNO3 = 0.000
DO 170 J = 1 , NCHEMO
YNO3 = YNO3 + CHR G(J)*XN(J, 1, INDSA)
170 CONTINUE
C
FO = FL(2, 1, IEXTRA) * DENFCO
WRITE(IFOUT,5220) I ,
*          YMIC1,
*          YMIC2,
*          YMIC3,
*          YMIC4,
*          YMIC5,
*          YMIC6,
*          YMIC7,
*          YMIC8,
*          YMIC9
180 CONTINUE
C
IF ( IPPDIS .EQ. 1 ) THEN
C
WRITE(IFOUT,5010) TITLE, TIME
WRITE(IFOUT,5100) ID(IUNT), TYPE(IUNT),
*          'DISTRIBUTION COEFFICIENTS,'
WRITE(IFOUT,5300) CNAME(MIC1),
*          CNAME(MIC2),
*          CNAME(MIC3),
*          CNAME(MIC4),

```

```

*          CNAME(MIC5),
*          CNAME(MIC6),
*          CNAME(MIC7),
*          CNAME(MIC8),
*          CNAME(MIC9)
C
DO 191 I = 1, NSTG
C
    WRITE(1FOUT,5320) I
*          DIS(MIC1,I, IEXTRA),
*          DIS(MIC2,I, IEXTRA),
*          DIS(MIC3,I, IEXTRA),
*          DIS(MIC4,I, IEXTRA),
*          DIS(MIC5,I, IEXTRA),
*          DIS(MIC6,I, IEXTRA),
*          DIS(MIC7,I, IEXTRA),
*          DIS(MIC8,I, IEXTRA),
*          DIS(MIC9,I, IEXTRA)
191    CONTINUE
C
DO 192 J = 1, MXCOMP
    IF (IEFF(J, IEXTRA) .EQ. 1) THEN
        CPHASE(J) = 'AQUEOUS'
    ELSE
        CPHASE(J) = 'ORGANIC'
    END IF
192    CONTINUE
C
    WRITE(1FOUT,5010) TITLE, TIME
    WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
*          'EFFICIENCY OF RECOVERY RATE.'
*          WRITE(1FOUT,5300) CNAME(MIC1),
*          CNAME(MIC2),
*          CNAME(MIC3),
*          CNAME(MIC4),
*          CNAME(MIC5),
*          CNAME(MIC6),
*          CNAME(MIC7),
*          CNAME(MIC8),
*          CNAME(MIC9)
*          WRITE(1FOUT,5310) CPHASE(MIC1),
*          CPHASE(MIC2),
*          CPHASE(MIC3),
*          CPHASE(MIC4),
*          CPHASE(MIC5),
*          CPHASE(MIC6),
*          CPHASE(MIC7),
*          CPHASE(MIC8),
*          CPHASE(MIC9)
C
DO 193 I = 1, NSTG
C
    WRITE(1FOUT,5320) I
*          EFF(MIC1,I, IEXTRA),
*          EFF(MIC2,I, IEXTRA),
*          EFF(MIC3,I, IEXTRA),
*          EFF(MIC4,I, IEXTRA),
*          EFF(MIC5,I, IEXTRA),
*          EFF(MIC6,I, IEXTRA),
*          EFF(MIC7,I, IEXTRA),
*          EFF(MIC8,I, IEXTRA),
*          EFF(MIC9,I, IEXTRA)
193    CONTINUE
C
    END IF
C
C
C
ELSE
... TANK CONCENTRATIONS ...
WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS IN TANK(AQUEOUS)'
WRITE(1FOUT,5200) ( CNAME(J), J=32, MXCOMP)
WRITE(1FOUT,5210) ( CUNIT(J), J=32, MXCOMP)
DO 220 I = 1, NSTG
    IND = IPOS(1, I, IUNT)
    IND = IPOS(2, I, IUNT)
    DENFCA = 1.000
    XMI C1 = XN(MI C1, 1, IND) / DENFCA * CVUNIT(MI C1)
    XMI C2 = XN(MI C2, 1, IND) / DENFCA * CVUNIT(MI C2)
    XMI C3 = XN(MI C3, 1, IND) / DENFCA * CVUNIT(MI C3)
    XMI C4 = XN(MI C4, 1, IND) / DENFCA * CVUNIT(MI C4)
    XMI C5 = XN(MI C5, 1, IND) / DENFCA * CVUNIT(MI C5)
    XMI C6 = XN(MI C6, 1, IND) / DENFCA * CVUNIT(MI C6)
    XMI C7 = XN(MI C7, 1, IND) / DENFCA * CVUNIT(MI C7)
    XMI C8 = XN(MI C8, 1, IND) / DENFCA * CVUNIT(MI C8)
    XMI C9 = XN(MI C9, 1, IND) / DENFCA * CVUNIT(MI C9)
C
    FA = FOUT(1, IUNT) * DENFCA
    WRITE(1FOUT,5220) I
*          XMI C1,
*          XMI C2,
*          XMI C3,
*          XMI C4,
*          XMI C5,
*          XMI C6,
*          XMI C7,
*          XMI C8,
*          XMI C9
220    CONTINUE
C
    WRITE(1FOUT,5010) TITLE, TIME
    WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
*          'CONCENTRATIONS IN TANK(ORGANIC)'
    WRITE(1FOUT,5200) ( CNAME(J), J=32, MXCOMP)
    WRITE(1FOUT,5210) ( CUNIT(J), J=32, MXCOMP)
    DO 240 I = 1, NSTG
        IND = IPOS(1, I, IUNT)
        IND = IPOS(2, I, IUNT)
        DENFCO = 1.000
        YMI C1 = XN(MI C1, 2, IND) / DENFCO * CVUNIT(MI C1)
        YMI C2 = XN(MI C2, 2, IND) / DENFCO * CVUNIT(MI C2)
        YMI C3 = XN(MI C3, 2, IND) / DENFCO * CVUNIT(MI C3)
        YMI C4 = XN(MI C4, 2, IND) / DENFCO * CVUNIT(MI C4)
        YMI C5 = XN(MI C5, 2, IND) / DENFCO * CVUNIT(MI C5)
        YMI C6 = XN(MI C6, 2, IND) / DENFCO * CVUNIT(MI C6)
        YMI C7 = XN(MI C7, 2, IND) / DENFCO * CVUNIT(MI C7)
        YMI C8 = XN(MI C8, 2, IND) / DENFCO * CVUNIT(MI C8)
        YMI C9 = XN(MI C9, 2, IND) / DENFCO * CVUNIT(MI C9)
C
        YNO3 = 0.000
        DO 230 J = 1, NCHEMO
            YNO3 = YNO3 + CHR(J)*XN(J, 1, IND)
        CONTINUE
230    CONTINUE
C
        FO = FOUT(2, IUNT) * DENFCO
        WRITE(1FOUT,5220) I
*          YMI C1,
*          YMI C2,
*          YMI C3,
*          YMI C4,
*          YMI C5,
*          YMI C6,
*          YMI C7,
*          YMI C8,
*          YMI C9
240    CONTINUE
C
    END IF
C
C
C
1000 CONTINUE
C
C
C
    RETURN
C
C
C
5010 FORMAT(1H /1H /1H ,A,5X,' TIME:',1P,E12.5,' (HR)')
5100 FORMAT(1H /1H , ' UNIT ID,'15.1X,A2.5X,A)
5200 FORMAT(1H , 'STAGE ',A8,9(5X,A8))
5210 FORMAT(1H , ' ,',A8,9(5X,A8))
5220 FORMAT(1H ,12,1P,10E13.5)
C
5300 FORMAT(1H , 'STAGE ',A8,1X,9(4X,A8,1X))
5310 FORMAT(1H , ' ,',A8,1X,9(4X,A8,1X))
5320 FORMAT(1H ,12,1P,15E13.5)
C
    END

```

```

SUBROUTINE OUTMIC(IFOUT,IPRDIS,
*      TITLE,TIME,ENDTIM,DT,
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,CPHASE,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,WLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
C
C
C
C PRINT TRANSIENT DATA
C
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C
C CHARACTER*72 TITLE
C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAYH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRG (MXCOMP),
*      CNTBP (MXCOMP)
C
C CHARACTER*8 CPHASE (MXCOMP)
C
C DIMENSION ID ( MXUNIT)
C CHARACTER*2 TYPE ( MXUNIT)
C DIMENSION FOUT ( 2 ,MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION MEXTRA ( MXUNIT)
C
C DIMENSION FL (2 ,MXSTG,MXEXTR),
*      TSTG ( MXSTG,MXEXTR),
*      WLT ( MXSTG,MXEXTR),
*      WST ( MXSTG,MXEXTR),
*      HOS ( MXSTG,MXEXTR),
*      TF ( MXSTG,MXEXTR)
C
C DIMENSION IEFF (MXCOMP ,MXEXTR)
C
C DIMENSION EFF (MXCOMP,MXSTG,MXEXTR),
*      DIS (MXCOMP,MXSTG,MXEXTR)
C
C DIMENSION IPOS (2,MXSTG2,MXUNIT)
C
C DIMENSION XN (MXCOMP,2,MXBLK )
C
COMMON /IXCOMP/NCHEM,
*      IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*      IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*      IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*      IOTC5 ,IOTC6 ,IOTC7 ,IOTC6 ,IOTCP4 ,
*      IOTCZR ,IOSR ,IORUDI ,IORUTR ,IORUNI ,
*      IORUMO ,IOCS ,IOCE ,IOGD ,IOAM ,
*      IOCM
C
C
C DO 100 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA(IUNT)
C CCO = CO
C TTD = TO
C
C CALL DENST( MXSTG ,NSTG ,
*      ISFIN ,FRC ,
*      CCO ,TTD ,
*      TSTG(1,IEXTRA) ,WLT(1,IEXTRA) ,
*      WST (1,IEXTRA) ,HOS(1,IEXTRA) )
100 CONTINUE
C
C
C DO 1000 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA(IUNT)

```

```

C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
... MIXER-SETTLER CONCENTRATIONS ...
C
WRITE(IFOUT,5010) TITLE,TIME
WRITE(IFOUT,5100) ID(IUNT),TYPE(IUNT),
*      'CONCENTRATIONS AT MIXERS(AQUEOUS)'
WRITE(IFOUT,5200) CNAME(10SR),
*      CNAME(10RUDI),
*      CNAME(10RUTR),
*      CNAME(10RUNI),
*      CNAME(10RUMO),
*      CNAME(10CS),
*      CNAME(10CE),
*      CNAME(10GD),
*      CNAME(10AM),
*      CNAME(10CM)
WRITE(IFOUT,5210) CUNIT(10SR),
*      CUNIT(10RUDI),
*      CUNIT(10RUTR),
*      CUNIT(10RUNI),
*      CUNIT(10RUMO),
*      CUNIT(10CS),
*      CUNIT(10CE),
*      CUNIT(10GD),
*      CUNIT(10AM),
*      CUNIT(10CM)
DO 120 I = 1 , NSTG
IND = IPOS(1,2*I-1,IUNT)
IND = IPOS(2,2*I-1,IUNT)
DENFCA = DEN SMA( MXCOMP ,XN(1,1,IND) )
XSR = XN(10SR ,1,IND) / DENFCA * CVUNIT(10SR )
XRUDI = XN(10RUDI ,1,IND) / DENFCA * CVUNIT(10RUDI)
XRUTR = XN(10RUTR ,1,IND) / DENFCA * CVUNIT(10RUTR)
XRUNI = XN(10RUNI ,1,IND) / DENFCA * CVUNIT(10RUNI)
XRUMO = XN(10RUMO ,1,IND) / DENFCA * CVUNIT(10RUMO)
XCS = XN(10CS ,1,IND) / DENFCA * CVUNIT(10CS )
XCE = XN(10CE ,1,IND) / DENFCA * CVUNIT(10CE )
XGD = XN(10GD ,1,IND) / DENFCA * CVUNIT(10GD )
XAM = XN(10AM ,1,IND) / DENFCA * CVUNIT(10AM )
XCM = XN(10CM ,1,IND) / DENFCA * CVUNIT(10CM )
XTF = TF(1,IEXTRA)
C
XNO3 = 0.000
DO 110 J = 1 , NCHEM
XNO3 = XNO3 + CHRG(J)*XN(J,1,IND)
110 CONTINUE
C
FA = FL(1,1,IEXTRA) * DENFCA
WRITE(IFOUT,5220) I ,
*      XSR ,
*      XRUDI ,
*      XRUTR ,
*      XRUNI ,
*      XRUMO ,
*      XCS ,
*      XCE ,
*      XGD ,
*      XAM ,
*      XCM
120 CONTINUE
C
WRITE(IFOUT,5010) TITLE,TIME
WRITE(IFOUT,5100) ID(IUNT),TYPE(IUNT),
*      'CONCENTRATIONS AT MIXERS(ORGANIC)'
WRITE(IFOUT,5200) CNAME(10SR),
*      CNAME(10RUDI),
*      CNAME(10RUTR),
*      CNAME(10RUNI),
*      CNAME(10RUMO),
*      CNAME(10CS),
*      CNAME(10CE),
*      CNAME(10GD),
*      CNAME(10AM),
*      CNAME(10CM)
WRITE(IFOUT,5210) CUNIT(10SR),
*      CUNIT(10RUDI),
*      CUNIT(10RUTR),
*      CUNIT(10RUNI),
*      CUNIT(10RUMO),
*      CUNIT(10CS),
*      CUNIT(10CE),
*      CUNIT(10GD),
*      CUNIT(10AM),
*      CUNIT(10CM)
DO 140 I = 1 , NSTG
IND = IPOS(1,2*I-1,IUNT)
IND = IPOS(2,2*I-1,IUNT)

```

```

DENFCO = DENSMO( MXCOMP ,WST(1, IEXTRA) ,XN(1,2, IND) )
YSR = XN(1OSR ,2, IND) / DENFCO * CVUNIT(1OSR )
YRUDI = XN(1ORUDI,2, IND) / DENFCO * CVUNIT(1ORUDI)
YRUTR = XN(1ORUTR,2, IND) / DENFCO * CVUNIT(1ORUTR)
YRUNI = XN(1ORUNI,2, IND) / DENFCO * CVUNIT(1ORUNI)
YRUMO = XN(1ORUMO,2, IND) / DENFCO * CVUNIT(1ORUMO)
YCS = XN(1OCS ,2, IND) / DENFCO * CVUNIT(1OCS )
YCE = XN(1OCE ,2, IND) / DENFCO * CVUNIT(1OCE )
YGD = XN(1OGD ,2, IND) / DENFCO * CVUNIT(1OGD )
YAM = XN(1OAM ,2, IND) / DENFCO * CVUNIT(1OAM )
YCM = XN(1OCM ,2, IND) / DENFCO * CVUNIT(1OCM )
YTF = TF(1, IEXTRA)

C
YNO3 = 0.000
DO 130 J = 1 , NCHEMO
  YNO3 = YNO3 + CHR( J)*XN( J,1, IND)
CONTINUE

130
C
FO = FL(2,1, IEXTRA) * DENFCO
WRITE( IFOUT, 5220) I ,
*
* YSR ,
* YRUDI ,
* YRUTR ,
* YRUNI ,
* YRUMO ,
* YCS ,
* YCE ,
* YGD ,
* YAM ,
* YCM

140 CONTINUE
C
WRITE( IFOUT, 5010) TITLE, TIME
WRITE( IFOUT, 5100) ID( IUNT), TYPE( IUNT),
* 'CONCENTRATIONS AT SETTLERS (AQUEOUS)'
WRITE( IFOUT, 5200) CNAME(1OSR ),
* CNAME(1ORUDI),
* CNAME(1ORUTR),
* CNAME(1ORUNI),
* CNAME(1ORUMO),
* CNAME(1OCS ),
* CNAME(1OCE ),
* CNAME(1OGD ),
* CNAME(1OAM ),
* CNAME(1OCM )
WRITE( IFOUT, 5210) CUNIT(1OSR ),
* CUNIT(1ORUDI),
* CUNIT(1ORUTR),
* CUNIT(1ORUNI),
* CUNIT(1ORUMO),
* CUNIT(1OCS ),
* CUNIT(1OCE ),
* CUNIT(1OGD ),
* CUNIT(1OAM ),
* CUNIT(1OCM )

DO 160 I = 1 , NSTG
  INDSA = IPOS(1, 2*1, IUNT)
  INDSO = IPOS(2, 2*1, IUNT)
  DENFCA = DENSMO( MXCOMP ,XN(1,1, INDSA) )
  XSR = XN(1OSR ,1, INDSA) / DENFCA * CVUNIT(1OSR )
  XRUDI = XN(1ORUDI,1, INDSA) / DENFCA * CVUNIT(1ORUDI)
  XRUTR = XN(1ORUTR,1, INDSA) / DENFCA * CVUNIT(1ORUTR)
  XRUNI = XN(1ORUNI,1, INDSA) / DENFCA * CVUNIT(1ORUNI)
  XRUMO = XN(1ORUMO,1, INDSA) / DENFCA * CVUNIT(1ORUMO)
  XCS = XN(1OCS ,1, INDSA) / DENFCA * CVUNIT(1OCS )
  XCE = XN(1OCE ,1, INDSA) / DENFCA * CVUNIT(1OCE )
  XGD = XN(1OGD ,1, INDSA) / DENFCA * CVUNIT(1OGD )
  XAM = XN(1OAM ,1, INDSA) / DENFCA * CVUNIT(1OAM )
  XCM = XN(1OCM ,1, INDSA) / DENFCA * CVUNIT(1OCM )
  XTF = TF(1, IEXTRA)

C
XNO3 = 0.000
DO 150 J = 1 , NCHEMO
  XNO3 = XNO3 + CHR( J)*XN( J,1, INDSA)
CONTINUE

150
C
FA = FL(1,1, IEXTRA) * DENFCA
WRITE( IFOUT, 5220) I ,
*
* XSR ,
* XRUDI ,
* XRUTR ,
* XRUNI ,
* XRUMO ,
* XCS ,
* XCE ,
* XGD ,
* XAM ,
* XCM

160 CONTINUE

```

```

C
WRITE( IFOUT, 5010) TITLE, TIME
WRITE( IFOUT, 5100) ID( IUNT), TYPE( IUNT),
* 'CONCENTRATIONS AT SETTLERS (ORGANIC)'
WRITE( IFOUT, 5200) CNAME(1OSR ),
* CNAME(1ORUDI),
* CNAME(1ORUTR),
* CNAME(1ORUNI),
* CNAME(1ORUMO),
* CNAME(1OCS ),
* CNAME(1OCE ),
* CNAME(1OGD ),
* CNAME(1OAM ),
* CNAME(1OCM )
WRITE( IFOUT, 5210) CUNIT(1OSR ),
* CUNIT(1ORUDI),
* CUNIT(1ORUTR),
* CUNIT(1ORUNI),
* CUNIT(1ORUMO),
* CUNIT(1OCS ),
* CUNIT(1OCE ),
* CUNIT(1OGD ),
* CUNIT(1OAM ),
* CUNIT(1OCM )

DO 180 I = 1 , NSTG
  INDSA = IPOS(1, 2*1, IUNT)
  INDSO = IPOS(2, 2*1, IUNT)
  DENFCO = DENSMO( MXCOMP ,WST(1, IEXTRA) ,XN(1,2, INDSO) )
  YSR = XN(1OSR ,2, INDSO) / DENFCO * CVUNIT(1OSR )
  YRUDI = XN(1ORUDI,2, INDSO) / DENFCO * CVUNIT(1ORUDI)
  YRUTR = XN(1ORUTR,2, INDSO) / DENFCO * CVUNIT(1ORUTR)
  YRUNI = XN(1ORUNI,2, INDSO) / DENFCO * CVUNIT(1ORUNI)
  YRUMO = XN(1ORUMO,2, INDSO) / DENFCO * CVUNIT(1ORUMO)
  YCS = XN(1OCS ,2, INDSO) / DENFCO * CVUNIT(1OCS )
  YCE = XN(1OCE ,2, INDSO) / DENFCO * CVUNIT(1OCE )
  YGD = XN(1OGD ,2, INDSO) / DENFCO * CVUNIT(1OGD )
  YAM = XN(1OAM ,2, INDSO) / DENFCO * CVUNIT(1OAM )
  YCM = XN(1OCM ,2, INDSO) / DENFCO * CVUNIT(1OCM )
  YTF = TF(1, IEXTRA)

C
YNO3 = 0.000
DO 170 J = 1 , NCHEMO
  YNO3 = YNO3 + CHR( J)*XN( J,1, INDSA)
CONTINUE

170
C
FO = FL(2,1, IEXTRA) * DENFCO
WRITE( IFOUT, 5220) I ,
*
* YSR ,
* YRUDI ,
* YRUTR ,
* YRUNI ,
* YRUMO ,
* YCS ,
* YCE ,
* YGD ,
* YAM ,
* YCM

180 CONTINUE
C
IF ( IPRDIS .EQ. 1 ) THEN
  WRITE( IFOUT, 5010) TITLE, TIME
  WRITE( IFOUT, 5100) ID( IUNT), TYPE( IUNT),
  * 'DISTRIBUTION COEFFICIENTS.'
  WRITE( IFOUT, 5300) CNAME(1OSR ),
  * CNAME(1ORUDI),
  * CNAME(1ORUTR),
  * CNAME(1ORUNI),
  * CNAME(1ORUMO),
  * CNAME(1OCS ),
  * CNAME(1OCE ),
  * CNAME(1OGD ),
  * CNAME(1OAM ),
  * CNAME(1OCM )

C
DO 191 I = 1 , NSTG
  WRITE( IFOUT, 5320) I ,
  * DIS(1OSR ,1, IEXTRA),
  * DIS(1ORUDI,1, IEXTRA),
  * DIS(1ORUTR,1, IEXTRA),
  * DIS(1ORUNI,1, IEXTRA),
  * DIS(1ORUMO,1, IEXTRA),
  * DIS(1OCS ,1, IEXTRA),
  * DIS(1OCE ,1, IEXTRA),
  * DIS(1OGD ,1, IEXTRA),
  * DIS(1OAM ,1, IEXTRA),
  * DIS(1OCM ,1, IEXTRA)

191 CONTINUE

```

```

C
DO 192 J = 1, MXCOMP
  IF ( I EFF(J, IEXTRA) .EQ. 1 ) THEN
    CPHASE(J) = 'AQUEOUS'
  ELSE
    CPHASE(J) = 'ORGANIC'
  END IF
192 CONTINUE
C
WRITE( IFOUT, 5010) TITLE, TIME
WRITE( IFOUT, 5100) ID( IUNT ), TYPE( IUNT ),
*   'EFFICIENCY OF RECOVERY RATE.'
WRITE( IFOUT, 5300) CNAME( IOSR ),
*   CNAME( IORUDI ),
*   CNAME( IORUTR ),
*   CNAME( IORUNI ),
*   CNAME( IORUMO ),
*   CNAME( IOCS ),
*   CNAME( IOCE ),
*   CNAME( IOGD ),
*   CNAME( IOAM ),
*   CNAME( IOCM )
WRITE( IFOUT, 5310) CPHASE( IOSR ),
*   CPHASE( IORUDI ),
*   CPHASE( IORUTR ),
*   CPHASE( IORUNI ),
*   CPHASE( IORUMO ),
*   CPHASE( IOCS ),
*   CPHASE( IOCE ),
*   CPHASE( IOGD ),
*   CPHASE( IOAM ),
*   CPHASE( IOCM )
C
DO 193 I = 1, NSTG
C
  WRITE( IFOUT, 5320) I ,
*   EFF( IOSR , I, IEXTRA ),
*   EFF( IORUDI , I, IEXTRA ),
*   EFF( IORUTR, I, IEXTRA ),
*   EFF( IORUNI , I, IEXTRA ),
*   EFF( IORUMO, I, IEXTRA ),
*   EFF( IOCS  , I, IEXTRA ),
*   EFF( IOCE  , I, IEXTRA ),
*   EFF( IOGD  , I, IEXTRA ),
*   EFF( IOAM  , I, IEXTRA ),
*   EFF( IOCM  , I, IEXTRA )
193 CONTINUE
C
  END IF
C
  ELSE
C
  ... TANK CONCENTRATIONS ...
C
  WRITE( IFOUT, 5010) TITLE, TIME
  WRITE( IFOUT, 5100) ID( IUNT ), TYPE( IUNT ),
*   'CONCENTRATIONS IN TANK(AQUEOUS)'
  WRITE( IFOUT, 5200) CNAME( IOSR ),
*   CNAME( IORUDI ),
*   CNAME( IORUTR ),
*   CNAME( IORUNI ),
*   CNAME( IORUMO ),
*   CNAME( IOCS ),
*   CNAME( IOCE ),
*   CNAME( IOGD ),
*   CNAME( IOAM ),
*   CNAME( IOCM )
  WRITE( IFOUT, 5210) CUNIT( IOSR ),
*   CUNIT( IORUDI ),
*   CUNIT( IORUTR ),
*   CUNIT( IORUNI ),
*   CUNIT( IORUMO ),
*   CUNIT( IOCS ),
*   CUNIT( IOCE ),
*   CUNIT( IOGD ),
*   CUNIT( IOAM ),
*   CUNIT( IOCM )
  DO 220 I = 1, NSTG
    IND = IPOS( I, IUNT )
    IND = IPOS( 2, I, IUNT )
    DENFCA = 1.000
    XSR = XN( IOSR , I, IND ) / DENFCA * CVUNIT( IOSR )
    XRUDI = XN( IORUDI, I, IND ) / DENFCA * CVUNIT( IORUDI )
    XRUTR = XN( IORUTR, I, IND ) / DENFCA * CVUNIT( IORUTR )
    XRUNI = XN( IORUNI, I, IND ) / DENFCA * CVUNIT( IORUNI )
    XRUMO = XN( IORUMO, I, IND ) / DENFCA * CVUNIT( IORUMO )
    XCS = XN( IOCS , I, IND ) / DENFCA * CVUNIT( IOCS )
    XCE = XN( IOCE , I, IND ) / DENFCA * CVUNIT( IOCE )
    XGD = XN( IOGD , I, IND ) / DENFCA * CVUNIT( IOGD )
    XAM = XN( IOAM , I, IND ) / DENFCA * CVUNIT( IOAM )
    XCM = XN( IOCM , I, IND ) / DENFCA * CVUNIT( IOCM )
  C
    XNO3 = 0.000
    DO 210 J = 1, NCHEMO
      XNO3 = XNO3 + CHR( J ) * XN( J, I, IND )
    CONTINUE
  C
    FA = FOUT( I, IUNT ) * DENFCA
    WRITE( IFOUT, 5220) I ,
*   XSR ,
*   XRUDI ,
*   XRUTR ,
*   XRUNI ,
*   XRUMO ,
*   XCS ,
*   XCE ,
*   XGD ,
*   XAM ,
*   XCM
  220 CONTINUE
  C
    WRITE( IFOUT, 5010) TITLE, TIME
    WRITE( IFOUT, 5100) ID( IUNT ), TYPE( IUNT ),
*   'CONCENTRATIONS IN TANK(ORGANIC)'
    WRITE( IFOUT, 5200) CNAME( IOSR ),
*   CNAME( IORUDI ),
*   CNAME( IORUTR ),
*   CNAME( IORUNI ),
*   CNAME( IORUMO ),
*   CNAME( IOCS ),
*   CNAME( IOCE ),
*   CNAME( IOGD ),
*   CNAME( IOAM ),
*   CNAME( IOCM )
    WRITE( IFOUT, 5210) CUNIT( IOSR ),
*   CUNIT( IORUDI ),
*   CUNIT( IORUTR ),
*   CUNIT( IORUNI ),
*   CUNIT( IORUMO ),
*   CUNIT( IOCS ),
*   CUNIT( IOCE ),
*   CUNIT( IOGD ),
*   CUNIT( IOAM ),
*   CUNIT( IOCM )
  DO 240 I = 1, NSTG
    IND = IPOS( I, IUNT )
    IND = IPOS( 2, I, IUNT )
    DENFCO = 1.000
    YSR = XN( IOSR , 2, IND ) / DENFCO * CVUNIT( IOSR )
    YRUDI = XN( IORUDI, 2, IND ) / DENFCO * CVUNIT( IORUDI )
    YRUTR = XN( IORUTR, 2, IND ) / DENFCO * CVUNIT( IORUTR )
    YRUNI = XN( IORUNI, 2, IND ) / DENFCO * CVUNIT( IORUNI )
    YRUMO = XN( IORUMO, 2, IND ) / DENFCO * CVUNIT( IORUMO )
    YCS = XN( IOCS , 2, IND ) / DENFCO * CVUNIT( IOCS )
    YCE = XN( IOCE , 2, IND ) / DENFCO * CVUNIT( IOCE )
    YGD = XN( IOGD , 2, IND ) / DENFCO * CVUNIT( IOGD )
    YAM = XN( IOAM , 2, IND ) / DENFCO * CVUNIT( IOAM )
    YCM = XN( IOCM , 2, IND ) / DENFCO * CVUNIT( IOCM )
  C
    YNO3 = 0.000
    DO 230 J = 1, NCHEMO
      YNO3 = YNO3 + CHR( J ) * XN( J, I, IND )
    CONTINUE
  C
    FO = FOUT( 2, IUNT ) * DENFCO
    WRITE( IFOUT, 5220) I ,
*   YSR ,
*   YRUDI ,
*   YRUTR ,
*   YRUNI ,
*   YRUMO ,
*   YCS ,
*   YCE ,
*   YGD ,
*   YAM ,
*   YCM
  240 CONTINUE
  C
  END IF
  C
  1000 CONTINUE

```



```

RETURN
C
C
5010 FORMAT(1H /1H /1H .A,5X,'TIME:',1P,E12.5,' (HR)')
5100 FORMAT(1H /1H .' UNIT ID.' 15, 1X, A2, 5X, A)
5200 FORMAT(1H .' STAGE ', A8, 9(5X, A8))
5210 FORMAT(1H .' . ', A8, 9(5X, A8))
5220 FORMAT(1H . 12, 1P, 10E13. 5)
C
5300 FORMAT(1H .' STAGE ', A8, 1X, 9(4X, A8, 1X))
5310 FORMAT(1H .' . ', A8, 1X, 9(4X, A8, 1X))
5320 FORMAT(1H . 12, 1P, 15E13. 5)
C
END

```

```

SUBROUTINE OUTPLT (IUPLT ,
* TITLE , TIME ,
* CNAME , FMASS , DECAYH , CVUNIT ,
* CUNIT , CHRG , CNTBP , CPHASE ,
* MXCOMP , MXUNIT , MXSTG , MXSTG2 , NCHEM , MXBLK ,
* MEXEXTR , NUNIT , ISFIN , FRC , CO , TO ,
* ID , TYPE , FOUT , MSTG , MEXTRA ,
* FL , TSTG , WLT , WST , HOS , TF ,
* IEFF , EFF , DIS ,
* IPOS ,
* XN )
C
C
C PUT CONCENTRATIONS IN PLOTFILE.
C
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER* 72 TITLE
C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAYH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRG (MXCOMP) ,
* CNTBP (MXCOMP)
C
C CHARACTER*8 CPHASE (MXCOMP)
C
C DIMENSION ID ( MXUNIT)
C CHARACTER*2 TYPE ( MXUNIT)
C DIMENSION FOUT ( 2 , MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION MEXTRA ( MXUNIT)
C
C DIMENSION FL (2 , MXSTG , MEXEXTR) ,
* TSTG ( MXSTG , MEXEXTR) ,
* WLT ( MXSTG , MEXEXTR) ,
* WST ( MXSTG , MEXEXTR) ,
* HOS ( MXSTG , MEXEXTR) ,
* TF ( MXSTG , MEXEXTR)
C
C DIMENSION IEFF (MXCOMP , MEXEXTR)
C
C DIMENSION EFF (MXCOMP , MXSTG , MEXEXTR) ,
* DIS (MXCOMP , MXSTG , MEXEXTR)
C
C DIMENSION IPOS (2, MXSTG2, MXUNIT)
C
C DIMENSION XN (MXCOMP, 2, MXBLK )
C
COMMON /IXCOMP/NCHEM,
* IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
* IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
* IOHP4 , IOHP5 , IOHP6 , IOZR , IO TC4 ,
* IO TC5 , IO TC6 , IO TC7 , IO TC6 , IO TC4 ,
* IO TCZR , IOSR , IO RUD1 , IO RUTR , IO RUNI ,
* IO RUMO , IOCS , IOCE , IOGD , IOAM ,
* IOCM
C
C
C
C DO 100 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA (IUNT)
C CO = CO
C TTO = TO
C
C CALL DENST( MXSTG , NSTG ,
* ISFIN , FRC ,
* CO , TTO ,
* TSTG (1, IEXTRA) , WLT (1, IEXTRA) ,
* WST (1, IEXTRA) , HOS (1, IEXTRA) )
100 CONTINUE
C
C
C WRITE (IUPLT, 5000) NUNIT , TIME
C
C DO 200 IUNT = 1 , NUNIT
C WRITE (IUPLT, 5100) ID (IUNT) , TYPE (IUNT) , MSTG (IUNT)

```

```

200 CONTINUE
C
C
C
DO 2000 IUNT = 1 , NUNIT
C
    NSTG = MSTG (IUNT)
    IEXTRA = MEXTRA(IUNT)
C
    IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
        ... MIXER-SETTLER CONCENTRATIONS ...
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'MIXER(AQUEOUS)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''
C
        DO 1110 I = 1 , NSTG
            IND = IPOS(1,2*I-1,IUNT)
            IND = IPOS(2,2*I-1,IUNT)
            DENFCA = DEN SMA( MXCOMP ,XN(1,1,IND) )
            FA = FL(1,1,IEXTRA) * DENFCA
            WRITE(IUPLT,5300)
*           I, ( XN(J,1,IND)/DENFCA*CVUNIT(J),J=1,MXCOMP),FA
1110 CONTINUE
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'MIXER(ORGANIC)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''
C
        DO 1120 I = 1 , NSTG
            IND = IPOS(1,2*I-1,IUNT)
            IND = IPOS(2,2*I-1,IUNT)
            DENFCO = DEN SMO( MXCOMP ,WST(1,IEXTRA) ,XN(1,2,IND) )
            FO = FL(2,1,IEXTRA) * DENFCO
            WRITE(IUPLT,5300)
*           I, ( XN(J,2,IND)/DENFCO*CVUNIT(J),J=1,MXCOMP),FO
1120 CONTINUE
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'SETTLER(AQUEOUS)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''
C
        DO 1130 I = 1 , NSTG
            INDSA = IPOS(1,2*I,IUNT)
            INDSO = IPOS(2,2*I,IUNT)
            DENFCA = DEN SMA( MXCOMP ,XN(1,1,INDSA) )
            FA = FL(1,1,IEXTRA) * DENFCA
            WRITE(IUPLT,5300)
*           I, ( XN(J,1,INDSA)/DENFCA*CVUNIT(J),J=1,MXCOMP),FA
1130 CONTINUE
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'SETTLER(ORGANIC)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''
C
        DO 1140 I = 1 , NSTG
            INDSA = IPOS(1,2*I,IUNT)
            INDSO = IPOS(2,2*I,IUNT)
            DENFCO = DEN SMO( MXCOMP ,WST(1,IEXTRA) ,XN(1,2,INDSO) )
            FO = FL(2,1,IEXTRA) * DENFCO
            WRITE(IUPLT,5300)
*           I, ( XN(J,2,INDSO)/DENFCO*CVUNIT(J),J=1,MXCOMP),FO
1140 CONTINUE
C
C
C
ELSE
C
C
C
        ... TANK CONCENTRATIONS ...
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'TANK(AQUEOUS)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''

```

```

C
DO 1210 I = 1 , NSTG
    IND = IPOS(1,1,IUNT)
    IND = IPOS(2,1,IUNT)
    DENFCA = 1.0D0
    FA = FOUT(1,IUNT) * DENFCA
    WRITE(IUPLT,5300)
*           I, ( XN(J,1,IND)/DENFCA*CVUNIT(J),J=1,MXCOMP),FA
1210 CONTINUE
C
        WRITE(IUPLT,' (2I20,1X,A)') ID(IUNT),MSTG(IUNT),
*           'TANK(ORGANIC)'
C
        WRITE(IUPLT,5200) ( CNAME(J),J=1,MXCOMP),'FLW_RATE'
        WRITE(IUPLT,5200) ( CUNIT(J),J=1,MXCOMP),'(L/HR)''
C
DO 240 I = 1 , NSTG
    IND = IPOS(1,1,IUNT)
    IND = IPOS(2,1,IUNT)
    DENFCO = 1.0D0
    FO = FOUT(2,IUNT) * DENFCO
    WRITE(IUPLT,5300)
*           I, ( XN(J,2,IND)/DENFCO*CVUNIT(J),J=1,MXCOMP),FO
240 CONTINUE
C
    END IF
C
C
2000 CONTINUE
C
        WRITE(IUPLT,' (5X,A,5X,A)') 'CONCENTRATION PROFILE OF ',TITLE
C
    RETURN
C
C
5000 FORMAT(120,2X,1P,E18.5)
5100 FORMAT(120,9X,A2,9X,120)
5200 FORMAT(20X,50(6X,A8,6X))
5300 FORMAT(120,1P,50E20.5)
C
    END

```

```

SUBROUTINE OUTPTF( IFOUT , IPRDIS,
*      TITLE , TIME , ENDTIM, DT ,
*      CNAME , FMASS , DECAH, CVUNIT,
*      CUNIT , CHRG , CNTBP , CPHASE,
*      MXCOMP, MXUNIT, MXSTG , MXSTG2, NCHEM , MXBLK ,
*      MXEXTR, NUNIT, ISFIN , FRC , CO , TO ,
*      ID , TYPE , FOUT , MSTG , MEXTRA,
*      FL , TSTG , WLT , WST , HOS , TF ,
*      IEFF , EFF , DIS ,
*      IPOS ,
*      XN )
C
C
C
C PRINT TRANSIENT DATA
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C CHARACTER* 72 TITLE
C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRG (MXCOMP) ,
*      CNTBP (MXCOMP)
C
C CHARACTER*8 CPHASE (MXCOMP)
C
C DIMENSION ID ( MXUNIT)
C CHARACTER*2 TYPE ( MXUNIT)
C DIMENSION FOUT ( 2 , MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION MEXTRA ( MXUNIT)
C
C DIMENSION FL (2 , MXSTG , MEXEXTR) ,
*      TSTG ( MXSTG , MEXEXTR) ,
*      WLT ( MXSTG , MEXEXTR) ,
*      WST ( MXSTG , MEXEXTR) ,
*      HOS ( MXSTG , MEXEXTR) ,
*      TF ( MXSTG , MEXEXTR)
C
C DIMENSION IEFF (MXCOMP , MEXEXTR)
C
C DIMENSION EFF (MXCOMP, MXSTG , MEXEXTR) ,
*      DIS (MXCOMP, MXSTG , MEXEXTR)
C
C DIMENSION IPOS (2, MXSTG2, MXUNIT)
C
C DIMENSION XN (MXCOMP, 2, MXBLK )
C
COMMON /IXCOMP/NCHEMO,
*      IOHNO3, IOU6 , IOU4 , IOU3 , IOU4 ,
*      IOHNO2, IOHYD , IOHAN , IOU5 , IOU6 ,
*      IOHP4 , IOHP5 , IOHP6 , IOZR , IOZC4 ,
*      IOZC5 , IOZC6 , IOZC7 , IOZC6 , IOZCP4 ,
*      IOZCZR , IOSR , IOZUDI , IOZUTR , IOZUNI ,
*      IOZUMO , IOCS , IOCE , IOGD , IOAM ,
*      IOCM
C
C
C DO 100 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA(IUNT)
C CO = CO
C TTO = TO
C
C CALL DENST( MXSTG , NSTG ,
*      ISFIN , FRC ,
*      CO , TTO ,
*      TSTG(1, IEXTRA) , WLT(1, IEXTRA) ,
*      WST(1, IEXTRA) , HOS(1, IEXTRA) )
100 CONTINUE
C
C DO 1000 IUNT = 1 , NUNIT
C
C NSTG = MSTG (IUNT)
C IEXTRA = MEXTRA(IUNT)
C
IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
... MIXER-SETTLER CONCENTRATIONS ...
WRITE( IFOUT, 5010) TITLE, TIME
WRITE( IFOUT, 5100) ID(IUNT), TYPE(IUNT),
' CONCENTRATIONS AT MIXERS (AQUEOUS)'
WRITE( IFOUT, 5200) CNAME( IOPU5 ),
*      CNAME( IOPU6 ),
*      CNAME( IONP4 ),
*      CNAME( IONP5 ),
*      CNAME( IONP6 ),
*      CNAME( IOZR ),
*      CNAME( IOTC4 ),
*      CNAME( IOTC5 ),
*      CNAME( IOTC6 ),
*      CNAME( IOTC7 )
WRITE( IFOUT, 5210) CUNIT( IOPU5 ),
*      CUNIT( IOPU6 ),
*      CUNIT( IONP4 ),
*      CUNIT( IONP5 ),
*      CUNIT( IONP6 ),
*      CUNIT( IOZR ),
*      CUNIT( IOTC4 ),
*      CUNIT( IOTC5 ),
*      CUNIT( IOTC6 ),
*      CUNIT( IOTC7 )
DO 120 I = 1 , NSTG
IND = IPOS(1, 2*I-1, IUNT)
IND = IPOS(2, 2*I-1, IUNT)
DENFCA = DEN SMA( MXCOMP , XN(1, 1, IND) )
XPU5 = XN( IOPU5 , 1, IND) / DENFCA * CVUNIT( IOPU5 )
XPU6 = XN( IOPU6 , 1, IND) / DENFCA * CVUNIT( IOPU6 )
XNP4 = XN( IONP4 , 1, IND) / DENFCA * CVUNIT( IONP4 )
XNP5 = XN( IONP5 , 1, IND) / DENFCA * CVUNIT( IONP5 )
XNP6 = XN( IONP6 , 1, IND) / DENFCA * CVUNIT( IONP6 )
XZR = XN( IOZR , 1, IND) / DENFCA * CVUNIT( IOZR )
XTC4 = XN( IOTC4 , 1, IND) / DENFCA * CVUNIT( IOTC4 )
XTC5 = XN( IOTC5 , 1, IND) / DENFCA * CVUNIT( IOTC5 )
XTC6 = XN( IOTC6 , 1, IND) / DENFCA * CVUNIT( IOTC6 )
XTC7 = XN( IOTC7 , 1, IND) / DENFCA * CVUNIT( IOTC7 )
FA = FL(1, 1, IEXTRA) * DENFCA
WRITE( IFOUT, 5220) I ,
*      XPU5 ,
*      XPU6 ,
*      XNP4 ,
*      XNP5 ,
*      XNP6 ,
*      XZR ,
*      XTC4 ,
*      XTC5 ,
*      XTC6 ,
*      XTC7
CONTINUE
WRITE( IFOUT, 5010) TITLE, TIME
WRITE( IFOUT, 5100) ID(IUNT), TYPE(IUNT),
' CONCENTRATIONS AT MIXERS (ORGANIC)'
WRITE( IFOUT, 5200) CNAME( IOPU5 ),
*      CNAME( IOPU6 ),
*      CNAME( IONP4 ),
*      CNAME( IONP5 ),
*      CNAME( IONP6 ),
*      CNAME( IOZR ),
*      CNAME( IOTCU6 ),
*      CNAME( IOTCP4 ),
*      CNAME( IOTCZR ),
*      CNAME( IOTC7 )
WRITE( IFOUT, 5210) CUNIT( IOPU5 ),
*      CUNIT( IOPU6 ),
*      CUNIT( IONP4 ),
*      CUNIT( IONP5 ),
*      CUNIT( IONP6 ),
*      CUNIT( IOZR ),
*      CUNIT( IOTCU6 ),
*      CUNIT( IOTCP4 ),
*      CUNIT( IOTCZR ),
*      CUNIT( IOTC7 )
DO 140 I = 1 , NSTG
IND = IPOS(1, 2*I-1, IUNT)
IND = IPOS(2, 2*I-1, IUNT)
DENFCO = DEN SMO( MXCOMP , WST(1, IEXTRA) , XN(1, 2, IND) )
YPU5 = XN( IOPU5 , 2, IND) / DENFCO * CVUNIT( IOPU5 )
YPU6 = XN( IOPU6 , 2, IND) / DENFCO * CVUNIT( IOPU6 )
YNP4 = XN( IONP4 , 2, IND) / DENFCO * CVUNIT( IONP4 )
YNP5 = XN( IONP5 , 2, IND) / DENFCO * CVUNIT( IONP5 )
YNP6 = XN( IONP6 , 2, IND) / DENFCO * CVUNIT( IONP6 )

```

```

    YZR = XN(10ZR ,2, IND) / DENFCO * CVUNIT(10ZR )
    YTC7 = XN(10TC7 ,2, IND) / DENFCO * CVUNIT(10TC7 )
    YTCU6 = XN(10TCU6,2, IND) / DENFCO * CVUNIT(10TCU6)
    YTCP4 = XN(10TCP4,2, IND) / DENFCO * CVUNIT(10TCP4)
    YTCZR = XN(10TCZR,2, IND) / DENFCO * CVUNIT(10TCZR)
    CUNIT(10NP6 ),
    CUNIT(10ZR ),
    CUNIT(10TCU6),
    CUNIT(10TCP4),
    CUNIT(10TCZR),
    CUNIT(10TC7 )

C
    FO = FL(2,1, IEXTRA) * DENFCO
    WRITE(1FOUT,5220) I ,
    *
    * YPU5 ,
    * YPU6 ,
    * YNP4 ,
    * YNP5 ,
    * YNP6 ,
    * YZR ,
    * YTCU6 ,
    * YTCP4 ,
    * YTCZR ,
    * YTC7
140 CONTINUE

WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
* 'CONCENTRATIONS AT SETTLERS(AQUEOUS)'
WRITE(1FOUT,5200) CNAME(10PU5 ),
* CNAME(10PU6 ),
* CNAME(10NP4 ),
* CNAME(10NP5 ),
* CNAME(10NP6 ),
* CNAME(10ZR ),
* CNAME(10TC4 ),
* CNAME(10TC5 ),
* CNAME(10TC6 ),
* CNAME(10TC7 )
WRITE(1FOUT,5210) CUNIT(10PU5 ),
* CUNIT(10PU6 ),
* CUNIT(10NP4 ),
* CUNIT(10NP5 ),
* CUNIT(10NP6 ),
* CUNIT(10ZR ),
* CUNIT(10TC4 ),
* CUNIT(10TC5 ),
* CUNIT(10TC6 ),
* CUNIT(10TC7 )
DO 160 I = 1 , NSTG
    INDSA = IPOS(1,2*I, IUNT)
    INDSO = IPOS(2,2*I, IUNT)
    DENFCA = DENFCA( MXCOMP ,XN(1,1, INDSA) )
    XPU5 = XN(10PU5 ,1, INDSA) / DENFCA * CVUNIT(10PU5 )
    XPU6 = XN(10PU6 ,1, INDSA) / DENFCA * CVUNIT(10PU6 )
    XNP4 = XN(10NP4 ,1, INDSA) / DENFCA * CVUNIT(10NP4 )
    XNP5 = XN(10NP5 ,1, INDSA) / DENFCA * CVUNIT(10NP5 )
    XNP6 = XN(10NP6 ,1, INDSA) / DENFCA * CVUNIT(10NP6 )
    XZR = XN(10ZR ,1, INDSA) / DENFCA * CVUNIT(10ZR )
    XTC4 = XN(10TC4 ,1, INDSA) / DENFCA * CVUNIT(10TC4 )
    XTC5 = XN(10TC5 ,1, INDSA) / DENFCA * CVUNIT(10TC5 )
    XTC6 = XN(10TC6 ,1, INDSA) / DENFCA * CVUNIT(10TC6 )
    XTC7 = XN(10TC7 ,1, INDSA) / DENFCA * CVUNIT(10TC7 )
C
    FA = FL(1,1, IEXTRA) * DENFCA
    WRITE(1FOUT,5220) I ,
    * XPU5 ,
    * XPU6 ,
    * XNP4 ,
    * XNP5 ,
    * XNP6 ,
    * XZR ,
    * XTC4 ,
    * XTC5 ,
    * XTC6 ,
    * XTC7
160 CONTINUE

WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
* 'CONCENTRATIONS AT SETTLERS(ORGANIC)'
WRITE(1FOUT,5200) CNAME(10PU5 ),
* CNAME(10PU6 ),
* CNAME(10NP4 ),
* CNAME(10NP5 ),
* CNAME(10NP6 ),
* CNAME(10ZR ),
* CNAME(10TCU6),
* CNAME(10TCP4),
* CNAME(10TCZR),
* CNAME(10TC7 )
WRITE(1FOUT,5210) CUNIT(10PU5 ),
* CUNIT(10PU6 ),
* CUNIT(10NP4 ),
* CUNIT(10NP5 ),
    CUNIT(10NP6 ),
    CUNIT(10ZR ),
    CUNIT(10TCU6),
    CUNIT(10TCP4),
    CUNIT(10TCZR),
    CUNIT(10TC7 )

C
    FO = FL(2,1, IEXTRA) * DENFCO
    WRITE(1FOUT,5220) I ,
    *
    * YPU5 ,
    * YPU6 ,
    * YNP4 ,
    * YNP5 ,
    * YNP6 ,
    * YZR ,
    * YTCU6 ,
    * YTCP4 ,
    * YTCZR ,
    * YTC7
180 CONTINUE

IF ( IPRDIS .EQ. 1 ) THEN
    WRITE(1FOUT,5010) TITLE, TIME
    WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
    * 'DISTRIBUTION COEFFICIENTS.'
    WRITE(1FOUT,5300) CNAME(10PU5 ),
    * CNAME(10PU6 ),
    * CNAME(10NP4 ),
    * CNAME(10NP5 ),
    * CNAME(10NP6 ),
    * CNAME(10ZR ),
    * CNAME(10TC7 ),
    * CNAME(10TCU6),
    * CNAME(10TCP4),
    * CNAME(10TCZR)
DO 191 I = 1 , NSTG
    WRITE(1FOUT,5320) I ,
    * DIS(10PU5 ,1, IEXTRA),
    * DIS(10PU6 ,1, IEXTRA),
    * DIS(10NP4 ,1, IEXTRA),
    * DIS(10NP5 ,1, IEXTRA),
    * DIS(10NP6 ,1, IEXTRA),
    * DIS(10ZR ,1, IEXTRA),
    * DIS(10TC7 ,1, IEXTRA),
    * DIS(10TCU6,1, IEXTRA),
    * DIS(10TCP4,1, IEXTRA),
    * DIS(10TCZR,1, IEXTRA)
191 CONTINUE

DO 192 J = 1 , MXCOMP
    IF ( IEFF(J, IEXTRA) .EQ. 1 ) THEN
        CPHASE(J) = 'AQUEOUS'
    ELSE
        CPHASE(J) = 'ORGANIC'
    END IF
192 CONTINUE

WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
* 'EFFICIENCY OF RECOVERY RATE.'
WRITE(1FOUT,5300) CNAME(10PU5 ),
* CNAME(10PU6 ),
* CNAME(10NP4 ),
* CNAME(10NP5 ),
* CNAME(10NP6 ),
* CNAME(10ZR ),
* CNAME(10TC7 ),
* CNAME(10TCU6),
* CNAME(10TCP4),
* CNAME(10TCZR)
WRITE(1FOUT,5310) CPHASE(10PU5 ),
* CPHASE(10PU6 ),

```

```

*          CPHASE(10NP4 ),
*          CPHASE(10NP5 ),
*          CPHASE(10NP6 ),
*          CPHASE(10ZR ),
*          CPHASE(10TC7 ),
*          CPHASE(10TCU6),
*          CPHASE(10TCP4),
*          CPHASE(10TCZR)
C
DO 193 I = 1 , NSTG
C
WRITE(1FOUT,5320) I
*          EFF(10PU5 ,1,1,EXTRA),
*          EFF(10PU6 ,1,1,EXTRA),
*          EFF(10NP4 ,1,1,EXTRA),
*          EFF(10NP5 ,1,1,EXTRA),
*          EFF(10NP6 ,1,1,EXTRA),
*          EFF(10ZR ,1,1,EXTRA),
*          EFF(10TC7 ,1,1,EXTRA),
*          EFF(10TCU6,1,1,EXTRA),
*          EFF(10TCP4,1,1,EXTRA),
*          EFF(10TCZR,1,1,EXTRA)
193 CONTINUE
C
END IF
C
C
C ELSE
C
... TANK CONCENTRATIONS ...
C
WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
* CONCENTRATIONS IN TANK(AQUEOUS)
WRITE(1FOUT,5200) CNAME(10PU5 ),
* CNAME(10PU6 ),
* CNAME(10NP4 ),
* CNAME(10NP5 ),
* CNAME(10NP6 ),
* CNAME(10ZR ),
* CNAME(10TC4 ),
* CNAME(10TC5 ),
* CNAME(10TC6 ),
* CNAME(10TC7 )
WRITE(1FOUT,5210) CUNIT(10PU5 ),
* CUNIT(10PU6 ),
* CUNIT(10NP4 ),
* CUNIT(10NP5 ),
* CUNIT(10NP6 ),
* CUNIT(10ZR ),
* CUNIT(10TC4 ),
* CUNIT(10TC5 ),
* CUNIT(10TC6 ),
* CUNIT(10TC7 )
DO 220 I = 1 , NSTG
IND = IPOS(1,I,IUNT)
IND = IPOS(2,I,IUNT)
DENFCA = 1.000
XPU5 = XN(10PU5 ,1,IND) / DENFCA * CVUNIT(10PU5 )
XPU6 = XN(10PU6 ,1,IND) / DENFCA * CVUNIT(10PU6 )
XNP4 = XN(10NP4 ,1,IND) / DENFCA * CVUNIT(10NP4 )
XNP5 = XN(10NP5 ,1,IND) / DENFCA * CVUNIT(10NP5 )
XNP6 = XN(10NP6 ,1,IND) / DENFCA * CVUNIT(10NP6 )
XZR = XN(10ZR ,1,IND) / DENFCA * CVUNIT(10ZR )
XTC4 = XN(10TC4 ,1,IND) / DENFCA * CVUNIT(10TC4 )
XTC5 = XN(10TC5 ,1,IND) / DENFCA * CVUNIT(10TC5 )
XTC6 = XN(10TC6 ,1,IND) / DENFCA * CVUNIT(10TC6 )
XTC7 = XN(10TC7 ,1,IND) / DENFCA * CVUNIT(10TC7 )
C
FA = FOUT(1,IUNT) * DENFCA
WRITE(1FOUT,5220) I
* XPU5 ,
* XPU6 ,
* XNP4 ,
* XNP5 ,
* XNP6 ,
* XZR ,
* XTC4 ,
* XTC5 ,
* XTC6 ,
* XTC7
220 CONTINUE
C
WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
* CONCENTRATIONS IN TANK(ORGANIC)
WRITE(1FOUT,5200) CNAME(10PU5 ),

```

```

*          CNAME(10PU6 ),
*          CNAME(10NP4 ),
*          CNAME(10NP5 ),
*          CNAME(10NP6 ),
*          CNAME(10ZR ),
*          CNAME(10TCU6),
*          CNAME(10TCP4),
*          CNAME(10TCZR),
*          CNAME(10TC7 )
*
WRITE(1FOUT,5210) CUNIT(10PU5 ),
* CUNIT(10PU6 ),
* CUNIT(10NP4 ),
* CUNIT(10NP5 ),
* CUNIT(10NP6 ),
* CUNIT(10ZR ),
* CUNIT(10TCU6),
* CUNIT(10TCP4),
* CUNIT(10TCZR),
* CUNIT(10TC7 )
DO 240 I = 1 , NSTG
IND = IPOS(1,I,IUNT)
IND = IPOS(2,I,IUNT)
DENFCO = 1.000
YPU5 = XN(10PU5 ,2,IND) / DENFCO * CVUNIT(10PU5 )
YPU6 = XN(10PU6 ,2,IND) / DENFCO * CVUNIT(10PU6 )
YNP4 = XN(10NP4 ,2,IND) / DENFCO * CVUNIT(10NP4 )
YNP5 = XN(10NP5 ,2,IND) / DENFCO * CVUNIT(10NP5 )
YNP6 = XN(10NP6 ,2,IND) / DENFCO * CVUNIT(10NP6 )
YZR = XN(10ZR ,2,IND) / DENFCO * CVUNIT(10ZR )
YTC7 = XN(10TC7 ,2,IND) / DENFCO * CVUNIT(10TC7 )
YTCU6 = XN(10TCU6,2,IND) / DENFCO * CVUNIT(10TCU6)
YTCP4 = XN(10TCP4,2,IND) / DENFCO * CVUNIT(10TCP4)
YTCZR = XN(10TCZR,2,IND) / DENFCO * CVUNIT(10TCZR)
C
FO = FOUT(2,IUNT) * DENFCO
WRITE(1FOUT,5220) I
* YPU5 ,
* YPU6 ,
* YNP4 ,
* YNP5 ,
* YNP6 ,
* YZR ,
* YTCU6 ,
* YTCP4 ,
* YTCZR ,
* YTC7
240 CONTINUE
C
END IF
C
1000 CONTINUE
C
RETURN
C
5010 FORMAT(1H /1H /1H ,A,5X,' TIME:',1P,E12.5,' (HR)')
5100 FORMAT(1H /1H ,A,UNIT ID,' 15.1X,A2.5X,A)
5200 FORMAT(1H ,A,STAGE ',A8,9(5X,A8))
5210 FORMAT(1H ,A,A8,9(5X,A8))
5220 FORMAT(1H ,I2,1P,10E13.5)
C
5300 FORMAT(1H ,A,STAGE ',A8,1X,9(4X,A8,1X))
5310 FORMAT(1H ,A,A8,1X,9(4X,A8,1X))
5320 FORMAT(1H ,I2,1P,15E13.5)
C
END

```

```

SUBROUTINE OUTPUT(IFOUT,IPRDIS,
*      TITLE,TIME,ENDTIM,DT
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,CPHASE,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,MLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
C
C
C      PRINT TRANSIENT DATA
C
C      IMPLICIT REAL*8 ( A-H , 0-Z )
C
C      CHARACTER*72 TITLE
C
C      CHARACTER*8 CNAME (MXCOMP)
C      DIMENSION FMASS (MXCOMP)
C      DIMENSION DECAYH (MXCOMP)
C      DIMENSION CVUNIT (MXCOMP)
C      CHARACTER*8 CUNIT (MXCOMP)
C      DIMENSION CHRG (MXCOMP),
*      CNTBP (MXCOMP)
C
C      CHARACTER*8 CPHASE (MXCOMP)
C
C      DIMENSION ID ( MXUNIT)
C      CHARACTER*2 TYPE ( MXUNIT)
C      DIMENSION FOUT ( 2 ,MXUNIT)
C
C      DIMENSION MSTG ( MXUNIT)
C      DIMENSION MEXTRA ( MXUNIT)
C
C      DIMENSION FL (2 ,MXSTG ,MEXTR),
*      TSTG ( MXSTG ,MEXTR),
*      MLT ( MXSTG ,MEXTR),
*      WST ( MXSTG ,MEXTR),
*      HOS ( MXSTG ,MEXTR),
*      TF ( MXSTG ,MEXTR)
C
C      DIMENSION IEFF (MXCOMP ,MXEXTR)
C
C      DIMENSION EFF (MXCOMP,MXSTG,MEXTR),
*      DIS (MXCOMP,MXSTG,MEXTR)
C
C      DIMENSION IPOS (2,MXSTG2,MXUNIT)
C
C      DIMENSION XN (MXCOMP,2,MXBLK )
C
C      COMMON /DAY/CDATE
C      CHARACTER CDATE*9
C
C      COMMON /IXCOMP/NCHEMO,
*      IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*      IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOU6 ,
*      IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*      IOTC5 ,IOTC6 ,IOTC7 ,IOTC6 ,IOTCP4 ,
*      IOTCZR,IOSR ,IORU1 ,IORUTR,IORUN1 ,
*      IORUMO,IOCS ,IOCE ,IOGD ,IOAM ,
*      IOCM
C
C      WRITE(IFOUT,5000) NUNIT ,CDATE
C
C      DO 100 IUNT = 1 , NUNIT
C
C          NSTG = MSTG (IUNT)
C          IEXTRA = MEXTRA(IUNT)
C          CCO = CO
C          TTO = TO
C
C          CALL DENST( MXSTG ,NSTG ,
*      ISFIN ,FRC ,
*      CCO ,TTO ,
*      TSTG(1,IEXTRA) ,MLT(1,IEXTRA) ,
*      WST (1,IEXTRA) ,HOS(1,IEXTRA) )
100 CONTINUE
C
C
C      DO 1000 IUNT = 1 , NUNIT
C
C          NSTG = MSTG (IUNT)
C          IEXTRA = MEXTRA(IUNT)
C
C          IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C              ... MIXER-SETTLER CONCENTRATIONS ...
C
C              WRITE(IFOUT,5010) TITLE,TIME
C              WRITE(IFOUT,5100) ID(IUNT),TYPE(IUNT),
*      'CONCENTRATIONS AT MIXERS(AQUEOUS)'
C              WRITE(IFOUT,5200) CNAME(IOHNO3),
*      CNAME(IOU6 ),
*      CNAME(IOPU4 ),
*      CNAME(IOPU3 ),
*      CNAME(IOU4 ),
*      CNAME(IOHNO2),
*      CNAME(IOHYD ),
*      CNAME(IOHAN )
C              WRITE(IFOUT,5210) CUNIT(IOHNO3),
*      CUNIT(IOU6 ),
*      CUNIT(IOPU4 ),
*      CUNIT(IOPU3 ),
*      CUNIT(IOU4 ),
*      CUNIT(IOHNO2),
*      CUNIT(IOHYD ),
*      CUNIT(IOHAN )
C
C              DO 120 I = 1 , NSTG
C                  IND = IPOS(1,2*I-1,IUNT)
C                  IND = IPOS(2,2*I-1,IUNT)
C                  DENFCA = DENFCA( MXCOMP ,XN(1,1,IND) )
C                  XH = XN(IOHNO3,1,IND) / DENFCA * CVUNIT(IOHNO3)
C                  XU6 = XN(IOU6 ,1,IND) / DENFCA * CVUNIT(IOU6 )
C                  XPU4 = XN(IOPU4 ,1,IND) / DENFCA * CVUNIT(IOPU4 )
C                  XPU3 = XN(IOPU3 ,1,IND) / DENFCA * CVUNIT(IOPU3 )
C                  XU4 = XN(IOU4 ,1,IND) / DENFCA * CVUNIT(IOU4 )
C                  XNO2 = XN(IOHNO2,1,IND) / DENFCA * CVUNIT(IOHNO2)
C                  XHYD = XN(IOHYD ,1,IND) / DENFCA * CVUNIT(IOHYD )
C                  XHAN = XN(IOHAN ,1,IND) / DENFCA * CVUNIT(IOHAN )
C                  XTF = TF(1,IEXTRA)
C
C                  XNO3 = 0.000
C                  DO 110 J = 1 , NCHEMO
C                      XNO3 = XNO3 + CHRG(J)*XN(J,1,IND)
110 CONTINUE
C
C                  FA = FL(1,1,IEXTRA) * DENFCA
C                  WRITE(IFOUT,5220) I ,
*      XH ,
*      XU6 ,
*      XPU4 ,
*      XPU3 ,
*      XU4 ,
*      XNO2 ,
*      XHYD ,
*      XHAN ,
*      XNO3 ,
*      XTF ,
*      FA
C
C              CONTINUE
C
C              WRITE(IFOUT,5010) TITLE,TIME
C              WRITE(IFOUT,5100) ID(IUNT),TYPE(IUNT),
*      'CONCENTRATIONS AT MIXERS(ORGANIC)'
C              WRITE(IFOUT,5200) CNAME(IOHNO3),
*      CNAME(IOU6 ),
*      CNAME(IOPU4 ),
*      CNAME(IOPU3 ),
*      CNAME(IOU4 ),
*      CNAME(IOHNO2),
*      CNAME(IOHYD ),
*      CNAME(IOHAN )
C              WRITE(IFOUT,5210) CUNIT(IOHNO3),
*      CUNIT(IOU6 ),
*      CUNIT(IOPU4 ),
*      CUNIT(IOPU3 ),
*      CUNIT(IOU4 ),
*      CUNIT(IOHNO2),
*      CUNIT(IOHYD ),
*      CUNIT(IOHAN )
C
C              DO 140 I = 1 , NSTG
C                  IND = IPOS(1,2*I-1,IUNT)
C                  IND = IPOS(2,2*I-1,IUNT)
C                  DENFCO = DENFCO( MXCOMP ,WST(1,IEXTRA) ,XN(1,2,IND) )
C                  YH = XN(IOHNO3,2,IND) / DENFCO * CVUNIT(IOHNO3)
C                  YU6 = XN(IOU6 ,2,IND) / DENFCO * CVUNIT(IOU6 )

```

```

YPU4 = XN(10PU4 ,2, IND) / DENFCO * CVUNIT(10PU4 )
YPU3 = XN(10PU3 ,2, IND) / DENFCO * CVUNIT(10PU3 )
YU4 = XN(10U4 ,2, IND) / DENFCO * CVUNIT(10U4 )
YNO2 = XN(10HNO2,2, IND) / DENFCO * CVUNIT(10HNO2)
YHYD = XN(10HYD ,2, IND) / DENFCO * CVUNIT(10HYD )
YHAN = XN(10HAN ,2, IND) / DENFCO * CVUNIT(10HAN )
YTF = TF(1, IEXTRA)

```

```

C
YNO3 = 0.0D0
DO 130 J = 1 , NCHEMO
  YNO3 = YNO3 + CHR(J)*XN(J,1, IND)
CONTINUE

```

```

130 C
FO = FL(2,1, IEXTRA) * DENFCO
WRITE(1FOUT, 5220) I ,
*
* YH ,
* YU6 ,
* YPU4 ,
* YPU3 ,
* YU4 ,
* YNO2 ,
* YHYD ,
* YHAN ,
* YNO3 ,
* YTF ,
*
FO

```

```

140 C
CONTINUE
WRITE(1FOUT, 5010) TITLE, TIME
WRITE(1FOUT, 5100) ID(IUNT), TYPE(IUNT),
*
* 'CONCENTRATIONS AT SETTLERS(AQUEOUS)'
WRITE(1FOUT, 5200) CNAME(10HNO3),
*
* CNAME(10U6 ),
*
* CNAME(10PU4 ),
*
* CNAME(10PU3 ),
*
* CNAME(10U4 ),
*
* CNAME(10HNO2),
*
* CNAME(10HYD ),
*
* CNAME(10HAN )
WRITE(1FOUT, 5210) CUNIT(10HNO3),
*
* CUNIT(10U6 ),
*
* CUNIT(10PU4 ),
*
* CUNIT(10PU3 ),
*
* CUNIT(10U4 ),
*
* CUNIT(10HNO2),
*
* CUNIT(10HYD ),
*
* CUNIT(10HAN )

```

```

DO 160 I = 1 , NSTG
INDSA = IPOS(1,2*1, IUNT)
INDSO = IPOS(2,2*1, IUNT)
DENFCA = DENSMO( MXCOMP ,XN(1,1, INDSA) )
XH = XN(10HNO3,1, INDSA) / DENFCA * CVUNIT(10HNO3)
XU6 = XN(10U6 ,1, INDSA) / DENFCA * CVUNIT(10U6 )
XPU4 = XN(10PU4 ,1, INDSA) / DENFCA * CVUNIT(10PU4 )
XPU3 = XN(10PU3 ,1, INDSA) / DENFCA * CVUNIT(10PU3 )
XU4 = XN(10U4 ,1, INDSA) / DENFCA * CVUNIT(10U4 )
XNO2 = XN(10HNO2,1, INDSA) / DENFCA * CVUNIT(10HNO2)
XHYD = XN(10HYD ,1, INDSA) / DENFCA * CVUNIT(10HYD )
XHAN = XN(10HAN ,1, INDSA) / DENFCA * CVUNIT(10HAN )
XTF = TF(1, IEXTRA)

```

```

C
XNO3 = 0.0D0
DO 150 J = 1 , NCHEMO
  XNO3 = XNO3 + CHR(J)*XN(J,1, INDSA)
CONTINUE

```

```

150 C
FA = FL(1,1, IEXTRA) * DENFCA
WRITE(1FOUT, 5220) I ,
*
* XH ,
*
* XU6 ,
*
* XPU4 ,
*
* XPU3 ,
*
* XU4 ,
*
* XNO2 ,
*
* XHYD ,
*
* XHAN ,
*
* XNO3 ,
*
* XTF ,
*
FA

```

```

160 C
CONTINUE
WRITE(1FOUT, 5010) TITLE, TIME
WRITE(1FOUT, 5100) ID(IUNT), TYPE(IUNT),
*
* 'CONCENTRATIONS AT SETTLERS(ORGANIC)'
WRITE(1FOUT, 5200) CNAME(10HNO3),
*
* CNAME(10U6 ),
*
* CNAME(10PU4 ),
*
* CNAME(10PU3 ),
*
* CNAME(10U4 ),

```

```

*
* CNAME(10HNO2),
*
* CNAME(10HYD ),
*
* CNAME(10HAN )
WRITE(1FOUT, 5210) CUNIT(10HNO3),
*
* CUNIT(10U6 ),
*
* CUNIT(10PU4 ),
*
* CUNIT(10PU3 ),
*
* CUNIT(10U4 ),
*
* CUNIT(10HNO2),
*
* CUNIT(10HYD ),
*
* CUNIT(10HAN )

```

```

DO 180 I = 1 , NSTG
INDSA = IPOS(1,2*1, IUNT)
INDSO = IPOS(2,2*1, IUNT)
DENFCO = DENSMO( MXCOMP ,WST(1, IEXTRA) ,XN(1,2, INDSO) )
YH = XN(10HNO3,2, INDSO) / DENFCO * CVUNIT(10HNO3)
YU6 = XN(10U6 ,2, INDSO) / DENFCO * CVUNIT(10U6 )
YPU4 = XN(10PU4 ,2, INDSO) / DENFCO * CVUNIT(10PU4 )
YPU3 = XN(10PU3 ,2, INDSO) / DENFCO * CVUNIT(10PU3 )
YU4 = XN(10U4 ,2, INDSO) / DENFCO * CVUNIT(10U4 )
YNO2 = XN(10HNO2,2, INDSO) / DENFCO * CVUNIT(10HNO2)
YHYD = XN(10HYD ,2, INDSO) / DENFCO * CVUNIT(10HYD )
YHAN = XN(10HAN ,2, INDSO) / DENFCO * CVUNIT(10HAN )
YTF = TF(1, IEXTRA)

```

```

C
YNO3 = 0.0D0
DO 170 J = 1 , NCHEMO
  YNO3 = YNO3 + CHR(J)*XN(J,1, INDSA)
CONTINUE

```

```

170 C
FO = FL(2,1, IEXTRA) * DENFCO
WRITE(1FOUT, 5220) I ,
*
* YH ,
*
* YU6 ,
*
* YPU4 ,
*
* YPU3 ,
*
* YU4 ,
*
* YNO2 ,
*
* YHYD ,
*
* YHAN ,
*
* YNO3 ,
*
* YTF ,
*
FO

```

```

180 C
CONTINUE
IF ( IPRDIS .EQ. 1 ) THEN
C
WRITE(1FOUT, 5010) TITLE, TIME
WRITE(1FOUT, 5100) ID(IUNT), TYPE(IUNT),
*
* 'DISTRIBUTION COEFFICIENTS.'
WRITE(1FOUT, 5300) CNAME(10HNO3),
*
* CNAME(10U6 ),
*
* CNAME(10PU4 ),
*
* CNAME(10PU3 ),
*
* CNAME(10U4 ),
*
* CNAME(10HNO2),
*
* CNAME(10HYD ),
*
* CNAME(10HAN )

```

```

DO 191 I = 1 , NSTG
WRITE(1FOUT, 5320) I ,
*
* DIS(10HNO3, I, IEXTRA),
*
* DIS(10U6 , I, IEXTRA),
*
* DIS(10PU4 , I, IEXTRA),
*
* DIS(10PU3 , I, IEXTRA),
*
* DIS(10U4 , I, IEXTRA),
*
* DIS(10HNO2, I, IEXTRA),
*
* DIS(10HYD , I, IEXTRA),
*
* DIS(10HAN , I, IEXTRA)

```

```

191 C
CONTINUE
DO 192 J = 1 , MXCOMP
IF ( IEFF(J, IEXTRA) .EQ. 1 ) THEN
  CPHASE(J) = 'AQUEOUS'
ELSE
  CPHASE(J) = 'ORGANIC'
END IF

```

```

192 C
CONTINUE
WRITE(1FOUT, 5010) TITLE, TIME
WRITE(1FOUT, 5100) ID(IUNT), TYPE(IUNT),
*
* 'EFFICIENCY OF RECOVERY RATE.'
WRITE(1FOUT, 5300) CNAME(10HNO3),
*
* CNAME(10U6 ),
*
* CNAME(10PU4 ),
*
* CNAME(10PU3 ),
*
* CNAME(10U4 ),
*
* CNAME(10HNO2),

```

```

*          CNAME(10HYD ),
*          CNAME(10HAN )
WRITE(1FOUT,5310) CPHASE(10HNO3),
*                CPHASE(10U6 ),
*                CPHASE(10PU4 ),
*                CPHASE(10PU3 ),
*                CPHASE(10U4 ),
*                CPHASE(10HNO2),
*                CPHASE(10HYD ),
*                CPHASE(10HAN )
C
DO 193 I = 1 , NSTG
C
    WRITE(1FOUT,5320) I ,
*                EFF(10HNO3,1,1,EXTRA),
*                EFF(10U6 ,1,1,EXTRA),
*                EFF(10PU4 ,1,1,EXTRA),
*                EFF(10PU3 ,1,1,EXTRA),
*                EFF(10U4 ,1,1,EXTRA),
*                EFF(10HNO2,1,1,EXTRA),
*                EFF(10HYD ,1,1,EXTRA),
*                EFF(10HAN ,1,1,EXTRA)
193 CONTINUE
C
    END IF
C
C
C
ELSE
C
C
C
... TANK CONCENTRATIONS ...
C
WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
*                'CONCENTRATIONS IN TANK(AQUEOUS)'
WRITE(1FOUT,5200) CNAME(10HNO3),
*                CNAME(10U6 ),
*                CNAME(10PU4 ),
*                CNAME(10PU3 ),
*                CNAME(10U4 ),
*                CNAME(10HNO2),
*                CNAME(10HYD ),
*                CNAME(10HAN )
WRITE(1FOUT,5210) CUNIT(10HNO3),
*                CUNIT(10U6 ),
*                CUNIT(10PU4 ),
*                CUNIT(10PU3 ),
*                CUNIT(10U4 ),
*                CUNIT(10HNO2),
*                CUNIT(10HYD ),
*                CUNIT(10HAN )
DO 220 I = 1 , NSTG
    IND = IPOS(1,1, IUNT)
    IND = IPOS(2,1, IUNT)
    DENFCA = 1.000
    XH = XN(10HNO3,1,IND) / DENFCA * CVUNIT(10HNO3)
    XU6 = XN(10U6 ,1,IND) / DENFCA * CVUNIT(10U6 )
    XPU4 = XN(10PU4 ,1,IND) / DENFCA * CVUNIT(10PU4 )
    XPU3 = XN(10PU3 ,1,IND) / DENFCA * CVUNIT(10PU3 )
    XU4 = XN(10U4 ,1,IND) / DENFCA * CVUNIT(10U4 )
    XNO2 = XN(10HNO2,1,IND) / DENFCA * CVUNIT(10HNO2)
    XHYD = XN(10HYD ,1,IND) / DENFCA * CVUNIT(10HYD )
    XHAN = XN(10HAN ,1,IND) / DENFCA * CVUNIT(10HAN )
    XTF = 0.000
C
    XNO3 = 0.000
    DO 210 J = 1 , NCHEMO
        XNO3 = XNO3 + CHR(J)*XN(J,1,IND)
210 CONTINUE
    FA = FOUT(1, IUNT) * DENFCA
    WRITE(1FOUT,5220) I ,
*                XH ,
*                XU6 ,
*                XPU4 ,
*                XPU3 ,
*                XU4 ,
*                XNO2 ,
*                XHYD ,
*                XHAN ,
*                XNO3 ,
*                XTF ,
*                FA
220 CONTINUE
C
WRITE(1FOUT,5010) TITLE, TIME
WRITE(1FOUT,5100) ID(IUNT), TYPE(IUNT),
*                'CONCENTRATIONS IN TANK(ORGANIC)'

```

```

*          WRITE(1FOUT,5200) CNAME(10HNO3),
*                CNAME(10U6 ),
*                CNAME(10PU4 ),
*                CNAME(10PU3 ),
*                CNAME(10U4 ),
*                CNAME(10HNO2),
*                CNAME(10HYD ),
*                CNAME(10HAN )
WRITE(1FOUT,5210) CUNIT(10HNO3),
*                CUNIT(10U6 ),
*                CUNIT(10PU4 ),
*                CUNIT(10PU3 ),
*                CUNIT(10U4 ),
*                CUNIT(10HNO2),
*                CUNIT(10HYD ),
*                CUNIT(10HAN )
DO 240 I = 1 , NSTG
    IND = IPOS(1,1, IUNT)
    IND = IPOS(2,1, IUNT)
    DENFCO = 1.000
    YH = XN(10HNO3,2,IND) / DENFCO * CVUNIT(10HNO3)
    YU6 = XN(10U6 ,2,IND) / DENFCO * CVUNIT(10U6 )
    YPU4 = XN(10PU4 ,2,IND) / DENFCO * CVUNIT(10PU4 )
    YPU3 = XN(10PU3 ,2,IND) / DENFCO * CVUNIT(10PU3 )
    YU4 = XN(10U4 ,2,IND) / DENFCO * CVUNIT(10U4 )
    YNO2 = XN(10HNO2,2,IND) / DENFCO * CVUNIT(10HNO2)
    YHYD = XN(10HYD ,2,IND) / DENFCO * CVUNIT(10HYD )
    YHAN = XN(10HAN ,2,IND) / DENFCO * CVUNIT(10HAN )
    YTF = 0.000
C
    YNO3 = 0.000
    DO 230 J = 1 , NCHEMO
        YNO3 = YNO3 + CHR(J)*XN(J,1,IND)
230 CONTINUE
    FO = FOUT(2, IUNT) * DENFCO
    WRITE(1FOUT,5220) I ,
*                YH ,
*                YU6 ,
*                YPU4 ,
*                YPU3 ,
*                YU4 ,
*                YNO2 ,
*                YHYD ,
*                YHAN ,
*                YNO3 ,
*                YTF ,
*                FO
240 CONTINUE
C
    END IF
C
C
C
1000 CONTINUE
C
C
C
RETURN
C
5000 FORMAT(1H1/1H /1H , 'NUMBER_OF_UNIT(S) :', I3, T100, 'DATE ' , A9)
5010 FORMAT(1H /1H /1H , A, 5X, 'TIME :', 1P, E12.5, ' (HR)')
5100 FORMAT(1H /1H , 'UNIT ID.', 15, 1X, A2, 5X, A)
5200 FORMAT(1H , 'STAGE' , 8(1X, A8, 3X),
*          ' NO3- ',
*          ' TBP (FREE)',
*          ' FLOW_RATE')
5210 FORMAT(1H , ' , 8(1X, A8, 3X),
*          '(MOL/L) ',
*          '(MOL/L) ',
*          '(L/HR)')
5220 FORMAT(1H , I2, 1P, E11.4, 7E12.4, 2E11.3, E12.4)
C
5300 FORMAT(1H , 'STAGE ' , A8, 1X, 7(4X, A8, 1X))
5310 FORMAT(1H , ' , , A8, 1X, 7(4X, A8, 1X))
5320 FORMAT(1H , I2, 1P, 15E13.5)
C
END

```



```

SUBROUTINE POLATE(XX ,NYY ,
*          LENGTH,
*          TAB ,YY )
C
C*****
C
C LINEARLY INTERPOLATE TABLE TAB.
C   TAB(I,1)      : INDEPENDENT VARIABLES
C   TAB(J,1) (J>1) : DEPENDENT VARIABLES
C   XX            : THE POINT OF INTERPLATION
C   YY            : INTERPOLATED VALUES
C   NYY           : NUMBER OF INDEPENDENT VARIABLES
C   LENGTH        : TABLE LENGTH
C
C (N.B.) I ASSUME THAT TAB(1,1) ARE IN RISING ORDER AND DON'T CHECK
C          THE ORDER FOR FEAR OF CALCULATION TIME LOSS.
C
C*****
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C DIMENSION TAB(NYY+1,LENGTH) ,YY(NYY)
C
C IF ( LENGTH .LE. 1 ) THEN
C
C   DO 1 I = 1 , NYY
C     YY(I) = TAB(I+1,1)
1 CONTINUE
C   RETURN
C END IF
C
C
C IP = ( LENGTH - 1 )/2 + 1
C
C IF ( XX .GE. TAB(1,1) .AND. XX .LE. TAB(1,LENGTH) ) THEN
C
10 CONTINUE
C
C   IF ( XX .LT. TAB(1,IP) ) THEN
C     IP = IP - 1
C     GO TO 10
C   ELSE IF( XX .EQ. TAB(1,IP) ) THEN
C     DO 20 I = 1 , NYY
C       YY(I) = TAB(I+1,IP)
20 CONTINUE
C   ELSE IF( XX .GT. TAB(1,IP+1) ) THEN
C     IP = IP + 1
C     GO TO 10
C   ELSE
C     DENOM = TAB(1,IP+1) - TAB(1,IP)
C     F1 = ( TAB(1,IP+1) - XX ) / DENOM
C     F2 = ( XX - TAB(1,IP) ) / DENOM
C     DO 30 I = 1 , NYY
C       YY(I) = F1 * TAB(I+1,IP) + F2 * TAB(I+1,IP+1)
30 CONTINUE
C   END IF
C
C ELSE IF( XX .LT. TAB(1,1) ) THEN
C
C   DO 100 I = 1 , NYY
C     YY(I) = TAB(I+1,1)
100 CONTINUE
C
C ELSE
C
C   DO 200 I = 1 , NYY
C     YY(I) = TAB(I+1,LENGTH)
200 CONTINUE
C
C END IF
C
C RETURN
C END

```

```

SUBROUTINE RDCARD( IN ,IOUT ,NWS ,PROGRAM )
C
C READ INPUT DATA UNIT(IN) AND KEEP THEM IN UNIT(IOUT)
C   IN  --- UNIT NO. FOR INPUT DATA
C   IOUT --- UNIT NO. FOR OUTPUT DATA
C   NWS --- UNIT NO. FOR PRINT
C   PROGRAM--- PROGRAM NAME
C
C IMODE --- FLAG ( PROCESS MODE )
C   0 --- STANDARD MODE
C   1 --- CHARACTER STRING MODE
C   2 --- COMMENT MODE
C
C ICMODE--- FLAG ( PROCESS MODE )
C   0 --- OTHER MODE
C   1 --- AFTER CHARACTER STRING MODE
C
C ICOMME--- FLAG ( CARD TYPE )
C   0 --- DATA CARD
C   1 --- COMMENT CARD '*,.....'
C           OR '#,.....'
C   2 --- INCLUDE CARD '@INCLUDE DSN, MEMBER, OPT '
C   3 --- LISTON CARD '@LISTON '
C   4 --- LISTOFF CARD '@LISTOFF '
C   5 --- EJECT CARD '@EJECT '
C
C IPRT --- FLAG ( PRINT INCLUDE DATA )
C   1 --- PRINT ON
C   OTHER - PRINT OFF
C
C ILIST --- FLAG ( LIST OF MAIN DATA )
C   1 --- LIST ON
C   OTHER - LIST OFF
C
C ILISTM--- FLAG ( LIST OF INCLUDE DATA )
C   1 --- LIST ON
C   OTHER - LIST OFF
C
C PARAMETER ( NLINE=500 ,NDATA=80 ,NCDEND=72 )
C
C NLINE --- CONSTANT 1'500' FOR PAGE CONTROL
C NDATA --- RECORD LENGTH OF INPUT DATA( 80 BYTES )
C NCDEND--- EFFECTIVE RECORD LENGTH OF INPUT DATA( 72 BYTES )
C
C COMMON /DAY/CDATE
C LOGICAL ERROR
C CHARACTER PROGMM*8 ,CDATE *9 ,
* DSN *44 ,MEMBER*8 ,
* OPTION*8 ,OPTTAB*8 ,
* NOTICE*10 ,ERRMSG(0:9)*20 ,
* LETTER*(NDATA) ,
* DATACD*(NCDEND) ,
* DATA2*(NCDEND)
C
C-----
C CONSTANT DATA SECTION
C
C DATA OPTTAB/'P '/'
C DATA ERRMSG/'
* 'OPERAND MISSING '
* 'DSN OPERAND ERROR '
* 'TOO LONG DS-NAME '
* 'MEMBER OPERAND ERROR'
* 'TOO LONG MEMBER NAME'
* 'OPTION OPERAND ERROR'
* 'TOO LONG OPTION NAME'
* 'UNDEFINED OPTION '
* 'INCLUDE CARD DELETED' /
C
C-----
C
C ERROR = .FALSE.
C NCTOT = 0
C NCEFF = 0
C NTCD = 0
C ICOMME = 0
C MERR = 0
C IMODE = 0
C ICMODE = 0
C ILIST = 1
C
C CALL DATE(CDATE)
C REWIND IOUT
C
1000 CONTINUE
C
C NCAFD = 0
C
C HEADDING

```

```

WRITE(MM6,90010) PRGNUM ,CDATE
C
C      READ ONE CARD
2000 CONTINUE
C
      READ ( IN ,90040 ,END=3000 ) LETTER
      NCTOT = NCTOT + 1
      NTCO  = NTCO  + 1
      MERR  = 0
C
      IF ( IMODE .EQ. 1 .OR. IMODE .EQ. 2 )
*      THEN
          ICOMME = 0
C
      ELSE
C
C      IF (
*      LETTER(1:1) .EQ. '*'
*      .OR. LETTER(1:1) .EQ. '#' )
*      THEN
          ..COMMENT CARD
          ICOMME = 1
C
      ELSE IF ( LETTER(1:9) .EQ. '@INCLUDE ' )
*      THEN
          ..INCLUDE CARD
          IPRT = 0
          MEFR = 0
          ICOMME = 2
          DO 2010 M = 1 , 44
              DSN(M:M) = ' '
2010      CONTINUE
          DO 2020 M = 1 , 8
              MEMBER(M:M) = ' '
2020      CONTINUE
          DO 2030 M = 1 , 8
              OPTION(M:M) = ' '
2030      CONTINUE
C
          DO 2110 M = 10 , NCDEND
              MDSN = M
              IF ( LETTER(M:M) .NE. ' ' ) GO TO 2120
2110      CONTINUE
              MEFR = 1
              GO TO 2490
C
2120      CONTINUE
          IF ( LETTER(MDSN:MDSN) .EQ. ' , ' )
*      THEN
              MEFR = 2
              GO TO 2490
          ELSE
              CONTINUE
          END IF
C
C      ..DSN NAME CHECK
          K = 0
          DO 2130 M = MDSN,NCDEND
              MLOC = M
              IF (
*      LETTER(M:M) .EQ. ' '
*      .OR. LETTER(M:M) .EQ. ' , ' ) GO TO 2140
              K = K + 1
              IF ( K .LE. 44 )
*      THEN
                  DSN(K:K) = LETTER(M:M)
              ELSE
                  MEFR = 3
                  GO TO 2490
              END IF
C
2130      CONTINUE
2140      CONTINUE
C
          IF ( MLOC .GE. NCDEND ) GO TO 2490
          IF ( LETTER(MLOC:MLOC) .EQ. ' ' ) GO TO 2490
C
C      ..MEMBER NAME CHECK
          MMEMB = MLOC + 1
          IF ( MMEMB .GT. NCDEND )
*      THEN
              MEFR = 4
              GO TO 2490
          ELSE
              CONTINUE
          END IF
          IF ( LETTER(MMEMB:MMEMB) .EQ. ' ' )
*      THEN
              MEFR = 4

```

```

GO TO 2490
ELSE
CONTINUE
END IF
C
K = 0
DO 2210 M = MMEMB,NCDEND
    MLOC = M
    IF (
*      LETTER(M:M) .EQ. ' '
*      .OR. LETTER(M:M) .EQ. ' , ' ) GO TO 2220
    K = K + 1
    IF ( K .LE. 8 )
*      THEN
        MEMBER(K:K) = LETTER(M:M)
    ELSE
        MEFR = 5
        GO TO 2490
    END IF
C
2210      CONTINUE
2220      CONTINUE
C
    IF ( MLOC .GE. NCDEND ) GO TO 2490
    IF ( LETTER(MLOC:MLOC) .EQ. ' ' ) GO TO 2490
C
C      ..OPTION CHECK
    MOPT = MLOC + 1
    IF ( MOPT .GT. NCDEND )
*      THEN
        MEFR = 6
        GO TO 2490
    ELSE
        CONTINUE
    END IF
    IF ( LETTER(MOPT:MOPT) .EQ. ' ' )
*      THEN
        MEFR = 6
        GO TO 2490
    ELSE
        CONTINUE
    END IF
C
    K = 0
    DO 2310 M = MOPT,NCDEND
        MLOC = M
        IF (
*      LETTER(M:M) .EQ. ' ' ) GO TO 2320
        K = K + 1
        IF ( K .LE. 8 )
*      THEN
            OPTION(K:K) = LETTER(M:M)
        ELSE
            MEFR = 7
            GO TO 2490
        END IF
C
2310      CONTINUE
2320      CONTINUE
C
    IF ( OPTION .EQ. OPTTAB )
*      THEN
        IPRT = 1
    ELSE
        MEFR = 8
        GO TO 2490
    END IF
C
2490      CONTINUE
C
    ELSE IF ( LETTER(1:8) .EQ. '@LISTON ' )
*      THEN
        ..LISTON CARD
        ICOMME = 3
        ILIST = 1
        GO TO 2000
C
    ELSE IF ( LETTER(1:9) .EQ. '@LISTOFF ' )
*      THEN
        ..LISTOFF CARD
        ICOMME = 4
        ILIST = 0
        GO TO 2000
C
    ELSE IF ( LETTER(1:7) .EQ. '@EJECT ' )

```

```

*      THEN
C      ..EJECT CARD
      ICOMME = 5
C
C      ELSE
C      ..DATA CARD
      ICOMME = 0
      END IF
      END IF
C
C
C      IF ( ICOMME .EQ. 0 )
*      THEN
C
      MLOC = 0
      DO 2510 M = 1 ,NCDEND
        DATACD(M:M) = ' '
2510    CONTINUE
C
      IF ( IMODE .EQ. 1 ) GO TO 2700
      IF ( IMODE .EQ. 2 ) GO TO 2800
C
2600    CONTINUE
      ..STANDARD MODE
      IMODE = 0
C
      MM = MLOC + 1
      IF ( MM .GT. NCDEND ) GO TO 2900
      DO 2610 M = MM , NCDEND
        DATACD(M:M) = LETTER(M:M)
        IF ( LETTER(M:M) .EQ. '***')
*      THEN
*      IF ( M .EQ. 1 .OR. ICMODE .EQ. 1 )
*      THEN
          MLOC = M
          GO TO 2700
        ELSE IF ( M .GT. 1 )
*      THEN
          IF ( LETTER(M-1:M-1) .EQ. ' '
            .OR. LETTER(M-1:M-1) .EQ. ',')
*      THEN
            MLOC = M
            GO TO 2700
          ELSE
            CONTINUE
          END IF
        ELSE
          CONTINUE
        END IF
      END IF
      ELSE IF ( LETTER(M:M) .EQ. '!')
*      THEN
*      IF ( M .LT. NCDEND )
*      THEN
*      IF ( LETTER(M+1:M+1) .EQ. '**')
*      THEN
          DATACD(M:M) = ' '
          DATACD(M+1:M+1) = ' '
          MLOC = M + 1
          GO TO 2800
        ELSE
          MLOC = NCDEND
          GO TO 2900
        END IF
      ELSE
        CONTINUE
      END IF
      ELSE
        CONTINUE
      END IF
C
      ICMODE = 0
      CONTINUE
      GO TO 2900
C
C
C
2610    CONTINUE
      GO TO 2900
C
C
C
2700    CONTINUE
      .. CHARACTER STRING MODE
      IMODE = 1
      ICMODE = 0
C
      MM = MLOC + 1
      IF ( MM .GT. NCDEND ) GO TO 2900
      DO 2710 M = MM , NCDEND
        DATACD(M:M) = LETTER(M:M)
        IF ( LETTER(M:M) .EQ. '***')
*      THEN
*      MLOC = M
*      ICMODE = 1
*      GO TO 2600
      ELSE
        CONTINUE
      END IF
      CONTINUE
      GO TO 2900
C
2800    CONTINUE
      .. COMMENT MODE
      IMODE = 2
      ICMODE = 0
C
      MM = MLOC + 1
      IF ( MM .GT. NCDEND ) GO TO 2900
      DO 2810 M = MM , NCDEND
        DATACD(M:M) = ' '
        IF ( LETTER(M:M) .EQ. '**')
*      THEN
*      IF ( M .LT. NCDEND )
*      THEN
*      IF ( LETTER(M+1:M+1) .EQ. '!')
*      THEN
          DATACD(M+1:M+1) = ' '
          MLOC = M + 1
          GO TO 2600
        ELSE
          CONTINUE
        END IF
      ELSE
        CONTINUE
      END IF
      ELSE
        CONTINUE
      END IF
      CONTINUE
      GO TO 2900
C
C
C
2900    CONTINUE
      ICMODE = 0
      ..OUTPUT CARD DATA
      IF ( NCEFF .EQ. 0 ) THEN
        WRITE(1OUT,90040) DATACD
      ELSE
        IF ( DATACD(1:1) .EQ. ' ' ) THEN
          WRITE(1OUT,90040) DATACD
        ELSE
          DO 2910 M = 1 , NCDEND
            MEND = M
            IF ( DATACD(M:M) .EQ. ' ' ) THEN
              GO TO 2920
            END IF
          CONTINUE
          CONTINUE
          IF ( MEND .EQ. NCDEND ) THEN
            WRITE(1OUT,90040) DATACD
          ELSE
            DATACD2 = DATACD(MEND:NCDEND)
            DATACD = DATACD(1:MEND)
            WRITE(1OUT,90040) DATACD
            WRITE(1OUT,'(8X,13)') NCEFF+1
          END IF
          IF ( DATACD2 .NE. ' ' ) THEN
            DATACD = DATACD2
            DO 2930 M = 1 , NCDEND
              MINI = M
              IF ( DATACD(M:M) .NE. ' ' ) THEN
                GO TO 2940
              END IF
            CONTINUE
            CONTINUE
          END IF
        END IF
      END IF
      CONTINUE
      CONTINUE

```

```

IF ( DATA2(MINI:MINI) .EQ. ' ' ) THEN
WRITE(1OUT,90040) DATA2
ELSE
DO 2950 M = MINI , NCDEND
MEND = M
IF ( DATA2(M:M) .EQ. ' '
.OR. DATA2(M:M) .EQ. ', ' ) THEN
GO TO 2960
END IF
2950 CONTINUE
2960 CONTINUE
IF ( DATA2(MEND:MEND) .EQ. ', ' ) THEN
DATA2(MEND:MEND) = ' '
END IF
IF ( MEND .EQ. NCDEND ) THEN
WRITE(1OUT,90040) DATA2
ELSE
DATA2 = DATA2(MEND+1:NCDEND)
DATA2 = DATA2(1:MEND)
WRITE(1OUT,90040) DATA2
C
DO 2970 M = 1 , NCDEND
MEND = M
IF ( DATA2(M:M) .NE. ' ' ) THEN
GO TO 2980
END IF
2970 CONTINUE
2980 CONTINUE
IF ( DATA2(MEND:MEND) .EQ. ', ' ) THEN
DATA2(MEND:MEND) = ' '
END IF
IF ( DATA2 .NE. ' ' ) THEN
WRITE(1OUT,90040) DATA2
END IF
C
END IF
C
END IF
C
END IF
C
END IF
NCEFF = NCEFF + 1
C
ELSE
CONTINUE
C
C
END IF
C
IF ( ILIST .NE. 1
* .AND. MERR .EQ. 0 ) GO TO 2000
C
C
C
IF ( ICOMME .EQ. 5 )
* THEN
IF ( NCARDS .EQ. 0 )
* THEN
GO TO 2000
ELSE
..END LINE
WRITE(NW6,90030) ( I , I = 1 , 8 )
END IF
GO TO 1000
C
ELSE
CONTINUE
END IF
C
IF ( NCARDS .EQ. 0 ) THEN
WRITE(NW6,90020) ( I , I = 1 , 8 )
END IF
NCARDS = NCARDS + 1
C
IF ( MERR .EQ. 0 )
* THEN
NOTICE = ' '
ELSE
NOTICE = ' **ERROR** '
ERROR = .TRUE.
END IF
C
IF ( ICOMME .EQ. 0 ) THEN
WRITE(NW6,90110) NCEFF ,LETTER,NTCD
END IF
IF ( ICOMME .EQ. 1 ) THEN
WRITE(NW6,90120) LETTER,NTCD
END IF
IF ( ICOMME .EQ. 2 ) THEN
WRITE(NW6,90130) NOTICE,LETTER,NTCD,EPRMSG(MERR)
END IF
IF ( NCARDS .LT. NLINE ) GO TO 2000
C
END LINE
WRITE(NW6,90030) ( I , I = 1 , 8 )
GO TO 1000
C
END OF PROCESS
3000 CONTINUE
IF ( NCARDS .GT. 0 ) WRITE(NW6,90030) ( I , I = 1 , 8 )
WRITE(NW6,91010) NCTOT ,NCEFF
C
IF ( ERROR )
* THEN
WRITE(NW6,99010)
STOP 20
ELSE
WRITE(NW6,91020)
END IF
C
REWIND 1OUT
RETURN
C
C-----
C
90010 FORMAT(1H1/1H /1H ,T10,'-',A8,'-INPUT DATA LIST',T100,'DATE ',A9)
90020 FORMAT(1H0, T16,' ',8I10/
* 1H ,T5,'DATA NO.',T16,' 0',8('---5---0'),' CARD SEQ. ')
90030 FORMAT(1H , T16,' 0',8('---5---0')/
* 1H , T16,' ',8I10 )
90040 FORMAT(A)
C
90110 FORMAT(1H ,110, T16,' :',A,' ', ,18.7)
90120 FORMAT(1H , T16,' C',A,' ', ,18.7)
90130 FORMAT(1H ,A10, T16,' >',A,' ', ,18.7,' ',A20)
C
91010 FORMAT(1H0,T10,'TOTAL NUMBER OF INPUT CARDS ',110/
* 1H ,T10,'TOTAL NUMBER OF EFFECTIVE INPUT CARDS',110)
91020 FORMAT(1H1)
99010 FORMAT(1H , '***** ERROR DETECTED AT READ CARDS ROUTINE *****' )
END

```

```

SUBROUTINE REACT(IReact,MXCOMP,JMIC ,NCOMP ,NCHEM ,MXREAC,
*          T ,TO ,RCON ,
*          X ,XN ,
*          FCHEM ,
*          ECHEM ,
*          REAC ,DRDX ,
*          RXHYD ,DRYH ,DRXHYD,DRYNO2)
C
C
C
C MAKE CHEMICAL REACTION TERM
C IReact = 0 --- BOTH PHASE (MIXER)
C          = 1 --- AQUEOUS PHASE (SETTLER)
C          = 2 --- ORGANIC PHASE (SETTLER)
C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C DIMENSION X (MXCOMP,2)
C DIMENSION XN (MXCOMP,2)
C
C DIMENSION FCHEM (MXREAC,2)
C DIMENSION ECHEM (MXREAC,2)
C
C DIMENSION REAC (NCHEM,2) ,
*          DRDX(NCHEM,NCHEM,2)
C
C COMMON /IXCOMP/NCHEMO,
*          IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
*          IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
*          IONP4 , IONP5 , IONP6 , IOZR , IOTC4 ,
*          IOTC5 , IOTC6 , IOTC7 , IOTC6 , IOTC4 ,
*          IOTCZR, IOSR , IOJDI, IOJUTR, IOJUNI,
*          IOJUMO, IOCS , IOCE , IOGD , IOAM ,
*          IOCM
C
C DATA CMIN/60.0D0/
C
C IF ( JMIC .LE. 0 ) THEN
C
C XH = XN( IOHNO3,1)
C XU6 = XN( IOU6 ,1)
C XPU4 = XN( IOPU4 ,1)
C XPU3 = XN( IOPU3 ,1)
C XU4 = XN( IOU4 ,1)
C XHAN = XN( IOHAN ,1)
C XHYD = XN( IOHYD ,1)
C XNO2 = XN( IOHNO2,1)
C XPU5 = XN( IOPU5 ,1)
C XPU6 = XN( IOPU6 ,1)
C XTC4 = XN( IOTC4 ,1)
C XTC5 = XN( IOTC5 ,1)
C XTC6 = XN( IOTC6 ,1)
C XTC7 = XN( IOTC7 ,1)
C XNP4 = XN( IONP4 ,1)
C XNP5 = XN( IONP5 ,1)
C XNP6 = XN( IONP6 ,1)
C
C XH = DMAX1( XH , 0.0D0 )
C XU6 = DMAX1( XU6 , 0.0D0 )
C XPU4 = DMAX1( XPU4 , 0.0D0 )
C XPU3 = DMAX1( XPU3 , 0.0D0 )
C XU4 = DMAX1( XU4 , 0.0D0 )
C XHAN = DMAX1( XHAN , 0.0D0 )
C XHYD = DMAX1( XHYD , 0.0D0 )
C XNO2 = DMAX1( XNO2 , 0.0D0 )
C XPU5 = DMAX1( XPU5 , 0.0D0 )
C XPU6 = DMAX1( XPU6 , 0.0D0 )
C XTC4 = DMAX1( XTC4 , 0.0D0 )
C XTC5 = DMAX1( XTC5 , 0.0D0 )
C XTC6 = DMAX1( XTC6 , 0.0D0 )
C XTC7 = DMAX1( XTC7 , 0.0D0 )
C XNP4 = DMAX1( XNP4 , 0.0D0 )
C XNP5 = DMAX1( XNP5 , 0.0D0 )
C XNP6 = DMAX1( XNP6 , 0.0D0 )
C
C YH = XN( IOHNO3,2)
C YU6 = XN( IOU6 ,2)
C YPU4 = XN( IOPU4 ,2)
C YPU3 = XN( IOPU3 ,2)
C YU4 = XN( IOU4 ,2)
C YHAN = XN( IOHAN ,2)
C YHYD = XN( IOHYD ,2)

```

```

YNO2 = XN( IOHNO2,2)
YPU5 = XN( IOPU5 ,2)
YPU6 = XN( IOPU6 ,2)
YTC4 = XN( IOTC4 ,2)
YTC5 = XN( IOTC5 ,2)
YTC6 = XN( IOTC6 ,2)
YTC7 = XN( IOTC7 ,2)
YNP4 = XN( IONP4 ,2)
YNP5 = XN( IONP5 ,2)
YNP6 = XN( IONP6 ,2)
C
C YH = DMAX1( YH , 0.0D0 )
C YU6 = DMAX1( YU6 , 0.0D0 )
C YPU4 = DMAX1( YPU4 , 0.0D0 )
C YPU3 = DMAX1( YPU3 , 0.0D0 )
C YU4 = DMAX1( YU4 , 0.0D0 )
C YHAN = DMAX1( YHAN , 0.0D0 )
C YHYD = DMAX1( YHYD , 0.0D0 )
C YNO2 = DMAX1( YNO2 , 0.0D0 )
C YPU5 = DMAX1( YPU5 , 0.0D0 )
C YPU6 = DMAX1( YPU6 , 0.0D0 )
C YTC4 = DMAX1( YTC4 , 0.0D0 )
C YTC5 = DMAX1( YTC5 , 0.0D0 )
C YTC6 = DMAX1( YTC6 , 0.0D0 )
C YTC7 = DMAX1( YTC7 , 0.0D0 )
C YNP4 = DMAX1( YNP4 , 0.0D0 )
C YNP5 = DMAX1( YNP5 , 0.0D0 )
C YNP6 = DMAX1( YNP6 , 0.0D0 )
C
C IF ( IReact .EQ. 0 ) THEN
C
C CALL REACTX( IReact,NCHEM ,T ,TO ,RCON ,
*          MXCOMP,REAC (1,1) ,DRDX(1,1,1) ,
*          MXREAC,FCHEM(1,1) ,ECHEM(1,1) ,
*          XH ,XU6 ,XPU4 ,XPU3 ,XU4 ,XHAN ,
*          XHYD ,XNO2 ,XPU5 ,XPU6 ,
*          XTC4 ,XTC5 ,XTC6 ,XTC7 ,
*          XNP4 ,XNP5 ,XNP6 ,X(1,1))
C
C CALL REACTY( IReact,NCHEM ,T ,TO ,RCON ,
*          MXCOMP,REAC (1,2) ,DRDX(1,1,2) ,
*          MXREAC,FCHEM(1,2) ,ECHEM(1,2) ,
*          YH ,YU6 ,YPU4 ,YPU3 ,YU4 ,YHAN ,
*          YHYD ,YNO2 ,YPU5 ,YPU6 ,
*          YTC4 ,YTC5 ,YTC6 ,YTC7 ,
*          YNP4 ,YNP5 ,YNP6 ,X(1,2),
*          XHYD ,RXHYD ,DRYH ,DRXHYD,DRYNO2)
C
C DO 120 J = 1 , NCHEM
C
C REAC(J,1) = REAC(J,1)*CMIN
C REAC(J,2) = REAC(J,2)*CMIN
C
C DO 110 K = 1 , NCHEM
C DRDX(J,K,1) = DRDX(J,K,1)*CMIN
C DRDX(J,K,2) = DRDX(J,K,2)*CMIN
110 CONTINUE
C
C 120 CONTINUE
C
C RXHYD = RXHYD *CMIN
C
C DRYH = DRYH *CMIN
C DRXHYD = DRXHYD*CMIN
C DRYNO2 = DRYNO2*CMIN
C
C ELSE IF( IReact .EQ. 1 ) THEN
C
C CALL REACTX( IReact,NCHEM ,T ,TO ,RCON ,
*          MXCOMP,REAC (1,1) ,DRDX(1,1,1) ,
*          MXREAC,FCHEM(1,1) ,ECHEM(1,1) ,
*          XH ,XU6 ,XPU4 ,XPU3 ,XU4 ,XHAN ,
*          XHYD ,XNO2 ,XPU5 ,XPU6 ,
*          XTC4 ,XTC5 ,XTC6 ,XTC7 ,
*          XNP4 ,XNP5 ,XNP6 ,X(1,1))
C
C DO 220 J = 1 , NCHEM
C
C REAC(J,1) = REAC(J,1)*CMIN
C
C DO 210 K = 1 , NCHEM
C DRDX(J,K,1) = DRDX(J,K,1)*CMIN
210 CONTINUE
C

```

```

220      CONTINUE
C
C
C      ELSE
C
C      CALL REACTX( IREACT,NCHEM ,T ,TO ,RCON ,
*
*      MXCOMP,REAC (1,2) ,DRDX(1,1,2) ,
*      MXREAC,FCHEM(1,2) ,ECHEM(1,2) ,
*      YH ,YU6 ,YPU4 ,YPU3 ,YU4 ,YHAN ,
*      YHYD ,YNQ2 ,YPU5 ,YPU6 ,
*      YTC4 ,YTC5 ,YTC6 ,YTC7 ,
*      YNP4 ,YNP5 ,YNP6 ,X(1,2),
*      XHYD ,RXHYD ,DRYH ,DRXHYD,DRYNO2)
C
C
C      DO 320 J = 1 , NCHEM
C
C      REAC(J,2) = REAC(J,2)*CMIN
C
C      DO 310 K = 1 , NCHEM
C      DRDX(J,K,2) = DRDX(J,K,2)*CMIN
310      CONTINUE
C
320      CONTINUE
C
C      RXHYD = RXHYD *CMIN
C
C      DRYH = DRYH *CMIN
C      DRXHYD = DRXHYD*CMIN
C      DRYNO2 = DRYNO2*CMIN
C
C      END IF
C
C
C      ELSE
C
C      DO 420 J = 1 , NCOMP
C
C      REAC(J,1) = 0.000
C      REAC(J,2) = 0.000
C
C      DO 410 K = 1 , NCOMP
C      DRDX(J,K,1) = 0.000
C      DRDX(J,K,2) = 0.000
410      CONTINUE
C
420      CONTINUE
C
C
C      END IF
C
C
C
C
C      RETURN
C      END

```

```

SUBROUTINE REACTX(IREACT,NCHEM ,T ,TO ,RCON ,
*
*      MXCOMP,REAC ,DRDX ,
*      MXREAC,FCHEM ,ECHEM ,
*      XH ,XU6 ,XPU4 ,XPU3 ,XU4 ,XHAN ,
*      XHYD ,XNO2 ,XPU5 ,XPU6 ,
*      XTC4 ,XTC5 ,XTC6 ,XTC7 ,
*      XNP4 ,XNP5 ,XNP6 ,X )
C
C
C      JACOBIAN CALCULATION OF CHEMICAL REACTION TERM IN AQUEOUS PHASE
C
C
C      IMPLICIT REAL*8 ( A-H , O-Z )
C
C      DIMENSION FCHEM (MXREAC)
C      DIMENSION ECHEM (MXREAC)
C
C      DIMENSION X (MXCOMP)
C
C      DIMENSION REAC(NCHEM) , DRDX(NCHEM,NCHEM)
C
C      COMMON /IXCOMP/NCHEMO,
*      IOHNO3, IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*      IOHNO2, IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*      IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*      IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6, IOTCP4,
*      IOTCZR, IOZR ,IORUDI, IORJTR, IORUNI,
*      IORUMO, IOCS ,IOCE ,IOGD ,IOAM ,
*      IOCM
C
C
C      DATA CK1 / 1.5000D+02 /
C      DATA CK201 / 6.5000D+00 /
C      DATA CK201 / 1.2910D+01 /
C      DATA CK202 / 4.3000D+00 /
C      DATA CK202 / 4.7380D+00 /
C      DATA CK3 / 0.0380D+00 /
C      DATA CK3A / 0.3500D+00 /
C      DATA CK401 / 5.1000D-03 /,
*      CK402 / 1.3000D+00 /,
*      CK403 / 0.5400D+00 /,
*      CK404 / 0.4400D+00 /,
*      CK405 / 0.7600D+00 /
*      CK406 / 5.5000D-02 /
C      DATA CK5 / 3.7000D+04 /
C
C
C      DATA CK6 / 2.5000D+00 /
C      DATA CK7 / 70.000D+00 /
C      DATA CK8 / 8.3000D+00 /
C      DATA CK9 / 7.0000D+00 /
C      DATA CK10 / 2.2500D+00 /
C      DATA CK11 / 92.100D+00 /
C      DATA CK12 / 250.00D+00 /
C      DATA CK13 / 37.000D+00 /
C      DATA CK14 / 7.9000D+00 /
C      DATA CK15 / 276.00D+00 /
C      DATA CK16 / 7.1500D+00 /
C      DATA CK17 / 1.0000D-05 /
C      DATA CK18 / 2.0000D-04 /
C      DATA CK19 / 8.0000D-04 /
C      DATA CK20 / 9.0000D-04 /
C      DATA CK21 / 4.8000D-04 /
C      DATA CK22 / 1.0000D+00 /
C      DATA CK23 / 0.0598D+00 /
C      DATA CK24 / 0.3140D+00 /
C      DATA CK25 / 1.6500D-03 /
C      DATA CK26 / 0.3450D+00 /
C      DATA CK27 / 320.00D+00 /
C      DATA CK27 / 438.50D+00 /
C      DATA CK28 / 0.1530D+00 /
C      DATA CK29 / 0.0664D+00 /
C      DATA CK30 / 372.00D+00 /
C      DATA CK31 / 6.6000D-05 /
C      DATA CK32 / 2.5000D+00 /
C      DATA CK33 / 2.5000D+00 /
C      DATA CK34 / 0.1000D+00 /
C      DATA CK35 / 3.3000D-02 /
C      DATA CK36 / 6.6000D-03 /
C      DATA CK37 / 3.3000D+01 /
C      DATA CK38 / 7.8200D-04 /
C      DATA CK39 / 1.3900D+00 /
C      DATA CK40 / 1.0000D-07 /
C
C
C      DATA CK411 / 2.5000D-02 /,

```

```

* CK412 / 1.3000D-02 /
DATA CK42 / 2.5000D-04 /
DATA CK43 / 5.8000D-04 /
DATA CK44 / 1.8000D+01 /
C DATA CK44 / 0.0D0 /
DATA CK45 / 100.00D+00 /

C
C
C DATA EK1 / 104.0D3 /
C DATA EK2 / 130.0D3 /
C DATA EK201 / 183.7D3 /
DATA EK202 / 22.2D3 /
DATA EK3 / 93.0D3 /
C DATA EK4M / 35.0D3 /
C DATA EK4S / 60.7D3 /
C DATA EK5 / 36.0D3 /
DATA EK6 / 103.0D3 /
DATA EK7 / 145.0D3 /
DATA EK8 / 70.3D3 /
DATA EK10 / 40.2D3 /

C
DATA EK11 / 82.0D3 /
DATA EK12 / 59.0D3 /
DATA EK13 / 43.0D3 /
DATA EK14 / 111.0D3 /
DATA EK15 / 23.0D3 /
DATA EK16 / 92.0D3 /

C
DATA EK19 / 68.7D3 /
DATA EK20 / 62.8D3 /
DATA EK21 / 105.0D3 /
DATA EK22 / 17.2D3 /
DATA EK23 / 85.0D3 /
DATA EK24 / 67.4D3 /
DATA EK25 / 105.0D3 /
DATA EK26 / 82.0D3 /
C DATA EK27 / 48.1D3 /
DATA EK27 / 51.9D3 /
DATA EK28 / 93.0D3 /
DATA EK29 / 73.0D3 /
DATA EK30 / 80.0D3 /
DATA EK31 / 50.0D3 /
DATA EK32 / 50.0D3 /
DATA EK33 / 50.0D3 /
DATA EK34 / 50.0D3 /
DATA EK35 / 50.0D3 /
DATA EK36 / 50.0D3 /
DATA EK37 / 50.0D3 /
DATA EK38 / 29.7D3 /
DATA EK39 / 76.6D3 /
DATA EK40 / 50.0D3 /

C
C
C DATA EPSX / 1.0000D-10 /

C... PREPARATION
C
DO 110 J = 1 , NCHEM
REAC(J) = 0.0D0
110 CONTINUE
C
DO 130 K = 1 , NCHEM
DO 120 J = 1 , NCHEM
DRDX(J,K) = 0.0D0
120 CONTINUE
130 CONTINUE

C
C
C
C XCN = XH + XHYD + XHAN + XNP5 + XPU5
* + 2.0D0*(XU6 + XPU6 + XNP6 )
* + 3.0D0*(XPU3 )
* + 4.0D0*(XU4 + XPU4 + XNP4 )

C
DCXH = 1.0D0
DCXU6 = 2.0D0
DCXPU4 = 4.0D0
DCXPU3 = 3.0D0
DCXU4 = 4.0D0
DCXHYD = 1.0D0
DCXHAN = 1.0D0
DCXPU5 = 1.0D0
DCXPU6 = 2.0D0
DCXNP4 = 4.0D0
DCXNP5 = 1.0D0
DCXNP6 = 2.0D0
    
```

```

C
ZH = DMAX1( XH , EPSX )
ZPU3 = DMAX1( XPU3 , EPSX )
ZCN = DMAX1( XCN , EPSX )

C
TAU = 1.0D0/T - 1.0D0/( T0 + 25.0D0 )

C
DO 210 L = 1 , MXREAC
ECHEM(L) = 0.0D0
210 CONTINUE
C
ECHEM( 1 ) = EK1
C
ECHEM( 2 ) = EK2
ECHEM( 2 ) = EK201
ECHEM( 3 ) = EK3

C
ECHEM( 6 ) = EK6
ECHEM( 7 ) = EK7
ECHEM( 8 ) = EK8

C
ECHEM(10) = EK10
ECHEM(11) = EK11
ECHEM(12) = EK12
ECHEM(13) = EK13
ECHEM(14) = EK14
ECHEM(15) = EK15
ECHEM(16) = EK16
ECHEM(17) = 0.0D0
ECHEM(18) = 0.0D0
ECHEM(19) = EK19
ECHEM(20) = EK20
ECHEM(21) = EK21
ECHEM(22) = EK22
ECHEM(23) = EK23
ECHEM(24) = EK24
ECHEM(25) = EK25
ECHEM(26) = EK26
ECHEM(27) = EK27
ECHEM(28) = EK28
ECHEM(29) = EK29
ECHEM(30) = EK30
ECHEM(31) = EK31
ECHEM(32) = EK32
ECHEM(33) = EK33
ECHEM(34) = EK34
ECHEM(35) = EK35
ECHEM(36) = EK36
ECHEM(37) = EK37
ECHEM(38) = EK38
ECHEM(39) = EK39
ECHEM(40) = EK40

C
DO 220 L = 1 , MXREAC
ECHEM(L) = DEXP( - ECHEM(L)/RCON*TAU )
220 CONTINUE
C
EB2 = DEXP( - EK202/RCON*TAU )

C
C... (1) REDUCTION OF PU(IV) BY URANOUS ( FROM R.MIXSET )
C
CK = ECHEM(1)*FCHEM(1)*CK1

C
RR = CK*XU4*XPU4/ZH/ZH
DXH = -2.0D0*CK*XU4*XPU4/ZH/ZH
DXPU4 = CK*XU4 /ZH/ZH
DXU4 = CK *XPU4/ZH/ZH

C
C
C
REAC(10HN03) = REAC(10HN03) + 2.0D0*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 2.0D0*DXH
DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 2.0D0*DXPU4
DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 2.0D0*DXU4

C
REAC(10U6) = REAC(10U6) + 0.5D0*RR
DRDX(10U6,10HN03) = DRDX(10U6,10HN03) + 0.5D0*DXH
DRDX(10U6,10PU4) = DRDX(10U6,10PU4) + 0.5D0*DXPU4
DRDX(10U6,10U4) = DRDX(10U6,10U4) + 0.5D0*DXU4

C
REAC(10PU4) = REAC(10PU4) - RR
DRDX(10PU4,10HN03) = DRDX(10PU4,10HN03) - DXH
DRDX(10PU4,10PU4) = DRDX(10PU4,10PU4) - DXPU4
DRDX(10PU4,10U4) = DRDX(10PU4,10U4) - DXU4

C
REAC(10PU3) = REAC(10PU3) + RR
DRDX(10PU3,10HN03) = DRDX(10PU3,10HN03) + DXH
DRDX(10PU3,10PU4) = DRDX(10PU3,10PU4) + DXPU4
DRDX(10PU3,10U4) = DRDX(10PU3,10U4) + DXU4
    
```

```

C
  REAC(10U4 ) = REAC(10U4 ) - 0.50*RR
  DRDX(10U4 ,10HN3) = DRDX(10U4 ,10HN3) - 0.50*DXH
  DRDX(10U4 ,10PU4 ) = DRDX(10U4 ,10PU4 ) - 0.50*DXPU4
  DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - 0.50*DXU4

C
C... (2) REDUCTION OF PU(IV) BY HAN ( FROM R.MIXSET )
C
  CK = ECHEM(2)*FCHEM(2)*CK201
  CB = EB2*CK202

C
  XPU42 = XPU4*XPU4
  XHAN2 = XHAN*XHAN
  ZPU32 = 1.00/( ZPU3*ZPU3 )
  XH4 = 1.00/( ZH*ZH*ZH*ZH )
  AA = 1.00/( 1.000 + CB*XCN )

C
  RR = CK*XPU42*XHAN2*ZPU32*XH4*AA*AA
  DXH = - 4.000*CK*XPU42*XHAN2*ZPU32*XH4*AA*AA/ZH
  DXPU4 = 2.000*CK*XPU4 *XHAN2*ZPU32*XH4*AA*AA
  DXPU3 = - 2.000*CK*XPU42*XHAN2*ZPU32*XH4*AA*AA/ZPU3
  DXHAN = 2.000*CK*XPU42*XHAN *ZPU32*XH4*AA*AA
  DCN = - CB*2.000*CK*XPU42*XHAN2*ZPU32*XH4*AA*AA*AA

C
  REAC(10HN3) = REAC(10HN3) + 2.000*RR
  DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) + 2.000*DXH
  DRDX(10HN3,10PU4 ) = DRDX(10HN3,10PU4 ) + 2.000*DXPU4
  DRDX(10HN3,10PU3 ) = DRDX(10HN3,10PU3 ) + 2.000*DXPU3
  DRDX(10HN3,10HAN ) = DRDX(10HN3,10HAN ) + 2.000*DXHAN

C
  DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) + 2.000*DCN*DCXH
  DRDX(10HN3,10U6 ) = DRDX(10HN3,10U6 ) + 2.000*DCN*DCXU6
  DRDX(10HN3,10PU4 ) = DRDX(10HN3,10PU4 ) + 2.000*DCN*DCXPU4
  DRDX(10HN3,10PU3 ) = DRDX(10HN3,10PU3 ) + 2.000*DCN*DCXPU3
  DRDX(10HN3,10U4 ) = DRDX(10HN3,10U4 ) + 2.000*DCN*DCXU4
  DRDX(10HN3,10HYD ) = DRDX(10HN3,10HYD ) + 2.000*DCN*DCXHYD
  DRDX(10HN3,10HAN ) = DRDX(10HN3,10HAN ) + 2.000*DCN*DCXHAN
  DRDX(10HN3,10PU5 ) = DRDX(10HN3,10PU5 ) + 2.000*DCN*DCXPU5
  DRDX(10HN3,10PU6 ) = DRDX(10HN3,10PU6 ) + 2.000*DCN*DCXPU6
  DRDX(10HN3,10NP4 ) = DRDX(10HN3,10NP4 ) + 2.000*DCN*DCXNP4
  DRDX(10HN3,10NP5 ) = DRDX(10HN3,10NP5 ) + 2.000*DCN*DCXNP5
  DRDX(10HN3,10NP6 ) = DRDX(10HN3,10NP6 ) + 2.000*DCN*DCXNP6

C
  REAC(10PU4 ) = REAC(10PU4 ) - RR
  DRDX(10PU4 ,10HN3) = DRDX(10PU4 ,10HN3) - DXH
  DRDX(10PU4 ,10PU4 ) = DRDX(10PU4 ,10PU4 ) - DXPU4
  DRDX(10PU4 ,10PU3 ) = DRDX(10PU4 ,10PU3 ) - DXPU3
  DRDX(10PU4 ,10HAN ) = DRDX(10PU4 ,10HAN ) - DXHAN

C
  DRDX(10PU4 ,10HN3) = DRDX(10PU4 ,10HN3) - DCN*DCXH
  DRDX(10PU4 ,10U6 ) = DRDX(10PU4 ,10U6 ) - DCN*DCXU6
  DRDX(10PU4 ,10PU4 ) = DRDX(10PU4 ,10PU4 ) - DCN*DCXPU4
  DRDX(10PU4 ,10PU3 ) = DRDX(10PU4 ,10PU3 ) - DCN*DCXPU3
  DRDX(10PU4 ,10U4 ) = DRDX(10PU4 ,10U4 ) - DCN*DCXU4
  DRDX(10PU4 ,10HYD ) = DRDX(10PU4 ,10HYD ) - DCN*DCXHYD
  DRDX(10PU4 ,10HAN ) = DRDX(10PU4 ,10HAN ) - DCN*DCXHAN
  DRDX(10PU4 ,10PU5 ) = DRDX(10PU4 ,10PU5 ) - DCN*DCXPU5
  DRDX(10PU4 ,10PU6 ) = DRDX(10PU4 ,10PU6 ) - DCN*DCXPU6
  DRDX(10PU4 ,10NP4 ) = DRDX(10PU4 ,10NP4 ) - DCN*DCXNP4
  DRDX(10PU4 ,10NP5 ) = DRDX(10PU4 ,10NP5 ) - DCN*DCXNP5
  DRDX(10PU4 ,10NP6 ) = DRDX(10PU4 ,10NP6 ) - DCN*DCXNP6

C
  REAC(10PU3 ) = REAC(10PU3 ) + RR
  DRDX(10PU3 ,10HN3) = DRDX(10PU3 ,10HN3) + DXH
  DRDX(10PU3 ,10PU4 ) = DRDX(10PU3 ,10PU4 ) + DXPU4
  DRDX(10PU3 ,10PU3 ) = DRDX(10PU3 ,10PU3 ) + DXPU3
  DRDX(10PU3 ,10HAN ) = DRDX(10PU3 ,10HAN ) + DXHAN

C
  DRDX(10PU3 ,10HN3) = DRDX(10PU3 ,10HN3) + DCN*DCXH
  DRDX(10PU3 ,10U6 ) = DRDX(10PU3 ,10U6 ) + DCN*DCXU6
  DRDX(10PU3 ,10PU4 ) = DRDX(10PU3 ,10PU4 ) + DCN*DCXPU4
  DRDX(10PU3 ,10PU3 ) = DRDX(10PU3 ,10PU3 ) + DCN*DCXPU3
  DRDX(10PU3 ,10U4 ) = DRDX(10PU3 ,10U4 ) + DCN*DCXU4
  DRDX(10PU3 ,10HYD ) = DRDX(10PU3 ,10HYD ) + DCN*DCXHYD
  DRDX(10PU3 ,10HAN ) = DRDX(10PU3 ,10HAN ) + DCN*DCXHAN
  DRDX(10PU3 ,10PU5 ) = DRDX(10PU3 ,10PU5 ) + DCN*DCXPU5
  DRDX(10PU3 ,10PU6 ) = DRDX(10PU3 ,10PU6 ) + DCN*DCXPU6
  DRDX(10PU3 ,10NP4 ) = DRDX(10PU3 ,10NP4 ) + DCN*DCXNP4
  DRDX(10PU3 ,10NP5 ) = DRDX(10PU3 ,10NP5 ) + DCN*DCXNP5
  DRDX(10PU3 ,10NP6 ) = DRDX(10PU3 ,10NP6 ) + DCN*DCXNP6

C
  REAC(10HAN ) = REAC(10HAN ) - RR
  DRDX(10HAN ,10HN3) = DRDX(10HAN ,10HN3) - DXH
  DRDX(10HAN ,10PU4 ) = DRDX(10HAN ,10PU4 ) - DXPU4

```

```

DRDX(10HAN ,10PU3 ) = DRDX(10HAN ,10PU3 ) - DXPU3
DRDX(10HAN ,10HAN ) = DRDX(10HAN ,10HAN ) - DXHAN

C
  DRDX(10HAN ,10HN3) = DRDX(10HAN ,10HN3) - DCN*DCXH
  DRDX(10HAN ,10U6 ) = DRDX(10HAN ,10U6 ) - DCN*DCXU6
  DRDX(10HAN ,10PU4 ) = DRDX(10HAN ,10PU4 ) - DCN*DCXPU4
  DRDX(10HAN ,10PU3 ) = DRDX(10HAN ,10PU3 ) - DCN*DCXPU3
  DRDX(10HAN ,10U4 ) = DRDX(10HAN ,10U4 ) - DCN*DCXU4
  DRDX(10HAN ,10HYD ) = DRDX(10HAN ,10HYD ) - DCN*DCXHYD
  DRDX(10HAN ,10HAN ) = DRDX(10HAN ,10HAN ) - DCN*DCXHAN
  DRDX(10HAN ,10PU5 ) = DRDX(10HAN ,10PU5 ) - DCN*DCXPU5
  DRDX(10HAN ,10PU6 ) = DRDX(10HAN ,10PU6 ) - DCN*DCXPU6
  DRDX(10HAN ,10NP4 ) = DRDX(10HAN ,10NP4 ) - DCN*DCXNP4
  DRDX(10HAN ,10NP5 ) = DRDX(10HAN ,10NP5 ) - DCN*DCXNP5
  DRDX(10HAN ,10NP6 ) = DRDX(10HAN ,10NP6 ) - DCN*DCXNP6

C
C... (3) DISINTEGRATION OF HYDRAZINE BY PU(IV)
FROM EXTRA.M-CODE EQ-(2.5-3)
C
  CK = ECHEM(3)*FCHEM(3)*CK3
C
  XH1 = 1.00/(XH+CK3A)
C
  RR = CK*XPU4*XHYD*XH1
C
  DXH = - CK*XPU4*XHYD*XH1*XH1
  DXPU4 = CK *XHYD*XH1
  DXHYD = CK*XPU4 *XH1

C
  REAC(10PU4 ) = REAC(10PU4 ) - RR
  DRDX(10PU4 ,10HN3) = DRDX(10PU4 ,10HN3) - DXH
  DRDX(10PU4 ,10PU4 ) = DRDX(10PU4 ,10PU4 ) - DXPU4
  DRDX(10PU4 ,10HYD ) = DRDX(10PU4 ,10HYD ) - DXHYD

C
  REAC(10PU3 ) = REAC(10PU3 ) + RR
  DRDX(10PU3 ,10HN3) = DRDX(10PU3 ,10HN3) + DXH
  DRDX(10PU3 ,10PU4 ) = DRDX(10PU3 ,10PU4 ) + DXPU4
  DRDX(10PU3 ,10HYD ) = DRDX(10PU3 ,10HYD ) + DXHYD

C
  REAC(10HYD ) = REAC(10HYD ) - RR
  DRDX(10HYD ,10HN3) = DRDX(10HYD ,10HN3) - DXH
  DRDX(10HYD ,10PU4 ) = DRDX(10HYD ,10PU4 ) - DXPU4
  DRDX(10HYD ,10HYD ) = DRDX(10HYD ,10HYD ) - DXHYD

C
C... (4) REOXYDATION OF PU(III) ( FROM R.MIXSET)
C
IF ( XN02 .LT. 1.0D-4 ) THEN
C
  CK = ECHEM(4)*FCHEM(4)*CK401
C
  RR = CK*XPU3*(XH**1.800)
  DXH = 1.800*CK*XPU3*(XH**0.800)
  DXPU3 = CK *(XH**1.800)
C
  REAC(10HN3) = REAC(10HN3) - 1.500*RR
  DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 1.500*DXH
  DRDX(10HN3,10PU3 ) = DRDX(10HN3,10PU3 ) - 1.500*DXPU3

C
  REAC(10PU4 ) = REAC(10PU4 ) + RR
  DRDX(10PU4 ,10HN3) = DRDX(10PU4 ,10HN3) + DXH
  DRDX(10PU4 ,10PU3 ) = DRDX(10PU4 ,10PU3 ) + DXPU3

C
  REAC(10PU3 ) = REAC(10PU3 ) - RR
  DRDX(10PU3 ,10HN3) = DRDX(10PU3 ,10HN3) - DXH
  DRDX(10PU3 ,10PU3 ) = DRDX(10PU3 ,10PU3 ) - DXPU3

C
  REAC(10HN02) = REAC(10HN02) + 0.500*RR
  DRDX(10HN02,10HN3) = DRDX(10HN02,10HN3) + 0.500*DXH
  DRDX(10HN02,10PU3 ) = DRDX(10HN02,10PU3 ) + 0.500*DXPU3

ELSE IF ( XN02 .LT. 2.3D-2 ) THEN
C
  CK = ECHEM(4)*FCHEM(4)
C
  A1 = 10.000**(-CK402*DLOG10(ZH)-CK403)
  A2 = XN02**(CK404-CK405*DLOG10(ZH))
  A3 = A1*A2
C
  RR = XPU3*A3*CK
  DXH = - CK402 /ZH *XPU3*A3*CK
  - DLOG(XN02)*CK405
  DXPU3 = /DLOG(10.000)/ZH *XPU3*A3*CK
  A3*CK
  DXN02 = (CK404-CK405*DLOG10(ZH))/XN02 *XPU3*A3*CK

```



```

C
    REAC(10HN03) = REAC(10HN03) - 1.5D0*RR
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 1.5D0*DXH
    DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) - 1.5D0*DXPU3
    DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) - 1.5D0*DXN02

C
    REAC(10PU4) = REAC(10PU4) + RR
    DRDX(10PU4,10HN03) = DRDX(10PU4,10HN03) + DXH
    DRDX(10PU4,10PU3) = DRDX(10PU4,10PU3) + DXPU3
    DRDX(10PU4,10HN02) = DRDX(10PU4,10HN02) + DXN02

C
    REAC(10PU3) = REAC(10PU3) - RR
    DRDX(10PU3,10HN03) = DRDX(10PU3,10HN03) - DXH
    DRDX(10PU3,10PU3) = DRDX(10PU3,10PU3) - DXPU3
    DRDX(10PU3,10HN02) = DRDX(10PU3,10HN02) - DXN02

C
    REAC(10HN02) = REAC(10HN02) + 0.5D0*RR
    DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) + 0.5D0*DXH
    DRDX(10HN02,10PU3) = DRDX(10HN02,10PU3) + 0.5D0*DXPU3
    DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + 0.5D0*DXN02

C
ELSE
C
    CK = ECHEM(4)*FCHEM(4)*CK406

C
    RR = CK*XPUS
    DXPU3 = CK

C
    REAC(10HN03) = REAC(10HN03) - 1.5D0*RR
    DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) - 1.5D0*DXPU3

C
    REAC(10PU4) = REAC(10PU4) + RR
    DRDX(10PU4,10PU3) = DRDX(10PU4,10PU3) + DXPU3

C
    REAC(10PU3) = REAC(10PU3) - RR
    DRDX(10PU3,10PU3) = DRDX(10PU3,10PU3) - DXPU3

C
    REAC(10HN02) = REAC(10HN02) + 0.5D0*RR
    DRDX(10HN02,10PU3) = DRDX(10HN02,10PU3) + 0.5D0*DXPU3

C
END IF

C... (5) DISINTEGRATION OF NITROUS ACID BY HYDRAZINE ( FROM R.MIXSET)
C
    CK = ECHEM(5)*FCHEM(5)*CK5

C
    RR = CK*XH*XNO2*XHYD
    DXH = CK *XNO2*XHYD
    DXHYD = CK*XH*XNO2
    DXN02 = CK*XH *XHYD

C
    REAC(10HN03) = REAC(10HN03) + RR
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DXH
    DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + DXHYD
    DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DXN02

C
    REAC(10HYD) = REAC(10HYD) - RR
    DRDX(10HYD,10HN03) = DRDX(10HYD,10HN03) - DXH
    DRDX(10HYD,10HYD) = DRDX(10HYD,10HYD) - DXHYD
    DRDX(10HYD,10HN02) = DRDX(10HYD,10HN02) - DXN02

C
    REAC(10HN02) = REAC(10HN02) - RR
    DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) - DXH
    DRDX(10HN02,10HYD) = DRDX(10HN02,10HYD) - DXHYD
    DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) - DXN02

C
C... (17) GENERATION OF NITROUS ACID THROUGH RADIOLYSIS
FROM EXTRA.M-CODE EQ-(2.5-6)
C
    CK = ECHEM(17)*FCHEM(17)*CK17
    RR = CK

C
    REAC(10HN03) = REAC(10HN03) - RR

C
    REAC(10HN02) = REAC(10HN02) + RR

C
C... (18) GENERATION OF NITRIC ACID THROUGH RADIOLYSIS
FROM EXTRA.M-CODE EQ-(2.5-7)
C
    CK = ECHEM(18)*FCHEM(18)*CK18

C
    RR = CK * XNO2
    DXN02 = CK
    
```

```

C
    REAC(10HN03) = REAC(10HN03) + RR
    DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DXN02

C
    REAC(10HN02) = REAC(10HN02) - RR
    DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) - DXN02

C
C... (27) DISINTEGRATION OF NITROUS ACID BY HAN ( FROM R.MIXSET )
( CK27 IN EXTRA.M-CODE EQ-(2.5-8) IS 436.0 )
C
    CK = ECHEM(27)*FCHEM(27)*CK27

C
    RR = CK*XNO2*XHAN*XH
    DXH = CK*XNO2*XHAN
    DXHAN = CK*XNO2 *XH
    DXN02 = CK *XH*XHAN*XH

C
    REAC(10HN03) = REAC(10HN03) + RR
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DXH
    DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + DXHAN
    DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DXN02

C
    REAC(10HAN) = REAC(10HAN) - RR
    DRDX(10HAN,10HN03) = DRDX(10HAN,10HN03) - DXH
    DRDX(10HAN,10HAN) = DRDX(10HAN,10HAN) - DXHAN
    DRDX(10HAN,10HN02) = DRDX(10HAN,10HN02) - DXN02

C
    REAC(10HN02) = REAC(10HN02) - RR
    DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) - DXH
    DRDX(10HN02,10HAN) = DRDX(10HN02,10HAN) - DXHAN
    DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) - DXN02

C
C... (6) DISPROPORTIONATION OF NP
FROM EXTRA.M-CODE EQ-(2.5-9)
C
    CK = ECHEM(6)*FCHEM(6)*CK6

C
    RR = CK*XNP4*XNP6*(2.16D0+12.5D0*XCN)
    DXNP4 = CK *XNP6*(2.16D0+12.5D0*XCN)
    DXNP6 = CK*XNP4 *(2.16D0+12.5D0*XCN)
    DCN = CK*XNP4*XNP6*12.5D0

C
    REAC(10HN03) = REAC(10HN03) + 4.0D0*RR
    DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.0D0*DXNP4
    DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.0D0*DXNP6

C
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.0D0*DCN*DCXH
    DRDX(10HN03,10U6) = DRDX(10HN03,10U6) + 4.0D0*DCN*DCXU6
    DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 4.0D0*DCN*DCXPU4
    DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) + 4.0D0*DCN*DCXPU3
    DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 4.0D0*DCN*DCXU4
    DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + 4.0D0*DCN*DCXHYD
    DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + 4.0D0*DCN*DCXHAN
    DRDX(10HN03,10PU5) = DRDX(10HN03,10PU5) + 4.0D0*DCN*DCXPU5
    DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 4.0D0*DCN*DCXPU6
    DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.0D0*DCN*DCXNP4
    DRDX(10HN03,10NP5) = DRDX(10HN03,10NP5) + 4.0D0*DCN*DCXNP5
    DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.0D0*DCN*DCXNP6

C
    REAC(10NP4) = REAC(10NP4) - RR
    DRDX(10NP4,10NP4) = DRDX(10NP4,10NP4) - DXNP4
    DRDX(10NP4,10NP6) = DRDX(10NP4,10NP6) - DXNP6

C
    DRDX(10NP4,10HN03) = DRDX(10NP4,10HN03) - DCN*DCXH
    DRDX(10NP4,10U6) = DRDX(10NP4,10U6) - DCN*DCXU6
    DRDX(10NP4,10PU4) = DRDX(10NP4,10PU4) - DCN*DCXPU4
    DRDX(10NP4,10PU3) = DRDX(10NP4,10PU3) - DCN*DCXPU3
    DRDX(10NP4,10U4) = DRDX(10NP4,10U4) - DCN*DCXU4
    DRDX(10NP4,10HYD) = DRDX(10NP4,10HYD) - DCN*DCXHYD
    DRDX(10NP4,10HAN) = DRDX(10NP4,10HAN) - DCN*DCXHAN
    DRDX(10NP4,10PU5) = DRDX(10NP4,10PU5) - DCN*DCXPU5
    DRDX(10NP4,10PU6) = DRDX(10NP4,10PU6) - DCN*DCXPU6
    DRDX(10NP4,10NP4) = DRDX(10NP4,10NP4) - DCN*DCXNP4
    DRDX(10NP4,10NP5) = DRDX(10NP4,10NP5) - DCN*DCXNP5
    DRDX(10NP4,10NP6) = DRDX(10NP4,10NP6) - DCN*DCXNP6

C
    REAC(10NP5) = REAC(10NP5) + 2.0D0*RR
    DRDX(10NP5,10NP4) = DRDX(10NP5,10NP4) + 2.0D0*DXNP4
    DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) + 2.0D0*DXNP6

C
    DRDX(10NP5,10HN03) = DRDX(10NP5,10HN03) + 2.0D0*DCN*DCXH
    DRDX(10NP5,10U6) = DRDX(10NP5,10U6) + 2.0D0*DCN*DCXU6
    DRDX(10NP5,10PU4) = DRDX(10NP5,10PU4) + 2.0D0*DCN*DCXPU4
    DRDX(10NP5,10PU3) = DRDX(10NP5,10PU3) + 2.0D0*DCN*DCXPU3
    
```

```
DRDX(10NP5 ,10U4 ) = DRDX(10NP5 ,10U4 ) + 2.000*DCN*DCXU4  
DRDX(10NP5 ,10HYD ) = DRDX(10NP5 ,10HYD ) + 2.000*DCN*DCXHYD  
DRDX(10NP5 ,10HAN ) = DRDX(10NP5 ,10HAN ) + 2.000*DCN*DCXHAN  
DRDX(10NP5 ,10PU5 ) = DRDX(10NP5 ,10PU5 ) + 2.000*DCN*DCXPU5  
DRDX(10NP5 ,10PU6 ) = DRDX(10NP5 ,10PU6 ) + 2.000*DCN*DCXPU6  
DRDX(10NP5 ,10NP4 ) = DRDX(10NP5 ,10NP4 ) + 2.000*DCN*DCXNP4  
DRDX(10NP5 ,10NP5 ) = DRDX(10NP5 ,10NP5 ) + 2.000*DCN*DCXNP5  
DRDX(10NP5 ,10NP6 ) = DRDX(10NP5 ,10NP6 ) + 2.000*DCN*DCXNP6  
  
C  
C  
C  
C  
C  
C  
C... (7) OXYDATION OF NP(IV) BY PU(IV)  
C FROM EXTRA.M-CODE EQ-(2.5-10)  
C  
C CK = ECHEM(7)*FCHEM(7)*CK7  
  
C ZH2 = ZH *ZH  
C ZH4 = ZH2*ZH2  
C RR = CK*XNP4*XP4/ZH4  
  
C DXH =-4.000*CK*XNP4*XP4/ZH4/ZH  
C DXPU4 = CK*XNP4 /ZH4  
C DXNP4 = CK *XP4/ZH4  
  
C REAC(10HN03) = REAC(10HN03) + 4.000*RR  
C DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.000*DXH  
C DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 4.000*DXPU4  
C DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.000*DXNP4  
  
C REAC(10PU4) = REAC(10PU4) - RR  
C DRDX(10PU4,10HN03) = DRDX(10PU4,10HN03) - DXH  
C DRDX(10PU4,10PU4) = DRDX(10PU4,10PU4) - DXPU4  
C DRDX(10PU4,10NP4) = DRDX(10PU4,10NP4) - DXNP4  
  
C REAC(10PU3) = REAC(10PU3) + RR  
C DRDX(10PU3,10HN03) = DRDX(10PU3,10HN03) + DXH  
C DRDX(10PU3,10PU4) = DRDX(10PU3,10PU4) + DXPU4  
C DRDX(10PU3,10NP4) = DRDX(10PU3,10NP4) + DXNP4  
  
C REAC(10NP4) = REAC(10NP4) - RR  
C DRDX(10NP4,10HN03) = DRDX(10NP4,10HN03) - DXH  
C DRDX(10NP4,10PU4) = DRDX(10NP4,10PU4) - DXPU4  
C DRDX(10NP4,10NP4) = DRDX(10NP4,10NP4) - DXNP4  
  
C REAC(10NP5) = REAC(10NP5) + RR  
C DRDX(10NP5,10HN03) = DRDX(10NP5,10HN03) + DXH  
C DRDX(10NP5,10PU4) = DRDX(10NP5,10PU4) + DXPU4  
C DRDX(10NP5,10NP4) = DRDX(10NP5,10NP4) + DXNP4  
  
C  
C  
C... (8) REDUCTION OF NP(VI) BY HYDRAZINE  
C FROM EXTRA.M-CODE EQ-(2.5-11)  
C  
C CK = ECHEM(8)*FCHEM(8)*CK8  
  
C RR = CK*XNP6*XHYD*(ZH**(-1.300))  
C DXH = -1.300*CK*XNP6*XHYD*(ZH**(-1.300))/ZH  
C DXHYD = CK*XNP6 *(ZH**(-1.300))  
C DXNP6 = CK *XHYD*(ZH**(-1.300))  
  
C REAC(10HN03) = REAC(10HN03) + RR  
C DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DXH  
C DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + DXHYD  
C DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + DXNP6  
  
C REAC(10HYD) = REAC(10HYD) - RR  
C DRDX(10HYD,10HN03) = DRDX(10HYD,10HN03) - DXH  
C DRDX(10HYD,10HYD) = DRDX(10HYD,10HYD) - DXHYD  
C DRDX(10HYD,10NP6) = DRDX(10HYD,10NP6) - DXNP6  
  
C REAC(10NP5) = REAC(10NP5) + RR  
C DRDX(10NP5,10HN03) = DRDX(10NP5,10HN03) + DXH
```

```
DRDX(10NP5 ,10HYD ) = DRDX(10NP5 ,10HYD ) + DXHYD  
DRDX(10NP5 ,10NP6 ) = DRDX(10NP5 ,10NP6 ) + DXNP6  
  
C  
C REAC(10NP6 ) = REAC(10NP6 ) - RR  
C DRDX(10NP6 ,10HN03) = DRDX(10NP6 ,10HN03) - DXH  
C DRDX(10NP6 ,10HYD) = DRDX(10NP6 ,10HYD) - DXHYD  
C DRDX(10NP6 ,10NP6) = DRDX(10NP6 ,10NP6) - DXNP6  
  
C  
C  
C... (9) REDUCTION OF NP(VI) BY URANOUS  
C FROM EXTRA.M-CODE EQ-(2.5-12)  
C  
C CK = ECHEM(9)*FCHEM(9)*CK9  
  
C RR = CK*XNP6*XU4  
C DXU4 = CK*XNP6  
C DXNP6 = CK *XU4  
  
C REAC(10HN03) = REAC(10HN03) + 4.000*RR  
C DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 4.000*DXU4  
C DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.000*DXNP6  
  
C REAC(10U6) = REAC(10U6) + 1.000*RR  
C DRDX(10U6 ,10U4) = DRDX(10U6 ,10U4) + 1.000*DXU4  
C DRDX(10U6 ,10NP6) = DRDX(10U6 ,10NP6) + 1.000*DXNP6  
  
C REAC(10U4) = REAC(10U4) - 1.000*RR  
C DRDX(10U4 ,10U4) = DRDX(10U4 ,10U4) - 1.000*DXU4  
C DRDX(10U4 ,10NP6) = DRDX(10U4 ,10NP6) - 1.000*DXNP6  
  
C  
C REAC(10NP5) = REAC(10NP5) + 2.000*RR  
C DRDX(10NP5 ,10U4) = DRDX(10NP5 ,10U4) + 2.000*DXU4  
C DRDX(10NP5 ,10NP6) = DRDX(10NP5 ,10NP6) + 2.000*DXNP6  
  
C  
C REAC(10NP6) = REAC(10NP6) - 2.000*RR  
C DRDX(10NP6 ,10U4) = DRDX(10NP6 ,10U4) - 2.000*DXU4  
C DRDX(10NP6 ,10NP6) = DRDX(10NP6 ,10NP6) - 2.000*DXNP6  
  
C  
C  
C... (10) REDUCTION OF NP(V) BY URANOUS  
C FROM EXTRA.M-CODE EQ-(2.5-13)  
C  
C CK = ECHEM(10)*FCHEM(10)*CK10  
  
C RR = CK*XNP5*XU4*(1.6D0/ZH/ZH +1.42D0*XH)  
C DXH = CK*XNP5*XU4*(-3.2D0/ZH/ZH+1.42D0 )  
C DXU4 = CK*XNP5 *(1.6D0/ZH/ZH +1.42D0*XH)  
C DXNP5 = CK *XU4*(1.6D0/ZH/ZH +1.42D0*XH)  
  
C REAC(10HN03) = REAC(10HN03) - 4.000*RR  
C DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 4.000*DXH  
C DRDX(10HN03,10U4) = DRDX(10HN03,10U4) - 4.000*DXU4  
C DRDX(10HN03,10NP5) = DRDX(10HN03,10NP5) - 4.000*DXNP5  
  
C REAC(10U6) = REAC(10U6) + 1.000*RR  
C DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + 1.000*DXH  
C DRDX(10U6 ,10U4) = DRDX(10U6 ,10U4) + 1.000*DXU4  
C DRDX(10U6 ,10NP5) = DRDX(10U6 ,10NP5) + 1.000*DXNP5  
  
C REAC(10U4) = REAC(10U4) - 1.000*RR  
C DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - 1.000*DXH  
C DRDX(10U4 ,10U4) = DRDX(10U4 ,10U4) - 1.000*DXU4  
C DRDX(10U4 ,10NP5) = DRDX(10U4 ,10NP5) - 1.000*DXNP5  
  
C REAC(10NP4) = REAC(10NP4) + 2.000*RR  
C DRDX(10NP4,10HN03) = DRDX(10NP4,10HN03) + 2.000*DXH  
C DRDX(10NP4,10U4) = DRDX(10NP4,10U4) + 2.000*DXU4  
C DRDX(10NP4,10NP5) = DRDX(10NP4,10NP5) + 2.000*DXNP5  
  
C  
C REAC(10NP5) = REAC(10NP5) - 2.000*RR  
C DRDX(10NP5 ,10HN03) = DRDX(10NP5 ,10HN03) - 2.000*DXH  
C DRDX(10NP5 ,10U4) = DRDX(10NP5 ,10U4) - 2.000*DXU4  
C DRDX(10NP5 ,10NP5) = DRDX(10NP5 ,10NP5) - 2.000*DXNP5  
  
C  
C  
C... (11) REDUCTION OF NP(VI) BY HAN  
C FROM EXTRA.M-CODE EQ-(2.5-14)  
C  
C CK = ECHEM(11)*FCHEM(11)*CK11  
  
C RR = CK*XNP6*XHAN/ZH  
C DXH = -CK*XNP6*XHAN/ZH/ZH  
C DXHAN = CK*XNP6 /ZH  
C DXNP6 = CK *XHAN/ZH  
  
C REAC(10HN03) = REAC(10HN03) + 2.000*RR  
C DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 2.000*DXH  
C DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + 2.000*DXHAN  
C DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 2.000*DXNP6
```

```
C
REAC(10HAN) = REAC(10HAN) - RR
DRDX(10HAN,10HN3) = DRDX(10HAN,10HN3) - DXH
DRDX(10HAN,10HAN) = DRDX(10HAN,10HAN) - DXHAN
DRDX(10HAN,10NP6) = DRDX(10HAN,10NP6) - DXNP6

C
REAC(10NP5) = REAC(10NP5) + RR
DRDX(10NP5,10HN3) = DRDX(10NP5,10HN3) + DXH
DRDX(10NP5,10HAN) = DRDX(10NP5,10HAN) + DXHAN
DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) + DXNP6

C
REAC(10NP6) = REAC(10NP6) - RR
DRDX(10NP6,10HN3) = DRDX(10NP6,10HN3) - DXH
DRDX(10NP6,10HAN) = DRDX(10NP6,10HAN) - DXHAN
DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - DXNP6
```

C
C
C... (12) REOXYDATION OF PU(V) AND REDUCTION OF PU(IV)
FROM EXTRA.M-CODE EQ-(2.5-15)

```
C
CK = ECHEM(12)*FCHEM(12)*CK12

C
RR = CK*XPU4*XPU5*XH/(XH+0.05400)
DXH = 0.05400*CK*XPU4*XPU5/(XH+0.05400)/(XH+0.05400)
DXPU5 = CK*XPU4 *XH/(XH+0.05400)
DXPU4 = CK *XPU5*XH/(XH+0.05400)
```

```
C
REAC(10PU4) = REAC(10PU4) - RR
DRDX(10PU4,10HN3) = DRDX(10PU4,10HN3) - DXH
DRDX(10PU4,10PU4) = DRDX(10PU4,10PU4) - DXPU4
DRDX(10PU4,10PU5) = DRDX(10PU4,10PU5) - DXPU5

C
REAC(10PU5) = REAC(10PU5) - RR
DRDX(10PU5,10HN3) = DRDX(10PU5,10HN3) - DXH
DRDX(10PU5,10PU4) = DRDX(10PU5,10PU4) - DXPU4
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - DXPU5

C
REAC(10PU6) = REAC(10PU6) + RR
DRDX(10PU6,10HN3) = DRDX(10PU6,10HN3) + DXH
DRDX(10PU6,10PU4) = DRDX(10PU6,10PU4) + DXPU4
DRDX(10PU6,10PU5) = DRDX(10PU6,10PU5) + DXPU5
```

```
C
REAC(10PU3) = REAC(10PU3) + RR
DRDX(10PU3,10HN3) = DRDX(10PU3,10HN3) + DXH
DRDX(10PU3,10PU4) = DRDX(10PU3,10PU4) + DXPU4
DRDX(10PU3,10PU5) = DRDX(10PU3,10PU5) + DXPU5
```

C
C
C... (13) REOXYDATION OF PU(III) AND REDUCTION OF PU(V)
FROM EXTRA.M-CODE EQ-(2.5-16)

```
C
CK = ECHEM(13)*FCHEM(13)*CK13

C
RR = CK*XPU3*XPU5*XH
DXH = CK*XPU3*XPU5
DXPU3 = CK *XPU5*XH
DXPU5 = CK*XPU3 *XH
```

```
C
REAC(10HN3) = REAC(10HN3) - 4.000*RR
DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 4.000*DXH
DRDX(10HN3,10PU3) = DRDX(10HN3,10PU3) - 4.000*DXPU3
DRDX(10HN3,10PU5) = DRDX(10HN3,10PU5) - 4.000*DXPU5

C
REAC(10PU3) = REAC(10PU3) - RR
DRDX(10PU3,10HN3) = DRDX(10PU3,10HN3) - DXH
DRDX(10PU3,10PU3) = DRDX(10PU3,10PU3) - DXPU3
DRDX(10PU3,10PU5) = DRDX(10PU3,10PU5) - DXPU5

C
REAC(10PU5) = REAC(10PU5) - RR
DRDX(10PU5,10HN3) = DRDX(10PU5,10HN3) - DXH
DRDX(10PU5,10PU3) = DRDX(10PU5,10PU3) - DXPU3
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - DXPU5
```

```
C
REAC(10PU4) = REAC(10PU4) + 2.000*RR
DRDX(10PU4,10HN3) = DRDX(10PU4,10HN3) + 2.000*DXH
DRDX(10PU4,10PU3) = DRDX(10PU4,10PU3) + 2.000*DXPU3
DRDX(10PU4,10PU5) = DRDX(10PU4,10PU5) + 2.000*DXPU5
```

C
C
C... (15) REOXYDATION OF PU(III) AND REDUCTION OF PU(VI)
FROM EXTRA.M-CODE EQ-(2.5-17)

```
C
CK = ECHEM(15)*FCHEM(15)*CK15

C
RR = CK*XPU3*XPU6
DXPU3 = CK *XPU6
DXPU6 = CK*XPU3

C
REAC(10PU3) = REAC(10PU3) - RR
```

```
DRDX(10PU3,10PU3) = DRDX(10PU3,10PU3) - DXPU3
DRDX(10PU3,10PU6) = DRDX(10PU3,10PU6) - DXPU6

C
REAC(10PU6) = REAC(10PU6) - RR
DRDX(10PU6,10PU3) = DRDX(10PU6,10PU3) - DXPU3
DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) - DXPU6

C
REAC(10PU4) = REAC(10PU4) + RR
DRDX(10PU4,10PU3) = DRDX(10PU4,10PU3) + DXPU3
DRDX(10PU4,10PU6) = DRDX(10PU4,10PU6) + DXPU6

C
REAC(10PU5) = REAC(10PU5) + RR
DRDX(10PU5,10PU3) = DRDX(10PU5,10PU3) + DXPU3
DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) + DXPU6
```

C
C
C... (16) REOXYDATION OF PU(V) BY NITRIC ACID
FROM EXTRA.M-CODE EQ-(2.5-18)

```
C
C16EPS = 1.00-10
ZN02 = XN02 + C16EPS

C
CK = ECHEM(16)*FCHEM(16)*CK16

C
RR = CK*XPU5*(ZN02**0.500)*(ZH**0.600)*(ZCN**0.400)
DXH = 0.600*CK*XPU5*(ZN02**0.500)/(ZH**0.600)*(ZCN**0.400)
DXN02 = 0.500*CK*XPU5/(ZN02**0.500)*(ZH**0.600)*(ZCN**0.400)
DXPU5 = CK *(ZN02**0.500)*(ZH**0.600)*(ZCN**0.400)
DCN = 0.400*CK*XPU5*(ZN02**0.500)*(ZH**0.600)/(ZCN**0.600)
```

```
C
REAC(10HN3) = REAC(10HN3) - 3.000*RR
DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 3.000*DXH
DRDX(10HN3,10HN2) = DRDX(10HN3,10HN2) - 3.000*DXN02
DRDX(10HN3,10PU5) = DRDX(10HN3,10PU5) - 3.000*DXPU5

C
DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 3.000*DCN*DCXH
DRDX(10HN3,10U6) = DRDX(10HN3,10U6) - 3.000*DCN*DCXU6
DRDX(10HN3,10PU4) = DRDX(10HN3,10PU4) - 3.000*DCN*DCXPU4
DRDX(10HN3,10PU3) = DRDX(10HN3,10PU3) - 3.000*DCN*DCXPU3
DRDX(10HN3,10U4) = DRDX(10HN3,10U4) - 3.000*DCN*DCXU4
DRDX(10HN3,10HYD) = DRDX(10HN3,10HYD) - 3.000*DCN*DCXHYD
DRDX(10HN3,10HAN) = DRDX(10HN3,10HAN) - 3.000*DCN*DCXHAN
DRDX(10HN3,10PU5) = DRDX(10HN3,10PU5) - 3.000*DCN*DCXPU5
DRDX(10HN3,10PU6) = DRDX(10HN3,10PU6) - 3.000*DCN*DCXPU6
DRDX(10HN3,10NP4) = DRDX(10HN3,10NP4) - 3.000*DCN*DCXNP4
DRDX(10HN3,10NP5) = DRDX(10HN3,10NP5) - 3.000*DCN*DCXNP5
DRDX(10HN3,10NP6) = DRDX(10HN3,10NP6) - 3.000*DCN*DCXNP6
```

```
C
REAC(10PU5) = REAC(10PU5) - 2.000*RR
DRDX(10PU5,10HN3) = DRDX(10PU5,10HN3) - 2.000*DXH
DRDX(10PU5,10HN2) = DRDX(10PU5,10HN2) - 2.000*DXN02
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - 2.000*DXPU5

C
DRDX(10PU5,10HN3) = DRDX(10PU5,10HN3) - 2.000*DCN*DCXH
DRDX(10PU5,10U6) = DRDX(10PU5,10U6) - 2.000*DCN*DCXU6
DRDX(10PU5,10PU4) = DRDX(10PU5,10PU4) - 2.000*DCN*DCXPU4
DRDX(10PU5,10PU3) = DRDX(10PU5,10PU3) - 2.000*DCN*DCXPU3
DRDX(10PU5,10U4) = DRDX(10PU5,10U4) - 2.000*DCN*DCXU4
DRDX(10PU5,10HYD) = DRDX(10PU5,10HYD) - 2.000*DCN*DCXHYD
DRDX(10PU5,10HAN) = DRDX(10PU5,10HAN) - 2.000*DCN*DCXHAN
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - 2.000*DCN*DCXPU5
DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) - 2.000*DCN*DCXPU6
DRDX(10PU5,10NP4) = DRDX(10PU5,10NP4) - 2.000*DCN*DCXNP4
DRDX(10PU5,10NP5) = DRDX(10PU5,10NP5) - 2.000*DCN*DCXNP5
DRDX(10PU5,10NP6) = DRDX(10PU5,10NP6) - 2.000*DCN*DCXNP6
```

```
C
REAC(10PU6) = REAC(10PU6) + 2.000*RR
DRDX(10PU6,10HN3) = DRDX(10PU6,10HN3) + 2.000*DXH
DRDX(10PU6,10HN2) = DRDX(10PU6,10HN2) + 2.000*DXN02
DRDX(10PU6,10PU5) = DRDX(10PU6,10PU5) + 2.000*DXPU5

C
DRDX(10PU6,10HN3) = DRDX(10PU6,10HN3) + 2.000*DCN*DCXH
DRDX(10PU6,10U6) = DRDX(10PU6,10U6) + 2.000*DCN*DCXU6
DRDX(10PU6,10PU4) = DRDX(10PU6,10PU4) + 2.000*DCN*DCXPU4
DRDX(10PU6,10PU3) = DRDX(10PU6,10PU3) + 2.000*DCN*DCXPU3
DRDX(10PU6,10U4) = DRDX(10PU6,10U4) + 2.000*DCN*DCXU4
DRDX(10PU6,10HYD) = DRDX(10PU6,10HYD) + 2.000*DCN*DCXHYD
DRDX(10PU6,10HAN) = DRDX(10PU6,10HAN) + 2.000*DCN*DCXHAN
DRDX(10PU6,10PU5) = DRDX(10PU6,10PU5) + 2.000*DCN*DCXPU5
DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) + 2.000*DCN*DCXPU6
DRDX(10PU6,10NP4) = DRDX(10PU6,10NP4) + 2.000*DCN*DCXNP4
DRDX(10PU6,10NP5) = DRDX(10PU6,10NP5) + 2.000*DCN*DCXNP5
DRDX(10PU6,10NP6) = DRDX(10PU6,10NP6) + 2.000*DCN*DCXNP6
```

```
C
REAC(10HN2) = REAC(10HN2) + RR
DRDX(10HN2,10HN3) = DRDX(10HN2,10HN3) + DXH
```

```

DFDX(1OHNO2,1OHNO2) = DRDX(1OHNO2,1OHNO2) + DXNO2
DFDX(1OHNO2,1OPU5) = DRDX(1OHNO2,1OPU5) + DXPU5
C
DFDX(1OHNO2,1OHNO3) = DRDX(1OHNO2,1OHNO3) + DCN*DCXH
DFDX(1OHNO2,1OU6) = DRDX(1OHNO2,1OU6) + DCN*DCXU6
DFDX(1OHNO2,1OPU4) = DRDX(1OHNO2,1OPU4) + DCN*DCXPU4
DFDX(1OHNO2,1OPU3) = DRDX(1OHNO2,1OPU3) + DCN*DCXPU3
DFDX(1OHNO2,1OU4) = DRDX(1OHNO2,1OU4) + DCN*DCXU4
DFDX(1OHNO2,1OHYD) = DRDX(1OHNO2,1OHYD) + DCN*DCXHYD
DFDX(1OHNO2,1OHAN) = DRDX(1OHNO2,1OHAN) + DCN*DCXHAN
DFDX(1OHNO2,1OPU5) = DRDX(1OHNO2,1OPU5) + DCN*DCXPU5
DFDX(1OHNO2,1OPU6) = DRDX(1OHNO2,1OPU6) + DCN*DCXPU6
DFDX(1OHNO2,1ONP4) = DRDX(1OHNO2,1ONP4) + DCN*DCXNP4
DFDX(1OHNO2,1ONP5) = DRDX(1OHNO2,1ONP5) + DCN*DCXNP5
DFDX(1OHNO2,1ONP6) = DRDX(1OHNO2,1ONP6) + DCN*DCXNP6
C
C
C... (19) DISPROPORTIONATION OF NP
C FROM EXTRA.M-CODE EQ-(2.5-19)
C
CK = ECHEM(19)*FCHEM(19)*CK19
C
RR = CK*XNP5*XNP5*XH*XH
DXH = 2.00*CK*XNP5*XNP5*XH
DXNP5 = 2.00*CK*XNP5 *XH*XH
C
REAC(1OHNO3) = REAC(1OHNO3) - 4.00*RR
DFDX(1OHNO3,1OHNO3) = DRDX(1OHNO3,1OHNO3) - 4.00*DXH
DFDX(1OHNO3,1ONP5) = DRDX(1OHNO3,1ONP5) - 4.00*DXNP5
C
REAC(1ONP4) = REAC(1ONP4) + 1.00*RR
DFDX(1ONP4,1OHNO3) = DRDX(1ONP4,1OHNO3) + 1.00*DXH
DFDX(1ONP4,1ONP5) = DRDX(1ONP4,1ONP5) + 1.00*DXNP5
C
REAC(1ONP5) = REAC(1ONP5) - 2.00*RR
DFDX(1ONP5,1OHNO3) = DRDX(1ONP5,1OHNO3) - 2.00*DXH
DFDX(1ONP5,1ONP5) = DRDX(1ONP5,1ONP5) - 2.00*DXNP5
C
REAC(1ONP6) = REAC(1ONP6) + 1.00*RR
DFDX(1ONP6,1OHNO3) = DRDX(1ONP6,1OHNO3) + 1.00*DXH
DFDX(1ONP6,1ONP5) = DRDX(1ONP6,1ONP5) + 1.00*DXNP5
C
C
C... (20) OXYDATION OF NP(V) BY HNO2
C FROM EXTRA.M-CODE EQ-(2.5-20)
C
C2OEPS = 1.00-25
ZNO2 = XNO2 + C2OEPS
C
CK = ECHEM(20)*FCHEM(20)*CK20
C
FF = CK*(XH**1.3000)*XCN*XCN
C
RR = CK*(XH**1.3000)*XCN*XCN*XNP5*ZNO2/(ZNO2+XNP5)
DXH = 1.3000*CK*(XH**0.3000)*XCN*XCN*XNP5*ZNO2/(ZNO2+XNP5)
DXNO2 = FF*XNP5*XNP5/(ZNO2+XNP5)/(ZNO2+XNP5)
DXNP5 = FF*ZNO2*ZNO2/(ZNO2+XNP5)/(ZNO2+XNP5)
DCN = 2.0000*CK*(XH**1.3000)*XCN*XNP5*ZNO2/(ZNO2+XNP5)
C
REAC(1OHNO3) = REAC(1OHNO3) - 3.00*RR
DFDX(1OHNO3,1OHNO3) = DRDX(1OHNO3,1OHNO3) - 3.00*DXH
DFDX(1OHNO3,1OHNO2) = DRDX(1OHNO3,1OHNO2) - 3.00*DXNO2
DFDX(1OHNO3,1ONP5) = DRDX(1OHNO3,1ONP5) - 3.00*DXNP5
C
DFDX(1OHNO3,1OHNO3) = DRDX(1OHNO3,1OHNO3) - 3.00*DCN*DCXH
DFDX(1OHNO3,1OU6) = DRDX(1OHNO3,1OU6) - 3.00*DCN*DCXU6
DFDX(1OHNO3,1OPU4) = DRDX(1OHNO3,1OPU4) - 3.00*DCN*DCXPU4
DFDX(1OHNO3,1OPU3) = DRDX(1OHNO3,1OPU3) - 3.00*DCN*DCXPU3
DFDX(1OHNO3,1OU4) = DRDX(1OHNO3,1OU4) - 3.00*DCN*DCXU4
DFDX(1OHNO3,1OHYD) = DRDX(1OHNO3,1OHYD) - 3.00*DCN*DCXHYD
DFDX(1OHNO3,1OHAN) = DRDX(1OHNO3,1OHAN) - 3.00*DCN*DCXHAN
DFDX(1OHNO3,1OPU5) = DRDX(1OHNO3,1OPU5) - 3.00*DCN*DCXPU5
DFDX(1OHNO3,1OPU6) = DRDX(1OHNO3,1OPU6) - 3.00*DCN*DCXPU6
DFDX(1OHNO3,1ONP4) = DRDX(1OHNO3,1ONP4) - 3.00*DCN*DCXNP4
DFDX(1OHNO3,1ONP5) = DRDX(1OHNO3,1ONP5) - 3.00*DCN*DCXNP5
DFDX(1OHNO3,1ONP6) = DRDX(1OHNO3,1ONP6) - 3.00*DCN*DCXNP6
C
REAC(1OHNO2) = REAC(1OHNO2) + RR
DFDX(1OHNO2,1OHNO3) = DRDX(1OHNO2,1OHNO3) + DXH
DFDX(1OHNO2,1OHNO2) = DRDX(1OHNO2,1OHNO2) + DXNO2
DFDX(1OHNO2,1ONP5) = DRDX(1OHNO2,1ONP5) + DXNP5
C
DFDX(1OHNO2,1OHNO3) = DRDX(1OHNO2,1OHNO3) + DCN*DCXH
DFDX(1OHNO2,1OU6) = DRDX(1OHNO2,1OU6) + DCN*DCXU6
DFDX(1OHNO2,1OPU4) = DRDX(1OHNO2,1OPU4) + DCN*DCXPU4
DFDX(1OHNO2,1OPU3) = DRDX(1OHNO2,1OPU3) + DCN*DCXPU3
DFDX(1OHNO2,1OU4) = DRDX(1OHNO2,1OU4) + DCN*DCXU4
DFDX(1OHNO2,1OHYD) = DRDX(1OHNO2,1OHYD) + DCN*DCXHYD

```

```

DFDX(1OHNO2,1OHAN) = DRDX(1OHNO2,1OHAN) + DCN*DCXHAN
DFDX(1OHNO2,1OPU5) = DRDX(1OHNO2,1OPU5) + DCN*DCXPU5
DFDX(1OHNO2,1OPU6) = DRDX(1OHNO2,1OPU6) + DCN*DCXPU6
DFDX(1OHNO2,1ONP4) = DRDX(1OHNO2,1ONP4) + DCN*DCXNP4
DFDX(1OHNO2,1ONP5) = DRDX(1OHNO2,1ONP5) + DCN*DCXNP5
DFDX(1OHNO2,1ONP6) = DRDX(1OHNO2,1ONP6) + DCN*DCXNP6
C
REAC(1ONP5) = REAC(1ONP5) - 2.00*RR
DFDX(1ONP5,1OHNO3) = DRDX(1ONP5,1OHNO3) - 2.00*DXH
DFDX(1ONP5,1OHNO2) = DRDX(1ONP5,1OHNO2) - 2.00*DXNO2
DFDX(1ONP5,1ONP5) = DRDX(1ONP5,1ONP5) - 2.00*DXNP5
C
DFDX(1ONP5,1OHNO3) = DRDX(1ONP5,1OHNO3) - 2.00*DCN*DCXH
DFDX(1ONP5,1OU6) = DRDX(1ONP5,1OU6) - 2.00*DCN*DCXU6
DFDX(1ONP5,1OPU4) = DRDX(1ONP5,1OPU4) - 2.00*DCN*DCXPU4
DFDX(1ONP5,1OPU3) = DRDX(1ONP5,1OPU3) - 2.00*DCN*DCXPU3
DFDX(1ONP5,1OU4) = DRDX(1ONP5,1OU4) - 2.00*DCN*DCXU4
DFDX(1ONP5,1OHYD) = DRDX(1ONP5,1OHYD) - 2.00*DCN*DCXHYD
DFDX(1ONP5,1OHAN) = DRDX(1ONP5,1OHAN) - 2.00*DCN*DCXHAN
DFDX(1ONP5,1OPU5) = DRDX(1ONP5,1OPU5) - 2.00*DCN*DCXPU5
DFDX(1ONP5,1OPU6) = DRDX(1ONP5,1OPU6) - 2.00*DCN*DCXPU6
DFDX(1ONP5,1ONP4) = DRDX(1ONP5,1ONP4) - 2.00*DCN*DCXNP4
DFDX(1ONP5,1ONP5) = DRDX(1ONP5,1ONP5) - 2.00*DCN*DCXNP5
DFDX(1ONP5,1ONP6) = DRDX(1ONP5,1ONP6) - 2.00*DCN*DCXNP6
C
REAC(1ONP6) = REAC(1ONP6) + 2.00*RR
DFDX(1ONP6,1OHNO3) = DRDX(1ONP6,1OHNO3) + 2.00*DXH
DFDX(1ONP6,1OHNO2) = DRDX(1ONP6,1OHNO2) + 2.00*DXNO2
DFDX(1ONP6,1ONP5) = DRDX(1ONP6,1ONP5) + 2.00*DXNP5
C
DFDX(1ONP6,1OHNO3) = DRDX(1ONP6,1OHNO3) + 2.00*DCN*DCXH
DFDX(1ONP6,1OU6) = DRDX(1ONP6,1OU6) + 2.00*DCN*DCXU6
DFDX(1ONP6,1OPU4) = DRDX(1ONP6,1OPU4) + 2.00*DCN*DCXPU4
DFDX(1ONP6,1OPU3) = DRDX(1ONP6,1OPU3) + 2.00*DCN*DCXPU3
DFDX(1ONP6,1OU4) = DRDX(1ONP6,1OU4) + 2.00*DCN*DCXU4
DFDX(1ONP6,1OHYD) = DRDX(1ONP6,1OHYD) + 2.00*DCN*DCXHYD
DFDX(1ONP6,1OHAN) = DRDX(1ONP6,1OHAN) + 2.00*DCN*DCXHAN
DFDX(1ONP6,1OPU5) = DRDX(1ONP6,1OPU5) + 2.00*DCN*DCXPU5
DFDX(1ONP6,1OPU6) = DRDX(1ONP6,1OPU6) + 2.00*DCN*DCXPU6
DFDX(1ONP6,1ONP4) = DRDX(1ONP6,1ONP4) + 2.00*DCN*DCXNP4
DFDX(1ONP6,1ONP5) = DRDX(1ONP6,1ONP5) + 2.00*DCN*DCXNP5
DFDX(1ONP6,1ONP6) = DRDX(1ONP6,1ONP6) + 2.00*DCN*DCXNP6
C
C
C... (21) OXYDATION OF NP(IV) BY HNO3
C FROM EXTRA.M-CODE EQ-(2.5-21)
C
C21EPS = 1.00-10
ZNO2 = XNO2 + C21EPS
C
CK = ECHEM(21)*FCHEM(21)*CK21
C
RR = CK/DSORT(ZNO2)*XNP4*XNP5
DXNO2 = -0.500*CK/DSORT(ZNO2)*XNP4*XNP5/ZNO2
DXNP4 = CK/DSORT(ZNO2)*XNP5
DXNP5 = CK/DSORT(ZNO2)*XNP4
C
REAC(1OHNO3) = REAC(1OHNO3) + 5.00*RR
DFDX(1OHNO3,1OHNO2) = DRDX(1OHNO3,1OHNO2) + 5.00*DXNO2
DFDX(1OHNO3,1ONP4) = DRDX(1OHNO3,1ONP4) + 5.00*DXNP4
DFDX(1OHNO3,1ONP5) = DRDX(1OHNO3,1ONP5) + 5.00*DXNP5
C
REAC(1OHNO2) = REAC(1OHNO2) + 1.00*RR
DFDX(1OHNO2,1OHNO2) = DRDX(1OHNO2,1OHNO2) + 1.00*DXNO2
DFDX(1OHNO2,1ONP4) = DRDX(1OHNO2,1ONP4) + 1.00*DXNP4
DFDX(1OHNO2,1ONP5) = DRDX(1OHNO2,1ONP5) + 1.00*DXNP5
C
REAC(1ONP4) = REAC(1ONP4) - 2.00*RR
DFDX(1ONP4,1OHNO2) = DRDX(1ONP4,1OHNO2) - 2.00*DXNO2
DFDX(1ONP4,1ONP4) = DRDX(1ONP4,1ONP4) - 2.00*DXNP4
DFDX(1ONP4,1ONP5) = DRDX(1ONP4,1ONP5) - 2.00*DXNP5
C
REAC(1ONP5) = REAC(1ONP5) + 2.00*RR
DFDX(1ONP5,1OHNO2) = DRDX(1ONP5,1OHNO2) + 2.00*DXNO2
DFDX(1ONP5,1ONP4) = DRDX(1ONP5,1ONP4) + 2.00*DXNP4
DFDX(1ONP5,1ONP5) = DRDX(1ONP5,1ONP5) + 2.00*DXNP5
C
C
C... (22) REDUCTION OF NP(VI) BY PU(III)
C FROM EXTRA.M-CODE EQ-(2.5-22)
C
CK = ECHEM(22)*FCHEM(22)*CK22
C
RR = CK*XPU3*XNP6*(2.13D3+1.86D2/ZH)
DXH = -CK*XPU3*XNP6* 1.86D2/ZH/ZH
DXPU3 = CK* XNP6*(2.13D3+1.86D2/ZH)
DXNP6 = CK*XPU3 *(2.13D3+1.86D2/ZH)

```

C
REAC(10PU4) = REAC(10PU4) + RR
DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) + DXH
DRDX(10PU4 ,10PU3) = DRDX(10PU4 ,10PU3) + DXPU3
DRDX(10PU4 ,10NP6) = DRDX(10PU4 ,10NP6) + DXNP6
C
REAC(10PU3) = REAC(10PU3) - RR
DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) - DXH
DRDX(10PU3 ,10PU3) = DRDX(10PU3 ,10PU3) - DXPU3
DRDX(10PU3 ,10NP6) = DRDX(10PU3 ,10NP6) - DXNP6
C
REAC(10NP5) = REAC(10NP5) + RR
DRDX(10NP5 ,10HN03) = DRDX(10NP5 ,10HN03) + DXH
DRDX(10NP5 ,10PU3) = DRDX(10NP5 ,10PU3) + DXPU3
DRDX(10NP5 ,10NP6) = DRDX(10NP5 ,10NP6) + DXNP6
C
REAC(10NP6) = REAC(10NP6) - RR
DRDX(10NP6 ,10HN03) = DRDX(10NP6 ,10HN03) - DXH
DRDX(10NP6 ,10PU3) = DRDX(10NP6 ,10PU3) - DXPU3
DRDX(10NP6 ,10NP6) = DRDX(10NP6 ,10NP6) - DXNP6
C
C
C... (23) REDUCTION OF NP(V) BY PU(III)
FROM EXTRA.M-CODE EQ-(2.5-23)
C
CK = ECHEM(23)*FCHEM(23)*CK23
C
RR = CK*NP5*XP3*(ZH**(-1.300))
DXH = -1.300*CK*NP5*XP3*(ZH**(-2.300))
DXPU3 = CK*NP5 *(ZH**(-1.300))
DXNP5 = CK *XP3*(ZH**(-1.300))
C
REAC(10HN03) = REAC(10HN03) - 4.000*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 4.000*DXH
DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) - 4.000*DXPU3
DRDX(10HN03,10NP5) = DRDX(10HN03,10NP5) - 4.000*DXNP5
C
REAC(10PU4) = REAC(10PU4) + RR
DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) + DXH
DRDX(10PU4 ,10PU3) = DRDX(10PU4 ,10PU3) + DXPU3
DRDX(10PU4 ,10NP5) = DRDX(10PU4 ,10NP5) + DXNP5
C
REAC(10PU3) = REAC(10PU3) - RR
DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) - DXH
DRDX(10PU3 ,10PU3) = DRDX(10PU3 ,10PU3) - DXPU3
DRDX(10PU3 ,10NP5) = DRDX(10PU3 ,10NP5) - DXNP5
C
REAC(10NP4) = REAC(10NP4) + RR
DRDX(10NP4 ,10HN03) = DRDX(10NP4 ,10HN03) + DXH
DRDX(10NP4 ,10PU3) = DRDX(10NP4 ,10PU3) + DXPU3
DRDX(10NP4 ,10NP5) = DRDX(10NP4 ,10NP5) + DXNP5
C
REAC(10NP5) = REAC(10NP5) - RR
DRDX(10NP5 ,10HN03) = DRDX(10NP5 ,10HN03) - DXH
DRDX(10NP5 ,10PU3) = DRDX(10NP5 ,10PU3) - DXPU3
DRDX(10NP5 ,10NP5) = DRDX(10NP5 ,10NP5) - DXNP5
C
C
C... (24) REDUCTION OF PU(VI) BY HYDRAZINE
FROM EXTRA.M-CODE EQ-(2.5-24)
C
CK = ECHEM(24)*FCHEM(24)*CK24
C
RR = CK*XP6*XHYD/ZH
DXH = - CK*XP6*XHYD/ZH/ZH
DXPU6 = CK *XHYD/ZH
DXHYD = CK*XP6 /ZH
C
REAC(10PU6) = REAC(10PU6) - RR
DRDX(10PU6 ,10HN03) = DRDX(10PU6 ,10HN03) - DXH
DRDX(10PU6 ,10PU6) = DRDX(10PU6 ,10PU6) - DXPU6
DRDX(10PU6 ,10HYD) = DRDX(10PU6 ,10HYD) - DXHYD
C
REAC(10HYD) = REAC(10HYD) - RR
DRDX(10HYD ,10HN03) = DRDX(10HYD ,10HN03) - DXH
DRDX(10HYD ,10PU6) = DRDX(10HYD ,10PU6) - DXPU6
DRDX(10HYD ,10HYD) = DRDX(10HYD ,10HYD) - DXHYD
C
REAC(10PU5) = REAC(10PU5) + RR
DRDX(10PU5 ,10HN03) = DRDX(10PU5 ,10HN03) + DXH
DRDX(10PU5 ,10PU6) = DRDX(10PU5 ,10PU6) + DXPU6
DRDX(10PU5 ,10HYD) = DRDX(10PU5 ,10HYD) + DXHYD
C
REAC(10HN03) = REAC(10HN03) + RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DXH
DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + DXPU6
DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + DXHYD
C
C
C

C... (25) DISPROPORTIONATION OF PU(IV) BY H2O
FROM EXTRA.M-CODE EQ-(2.5-25)
C
CK = ECHEM(25)*FCHEM(25)*CK25
C
RR = CK*XP4*XP4*(ZH**(-2.200))*(ZCN**(-0.5400))
DXH = - 2.200 *CK*XP4*XP4*(ZH**(-3.200))*(ZCN**(-0.5400))
DXPU4 = 2.000 *CK *XP4*(ZH**(-2.200))*(ZCN**(-0.5400))
DCN = - 0.5400*CK*XP4*XP4*(ZH**(-2.200))*(ZCN**(-1.5400))
C
REAC(10HN03) = REAC(10HN03) + 4.000*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.000*DXH
DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 4.000*DXPU4
C
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.000*DCN*DCXH
DRDX(10HN03,10U6) = DRDX(10HN03,10U6) + 4.000*DCN*DCXU6
DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 4.000*DCN*DCXPU4
DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) + 4.000*DCN*DCXPU3
DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 4.000*DCN*DCXU4
DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + 4.000*DCN*DCXHYD
DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + 4.000*DCN*DCXHAN
DRDX(10HN03,10PU5) = DRDX(10HN03,10PU5) + 4.000*DCN*DCXPU5
DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 4.000*DCN*DCXPU6
DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.000*DCN*DCXNP4
DRDX(10HN03,10NP5) = DRDX(10HN03,10NP5) + 4.000*DCN*DCXNP5
DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.000*DCN*DCXNP6
C
C
REAC(10PU4) = REAC(10PU4) - 2.000*RR
DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) - 2.000*DXH
DRDX(10PU4 ,10HN02) = DRDX(10PU4 ,10HN02) - 2.000*DXNO2
DRDX(10PU4 ,10PU4) = DRDX(10PU4 ,10PU4) - 2.000*DXPU4
C
DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) - 2.000*DCN*DCXH
DRDX(10PU4 ,10U6) = DRDX(10PU4 ,10U6) - 2.000*DCN*DCXU6
DRDX(10PU4 ,10PU4) = DRDX(10PU4 ,10PU4) - 2.000*DCN*DCXPU4
DRDX(10PU4 ,10PU3) = DRDX(10PU4 ,10PU3) - 2.000*DCN*DCXPU3
DRDX(10PU4 ,10U4) = DRDX(10PU4 ,10U4) - 2.000*DCN*DCXU4
DRDX(10PU4 ,10HYD) = DRDX(10PU4 ,10HYD) - 2.000*DCN*DCXHYD
DRDX(10PU4 ,10HAN) = DRDX(10PU4 ,10HAN) - 2.000*DCN*DCXHAN
DRDX(10PU4 ,10PU5) = DRDX(10PU4 ,10PU5) - 2.000*DCN*DCXPU5
DRDX(10PU4 ,10PU6) = DRDX(10PU4 ,10PU6) - 2.000*DCN*DCXPU6
DRDX(10PU4 ,10NP4) = DRDX(10PU4 ,10NP4) - 2.000*DCN*DCXNP4
DRDX(10PU4 ,10NP5) = DRDX(10PU4 ,10NP5) - 2.000*DCN*DCXNP5
DRDX(10PU4 ,10NP6) = DRDX(10PU4 ,10NP6) - 2.000*DCN*DCXNP6
C
C
REAC(10PU3) = REAC(10PU3) + RR
DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) + DXH
DRDX(10PU3 ,10HN02) = DRDX(10PU3 ,10HN02) + DXNO2
DRDX(10PU3 ,10PU4) = DRDX(10PU3 ,10PU4) + DXPU4
C
DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) + DCN*DCXH
DRDX(10PU3 ,10U6) = DRDX(10PU3 ,10U6) + DCN*DCXU6
DRDX(10PU3 ,10PU4) = DRDX(10PU3 ,10PU4) + DCN*DCXPU4
DRDX(10PU3 ,10PU3) = DRDX(10PU3 ,10PU3) + DCN*DCXPU3
DRDX(10PU3 ,10U4) = DRDX(10PU3 ,10U4) + DCN*DCXU4
DRDX(10PU3 ,10HYD) = DRDX(10PU3 ,10HYD) + DCN*DCXHYD
DRDX(10PU3 ,10HAN) = DRDX(10PU3 ,10HAN) + DCN*DCXHAN
DRDX(10PU3 ,10PU5) = DRDX(10PU3 ,10PU5) + DCN*DCXPU5
DRDX(10PU3 ,10PU6) = DRDX(10PU3 ,10PU6) + DCN*DCXPU6
DRDX(10PU3 ,10NP4) = DRDX(10PU3 ,10NP4) + DCN*DCXNP4
DRDX(10PU3 ,10NP5) = DRDX(10PU3 ,10NP5) + DCN*DCXNP5
DRDX(10PU3 ,10NP6) = DRDX(10PU3 ,10NP6) + DCN*DCXNP6
C
REAC(10PU5) = REAC(10PU5) + RR
DRDX(10PU5 ,10HN03) = DRDX(10PU5 ,10HN03) + DXH
DRDX(10PU5 ,10HN02) = DRDX(10PU5 ,10HN02) + DXNO2
DRDX(10PU5 ,10PU4) = DRDX(10PU5 ,10PU4) + DXPU4
C
DRDX(10PU5 ,10HN03) = DRDX(10PU5 ,10HN03) + DCN*DCXH
DRDX(10PU5 ,10U6) = DRDX(10PU5 ,10U6) + DCN*DCXU6
DRDX(10PU5 ,10PU4) = DRDX(10PU5 ,10PU4) + DCN*DCXPU4
DRDX(10PU5 ,10PU3) = DRDX(10PU5 ,10PU3) + DCN*DCXPU3
DRDX(10PU5 ,10U4) = DRDX(10PU5 ,10U4) + DCN*DCXU4
DRDX(10PU5 ,10HYD) = DRDX(10PU5 ,10HYD) + DCN*DCXHYD
DRDX(10PU5 ,10HAN) = DRDX(10PU5 ,10HAN) + DCN*DCXHAN
DRDX(10PU5 ,10PU5) = DRDX(10PU5 ,10PU5) + DCN*DCXPU5
DRDX(10PU5 ,10PU6) = DRDX(10PU5 ,10PU6) + DCN*DCXPU6
DRDX(10PU5 ,10NP4) = DRDX(10PU5 ,10NP4) + DCN*DCXNP4
DRDX(10PU5 ,10NP5) = DRDX(10PU5 ,10NP5) + DCN*DCXNP5
DRDX(10PU5 ,10NP6) = DRDX(10PU5 ,10NP6) + DCN*DCXNP6
C
C
C... (26) DISPROPORTIONATION OF PU(V) BY NITRIC ACID
FROM EXTRA.M-CODE EQ-(2.5-26)
C
CK = ECHEM(26)*FCHEM(26)*CK26
C
C

```

RR = CK*XPUS*XPUS*XH
DXH = CK*XPUS*XPUS
DXPUS = 2.0D0*CK*XPUS *XH
C
REAC(10HN03) = REAC(10HN03) - 4.0D0*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 4.0D0*DXH
DRDX(10HN03,10PU5) = DRDX(10HN03,10PU5) - 4.0D0*DXPUS
C
REAC(10PU5) = REAC(10PU5) - 2.0D0*RR
DRDX(10PU5,10HN03) = DRDX(10PU5,10HN03) - 2.0D0*DXH
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - 2.0D0*DXPUS
C
REAC(10PU4) = REAC(10PU4) + RR
DRDX(10PU4,10HN03) = DRDX(10PU4,10HN03) + DXH
DRDX(10PU4,10PU5) = DRDX(10PU4,10PU5) + DXPUS
C
REAC(10PU6) = REAC(10PU6) + RR
DRDX(10PU6,10HN03) = DRDX(10PU6,10HN03) + DXH
DRDX(10PU6,10PU5) = DRDX(10PU6,10PU5) + DXPUS
C
C... (14) REDUCTION OF PU(VI) BY HN02
FROM EXTRA.M-CODE EQ-(2.5-27)
C
CK = ECHEM(14)*FCHEM(14)*CK14
C
RR = CK*XPUG*XPUG/ZH
DXH = - CK*XPUG*XPUG/ZH/ZH
DXPUG = CK *XPUG/ZH
DXN02 = CK*XPUG /ZH
C
REAC(10PU6) = REAC(10PU6) - 2.0D0*RR
DRDX(10PU6,10HN03) = DRDX(10PU6,10HN03) - 2.0D0*DXH
DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) - 2.0D0*DXPUG
DRDX(10PU6,10HN02) = DRDX(10PU6,10HN02) - 2.0D0*DXN02
C
REAC(10HN02) = REAC(10HN02) - 1.0D0*RR
DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) - 1.0D0*DXH
DRDX(10HN02,10PU6) = DRDX(10HN02,10PU6) - 1.0D0*DXPUG
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) - 1.0D0*DXN02
C
REAC(10PU5) = REAC(10PU5) + 2.0D0*RR
DRDX(10PU5,10HN03) = DRDX(10PU5,10HN03) + 2.0D0*DXH
DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) + 2.0D0*DXPUG
DRDX(10PU5,10HN02) = DRDX(10PU5,10HN02) + 2.0D0*DXN02
C
REAC(10HN03) = REAC(10HN03) + 3.0D0*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 3.0D0*DXH
DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 3.0D0*DXPUG
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + 3.0D0*DXN02
C
C... (28) REDUCTION OF PU(VI) BY HAN
FROM EXTRA.M-CODE EQ-(2.5-28)
C
CK = ECHEM(28)*FCHEM(28)*CK28
C
RR = CK*XPUG*XHAN/ZH
DXH = - CK*XPUG*XHAN/ZH/ZH
DXPUG = CK *XHAN/ZH
DXHAN = CK*XPUG /ZH
C
REAC(10PU6) = REAC(10PU6) - 2.0D0*RR
DRDX(10PU6,10HN03) = DRDX(10PU6,10HN03) - 2.0D0*DXH
DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) - 2.0D0*DXPUG
DRDX(10PU6,10HAN) = DRDX(10PU6,10HAN) - 2.0D0*DXHAN
C
REAC(10HAN) = REAC(10HAN) - 2.0D0*RR
DRDX(10HAN,10HN03) = DRDX(10HAN,10HN03) - 2.0D0*DXH
DRDX(10HAN,10PU6) = DRDX(10HAN,10PU6) - 2.0D0*DXPUG
DRDX(10HAN,10HAN) = DRDX(10HAN,10HAN) - 2.0D0*DXHAN
C
REAC(10PU5) = REAC(10PU5) + 2.0D0*RR
DRDX(10PU5,10HN03) = DRDX(10PU5,10HN03) + 2.0D0*DXH
DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) + 2.0D0*DXPUG
DRDX(10PU5,10HAN) = DRDX(10PU5,10HAN) + 2.0D0*DXHAN
C
REAC(10HN03) = REAC(10HN03) + 4.0D0*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.0D0*DXH
DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 4.0D0*DXPUG
DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + 4.0D0*DXHAN
C
C... (29) REDUCTION OF PU(V) BY HAN AND NITRIC ACID
FROM EXTRA.M-CODE EQ-(2.5-29)
C
CK = ECHEM(29)*FCHEM(29)*CK29
C
RR = CK*XPUS*XHAN

```

```

DXPUS = CK* XHAN
DXHAN = CK*XPUS
C
REAC(10HN03) = REAC(10HN03) - 2.0D0*RR
DRDX(10HN03,10PU5) = DRDX(10HN03,10PU5) - 2.0D0*DXPUS
DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) - 2.0D0*DXHAN
C
REAC(10PU5) = REAC(10PU5) - 2.0D0*RR
DRDX(10PU5,10PU5) = DRDX(10PU5,10PU5) - 2.0D0*DXPUS
DRDX(10PU5,10HAN) = DRDX(10PU5,10HAN) - 2.0D0*DXHAN
C
REAC(10HAN) = REAC(10HAN) - 2.0D0*RR
DRDX(10HAN,10PU5) = DRDX(10HAN,10PU5) - 2.0D0*DXPUS
DRDX(10HAN,10HAN) = DRDX(10HAN,10HAN) - 2.0D0*DXHAN
C
REAC(10PU3) = REAC(10PU3) + 2.0D0*RR
DRDX(10PU3,10PU5) = DRDX(10PU3,10PU5) + 2.0D0*DXPUS
DRDX(10PU3,10HAN) = DRDX(10PU3,10HAN) + 2.0D0*DXHAN
C
C... (30) REDUCTION OF PU(VI) BY URANOUS
FROM EXTRA.M-CODE EQ-(2.5-30)
C
CK = ECHEM(30)*FCHEM(30)*CK30
C
RR = CK*XPUG*XU4/(ZH**1.2D0)
DXH = - 1.2D0*CK*XPUG*XU4/(ZH**2.2D0)
DXPUG = CK *XU4/(ZH**1.2D0)
DXU4 = CK*XPUG /ZH**1.2D0
C
REAC(10PU6) = REAC(10PU6) - 2.0D0*RR
DRDX(10PU6,10HN03) = DRDX(10PU6,10HN03) - 2.0D0*DXH
DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) - 2.0D0*DXPUG
DRDX(10PU6,10U4) = DRDX(10PU6,10U4) - 2.0D0*DXU4
C
REAC(10U4) = REAC(10U4) - 1.0D0*RR
DRDX(10U4,10HN03) = DRDX(10U4,10HN03) - 1.0D0*DXH
DRDX(10U4,10PU6) = DRDX(10U4,10PU6) - 1.0D0*DXPUG
DRDX(10U4,10U4) = DRDX(10U4,10U4) - 1.0D0*DXU4
C
REAC(10U6) = REAC(10U6) + 1.0D0*RR
DRDX(10U6,10HN03) = DRDX(10U6,10HN03) + 1.0D0*DXH
DRDX(10U6,10PU6) = DRDX(10U6,10PU6) + 1.0D0*DXPUG
DRDX(10U6,10U4) = DRDX(10U6,10U4) + 1.0D0*DXU4
C
REAC(10PU5) = REAC(10PU5) + 2.0D0*RR
DRDX(10PU5,10HN03) = DRDX(10PU5,10HN03) + 2.0D0*DXH
DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) + 2.0D0*DXPUG
DRDX(10PU5,10U4) = DRDX(10PU5,10U4) + 2.0D0*DXU4
C
REAC(10HN03) = REAC(10HN03) + 4.0D0*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.0D0*DXH
DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 4.0D0*DXPUG
DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 4.0D0*DXU4
C
C... (31) DISINTEGRATION OF HYDRAZINE BY TC(VII)
FROM EXTRA.M-CODE EQ-(2.5-31)
C
CK = ECHEM(31)*FCHEM(31)*CK31
C
RR = CK*XTC7*XHYD
DXHYD = CK*XTC7
DXTC7 = CK *XHYD
C
REAC(10HYD) = REAC(10HYD) - RR
DRDX(10HYD,10HYD) = DRDX(10HYD,10HYD) - DXHYD
DRDX(10HYD,10TC7) = DRDX(10HYD,10TC7) - DXTC7
C
REAC(10TC7) = REAC(10TC7) - RR
DRDX(10TC7,10HYD) = DRDX(10TC7,10HYD) - DXHYD
DRDX(10TC7,10TC7) = DRDX(10TC7,10TC7) - DXTC7
C
REAC(10TC6) = REAC(10TC6) + RR
DRDX(10TC6,10HYD) = DRDX(10TC6,10HYD) + DXHYD
DRDX(10TC6,10TC7) = DRDX(10TC6,10TC7) + DXTC7
C
C... (32) DISINTEGRATION OF HYDRAZINE BY TC(VI)
FROM EXTRA.M-CODE EQ-(2.5-32)
C
CK = ECHEM(32)*FCHEM(32)*CK32
C
RR = CK*XTC6*XHYD
DXHYD = CK*XTC6
DXTC6 = CK *XHYD
C
REAC(10HYD) = REAC(10HYD) - RR
DRDX(10HYD,10HYD) = DRDX(10HYD,10HYD) - DXHYD

```

C DRDX(10HYD ,10TC6) = DRDX(10HYD ,10TC6) - DXTC6
 REAC(10TC6) = REAC(10TC6) - RR
 DRDX(10TC6 ,10HYD) = DRDX(10TC6 ,10HYD) - DXHYD
 DRDX(10TC6 ,10TC6) = DRDX(10TC6 ,10TC6) - DXTC6

C REAC(10TC5) = REAC(10TC5) + RR
 DRDX(10TC5 ,10HYD) = DRDX(10TC5 ,10HYD) + DXHYD
 DRDX(10TC5 ,10TC6) = DRDX(10TC5 ,10TC6) + DXTC6

C... (33) DISINTEGRATION OF HYDRAZINE BY TC(V)
 FROM EXTRA.M-CODE EQ-(2.5-33)

C CK = ECHEM(33)*FCHEM(33)*CK33
 RR = CK*XTC5*XHYD
 DXHYD = CK*XTC5
 DXTC5 = CK *XHYD

C REAC(10HYD) = REAC(10HYD) - RR
 DRDX(10HYD ,10HYD) = DRDX(10HYD ,10HYD) - DXHYD
 DRDX(10HYD ,10TC5) = DRDX(10HYD ,10TC5) - DXTC5

C REAC(10TC5) = REAC(10TC5) - RR
 DRDX(10TC5 ,10HYD) = DRDX(10TC5 ,10HYD) - DXHYD
 DRDX(10TC5 ,10TC5) = DRDX(10TC5 ,10TC5) - DXTC5

C REAC(10TC4) = REAC(10TC4) + RR
 DRDX(10TC4 ,10HYD) = DRDX(10TC4 ,10HYD) + DXHYD
 DRDX(10TC4 ,10TC5) = DRDX(10TC4 ,10TC5) + DXTC5

C... (34) REOXYDATION OF TC(IV) BY NITRIC ACID
 FROM EXTRA.M-CODE EQ-(2.5-34)

C CK = ECHEM(34)*FCHEM(34)*CK34
 RR = CK*XTC4*XCN
 DXTC4 = CK *XCN
 DCN = CK*XTC4

C REAC(10HNO3) = REAC(10HNO3) - RR
 DRDX(10HNO3,10TC4) = DRDX(10HNO3,10TC4) - DXTC4

C DRDX(10HNO3,10HNO3) = DRDX(10HNO3,10HNO3) - DCN*DCXH
 DRDX(10HNO3,10U6) = DRDX(10HNO3,10U6) - DCN*DCXU6
 DRDX(10HNO3,10PU4) = DRDX(10HNO3,10PU4) - DCN*DCXPU4
 DRDX(10HNO3,10PU3) = DRDX(10HNO3,10PU3) - DCN*DCXPU3
 DRDX(10HNO3,10U4) = DRDX(10HNO3,10U4) - DCN*DCXU4
 DRDX(10HNO3,10HYD) = DRDX(10HNO3,10HYD) - DCN*DCXHYD
 DRDX(10HNO3,10HAN) = DRDX(10HNO3,10HAN) - DCN*DCXHAN
 DRDX(10HNO3,10PU5) = DRDX(10HNO3,10PU5) - DCN*DCXPU5
 DRDX(10HNO3,10PU6) = DRDX(10HNO3,10PU6) - DCN*DCXPU6
 DRDX(10HNO3,10NP4) = DRDX(10HNO3,10NP4) - DCN*DCXNP4
 DRDX(10HNO3,10NP5) = DRDX(10HNO3,10NP5) - DCN*DCXNP5
 DRDX(10HNO3,10NP6) = DRDX(10HNO3,10NP6) - DCN*DCXNP6

C REAC(10HNO2) = REAC(10HNO2) + RR
 DRDX(10HNO2,10TC4) = DRDX(10HNO2,10TC4) + DXTC4

C DRDX(10HNO2,10HNO3) = DRDX(10HNO2,10HNO3) + DCN*DCXH
 DRDX(10HNO2,10U6) = DRDX(10HNO2,10U6) + DCN*DCXU6
 DRDX(10HNO2,10PU4) = DRDX(10HNO2,10PU4) + DCN*DCXPU4
 DRDX(10HNO2,10PU3) = DRDX(10HNO2,10PU3) + DCN*DCXPU3
 DRDX(10HNO2,10U4) = DRDX(10HNO2,10U4) + DCN*DCXU4
 DRDX(10HNO2,10HYD) = DRDX(10HNO2,10HYD) + DCN*DCXHYD
 DRDX(10HNO2,10HAN) = DRDX(10HNO2,10HAN) + DCN*DCXHAN
 DRDX(10HNO2,10PU5) = DRDX(10HNO2,10PU5) + DCN*DCXPU5
 DRDX(10HNO2,10PU6) = DRDX(10HNO2,10PU6) + DCN*DCXPU6
 DRDX(10HNO2,10NP4) = DRDX(10HNO2,10NP4) + DCN*DCXNP4
 DRDX(10HNO2,10NP5) = DRDX(10HNO2,10NP5) + DCN*DCXNP5
 DRDX(10HNO2,10NP6) = DRDX(10HNO2,10NP6) + DCN*DCXNP6

C REAC(10TC4) = REAC(10TC4) - RR
 DRDX(10TC4 ,10TC4) = DRDX(10TC4 ,10TC4) - DXTC4

C DRDX(10TC4 ,10HNO3) = DRDX(10TC4 ,10HNO3) - DCN*DCXH
 DRDX(10TC4 ,10U6) = DRDX(10TC4 ,10U6) - DCN*DCXU6
 DRDX(10TC4 ,10PU4) = DRDX(10TC4 ,10PU4) - DCN*DCXPU4
 DRDX(10TC4 ,10PU3) = DRDX(10TC4 ,10PU3) - DCN*DCXPU3
 DRDX(10TC4 ,10U4) = DRDX(10TC4 ,10U4) - DCN*DCXU4
 DRDX(10TC4 ,10HYD) = DRDX(10TC4 ,10HYD) - DCN*DCXHYD
 DRDX(10TC4 ,10HAN) = DRDX(10TC4 ,10HAN) - DCN*DCXHAN
 DRDX(10TC4 ,10PU5) = DRDX(10TC4 ,10PU5) - DCN*DCXPU5
 DRDX(10TC4 ,10PU6) = DRDX(10TC4 ,10PU6) - DCN*DCXPU6

DRDX(10TC4 ,10NP4) = DRDX(10TC4 ,10NP4) - DCN*DCXNP4
 DRDX(10TC4 ,10NP5) = DRDX(10TC4 ,10NP5) - DCN*DCXNP5
 DRDX(10TC4 ,10NP6) = DRDX(10TC4 ,10NP6) - DCN*DCXNP6

C REAC(10TC5) = REAC(10TC5) + RR
 DRDX(10TC5 ,10TC4) = DRDX(10TC5 ,10TC4) + DXTC4

C DRDX(10TC5 ,10HNO3) = DRDX(10TC5 ,10HNO3) + DCN*DCXH
 DRDX(10TC5 ,10U6) = DRDX(10TC5 ,10U6) + DCN*DCXU6
 DRDX(10TC5 ,10PU4) = DRDX(10TC5 ,10PU4) + DCN*DCXPU4
 DRDX(10TC5 ,10PU3) = DRDX(10TC5 ,10PU3) + DCN*DCXPU3
 DRDX(10TC5 ,10U4) = DRDX(10TC5 ,10U4) + DCN*DCXU4
 DRDX(10TC5 ,10HYD) = DRDX(10TC5 ,10HYD) + DCN*DCXHYD
 DRDX(10TC5 ,10HAN) = DRDX(10TC5 ,10HAN) + DCN*DCXHAN
 DRDX(10TC5 ,10PU5) = DRDX(10TC5 ,10PU5) + DCN*DCXPU5
 DRDX(10TC5 ,10PU6) = DRDX(10TC5 ,10PU6) + DCN*DCXPU6
 DRDX(10TC5 ,10NP4) = DRDX(10TC5 ,10NP4) + DCN*DCXNP4
 DRDX(10TC5 ,10NP5) = DRDX(10TC5 ,10NP5) + DCN*DCXNP5
 DRDX(10TC5 ,10NP6) = DRDX(10TC5 ,10NP6) + DCN*DCXNP6

C... (35) REOXYDATION OF TC(V) BY NITRIC ACID
 FROM EXTRA.M-CODE EQ-(2.5-35)

C CK = ECHEM(35)*FCHEM(35)*CK35
 RR = CK*XTC5*XCN
 DXTC5 = CK *XCN
 DCN = CK*XTC5

C REAC(10HNO3) = REAC(10HNO3) - RR
 DRDX(10HNO3,10TC5) = DRDX(10HNO3,10TC5) - DXTC5

C DRDX(10HNO3,10HNO3) = DRDX(10HNO3,10HNO3) - DCN*DCXH
 DRDX(10HNO3,10U6) = DRDX(10HNO3,10U6) - DCN*DCXU6
 DRDX(10HNO3,10PU4) = DRDX(10HNO3,10PU4) - DCN*DCXPU4
 DRDX(10HNO3,10PU3) = DRDX(10HNO3,10PU3) - DCN*DCXPU3
 DRDX(10HNO3,10U4) = DRDX(10HNO3,10U4) - DCN*DCXU4
 DRDX(10HNO3,10HYD) = DRDX(10HNO3,10HYD) - DCN*DCXHYD
 DRDX(10HNO3,10HAN) = DRDX(10HNO3,10HAN) - DCN*DCXHAN
 DRDX(10HNO3,10PU5) = DRDX(10HNO3,10PU5) - DCN*DCXPU5
 DRDX(10HNO3,10PU6) = DRDX(10HNO3,10PU6) - DCN*DCXPU6
 DRDX(10HNO3,10NP4) = DRDX(10HNO3,10NP4) - DCN*DCXNP4
 DRDX(10HNO3,10NP5) = DRDX(10HNO3,10NP5) - DCN*DCXNP5
 DRDX(10HNO3,10NP6) = DRDX(10HNO3,10NP6) - DCN*DCXNP6

C REAC(10HNO2) = REAC(10HNO2) + RR
 DRDX(10HNO2,10TC5) = DRDX(10HNO2,10TC5) + DXTC5

C DRDX(10HNO2,10HNO3) = DRDX(10HNO2,10HNO3) + DCN*DCXH
 DRDX(10HNO2,10U6) = DRDX(10HNO2,10U6) + DCN*DCXU6
 DRDX(10HNO2,10PU4) = DRDX(10HNO2,10PU4) + DCN*DCXPU4
 DRDX(10HNO2,10PU3) = DRDX(10HNO2,10PU3) + DCN*DCXPU3
 DRDX(10HNO2,10U4) = DRDX(10HNO2,10U4) + DCN*DCXU4
 DRDX(10HNO2,10HYD) = DRDX(10HNO2,10HYD) + DCN*DCXHYD
 DRDX(10HNO2,10HAN) = DRDX(10HNO2,10HAN) + DCN*DCXHAN
 DRDX(10HNO2,10PU5) = DRDX(10HNO2,10PU5) + DCN*DCXPU5
 DRDX(10HNO2,10PU6) = DRDX(10HNO2,10PU6) + DCN*DCXPU6
 DRDX(10HNO2,10NP4) = DRDX(10HNO2,10NP4) + DCN*DCXNP4
 DRDX(10HNO2,10NP5) = DRDX(10HNO2,10NP5) + DCN*DCXNP5
 DRDX(10HNO2,10NP6) = DRDX(10HNO2,10NP6) + DCN*DCXNP6

C REAC(10TC5) = REAC(10TC5) - RR
 DRDX(10TC5 ,10TC5) = DRDX(10TC5 ,10TC5) - DXTC5

C DRDX(10TC5 ,10HNO3) = DRDX(10TC5 ,10HNO3) - DCN*DCXH
 DRDX(10TC5 ,10U6) = DRDX(10TC5 ,10U6) - DCN*DCXU6
 DRDX(10TC5 ,10PU4) = DRDX(10TC5 ,10PU4) - DCN*DCXPU4
 DRDX(10TC5 ,10PU3) = DRDX(10TC5 ,10PU3) - DCN*DCXPU3
 DRDX(10TC5 ,10U4) = DRDX(10TC5 ,10U4) - DCN*DCXU4
 DRDX(10TC5 ,10HYD) = DRDX(10TC5 ,10HYD) - DCN*DCXHYD
 DRDX(10TC5 ,10HAN) = DRDX(10TC5 ,10HAN) - DCN*DCXHAN
 DRDX(10TC5 ,10PU5) = DRDX(10TC5 ,10PU5) - DCN*DCXPU5
 DRDX(10TC5 ,10PU6) = DRDX(10TC5 ,10PU6) - DCN*DCXPU6
 DRDX(10TC5 ,10NP4) = DRDX(10TC5 ,10NP4) - DCN*DCXNP4
 DRDX(10TC5 ,10NP5) = DRDX(10TC5 ,10NP5) - DCN*DCXNP5
 DRDX(10TC5 ,10NP6) = DRDX(10TC5 ,10NP6) - DCN*DCXNP6

C REAC(10TC7) = REAC(10TC7) + RR
 DRDX(10TC7 ,10TC5) = DRDX(10TC7 ,10TC5) + DXTC5

C DRDX(10TC7 ,10HNO3) = DRDX(10TC7 ,10HNO3) + DCN*DCXH
 DRDX(10TC7 ,10U6) = DRDX(10TC7 ,10U6) + DCN*DCXU6
 DRDX(10TC7 ,10PU4) = DRDX(10TC7 ,10PU4) + DCN*DCXPU4

```

DRDX(10TC7 ,10PU3 ) = DRDX(10TC7 ,10PU3 ) + DCN*DCXPU3
DRDX(10TC7 ,10U4 ) = DRDX(10TC7 ,10U4 ) + DCN*DCXU4
DRDX(10TC7 ,10HYD ) = DRDX(10TC7 ,10HYD ) + DCN*DCXHYD
DRDX(10TC7 ,10HAN ) = DRDX(10TC7 ,10HAN ) + DCN*DCXHAN
DRDX(10TC7 ,10PU5 ) = DRDX(10TC7 ,10PU5 ) + DCN*DCXPU5
DRDX(10TC7 ,10PU6 ) = DRDX(10TC7 ,10PU6 ) + DCN*DCXPU6
DRDX(10TC7 ,10NP4 ) = DRDX(10TC7 ,10NP4 ) + DCN*DCXNP4
DRDX(10TC7 ,10NP5 ) = DRDX(10TC7 ,10NP5 ) + DCN*DCXNP5
DRDX(10TC7 ,10NP6 ) = DRDX(10TC7 ,10NP6 ) + DCN*DCXNP6

C
C
C... (36) REOXYDATION OF TC(VI) BY NITRIC ACID
C FROM EXTRA.M-CODE EQ-(2.5-36)
C
C CK = ECHEM(36)*FCHEM(36)*CK36
C
C RR = CK*XTC6*XCN
DXTC6 = CK *XCN
DCN = CK*XTC6
C
C REAC(10HNO3) = REAC(10HNO3) - RR
DRDX(10HNO3 ,10TC6 ) = DRDX(10HNO3 ,10TC6 ) - DXTC6
C
C DRDX(10HNO3 ,10HNO3 ) = DRDX(10HNO3 ,10HNO3 ) - DCN*DCXH
DRDX(10HNO3 ,10U6 ) = DRDX(10HNO3 ,10U6 ) - DCN*DCXU6
DRDX(10HNO3 ,10PU4 ) = DRDX(10HNO3 ,10PU4 ) - DCN*DCXPU4
DRDX(10HNO3 ,10PU3 ) = DRDX(10HNO3 ,10PU3 ) - DCN*DCXPU3
DRDX(10HNO3 ,10U4 ) = DRDX(10HNO3 ,10U4 ) - DCN*DCXU4
DRDX(10HNO3 ,10HYD ) = DRDX(10HNO3 ,10HYD ) - DCN*DCXHYD
DRDX(10HNO3 ,10HAN ) = DRDX(10HNO3 ,10HAN ) - DCN*DCXHAN
DRDX(10HNO3 ,10PU5 ) = DRDX(10HNO3 ,10PU5 ) - DCN*DCXPU5
DRDX(10HNO3 ,10PU6 ) = DRDX(10HNO3 ,10PU6 ) - DCN*DCXPU6
DRDX(10HNO3 ,10NP4 ) = DRDX(10HNO3 ,10NP4 ) - DCN*DCXNP4
DRDX(10HNO3 ,10NP5 ) = DRDX(10HNO3 ,10NP5 ) - DCN*DCXNP5
DRDX(10HNO3 ,10NP6 ) = DRDX(10HNO3 ,10NP6 ) - DCN*DCXNP6
C
C REAC(10HNO2) = REAC(10HNO2) + RR
DRDX(10HNO2 ,10TC6 ) = DRDX(10HNO2 ,10TC6 ) + DXTC6
C
C DRDX(10HNO2 ,10HNO3 ) = DRDX(10HNO2 ,10HNO3 ) + DCN*DCXH
DRDX(10HNO2 ,10U6 ) = DRDX(10HNO2 ,10U6 ) + DCN*DCXU6
DRDX(10HNO2 ,10PU4 ) = DRDX(10HNO2 ,10PU4 ) + DCN*DCXPU4
DRDX(10HNO2 ,10PU3 ) = DRDX(10HNO2 ,10PU3 ) + DCN*DCXPU3
DRDX(10HNO2 ,10U4 ) = DRDX(10HNO2 ,10U4 ) + DCN*DCXU4
DRDX(10HNO2 ,10HYD ) = DRDX(10HNO2 ,10HYD ) + DCN*DCXHYD
DRDX(10HNO2 ,10HAN ) = DRDX(10HNO2 ,10HAN ) + DCN*DCXHAN
DRDX(10HNO2 ,10PU5 ) = DRDX(10HNO2 ,10PU5 ) + DCN*DCXPU5
DRDX(10HNO2 ,10PU6 ) = DRDX(10HNO2 ,10PU6 ) + DCN*DCXPU6
DRDX(10HNO2 ,10NP4 ) = DRDX(10HNO2 ,10NP4 ) + DCN*DCXNP4
DRDX(10HNO2 ,10NP5 ) = DRDX(10HNO2 ,10NP5 ) + DCN*DCXNP5
DRDX(10HNO2 ,10NP6 ) = DRDX(10HNO2 ,10NP6 ) + DCN*DCXNP6
C
C REAC(10TC6 ) = REAC(10TC6 ) - RR
DRDX(10TC6 ,10TC6 ) = DRDX(10TC6 ,10TC6 ) - DXTC6
C
C DRDX(10TC6 ,10HNO3 ) = DRDX(10TC6 ,10HNO3 ) - DCN*DCXH
DRDX(10TC6 ,10U6 ) = DRDX(10TC6 ,10U6 ) - DCN*DCXU6
DRDX(10TC6 ,10PU4 ) = DRDX(10TC6 ,10PU4 ) - DCN*DCXPU4
DRDX(10TC6 ,10PU3 ) = DRDX(10TC6 ,10PU3 ) - DCN*DCXPU3
DRDX(10TC6 ,10U4 ) = DRDX(10TC6 ,10U4 ) - DCN*DCXU4
DRDX(10TC6 ,10HYD ) = DRDX(10TC6 ,10HYD ) - DCN*DCXHYD
DRDX(10TC6 ,10HAN ) = DRDX(10TC6 ,10HAN ) - DCN*DCXHAN
DRDX(10TC6 ,10PU5 ) = DRDX(10TC6 ,10PU5 ) - DCN*DCXPU5
DRDX(10TC6 ,10PU6 ) = DRDX(10TC6 ,10PU6 ) - DCN*DCXPU6
DRDX(10TC6 ,10NP4 ) = DRDX(10TC6 ,10NP4 ) - DCN*DCXNP4
DRDX(10TC6 ,10NP5 ) = DRDX(10TC6 ,10NP5 ) - DCN*DCXNP5
DRDX(10TC6 ,10NP6 ) = DRDX(10TC6 ,10NP6 ) - DCN*DCXNP6
C
C REAC(10TC7 ) = REAC(10TC7 ) + RR
DRDX(10TC7 ,10TC6 ) = DRDX(10TC7 ,10TC6 ) + DXTC6
C
C DRDX(10TC7 ,10HNO3 ) = DRDX(10TC7 ,10HNO3 ) + DCN*DCXH
DRDX(10TC7 ,10U6 ) = DRDX(10TC7 ,10U6 ) + DCN*DCXU6
DRDX(10TC7 ,10PU4 ) = DRDX(10TC7 ,10PU4 ) + DCN*DCXPU4
DRDX(10TC7 ,10PU3 ) = DRDX(10TC7 ,10PU3 ) + DCN*DCXPU3
DRDX(10TC7 ,10U4 ) = DRDX(10TC7 ,10U4 ) + DCN*DCXU4
DRDX(10TC7 ,10HYD ) = DRDX(10TC7 ,10HYD ) + DCN*DCXHYD
DRDX(10TC7 ,10HAN ) = DRDX(10TC7 ,10HAN ) + DCN*DCXHAN
DRDX(10TC7 ,10PU5 ) = DRDX(10TC7 ,10PU5 ) + DCN*DCXPU5
DRDX(10TC7 ,10PU6 ) = DRDX(10TC7 ,10PU6 ) + DCN*DCXPU6
DRDX(10TC7 ,10NP4 ) = DRDX(10TC7 ,10NP4 ) + DCN*DCXNP4
DRDX(10TC7 ,10NP5 ) = DRDX(10TC7 ,10NP5 ) + DCN*DCXNP5
DRDX(10TC7 ,10NP6 ) = DRDX(10TC7 ,10NP6 ) + DCN*DCXNP6
C
C
C

```

```

C... (37) REOXYDATION OF TC(IV) AND REDUCTION OF TC(VII)
C FROM EXTRA.M-CODE EQ-(2.5-37)
C
C CK = ECHEM(37)*FCHEM(37)*CK37
C
C RR = CK*XTC4*XTC7
DXTC4 = CK *XTC7
DXTC7 = CK*XTC4
C
C REAC(10TC4 ) = REAC(10TC4 ) - RR
DRDX(10TC4 ,10TC4 ) = DRDX(10TC4 ,10TC4 ) - DXTC4
DRDX(10TC4 ,10TC7 ) = DRDX(10TC4 ,10TC7 ) - DXTC7
C
C REAC(10TC7 ) = REAC(10TC7 ) - RR
DRDX(10TC7 ,10TC4 ) = DRDX(10TC7 ,10TC4 ) - DXTC4
DRDX(10TC7 ,10TC7 ) = DRDX(10TC7 ,10TC7 ) - DXTC7
C
C REAC(10TC5 ) = REAC(10TC5 ) + RR
DRDX(10TC5 ,10TC4 ) = DRDX(10TC5 ,10TC4 ) + DXTC4
DRDX(10TC5 ,10TC7 ) = DRDX(10TC5 ,10TC7 ) + DXTC7
C
C REAC(10TC6 ) = REAC(10TC6 ) + RR
DRDX(10TC6 ,10TC4 ) = DRDX(10TC6 ,10TC4 ) + DXTC4
DRDX(10TC6 ,10TC7 ) = DRDX(10TC6 ,10TC7 ) + DXTC7
C
C... (38) REDUCTION OF TC(VII) BY PU(III)
C FROM EXTRA.M-CODE EQ-(2.5-38)
C
C CK = ECHEM(38)*FCHEM(38)*CK38
C
C RR = CK*XPU3*XTC7*XH*XH*XH*XH
DXH = 4.000*CK*XPU3*XTC7*XH*XH*XH*XH
DXPU3 = CK *XTC7*XH*XH*XH*XH
DXTC7 = CK*XPU3 *XH*XH*XH*XH
C
C REAC(10HNO3) = REAC(10HNO3) - 4.000*RR
DRDX(10HNO3 ,10HNO3 ) = DRDX(10HNO3 ,10HNO3 ) - 4.000*DXH
DRDX(10HNO3 ,10PU3 ) = DRDX(10HNO3 ,10PU3 ) - 4.000*DXPU3
DRDX(10HNO3 ,10TC7 ) = DRDX(10HNO3 ,10TC7 ) - 4.000*DXTC7
C
C REAC(10PU3 ) = REAC(10PU3 ) - 3.000*RR
DRDX(10PU3 ,10HNO3 ) = DRDX(10PU3 ,10HNO3 ) - 3.000*DXH
DRDX(10PU3 ,10PU3 ) = DRDX(10PU3 ,10PU3 ) - 3.000*DXPU3
DRDX(10PU3 ,10TC7 ) = DRDX(10PU3 ,10TC7 ) - 3.000*DXTC7
C
C REAC(10PU4 ) = REAC(10PU4 ) + 3.000*RR
DRDX(10PU4 ,10HNO3 ) = DRDX(10PU4 ,10HNO3 ) + 3.000*DXH
DRDX(10PU4 ,10PU3 ) = DRDX(10PU4 ,10PU3 ) + 3.000*DXPU3
DRDX(10PU4 ,10TC7 ) = DRDX(10PU4 ,10TC7 ) + 3.000*DXTC7
C
C REAC(10TC7 ) = REAC(10TC7 ) - RR
DRDX(10TC7 ,10HNO3 ) = DRDX(10TC7 ,10HNO3 ) - DXH
DRDX(10TC7 ,10PU3 ) = DRDX(10TC7 ,10PU3 ) - DXPU3
DRDX(10TC7 ,10TC7 ) = DRDX(10TC7 ,10TC7 ) - DXTC7
C
C REAC(10TC4 ) = REAC(10TC4 ) + RR
DRDX(10TC4 ,10HNO3 ) = DRDX(10TC4 ,10HNO3 ) + DXH
DRDX(10TC4 ,10PU3 ) = DRDX(10TC4 ,10PU3 ) + DXPU3
DRDX(10TC4 ,10TC7 ) = DRDX(10TC4 ,10TC7 ) + DXTC7
C
C... (39) REDUCTION OF TC(VII) BY URANOUS
C FROM EXTRA.M-CODE EQ-(2.5-39)
C
C CK = ECHEM(39)*FCHEM(39)*CK39
C
C RR = CK*XU4*XTC7*XH*XH
DXU4 = 2.000*CK*XU4*XTC7*XH
DXTC7 = CK*XU4 *XH*XH
C
C REAC(10U4 ) = REAC(10U4 ) - 3.000*RR
DRDX(10U4 ,10HNO3 ) = DRDX(10U4 ,10HNO3 ) - 3.000*DXH
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - 3.000*DXU4
DRDX(10U4 ,10TC7 ) = DRDX(10U4 ,10TC7 ) - 3.000*DXTC7
C
C REAC(10U6 ) = REAC(10U6 ) + 3.000*RR
DRDX(10U6 ,10HNO3 ) = DRDX(10U6 ,10HNO3 ) + 3.000*DXH
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + 3.000*DXU4
DRDX(10U6 ,10TC7 ) = DRDX(10U6 ,10TC7 ) + 3.000*DXTC7
C
C REAC(10TC7 ) = REAC(10TC7 ) - 2.000*RR
DRDX(10TC7 ,10HNO3 ) = DRDX(10TC7 ,10HNO3 ) - 2.000*DXH
DRDX(10TC7 ,10U4 ) = DRDX(10TC7 ,10U4 ) - 2.000*DXU4
DRDX(10TC7 ,10TC7 ) = DRDX(10TC7 ,10TC7 ) - 2.000*DXTC7
C
C REAC(10TC4 ) = REAC(10TC4 ) + 2.000*RR
DRDX(10TC4 ,10HNO3 ) = DRDX(10TC4 ,10HNO3 ) + 2.000*DXH

```



```

DRDX(10TC4 ,10U4 ) = DRDX(10TC4 ,10U4 ) + 2.000*DXU4
DRDX(10TC4 ,10TC7 ) = DRDX(10TC4 ,10TC7 ) + 2.000*DXTC7
C
C
C... (40) REDUCTION OF TC(VII) BY HAN
C FROM EXTRA.M-CODE E0-(2.5-40)
C
CK = ECHEM(40)*FCHEM(40)*CK40
C
IF ( XHAN .GT. CK .AND. XTC7 .GT. CK ) THEN
RR = CK
C
REAC(10HAN ) = REAC(10HAN ) - RR
REAC(10TC7 ) = REAC(10TC7 ) - RR
REAC(10TC5 ) = REAC(10TC5 ) + RR
C
END IF
C
C... (41) OXYDATION OF U(IV) BY NITRIC ACID ( FROM R.MIXSET )
C
C41EPS = 1.00-10
C
ZNO2 = XNO2 + C41EPS
C
IF ( XH .LT. 0.8D0 ) THEN
C
CK = ECHEM(41)*FCHEM(41)*CK411
C
RR = CK*XU4*(ZNO2**0.38D0)*(XH**2.7D0)
DXH = 2.70D0*CK*XU4*(ZNO2**0.38D0)*(XH**1.7D0)
DXU4 = CK *(ZNO2**0.38D0)*(XH**2.7D0)
DXNO2 = 0.38D0*CK*XU4/(ZNO2**0.62D0)*(XH**2.7D0)
C
REAC(10HN03) = REAC(10HN03) + RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DXH
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + DXU4
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DXNO2
C
REAC(10U6 ) = REAC(10U6 ) + RR
DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + DXH
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DXU4
DRDX(10U6 ,10HN02) = DRDX(10U6 ,10HN02) + DXNO2
C
REAC(10U4 ) = REAC(10U4 ) - RR
DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - DXH
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DXU4
DRDX(10U4 ,10HN02) = DRDX(10U4 ,10HN02) - DXNO2
C
REAC(10HN02) = REAC(10HN02) + RR
DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) + DXH
DRDX(10HN02,10U4 ) = DRDX(10HN02,10U4 ) + DXU4
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + DXNO2
C
ELSE
C
CK = ECHEM(41)*FCHEM(41)*CK412
C
RR = CK*XU4*(ZNO2**0.38D0)
DXU4 = CK *(ZNO2**0.38D0)
DXNO2 = 0.38D0*CK*XU4/(ZNO2**0.62D0)
C
REAC(10HN03) = REAC(10HN03) + RR
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + DXU4
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DXNO2
C
REAC(10U6 ) = REAC(10U6 ) + RR
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DXU4
DRDX(10U6 ,10HN02) = DRDX(10U6 ,10HN02) + DXNO2
C
REAC(10U4 ) = REAC(10U4 ) - RR
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DXU4
DRDX(10U4 ,10HN02) = DRDX(10U4 ,10HN02) - DXNO2
C
REAC(10HN02) = REAC(10HN02) + RR
DRDX(10HN02,10U4 ) = DRDX(10HN02,10U4 ) + DXU4
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + DXNO2
C
END IF
C
C... (42) OXYDATION OF U(IV) BY OXYGEN ( FROM R.MIXSET )
C
CK = ECHEM(42)*FCHEM(42)*CK42
C
RR = CK*XU4/ZH
DXH = -CK*XU4/ZH/ZH

```

```

DXU4 = CK /ZH
C
C
REAC(10HN03) = REAC(10HN03) + 2.000*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 2.000*DXH
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + 2.000*DXU4
C
REAC(10U6 ) = REAC(10U6 ) + RR
DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + DXH
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DXU4
C
REAC(10U4 ) = REAC(10U4 ) - RR
DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - DXH
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DXU4
C
C... (43) REDUCTION OF NP(V) BY HYDRAZINE( FORM A.MIXSET)
C
CK = ECHEM(43)*FCHEM(43)*CK43
C
RR = CK*XNP5*XHYD*XH*XH*XH
DXH = 3.000*CK*XNP5*XHYD*XH*XH
DXHYD = CK*XNP5 *XH*XH*XH
DXNPS = CK *XHYD*XH*XH*XH
C
REAC(10HN03) = REAC(10HN03) - 4.000*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 4.000*DXH
DRDX(10HN03,10HYD ) = DRDX(10HN03,10HYD ) - 4.000*DXHYD
DRDX(10HN03,10NP5 ) = DRDX(10HN03,10NP5 ) - 4.000*DXNPS
C
REAC(10HYD ) = REAC(10HYD ) - RR
DRDX(10HYD ,10HN03) = DRDX(10HYD ,10HN03) - DXH
DRDX(10HYD ,10HYD ) = DRDX(10HYD ,10HYD ) - DXHYD
DRDX(10HYD ,10NP5 ) = DRDX(10HYD ,10NP5 ) - DXNPS
C
REAC(10NP4 ) = REAC(10NP4 ) + RR
DRDX(10NP4 ,10HN03) = DRDX(10NP4 ,10HN03) + DXH
DRDX(10NP4 ,10HYD ) = DRDX(10NP4 ,10HYD ) + DXHYD
DRDX(10NP4 ,10NP5 ) = DRDX(10NP4 ,10NP5 ) + DXNPS
C
REAC(10NP5 ) = REAC(10NP5 ) - RR
DRDX(10NP5 ,10HN03) = DRDX(10NP5 ,10HN03) - DXH
DRDX(10NP5 ,10HYD ) = DRDX(10NP5 ,10HYD ) - DXHYD
DRDX(10NP5 ,10NP5 ) = DRDX(10NP5 ,10NP5 ) - DXNPS
C
C... (44) REDUCTION OF NP(VI) BY HNO2
C FROM REACT-MOD CODE (56)
C
C44EPS = 1.00-25
ZNO2 = XNO2 + C44EPS
C
CK = ECHEM(44)*FCHEM(44)*CK44
C
FF = CK*XNO2*(XCN**1.5D0)
C
RR = FF*XNP6*XNO2/(ZNO2+XNP5)/ZH
DXH = -FF*XNP6*XNO2/(ZNO2+XNP5)/ZH/ZH
DXNO2 = 2.000*FF*XNP6 /ZNO2+XNP5)/ZH
*
DXNPS = -FF*XNP6*XNO2/(ZNO2+XNP5)/(ZNO2+XNP5)/ZH
DXNP6 = FF *XNO2/(ZNO2+XNP5)/ZH
DCN = 1.5D0*CK*XNP6*XNO2*XNO2*(XCN**0.5D0)/(ZNO2+XNP5)/ZH
C
REAC(10HN03) = REAC(10HN03) + 3.000*RR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 3.000*DXH
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + 3.000*DXNO2
DRDX(10HN03,10NP5 ) = DRDX(10HN03,10NP5 ) + 3.000*DXNPS
DRDX(10HN03,10NP6 ) = DRDX(10HN03,10NP6 ) + 3.000*DXNP6
C
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 3.000*DCN*DCXH
DRDX(10HN03,10U6 ) = DRDX(10HN03,10U6 ) + 3.000*DCN*DCXU6
DRDX(10HN03,10PU4 ) = DRDX(10HN03,10PU4 ) + 3.000*DCN*DCXPU4
DRDX(10HN03,10PU3 ) = DRDX(10HN03,10PU3 ) + 3.000*DCN*DCXPU3
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + 3.000*DCN*DCXU4
DRDX(10HN03,10HYD ) = DRDX(10HN03,10HYD ) + 3.000*DCN*DCXHYD
DRDX(10HN03,10HAN ) = DRDX(10HN03,10HAN ) + 3.000*DCN*DCXHAN
DRDX(10HN03,10PU5 ) = DRDX(10HN03,10PU5 ) + 3.000*DCN*DCXPU5
DRDX(10HN03,10PU6 ) = DRDX(10HN03,10PU6 ) + 3.000*DCN*DCXPU6
DRDX(10HN03,10NP4 ) = DRDX(10HN03,10NP4 ) + 3.000*DCN*DCXNP4
DRDX(10HN03,10NP5 ) = DRDX(10HN03,10NP5 ) + 3.000*DCN*DCXNP5
DRDX(10HN03,10NP6 ) = DRDX(10HN03,10NP6 ) + 3.000*DCN*DCXNP6
C
REAC(10HN02) = REAC(10HN02) - RR
DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) - DXH
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) - DXNO2
DRDX(10HN02,10NP5 ) = DRDX(10HN02,10NP5 ) - DXNPS
DRDX(10HN02,10NP6 ) = DRDX(10HN02,10NP6 ) - DXNP6

```

```

C
DRDX(10HNO2,10HNO3) = DRDX(10HNO2,10HNO3) - DCN*DCXH
DRDX(10HNO2,10U6) = DRDX(10HNO2,10U6) - DCN*DCXU6
DRDX(10HNO2,10PU4) = DRDX(10HNO2,10PU4) - DCN*DCXPU4
DRDX(10HNO2,10PU3) = DRDX(10HNO2,10PU3) - DCN*DCXPU3
DRDX(10HNO2,10U4) = DRDX(10HNO2,10U4) - DCN*DCXU4
DRDX(10HNO2,10HYD) = DRDX(10HNO2,10HYD) - DCN*DCXHYD
DRDX(10HNO2,10HAN) = DRDX(10HNO2,10HAN) - DCN*DCXHAN
DRDX(10HNO2,10PU5) = DRDX(10HNO2,10PU5) - DCN*DCXPU5
DRDX(10HNO2,10PU6) = DRDX(10HNO2,10PU6) - DCN*DCXPU6
DRDX(10HNO2,10NP4) = DRDX(10HNO2,10NP4) - DCN*DCXNP4
DRDX(10HNO2,10NP5) = DRDX(10HNO2,10NP5) - DCN*DCXNP5
DRDX(10HNO2,10NP6) = DRDX(10HNO2,10NP6) - DCN*DCXNP6

C
REAC(10NP5) = REAC(10NP5) + 2.000*RR
DRDX(10NP5,10HNO3) = DRDX(10NP5,10HNO3) + 2.000*DXH
DRDX(10NP5,10HNO2) = DRDX(10NP5,10HNO2) + 2.000*DXNO2
DRDX(10NP5,10NP5) = DRDX(10NP5,10NP5) + 2.000*DXNP5
DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) + 2.000*DXNP6

C
DRDX(10NP5,10HNO3) = DRDX(10NP5,10HNO3) + 2.000*DCN*DCXH
DRDX(10NP5,10U6) = DRDX(10NP5,10U6) + 2.000*DCN*DCXU6
DRDX(10NP5,10PU4) = DRDX(10NP5,10PU4) + 2.000*DCN*DCXPU4
DRDX(10NP5,10PU3) = DRDX(10NP5,10PU3) + 2.000*DCN*DCXPU3
DRDX(10NP5,10U4) = DRDX(10NP5,10U4) + 2.000*DCN*DCXU4
DRDX(10NP5,10HYD) = DRDX(10NP5,10HYD) + 2.000*DCN*DCXHYD
DRDX(10NP5,10HAN) = DRDX(10NP5,10HAN) + 2.000*DCN*DCXHAN
DRDX(10NP5,10PU5) = DRDX(10NP5,10PU5) + 2.000*DCN*DCXPU5
DRDX(10NP5,10PU6) = DRDX(10NP5,10PU6) + 2.000*DCN*DCXPU6
DRDX(10NP5,10NP4) = DRDX(10NP5,10NP4) + 2.000*DCN*DCXNP4
DRDX(10NP5,10NP5) = DRDX(10NP5,10NP5) + 2.000*DCN*DCXNP5
DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) + 2.000*DCN*DCXNP6

C
REAC(10NP6) = REAC(10NP6) - 2.000*RR
DRDX(10NP6,10HNO3) = DRDX(10NP6,10HNO3) - 2.000*DXH
DRDX(10NP6,10HNO2) = DRDX(10NP6,10HNO2) - 2.000*DXNO2
DRDX(10NP6,10NP5) = DRDX(10NP6,10NP5) - 2.000*DXNP5
DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - 2.000*DXNP6

C
DRDX(10NP6,10HNO3) = DRDX(10NP6,10HNO3) - 2.000*DCN*DCXH
DRDX(10NP6,10U6) = DRDX(10NP6,10U6) - 2.000*DCN*DCXU6
DRDX(10NP6,10PU4) = DRDX(10NP6,10PU4) - 2.000*DCN*DCXPU4
DRDX(10NP6,10PU3) = DRDX(10NP6,10PU3) - 2.000*DCN*DCXPU3
DRDX(10NP6,10U4) = DRDX(10NP6,10U4) - 2.000*DCN*DCXU4
DRDX(10NP6,10HYD) = DRDX(10NP6,10HYD) - 2.000*DCN*DCXHYD
DRDX(10NP6,10HAN) = DRDX(10NP6,10HAN) - 2.000*DCN*DCXHAN
DRDX(10NP6,10PU5) = DRDX(10NP6,10PU5) - 2.000*DCN*DCXPU5
DRDX(10NP6,10PU6) = DRDX(10NP6,10PU6) - 2.000*DCN*DCXPU6
DRDX(10NP6,10NP4) = DRDX(10NP6,10NP4) - 2.000*DCN*DCXNP4
DRDX(10NP6,10NP5) = DRDX(10NP6,10NP5) - 2.000*DCN*DCXNP5
DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - 2.000*DCN*DCXNP6

C
C... (45) OXIDATION OF PU(III) BY HNO2
BY PLUTONIUM HANDBOOK

C
CK = ECHEM(45)*FCHEM(45)*CK45

C
RR = CK*XPUS*XNO2*XH
DXH = CK*XPUS*XNO2
DXNO2 = CK*XPUS*XH
DXPU3 = CK*XNO2*XH

C
REAC(10HNO3) = REAC(10HNO3) - RR
DRDX(10HNO3,10HNO3) = DRDX(10HNO3,10HNO3) - DXH
DRDX(10HNO3,10HNO2) = DRDX(10HNO3,10HNO2) - DXNO2
DRDX(10HNO3,10PU3) = DRDX(10HNO3,10PU3) - DXPU3

C
REAC(10HNO2) = REAC(10HNO2) - RR
DRDX(10HNO2,10HNO3) = DRDX(10HNO2,10HNO3) - DXH
DRDX(10HNO2,10HNO2) = DRDX(10HNO2,10HNO2) - DXNO2
DRDX(10HNO2,10PU3) = DRDX(10HNO2,10PU3) - DXPU3

C
REAC(10PU3) = REAC(10PU3) - RR
DRDX(10PU3,10HNO3) = DRDX(10PU3,10HNO3) - DXH
DRDX(10PU3,10HNO2) = DRDX(10PU3,10HNO2) - DXNO2
DRDX(10PU3,10PU3) = DRDX(10PU3,10PU3) - DXPU3

C
REAC(10PU4) = REAC(10PU4) + RR
DRDX(10PU4,10HNO3) = DRDX(10PU4,10HNO3) + DXH
DRDX(10PU4,10HNO2) = DRDX(10PU4,10HNO2) + DXNO2
DRDX(10PU4,10PU3) = DRDX(10PU4,10PU3) + DXPU3

C
RETURN
END

SUBROUTINE REACTY(IREACT,NCHEM,T,TO,RCON,
* MXCOMP,REAC,DRDX,
* MXREAC,FCHEM,ECHEM,
* YH,YU6,YPU4,YPU3,YU4,YHAN,
* YHYD,YNO2,YPU5,YPU6,
* YTC4,YTC5,YTC6,YTC7,
* YNP4,YNP5,YNP6,Y,
* XHYD,RXHYD,DRYH,DRXHYD,DRYNO2)

C
C*****
C
C JACOBIAN CALCULATION OF CHEMICAL REACTION TERM IN ORGANIC PHASE
C*****
C
C IMPLICIT REAL*8 (A-H, O-Z)
C
C DIMENSION FCHEM (MXREAC)
C DIMENSION ECHEM (MXREAC)
C
C DIMENSION Y (MXCOMP)
C
C DIMENSION REAC(NCHEM), DRDX(NCHEM,NCHEM)
C
COMMON /IXCOMP/NCHEMO,
* 10HNO3,10U6,10PU4,10PU3,10U4,
* 10HNO2,10HYD,10HAN,10PU5,10PU6,
* 10NP4,10NP5,10NP6,10ZR,10TC4,
* 10TC5,10TC6,10TC7,10TCU6,10TCP4,
* 10TCZR,10SR,10RJD1,10RJR,10RJR1,
* 10RJM0,10CS,10CE,10GD,10AM,
* 10CM

C
C DATA CK1 / 6.5000D+00 /
C DATA CK4 / 1.5000D-01 /
C DATA CK5 / 250.00D+00 /
C DATA CK6 / 2.0000D-04 /
C DATA CK9 / 7.0000D+00 /
C DATA CK14 / 7.5000D+00 /
C DATA CK19 / 0.0000D+00 /
C DATA CK20 / 1.0000D+00 /
C
C DATA CK411 / 1.6000D-02 /,
* CK412 / 4.0000D-02 /
C DATA CK42 / 3.2000D-03 /
C
C DATA EK5 / 79.5D3 /
C DATA EK6 / 96.0D3 /
C DATA EK20 / 63.0D3 /
C
C DATA EPSY / 1.00D-10 /
C
C ZH = DMAX1(YH, EPSY)

C
C... PREPARATION
C
DO 110 J = 1, NCHEM
REAC(J) = 0.000
110 CONTINUE
C
DO 130 K = 1, NCHEM
DO 120 J = 1, NCHEM
DRDX(J,K) = 0.000
120 CONTINUE
130 CONTINUE
C
C
C YCN = YH + YHYD + YHAN + YNP5 + YPU5
* + 2.000*(YU6 + YPU6 + YNP6)
* + 3.000*(YPU3)
* + 4.000*(YU4 + YPU4 + YNP4)
C
DCYH = 1.000
DCYU6 = 2.000
DCYPU4 = 4.000
DCYPU3 = 3.000
DCYU4 = 4.000
DCYHYD = 1.000
DCYHAN = 1.000
DCYPU5 = 1.000
DCYPU6 = 2.000
DCYNP4 = 4.000
DCYNP5 = 1.000
DCYNP6 = 2.000

```

```

C
C
C
C
    TAU = 1.000/T - 1.000/( T0 + 25.000 )
C
    DO 210 L = 1 , MXREAC
        ECHEM(L) = 0.000
210 CONTINUE
C
    ECHEM( 5 ) = EK5
    ECHEM( 6 ) = EK6
    ECHEM(20) = EK20
C
    DO 220 L = 1 , MXREAC
        ECHEM(L) = DEXP( - ECHEM(L)/RCON*TAU )
220 CONTINUE
C
C
C
C... (1) REDUCTION OF PU(IV) BY URANOUS ( FROM R.MIXSET )
C
    CK = ECHEM(1)*FCHEM(1)*CK1
C
    RR = CK*YU4*YPU4/ZH/ZH
    DYH = -2.000*CK*YU4*YPU4/ZH/ZH/ZH
    DYP4 = CK*YU4 /ZH/ZH
    DYU4 = CK *YPU4/ZH/ZH
C
    REAC(10HN03) = REAC(10HN03) + 2.000*RR
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 2.000*DYH
    DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 2.000*DYP4
    DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 2.000*DYU4
C
    REAC(10U6 ) = REAC(10U6 ) + 0.500*RR
    DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + 0.500*DYH
    DRDX(10U6 ,10PU4) = DRDX(10U6 ,10PU4) + 0.500*DYP4
    DRDX(10U6 ,10U4) = DRDX(10U6 ,10U4) + 0.500*DYU4
C
    REAC(10PU4) = REAC(10PU4) - RR
    DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) - DYH
    DRDX(10PU4 ,10PU4) = DRDX(10PU4 ,10PU4) - DYP4
    DRDX(10PU4 ,10U4) = DRDX(10PU4 ,10U4) - DYU4
C
    REAC(10PU3) = REAC(10PU3) + RR
    DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) + DYH
    DRDX(10PU3 ,10PU4) = DRDX(10PU3 ,10PU4) + DYP4
    DRDX(10PU3 ,10U4) = DRDX(10PU3 ,10U4) + DYU4
C
    REAC(10U4) = REAC(10U4) - 0.500*RR
    DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - 0.500*DYH
    DRDX(10U4 ,10PU4) = DRDX(10U4 ,10PU4) - 0.500*DYP4
    DRDX(10U4 ,10U4) = DRDX(10U4 ,10U4) - 0.500*DYU4
C
C
C... (4) REOXYDATION OF PU(III) ( FROM R.MIXSET )
C
    CK = ECHEM(4)*FCHEM(4)*CK4
C
    RR = CK*YPU3*YNO2*(YH**3,1D0)
    DYH = 3.1D0*CK*YPU3*YNO2*(YH**2,1D0)
    DYP3 = CK *YNO2*(YH**3,1D0)
    DYN02 = CK*YPU3 *(YH**3,1D0)
C
    REAC(10HN03) = REAC(10HN03) - 1.500*RR
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) - 1.500*DYH
    DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) - 1.500*DYP3
    DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) - 1.500*DYN02
C
    REAC(10PU4) = REAC(10PU4) + RR
    DRDX(10PU4 ,10HN03) = DRDX(10PU4 ,10HN03) + DYH
    DRDX(10PU4 ,10PU3) = DRDX(10PU4 ,10PU3) + DYP3
    DRDX(10PU4 ,10HN02) = DRDX(10PU4 ,10HN02) + DYN02
C
    REAC(10PU3) = REAC(10PU3) - RR
    DRDX(10PU3 ,10HN03) = DRDX(10PU3 ,10HN03) - DYH
    DRDX(10PU3 ,10PU3) = DRDX(10PU3 ,10PU3) - DYP3
    DRDX(10PU3 ,10HN02) = DRDX(10PU3 ,10HN02) - DYN02
C
    REAC(10HN02) = REAC(10HN02) + 0.500*RR
    DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) + 0.500*DYH
    DRDX(10HN02,10PU3) = DRDX(10HN02,10PU3) + 0.500*DYP3
    DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + 0.500*DYN02
C
C
C... (5) DISINTEGRATION OF HNO2 BY HYDRAZINE
C
    FROM EXTRA.M-CODE EQ-(2.5-43)
C (THIS REACTION TERM WILL BE ADDED AT SUBROUTINE JACOB.M. SPECIAL CODING.)
C
RXHYD = 0.000
C
DRYH = 0.000
DRXHYD = 0.000
DRYNO2 = 0.000
C
IF ( IREACT .EQ. 0 ) THEN
C
    C5EPS = 1.0D-25
    ZXHYD = XHYD + C5EPS
C
    CK = ECHEM(5)*FCHEM(5)*CK5
C
    RR = CK*YNO2*(ZXHYD**0.2) *(YH**1.3)
    DYN02 = CK *(ZXHYD**0.2) *(YH**1.3)
    DYH = 1.3D0*CK*YNO2*(ZXHYD**0.2) *(YH**0.3)
    DXHYD = 0.2D0*CK*YNO2/(ZXHYD**0.8D0)*(YH**1.3)
C
    RXHYD = RR
C
    DRYH = DYH
    DRXHYD = DXHYD
    DRYNO2 = DYN02
C
END IF
C
C... (6) DISPROPORTIONATION OF NP
    FROM EXTRA.M-CODE EQ-(2.5-44)
C
    CK = ECHEM(6)*FCHEM(6)*CK6
    RR = CK*YNP4*YNP6*(2.16D0+12.5D0*YCN)
    DYNP4 = CK*YNP6*(2.16D0+12.5D0*YCN)
    DYNP6 = CK*YNP4*(2.16D0+12.5D0*YCN)
    DCN = CK*YNP4*YNP6*12.5D0
C
    REAC(10HN03) = REAC(10HN03) + 4.000*RR
    DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.000*DYNP4
    DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.000*DYNP6
C
    DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 4.000*DCN*DCYH
    DRDX(10HN03,10U6) = DRDX(10HN03,10U6) + 4.000*DCN*DCYU6
    DRDX(10HN03,10PU4) = DRDX(10HN03,10PU4) + 4.000*DCN*DCYPU4
    DRDX(10HN03,10PU3) = DRDX(10HN03,10PU3) + 4.000*DCN*DCYPU3
    DRDX(10HN03,10U4) = DRDX(10HN03,10U4) + 4.000*DCN*DCYU4
    DRDX(10HN03,10HYD) = DRDX(10HN03,10HYD) + 4.000*DCN*DCYHYD
    DRDX(10HN03,10HAN) = DRDX(10HN03,10HAN) + 4.000*DCN*DCYHAN
    DRDX(10HN03,10PU5) = DRDX(10HN03,10PU5) + 4.000*DCN*DCYPU5
    DRDX(10HN03,10PU6) = DRDX(10HN03,10PU6) + 4.000*DCN*DCYPU6
    DRDX(10HN03,10NP4) = DRDX(10HN03,10NP4) + 4.000*DCN*DCYNP4
    DRDX(10HN03,10NP5) = DRDX(10HN03,10NP5) + 4.000*DCN*DCYNP5
    DRDX(10HN03,10NP6) = DRDX(10HN03,10NP6) + 4.000*DCN*DCYNP6
C
    REAC(10NP4) = REAC(10NP4) - RR
    DRDX(10NP4 ,10NP4) = DRDX(10NP4 ,10NP4) - DYNP4
    DRDX(10NP4 ,10NP6) = DRDX(10NP4 ,10NP6) - DYNP6
C
    DRDX(10NP4 ,10HN03) = DRDX(10NP4 ,10HN03) - DCN*DCYH
    DRDX(10NP4 ,10U6) = DRDX(10NP4 ,10U6) - DCN*DCYU6
    DRDX(10NP4 ,10PU4) = DRDX(10NP4 ,10PU4) - DCN*DCYPU4
    DRDX(10NP4 ,10PU3) = DRDX(10NP4 ,10PU3) - DCN*DCYPU3
    DRDX(10NP4 ,10U4) = DRDX(10NP4 ,10U4) - DCN*DCYU4
    DRDX(10NP4 ,10HYD) = DRDX(10NP4 ,10HYD) - DCN*DCYHYD
    DRDX(10NP4 ,10HAN) = DRDX(10NP4 ,10HAN) - DCN*DCYHAN
    DRDX(10NP4 ,10PU5) = DRDX(10NP4 ,10PU5) - DCN*DCYPU5
    DRDX(10NP4 ,10PU6) = DRDX(10NP4 ,10PU6) - DCN*DCYPU6
    DRDX(10NP4 ,10NP4) = DRDX(10NP4 ,10NP4) - DCN*DCYNP4
    DRDX(10NP4 ,10NP5) = DRDX(10NP4 ,10NP5) - DCN*DCYNP5
    DRDX(10NP4 ,10NP6) = DRDX(10NP4 ,10NP6) - DCN*DCYNP6
C
    REAC(10NP5) = REAC(10NP5) + 2.000*RR
    DRDX(10NP5 ,10NP4) = DRDX(10NP5 ,10NP4) + 2.000*DYNP4
    DRDX(10NP5 ,10NP6) = DRDX(10NP5 ,10NP6) + 2.000*DYNP6
C
    DRDX(10NP5 ,10HN03) = DRDX(10NP5 ,10HN03) + 2.000*DCN*DCYH
    DRDX(10NP5 ,10U6) = DRDX(10NP5 ,10U6) + 2.000*DCN*DCYU6
    DRDX(10NP5 ,10PU4) = DRDX(10NP5 ,10PU4) + 2.000*DCN*DCYPU4
    DRDX(10NP5 ,10PU3) = DRDX(10NP5 ,10PU3) + 2.000*DCN*DCYPU3
    DRDX(10NP5 ,10U4) = DRDX(10NP5 ,10U4) + 2.000*DCN*DCYU4
    DRDX(10NP5 ,10HYD) = DRDX(10NP5 ,10HYD) + 2.000*DCN*DCYHYD
    DRDX(10NP5 ,10HAN) = DRDX(10NP5 ,10HAN) + 2.000*DCN*DCYHAN
    DRDX(10NP5 ,10PU5) = DRDX(10NP5 ,10PU5) + 2.000*DCN*DCYPU5
    DRDX(10NP5 ,10PU6) = DRDX(10NP5 ,10PU6) + 2.000*DCN*DCYPU6
    DRDX(10NP5 ,10NP4) = DRDX(10NP5 ,10NP4) + 2.000*DCN*DCYNP4
    DRDX(10NP5 ,10NP5) = DRDX(10NP5 ,10NP5) + 2.000*DCN*DCYNP5
    DRDX(10NP5 ,10NP6) = DRDX(10NP5 ,10NP6) + 2.000*DCN*DCYNP6

```

```

C
C
    REAC(10NP6) = REAC(10NP6) - RR
    DRDX(10NP6,10NP4) = DRDX(10NP6,10NP4) - DYNP4
    DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - DYNP6

C
    DRDX(10NP6,10HN3) = DRDX(10NP6,10HN3) - DCN*DCYH
    DRDX(10NP6,10U6) = DRDX(10NP6,10U6) - DCN*DCYU6
    DRDX(10NP6,10PU4) = DRDX(10NP6,10PU4) - DCN*DCYPU4
    DRDX(10NP6,10PU3) = DRDX(10NP6,10PU3) - DCN*DCYPU3
    DRDX(10NP6,10U4) = DRDX(10NP6,10U4) - DCN*DCYU4
    DRDX(10NP6,10HYD) = DRDX(10NP6,10HYD) - DCN*DCYHYD
    DRDX(10NP6,10HAN) = DRDX(10NP6,10HAN) - DCN*DCYHAN
    DRDX(10NP6,10PU5) = DRDX(10NP6,10PU5) - DCN*DCYPU5
    DRDX(10NP6,10PU6) = DRDX(10NP6,10PU6) - DCN*DCYPU6
    DRDX(10NP6,10NP4) = DRDX(10NP6,10NP4) - DCN*DCYNP4
    DRDX(10NP6,10NP5) = DRDX(10NP6,10NP5) - DCN*DCYNP5
    DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - DCN*DCYNP6

C
C... (9) REDUCTION OF NP(VI) BY URANOUS
C FROM EXTRA.M-CODE EQ-(2.5-45)
C
    CK = ECHEM(9)*FCHEM(9)*CK9
    RR = CK*YNP6*YU4
    DYU4 = CK*YNP6
    DYNP6 = CK*YU4

C
    REAC(10HN3) = REAC(10HN3) + 4.00*RR
    DRDX(10HN3,10U4) = DRDX(10HN3,10U4) + 4.00*DYU4
    DRDX(10HN3,10NP6) = DRDX(10HN3,10NP6) + 4.00*DYNP6

C
    REAC(10U6) = REAC(10U6) + RR
    DRDX(10U6,10U4) = DRDX(10U6,10U4) + DYU4
    DRDX(10U6,10NP6) = DRDX(10U6,10NP6) + DYNP6

C
    REAC(10U4) = REAC(10U4) - RR
    DRDX(10U4,10U4) = DRDX(10U4,10U4) - DYU4
    DRDX(10U4,10NP6) = DRDX(10U4,10NP6) - DYNP6

C
    REAC(10NP5) = REAC(10NP5) + 2.00*RR
    DRDX(10NP5,10U4) = DRDX(10NP5,10U4) + 2.00*DYU4
    DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) + 2.00*DYNP6

C
    REAC(10NP6) = REAC(10NP6) - 2.00*RR
    DRDX(10NP6,10U4) = DRDX(10NP6,10U4) - 2.00*DYU4
    DRDX(10NP6,10NP6) = DRDX(10NP6,10NP6) - 2.00*DYNP6

C
C... (14) REDUCTION OF PU(VI) BY HNO2
C FROM EXTRA.M-CODE EQ-(2.5-46)
C
    CK = ECHEM(14)*FCHEM(14)*CK14

C
    YH1 = CK/ZH
    RR = YPU6*YNO2*YH1
    DYH = - YPU6*YNO2*YH1/ZH
    DYP6 = YNO2*YH1
    DYN2 = YPU6 *YH1

C
    REAC(10PU6) = REAC(10PU6) - 2.00*RR
    DRDX(10PU6,10HN3) = DRDX(10PU6,10HN3) - 2.00*DYH
    DRDX(10PU6,10PU6) = DRDX(10PU6,10PU6) - 2.00*DYPU6
    DRDX(10PU6,10HN2) = DRDX(10PU6,10HN2) - 2.00*DYNO2

C
    REAC(10HN2) = REAC(10HN2) - 1.00*RR
    DRDX(10HN2,10HN3) = DRDX(10HN2,10HN3) - 1.00*DYH
    DRDX(10HN2,10PU6) = DRDX(10HN2,10PU6) - 1.00*DYPU6
    DRDX(10HN2,10HN2) = DRDX(10HN2,10HN2) - 1.00*DYNO2

C
    REAC(10PU5) = REAC(10PU5) + 2.00*RR
    DRDX(10PU5,10HN3) = DRDX(10PU5,10HN3) + 2.00*DYH
    DRDX(10PU5,10PU6) = DRDX(10PU5,10PU6) + 2.00*DYPU6
    DRDX(10PU5,10HN2) = DRDX(10PU5,10HN2) + 2.00*DYNO2

C
    REAC(10HN3) = REAC(10HN3) + 3.00*RR
    DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) + 3.00*DYH
    DRDX(10HN3,10PU6) = DRDX(10HN3,10PU6) + 3.00*DYPU6
    DRDX(10HN3,10HN2) = DRDX(10HN3,10HN2) + 3.00*DYNO2

C
C... (19) DISPROPORTIONATION OF NP
C FROM EXTRA.M-CODE EQ-(2.5-47)
C
    CK = ECHEM(19)*FCHEM(19)*CK19
    RR = CK*YNP5*YNP5*YH*YH
    DYH = 2.00*CK*YNP5*YNP5*YH
    DYNP5 = 2.00*CK*YNP5*YH*YH

C
    REAC(10HN3) = REAC(10HN3) - 4.00*RR
    DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 4.00*DYH
    DRDX(10HN3,10NP5) = DRDX(10HN3,10NP5) - 4.00*DYNP5

C
    REAC(10NP4) = REAC(10NP4) + RR
    DRDX(10NP4,10HN3) = DRDX(10NP4,10HN3) + DYH
    DRDX(10NP4,10NP5) = DRDX(10NP4,10NP5) + DYNP5

C
    REAC(10NP5) = REAC(10NP5) - 2.00*RR
    DRDX(10NP5,10HN3) = DRDX(10NP5,10HN3) - 2.00*DYH
    DRDX(10NP5,10NP5) = DRDX(10NP5,10NP5) - 2.00*DYNP5

C
    REAC(10NP6) = REAC(10NP6) + RR
    DRDX(10NP6,10HN3) = DRDX(10NP6,10HN3) + DYH
    DRDX(10NP6,10NP5) = DRDX(10NP6,10NP5) + DYNP5

C
C... (20) OXYDATION OF NP(V) BY HNO2
C FROM EXTRA.M-CODE EQ-(2.5-48)
C
    C2OEPS = 1.00-25
    ZNO2 = YNO2 + C2OEPS

C
    CK = ECHEM(20)*FCHEM(20)*CK20

C
    FF = CK*(YH**1.3000)*YCN*YCN

C
    RR = CK*(YH**1.3000)*YCN*YCN*YNP5*ZNO2/(ZNO2+YNP5)
    DYH = 1.3000*CK*(YH**0.3000)*YCN*YCN*YNP5*ZNO2/(ZNO2+YNP5)
    YNO2 = FF*YNP5*YNP5/(ZNO2+YNP5)/(ZNO2+YNP5)
    DYNP5 = FF*ZNO2*ZNO2/(ZNO2+YNP5)/(ZNO2+YNP5)
    DCN = 2.00*CK*(YH**1.3000)*YCN*YNP5*ZNO2/(ZNO2+YNP5)

C
    REAC(10HN3) = REAC(10HN3) - 3.00*RR
    DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 3.00*DYH
    DRDX(10HN3,10HN2) = DRDX(10HN3,10HN2) - 3.00*DYNO2
    DRDX(10HN3,10NP5) = DRDX(10HN3,10NP5) - 3.00*DYNP5

C
    DRDX(10HN3,10HN3) = DRDX(10HN3,10HN3) - 3.00*DCN*DCYH
    DRDX(10HN3,10U6) = DRDX(10HN3,10U6) - 3.00*DCN*DCYU6
    DRDX(10HN3,10PU4) = DRDX(10HN3,10PU4) - 3.00*DCN*DCYPU4
    DRDX(10HN3,10PU3) = DRDX(10HN3,10PU3) - 3.00*DCN*DCYPU3
    DRDX(10HN3,10U4) = DRDX(10HN3,10U4) - 3.00*DCN*DCYU4
    DRDX(10HN3,10HYD) = DRDX(10HN3,10HYD) - 3.00*DCN*DCYHYD
    DRDX(10HN3,10HAN) = DRDX(10HN3,10HAN) - 3.00*DCN*DCYHAN
    DRDX(10HN3,10PU5) = DRDX(10HN3,10PU5) - 3.00*DCN*DCYPU5
    DRDX(10HN3,10PU6) = DRDX(10HN3,10PU6) - 3.00*DCN*DCYPU6
    DRDX(10HN3,10NP4) = DRDX(10HN3,10NP4) - 3.00*DCN*DCYNP4
    DRDX(10HN3,10NP5) = DRDX(10HN3,10NP5) - 3.00*DCN*DCYNP5
    DRDX(10HN3,10NP6) = DRDX(10HN3,10NP6) - 3.00*DCN*DCYNP6

C
    REAC(10HN2) = REAC(10HN2) + RR
    DRDX(10HN2,10HN3) = DRDX(10HN2,10HN3) + DYH
    DRDX(10HN2,10HN2) = DRDX(10HN2,10HN2) + DYN2
    DRDX(10HN2,10NP5) = DRDX(10HN2,10NP5) + DYNP5

C
    DRDX(10HN2,10HN3) = DRDX(10HN2,10HN3) + DCN*DCYH
    DRDX(10HN2,10U6) = DRDX(10HN2,10U6) + DCN*DCYU6
    DRDX(10HN2,10PU4) = DRDX(10HN2,10PU4) + DCN*DCYPU4
    DRDX(10HN2,10PU3) = DRDX(10HN2,10PU3) + DCN*DCYPU3
    DRDX(10HN2,10U4) = DRDX(10HN2,10U4) + DCN*DCYU4
    DRDX(10HN2,10HYD) = DRDX(10HN2,10HYD) + DCN*DCYHYD
    DRDX(10HN2,10HAN) = DRDX(10HN2,10HAN) + DCN*DCYHAN
    DRDX(10HN2,10PU5) = DRDX(10HN2,10PU5) + DCN*DCYPU5
    DRDX(10HN2,10PU6) = DRDX(10HN2,10PU6) + DCN*DCYPU6
    DRDX(10HN2,10NP4) = DRDX(10HN2,10NP4) + DCN*DCYNP4
    DRDX(10HN2,10NP5) = DRDX(10HN2,10NP5) + DCN*DCYNP5
    DRDX(10HN2,10NP6) = DRDX(10HN2,10NP6) + DCN*DCYNP6

C
    REAC(10NP5) = REAC(10NP5) - 2.00*RR
    DRDX(10NP5,10HN3) = DRDX(10NP5,10HN3) - 2.00*DYH
    DRDX(10NP5,10HN2) = DRDX(10NP5,10HN2) - 2.00*DYNO2
    DRDX(10NP5,10NP5) = DRDX(10NP5,10NP5) - 2.00*DYNP5

C
    DRDX(10NP5,10HN3) = DRDX(10NP5,10HN3) - 2.00*DCN*DCYH
    DRDX(10NP5,10U6) = DRDX(10NP5,10U6) - 2.00*DCN*DCYU6
    DRDX(10NP5,10PU4) = DRDX(10NP5,10PU4) - 2.00*DCN*DCYPU4
    DRDX(10NP5,10PU3) = DRDX(10NP5,10PU3) - 2.00*DCN*DCYPU3
    DRDX(10NP5,10U4) = DRDX(10NP5,10U4) - 2.00*DCN*DCYU4
    DRDX(10NP5,10HYD) = DRDX(10NP5,10HYD) - 2.00*DCN*DCYHYD
    DRDX(10NP5,10HAN) = DRDX(10NP5,10HAN) - 2.00*DCN*DCYHAN
    DRDX(10NP5,10PU5) = DRDX(10NP5,10PU5) - 2.00*DCN*DCYPU5
    DRDX(10NP5,10PU6) = DRDX(10NP5,10PU6) - 2.00*DCN*DCYPU6
    DRDX(10NP5,10NP4) = DRDX(10NP5,10NP4) - 2.00*DCN*DCYNP4
    DRDX(10NP5,10NP5) = DRDX(10NP5,10NP5) - 2.00*DCN*DCYNP5
    DRDX(10NP5,10NP6) = DRDX(10NP5,10NP6) - 2.00*DCN*DCYNP6

```

```

C
REAC(10NP6 ) = REAC(10NP6 ) + 2.000*FR
DRDX(10NP6 ,10HN03) = DRDX(10NP6 ,10HN03) + 2.000*DYH
DRDX(10NP6 ,10HN02) = DRDX(10NP6 ,10HN02) + 2.000*DYNO2
DRDX(10NP6 ,10NP5 ) = DRDX(10NP6 ,10NP5 ) + 2.000*DYNP5
C
DRDX(10NP6 ,10HN03) = DRDX(10NP6 ,10HN03) + 2.000*DCN*DCYH
DRDX(10NP6 ,10U6 ) = DRDX(10NP6 ,10U6 ) + 2.000*DCN*DCYU6
DRDX(10NP6 ,10PU4 ) = DRDX(10NP6 ,10PU4 ) + 2.000*DCN*DCYPU4
DRDX(10NP6 ,10PU3 ) = DRDX(10NP6 ,10PU3 ) + 2.000*DCN*DCYPU3
DRDX(10NP6 ,10U4 ) = DRDX(10NP6 ,10U4 ) + 2.000*DCN*DCYU4
DRDX(10NP6 ,10HYD ) = DRDX(10NP6 ,10HYD ) + 2.000*DCN*DCYHYD
DRDX(10NP6 ,10HAN ) = DRDX(10NP6 ,10HAN ) + 2.000*DCN*DCYHAN
DRDX(10NP6 ,10PU5 ) = DRDX(10NP6 ,10PU5 ) + 2.000*DCN*DCYPU5
DRDX(10NP6 ,10PU6 ) = DRDX(10NP6 ,10PU6 ) + 2.000*DCN*DCYPU6
DRDX(10NP6 ,10NP4 ) = DRDX(10NP6 ,10NP4 ) + 2.000*DCN*DCYNP4
DRDX(10NP6 ,10NP5 ) = DRDX(10NP6 ,10NP5 ) + 2.000*DCN*DCYNP5
DRDX(10NP6 ,10NP6 ) = DRDX(10NP6 ,10NP6 ) + 2.000*DCN*DCYNP6

```

C
C... (41) OXYDATION OF U(IV) THROUGH NITROUS ACID (R.MIXSET)

```

C
C41EPS = 1.00-10
C
ZNO2 = YNO2 + C41EPS
C
IF ( YH .LE. 0.3400 ) THEN
C
CK = ECHEM(41)*FCHEM(41)*CK411
C
RR = CK*YU4*(ZNO2**0.4900)
DYU4 = CK *(ZNO2**0.4900)
DYNO2 = 0.4900*CK*YU4/(ZNO2**0.5100)

```

```

C
REAC(10HN03) = REAC(10HN03) + FR
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + DYU4
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DYNO2
C
REAC(10U6 ) = REAC(10U6 ) + FR
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DYU4
DRDX(10U6 ,10HN02) = DRDX(10U6 ,10HN02) + DYNO2
C
REAC(10U4 ) = REAC(10U4 ) - FR
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DYU4
DRDX(10U4 ,10HN02) = DRDX(10U4 ,10HN02) - DYNO2
C
REAC(10HN02) = REAC(10HN02) + FR
DRDX(10HN02,10U4 ) = DRDX(10HN02,10U4 ) + DYU4
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + DYNO2

```

```

C
ELSE
C
CK = ECHEM(41)*FCHEM(41)*CK412
C
RR = CK*YU4*(ZH**0.6300)*(ZNO2**0.4900)
DYH = 0.6300*CK*YU4/(ZH**0.3700)*(ZNO2**0.4900)
DYU4 = CK *(ZH**0.6300)*(ZNO2**0.4900)
DYNO2 = 0.4900*CK*YU4*(ZH**0.6300)/(ZNO2**0.5100)

```

```

C
REAC(10HN03) = REAC(10HN03) + FR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + DYH
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + DYU4
DRDX(10HN03,10HN02) = DRDX(10HN03,10HN02) + DYNO2
C
REAC(10U6 ) = REAC(10U6 ) + FR
DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + DYH
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DYU4
DRDX(10U6 ,10HN02) = DRDX(10U6 ,10HN02) + DYNO2
C
REAC(10U4 ) = REAC(10U4 ) - FR
DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - DYH
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DYU4
DRDX(10U4 ,10HN02) = DRDX(10U4 ,10HN02) - DYNO2
C
REAC(10HN02) = REAC(10HN02) + FR
DRDX(10HN02,10HN03) = DRDX(10HN02,10HN03) + DYH
DRDX(10HN02,10U4 ) = DRDX(10HN02,10U4 ) + DYU4
DRDX(10HN02,10HN02) = DRDX(10HN02,10HN02) + DYNO2

```

C
C
C
C... (42) OXYDATION OF U(IV) BY OXYGEN (R.MIXSET)

```

C
CK = ECHEM(42)*FCHEM(42)*CK42
C
RR = CK*YU4/(ZH**0.8600)
DYH = -0.8600*CK*YU4/(ZH**1.8600)

```

```

C
DYU4 = CK / (ZH**0.8600)
C
REAC(10HN03) = REAC(10HN03) + 2.000*FR
DRDX(10HN03,10HN03) = DRDX(10HN03,10HN03) + 2.000*DYH
DRDX(10HN03,10U4 ) = DRDX(10HN03,10U4 ) + 2.000*DYU4
C
REAC(10U6 ) = REAC(10U6 ) + FR
DRDX(10U6 ,10HN03) = DRDX(10U6 ,10HN03) + DYH
DRDX(10U6 ,10U4 ) = DRDX(10U6 ,10U4 ) + DYU4
C
REAC(10U4 ) = REAC(10U4 ) - FR
DRDX(10U4 ,10HN03) = DRDX(10U4 ,10HN03) - DYH
DRDX(10U4 ,10U4 ) = DRDX(10U4 ,10U4 ) - DYU4
C
RETURN
END

```

```

SUBROUTINE READPL(IUPRT ,IUPLT ,INFILE,
*          CVUNIT,
*          MXCOMP,MXUNIT,MXSTG ,MXSTG2,MXBLK ,
*          NUNIT ,NBLK ,
*          ID ,TYPE ,MSTG ,
*          IPOS ,
*          X )
C
C
C          READ CONCENTRATIONS FROM PLOTFILE.
C
C          IMPLICIT REAL*8 ( A-H , 0-Z )
C
C          CHARACTER*33 INFILE
C          CHARACTER*8 CNAMEX
C          CHARACTER*8 CUNITX
C          CHARACTER*2 TYPEX
C
C          DIMENSION CVUNIT(MXCOMP)
C
C          DIMENSION ID (          MXUNIT)
C          CHARACTER*2 TYPE (          MXUNIT)
C          DIMENSION MSTG (          MXUNIT)
C
C          DIMENSION IPOS (2,MXSTG2,MXUNIT)
C
C          DIMENSION X (MXCOMP,2,MXBLK )
C
COMMON /IXCOMP/NCHEMO,
*          IOHNO3,IOU6 ,IOPU4 ,IOPU3 ,IOU4 ,
*          IOHNO2,IOHYD ,IOHAN ,IOPU5 ,IOPU6 ,
*          IONP4 ,IONP5 ,IONP6 ,IOZR ,IOTC4 ,
*          IOTC5 ,IOTC6 ,IOTC7 ,IOTCU6 ,IOTCP4 ,
*          IOTCZR,IOSR ,IORUDI,IOFUTR,IORUNI,
*          IORJMO,IOCS ,IOCE ,IOGD ,IOAM ,
*          IOCM
C
C
C          OPEN(UNIT=IUPLT,FILE=INFILE,FORM='FORMATTED',
*          STATUS='OLD')
C
1000 CONTINUE
C
C          READ(IUPLT,*,END=3000) NUNITX ,TIMEX
C
C          IF ( NUNITX .NE. NUNIT ) THEN
C            WRITE(IUPRT,*) ' PROCESS SYSTEM IN PLOTFILE IS MISMATCH.'
C            STOP
C          END IF
C
C          DO 1010 IUNT = 1 , NUNIT
C            READ(IUPLT,*) IDX,TYPEX,MSTGX
C
C            IF ( TYPEX .NE. TYPE(IUNT) .OR. MSTGX .NE. MSTG(IUNT) ) THEN
C              WRITE(IUPRT,*) ' PROCESS SYSTEM IN PLOTFILE IS MISMATCH.'
C              STOP
C            END IF
C
1010 CONTINUE
C
C
C          DO 2000 IUNT = 1 , NUNIT
C            NSTG = MSTG (IUNT)
C
C            IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C              ... MIXER-SETTLER CONCENTRATIONS ...
C
C              READ(IUPLT,*) IDX,MSTGX
C
C              READ(IUPLT,*) ( CNAMEX,J=1,MXCOMP)
C              READ(IUPLT,*) ( CUNITX,J=1,MXCOMP)
C
C              DO 1110 I = 1 , NSTG
C                IND = IPOS(1,2*I-1,IUNT)
C                IND = IPOS(2,2*I-1,IUNT)
C                READ(IUPLT,*) IDUM,(X(J,1,IND),J=1,MXCOMP)
1110 CONTINUE
C
C
C              READ(IUPLT,*) IDX,MSTGX
C
C              READ(IUPLT,*) ( CNAMEX,J=1,MXCOMP)
C              READ(IUPLT,*) ( CUNITX,J=1,MXCOMP)
C
C              DO 1130 I = 1 , NSTG
C                INDSA = IPOS(1,2*I,IUNT)
C                INDSO = IPOS(2,2*I,IUNT)
C                READ(IUPLT,*) IDUM,(X(J,1,INDSA),J=1,MXCOMP)
1130 CONTINUE
C
C              READ(IUPLT,*) IDX,MSTGX
C
C              READ(IUPLT,*) ( CNAMEX,J=1,MXCOMP)
C              READ(IUPLT,*) ( CUNITX,J=1,MXCOMP)
C
C              DO 1140 I = 1 , NSTG
C                INDSA = IPOS(1,2*I,IUNT)
C                INDSO = IPOS(2,2*I,IUNT)
C                READ(IUPLT,*) IDUM,(X(J,2,INDSO),J=1,MXCOMP)
1140 CONTINUE
C
C              ELSE
C
C                ... TANK CONCENTRATIONS ...
C
C              READ(IUPLT,*) IDX,MSTGX
C
C              READ(IUPLT,*) ( CNAMEX,J=1,MXCOMP)
C              READ(IUPLT,*) ( CUNITX,J=1,MXCOMP)
C
C              DO 1210 I = 1 , NSTG
C                IND = IPOS(1,1,IUNT)
C                IND = IPOS(2,1,IUNT)
C                READ(IUPLT,*) IDUM,(X(J,1,IND),J=1,MXCOMP)
1210 CONTINUE
C
C              READ(IUPLT,*) IDX,MSTGX
C
C              READ(IUPLT,*) ( CNAMEX,J=1,MXCOMP)
C              READ(IUPLT,*) ( CUNITX,J=1,MXCOMP)
C
C              DO 240 I = 1 , NSTG
C                IND = IPOS(1,1,IUNT)
C                IND = IPOS(2,1,IUNT)
C                READ(IUPLT,*) IDUM,(X(J,2,IND),J=1,MXCOMP)
240 CONTINUE
C
C              END IF
C
C              2000 CONTINUE
C
C              3000 CONTINUE
C
C              CLOSE(UNIT=IUPLT)
C
C              DO 3200 I = 1 , NBLK
C                DO 3100 J = 1 , MXCOMP
C                  X(J,1,I) = X(J,1,I)/CVUNIT(J)
C                  X(J,2,I) = X(J,2,I)/CVUNIT(J)
3100 CONTINUE
3200 CONTINUE
C
C              RETURN
C
C              END

```

```

SUBROUTINE RUCOMP(XH ,RRUDI ,RRUTR ,RRUNI ,RRUMO )
C
C CALCULATE RUTHENIUM COMPLEX COMPOSITION
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
RRUDI = 1.61511D0*( -DEXP( -0.112066D0*(XH +0.61049D0) )
*
+DEXP( -0.0069D0 *(XH*XH+0.85399D0) ) )
IF ( RRUDI .LT. 0.0D0 ) THEN
RRUDI = 0.0D0
END IF
C
C
RRUTR = 0.0799708D0*( XH + 2.20226D0 )
*
*DEXP( 2.69315D0/( XH + 2.20226D0) )
*
- 0.586792D0
IF ( RRUTR .LT. 0.0D0 ) THEN
RRUTR = 0.0D0
END IF
C
C
RRUNI = 0.489439D0*DEXP( 0.234136D0/( XH + 0.493768D0) )
*
- 0.488788D0
IF ( RRUNI .LT. 0.0D0 ) THEN
RRUNI = 0.0D0
END IF
C
C
RRUMO = 1.0D0 - RRUDI - RRUTR - RRUNI
IF ( RRUMO .LT. 0.0D0 ) THEN
RRUMO = 0.0D0
END IF
C
C
RETURN
END

```

```

SUBROUTINE SETRMP(MXDEST,NXRM ,NUNIT ,MXUNIT,MXRM ,MXSTG2,
*
TYPE ,MDEST ,
*
MSTG ,IUDSTA,IUDSTO,
*
IROWFM,ICOLRM,
*
IPOS )
C
C SET ROW POSITIONS AND COLUMN POSITIONS OF SUB-MATRICES
C
C IMPLICIT REAL*8 ( A-H , 0-Z )
C
C CHARACTER*2 TYPE(MXUNIT)
C
C DIMENSION MSTG ( MXUNIT)
C DIMENSION IUDSTA(3,MXDEST,MXUNIT),
*
IUDSTO(3,MXDEST,MXUNIT)
C DIMENSION MDEST ( 2 ,MXUNIT)
C
C DIMENSION IROWFM( MXRM ),
*
ICOLRM( MXRM )
C
C DIMENSION IPOS(2,MXSTG2,MXUNIT)
C
C NXRM = 0
C
C DO 1400 IUNTS = 1 , NUNIT
C
C NDEST = MDEST(1, IUNTS)
C
C IF ( NDEST .GT. 0 ) THEN
C
C DO 1300 KDEST = 1 , NDEST
C
C NSTGS = MSTG ( IUNTS)
C IUNT = IUDSTA(1,KDEST, IUNTS)
C I = IUDSTA(2,KDEST, IUNTS)
C IS = 1
C
C IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
C IMS = 2*IS
C ELSE
C IMS = IS
C END IF
C
C IUDSTA(3,KDEST, IUNTS) = 0
C
C INDAS = IPOS(1,IMS, IUNTS)
C INDOS = IPOS(2,IMS, IUNTS)
C
C IF ( IUNT .NE. 0 ) THEN
C
C IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C IM = 2*I - 1
C ELSE
C IM = I
C END IF
C
C INDA = IPOS(1,IM, IUNT)
C INDO = IPOS(2,IM, IUNT)
C
C IF (
*
INDAS .NE. (INDA-1)
,AND. INDAS .NE. (INDA+1) ) THEN
C
C IF ( NXRM .GT. 0 ) THEN
C
C DO 1100 IXRMM = 1 , NXRM
C IXRM = IXRMM
C IF (
*
INDAS .EQ. ICOLRM(IXRMM)
,AND. INDA .EQ. IROWFM(IXRMM) ) THEN
C GO TO 1200
C END IF
1100 CONTINUE
C
C END IF
C
C IF ( NXRM .EQ. MXRM ) THEN
C WRITE(IUPRT,*)
*
' TOO MANY SOURCE TERM(AQUEOUS PHASE).'
C WRITE(IUPRT,*) ' MAXIMUM IS MXRM = ',MXRM
C END IF
C
C NXRM = NXRM + 1
C IXRM = NXRM
C
C 1200 CONTINUE
C IUDSTA(3,KDEST, IUNTS) = IXRM

```

```

        IROWRM(IXRM)      = INDA
        ICOLRM(IXRM)      = INDAS
C
        END IF
C
        END IF
C
1300    CONTINUE
C
        END IF
C
1400    CONTINUE
C
        DO 2400 IUNTS = 1 , NUNIT
C
            NDEST = MDEST(2, IUNTS)
C
            IF ( NDEST .GT. 0 ) THEN
C
                DO 2300 KDEST = 1 , NDEST
C
                    NSTGS = MSTG ( IUNTS)
                    IUNT  = IUDSTO(1, KDEST, IUNTS)
                    I     = IUDSTO(2, KDEST, IUNTS)
                    IS    = NSTGS
C
                    IF ( TYPE(IUNTS) .EQ. 'MS' ) THEN
                        IMS = 2*IS
                    ELSE
                        IMS = IS
                    END IF
C
                    IUDSTO(3, KDEST, IUNTS) = 0
C
                    INDA = IPOS(1, IMS, IUNTS)
                    INDO = IPOS(2, IMS, IUNTS)
C
                    IF ( IUNT .NE. 0 ) THEN
C
                        IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
                            IM = 2*I - 1
                        ELSE
                            IM = I
                        END IF
C
                        INDA = IPOS(1, IM, IUNT)
                        INDO = IPOS(2, IM, IUNT)
C
                        IF (
                            INDO .NE. (INDO-1)
                            .AND. INDO .NE. (INDO+1) ) THEN
C
                            IF ( NXRM .GT. 0 ) THEN
C
                                DO 2100 IXRMM = 1 , NXRM
                                    IXRM = IXRMM
                                    IF (
                                        INDO .EQ. ICOLRM(IXRMM)
                                        .AND. INDO .EQ. IROWRM(IXRMM) ) THEN
                                        GO TO 2200
                                    END IF
                                CONTINUE
C
                                END IF
C
                                IF ( NXRM .EQ. MXRM ) THEN
                                    WRITE(IUPRT, *)
                                        ' TOO MANY SOURCE TERM(ORGANIC PHASE).'
                                    WRITE(IUPRT, *) ' MAXIMUM IS NXRM = ', MXRM
                                END IF
C
                                NXRM = NXRM + 1
                                IXRM = NXRM
C
                                CONTINUE
                                IUDSTO(3, KDEST, IUNTS) = IXRM
                                IROWRM(IXRM) = INDO
                                ICOLRM(IXRM) = INDO
C
                            END IF
C
                        CONTINUE
C
2300    CONTINUE
C
                    END IF
C
2400    CONTINUE
C
                RETURN
            END
    END

```

```

SUBROUTINE SOLFFF( ISFIN , FRC ,
*                CO , TO ,
*                MXCOMP ,
*                MXFEED, NFA , NFO ,
*                TFEED ,
*                XF , FF )
C
C*****
C
C    TRANSFORM XF AND FF TO SOLUT-FREE UNIT
C
C*****
C
C    IMPLICIT REAL*8 ( A-H , 0-Z )
C
C    DIMENSION XF(MXCOMP, MXFEED, 2) , FF(MXFEED, 2)
C
C
C    DO 120 I = 1 , NFA
C
C        CCO = CO
C        TTO = TO
C
C        CALL DENST( 1 , 1 ,
*                ISFIN , FRC ,
*                CCO , TTO ,
*                TFEED , WLTF ,
*                WSTF , HOSF )
C
C        DENFCA = DENMSA( MXCOMP , XF(1, I, 1) )
C
C        FF(1, I) = DENFCA * FF(1, I)
C
C        DO 110 J = 1 , MXCOMP
            XF(J, I, 1) = XF(J, I, 1) / DENFCA
110    CONTINUE
120    CONTINUE
C
C
C    DO 220 I = 1 , NFO
C
C        CCO = CO
C        TTO = TO
C
C        CALL DENST( 1 , 1 ,
*                ISFIN , FRC ,
*                CCO , TTO ,
*                TFEED , WLTF ,
*                WSTF , HOSF )
C
C        DENFCO = DENMSO( MXCOMP ,
*                WLTF , HOSF ,
*                XF(1, I, 2) )
C
C        FF(1, 2) = DENFCO * FF(1, 2)
C
C        DO 210 J = 1 , MXCOMP
            XF(J, I, 2) = XF(J, I, 2) / DENFCO
210    CONTINUE
220    CONTINUE
C
        RETURN
    END

```



```

SUBROUTINE TBPDEG(IFOUT ,
*      TITLE , TIME , ENDTIME , DT ,
*      CNAME , FMASS , DECAYH , CVUNIT ,
*      CUNIT , CHR9 , CNTBP , FDEG ,
*      MXCOMP , MXUNIT , MXSTG , MXSTG2 , NCHEM , MXBLK ,
*      MXEXTR , NUNIT , FRC , CO , TO ,
*      ID , TYPE , MSTG , MEXTRA , FL ,
*      TSTG , V , VS , PHI , PHIS ,
*      IPOS ,
*      XN )
C
C
C      TBP DEGRADATION CALCULATION
C
C      IMPLICIT REAL*8 ( A-H , O-Z )
C
C      CHARACTER*72 TITLE
C
C      CHARACTER*8 CNAME (MXCOMP)
C      DIMENSION FMASS (MXCOMP)
C      DIMENSION DECAYH (MXCOMP)
C      DIMENSION CVUNIT (MXCOMP)
C      CHARACTER*8 CUNIT (MXCOMP)
C      DIMENSION CHR9 (MXCOMP) ,
*      CNTBP (MXCOMP)
C
C      DIMENSION ID ( MXUNIT)
C      CHARACTER*2 TYPE ( MXUNIT)
C
C      DIMENSION MSTG ( MXUNIT)
C      DIMENSION MEXTRA ( MXUNIT)
C
C      DIMENSION FL (2 , MXSTG , MXEXTR) ,
*      TSTG ( MXSTG , MXEXTR) ,
*      V ( MXSTG , MXEXTR) ,
*      VS ( MXSTG , MXEXTR) ,
*      PHI ( MXSTG , MXEXTR) ,
*      PHIS ( MXSTG , MXEXTR)
C
C      DIMENSION IPOS (2 , MXSTG2 , MXUNIT)
C
C      DIMENSION XN (MXCOMP , 2 , MXBLK )
C
C      COMMON /IXCOMP/NCHEM ,
*      IOHNO3 , IOU6 , IOU4 , IOU3 , IOU4 ,
*      IOHNO2 , IOHYD , IOHAN , IOU5 , IOU6 ,
*      IOHP4 , IOHP5 , IOHP6 , IOZR , IOTC4 ,
*      IOTC5 , IOTC6 , IOTC7 , IOTC6 , IOTCP4 ,
*      IOTCZR , IOSR , IORUD1 , IORUTR , IORUNI ,
*      IORUNO , IOCS , IOCE , IOGD , IOAM ,
*      IOCM
C
C      DATA ONE/1.0D0/
C
C      FMSTG = 266.3D0
C      FMDBP = 210.2D0
C      EV = 1.6021892D-19
C      ANA = 6.022045D23
C      CTBP = CO * FRC
C      FRAD = FDEG*FRC
C
C      WET CONDITION
C
C      A = 1.263D-2
C      B = - 6.972D-2
C      C = - 1.525D-2
C      D = 8.631D-1
C      GVAL = A *CTBP**4 + B *CTBP**3 + C *CTBP**2 + D *CTBP
C
C      DRY CONDITION
C
C      AD = - 2.3D-2
C      BD = 2.1D-1
C      CD = - 7.3D-1
C      DD = 1.8D0
C      GVALD = AD*CTBP**4 + BD*CTBP**3 + CD*CTBP**2 + DD*CTBP
C
C      WRITE (IFOUT , 5000) NUNIT , TIME
C
C      DO 1000 IUNT = 1 , NUNIT

```

```

C
C      NSTG = MSTG (IUNT)
C      IEXTRA = MEXTRA(IUNT)
C
C      IF ( TYPE(IUNT) .EQ. 'MS' ) THEN
C
C      ... MIXER-SETTLER CONCENTRATIONS ...
C
C      WRITE (IFOUT , 5010) TITLE , TIME
C      WRITE (IFOUT , 5100) ID (IUNT) , TYPE (IUNT) ,
*      'TBP DEGRADATION CALCULATION'
C      WRITE (IFOUT , 5200) 'TEMP.' ,
*      'VOL (ORG)' ,
*      'FLOW (ORG)' ,
*      'RETAIN-T.' ,
*      'HEAT' ,
*      'G-VALUE' ,
*      'RATE CONS.' ,
*      'DBP (CHEM.)' ,
*      'DBP (RAD.)' ,
*      'DBP (TOT)' ,
*      'DBP' ,
*      'DBP'
C      WRITE (IFOUT , 5210) ' (C)' ,
*      ' (L)' ,
*      ' (L/HR)' ,
*      ' (HR)' ,
*      ' (W)' ,
*      ' (1/100EV)' ,
*      ' (1/HR)' ,
*      ' (MOL/HR)' ,
*      ' (MOL/HR)' ,
*      ' (MOL/HR)' ,
*      ' (G/HR)' ,
*      ' (PPM/HR)'
C      DO 120 I = 1 , NSTG
C
C      IND = IPOS(2,2*I-1 , IUNT)
C      INDSO = IPOS(2,2*I , IUNT)
C
C      TORG = TSTG(I , IEXTRA) - TO
C      VORG = V (I , IEXTRA) * ( ONE - PHI (I , IEXTRA) )
*      + VS (I , IEXTRA) * ( ONE - PHIS (I , IEXTRA) )
C      FORG = FL(2 , I , IEXTRA)
C      RTMORG = VORG/FORG
C      RHOORG = - 7.3D-4 * ( TORG - 25.0D0 ) + 0.813D0
C
C      HORG = 0.0D0
C      DO 110 J = 1 , MXCOMP
C      HORG = HORG + DECAYH (J)
*      *FMASS (J)
*      *XN (J , 2 , IND)
*      *V (I , IEXTRA) * ( ONE - PHI (I , IEXTRA) )
*      + DECAYH (J)
*      *FMASS (J)
*      *XN (J , 2 , INDSO)
*      *VS (I , IEXTRA) * ( ONE - PHIS (I , IEXTRA) )
C      CONTINUE
C
C      RATEC = 2.64D-08 *DEXP (9.46D-2 *TORG)
C      DBPCHE = CTBP *RATEC *VORG
C      DBPRAD = FRAD *HORG *3600.0D0 /EV *GVAL /100.0D0 /ANA
C      DBPTOT = DBPCHE + DBPRAD
C      DBP = DBPTOT *FMDBP
C      DBPPPM = DBP /VORG *1.0D3 /RHOORG
C
C      WRITE (IFOUT , 5220) I ,
*      TORG ,
*      VORG ,
*      FORG ,
*      RTMORG ,
*      HORG ,
*      GVAL ,
*      RATEC ,
*      DBPCHE ,
*      DBPRAD ,
*      DBPTOT ,
*      DBP ,
*      DBPPPM
C      CONTINUE
C
C      END IF
C
C      1000 CONTINUE
C
C
C      RETURN

```

```

C
C
5000 FORMAT(1H , 'NUMBER OF UNIT(S) :', I3, 5X, 'TIME:', 1P, E12.5, ' (HR)')
5010 FORMAT(1H /1H /1H , A, 5X, 'TIME:', 1P, E12.5, ' (HR)')
5100 FORMAT(1H /1H , 'UNIT ID.', I5, 1X, A2, 5X, A)
5200 FORMAT(1H , 'STAGE ', A5, 11(1X, A10))
5210 FORMAT(1H , ' ', A5, 11(1X, A10))
5220 FORMAT(1H , I2, F7.2, 1P, 11E11.3)
C
C
END
    
```

```

SUBROUTINE TRAN
* (IUPRT , IUPLT , IRST , IPFDIS, TITLE , CTFILE, RXLIM ,
* MXTAUC, TMCONB, TMCONC, DTCONS,
* MXMCOMP, MXCOMP, MXUNIT, MXSTG , MXSTG2, NCHEM , MXBLK , MXREAC, MXDEST ,
* MEXTR, NUNIT , NBLK , NEXTRA, ENDTIM, DTO , DTMAX , DTMIN , DTPRT ,
* DTPLT , TRATE , ITMAX , EPS , ITMIN , EPSLIM, ISFIN , FRC , CO ,
* TO , CDBP , FDEG , RCON , CNAME , FMASS , DECAYH, CVUNIT , CUNIT ,
* CHRG , CNTBP , CPHASE, MXFEED, MXTABL, NFA , NFO , TFEED , FF ,
* NFFTAB, FFTAB , XF , NXFTAB, XTAB , IFDSTA, IFDSTO,
* ID , IREACT, FCHEM , ECHEM , TYPE , MSTG , IUDSTA , IUDSTO, MDEST ,
* MEXTRA, TAV , VTOT , ACID , ACIDIN, OXID , FIN , FOUT , FFOUT ,
* FFOUT , FMSIN , FMSOUT, FMTIN , FMTOUT, FMTBL ,
* IDEXTR, CHRDIS, IDIST , IEFF , CDIS , CKEQU , EFF , DIS , TSTG ,
* V , VS , HEIGHT, FL , FR , PHIOLD, PHI , PHIS , WLT ,
* WST , HOS , TF , FLS , FLSMT , IPFLS , IPOS , XN , X ,
* DX , REAC , DRDX , DDDX , FKEO1 , FKEO2 , FKEO3 , FKEOC ,
* DK1 , DK2 , DK3 , DKC , DK1DX , DK2DX , DK3DX , DKCDX , DADX ,
* DBOX , DTFDX , DISTC , DDCCX , DDCCDF,
* MXN , MXSYS , MXRM , NXRM , FX , DFDXD , DFDXL , DFDXU ,
* IP , WK , FM , IROWFM, ICOLFM,
* IPOPFM, EM , IROWEM, SYSM , Y , IPSYS )
    
```

```

C
C*****
C
C CONTROL TRANSIENT CALCULATION.
C*****
    
```

```

C
C IMPLICIT REAL*8 ( A-H , O-Z )
C
C CHARACTER* 72 TITLE
C CHARACTER* 30 CTFILE
C CHARACTER* 33 COUTF
C CHARACTER* 33 COUTP
C INTEGER IUPRT, IUPLT, LFNAME
    
```

```

C
C COMMON /IXCOMP/NCHEM,
* IOHNO3, IOU6 , IOPU4 , IOPU3 , IOU4 ,
* IOHNO2, IOHYD , IOHAN , IOPU5 , IOPU6 ,
* IONP4 , IONP5 , IONP6 , IOZR , IOTC4 ,
* IOTC5 , IOTC6 , IOTC7 , IOTCJ6, IOTCP4,
* IOTCZR, IOSR , IOJUDI, IOJUTR, IOJUNI,
* IOJUMO, IOCS , IOCE , IOGD , IOAM ,
* IOCM
    
```

```

C
C CHARACTER*8 CNAME (MXCOMP)
C DIMENSION FMASS (MXCOMP)
C DIMENSION DECAYH (MXCOMP)
C DIMENSION CVUNIT (MXCOMP)
C CHARACTER*8 CUNIT (MXCOMP)
C DIMENSION CHRG (MXCOMP),
* CNTBP (MXCOMP)
    
```

```

C
C CHARACTER*8 CPHASE (MXCOMP)
C
C DIMENSION FF ( MXFEED, 2),
* NFFTAB( MXFEED, 2),
* FFTAB (2 , MXTABL, MXFEED, 2),
* XF (MXCOMP , MXFEED, 2),
* NXFTAB( MXFEED, 2),
* XTAB (MXCOMP+1, MXTABL, MXFEED, 2),
* IFDSTA(2 , MXFEED ),
* IFDSTO(2 , MXFEED )
    
```

```

C
C DIMENSION ID ( MXUNIT)
C DIMENSION IREACT( MXUNIT)
C DIMENSION FCHEM (MXREAC, 2 , MXUNIT),
* ECHEM (MXREAC, 2 , MXUNIT)
    
```

```

C
C CHARACTER*2 TYPE ( MXUNIT)
C
C DIMENSION MSTG ( MXUNIT),
* IUDSTA(3 , MXDEST, MXUNIT),
* IUDSTO(3 , MXDEST, MXUNIT)
C DIMENSION MDEST ( 2 , MXUNIT),
* MEXTRA( MXUNIT)
    
```

```

C
C DIMENSION TAV ( MXUNIT),
* VTOT ( MXUNIT),
* ACID ( MXUNIT),
* ACIDIN( MXUNIT),
* OXID ( MXUNIT),
* FIN ( 2 , MXUNIT),
* FOUT ( 2 , MXUNIT)
    
```

```

*          RFOUT (MXDEST, 2, MXUNIT),
*          FMSIN (MXMCOMP, 2, MXUNIT),
*          FMSOUT (MXMCOMP, 2, MXUNIT)
C
  DIMENSION FMTIN (MXMCOMP, 2),
*           FMTOUT (MXMCOMP, 2),
*           RMTBL (MXMCOMP, 3)
C
  DIMENSION IDEXTR (MXEXTR),
CHARACTER*8 CHRDIS (MXCOMP, MXEXTR)
C
  DIMENSION IDIST (MXCOMP, MXEXTR)
  DIMENSION IEFF (MXCOMP, MXEXTR)
C
  DIMENSION CDIS (MXCOMP, MXSTG, MXEXTR),
*           CKEOU (MXCOMP, MXSTG, MXEXTR),
*           EFF (MXCOMP, MXSTG, MXEXTR),
*           DIS (MXCOMP, MXSTG, MXEXTR)
C
  DIMENSION TSTG (MXSTG, MXEXTR),
*           V (MXSTG, MXEXTR),
*           VS (MXSTG, MXEXTR),
*           HEIGHT (MXSTG, MXEXTR),
*           FL (2, MXSTG, MXEXTR),
*           FR (MXSTG, MXEXTR),
*           PHIOLD (MXSTG, MXEXTR),
*           PHI (MXSTG, MXEXTR),
*           PHIS (MXSTG, MXEXTR),
*           WLT (MXSTG, MXEXTR),
*           WST (MXSTG, MXEXTR),
*           HOS (MXSTG, MXEXTR),
*           TF (MXSTG, MXEXTR)
C
  DIMENSION FLS (MXUNIT, 2),
*           FLSMT (MXUNIT, MXUNIT, 2),
*           IPFLS (MXUNIT, 2)
C
  DIMENSION IPOS (2, MXSTG2, MXUNIT)
C
  DIMENSION DTPRT (100)
  DIMENSION DTPLT (100)
C
  DIMENSION XN (MXCOMP, 2, MXBLK),
*           X (MXCOMP, 2, MXBLK),
*           DX (MXCOMP, 2, MXBLK)
C
  DIMENSION REAC (NCHEM, 2),
*           DRDX (NCHEM, NCHEM, 2)
C
  DIMENSION DDDX (NCHEM, NCHEM)
C
  DIMENSION FKE01 (NCHEM),
*           FKE02 (NCHEM),
*           FKE03 (NCHEM),
*           FKE0C (NCHEM)
  DIMENSION DK1 (NCHEM),
*           DK2 (NCHEM),
*           DK3 (NCHEM),
*           DKC (NCHEM)
  DIMENSION DK1DX (NCHEM, NCHEM),
*           DK2DX (NCHEM, NCHEM),
*           DK3DX (NCHEM, NCHEM),
*           DKCDX (NCHEM, NCHEM)
  DIMENSION DADX (NCHEM),
*           DBDX (NCHEM),
*           DTFDX (NCHEM)
  DIMENSION DISTC (NCHEM),
*           DDCDX (NCHEM, NCHEM),
*           DDCDTF (NCHEM)
C
  DIMENSION FX (MXN, MXBLK),
*           DFDXD (MXN, MXN, MXBLK),
*           DFDXL (MXN, MXN, MXBLK),
*           DFDXU (MXN, MXN, MXBLK),
*           IP (MXN, MXBLK),
*           WK (MXN)
  DIMENSION FM (MXN, MXN, MXFRM),
*           IROWFM (MXFRM),
*           ICOLFM (MXFRM)
  DIMENSION IPOSFM (MXBLK, MXSYS)
C
  DIMENSION EM (MXN, MXN, MXBLK, MXSYS)
  DIMENSION IROWEM (MXSYS)
C
  DIMENSION SYM (MXN*MXSYS, MXN*MXSYS),

```

```

*          Y (MXN*MXSYS),
*          IPSYS (MXN*MXSYS)
C
  DIMENSION TMCONB (MXTAUC), TMCONC (MXTAUC), DTCONS (MXTAUC)
  DIMENSION FFOUT (2, 2, MXUNIT)
C
  NCHEM2 = 2*NCHEM
C
  EPSX = 1.0D-50
C
  DT = DTO
  ISTDY = 0
  NSTEP = 0
  ITER = 0
  ITERM = 0
  IITER = 0
C
  DTPEPS = 1.0D-6
  IF (DTPEPS .GT. DTMIN*0.9D0) THEN
    DTPEPS = DTMIN*0.9D0
  END IF
C
  IF (NCHEM2 .GT. MXN) THEN
    WRITE (6, *) 'ARRAY SIZE ERROR AT TRAN.'
    WRITE (6, *) 'NCHEM : ', NCHEM
    WRITE (6, *) 'MXN : ', MXN
    STOP
  END IF
C
  CALL SETRMP (MXDEST, NXFRM, NUNIT, MXUNIT, MXFRM, MXSTG2,
*            TYPE, MDEST,
*            MSTG, IUDSTA, IUDSTO,
*            IROWFM, ICOLFM,
*            IPOS)
C
  --- TIME ZERO ---
C
  TIME = 0.0D0
C
  --- INTERPOLATE FEED FLOW RATES:FF AND CONCENTRATIONS:XF ---
C
  CALL FEEDS (TIME,
*            ISFIN, FRC,
*            CO, TO,
*            MXCOMP,
*            MXFEED, MXTABL, NFA, NFO,
*            TFEED,
*            FF, NFFTAB, FFTAB,
*            XF, NXFTAB, XFTAB)
C
  --- CALCULATE FLOW RATE ---
C
  CALL CALFLW (MXUNIT, NUNIT, MXSTG, MXDEST, MXEXTR, NEXTRA,
*            MXFEED, NFA, NFO,
*            IFDSTA, IFDSTO, IUDSTA, IUDSTO, MDEST, MEXTRA,
*            FOUT, FFOUT, ACIDIN, FFOUT,
*            FF, FL,
*            FLS, FLSMT, IPFLS,
*            MSTG)
C
  --- CALCULATE VOLUME FRACTION ---
C
  CALL CALCPH (MXUNIT, NUNIT, MXSTG, MXEXTR,
*            TYPE, MSTG, MEXTRA,
*            HEIGHT, PHIOLD, PHI, PHIS,
*            FL, FR)
C
  IF (CTFILE .NE. ' ') THEN
    LFNAM = 30
    DO 200 I = 30, 1, -1
      IF (CTFILE(1:I) .EQ. ' ') LFNAM = I - 1
    CONTINUE
    COUTF=CTFILE(1:LFNAM) //' .outlist'
  END IF
C
  IFOUT = IRST
  OPEN (UNIT=IFOUT, FILE=COUTF, FORM='FORMATTED',
*       STATUS='UNKNOWN')

```

```

ELSE
C      IFOUT = IUPRT
C
C      END IF
C
C      CALL INPRRT(IFOUT ,
*        TITLE ,TIME ,ENDTIM,DT ,
*        CNAME ,FMASS ,DECAYH,CVUNIT,
*        CUNIT ,CHRG ,CNTBP ,FDEG ,
*        MXCOMP,MXUNIT,MXSTG ,MXSTG2,NCHEM ,MXBLK ,
*        MXREAC,MXDEST,MXFEED,MXTABL,
*        MXEXTR,NUNIT ,FRC ,CO ,TO ,CDBP ,
*        NFA ,NFO ,NEXTRA,
*        FF ,NFFTAB,FFTAB ,XF ,NXFTAB,XFTAB ,
*        IFDSTA,IFDSTO,
*        ID ,IREACT,FCHEM ,ECHEM ,TYPE ,MSTG ,
*        IUDSTA,IUDSTO,MDEST ,MEXTRA,TAV ,VTOT ,
*        ACID ,ACIDIN,OXID ,FOUT ,RFOUT ,FFOUT ,
*        IDEXTR,CHRDIS,
*        FL ,FR ,TSTG ,V ,VS ,
*        PHI ,PHIS ,
*        IPOS ,
*        XN )
C
C      WRITE(IUPRT,9000) TIME ,DT ,NSTEP ,ITER ,ITERM ,ITER
C
C      IIPRT = 1
C      TIMEPR = DTPRT(IIPRT)
C      IIPLT = 1
C      TIMEPL = DTPLT(IIPLT)
C
C      IF ( TIMEPL .LT. DT ) THEN
C        IF ( CTFILE .NE. ' ' ) THEN
C          WRITE(COUTP,'(2A,12.2,A)')
*          CTFILE(1:LNAME),',',IIPLT,'plot'
C        ELSE
C          WRITE(COUTP,'(A,12.2,A)') 'mixset.',IIPLT,'plot'
C        ENDIF
C
C      OPEN(UNIT=IIPLT,FILE=COUTP,FORM='FORMATTED',
*        STATUS='UNKNOWN')
C
C      CALL OUTPLT(IIPLT ,
*        TITLE ,TIME ,
*        CNAME ,FMASS ,DECAYH,CVUNIT,
*        CUNIT ,CHRG ,CNTBP ,CPHASE,
*        MXCOMP,MXUNIT,MXSTG ,MXSTG2,NCHEM ,MXBLK ,
*        MXEXTR,NUNIT ,ISFIN ,FRC ,CO ,TO ,
*        ID ,TYPE ,FOUT ,MSTG ,MEXTRA,
*        FL ,TSTG ,WLT ,WST ,HOS ,TF ,
*        IEFF ,EFF ,DIS ,
*        IPOS ,
*        XN )
C
C      CLOSE(UNIT=IIPLT)
C
C      IIPLT = IIPLT + 1
C
C      IF ( DTPLT(IIPLT) .GT. TIMEPL ) THEN
C        TIMEPL = DTPLT(IIPLT)
C      ELSE
C        TIMEPL = ENDTIM
C      ENDIF
C
C      END IF
C
C      1000 CONTINUE
C
C      IF ( DT .GT. DTMAX ) DT = DTMAX
C
C      DTSAVE = DT
C
C      IF ( TIME + DT + 1.0D-5 .GT. TIMEPR ) THEN
C        DT = TIMEPR - TIME
C      END IF
C
C      IF ( TIME + DT + 1.0D-5 .GT. TIMEPL ) THEN
C        DT = TIMEPL - TIME
C      END IF
C
C      IF ( DT .LT. DMIN1( DTO , DTMIN ) ) THEN

```

```

DT = DMIN1( DTO , DTMIN )
END IF
C
C
C
IF ( TIME + DT .GE. TIMEPR ) THEN
  IPR = 1
ELSE
  IPR = 0
END IF
C
IF ( TIME + DT .GE. TIMEPL ) THEN
  IPL = 1
ELSE
  IPL = 0
END IF
C
C
TIME = TIME + DT
NSTEP = NSTEP + 1
C
C
--- INTERPOLATE FEED FLOW RATES:FF AND CONCENTRATIONS:XF ---
C
CALL FEEDS(TIME ,
*   ISFIN ,FRC ,
*   CO ,TO ,
*   MXCOMP,
*   MXFEED,MXTABL,NFA ,NFO ,
*   TFEED ,
*   FF ,NFFTAB,FFTAB ,
*   XF ,NXFTAB,XFTAB )
C
C
--- CALCULATE FLOW RATE ---
C
CALL CALFLW(MXUNIT,NUNIT ,MXSTG ,MXDEST,MXEXTR,NEXTRA,
*   MXFEED,NFA ,NFO ,
*   IFDSTA,IFDSTO,IUDSTA,IUDSTO,MDEST ,MEXTRA,
*   FOUT ,RFOUT ,ACIDIN,FFOUT ,
*   FL ,FL ,
*   FLS ,FLSMT ,IPFLS ,
*   MSTG )
C
C
--- CALCULATE VOLUME FRACTION ---
C
CALL CALCPH(MXUNIT,NUNIT ,MXSTG ,MXEXTR,
*   TYPE ,MSTG ,MEXTRA,
*   HEIGHT,PHIOLD,PHI ,PHIS ,
*   FL ,FR )
C
C
ITER = 0
ITERM = 0
C
DO 1700 JJJ = NCHEM , MXCOMP
C
  IF ( JJJ .EQ. NCHEM ) THEN
    JM1C = 0
    JBASE = 0
    NCOMP = NCHEM
  ELSE
    JM1C = JJJ
    JBASE = JJJ - 1
    NCOMP = 1
  END IF
C
  NM = NCOMP*2
C
  --- ITERATION OF NEWTON'S METHOD ---
C
  ICONV = 0
C
  DO 1500 L = 1 , ITMAX
C
    IF ( ITER .LT. L ) THEN
      ITER = L
    END IF
C
    --- MAKE JACOBIAN ---
C
    CALL MAKEJB(MXDEST,MXCOMP,MXUNIT,MXSTG ,MXSTG2,NCHEM ,
*   MXBLK ,JM1C ,JBASE ,NCOMP ,MXREAC,MXEXTR,
*   NUNIT ,NBLK ,ISTDY ,DT ,
*   FRC ,CO ,TO ,RCON ,CHRG ,CNTBP ,
*   MXFEED,NFA ,NFO ,
*   FF ,XF ,IFDSTA,IFDSTO,
*   TYPE ,MSTG ,IUDSTA,IUDSTO,MDEST ,MEXTRA,
*   TAV ,VTOT ,ACID ,OXID ,FOUT ,RFOUT ,

```

```

*          IDIST ,FCHEM ,ECHEM ,CDIS ,CKEQU ,
*          IEFF ,EFF ,
*          DIS ,TSTG ,V ,VS ,FL ,FR ,
*          PHIOLD,PHI ,PHIS ,TF ,CDBP ,
*          IPOS ,XN ,X ,
*          REAC ,DRDX ,DDDX ,
*          FKEO1 ,FKEO2 ,FKEO3 ,FKEOC ,
*          DK1 ,DK2 ,DK3 ,DKC ,
*          DK1DX ,DK2DX ,DK3DX ,DKCDX ,
*          DADX ,DBDX ,DTFDX ,DISTC ,DDCDX ,DDCDTF ,
*          MXN ,MXRM ,NXRM ,
*          FX ,DFDX ,DFDXL ,DFDXU ,RM )
C
C          --- SOLVE EQUATION ---
C
*          CALL CAPSLV(NXRM ,MXRM ,MXSYS ,
*          NBLK ,MXN ,MXBLK ,
*          NM ,DFDX ,DFDXL ,DFDXU ,
*          FX ,IP ,WK ,
*          RM ,IROWRM ,ICOLRM ,IPOSRM ,
*          EM ,IROWEM ,
*          SYM ,Y ,IPSYS ,EPSX ,IER )
C
C          IF ( IER .NE. 0 ) THEN
C              WRITE(6,*) ' MATRIX SINGULAR.'
C              GO TO 1600
C          END IF
C
C          DO 1200 I = 1 , NBLK
C              DO 1100 J = 1 , NCOMP
C                  JJ = J + JBASE
C                  DX(JJ,1,I) = FX(J ,I)
C                  DX(JJ,2,I) = FX(J+NCOMP,I)
C              CONTINUE
C          CONTINUE
C
C          --- ADD INCREMENTS ---
C
C          DO 1400 I = 1 , NBLK
C              DO 1300 J = 1 , NCOMP
C
C                  JJ = J + JBASE
C
C                  XN(JJ,1,I) = XN(JJ,1,I) + DX(JJ,1,I)
C                  XN(JJ,2,I) = XN(JJ,2,I) + DX(JJ,2,I)
C
C                  IF ( DABS(XN(JJ,1,I)) .GT. 10.000 ) GO TO 1600
C                  IF ( DABS(XN(JJ,2,I)) .GT. 10.000 ) GO TO 1600
C
C                  IF (
C                      JJ .NE. 10TCU6
C                      .AND. JJ .NE. 10TCP4
C                      .AND. JJ .NE. 10TCZR ) THEN
C                      IF ( XN(JJ,1,I) .LT. 0.000 ) THEN
C                          XN(JJ,1,I) = 0.000
C                      END IF
C                  END IF
C
C                  IF ( XN(JJ,2,I) .LT. 0.000 ) THEN
C                      XN(JJ,2,I) = 0.000
C                  END IF
C
C              CONTINUE
C          CONTINUE
C
C          --- CALCULATE RELATIVE ERROR ---
C
C          ICONVO = ICONV
C          ITERL = L
C
C          CALL CONVCK(ICONV ,EPS ,RXLIM ,EPSLIM ,ITMAX ,ITERL ,
C          MXCOMP ,MXUNIT ,MXSTG2 ,
C          JMIC ,JBASE ,NCOMP ,MXBLK ,NJUNIT ,
C          NBLK ,TYPE ,MSTG ,
C          IPOS ,
C          XN ,DX )
C
C          IF ( ITERM .LT. L ) THEN
C              IF ( ICONVO .EQ. 0 .AND. ICONV .NE. 0 ) THEN
C                  ITERM = L
C              ELSE IF ( ICONV .EQ. 0 ) THEN
C                  ITERM = L
C              END IF
C          END IF
C
C          --- JUDGEMENT OF CONVERGENCE ---
C
C          IF( ICONV .EQ. 2 ) GO TO 1700
C
C          --- JUDGEMENT OF CONVERGENCE(WEAK CONVERGENCE) ---
C
C          IF( ICONV .EQ. 1 ) GO TO 1700
C
C          1600 CONTINUE
C
C          ... ITERATION EXCEEDED ITMAX OR MATRIX SINGULAR
C
C          IITER = IITER + ITER
C          WRITE(IUPRT,9100) TIME ,DT ,NSTEP ,ITER ,IITER ,JMIC
C
C          TIME = TIME - DT
C          NSTEP = NSTEP - 1
C          DT = 0.500 * DT
C
C          IF( DT .LT. DTMIN ) THEN
C              WRITE(IUPRT,*) ' ***** DT BECAME LESS THAN DTMIN *****'
C              GO TO 8000
C          END IF
C
C          ... RETRIEVE OLD VALUES
C
C          DO 1620 I = 1 , NBLK
C              DO 1610 J = 1 , MXCOMP
C                  XN(J,1,I) = X(J,1,I)
C                  XN(J,2,I) = X(J,2,I)
C              CONTINUE
C          CONTINUE
C
C          GO TO 1000
C
C          1700 CONTINUE
C
C          --- ITERATION HAS CONVERGED. ---
C
C          CALL CUT(MXCOMP ,MXBLK ,NBLK ,XN )
C
C          DO 3200 I = 1 , NBLK
C              DO 3100 J = 1 , MXCOMP
C                  X(J,1,I) = XN(J,1,I)
C                  X(J,2,I) = XN(J,2,I)
C              CONTINUE
C          CONTINUE
C
C          --- OUTPUT ---
C
C          IITER = IITER + ITER
C
C          IF (NSTEP .LT. 1000) THEN
C              WRITE(IUPRT,9000) TIME ,DT ,NSTEP ,ITER ,ITERM ,IITER
C          ENDIF
C
C          IF ( IPR .NE. 0 ) THEN
C              CALL OUTPUT(IFOUT ,IPRDIS ,
C              TITLE ,TIME ,ENDTIM ,DT ,
C              CNAME ,FMASS ,DECAYH ,CVUNIT ,
C              CUNIT ,CHRG ,CNTBP ,CPHASE ,
C              MXCOMP ,MXUNIT ,MXSTG ,MXSTG2 ,NCHEM ,MXBLK ,
C              MEXTR ,NJUNIT ,ISFIN ,FRC ,CO ,TO ,
C              ID ,TYPE ,FOUT ,MSTG ,MEXTRA ,
C              FL ,TSTG ,WLT ,WST ,HOS ,TF ,
C              IEFF ,EFF ,DIS ,
C              IPOS ,
C              XN )
C              CALL OUTPTF(IFOUT ,IPRDIS ,
C              TITLE ,TIME ,ENDTIM ,DT ,
C              CNAME ,FMASS ,DECAYH ,CVUNIT ,
C              CUNIT ,CHRG ,CNTBP ,CPHASE ,

```

```

*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,WLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
CALL OUTMIC(IFOUT,IPRDIS,
*      TITLE,TIME,ENDTIM,DT,
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,CPHASE,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,WLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
CALL OUTADD(IFOUT,IPRDIS,
*      TITLE,TIME,ENDTIM,DT,
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,CPHASE,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,WLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
C
IIPRT = IIPRT + 1
C
IF ( DTPRT(IIPRT) .GT. TIMEPR ) THEN
    TIMEPR = DTPRT(IIPRT)
ELSE
    TIMEPR = ENDTIM
ENDIF
C
END IF
C
IF ( IPL .EQ. 1 ) THEN
C
    IF ( TIMEPL .GE. ENDTIM ) THEN
        TIMEPL = ENDTIM
        IF ( CTFILE .NE. ' ' ) THEN
            COUPT=CTFILE(1:LFNAME)///'.plot'
        ELSE
            COUPT='mixset'///'.plot'
        ENDIF
    ELSE
        IF ( CTFILE .NE. ' ' ) THEN
            WRITE(COUPT,'(2A,12.2,A)')
            CTFILE(1:LFNAME),'.',IIPRT,'plot'
        ELSE
            WRITE(COUPT,'(A,12.2,A)') 'mixset.',IIPRT,'plot'
        ENDIF
    END IF
    OPEN(UNIT=IUPLT,FILE=COUPT,FORM='FORMATTED',
        *      STATUS='UNKNOWN')
C
    CALL OUTPLT(IUPLT,
*      TITLE,TIME,
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,CPHASE,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,ISFIN,FRC,CO,TO,
*      ID,TYPE,FOUT,MSTG,MEXTRA,
*      FL,TSTG,WLT,WST,HOS,TF,
*      IEFF,EFF,DIS,
*      IPOS,
*      XN )
C
    CLOSE(UNIT=IUPLT)
C
    IIPLT = IIPLT + 1
C
    IF ( DTPLT(IIPLT) .GT. TIMEPL ) THEN
        TIMEPL = DTPLT(IIPLT)
    ELSE
        TIMEPL = ENDTIM
    ENDIF
C
END IF
C
C
C
IF ( IPR .EQ. 1 .OR. IPL .EQ. 1 ) THEN

```

```

C      DT = DTSAVE
C
ELSE
C
    IF ( ITERM .LE. ITMIN ) THEN
        DT = TRATE * DT
    END IF
C
END IF
C
DO I = 1, MXTAUC
    IF ( TIME+DT+1.0D-5 .GE. TMCONB(I) ) THEN
        IF ( TIME .LE. TMCONE(I) ) THEN
            IF ( TIME .GE. TMCONB(I) ) THEN
                IF ( DT .GT. DTCONS(I) ) THEN
                    DT = DTCONS(I)
                    I = MXTAUC
                END IF
            ELSE
                DT = TMCONB(I) - TIME
                I = MXTAUC
            END IF
        END IF
    END IF
END DO
C
C
C
IF ( TIME .LT. ENDTIM ) GO TO 1000
C
C
8000 CONTINUE
C
CALL TBPDEG(IFOUT,
*      TITLE,TIME,ENDTIM,DT,
*      CNAME,FMASS,DECAYH,CVUNIT,
*      CUNIT,CHRG,CNTBP,FDEG,
*      MXCOMP,MXUNIT,MXSTG,MXSTG2,NCHEM,MXBLK,
*      MXEXTR,NUNIT,FRC,CO,TO,
*      ID,TYPE,MSTG,MEXTRA,FL,
*      TSTG,V,VS,PHI,PHIS,
*      IPOS,
*      XN )
C
CALL MATBL (IFOUT,
*      TITLE,TIME,ENDTIM,DT,
*      MXCOMP,MXDEST,MXUNIT,
*      MXSTG2,MXBLK,NUNIT,
*      FMASS,
*      RXLIM,EPS,
*      MXFEED,NFA,NFO,
*      FF,XF,IFDSTA,IFDSTO,
*      ID,TYPE,MSTG,IUDSTA,IUDSTO,
*      MDEST,MEXTRA,TAV,VTOT,ACID,
*      ACIDIN,OXID,FIN,FOUT,RFOUT,
*      FMSIN,FMSOUT,FMTIN,FMTOUT,RMTBL,
*      IPOS,XN )
C
IF ( CTFILE .NE. ' ' ) THEN
    CLOSE(UNIT=IFOUT)
END IF
C
RETURN
C
C
9000 FORMAT(1H,'TIME =',F9.3,' (HR)',
*      1X,'STEP =',F7.3,
*      1X,'NO. =',15,
*      1X,'ITER =',12,
*      1X,'ITERM =',12,
*      1X,'SUM =',16)
9100 FORMAT(1H,'TIME =',F9.3,' (HR)',
*      1X,'STEP =',F7.3,
*      1X,'NO. =',15,
*      1X,'ITER =',12,
*      1X,'SUM =',16,
*      1X,'JMIC =',12/
*      1H,'ERROR DETECTED AND TIME STEP HAS BEEN CHANGED.')
C
END

```

```

SUBROUTINE TRIBLK(MXN,M,NM,D,E,F,B,IP,WK,EPSX,IER)
C
C LU DECOMPOSITION METHOD FOR A TRI-DIAGONAL BLOCK MATRIX.
C
C
C | D(1) F(1) |
C | E(2) D(2) F(2) |
C | ..... |
C | ..... |
C | ..... |
C | ..... |
C | ..... |
C | ..... |
C | ..... E(M-1) D(M-1) F(M-1) |
C | ..... E(M) D(M) |
C
C
C PARAMETERS:
C
C M : NUMBER OF DIAGONAL BLOCKS.
C MXN : THE LEADING DIMENSION OF EACH DIAGONAL BLOCK.
C NM : THE ORDER OF SUB-MATRIX ON BLOCK DIAGONAL PART.
C D : SUB-MATRIX : DIAGONAL BLOCK OF TRI-DIAGONAL PART.
C E : SUB-MATRIX : LOWER BLOCK OF TRI-DIAGONAL PART.
C F : SUB-MATRIX : UPPER BLOCK OF TRI-DIAGONAL PART.
C B : THE RIGHT HAND SIDE VECTOR FOR EACH DIAGONAL BLOCK.
C OR SOLUTION VECTOR FOR EACH DIAGONAL BLOCK.
C IP : PIVOT NUMBERS.
C WK : THE ARRAY FOR WORKING.
C EPSX : THE TOLERANCE FOR PIVOTAL ELEMENTS.
C IER : THE ERROR CODE. IF IER=0, NORMAL RETURN.
C
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C DIMENSION D (MXN,MXN,M),
* E (MXN,MXN,M),
* F (MXN,MXN,M),
* B (MXN,M),
* IP (MXN,M),
* WK (MXN,M)
C
C IER = 0
C
C DO 700 I = 2, M
C
C CALL INVGJ(MXN, NM, D(1,1,I-1), IP, EPSX, IER)
C IF ( IER .NE. 0 ) THEN
C RETURN
C END IF
C
C DO 300 J = 1, NM
C
C DO 100 L = 1, NM
C WK(L) = F(L,J,I-1)
C 100 CONTINUE
C
C DO 220 K = 1, NM
C S = 0.0D0
C DO 210 L = 1, NM
C S = S + D(K,L,I-1) * WK(L)
C 210 CONTINUE
C F(K,J,I-1) = S
C
C 220 CONTINUE
C
C
C
C
C DO 600 J = 1, NM
C
C DO 500 K = 1, NM
C
C S = 0.0D0
C DO 400 L = 1, NM
C S = S + E(K,L,I) * F(L,J,I-1)
C 400 CONTINUE
C D(K,J,I) = D(K,J,I) - S
C
C 500 CONTINUE
C
C
C
C 600 CONTINUE

```

```

C
C 700 CONTINUE
C
C CALL INVGJ(MXN, NM, D(1,1,M), IP, EPSX, IER)
C IF ( IER .NE. 0 ) THEN
C RETURN
C END IF
C
C
C FORWARD SUBSTITUTION
C
C DO 1100 K = 1, NM
C WK(K) = B(K,1)
C 1100 CONTINUE
C
C DO 1220 K = 1, NM
C S = 0.0D0
C DO 1210 L = 1, NM
C S = S + D(K,L,1) * WK(L)
C 1210 CONTINUE
C B(K,1) = S
C
C 1220 CONTINUE
C
C DO 1700 I = 2, M
C
C DO 1400 K = 1, NM
C S = 0.0D0
C DO 1300 L = 1, NM
C S = S + E(K,L,I) * B(L,I-1)
C 1300 CONTINUE
C B(K,I) = B(K,I) - S
C 1400 CONTINUE
C
C DO 1500 K = 1, NM
C WK(K) = B(K,I)
C 1500 CONTINUE
C
C DO 1620 K = 1, NM
C S = 0.0D0
C DO 1610 L = 1, NM
C S = S + D(K,L,I) * WK(L)
C 1610 CONTINUE
C B(K,I) = S
C
C 1620 CONTINUE
C
C 1700 CONTINUE
C
C
C BACKWARD SUBSTITUTION
C
C DO 2300 I = M-1, 1, -1
C
C DO 2200 K = 1, NM
C S = 0.0D0
C DO 2100 L = 1, NM
C S = S + F(K,L,I) * B(L,I+1)
C 2100 CONTINUE
C B(K,I) = B(K,I) - S
C 2200 CONTINUE
C
C 2300 CONTINUE
C RETURN
C END

```