

JAERI-M
90-018

アクチノイド元素の硝酸水溶液系
化学反応数値モデル：

R E A C T

1990年2月

館盛 勝一

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1990

編集兼発行 日本原子力研究所
印刷 日立高速印刷株式会社

アクチニド元素の硝酸水溶液系化学反応数値モデル：REACT

日本原子力研究所東海研究所燃料安全工学部

館盛 勝一

(1990年1月26日受理)

ウラン、プルトニウム、ネプツニウムの硝酸水溶液系における原子価変化をシミュレートする、アクチニド元素の化学反応数値モデル：REACTコードを開発した。いくつかの還元剤も含む酸化還元反応や不均化反応、放射線による亜硝酸の生成と分解反応等27個の化学反応の速度式がREACTコードに組込まれている。これらの速度式を解くために、porsing法等8種の数値解法が反応系に応じて選択できるようになっている。

本報では、化学反応と反応速度式、数値解法といったコードの内容といくつかの計算例を示し、Appendixにコードのマニュアルとプログラムのソースファイル一式を載せた。

A mathematical Model for Chemical Reactions
with Actinide Elements in the Aqueous Nitric Acid Solution:
REACT

Shoichi TACHIMORI

Department of Fuel Safety Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received January 26, 1990)

A mathematical model of chemical reactions with actinide elements: REACT code, was developed to simulate change of valency states of U, Pu and Np in the aqueous nitric acid solution. Twenty seven rate equations for the redox reactions involving some reductants, disproportionation reactions, and radiolytic growth and decay reactions of nitrous acid were programmed in the code. Eight numerical solution methods such as Porsing method to solve the rate equations were incorporated parallel as options depending on the characteristics of the reaction systems.

The present report gives a description of the REACT code, e.g., chemical reactions and their rate equations, numerical solution methods, and some examples of the calculation results. A manual and a source file of the program was attached to the appendix.

Keywords: Mathematical Model, Actinide Elements, REACT Code, Nitric Acid, Redox Reactions, Uranium, Neptunium, Plutonium, Nitrous Acid, Disproportionation, Program

目 次

1. 序 言	1
2. モデルの概要	2
2.1 化学反応式	2
2.2 計算モデル	7
3. 数値計算手法	10
3.1 常微分方程式の解法	10
3.2 連立方程式	11
3.3 Jacobian の解析解	11
3.4 時間キザミ幅	18
3.5 その他の解法	18
3.6 解法を選択について	19
4. REACTによる計算実行例	20
4.1 Pu(IV)のウラナスU(IV)またはHAN(硝酸ヒドロキシルアミン)による還元反応	20
4.2 Pu(III)の再酸化反応	20
5. 結 び	25
謝 辞	25
引用文献	25
付 録	27
A.1 入力データ及びJCLの説明	27
1.1 入力データの説明	27
1.2 JCLの説明	35
A.2 プログラムの概要	38
2.1 木構造	38
2.2 コモンの説明	40
2.3 ルーチンの説明	46
A.3 プログラムリスト	51

Contents

1. Introduction	1
2. REACT model	2
2.1 Chemical reactions	2
2.2 Mathematical model	7
3. Numerical solution	10
3.1 Numerical solution of differential equations	10
3.2 Simultaneous equations	11
3.3 Analytic solution of Jacobian	11
3.4 Time interval for iterative procedure	18
3.5 Other solution methods	18
3.6 Selection of the solution method	19
4. Results of calculation by REACT for some reaction systems	20
4.1 Reduction of Pu(IV) with U(IV) or HAN(hydroxylamine nitrate) ...	20
4.2 Reoxidation reaction of Pu(III)	20
5. Conclusion	25
Acknowledgements	25
References	25
Appendices	27
A.1 Input data and JCL command	27
1.1 Input data	27
1.2 JCL	35
A.2 Description of Program	38
2.1 Tree structure	38
2.2 Common	40
2.3 Routine	46
A.3 Fortran listing of program REACT	51

1. 序 言

核燃料サイクル化学工程における主要なアクチニド元素（特にU, Pu, Np）のふるまいを解析する場合，それぞれの工程で起こる化学反応等の現象を，適切な数値モデルに置き換え，工程シミュレーションを実施すると共に，実際の工程条件での実験により，上記モデルを検証する方法が考えられる。

著者らが進めてきた再処理抽出工程（Purex）のモデル開発においても，アクチニド元素等の抽出に伴う物質移行の他に，硝酸水溶液あるいは抽出溶媒（有機相）中での化学反応（放射線化学的反応，酸化還元反応，不均化反応等）による各成分濃度および原子価分布の変化を評価しなければならない。

そのような理由から，ある特定の溶媒中でのアクチニド元素の化学的ふるまいをシミュレートする計算モデルが要求される。

再処理工程等核燃料サイクル諸工程の場合，アクチニド元素の溶液化学の主な舞台は硝酸水溶液である。プルトニウムやネプツニウムの，硝酸水溶液系における各種化学反応の速度論的研究は，今迄に数多く行われており，各反応の速度式も求まっている¹⁾。また，放射線効果に関して各国で詳細な研究が行われてきた^{2), 3)}。しかし，それらを数値モデルに変換し，多成分系における多数の化学反応の競合過程を解析評価した研究はほとんどないといってさしつかえない。ここに例としてあげることができるものは，硝酸水溶液の放射線分解反応を計算機シミュレーションし，各生成物の生成量を求めた我が国の泉田（日立エネ研）の研究⁴⁾と，プルトニウムの硝酸水溶液について，放射線分解生成物とプルトニウムとの反応を取扱ったソ連のVladimirovaの研究^{5), 6)}のみである。彼らの研究成果は，並列的に同時進行する多くの反応を数値モデルとして解く方法が，実際の水溶液系で起こる現象を十分正しく予測し得る展望を示している。

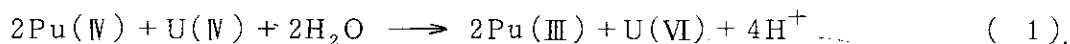
本計算コード：REACTは以上の背景のもとに，U, Pu, Np等が硝酸水溶液中で示す各種の化学反応を，数値モデル化し，平衡論よりはむしろ速度論に基づいて，過渡現象を追跡するために開発したものである。

本報では，REACTコードに現在組込まれている化学反応式とその速度式およびその数式解を示すと共に，微分方程式のいくつかの解法について計算機のCPU時間と計算結果について比較した結果を示した。プログラムでは，8種類の解法から選択することにした。また，付録にREACTのプログラムソースファイルを載せた。

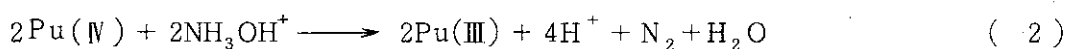
2. モデルの概要

2.1 化学反応式

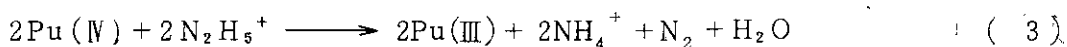
REACTコードには、次に示す27個の化学反応式の数式が組込んである。



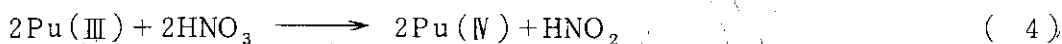
$$k_1 \frac{[\text{Pu}(\text{V})][\text{U}(\text{V})]}{([\text{H}^+] + 0.05)^2}$$



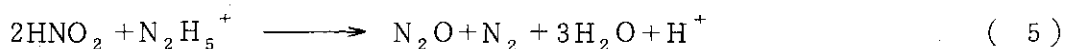
$$k_2 \frac{[\text{Pu}(\text{V})]^2 [\text{HAN}]^2}{[\text{Pu}(\text{III})]^2 [\text{H}^+]^4 (\text{C}_\text{N} + 0.33)^2}$$



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



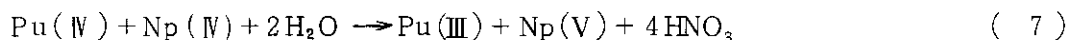
$$k_4 [\text{Pu}(\text{III})][\text{HNO}_2]^{0.5} [\text{H}^+]^{0.5} [\text{C}_\text{N}]^{0.4}$$



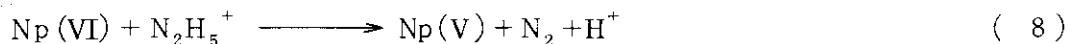
$$k_5 [\text{HNO}_2][\text{N}_2\text{H}_5^+][\text{H}^+]$$



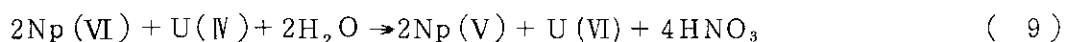
$$k_6 [\text{Np}(\text{IV})][\text{Np}(\text{VI})](2.16 + 12.5\text{C}_\text{N})$$



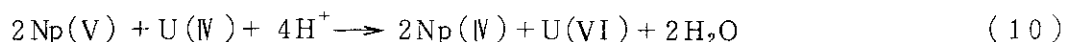
$$k_7 [\text{Np}(\text{IV})][\text{Pu}(\text{V})][\text{H}^+]^{-4}$$



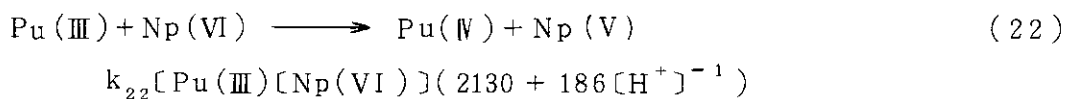
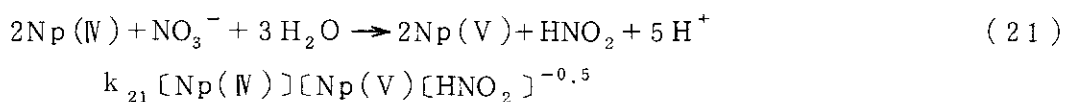
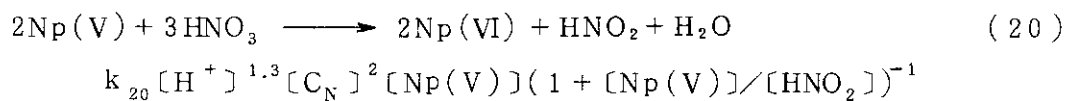
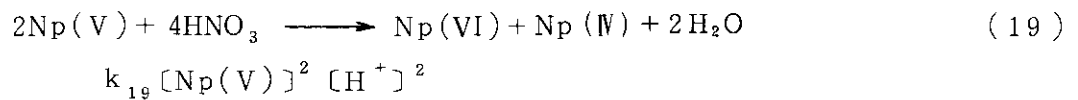
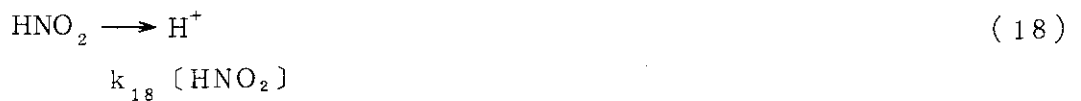
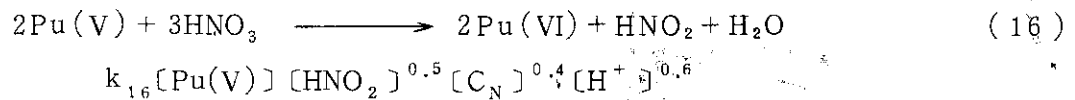
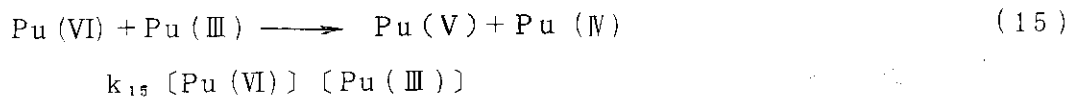
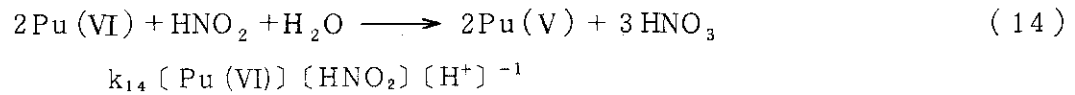
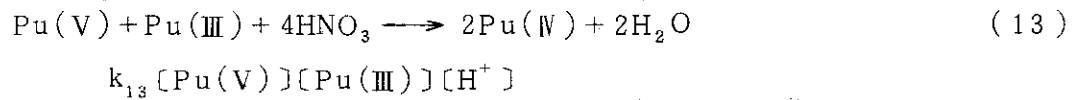
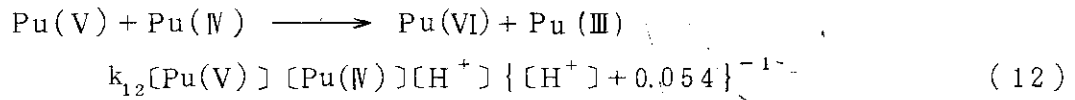
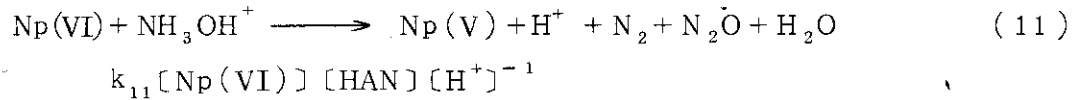
$$k_8 [\text{Np}(\text{VI})][\text{N}_2\text{H}_5^+][\text{H}^+]^{-1.3}$$

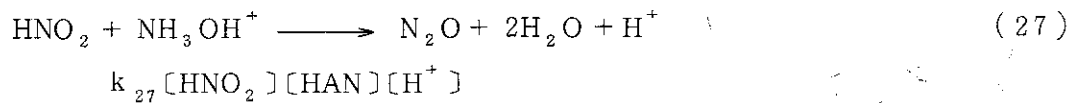
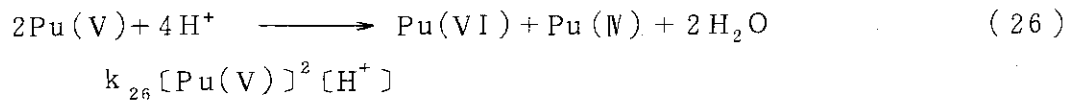
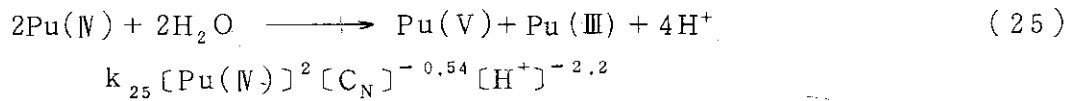
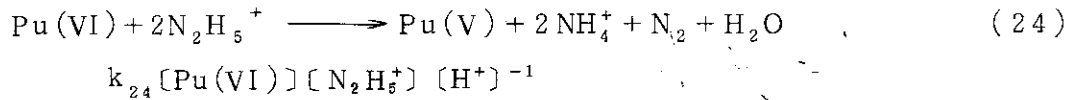
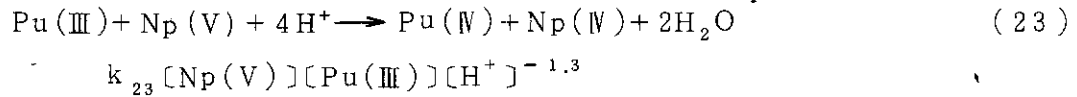


$$k_9 [\text{Np}(\text{VI})][\text{U}(\text{V})]$$



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





この場合、 $C_N = C_H + C_{\text{NP}_5} + C_{\text{PU}_5} + C_{\text{HAN}} + C_{\text{Hydrazin}} + 2(C_{\text{U}_6} + C_{\text{PU}_6} + C_{\text{NP}_6})$
 $+ 3C_{\text{PU}_3} + 4(C_{\text{U}_4} + C_{\text{PU}_4} + C_{\text{NP}_4})$

ここで C_M は全て mole/dm³ 濃度で表記する。

以上の速度式においては、濃度単位は体積モル濃度 (mole/dm³) を、時間単位は分 (min) を採用した。これら27個の反応速度定数を表2.1に示す。

表 2.1 化学量論係数 C_{ij} 及び反応速度定数 k_i (25°C)

反応式	U(VI)	Pu(W)	HNO ₃	U(W)	Pu(III)	HNO ₃	N ₂ H ₅ ⁺	HAN	Pu(V)	Pu(VI)	Np(W)	Np(V)	Np(VI)	K _j mol·min	E _i kJ/mol	引用 文献
	x ₁₀	x ₃	x ₅	x ₁₁	x ₄	x ₆	x ₁₃	x ₁₂	x ₂	x ₁	x ₉	x ₈	x ₇			
①	+1	-2	+4	-1	+2									5000	104	7)
②		-1	+2		+1			-1						0732	130	8)
③		-1			+1		-1							0038	93	9)
④		+2	-2		-2	+1								0.7	59	10)
⑤			+1			-2	-1							30000	36	11) 12)
⑥			+4								-1	+2	-1	2.5	103	13)
⑦		-1	+4		+1						-1	+1		70	145	14)
⑧							-1					+1	-1	8.3	7.0.3	15)
⑨	+1		+4	-1								+2	-2	7		16)
⑩	+1		-4	-1							+2	-2		2.25	40.2	17)
⑪			+1					-1				+1	-1	92.1	82	18)
⑫		-1			+1				-1	+1				250	59	19)
⑬		+2	-4		-1				-1					37	43	20)

表 2.1 (続き)

反応式	U(VI)	Pu(W)	HNO ₃	U(W)	Pu(III)	HNO ₂	N ₂ H ₅ ⁺	HAN	Pu(V)	Pu(VI)	Np(W)	Np(V)	Np(VI)	K _j	E _i	引用 文献
	X ₁₀	X ₃	X ₅	X ₁₁	X ₄	X ₆	X ₁₃	X ₁₂	X ₂	X ₁	X ₉	X ₈	X ₇	mol·min kJ/mol	kJ/mol	
⑭			+3			-1			+2	-2				7.9	111	21)
⑮		+1			-1				+1	-1				276	23	22)
⑯			-3			+1			-2	+2				3.575	92	23)
⑰			-1			+1								10 ⁻⁴		
⑱			+1			-1								0.02		
⑲			-4								+1	-2	+1	8×10 ⁻⁴	68.7	24)
⑳			-3			+1						-2	+2	9×10 ⁻⁴	62.8	25)
㉑			+5			+1					-2	+2		4.8×10 ⁻⁴	105	25)
㉒		+1			-1							+1	-1	1.0	17.2	26)
㉓		+1	-4		-1						+1	-1		0.0598	85	27)
㉔							-2		+1	-1				0.314	67.4	28)
㉕		-2	+4		+1				+1					1.65×10 ⁻³	105	29)
㉖		+1	-4						-2	+1				0.345	82	22)
㉗			+1			-1		-1						436	48.1	30)

2.2 計算モデル

2.1節で示したように、考慮されている化学反応に係わる化学種は13個であり、それぞれを次の様におく。

$$x_1 = [\text{Pu(VI)}]$$

$$x_2 = [\text{Pu(V)}]$$

$$x_3 = [\text{Pu(IV)}]$$

$$x_4 = [\text{Pu(III)}]$$

$$x_5 = [\text{H}^+]$$

$$x_6 = [\text{HNO}_2]$$

$$x_7 = [\text{Np(VI)}]$$

$$x_8 = [\text{Np(V)}]$$

$$x_9 = [\text{Np(IV)}]$$

$$x_{10} = [\text{U(VI)}]$$

$$x_{11} = [\text{U(IV)}]$$

$$x_{12} = [\text{HAN}]$$

$$x_{13} = [\text{N}_2\text{H}_5^+]$$

また、(1)～(26)式の反応速度を v_1, v_2, \dots, v_{27} と表記すれば、上式よりそれぞれは次の様に表わされる。

$$v_1 = k_1 x_3 x_{11} (x_5 + 0.05)^{-2}$$

$$v_2 = k_2 x_3^2 x_{12}^2 x_4^{-2} x_5^{-4} (C_N + 0.35)^{-2}$$

$$v_3 = k_3 x_3 x_{13} (x_5 + 0.35)^{-1}$$

$$v_4 = k_4 x_4 x_6^{0.5} x_5^{0.5} C_N^{0.4}$$

$$v_5 = k_5 x_6 x_{13} x_5^2$$

$$v_6 = k_6 x_9 x_7 (2.16 + 12.5 C_N)$$

$$v_7 = k_7 x_9 x_3 x_5^{-4}$$

$$v_8 = k_8 x_7 x_{13} x_5^{-1.3}$$

$$v_9 = k_9 x_7 x_{11}$$

$$v_{10} = k_{10} x_8 x_{11} (1.6 x_5^{-2} + 1.42 x_5)$$

$$v_{11} = k_{11} x_7 x_{12} x_5^{-1}$$

$$v_{12} = k_{12} x_2 x_3 x_5 (x_5 + 0.054)^{-1}$$

$$v_{13} = k_{13} x_2 x_4 x_5$$

$$v_{14} = k_{14} x_1 x_6 x_5^{-1}$$

$$v_{15} = k_{15} x_1 x_4$$

$$v_{16} = k_{16} x_2 x_6^{0.5} C_N^{0.4} x_5^{0.6}$$

$$v_{17} = k_{17} x_5$$

$$v_{18} = k_{18} x_6$$

$$v_{19} = k_{19} x_8^2 x_5^2$$

$$v_{20} = k_{20} x_5^{1.3} C_N^2 x_8 \left(1 + \frac{x_8}{x_6}\right)^{-1}$$

$$v_{21} = k_{21} x_9 x_8 x_6^{-0.5}$$

$$v_{22} = k_{22} x_4 x_7 (2130 + 186/x_5)$$

$$v_{23} = k_{23} x_8 x_4 x_5^{-1.3}$$

$$v_{24} = k_{24} x_1 x_5^{-1} x_{13}$$

$$v_{25} = k_{25} x_3^2 C_N^{-0.54} x_5^{-2.2}$$

$$v_{26} = k_{26} x_2^2 x_5$$

$$v_{27} = k_{27} x_5 x_6 x_{12}$$

$$C_N = x_5 + x_2 + x_8 + x_{12} + x_{13} + 2(x_1 + x_7 + x_{10}) + 3x_4 + 4(x_3 + x_9 + x_{11})$$

化学反応式中の化学量論係数を C_{ik} (i : 化学種番号, k : 反応式の番号。左辺に表われる場合は消費されるので負号をつけ, 右辺では生成するので正号をつける。式中に表われない時は 0 となる) とすれば, 各イオン濃度の変化量 $f(X)$ は, 以下のように表わされる。

$$f_i(X) = \frac{dx_i}{dt} = \sum C_{ik} \cdot v_k$$

$$\text{例えば } f_1(X) = \frac{dx_1}{dt} = v_{12} - 2v_{14} - v_{15} + 2v_{16} - v_{24} + v_{26}$$

これより,

$$\frac{d}{dt} [\text{Pu(VI)}] = v_{12} - 2v_{14} - v_{15} + 2v_{16} - v_{24} + v_{26}$$

$$\frac{d}{dt} [\text{Pu(V)}] = -v_{12} - v_{13} + 2v_{14} + v_{15} - 2v_{16} + v_{24} + v_{25} - 2v_{26}$$

$$\frac{d}{dt} [\text{Pu(IV)}] = -2v_1 - v_2 - v_3 + 2v_4 - v_7 - v_{12} + 2v_{13} + v_{15} + v_{22} + v_{23} + v_{25} + v_{26}$$

$$\frac{d}{dt} [\text{Pu(III)}] = 2v_1 + v_2 + v_3 - 2v_4 + v_7 + v_{12} - v_{13} - v_{15} - v_{22} - v_{23} + v_{25}$$

$$\frac{d}{dt} [\text{H}^+] = 4v_1 + 2v_2 - 2v_4 + 4v_6 + 4v_7 + 4v_9 - 4v_{10} + v_{11} - 4v_{13} + 3v_{14} - 3v_{16} \\ - v_{17} + v_{18} - 4v_{19} - 3v_{20} + 5v_{21} - 4v_{23} + 4v_{25} - 4v_{26} + v_{27}$$

$$\frac{d}{dt} [\text{HNO}_2] = v_4 - v_5 - v_{14} + v_{16} + v_{17} - v_{18} + v_{20} + v_{21} - v_{27}$$

$$\frac{d}{dt} [\text{Np(VI)}] = -v_6 - v_8 - 2v_9 - v_{11} + v_{19} + 2v_{20} - v_{22}$$

$$\frac{d}{dt} [\text{Np(V)}] = 2v_6 + v_7 + v_8 + 2v_9 - 2v_{10} + v_{11} - 2v_{19} - 2v_{20} + 2v_{21} + v_{22} - v_{23}$$

$$\frac{d}{dt} [\text{Np(IV)}] = -v_6 - v_7 + 2v_{10} + v_{19} - 2v_{21} + v_{23}$$

$$\frac{d}{dt} [\text{U(VI)}] = v_1 + v_9 + v_{10}$$

$$\frac{d}{dt} [\text{U(IV)}] = -v_1 - v_9 - v_{10}$$

$$\frac{d}{dt} [\text{HAN}] = -v_2 - v_{11} - v_{27}$$

$$\frac{d}{dt} [\text{N}_2\text{H}_5^+] = -v_3 - v_5 - v_8 - 2v_{24}$$

3. 数値計算手法

3.1 常微分方程式の解法

本コードでは通常の Jacobian を用いて常微分方程式を解いている。これは2次までの Taylor 展開に基づいており、以下のような形式である。

a) Porsing 法

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

X^{n+1} について解くと、

$$X^{n+1} = X^n + \Delta t (I - \Delta t J_n)^{-1} F(X^n)$$

b) 修正 Porsing 法

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

$$= F(X^n) + \frac{1}{2} J_n (X^{n+1} - X^n)$$

X^{n+1} について解くと、

$$X^{n+1} = X^n + \Delta t \left(I - \frac{1}{2} \Delta t J_n \right)^{-1} F(X^n)$$

ここで、 I は単位行列、 Δt は時間きざみ幅、 J_n は Jacobian $(J_{ij})^n$ である (n はタイムステップを表わす)。 J_n は V を用いて以下のように表わされる。

$$(J_{ij}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \dots & \frac{\partial f_1}{\partial x_{13}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \dots & \frac{\partial f_2}{\partial x_{13}} \\ \vdots & \vdots & \vdots & & \\ \frac{\partial f_{13}}{\partial x_1} & \frac{\partial f_{13}}{\partial x_2} & \frac{\partial f_{13}}{\partial x_3} & \dots & \frac{\partial f_{13}}{\partial x_{13}} \end{pmatrix} = \left(\frac{\partial f_i}{\partial x_j} \right) = \left(\sum_k C_{ik} \frac{\partial v_k}{\partial x_j} \right)$$

3.2 連立方程式

3.1節 a), b) は逆行列を計算することによって X^{n+1} が求められるが, 演算コストを考慮して次のような連立方程式を解くことで代用する。

$$A \cdot Y = B$$

ここに $A = I - \Delta t J_n$ または $I - \frac{1}{2} \Delta t J_n$, $Y = (X^{n+1} - X^n) / \Delta t$, $B = F(X^n)$ であり,

Crout 法により簡単に計算できる。これより,

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

3.3 Jacobian の解析解

Jacobian $(J_{ij})^n$ は $(\partial v_k / \partial x_j)^n$ から直ちに計算できる。

i) $k = 1$

$$\partial v_1 / \partial x_3 = k_1 x_{11} (x_5 + 0.05)^{-2}$$

$$\partial v_1 / \partial x_5 = -2 k_1 x_3 x_{11} (x_5 + 0.05)^{-3}$$

$$\partial v_1 / \partial x_{11} = k_1 x_3 (x_5 + 0.05)^{-2}$$

他の $\partial v_1 / \partial x_i = 0$

ii) $k = 2$

$$\partial v_2 / \partial x_j = (\partial v_2 / \partial x_j)_{C_N} + (\partial v_2 / \partial C_N)_{x_j} \cdot (\partial C_N / \partial x_j) \text{ だから}$$

$$(\partial v_2 / \partial C_N)_{x_j} = -2 k_2 x_3^2 x_{12}^2 x_4^{-2} x_5^{-4} (C_N + 0.35)^{-3}$$

$$\partial v_2 / \partial x_1 = 2 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_2 = (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_3 = 2 k_2 x_3 x_{12}^2 x_4^{-2} x_5^{-4} (C_N + 0.35)^{-2} + 4 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_4 = -2 k_2 x_3^2 x_{12}^2 x_4^{-3} x_5^{-4} (C_N + 0.35)^{-2} + 3 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_5 = -4 k_2 x_3^2 x_{12}^2 x_4^{-2} x_5^{-5} (C_N + 0.35)^{-2} + (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_6 = 0$$

$$\partial v_2 / \partial x_7 = 2 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_8 = (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_9 = 4 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_{10} = 2 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_{11} = 4 (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_{12} = 2 k_2 x_3^2 x_{12} x_4^{-2} x_5^{-4} (C_N + 0.35)^{-2} + (\partial v_2 / \partial C_N)_{x_j}$$

$$\partial v_2 / \partial x_{13} = (\partial v_2 / \partial C_N)_{x_j}$$

iii) $k = 3$

$$\partial v_3 / \partial x_3 = k_3 x_{13} (x_5 + 0.35)^{-1}$$

$$\partial v_3 / \partial x_5 = -k_3 x_3 x_{13} (x_5 + 0.35)^{-2}$$

$$\partial v_3 / \partial x_{13} = k_3 x_3 (x_5 + 0.35)^{-1}$$

$$\text{他の } \partial v_3 / \partial x_j = 0$$

iv) $k = 4$

$$(\partial v_4 / \partial C_N)_{x_j} = 0.4 k_4 x_4 x_6^{0.5} x_5^{0.5} C_N^{-0.6}$$

$$\partial v_4 / \partial x_1 = 2(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_2 = (\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_3 = 4(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_4 = k_4 x_6^{0.5} x_5^{0.5} C_N^{0.4} + 3(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_5 = 0.5 k_4 x_4 x_6^{0.5} x_5^{-0.5} C_N^{0.4} + (\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_6 = 0.5 k_4 x_4 x_6^{-0.5} x_5^{0.5} C_N^{0.4}$$

$$\partial v_4 / \partial x_7 = 2(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_8 = (\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_9 = 4(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_{10} = 2(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_{11} = 4(\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_{12} = (\partial v_4 / \partial C_N)_{x_j}$$

$$\partial v_4 / \partial x_{13} = (\partial v_4 / \partial C_N)_{x_j}$$

v) $k = 5$

$$\partial v_5 / \partial x_5 = k_5 x_6 x_{13}$$

$$\partial v_5 / \partial x_6 = k_5 x_{13} x_5$$

$$\partial v_5 / \partial x_{13} = k_5 x_6 x_5$$

$$\text{他の } \partial v_5 / \partial x_j = 0$$

vi) $k = 6$

$$\begin{aligned}
(\partial v_6 / \partial C_N)_{x_j} &= 12.5 k_6 x_9 x_7 \\
\partial v_6 / \partial x_1 &= 2 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_2 &= (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_3 &= 4 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_4 &= 3 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_5 &= (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_6 &= 0 \\
\partial v_6 / \partial x_7 &= k_6 x_9 (2.16 + 12.5 C_N) + 2 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_8 &= (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_9 &= k_6 x_7 (2.16 + 12.5 C_N) + 4 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_{10} &= 2 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_{11} &= 4 (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_{12} &= (\partial v_6 / \partial C_N)_{x_j} \\
\partial v_6 / \partial x_{13} &= (\partial v_6 / \partial C_N)_{x_j}
\end{aligned}$$

vii) $k = 7$

$$\begin{aligned}
\partial v_7 / \partial x_3 &= k_7 x_9 x_5^{-4} \\
\partial v_7 / \partial x_5 &= -4 k_7 x_9 x_3 x_5^{-5} \\
\partial v_7 / \partial x_9 &= k_7 x_3 x_5^{-4} \\
\text{他の } \partial v_7 / \partial x_j &= 0
\end{aligned}$$

viii) $k = 8$

$$\begin{aligned}
\partial v_8 / \partial x_5 &= -1.3 k_8 x_7 x_{13} x_5^{-2.3} \\
\partial v_8 / \partial x_7 &= k_8 x_{13} x_5^{-1.3} \\
\partial v_8 / \partial x_{13} &= k_8 x_7 x_5^{-1.3} \\
\text{他の } \partial v_8 / \partial x_j &= 0
\end{aligned}$$

ix) $k = 9$

$$\begin{aligned}
\partial v_9 / \partial x_7 &= k_9 x_{11} \\
\partial v_9 / \partial x_{11} &= k_9 x_7 \\
\text{他の } \partial v_9 / \partial x_j &= 0
\end{aligned}$$

x) $k = 10$

$$\partial v_{10} / \partial x_5 = k_{10} x_8 x_{11} (-3.2 x_5^{-3} + 1.42)$$

$$\partial v_{10} / \partial x_8 = k_{10} x_{11} (1.6 x_5^{-2} + 1.42 x_5)$$

$$\partial v_{10} / \partial x_{11} = k_{10} x_8 (1.6 x_5^{-2} + 1.42 x_5)$$

$$\text{他の } \partial v_{10} / \partial x_j = 0$$

xi) $k = 11$

$$\partial v_{11} / \partial x_5 = -k_{11} x_7 x_{12} x_5^{-2}$$

$$\partial v_{11} / \partial x_7 = k_{11} x_{12} x_5^{-1}$$

$$\partial v_{11} / \partial x_{12} = k_{11} x_7 x_5^{-1}$$

$$\text{他の } \partial v_{11} / \partial x_j = 0$$

xii) $k = 12$

$$\partial v_{12} / \partial x_2 = k_{12} x_3 x_5 (x_5 + 0.054)^{-1}$$

$$\partial v_{12} / \partial x_3 = k_{12} x_2 x_5 (x_5 + 0.054)^{-1}$$

$$\partial v_{12} / \partial x_5 = k_{12} x_2 x_3 (x_5 + 0.054)^{-2}$$

$$\text{他の } \partial v_{12} / \partial x_j = 0$$

xiii) $k = 13$

$$\partial v_{13} / \partial x_2 = k_{13} x_4 x_5$$

$$\partial v_{13} / \partial x_4 = k_{13} x_2 x_5$$

$$\partial v_{13} / \partial x_5 = k_{13} x_2 x_4$$

$$\text{他の } \partial v_{13} / \partial x_j = 0$$

xiv) $k = 14$

$$\partial v_{14} / \partial x_1 = k_{14} x_6 x_5^{-1}$$

$$\partial v_{14} / \partial x_5 = -k_{14} x_1 x_6 x_5^{-2}$$

$$\partial v_{14} / \partial x_6 = k_{14} x_1 x_5^{-1}$$

$$\text{他の } \partial v_{14} / \partial x_j = 0$$

XV) $k = 15$

$$\partial v_{15} / \partial x_1 = k_{15} x_4$$

$$\partial v_{15} / \partial x_4 = k_{15} x_1$$

$$\text{他の } \partial v_{15} / \partial x_j = 0$$

XVI) $k = 16$

$$(\partial v_{16} / \partial C_N)_{x_j} = 0.4 k_{16} x_2 x_6^{0.5} C_N^{-0.6} x_5^{0.6}$$

$$\partial v_{16} / \partial x_1 = 2 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_2 = k_{16} x_6^{0.5} C_N^{0.4} x_5^{0.6} + (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_3 = 4 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_4 = 3 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_5 = 0.6 k_{16} x_2 x_6^{0.5} C_N^{0.4} x_5^{-0.4} + (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_6 = 0.5 k_{16} x_2 x_6^{-0.5} C_N^{0.4} x_5^{0.6}$$

$$\partial v_{16} / \partial x_7 = 2 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_8 = (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_9 = 4 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_{10} = 2 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_{11} = 4 (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_{12} = (\partial v_{16} / \partial C_N)_{x_j}$$

$$\partial v_{16} / \partial x_{13} = (\partial v_{16} / \partial C_N)_{x_j}$$

XVII) $k = 17$

$$\partial v_{17} / \partial x_5 = k_{17}$$

$$\text{他の } \partial v_{17} / \partial x_j = 0$$

XVIII) $k = 18$

$$\partial v_{18} / \partial x_6 = k_{18}$$

$$\text{他の } \partial v_{18} / \partial x_j = 0$$

xix) $k = 19$

$$\partial v_{19} / \partial x_5 = 2 k_{19} x_8^2 x_5$$

$$\partial v_{19} / \partial x_8 = 2 k_{19} x_8 x_5^2$$

$$\text{他の } \partial v_{19} / \partial x_j = 0$$

xx) $k = 20$

$$(\partial v_{20} / \partial C_N)_{x_j} = 2 k_{20} x_5^{1.3} C_N x_8 (1 + x_8/x_6)^{-1}$$

$$\partial v_{20} / \partial x_1 = 2 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_2 = (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_3 = 4 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_4 = 3 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_5 = (\partial v_{20} / \partial C_N)_{x_j} + 1.3 k_{20} x_5^{0.3} C_N^2 x_8 (1 + x_8/x_6)^{-1}$$

$$\partial v_{20} / \partial x_6 = k_{20} x_5^{1.3} C_N^2 x_8^2 (x_6 + x_8)^{-2}$$

$$\partial v_{20} / \partial x_7 = 2 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_8 = (\partial v_{20} / \partial C_N)_{x_j} + k_{20} x_5^{1.3} C_N^2 x_6^2 (x_6 + x_8)^{-2}$$

$$\partial v_{20} / \partial x_9 = 4 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_{10} = 2 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_{11} = 4 (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_{12} = (\partial v_{20} / \partial C_N)_{x_j}$$

$$\partial v_{20} / \partial x_{13} = (\partial v_{20} / \partial C_N)_{x_j}$$

xxi) $k = 21$

$$\partial v_{21} / \partial x_6 = -0.5 k_{21} x_6^{-1.5} x_8 x_9$$

$$\partial v_{21} / \partial x_8 = k_{21} x_6^{-0.5} x_9$$

$$\partial v_{21} / \partial x_9 = k_{21} x_6^{-0.5} x_8$$

$$\text{他の } \partial v_{21} / \partial x_j = 0$$

xxii) $k = 22$

$$\partial v_{22} / \partial x_4 = k_{22} x_7 (2130 + 186/x_5)$$

$$\partial v_{22} / \partial x_5 = k_{22} x_4 x_7 (-186/x_5^2)$$

$$\partial v_{22} / \partial x_7 = k_{22} x_4 (2130 + 186/x_5)$$

$$\text{他の } \partial v_{22} / \partial x_j = 0$$

XXIII) $k = 23$

$$\partial v_{23} / \partial x_4 = k_{23} x_5^{-1.3} x_8$$

$$\partial v_{23} / \partial x_5 = -1.3 k_{23} x_4 x_5^{-2.3} x_8$$

$$\partial v_{23} / \partial x_8 = k_{23} x_4 x_5^{-1.3}$$

$$\text{その他の } \partial v_{23} / \partial x_j = 0$$

XXIV) $k = 24$

$$\partial v_{24} / \partial x_1 = k_{24} x_5^{-1} x_{13}$$

$$\partial v_{24} / \partial x_5 = -k_{24} x_1 x_5^{-2} x_{13}$$

$$\partial v_{24} / \partial x_{13} = k_{24} x_1 x_5^{-1}$$

$$\text{その他の } \partial v_{24} / \partial x_j = 0$$

XXV) $k_{25} = 25$

$$(\partial v_{25} / \partial C_N)_{x_j} = -0.54 k_{25} x_3^2 x_5^{-2.2} C_N^{-1.54}$$

$$\partial v_{25} / \partial x_1 = 2 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_2 = (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_3 = 4 (\partial v_{25} / \partial C_N)_{x_j} + 2 k_{25} x_3 x_5^{-2.2} C_N^{-0.54}$$

$$\partial v_{25} / \partial x_4 = 3 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_5 = (\partial v_{25} / \partial C_N)_{x_j} - 2.2 k_{25} x_3^2 x_5^{-3.2} C_N^{-0.54}$$

$$\partial v_{25} / \partial x_6 = 0$$

$$\partial v_{25} / \partial x_7 = 2 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_8 = (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_9 = 4 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_{10} = 2 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_{11} = 4 (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_{12} = (\partial v_{25} / \partial C_N)_{x_j}$$

$$\partial v_{25} / \partial x_{13} = (\partial v_{25} / \partial C_N)_{x_j}$$

XXVI) $K = 26$

$$\partial v_{26} / \partial x_2 = 2 k_{26} x_2 x_5$$

$$\partial v_{26} / \partial x_5 = k_{26} x_2^2$$

$$\text{その他の } \partial v_{26} / \partial x_j = 0$$

XXVII) $k = 27$

$$\partial v_{27} / \partial x_5 = k_{27} x_6 x_{12}$$

$$\partial v_{27} / \partial x_6 = k_{27} x_5 x_{12}$$

$$\partial v_{27} / \partial x_{12} = k_{27} x_5 x_6$$

$$\text{その他の } \partial v_{27} / \partial x_j = 0$$

3.4 時間キザミ幅

時間キザミ幅は通常入力で与える（初期条件によって $10^{-3} \sim 10^{-7} \text{min}$ ）。特に 0 を指定した場合はコード内で自動的にキザミ幅を設定する。これは 1 ステップ毎に J_n , $F(X^n)$ の大きさによってキザミ幅 DT を決定するもので、次式によって与える。

$$F_{\max} = \max |F(X^n)|$$

$$J_{\max} = \max |J_n|$$

$$DT = 7 \min \left(\frac{F_f}{400 \cdot F_{\max}}, \frac{F_j}{40 \cdot \sqrt{J_{\max}}} \right)$$

ただし、 F_f , F_j は入力で与える定数（通常 1 とする）。

この DT を用いて計算した結果、濃度値が負になった場合は DT を半分にして再計算する。

3.5 その他の解法

本コードでは Porsing 系解法の他に Runge-Kutta-Gill, Haming, Runge-Kutta-Varner, Adams の各解法も使用できる。これらはある時刻 T^n のまわりで幾つかの $F(X)$ を求めて X^{n+1} を決定する方法及び T^{n-1} , T^{n-2} , …… T^{n-m} で既に計算された $F(X^{n-1})$, $F(X^{n-2})$, …… $F(X^{n-m})$ から $F(X^{n+1})$ を予測して X^{n+1} を決定する方法に分類される。なお、これらの解法では Jacobian J_n は必要としない。

本プログラムではさらに Euler-Romberg 法, Jacobi-Romberg 法（仮称）によっても計算を行うことができる。両法の概要を以下に示す。

(1) Euler-Romberg 法

これは常微分方程式の最も単純な解法である Euler 法と, Romberg 法による補間を組み合わせたもので、計算機向きの反復解法である。

〔手順〕

- 1) 区間を次々に二分しながら、くり返し Euler 法を適用する。
- 2) 反復線形補間を行い、 $DT = 0$ に外挿する。

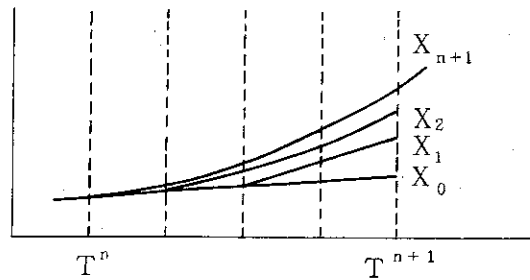


図 3.1 Euler 法の反復適用

Romberg の補間漸化式は、

$$X_k = \frac{2^m X_{k+1} - X_k}{2^m - 1} \quad (m \text{ は二分の回数})$$

$|X_k - X_{k-1}| \leq \epsilon$ となったら収束したとみなす。

実際に計算を行ってみると収束が悪いため反復回数が大きくなり、CPU時間をかなり使用する。そこで収束性のよい Jacobi-Romberg 法を適用してみた。

(2) Jacobi-Romberg 法

これは上記の Euler 法の部分に Porsing 法を組み入れたもので、収束性はかなりよい。反応速度定数が極端に大きいケースでは Porsing 法よりも短時間で計算できることが確認された。

3.6 解法の実行例について

3.5 節で述べた各種解法について、いくつかの試計算により評価を行った結果、以下の概括的結論を得た。

- i) 変化が比較的ゆるやかな反応系では、Adams 法が最も高速、高精度で計算できる。速い反応成分が含まれてくると、濃度が負値になったり、発散する場合がある。
- ii) Porsing 法や M-Porsing 法は、反応系の条件にあまり左右されず安定した解を出す。
Jacobian の大きさ、反応速度によりきざみ幅 DT を決定する場合、計算速度、精度の上で特に M-Porsing 法が有利である。
- iii) 反復解法 (Euler-Romberg, Jacobi-Romberg) は、反応速度が極端に大きい時、他解法では解けなかったり、計算時間が膨大となる場合に有利である。
- iv) R-K-G 法, Haming 法, R-K-V 法は系の条件に左右され易い。

以上の結論から次節 (4 節) の実行例は全て M-Porsing 法により計算した結果である。

4. REACT による計算実行例

4.1 Pu(IV)のウラナスU(IV)またはHAN(硝酸ヒドロキシルアミン)による還元反応

再処理ピューレックス工程では、ウランとプルトニウム分離方法として、Pu(IV)のPu(III)への還元法を採用している。通常、還元剤としてウラナス〔U(IV)〕、またはHAN〔NH₂OHNO₃〕が用いられる。

図 4.1, 4.2 は、Pu(IV) : 濃度 0.1M を U(IV) : 濃度 0.05M, または HAN : 濃度 0.1M で還元した時の Pu(IV), Pu(III) の濃度変化計算結果を示す。これらの計算結果は、

- i) U(IV) による Pu(IV) の還元速度 (破線) が HAN のそれ (実線) に比べて極めて大きく、0.01 分以内にはほぼ全ての Pu(IV) が Pu(III) に還元される (HAN では 10 分以上要する)。
- ii) どちらの反応も硝酸濃度が低いほど速い。
- iii) ヒドラジンが添加されていないと、ある時間経過後、Pu(III) の再酸化反応が起き、Pu(IV) が増大し始める (次の 4.2 節を参照) 等の特徴を示している。

4.2 Pu(III)の再酸化反応

Pu(III) は比較的酸化され易く、特に亜硝酸が存在するとその触媒的作用により、Pu(III) の Pu(IV) への酸化は加速されるので、通常はヒドラジンを添加して亜硝酸を常に分解することにより、再酸化反応を抑制している。図 4.3 は、Pu(III) の再酸化反応が Pu の濃度が大きいほど速く進むことを示し、図 4.4 は、これら再酸化反応が硝酸濃度に大きく依存することを示している。また図 4.5 では、硝酸濃度が 0.3~1.0M と低くてもヒドラジンが無いと約 20 分でほぼ全ての Pu(III) が Pu(IV) に酸化され、亜硝酸濃度も約 0.1M 以上に迄増大する事を示している。

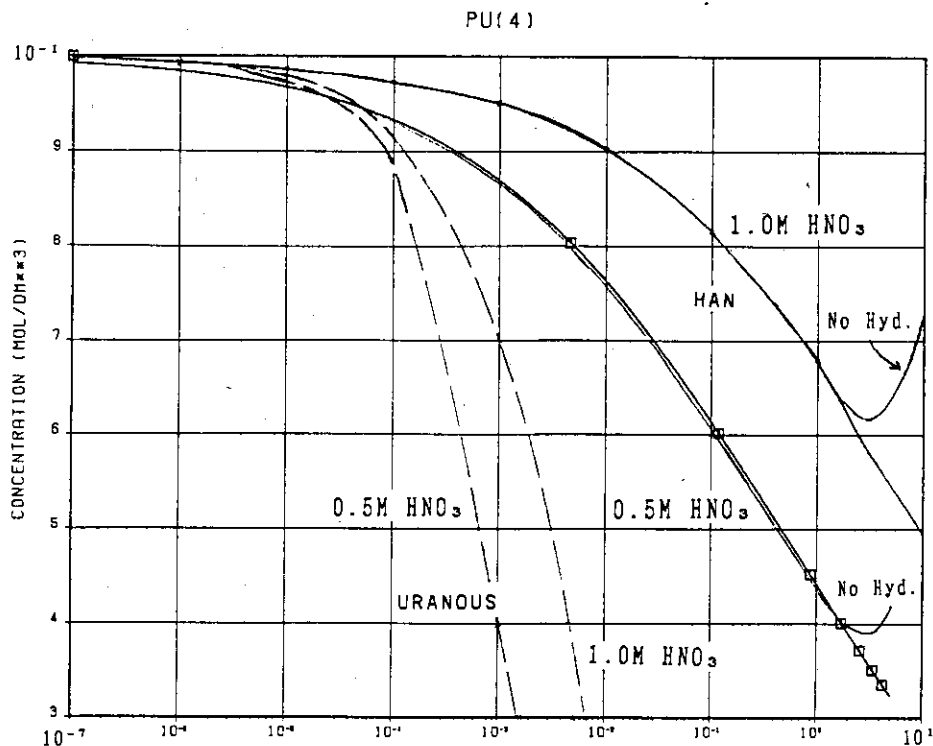


図 4.1 Pu (IV) のHANまたはU(IV)による還元：Pu (IV)の経時変化
Pu (IV); 0.1M, HNO₃; 1.0M or 0.5M, Reductant; 0.1M HAN
or 0.05M U(IV) Hydrazine; 0.1M or 0M

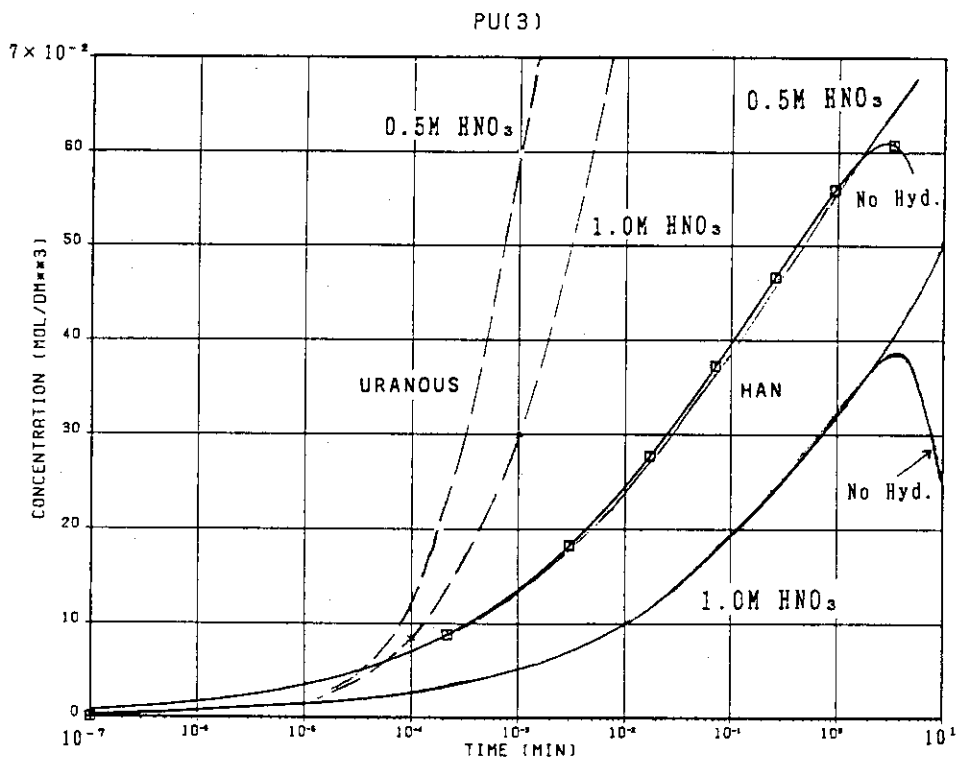


図 4.2 Pu (IV) のHANまたはU(IV)による還元：Pu (III)の生成条件は
図 4.1 に同じ

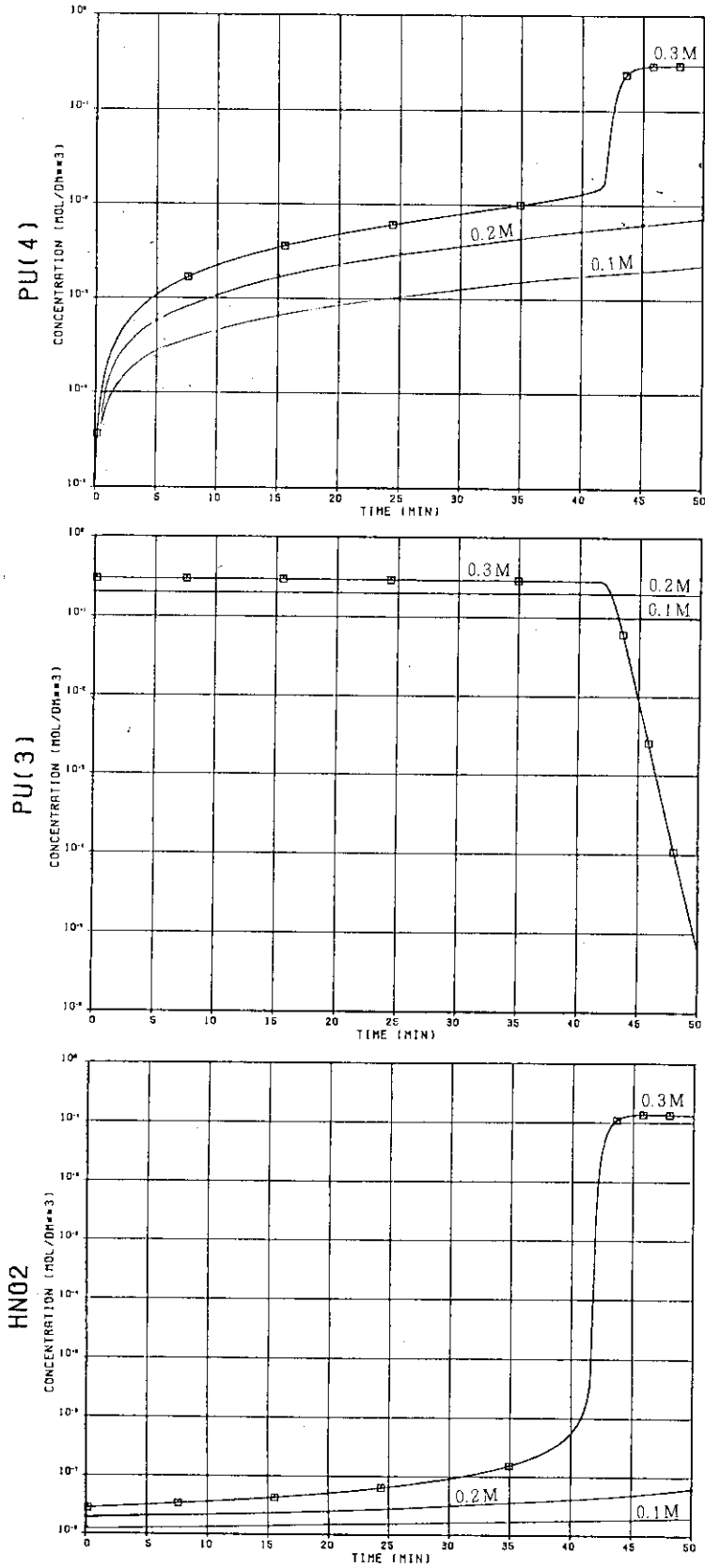


図 4.3 Pu (Ⅲ)の再酸化反応における初期Pu (Ⅲ)濃度の影響 ; Pu (Ⅲ),
 Pu (Ⅳ), HNO₂ の経時変化
 Pu (Ⅲ) : 0.1, 0.2, 0.3M, HNO₃ : 3.0M,
 Hydrazine : 1 × 10⁻²M

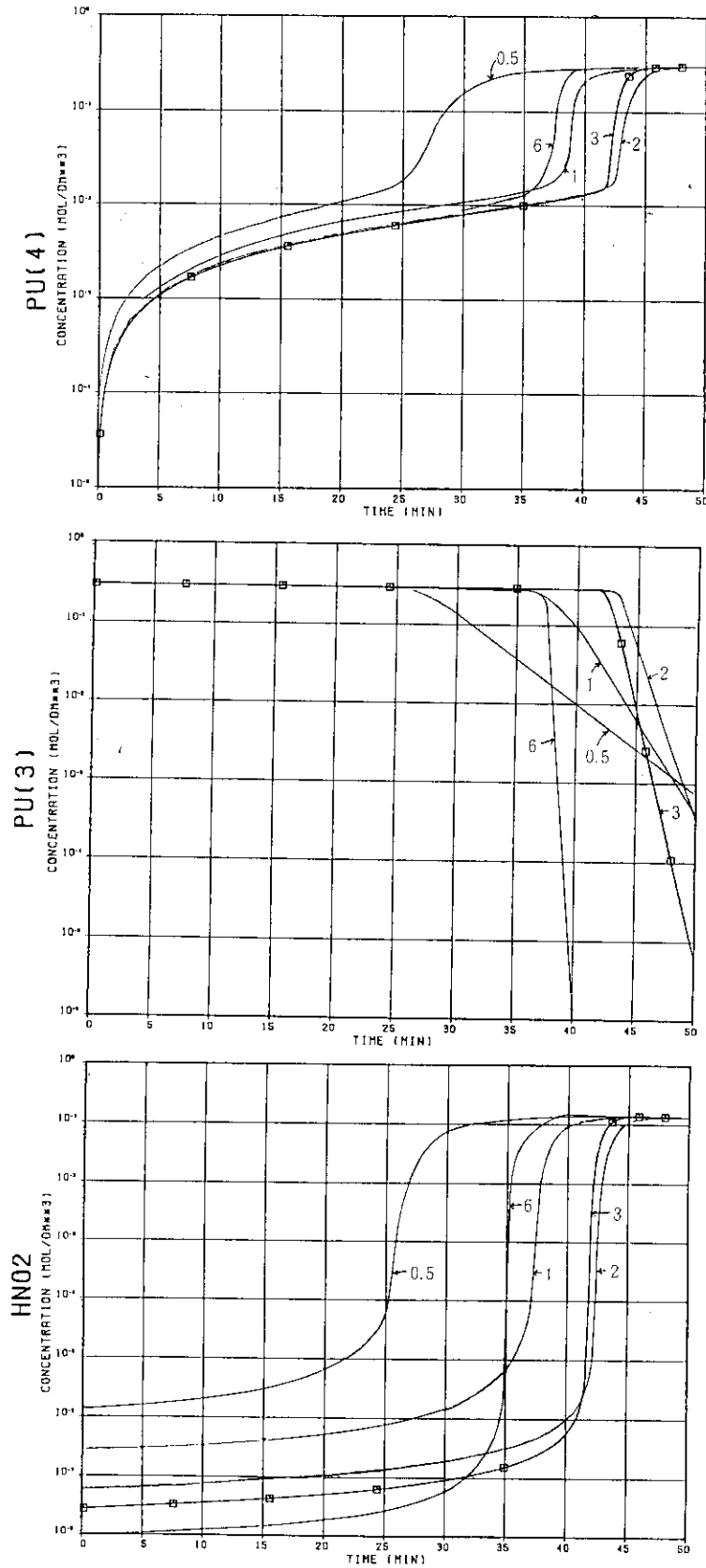


図 4.4 Pu (III) の再酸化反応における初期硝酸濃度の影響 ; Pu (III), Pu (IV), HNO_2 の経時変化
 Pu (III) : 0.3M, HNO_3 : 0.5, 1, 2, 3, 6M,
 Hydrazine : 1×10^{-2} M

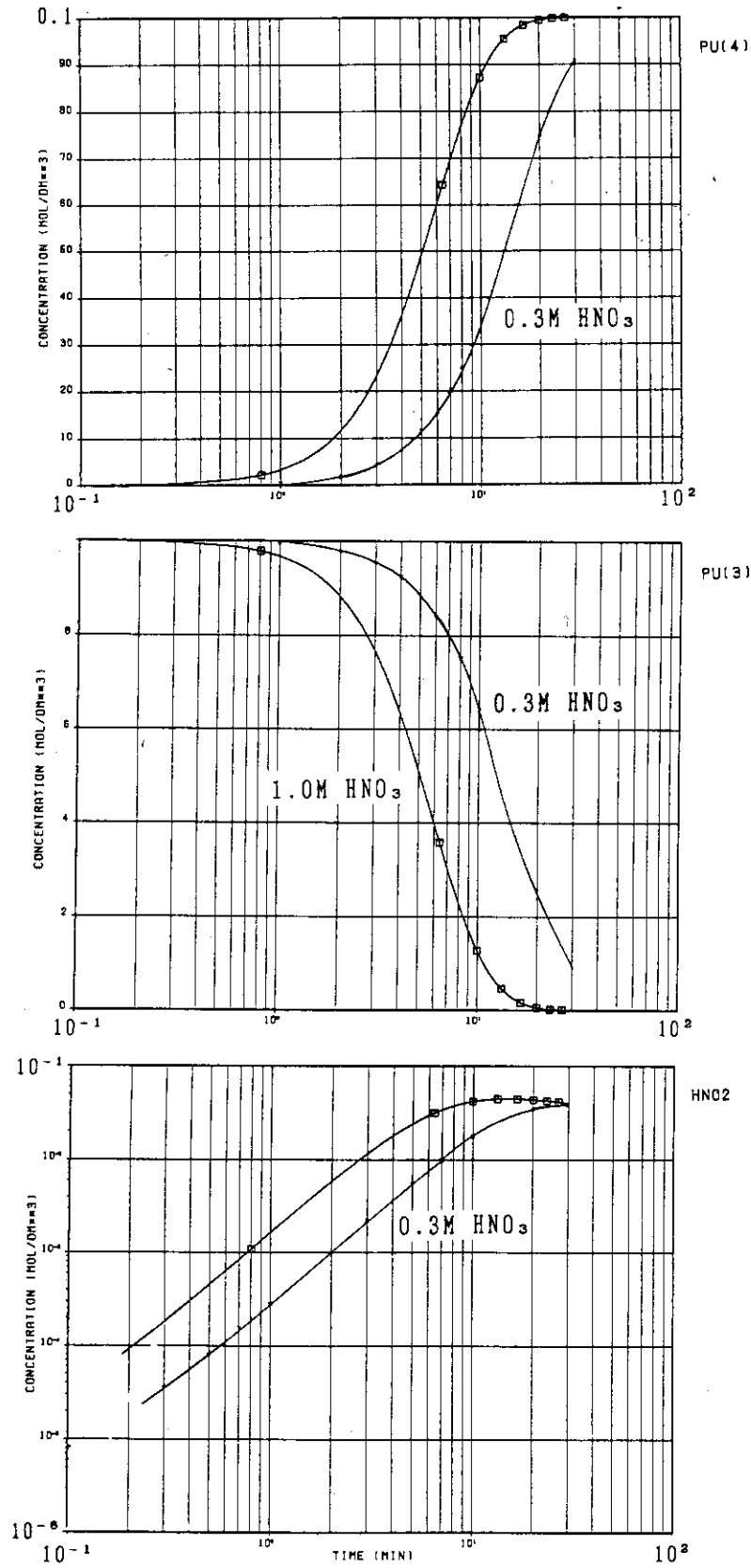


図 4.5 ヒドラジンが添加されていない系における Pu (III) の再酸化反応：硝酸濃度の影響；Pu (III), Pu (IV), HNO₂ の経時変化
 Pu (III)：0.1M, HNO₃ 0.3Mまたは 1.0M, Hydrazine：0M

5. 結 び

ここで報告した数値モデルは、計算実行例からも解るように、化学反応の速度論的シミュレーションの分野での大きな可能性を示している。REACTコードには、原理的には、現在組入れてある27個の化学反応速度式の他に、更に多くの反応速度式を追加することが出来る。今回の作業において、U, Pu, Np が関与する酸化還元反応を調査したところ、硝酸溶液系で反応速度式が求まっていないものが幾つもあった。平衡系を考えると、正反応と逆反応それぞれの速度定数を知る必要もある。核燃料サイクルにおいては、アクチニド元素が関与する多くの化学反応を利用しており、今後とも、まだ得られていない上記のような基本的なデータを取得する必要がある。

謝 辞

本コードの開発においては、日本原子力研究所計算センター、石黒美佐子氏に数値解法について多大の支援を得た。ここに深く感謝いたします。

引 用 文 献

- 1) Gourisse D., Madic C., "Redox Reactions of Transuranium Elements in the Aqueous Solution" Gmelin Handbuch der Anorganischen Chemie, Band 20, Transurane Teil D1, Chapter 3, pp27-105, Springer-Verlag, 1975.
- 2) Vladimirova M.V., "Advances in the radiation chemistry of plutonium and neptunium" Radiokhimiya, 27(3), 311(1985).
- 3) Miner F.J., Seed J.R., "Radiation chemistry of plutonium nitrate solutions" Chem. Rev., 67, 299(1967).
- 4) 泉田龍男, "硝酸の放射線化学", 放射線化学, 45, 44(1988)
- 5) Vladimirova M.V., Golub E.A., "Mathematical model of radiation-chemical processes in nitric acid solutions of plutonium, I. Selection of the model and reaction rate constant" Radiokhimiya, 30(4), 547(1988).
- 6) Vladimirova M.V., Golub E.A., "Mathematical modelling of radiation chemical processes in nitric acid solutions of plutonium" Proc. 6th Tihany Symposium on Rad. Chemistry, Akademiai Kiado, Budapest. Vol.1. p.167(1987).
- 7) Biddle P., Miles J.H., Waterman M.J., J.inorg.nucl.Chem., 28, 1736(1966).
- 8) Barney G.S., J.inorg.nucl.Chem., 38, 1677(1976).

5. 結 び

ここで報告した数値モデルは、計算実行例からも解るように、化学反応の速度論的シミュレーションの分野での大きな可能性を示している。REACTコードには、原理的には、現在組入れてある27個の化学反応速度式の他に、更に多くの反応速度式を追加することが出来る。今回の作業において、U, Pu, Np が関与する酸化還元反応を調査したところ、硝酸溶液系で反応速度式が求まっていないものが幾つもあった。平衡系を考えると、正反応と逆反応それぞれの速度定数を知る必要もある。核燃料サイクルにおいては、アクチニド元素が関与する多くの化学反応を利用しており、今後とも、まだ得られていない上記のような基本的なデータを取得する必要がある。

謝 辞

本コードの開発においては、日本原子力研究所計算センター、石黒美佐子氏に数値解法について多大の支援を得た。ここに深く感謝いたします。

引 用 文 献

- 1) Gourisse D., Madic C., "Redox Reactions of Transuranium Elements in the Aqueous Solution" Gmelin Handbuch der Anorganischen Chemie, Band 20, Transurane Teil D1, Chapter 3, pp27-105, Springer-Verlag, 1975.
- 2) Vladimirova M.V., "Advances in the radiation chemistry of plutonium and neptunium" Radiokhimiya, 27(3), 311(1985).
- 3) Miner F.J., Seed J.R., "Radiation chemistry of plutonium nitrate solutions" Chem. Rev., 67, 299(1967).
- 4) 泉田龍男, "硝酸の放射線化学", 放射線化学, 45, 44(1988)
- 5) Vladimirova M.V., Golub E.A., "Mathematical model of radiation-chemical processes in nitric acid solutions of plutonium, I. Selection of the model and reaction rate constant" Radiokhimiya, 30(4), 547(1988).
- 6) Vladimirova M.V., Golub E.A., "Mathematical modelling of radiation chemical processes in nitric acid solutions of plutonium" Proc. 6th Tihany Symposium on Rad. Chemistry, Akademiai Kiado, Budapest. Vol.1. p.167(1987).
- 7) Biddle P., Miles J.H., Waterman M.J., J.inorg.nucl.Chem., 28, 1736(1966).
- 8) Barney G.S., J.inorg.nucl.Chem., 38, 1677(1976).

5. 結 び

ここで報告した数値モデルは、計算実行例からも解るように、化学反応の速度論的シミュレーションの分野での大きな可能性を示している。REACTコードには、原理的には、現在組入れてある27個の化学反応速度式の他に、更に多くの反応速度式を追加することが出来る。今回の作業において、U, Pu, Np が関与する酸化還元反応を調査したところ、硝酸溶液系で反応速度式が求まっていないものが幾つもあった。平衡系を考えると、正反応と逆反応それぞれの速度定数を知る必要もある。核燃料サイクルにおいては、アクチノイド元素が関与する多くの化学反応を利用しており、今後とも、まだ得られていない上記のような基本的なデータを取得する必要がある。

謝 辞

本コードの開発においては、日本原子力研究所計算センター、石黒美佐子氏に数値解法について多大の支援を得た。ここに深く感謝いたします。

引 用 文 献

- 1) Gourisse D., Madic C., "Redox Reactions of Transuranium Elements in the Aqueous Solution" Gmelin Handbuch der Anorganischen Chemie, Band 20, Transurane Teil D1, Chapter 3, pp27-105, Springer-Verlag, 1975.
- 2) Vladimirova M.V., "Advances in the radiation chemistry of plutonium and neptunium" Radiokhimiya, 27(3), 311(1985).
- 3) Miner F.J., Seed J.R., "Radiation chemistry of plutonium nitrate solutions" Chem. Rev., 67, 299(1967).
- 4) 泉田龍男, "硝酸の放射線化学", 放射線化学, 45, 44(1988)
- 5) Vladimirova M.V., Golub E.A., "Mathematical model of radiation-chemical processes in nitric acid solutions of plutonium, I. Selection of the model and reaction rate constant" Radiokhimiya, 30(4), 547(1988).
- 6) Vladimirova M.V., Golub E.A., "Mathematical modelling of radiation chemical processes in nitric acid solutions of plutonium" Proc. 6th Tihany Symposium on Rad.Chemistry, Akademiai Kiado, Budapest. Vol.1. p.167(1987).
- 7) Biddle P., Miles J.H., Waterman M.J., J.inorg.nucl.Chem., 28, 1736(1966).
- 8) Barney G.S., J.inorg.nucl.Chem., 38, 1677(1976).

- 9) Koltunov.V.S., Zhuravleva G.I., Radiokhimiya, 16, 84(1974).
- 10) Koltunov.V.S., et al, Radiokhimiya, 15, 777(1973).
- 11) Biddle P., Miles J.H., J.inorg.nucl.Chem., 30, 1291(1968).
- 12) Koltunov V.S., Marchenko V.I., Kinet.Katal., 7(2), 224(1966).
- 13) Rykov.A.G., et al, Radiokhimiya, 14, 128(1972), 8, 27(1966).
- 14) Koltunov V.S., et al, Radiokhimiya, 18, 65(1976).
- 15) Koltunov V.S., Tikhonov M.F., Radiokhimiya, 15, 194(1973).
- 16) Salomon L., et al, ETR-233(1968).
- 17) Shilin I.V., Rumyantseva T.A., Radiokhimiya, 20, 536(1978).
- 18) Koltunov V.S., Tikhonov.M.F., Radiokhimiya, 19, 611(1977).
- 19) Ryabova A.A., et al, Radiokhimiya, 22, 254(1980).
- 20) Egorov G.F., et al, Radiokhimiya, 23, 369(1981).
- 21) Koltunov V.S., Zhuravleva G.I., Radiokhimiya, 10, 662(1968).
- 22) Artyukhin P.I., et al. Zh.Neorgan.Khim., 4, 1324(1959).
- 23) Koltunov V.S., Ryabova A.A., Radiokhimiya, 22, 635(1980).
- 24) Escure H., CEA-R-4574(1974).
- 25) Moulin J.P., CEA-R-4912(1978).
- 26) Fulton R.B., Newton T.W., J.Phys. Chem., 74, 1661(1970).
- 27) Koltunov V.S., et al, Radiokhimiya, 23, 559(1981), 24, 607(1982).
- 28) Koltunov V.S., Zhuravleva G.I., Radiokhimiya, 15, 74(1973).
- 29) Egorov G.F., et al, Radiokhimiya, 23, 369(1981).
- 30) Barney G.S., ARH-SA-97, (1971).

付 録

A.1 入力データ及びJCL説明

1.1 入力データの説明

(1) 計算コード(Solver)の入力

カード①

変 数 名	CTIT
フォーマット	A72

変 数 名	型	意 味
CTIT	CH*72	計算タイトル。グラフの主タイトルとしても用いられる。

カード②

変 数 名	NDTS	IOPT	ICLK	IFINT	NEPS
フォーマット	5 I 1 0				

変 数 名	型	意 味
NDTS	I * 4	時間データ(③)の数。最大5。
IOPT	I * 4	解法の指定。 0 = Porsing 法 1 = M-Porsing 法 2 = R-K-G 法 3 = Haming 法 4 = R-K-V 法 5 = Adams 法 6 = Euler-Romberg 法 7 = Jacobi-Romberg 法

変数名	型	意	味
ICLK	I*4	チェックリストフラグ 0 = チェックリストなし。 1 = 化学反応式のリスト 2 = Pu, Np, U の物質収支リスト	
IFINT	I*4	プロット用ファイルへの出力インターバル。 0 の場合は全ステップ出力。	
NEPS	I*4	ある化学種濃度が負になった場合の最大メッセージ出力回数。0 の場合は毎回出力する。 (~ 10)	

カード③

本カードはカード②で指定した NDTs 枚必要である。また、解法 (IOPT) によって指定方法が異なる。

変数名	TEND	TLST	DTT	TPRM1	TPRM2
フォーマット	5F10.0				

変数名	型	意	味
TEND	R*8	計算終了時刻 (min)。	
TLST	R*8	リスト出力間隔 (min) 。0 の場合は出力しない。	
DTT	R*8	以下の説明を参照。	
TPRM1	R*8		
TPRM2	R*8		

1) Porsing, M-Porsing の場合 (IOPT=0, 1)

DTT : 時間キザミ幅 (min) 。 DTT = 0 とした場合は自動設定される (3.4 節参照) 。

TPRM1 : DTT = 0 とした場合, F_i を指定する (省略値 1) 。

TPRM2 : DTT = 0 とした場合, F_j を指定する (省略値 1) 。

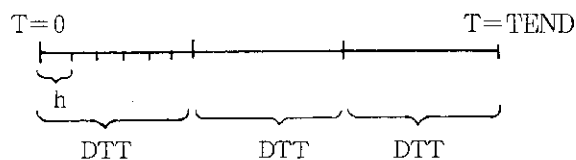
2) R-K-G の場合 (IOPT=2)

DTT : 時間キザミ幅 (min) 。 DTT > 0 であること。

TPRM1, TPRM2 : 無意味。

3) Haming の場合 (IOPT=3)

DTT : 時間キザミ区分 (min) 。



時間キザミは DTT 内で自動的に設定される。条件がよければ $DTT = TEND$ とすることも可能であるが、通常は $DTT < TEND$ とする。

TPRM1 : 初期時間キザミ幅 h_0 (min)。

TPRM2 : 解に対する相対誤差の上限。0 の場合標準値。

4) R-K-V, Adams の場合 (IOPT=4, 5)

DTT : 時間キザミ区分 (min)。Haming の場合と同様。

TPRM1 : 絶対誤差の上限。0 の場合は標準値。

TPRM2 : 相対誤差の上限。0 の場合は標準値。

5) Euler-Romberg, Jacobi-Romberg の場合 (IOPT=6, 7)

DTT : 時間キザミ幅 (min)。

TPRM1 : 相対誤差の上限 (0 の場合, 10^{-6} とする)。

TPRM2 : 最大反復回数 (0 の場合 16 とする)。整数型で指定すること。通常は 5~6 でよい。反復回数を n とすると、計算回数は $2^{n+1}-1$ 回となり、あまり大きな数にすると CPU 時間が膨大になる。

カード④ (2 枚)

本カード以降は省略できる。その場合はリスタートと見なす (リスタート・ファイルの指定が必要。詳細は A.1.2 節参照)。

変数名	VOLX	{ AX(i) i=1~6 }
フォーマット	7F10.0	

変数名	{ AX(i) , i=7~13 }
フォーマット	7F10.0

変数数	型	意味
VOLX	R*8	溶液体積 (dm^3)。⑤カードを指定する場合に意味をもつ。
AX(i)	R*8	化学種 i (1~13) の初濃度 (mol/dm^3) *

* $HNO_3, HNO_2, Pu(III)$ は 0 としない。

カード⑤ (2枚)

本カードは省略できる。時刻0で別溶液を混入する場合に指定する。

変 数 名	VOLY	{AY(i), i=1~6}
フォーマット		7F10.0

変 数 名	{AY(i), i=7~13}
フォーマット	7F10.0

変 数 名	型	意	味
VOLY	R*8	混入する溶液体積 (dm ³)。	
AY(i)	R*8	混入溶液内の化学種 i の濃度 (mol/dm ³)	

計算開始時の濃度は次式のようにになる。

$$AX(i) = \{ VOLX \cdot AX(i) + VOLY \cdot AY(i) \} / (VOLX + VOLY)$$

カード⑥ (ネームリスト形式)

本カードは省略できる。

NAMELIST/CONST/AK(k) k = 1~19

変 数 名	型	意	味
AK(k)	R*8	k 番目の反応式の反応速度定数。 省略値は本文第 2.1 節参照。	

(2) 作図カード (Graphic) の入力

カード①

変 数 名	CTIT
フォーマット	A72

変 数 名	型	意	味
CTIT	CH*72	主タイトル。空白の場合は計算コードのタイトルを使用する。	

カード②

変数名	NFILE	
フォーマット	I5	

変数名	型	意味
NFILE	I*4	作図ファイルの個数。最大2。

比較プロット用に、作図ファイルは2個まで入力できる。

作図ファイルのDD名は、

1 個目……FT11F001

2 個目……FT12F001

で定義する。

カード③

変数名	{NPLT(i), i=1, 2}	
フォーマット	2I5	

変数名	型	意味
NPLT(i)	I*4	i 番目の作図ファイルの最大有効データ点数。 最大1200点。

以下のカード④～⑥はグラフの数だけ繰り返す。

カード④

変数名	CAPT	
フォーマット	A72	

変数名	型	意味
CAPT	CH*72	サブ・タイトル。

主タイトルの下部に描かれる。

カード⑤

変数名	NGCT	NGTP	XINP0	XINP9	YINP0	YINP9
フォーマット	2 I 5		4 F 10.0			

変数名	型	意味
NGCT	I * 4	一図に描く曲線の数。最大 13。
NGTP	I * 4	各軸の指定。10・J+I で与える。 I = 0 x 軸線形, y 軸線形 = 1 " , y 軸対数 = 2 x 軸対数, y 軸線形 = 3 " , y 軸対数 J = 0 z 軸線形 = 1 z 軸対数 ここで, x 軸: 時間 y 軸: 化学種濃度 z 軸: 反応速度 (化学種生成速度) を表わす。
XINP0	R * 8	x 軸の最小, 最大目盛。無指定の場合は自動スケールリングを行う。
XINP9	R * 8	
YINP0	R * 8	y 軸の最小, 最大目盛。無指定の場合は自動スケールリングを行う。
YINP9	R * 8	

現在 z 軸の指定はできない (常に自動スケール)。

カード⑥ (NGCT 枚与える)

変数名	K 1	K 2	-	CPRM
フォーマット	2 I 5		5 X	A 3 2

変数名	型	意味
K 1	I * 4	作図ファイルの番号 (\leq NFILE)。1 又は 2
K 2	I * 4	項目の番号。100・J+I で与える。 I = 1~13 化学種番号 (J=0, 1 の場合) 1~19 反応式番号 (J=2 の場合)

変数名	型	意	味
		J = 0	化学種濃度
		1	化学種生成速度
		2	反応速度
CPRM	CH* 32	項目に付けるキャプション	

入力データ例 (計算コード)

TEST 1: OXIDATION OF NP(V) IN EXTRACTION STAGE

```

      1           5           1           10
1.0D02  1.0D01  1.0D00  1.0D-6  1.0D-6
1.0D00  1.0D-3  0.0D00  1.0D-2  1.0D-6  3.0D00  1.0D-6
1.0D-7  1.0D-3  1.0D-7  0.0D00  0.0D00  0.0D00  0.0D00
    
```

&CONST

AK = 5*0.0D0, 1.0D0, 70D0, 9*0.0D0, 1.5D-5, 1.0D-2,

&END

TEST 2: REOXIDATION OF PU(III)

```

      1           1           2           10
3.0D01  2.0D00  0           1.0D-2  1.0D00
1.0D00  0.0D00  0.0D00  0.0D00  1.0D-1  3.0D-1  1.0D-6
0.0D00  0.0D00  0.0D00  0.0D00  0.0D00  0.0D00  1.0D-2
    
```

&CONST

AK = 2*0.0D0, 3.8D-2, 7.0D-1, 6.0D04, 11*0.0D0, 1.5D-5, 1.0D-2,

8*0.0D0,

&END

TEST 3: URANOUS PU(IV) REACTION, M-PORSING METHOD

```

      1           1           2           100           10
1.0D-1  1.0D-4  0.0D00  1.0D-2  1.0D00
1.0D00  0.0D00  0.0D00  1.0D-1  1.0D-6  1.0D00  1.0D-6
0.0D00  0.0D00  0.0D00  0.0D00  5.0D-2  0.0D00  0.0D00
    
```

&CONST

AK(17) = 1.5D-5, AK(18) = 1.0D-2,

&END

TEST 4: HAN-PU(IV) REACTION, M-PORSING METHOD

1	1	2	10			
1.0D01	1.0D-1	0	1.0D-2	1.0D00		
1.0D00	0.0D00	0.0D00	1.0D-1	1.0D-6	1.0D00	1.0D-6
0.0D00	0.0D00	0.0D00	0.0D00	0.0D00	1.0D-1	1.0D-1

&CONST

AK = 0.0D00, 1.74D00, 3.8D-2, 7.0D-1, 6.0D04, 11*0.0D00, 1.5D-5,
1.0D-2, 5*0.0D0,

&END

TEST 6: NP(VI)-PU(III)-PU(IV)-HAN-HYDRAZ

1	1	2	10			
1.0D00	1.0D-1	0	1.0D-2	1.0D00		
1.0D00	1.0D-7	0.0D00	1.0D-2	1.0D-7	5.0D-1	1.0D-7
1.0D-3	1.0D-7	1.0D-7	1.0D-7	0.0D00	1.0D-1	1.0D-1

&CONST

AK(17) = 1.5D-5, AK(18) = 1.0D-2, AK(1) = 0.0D0,

&END

TEST 7: NP(VI)-NP(IV)-PU(III)-PU(IV)-URANOUS-HYDRAZ

1	1	2	10			
5.0D-1	1.0D-1	0	1.0D-2	1.0D00		
1.0D00	1.0D-7	1.0D-7	1.0D-2	1.0D-7	5.0D-1	1.0D-7
1.0D-3	1.0D-7	1.0D-7	0.0D00	1.0D-2	0.0D00	1.0D-1

&CONST

AK(17) = 1.5D-5, AK(18) = 1.0D-2,

&END

TEST 8: PU(VI)-TIME DEPENDENCY PU(V),(IV),(III)

1	1	2	100	10		
1.5D03	1.0D02	0	1.0D00	1.0D00		
1.0D00	1.0D00	0.0D00	3.0D-1	1.0D-7	2.0D00	1.0D-6
0.0D00	0.0D00	0.0D00	0.0D00	0.0D00	0.0D00	0.0D00

&CONST

AK(1) = 0.0D0, AK(5) = 0.0D0,

&END

1.2 JCLの説明

(1) システム構成

本コードのシステム構成を図A・1に示す。

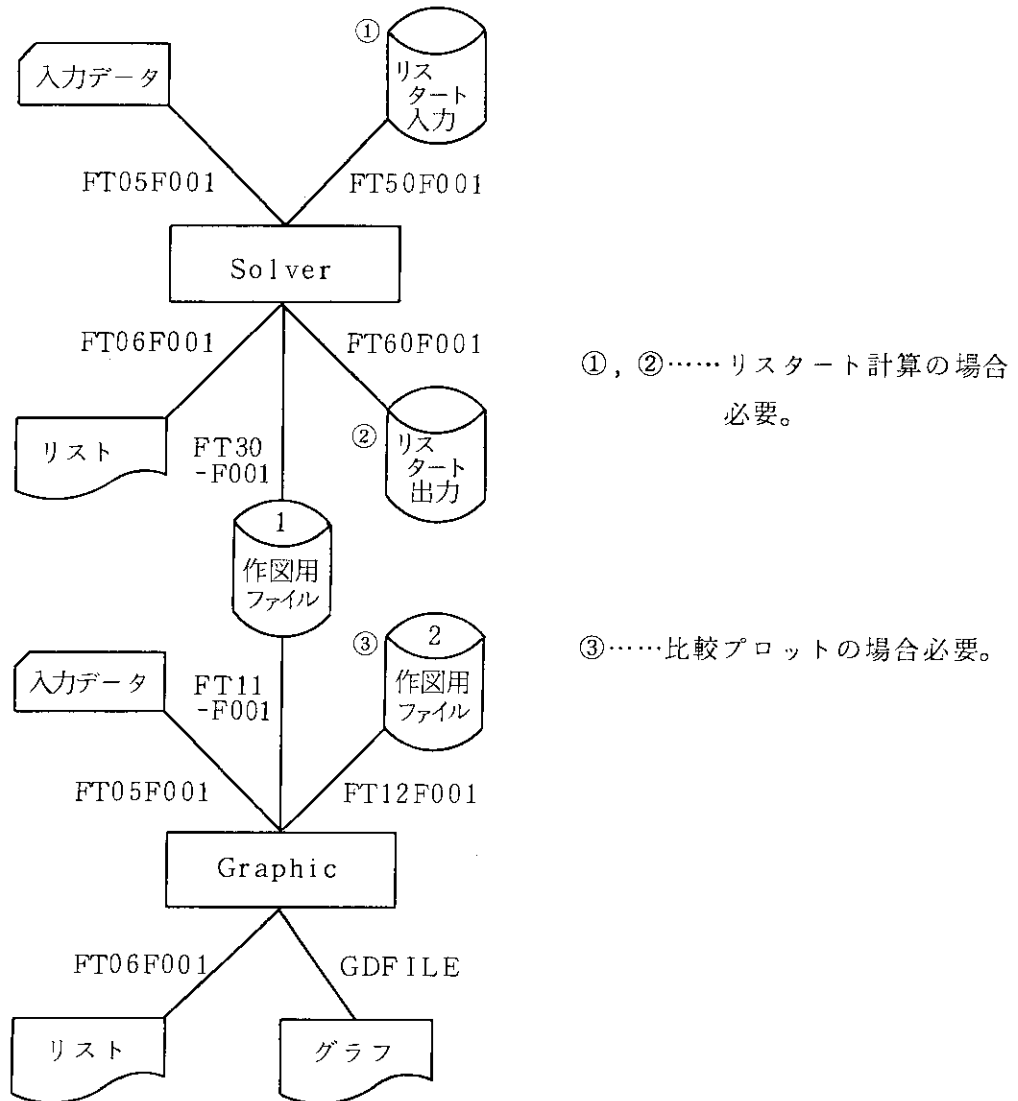


図 A.1 REACTコードのシステム構成

(2) JCL例

REACT コードを実行するための典型的な JCL 例を示す。

1) 計算コードと作図コードを同時に実行する JCL

```
//JCLG      JOB
//          EXEC JCLG
//SYSIN    DD DATA,DLM='++'
// JUSER  xxxxxxxx
           T.6 C.2 W.3 I.4 OPN GRP
           OPTP NOTIFY=J      PASSWORD=xxxxxx
//*****
//FORT77   EXEC FORT77,SO=J      .REACT,
//          A='ELM(*)',NONUM',B='NOPRINT',OPT=3
//SYSINC   DD DSN=J      .REACT.INC.FORT77,DISP=SHR
//LINK     EXEC LKED77
//GO       EXEC GO
//FT60F001 DD DUMMY,
//          DCB=(RECFM=VBS,LRECL=6208,BLKSIZE=6212,DSORG=PS)
//FT50F001 DD DUMMY
//FT30F001 DD DSN=&&GRP,DISP=(NEW,PASS),
//          UNIT=WK10,SPACE=(TRK,(50,50),RLSE),
//          DCB=(RECFM=VBS,LRECL=6208,BLKSIZE=6212,DSORG=PS)
//SYSIN    DD DSN=J      .REACT.DATA(TEST005),DISP=SHR
//*****
//FORT77P  EXEC FORT77,SO=J      .REACTG,
//          A='ELM(*)',NONUM',B='NOPRINT',OPT=3
//SYSINC   DD DSN=J      .REACT.INC.FORT77,DISP=SHR
//LINKP    EXEC LKED77,GRLIB=PNL
//GOP      EXEC GO
//FT11F001 DD DSN=&&GRP,DISP=(OLD,DELETE)
//SYSIN    DD DSN=J      .REACT.DATA(PLOT01),DISP=SHR
// EXPAND GRNLP,SYSOUT=
```

註) &&GRP : 作図用の計算結果を入力するファイルで一時データセットとしている。2つの計算結果を比較(作図)したい時等は保存ファイルに変更する。

TEST005 : 計算条件を与えるためのデータセット(予め準備しておく)。

PLOT01 : 作図のフォーマットを指定するデータセット(予め準備しておく)。

FT60F } : どちらもダミーとしておく。
FT50F }

なお、作図ファイルを一時から保存に変更するには、

FT30Fでは、DISP=(NEW,PASS), UNIT=WK10 を

DISP=(NEW,CATLG), UNIT=USERXX とし、

FT11Fでは、DISP=(OLD,DELETE) を DISP=SHR とする。

2) リスタート計算の場合 (作図実行せず)

```

1 //JCLG      JOB
2 //          EXEC JCLG
3 //SYSIN     DD DATA,DLM='++'
4 // JUSER 99999999,XX.XXXXXX,0000.00
5             T.4 C.2 W.1 I.3 OPN
6             OPTP NOTIFY=J9999 PASSWORD=XXXX
7 //*****
8 //FORT77   EXEC FORT77,SO=J9999.REACT,
9 //          A='ELM(*)',NONUM',B='NOPRINT',
10 //         LCT=62,OPT=3
11 //SYSINC   DD DSN=J9999.REACT.INC.FORT77,DISP=SHR
12 //LKED77   EXEC LKED77
13 //GOGOGOA  EXEC GO
14 //FT60F001 DD DSN=J9999.REACT.RES2,DISP=(NEW,CATLG),
15 //         UNIT=XXXXXX,SPACE=(TRK,(5,5),RLSE),
16 //         DCB=(RECFM=VBS,LRECL=6208,BLKSIZE=6212,DSORG=PS)
17 //FT50F001 DD DSN=J9999.REACT.RES1,DISP=SHR
18 //FT30F001 DD DSN=J9999.REACT.GRP1,DISP=(NEW,CATLG),
19 //         UNIT=XXXXXX,SPACE=(TRK,(30,20),RLSE),
20 //         DCB=(RECFM=VBS,LRECL=6208,BLKSIZE=6212,DSORG=PS)
21 //SYSIN    DD DSN=J9999.REACT.DATA(TESTR),DISP=SHR
22 ++
23 //

```

- ⑭ リスタート出力ファイルを定義
- ⑰ リスタート入力ファイルの指定
- ⑱ 作図 (計算結果入力) ファイルを定義

3) 比較プロットを行う場合

```

1 //JCLG      JOB
2 //          EXEC JCLG
3 //SYSIN     DD DATA,DLM='++'
4 // JUSER 99999999,XX.XXXXXX,0000.00
5             T.4 C.2 W.1 I.3 OPN GRP
6             OPTP NOTIFY=J9999 PASSWORD=XXXX
7 //*****
8 //FORT77P  EXEC FORT77,SO=J9999.REACT.GRAPH,
9 //          A='ELM(*)',NONUM',B='NOPRINT',
10 //         LCT=62,OPT=3
11 //SYSINC   DD DSN=J9999.REACT.INC.FORT77,DISP=SHR
12 //LKED77P  EXEC LKED77,GRLIB=PNL
13 //GOGOGOP  EXEC GO
14 //FT11F001 DD DSN=J9999.REACT.GRP1,DISP=SHR
15 //FT12F001 DD DSN=J9999.REACT.GRP2,DISP=SHR
16 //SYSIN    DD DSN=J9999.REACT.DATA(TESTP2),DISP=SHR
17 // EXPAND GRNLP
18 ++
19 //

```

- ⑭, ⑮ 作図ファイル1, 2の指定

A.2 プログラムの概要

2.1 木構造 <計算コード>

```

MAIN  ----INPUT  ----RSTIN
      |          +---ELMOUT
      |          +---EQCHK
      +---VCALC  ----EPSCHK
      +---FCALC
      +---OUTLST----ELMOUT
      |          +---CXTOUT
      |          +---FUNOUT
      |          +---VELOUT
      +---OUTFIL
      +---SOLVE  ----SOLJAC----TIMSET
      |          |          +---VCALC  ----EPSCHK
      |          |          +---RVKRXJ
      |          |          +---JCALC
      |          |          +---FCALC
      |          |          +---DEFDT
      |          |          +---*DLAX
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---SOLRKG----TIMSET
      |          |          +---*DRKG  ----FUNCO1----VCALC  ----EPSCHK
      |          |          |          +---FCALC
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---SOLHMG----TIMSET
      |          |          +---*DHAMNG----FUNCO1----VCALC  ----EPSCHK
      |          |          |          +---FCALC
      |          |          +---OUTHMG----OUTLST----ELMOUT
      |          |          |          |          +---CXTOUT
      |          |          |          |          +---FUNOUT
      |          |          |          |          +---VELOUT
      |          |          |          +---OUTFIL
      |          +---SOLRKV----TIMSET
      |          |          +---*ODDRK1----FUNCO2----VCALC  ----EPSCHK
      |          |          |          +---FCALC
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---SOLADM----TIMSET
      |          |          +---*DODAM  ----FUNCO2----VCALC  ----EPSCHK
      |          |          |          +---FCALC
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---SOLEUL----TIMSET
      |          |          +---EULRBG----FUNCO2----VCALC  ----EPSCHK
      |          |          |          +---FCALC
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---SOLJRB----TIMSET
      |          |          +---JACRBG----VCALC  ----EPSCHK
      |          |          |          +---RVKRXJ
      |          |          |          +---JCALC
      |          |          |          +---FCALC
      |          |          |          +---*DLAX
      |          |          +---OUTLST----ELMOUT
      |          |          |          +---CXTOUT
      |          |          |          +---FUNOUT
      |          |          |          +---VELOUT
      |          |          +---OUTFIL
      |          +---ELMOUT
      +---RSTOUT
    
```

<作図コード>

```

MAIN  ----*ERRSET
      +--FILGET
      +--GETCON
      +--PSCALE-----SCALEO
      +--PFRAME-----NEWPAG----*PLOTS
      I          I          +--*PLOT
      I          +--*NEWPEN
      I          +--*PLOT
      I          +--XDIVSN----*NEWPEN
      I          I          +--*PLOT
      I          I          +--*SYMBOL
      I          I          +--GEXP10----*SYMBOL
      I          +--YDIVSN----*NEWPEN
      I          I          +--*PLOT
      I          I          +--*SYMBOL
      I          I          +--GEXP10----*SYMBOL
      I          +--YDIVRY----*NEWPEN
      I          I          +--*PLOT
      I          I          +--*SYMBOL
      I          I          +--GEXP10----*SYMBOL
      I          +--*SYMBOL
      +--PLINES----*NEWPEN
      I          +--*PLOT
      I          +--*SYMBOL
      +--*PLOT
    
```

2.2 コモンの説明

本コードでは一部のCOMMON (COMHMG) を除いて全てインクルードファイルを用いて定義している。この中ではCOMMON文の他に、IMPLICIT文、PARAMETER文も使用している。

IMPLICIT文では実数型変数を倍精度として宣言している(作図コード内では一部単精度)。

PARAMETER文ではコード内で使用している定数を定義している。

定数の意味は、

(計算コード)

NELM	化学種の数(13)
NREF	反応式の数(27)
NDTC	時間データの最大数(5)

(作図コード)

NFS	作図ファイルの最大数(2)
NDC	1ファイル当りの最大有効データ数(1200)
NGS	1グラフに描ける項目の最大数(13)
NEM	NELMと同じ
NRF	NREFと同じ

である。()内は定数値である。これらの値を変えることによってコード内制限値を変えることができる。

以下に各COMMONの説明を行う。

Module Description Sheet				
Module name		COMC	Project name	計算コード
Module type		コモン	Coder	
Module number			Date	
変数名	型	サイズ	意味	
CPGM	CH*8	-	コード名 (REACT)	
CTIT	CH*72	-	計算タイトル	
CELM	CH*5	NELM	化学種 xi の名称	
Module name		COMHMG		
HHO	R*8	-	初期きざみ h。	
TL	R*8	-	リスト出力間隔用変数	
このコモンは、Haming 法を用いた際にサブルーチン SOLHMG から OUTHMG へ必要データを渡すためのものである。				
Module name		COMI		
IOPT	I*4	-	解法の指定	
ICLK	I*4	-	チェック・リスト・フラグ	
ICYC	I*4	-	サイクル数	
IDTS	I*4	-	時間データ番号	
NDTS	I*4	-	時間データの数	
IFINT	I*4	-	作図ファイルへの出力インターバル	
NEPS	I*4	-	エラーメッセージの最大出力回数	
NEPSC	I*4	-	エラー回数のカウンタ	
詳細は A.1.1 節参照。				
Module name		COMR1		
TIMEO	R*8	-	計算開始時刻 (リスタート以外は 0)	
TIME	R*8	-	現時刻	
DT	R*8	-	時間キザミ幅	
TEND	R*8	NDTC	計算終了時刻	
TLST	R*8	NDTC	リスト出力間隔	

Module Description Sheet			
Module name		COMR 1	Project name 計算コード
Module type		コモン	Coder
Module number			
変数名	型	サイズ	意味
DTT	R*8	NDTC	A.1.1節参照
TPRM1	R*8	NDTC	
TPRM2	R*8	NDTC	
Module name		COMR 2	
CN	R*8	-	全硝酸根濃度 C_N
CN04	R*8	-	$C_N^{0.4}$
AX	R*8	NELM	化学種 x_i 濃度
AK	R*8	NREF	反応速度定数 K_k
AV	R*8	NREF	反応速度 v_k
AF	R*8	NELM	化学種 x_i の生成速度 $f_i = dx_i/dt$
AJ	R*8	NELM, NELM	Jacobian J_{ij}
AC	R*8	NELM, NREF	化学量論係数 C_{ik}
RVRX	R*8	NREF, NELM	$\partial v_k / \partial x_j$
Module name		COMR 3	
XPUO	R*8	-	初期の全 Pu 濃度
XUO	R*8	-	初期の全 U 濃度
XNPO	R*8	-	初期の全 N_p 濃度
TEPSC	R*8	NELM	負濃度の積算用
Module name		COMR 4	
DTMAX	R*8	-	DTの最大値
DTMIN	R*8	-	DTの最小値
IDTMX	I*4	-	最大のDTとなったサイクル数
IDTMN	I*4	-	最小のDTとなったサイクル数
このデータはPorsing法, M-Porsing法のみで使用する。			

Module Description Sheet				
Module name		COMC	Project name	作図コード
Module type		コモン	Coder	
Module number			Date	
変数名	型	サイズ	意味	
CTIT	CH*72	—	主タイトル	
CAPT	CH*72	—	サブ・タイトル	
CPRM	CH*32	NGS	項目のキャプション	
CELM	CH*5	NEM,NFS	化学種の名称	
Module name		COMI 1		
NFILE	I*4	—	作図ファイルの数	
NGCT	I*4	—	一図に描く項目の数	
NGTP	I*4	—	各軸の指定	
IVFLG	I*4	—	z 軸 (生成反応速度) のフラグ 0 = z 軸なし 1 = z 軸のみあり 2 = y 軸, z 軸の両方あり	
NELM	I*4	NFS	ファイル i の化学種の数	
NREF	I*4	NFS	ファイル i の化学反応式の数	
NDCT	I*4	NFS	ファイル i のデータ点数	
NGPM	I*4	2,NGS	項目のファイル番号, 化学種番号の格納 (1, i)……項目 i のファイル番号 (2, i)……項目 i の化学種番号/反応式番号	
詳細はA.1.1節参照。なお NGPMは入力データ k1, k2 を格納。				
Module name		COMI 2		
IMX0	I*4	—	x 軸目盛の上・下限	
IMX9	I*4	—	線形の場合 $IMX0 \cdot 10^{IEXX} \sim IMX9 \cdot 10^{IEXX}$	
IEXX	I*4	—	対数の場合 $10^{IMX0} \sim 10^{IMX9}$	

Module Description Sheet				
Module name		COMI 2	Project name	作図コード
Module type		コモン	Coder	
Module number			Date	
変数名	型	サイズ	意味	
IMY0	I*4	—	y軸目盛の上・下限 x軸と同様	
IMY9	I*4	—		
IEXY	I*4	—		
IEM0	I*4	—	z軸目盛の上下限	
IEM9	I*4	—	線形の場合 $IEM0 \cdot 10^{IEP0} \sim IEM9 \cdot 10^{IEP0}$	
IEP0	I*4	—	対数の場合(負側) $10^{IEM0} \sim 10^{IEM9}$	
IEP9	I*4	—	(正側) $10^{IEP0} \sim 10^{IEP9}$	
IPN0	I*4	—	図に描く際のデータ点数	
Module name		COMR1		
XINP0	R*8	—	入力で与えられたx軸目盛の上下限	
XINP9	R*8	—		
YINP0	R*8	—	入力で与えられたy軸目盛の上下限	
YINP9	R*8	—		
YVEL0	R*8	—	入力で与えられたz軸目盛の上下限, 現在無意味	
YVEL9	R*8	—		
TIME	R*8	NDC, NFS	時刻データ	
CONC	R*8	NDC, NEM, NFS	濃度データ	
XFUN	R*8	NDC, NEM, NFS	生成速度データ	
XVEL	R*8	NDC, NRF, NFS	反応速度データ	

Module Description Sheet				
Module name		COMR2	Project name	作図コード
Module type		コモン	Coder	
Module number			Date	
変数名	型	サイズ	意味	
SFX	R*4	-	x軸のスケール・ファクター	
SFY	R*4	-	y軸のスケール・ファクター	
SFV	R*4	-	z軸のスケール・ファクター	
SXL	R*4	-	x軸長さ (mm)	
SYL	R*4	-	y, z軸長さ (mm)	
SX0	R*4	-	x軸目盛の上下限值	
SX9	R*4	-		
SY0	R*4	-	y軸目盛の上下限值	
SY9	R*4	-		
SV0	R*4	-	z軸目盛の上下限 (線形の場合)	
SV9	R*4	-		
SYM	R*4	-	z軸目盛が対数の場合, 正側, 負側目盛の絶対値が 最小となる座標値	
SYP	R*4	-		
SEM	R*4	-	z軸目盛が対数の場合, 正側, 負側目盛の絶対値の 最小値	
SEP	R*4	-		
SZP	R*4	-	z軸目盛が対数の場合, SYMとSYPの間を3等 分し, 正側をSZP, 負側をSZM, 中央をSZZ とする (極小部分を正負で区別する)。	
SZZ	R*4	-		
SZM	R*4	-		
SX	R*4	NDC, NGS	プロット座標	
SY	R*4	NDC, NGS		

2.3 ルーチンの説明

計算コード及び作図コードについて各ルーチンの名称と機能を示す。

各モジュール(サブルーチン)の名称と機能

(計算コード用モジュール)

モジュール名	機 能	備 考
BLOCKD	コモン変数に初期値を与える。	AKには標準的な反応速度定数, ACには反応における化学量論係数を与える。
CXTOUT	全Pu, 全Np, 全U濃度をリスト出力する。	
DEFDT	(修正)Porsing法において, DTを自動的に設定する。	詳細はA1.1節参照。
ELMOUT	各化学種濃度をリスト出力する。	
EPSCHK	負濃度のチェックを行う。	[Pu(III)], [HNO ₃], [HNO ₂]については, 濃度0ではいけないので最小値を10 ⁻¹⁰ としている。
EQCHK	化学反応式のチェック・リストを行う。	
EULRBG	Euler-Romberg法	詳細は本文第3.5節参照。
FCALC	f(x _i)を計算する。(v _k より)	$f_i = \sum C_{ik} v_k$
FUNCO1	f(x _i)を計算する。(x _i →v _k →f _i)	SSL II ルーチンDRKG(R-K-G法), DHAMNG(Haming法)より呼び出される。
FUNCO2	f(x _i)を計算する。(x _i →v _k →f _i)	SSL II ルーチンDODRK1(R-K-V法), DODAM(Adams法)及びEULRBG(Euler-Romberg法)より呼び出される。
FUNOUT	f(x _i)値をリスト出力する。	
INPUT	計算データを入力する。	変数の意味については第A1.1節参照

モジュール名	機 能	備 考
JACRBG	Jacobi - Romberg 法	詳細は本文第 3.5 節を参照。
JCALG	$\partial v_k / \partial x_j$ から Jacobian J_{ij} を計算する。	$J_{ij} = \sum_k C_{ik} \frac{\partial v_k}{\partial x_j}$
MAIN	計算コード・メイン・ルーチン	
OUTFIL	作図ファイルへの出力を行う。	
OUTHMG	Haming 法による計算結果をリスト、ファイル出力する。	SSLII ルーチン DHAMNG より呼び出される。
OUTLST	タイム・ステップ毎のリスト出力を行う。	
RSTIN	リスタートファイルの入力を行う。	
RSTOUT	リスタートファイルへの出力を行う。	
RVKRXJ	$\partial v_k / \partial x_j$ を計算する。	詳細は本文第 3.3 節を参照。
SOLADM	Adams 法の制御	Adams 法については「SSL II 使用手引書*」を参照。
SOLEUL	Euler - Romberg 法の制御	Euler - Romberg 法については本文第 3.5 節参照。
SOLHMG	Haming 法の制御	Haming 法については「SSL II 使用手引書*」を参照。
SOLJAC	Porsing , M-Porsing 法	Porsing 法, M-Porsing 法については本文第 3.1 節を参照。
* 富士通「SSL II 使用手引書」		

モジュール名	機能	備考
SOLJRB	Jacobi-Romberg 法の制御	Jacobi-Romberg 法については本文第 3.5 節を参照。
SOLRKG	R-K-G 法の制御	R-K-G 法については「SSL II 使用手引書」を参照。
SOLRKV	R-K-V 法の制御	R-K-V 法については「SSL II 使用手引書」を参照。
SOLVE	常微分方程式の解の制御	
TIMSET	時間データの標準化を行う。	時刻 T_1 で DT を変えた場合の例 $NC1=3$ $NC2=6$ $XDT=3 \cdot DT_2 - XTIM$ TL についても同様
VCALC	反応速度 v_k を計算する。	詳細は本文第 2.2 節を参照。
VELOUT	反応速度 v_k をリスト出力する。	

(以下作図用モジュール)

モジュール名	機能	備考
BLOCKD	コモン変数に初刻値を与える。	図のフレームの大きさ
FILGET	作図ファイルを入力する。	
GETCON	作図データを入力する。(一図毎)	変数の意味は第 A 1.1 節を参照
GEXP10	10^n の文字を描く。	

モジュール名	機 能	備 考
MAIN	作図コードのメイン・ルーチン	
NEWPAG	グラフ出力のページ替えを行う。	最初の呼び出しのときは、PNL オープン 原点を(30,10)とする。
PERAME	図のフレーム、キャプションを描く。	
PLINES	一図のグラフを描く。	座標は, $x_i = (x_i - x_0) \cdot f_x \dots\dots\dots \text{線形}$ $x_i = (\log x_i - \log x_0) \cdot f_x \dots\dots \text{対数}$ の変換で求める。(y, zも同様) ただし、反応(生成)速度で対数軸の場合、正側と負側を分ける必要がある。
PSCALE	データの最大値、最小値を求めてスケールリングを行う。	変数の意味についてはA.1.1節参照。 本ルーチンでは各軸の最大値、最小値を検索し、SCALEOルーチンで値を規格化したのちスケールリング・ファクターを出す。 $f_x = X_L / (S_9 - S_0)$ $f_x : \text{ファクター}$ $X_L : \text{軸長さ}$ $\left. \begin{array}{l} S_9 \\ S_0 \end{array} \right\} \text{規格化した最大・最小値}$ 反応・生成速度については PLINESルーチンを参照(対数軸の場合、正側と負側に分けて規格化する)。

モジュール名	機能	備考
SCALEO	最大値，最小値を規格化する。	<p>規格化とは</p> <p>(線形) $n_1 \times 10^m \leq R_{\min}$, $R_{\max} \leq n_2 \times 10^m$ $\{n_1, n_2 : 100 \text{未満の正整数},$ $m : \text{整数}\}$</p> <p>(対数) $10^{k_1} \leq R_{\min}, R_{\max} \leq 10^{k_2}$ $\{k_1, k_2 : \text{整数}\}$</p> <p>となるように最大，最小範囲を拡張することをいう。</p> <p>(1) 線形の場合 $LM0 = n_1$ $LM9 = n_2$ $LEX = m$ $SR0 = n_1 \times 10^m$ $SR9 = n_2 \times 10^m$</p> <p>(2) 対数の場合 $LM0 = k_1$ $LM9 = k_2$ LEX は使用しない。 $SR0 = k_1$ $SR9 = k_2$</p> <p>〔注意〕 n_1, n_2 についてはグラフ上の目盛が等間隔になるよう調整する。k_1 については $k_1 = \max(k_1, k_2 - 8)$ の調整をする。</p>
XDIVSN	x 軸の目盛を描く。	
YDIRV	z 軸 (反応・生成速度) の目盛を描く。	PLINES ルーチンを参照。
VDIVSN	y 軸 (濃度) の目盛を描く。	

A.3 プログラムリスト

REACT. INC. FORT77

```
*****
* COMGRP **
*****
```

```

      IMPLICIT REAL*8 ( A-H, O-R, T-Z )
      IMPLICIT REAL*4 ( S )
C
      PARAMETER ( NFS = 2, NDC = 1200, NGS = 13 )
      PARAMETER ( NEM = 13, NRF = 27 )
C
      COMMON / COMI1 / NFILE, NGCT, NGTP, IVFLG,
      . NELM(NFS), NREF(NFS), NDCT(NFS), NGPM(2, NGS)
      COMMON / COMI2 / IMXO, IMX9, IEXX,
      . IMYO, IMY9, IEXY,
      . IEMO, IEM9, IEPO, IEP9,
      . IPNO(NGS)
      COMMON / COMR1 / XINPO, XINP9, YINPO, YINP9,
      . YVELO, YVEL9,
      . TIME(NDC, NFS), CONC(NDC, NEM, NFS),
      . XFUN(NDC, NEM, NFS), XVEL(NDC, NRF, NFS)
      COMMON / COMR2 / SFX, SFY, SFV, SXL, SYL,
      . SXO, SX9, SYO, SY9,
      . SVO, SV9, SYM, SYP,
      . SEM, SEP, SZP, SZZ, S2M,
      . SX(NDC, NGS), SY(NDC, NGS)
      COMMON / COMC / CTIT, CAPT, CPRM(NGS), CELM(NEM, NFS)
      CHARACTER CTIT*72, CAPT*72, CPRM*32, CELM*5
```

```
*****
* COMMON **
*****
```

```

      IMPLICIT REAL*8 ( A-H, O-Z )
C
      PARAMETER ( NELM = 13, NREF = 27, NDTC = 5 )
C
      COMMON / COMI / IOPT, ICHK, ICYC, IDTS, NDTs, IFINT,
      . NEPS, NEPSC(NELM)
      COMMON / COMR1 / TIMEO, TIME, DT,
      . TEND(NDTC), TLST(NDTC), DTT(NDTC),
      . TPRM1(NDTC), TPRM2(NDTC)
      COMMON / COMR2 / CN, CNO4, AX(NELM),
      . AK(NREF), AV(NREF), AF(NELM),
      . AJ(NELM, NELM), AC(NELM, NREF), RVRX(NREF, NELM)
      COMMON / COMR3 / XPUO, XUO, XNPO, TEPSC(NELM)
      COMMON / COMR4 / DTMAX, DTMIN, IDTMX, IDTMN
      COMMON / COMC / CPGM, CTIT, CELM(NELM)
      CHARACTER CPGM*8, CTIT*72, CELM*5
```

REACT.FORT77

 ** BLOCKD **

```

      BLOCK DATA  BLOCKD
*INCLUDE  COMMON
C
      PARAMETER  ( NNN = NELM*NREF )
C
C**  CONSTANTS OF REACTION VELOCITY ***
      DATA  AK  / 5.00D+3, 7.32D-1, 3.80D-2, 7.00D-1, 3.00D+4,
.                2.50D+0, 7.00D+1, 8.30D+0, 7.00D+0, 2.25D+0,
.                9.21D+1, 2.50D+2, 3.70D+1, 7.90D+0, 2.76D+2,
.                3.58D+0, 1.00D-4, 2.00D-2, 8.00D-4, 9.00D-4,
.                4.75D-4, 1.00D+0, 5.98D-2, 3.14D-1, 1.65D-3,
.                3.45D-1, 4.36D+2/
C
C**  CHEMICAL COEFFICIENTS IN REACTION EQUATION ***
C.. EQUATION 1 ...
      DATA  (AC( 1, I), I = 1, NREF)
. /  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
.   ODO,  ODO, +1DO,  ODO, -2DO, -1DO, +2DO,  ODO,  ODO,
.   ODO,  ODO,  ODO,  ODO,  ODO,  ODO, -1DO,  ODO, +1DO,  ODO/
C.. EQUATION 2 ...
      DATA  (AC( 2, I), I = 1, NREF)
. /  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
.   ODO,  ODO, -1DO, -1DO, +2DO, +1DO, -2DO,  ODO,  ODO,
.   ODO,  ODO,  ODO,  ODO,  ODO,  ODO, +1DO, +1DO, -2DO,  ODO/
C.. EQUATION 3 ...
      DATA  (AC( 3, I), I = 1, NREF)
. / -2DO, -1DO, -1DO, +2DO,  ODO,  ODO, -1DO,  ODO,  ODO,
.   ODO,  ODO, -1DO, +2DO,  ODO, +1DO,  ODO,  ODO,  ODO,
.   ODO,  ODO,  ODO, +1DO, +1DO,  ODO, -2DO, +1DO,  ODO/
C.. EQUATION 4 ...
      DATA  (AC( 4, I), I = 1, NREF)
. / +2DO, +1DO, +1DO, -2DO,  ODO,  ODO, +1DO,  ODO,  ODO,
.   ODO,  ODO, +1DO, -1DO,  ODO, -1DO,  ODO,  ODO,  ODO,
.   ODO,  ODO,  ODO, -1DO, -1DO,  ODO, +1DO,  ODO,  ODO/
C.. EQUATION 5 ...
      DATA  (AC( 5, I), I = 1, NREF)
. / +2DO, +2DO,  ODO, -2DO,  ODO, +4DO, +4DO,  ODO, +4DO,
.   -4DO, +1DO,  ODO, -4DO, +3DO,  ODO, -3DO, -1DO, +1DO,
.   -4DO, -3DO, +5DO,  ODO, -4DO,  ODO, +4DO, -4DO, +1DO/
C.. EQUATION 6 ...
      DATA  (AC( 6, I), I = 1, NREF)
. /  ODO,  ODO,  ODO, +1DO, -1DO,  ODO,  ODO,  ODO,  ODO,
.   ODO,  ODO,  ODO,  ODO, -1DO,  ODO, +1DO, +1DO, -1DO,
.   ODO, +1DO, +1DO,  ODO,  ODO,  ODO,  ODO,  ODO, -1DO/
C.. EQUATION 7 ...
      DATA  (AC( 7, I), I = 1, NREF)
. /  ODO,  ODO,  ODO,  ODO,  ODO, -1DO,  ODO, -1DO, -2DO,
.   ODO, -1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
.   +1DO, +2DO,  ODO, -1DO,  ODO,  ODO,  ODO,  ODO,  ODO/
C.. EQUATION 8 ...

```

```

      DATA (AC( 8, I), I = 1, NREF)
      . /  ODO,  ODO,  ODO,  ODO,  ODO,  +2DO,  +1DO,  +1DO,  +2DO,
      . -2DO, +1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      . -2DO, -2DO, +2DO, +1DO, -1DO,  ODO,  ODO,  ODO,  ODO/
C.. EQUATION 9 ...
      DATA (AC( 9, I), I = 1, NREF)
      . /  ODO,  ODO,  ODO,  ODO,  ODO,  -1DO,  -1DO,  ODO,  ODO,
      . +2DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      . +1DO,  ODO, -2DO,  ODO, +1DO,  ODO,  ODO,  ODO,  ODO,  ODO/
C.. EQUATION 10 ...
      DATA (AC(10, I), I = 1, NREF)
      . / +1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  +1DO,
      . +1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      .  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO/
C.. EQUATION 11 ...
      DATA (AC(11, I), I = 1, NREF)
      . / -1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  -1DO,
      . -1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      .  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO/
C.. EQUATION 12 ...
      DATA (AC(12, I), I = 1, NREF)
      . /  ODO, -1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      .  ODO, -1DO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      .  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO, -1DO/
C.. EQUATION 13 ...
      DATA (AC(13, I), I = 1, NREF)
      . /  ODO,  ODO, -1DO,  ODO,  ODO, -1DO,  ODO,  ODO, -1DO,  ODO,
      .  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,  ODO,
      .  ODO,  ODO,  ODO,  ODO,  ODO,  ODO, -2DO,  ODO,  ODO,  ODO/
C
C** INITIAL ZERO CLEAR ***
      DATA  RVRX      / NNN*O.ODO /
      DATA  DTMAX, DTMIN / O.ODO, 9.999D20 /
      DATA  IDTMX, IDTMN / 2*O /
      DATA  NEPSC     / NELM*O /
      DATA  TEPSC     / NELM*O.ODO /
C
C** PROGRAM NAME ***
      DATA  CPGM      / ' REACT ' /
C
C** ELEMENTS NAME ***
      DATA  CELM
      . / 'PU(6)', 'PU(5)', 'PU(4)', 'PU(3)', 'HNO3 ', 'HNO2 ', 'NP(6)',
      . 'NP(5)', 'NP(4)', 'U(6) ', 'U(4) ', 'AAA ', 'BBB ' /
C
      END

```

 ** CXTOUT **

```

C*****
C** CXTOUT ***
C*****
      SUBROUTINE  CXTOUT

```

```

C
C** LIST OUTPUT OF TOTAL CONC. OF PU, NP, U ***
*INCLUDE COMMON
C
XPU = AX(1) + AX(2) + AX(3) + AX(4)
XNP = AX(7) + AX(8) + AX(9)
XU = AX(10) + AX(11)
C
IF (XPUO .GT. 0.000) THEN
  QPU = 1.0D2*(XPU/XPUO - 1.000)
ELSE
  QPU = 0.000
END IF
C
IF (XNPO .GT. 0.000) THEN
  QNP = 1.0D2*(XNP/XNPO - 1.000)
ELSE
  QNP = 0.000
END IF
C
IF (XUO .GT. 0.000) THEN
  QU = 1.0D2*(XU/XUO - 1.000)
ELSE
  QU = 0.000
END IF
C
WRITE(6, 6000) 'PU', XPUO, XPU, QPU,
.              'NP', XNPO, XNP, QNP,
.              'U ', XUO, XU, QU
RETURN
C
6000 FORMAT(1H0,3X,'*** TOTAL CONCENTRATION VALANCE ***'
.          /1H ,6X,12X,'INIT',10X,'NOW',11X,'DEF. (%)'
.          /1H ,6X,'TOTAL ',A,1P3D14.4))
END

```

```

*****
* DEFDT **
*****

```

```

C*****
C** DEFDT ***
C*****
SUBROUTINE DEFDT ( EPS, EPS2 )
C
C** DEFINE DT OF (M)-PORSING ***
*INCLUDE COMMON
C
QMAX = 0.000
QMAXR = 0.000
DO 4100 I = 1, NELM
  DO 4000 J = 1, NELM
    QMAXR = MAX( QMAXR, ABS( AJ(J, I) ) )
4000 CONTINUE
QMAX = MAX( QMAX, ABS( AF(I) ) )

```

```

4100 CONTINUE
C
      IF (QMAX .LE. 0.000) QMAX = 1.000
      IF (QMAXR .LE. 0.000) QMAXR = 1.000
      DT = 7.000*MIN( EPS/QMAX/400.000, EPS2/SQRT(QMAXR)/40.000 )
CCCCC DT = 7.000*MIN( EPS/QMAX/400.000, 5.000*EPS2/QMAXR )
C
      RETURN
      END

```

```

*****
ELMOUT **
*****

```

```

C*****
C** ELMOUT ***
C*****
      SUBROUTINE ELMOUT ( XX )
C
C** LIST OUTPUT OF ELEMENTAL CONCENTRATION ***
*INCLUDE COMMON
C
      DIMENSION XX(NELM)
C
      PARAMETER ( NLL = 7 )
      PARAMETER ( NNN = (NELM - 1)/NLL + 1 )
C
      IE = 0
*VOCL LOOP, SCALAR
      DO 4000 I = 1, NNN
          IS = IE + 1
          IE = MIN(NELM, IE + NLL)
          WRITE(6, 6000) (CELM(J), J = IS, IE)
          WRITE(6, 6100) (XX(J), J = IS, IE)
      4000 CONTINUE
C
      RETURN
C
      6000 FORMAT(1H0,7A14)
      6100 FORMAT(1H ,5X,1P7D14.4)
      END

```

```

*****
EPSCHK **
*****

```

```

C*****
C** EPSCHK ***
C*****
      SUBROUTINE EPSCHK ( WX )
C
C** CONCENTRATION MINIMUM CHECK ***
*INCLUDE COMMON
C
      DIMENSION WX(NELM)

```

```

      DIMENSION  EPSO(NELM),  ICKT(NELM)
      DATA      EPSO  / 3*0.000, 3*1.00-10, 7*0.000 /
      DATA      NECNT / 0 /
C
      IF (IOPT .GE. 6) GO TO 9999
      ICK      = 0
      DO 4000 I = 1, NELM
        IF (WX(I) .LT. EPSO(I)) THEN
          ICK      = ICK + 1
          ICKT(ICK) = I
          NEPSC(I) = NEPSC(I) + 1
          TEPSC(I) = TEPSC(I) + WX(I) - EPSO(I)
        END IF
      4000 CONTINUE
C
      IF (ICK .GT. 0) THEN
        IF (NEPS.EQ.0 .OR. NECNT.LE.NEPS) THEN
          NECNT      = NECNT + 1
          WRITE(6, 6000) TIME, ICYC
          WRITE(6, 6100) 'OLD', (CELM(ICKT(I)), WX(ICKT(I)), I=1, ICK)
          WRITE(6, 6100) 'NEW', (CELM(ICKT(I)), EPSO(ICKT(I)), I=1, ICK)
        END IF
      *VOCL LOOP, SCALAR
      DO 4100 I = 1, ICK
        WX(ICKT(I)) = EPSO(ICKT(I))
      4100 CONTINUE
      END IF
C
      9999 CONTINUE
      RETURN
C
      6000 FORMAT(1H0, '*** WARNING IN EPSCHK, NEXTS CONC. ARE MODIFIED',
        .      ' AT TIME :', 1PD13.4, ',', 18, ' CYCLES')
      6100 FORMAT(1H , ' <<', A, ' CONCENTRATION>>',
        .      3(/1H , 2X, 5(2X, A, 1PD9.1)))
      END

```

```

*****
* EQCHK **
*****

```

```

C*****
C** EQCHK ***
C*****
      SUBROUTINE  EQCHK
C
C** CHECK OF CHEMICAL EQUATIONS ***
C
*INCLUDE  COMMON
C
      CHARACTER*80  CP, CM
      CHARACTER*3   CW
C
      WRITE(6, 6000)  CPGM, CTIT
C

```



```

*VOCL LOOP, SCALAR
DO 4200 I = 1, NREF
  CP      = ' '
  CM      = ' '
  NP      = 0
  NM      = 0
C
  DO 4000 J = 1, NELM
    IF (AC(J, I) .GT. 0.000) THEN
      IF (NP .GT. 0) THEN
        CP(NP+1:NP+3) = ' + '
        NP      = NP + 3
      END IF
      IF (AC(J, I) .GT. 1.000) THEN
        WRITE(CW, '(F3.0)') AC(J, I)
        CP(NP+1:NP+2) = CW(2:2)//'*'
        NP      = NP + 2
      END IF
      CP(NP+1:NP+5) = CELM(J)
      NP      = NP + 5
C
      ELSE IF (AC(J, I) .LT. 0.000) THEN
        IF (NM .GT. 0) THEN
          CM(NM+1:NM+3) = ' + '
          NM      = NM + 3
        END IF
        IF (AC(J, I) .LT. -1.000) THEN
          WRITE(CW, '(F3.0)') -AC(J, I)
          CM(NM+1:NM+2) = CW(2:2)//'*'
          NM      = NM + 2
        END IF
        CM(NM+1:NM+5) = CELM(J)
        NM      = NM + 5
      END IF
    4000 CONTINUE
  C
    CM(NM+1:) = ' ==> '//CP(1:NP)
    WRITE(6, 6100) I, AK(I), CM
  4200 CONTINUE
C
  RETURN
C
6000 FORMAT(1H1, '<<< ', A, '>>> ', A
.          /1HO, '*****'
.          /1H, '** CHEMICAL REACTION EQUATIONS **'
.          /1H, '*****')
6100 FORMAT(1HO, ' K', I2.2, ' = ', 1PD11.3, 3X, A)
END

```

```

*****
* EULRBG **
*****

```

```

C
C** INTEGRATION BY EULER-ROMBERG

```

```

C
      SUBROUTINE  EULRBG  ( NN,  FX,  XX,  TX,  FUNC,  TIME,  DT,
                        EPS,  MXI,  ICON )
C
C** ARGUMENTS ***
C
C   NN   (IN)  NUMBER OF INDEPENDENT VARIABLES
C   FX   (WK)  ARRAY FOR FUNCTION VALUE (WORK)
C   XX   (IN)  ARRAY FOR INDEPENDENT VARIABLE'S VALUE
C   TX   (WK)  ROMBERG TABLE, SIZE FOR (MXI*NN)
C   FUNC (IN)  FUNCTION TO CALCULATE DF/DT
C           ARG LIST, CALL FUNC ( TIME, XX, FX )
C   TIME (IO)  TIME
C   DT   (IN)  TIME DIFFERENCE ( NEXT TIME IS TIME + DT )
C   EPS  (IN)  RELATIVE EPSILON
C   MXI  (IN)  MAXIMUM ITERATION NUMBER
C   ICON (OT)  CONDITION CODE ( NEGATIVE FOR ITERATION NUMBER )
C
      IMPLICIT  REAL*8 ( A-H, O-Z )
      DIMENSION  FX(NN), XX(NN), TX(NN, MXI)
      EXTERNAL  FUNC
C
C** INITIAL INTEGRATION ***
      CALL FUNC  ( TIME, XX, FX )
      DO 4000 I = 1, NN
         TX(I, 1) = XX(I) + DT*FX(I)
      4000 CONTINUE
C
C** DIVISIONAL INTEGRATION ***
      NDV      = 1
      DO 4800 L = 1, MXI - 1
         NDV    = 2*NDV
C
C..... INITIAL SET ...
         DO 4100 I = 1, NN
            TX(I, L + 1) = XX(I)
      4100 CONTINUE
C
C..... DIVISION ...
         DTX    = DT/NDV
         DO 4300 J = 1, NDV
            TIM  = TIME + (J - 1)*DTX
            CALL FUNC  ( TIM, TX(I, L + 1), FX )
            DO 4200 I = 1, NN
               TX(I, L + 1) = TX(I, L + 1) + DTX*FX(I)
      4200 CONTINUE
      4300 CONTINUE
C
C..... "ROMBERG" INTERPOLATION AND CONVERGE CHECK ...
         M2     = 1
         IF (L .LE. 1) GO TO 3100
         DO 4600 K = L, 2, -1
            M2   = 2*M2
            DO 4400 I = 1, NN
               TX(I, K) = (M2*TX(I, K + 1) - TX(I, K))/(M2 - 1)

```

```

4400      CONTINUE
C
      DO 4500 I = 1, NN
          ERR = ABS(TX(I, K) - TX(I, K - 1))
          IF (ERR .GT. EPS*ABS(TX(I, K - 1))) GO TO 3000
4500      CONTINUE
C
C..... CONVERGED ...
          ICON = -L
          ICNV = K
          GO TO 3300
3000      CONTINUE
4600      CONTINUE
C
C..... NOT CONVERGE, CALCULATE AT K = 1 ...
3100      CONTINUE
          M2 = 2*M2
          DO 4700 I = 1, NN
              TX(I, 1) = (M2*TX(I, 2) - TX(I, 1))/(M2 - 1)
4700      CONTINUE
C
4800 CONTINUE
C
C*** NOT CONVERGE IN MAXIMUM ITERATION ***
          ICON = 10
          ICNV = 1
C
C*** CONVERGED ***
3300 CONTINUE
          DO 4900 I = 1, NN
              XX(I) = TX(I, ICNV)
4900 CONTINUE
          TIME = TIME + DT
C
9999 CONTINUE
      RETURN
      END

```

```

*****
< FCALC **
*****

```

```

C*****
C** FCALC ***
C*****
      SUBROUTINE FCALC
C
C** CALCULATE FUNCTION F(X) ***
*INCLUDE COMMON
C
      DO 4200 I = 1, NELM
          AF(I) = 0.000
          DO 4000 K = 1, NREF
              AF(I) = AF(I) + AC(I, K)*AV(K)
4000      CONTINUE

```

```

4200 CONTINUE
C
      RETURN
      END

*****
** FUNCO1 **
*****

C*****
C** FUNCO1 ***
C*****
      SUBROUTINE FUNCO1 ( YY, FF )
C
C** CALCULATE @F/@T FOR RKG AND HAMNG (SSL-II) ***
*INCLUDE COMMON
C
      PARAMETER ( K = NELM + 1 )
      DIMENSION YY(K), FF(K)
C
      CALL VCALC ( YY(2) )
      CALL FCALC
C
      DO 4000 I = 1, NELM
          FF(I + 1) = AF(I)
4000 CONTINUE
C
      RETURN
      END

*****
** FUNCO2 **
*****

C*****
C** FUNCO2 ***
C*****
      SUBROUTINE FUNCO2 ( X, Y, YP )
C
C** CALCULATE @F/@T FOR ODRK1 AND ODAM (SSL-II) ***
*INCLUDE COMMON
C
      DIMENSION Y(NELM), YP(NELM)
C
      CALL VCALC ( Y )
      CALL FCALC
C
      DO 4000 I = 1, NELM
          YP(I) = AF(I)
4000 CONTINUE
C
      RETURN
      END

```

```
*****
** FUNOUT **
*****
```

```
C*****
C** FUNOUT ***
C*****
      SUBROUTINE  FUNOUT
C
C** LIST OUTPUT OF REACTION FUNCTION F(X,T) ***
*INCLUDE  COMMON
C
      PARAMETER  ( NLL      = 7 )
      PARAMETER  ( NNN      = (NELM - 1)/NLL + 1 )
C
      WRITE(6, 6000)
      IE        = 0
*VOCL LOOP, SCALAR
      DO 4000 I = 1, NNN
          IS      = IE + 1
          IE      = MIN(NELM, IE + NLL)
          WRITE(6, 6100) (J, J = IS, IE)
          WRITE(6, 6200) (AF(J), J = IS, IE)
      4000 CONTINUE
C
      RETURN
C
6000 FORMAT(1H0,3X,'*** REACTION FUNCTION, F(X,T) ***')
6100 FORMAT(1H0,9X,7('F-',I2.2,':',10X))
6200 FORMAT(1H ,5X,1P7D14.4)
      END
```

```
*****
** INPUT **
*****
```

```
C*****
C** INPUT ***
C*****
      SUBROUTINE  INPUT
C
C** INPUT DATA / CHANGE CONSTANTS ***
*INCLUDE  COMMON
C
      CHARACTER*14 CTYP(0:8)
      .      / '<PORSING>', '<M-PORSING>', '<R-K-G>', '<HAMING>'
      .      '<R-K-V>', '<ADAMS>', '<EULER-RBG>',
      .      '<JAC-RBG>', '<M-JAC-RBG>' /
      DIMENSION  AY(NELM)
      NAMELIST  / CONST / AK
C
C.. CONTROL DATA ...
      READ (5, 5000)  CTIT, NDTS, IOPT, ICHK, IFINT, NEPS
      NDTS = MAX(1, MIN(NDTC, NDTS))
```

```

      IOPT  = MAX(0, IOPT)
      IF (IOPT .GT. 8) IOPT = 0
      WRITE(6, 6000) CPGM, CTIT, NDT5
C
      IF (NDTS.LE.0 .OR. NDT5.GT.NDTC) THEN
        WRITE(6, 7000) NDT5, NDTC
        STOP 999
      END IF
      DO 4000 I = 1, NDT5
        READ (5, 5100) TEND(I), TLST(I), DTT(I), TPRM1(I), TPRM2(I)
4000 CONTINUE
C
      WRITE(6, 6100) (I, TEND(I), TLST(I), DTT(I), TPRM1(I), TPRM2(I),
        I = 1, NDT5 )
      IFINT = MAX(1, IFINT)
      WRITE(6, 6200) IOPT, CTYP(IOPT), ICHK, IFINT, NEPS
C
C.. INITIAL CONCENTRATION AND VOLUME ...
C.. IF NOT SPECIFIED, RESTART ASSUMED ...
      READ (5, 5100, END=3000) VOLX, AX
C
C.. ADDITIONAL CONCENTRATION AND VOLUME ...
      READ (5, 5100) VOLY, AY
C
C.. CHANGE REACTION VELOCITY CONSTANT ...
      READ (5, CONST, END=3100)
C
      TIME0 = 0.0D0
      GO TO 3100
C
C.. RESTART INPUT ...
3000 CONTINUE
      VOLX = 0.0D0
      VOLY = 0.0D0
      CALL RSTIN
C
C.. PRINT OUT ...
3100 CONTINUE
      WRITE(6, 6300) 'INITIAL', VOLX
      CALL ELMOUT ( AX )
      IF (VOLY .GT. 0.0D0) THEN
        WRITE(6, 6300) 'ADDITIONAL', VOLY
        CALL ELMOUT ( AY )
        DO 4100 I = 1, NELM
          AX(I) = (VOLX*AX(I) + VOLY*AY(I))/(VOLX + VOLY)
4100 CONTINUE
          VOLX = VOLX + VOLY
          WRITE(6, 6300) 'MODIFIED', VOLX
          CALL ELMOUT ( AX )
        END IF
      XPU0 = AX(1) + AX(2) + AX(3) + AX(4)
      XNPO = AX(7) + AX(8) + AX(9)
      XUO = AX(10) + AX(11)
      IF (ICLK .EQ. 0) THEN
        WRITE(6, 6400) (I, AK(I), I = 1, NREF)

```

```

ELSE
  CALL EQCHK
END IF
C
RETURN
C
5000 FORMAT(A72
.      /BN,5I10)
5100 FORMAT(BN,7F10.0)
C
6000 FORMAT(1H1,'<<< ',A,' >>>',3X,A
.      /1H,'*****'
.      /1H,'*** INPUT DATA LIST ***'
.      /1H,'*****'
.      /1H,2X,'... TIME CONTROL DATA ...'
.      /1H,2X,' CYCLE NO. =',I5)
6100 FORMAT(1H0,2X,18X,'TEND(MIN)      LIST(MIN)      DT (MIN)
.      'PARM-1      PARM-2'
.      /4(1H,2X,' CYCLE',I2,'      =',1P5D13.3))
6200 FORMAT(1H0,2X,' CALC. TYPE =',I5,1X,A
.      /1H,2X,' CHK PRINT =',I5
.      /1H,2X,' FILE INT. =',I5
.      /1H,2X,' EPS MAX P. =',I5)
6300 FORMAT(1H0,2X,'... ',A,' VOLUME AND CONCENTRATION ...'
.      /1H,2X,' VOLUME      =',1PD11.3,' (DM**3)')
6400 FORMAT(1H0,2X,'... REACTION CONSTANT ...'
.      /4(1H,5(2X,'K',I2.2,1PD11.3,,:2X)))
C
7000 FORMAT(1H0,'** ERROR IN INPUT, NPTS :',I5,' IS INVALID,',
.      ' MAX NO. :',I5)
END

```

** JACRBG **

```

C*****
C** JACRBG ***
C*****
SUBROUTINE JACRBG ( PRM, EPS, MXI, ICON )
C
C** SOLVE BY JACOBIAN-ROMBERG METHOD ***
*INCLUDE COMMON
C
DIMENSION VW(NELM), IP(NELM), XF(NELM), XJ(NELM, NELM)
DIMENSION TX(NELM, 20)
C
C<<< INITIAL INTGRATION >>>
C.. FIRST CONDITION, TX(I) GE 0.0 ...
NDV = 1
DO 4400 L = 1, MXI - 1
DO 4000 I = 1, NELM
TX(I, L) = AX(I)
4000 CONTINUE
C

```

```

C..... DIVISION ...
      DTX      = DT/NDV
      DO 4300 K = 1, NDV
        CALL VCALC ( TX(1, L) )
        CALL RVKRXJ
        CALL JCALC
        CALL FCALC
C
*VOCL LOOP,VECTOR
      DO 4150 J = 1, NELM
        DO 4100 I = 1, NELM
          XJ(I, J) = -DTX*AJ(I, J)*PRM
4100      CONTINUE
          XJ(J, J) = XJ(J, J) + 1.0D0
          XF(J)    = AF(J)
4150      CONTINUE
C
      CALL DLAX ( XJ, NELM, NELM, XF, 0.0D0, 1, IS, VW, IP,
                ICON )
      IF (ICON .NE. 0) THEN
        WRITE(6, *) '*** ERROR IN LAX, ICON = ', ICON
        STOP 999
      END IF
C
*VOCL LOOP,VECTOR
      DO 4200 I = 1, NELM
        TX(I, L) = TX(I, L) + DTX*XF(I)
        IF (TX(I, L) .LT. 0.0D0) GO TO 3000
        IF (TX(I, L) .GT. 1.0D2) GO TO 3000
4200      CONTINUE
4300      CONTINUE
        LS      = L
        GO TO 3100
3000      CONTINUE
        NDV     = 2*NDV
4400      CONTINUE
        STOP 'STOP AT JACRBG, FIRST CONDITION NOT CONVERGED.'
C
C** DIVISIONAL INTEGRATION ***
3100      CONTINUE
        DO 4800 L = LS, MXI - 1
          NDV     = 2*NDV
C
C..... INITIAL SET ...
        DO 4500 I = 1, NELM
          TX(I, L + 1) = AX(I)
4500      CONTINUE
C
C..... DIVISION ...
      DTX      = DT/NDV
      DO 4700 K = 1, NDV
        CALL VCALC ( TX(1, L + 1) )
        CALL RVKRXJ
        CALL JCALC
        CALL FCALC

```



```

C
*VOCL LOOP,VECTOR
      DO 4620 J = 1, NELM
      DO 4600 I = 1, NELM
        XJ(I, J) = -DTX*AJ(I, J)*PRM
4600      CONTINUE
        XJ(J, J) = XJ(J, J) + 1.0DO
        XF(J)    = AF(J)
4620      CONTINUE
C
      CALL DLAX ( XJ, NELM, NELM, XF, 0.0DO, 1, IS, VW, IP,
                ICON )
      IF (ICON .NE. 0) THEN
        WRITE(6, *) '*** ERROR IN LAX, ICON = ', ICON
        STOP 999
      END IF
C
*VOCL LOOP,VECTOR
      DO 4650 I = 1, NELM
        TX(I, L + 1) = TX(I, L + 1) + DTX*XF(I)
4650      CONTINUE
4700      CONTINUE
C
C..... FIRST CONVERGE CHECK ...
      DO 4720 I = 1, NELM
        ERR = ABS(TX(I, L + 1) - TX(I, L))
        IF (ERR .GT. EPS*ABS(TX(I, L))) GO TO 3200
4720      CONTINUE
        ICON = -L
        ICNV = L + 1
        GO TO 3500
C
C..... "ROMBERG" INTERPOLATION AND CONVERGE CHECK ...
3200      CONTINUE
        M2 = 1
        DO 4780 K = L, LS, -1
          M2 = 2*M2
          DO 4740 I = 1, NELM
            TX(I, K) = (M2*TX(I, K + 1) - TX(I, K))/(M2 - 1)
4740          CONTINUE
          IF (L .LE. LS) GO TO 3400
C
          DO 4760 I = 1, NELM
            ERR = ABS(TX(I, K) - TX(I, K - 1))
            IF (ERR .GT. EPS*ABS(TX(I, K - 1))) GO TO 3300
4760          CONTINUE
C
C..... CONVERGED ...
          ICON = -L
          ICNV = K
          GO TO 3500
3300          CONTINUE
4780          CONTINUE
C
C..... NOT CONVERGED ...

```

```

3400    CONTINUE
4800    CONTINUE
      ICON  = 10
      ICNV  = 1
C
C** CONVERGED ***
3500    CONTINUE
      DO 4900 I = 1, NELM
          AX(I) = TX(I, ICNV)
4900    CONTINUE
C
      RETURN
      END

```

```

*****
** JCALC **
*****

```

```

C*****
C** JCALC ***
C*****
      SUBROUTINE JCALC
C
C** CALCULATE JACOBIAN J(N) ***
*INCLUDE COMMON
C
      DO 4200 J = 1, NELM
      DO 4200 I = 1, NELM
          AJ(I, J) = 0.0D0
          DO 4000 K = 1, NREF
              AJ(I, J) = AJ(I, J) + AC(I, K)*RVRX(K, J)
4000    CONTINUE
4200    CONTINUE
C
      RETURN
      END

```

```

*****
** MAIN **
*****

```

```

C*****
C** ACTINIDE REACTION ANALYSIS IN AQUA
C*****
C
*INCLUDE COMMON
C
C** DATA INPUT ***
      CALL INPUT
          ICYC = 0
          IDTS = 1
          TIME = TIME0
          DT   = DTT(1)
C
C** INITIAL OUTPUT ***

```

```

*VOCL LOOP, SCALAR
  DO 4000 I = 1, NDT5
    IF (TLST(I) .GT. 0.0D0) THEN
      WRITE(6, '(1H1)')
      IF (ICLK .GE. 2) THEN
        CALL VCALL ( AX )
        CALL FCALC
      END IF
      CALL OUTLST
      GO TO 3000
    END IF
  4000 CONTINUE
C
  3000 CONTINUE
  CALL OUTFIL
C
C** SOLUVE REACTION EQUATION ***
  CALL SOLVE
C
  CALL RSTOUT
C
  STOP
  END

```

```

*****
OUTFIL **
*****

```

```

C*****
C** OUTFIL ***
C*****
  SUBROUTINE OUTFIL
C
C** FILE OUTPUT ***
*INCLUDE COMMON
C
  IF (ICYC .LE. 0) THEN
    WRITE(30) CTIT
    WRITE(30) NELM, NREF
    WRITE(30) CELM
  END IF
C
  IF (MOD(ICYC, IFINT).EQ.0 .OR. TEND(IDTS)-TIME.LE.1.0D-10) THEN
    WRITE(30) TIME, AX, AF, AV
  END IF
C
  RETURN
  END

```

```

*****
OUTHMG **
*****

```

```

C*****
C** OUTHMG ***

```

```

C*****
      SUBROUTINE  OUTHMG  ( YY, FF, N1, HH, IS )
C
C** OUTPUT FROM HAMNG (SSL-II) OF 1 STEP CALCULATED ***
*INCLUDE  COMMON
C
      COMMON / COMHMG / HHO,  TL
C
      DIMENSION  YY(N1),  FF(N1)
C
      DT      = HH
      TIME    = YY(1)
      DO 4000 I = 1, NELM
         AX(I) = YY(I + 1)
      4000 CONTINUE
C
      ICYC    = ICYC + 1
      IF (TL - TIME .LE. 1.0D-8)  THEN
         CALL OUTLST
         TL    = TIME + TLST(IDTS)*1.00000001D0
         TL    = TL - MOD(TL, TLST(IDTS))
      END IF
      CALL OUTFIL
C
      RETURN
      END

```

```

*****
** OUTLST **
*****

```

```

C*****
C** OUTLST ***
C*****
      SUBROUTINE  OUTLST
C
C** LIST OUTPUT ***
*INCLUDE  COMMON
C
      CHARACTER*4  CTH(NDTC) / '1-ST', '2-ND', '3-RD', '4-TH', '5-TH' /
C
      WRITE(6, 6000)  TIME, ICYC, CTH(IDTS), DT
      CALL ELMOUT ( AX )
      IF (ICLK .GE. 2)  THEN
         CALL CXTOUT
         CALL FUNOUT
         CALL VELOUT
      END IF
C
      RETURN
C
6000 FORMAT(1H0,83('*')
.          /1H , '** TIME',1PD12.4,' (MIN)',111,' CYCLES IN ',
.          A,' LOOP,  DT :',D11.3,' (MIN) **'
.          /1H ,83('*'))

```

```
*****
* RSTIN  **
*****
```

```
C*****
C** RSTIN  ***
C*****
      SUBROUTINE  RSTIN
C
C** RESTART FILE INPUT ***
*INCLUDE  COMMON
C
      WRITE(6, 6000)
      READ (50)  NELMX, NREFX, TIMEO
      IF (NELMX.NE.NELM .OR. NREFX.NE.NREF)  THEN
        WRITE(6, 6100)
        STOP 999
      END IF
      READ (50)  AX,  AK
      RETURN
C
      6000 FORMAT(1H0,2X,'... RESTART ASSUMED ...')
      6100 FORMAT(1H , '*** ERROR IN RSTIN, NELM OR NREF UNMATCHED')
      END
```

```
*****
* RSTOUT **
*****
```

```
C*****
C** RSTOUT ***
C*****
      SUBROUTINE  RSTOUT
C
C** RESTART FILE OUTPUT ***
*INCLUDE  COMMON
C
      WRITE(60)  NELM, NREF, TIME
      WRITE(60)  AX,  AK
      RETURN
      END
```

```
*****
* RVKRXJ **
*****
```

```
C*****
C** RVKRXJ ***
C*****
      SUBROUTINE  RVKRXJ
C
C** CALCULATE @V(K)/@X(J) ***
*INCLUDE  COMMON
```

```

C
C.. K = 1 ...
RVRX( 1, 3) = AK(1)*AX(11)/(AX(5) + 0.05D0)**2
RVRX( 1, 5) = -2.0D0*AK(1)*AX(3)*AX(11)/(AX(5) + 0.05D0)**3
RVRX( 1, 11) = AK(1)*AX(3)/(AX(5) + 0.05D0)**2
C
C.. K = 2 ...
RV2RCN = -2.0D0*AK(2)*AX(3)**2*AX(12)**2
          / ( AX(4)**2*AX(5)**4*(CN + 0.35D0)**3 )
RVRX( 2, 1) = 2.0D0*RV2RCN
RVRX( 2, 2) =          RV2RCN
RVRX( 2, 3) = 2.0D0*AK(2)*AX(3)*AX(12)**2
          / ( AX(4)**2*AX(5)**4*(CN + 0.35D0)**2 )
          + 4.0D0*RV2RCN
RVRX( 2, 4) = -2.0D0*AK(2)*AX(3)**2*AX(12)**2
          / ( AX(4)**3*AX(5)**4*(CN + 0.35D0)**2 )
          + 3.0D0*RV2RCN
RVRX( 2, 5) = -4.0D0*AK(2)*AX(3)**2*AX(12)**2
          / ( AX(4)**2*AX(5)**5*(CN + 0.35D0)**2 )
          +          RV2RCN
RVRX( 2, 7) = 2.0D0*RV2RCN
RVRX( 2, 8) =          RV2RCN
RVRX( 2, 9) = 4.0D0*RV2RCN
RVRX( 2, 10) = 2.0D0*RV2RCN
RVRX( 2, 11) = 4.0D0*RV2RCN
RVRX( 2, 12) = 2.0D0*AK(2)*AX(3)**2*AX(12)
          / ( AX(4)**2*AX(5)**4*(CN + 0.35D0)**2 )
          +          RV2RCN
RVRX( 2, 13) =          RV2RCN
C
C.. K = 3 ...
RVRX( 3, 3) = AK(3)*AX(13)/(AX(5) + 0.35D0)
RVRX( 3, 5) = -AK(3)*AX(3)*AX(13)/(AX(5) + 0.35D0)**2
RVRX( 3, 13) = AK(3)*AX(3)/(AX(5) + 0.35D0)
C
C.. K = 4 ...
RV4RCN = 0.4D0*AK(4)*AX(4)*SQRT(AX(6)*AX(5))/CN**0.6D0
RVRX( 4, 1) = 2.0D0*RV4RCN
RVRX( 4, 2) =          RV4RCN
RVRX( 4, 3) = 4.0D0*RV4RCN
RVRX( 4, 4) = AK(4)*SQRT(AX(6)*AX(5))*CN04
          + 3.0D0*RV4RCN
RVRX( 4, 5) = 0.5D0*AK(4)*AX(4)*SQRT(AX(6)/AX(5))*CN04
          +          RV4RCN
RVRX( 4, 6) = 0.5D0*AK(4)*AX(4)*SQRT(AX(5)/AX(6))*CN04
RVRX( 4, 7) = 2.0D0*RV4RCN
RVRX( 4, 8) =          RV4RCN
RVRX( 4, 9) = 4.0D0*RV4RCN
RVRX( 4, 10) = 2.0D0*RV4RCN
RVRX( 4, 11) = 4.0D0*RV4RCN
RVRX( 4, 12) =          RV4RCN
RVRX( 4, 13) =          RV4RCN
C
C.. K = 5 ...
RVRX( 5, 5) = AK(5)*AX(6)*AX(13)

```

```

RVRX( 5, 6) = AK(5)*AX(13)*AX(5)
RVRX( 5, 13) = AK(5)*AX(6)*AX(5)
C
C.. K = 6 ...
RV6RCN = 12.5D0*AK(6)*AX(9)*AX(7)
RVRX( 6, 1) = 2.0D0*RV6RCN
RVRX( 6, 2) = RV6RCN
RVRX( 6, 3) = 4.0D0*RV6RCN
RVRX( 6, 4) = 3.0D0*RV6RCN
RVRX( 6, 5) = RV6RCN
RVRX( 6, 7) = AK(6)*AX(9)*(2.16D0 + 12.5D0*CN)
+ 2.0D0*RV6RCN
RVRX( 6, 8) = RV6RCN
RVRX( 6, 9) = AK(6)*AX(7)*(2.16D0 + 12.5D0*CN)
+ 4.0D0*RV6RCN
RVRX( 6, 10) = 2.0D0*RV6RCN
RVRX( 6, 11) = 4.0D0*RV6RCN
RVRX( 6, 12) = RV6RCN
RVRX( 6, 13) = RV6RCN
C
C.. K = 7 ...
RVRX( 7, 3) = AK(7)*AX(9)/AX(5)**4
RVRX( 7, 5) = -4.0D0*AK(7)*AX(9)*AX(3)/AX(5)**5
RVRX( 7, 9) = AK(7)*AX(3)/AX(5)**4
C
C.. K = 8 ...
RVRX( 8, 5) = -1.3D0*AK(8)*AX(7)*AX(13)/AX(5)**2.3D0
RVRX( 8, 7) = AK(8)*AX(13)/AX(5)**1.3D0
RVRX( 8, 13) = AK(8)*AX(7)/AX(5)**1.3D0
C
C.. K = 9 ...
RVRX( 9, 7) = AK(9)*AX(11)
RVRX( 9, 11) = AK(9)*AX(7)
C
C.. K = 10 ...
RVRX(10, 5) = AK(10)*AX(8)*AX(11)*(-3.2D0/AX(5)**3 + 1.42D0)
RVRX(10, 8) = AK(10)*AX(11)*(1.6D0/AX(5)**2 + 1.42D0*AX(5))
RVRX(10, 11) = AK(10)*AX(8)*(1.6D0/AX(5)**2 + 1.42D0*AX(5))
C
C.. K = 11 ...
RVRX(11, 5) = -AK(11)*AX(7)*AX(12)/AX(5)**2
RVRX(11, 7) = AK(11)*AX(12)/AX(5)
RVRX(11, 12) = AK(11)*AX(7)/AX(5)
C
C.. K = 12 ...
RVRX(12, 2) = AK(12)*AX(3)*AX(5)/(AX(5) + 0.054D0)
RVRX(12, 3) = AK(12)*AX(2)*AX(5)/(AX(5) + 0.054D0)
RVRX(12, 5) = AK(12)*AX(2)*AX(3)/(AX(5) + 0.054D0)**2
C
C.. K = 13 ...
RVRX(13, 2) = AK(13)*AX(4)*AX(5)
RVRX(13, 4) = AK(13)*AX(2)*AX(5)
RVRX(13, 5) = AK(13)*AX(2)*AX(4)
C
C.. K = 14 ...

```

```

RVRX(14, 1) = AK(14)*AX(6)/AX(5)
RVRX(14, 5) = -AK(14)*AX(1)*AX(6)/AX(5)**2
RVRX(14, 6) = AK(14)*AX(1)/AX(5)
C
C.. K = 15 ...
RVRX(15, 1) = AK(15)*AX(4)
RVRX(15, 4) = AK(15)*AX(1)
C
C.. K = 16 ...
RV16RC = 0.4D0*AK(16)*AX(2)*SQRT(AX(6))*(AX(5)/CN)**0.6D0
RVRX(16, 1) = 2.0D0*RV16RC
RVRX(16, 2) = AK(16)*SQRT(AX(6))*CN**0.4D0*AX(5)**0.6D0
      + RV16RC
RVRX(16, 3) = 4.0D0*RV16RC
RVRX(16, 4) = 3.0D0*RV16RC
RVRX(16, 5) = 0.6D0*AK(16)*AX(2)*SQRT(AX(6))*CN**0.4D0
      + / AX(5)**0.4D0
      + RV16RC
RVRX(16, 6) = 0.5D0*AK(16)*AX(2)*CN**0.4D0*AX(5)**0.6D0
      + / SQRT(AX(6))
RVRX(16, 7) = 2.0D0*RV16RC
RVRX(16, 8) = RV16RC
RVRX(16, 9) = 4.0D0*RV16RC
RVRX(16, 10) = 2.0D0*RV16RC
RVRX(16, 11) = 4.0D0*RV16RC
RVRX(16, 12) = RV16RC
RVRX(16, 13) = RV16RC
C
C.. K = 17 ...
RVRX(17, 5) = AK(17)
C
C.. K = 18 ...
RVRX(18, 6) = AK(18)
C
C.. K = 19 ...
RVRX(19, 5) = 2.0D0*AK(19)*AX(8)**2*AX(5)
RVRX(19, 8) = 2.0D0*AK(19)*AX(8)*AX(5)**2
C
C.. K = 20 ...
RV20RC = 2.0D0*AK(20)*AX(5)**1.3D0*CN*AX(6)*AX(8)/(AX(6)+AX(8))
RVRX(20, 1) = 2.0D0*RV20RC
RVRX(20, 2) = RV20RC
RVRX(20, 3) = 4.0D0*RV20RC
RVRX(20, 4) = 3.0D0*RV20RC
RVRX(20, 5) = 1.3D0*AK(20)*AX(5)**0.2D0*CN**2*AX(8)/(AX(6)+AX(8))
> + RV20RC
RVRX(20, 6) = AK(20)*AX(5)**1.3D0*CN*2*AX(8)**2/(AX(6)+AX(8))**2
RVRX(20, 7) = 2.0D0*RV20RC
RVRX(20, 8) = AK(20)*AX(5)**1.3D0*CN*2*AX(6)**2/(AX(6)+AX(8))**2
> + RV20RC
RVRX(20, 9) = 4.0D0*RV20RC
RVRX(20, 10) = 2.0D0*RV20RC
RVRX(20, 11) = 4.0D0*RV20RC
RVRX(20, 12) = RV20RC
RVRX(20, 13) = RV20RC

```



```

C
C.. K = 21 ...
RVRX(21, 6) = -5.0D-1*AK(21)*AX(8)*AX(9)/AX(6)**1.5D0
RVRX(21, 8) = AK(21)*AX(9)/AX(6)**0.5D0
RVRX(21, 9) = AK(21)*AX(8)/AX(6)**0.5D0

C
C.. K = 22 ...
RVRX(22, 4) = AK(22)*AX(7)*(2.13D3 + 1.86D2/AX(5))
RVRX(22, 7) = AK(22)*AX(4)*(2.13D3 + 1.86D2/AX(5))
RVRX(22, 5) = -1.86D2*AK(22)*AX(4)*AX(7)/AX(5)**2D0

C
C.. K = 23 ...
RVRX(23, 4) = AK(23)*AX(8)/AX(5)**1.3D0
RVRX(23, 5) = -1.3D0*AK(23)*AX(4)*AX(8)/AX(5)**2.3D0
RVRX(23, 8) = AK(23)*AX(4)/AX(5)**1.3D0

C
C.. K = 24 ...
RVRX(24, 1) = AK(24)*AX(13)/AX(5)
RVRX(24, 5) = -1.0D0*AK(24)*AX(1)*AX(13)/AX(5)**2.0D0
RVRX(24, 13) = AK(24)*AX(1)/AX(5)

C
C.. K = 25 ...
RV25RC = -5.4D-1*AK(25)*AX(3)**2.0D0/(AX(5)**2.2D0*CN**1.54D0)
RVRX(25, 1) = 2.0D0*RV25RC
RVRX(25, 2) = RV25RC
RVRX(25, 3) = 2.0D0*AK(25)*AX(3)/(AX(5)**2.2D0*CN**0.54D0)
+ 4.0D0*RV25RC
RVRX(25, 4) = 3.0D0*RV25RC
RVRX(25, 5) = -2.2D0*AK(25)*AX(3)**2.0D0
/ (AX(5)**3.2D0*CN**0.54D0)
+ RV25RC
RVRX(25, 7) = 2.0D0*RV25RC
RVRX(25, 8) = RV25RC
RVRX(25, 9) = 4.0D0*RV25RC
RVRX(25, 10) = 2.0D0*RV25RC
RVRX(25, 11) = 4.0D0*RV25RC
RVRX(25, 12) = RV25RC
RVRX(25, 13) = RV25RC

C
C.. K = 26 ...
RVRX(26, 2) = 2.0D0*AK(26)*AX(2)*AX(5)
RVRX(26, 5) = AK(26)*AX(2)**2.0D0

C
C.. K = 27 ...
RVRX(27, 5) = AK(27)*AX(6)*AX(12)
RVRX(26, 6) = AK(27)*AX(5)*AX(12)
RVRX(26, 12) = AK(27)*AX(5)*AX(6)
RETURN
END

```

```

*****
**
*****

```

C*****

```

C** SOLADM ***
C*****
      SUBROUTINE  SOLADM
C
C** SOLVE BY ADAMS METHOD ***
*INCLUDE  COMMON
C
      PARAMETER  ( NN = 21*NELM + 110 )
      DIMENSION  VW(NN),  IVW(11)
C
      EXTERNAL   FUNCO2
C
      TIME      = TIME0
      ISW       = 10
C
C** TIME LOOP ***
*VOCL LOOP,SCALAR
      DO 4500 IDTS = 1, NDTS
          EPSA   = TPRM1(IDTS)
          EPSR   = TPRM2(IDTS)
          CALL TIMSET ( TIME,  TEND(IDTS), TLST(IDTS), DTT(IDTS),
                        TL,    DT,      NCYC1,  NCYC2  )
C
C..... DT LOOP ...
          DO 4200 I = NCYC1, NCYC2
              XEND  = TIME + DT
              DO 4000 WHILE (.TRUE.)
1000          CONTINUE
              CALL DODAM ( TIME,  AX,      FUNCO2, NELM,  XEND,  ISW,
                        EPSA,  EPSR,  VW,      IVW,  ICON  )
C
              IF (ICON .GE. 100) THEN
                  WRITE(6, 7000)  ICON,  TIME,  EPSA,  EPSR
                  IF (ICON .EQ. 100) GO TO 1000
                  IF (ICON .EQ. 200) GO TO 1000
                  IF (ICON .EQ. 10000) GO TO 1000
                  STOP 999
              END IF
C
              ICYC  = ICYC + 1
              IF (TL - TIME .LE. 1.0D-8) THEN
                  CALL OUTLST
                  TL    = TIME + TLST(IDTS)*1.00000001D0
                  TL    = TL - MOD(TL, TLST(IDTS))
              END IF
              CALL OUTFIL
              IF (XEND - TIME .LE. 1.0D-8) GO TO 3100
4000          CONTINUE
3100          CONTINUE
              DT      = DTT(IDTS)
C
4200          CONTINUE
4500          CONTINUE
C
      RETURN

```

```

C
7000 FORMAT(1H0,'** ERROR IN SOLADM, ICON =',I8
.         /1H /'   TIME :',1PD13.4
.         /1H /'   EPSA :',   D13.4
.         /1H /'   EPSR :',   D13.4)
END

```

```

*****
SOLEUL  **
*****

```

```

C*****
C** SOLEUL ***
C*****
      SUBROUTINE  SOLEUL
C
C** SOLVE BY EULER-ROMBERG METHODD ***
*INCLUDE  COMMON
C
      PARAMETER ( MXIO  = 31 )
      DIMENSION  FX(NELM), TX(NELM, MXIO)
C
      EXTERNAL  FUNCO2
C
C** TIME LOOP ***
*VOCL LOOP, SCALAR
      DO 4500 IDTS = 1, NDT5
C
          CALL TIMSET ( TIME,      TEND(IDTS), TLST(IDTS), DTT(IDTS),
.                   TL,      DT,      NCYC1,      NCYC2      )
C
          EPS      = TPRM1(IDTS)
          MXI      = TPRM2(IDTS)
          MXI      = MIN(MXI, MXIO)
          IF (EPS .LE. 0.0D0) EPS      = 1.0D-6
          IF (MXI .LE. 0)      MXI      = 16
C
C..... DT LOOP ...
          DO 4300 L = NCYC1, NCYC2
              CALL EULRBG ( NELM, FX, AX, TX, FUNCO2, TIME, DT,
.                   EPS, MXI, ICON )
C
              IF (ICON .GT. 0) THEN
                  WRITE(6, 6000)  TIME, ICON
              END IF
C
              ICYC  = ICYC + 1
              IF (TL - TIME .LE. 1.0D-8) THEN
                  CALL OUTLST
                  TL      = TL + TLST(IDTS)*1.00000001D0
                  TL      = TL - MOD(TL, TLST(IDTS))
              END IF
              CALL OUTFIL
              DT      = DTT(IDTS)
C

```

```

4300 CONTINUE
4500 CONTINUE
C
RETURN
C
6000 FORMAT(1H0,'*** ERROR IN SOLEUL AT TIME :',1PD13.4,
', ICON =',16)
END

```

```

*****
SOLHMG **
*****

```

```

C*****
C** SOLHMG ***
C*****
SUBROUTINE SOLHMG
C
C** SOLVE BY HAMMING METHOD ***
*INCLUDE COMMON
C
COMMON / COMHMG / HHO, TL
C
PARAMETER ( N1 = NELM + 1, NN = 18*N1 )
DIMENSION Y(N1), VW(NN)
C
EXTERNAL FUNC01, OUTHMG
C
C** TIME LOOP ***
*VOCL LOOP, SCALAR
DO 4500 IDTS = 1, NDT5
DT = 0.000
CALL TIMSET ( TIME, TEND(IDTS), TLST(IDTS), DTT(IDTS),
TL, DT, NCYC1, NCYC2 )
C
HH = TPRM1(IDTS)
EPS = TPRM2(IDTS)
XEND = TEND(IDTS)
1000 CONTINUE
H = HH
Y(1) = TIME
DO 4000 J = 1, NELM
Y(J + 1) = AX(J)
4000 CONTINUE
C
CALL DHAMNG ( Y, N1, H, XEND, EPS, FUNC01, OUTHMG,
VW, ICON )
C
IF (ICON .GT. 10000) THEN
WRITE(6, 6000) ICON, ICYC
WRITE(6, 6100) TIME, XEND, H, EPS, N1
IF (ICON .EQ. 20000) THEN
HH = HH/2.000
EPS = 2.000*EPS
GO TO 1000

```

```

      END IF
      STOP 999
    END IF
  C
  4500 CONTINUE
  C
      RETURN
  C
  6000 FORMAT(1H0,'** ERROR IN SOLHMG, ICON =',I8,I8,' CYCLES')
  6100 FORMAT(1H /' TIME :',1PD13.4
    . /1H /' XEND :',D13.4
    . /1H /' H :',D13.4
    . /1H /' EPS :',D13.4
    . /1H /' N1 :',I4)
      END

*****
SOLJAC **
*****

C*****
C** SOLJAC ***
C*****
      SUBROUTINE SOLJAC
  C
  C** SOLVE BY JACOBIAN METHOD ***
  *INCLUDE COMMON
  C
      DIMENSION VW(NELM), IP(NELM), XF(NELM), XJ(NELM, NELM)
  C
  C.. IOPT = 0 IS PORSING, = 1 IS M-PORSING ...
  C.. DTT = 0 FOR VARIABLE DT ...
      IF (IOPT .EQ.1) THEN
        PRM = 0.5D0
      ELSE
        PRM = 1.0D0
      END IF
  C
  C** TIME LOOP ***
  *VOCL LOOP, SCALAR
      DO 4500 IDTS = 1, NDT5
  C..... EPS FOR VARIABLE DT ...
      IF (DTT(IDTS) .GT. 0.0D0) THEN
        EPS = 0.0D0
        EPS2 = 0.0D0
      ELSE
        IF (TPRM1(IDTS) .LE. 0.0D0) THEN
          EPS = 1.0D0
        ELSE
          EPS = TPRM1(IDTS)
        END IF
        IF (TPRM2(IDTS) .LE. 0.0D0) THEN
          EPS2 = 1.0D0
        ELSE
          EPS2 = TPRM2(IDTS)
        END IF
      END IF
    END DO
  END SUBROUTINE SOLJAC

```

```

        END IF
      END IF
C
      CALL TIMSET ( TIME, TEND(IDTS), TLST(IDTS), DTT(IDTS),
                  TL, DT, NCYC1, NCYC2 )
C
C..... DT LOOP ...
      DO 4300 L = NCYC1, NCYC2
C
          CALL VCALC ( AX )
          CALL RVKRXJ
          CALL JCALC
          CALL FCALC
          IF (EPS .GT. 0.000) THEN
              CALL DEFDT ( EPS, EPS2 )
          END IF
          DT = MIN(TEND(IDTS) - TIME, DT)
C
C***** MATRIX OPERATION (I - DT*AJ) OR (I - DT*AJ/2) ***
      1000 CONTINUE
*VOCL LOOP,VECTOR
      DO 4100 J = 1, NELM
          DO 4000 I = 1, NELM
              XJ(I, J) = -DT*AJ(I, J)*PRM
          4000 CONTINUE
              XJ(J, J) = XJ(J, J) + 1.000
              XF(J) = AF(J)
          4100 CONTINUE
C
C** SOLVE SIMULTANEOUS EQUATIONS USING SSL2 ***
C.. "A"*"X" = "B",
C.. "A" = (I - DT*AJ) OR (I - DT*AJ/2) <== "XJ"
C.. "X" = (X(N + 1) - X(N))/DT ==> "XF"
C.. "B" = F(X) <== "XF"
          CALL DLAX ( XJ, NELM, NELM, XF, 0.000, 1, IS, VW, IP,
                  ICON )
          IF (ICON .NE. 0) THEN
              WRITE(6, *) '*** ERROR IN LAX, ICON = ', ICON
              STOP 999
          END IF
C
C.. NEW CONC. MINUS CHECK (IF VARIABLE DT) ...
          IF (EPS .GT. 0.000) THEN
              DO 4200 I = 1, NELM
                  IF (AX(I) + DT*XF(I) .LT. 0.000) THEN
                      DT = DT/2.000
                      GO TO 1000
                  END IF
              4200 CONTINUE
          END IF
C
C** NEW CONCENTRATION ***
*VOCL LOOP,VECTOR
      DO 4250 I = 1, NELM
          AX(I) = AX(I) + DT*XF(I)

```

```

4250      CONTINUE
C
      ICYC   = ICYC + 1
      TIME  = TIME + DT
      IF (TL - TIME .LE. 1.0D-8) THEN
          CALL OUTLST
          TL   = TL + TLST(IDTS)*1.00000001D0
          TL   = TL - MOD(TL, TLST(IDTS))
      END IF
      CALL OUTFIL
C
      IF (TEND(IDTS) - TIME .LE. 1.0D-10) GO TO 3100
      IF (DT .GT. DTMAX) THEN
          DTMAX = DT
          IDTMX = ICYC
      END IF
      IF (DT .LT. DTMIN) THEN
          DTMIN = DT
          IDTMN = ICYC
      END IF
C
      DT      = DTT(IDTS)
4300      CONTINUE
3100      CONTINUE
4500      CONTINUE
C
      WRITE(6, 6000)  CPGM, DTMAX, IDTMX, DTMIN, IDTMN
      RETURN
C
6000      FORMAT(1H0,'*** ',A,' ***  CALCULATION ENDED BY (M-)PORSING'
.          /1H0,'  MAXIMUM DT IS',1PD12.3,' AT CYCLE NO.',I7
.          /1H ,'  MINIMUM DT IS',  D12.3,' AT CYCLE NO.',I7)
C
7000      FORMAT(1H0,'** ERROR IN SOLADM, ICON =',I8
.          /1H ,'  TIME  :',1PD13.4
.          /1H ,'  EPSA  :',  D13.4
.          /1H ,'  EPSR  :',  D13.4)
      END

```

```

*****
SOLJRB **
*****

```

```

C*****
C** SOLJRB ***
C*****
      SUBROUTINE  SOLJRB
C
C** SOLVE BY JACOBIAN-ROMBERG METHOD ***
*INCLUDE  COMMON
C
C.. IOPT = 7 IS PORSING, = 8 IS M-PORSING ...
      IF (IOPT .EQ. 8) THEN
          PRM   = 0.5D0
      ELSE

```

```

        PRM      = 1.000
    END IF
    IWF      = NEPS
    IF (IWF .LE. 0)   IWF = 999999999
C
C** TIME LOOP ***
*VOCL LOOP, SCALAR
DO 4500 IDTS = 1, NDTs
    IF (TPRM1(IDTS) .LE. 0.000)   THEN
        EPS      = 1.0D-6
    ELSE
        EPS      = TPRM1(IDTS)
    END IF
    IF (TPRM2(IDTS) .LE. 0.000)   THEN
        MXI      = 16
    ELSE
        MXI      = TPRM2(IDTS)
        MXI      = MIN(MXI, 20)
    END IF
C
        CALL TIMSET ( TIME,      TEND(IDTS), TLST(IDTS), DTT(IDTS),
                     TL,        DT,        NCYC1,      NCYC2      )
C
C..... DT LOOP ...
DO 4300 L = NCYC1, NCYC2
C
        CALL JACRBG ( PRM,      EPS,      MXI,      ICON )
        IF (ICON.GT.0 .AND. IWF.GT.0)   THEN
            IWF      = IWF - 1
            WRITE(6, 7000)
        END IF
C
        ICYC      = ICYC + 1
        TIME      = TIME + DT
        IF (TL - TIME .LE. 1.0D-8)   THEN
            CALL OUTLST
            TL      = TL + TLST(IDTS)*1.00000001D0
            TL      = TL - MOD(TL, TLST(IDTS))
        END IF
        CALL OUTFIL
C
        IF (TEND(IDTS) - TIME .LE. 1.0D-10)   GO TO 3100
        IF (DT .GT. DTMAX)   THEN
            DTMAX    = DT
            IDTMX    = ICYC
        END IF
        IF (DT .LT. DTMIN)   THEN
            DTMIN    = DT
            IDTMN    = ICYC
        END IF
C
        DT        = DTT(IDTS)
4300    CONTINUE
3100    CONTINUE
4500    CONTINUE

```



```

C      WRITE(6, 6000)  CPGM, DTMAX, IDTMX, DTMIN, IDTMN
      RETURN
C
6000  FORMAT(1H0,'*** ',A,' ***  CALCULATION ENDED BY (M)-J.RB'
        /1H0,'  MAXIMUM DT IS',1PD12.3,'  AT CYCLE NO.',I7
        /1H ,'  MINIMUM DT IS',  D12.3,'  AT CYCLE NO.',I7)
C
7000  FORMAT(1H ,'*** ERROR IN SOLJRB, NOT CONVERGED ***')
      END

```

```

*****
SOLRKG  **
*****

```

```

C*****
C** SOLRKG ***
C*****
      SUBROUTINE  SOLRKG
C
C** SOLVE BY RUNGE-KUTTA-GILL METHOD ***
*INCLUDE  COMMON
C
      PARAMETER  ( K = NELM + 1 )
      DIMENSION  Y(K, 2), F(K, 2), VW(K)
C
      EXTERNAL  FUNCO1
C
C** TIME LOOP ***
*VOCL LOOP, SCALAR
      DO 4500 IDTS = 1, NDT5
C
          CALL TIMSET ( TIME, TEND(IDTS), TLST(IDTS), DTT(IDTS),
                     TL, DT, NCCYC1, NCCYC2 )
C
C..... DT LOOP ...
          DO 4300 L = NCCYC1, NCCYC2
              Y(1, 1) = TIME
              DO 4000 I = 1, NELM
                  Y(I + 1, 1) = AX(I)
          4000  CONTINUE
C
              CALL DRKG  ( Y, F, K, K, DT, 2, FUNCO1, VW, ICON )
C
              IF (ICON .NE. 0) THEN
                  WRITE(6, 6000)  TIME, ICON
                  STOP 999
              END IF
C
              DO 4100 I = 1, NELM
                  AX(I) = Y(I + 1, 2)
          4100  CONTINUE
C
              ICYC  = ICYC + 1
              TIME  = TIME + DT

```

```

        IF (TL - TIME .LE. 1.0D-8) THEN
            CALL OUTLST
            TL      = TL + TLST(IDTS)*1.00000001D0
            TL      = TL - MOD(TL, TLST(IDTS))
        END IF
        CALL OUTFIL
        DT      = DTT(IDTS)
C
C 4300 CONTINUE
C 4500 CONTINUE
C
C RETURN
C
C 6000 FORMAT(1H0,'*** ERROR IN SOLRKG AT TIME :',1PD13.4,
             ', ICON =',16)
C
C END

*****
* SOLRKV **
*****

C*****
C** SOLRKV ***
C*****
      SUBROUTINE SOLRKV
C
C** SOLVE BY RUNGE-KUTTA-VERNER METHOD ***
*INCLUDE COMMON
C
      PARAMETER ( NN = 9*NELM + 40 )
      DIMENSION VW(NN), IVW(5)
      EXTERNAL FUNCO2
C
      TIME = TIME0
      ISW = 10
C
C** TIME LOOP ***
*VOCL LOOP, SCALAR
      DO 4500 IDTS = 1, NDT5
          EPSA = TPRM1(IDTS)
          EPSR = TPRM2(IDTS)
          CALL TIMSET ( TIME, TEND(IDTS), TLST(IDTS), DTT(IDTS),
                     TL, DT, NCYC1, NCYC2 )
C
C..... DT LOOP ...
      DO 4200 I = NCYC1, NCYC2
          XEND = TIME + DT
          DO 4000 WHILE (.TRUE.)
1000          CONTINUE
              CALL DODRK1 ( TIME, AX, FUNCO2, NELM, XEND, ISW,
                          EPSA, EPSR, VW, IVW, ICON )
C
C
          IF (ICON .GE. 10000) THEN
              WRITE(6, 7000) ICON, TIME, EPSA, EPSR
              IF (ICON .EQ. 10000) GO TO 1000
          
```

```

                IF (ICON .EQ. 11000) GO TO 1000
                STOP 999
            END IF
C
            ICYC = ICYC + 1
            IF (TL - TIME .LE. 1.0D-8) THEN
                CALL OUTLST
                TL = TIME + TLST(IDTS)*1.00000001D0
                TL = TL - MOD(TL, TLST(IDTS))
            END IF
            CALL OUTFIL
            IF (XEND - TIME .LE. 1.0D-8) GO TO 3100
4000         CONTINUE
3100         CONTINUE
            DT = DTT(IDTS)
C
4200         CONTINUE
4500         CONTINUE
C
            RETURN
C
7000         FORMAT(1H0,'** ERROR IN SOLRKV, ICON =',I8
                . /1H ',' TIME :',1PD13.4
                . /1H ',' EPSA :', D13.4
                . /1H ',' EPSR :', D13.4)
            END

```

```

*****
SOLVE **
*****

```

```

C*****
C** SOLVE ***
C*****
        SUBROUTINE SOLVE
C
C** SOLVE DIFFERENCE EQUATIONS ***
*INCLUDE COMMON
C
        IF (IOPT.LE.1 .OR. IOPT.GE.9) THEN
            CALL SOLJAC
        ELSE IF (IOPT .EQ. 2) THEN
            CALL SOLRKG
        ELSE IF (IOPT .EQ. 3) THEN
            CALL SOLHMG
        ELSE IF (IOPT .EQ. 4) THEN
            CALL SOLRKV
        ELSE IF (IOPT .EQ. 5) THEN
            CALL SOLADM
        ELSE IF (IOPT .EQ. 6) THEN
            CALL SOLEUL
        ELSE IF (IOPT .GE. 7) THEN
            CALL SOLJRB
        END IF
C

```

```

C.. EPS CHECK OUTPUT ...
*VOCL LOOP, SCALAR
  DO 4000 I = 1, NELM
    IF (NEPSC(I) .GT. 0) THEN
      WRITE(6, 6000) CELM, NEPSC
      CALL ELMOUT ( TEPSC )
      GO TO 9999
    END IF
  4000 CONTINUE
C
  9999 CONTINUE
  RETURN
C
  6000 FORMAT(1H0, '*** ERROR SUMMARY OF EPS CHECK ***'
    /1H0, ' ', 13(3X, A5)
    /1H, 'COUNTS --', 13(1X, I7))
  END

```

```

*****
* TIMSET **
*****

```

```

C*****
C** TIMSET ***
C*****
  SUBROUTINE TIMSET ( XTIM, XEND, XLST, DTX, TL, XDT, NC1, NC2 )
C
*INCLUDE COMMON
C
  IF (XLST .LE. 0.000) THEN
    TL = XEND*2.000
  ELSE
    NL1 = XTIM/XLST + 1.000000100
    TL = NL1*XLST
  END IF
  IF (DTX .GT. 0.000) THEN
    NC1 = XTIM/DTX + 1.0000000100
    NC2 = XEND/DTX + 1.00-8
    XDT = NC1*DTX - XTIM
  ELSE
    NC1 = 1
    NC2 = 999999999
    XDT = 0.000
  END IF
C
  RETURN
  END

```

```

*****
* VCALC **
*****

```

```

C*****
C** VCALC ***
C*****

```

```

      SUBROUTINE VCALC ( WX )
C
C** CALCULATE REACTION VELOCITY "V" ***
*INCLUDE COMMON
C
      DIMENSION WX(NELM)
C
C.. CONCENTRATION MINIMUM CHECK ...
      CALL EPSCHK ( WX )
C
      CN      = WX(5) + WX(8) + WX(2) + 3*WX(4) + WX(12) + WX(13)
      + 2*(WX(10) + WX(1) + WX(7)) + 4*(WX(11) + WX(3) + WX(9))
      CN04    = CN**0.4D0
C
      AV(1)   = AK(1)*WX(3)*WX(11)/(WX(5) + 0.05D0)**2
      AV(2)   = AK(2)*WX(3)**2*WX(12)**2
      + ( WX(4)**2*WX(5)**4*(CN + 0.35D0)**2 )
      AV(3)   = AK(3)*WX(3)*WX(13)/(WX(5) + 0.35D0)
      AV(4)   = AK(4)*WX(4)*SQRT(WX(6)*WX(5))*CN04
      AV(5)   = AK(5)*WX(6)*WX(13)*WX(5)
      AV(6)   = AK(6)*WX(9)*WX(7)*(2.16D0 + 12.5D0*CN)
      AV(7)   = AK(7)*WX(9)*WX(3)/WX(5)**4
      AV(8)   = AK(8)*WX(7)*WX(13)/WX(5)**1.3D0
      AV(9)   = AK(9)*WX(7)*WX(11)
      AV(10)  = AK(10)*WX(8)*WX(11)*(1.6D0/WX(5)**2 + 1.42D0*WX(5))
      AV(11)  = AK(11)*WX(7)*WX(12)/WX(5)
      AV(12)  = AK(12)*WX(2)*WX(3)*WX(5)/(WX(5) + 0.054D0)
      AV(13)  = AK(13)*WX(2)*WX(4)*WX(5)
      AV(14)  = AK(14)*WX(1)*WX(6)/WX(5)
      AV(15)  = AK(15)*WX(1)*WX(4)
      AV(16)  = AK(16)*WX(2)*SQRT(WX(6))*CN04*WX(5)**0.6D0
      AV(17)  = AK(17)*WX(5)
      AV(18)  = AK(18)*WX(6)
      AV(19)  = AK(19)*WX(8)**2*WX(5)**2
      AV(20)  = AK(20)*WX(5)**1.3D0*CN**2*WX(6)*WX(8)/(WX(6)+WX(8))
      AV(21)  = AK(21)*WX(8)*WX(9)/WX(6)**0.5D0
      AV(22)  = AK(22)*WX(4)*WX(7)*(2.13D3 + 1.86D2/WX(5))
      AV(23)  = AK(23)*WX(4)*WX(8)/WX(5)**1.3D0
      AV(24)  = AK(24)*WX(1)*WX(13)/WX(5)
      AV(25)  = AK(25)*WX(3)**2.0D0/(WX(5)**2.2D0*CN**0.54D0)
      AV(26)  = AK(26)*WX(2)**2.0D0*WX(5)
      AV(27)  = AK(27)*WX(5)*WX(6)*WX(12)
C
      RETURN
      END

*****
VELOUT **
*****

C*****
C** VELOUT ***
C*****
      SUBROUTINE VELOUT
C

```

```

C** LIST OUTPUT OF CHEMICAL REACTION VELOCITY ***
*INCLUDE COMMON
C
  PARAMETER ( NLL = 10 )
  PARAMETER ( NNN = (NREF - 1)/NLL + 1 )
C
  WRITE(6, 6000)
  IE = 0
*VOCL LOOP, SCALAR
  DO 4000 I = 1, NNN
    IS = IE + 1
    IE = MIN(NREF, IE + NLL)
    WRITE(6, 6100) (J, J = IS, IE)
    WRITE(6, 6200) (AV(J), J = IS, IE)
  4000 CONTINUE
C
C
  RETURN
C
6000 FORMAT(1H0,3X,'*** REACTION VELOCITY, V(X,T) ***')
6100 FORMAT(1H0,9X,10('V-',I2.2,':',7X))
6200 FORMAT(1H ,7X,1P10D11.3)
END

```