

(U, Pu) O_{2-x} 系燃料の固・液相線温度の解析的評価

(技術報告)

2003年2月

核燃料サイクル開発機構
東 海 事 業 所

本資料の全部または一部を複写・複製・転載する場合は、下記にお問い合わせください。

〒319-1184 茨城県那珂郡東海村村松4番地49
核燃料サイクル開発機構
技術展開部 技術協力課

Inquiries about copyright and reproduction should be addressed to :

Technical Cooperation Section,
Technology Management Division,
Japan Nuclear Cycle Development Institute
4-49 Muramatsu, Tokai-mura, Naka-gun, Ibaraki 319-1184,
Japan

© 核燃料サイクル開発機構 (Japan Nuclear Cycle Development Institute)
2003

(U, Pu)O_{2-x}系 燃料の固・液相線温度の解析的評価

(技術報告)

長井修一朗*

要旨

- ・定温・定圧での平衡の基準 “Gibbs 自由エネルギーは極小” の原理に基づいた、2相2成分及び2相3成分系状態図計算プログラムを作成した。
- ・このプログラムを用いて、理想溶液モデルで記述される UO₂-PuO₂系、正則溶液モデルで記述した Ag-Au-Cu系状態図の計算結果を評価し、解析プログラムの妥当性を確認した
- ・U-O及びPu-O系の固・液相線を最も良く再現するように設定したパラメータを用いたU-Pu-O系状態図は、O/Mの低下に従い小さな極大を持つPu-O系と類似の固・液相線となり、Aitken等及び森本の測定結果と良く一致した。
- ・Aitken等によるU-Pu-O系の固・液相線の測定結果は測定精度向上を図る必要があるが、Pu-O系の固・液相線と自己矛盾のない優れた測定データであることが確認できた。

*ブルトニウム燃料センター

Analytical Evaluation of Solidus and Liquidus Temperature
on (U,Pu) O_{2-x} Fuel
(Technical Document)

Shuichirou Nagai*

Phase diagram calculation code PD32 for 2 phases-2components and 2 phases-3components systems was developed based on Gibbs Free Energy Minimum Principal under equilibrium condition.

Phase diagram calculation of UO_2 - PuO_2 system described by ideal solution model and Ag-Au-Cu system described by regular solution model showed that PD32 gave the reasonable results, that was, the chemical potential of each component had good agreement between two phases.

Free energy of solid U-O, Pu-O and (U,Pu) O_{2-x} systems was calculated based on Blackburn's oxygen potential model. For the calculation of free energy of liquid systems, three parameters which modified the equilibrium constants between the metal ions and oxygen pressure in Blackburn model was introduced, and was established to fit the solidus and liquidus line in U-O and Pu-O systems.

Calculated solidus and liquidus line and their O/M dependence in (U,Pu) O_{2-x} system agreed well with measured data by Aitken et al.

It was confirmed that the solidus and liquidus line between Pu-O and (U,Pu) O_{2-x} systems by Aitken et al was self-consistent.

* Plutonium Fuel Center

目 次

1. はじめに	1
2. 状態図計算の熱力学的考え方	1
3. 状態図の計算方法	2
3.1 計算方法	3
3.2 自由エネルギー極小値の探索方法	3
4. 簡単な系の状態図計算	4
4.1 2相2成分系状態図	4
4.1.1 UO_2 - PuO_2 系の状態図	4
4.1.2 正則溶液モデルから得られる状態図	4
4.1.3 共晶混合物を持つ系の状態図	5
4.2 2相3成分系状態図	6
4.2.1 理想溶液状態図	6
4.2.2 正則溶液の状態図	6
4.3 状態図計算プログラムの評価	6
5. 燃料の状態図計算	7
5.1 燃料の酸素分圧モデル	7
5.2 U-0系状態図	8
5.2.1 正則溶液モデル	8
5.2.2 固相の相互作用パラメータの評価	9
5.2.3 固相と酸素濃度の高い液相の状態図	9
5.2.4 固相と酸素が若干溶解している液体金属ウランとの状態図	10
5.2.5 2液相状態図	10
5.3 Pu-0系状態図	11
5.3.1 Aitken等によるPu-0及びU-Pu-0系状態図の測定結果	11
5.3.2 Pu-0系状態図の計算	11
5.4 U-Pu-0系の状態図	12
5.4.1 系の自由エネルギーの評価方法	12
5.4.2 基準組成のU及びPuのケミカルポテンシャル	13
5.4.3 液相U-Pu-0系のモデル	15
5.4.4 U-Pu-0系の自由エネルギー	15
5.4.5 U-Pu-0系状態図計算結果	15
5.5 考察	16
5.5.1 U-Pu-0系の熱力学的挙動	16
5.5.2 UO_2 - PuO_2 系は理想溶液か	17
5.5.3 Blackburnモデルに基づく状態図計算の妥当性	17
5.5.4 イオン間平衡定数比	18

(1) イオン間平衡定数比の評価	18
(2) $f_{L^{21}}$ の感度	18
(3) $f_{L^{22}}$ の感度	19
5.5.5 等組成状態図	19
5.5.6 Aitken 等の測定結果の評価	20
6. まとめ	21
7. おわりに	21
8. 謝辞	22
9. 参考文献	22

付録

A-1 状態図計算プログラム PD32 概要	付-A1(67)
A-2 構成元素、化合物の自由エネルギー	付-A2(69)
A.3 PD32 ソースプログラム	付-A3(72)

< 表図目次：本文 >

Table 4-1	Ag,Au,Cu の熱力学データ	24
fig 2-1	2 相共存時の自由エネルギーとケミカルポテンシャルの関係	25
fig 3-1	自由エネルギー極小値探索模式図	25
fig 4-1	UO ₂ -PuO ₂ 系状態図の計算結果と測定値の比較	26
fig 4-2	UO ₂ -PuO ₂ 系 固相及び液相の自由エネルギー (2600C)	26
fig 4-3	UO ₂ -PuO ₂ 系 固・液相の自由エネルギー (拡大図)	27
fig 4-4	UO ₂ -PuO ₂ 系における固相及び液相の組成、固層の存在割合の 組成依存性 (2600C)	27
fig 4-5	Ag-Au 系状態図 (理想溶液モデル)	28
fig 4-6	Ag-Au 系状態図 (w ^S =-700,w ^L =0.0 Joul)	28
fig 4-7	Ag-Au 系状態図 (正則溶液モデル、w ^S =-5000,w ^L =0.0 Joul)	29
fig 4-8	Ag-Au 系の状態図 (正則溶液モデル、w ^L =-5000,w ^S =0.0 Joul)	29
fig 4-9	共晶混合物を持つ系の自由エネルギーと状態図	30
fig 4-10	Ag-Au-Cu 系状態図 (理想溶液モデル、1050C)	30
fig 4-11	Ag-Au-Cu 系状態図 (正則溶液モデル、1100C)	31
fig 4-12	Ag-Au-Cu 系状態図 (正則溶液モデル、1110C)	31
fig 4-13	2 相間のケミカルポテンシャルの計算誤差	32
fig 5-1	酸素分圧の測定値と解析値の比較 (U-O 系、O/U < 2.0)	32
fig 5-2	酸素分圧の測定値と解析値の比較 (U-O 系、O/U > 2.0)	33
fig 5-3	酸素分圧の測定値と解析値の比較 (Pu-O 系)	33
fig 5-4	酸素分圧測定値と解析値の比較 (U-Pu-O 系)	34
fig 5-5	酸素分圧の測定値と解析値の比較 (U-O 系)	34
fig 5-6	酸素分圧の測定値と解析値の比較 (Pu-O 系)	35
fig 5-7	酸素分圧の測定値と解析値の比較 (U-Pu-O 系)	35
fig 5-8	(0.2Pu,0.8U) O _{2-x} 系での Pu イオンの存在割合	36
fig 5-9	O/U < 2.0 領域の状態図 (模式図)	37
fig 5-10	U-O 系の相互作用パラメータ (w)	38
fig 5-11	U-O 系の自由エネルギー	38
fig 5-12	U-O 系の U の活量	39
fig 5-13	U-O 系の状態図 解析結果と測定値の比較	39
fig 5-14	U-O 系の固相・液相 (L2) の自由エネルギー (2600C)	40
fig 5-15	2 相間のケミカルポテンシャルの計算誤差 (U-O 系)	40
fig 5-16	液相 L1 の相互作用パラメータ (w)	41
fig 5-17	U-O 系の固相・液相 (L1) の自由エネルギー	41
fig 5-18	U- O _{2-x} 系状態図 計算結果と測定値の比較	42
fig 5-19	液相 L1 及び L2 の自由エネルギー	42
fig 5-20	U (L1) -UO _{2-x} (L2) 系の状態図	43

fig 5-21	Pu-O 及び U-Pu-O 系の固・液相線の測定結果 (Aitken 等)	43
fig 5-22	固相 Pu-O 系の相互作用パラメータ (w)	44
fig 5-23	固相 Pu-O 系の自由エネルギー	44
fig 5-24	Pu-O 系の状態図の解析結果と測定値の比較	45
fig 5-25	Pu-O 系 固・液相の自由エネルギー (2400C)	45
fig 5-26	Pu-O 系 固・液相の自由エネルギー (2400C 部分拡大図)	46
fig 5-27	U-Pu-O 系にて、O のケミカルポテンシャルが既知の時、他の 2 成分の ケミカルポテンシャルを計算する模式図	46
fig 5-28	U-Pu-O 系の自由エネルギー (2600C)	47
fig 5-29	U-Pu-O 系の自由エネルギー (2600C)	47
fig 5-30	U-Pu-O 系の状態図の計算結果 (2600C)	48
fig 5-31	U-Pu-O 系の計算誤差 その 1 (U 及び Pu, 2600C)	48
fig 5-32	U-Pu-O 系の計算誤差 その 2 (O, 2600C)	49
fig 5-33	U-Pu-O 系状態図の計算結果 (2500C)	49
fig 5-34	U-Pu-O 系状態図の計算結果 (2700C)	50
fig 5-35	U-Pu-O 系状態図の計算結果 (2800C)	50
fig 5-36	U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu)=0.2)	51
fig 5-37	U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu)=0.4)	51
fig 5-38	U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu)=0.6)	52
fig 5-39	U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu)=0.28)	52
fig 5-40	(U,Pu) O ₂ 系での UO ₂ 及び PuO ₂ の活量 (2600C)	53
fig 5-41	(U,Pu) O _{1.7} 系での UO _{1.7} 及び PuO _{1.7} の活量	53
fig 5-42	(U,Pu) O _{2.0} 系の酸素分圧、及び U,Pu の活量 (2600C)	54
fig 5-43	(U,Pu) O _{1.7} 系の酸素分圧、及び U,Pu の活量 (2600C)	54
fig 5-44	U-Pu-O 系の Pu の等活量線 (2600C)	55
fig 5-45	U-Pu-O 系の U 及び Pu の平均原子価	55
fig 5-46	UO ₂ -PuO ₂ 系状態図のパラメータサーベイ	56
fig 5-47	U-Pu-O 系状態図のパラメータサーベイ (2600C)	56
fig 5-48	Blackburn モデルのイオン間の平衡定数と状態図の関係 (U-O 系)	57
fig 5-49	Blackburn モデルのイオン間の平衡定数と状態図の関係 (U-Pu-O 系)	57
fig 5-50	Blackburn モデルのイオン間の平衡定数と状態図の関係 (Pu-O 系)	58
fig 5-51	イオン間平衡定数比	58
fig 5-52	Pu-O 系の感度解析 (その 1)	59
fig 5-53	U-Pu-O 系状態図の感度解析 (その 1、2600C)	59
fig 5-54	U-Pu-O 系の感度解析 (その 1、2450C)	60
fig 5-55	Pu-O 系の感度解析 (その 2)	60
fig 5-56	U-Pu-O 系の感度解析 (その 2)	61
fig 5-57	U-Pu-O 系の等組成状態図 (その 1)	61
fig 5-58	U-Pu-O 系の等組成状態図 (その 2)	62

fig 5-59	Pu-O 系の状態図 (case U, $f_L^{21}=1.0, f_L^{22}=0.0$)	62
fig 5-60	U-Pu-O 系の状態図 (2600C) (case U, $f_L^{21}=1.0, f_L^{22}=0.0$)	63
fig 5-61	U-Pu-O 系等組成状態図 その 3 (case U, $f_L^{21}=1.0, f_L^{22}=0.0$)	63
fig 5-62	U-Pu-O 系等組成状態図 その 4 (case U, $f_L^{21}=1.0, f_L^{22}=0.0$)	64
fig 5-63	Ag-Au-Cu 系を利用して U-Pu-O 系を模擬した状態図 (その 1、等組成状態図)	64
fig 5-64	Ag-Au-Cu 系を利用して U-Pu-O 系を模擬した状態図 65 (その 2、2000C 等組成状態図)	65

< 表図目次 : 付録 >

fig A-1	O, O ₂ , U 及び Pu の自由エネルギー	A2-2
fig A-2	UO ₂ , U 及び PuO ₂ の自由エネルギー	A2-3

1. はじめに

・MOX 燃料の融点は燃料要素の最高線出力を決める重要な物性値の一つであり、プルトニウム燃料センター製造加工部技術開発室でその測定を実施している。

・MOX 燃料は U-Pu-O の三成分から構成され、この系の融点測定には

O/M が 2.0 に近づくと、るつぼ材のタングステンと反応しやすくなり、正確な融点の測定が難しい

三成分系は測定すべき点が多く、それら全てを実験により求めるには、膨大な時間を要する等の困難な課題がある。上記課題解決の一助として、熱力学的手法に基づいた、2 相 2 成分系及び 2 相 3 成分系の状態図解析プログラムを作成した。

・熱力学的手法に基づく状態図の計算プログラムは長谷部等による総説⁽¹⁾⁽²⁾で述べられているように現在まで多くの研究がなされており、完成された計算システムとして市販されている。しかしながら、

対象である U-Pu-O 系の固・液相線の計算は高々 2 相 3 成分系と変数の数も少なく比較的容易にコーディング可能

U-Pu-O 系の自由エネルギーは O/M が 2.0 近傍で急変し、解析関数でそれを記述するのは困難なことから、系の自由エネルギー計算ルーチンが必要となり、このルーチンと既存の計算システムの両立が疑問

自作することにより、システムの改良、維持管理が容易であり、コストも掛からない

等の理由で自作した。

・融点近傍での U-Pu-O 系燃料の固・液相線の研究の主たる物は次の 3 つである。

Lyon 等⁽³⁾は UO_2 - PuO_2 系の固・液相線の測定を行い、この系は理想溶液モデルで記述し得ることを示した。

Latta 等⁽⁴⁾は UO_{2+x} 系の融点を測定し、固・液相温度が O/M の減少と共に単調に低下することを示した。

Aitken 等⁽⁵⁾は Pu-O 系及び U-Pu-O 系の融点を測定し、これらの系の固・液相温度の O/M 依存性が極めて小さいとの結果を得た。

・Lyon 等及び Latta 等の結果は測定点数も多く信頼性が高いが、一方 Aitken らの測定結果は測定点数が少なく、かつ固・液相線の O/M 依存性が U-O 系と大きく異なることから、熱力学的な面からの検討が必要と思われた。プルトニウムセンター製造加工部技術開発室で実施している MOX 燃料の融点測定の目的は、信頼性の高い測定データを取得することであり、ここで報告する状態図解析プログラムの開発及び MOX 燃料の固・液相線解析の目的は

状態図解析用のツールを準備し、測定の効率化に資する

Aitken 等の測定結果の評価を行う

ことにある。

2. 状態図計算の熱力学的考え方

・熱力学によると、定温・定圧状態での平衡の基準は "Gibbs 自由エネルギー G は極小" すなわち

$$G = \min.$$

である。この基準を多相系に適用すると、"相を構成している各成分のケミカルポテンシャルは相間で

等しい”

$$\begin{aligned}\mu_1^A &= \mu_1^B = \mu_1^C = \mu_1^D \cdots \cdots \\ \mu_2^A &= \mu_2^B = \mu_2^C = \mu_2^D \cdots \cdots \\ &\cdot \\ &\cdot\end{aligned}$$

と表現される新たな基準を導くことができる。ここで、1,2,・・・は成分番号、A,B,C,D・・・は相の名前。
 ・これらの基準を2成分系の固・液平衡に対して適用したものを fig 2-1 に示す。横軸は成分1のモル分率(x),縦軸は固相及び液相の1モルあたりの自由エネルギー g^s 及び g^l で、それぞれ以下のように定義される。

$$\begin{aligned}x &= n_1 / (n_1 + n_2) \\ g^s &= G^s / (n_1 + n_2) \quad , \quad g^l = G^l / (n_1 + n_2)\end{aligned}$$

ここで、 G^s 及び G^l は固相及び液相の自由エネルギー、 n_1, n_2 は各成分のモル数。

・各成分のケミカルポテンシャル μ は

$$\begin{aligned}\mu_1 &= (G / n_1)_{n_2} = g + (1-x)(g^l / x) \\ \mu_2 &= (G / n_2)_{n_1} = g - x(g^l / x)\end{aligned}$$

と書けるから、例えば液相の1モルあたりの自由エネルギー g^l 上の任意の点 A での接線の $x=0$ 及び $x=1$ 軸との交点はそれぞれ、 μ_1^L, μ_2^L となる。

・fig 2-1 の直線 $p^s - p^l$ のように両相の自由エネルギー曲線に対して、接点 p^s 及び p^l で共通接線が引ける場合には、固相の組成 x^s でのケミカルポテンシャルは液相の組成 x^l のケミカルポテンシャルと等しくなる。従って、組成 x^s の固相と組成 x^l の液相は共存し平衡状態にある。

・組成 x が x^s より小さい場合、 g^s が g^l より小さいので、平衡状態では、自由エネルギー極小の基準から、固相のみが存在する領域となり、 x が x^l より大きい場合、液相のみが存在する領域となる。

・ x^s と x^l 間の領域については、共通接線の線分 $p^s - p^l$ に沿っての1モルあたりの自由エネルギー g^* は

$$g^* = A_p g^s(x^s) + (1-A_p)g^l(x^l) \quad \text{----- (2-1)}$$

と、各相の g の相加平均で書き表せる。ここで、 $g^s(x^s)$ 及び $g^l(x^l)$ は x^s 及び x^l の組成における固相及び液相の1モルあたりの自由エネルギー、 A_p は $(x^l - x) / (x^l - x^s)$ で表される固相の割合である。

・ g^* は $x^s - x^l$ 間で必ず g^s 及び g^l より小さいので、自由エネルギー極小の基準から(2-1)式で表される固相・液相の2相共存状態が平衡状態となる。

3. 状態図の計算手法

・状態図の解析は2章で述べた(1)自由エネルギーの極小値に対応する組成を求める方法及び(2)相間のケミカルポテンシャルが等しい組成を求める方法に大別され多くの方法が開発されている。詳細は長谷部等の総説⁽¹⁾⁽²⁾を参照されたい。

・本報告では

*微分操作が不要である

*FORTRAN による coding が容易

等の理由で、自由エネルギーの極小値に対応する組成を求める方法を用いた。

3.1 計算方法

・2相3成分系に対しては、組成の関係

$$x^S + y^S + z^S = 1 \quad \text{----- (3-1)}$$

$$x^L + y^L + z^L = 1 \quad \text{----- (3-2)}$$

及び、相の存在割合 A_p と組成の間に

$$A_p x^S + (1 - A_p) x^L = A_x \quad \text{----- (3-3)}$$

$$A_p y^S + (1 - A_p) y^L = A_y \quad \text{----- (3-4)}$$

の関係が成立する。ここで A_x, A_y は2つの相 (S 及び L) からなる系全体の X, Y 成分のモル分率。(成分 Z については組成の関係から導かれ独立の関係式とはならない。)

・従って、解析対象の組成 A_x, A_y ($A_z = 1 - A_x - A_y$) を既知とすると、7個の未知変数に対して(3-1)~(3-4)の4個の関係式が存在することから、残りの3個の変数に対して自由エネルギー

$$g^* = A_p g^S(x^S, y^S) + (1 - A_p) g^L(x^L, y^L) \quad \text{----- (3-5)}$$

の極小値に対応する固相・液相の組成、及び相の存在割合を求める。

・2相2成分系では、成分 Z が無く、相の存在割合と組成の関係式が1つ減るため、5個の未知変数に対して3つの関係式が存在することになり、残り2個の変数をパラメータとして g^* の極小値に対応する組成、相の存在割合を求める。

3.2 自由エネルギー極小値の探査方法

・(3-5) 式の未知変数の組 $V = V(x^S, y^S, A_p)$ をベクトルとみなし、 g^* を小さくするベクトルの方向に未知変数を変化させることを繰り返して、 g^* の極小値に対応する未知変数の組を求める。fig 3-1 参照。

・未知変数の初期値を A_{p0}, x_0^S 及び y_0^S とすると、

$$V = V_0(x_0^S, y_0^S, A_{p0}) + V(x^S, y^S, A_p)$$

$$A_p = A_{p0} + D_2 \cos$$

$$x^S = x_0^S + D_1 \sin$$

$$y^S = y_0^S + D_1 \cos$$

に対応する

$$g^* = g^*(x^S, y^S, A_p)$$

を計算し、制限条件(3-1)~(3-4)を満たす (x^S, y^S, A_p) の組の内、 g^* を最も小さくする組 を求め、それを初期値(g_0^*)として同様の計算を繰り返す。

ここで、 D_1, D_2 は x^S, y^S 及び A_p のきざみ幅、 D_1 は $0 \sim \pi$ 、 D_2 は $0 \sim 2$ 。

・いかなる D_1 及び D_2 に対しても $g^* > g_0^*$ となった場合、 D_1, D_2 を小さくして同様の計算を繰り返し、 D_1, D_2 が一定値 (10^{-10}) 以下となった場合の x^S, y^S, A_p を求める解とし、この時の g^* を自由エネルギーの極小値とした。

・本プログラムでは設定した温度(T) 及び組成 (A_x, A_y) に対して、得られた A_p が $0 < A_p < 1$ の場合、系は2相状態にあり、 A_p が一定値 (10^{-7}) 以下であればL相の单相状態、 $(1 - A_p)$ が 10^{-7} 以下であればS相の单相状態としている。

・自由エネルギーの極小値探査ルーチンは2相2成分系(SSCN)及び2相3成分系(SSCN3)と別々に作成され、プログラム中に組み込まれている。

4. 簡単な系の状態図計算

系の自由エネルギーが組成の簡単な関係式で表されるような系を取り上げて

- ・作成プログラムの妥当性
- ・系の自由エネルギーと状態図の関連

を確認する。

4.1 2相2成分系状態図

4.1.1 UO₂ - PuO₂系の状態図

・標記の系の状態図は Lyon 等⁽²⁾により報告されており、その固相線及び液相線は理想溶液モデルにて説明できることが知られている。

・固相、液相とも理想溶液モデルにて記述し得るならば、系の自由エネルギーは

$$g^s = x^s \mu^{*s}_{UO_2} + (1 - x^s) \mu^{*s}_{PuO_2} + RT\{x^s \ln x^s + (1 - x^s) \ln(1 - x^s)\}$$

$$g^l = x^l \mu^{*l}_{UO_2} + (1 - x^l) \mu^{*l}_{PuO_2} + RT\{x^l \ln x^l + (1 - x^l) \ln(1 - x^l)\}$$

$$g^* = A_p g^s + (1 - A_p) g^l$$

と書ける。

ここで、 μ^* は固体又は液体状態のUO₂, PuO₂単体のケミカルポテンシャル(= 1モルあたりの自由エネルギー)、Rは気体定数、Tは温度(K)、xはUO₂のモル分率である。固体、液体状態のUO₂及びPuO₂の自由エネルギーは巻末の付録参照。

・fig 4-1に計算により求めた状態図とLyon等⁽²⁾及びAitken等⁽⁴⁾により報告されている測定結果を示す。両者は良く一致しており、UO₂ - PuO₂系は理想溶液モデルにて記述し得ることを示している。但しAitkenらの測定結果に対しては55Cの補正を施してある(5.4.1節参照)

・fig 4-2に2600CにおけるUO₂-PuO₂系の固相及び液相の自由エネルギーをUO₂のモル分率の関数として示す。x(UO₂)が0.47近傍で交差し、交点よりx(UO₂)が小さい領域では液相の自由エネルギーが小さく、大きい領域では固相の自由エネルギーが小さい。

・fig 4-3に交点近傍の拡大図に2相系の自由エネルギー g^* をオーバープロットしたものを示す。2相領域において g^* は g^s 及び g^l の共通接線となっており、2章で述べたことを裏付けている。

・fig 4-4にsub.SSCNから出力された2600Cにおける固相・液相の組成(x^s , x^l)、及び固相の存在割合 A_p を2相からなる系全体の組成 A_{UO_2} の関数として示した。但し、単一相の領域で存在しない相の組成は無意味となるが、その場合は存在しない相の組成は0としている。単一相の領域では存在する相の組成xは系全体のUO₂の組成 A_{UO_2} と一致している。2相領域では固相及び液相の組成は一定値を持ち、 A_p が0~1まで直線的に変化し、この関係(lever relation)が成立している。

・以上のことから、2相2成分系状態図計算ルーチンSSCNの解析手法の妥当性を確認できた。

4.1.2 正則溶液モデルから得られる状態図

・本節はAg-Au系を題材として、正則溶液モデル⁽⁶⁾を用いて、モデルのパラメータと状態図の関係を調べ、これからの状態図解析の理解に資する。従って、以下に図示するAg-Au系の状態図は本物の状態図とはまったく異なったものである。

・正則溶液モデルは系の自由エネルギーを以下の式で表す。

$$g^s = x^s_{Ag} \mu^{*s}_{Ag} + x^s_{Au} \mu^{*s}_{Au} + RT(x^s_{Ag} \ln x^s_{Ag} + x^s_{Au} \ln x^s_{Au}) + x^s_{Ag} x^s_{Au} W^s \quad \text{----- (4-1)}$$

$$g^L = x_{Ag}^L \mu_{Ag}^{*L} + x_{Au}^L \mu_{Au}^{*L} + RT(x_{Ag}^L \ln x_{Ag}^L + x_{Au}^L \ln x_{Au}^L) + x_{Ag}^L x_{Au}^L w^L \quad \text{----- (4-2)}$$

ここで、w は理想溶液からのずれを表すパラメータで、系の混合熱と直接関連付けられる。

・構成成分(Ag 及び Au)の自由エネルギーは測定された厳密なものは使用せず、以下のように単純化した近似式を用いた。

$$\begin{aligned} \mu^{*S} &= H^S - TS^S \\ &= H_m^S + C_p^S(T - T_m) - T\{S_m^S + C_p^S \ln(T/T_m)\} \\ \mu^{*L} &= H^L - TS^L \\ &= (H_m^S + \Delta H_f) + C_p^L(T - T_m) - T\{(S_m^S + \Delta S_f/T_m + C_p^L \ln(T/T_m))\} \end{aligned}$$

ここで、H はエンタルピー(Joul/mole)、S はエントロピー(Joul/mole,K)、Cp は比熱(Joul/mole,K)、 ΔH_f は融解熱(Joul/mole)、サフィックス m は融点での値。このように g を設定すると、融点にて $\mu^{*S} = \mu^{*L}$,融点以下では $\mu^{*S} < \mu^{*L}$, 融点以上では $\mu^{*S} > \mu^{*L}$ となる。個々の成分の Cp, ΔH_f 等は表 4 - 1 にまとめて示してある。

・fig4-5 に理想溶液モデル($w^S = w^L = 0$)で計算した Ag - Au 系の状態図を示す。理想溶液の特色であるレンズ状の2相領域が著しく狭まっているのは両者の融解熱がほぼ等しいことによる。

・fig4-6 に $w^S = -700\text{Joul}$, $w^L = 0$ の時の状態図を示す。固相の自由エネルギーの低下によって、固相領域が拡大し、実際の状態図⁽⁷⁾に近いものとなっている。

・fig4-7 に $w^S = -5000\text{Joul}$, $w^L = 0$ のときの状態図を示す。固・液相線が極大値を持つ化合物形成を示す状態図となる。全組成範囲でこのような状態図となる例は Hansen の状態図集⁽⁷⁾では見つからなかったが、状態図の一部ではしばしば出現する。

・fig4-8 に $w^S = 0$, $w^L = -5000\text{Joul}$ のときの状態図を示す。固・液相線が極小値を持ち、溶融塩系(NaCl-KCl 系) や性質の類似した成分からなる系(K-Cs 系)などで出現する。

・正則溶液モデルの相互作用パラメータ w^S 及び w^L について、 $w^S < w^L$ ならば固相の自由エネルギーが減少するため、固相の存在領域が拡大し、逆ならば、液相の存在領域が拡大する。

・以上の例から、正則溶液モデルは w をパラメータとすることにより様々の状態図を画くことが可能であり、状態図計算には簡単且つ有効なモデルであることが判る。

4.1.3 共晶混合物を持つ系の状態図

・液体は相互にいかなる割合でも溶解するが、固体は相互にまったく溶け合わない場合、共晶状態図と呼ばれる簡単な状態図となる。

・このような系の固相の自由エネルギーは

$$g^S = x_{Ag}^S g_{Ag}^S + x_{Au}^S g_{Au}^S = x_{Ag}^S \mu_{Ag}^{*S} + x_{Au}^S \mu_{Au}^{*S}$$

となり、液相の自由エネルギーは(4-2)で書ける。700C における固・液相の自由エネルギーの組成依存及び計算により求めた状態図を fig4-9 に示す。但し、液相については簡単のため、 $w^L = 0$, すなわち理想溶液モデルとした。

・上式で表される固相の自由エネルギーの $x(\text{Ag}) = 1$ 軸との交点は単体 Ag 1 モルの自由エネルギー、すなわち固体純 Ag のケミカルポテンシャル(μ_{Ag}^{*S})である。この点からの液相の自由エネルギー曲線への接線の接点を x_1 とすると、2章の議論から液相の組成 x_1 での Ag のケミカルポテンシャルは固体純 Ag のケミカルポテンシャルと一致し、 x_1 と温度の関係を図示すると共晶混合物を持つ系の状態図となる。Au のリッチな側でも同様である。

・ $x_1 \sim 1$ 、及び $0 \sim x_2$ 間では、接線が固相、液相の自由エネルギーより低いので、固体 Ag 又は Au 単体と液相の 2 相領域となり、 $x_2 \sim x_1$ 間で液相の単相となる。

4.2 2 相 3 成分系状態図

・ Ag-Au-Cu 系の仮想的状態図の計算を行い、3 成分系状態図解析ルーチン SSCN3 の解析結果の妥当性を確認する。

・ Ag-Au-Cu 系は以下の正則溶液モデルを用いた。

$$g^S = x^S \mu^{*S}_{Ag} + y^S \mu^{*S}_{Au} + z^S \mu^{*S}_{Cu} + RT(x^S \ln x^S + y^S \ln y^S + z^S \ln z^S) + x^S y^S w^S_{AgAu} + y^S z^S w^S_{AuCu} + z^S x^S w^S_{CuAg} \quad \text{----- (4-3)}$$

$$g^L = x^L \mu^{*L}_{Ag} + y^L \mu^{*L}_{Au} + z^L \mu^{*L}_{Cu} + RT(x^L \ln x^L + y^L \ln y^L + z^L \ln z^L) + x^L y^L w^L_{AgAu} + y^L z^L w^L_{AuCu} + z^L x^L w^L_{CuAg} \quad \text{----- (4-4)}$$

ここで、 x, y, z はそれぞれ Ag, Au, Cu のモル分率、 w はサフィックスの原子間の相互作用パラメータ (理想溶液からのずれを表すパラメータ) で定数とした。

4.2.1 理想溶液モデルによる状態図

・ fig4-10 に 1050 C における状態図を示す。fig4-5 に示すように理想溶液モデルでは融点は組成の関数で単調に変化するので、2 成分系 Ag-Au 及び Ag-Cu 系で 2 相領域が現れ、2 成分系 Au-Cu 系では固相だけとなる。計算された状態図はこのことを良く再現している。又、fig4-5 に示した Ag-Au 系の 1050C での切片の組成は fig4-10 の Ag-Au 線 ($z^S = 0$) の固相・液相組成と一致している。

4.2.2 正則溶液モデルによる状態図

・ fig4-11 に $w^S_{AgAu} = -5000 \text{Joul}$ 、他の相互作用パラメータは全て 0 とした場合、及び fig4-12 に $w^S_{AgAu} = -5000$ 、 $w^S_{AgCu} = -4000$ 、 $w^S_{AuCu} = 3000 \text{Joul}$ とした場合の状態図を示す。

・ fig4-11 では固相の Ag-Au 間の g のみ低下させているので、その領域で固相が表れている。

・ fig4-12 では固相の Au-Cu 間の g を増加させ、Ag-Au 及び Ag-Cu 間の g を低下させているので、その意図に従った状態図となっている。

・ 2 章で述べたように 2 相が共存している場合、構成成分のケミカルポテンシャルは相間で等しくなる必要がある。3 成分正則溶液で、相互作用パラメータ w が組成に依存しない場合、成分のケミカルポテンシャルは

$$\mu_A = (g / n_A)_{nB, nC} = \mu^*_A + RT \ln(x_A) + x_B w_{AB} + x_C w_{AC} - x_A x_B w_{AB} - x_B x_C w_{BC} - x_C x_A w_{AC}$$

と書けるので、2 相間のケミカルポテンシャルの計算誤差を以下の式で評価できる。

$$= 2 | \mu^S_A - \mu^L_A | / | \mu^S_A + \mu^L_A | \quad \text{----- (4-5)}$$

・ fig4-13 に の計算結果を示す。3 個の成分いずれに対しても、 の計算値は 10^{-7} 以下と良い一致が得られた。

4.3 状態図計算プログラムの評価

・ 2 相 2 成分系状態図及び 2 相 3 成分系状態図の解析ルーチン、SSCN 及び SSCN3 は、自由エネルギー極小に対応する各成分のケミカルポテンシャルは相間で一致した

固・液溶液のモデルに対応した妥当な状態図の計算結果を与えたことから、状態図解析プログラムの計算結果は妥当なものと判断される。

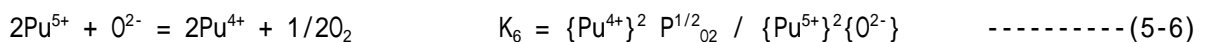
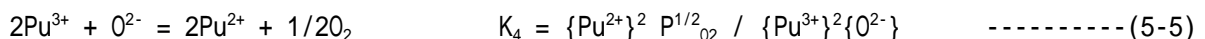
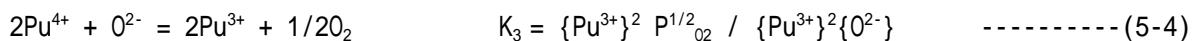
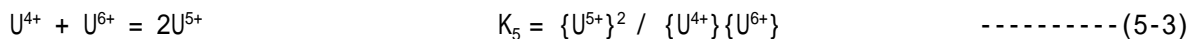
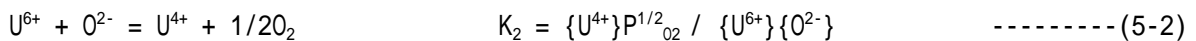
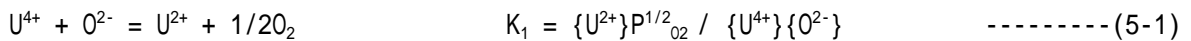
5. 燃料の状態図計算

- ・MOX 燃料の状態図、特に融点(固・液相線)を計算するためには、固・液両相について温度、組成の関数として系の自由エネルギーを求める必要がある。
- ・固相については燃料の酸素分圧の測定結果及びそれに基づいた酸素分圧計算モデルが報告されており、それから系の自由エネルギーを計算し得る。しかし、液相については酸素分圧の測定は皆無であり、むしろ状態図(固・液相線)の測定が液相に対する唯一の熱力学的情報と言える。
- ・従って、U-O 系及び Pu-O 系については、液相においても固相と同様のイオン間平衡が成立していると仮定し、これらの系の固・液相線を良く再現するように液相のイオン間平衡定数を求めた。
- ・上記の固相と液相間の平衡定数の比がそのまま U-Pu-O 系に当てはまると仮定して、この系の状態図を計算し、測定結果と比較した。

5.1 燃料の酸素分圧モデル

・U - O , Pu - O 及び U - Pu - O 系の状態図を解析的に求めるためには、各系の自由エネルギーを組成の関数として作成する必要がある。この目的のために、Blackburn による酸素分圧モデルを若干改良し、過去に報告されている酸素分圧の測定結果で検証したもの⁽⁸⁾を用いた。

・このモデルは、基本的に以下の6つの平衡式



を、マスバランスの条件

$$\{U^{2+}\} + \{U^{4+}\} + \{U^{5+}\} + \{U^{6+}\} = 1 - q$$

$$\{Pu^{2+}\} + \{Pu^{3+}\} + \{Pu^{4+}\} + \{Pu^{5+}\} = q$$

及び、電気的中性の条件

$$\{U^{2+}\} + 2\{U^{4+}\} + 2.5\{U^{5+}\} + 3\{U^{6+}\} + \{Pu^{2+}\} + 1.5\{Pu^{3+}\} + 2\{Pu^{4+}\} + 2.5\{Pu^{5+}\} = \{O^{2-}\}$$

を解くことにより、酸素分圧を求めている。ここで、 $q = \{Pu\} / [\{Pu\} + \{U\}]$ で定義される Pu 分率。

・ K_1 ~ K_6 の値は以下の通り。

$$\ln(K_1) = \{-1.69146196/T + 1.8030446 \cdot 10^{10} / T^2 - 6.483023 \cdot 10^6 / T + 773.1501\} / 2$$

$$\ln(K_2) = -16350/T + 4.96$$

$$\ln(K_3) = -54931/T + 13.37$$

$$\ln(K_4) = -92500/T + 21.3$$

$$\ln(K_5) = 11000/T - 12.75$$

$$\ln(K_6) = 4.3$$

・上記の関係式を用いて計算した酸素分圧と測定結果の比較を fig5-1~fig5-4 に示す。測定値と解析結果は良く一致している。但し、fig5-2 及び 5-3 では測定結果と解析値の間で大きな不一致が認められる測定点があるが、これは O/U が 2.0 近傍で酸素分圧が急激に変化するために、測定に供した燃料組成 (O/U) の分析誤差によるものと考えられる。

・fig5-5 に U-0 系の酸素分圧の測定値と解析結果の比較を O/U の関数としてを示す。解析結果は測定値を良く再現している。又、同図において、Ackermann らの測定値⁽¹⁰⁾ より低 O/U 側は 2 相領域である。U-0 系の酸素分圧の測定は最高温度 2431C まで為されておりこれらのデータで検証された平衡定数 K_1 は信頼性が高い。

・fig5-6 に Pu-0 系の酸素分圧の測定結果と解析結果の比較を O/Pu の関数として示す。Pu-0 系の酸素分圧はイオン反応式から明らかなように、O/Pu が 1.5 以下は K_4 、O/Pu が 1.5 以上は K_3 により制御される。fig5-3 に示した過去に測定されたデータは全て O/Pu が 1.6 以上、温度が 1545C 以下である。解析値は測定結果を良く再現していることから K_3 は妥当なものと判断されるが、 K_4 の妥当性を判断できるほど広い O/M 範囲での酸素分圧の測定は為されていない。又、 K_3 及び K_4 とともに状態図解析に必要な 2200C 以上の高温領域での検証には、次に述べる U-Pu-0 系の酸素分圧測定結果が必要である。

・fig5-7 に U-Pu-0 系の酸素分圧の解析結果と測定値の比較を示す。U-0 系と比べると一致の度合いは劣る。U-Pu-0 系の酸素分圧測定は温度 2276C、O/M が 1.92 までカバーしている。又、fig5-8 に Blackburn モデルから計算される Pu-0 系及び 0.2Pu-0.8U-0 系での Pu イオンの存在割合を示す。O/M が 1.9 まで還元された領域では、fig5-8 に示すように、 Pu^{2+} が 10% 程度存在し、平衡定数 K_3 と K_4 のバランスで酸素分圧が決まる。従って、U-Pu-0 系の酸素分圧の測定結果より、Pu-0 系では検証し得なかった K_3 及び K_4 の高温、低 O/M での妥当性を部分的に確認できた。

5.2 U-0 系の状態図

- ・fig5-9 に O/U が 2.0 より小さい領域の状態図を模式的に示す。
- ・一つの固相(S)と二つの液相 (L1, L2) の存在する単相領域、これらの相の何れか二つが共存する三つの 2 相領域に分けられる。
- ・本報告では、三つの単相領域の自由エネルギーを正則溶液モデルで記述し、それに基づいて 2 相共存線を計算する。

5.2.1 正則溶液モデル

・U - 0 系の自由エネルギーを次のような正則溶液モデルで記述する。

$$G = n_u \mu_u^*(T) + n_o \mu_o^*(T) + RT(n_u \ln x_u + n_o \ln x_o) + (n_u + n_o) x_u x_o w \quad \text{----- (5-7)}$$

ここで、G : Gibbs 自由エネルギー、 μ_u^* , μ_o^* : 標準状態にある、ウラン金属、原子状酸素(気体)のケミカルポテンシャル、 n_u , n_o : ウラン、原子状酸素のモル数、 x_u , x_o : ウラン、原子状酸素のモル分率、w : 相互作用パラメータ (T, x の関数)

・以後の取り扱いの容易さを考えて、 $n_u + n_o = 1$ とし、その時の自由エネルギーを $g (= G/(n_u + n_o))$ とすると、

$$g = x^u \mu_u^*(T) + x_o \mu_o^*(T) + RT(x_u \ln x_u + x_o \ln x_o) + x_u x_o w \quad \text{----- (5-8)}$$

・U-0 系について我々が利用可能な熱力学データは、 UO_2 の生成自由エネルギーと、酸素分圧の測定結果であることから、各相について以下の方法で w を評価した。

5.2.2 固相の相互作用パラメータの評価

・酸素のケミカルポテンシャル μ_o は

$$\begin{aligned} \mu_o &= (G / n_o)_{nu} \\ &= \mu_o^* + RT \ln x_o + x_u^2 w + x_u^2 x_o (w / x_o)_{nu} \end{aligned} \quad \text{----- (5-9)}$$

又、圧力 P_o の酸素のケミカルポテンシャルは

$$\mu_o = \mu_o^* + RT \ln P_o \quad \text{----- (5-10)}$$

と書けるから、両者を一緒にすると以下の微分方程式が得られる。

$$w / x_o + w / x_o + RT \{ \ln(x_o / P_o) / (x_u^2 x_o) \} = 0 \quad \text{----- (5-11)}$$

この方程式は容易に解くことができ、その一般解は以下のように求まる。⁽⁹⁾

$$w = C / x_o + (1 / x_o) \int_{x_o}^{x_o^*} f(x_o) dx_o \quad \text{----- (5-12)}$$

$$f(x_o) = RT \ln(P_o / x_o) / (1 - x_o)^2 \quad \text{----- (5-13)}$$

ここで、 x_o^* は 0/M が 2.0 の時のモル分率、C は積分定数。

・積分定数 C は (5-8) 式を用いて、 UO_2 の自由エネルギーより求まるので、酸素分圧 P_o から (5-12) 式の積分を行うことにより w を計算でき、系の自由エネルギー g を評価できる。

- ・ fig5-10 及び fig5-11 に固相の w とそれから計算された自由エネルギーを O/U の関数として示す。
- ・ 系の U の活量 (a_U) は

$$\begin{aligned} \mu_U &= \mu_U^* + RT \{ \ln x_U + (x_o / x_U) \ln(x_o / P_o) \} + x_o w \\ &= \mu_U^* + RT \ln a_U \end{aligned}$$

より計算され、その結果を fig5-12 に示す。 $\log(a_U) = 0$ 、すなわち $a_U = 1.0$ と交わる O/U 比は 1600 C で 1.96、2100 C で 1.80 と Edwards 等⁽¹¹⁾ による UO_{2-x} と金属 U が平衡にある O/U 比と一致する。 UO_{2-x} と平衡にある金属 U への酸素の溶解度は極めて小さく U の活量は 1 であるから、上記の a_U の計算結果は妥当と思われる。

・状態図の計算では、 w を O/U 又は x_o の関数として正確に多項式で近似することは困難であったため、 O/U が 1.1-2.1 の区間を 1000 等分した各点の w をあらかじめ (5-12) 式で計算し、必要とされる O/U に対する値は Lagrange 補間法⁽¹²⁾ により求めた。

5.2.3 固相と酸素濃度の高い液相の状態図

・ fig5-9 に於いて、温度 T_1 と固相線・液相線の交わる O/U を $O/U^s, O/U^l$ とすると、両相は平衡状態にあるので

$$P_o^s(O/U^s) = P_o^l(O/U^l)$$

となる。 $U-O$ 系のような状態図の場合、 $O/U^l < O/U^s$ であり、且つ酸素分圧は O/U の低下と共に減少するので、任意の O/U に対して

$$P_o^l(O/U) > P_o^s(O/U)$$

が成立する。

・液相 L_2 の酸素分圧も 5-1 節で述べた Blackburn モデルで記述できると仮定すると、液相のイオン間平衡定数 K_1^l は $U-O$ 系の固・液相線を最も良く再現するように

$$K_1^l / K_1 = \exp(f_L^{11})$$

で表されるイオン間平衡定数比 f_L^{11} を決めることにより、液相 L_2 の酸素分圧が計算でき、前節で述べ

た方法で自由エネルギーを評価し得る。

・fig5-13 に f_L^{11} として以下の値を用いて計算した固相線・液相線を Latta 等により報告された測定結果と共に示す。

$$f_L^{11} = 0.75 + (3073 - T_k) \cdot 10.625 \cdot 10^{-4}$$

この時の固・液相の自由エネルギーを O/U の関数として fig5-14 に示す。

・この系での 2 相間のケミカルポテンシャルの差異を (4-5) 式で評価したものを fig5-15 に示す。温度が 2700C 以下では f_L^{11} は 10^{-6} 以下となり、4-2-2 節の結果より 1 桁悪くなった。これは状態図を計算するときの自由エネルギー、及び一致の度合いを評価するときのケミカルポテンシャルを補間法で求めたことによると考えられる。また温度が UO_2 の融点近くまで増加すると f_L^{11} は 10^{-5} まで悪化する。これは fig 5-5 や fig5-12 に見られるように、 O/U が 2.0 近傍では、酸素及び U のケミカルポテンシャルの組成に対する変化の割合が急激に増加するためと考えられる。

5.2.4 固相と酸素が若干溶解している液体金属ウランの状態図

・fig5-9 に示すように、2425C 以下の温度の S+L1 領域では、酸素の若干溶解している液体金属 U と固体 UO_{2-x} とは平衡にある。この 2 相領域では

$$\mu_{O_0}^S = \mu_{O_0}^{L1}$$

であるので、液相線及び固相線の組成を $O/U^{L1}, O/U^S$ とすると

$$\mu_{O_0}^S(O/U^S) = \mu_{O_0}^{L1}(O/U^{L1})$$

が成立する。

・液相 L1 を正則溶液モデル (5-9) で記述すると、酸素のケミカルポテンシャルは

$$\mu_{O_0} = \mu_{O_0}^* + RT \ln x_{O_0} + x_{O_0}^2 w$$

となる。ここで x_{O_0} は液相 L1 中の酸素のモル分率。

・また、固相線に沿っての酸素分圧は Ackermann 等⁽¹⁰⁾ により測定されており、それを $P_{O_0}^n$ とすると、先の (5-9) の関係から

$$w = RT \ln(P_{O_0}^n / x_{O_0}) / x_{O_0}^2$$

によって液相 L1 の熱力学関数は計算できる。

・Ackermann 等⁽¹⁰⁾ による $P_{O_0}^n$ 、及び Edwards 等⁽¹¹⁾ による x_{O_0} の結果を基に計算した w を fig5-16 に示す。得られた w を温度の 2 次式で書くと

$$w = -0.0425Tk^2 + 260.25Tk - 718514$$

となる。

・これより計算された 1600C における液相 L1 の自由エネルギーを固相のそれと一緒に fig5-17 に示す。

・計算により求めた U - UO_{2-x} 系の状態図を fig5-18 に示す。Edwards らの測定結果と良く一致しているが、それは fig5-5 に示すように Blackburn モデルによる酸素分圧の計算値が Ackermann 等⁽¹⁰⁾ による固相線上の酸素分圧の測定結果を良く再現していることが主たる要因である。

5.2.5 2 液相状態図

・fig5-9 に示すように、2425C 以上の高温で、低 O/U 液体混合物は酸素濃度の異なる 2 種類の液体 (L1 及び L2) に分離する。この 2 液相分離は上記で定式化した L1, L2 の熱力学関数で計算できる。

・両液相の 2600C の自由エネルギーを fig5-19 に、状態図の計算結果を fig5-20 に示す

・L2側の液相線に不連続が見られる。これは Blackburn モデルでは(5-1)式から明らかなように、Uの原子価が2より小さい場合、すなわち O/U が1.0以下はモデルの対象外のため、本計算では固相及び液相(L2)の自由エネルギーは $1.1 < O/U < 2.1$ の範囲でのみ Blackburn モデルで計算し、この範囲外の自由エネルギーは外挿により求めているため、自由エネルギーの微係数、即ち μ が不連続となったことによる。

・計算結果は C. Gueneau 等の測定結果と一致している。

5.3 Pu-0 系状態図

5.3.1 Aitken 等による Pu-0 及び U-Pu-0 系状態図の測定結果

・fig5-21に Aitken 等⁽⁵⁾により報告されている Pu-0 系及び U-Pu-0 系の固・液相線を Pu 含有量をパラメータとして O/M の関数として示す。但し Aitken 等の報告している PuO_2 の融点は 2445C と fig4-1 示した Lyon 等の 2390C より 55 度高い。又、 PuO_2 の融点は E.H.P.Cordfunk 監修のテキスト⁽¹³⁾では 2390C、I.Barin 監修の熱力学テーブル⁽¹⁴⁾では 2440C とまちまちである。この原因として E.H.P.Cordfunk 監修のテキスト⁽¹³⁾では、 PuO_2 は溶融前に酸素を放出し低 O/M の酸化物に変化するためとしている。ここでは、Aitken 等のデータを 55 度下げることにより、 $O/M = 2.0$ の固・液相温度が Lyon 等の結果と良く一致することから、55 度の補正を加えた温度をプロットした。

・これらの測定結果を U-0 系と比較すると、U-0 系では固相線温度が O/U が 0.1 減少すると約 100C 低下するのに対して、Pu を含んだ系は O/M が変化しても固・液相線温度は粗一定である。

・本節では Pu-0 系の状態図の解析を行い、次節では本節の結果と 5-2 節で述べた U-0 系の結果を基に U-Pu-0 系の状態図の解析を実施し、Aitken 等による測定結果を評価する。

5.3.2 Pu-0 系状態図の計算

・固相 Pu-0 系の自由エネルギーは 5.2.1~5.2.2 で述べた相互作用パラメータ w を用いる方法で計算した。fig5-22~5-23 に Pu-0 系の w と自由エネルギーを O/Pu の関数として示す。

・Aitken 等による Pu-0 系の状態図は U-0 系とは異なり、 O/M の減少に伴い固・液相線温度が若干増加し、 O/M が 1.8 付近で極大値となり、さらに O/M が減少すると固・液相線温度は低下する。これは fig4-7 に示したように Ag-Au 系にて、固相の相互作用パラメータ w を液相のそれより若干負の値とした状態図と類似している。

・固相と液相が平衡状態にあるとき、両相のケミカルポテンシャルが等しいことから、

$$P_{O_2}^{S_{O_2}}(O/M^S) = P_{O_2}^{L_{O_2}}(O/M^L)$$

が成立する。この関係を Pu-0 系に適用すると、

$$O/M \text{ が } 1.8 \text{ より大きい場合} \quad \text{-----} \quad O/M^S < O/M^L \quad \text{及び} \quad P_{O_2}^{S_{O_2}}(O/M) > P_{O_2}^{L_{O_2}}(O/M)$$

$$O/M \text{ が } 1.8 \text{ より小さい場合} \quad \text{-----} \quad O/M^S > O/M^L \quad \text{及び} \quad P_{O_2}^{S_{O_2}}(O/M) < P_{O_2}^{L_{O_2}}(O/M)$$

の関係が成り立つ。

・液相の酸素分圧も 5-1 節で述べた Blackburn モデルで記述できると仮定すると、液相のイオン間平衡定数 K^L は Pu-0 系の固・液相線を最も良く再現するように

$$K_3^L / K_3 = \exp(f_L^{21})$$

$$K_4^L / K_4 = \exp(f_L^{22})$$

で表されるイオン間平衡定数比 f_{21}^L 及び f_{22}^L を決めることにより、液相の酸素分圧が計算でき、5.2.2

節で述べた方法で自由エネルギーを評価し得る。

・ fig5-24 に f_L^{21} と f_L^{22} して以下の値を用いて計算した固相線・液相線を Aitken 等により報告された測定結果と共に示す。Pu-0 系の測定点数が少ないことから、U-0 系で評価された f_L^{11} と比べて、 f_L^{21} 及び f_L^{22} は信頼性が劣ると推定される。

$$f_L^{21} = -0.35$$

$$f_L^{22} = 2.0 + (2743.0 - T_k) * 8.89 * 10^{-3}$$

・これらのパラメータを用いたときの固・液相の自由エネルギーを O/M の関数として fig5-25 に、O/M=1.94 近傍の拡大図を fig5-26 に示す。自由エネルギー曲線は O/M がおよそ 1.7 と 1.94 の 2 点で交差しており、O/M の低い側から液相、固相、液相の順番で相が表れ、各相の境界で固・液 2 相領域の存在を示している。

5.4 U - Pu - O 系状態図の計算

5.4.1 系の自由エネルギーの評価方法

・この系の自由エネルギーは、Pitzer & Brewer⁽¹⁵⁾ のテキストで紹介されている Darken の考え方で計算した。Darken は「多成分溶液は、溶質の部分モル量の積分を他の成分比が固定されている pseudo-binary line に沿って行なうならば、他成分は純溶媒と考えることができる」

との原理を応用した。この原理に従うと、一つの成分の部分モル量が全組成にて既知であれば他の全ての成分の部分モル量の計算が可能である。以下その手法を紹介する。

・次のような関数を定義する。

$$t = n_{Pu} / (n_U + n_{Pu}) = x_U / (x_U + x_{Pu}) \quad \text{----- (5-14)}$$

$$s_2 = n_{Pu} / (n_O + n_{Pu}) = x_{Pu} / (x_O + x_{Pu}) \quad \text{----- (5-15)}$$

$$s_3 = n_U / (n_O + n_U) = x_U / (x_O + x_U) \quad \text{----- (5-16)}$$

・これらの関数の組成による偏微分は

$$\left(\frac{s_2}{n_O} \right)_{n_U, n_{Pu}} = - n_{Pu} / (n_O + n_{Pu})^2$$

$$\left(\frac{s_3}{n_O} \right)_{n_U, n_{Pu}} = - n_U / (n_O + n_U)^2$$

$$\left(\frac{t}{n_U} \right)_{n_O, n_{Pu}} = - n_{Pu} / (n_U + n_{Pu})^2$$

$$\left(\frac{t}{n_{Pu}} \right)_{n_O, n_U} = n_U / (n_U + n_{Pu})^2$$

と書ける。

・自由エネルギーの全微分

$$dG = \mu_O dn_O + \mu_U dn_U + \mu_{Pu} dn_{Pu}$$

から

$$\left(\frac{\mu_O}{n_U} \right)_{n_O, n_{Pu}} = \left(\frac{\mu_U}{n_O} \right)_{n_U, n_{Pu}}$$

と書け、かつ n_U と n_{Pu} が一定は t が一定に対応するので

$$\begin{aligned} \left(\frac{\mu_U}{s_2} \right)_t &= \left(\frac{\mu_U}{n_O} \right)_{n_U, n_{Pu}} \left(\frac{n_O}{s_2} \right)_{n_U, n_{Pu}} \\ &= \left(\frac{\mu_O}{n_U} \right)_{n_O, n_{Pu}} \left(\frac{n_O}{s_2} \right)_{n_U, n_{Pu}} \\ &= \left(\frac{\mu_O}{t} \right)_{n_O, n_{Pu}} \left(\frac{t}{n_U} \right)_{n_O, n_{Pu}} \left(\frac{n_O}{s_2} \right)_{n_U, n_{Pu}} \\ &= \left(\frac{\mu_O}{t} \right) s_2 (n_O + n_{Pu})^2 / (n_U + n_{Pu})^2 \\ &= (t/s_2)^2 \left(\frac{\mu_O}{t} \right)_{s_2} \quad \text{----- (5-17)} \end{aligned}$$

同様に

$$(\mu_{Pu}/s_3)_t = -((1-t)/s_3)^2(\mu_0/t)_{s_3} \quad \text{-----(5-18)}$$

・積分すると、

$$\mu_U(s_2, t) - \mu_U(s_2^0, t) = \int_{s_2^0}^{s_2} (t/s_2)^2 (\mu_0/t)_{s_2} ds_2 \quad \text{-----(5-19)}$$

$$\mu_{Pu}(s_3, t) - \mu_{Pu}(s_3^0, t) = - \int_{s_3^0}^{s_3} ((1-t)/s_3)^2 (\mu_0/t)_{s_3} ds_3 \quad \text{-----(5-20)}$$

となり、酸素のケミカルポテンシャルから、U 及び Pu のケミカルポテンシャルを計算できる。

・また、O, U 及び Pu のケミカルポテンシャルは

$$\mu_0 = \mu_0^* + RT \ln P_0 \quad d\mu_0 = RT d \ln P_0$$

$$\mu_U = \mu_U^* + RT \ln a_U \quad d\mu_U = RT d \ln a_U$$

$$\mu_{Pu} = \mu_{Pu}^* + RT \ln a_{Pu} \quad d\mu_{Pu} = RT d \ln a_{Pu}$$

のように書けるから、

$$\ln a_U(s_2, t) - \ln a_U(s_2^0, t) = \int_{s_2^0}^{s_2} (t/s_2)^2 (\ln P_0/t)_{s_2} ds_2 \quad \text{-----(5-21)}$$

$$\ln a_{Pu}(s_3, t) - \ln a_{Pu}(s_3^0, t) = - \int_{s_3^0}^{s_3} ((1-t)/s_3)^2 (\ln P_0/t)_{s_3} ds_3 \quad \text{-----(5-22)}$$

・fig5-27 に示すように、s₂一定線上で t が t 変化したときの酸素のケミカルポテンシャル変化

$$\begin{aligned} (\ln P_0/t)_{s_2} &= (\mu_0^2 - \mu_0^1)/t \\ &= \{ \mu_0^2(s_2, t=t_0 + t/2) - \mu_0^1(s_2, t=t_0 - t/2) \} / t \end{aligned} \quad \text{-----(5-23)}$$

を、t一定線に沿って s₂ で積分することにより、基準組成(s₂⁰)からの U のケミカルポテンシャルの変化量を計算することができる。Pu のケミカルポテンシャルについても同様。上記の積分はケミカルポテンシャルの計算精度を向上させるため、”未定係数法による数値積分”⁽¹²⁾を用いた。

・fig5-27における、O-U線上及びO-Pu線上のU, Puのケミカルポテンシャルは2成分系でのGibbs-Duhem式

$$n_0 d\mu_0 + n_U d\mu_U = 0 \quad \text{-----(5-24)}$$

$$n_0 d\mu_0 + n_{Pu} d\mu_{Pu} = 0 \quad \text{-----(5-25)}$$

の積分より求めた。この方法にて得た U 及び Pu のケミカルポテンシャルは 5.2.2 節にて述べた方法にて計算したものと一致する。

・従って、Blackburn モデルによって U-Pu-O 系の酸素分圧と基準組成における U 及び Pu のケミカルポテンシャルが与えられるならば、系の自由エネルギーの評価が可能である。

5.4.2 基準組成の U 及び Pu のケミカルポテンシャル

・組成 (U, Pu)O_{2.0} は

UO₂ 及び PuO₂ の自由エネルギーが既知である

4.1.1 節で述べたように、UO₂ - PuO₂ 系を理想溶液と仮定すると、混合物の自由エネルギーを容易に評価できる

ことから基準組成とした

・UO₂ 及び PuO₂ のモル当たりの自由エネルギー g は

$$g_{UO_2} = 2\mu_{O_0}^{UO_2} + \mu_U = 2(\mu_{O_0}^* + RT \ln P_0^{UO_2}) + \mu_U^* + RT \ln a_U^{\#} \quad \text{----- (5-26)}$$

$$g_{PuO_2} = 2\mu_{O_0}^{PuO_2} + \mu_{Pu} = 2(\mu_{O_0}^* + RT \ln P_0^{PuO_2}) + \mu_{Pu}^* + RT \ln a_{Pu}^{\#} \quad \text{----- (5-27)}$$

ここで

$\mu_{O_0}^{UO_2}$ 及び $\mu_{O_0}^{PuO_2}$ は UO_2 及び PuO_2 中の原子状酸素のケミカルポテンシャル

μ_U 及び μ_{Pu} は U 及び Pu のケミカルポテンシャル

$\mu_{O_0}^*$ 、 μ_U^* 及び μ_{Pu}^* は、それぞれの元素の標準状態でのケミカルポテンシャル

$P_0^{UO_2}$ 及び $P_0^{PuO_2}$ は UO_2 及び PuO_2 中の原子状酸素の分圧

$a_U^{\#}$ 及び $a_{Pu}^{\#}$ は UO_2 及び PuO_2 中の U 及び Pu の活量

・ $xUO_2 + (1-x)PuO_2$ の理想溶液混合物を考えると、混合物の自由エネルギーは

$$\begin{aligned} G_{\text{mix}}^S &= xg_{UO_2} + (1-x)g_{PuO_2} + RT\{x \ln x + (1-x) \ln(1-x)\} \\ &= 2\mu_{O_0}^* + 2RT\{x \ln P_0^{UO_2} + (1-x) \ln P_0^{PuO_2}\} + x\mu_U^* + (1-x)\mu_{Pu}^* \\ &\quad + RT\{x \ln(a_U^{\#} x) + (1-x) \ln(a_{Pu}^{\#} (1-x))\} \quad \text{----- (5-28)} \end{aligned}$$

・ 上記混合物を 3 成分系として一般化して記述すると

$$G_{\text{mix}} = x_0(\mu_{O_0}^* + RT \ln P_0) + x_U(\mu_U^* + RT \ln a_U) + x_{Pu}(\mu_{Pu}^* + RT \ln a_{Pu}) \quad \text{----- (5-29)}$$

ここで、 x_0, x_U, x_{Pu} は原子状酸素、ウラン、プルトニウムのモル分率。

$UO_2 - PuO_2$ 系は $x_0 = 2/3$, $x_U + x_{Pu} = 1/3$ であり、且つ $G_{\text{mix}}^S = 3G_{\text{mix}}$, $x = 3x_U$ であるから

$$\begin{aligned} &2\{x \ln P_0^{UO_2} + (1-x) \ln P_0^{PuO_2}\} + x \ln(a_U^{\#} x) + (1-x) \ln(a_{Pu}^{\#} (1-x)) \\ &= 2 \ln P_0 + x \ln a_U + (1-x) \ln a_{Pu} \quad \text{----- (5-30)} \end{aligned}$$

この式の左辺は x のみの関数であり、右辺は x の関数である P_0, a_U, a_{Pu} を含んでいる。ここで左辺を $K(x)$, 右辺を $J(x, P_0, a_U, a_{Pu})$ とする。

・ 右辺の全微分は

$$dJ = 2d(\ln P_0) + (\ln a_U) dx + x d(\ln a_U) - (\ln a_{Pu}) dx + (1-x) d(\ln a_{Pu}) \quad \text{----- (5-31)}$$

・ Gibbs-Duhem の式

$$x_0 d \ln P_0 + x_U d \ln a_U + x_{Pu} d \ln a_{Pu} = 0$$

を、上記と同様に $x_0 = 2/3$, $x_U + x_{Pu} = 1/3$ 、 $x = 3x_U$ の関係を用いて書き換えると

$$2d \ln P_0 + x d \ln a_U + (1-x) d \ln a_{Pu} = 0$$

を (5-31) に代入すると

$$dJ/dx = \ln(a_U/a_{Pu}) \quad \text{----- (5-32)}$$

の関係が得られる。

・ x のみの関数である $K(x)$ の微分は

$$dK(x)/dx = 2 \ln(P_0^{UO_2} / P_0^{PuO_2}) + \ln(a_U^{\#} / a_{Pu}^{\#}) + \ln(x/(1-x))$$

となるから

$$\ln(a_U/a_{Pu}) = 2 \ln(P_0^{UO_2} / P_0^{PuO_2}) + \ln(a_U^{\#} / a_{Pu}^{\#}) + \ln(x/(1-x)) \quad \text{----- (5-33)}$$

・ (5-30) 式と (5-33) 式より a_U 及び a_{Pu} と酸素分圧の関係は

$$P_0^2 a_{Pu} = (P_0^{PuO_2})^2 a_{Pu}^{\#} (1-x) \quad \text{----- (5-34)}$$

$$P_0^2 a_U = (P_0^{UO_2})^2 a_U^{\#} x \quad \text{----- (5-35)}$$

となり、 $UO_2 - PuO_2$ 系の酸素分圧が得られるならば、 U 及び Pu の活量を計算できるとともに、 UO_2 及び PuO_2 の活量を

$$a_{UO_2} = P_0^2 a_U \quad \text{----- (5-36)}$$

$$a_{\text{PuO}_2} = P_0^2 a_{\text{Pu}} \quad \text{----- (5-37)}$$

と書くと、 UO_2 - PuO_2 系は理想溶液となり、先の前提を満足している。

- ・以上の手法で基準組成のU及びPuのケミカルポテンシャルを評価した。

5.4.3 液相 U-Pu-O系のモデル

・液相U-Pu-O系の酸素分圧の計算には測定されたU-O系及びPu-O系状態図を計算上良く再現するように設定したイオン間平衡定数比 f_L^{11} , f_L^{21} 及び f_L^{22} をそのまま用いた。

・Blackburn モデルでは、U-Pu-O系の酸素分圧はUイオンと O^{2-} イオン及びPuイオンと O^{2-} イオン間の平衡と電気的中性の条件によって決定される。このときのイオン間の平衡定数は相互に無関係・独立、すなわちUイオンと O^{2-} イオン間の平衡定数はPuイオンの有無とは無関係に定義されている。fig5-4に示すように、このような考え方のモデルで固体U-Pu-O系の酸素分圧挙動を説明し得るので、液体U-Pu-O系についてもこの考え方が成り立つであろう。

5.4.4 U-Pu-O系の自由エネルギー

・fig5-28及び5-29にU-Pu-O系の2600Cにおける固相、液相の自由エネルギーをO/M及びPu/(U+Pu)の関数として示す。自由エネルギーは組成に対して2次関数的変化をし、U-O及びPu-O系と類似している。

・系の自由エネルギーおよびケミカルポテンシャルをPu/(U+Pu) = 0~1の間201組成、O/M=1.1~2.1の間1001組成についてあらかじめ求めテーブル化し、状態図計算及び相間のケミカルポテンシャル一致の確認に使用した。

5.4.5 U-Pu-O系状態図解析結果

・fig5-30に一定温度(2600C)での固・液相線のO/MとPu/(U+Pu)の関係を表す等温状態図(3成分系状態図は3次元の立体となり、その等温断面図)を示す。この状態図はU-O及び UO_2 - PuO_2 状態図の固・液相線と2600C定温線の交点を起点とし、起点間の固・液相線のO/MとPu/(U+Pu)の関係を図示したものである。

・U-O及びPu-O系状態図で最適化した f_L^{11} , f_L^{21} 及び f_L^{22} を用いてU-Pu-O系状態図を計算した場合(以下「最適化ケース」と呼ぶ) Pu/(U+Pu)=0、即ちU-O系状態図を起点とした固・液相線はPu濃度が増加してもO/Mはほとんど変化せず、O/M=2.0、即ち UO_2 - PuO_2 系状態図を起点とした固・液相線はO/Mが低下してもPu/(U+Pu)比はほとんど変化しない、「逆L型」の状態図となった。

・図中の測定点はfig5-21に示したAitken等の限られた測定結果から内外挿により求めたものである。O/Mが1.84より高い測定結果とは比較的良く一致しているが、O/Mがそれより低い測定結果との一致は悪い。この原因としては

測定データの面からは、O/Mの低い場合の測定値の数が少なく内外挿の精度が悪い

解析法の面からは、Pu-O系の低O/Mでの状態図の情報が少ないため、イオン間平衡定数比 f_L^{22} の信頼性に乏しい

ことがあげられる。

・fig5-30に示す破線は2相領域での共存する組成間を結んだ平衡連結線(tie line)を示している。この図より、低O/M MOX燃料で2相が共存する場合、

$$(O/M)^S > (O/M)^L$$

$$\{Pu/(U+Pu)\}^S < \{Pu/(U+Pu)\}^L$$

となっている。

・2相間のケミカルポテンシャルの差異を(4-5)式で評価したものを fig 5-31 及び fig 5-32 に示す。U-0系と比べて2桁大きくなっている。これは、U-0系と比較すると

自由エネルギー極小を探索するパラメータの増加

系の組成、O/M 及び Pu/(U+Pu)、に対応する自由エネルギーをテーブルから求めるとき、及び

自由エネルギーの極小に対応する組成が求まった後にケミカルポテンシャルの一致の度合い

を確認する際に Lagrange 補間法を繰り返し使用すること

に因ると考えられる。又、 $\log(\quad)$ が-4より大きくなっている点は O/M=2.0 及び Pu/(U+Pu)=0 近傍に対応し、組成の変動に対するケミカルポテンシャルの変動の割合が大きいことによる。

・fig 5-33-5-35 に 2500C~2800C の等温状態図を示す。固・液相線は 2600C と同様「逆 L 型」となった。解析結果は低 O/M 領域で測定結果との一致が悪いのは、5.2.3 で U-0 系で述べたと同様の原因と思われる。

・測定結果と直接の比較を目的として Pu/(U+Pu)比が一定の状態図の計算を行った。fig 5-36-5-38 に Aitken 等により測定された Pu/(U+Pu)=0.2, 0.4 及び 0.6 の組成における固・液相線の O/M 依存性との比較を示す。解析結果は測定値と固・液相線温度の絶対値が若干異なるものの、固・液相線温度の O/M 依存性が極めて小さいことを良く表している。又、Pu センター技術開発室で測定された Pu/(U+Pu) = 0.28 における固・液相線の O/M 依存性の結果⁽¹⁶⁾との比較を fig 5-39 に示す。液相線の解析結果は測定値と比べて低めであるが、解析結果の O/M 依存性は測定値と良く一致している。

・U-0 及び Pu-0 系状態図で最適化したイオン間平衡定数比 f_L^{11} , f_L^{21} 及び f_L^{22} を用いて U-Pu-0 系状態図を計算した「最適化ケース」は、U-Pu-0 系状態図を良く再現した。

5.5 考察

5.5.1 U-Pu-0 系の熱力学的挙動

・U-Pu-0 系を UO_{2-x} と PuO_{2-x} の混合物とみなすと、混合物中の構成化合物の活量は(5-36)及び(5-37)より

$$a_{UO_{2-x}} = a_U(P_O)^{2-x}$$

$$a_{PuO_{2-x}} = a_{Pu}(P_O)^{2-x}$$

と書ける。fig5-40 及び fig5-41 に O/M = 2.0 と 1.7 における、 UO_{2-x} , PuO_{2-x} の活量、を Pu/(U+Pu)の関数として示す。O/M=2.0 では組成に対して直線となり 5.4.2 節で仮定した理想溶液の挙動を示しており、O/M=1.7 では理想溶液から負側にずれている。何ゆえ負がわにずれるか詳細は不明であるが、高濃度では Raoult の法則、低濃度では Henry の法則に従い妥当な結果である。

・各成分のケミカルポテンシャルは U と Pu の化学的性質の違いを反映して興味深い挙動を示す。fig5-42 に O/M = 2.0 における酸素分圧、U 及び Pu の活量を Pu/(U+Pu)の関数として示す。酸素分圧は Pu 濃度の増加につれて増加、U の活量は Pu 濃度の減少に従い増加、と予想された挙動を示すが、Pu の活量は Pu 濃度の増加と共に減少する特異な挙動を示している。これは UO_2 と PuO_2 の酸素分圧が大きく異なるため、Pu 濃度が増加するに従い酸素分圧が大きく増加し、(5-34)~(5-37)式より求められる Pu の活量が常識とは異なり減少したものと考えられる。fig5-43 に O/M=1.7 での酸素分圧、活量の挙動を示す。Pu/(U+Pu)比が 0.1 以下では Pu の活量が Pu 濃度の増加に従い増える常識的な挙動となっている。

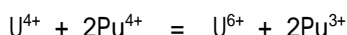
・ fig5-44 に Pu の等活量線を示す。Pu の活量は先に述べたことに加えて、低 O/M では Pu/(U+Pu) が 0.3 近傍で極小値を持つ等、特異な挙動を示している。

・ 5 章で述べた Blackburn モデルでは、 UO_{2-x} と PuO_{2-x} を混合するとマスバランスと電気的中性の条件を満たすように電荷の再配分が起こり、その結果 Pu は還元され U は酸化される。この電荷の再配分の挙動を fig5-45 に示す。例えば、 $UO_{1.7}$ と $PuO_{1.7}$ を 1:1 の割合で混合すると、混合前の U 及び Pu の原子価は 3.4 であるが、混合後 U は 4.0 近くまで酸化され、Pu は 2.8 まで還元される。従って、Pu/(U+Pu) が減少するにもかかわらず Pu の活量は増加する。しかし O/M=1.7 の場合、Pu/(U+Pu) 比が 0.2 以下では Pu の還元が頭打ちおなり、活量に対する濃度の項、すなわちエントロピーの寄与が大きくなり、Pu 濃度の増加に従い活量が増えていると考えられる。

5.5.2 UO_2 - PuO_2 系は理想溶液か

・ 前節で述べたように、 UO_2 - PuO_2 系は混合により U と Pu の間で電荷の移動が生じ、活量が濃度と等しい理想溶液モデルが適用可能かの疑問が生ずることから、以下の検討を行った。

・ UO_2 と PuO_2 を混合すると



の反応が起こると考えると、上記の反応熱は UO_2 , UO_3 , PuO_2 及び Pu_2O_3 の生成熱から約 317kJJou と推定される。O/M が 2.0 での上記反応の進行割合は fig5-45 に示した平均原子価の結果から約 1.5%程度であるから、正則溶液モデルにおける相互作用パラメータ w は 7kJJou/mole 程度となる。

・ この項の自由エネルギーへの寄与は(5-8)式より $UO_2:PuO_2 = 1:1$ の組成で 1.75kJJou であり、エントロピーの項 $RT\{x \ln x + (1-x) \ln(1-x)\}$ は 2800K にて -32kJJou であることから w の寄与は約 5%と小さく、 UO_2 - PuO_2 系は理想溶液で近似することは妥当と判断される。

・ fig5-46 に、固・液相とも同一の相互作用パラメータ ($w = 7kJJou$) にて記述し得る正則溶液とした場合の状態図を示す。比較のため、理想溶液モデル及び固・液相間で相互作用パラメータが $w^s = 7kJJou$, $w^l = 0$ と異なる場合を併記した。相互作用パラメータがエントロピー項と比べて小さく、かつ固・液相間で等しいならば理想溶液モデルとほぼ等しい状態図となる。

・ 仮に、 UO_2 - PuO_2 系が正則溶液モデルで記述されるならば、5.4.2 節の結論は

$$P_o^2 a_{Pu} = (P^{PuO_2}_o)^2 a_{Pu}^{\#} (1-x) \exp(x^2 w / RT)$$

$$P_o^2 a_U = (P^{UO_2}_o)^2 a_U^{\#} x \exp\{(1-x)^2 w / RT\}$$

と変形できる。fig 5-47 に $w^s = w^l = 7kJJou$ とした場合の、2600C での U-Pu-O 系の状態図を理想溶液モデル及び $w^s = 7kJJou$, $w^l = 0$ の場合と共に示す。相互作用パラメータが固・液両相で等しくかつエントロピー項と比べて小さいと、2 相領域の幅が若干変動するが状態図は UO_2 - PuO_2 系を理想溶液とした場合とほぼ一致し、相間で異なると状態図は大きく変化する。

・ 以上のことから、 UO_2 - PuO_2 系は相互作用パラメータが $w^s = w^l = 7kJJou/mole$ と両相間で等しく、かつエントロピー項と比べて小さいことから、理想溶液と考えて差し支えない。

5.5.3 Blackburn モデルに基づく状態図計算の妥当性

・ Blackburn による酸素分圧モデルは U 及び Pu の酸化還元平衡に基礎を置き直感的に分かり良く、かつイオン間の高次方程式を解くことにより容易に酸素分圧が求まり、本報告で述べたような状態図計算に容易に応用可能である。

・本モデルは、fig5-1-5-4 に示したように酸素分圧の計算値には一桁程度の不確かさがあるので、本節ではこの不確かさと状態図の計算結果の関係を検証する。

・(5-1)式は O/U が 2.0 以下の酸素分圧を制御する平衡定数で、

$$P_{O_2} = (K_1)^2$$

が近似的に成り立つ。fig5-48 と 5-49 に P_{O_2} が 10 倍及び 10^4 倍に成るよう K_1 にファクターを乗じた場合の状態図の計算結果を示す。ファクターが 10 程度では状態図の計算結果はほとんど変化しない。又、ファクターが 10^4 の場合でも U-0 系では状態図はほとんど変化しないが、U-Pu-0 系では状態図が大きく変化した。これは、次の Pu-0 系で K_3 と K_4 の一方を変化させたとき状態図が変わるのと同様、U-Pu-0 系では K_3 及び K_4 と K_1 の間の相対的なバランスが変化したことによる。

・Pu-0 系の同様の計算結果を fig5-50 に示す。 K_3 及び K_4 に共通にファクター 10 を乗じた場合 Pu-0 系の状態図はほとんど変化がなく、 K_4 にのみファクターを乗じた場合状態図は大きく変化した。

・これらのことは、2 章で述べた相平衡の条件（各成分のケミカルポテンシャルは相間で等しい）を状態図計算によって示したものと等しい。即ち相間のケミカルポテンシャルが変化しないようにファクターを選べば状態図は変化せず、逆の場合は状態図は変化する。

・以上のことから、Blackburn モデルの酸素分圧計算の不確かさは状態図計算に与える影響は小さい。しかし酸素分圧の組成(O/M 等)依存性を変化させると状態図は変わる。その時はイオン間平衡定数比の見直しが必要となろう。

5.5.4 イオン間平衡定数比

(1) イオン間平衡定数比の評価

・液相の酸素分圧は固相に対して構築された Blackburn による酸素分圧モデルのイオン間平衡定数に U-0 系及び Pu-0 系固・液状態図を再現するようにイオン間平衡定数比を乗ずることにより求めた。

・イオン間平衡定数比 $f_L^{11} \sim f_L^{22}$ と液相の固・酸素分圧比の関係は

$$f_L^{11} = (1/2) \ln (P_{O_2}^L / P_{O_2}^S), \quad f_L^{21} \text{ 及び } f_L^{22} \text{ についても同様}$$

である。fig 5-51 に上記 3 パラメータと温度の関係を図示した。

・これらのイオン間平衡定数比を固相での平衡定数に乗じて液相の平衡定数を計算し、それを用いて液相の酸素分圧、ケミカルポテンシャルを算出することから、状態図への感度は高い。

・ f_L^{11} は Latta 等による U-0 系の状態図測定に基づいており信頼性は高い (fig5-13 参照)。

・ f_L^{21} 及び f_L^{22} は Aitken 等による Pu-0 系の状態図測定に基づいている (fig5-24 参照)。 f_L^{21} は Pu-0 系の O/Pu が約 1.8~2.0 の間の酸素分圧比に効果的で、固・液相温度と O/Pu の関係から -0.35 の一定値を用いた。 f_L^{22} は O/Pu が 1.8 近傍の状態図の極大と O/Pu~1.6 の固相線温度 2245C に計算結果が合致するように定めた。その結果、2245C において f_L^{22} は約 4.0 と極めて大きな値となり、その温度依存性も他のパラメータと異なる傾向となった。同一組成の Pu-0 系で固相・液相間で 3000 倍近く酸素分圧が異なるとは考えにくく、 f_L^{22} には大きな誤差が含まれているものと考えられる。従って Pu-0 系状態図について新たな情報が得られたならば f_L^{22} の修正 (Pu-0 系の状態図は f_L^{21} と f_L^{22} のバランスで決まることを考えに入れると f_L^{21} についても同様) が必要であろう。

(2) f_L^{21} の感度

・ f_L^{22} は変更せず、 f_L^{21} のみを変えた場合の Pu-0 系の状態図の変化を fig 5-52 に示す。case S は 5.3.2

節で述べた Pu-0 系の状態図を最も良く再現するパラメータの組を用いたもので、 $f_L^{21}=-0.35$ である。 f_L^{21} が負、 f_L^{22} が正であるため、高 O/Pu 側で固・液相線温度を高める効果と、低 O/Pu 側で固・液相線温度を下げる効果のせめぎあい、O/Pu=1.8 にて固・液相線の極大が生じている。case A は $f_L^{21}=0.0$ とすることにより、液相酸素分圧が f_L^{22} のみの効果で固相より小さくなり、固・液相線は O/Pu の減少につれ徐々に低下している。case B は $f_L^{21}=-0.7$ と case S より負側に大きくしているため、固・液相線の極大が O/Pu 側にずれ、かつ温度が高くなっている。

・上記 3 ケースで使用したパラメータを用いて、2600C 及び 2450C での U-Pu-0 系の状態図計算を行った結果を fig5-53-5.54 に示す。両図とも case A では固相領域が減少し、case B では固相領域が増加と、Pu-0 系状態図の結果と整合が取れている。また fig5-54 の Pu/(U+Pu)=1.0 の線上で O/M=1.56~1.84 の範囲が固相領域であるが、これは fig5-52 の 2450C での固相領域と一致している。

(3) f_L^{22} の感度

・ f_L^{22} は fig5-51 に示したように大きな温度依存性を持たざるを得なかった。しかし感度解析上は解析結果が複雑になるので、2600C でおおよそ 1.0 であることからこれを基準とし (case D)、 $f_L^{22}=0.0$ (case E)、 $f_L^{22}=0.5$ (case F) 及び $f_L^{22}=2.0$ (case G) の計算結果を fig5-55 及び 5-56 に示す。いずれのケースも、 f_L^{21} は -0.35 とした。

・fig5-55 の Pu-0 系では O/Pu が 1.8 以上では D-F のいずれの場合でも、固・液相線は O/Pu の減少につれてほとんど同じ勾配で増加している。これは D-G では、高 O/Pu 側で有効な f_L^{21} として -0.35 と一定値を用いたためであり、fig5-56 の U-Pu-0 系の状態図の D-G にて O/M=2.0 近傍の固・液相線の O/M と Pu/(U+Pu) の関係が fig5-53 とは異なり全てのケースで同一であることと対応している。すなわち、O/M=2.0 近傍の状態図は f_L^{21} によって決まる。

・低 O/M の状態図は f_L^{21} と f_L^{22} の相対的大きさによって決まる。

5.5.5 等組成状態図

・ f_L^{11} ~ f_L^{22} のパラメータを最適化ケースに設定した場合での、O/M が 1.75~2.00、Pu/(U+Pu) が 0~0.6 の範囲の等組成状態図 (3 次元立体状態図の Pu/(U+Pu) または O/M 一定の断面図) を fig5-57 と fig5-58 に示す。

・fig5-57 は Pu/(U+Pu) 一定の等組成状態図で既に fig5-36-5-39 に Aitken らの測定結果の比較のためその一部を示してある。様々の Pu 濃度の固・液相線は、O/M=2.0、即ち UO₂-PuO₂ 系状態図の固・液相線温度を起点とし、O/M の低下にも係らず一定温度を保持し、O/M が U-0 系の 2 相領域に相当する O/M まで低下すると、固相線温度は急激に低下し、液相線は U-0 系の液相線と重なり合って低下する。

・一方、O/M 一定の等組成状態図、fig5-58 では、Pu/(U+Pu)=0、即ち U-0 系状態図の固・液相線温度を起点とし、固相線温度は融点の低い PuO_{2-x} 濃度の増加にもかかわらず増加し、液相線温度は粗一定に保持されている。そして、UO₂-PuO₂ 系の 2 相領域まで PuO_{2-x} 濃度が増加すると、それ以上の濃度では固・液相線とも UO₂-PuO₂ 系の固・液相線と重なり合って低下する。

・以上 2 つの等組成状態図は 5.4.5 節で述べた等温状態図が「逆 L 型」であることと符合する。

・このことをより一層明確にするため、Pu-0 系状態図が U-0 系状態図 (fig5-13 参照) と類似の固・液相挙動を示すように、 $f_L^{21}=1.0$ 、 $f_L^{22}=0.0$ のケースの状態図計算を行った (case U)、fig5-59 に Pu-0 系状態図の計算結果を最適化ケースと比較して示す。固・液相線が U-0 系と同様 O/Pu の低下と共に単

調に低下している。fig5-60 に同様のケースの 2600C での U-Pu-0 系等温状態図を示す。等温状態図は逆 L 型にはならず、「/ 型」となる。さらに fig5-61 及び 5-62 に示した等組成状態図では、固・液相線温度は Pu 濃度の増加、O/M の減少にしたがって単調に減少し、最適ケースと比べて固・液相線挙動は大きく異なったものとなる。

- ・最適化ケースと case U の等温及び等組成状態図を比較すると、
最適化ケースの等温状態図で O/M 比が 2.0 から減少するにつれて、固・液相線の Pu/(U+Pu) が増加するのは、Pu/(U+Pu) が一定の等組成状態図にて O/M が 2.0 から減少するに従い固・液相線温度が増加するのに相当
最適化ケースの等温状態図で Pu/(U+Pu) 比が 0 から増加するにつれての固・液相線の O/M が減少するのは、O/M が一定の等組成状態図にて Pu/(U+Pu) が増加するに従い固・液相線温度が増加するのに相当

することが判る。

- ・これらの現象は、既に Aitken 等が指摘しているように、O/U の減少に従い単調に固・液相線温度が低下する U-0 系に、固相の安定化(又は液相の不安定化)を引き起こす Pu-0 系を加えることにより、U-Pu-0 系固相の安定化(又は液相の不安定化)が生じていることに対応する。
- ・又、蛇足ながら付け加えておくと、U-Np-0 系状態図は Np-0 系状態図⁽¹⁷⁾が U-0 系状態図と類似していることから「/ 型」となり、U-Am-0 系状態図は Am-0 系の固・液相線が不明であるが低温の状態図が Pu-0 系と類似していることから、それと類似の固・液相線と推定すると、「逆 L 型」と予想される。

5.5.6 Aitken 等の測定結果の評価

- ・U-0 及び Pu-0 系固・液状態図を用いて最適化を図ったイオン間平衡定数比 f_L^{11} , f_L^{21} 及び f_L^{22} を用いて U-Pu-0 系の状態図の評価を行った。その結果は、固・液相線温度が O/M の低下と共にわずかに増加し、Aitken 等及び森本の測定結果⁽¹⁶⁾をよく説明するものであった。
- ・本状態図計算の初期段階では、U-Pu-0 系固・液相線温度は、特に Pu 濃度の小さい領域については、U-0 の固・液相線と同様単調に減少するものと予想した。しかしながら、Pu-0 系状態図において、固・液相線が小さな極大を持つような挙動が直接 U-Pu-0 系状態図に反映され、等組成状態図の固・液相線温度は O/M の低下に伴い小さな極大を持ち、かつ Pu-0 系状態図と U-0 系状態図の濃度を加味した平均的なものとなった。
- ・これは U-Pu 間の相互作用が U-0 又は Pu-0 間の相互作用と比べて著しく小さいとして無視(正確には電気的中性の条件を介した 2 次的な相互作用、5-1 節参照)している Blackburn モデルに由来するものである。5.4.3 節で述べたように Blackburn モデルでは、固体 MOX 中で Pu イオンと酸素イオン間の平衡定数は U イオンの存在の有無とは無関係に定義しており、この仮定は MOX の酸素分圧測定結果を良く説明し得ることから妥当と考えられる。この考え方は液体 MOX に対しても成立するであろうから、最適化した f_L^{21} 及び f_L^{22} を乗じた液相の平衡定数及びそれから求められる液相の酸素分圧は固体 MOX より不確定さは増大するが妥当なものと判断される。
- ・fig5-63 及び 5-64 に上記の Blackburn モデルの考え方を Ag-Au-Cu 系にて正則溶液モデルで模擬した結果を示す。ここで(4-3)及び(4-4)式にて定義される相互作用パラメータ w は、Cu を U、Au を Pu、及び Ag を Oに見立てて、以下のように設定した。

$$w_{AgAu}^S = -140 * Sx \text{ (kJoul)}$$

$$W_{AgCu}^S = -150 \cdot Qx \text{ (kJouI)}$$

$$W_{AuCu}^S = 0.0$$

$$W_{AgAu}^L = -100 \cdot Sx \text{ (kJouI)}$$

$$W_{AgCu}^L = -100 \cdot Qx \text{ (kJouI)}$$

$$W_{AuCu}^L = 0.0$$

$$Sx = -51194 \cdot x^4 + 12.353 \cdot x^3 - 10.377 \cdot x^2 + 3.1393 \cdot x + 0.7$$

$$Qx = -1.2 \cdot x^2 + 1.2x + 0.7$$

Sx 及び Qx に濃度依存性を持たせることにより、fig5-63 に見られるように、Au/(Au+Cu)=0.0 の場合、即ち Cu-Ag 系は Ag/(Au+Cu)=1.0 にて固・液相線に極大を持ち、Au/(Au+Cu)=1.0 の場合、即ち Au-Ag 系は Ag/(Au+Cu)=0.7 にて極大を持つ。

- ・ Au/(Au+Cu) が任意の値に対応する状態図は、Au-Cu 間の相互作用が無いとしているので、Au-Ag 及び Cu-Ag 状態図の濃度を加味した平均となっており、かつ fig5-64 に示した等温状態図は「逆 L」型となり、U-Pu-0 系状態図と類似の状態図となった。

- ・ 従って、U-Pu-0 系状態図は、固・液相線の極大の組成が異なる U-0 系及び Pu-0 系の混合物で、かつ U-Pu 間の相互作用が U-0 又は Pu-0 間と比べて無視し得るほど小さい場合の状態図と言える。これは状態図解析の面から、Blackburn モデルの考え方が妥当であることを示している。

- ・ Pu-0 系状態図で最適化を図ったパラメータ f_L^{21} 及び f_L^{22} を用いた計算結果は、Aitken 等による U-Pu-0 系の等組成状態図の測定結果と良く一致した (fig5-36~5-38 参照)。このことは、言い換えると Aitken らにより測定された U-Pu-0 系の固・液相線の O/M 依存性は Pu-0 系の固・液相線と自己矛盾の無い優れた測定データと言える。

- ・ しかしながら、彼等の測定結果は測定点数が少なく、かつ他の測定結果と整合を取るためには 55C の温度補正が必要であった。今後データの拡充、測定精度の向上を図る必要がある (特に Pu-0 系について)。

6. まとめ

- ・ 定温・定圧での平衡の基準 "Gibbs 自由エネルギーは極小" の原理に基づいた、2 相 2 成分及び 2 相 3 成分系状態図計算プログラムを作成した。

- ・ このプログラムを用いて、理想溶液モデルで記述される UO₂-PuO₂ 系、正則溶液モデルで記述した Ag-Au-Cu 系状態図の計算結果を評価し、解析プログラムの妥当性を確認した

- ・ U-0 及び Pu-0 系の固・液相線を最も良く再現するように設定したイオン間平衡定数比を用いて計算した U-Pu-0 系状態図は、O/M の低下に従い小さな極大を持つ Pu-0 系と類似の固・液相線となり、Aitken 等及び森本の測定結果と良く一致した。

- ・ Aitken 等による U-Pu-0 系の固・液相線の測定結果は測定精度向上を図る必要があるが、Pu-0 系の固・液相線と自己矛盾のない優れた測定データであることが確認できた。

- ・ 今後 Pu-0 系も含めて、MOX 燃料の固・液相線データの拡充、測定精度の向上を図る必要がある

7. おわりに

- ・ 定温・定圧での平衡の基準 "Gibbs 自由エネルギーは極小" の原理に基づいた、2 相 2 成分及び 2 相

3 成分系状態図計算プログラムを作成した。

・このプログラムを用いて、理想溶液モデルで記述される UO_2 - PuO_2 系、正則溶液モデルで記述した Ag-Au-Cu 系状態図の計算結果を評価し、解析プログラムの妥当性を確認した

・U-0 及び Pu-0 系の固・液相線を最も良く再現するように設定したパラメータを用いた U-Pu-0 系状態図は、O/M の低下に従い小さな極大を持つ Pu-0 系と類似の固・液相線となり、Aitken 等及び森本の測定結果と良く一致した。

・Aitken 等による U-Pu-0 系の固・液相線の測定結果は測定精度向上を図る必要があるが、Pu-0 系の固・液相線と自己矛盾のない優れた測定データであることが確認できた。

8. 謝辞

・プルトニウムセンター製造加工部設計評価 Gr 中島弘さんには WINDOWS 版 FORTRAN の使用方法等ご教授いただいた。心からお礼申し上げます。

・プルトニウムセンター製造加工部技術開発室 森本恭一さんには U-Pu-0 系の融点測定データの提供をしていただいた。感謝したい。

9. 参考文献

- (1) 長谷部光弘、小林俊雄 "コンピュータによる多相間平衡計算" 金属 1991 (7) p14
- (2) 長谷部光弘、西沢泰二 "最近の状態図に関する研究" 日本金属学会会報 11(1972)879
- (3) W.L.Lyon and W.E.Baily, "The solid-liquid phase diagram for the UO_2 - PuO_2 system" J.Nuc.Mat. 22(1967)332
- (4) R.E.Latta and R.E.Fryxell, "Determination of solidus-liquidus temperatures in the UO_{2+x} system" J.Nucl.Mat. 35(1970)195
- (5) E.A.Aitken and S.K.Evans, "Thermodynamic data program involving Plutonium and Uranium at high temperature" GEAP-5672(1968)
- (6) 長井修一郎 "化学熱力学入門及び燃料への応用" JNC TN8410 2001-022
- (7) P.E.Hansen, "Constitution of binary alloy", McGraw-Hill Book Company(1958)
- (8) 長井修一郎 " UO_2 , PuO_2 及び MOX 燃料の酸素分圧" JNC TN8410 99-019
- (9) 安達忠次、一戸明 "微分方程式の解き方" 宝文館出版 (1963)
- (10) R.J.Ackerman, E.G.Rauth, M.S.Chandrasekharaiah "A thermodynamic study of the Uranium-Uranium system" J.Physcal Chem. 73(1969)762
- (11) R.K.Edwards and A.E.Martin, "Phase relations in the Uranium-Uranium dioxide system at high temperatures" IAEA-SM-66/81(1966)423
- (12) T.R.McCalla, "計算機のための数値計算法概論"、三浦攻ほか訳、サイエンス社 (1972)
- (13) E.H.P.Cordfunke, R.J.M.Konings ed. "Thermochemical data for reactor materials and fission products" ,North-Holland(1990)
- (14) Ihsan Barin, "Thermochemical data of pure substances", VCH(1989)
- (15) K.S.Pitzer, L.Brewer, "Thermodynamics", McGraw-Hill Book Company

- (16) 森本恭一 "私信"
- (17) 大阪科学技術センター アクチド[®] 研究調査委員会編 "アクチド[®] 燃料開発にかかわる技術情報(解説編)" 2001年12月発行
- (18) C.Gueneau, V.Dauvois, P.Perodeand, C.Gonella and O.Dugne "Liquid immiscibility in (O,U,Zr) model corium" J.Nuc.Mat 254(1998)158

124-met-

Table 4-1 Ag, Au, Cu の熱力学データ

構成成分	H_m^S (Joul/mole)	S_m^S (Joue/mole/K)	融点 (K)	融解熱(Joul/mole)	C_p^S (Joul/mole/K)	C_p^L (Joul/mole/K)
Ag	26412	81.730	1233.95	11297	30	33.500
Au	29099	88.429	1337.58	12552	28	30.962
Cu	29658	74.291	1358.00	13138	28	32.844

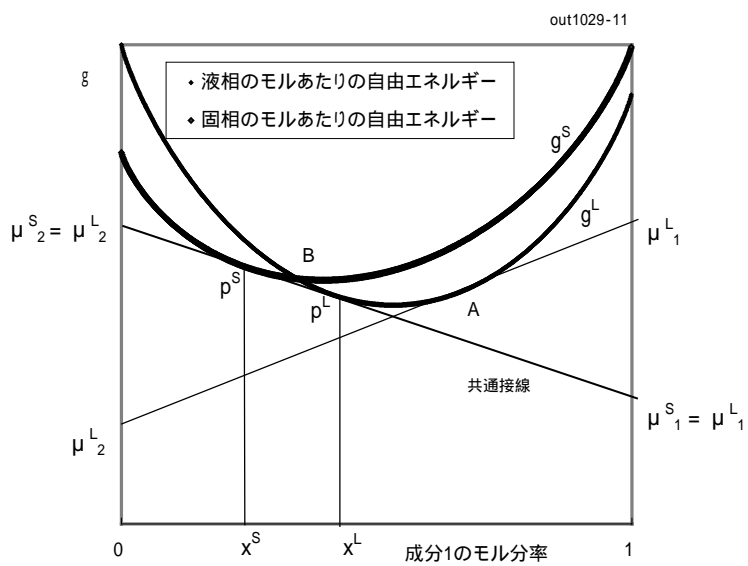


fig 2-1 2相共存時の自由エネルギーとケミカルポテンシャルの関係

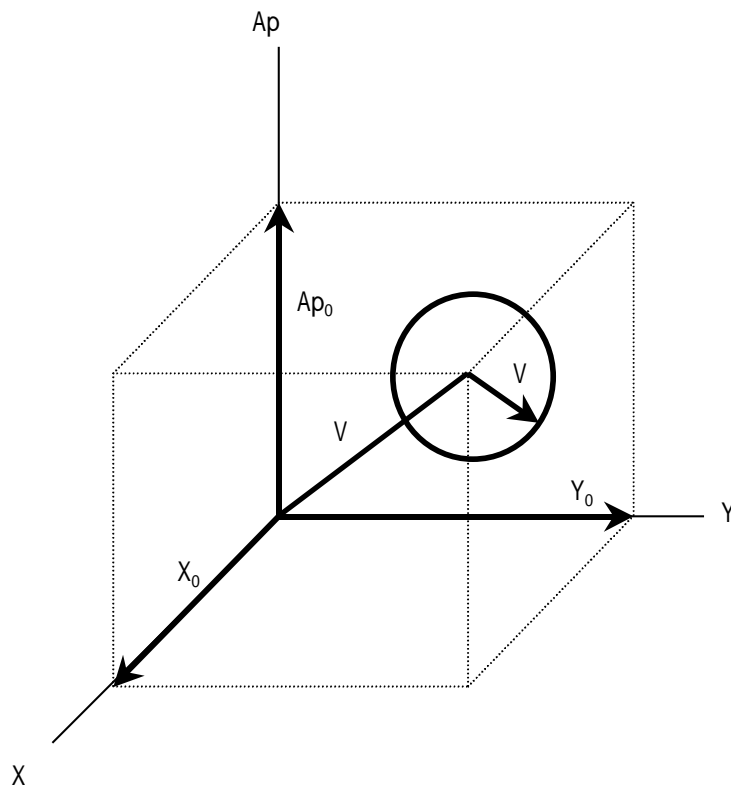


fig 3-1 自由エネルギー極小値探索模式図

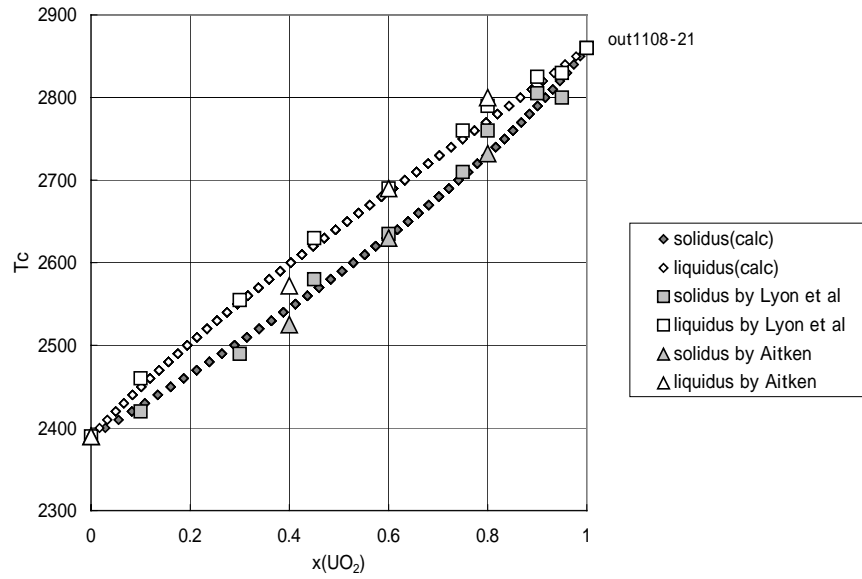


fig 4-1 UO₂-PuO₂系状態図の計算結果と測定値の比較

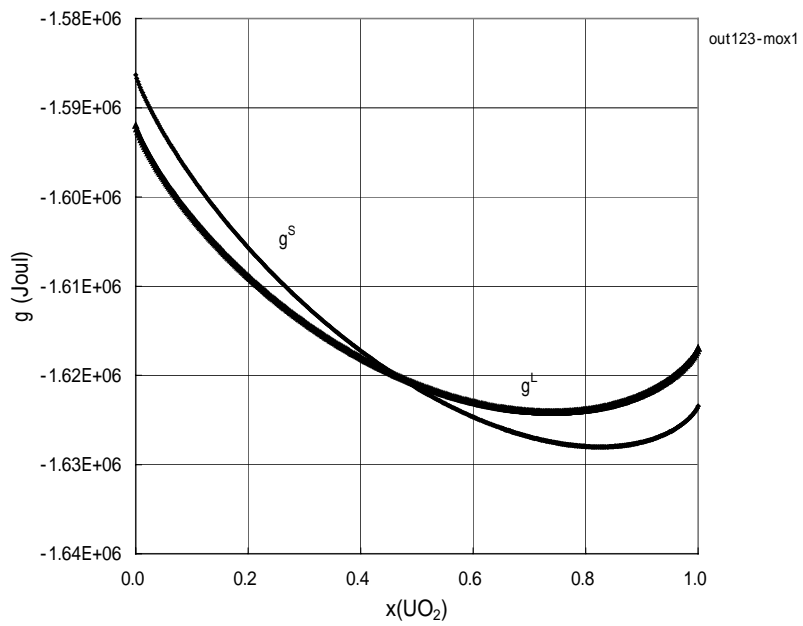


fig 4-2 UO₂-PuO₂系 固相及び液相の自由エネルギー(2600C)

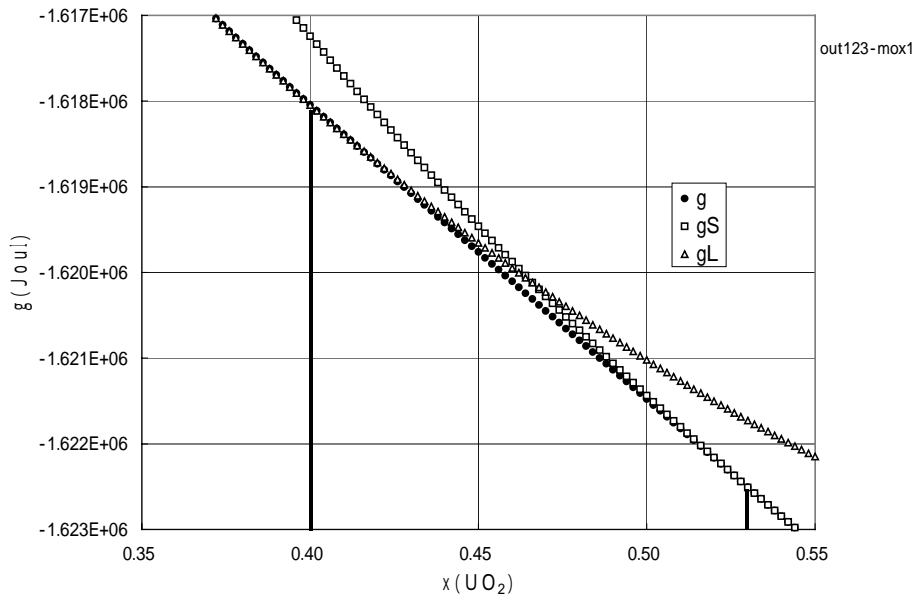


fig 4-3 UO₂-PuO₂系 固・液相の自由エネルギー(拡大図)

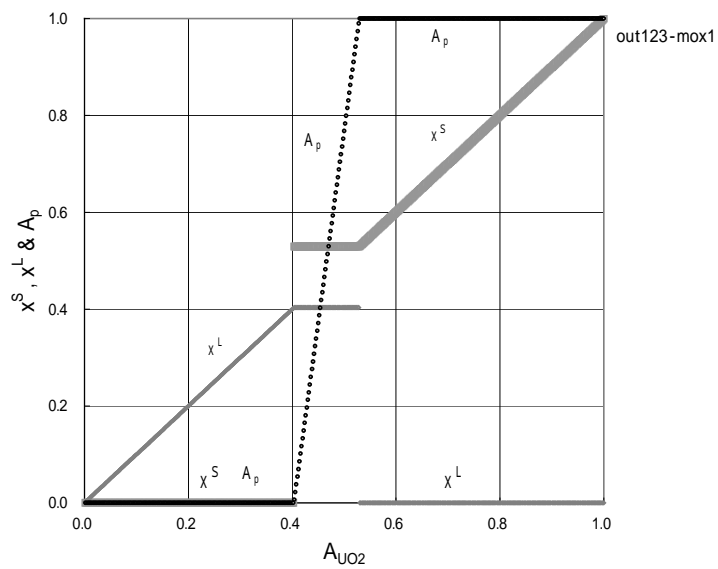


fig 4-4 UO₂-PuO₂系における固相及び液相の組成, 固相の存在割合の組成依存性 (2600C)

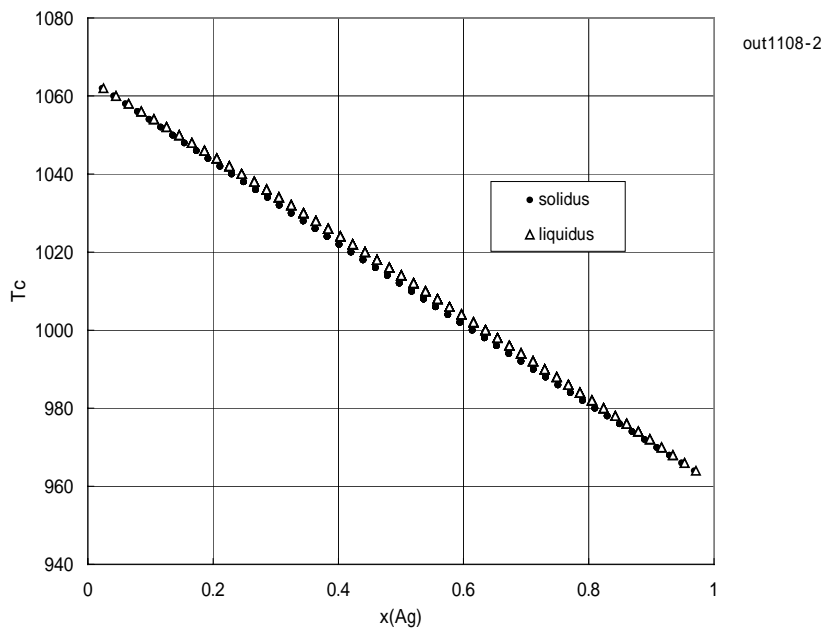


fig 4-5 Ag-Au系状態図 (理想溶液モデル)

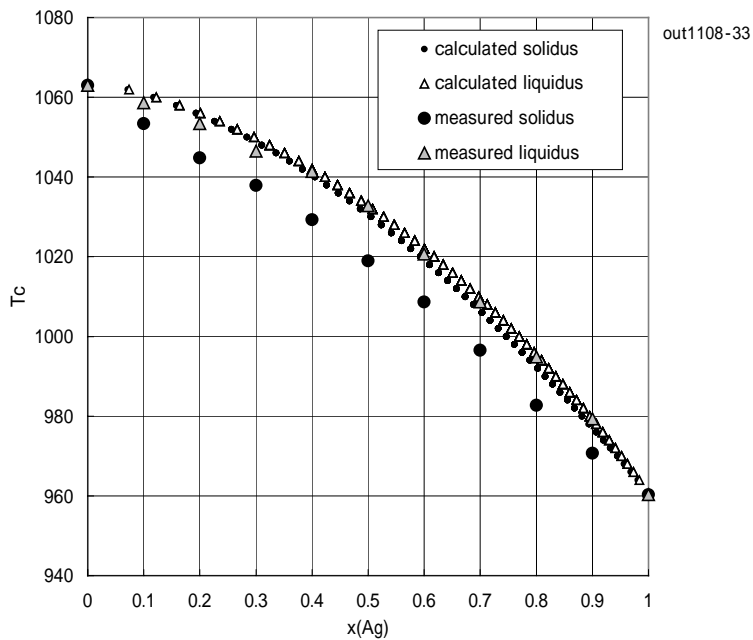


fig 4-6 Ag-Au系状態図 ($w^S = -700, w^L = 0.0$ Joul)

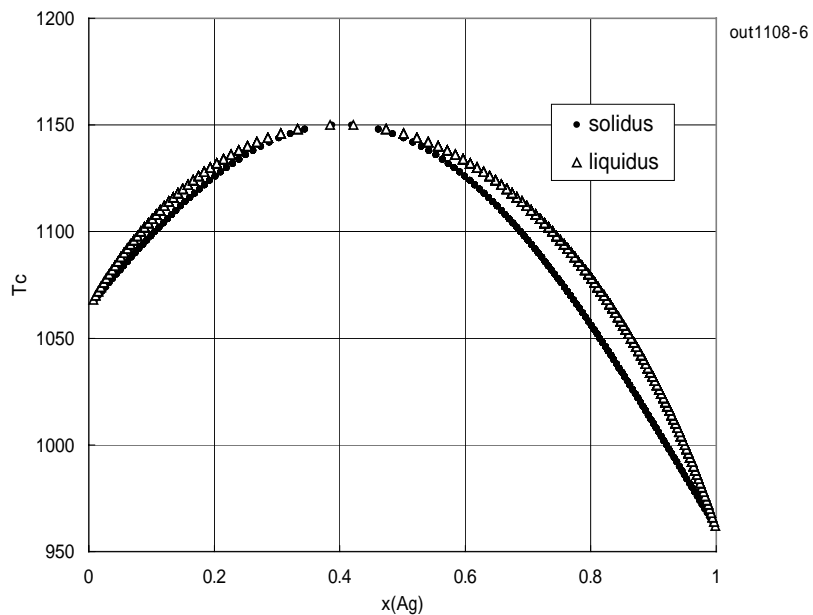


fig 4-7 Ag-Au系状態図 (正則溶液モデル, $w^S = -5000$, $w^L = 0.0$ Joul)

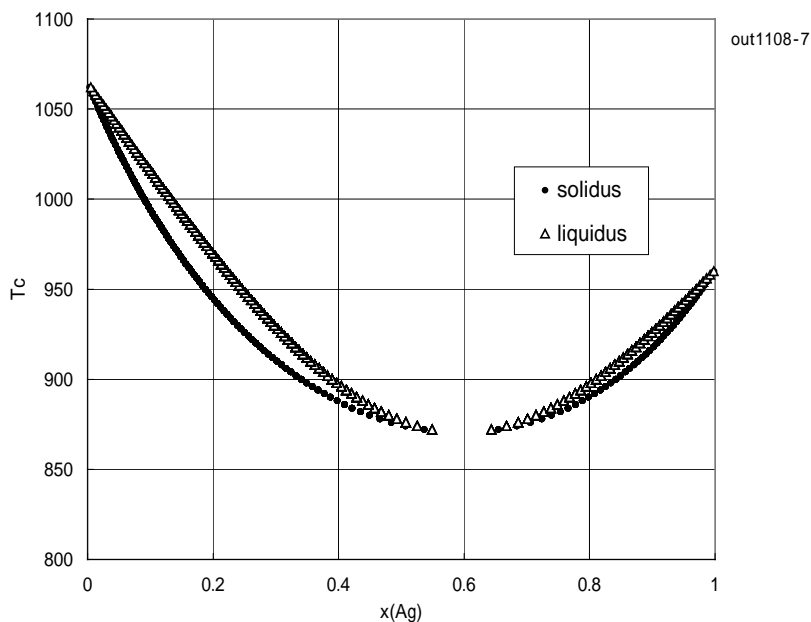


fig 4-8 Ag-Au系の状態図 (正則溶液モデル, $w^L = -5000$, $w^S = 0.0$ Joul)

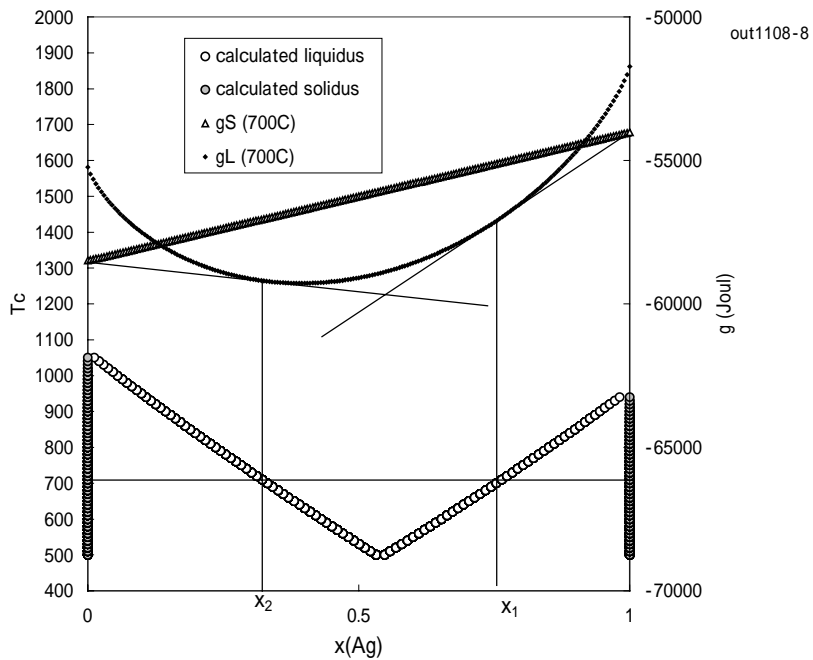


fig 4-9 共晶混合物を持つ系の自由エネルギーと状態図

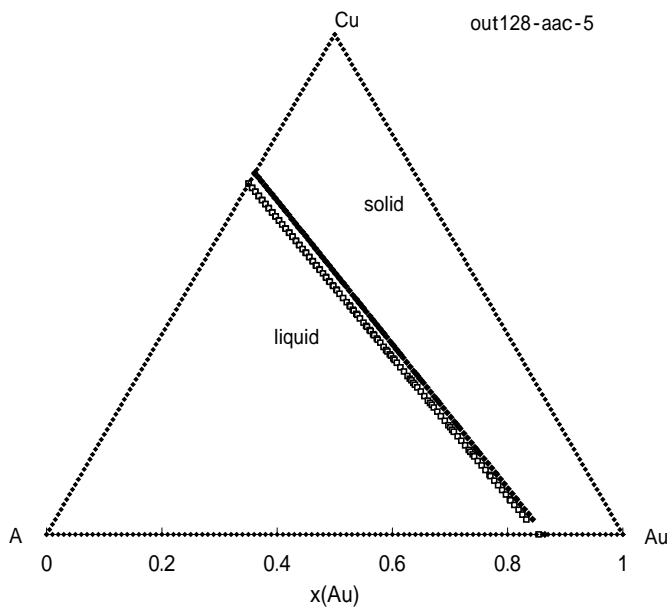


fig 4-10 Ag-Au-Cu系状態図 (理想溶液モデル,1050C)

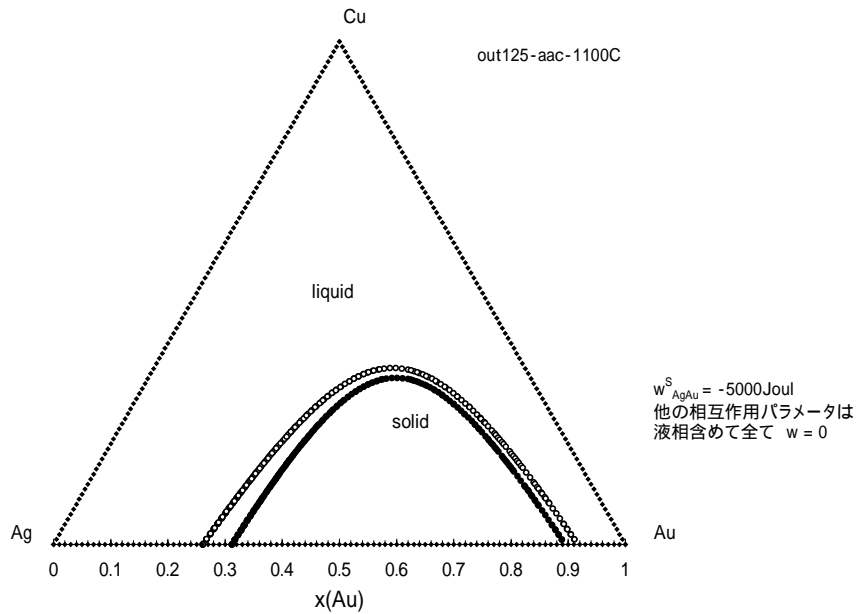


fig 4-11 Ag-Au-Cu 系状態図 (正則溶液モデル、1100 C)

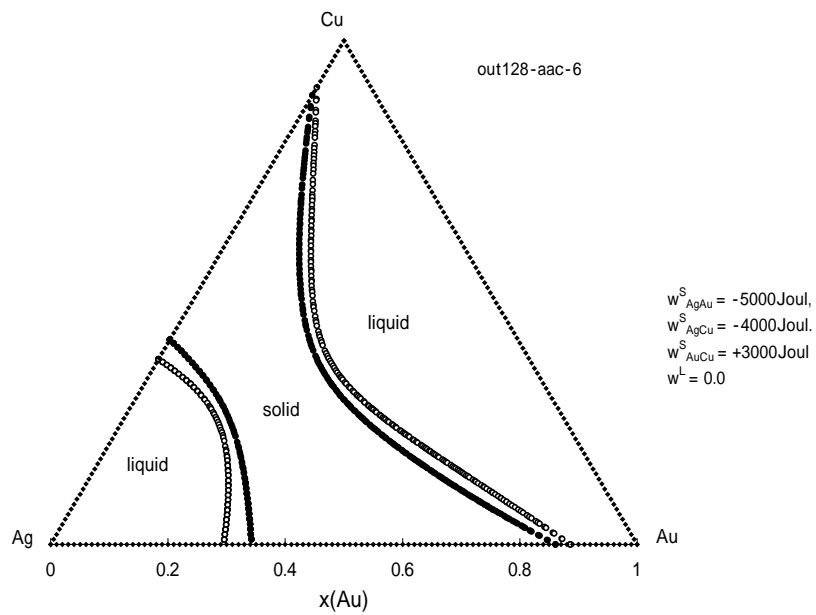


fig 4-12 Ag-Au-Cu 系状態図 (正則溶液モデル、1110 C)

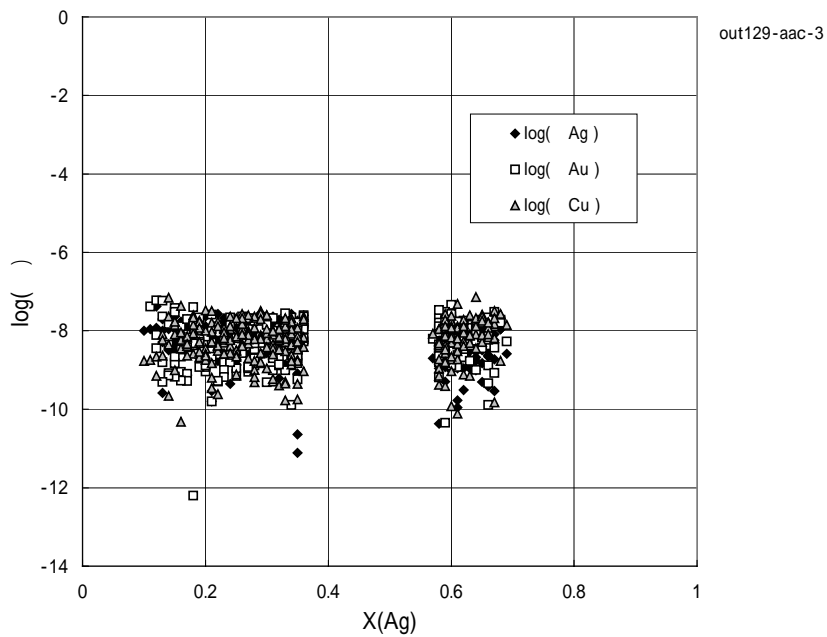


fig 4-13 2相間のケミカルポテンシャルの計算誤差

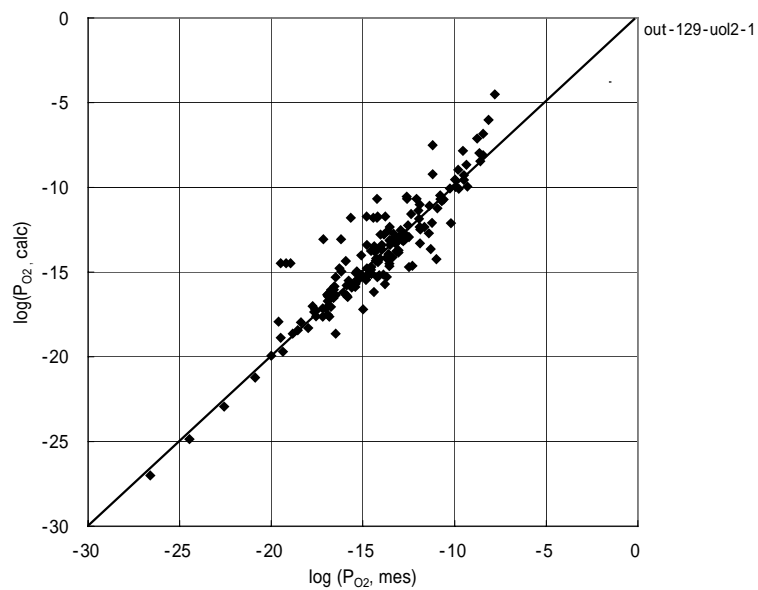


fig 5-1 酸素分圧の測定値と解析値の比較 (U-O系、0/U < 2.0)

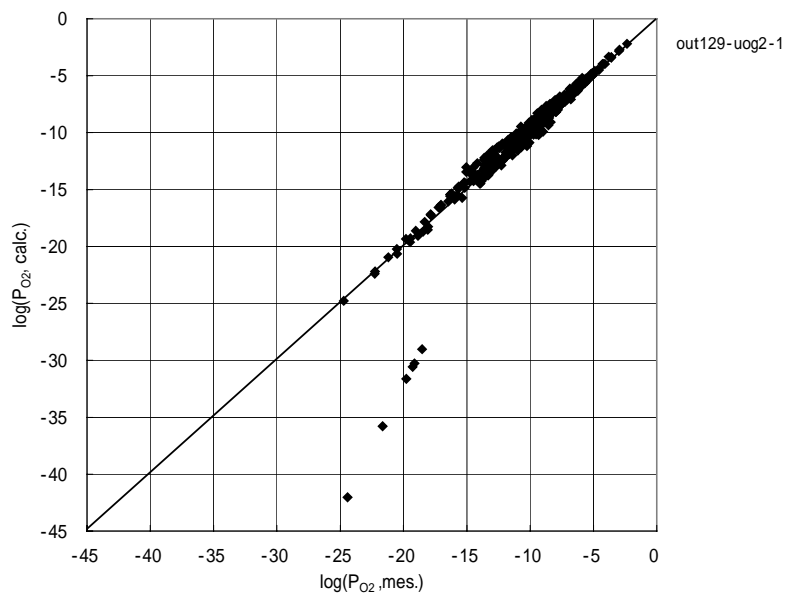


fig 5-2 酸素分圧の測定値と解析値の比較 (U-O系、O/U>2.0)

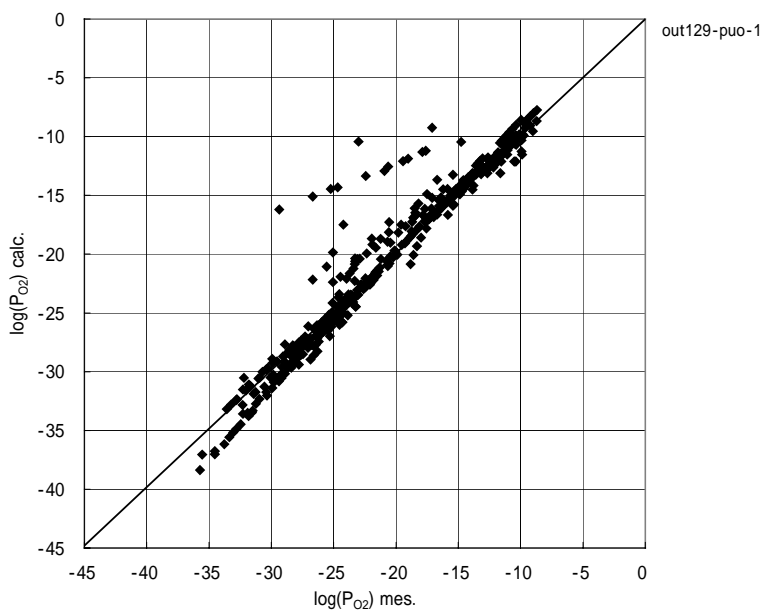


fig 5-3 酸素分圧の測定値と解析値の比較 (Pu-O系)

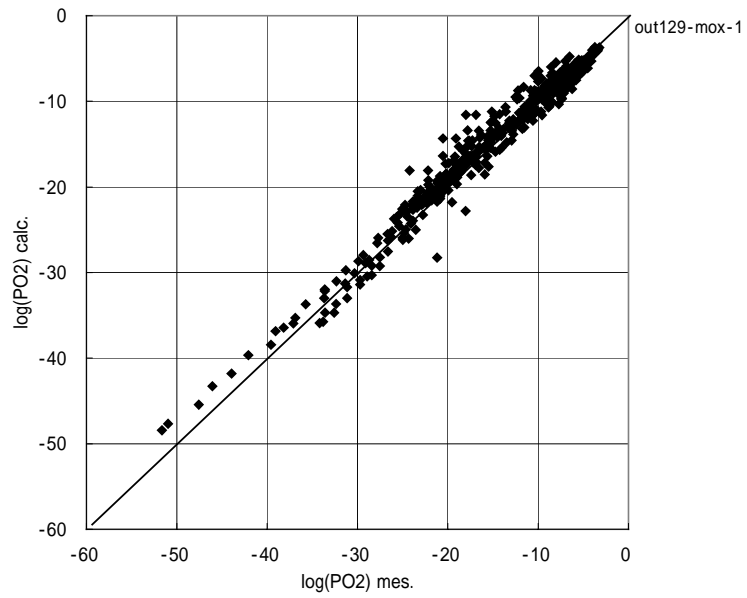


fig 5-4 酸素分圧測定値と解析値の比較 (U-Pu-O系)

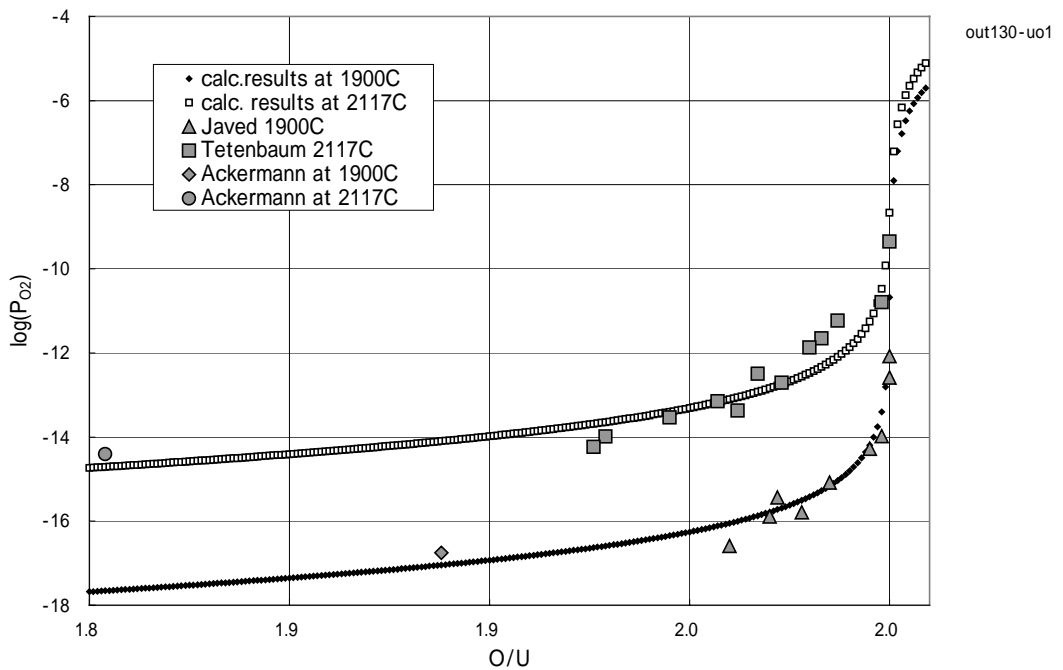


fig 5-5 酸素分圧の解析値と測定結果の比較 (U-O系)

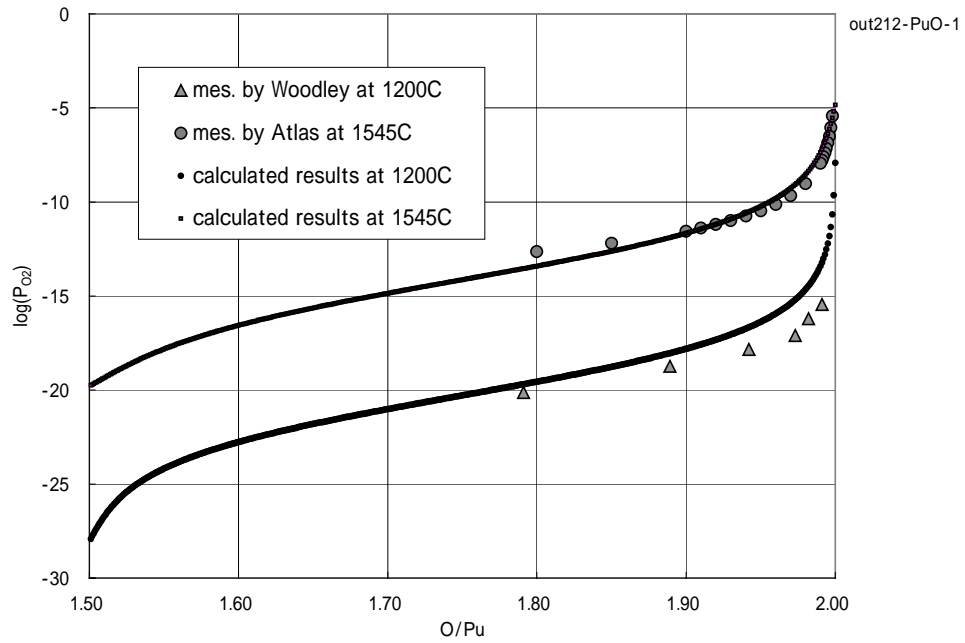


fig 5-6 酸素分圧の解析結果と測定値の比較 (Pu-O系)

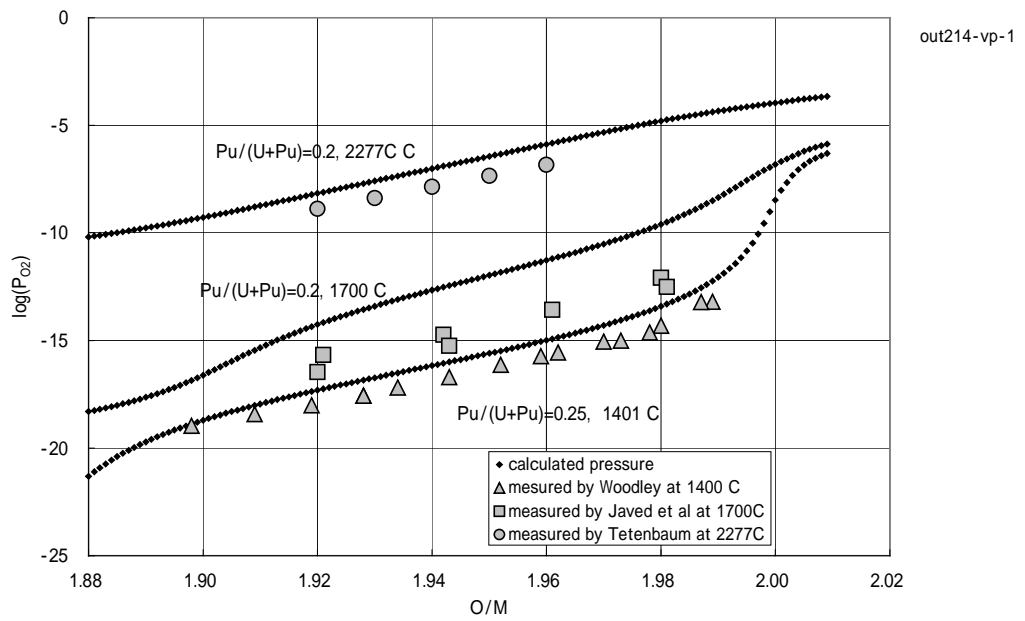


fig 5-7 酸素分圧の解析結果と測定値の比較 (U-Pu-O系)

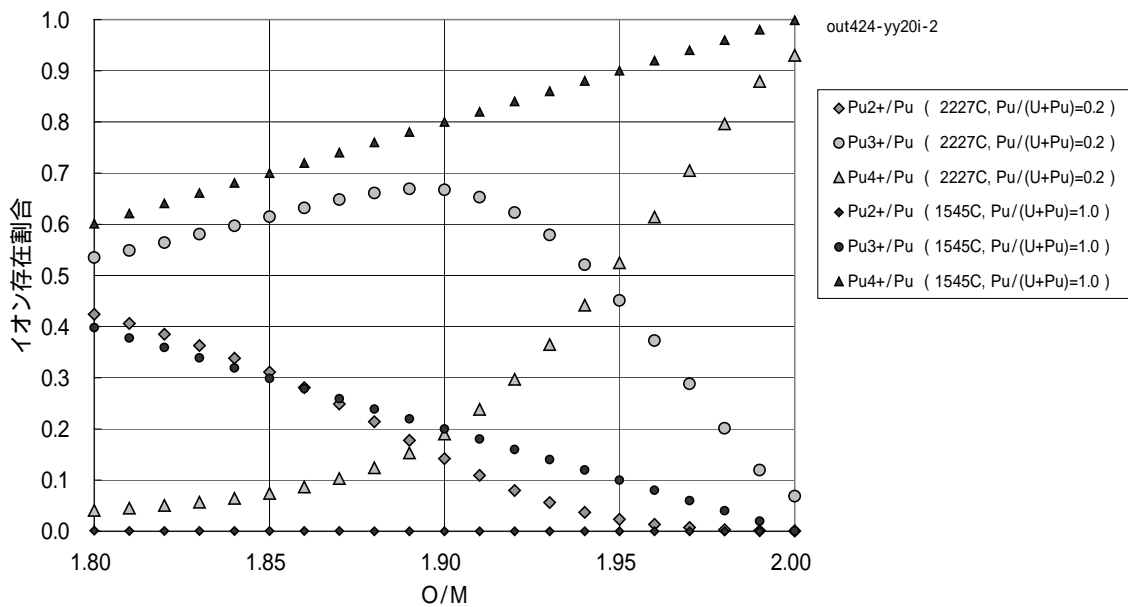


fig 5-8 (0.2Pu, 0.8U)O_{2-x}系でのPuイオンの存在割合

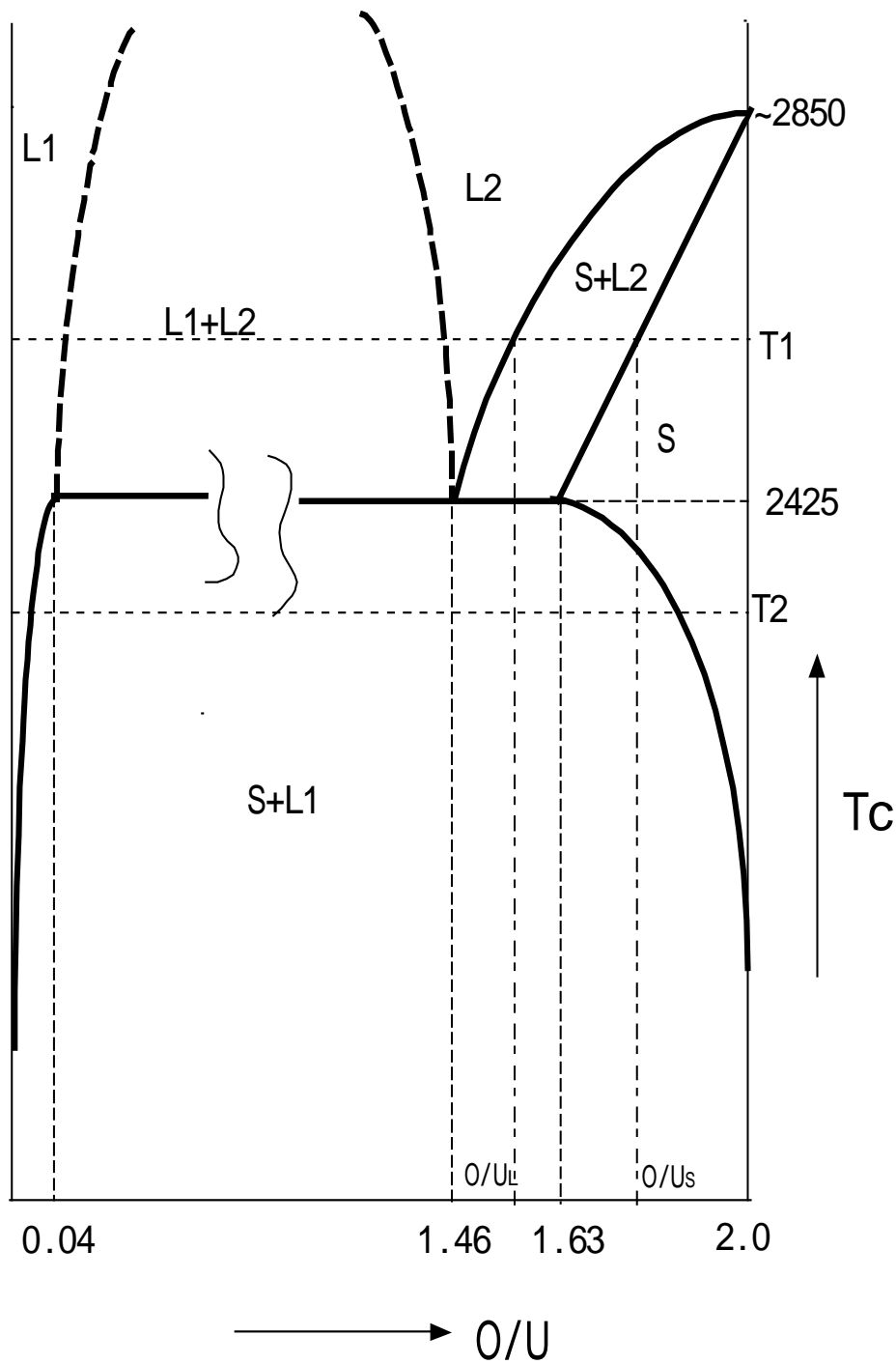


fig 5-9 $O/U < 2.0$ 領域の状態図(模式図)

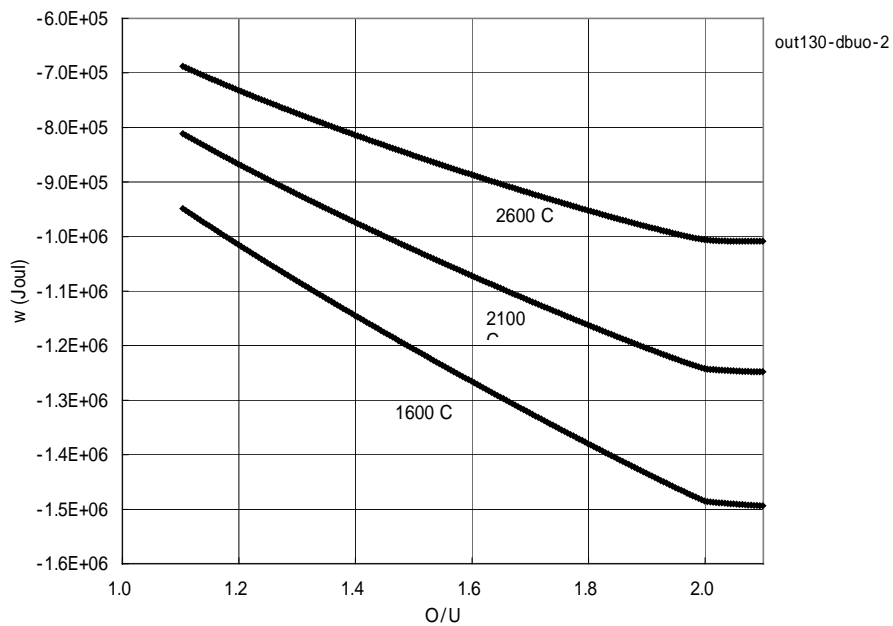


fig 5-10 U-O 系の相互作用パラメータ (w)

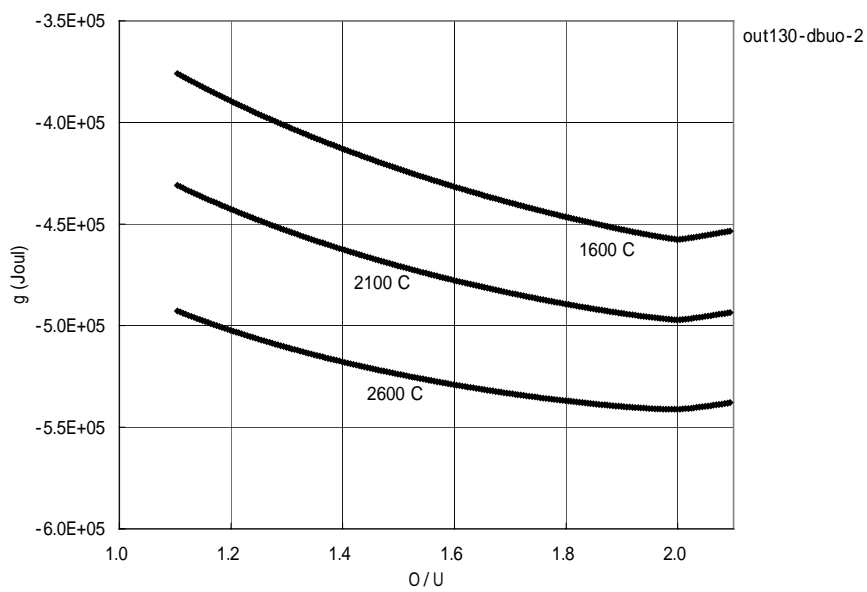


fig 5-11 U-O 系の自由エネルギー

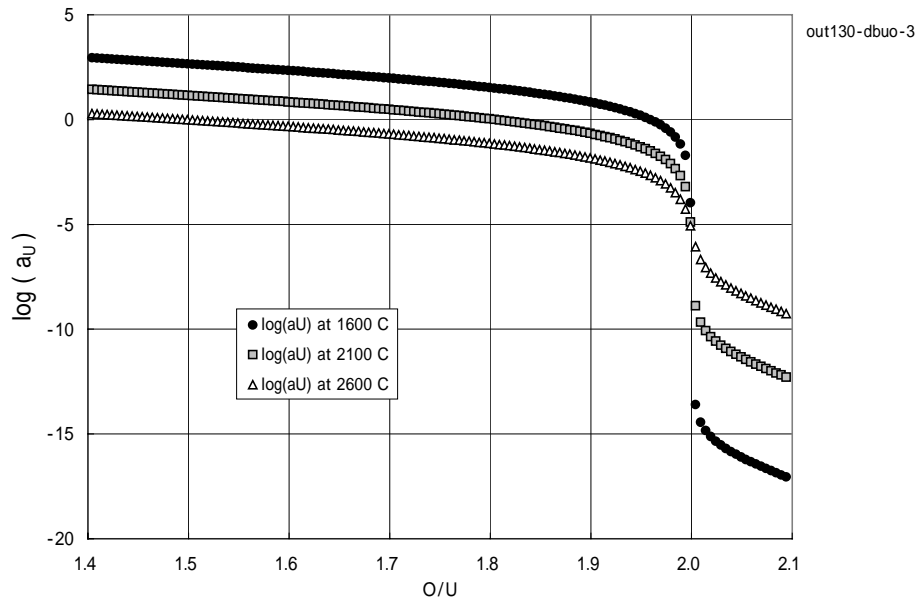


fig 5-12 U-O系のUの活量

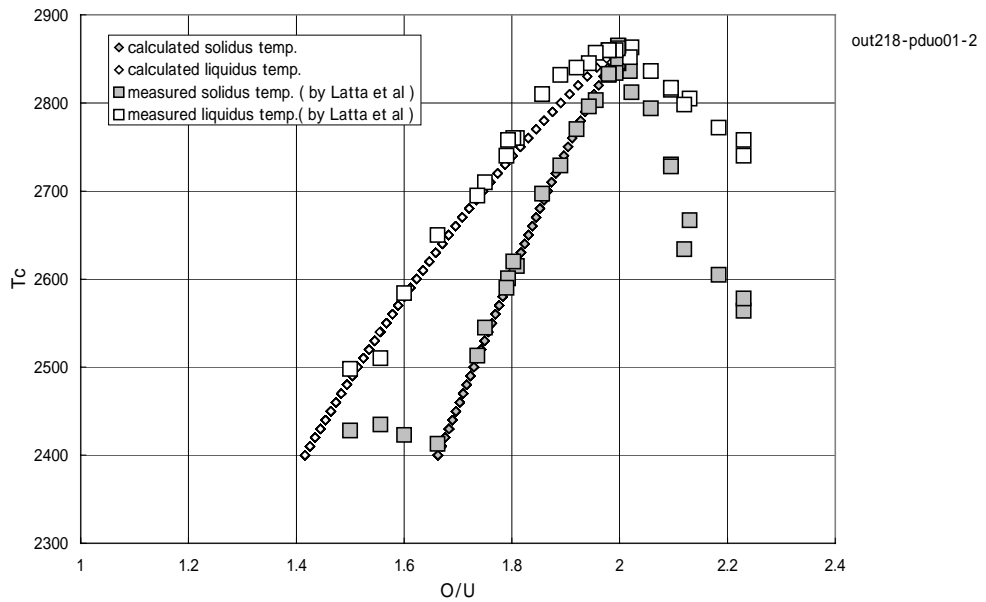


fig 5-13 U-O系の状態図 解析結果と測定値の比較

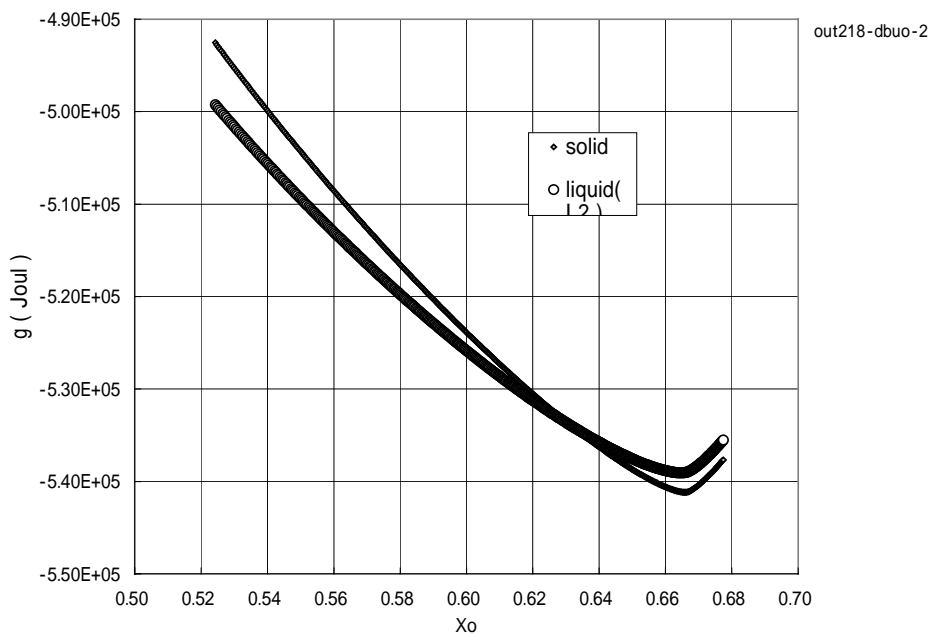


fig 5-14 U-O 系の固相・液相(L2)の自由エネルギー (2600 C)

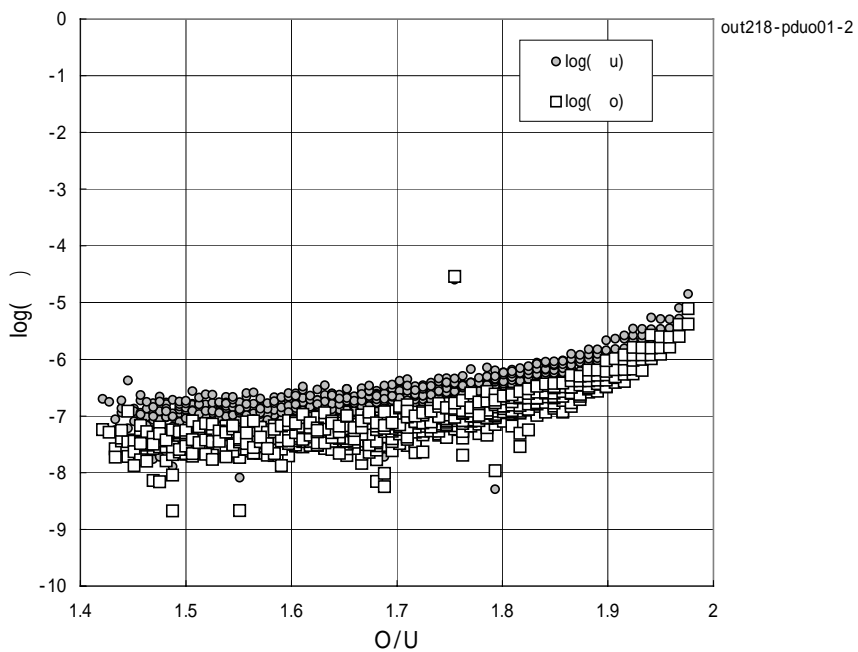


fig 5-15 2相間のケミカルポテンシャルの計算誤差(U-O系)

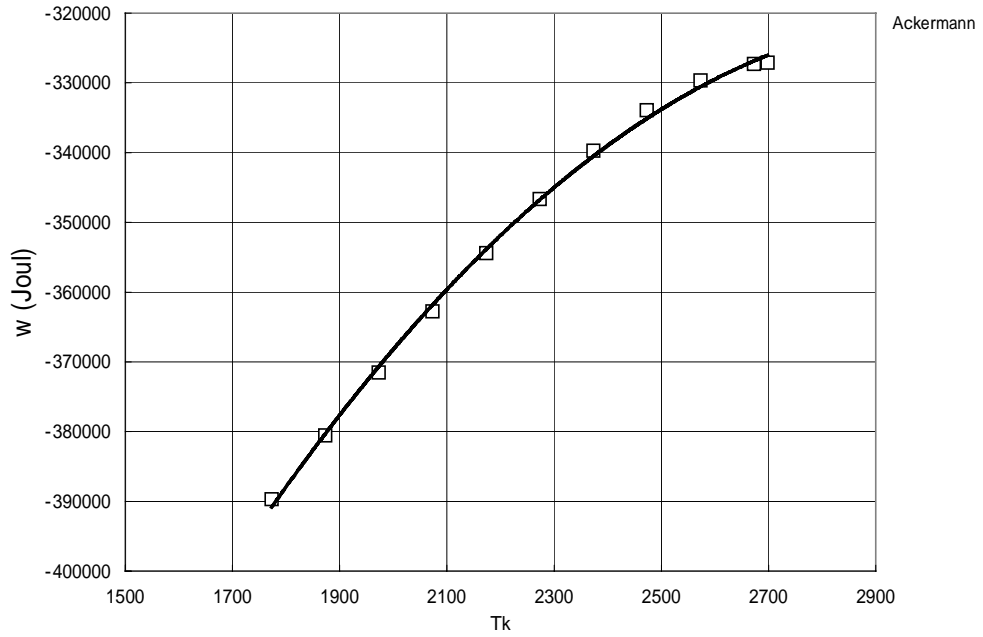


fig 5-16 液相L1の相互作用パラメータ (w)

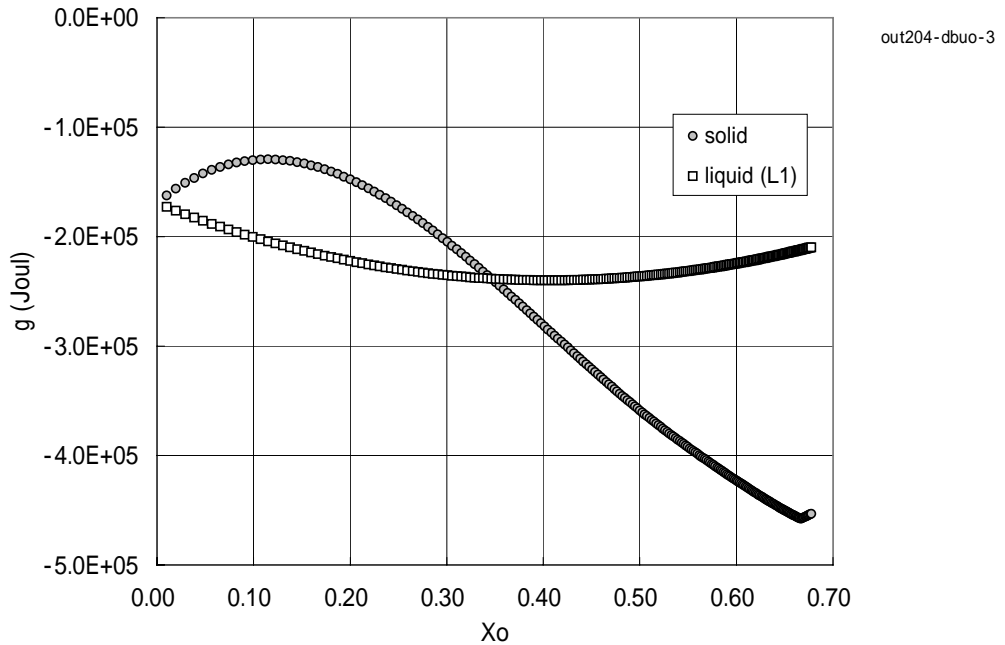


fig 5-17 U-O 系の固相・液相 (L1) の自由エネルギー

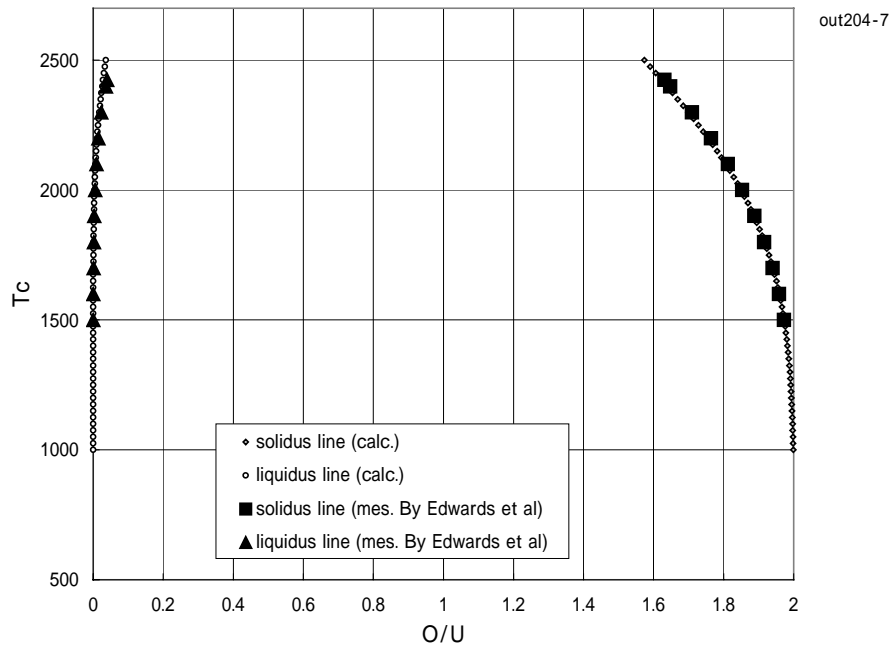


fig 5-18 U-UO_{2-x}系状態図 計算結果と測定値の比較

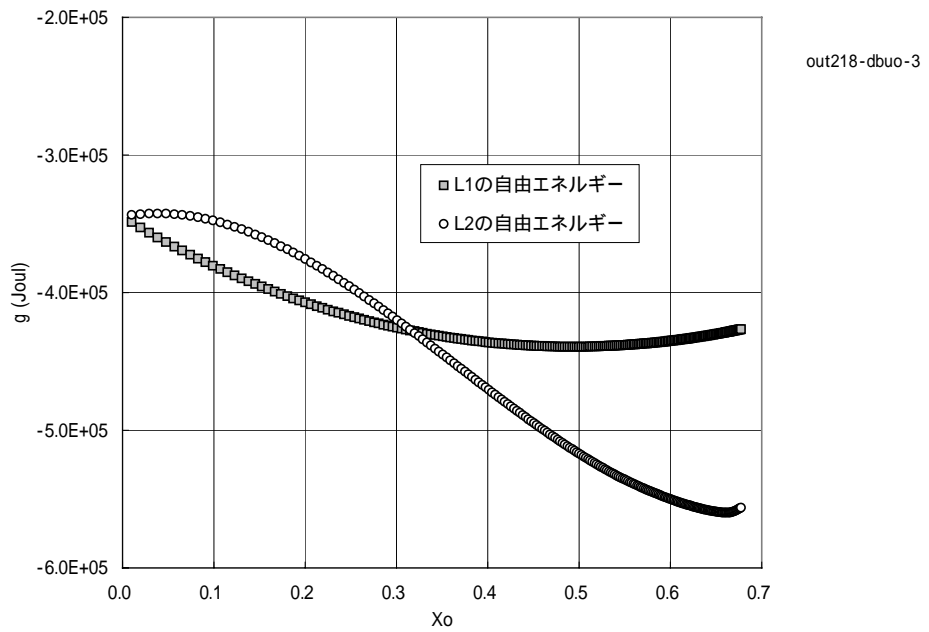


fig 5-19 液相L1及びL2の自由エネルギー

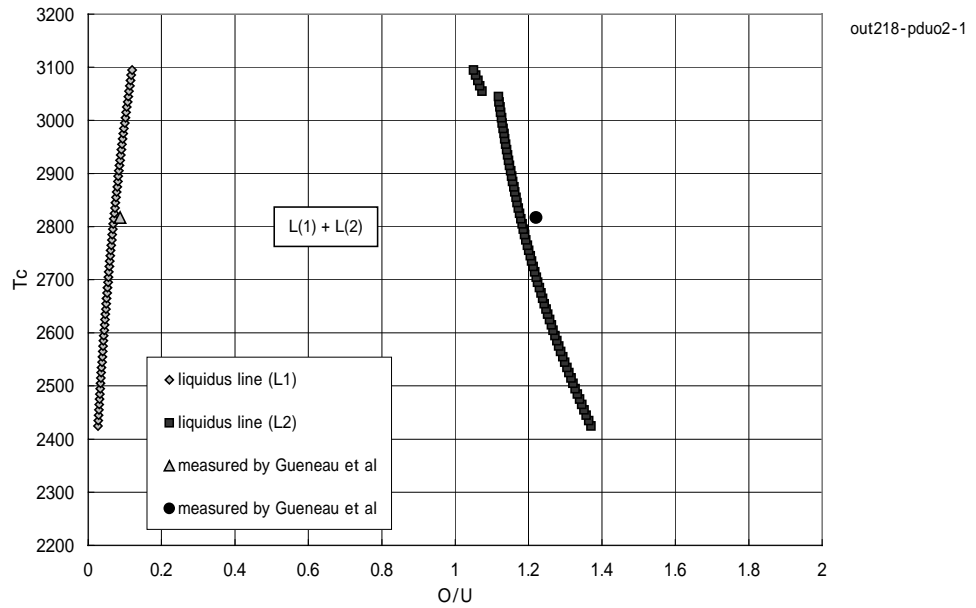


fig 5-20 U (L1) - UO_{2-x} (L2) 系の状態図

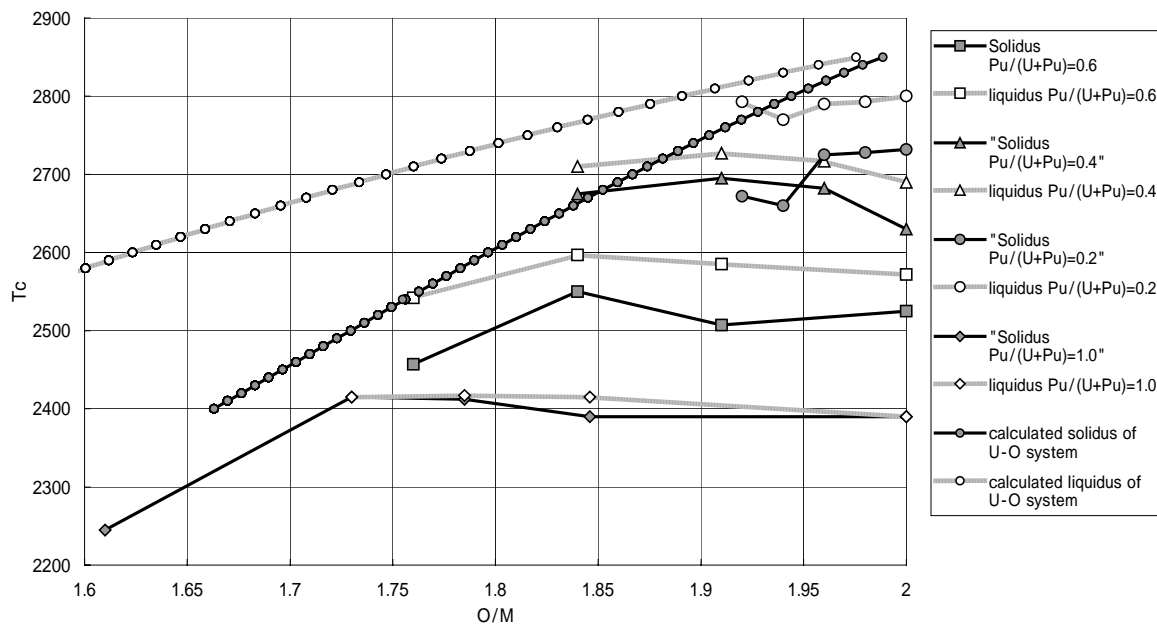


fig 5-21 Pu-O及びU-Pu-O系の固・液相線の測定結果 (Aitken等)

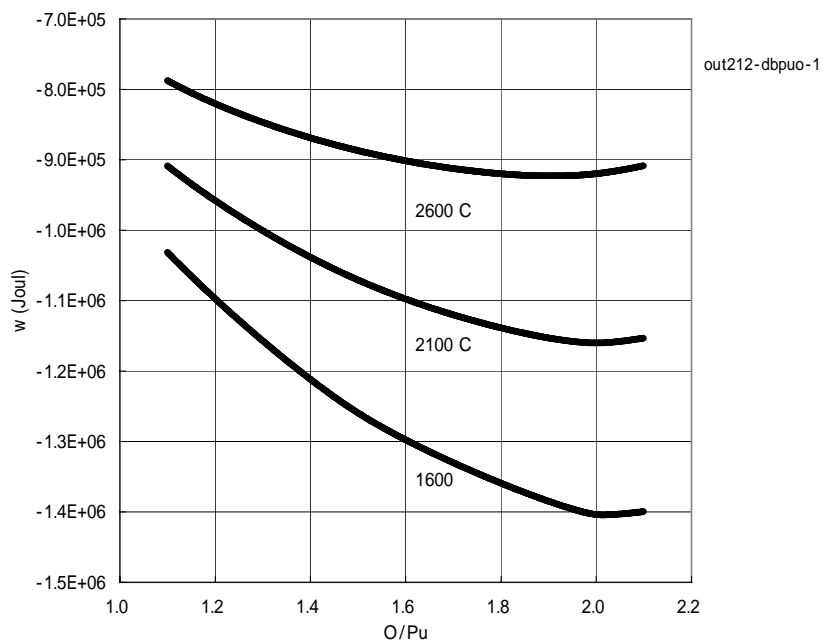


fig 5-22 固相 Pu-O系の相互作用パラメータ (w)

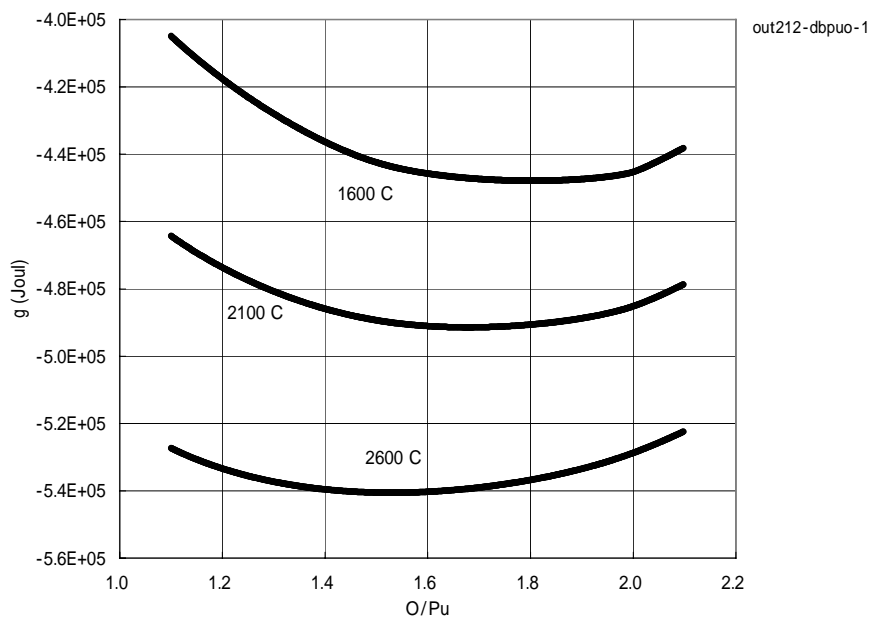


fig 5-23 固相 Pu-O系の自由エネルギー

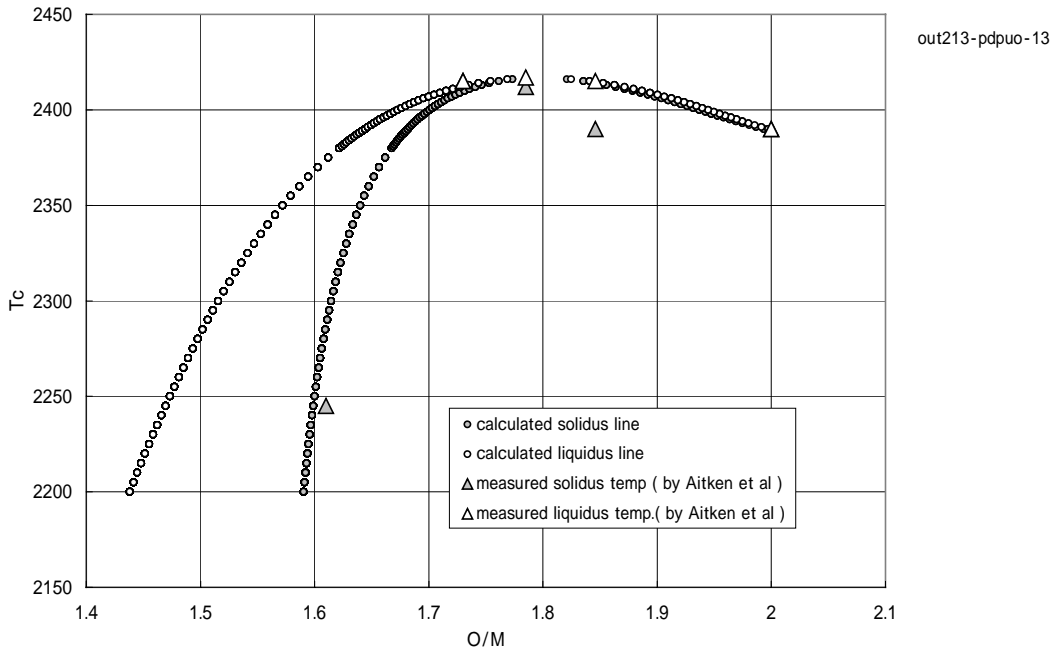


fig 5-24 Pu-O 系の状態図の解析結果と測定値の比較

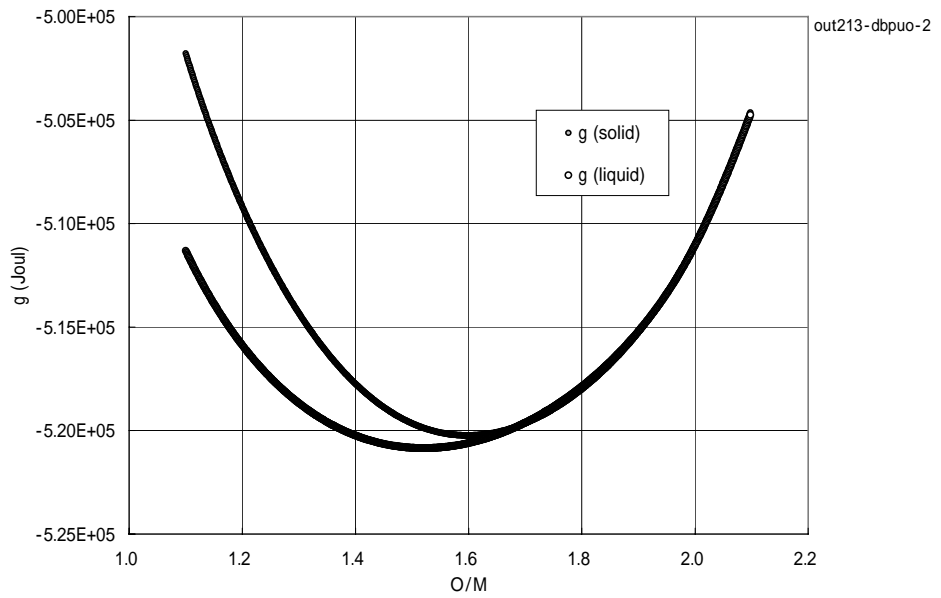


fig 5-25 Pu-O系 固・液相の自由エネルギー(2400 C)

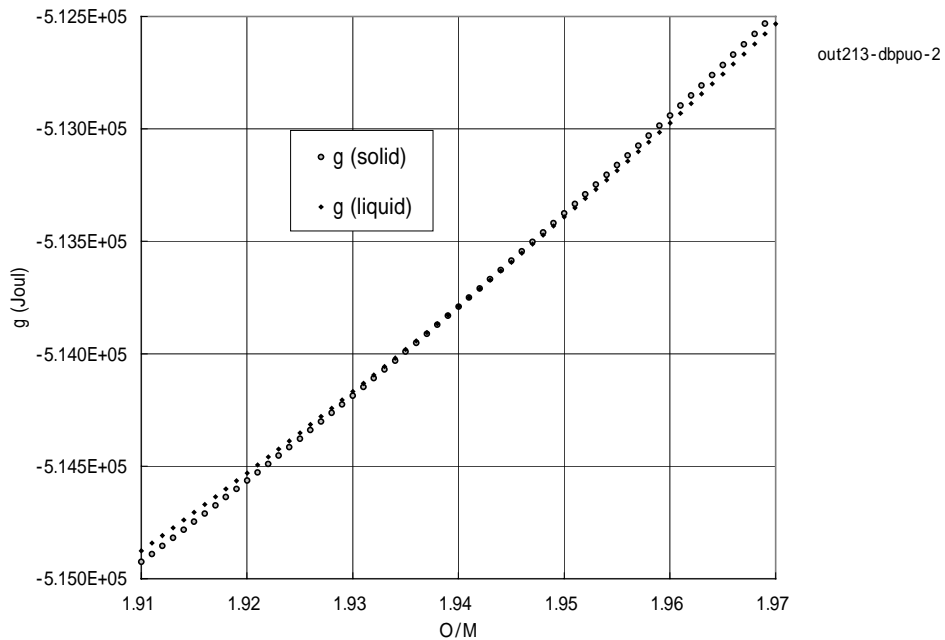


fig 5-26 Pu-O系 固・液相の自由エネルギー (2400 C 部分拡大図)

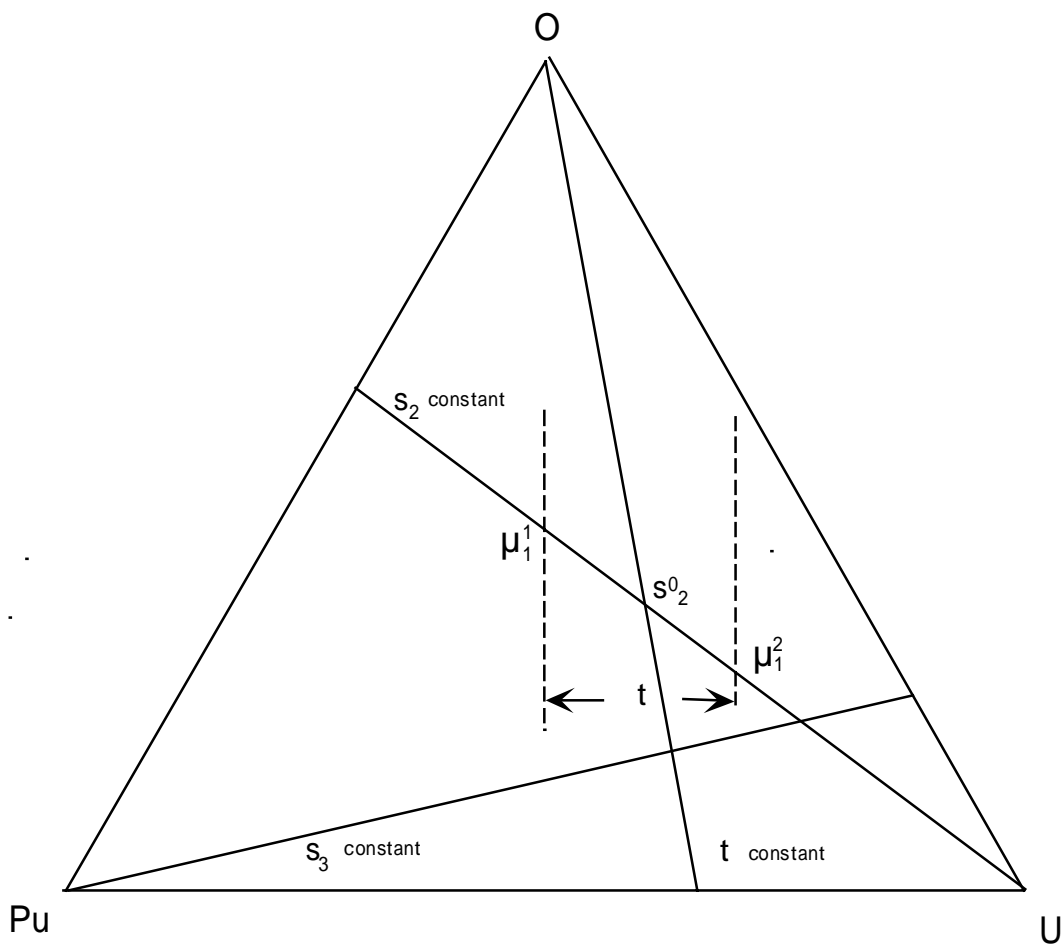


fig 5-27 U-Pu-O系にて、0のケミカルポテンシャルが既知の時、他の2成分のケミカルポテンシャルを計算する模式図

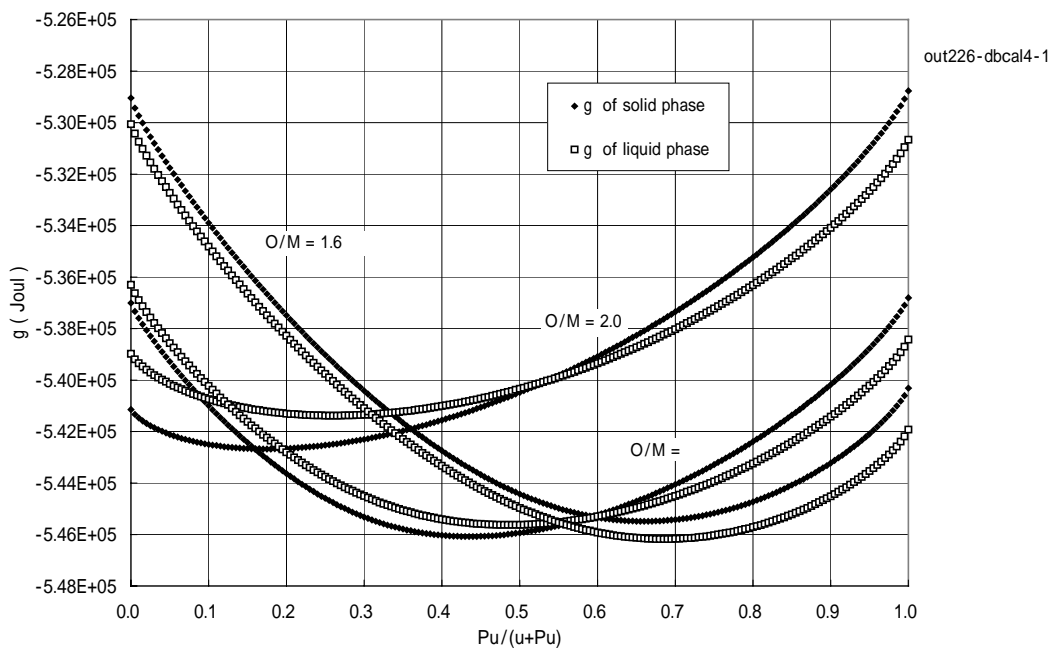


fig 5-28 U-Pu-O系の自由エネルギー (2600 C)

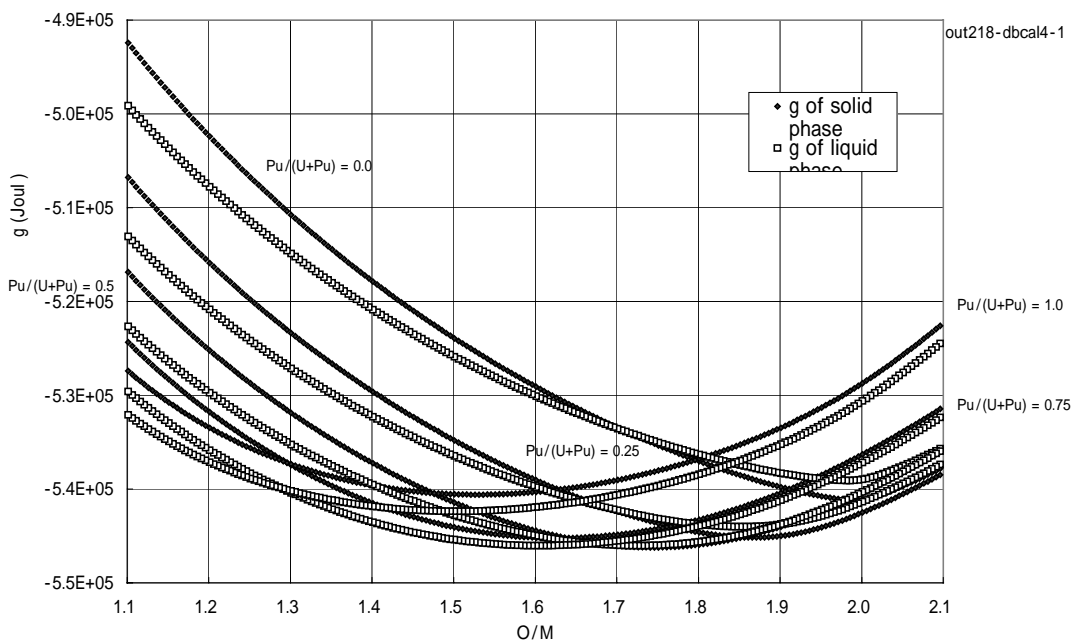


fig 5-29 U-Pu-O系の自由エネルギー (2600 C)

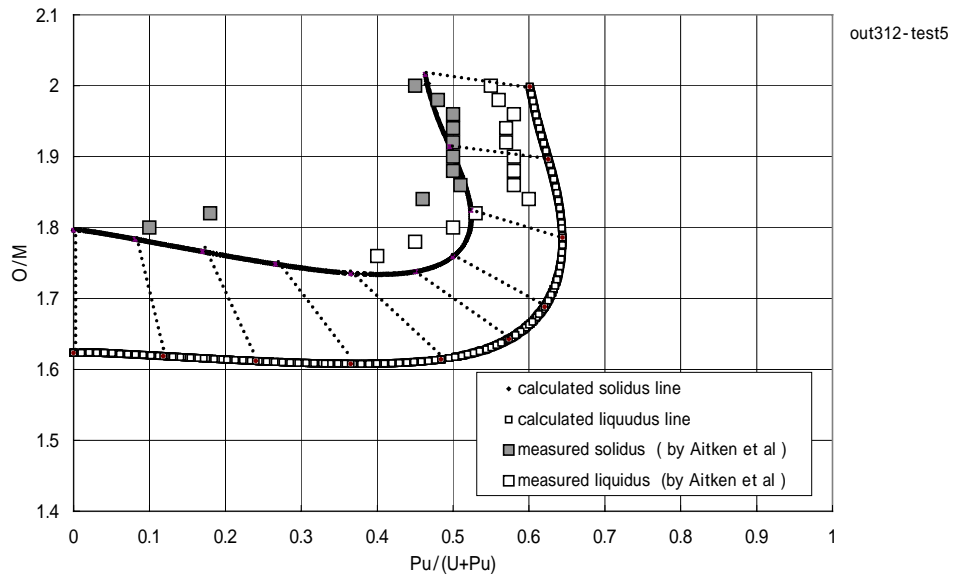


fig 5-30 U-Pu-O系の状態図の計算結果 (2600 C)

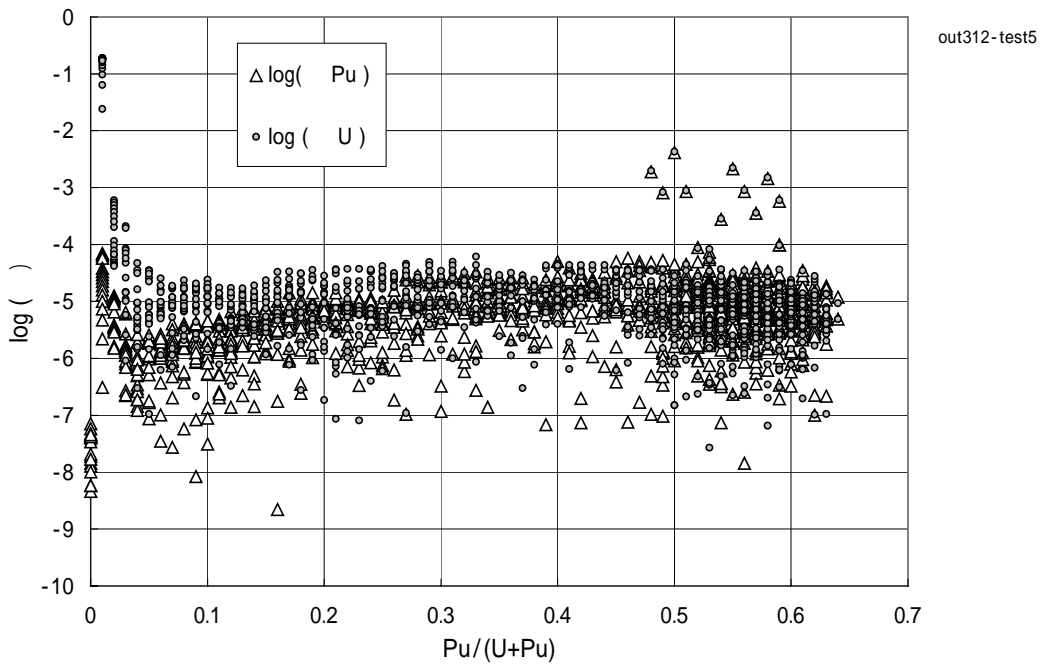


fig 5-31 U-Pu-O 系の計算誤差 その1 (U 及びPu , 2600 C)

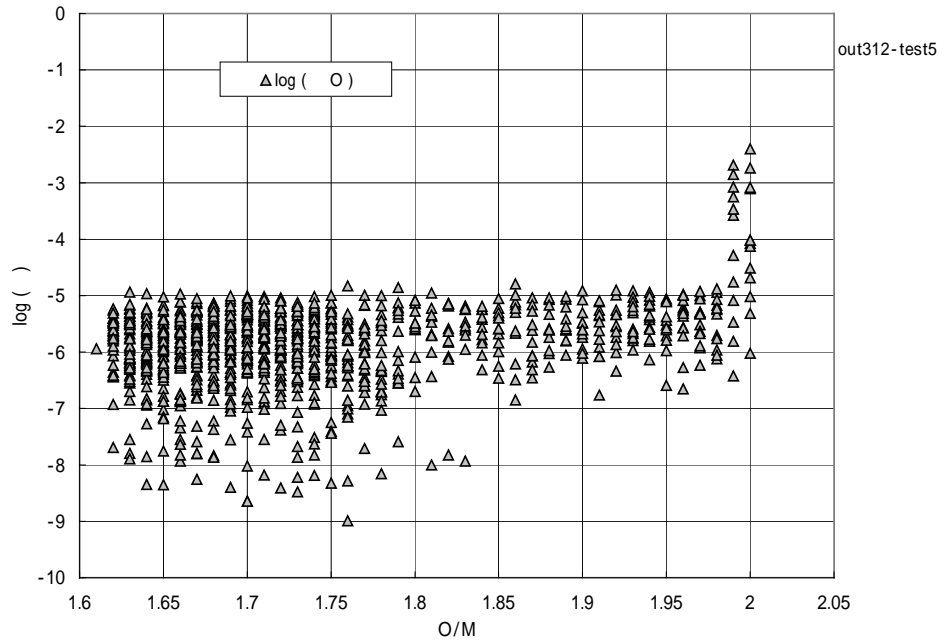


fig 5-32 U-Pu-O 系の計算誤差 その2 (O, 2600 C)

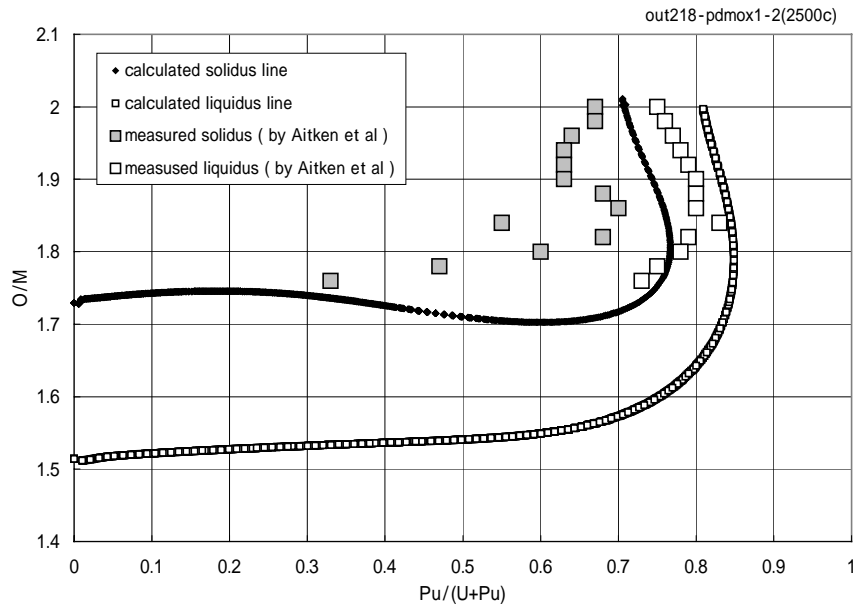


fig 5-33 U-Pu-O系状態図の計算結果 (2500 C)

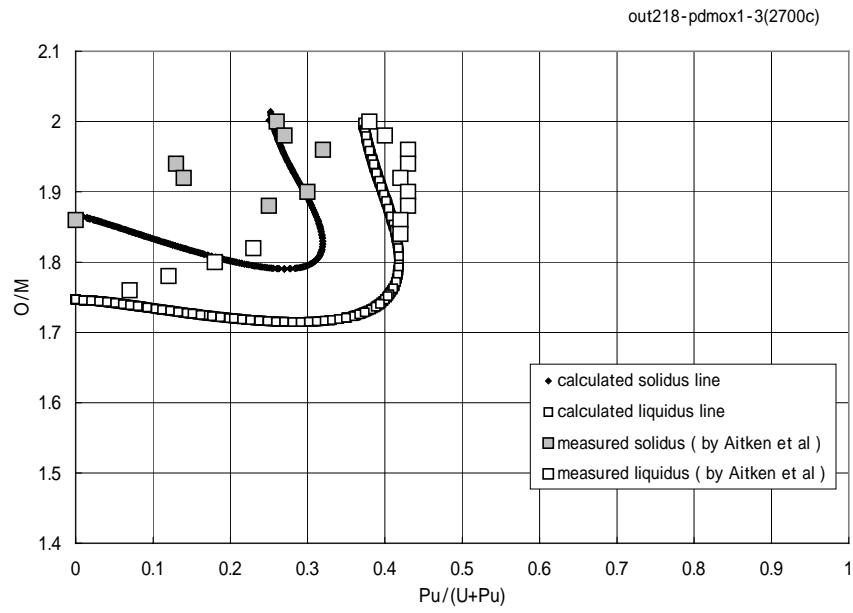


fig 5-34 U-Pu-O系状態図の計算結果 (2700 C)

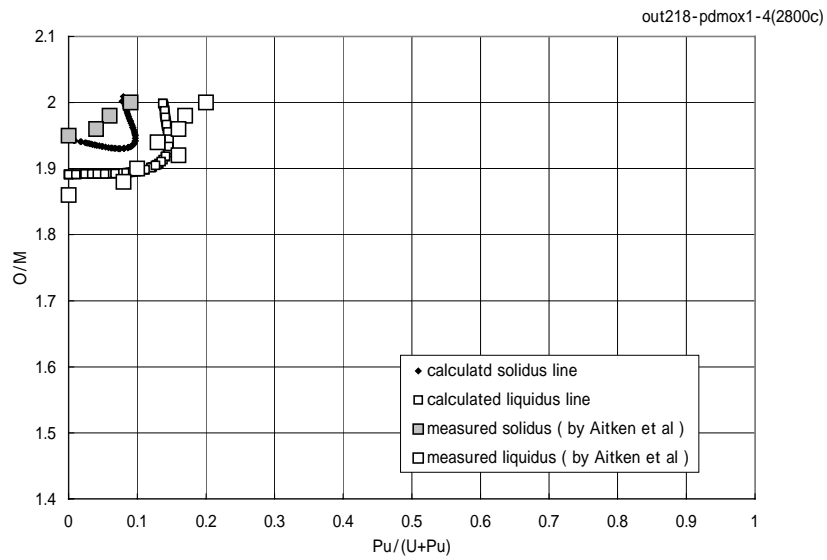


fig 5-35 U-Pu-O系状態図の計算結果 (2800 C)

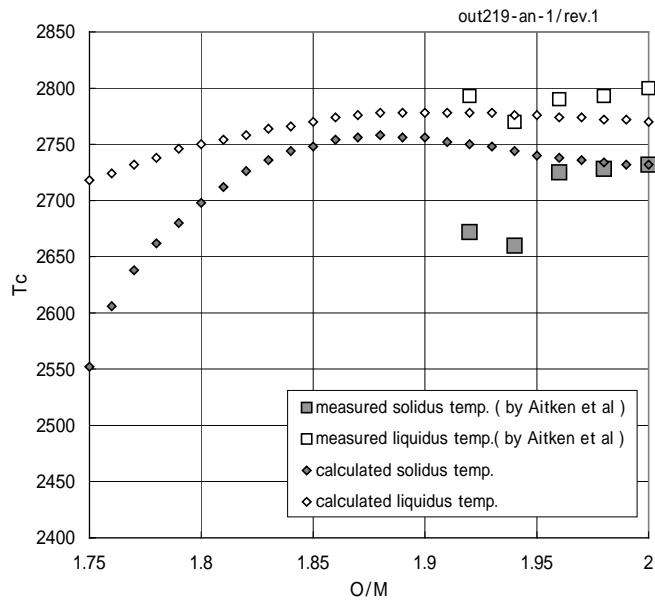


fig 5-36 U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu) = 0.2)

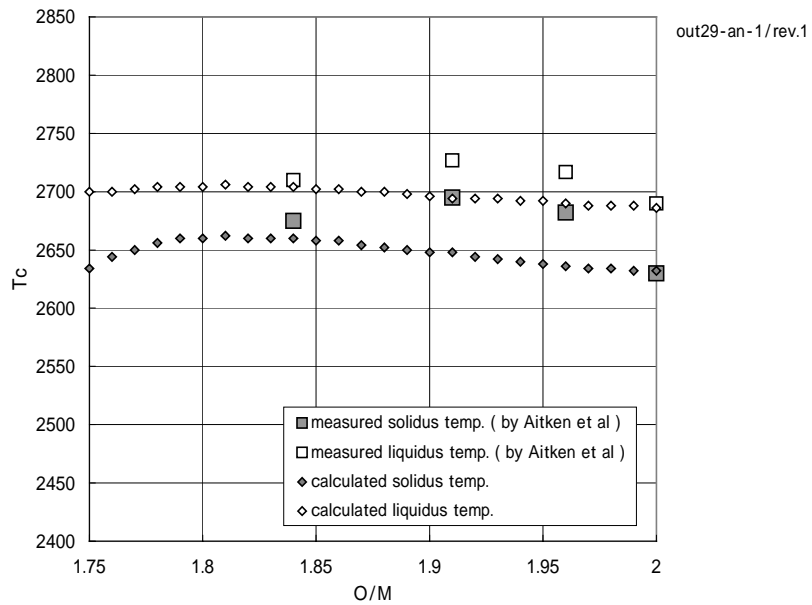


fig 5-37 U-Pu-O系状態図の計算結果と測定値の比較 (Pu/(U+Pu) = 0.4)

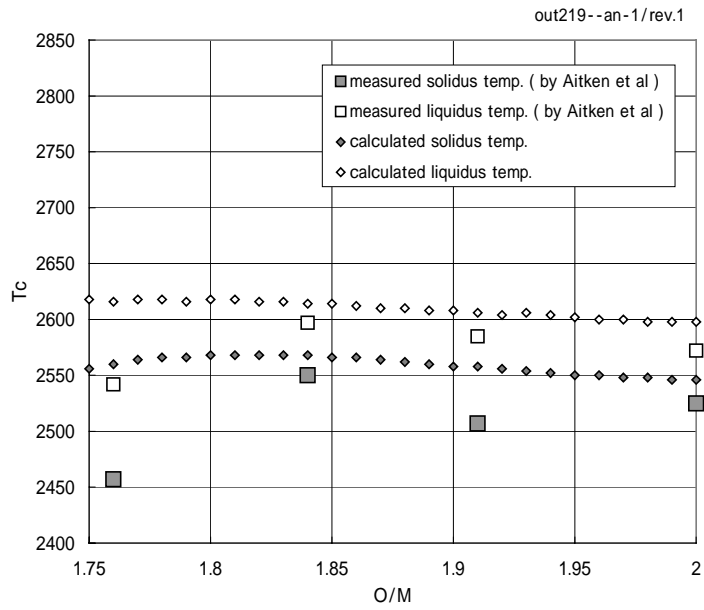


fig 5-38 U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu) = 0.6)

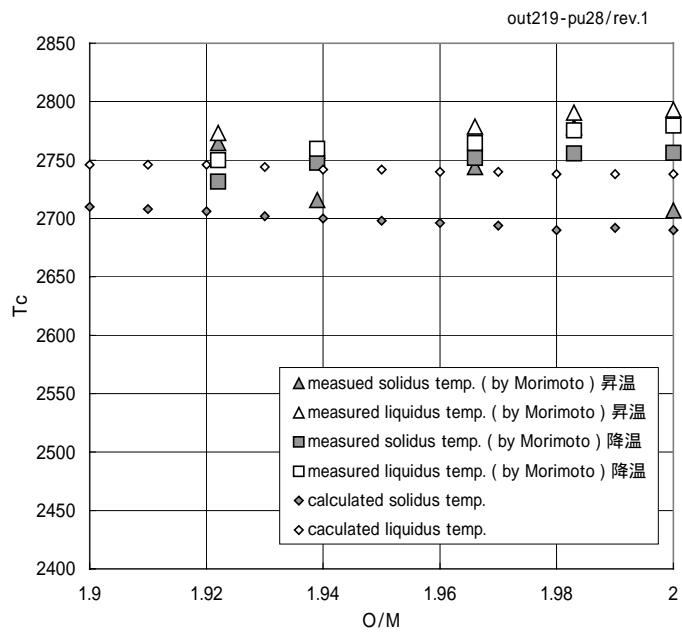


fig 5-39 U-Pu-O 系状態図の計算結果と測定値の比較 (Pu/(U+Pu) = 0.28)

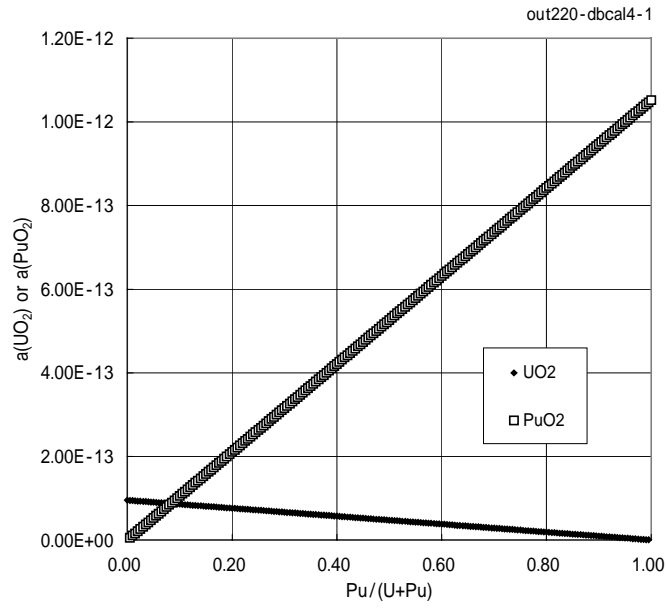


fig 5-40 (U, Pu) O₂ 系での UO₂ 及び PuO₂ の活量 (2600 C)

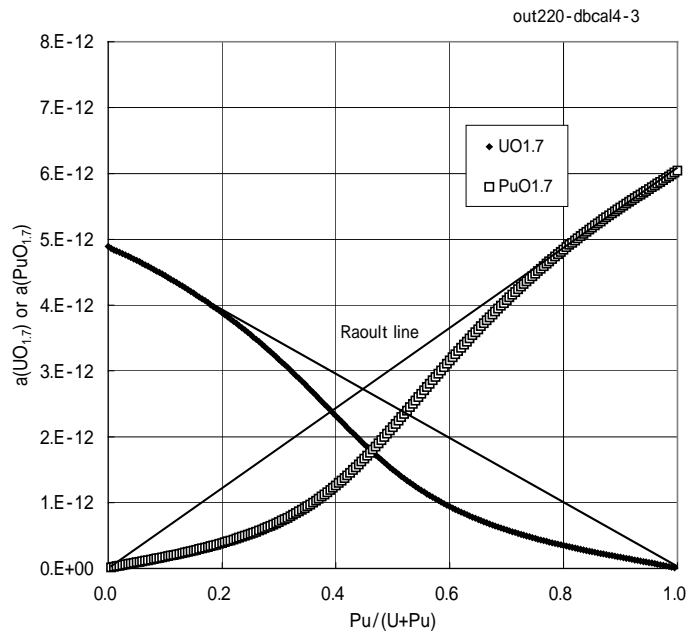


fig 5-41 (U, Pu) O_{1.7} 系での UO_{1.7} 及び PuO_{1.7} の活量

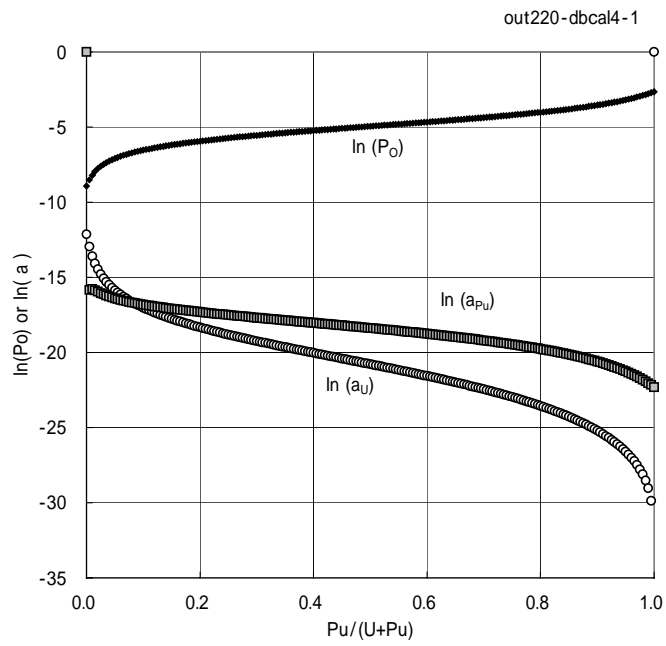


fig 5-42 (U, Pu) $O_{2.0}$ 系の酸素分圧、及び U, Pu の活量 (2600 C)

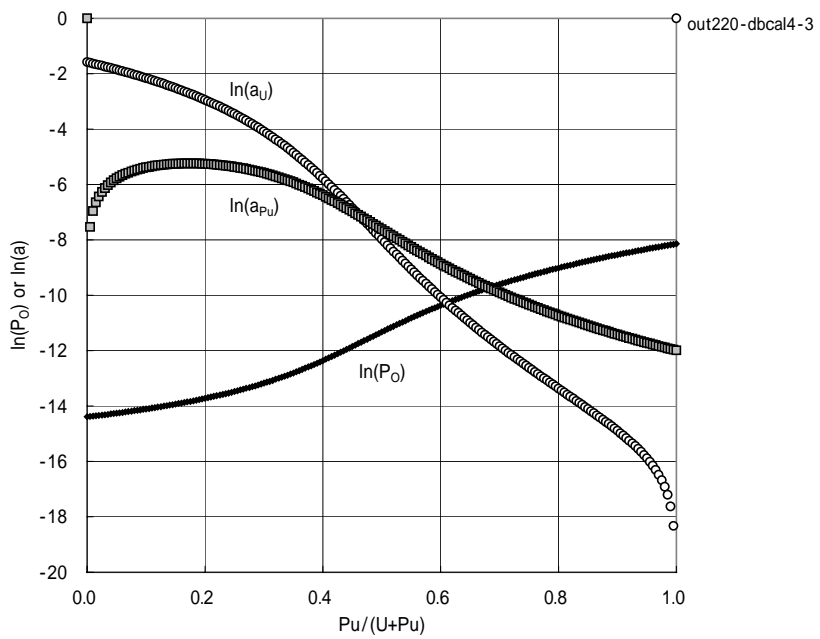


fig 5-43 (U,Pu) $O_{1.7}$ 系の酸素分圧 及び U, Pu の活量 (2600 C)

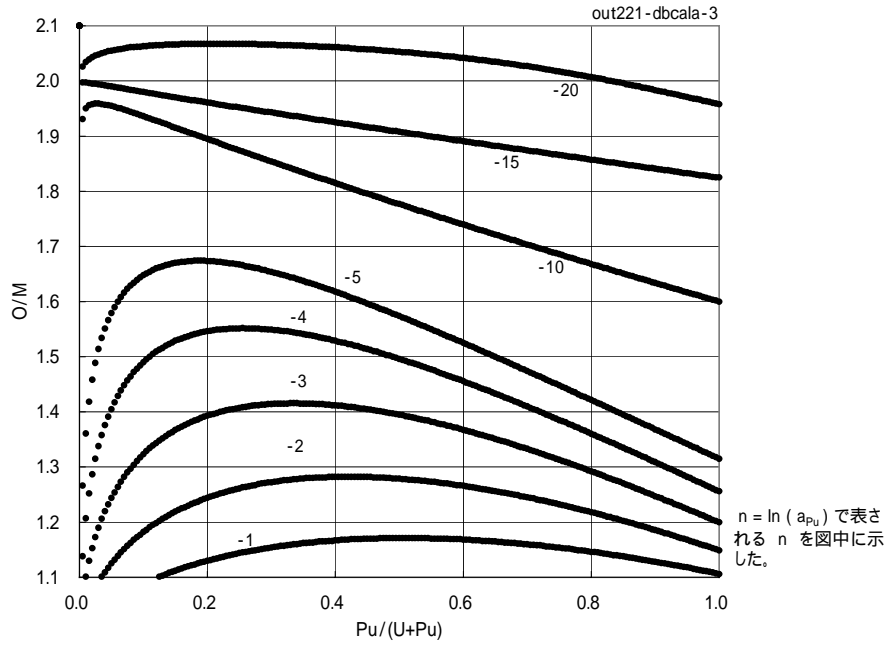


fig 5-44 U-Pu-O 系の Pu の等活量線 (2600 C)

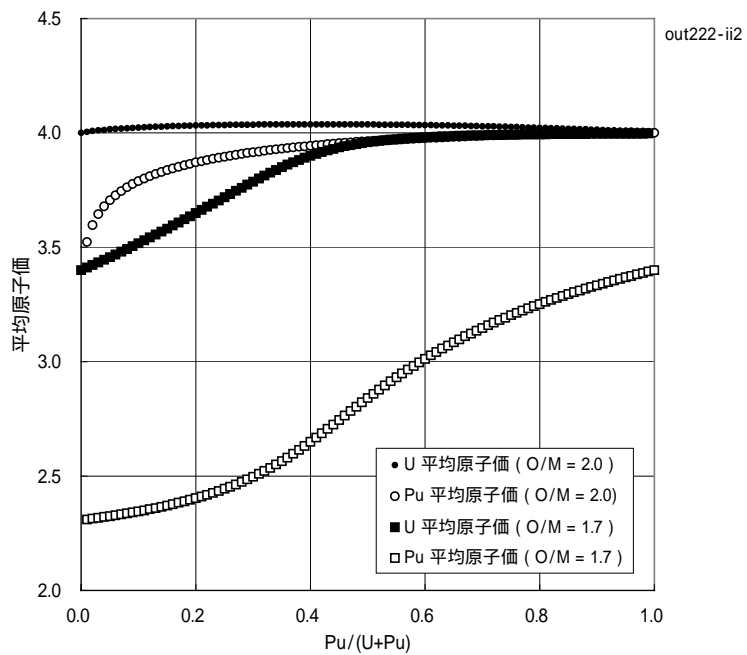


fig 5-45 U-Pu-O 系の U 及び Pu の平均原子価

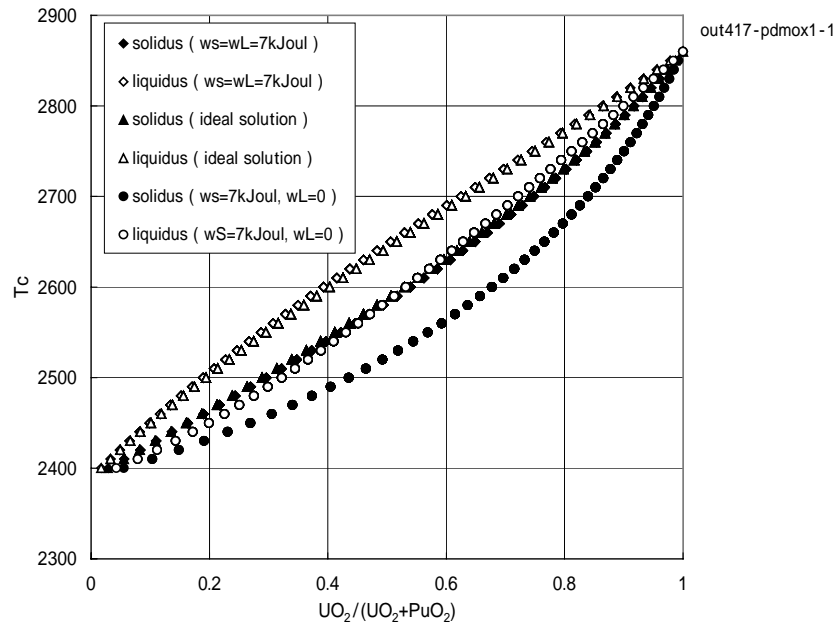


fig 5-46 UO_2 - PuO_2 系状態図のパラメータサーベイ

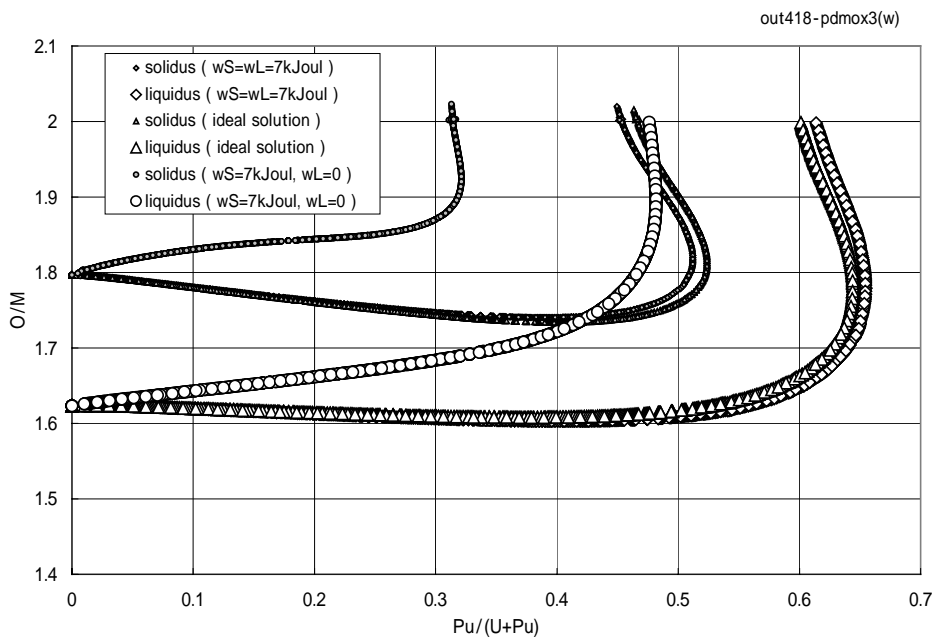


fig 5-47 U-Pu-O 系状態図のパラメータサーベイ (2600 C)

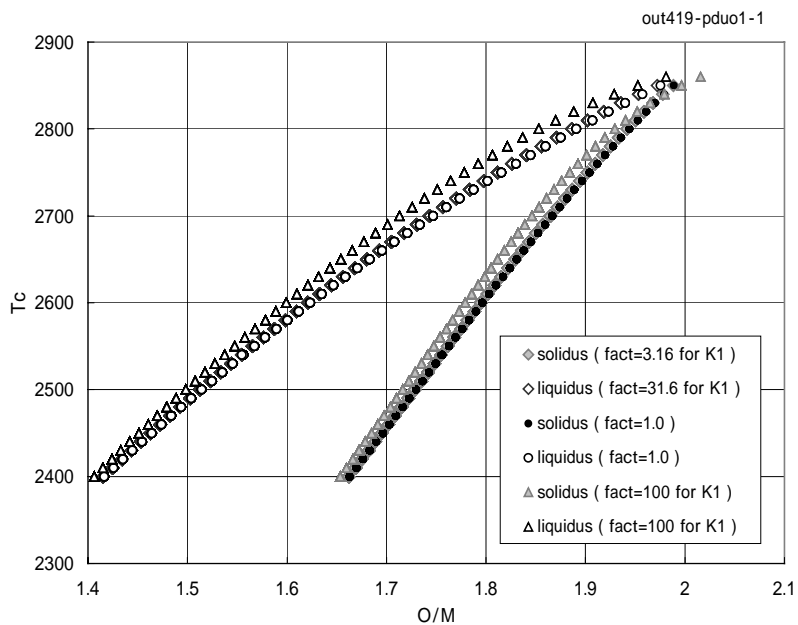


fig 5-48 Blackburn モデルのイオン間の平衡定数と状態図の関係 (U-O 系)

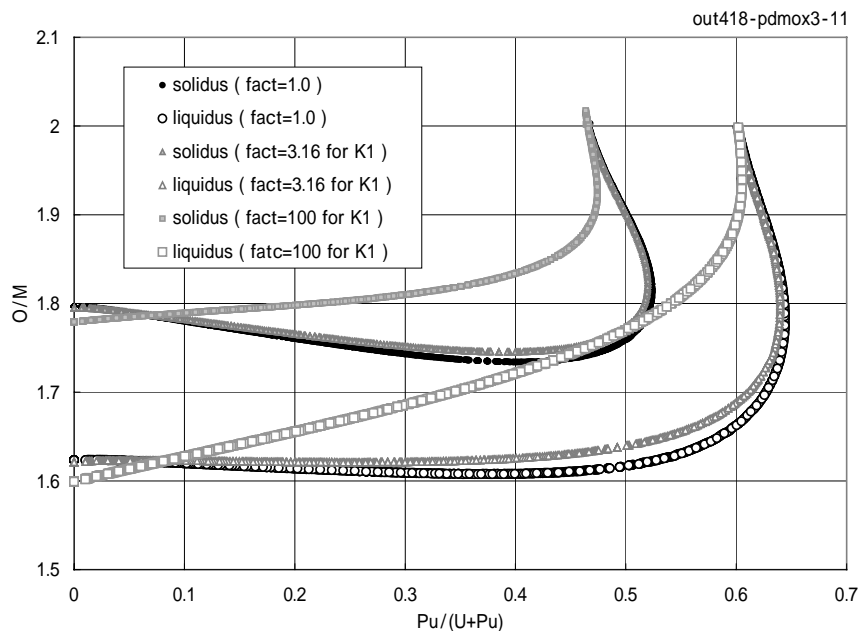


fig 5-49 Blackburn モデルのイオン間の平衡定数と状態図の関係 (U-Pu-O 系)

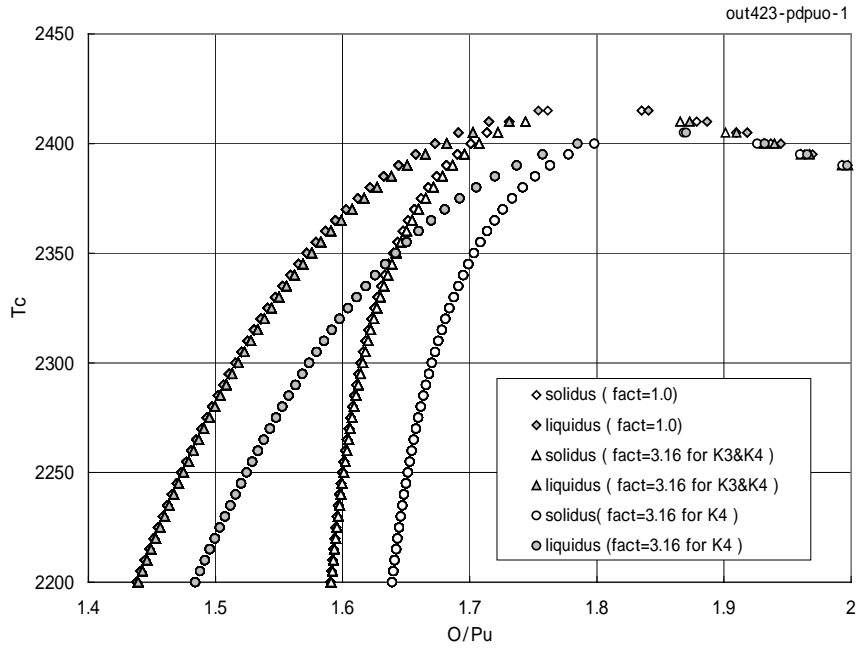


fig 5-50 Blackburn モデルのイオン間の平衡定数と状態図の関係 (Pu-O系)

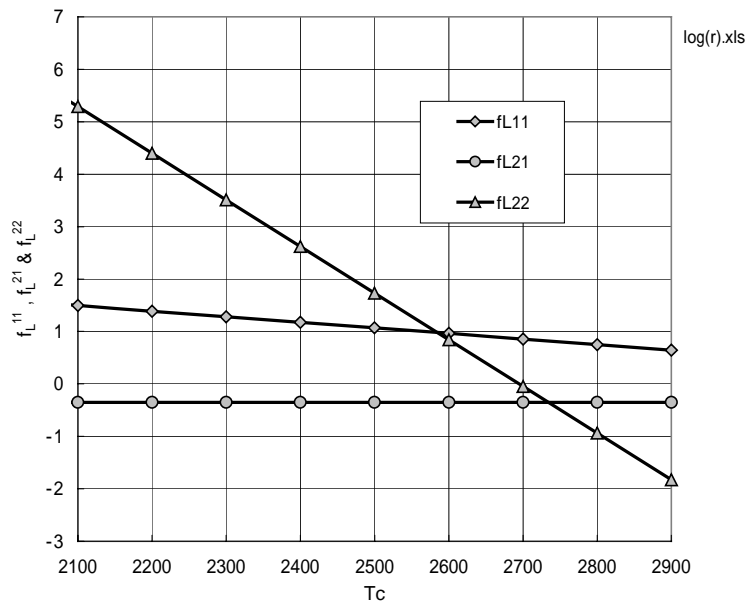


fig 5-51 イオン間平衡定数比

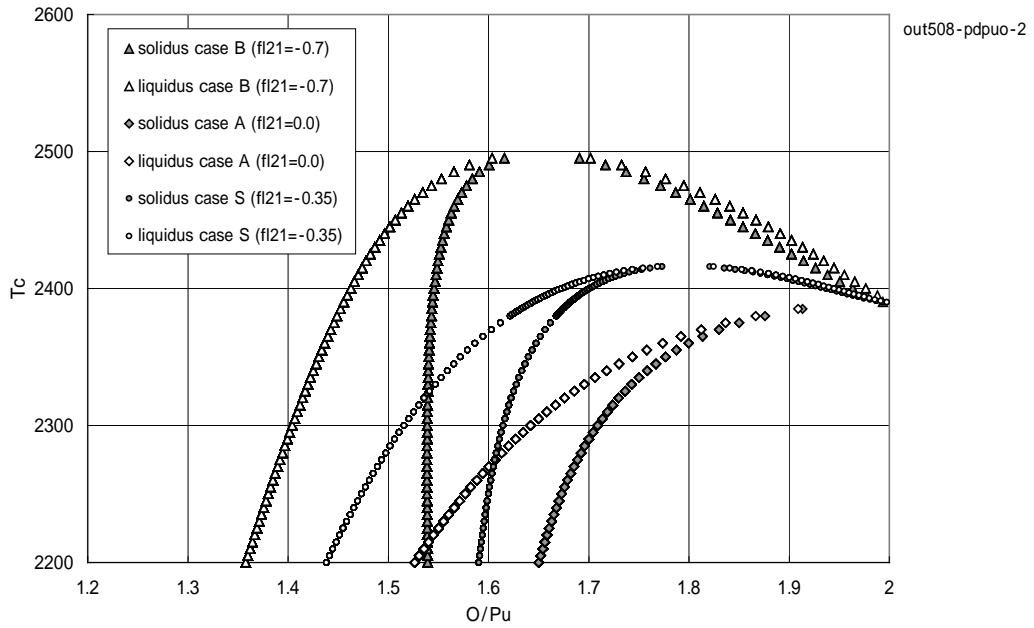


fig 5-52 Pu-O系の感度解析 (その1)

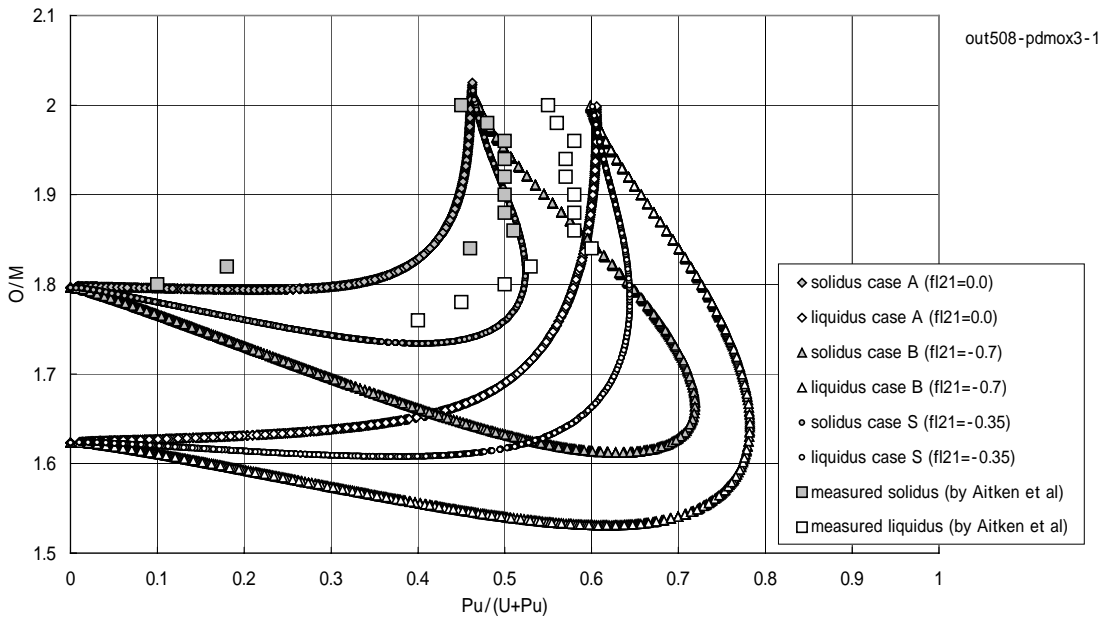


fig 5-53 U-Pu-O 系状態図の感度解析 (その1、2600C)

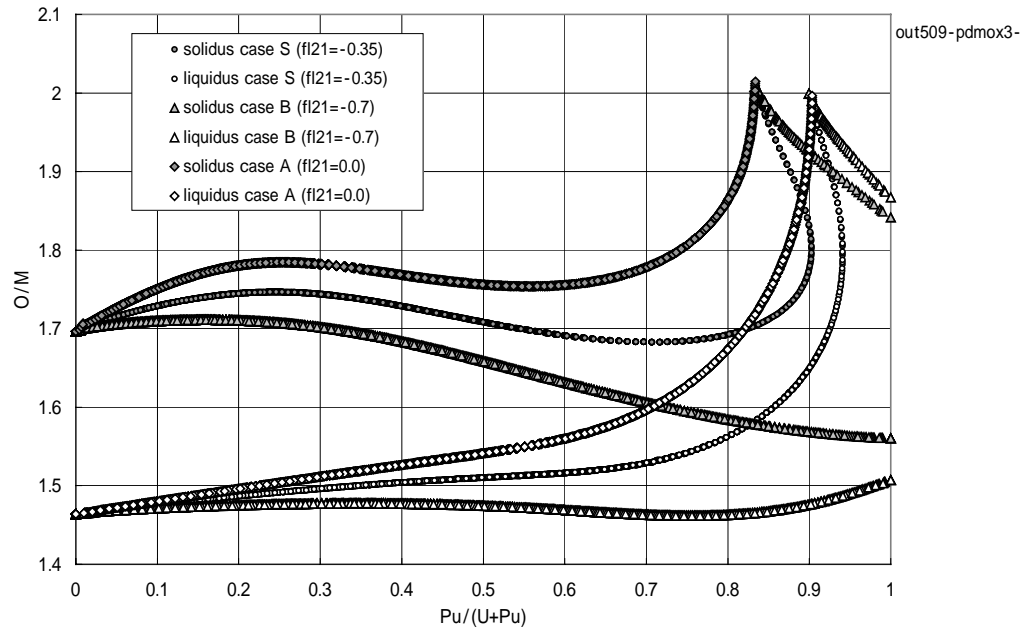


fig 5-54 U-Pu-O 系の感度解析 (その1, 2450 C)

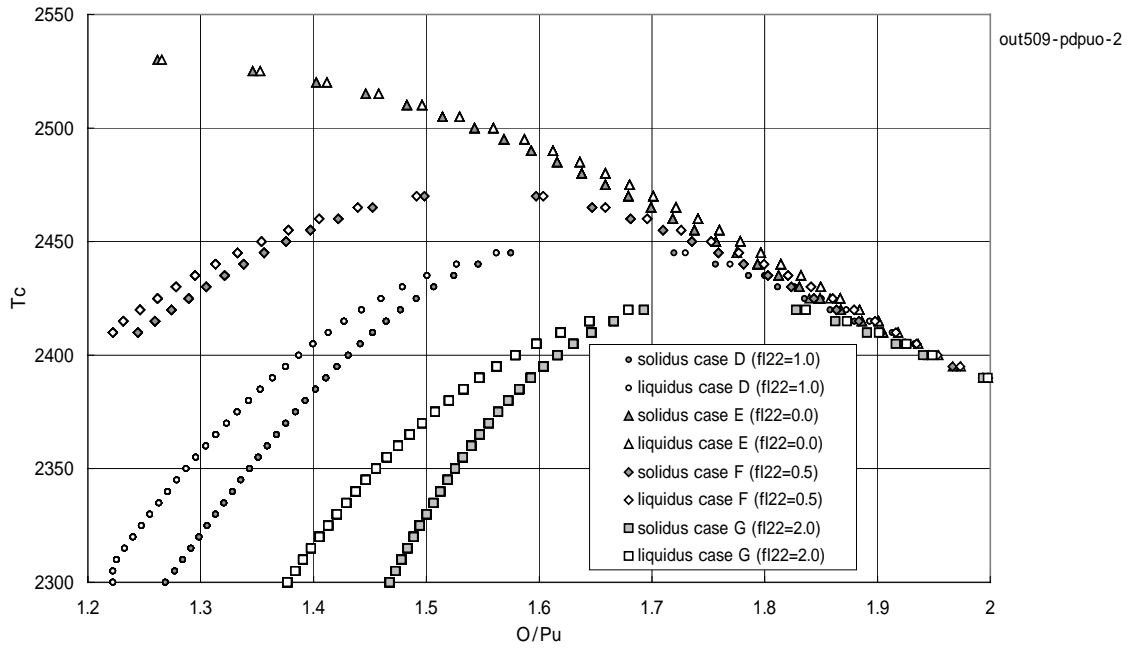


fig 5-55 Pu-O 系の感度解析 (その2)

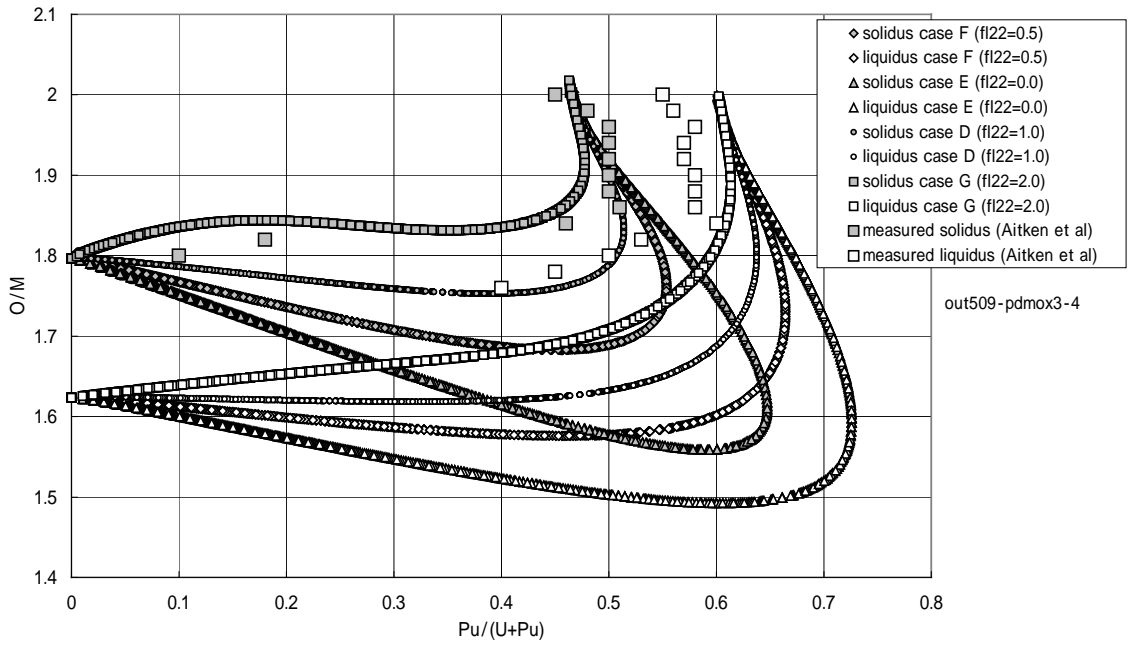


fig 5-56 U-Pu-O 系の感度解析 (その2)

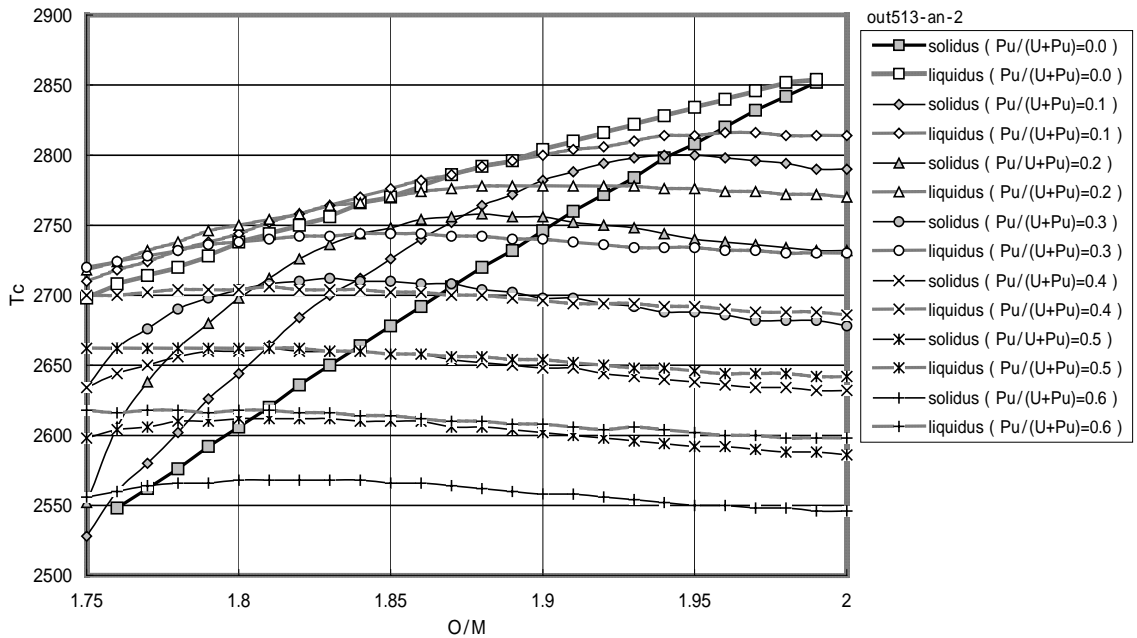


fig 5-57 U-Pu-O 系の等組成状態図 (その1)

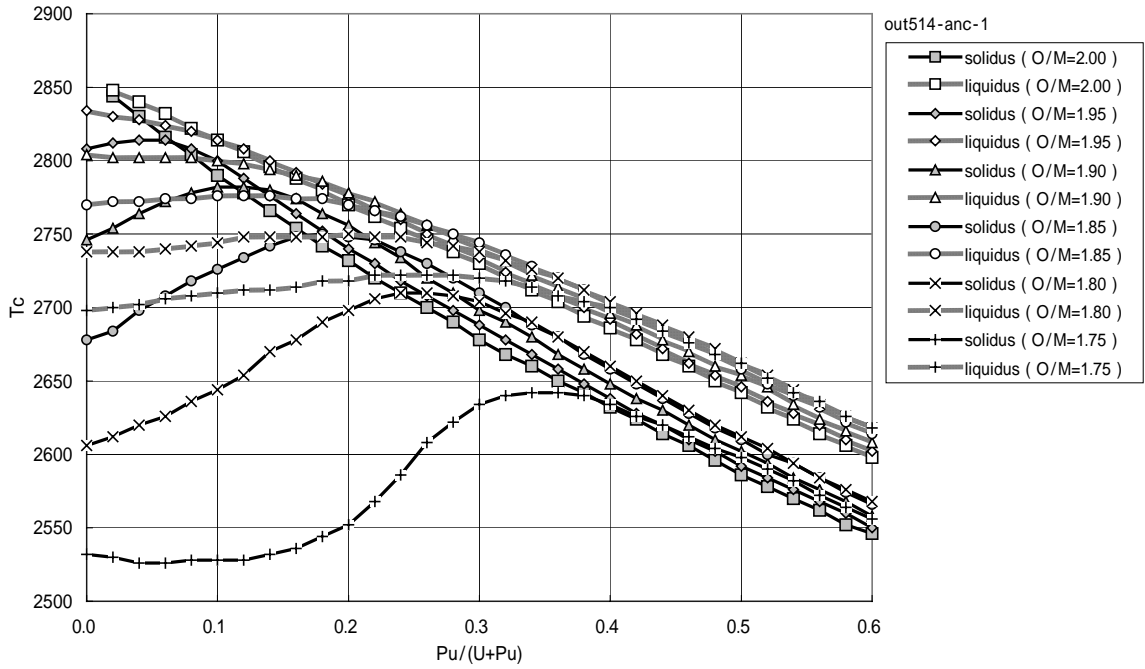


fig 5-58 U-Pu-O 系の等組成状態図 (その2)

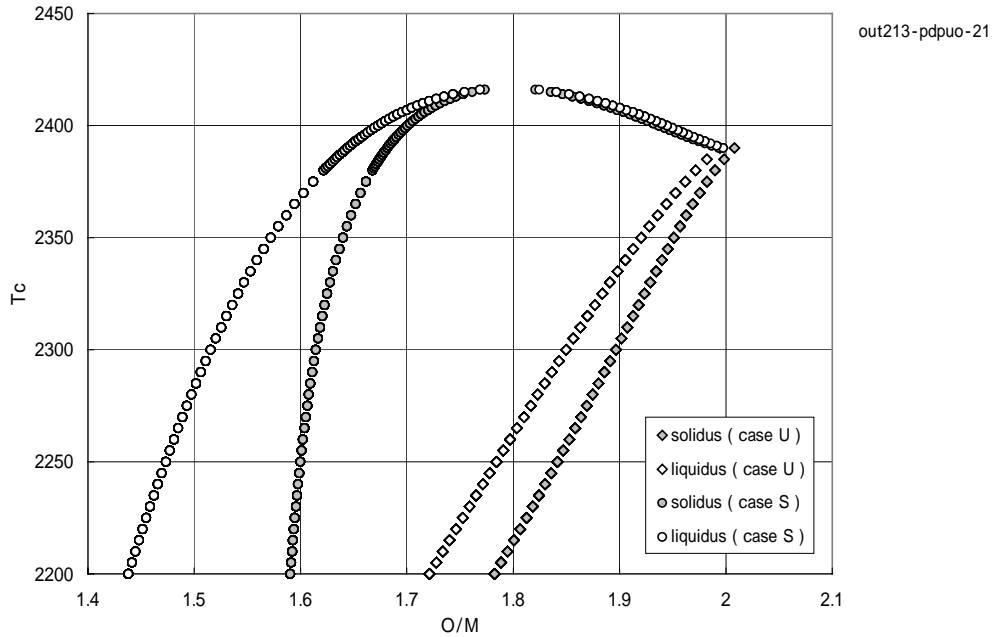


fig 5-59 Pu-O 系の状態図 (case U, $f_L^{21}=1.0$, $f_L^{22}=0.0$)

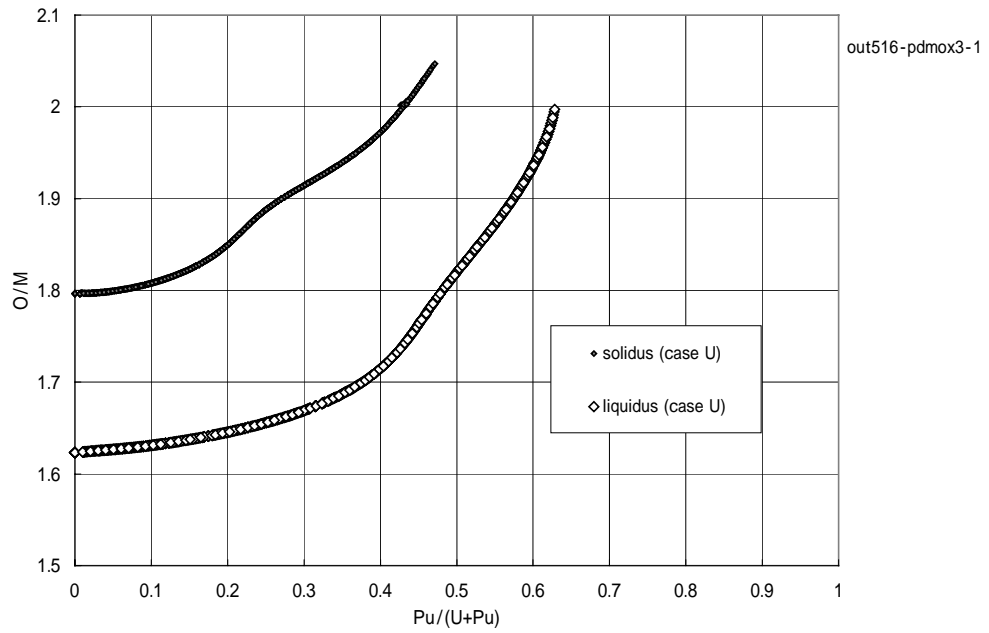


fig 5-60 U-Pu-O 系の状態図 (2600C) (case U , $f_L^{21}=1.0$, $f_L^{22}=0.0$)

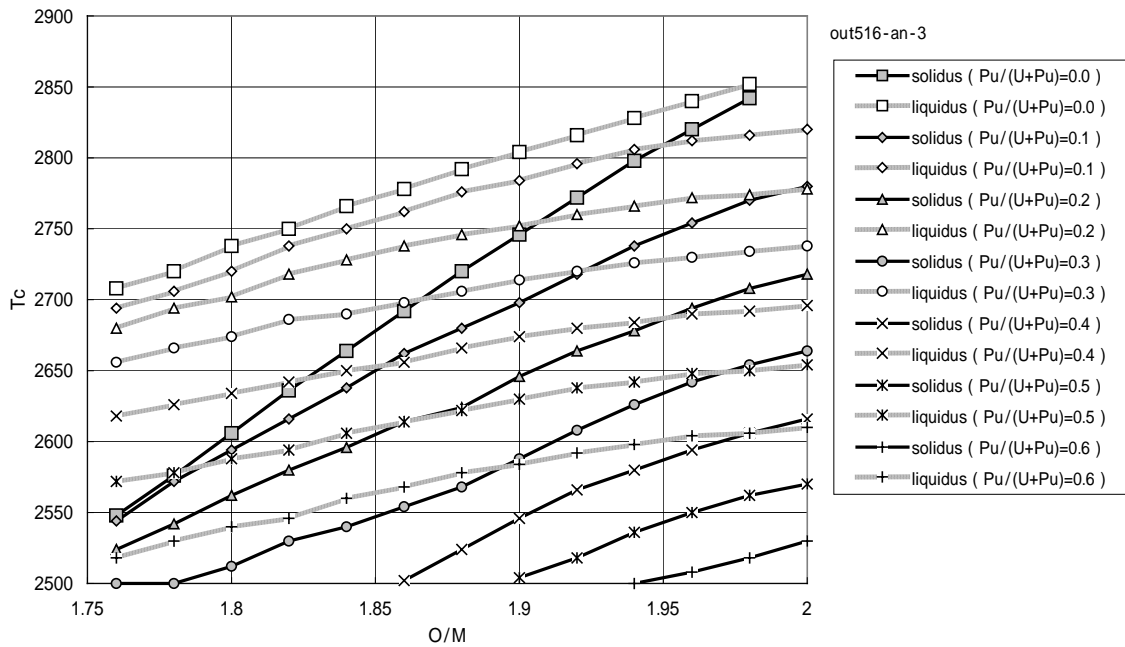


fig 5-61 U-Pu-O 系 等組成状態図 その3 (case U , $f_{L21}=1.0$, $f_{L22}=0.0$)

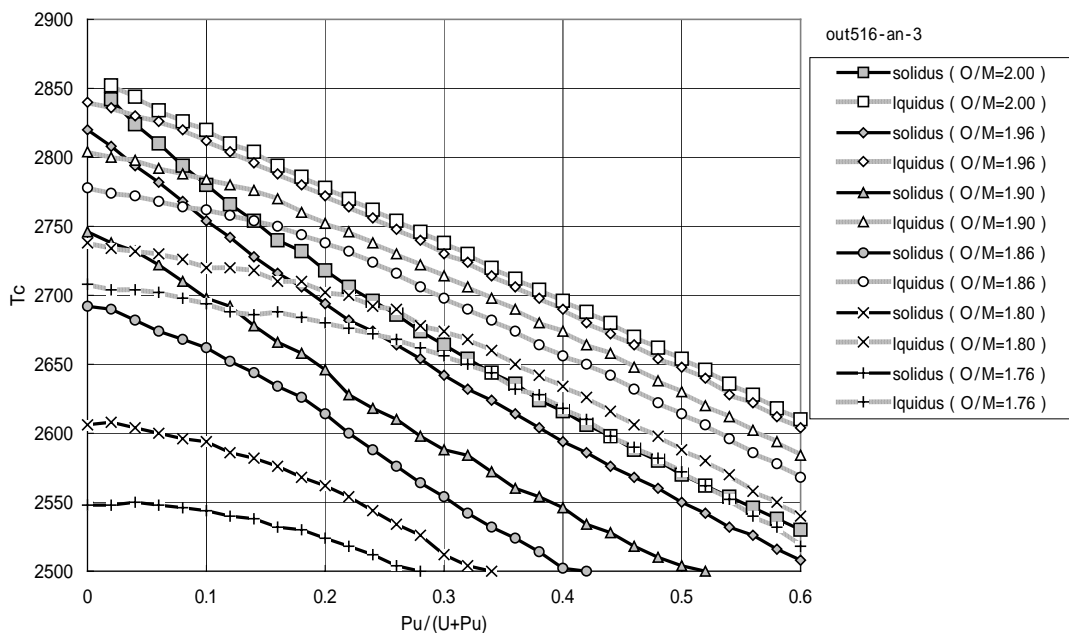


fig 5-62 U-Pu-O 系 等組成状態図 その4 (case U , $f_L^{21}=1.0$, $f_L^{22}=0.0$)

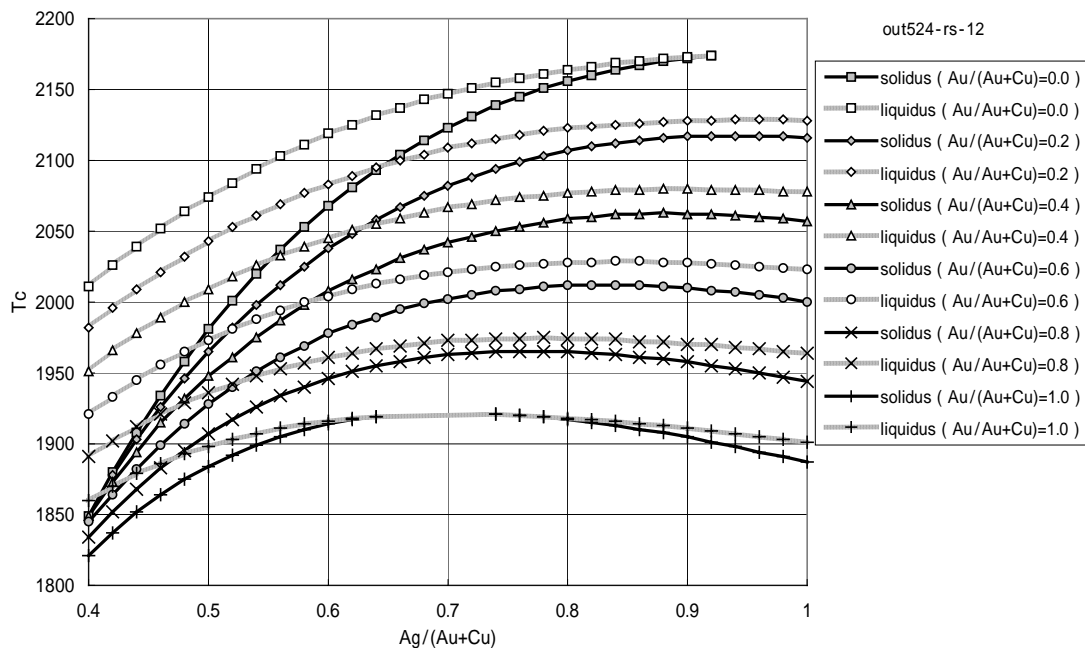


fig 5-63 Ag-Au-Cu 系を利用してU-Pu-O 系を模擬した状態図 (その1, 等組成状態図)

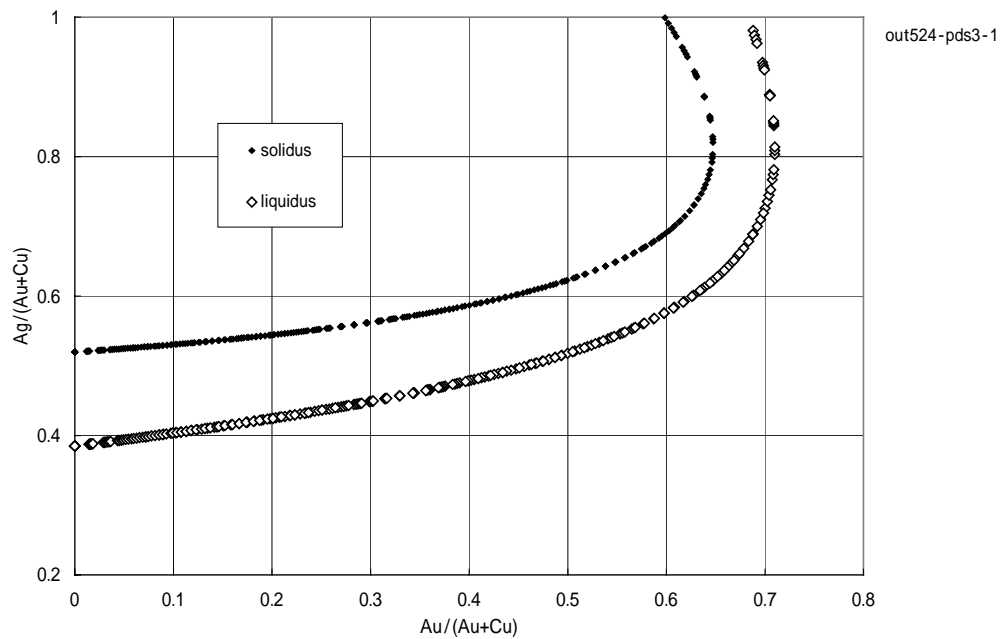


fig 5-64 Ag-Au-Cu 系を利用してU-Pu-O系を模擬した状態図 (その2, 2000 C 等温状態図)

付 録

付録

- A-1 状態図計算プログラム PD32 概要
- A-2 構成元素、化合物の自由エネルギー
- A.3 PD32 ソースプログラム

A-1 状態図計算プログラム PD32 概要

PD32 は以下のサブプログラムから構成されている。

(1) 状態図計算の制御サブプログラム

PDMOX1	U02-Pu02 理想溶液状態図計算
PDU01	U-0 系状態図計算
PDPU0	Pu-0 系状態図計算
PDMOX3	U-Pu-0 系状態図計算
PDAGAU1	Ag-Au 系状態図計算
PDS3	Ag-Au-Cu 系状態図計算

これらのサブプログラムは計算すべき温度、組成を設定し、計算後の 2 相の組成、自由エネルギー等を出力する。

(2) 状態図計算サブプログラム

SSCN	2 相 2 成分系状態図計算
SSCN3	2 相 3 成分系状態図計算
AX1 及び AX3	組成の制限条件が満たされていることの確認

(3) 燃料系の固相及び液相の自由エネルギー又は相互作用パラメータの計算サブプログラム

WUOS	固相 U-0 の W、自由エネルギーの計算
WUOL	液相 U-0 の W、自由エネルギーの計算
WPUOS	固相 P u-0 の W、自由エネルギーの計算
WPUOLL	液相 P u-0 の W、自由エネルギーの計算
GUPUOS	固相 U-Pu-0 系の自由エネルギーの計算
GUPUOL	固相 U-Pu-0 系の自由エネルギーの計算

(4) 固相及び液相の自由エネルギー計算関数サブプログラム

- ・系の自由エネルギーは sub SSCN 又は sub SSCN3 の内容を修正することなしに様々の系の計算が可能ないように、関数サブプログラムとした。
- ・関数サブプログラムは

$$F = A_p \cdot G_s + (1 - A_p) \cdot G_L$$

の形式でコーディングされている。

ここで、F は 2 相共存系の自由エネルギー、G_S は固相の自由エネルギー、G_L は液相の自由エネルギー、A_p は固相の存在割合 である。

- ・自由エネルギーの関数サブプログラムリストを別表に示す。

(5) 元素、化合物の自由エネルギー及び酸素分圧計算サブプログラム

GG02	酸素分子の自由エネルギー計算
GOXA	原子状酸素の自由エネルギー計算
GUMET	固体又は液体金属 U の自由エネルギー計算
GPUMET	固体又は液体金属 Pu の自由エネルギー計算
GU02S	固体 U02 の自由エネルギー計算
GPU02S	固体 Pu02 の自由エネルギー計算
GU02L	液体 U02 の自由エネルギー計算

GPU02L	液体 PuO ₂ の自由エネルギー計算
SGAGS	融点近傍の固体 Ag の自由エネルギー計算
SGAGL	融点近傍の液体 Ag の自由エネルギー計算
SGAUS	融点近傍の固体 Au の自由エネルギー計算
SGAUL	融点近傍の液体 Au の自由エネルギー計算
SGCUS	融点近傍の固体 Cu の自由エネルギー計算
SGCUS	融点近傍の液体 Cu の自由エネルギー計算
MOX0XPZ	U-O , Pu-O, U-Pu-O 系の酸素分圧の計算

(6) 支援サブプログラム

・状態図計算に必要とされるサブルーチン呼び出し、計算された諸量の組成及び温度依存性の妥当性を確認するためのもので、ユーザーが必要に応じて内容を変更しても差し支えなく、内容の変更によって状態図計算には影響を及ぼさない。詳細はプログラム参照

(7) その他のサブプログラム・SSCN 又は SSCN3 で得られた解が 2 相共存状態ならば、構成成分のケミカルポテンシャルは 2 相間で合い等しい。このことを確認するために以下のサブプログラムを内蔵している。

CHK1	U-Pu-O 系のケミカルポテンシャルの一致度の確認
CHK2	Ag-Au-Cu 系のケミカルポテンシャルの一致度の確認
CHK3	U-O 系のケミカルポテンシャルの一致度の確認

・U-Pu-O 系で組成の関数として離散的に計算された自由エネルギーを用いて、必要とする組成の自由エネルギーを 3 次式の Lagrange 補間を行うサブプログラム LAG を内蔵している。

(8) メインプログラム

・状態図計算制御サブルーチン又は支援サブルーチンを選択する。

別表 自由エネルギーの関数サブプログラム

計算対象系	F	G _s	G _L
U-O 系	FUOSL1, FUOSL2, FUOL12	FUOS	FUOL1, FUOL2
Pu-O 系	FPUOSL	FPUOS	FPUOL
UO ₂ -PuO ₂ 系	FG01	FMOXS	FMOXL
U-Pu-O 系	FM3SL	FM3S	FM3L
Ag-Au 系	FG1	FS1	FL1
Ag-Au-Cu 系	FAAC	FS	FL

A-2 構成元素、化合物の自由エネルギー

(1) 酸素(O , O_2)、ウラン、プルトニウム

Ihsan Barin の監修した熱化学データ集⁽¹⁾を用いた。

(2) 固体 UO_2

室温から 2000K までは Cordfunke 監修の熱力学テーブル⁽²⁾、それ以上の温度に対しては R.A.Hein⁽³⁾ によるエンタルピーデータを使用し求めた。

(3) 固体 PuO_2

Cordfunke 監修の熱力学テーブルを用いた

(4) 液体 UO_2

L.Leibowitz 等⁽⁴⁾ による比熱、融解熱の測定結果を用いた

(5) 液体 PuO_2

Cordfunke 監修の熱力学テーブルを用いた

(6) 化合物の融点

UO_2 の融点は 3133.15 K (2860 C)

PuO_2 の融点は 2663 K (2389.85 C)

を用いた。

・上記の元素、化合物の自由エネルギーは温度の多項式で表し (fig A-1,A-2 に示す) 計算に用いた。

参考文献

(1) Ihsan Barin, Thermochemical data of pure substances, VCH(1989)

(2) E.H.P.Cordfunke, R.J.M.Konings ed. Thermochemical data for reactor materials and fission products, North-Holland(1990)

(3) R.A.Hein, L.H.Sjodahl and R.Szwarc J.Nuc.Mat. 25(1968)99

(4) L.Leibowitz, M.G.Chasanov, I.L.W.Mishler and D.F.Fischer J.Nuc.Mat. 39(1971)115

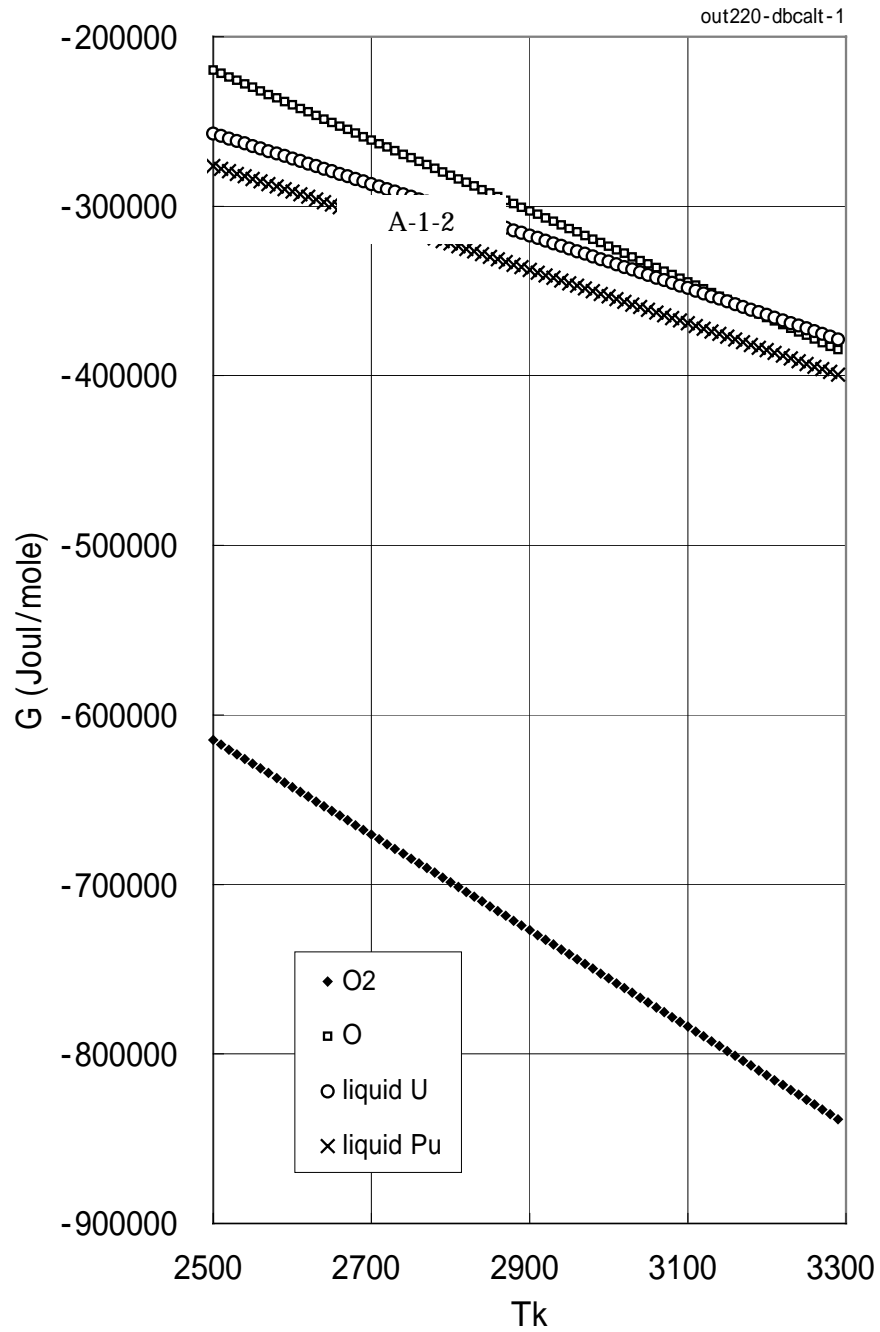


fig A-1 O, O₂, U, 及び Pu の自由エネルギー

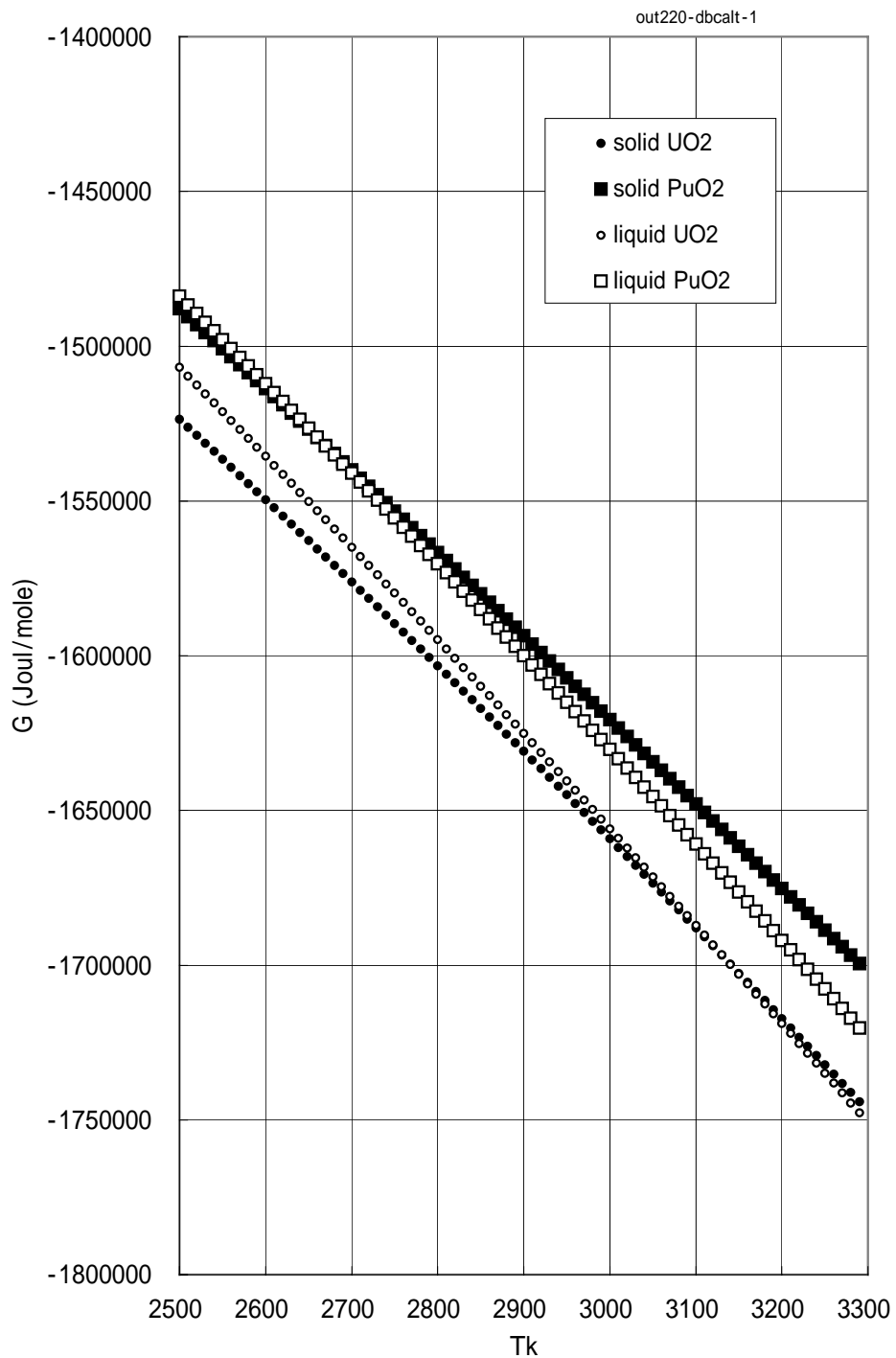


fig A-2 UO₂ 及び PuO₂ の自由エネルギー

JNC TN8410 2002-014

A.3 PD32 ソースプログラム

```

1: PROGRAM MAIN
2: IMPLICIT REAL*8 (A-H, O-Z)
3:c CALL PHDMOX1
4:c CALL PDAGAU1
5:c CALL PDAGCU1
6:c CALL PDUO1
7:c CALL PDUO2
8:c CALL PDPUIO
9: CALL PDMOX3
10:c CALL PUS3
11:c CALL PDS32
12:c CALL PDMOXR1
13:c CALL PDMOXR2
14:c CALL DBCAL3
15:c CALL DBCAL4
16:c CALL DBCPUO
17:c CALL DBUO
18:c CALL VPO
19:c CALL DBCAL
20:c CALL DBCAL2
21:c CALL DBCALT
22:c CALL DBCALA
23: STOP
24: END
25:C*****
26: SUBROUTINE PDMOXR1
27:CCCCC
28:C
29: IMPLICIT REAL*8 (A-H, O-Z)
30: COMMON/TZ/TK, XZ, YZ
31: COMMON/LLS/LS, NS
32: COMMON/GMS/XPUS, OMSS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
33: COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PLS(1001, 1001)
34: COMMON/LLL/LL, NL
35: COMMON/GML/XPUSL, OMSL, DXL, DML, XPRL(1001), OMRL(1001), GMSL(1001, 1001)
36: COMMON/AL/AUSL(1001, 1001), APUSL(1001, 1001), PLL(1001, 1001)
37: COMMON /GO3/GO, GU, GPU
38: TC=2600.0
39: TK=TC+273.15
40: CALL GUPUOS
41: CALL GUPUOL
42: OPEN(7, FILE=' T2600n', STATUS=' NEW' )
43: WRITE(7, *)LS, NS, LL, NL, TK, GO, GU, GPU, XPUS, OMSS, DX, DM
44: 1, XPUSL, OMSL, DXL, DML
45: DO 100 J=1, 1000
46: WRITE(7, *) OMR(J), OMRL(J)
47: 100 CONTINUE
48: DO 200 J=1, 1001
49: WRITE(7, *) XPR(J), XPRL(J)
50: 200 CONTINUE
51: DO 300 J=1, 1001
52: DO 400 K=1, 1000
53: WRITE(7, *) GMS(J, K), AUS(J, K), APUS(J, K), PLS(J, K), GMSL(J, K)
54: 1, AUSL(J, K), APUSL(J, K), PLL(J, K)
55: 400 CONTINUE
56: 300 CONTINUE
57: CLOSE(7)
58: RETURN
59: END
60:C*****
61:C*****
62: SUBROUTINE PDMOXR2
63:CCCCC
64:C
65: IMPLICIT REAL*8 (A-H, O-Z)
66: EXTERNAL FM3SI, FM3S, FM3I
67: DIMENSION X(2), Y(2), TS(2)
68: COMMON /GO3/GO, GU, GPU
69: COMMON/TZ/TK, XZ, YZ
70: COMMON/FL11/FL11, FL21, FL22, FL31
71: COMMON/LLS/LS, NS
72: COMMON/GMS/XPUS, OMSS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
73: COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PLS(1001, 1001)
74: COMMON/LLL/LL, NL
75: COMMON/GML/XPUSL, OMSL, DXL, DML, XPRL(1001), OMRL(1001), GMSL(1001, 1001)
76: COMMON/AL/AUSL(1001, 1001), APUSL(1001, 1001), PLL(1001, 1001)
77: READ(5, *)LLW, LO2, KOUT, TC1, TC2, DT
78: READ(5, *)NC, OMS, CV
79: READ(5, *)MC, XXPUS, CW
80: OPEN(7, FILE=' T2600n', STATUS=' OLD' )
81: READ(7, *)LS, NS, LL, NL, TK, GO, GU, GPU, XPUS, OMSS, DX, DM
82: 1, XPUSL, OMSL, DXL, DML
83: DO 100 J=1, 1000
84: READ(7, *) OMR(J), OMRL(J)
85: 100 CONTINUE
86: DO 200 J=1, 1001
87: READ(7, *) XPR(J), XPRL(J)
88: 200 CONTINUE
89: DO 300 J=1, 1001

```

```

90:      DO 400 K=1, 1000
91:      READ (7, *) GMS(J, K), AUS(J, K), APUS(J, K), PLS(J, K), GMSL(J, K)
92:      1, AUSL(J, K), APUSL(J, K), PLL(J, K)
93:      400 CONTINUE
94:      300 CONTINUE
95:      CLOSE(7)
96:c    CALL DBCAL3
97:c    RETURN
98:      IC=1
99:      R32=SQRT(3.0)/2.0
100:     OMMAX=2.1
101:     OMMIN=1.1
102:     XMAX=OMMAX/(1.0+OMMAX)
103:     XMIN=OMMIN/(1.0+OMMIN)
104:     YMAX=1.0-XMIN
105:     YMIN=0.0
106:     SMAX=1.0
107:     OMF=1.3
108:     TTC=2600.0
109:c
110:c c c c c c
111:     DO 3000 NZ=1, NC
112:     OM=OMS-CV*(NZ-1)
113:     IF(OM.LE.OMF) GO TO 2100
114:     XZ=OM/(1.0+OM)
115:     DO 2000 KZ=1, MC
116:     XPU=XXPUS+CW*(KZ-1)
117:     YZ=XPU*(1.0-XZ)
118:     ZZ=1.0-YZ-XZ
119:     IF(ZZ.LT.0.0) GO TO 2100
120:     CALL SSCN3(FM3SL, XMAX, XMIN, YMAX, YMIN, SMAX, X, Y, XZ, YZ, AP, FMIN,
121:     1 ITR, JPH, LLW)
122:     IF(JPH.LE.0.AND.L02.EQ.2) GO TO 2000
123:c    IF(JPH.EQ.2900) GO TO 2000
124:     IF(KOUT.EQ.1) THEN
125:c c c c c c CONVERSION TO TRIANGULAR COORDINATE
126:     XPS=Y(1)
127:     XPL=Y(2)
128:     XUS=1.0-X(1)-Y(1)
129:     XUL=1.0-X(2)-Y(2)
130:     X31=(1.0+XPS-XUS)/2.0
131:     Y31=R32*(1.0-XPS-XUS)
132:     X32=(1.0+XPL-XUL)/2.0
133:     Y32=R32*(1.0-XPL-XUL)
134:     CPS=ETIME(TS)
135:c c c c c c
136:     WRITE (6, 120) TTC, XZ, YZ, ITR, JPH, X(1), X(2), Y(1), Y(2), AP, FMIN
137:     1, X31, Y31, X32, Y32, CPS
138:     ELSE
139:     CPS=ETIME(TS)
140:     OCS=X(1)/(1.0-X(1))
141:     OCL=X(2)/(1.0-X(2))
142:     XXPS=Y(1)/(1.0-X(1))
143:     XXPL=Y(2)/(1.0-X(2))
144:     OMZ=OM
145:     XPUZ=XPU
146:     IF(IC.EQ.0) THEN
147:     WRITE (6, 120) TTC, OMZ, XPUZ, ITR, JPH, X(1), X(2), Y(1), Y(2), AP, FMIN
148:     1, XXPS, OCS, XXPL, OCL, DAU, DPU, DPP, CPS
149:     ELSE
150:     IF(JPH.EQ.2) THEN
151:     CALL CHK1(OCS, XXPS, OCL, XXPL, DAU, DPU, DPP)
152:     ELSE
153:     DAU=-50.0
154:     DPU=-50.0
155:     DPP=-50.0
156:     END IF
157:     WRITE (6, 130) TTC, OMZ, XPUZ, ITR, JPH, X(1), X(2), Y(1), Y(2), AP, FMIN
158:     1, XXPS, OCS, XXPL, OCL, DAU, DPU, DPP, CPS
159:     END IF
160:     END IF
161:     2000 CONTINUE
162:     2100 CONTINUE
163:     3000 CONTINUE
164:     1000 CONTINUE
165:     120 FORMAT (F12.1, 2F13.8, 215, 5F13.6, E30.18, 5F13.6)
166:     130 FORMAT (F12.1, 2F13.8, 215, 5F13.6, E30.18, 8F13.6)
167:     140 FORMAT (20E15.7)
168:     RETURN
169:     END
170:c *****
171:     SUBROUTINE PDMOX3
172:c c c c c c
173:c
174:     IMPLICIT REAL*8 (A-H, O-Z)
175:     EXTERNAL FM3SL, FM3S, FM3L
176:     DIMENSION X(2), Y(2), TS(2)
177:     COMMON/TZ/TK, XZ, YZ
178:     COMMON /GO3/GO, GU, GPU

```

```

179:      COMMON/FL11/FL11, FL21, FL22, FL31
180:      READ (5, *) LLW, L02, KOUT, TC1, TC2, DT
181:      READ (5, *) NC, OMS, CV
182:      READ (5, *) MC, XPUS, CW
183:c     WRITE (6, *) FL11, FL21, FL22, FL31
184:c     WRITE (6, *) LLW, L02, TC1, TC2, DT
185:c     WRITE (6, *) NC, OMS, CV
186:c     WRITE (6, *) MC, XPUS, CW
187:      IC=1
188:      R32=SQRT (3.0)/2.0
189:      OMMAX=2.1
190:      OMMIN=1.1
191:      XMAX=OMMAX/(1.0+OMMAX)
192:      XMIN=OMMIN/(1.0+OMMIN)
193:      YMAX=1.0-XMIN
194:      YMIN=0.0
195:      SMAX=1.0
196:      OMF=1.3
197:C
198:      ITTN = INT ((TC2-TC1+0.001)/DT)+1
199:      DO 1000 IT=1, ITTN
200:      TTC=TC1+(IT-1)*DT
201:      TK=TTC+273.15
202:CCCC  G OF COMPONENT
203:c     CALL BOXA (TK, GO, GFO)
204:c     CALL GUMET (TK, GU, GFU)
205:c     CALL GPUMET (TK, GPU, GFPU)
206:      CALL GUPUOS
207:C     write (6, *) ttc
208:      CALL GUPUOL
209:C     WRITE (6, *) TK
210:CCCCCCC
211:      DO 3000 NZ=1, NC
212:      OM=OMS-CV*(NZ-1)
213:      IF (OM.LE.OMF) GO TO 2100
214:      XZ=OM/(1.0+OM)
215:      DO 2000 KZ=1, MC
216:      XPU=XPUS+CW*(KZ-1)
217:      YZ=XPU*(1.0-XZ)
218:      77=1.0-Y7-XZ
219:      IF (ZZ.LT.0.0) GO TO 2100
220:C     QS=FM3S (XZ, YZ)
221:C     QL=FM3L (XZ, YZ)
222:c     WRITE (6, *) QS, QL
223:      CALL SSCN3 (FM3SL, XMAX, XMIN, YMAX, YMIN, SMAX, X, Y, XZ, YZ, AP, FMIN,
224:      1 ITR, JPH, LLW)
225:      IF (JPH.LE.0.AND.L02.EQ.2) GO TO 2000
226:C     IF (JPH.EQ.2900) GO TO 2000
227:      IF (KOUT.EQ.1) THEN
228:CCCCCC CONVERSION TO TRIANGULAR COORDINATE
229:      XPS=Y (1)
230:      XPL=Y (2)
231:      XUS=1.0-X (1)-Y (1)
232:      XUL=1.0-X (2)-Y (2)
233:      X31=(1.0+XPS-XUS)/2.0
234:      Y31=R32*(1.0-XPS-XUS)
235:      X32=(1.0+XPL-XUL)/2.0
236:      Y32=R32*(1.0-XPL-XUL)
237:      CPS=ETIME (TS)
238:CCCCCC
239:      WRITE (6, 120) TTC, XZ, YZ, ITR, JPH, X (1), X (2), Y (1), Y (2), AP, FMIN
240:      1 , X31, Y31, X32, Y32, CPS
241:      ELSE
242:      CPS=ETIME (TS)
243:      OCS=X (1)/(1.0-X (1))
244:      OCL=X (2)/(1.0-X (2))
245:      XXPS=Y (1)/(1.0-X (1))
246:      XXPL=Y (2)/(1.0-X (2))
247:      OMZ=OM
248:      XPUZ=XPU
249:      IF (IC.EQ.0) THEN
250:      WRITE (6, 120) TTC, OMZ, XPUZ, ITR, JPH, X (1), X (2), Y (1), Y (2), AP, FMIN
251:      1 , XXPS, OCS, XXPL, OCL, CPS
252:      ELSE
253:      CALL CHK1 (OCS, XXPS, OCL, XXPL, DAU, DPU, DPP)
254:      WRITE (6, 130) TTC, OMZ, XPUZ, ITR, JPH, X (1), X (2), Y (1), Y (2), AP, FMIN
255:      1 , XXPS, OCS, XXPL, OCL, DAU, DPU, DPP, CPS
256:      END IF
257:      END IF
258: 2000 CONTINUE
259: 2100 CONTINUE
260: 3000 CONTINUE
261: 1000 CONTINUE
262: 120 FORMAT (F12.1, 2F13.8, 2I5, 5F13.6, E30.18, 5F13.6)
263: 130 FORMAT (F12.1, 2F13.8, 2I5, 5F13.6, E30.18, 8F13.6)
264:      RETURN
265:      END
266:C*****
267:C*****

```

```

268: SUBROUTINE CHK1 (OMS, XRS, OML, XRL, DAU, DPU, DPP)
269: IMPLICIT REAL*8 (A-H, O-Z)
270: COMMON/LLS/LS, NS
271: COMMON/GMS/XPUS, OMSS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
272: COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PLS(1001, 1001)
273: COMMON/LLL/LL, NL
274: COMMON/GML/XPUSL, OMSL, DXL, DML, XPRL(1001), OMRL(1001), GMSL(1001, 1001)
275: COMMON/AL/AUSL(1001, 1001), APUSL(1001, 1001), PLL(1001, 1001)
276: COMMON /G03/GO, GU, GPU
277: COMMON/TZ/TK, XZ, YZ
278: DIMENSION UU(4), V1(4), V2(4), V3(4), AA(4), AB(4), AC(4), WW(4)
279: N=4
280: EPS=1.0D-20
281: RG=8.313606
282: RT=R0*TK
283: CCCCCCCC FOR SOLID PHASE
284: KOM=INT((OMSS-OMS)/DM)+1
285: KPU=INT((XRS-XPUS)/DX)+1
286: KP2=NS-2
287: C*****
288: IF (KPU.GT.1.AND.KPU.LT.KP2) THEN
289: KS=KPU-1
290: ELSE
291: IF (KPU.EQ.1) KS=1
292: IF (KPU.GE.KP2) KS=NS-3
293: END IF
294: LOM=KOM-2
295: c write(6,*) kom, kpu, kp2, ks, lom
296: C*****
297: DO 100 K=1, N
298: WW(K)=XPR(KS+K-1)
299: DO 200 J=1, N
300: UU(J)=OMR(LOM+J)
301: V1(J)=AUS(KS+K-1, LOM+J)
302: V2(J)=APUS(KS+K-1, LOM+J)
303: V3(J)=PLS(KS+K-1, LOM+J)
304: c nnnn=ks+k-1
305: c llll=lom+j
306: c write(6,*) nnnn, llll, uu(j), v1(j), v2(j), v3(j), oms
307: 200 CONTINUE
308: CALL LAG(N, UU, V1, OMS, A1)
309: CALL LAG(N, UU, V2, OMS, A2)
310: CALL LAG(N, UU, V3, OMS, A3)
311: AA(K)=A1
312: AB(K)=A2
313: AC(K)=A3
314: c WRITE(6,*) A1, A2, A3
315: 100 CONTINUE
316: CALL LAG(N, WW, AA, XRS, A1)
317: CALL LAG(N, WW, AB, XRS, A2)
318: CALL LAG(N, WW, AC, XRS, A3)
319: CUS=GU+RT*A1
320: CPS=GPU+RT*A2
321: COS=GO+RT*A3
322: c WRITE(6,*) ww(1), ww(2), ww(3), ww(4), CUS, CPS, COS, xrs
323: CCCCCCCC FOR LIQUID PHASE
324: KOM=INT((OMSL-OML)/DML)+1
325: KPU=INT((XRL-XPUSL)/DXL)+1
326: KP2=NL-2
327: C*****
328: IF (KPU.GT.1.AND.KPU.LT.KP2) THEN
329: KS=KPU-1
330: ELSE
331: IF (KPU.EQ.1) KS=1
332: IF (KPU.GE.KP2) KS=NS-3
333: END IF
334: LOM=KOM-2
335: C*****
336: DO 300 K=1, N
337: WW(K)=XPRL(KS+K-1)
338: DO 400 J=1, N
339: UU(J)=OMRL(LOM+J)
340: V1(J)=AUSL(KS+K-1, LOM+J)
341: V2(J)=APUSL(KS+K-1, LOM+J)
342: V3(J)=PLL(KS+K-1, LOM+J)
343: 400 CONTINUE
344: CALL LAG(N, UU, V1, OML, A1)
345: CALL LAG(N, UU, V2, OML, A2)
346: CALL LAG(N, UU, V3, OML, A3)
347: AA(K)=A1
348: AB(K)=A2
349: AC(K)=A3
350: 300 CONTINUE
351: CALL LAG(N, WW, AA, XRL, A1)
352: CALL LAG(N, WW, AB, XRL, A2)
353: CALL LAG(N, WW, AC, XRL, A3)
354: CUL=GU+RT*A1
355: CPL=GPU+RT*A2
356: COL=GO+RT*A3

```

```

357:c      WRITE(6,*) ww(1), ww(2), ww(3), ww(4), CUL, CPL, COL
358:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
359:      DAU=DLOG10(DABS(2.0*(CUS-CUL)/(CUS+CUL))+EPS)
360:      DPU=DLOG10(DABS(2.0*(CPS-CPL)/(CPS+CPL))+EPS)
361:      DPP=DLOG10(DABS(2.0*(COS-COL)/(COS+COL))+EPS)
362:      RETURN
363:      END
364:C*****
365:C*****
366:      FUNCTION FM3SL(AP, X, Y)
367:      IMPLICIT REAL*8 (A-H, O-Z)
368:      DIMENSION X(2), Y(2)
369:      FM3SL=AP*FM3S(X(1), Y(1)) + (1.0-AP)*FM3L(X(2), Y(2))
370:      RETURN
371:      END
372:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
373:C*****
374:C*****
375:      FUNCTION FM3S(X, Y)
376:      IMPLICIT REAL*8 (A-H, O-Z)
377:      COMMON/TZ,TK,XZ,YZ
378:      COMMON/LLS,LL,NN
379:      COMMON/GMS/XPUS,OMS,DX,DM,XPR(1001),OMR(1001),GMS(1001,1001)
380:      DIMENSION UU(4),VV(4),WW(4),GG(4)
381:      N=4
382:      EPS=1.0D-30
383:      GTT=5.0D5
384:      X32=OMS/(1.0+OMS)
385:      IF(X.GT.1.0) X=1.0
386:      IF(X.LT.0.0) X=0.0
387:      IF(Y.GT.1.0) Y=1.0
388:      IF(Y.LT.0.0) Y=0.0
389:      XXX=X
390:      OM=X/(1.0-X+EPS)
391:      XPU=Y/(1.0-X+EPS)
392:      KOM=INI((OMS-OM)/DM)+1
393:      KPU=INT((XPU-XPUS)/DX)+1
394:      KO2=LL-2
395:      KP2=NN-2
396:c      write(6,*) x, y, ll, nn, xpus, oms, dx, dm, om, xpu, kom, kpu, ko2, kp2
397:C*****
398:      IF(OM.LT.OMS.AND.OM.GT.OMR(LL)) THEN
399:C*****
400:      IF(KPU.GT.1.AND.KPU.LT.KP2) THEN
401:      KS=KPU-1
402:      ELSE
403:      IF(KPU.EQ.1) KS=1
404:      IF(KPU.GE.KP2) KS=NN-3
405:      END IF
406:C*****
407:      IF(KOM.GT.1.AND.KOM.LT.KO2) THEN
408:      LOM=KOM-2
409:      ELSE
410:      IF(KOM.EQ.1) LOM=1
411:      IF(KOM.GE.KO2) LOM=LL-3
412:      END IF
413:c      write(6,*) ks, lom
414:C*****
415:      DO 100 K=1, N
416:      WW(K)=XPR(KS+K-1)
417:c      ka=ks+k-1
418:      DO 200 J=1, N
419:      UU(J)=OMR(LOM+J)
420:      VV(J)=GMS(KS+K-1, LOM+J)
421:c      lj=lom+j
422:c      write(6,*) ka, lj, uu(j), vv(j), ww(k)
423: 200 CONTINUE
424:      CALL LAG(N, UU, VV, OM, AA)
425:      GG(K)=AA
426:c      write(6,*) k, gg(k)
427: 100 CONTINUE
428:      CALL LAG(N, WW, GG, XPU, GG3)
429:      FM3S=GG3
430:c      write(6,*) gg3
431:C*****
432:      ELSE
433:      IF(OM.GE.OMS) THEN
434:      DGG=(GTT-GMS(KPU,1))/(1.0-X32)
435:      FM3S=GMS(KPU,1)+DGG*(XXX-X32)
436:      ELSE
437:      XXR=OMR(LL)/(1.0+OMR(LL))
438:      DGG=(GTT-GMS(KPU,LL))/XXR
439:      FM3S=GMS(KPU,LL)+DGG*(XXR-XXX)
440:      END IF
441:      END IF
442:      RETURN
443:      END
444:C*****
445:C*****

```

```

446: FUNCTION FM3L(X, Y)
447: IMPLICIT REAL*8 (A-H, O-Z)
448: COMMON/TZ/TK, XZ, YZ
449: COMMON/LLS/LL, NN
450: COMMON/GML/XPUS, OMS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
451: DIMENSION UU(4), VV(4), WW(4), GG(4)
452: N=4
453: EPS=1.0D-30
454: GTT=5.0D5
455: X32=OMS/(1.0+OMS)
456: IF(X.GT.1.0) X=1.0
457: IF(X.LT.0.0) X=0.0
458: IF(Y.GT.1.0) Y=1.0
459: IF(Y.LT.0.0) Y=0.0
460: XXX=X
461: OM=X/(1.0-X+EPS)
462: XPU=Y/(1.0-X+EPS)
463: KOM=INT((OMS-OM)/DM)+1
464: KPU=INT((XPU-XPUS)/DX)+1
465: KO2=LL-2
466: KP2=NN-2
467: C*****
468: IF(OM.LT.OMS.AND.OM.GT.OMR(LL)) THEN
469: C*****
470: IF(KPU.GT.1.AND.KPU.LT.KP2) THEN
471: KS=KPU-1
472: ELSE
473: IF(KPU.EQ.1) KS=1
474: IF(KPU.GE.KP2) KS=NN-3
475: END IF
476: C*****
477: IF(KOM.GT.1.AND.KOM.LT.KO2) THEN
478: LOM=KOM-2
479: ELSE
480: IF(KOM.EQ.1) LOM=1
481: IF(KOM.GE.KO2) LOM=LL-3
482: END IF
483: C*****
484: DO 100 K=1, N
485: WW(K)=XPR(KS+K-1)
486: DO 200 J=1, N
487: UU(J)=OMR(LOM+J)
488: VV(J)=GMS(KS+K-1, LOM+J)
489: 200 CONTINUE
490: CALL LAG(N, UU, VV, OM, AA)
491: GG(K)=AA
492: 100 CONTINUE
493: CALL LAG(N, WW, GG, XPU, GG3)
494: FM3L=GG3
495: C*****
496: ELSE
497: IF(OM.GE.OMS) THEN
498: DGG=(GTT-GMS(KPU, 1))/(1.0-X32)
499: FM3L=GMS(KPU, 1)+DGG*(XXX-X32)
500: ELSE
501: XXR=OMR(LL)/(1.0+OMR(LL))
502: DGG=(GTT-GMS(KPU, LL))/XXR
503: FM3L=GMS(KPU, LL)+DGG*(XXR-XXX)
504: END IF
505: END IF
506: RETURN
507: END
508: C*****
509: C*****
510: SUBROUTINE PDS3
511: CCCCC
512: C
513: IMPLICIT REAL*8 (A-H, O-Z)
514: EXTERNAL FAAC, FS, FL
515: DIMENSION X(2), Y(2), TS(2)
516: COMMON /GS/GAGS, GAUS, GCUS
517: COMMON /GL/GAGL, GAUL, GCUL
518: COMMON/TZ/TK, XZ, YZ
519: COMMON/QAXY/QAXY
520: C OPEN(6, FILE=' ', STATUS='NEW')
521: READ(5, *) LLW, LO2, TC1, TC2, DT
522: READ(5, *) NC, ZA, CV
523: READ(5, *) MC, ZB, CW
524: WRITE(6, *) LLW, LO2, TC1, TC2, DT
525: WRITE(6, *) NC, ZA, CV
526: WRITE(6, *) MC, ZB, CW
527: R32=SQRT(3.0)/2.0
528: XMAX=1.0
529: XMIN=0.0
530: YMAX=1.0
531: XMIN=0.0
532: SMAX=1.0
533: C
534: ITTN = INT((TC2-TC1+0.001)/DT)+1

```

```

535: DO 1000 IT=1, ITTN
536: TTC=TC1+(IT-1)*DT
537: TK=TTC+273.15
538: CCCC G OF COMPONENT
539: CALL SGAGS(TK, GAGS)
540: CALL SGAGL(TK, GAGL)
541: CALL SGAUS(TK, GAUS)
542: CALL SGAUL(TK, GAUL)
543: CALL SGCUS(TK, GCUS)
544: CALL SGCUL(TK, GCUL)
545: C WRITE(6, *) TK, GAGS, GAGL, GAUS, GAUL, GCUS, GCUL
546: CCCCCC
547: DO 3000 NZ=1, MC
548: YZ=ZB+CW*(NZ-1)
549: IF(MC.EQ.1) YZ=ZB
550: DO 2000 KZ=1, NC
551: XZ=ZA+CV*(KZ-1)
552: C YZ=(1.0-XZ)*0.5
553: IF(NC.EQ.1) XZ=ZA
554: C XZ=1.0-YZ
555: ZZ=1.0-YZ-XZ
556: IF(ZZ.LT.0.0) GO TO 2100
557: CALL SSCN3(FAAC, XMAX, XMIN, YMAX, YMIN, SMAX, X, Y, XZ, YZ, AP,
558: 1 FMIN, ITR, JPH, LLW)
559: IF(JPH.LE.0.AND.LO2.EQ.2) GO TO 2000
560: C IF(JPH.EQ.2900) GO TO 2000
561: CCCCCC CONVERSION TO TRIANGULAR COORDINATE
562: X31=(1.0-X(1)+Y(1))/2.0
563: Y31=R32*(1.0-X(1)-Y(1))
564: X32=(1.0-X(2)+Y(2))/2.0
565: Y32=R32*(1.0-X(2)-Y(2))
566: CPS=ETIME(TS)
567: CALL CHK2(X(1), Y(1), X(2), Y(2), DCX, DCY, DCZ)
568: CCCCCC
569: WRITE(6,120) TTC, XZ, YZ, ITR, JPH, X(1), X(2), Y(1), Y(2), AP, FMIN
570: 1, X31, Y31, X32, Y32, DCX, DCY, DCZ, CPS
571: 2000 CONTINUE
572: 2100 CONTINUE
573: 3000 CONTINUE
574: 1000 CONTINUE
575: 120 FORMAT(F12.1, 2F13.8, 2I5, 5F13.6, E30.18, 4F13.6, 3E11.2, F10.1)
576: RETURN
577: END
578: C*****
579: SUBROUTINE PDS32
580: CCCCCC
581: C
582: IMPLICIT REAL*8 (A-H, O-Z)
583: EXTERNAL FAAC, FS, FL
584: DIMENSION X(2), Y(2), TS(2)
585: COMMON /GS/GAGS, GAUS, GCUS
586: COMMON /GL/GAGL, GAUL, GCUL
587: COMMON /TZ/TK, XZ, YZ
588: COMMON/GAXY/QAXY
589: C OPEN(6, FILE=' ', STATUS='NEW')
590: READ(5, *) LLW, LO2, TC1, TC2, DT
591: READ(5, *) NC, ZA, CV
592: READ(5, *) MC, ZB, CW
593: WRITE(6, *) LLW, LO2, TC1, TC2, DT
594: WRITE(6, *) NC, ZA, CV
595: WRITE(6, *) MC, ZB, CW
596: R32=SQRT(3.0)/2.0
597: XMAX=1.0
598: XMIN=0.0
599: YMAX=1.0
600: YMIN=0.0
601: C SMAX=1.0
602: C
603: ITTN = INT((TC2-TC1+0.001)/DT)+1
604: DO 1000 IT=1, ITTN
605: TTC=TC1+(IT-1)*DT
606: TK=TTC+273.15
607: CCCC G OF COMPONENT
608: CALL SGAGS(TK, GAGS)
609: CALL SGAGL(TK, GAGL)
610: CALL SGAUS(TK, GAUS)
611: CALL SGAUL(TK, GAUL)
612: CALL SGCUS(TK, GCUS)
613: CALL SGCUL(TK, GCUL)
614: C WRITE(6, *) TK, GAGS, GAGL, GAUS, GAUL, GCUS, GCUL
615: CCCCCC
616: DO 3000 NZ=1, MC
617: BB=ZB+CW*(NZ-1)
618: DO 2000 KZ=1, NC
619: AA=ZA+CV*(KZ-1)
620: AA1=1.0+AA
621: XZ=AA/AA1
622: YZ=BB/AA1
623: ZZ=1.0-YZ-XZ

```



```

624:      IF (ZZ. LT. 0. 0) ZZ=0. 0
625:      CALL SSON3 (FAAC, XMAX, XMIN, YMAX, YMIN, SMAX, X, Y, XZ, YZ, AP,
626:      1 FMIN, ITR, JPH, LLW)
627:      IF (JPH. LE. 0. AND. L02. EQ. 2) GO TO 2000
628: C      IF (JPH. EQ. 2900) GO TO 2000
629: CCCCC  CONVERSION TO TRIANGULAR COORDINATE
630:      X31=(1. 0-X(1)+Y(1))/2. 0
631:      Y31=R32*(1. 0-X(1)-Y(1))
632:      X32=(1. 0-X(2)+Y(2))/2. 0
633:      Y32=R32*(1. 0-X(2)-Y(2))
634:      CPS=ETIME (TS)
635: C      CALL CHK2 (X(1), Y(1), X(2), Y(2), DCX, DCY, DCZ)
636: CCCCC
637:      WRITE (6, 120) TTC, BB, AA, ITR, JPH, X(1), X(2), Y(1), Y(2), AP, FMIN
638:      1, X31, Y31, X32, Y32, DCX, DCY, DCZ, CPS
639: 2000 CONTINUE
640: 2100 CONTINUE
641: 3000 CONTINUE
642: 1000 CONTINUE
643: 120  FORMAT (F12. 1, 2F13. 8, 2I5, 5F13. 6, E30. 18, 4F13. 6, 3E11. 2, F10. 1)
644:      RETURN
645:      END
646: C*****
647:      SUBROUTINE CHK2 (XS, YS, XL, YL, DCX, DCY, DCZ)
648:      IMPLICIT REAL*8 (A-H, O-Z)
649:      COMMON/GS/G1S, G2S, G3S
650:      COMMON/TZ/TK, XZ, YZ
651:      COMMON/WXYZS/WXYS, WYZS, WZXS
652:      COMMON/GL/G1L, G2L, G3L
653:      COMMON/WXYZL/WXYL, WYZL, WZXL
654:      EX=1. 0D-15
655:      RG=8. 313606
656:      RT=RG*TK
657:      ZS=1. 0-XS-YS
658:      ZL=1. 0-XL-YL
659:      IF (ZS. LT. 0. 0) ZS=0. 0
660:      IF (ZL. LT. 0. 0) ZL=0. 0
661:      WWS=XS*YS*WXYS+YS*ZS*WYZS+ZS*XS*WZXS
662:      WWL=XL*YL*WXYL+YL*ZL*WYZL+ZL*XL*WZXL
663:      CXS=G1S+RT*DLOG (XS+EX) -WWS+YS*WXYS+ZS*WZXS
664:      CYS=G2S+RT*DLOG (YS+EX) -WWS+ZS*WYZS+XS*WXYS
665:      CZS=G3S+RT*DLOG (ZS+EX) -WWS+XS*WZXS+YS*WYZS
666:      CXL=G1L+RT*DLOG (XL+EX) -WWL+YL*WXYL+ZL*WZXL
667:      CYL=G2L+RT*DLOG (YL+EX) -WWL+ZL*WYZL+XL*WXYL
668:      CZL=G3L+RT*DLOG (ZL+EX) -WWL+XL*WZXL+YL*WYZL
669:      DCX=2*(CXS-CXL)/(CXS+CXL)
670:      DCY=2*(CYS-CYL)/(CYS+CYL)
671:      DCZ=2*(CZS-CZL)/(CZS+CZL)
672:      RETURN
673:      END
674: C*****
675: C*****
676: C*****
677: C*****
678:      FUNCTION FS (X, Y)
679:      IMPLICIT REAL*8 (A-H, O-Z)
680:      COMMON/GS/G1, G2, G3
681:      COMMON/TZ/TK, XZ, YZ
682:      COMMON/WXYZS/WXY, WYZ, WZX
683:      RR=8. 313608
684:      RT=RR*TK
685:      QX=-1. 2*X*X+1. 2*X+0. 7
686:      SX=-5. 1194*X**4+12. 353*X**3-10. 377*X**2+3. 1393*X+0. 7
687:      WXY=-140000. 0*SX
688:      WZX=-150000. 0*QX
689:      WYZ=0. 0
690:      EX=1. 0D-15
691:      IF (X. LT. 0. 0) X=0. 0
692:      IF (Y. LT. 0. 0) Y=0. 0
693:      Z=1. 0-X-Y
694:      IF (Z. LT. 0. 0) Z=0. 0
695:      FS=X*(G1+RT*DLOG (X+EX))
696:      1 +Y*(G2+RT*DLOG (Y+EX))
697:      2 +Z*(G3+RT*DLOG (Z+EX))
698:      3 +X*Y*WXY+X*Z*WZX+Y*Z*WYZ
699: C      FS=X*G1+Y*G2+Z*G3
700:      RETURN
701:      END
702: C*****
703: C*****
704: C*****
705:      FUNCTION FL (X, Y)
706:      IMPLICIT REAL*8 (A-H, O-Z)
707:      COMMON/GL/G1, G2, G3
708:      COMMON/TZ/TK, XZ, YZ
709:      COMMON/WXYZL/WXY, WYZ, WZX
710:      RR=8. 313608
711:      RT=RR*TK
712:      EX=1. D-15

```

```

713:      QX=-1.2*X*X+1.2*X+0.7
714:      SX=-5.1194*X**4+12.353*X**3-10.377*X**2+3.1393*X+0.7
715:      WXY=-100000.0*SX
716:      WZX=-100000.0*QX
717:      WYZ=0.0
718:      IF (X.LT.0.0) X=0.0
719:      IF (Y.LT.0.0) Y=0.0
720:      Z=1.0-X-Y
721:      IF (Z.LT.0.0) Z=0.0
722:      FL=X*(G1+RT*DLOG(X+EX))
723:      1 +Y*(G2+RT*DLOG(Y+EX))
724:      2 +Z*(G3+RT*DLOG(Z+EX))
725:      3 +X*Y*WXY+X*Z*WZX+Y*Z*WYZ
726:      RETURN
727:      END
728: C*****
729: C*****
730: C*****
731: C*****
732: C*****
733: C*****
734:      FUNCTION FAAC(AP,X,Y)
735:      IMPLICIT REAL*8 (A-H,O-Z)
736:      DIMENSION X(2),Y(2)
737:      FAAC=AP*FS(X(1),Y(1))+(1.0-AP)*FL(X(2),Y(2))
738: C      WRITE(6,*) AP,X(1),X(2),Y(1),Y(2),FAAC
739:      RETURN
740:      END
741: CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
742: C*****
743:      SUBROUTINE PDUO
744: C      CALCURATION OF PU-O SYSTEM //01
745: CCCCC
746: C
747:      IMPLICIT REAL*8 (A-H,O-Z)
748:      PARAMETER(N=2)
749:      EXTERNAL FPUOSL,FPUOS,FPUOL
750:      DIMENSION X(N),TS(2)
751:      COMMON /GO2/GO,GPU,G1L,G2L
752:      COMMON /TZ/TK,XZ
753:      COMMON/FL11/FL11,FL21,FL22,FL31
754: C      READ(5,*)FL11,FL21,FL22,FL31
755:      READ(5,*)LLW,LO2,TC1,TC2,DT
756:      READ(5,*)NC,ZA,CV
757: C      WRITE(6,*)FL11,FL21,FL22,FL31
758:      WRITE(6,*)LLW,LO2,TC1,TC2,DT
759:      WRITE(6,*)NC,ZA,CV
760:      XMAX=0.68
761:      XMIN=0.55
762:      XF=2.0/3.0
763: C
764: C      ITTN = INT((TC2-TC1+0.001)/DT)+1
765: C      DO 1000 IT=1,ITTN
766:      IT=0
767: 3000 CONTINUE
768:      IT=IT+1
769:      IF (IT.EQ.1) THEN
770:      TTC=TC1
771:      ELSE
772:      IF (TTC.GE.4000.0) THEN
773:      DDT=0.2*DT
774:      ELSE
775:      DDT=DT
776:      END IF
777:      TTC=TTC+DDT
778:      END IF
779:      IF (TTC.GT.TC2) GO TO 1000
780:      TK=TTC+273.15
781:      CALL WPUOS
782:      CALL WPUOLL
783:      CALL GOXA(TK,GO,GFO)
784:      CALL GPUMET(TK,GPU,GFU)
785:      DO 2000 KZ=1,NC
786:      OM=ZA+CV*(KZ-1)
787:      XZ=OM/(1.0+OM)
788:      IF (XZ.GT.XF) GO TO 2100
789:      IF (NC.EQ.1) XZ=ZA
790:      CALL SSCN(FPUOSL,XMAX,XMIN,X,XZ,AP,FMIN,ITR,JPH,LLW)
791:      IF (JPH.LE.0.AND.LO2.EQ.2) GO TO 2000
792:      CPS=ETIME(TS)
793:      OM2=X(2)/(1.0-X(2))
794:      OM1=X(1)/(1.0-X(1))
795:      WRITE(6,120) TTC,OM,ITR,JPH,OM1,OM2,AP,FMIN,CPS
796: 2000 CONTINUE
797: 2100 CONTINUE
798:      GO TO 3000
799: 1000 CONTINUE
800: 120 FORMAT(F12.1,F13.8,2I5,3F13.6,E30.18,F13.6)
801:      RETURN

```

```

802:      END
803: C*****
804: C*****
805:      FUNCTION FPUOSL (AP, X)
806:      IMPLICIT REAL*8 (A-H, O-Z)
807:      DIMENSION X (2)
808:      COMMON /TZ/ TK, XZ
809:      FPUOSL=AP*FPUOS (X (1)) + (1. 0-AP)*FPUOL (X (2))
810:      RETURN
811:      END
812: C*****
813: C*****
814: C*****
815:      SUBROUTINE PDUO2
016: C CALCURATION OF U-O SYSTEM //01
817: CCCCC
818: C
819:      IMPLICIT REAL*8 (A-H, O-Z)
820:      PARAMETER (N=2)
821:      EXTERNAL FUOSL1, FUOSL2, FUOL12, FUOS, FUOL1, FUOL2
822:      DIMENSION X (N), TS (2)
823:      COMMON /GO1/GO, GU, G1L, G2L
824:      COMMON/TZ/TK, XZ
825:      COMMON/FL11/FL11, FL21, FL22, FL31
826: c   READ (5, *) FL11, FL21, FL22, FL31
827:      READ (5, *) LLW, LQ2, TC1, TC2, DT
828:      READ (5, *) NC, ZA, CV
829: c   WRITE (6, *) FL11, FL21, FL22, FL31
830:      WRITE (6, *) LLW, LQ2, TC1, TC2, DT
831:      WRITE (6, *) NC, ZA, CV
832:      XMAX=0. 675
833:      XMIN=0. 0
834:      XF=2. 0/3. 0
835: C
836: C      ITTN = INT ((TC2-TC1+0. 001)/DT)+1
837: C      DO 1000 IT=1, ITTN
838:      IT=0
839: 3000 CONTINUE
840:      IT=IT+1
841:      IF (IT. EQ. 1) THEN
842:      TTC=TC1
843:      ELSE
844:      IF (TTC. GE. 4000. 0) THEN
845:      DDT=0. 2*DT
846:      ELSE
847:      DDT=DT
848:      END IF
849:      TTC=TTC+DDT
850:      END IF
851:      IF (TTC. GT. TC2) GO TO 1000
852:      TK=TTC+273. 15
853:      CALL WUOS
854:      CALL WUOL
855:      CALL GOXA (TK, GO, GFO)
856:      CALL GUMET (TK, GU, GFU)
857:      DO 2000 KZ=1, NC
858:      XZ=ZA+CV*(KZ-1)
859:      IF (XZ. GT. XF) GO TO 2100
860:      IF (NC. EQ. 1) XZ=ZA
861:      CALL SSCN (FUOL12, XMAX, XMIN, X, XZ, AP, FMIN, ITR, JPH, LLW)
862:      IF (JPH. LE. 0. AND. LQ2. EQ. 2) GO TO 2000
863:      CPS=ETIME (TS)
864:      OML=X (2) / (1. 0-X (2))
865:      OMS=X (1) / (1. 0-X (1))
866:      WRITE (6, 120) TTC, XZ, ITR, JPH, OMS, OML, AP, FMIN, CPS
867: 2000 CONTINUE
868: 2100 CONTINUE
869:      GO TO 3000
870: 1000 CONTINUE
871: 120 FORMAT (F12. 1, F13. 8, 2I5, 3F13. 6, E30. 18, F13. 2)
872:      RETURN
873:      END
874: C*****
875: C*****
876:      SUBROUTINE PDUO1
877: C CALCURATION OF U-O SYSTEM //01
878: CCCCC
879: C
880:      IMPLICIT REAL*8 (A-H, O-Z)
881:      PARAMETER (N=2)
882:      EXTERNAL FUOSL1, FUOSL2, FUOL12, FUOS, FUOL1, FUOL2
883:      DIMENSION X (N), TS (2)
884:      COMMON /GO1/GO, GU, G1L, G2L
885:      COMMON/TZ/TK, XZ
886:      COMMON/FL11/FL11, FL21, FL22, FL31
887: c   READ (5, *) FL11, FL21, FL22, FL31
888:      READ (5, *) LLW, LQ2, TC1, TC2, DT
889:      READ (5, *) NC, ZA, CV
890: c   WRITE (6, *) FL11, FL21, FL22, FL31

```

```

891: WRITE(6,*)LLW,LO2,TC1,TC2,DT
892: WRITE(6,*)NC,ZA,CV
893: XMAX=0.675
894: XMIN=0.53
895: XF=2.0/3.0
896:C
897:C ITTN = INT((TC2-TC1+0.001)/DT)+1
898:C DO 1000 IT=1,ITTN
899: IT=0
900: 3000 CONTINUE
901: IT=IT+1
902: IF (IT.EQ.1) THEN
903: TTC=TC1
904: ELSE
905: IF (TTC.GE.4000.0) THEN
906: DDT=0.2*DT
907: ELSE
908: DDT=DT
909: END IF
910: TTC=TTC+DDT
911: END IF
912: IF (TTC.GT.TC2) GO TO 1000
913: TK=TTC+273.15
914: CALL WUOS
915: CALL WUOL
916: CALL GOXA(TK,GO,GFO)
917: CALL GUMET(TK,GU,GFU)
918: DO 2000 KZ=1,NC
919: XZ=ZA+CV*(KZ-1)
920: IF (XZ.GT.XF) GO TO 2100
921: IF (NC.EQ.1) XZ=ZA
922: CALL SSCN(FUOSL2,XMAX,XMIN,X,XZ,AP,FMIN,ITR,JPH,LLW)
923: IF (JPH.LE.0.AND.LO2.EQ.2) GO TO 2000
924: CPS=ETIME(TS)
925: OML=X(2)/(1.0-X(2))
926: OMS=X(1)/(1.0-X(1))
927: CALL CHK3(OMS,OML,DCU,DCO)
928: WRITE(6,120)TTC,XZ,ITR,JPH,OMS,OML,AP,FMIN,DCU,DCO,CPS
929: 2000 CONTINUE
930: 2100 CONTINUE
931: GO TO 3000
932: 1000 CONTINUE
933: 120 FORMAT(F12.1,F13.8,2I5,3F13.6,E30.18,2E13.3,F13.2)
934: RETURN
935: END
936:C*****
937:C*****
938: SUBROUTINE CHK3(OS,OL,DCU,DCO)
939: IMPLICIT REAL*8 (A-H,O-Z)
940: COMMON/TZ/TK,XZ
941: COMMON /GO1/GO,GU,G1L,G2L
942: COMMON/CUOL/CPOL(1000),CPUL(1000),AUL(1000),PPL(1000),GGL(1000)
943: COMMON/WUOLL/OMSL,OM2L,DOML,OMRL(1000),WWL(1000)
944: COMMON/NOL/NOL
945: COMMON/CUOS/CPOS(1000),CPUS(1000),AUS(1000),PPS(1000),GGS(1000)
946: COMMON/WUO/OMS,OM2,DOM,OMR(1000),WW(1000)
947: COMMON/NOM/NOM
948: DIMENSION UU(4),VV(4),SS(4)
949: N=4
950: RR=8.313608
951: RT=RR*TK
952: CCCCC FOR SOLID PHASE
953: KOM=INT((OS-OMS)/DOM)+1
954: LOM=KOM-1
955: DO 200 J=1,N
956: UU(J)=OMR(LOM+J)
957: VV(J)=CPOS(LOM+J)
958: SS(J)=CPUS(LOM+J)
959: 200 CONTINUE
960: CALL LAG(N,UU,VV,OS,COS)
961: CALL LAG(N,UU,SS,OS,CUS)
962: CCCCC FOR LIQUID PHASE
963: KOM=INT((OL-OMSL)/DOML)+1
964: LOM=KOM-1
965: DO 300 J=1,N
966: UU(J)=OMRL(LOM+J)
967: VV(J)=CPOL(LOM+J)
968: SS(J)=CPUL(LOM+J)
969: 300 CONTINUE
970: CALL LAG(N,UU,VV,OL,COL)
971: CALL LAG(N,UU,SS,OL,CUL)
972: CCCCCCCCCCCCCC
973: DCU=2*(CUS-CUL)/(CUS+CUL)
974: DCO=2*(COS-COL)/(COS+COL)
975: CCC CHECK ROUTINE
976: C XPU=0.0
977: C LS=0
978: C CALL MOXOPZ(XPU,OS,TK,PO2,POS,XP,LS)
979: C LS=1

```

```

980:c      CALL MOXOPZ (XPU, OL, TK, P02, POL, OXP, LS)
981:c      ACS=GO+RT*DLOG (POS)
982:c      ACL=GO+RT*DLOG (POL)
983:c      DD=(ACS-ACL)/ACS
984:c      WRITE (6, 100) OS, OL, ACS, ACL, DD
985: 100 FORMAT (15E20, 10)
986:      RETURN
987:      END
988:C*****
989:C*****
990:      FUNCTION FUOSL1 (AP, X)
991:      IMPLICIT REAL*8 (A-H, O-Z)
992:      DIMENSION X (2)
993:      COMMON /TZ/ TK, XZ
994:      FUOSL1=AP*FUOS (X (1)) + (1. 0-AP)*FUOL1 (X (2))
995:C      WRITE (6, *) AP, X (1), X (2), FUOSL
996:      RETURN
997:      END
998:C*****
999:C*****
1000:     FUNCTION FUOSL2 (AP, X)
1001:     IMPLICIT REAL*8 (A-H, O-Z)
1002:     DIMENSION X (2)
1003:     COMMON /TZ/ TK, XZ
1004:     FUOSL2=AP*FUOS (X (1)) + (1. 0-AP)*FUOL2 (X (2))
1005:     RETURN
1006:     END
1007:C*****
1008:C*****
1009:     FUNCTION FUOL12 (AP, X)
1010:     IMPLICIT REAL*8 (A-H, O-Z)
1011:     DIMENSION X (2)
1012:     COMMON /TZ/ TK, XZ
1013:     FUOL12=AP*FUOL1 (X (1)) + (1. 0-AP)*FUOL2 (X (2))
1014:     RETURN
1015:     END
1016:C*****
1017:     FUNCTION FUOS (X)
1018:     IMPLICIT REAL*8 (A-H, O-Z)
1019:     COMMON /G01/G1S, G2S, G1L, G2L
1020:     COMMON /TZ/TK, XZ
1021:     COMMON/WREF/W, WL1, WL2
1022:     COMMON/WUO/OMS, OM2, DOM, OMR (1000), WW (1000)
1023:     COMMON/NOM/NOM
1024:     DIMENSION UU (4), VV (4)
1025:     N=4
1026:     NOT=10
1027:     RR=8. 313608
1028:     RT=RR*TK
1029:     OM=X/(1. 0-X)
1030:     IF (OM. LT. OM2. AND. OM. GE. OMS) THEN
1031:     KOW=INT ((OM-OMS)/DOM)+1
1032: CCCCC FOR SOLID PHASE
1033:     LOM=KOW-1
1034:     DO 200 J=1, N
1035:     UU (J)=OMR (LOM+J)
1036:     VV (J)=WW (LOM+J)
1037: 200 CONTINUE
1038:     CALL LAG (N, UU, VV, OM, W)
1039:     ELSE
1040:     IF (OM. GE. OM2) THEN
1041:     W=WW (NOM)
1042:     ELSE
1043:     W1=WW (1)
1044:     W2=WW (NOT)
1045:     XX1=OMR (1)/(1. 0+OMR (1))
1046:     XX2=OMR (NOT)/(1. 0+OMR (NOT))
1047:     RW=(W2-W1)/(XX2-XX1)
1048:     W=W1+RW*(X-XX1)
1049:     END IF
1050:     END IF
1051:     EX=1. 0D-50
1052:     G1=X*G1S+(1. 0-X)*G2S
1053:     G2=RT*(X*DLOG (X+EX)+(1. 0-X)*DLOG (1. 0-X+EX))
1054:     G3=X*(1. 0-X)*W
1055:     FUOS=G1+G2+G3
1056:     RETURN
1057:     END
1058:C*****
1059:     SUBROUTINE WPUOS
1060:     IMPLICIT REAL*8 (A-H, O-Z)
1061:     COMMON/DUM/D1, D2, D3, D4
1062:     COMMON/TZ/TK, XZ
1063:     COMMON/CPUOS/CPO (1000), CPU (1000), AU (1000), PP (1000), GGG (1000)
1064:     COMMON/WPUO/OMPS, OMP2S, DOMPS, OMRPS (1000), WPOS (1000)
1065:     COMMON/NOP/NOP
1066:     DIMENSION XX (1000), FF (1000), SS (1000)
1067:     LS=0
1068:     XXP=1. 0

```

```

1069:      RG=8. 313608
1070:      OMPS=1. 1
1071:      OMP2S=2. 099
1072:      DOMPS=0. 001
1073:      NOP=INT ((OMP2S-OMPS+0. 0000001)/DOMPS)+1
1074:      RT=TK*RG
1075:      CALL GPU02S(TK, GPU02, GFPU02)
1076:      CALL GOXA(TK, GO, GFO)
1077:      CALL GPUMET(TK, GPU, GFPU)
1078:      XO=2. 0/3. 0
1079:      XPU=1. 0-XO
1080:      B1=XPU*GPU+XO*GO
1081:      B2=RT*(XPU*DLOG(XPU)+XO*DLOG(XO))
1082:      B3=XPU*XO
1083:      GS=GPU02/3. 0
1084:      WW2=(GS-B1-B2)/B3
1085: ccccccc
1086:      DO 1000 K=1, NOP
1087:      OM=OMPS+DOMPS*DFLOAT(K-1)
1088:      OMRPS(K)=OM
1089:      IF(OM.LT.2. 000001.AND.OM.GT.1. 999999) KS=K
1090:      XX(K)=OM/(1+OM)
1091:      CALL MOXXPZ(XXPU, OM, TK, PO2, PO, OXP, LS)
1092:      FF(K)=DLOG(PO/XX(K))/(1. 0-XX(K))**2
1093: 1000 CONTINUE
1094: ccccccccccccccc DETERMINATION OF INTEGRAL CONSTANT
1095:      SS(1)=0. 0
1096:      DO 1050 K=1, NOP-1
1097:      SS(K+1)=SS(K)+(FF(K+1)+FF(K))*(XX(K+1)-XX(K))/2. 0
1098: 1050 CONTINUE
1099:      CI=WW2*XX(KS)/RT-SS(KS)
1100: ccccccccccccccc
1101:      DO 1100 K=1, NOP
1102:      WPOS(K)=(RT/XX(K))*(CI+SS(K))
1103: 1100 CONTINUE
1104: ccccccccccccccc
1105:      DO 1150 K=1, NOP-1
1106:      XO=XX(K)
1107:      XU=1. 0-XX(K)
1108:      IF(K.NE.1) THEN
1109:      DW=(WPOS(K+1)-WPOS(K-1))/(XX(K+1)-XX(K-1))
1110:      ELSE
1111:      DW=(WPOS(K+1)-WPOS(K))/(XX(K+1)-XX(K))
1112:      END IF
1113:      DWXO=XU**2*XO*DW
1114:      DWXU=-XU**2*XU*DW
1115:      CPO(K)=GO+RT*DLOG(XO)+WPOS(K)*XU**2+DWXO
1116:      CPU(K)=GPU+RT*DLOG(XU)+WPOS(K)*XO**2+DWXU
1117:      GGG(K)=XO*GO+XU*GPU+RT*(XO*DLOG(XO)+XU*DLOG(XU))+XO*XU*WPOS(K)
1118:      AAA=XU*DEXP(WPOS(K)*XO**2+DWXU)/RT
1119:      AU(K)=DLOG(AAA)
1120:      BBB=XO*DEXP(WPOS(K)*XU**2+DWXO)/RT
1121:      PP(K)=DLOG(BBB)
1122: 1150 CONTINUE
1123: ccccccccccccccc
1124: 100 FORMAT(15E20. 10)
1125:      RETURN
1126:      END
1127: C*****
1128:      SUBROUTINE WPUOLL
1129:      IMPLICIT REAL*8 (A-H, O-Z)
1130:      COMMON/DUM/D1, D2, D3, D4
1131:      COMMON/TZ/TK, XZ
1132:      COMMON/CPUOL/CPO(1000), CPU(1000), AU(1000), PP(1000), GGG(1000)
1133:      COMMON/WPUOL/OMPL, OMP2L, DOMPL, OMRPL(1000), WPOL(1000)
1134:      COMMON/NOP/NOP
1135:      DIMENSION XX(1000), FF(1000), SS(1000)
1136:      LS=1
1137:      XXPU=1. 0
1138:      RG=8. 313608
1139:      OMPL=1. 1
1140:      OMP2L=2. 099
1141:      DOMPL=0. 001
1142:      NOP=INT ((OMP2L-OMPL+0. 0000001)/DOMPL)+1
1143:      RT=TK*RG
1144:      CALL GPU02L(TK, GPU02, GFPU02)
1145:      CALL GOXA(TK, GO, GFO)
1146:      CALL GPUMET(TK, GPU, GFPU)
1147:      XO=2. 0/3. 0
1148:      XPU=1. 0-XO
1149:      B1=XPU*GPU+XO*GO
1150:      B2=RT*(XPU*DLOG(XPU)+XO*DLOG(XO))
1151:      B3=XPU*XO
1152:      GS=GPU02/3. 0
1153:      WW2=(GS-B1-B2)/B3
1154: ccccccc
1155:      DO 1000 K=1, NOP
1156:      OM=OMPL+DOMPL*DFLOAT(K-1)
1157:      OMRPL(K)=OM

```

```

1158:      IF (OM.LT.2.000001.AND.OM.GT.1.999999) KS=K
1159:      XX(K)=OM/(1+OM)
1160:      CALL MOXOPZ (XXPU, OM, TK, P02, PO, OXP, LS)
1161:      FF(K)=DLOG(PO/XX(K))/(1.0-XX(K))*2
1162: 1000 CONTINUE
1163: CCCCCCCCCCCCCC DETERMINATION OF INTEGRAL CONSTANT
1164:      SS(1)=0.0
1165:      DO 1050 K=1, N0P-1
1166:      SS(K+1)=SS(K)+(FF(K+1)+FF(K))*(XX(K+1)-XX(K))/2.0
1167: 1050 CONTINUE
1168:      CI=WW2*XX(KS)/RT-SS(KS)
1169: CCCCCCCCCCCCCC
1170:      DO 1100 K=1, N0P
1171:      WPOL(K)=(RT/XX(K))*(CI+SS(K))
1172: 1100 CONTINUE
1173: CCCCCCCCCCCCCC
1174:      DO 1150 K=1, N0P-1
1175:      XO=XX(K)
1176:      XU=1.0-XX(K)
1177:      IF (K.NE.1) THEN
1178:      DW=(WPOL(K+1)-WPOL(K-1))/(XX(K+1)-XX(K-1))
1179:      ELSE
1180:      DW=(WPOL(K+1)-WPOL(K))/(XX(K+1)-XX(K))
1181:      END IF
1182:      DWXO=XU**2*XO*DW
1183:      DWXU=-XO**2*XU*DW
1184:      CPO(K)=GO+RT*DLOG(XO)+WPOL(K)*XU**2+DWXO
1185:      CPU(K)=GPU+RT*DLOG(XU)+WPOL(K)*XO**2+DWXU
1186:      GGG(K)=XO*GO+XU*GPU+RT*(XO*DLOG(XO)+XU*DLOG(XU))+XO*XU*WPOL(K)
1187:      AAA=XU*DEXP((WPOL(K)*XO**2+DWXU)/RT)
1188:      AU(K)=DLOG(AAA)
1189:      BBB=XO*DEXP((WPOL(K)*XU**2+DWXO)/RT)
1190:      PP(K)=DLOG(BBB)
1191: 1150 CONTINUE
1192: CCCCCCCCCCCCCC
1193: 100 FORMAI (15E20.10)
1194: RETURN
1195: END
1196: C*****
1197: C*****
1198: FUNCTION FPUOS(X)
1199: IMPLICIT REAL*8 (A-H, O-Z)
1200: COMMON /G02/G1S, G2S, G1L, G2L
1201: COMMON /TZ/TK, XZ
1202: COMMON /WREF/W, WL1, WL2
1203: COMMON /WPUO/OMS, OM2, DOM, OMR(1000), WW(1000)
1204: COMMON /NOP/NOM
1205: DIMENSION UU(4), VV(4)
1206: N=4
1207: NOT=10
1208: RR=8.313608
1209: RT=RR*TK
1210: OM=X/(1.0-X)
1211: IF (OM.LT.OM2.AND.OM.GE.OMS) THEN
1212: KOM=INT((OM-OMS)/DOM)+1
1213: LOM=KOM-1
1214: DO 200 J=1, N
1215: UU(J)=OMR(LOM+J)
1216: VV(J)=WW(LOM+J)
1217: 200 CONTINUE
1218: CALL LAG(N, UU, VV, OM, W)
1219: ELSE
1220: IF (OM.GE.OM2) THEN
1221: W=WW(NOM)
1222: ELSE
1223: W1=WW(1)
1224: W2=WW(NOT)
1225: XX1=OMR(1)/(1.0+OMR(1))
1226: XX2=OMR(NOT)/(1.0+OMR(NOT))
1227: RW=(W2-W1)/(XX2-XX1)
1228: W=W1+RW*(X-XX1)
1229: END IF
1230: END IF
1231: EX=1.0D-50
1232: G1=X*G1S+(1.0-X)*G2S
1233: G2=RT*(X*DLOG(X+EX)+(1.0-X)*DLOG(1.0-X+EX))
1234: G3=X*(1.0-X)*W
1235: FPUOS=G1+G2+G3
1236: RETURN
1237: END
1238: C*****
1239: FUNCTION FPUOL(X)
1240: IMPLICIT REAL*8 (A-H, O-Z)
1241: COMMON /G02/G1S, G2S, G1L, G2L
1242: COMMON /TZ/TK, XZ
1243: COMMON /WREF/W, WL1, WL2
1244: COMMON /WPUOL/OMS, OM2, DOM, OMR(1000), WW(1000)
1245: COMMON /NOP/NOM
1246: DIMENSION UU(4), VV(4)

```

```

1247:      N=4
1248:      NOT=10
1249:      RR=8. 313608
1250:      RT=RR*TK
1251:      OM=X/(1.0-X)
1252:      IF (OM. LT. OM2. AND. OM. GE. OMS) THEN
1253:      KOM=INT((OM-OMS)/DOM)+1
1254:      LOM=KOM-1
1255:      DO 200 J=1, N
1256:      UU(J)=OMR(LOM+J)
1257:      VV(J)=WW(LOM+J)
1258: 200 CONTINUE
1259:      CALL LAG(N, UU, VV, OM, W)
1260:      ELSE
1261:      IF (OM. GE. OM2) THEN
1262:      W=WW(NOM)
1263:      ELSE
1264:      W1=WW(1)
1265:      W2=WW(NOT)
1266:      XX1=OMR(1)/(1.0+OMR(1))
1267:      XX2=OMR(NOT)/(1.0+OMR(NOT))
1268:      RW=(W2-W1)/(XX2-XX1)
1269:      W=W1+RW*(X-XX1)
1270:      END IF
1271:      END IF
1272:      EX=1.0D-50
1273:      G1=X*G1S+(1.0-X)*G2S
1274:      G2=RT*(X*DLOG(X+EX)+(1.0-X)*DLOG(1.0-X+EX))
1275:      G3=X*(1.0-X)*W
1276:      FPUOL=G1+G2+G3
1277:      RETURN
1278:      END
1279: C*****
1280:      SUBROUTINE WUOS
1281:      IMPLICIT REAL*8 (A-H, O-Z)
1282:      COMMON/DUM/D1, D2, D3, D4
1283:      COMMON/TZ/TK, XZ
1284:      COMMON/CUOS/CPO(1000), CPU(1000), AU(1000), PP(1000), GGG(1000)
1285:      COMMON/WUO/OMS, OM2, DOM, OMR(1000), WW(1000)
1286:      COMMON/NOM/NOM
1287:      DIMENSION XX(1000), FF(1000), SS(1000), DWDX(1000)
1288:      LS=0
1289:      XPU=0.0
1290:      RG=8. 313608
1291:      OMS=1.1
1292:      OM2=2.099
1293:      DOM=0.001
1294:      NOM=INT((OM2-OMS+0.000001)/DOM)+1
1295:      RT=TK*RG
1296:      CALL GUO2S(TK, GUO2, GFUO2)
1297:      CALL GOXA(TK, GO, GFO)
1298:      CALL GUMET(TK, GU, GFU)
1299:      X0=2.0/3.0
1300:      XU=1.0-X0
1301:      B1=XU*GU+X0*GO
1302:      B2=RT*(XU*DLOG(XU)+X0*DLOG(X0))
1303:      B3=XU*X0
1304:      GS=GUO2/3.0
1305:      WW2=(GS-B1-B2)/B3
1306: cccccccc
1307:      DO 1000 K=1, NOM
1308:      OM=OMS+DOM*DFLOAT(K-1)
1309:      OMR(K)=OM
1310:      IF (OM. LT. 2.000001. AND. OM. GT. 1.999999) KS=K
1311:      XX(K)=OM/(1+OM)
1312:      CALL MOXOPZ(XPU, OM, TK, PO2, PO, OXP, LS)
1313:      PP(K)=PO
1314:      FF(K)=DLOG(PO/XX(K))/(1.0-XX(K))*2
1315: 1000 CONTINUE
1316: cccccccccccccccc DETERMINATION OF INTEGRAL CONSTANT
1317:      SS(1)=0.0
1318:      DO 1050 K=1, NOM-1
1319:      SS(K+1)=SS(K)+(FF(K+1)+FF(K))*(XX(K+1)-XX(K))/2.0
1320: 1050 CONTINUE
1321:      C1=WW2*XX(KS)/RT-SS(KS)
1322: cccccccccccccccc
1323:      DO 1100 K=1, NOM
1324:      WW(K)=(RT/XX(K))*(C1+SS(K))
1325: C      WRITE(6, 100) XX(K), WW(K), WW2
1326: 1100 CONTINUE
1327: cccccccccccccccc
1328: cccccccccccccccc
1329:      DO 1150 K=1, NOM
1330:      X0=XX(K)
1331:      XU=1.0-X0
1332:      CPO(K)=GO+RT*DLOG(PP(K))
1333:      CPU(K)=GU+RT*(DLOG(XU)+(X0/XU)*DLOG(X0/PP(K)))+X0*WW(K)
1334:      GGG(K)=X0*GO+XU*GU+RT*(X0*DLOG(X0)+XU*DLOG(XU))+X0*XU*WW(K)
1335:      AU(K)=DEXP(CPU(K)-GU)/RT

```



```

1336: 1150 CONTINUE
1337: CCCCCCCCCCCCCC
1338: CCCCCCCCCCCCCC
1339: 100 FORMAT(15E20,10)
1340: RETURN
1341: END
1342: C*****
1343: C*****
1344: C*****
1345: FUNCTION FUOL1(X)
1346: IMPLICIT REAL*8 (A-H,O-Z)
1347: COMMON /G01/G1S, G2S, G1L, G2L
1348: COMMON /TZ,TK, XZ
1349: COMMON /WREF/WS, W, WDUM
1350: RR=8. 313608
1351: RT=RR*TK
1352: TMAX=4300.0
1353: C***** original model
1354: C CC=-534728.876
1355: C BB=82.84768
1356: C AA=0.0
1357: C***** Ackerman Po pressure base
1358: C CC=-630695.17
1359: C BB=170.8755
1360: C AA=-0.01985676
1361: C***** model based on Po from MOXOPZ
1362: C CC=-682810.0
1363: C BB=216.7959
1364: C AA=-0.0314081
1365: CCCCCCCCCCCCCC
1366: C CC=-718514.0
1367: C BB=260.25
1368: C AA=-0.0425
1369: CCCCCCCCCCCCCC
1370: C XXS=2.0/3.0
1371: C IF (TK. LE. TMAX) THEN
1372: C W=AA*TK**2+BB*TK+CC
1373: C ELSE
1374: C TT=TMAX
1375: C W=AA*TT**2+BB*TT+CC
1376: C END IF
1377: C EX=1.0D-50
1378: C G1=X*G1S+(1.0-X)*G2S
1379: C G2=RT*(X*DLOG(X+EX)+(1.0-X)*DLOG(1.0-X+EX))
1380: C G3=X*(1.0-X)*W
1381: C FUOL1=G1+G2+G3
1382: C CPO=G1S+RT*DLOG(X+EX)+2*X*(1.0-X)**2*W+X**2*(1.0-X)**2*AA
1383: C CPU=G2S+RT*DLOG(1.0-X)+X**2*(2*X-1.0)*W-X**3*(1.0-X)*AA
1384: C WRITE(6,100)TK, OM, X, FUOL
1385: 100 FORMAT(6F20.8)
1386: RETURN
1387: END
1388: C*****
1389: C*****
1390: FUNCTION FUOL2(X)
1391: IMPLICIT REAL*8 (A-H,O-Z)
1392: COMMON /G01/G1S, G2S, G1L, G2L
1393: COMMON /TZ,TK, XZ
1394: COMMON /WREF/WS, WDUM, W
1395: COMMON /WUOLL/OMSL, OM2L, DOML, OMRL(1000), WWL(1000)
1396: COMMON /NOL/NOL
1397: DIMENSION UU(4), VV(4)
1398: N=4
1399: NOT=500
1400: RR=8. 313608
1401: RT=RR*TK
1402: OM=X/(1.0-X)
1403: IF (OM. LT. OM2L. AND. OM. GE. OMSL) THEN
1404: KOW=INT((OM-OMSL)/DOML)+1
1405: LOM=KOW-1
1406: DO 200 J=1, N
1407: UU(J)=OMRL(LOM+J)
1408: VV(J)=WWL(LOM+J)
1409: 200 CONTINUE
1410: CALL LAG(N, UU, VV, OM, W)
1411: ELSE
1412: IF (OM. GE. OM2L) THEN
1413: W=WWL(NOL)
1414: ELSE
1415: W1=WWL(1)
1416: W2=WWL(NOT)
1417: XX1=OMRL(1)/(1.0+OMRL(1))
1418: XX2=OMRL(NOT)/(1.0+OMRL(NOT))
1419: RW=(W2-W1)/(XX2-XX1)
1420: W=W1+RW*(X-XX1)
1421: END IF
1422: END IF
1423: EX=1.0D-50
1424: G1=X*G1S+(1.0-X)*G2S

```

```

1425:      G2=RT*(X*DLOG(X+EX)+(1,0-X)*DLOG(1,0-X+EX))
1426:      G3=X*(1,0-X)*W
1427:      FUOL2=G1+G2+G3
1428:      RETURN
1429:      END
1430: C*****
1431: C*****
1432: C*****
1433:      SUBROUTINE WUOL
1434:      IMPLICIT REAL*8 (A-H, O-Z)
1435:      COMMON/DUM/D1, D2, D3, D4
1436:      COMMON/TZ/TK, XZ
1437:      COMMON/GUOL/CPO(1000), CPU(1000), AU(1000), PP(1000), GGG(1000)
1438:      COMMON/WUOL/OMSL, OM2L, DOML, OMRL(1000), WWL(1000)
1439:      COMMON/NOL/NOL
1440:      DIMENSION XX(1000), FF(1000), SS(1000), DWDX(1000)
1441:      LS=1
1442:      XPU=0.0
1443:      RG=8.313608
1444:      OMSL=1.1
1445:      OM2L=2.099
1446:      DOML=0.001
1447:      NOL=INT((OM2L-OMSL+0.000001)/DOML)+1
1448:      RT=TK*RG
1449:      CALL GLUO2(TK, GLUO2, GLFUO2)
1450:      CALL GOXA(TK, GO, GFO)
1451:      CALL GUMET(TK, GU, GFU)
1452:      XO=2.0/3.0
1453:      XU=1.0-XO
1454:      B1=XU*GU+XO*GO
1455:      B2=RT*(XU*DLOG(XU)+XO*DLOG(XO))
1456:      B3=XU*XO
1457:      GS=GLUO2/3.0
1458:      WWL2=(GS-B1-B2)/B3
1459: ccccccc
1460:      DO 1000 K=1, NOL
1461:      OM=OMSL+DOML*DLOAT(K-1)
1462:      OMRL(K)=OM
1463:      IF(OM.LT.2.000001.AND.OM.GT.1.999999) KS=K
1464:      XX(K)=OM/(1+OM)
1465:      CALL MOXOPZ(XPU, OM, TK, PO2, PO, OXP, LS)
1466:      PP(K)=PO
1467:      FF(K)=DLOG(PO/XX(K))/(1.0-XX(K))**2
1468: 1000 CONTINUE
1469: ccccccccccccccc DETERMINATION OF INTEGRAL CONSTANT
1470:      SS(1)=0.0
1471:      DO 1050 K=1, NOL-1
1472:      SS(K+1)=SS(K)+(FF(K+1)+FF(K))*(XX(K+1)-XX(K))/2.0
1473: 1050 CONTINUE
1474:      C1=WWL2*XX(KS)/RT-SS(KS)
1475: ccccccccccccccc
1476:      DO 1100 K=1, NOL
1477:      WWL(K)=(RT/XX(K))*(C1+SS(K))
1478: 1100 CONTINUE
1479: ccccccccccccccccccc
1480: ccccccccccccccccccc
1481:      DO 1150 K=1, NOL
1482:      XO=XX(K)
1483:      XU=1.0-XX(K)
1484:      CPO(K)=GO+RT*DLOG(PP(K))
1485:      CPU(K)=GU+RT*(DLOG(XU)+(XO/XU)*DLOG(XO/PP(K)))+XO*WWL(K)
1486:      GGG(K)=XO*GO+XU*GU+RT*(XO*DLOG(XO)+XU*DLOG(XU))+XO*XU*WWL(K)
1487:      AU(K)=DEXP((CPU(K)-GU)/RT)
1488: 1150 CONTINUE
1489: ccccccccccccccccccc
1490: ccccccccccccccccccc
1491:      RETURN
1492:      END
1493: ccccccccccccccccccc
1494: ccccccccccccccccccc
1495: ccccccccccccccccccc
1496:      SUBROUTINE INTEGRA(N, XX, YY, AS, AT, SSS)
1497:      IMPLICIT REAL*8 (A-H, O-Z)
1498:      DIMENSION A(10, 11), B(10), XX(10), YY(10)
1499:      DO 200 K=1, N
1500:      A(1, K)=1.0
1501:      A(2, K)=XX(K)
1502:      A(3, K)=XX(K)**2
1503:      A(4, K)=XX(K)**3
1504:      A(K, N+1)=(AT**K-AS**K)/DLOAT(K)
1505: 200 CONTINUE
1506:      CALL GAUSS(N, A, B)
1507:      SSS=0.0
1508:      DO 300 J=1, N
1509:      SSS=SSS+B(J)*YY(J)
1510: 300 CONTINUE
1511:      RETURN
1512:      END
1513: ccccccccccccccccccc

```

```

1514:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1515:  SUBROUTINE GAUSS (N, A, B)
1516:  IMPLICIT REAL*8 (A-H, O-Z)
1517:  DIMENSION A(10, 11), B(10)
1518:  NP1=N+1
1519:  NM1=N-1
1520:  DO 30 K=1, NM1
1521:  AKK=A(K, K)
1522:  DO 40 J=K, NP1
1523:  A(K, J)=A(K, J)/AKK
1524:  40 CONTINUE
1525:  KP1=K+1
1526:  DO 50 I=KP1, N
1527:  IF (I.EQ.K) GO TO 50
1528:  AIK=A(I, K)
1529:  DO 60 J=K, NP1
1530:  A(I, J)=A(I, J)-AIK*A(K, J)
1531:  60 CONTINUE
1532:  50 CONTINUE
1533:  30 CONTINUE
1534:  ANN=A(N, N)
1535:  A(N, N)=A(N, N)/ANN
1536:  A(N, NP1)=A(N, NP1)/ANN
1537:  B(N)=A(N, NP1)
1538:CCCCCCCCCCC
1539:  DO 70 I=1, NM1
1540:  K=N-I
1541:  B(K)=A(K, NP1)
1542:  DO 80 J=K, N
1543:  B(K)=B(K)-A(K, J+1)*B(J+1)
1544:  80 CONTINUE
1545:  70 CONTINUE
1546:  RETURN
1547:  END
1548:C*****
1549:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1550:  SUBROUTINE GUPUOS
1551:  IMPLICIT REAL*8 (A-H, O-Z)
1552:  COMMON /G03/GO, GU, GPU
1553:  COMMON/TZ/TK, XZ, YZ
1554:  COMMON/LLS/LL, NN
1555:  COMMON/GMS/XPUS, OMS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
1556:  COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PL(1001, 1001)
1557:  DIMENSION SF2(1001), SF3(1001), S2(1001), S3(1001)
1558:  1, RX(1001), PPL(1001)
1559:  DIMENSION XT(10), Y1(10), X2(10), Y2(10)
1560:C***** MESH POINT SETTING
1561:C***** NN = XPU MESH NO. LL= OM MESH NO. L2 = MESH POINT AT OM=2.0
1562:  NN=201
1563:  LL=1001
1564:  L2=(LL-1)/10+1
1565:  NS=4
1566:  NXY=NS-2
1567:  XPUS=0.0
1568:  OMS=2.1
1569:  DX=1.0/DFLOAT(NN-1)
1570:  DM=1.0/DFLOAT(LL-1)
1571:  DDP=2.0D-8
1572:  EPS=1.0D-30
1573:  EPAL=DLOG(EPS)
1574:  LS=0
1575:  RG=8.313608
1576:  RT=TK*RG
1577:  WSL=0.0/RT
1578:C***** SETTING END
1579:  CALL GU02S(TK, GU02, GFU02)
1580:  CALL GOXA(TK, GO, GFO)
1581:  CALL GUMET(TK, GU, GFU)
1582:  CALL GPU02S(TK, GPU02, GFPU02)
1583:  CALL GPUMET(TK, GPU, GFPU)
1584:CCCCC
1585:  U02
1586:  XPU=0.0
1587:  OM=2.0
1588:  CALL MOXOPZ(XPU, OM, TK, P02, PU0, OXP, LS)
1589:  AUL2=(GU02-2.0*GO-GU-2.0*RT*DLOG(PU0))/RT
1590:CCCCC
1591:  PU02
1592:  XPU=1.0
1593:  OM=2.0
1594:  CALL MOXOPZ(XPU, OM, TK, P02, PPU0, OXP, LS)
1595:  APUL2=(GPU02-2.0*GO-GPU-2.0*RT*DLOG(PPU0))/RT
1596:CCCCC
1597:  CALC. OF U & PU ACTIVITY IN U02-PU02
1598:  DO 200 K=1, NN
1599:  OM=2.0
1600:  XPU=XPUS+DX*DFLOAT(K-1)
1601:  XU=1.0-XPU
1602:  XPR(K)=XPU
1603:  CALL MOXOPZ(XPU, OM, TK, P02, PO, OXP, LS)
1604:  POL=DLOG(PO)
1605:  AUL=2.0*DLOG(PU0/PO)+AUL2+DLOG(XU+EPS)+XPU**2*WSL

```

```

1603:      APUL=2. 0*DLOG (PPUO/PO) +APUL2+DLOG (XPU+EPS) +XU**2*WSL
1604:      AUS (K, L2) =AUL
1605:      APUS (K, L2) =APUL
1606:      PL (K, L2) =POL
1607: C      GSM=XU*GUO2+XPU*GPUO2+RT* (XU*DLOG (XU+EPS) +XPU*DLOG (XU+EPS))
1608:      GM3=2. 0*(GO+RT*DLOG (PO)) + (XU*(GU+RT*AUL) +XPU*(GPU+RT*APUL))
1609:      GMS (K, L2) =GM3/3. 0
1610: C*****
1611:      IF (K. NE. 1. AND. K. NE. NN) THEN
1612:          ST=XPU
1613:          STR=1. 0-ST
1614:          DO 300 J=1, LL
1615:              OM=OMS-DM*DFLOAT (J-1)
1616:              CALL MOXOPZ (XPU, OM, TK, PO2, PO, OXP, LS)
1617:              POL=DLOG (PO)
1618:              PL (K, J) =POL
1619:              OMR (J) =OM
1620:              XO=OM/ (1. 0+OM)
1621:              XP=XPU/ (1. 0+OM)
1622:              XU= (1. 0-XPU) / (1. 0+OM)
1623:              S3 (J) =XU/ (XO+XU)
1624:              S2 (J) =XP/ (XO+XP)
1625:              XP1=XP+DDP
1626:              XP2=XP-DDP
1627:              XU31= (1. 0-XP1) *S3 (J)
1628:              XO31= (1. 0-XP1) * (1. 0-S3 (J))
1629:              XU32= (1. 0-XP2) *S3 (J)
1630:              XO32= (1. 0-XP2) * (1. 0-S3 (J))
1631: CCCCC
1632:              XU21=1. 0-XP1/S2 (J)
1633:              XO21= (1. 0-S2 (J)) *XP1/S2 (J)
1634:              XU22=1. 0-XP2/S2 (J)
1635:              XO22= (1. 0-S2 (J)) *XP2/S2 (J)
1636: CCCCC
1637:              DST3= (XP1/ (XP1+XU31)) - (XP2/ (XP2+XU32))
1638:              DST2= (XP1/ (XP1+XU21)) - (XP2/ (XP2+XU22))
1639: CCCCC
1640:              OM31=XO31/ (1. 0-XO31)
1641:              XPU31=XPU/ (1. 0-XO31)
1642:              OM32=XO32/ (1. 0-XO32)
1643:              XPU32=XPU/ (1. 0-XO32)
1644:              CALL MOXOPZ (XPU31, OM31, TK, PO2, PO31, OXP, LS)
1645:              CALL MOXOPZ (XPU32, OM32, TK, PO2, PO32, OXP, LS)
1646:              SF3 (J) =- (STR/S3 (J)) **2*DLOG (PO31/PO32) /DST3
1647:              OM21=XO21/ (1. 0-XO21)
1648:              XPU21=XPU/ (1. 0-XO21)
1649:              OM22=XO22/ (1. 0-XO22)
1650:              XPU22=XPU/ (1. 0-XO22)
1651:              CALL MOXOPZ (XPU21, OM21, TK, PO2, PO21, OXP, LS)
1652:              CALL MOXOPZ (XPU22, OM22, TK, PO2, PO22, OXP, LS)
1653:              SF2 (J) = (ST/S2 (J)) **2*DLOG (PO21/PO22) /DST2
1654: C      WRITE (6, 100) DST2, DST3, SF2 (J), SF3 (J)
1655:      300 CONTINUE
1656: C*****
1657:      SSU=0. 0
1658:      SSP=0. 0
1659:      DO 320 J=1, L2-1
1660:          DO 302 M=1, NS
1661:              IF (J. GT. NXY) THEN
1662:                  X1 (M) =S3 (J-NXY+M)
1663:                  Y1 (M) =SF3 (J-NXY+M)
1664:                  X2 (M) =S2 (J-NXY+M)
1665:                  Y2 (M) =SF2 (J-NXY+M)
1666:              ELSE
1667:                  X1 (M) =S3 (M)
1668:                  X2 (M) =S2 (M)
1669:                  Y1 (M) =SF3 (M)
1670:                  Y2 (M) =SF2 (M)
1671:              END IF
1672:          302 CONTINUE
1673:          AS1=S3 (J)
1674:          AT1=S3 (J+1)
1675:          AS2=S2 (J)
1676:          AT2=S2 (J+1)
1677:          CALL INTEGRA (NS, X1, Y1, AS1, AT1, DSP)
1678:          CALL INTEGRA (NS, X2, Y2, AS2, AT2, DSU)
1679: C      DSP= (SF3 (J+1) +SF3 (J)) * (S3 (J+1) -S3 (J)) /2. 0
1680: C      DSU= (SF2 (J+1) +SF2 (J)) * (S2 (J+1) -S2 (J)) /2. 0
1681:          SSP=SSP+DSP
1682:          SSU=SSU+DSU
1683:      320 CONTINUE
1684:      APUL=APUL-SSP
1685:      AUL=AUL-SSU
1686:      AUS (K, 1) =AUL
1687:      APUS (K, 1) =APUL
1688:      POL=PL (K, 1)
1689:      OM=OMS
1690:      XO=OM/ (1. 0+OM)
1691:      XP=XPU/ (1. 0+OM)

```

```

1692:      XU=(1.0-XPU)/(1.0+OM)
1693:      GMS(K,1)=XO*(GO+RT*POL)+XP*(GPU+RT*APUL)+XU*(GU+RT*AUL)
1694: C*****
1695:      DO 350 J=1,LL-1
1696:      DO 305 M=1,NS
1697:      IF (J.GT.NXY.AND.J.LT.(LL-NXY)) THEN
1698:      X1(M)=S3(J-NXY+M)
1699:      Y1(M)=SF3(J-NXY+M)
1700:      X2(M)=S2(J-NXY+M)
1701:      Y2(M)=SF2(J-NXY+M)
1702:      ELSE
1703:      IF (J.LE.2) THEN
1704:      X1(M)=S3(M)
1705:      Y1(M)=SF3(M)
1706:      X2(M)=S2(M)
1707:      Y2(M)=SF2(M)
1708:      ELSE
1709:      X1(M)=S3(LL-NS+M)
1710:      Y1(M)=SF3(LL-NS+M)
1711:      X2(M)=S2(LL-NS+M)
1712:      Y2(M)=SF2(LL-NS+M)
1713:      END IF
1714:      END IF
1715: 305 CONTINUE
1716:      AS1=S3(J)
1717:      AT1=S3(J+1)
1718:      AS2=S2(J)
1719:      AT2=S2(J+1)
1720:      CALL INTEGRA(NS,X1,Y1,AS1,AT1,DAPL)
1721:      CALL INTEGRA(NS,X2,Y2,AS2,AT2,DAUL)
1722: C      DAPL=(SF3(J+1)+SF3(J))*(S3(J+1)-S3(J))/2.0
1723:      APUL=APUL+DAPL
1724: C      DAUL=(SF2(J+1)+SF2(J))*(S2(J+1)-S2(J))/2.0
1725:      AUL=AUL+DAUL
1726:      APUS(K,J+1)=APUL
1727:      AUS(K,J+1)=AUL
1728:      POL=PL(K,J+1)
1729:      OM=OMS-DM*DFLOAT(J)
1730:      XO=OM/(1.0+OM)
1731:      XPU=XPU/(1.0+OM)
1732:      XU=(1.0-XPU)/(1.0+OM)
1733:      GMS(K,J+1)=XO*(GO+RT*POL)+XP*(GPU+RT*APUL)+XU*(GU+RT*AUL)
1734: C      WRITE(6,100)OM,POL,AUL,APUL,GMS(K,J+1)
1735: 350 CONTINUE
1736: C***** FOR UO2 OR PUO2 CASE
1737:      ELSE
1738:      DO 400 J=1,LL
1739:      OM=OMS-DM*DFLOAT(J-1)
1740:      XO=OM/(1.0+OM)
1741:      XM=1.0-XO
1742:      RX(J)=XO/XM
1743:      IF (K.EQ.1) THEN
1744:      XPU=0.0
1745:      CALL MOXOPZ(XPU,OM,TK,P02,PO,XP,LS)
1746:      PPL(J)=DLOG(PO)
1747:      ELSE
1748:      XPU=1.0
1749:      CALL MOXOPZ(XPU,OM,TK,P02,PO,XP,LS)
1750:      PPL(J)=DLOG(PO)
1751:      END IF
1752: 400 CONTINUE
1753: C*****
1754:      SS=0.0
1755:      DO 420 J=1,L2-1
1756:      PL(K,J+1)=PPL(J+1)
1757:      DO 402 M=1,NS
1758:      IF (J.GT.NXY) THEN
1759:      X1(M)=PPL(J-NXY+M)
1760:      Y1(M)=RX(J-NXY+M)
1761:      ELSE
1762:      X1(M)=PPL(M)
1763:      Y1(M)=RX(M)
1764:      END IF
1765: 402 CONTINUE
1766:      AS1=PPL(J)
1767:      AT1=PPL(J+1)
1768:      CALL INTEGRA(NS,X1,Y1,AS1,AT1,DAUL)
1769: C      DAUL=(RX(J+1)+RX(J))*(PPL(J+1)-PPL(J))/2.0
1770:      SS=SS-DAUL
1771: 420 CONTINUE
1772:      OM=OMS
1773:      XO=OM/(1.0+OM)
1774:      XM=1.0-XO
1775:      PL(K,1)=PPL(1)
1776:      IF (K.EQ.1) THEN
1777:      AUL=AUL2-SS
1778:      AUS(K,1)=AUL
1779:      APUS(K,1)=EPAL
1780:      GMS(K,1)=XO*(GO+RT*PPL(1))+XM*(GU+RT*AUL)

```

```

1781:      ELSE
1782:      APUL=APUL2-SS
1783:      APUS(K,1)=APUL
1784:      AUS(K,1)=EPAL
1785:      GMS(K,1)=XO*(GO+RT*PPL(1))+XM*(GPU+RT*APUL)
1786:      END IF
1787: C*****
1788:      DO 450 J=1, LL-1
1789:      OM=OMS-DM*DFLOAT(J)
1790:      XO=OM/(1.0+OM)
1791:      XM=1.0-XO
1792:      PL(K,J+1)=PPL(J+1)
1793:      DO 455 M=1, NS
1794:      IF (J.GT.NXY.AND.J.LT.(LL-NXY)) THEN
1795:      X1(M)=PPL(J-NXY+M)
1796:      Y1(M)=RX(J-NXY+M)
1797:      ELSE
1798:      IF (J.LE.2) THEN
1799:      X1(M)=PPL(M)
1800:      Y1(M)=RX(M)
1801:      ELSE
1802:      X1(M)=PPL(LL-NS+M)
1803:      Y1(M)=RX(LL-NS+M)
1804:      END IF
1805:      END IF
1806: 455 CONTINUE
1807:      AS1=PPL(J)
1808:      AT1=PPL(J+1)
1809:      CALL INTEGRA(NS,X1,Y1,AS1,AT1,DAUL)
1810: C      DAUL=- (RX(J+1)+RX(J))*(PPL(J+1)-PPL(J))/2.0
1811:      IF (K.EQ.1) THEN
1812:      AUL=AUL-DAUL
1813:      AUS(K,J+1)=AUL
1814:      APUS(K,J+1)=EPAL
1815:      GMS(K,J+1)=XO*(GO+RT*PPL(J+1))+XM*(GU+RT*AUL)
1816:      ELSE
1817:      APUL=APUL-DAUL
1818:      APUS(K,J+1)=APUL
1819:      AUS(K,J+1)=EPAL
1820:      GMS(K,J+1)=XO*(GO+RT*PPL(J+1))+XM*(GPU+RT*APUL)
1821:      END IF
1822: C
1823: c      WRITE(6,100) OM,XO,PPL(J),AUL,APUL,GMS(K,J+1)
1824: 450 CONTINUE
1825:      END IF
1826: 200 CONTINUE
1827: C      DO 700 K=1,3
1828: C      KK=1+100*(K-1)
1829: C      DO 700 J=1,LL
1830: C      WRITE(6,100) XPR(KK),OMR(J),PL(KK,J),AUS(KK,J),APUS(KK,J),GMS(KK,J)
1831: C      700 CONTINUE
1832: ccccccc
1833: 100 FORMAT(15E20.10)
1834:      RETURN
1835:      END
1836: C*****
1837: C*****
1838:      SUBROUTINE GUPUOL
1839:      IMPLICIT REAL*8 (A-H,O-Z)
1840:      COMMON /G03/GO,GU,GPU
1841:      COMMON/TZ/TK,XZ,YZ
1842:      COMMON/LLL/LL,NN
1843:      COMMON/GML/XPUS,OMS,DX,DM,XPR(1001),OMR(1001),GMS(1001,1001)
1844:      COMMON/AL/AUS(1001,1001),APUS(1001,1001),PL(1001,1001)
1845:      DIMENSION SF2(1001),SF3(1001),S2(1001),S3(1001)
1846:      1 .RX(1001),PPL(1001)
1847:      DIMENSION X1(10),Y1(10),X2(10),Y2(10)
1848: c***** MESH POINT SETTING
1849: C***** NN = XPU MESH NO. LL= OM MESH NO. L2 = MESH POINT AT OM=2.0
1850:      NN=201
1851:      LL=1001
1852:      L2=(LL-1)/10+1
1853:      NS=4
1854:      NXY=NS-2
1855:      XPUS=0.0
1856:      OMS=2.1
1857:      DX=1.0/DFLOAT(NN-1)
1858:      DM=1.0/DFLOAT(LL-1)
1859:      DDP=2.0D-8
1860:      EPS=1.0D-30
1861:      EPAL=DLOG(EPS)
1862:      LS=1
1863:      RG=8.313608
1864:      RT=TK*RG
1865:      WSL=0.0/RT
1866: C***** SETTING END
1867:      CALL GUO2L(TK,GUO2,GFUO2)
1868:      CALL GOXA(TK,GO,GFO)
1869:      CALL GUMET(TK,GU,GFU)

```

```

1870:      CALL GPU02L (TK, GPU02, GFPU02)
1871:      CALL GPUMET (TK, GPU, GFPU)
1872: CCCCC  U02
1873:      XPU=0. 0
1874:      OM=2. 0
1875:      CALL MOXOPZ (XPU, OM, TK, P02, PU0, OXP, LS)
1876:      AUL2=(GU02-2. 0*GO-GU-2. 0*RT*DLOG (PU0))/RT
1877: CCCCC  PU02
1878:      XPU=1. 0
1879:      OM=2. 0
1880:      CALL MOXOPZ (XPU, OM, TK, P02, PPU0, OXP, LS)
1881:      APUL2=(GPU02-2. 0*GO-GPU-2. 0*RT*DLOG (PPU0))/RT
1882: CCCCC  CALC. OF U & PU ACTIVITY IN U02-PU02
1883:      DO 200 K=1, NN
1884:      OM=2. 0
1885:      XPU=XPU+DX*DFLOAT (K-1)
1886:      XU=1. 0-XPU
1887:      XPR (K) =XPU
1888:      CALL MOXOPZ (XPU, OM, TK, P02, PO, OXP, LS)
1889:      POL=DLOG (PO)
1890:      AUL=2. 0*DLOG (PU0/PO) +AUL2+DLOG (XU+EPS) +XPU**2*WSL
1891:      APUL=2. 0*DLOG (PPU0/PO) +APUL2+DLOG (XPU+EPS) +XU**2*WSL
1892:      AUS (K, L2) =AUL
1893:      APUS (K, L2) =APUL
1894:      PL (K, L2) =POL
1895: C      GSM=XU*GU02+XPU*GPU02+RT* (XU*DLOG (XU+EPS) +XPU*DLOG (XPU+EPS))
1896:      GM3=2. 0* (GO+RT*DLOG (PO)) + (XU* (GU+RT*AUL) +XPU* (GPU+RT*APUL))
1897:      GMS (K, L2) =GM3/3. 0
1898: C      WRITE (6, 100) XU, POL, APUL, AUL, AUL2, APUL2, GSM, GM3
1899: C*****
1900:      IF (K. NE. 1. AND. K. NE. NN) THEN
1901:      ST=XPU
1902:      STR=1. 0-ST
1903:      DO 300 J=1, LL
1904:      OM=OMS-DM*DFLOAT (J-1)
1905:      CALL MOXOPZ (XPU, OM, TK, P02, PO, OXP, LS)
1906:      POL=DLOG (PO)
1907:      PL (K, J) =POL
1908:      OMR (J) =OM
1909:      XO=OM/ (1. 0+OM)
1910:      XP=XPU/ (1. 0+OM)
1911:      XU= (1. 0-XPU) / (1. 0+OM)
1912:      S3 (J) =XU/ (XO+XU)
1913:      S2 (J) =XP/ (XO+XP)
1914:      XP1=XP+DDP
1915:      XP2=XP-DDP
1916:      XU31= (1. 0-XP1) *S3 (J)
1917:      X031= (1. 0-XP1) * (1. 0-S3 (J))
1918:      XU32= (1. 0-XP2) *S3 (J)
1919:      X032= (1. 0-XP2) * (1. 0-S3 (J))
1920: CCCCC
1921:      XU21=1. 0-XP1/S2 (J)
1922:      X021= (1. 0-S2 (J)) *XP1/S2 (J)
1923:      XU22=1. 0-XP2/S2 (J)
1924:      X022= (1. 0-S2 (J)) *XP2/S2 (J)
1925: CCCCC
1926:      DST3= (XP1/ (XP1+XU31)) - (XP2/ (XP2+XU32))
1927:      DST2= (XP1/ (XP1+XU21)) - (XP2/ (XP2+XU22))
1928: CCCCC
1929:      OM31=X031/ (1. 0-X031)
1930:      XPU31=XP1/ (1. 0-X031)
1931:      OM32=X032/ (1. 0-X032)
1932:      XPU32=XP2/ (1. 0-X032)
1933:      CALL MOXOPZ (XPU31, OM31, TK, P02, P031, OXP, LS)
1934:      CALL MOXOPZ (XPU32, OM32, TK, P02, P032, OXP, LS)
1935:      SF3 (J) =- (STR/S3 (J)) **2*DLOG (P031/P032) /DST3
1936:      OM21=X021/ (1. 0-X021)
1937:      XPU21=XP1/ (1. 0-X021)
1938:      OM22=X022/ (1. 0-X022)
1939:      XPU22=XP2/ (1. 0-X022)
1940:      CALL MOXOPZ (XPU21, OM21, TK, P02, P021, OXP, LS)
1941:      CALL MOXOPZ (XPU22, OM22, TK, P02, P022, OXP, LS)
1942:      SF2 (J) = (ST/S2 (J)) **2*DLOG (P021/P022) /DST2
1943: C      WRITE (6, 100) DST2, DST3, SF2 (J), SF3 (J)
1944:      300 CONTINUE
1945: C*****
1946:      SSU=0. 0
1947:      SSP=0. 0
1948:      DO 320 J=1, L2-1
1949:      DO 302 M=1, NS
1950:      IF (J. GT. NXY) THEN
1951:      X1 (M) =S3 (J-NXY+M)
1952:      Y1 (M) =SF3 (J-NXY+M)
1953:      X2 (M) =S2 (J-NXY+M)
1954:      Y2 (M) =SF2 (J-NXY+M)
1955:      ELSE
1956:      X1 (M) =S3 (M)
1957:      X2 (M) =S2 (M)
1958:      Y1 (M) =SF3 (M)

```

```

1959:      Y2(M)=SF2(M)
1960:      END IF
1961: 302 CONTINUE
1962:      AS1=S3(J)
1963:      AT1=S3(J+1)
1964:      AS2=S2(J)
1965:      AT2=S2(J+1)
1966:      CALL INTEGRA(NS, X1, Y1, AS1, AT1, DSP)
1967:      CALL INTEGRA(NS, X2, Y2, AS2, AT2, DSU)
1968:C      DSP=(SF3(J+1)+SF3(J))*(S3(J+1)-S3(J))/2.0
1969:C      DSU=(SF2(J+1)+SF2(J))*(S2(J+1)-S2(J))/2.0
1970:      SSP=SSP+DSP
1971:      SSU=SSU+DSU
1972: 320 CONTINUE
1973:      APUL=APUL-SSP
1974:      AUL=AUL-SSU
1975:      AUS(K,1)=AUL
1976:      APUS(K,1)=APUL
1977:      POL=PL(K,1)
1978:      OM=OMS
1979:      XO=OM/(1.0+OM)
1980:      XP=XPU/(1.0+OM)
1981:      XU=(1.0-XPU)/(1.0+OM)
1982:      GMS(K,1)=XO*(GO+RT*POL)+XP*(GPU+RT*APUL)+XU*(GU+RT*AUL)
1983:C*****
1984:      DO 350 J=1, LL-1
1985:      DO 305 M=1, NS
1986:      IF (J.GT.NXY.AND.J.LT.(LL-NXY)) THEN
1987:      X1(M)=S3(J-NXY+M)
1988:      Y1(M)=SF3(J-NXY+M)
1989:      X2(M)=S2(J-NXY+M)
1990:      Y2(M)=SF2(J-NXY+M)
1991:      ELSE
1992:      IF (J.LE.2) THEN
1993:      X1(M)=S3(M)
1994:      Y1(M)=SF3(M)
1995:      X2(M)=S2(M)
1996:      Y2(M)=SF2(M)
1997:      ELSE
1998:      X1(M)=S3(LL-NS+M)
1999:      Y1(M)=SF3(LL-NS+M)
2000:      X2(M)=S2(LL-NS+M)
2001:      Y2(M)=SF2(LL-NS+M)
2002:      END IF
2003:      END IF
2004: 305 CONTINUE
2005:      AS1=S3(J)
2006:      AT1=S3(J+1)
2007:      AS2=S2(J)
2008:      AT2=S2(J+1)
2009:      CALL INTEGRA(NS, X1, Y1, AS1, AT1, DAPL)
2010:      CALL INTEGRA(NS, X2, Y2, AS2, AT2, DAUL)
2011:C      DAPL=(SF3(J+1)+SF3(J))*(S3(J+1)-S3(J))/2.0
2012:      APUL=APUL+DAPL
2013:C      DAUL=(SF2(J+1)+SF2(J))*(S2(J+1)-S2(J))/2.0
2014:      AUL=AUL+DAUL
2015:      APUS(K,J+1)=APUL
2016:      AUS(K,J+1)=AUL
2017:      POL=PL(K,J+1)
2018:      OM=OMS-DM*DFLOAT(J)
2019:      XO=OM/(1.0+OM)
2020:      XP=XPU/(1.0+OM)
2021:      XU=(1.0-XPU)/(1.0+OM)
2022:      GMS(K,J+1)=XO*(GO+RT*POL)+XP*(GPU+RT*APUL)+XU*(GU+RT*AUL)
2023:C      WRITE(6,100)OM,POL,AUL,APUL,GMS(K,J+1)
2024:      350 CONTINUE
2025:C***** FOR U02 OR PU02 CASE
2026:      ELSE
2027:      DO 400 J=1, LL
2028:      OM=OMS-DM*DFLOAT(J-1)
2029:      XO=OM/(1.0+OM)
2030:      XM=1.0-XO
2031:      RX(J)=XO/XM
2032:      IF (K.EQ.1) THEN
2033:      XPU=0.0
2034:      CALL MOXOPZ(XPU,OM,TK,P02,PO,XP,LS)
2035:      PPL(J)=DLOG(PO)
2036:      ELSE
2037:      XPU=1.0
2038:      CALL MOXOPZ(XPU,OM,TK,P02,PO,XP,LS)
2039:      PPL(J)=DLOG(PO)
2040:      END IF
2041: 400 CONTINUE
2042:C*****
2043:      SS=0.0
2044:      DO 420 J=1, L2-1
2045:      PL(K,J+1)=PPL(J+1)
2046:      DO 402 M=1, NS
2047:      IF (J.GT.NXY) THEN

```



```

2048:      X1 (M)=PPL (J-NXY+M)
2049:      Y1 (M)=RX (J-NXY+M)
2050:      ELSE
2051:      X1 (M)=PPL (M)
2052:      Y1 (M)=RX (M)
2053:      END IF
2054: 402 CONTINUE
2055:      AS1=PPL (J)
2056:      AT1=PPL (J+1)
2057:      CALL INTEGRA (NS, X1, Y1, AS1, AT1, DAUL)
2058: C      DAUL=- (RX (J+1)+RX (J)) * (PPL (J+1)-PPL (J)) /2. 0
2059:      SS=SS-DAUL
2060: 420 CONTINUE
2061:      OM=OMS
2062:      XO=OM / (1. 0+OM)
2063:      XM=1. 0-XO
2064:      PL (K, 1)=PPL (1)
2065:      IF (K. EQ. 1) THEN
2066:      AUL=AUL2-SS
2067:      AUS (K, 1)=AUL
2068:      APUS (K, 1)=EPAL
2069:      GMS (K, 1)=XO* (GO+RT*PPL (1))+XM* (GU+RT*AUL)
2070:      ELSE
2071:      APUL=APUL2-SS
2072:      APUS (K, 1)=APUL
2073:      AUS (K, 1)=EPAL
2074:      GMS (K, 1)=XO* (GO+RT*PPL (1))+XM* (GPU+RT*APUL)
2075:      END IF
2076: C*****
2077:      DO 450 J=1, LL-1
2078:      OM=OMS-DM*DFLOAT (J)
2079:      XO=OM / (1. 0+OM)
2080:      XM=1. 0-XO
2081:      PL (K, J+1)=PPL (J+1)
2082:      DO 455 M=1, NS
2083:      IF (J. GT. NXY. AND. J. LT. (LL-NXY)) THEN
2084:      X1 (M)=PPL (J-NXY+M)
2085:      Y1 (M)=RX (J-NXY+M)
2086:      ELSE
2087:      IF (J. LE. 2) THEN
2088:      X1 (M)=PPL (M)
2089:      Y1 (M)=RX (M)
2090:      ELSE
2091:      X1 (M)=PPL (LL-NS+M)
2092:      Y1 (M)=RX (LL-NS+M)
2093:      END IF
2094:      END IF
2095: 455 CONTINUE
2096:      AS1=PPL (J)
2097:      AT1=PPL (J+1)
2098:      CALL INTEGRA (NS, X1, Y1, AS1, AT1, DAUL)
2099: C      DAUL=- (RX (J+1)+RX (J)) * (PPL (J+1)-PPL (J)) /2. 0
2100:      IF (K. EQ. 1) THEN
2101:      AUL=AUL-DAUL
2102:      AUS (K, J+1)=AUL
2103:      APUS (K, J+1)=EPAL
2104:      GMS (K, J+1)=XO* (GO+RT*PPL (J+1))+XM* (GU+RT*AUL)
2105:      ELSE
2106:      APUL=APUL-DAUL
2107:      APUS (K, J+1)=APUL
2108:      AUS (K, J+1)=EPAL
2109:      GMS (K, J+1)=XO* (GO+RT*PPL (J+1))+XM* (GPU+RT*APUL)
2110:      END IF
2111: C
2112: C      WRITE (6, 100) OM, XO, PPL (J), AUL, APUL, GMS (K, J+1)
2113: 450 CONTINUE
2114:      END IF
2115: 200 CONTINUE
2116: C      DO 700 K=1, 3
2117: C      KK=1+100* (K-1)
2118: C      DO 700 J=1, LL
2119: C      WRITE (6, 100) XPR (KK), OMR (J), PL (KK, J), AUS (KK, J), APUS (KK, J), GMS (KK, J)
2120: C      700 CONTINUE
2121: CCCCCCC
2122: 100 FORMAT (15E20, 10)
2123:      RETURN
2124:      END
2125: C*****
2126: C*****
2127: CCCCCCCCCCCCCCCCCC
2128: C*****
2129:      SUBROUTINE DBUO
2130:      IMPLICIT REAL*8 (A-H, O-Z)
2131:      COMMON/DUM/D1, D2, D3, D4
2132:      COMMON /G01/G1S, G2S, G1L, G2L
2133:      COMMON/TZ/TK, XZ
2134:      COMMON/WREF/WS, WL1, WL2
2135:      COMMON/CUOS/CPO (1000), CPU (1000), AU (1000), PP (1000), GGG (1000)
2136:      COMMON/WUO/OMS, OM2, DOM, OMR (1000), WW (1000)

```

```

2137: COMMON/CPUL/CPOL (1000), CPUL (1000), AUL (1000), PPL (1000), GGGL (1000)
2138: COMMON/WUOLL/OMSL, OM2L, DOML, OMRL (1000), WWL (1000)
2139: COMMON/NOM/NOM
2140: COMMON/NOL/NOL
2141: XPU=0.0
2142: NN=1
2143: DO 300 K=1, NN
2144: TK=2860.0+273.15
2145: CALL GOXA (TK, G1S, GFO)
2146: CALL GUMET (TK, G2S, GFU)
2147: CALL WUOS
2148: CALL WUOL
2149: DO 500 J=1, 210
2150: OM=2.005-DFLOAT (J) *0.0002
2151: XX=OM / (1.0+OM)
2152: GS=FUOS (XX)
2153: GL2=FUOL2 (XX)
2154: GL1=FUOL1 (XX)
2155: WRITE (6, 100) TK, OM, XX, GS, GL1, GL2
2156: 500 CONTINUE
2157: 300 CONTINUE
2158: 100 FORMAT (10E20, 7)
2159: RETURN
2160: END
2161: C*****
2162: CCCCCCCCCCCCCCCCCC
2163: C*****
2164: SUBROUTINE DBCAL
2165: IMPLICIT REAL*8 (A-H, O-Z)
2166: COMMON/DUM/D1, D2, D3, D4
2167: COMMON /G01/G1S, G2S, G1L, G2L
2168: COMMON/TZ/TK, XZ
2169: COMMON/WREF/WS, WL1, WL2
2170: COMMON/WUO/OMS, OM2, DOM, OMR (1000), WW (1000)
2171: COMMON/WUOLL/OMSL, OM2L, DOML, OMRL (1000), WWL (1000)
2172: COMMON/NOM/NOM
2173: COMMON/NOL/NOL
2174: COMMON/WPUO/OMPS, OMP2S, DOMPS, OMRPS (1000), WPOS (1000)
2175: COMMON/NOP/NOP
2176: C OPEN (6, FILE=' ', STATUS='NEW')
2177: XPU=0.0
2178: TK=2600.0+273.15
2179: CALL GU02S (TK, G1S, GUSF)
2180: CALL GPU02S (TK, G2S, GPUSF)
2181: CALL GU02L (TK, G1L, GULF)
2182: CALL GPU02L (TK, G2L, GPULF)
2183: DO 500 J=1, 501
2184: XX=0.0+0.002*DFLOAT (J-1)
2185: GS=FMOXS (XX)
2186: GL=FMOXL (XX)
2187: WRITE (6, 100) XX, GS, GL
2188: 500 CONTINUE
2189: 100 FORMAT (10E20, 7)
2190: RETURN
2191: END
2192: C*****
2193: C*****
2194: SUBROUTINE DBCPUO
2195: IMPLICIT REAL*8 (A-H, O-Z)
2196: COMMON/TZ/TK, XZ
2197: COMMON/WPUO/OMPS, OMP2S, DOMPS, OMRPS (1000), WPOS (1000)
2198: COMMON/NOP/NOP
2199: COMMON/CPUL/CPOL (1000), CPUL (1000), AUL (1000), PPL (1000), GGL (1000)
2200: COMMON/WPUOL/OMPL, OMP2L, DOMPL, OMRPL (1000), WPOL (1000)
2201: COMMON/CPUS/CPPOS (1000), CPUS (1000), AUS (1000), PPS (1000), GGS (1000)
2202: C OPEN (6, FILE=' ', STATUS='NEW')
2203: XPU=1.0
2204: DO 1100 J=1, 1
2205: TK=2245.0+273.15+ (J-1)*500.0
2206: CALL WPUOS
2207: CALL WPUOLL
2208: DO 1100 K=1, NOP
2209: WRITE (6, 100) TK, OMRPS (K), pps (k), ppl (k)
2210: 1100 CONTINUE
2211: 1050 CONTINUE
2212: 1000 CONTINUE
2213: 100 FORMAT (10E20, 7)
2214: RETURN
2215: END
2216: C*****
2217: C*****
2218: SUBROUTINE DBCAL3
2219: IMPLICIT REAL*8 (A-H, O-Z)
2220: COMMON/TZ/TK, XZ, YZ
2221: DIMENSION XX (2), YY (2)
2222: C OPEN (6, FILE=' ', STATUS='NEW')
2223: TK=2400.0+273.15
2224: C CALL GUPUOS
2225: C CALL GUPUOL

```

```

2226:      LOM=101
2227:      DO 1000 J=1, 1
2228:      OM=1.8637+0.0001*DFLOAT(J-1)
2229:      DO 1200 K=1, 1
2230:      XPU=0.63362+0.0001*DFLOAT(K)
2231:      XS=OM/(1.0+OM)
2232:      XM=1.0-XS
2233:      YS=XM*XPU
2234:      XL=XS
2235:      YL=YS
2236:      GGS=FM3S(XS,YS)
2237:      GGL=FM3L(XL,YL)
2238:      WRITE(6,100)XS,YS,XPU,OM,GGS,GGL
2239: 1200 CONTINUE
2240: 1000 CONTINUE
2241: 100 FORMAT(20E20.10)
2242:      RETURN
2243:      END
2244: C*****
2245: C*****
2246:      SUBROUTINE DBCAL4
2247:      IMPLICIT REAL*8 (A-H, O-Z)
2248:      COMMON/TZ/TK, XZ, YZ
2249:      COMMON/FL11/FL11, FL21, FL22, FL31
2250:      COMMON/LLS/LS, NS
2251:      COMMON/GMS/XPUS, OMSS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
2252:      COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PLS(1001, 1001)
2253:      COMMON/LLL/LL, NL
2254:      COMMON/GML/XPUSL, OMSL, DXL, DML, XPRL(1001), OMRL(1001)
2255: 1. GMSL(1001, 1000)
2256:      COMMON/AL/AUSL(1001, 1001), APUSL(1001, 1001), PLL(1001, 1001)
2257:      COMMON /GO3/GO, GU, GPU
2258:      DIMENSION XX(2), YY(2)
2259: C OPEN(6, FILE=' ', STATUS='NEW')
2260:      TK=2600.0+273.15
2261:      RG=8.313608
2262:      RT=RG*TK
2263:      EPS=1.D-30
2264: C CALL GUO2S(TK, GUO2, GFUO2)
2265: C CALL GOXA(TK, GO, GFO)
2266: C CALL GUMET(TK, GU, GFU)
2267: C CALL GPUO2S(TK, GPUO2, GFPUO2)
2268: C CALL GPUMET(TK, GPU, GFPU)
2269: C CALL GUO2L(TK, GUO2LL, GFUO2)
2270: C CALL GPUO2L(TK, GPUO2LL, GFPUO2)
2271:      CALL GUPUOS
2272:      CALL GUPUOL
2273:      DO 500 J=1, 3
2274:      N=101+200*(J-1)
2275:      OM=OMR(N)
2276:      DO 501 JJ=1, 201
2277:      K=JJ
2278:      XPU=XPR(K)
2279:      XU=1.0-XPU
2280:      WRITE(6,100)OM, XPU, GMS(K, N), AUS(K, N), APUS(K, N), PLS(K, N),
2281: 1 GMSL(K, N), AUSL(K, N), APUSL(K, N), PLL(K, N)
2282: 501 CONTINUE
2283: 500 CONTINUE
2284: 1000 CONTINUE
2285: 100 FORMAT(20E20.10)
2286:      RETURN
2287:      END
2288: C*****
2289: C*****
2290:      SUBROUTINE DBCALT
2291:      IMPLICIT REAL*8 (A-H, O-Z)
2292:      COMMON/TZ/TK, XZ, YZ
2293:      RG=8.313608
2294:      EPS=1.D-30
2295:      NT=280
2296:      DO 500 J=1, NT
2297:      TK=500.0+10.0*(J-1)
2298:      CALL GGO2(TK, GO2, GFO2)
2299:      CALL GOXA(TK, GO, GFO)
2300:      CALL GUMET(TK, GU, GFU)
2301:      CALL GPUMET(TK, GPU, GFPU)
2302:      CALL GUO2S(TK, GUO2, GFUO2)
2303:      CALL GPUO2S(TK, GPUO2, GFPUO2)
2304:      CALL GUO2L(TK, GUO2LL, GFUO2)
2305:      CALL GPUO2L(TK, GPUO2LL, GFPUO2)
2306:      T1=1.0/TK
2307:      WRITE(6,100)TK, T1, GO2, GO, GU, GPU, GUO2, GPUO2, GUO2LL, GPUO2LL
2308: 500 CONTINUE
2309: 100 FORMAT(20E20.10)
2310:      RETURN
2311:      END
2312: C*****
2313: C*****
2314:      SUBROUTINE DBCALA

```

```

2315: IMPLICIT REAL*8 (A-H, O-Z)
2316: COMMON/TZ/TK, XZ, YZ
2317: COMMON/FL11/FL11, FL21, FL22, FL31
2318: COMMON/LLS/LS, NS
2319: COMMON/GMS/XPUS, OMSS, DX, DM, XPR(1001), OMR(1001), GMS(1001, 1001)
2320: COMMON /AS/AUS(1001, 1001), APUS(1001, 1001), PLS(1001, 1001)
2321: COMMON/LLL/LL, NL
2322: COMMON/GML/XPUSL, OMSL, DXL, DML, XPRL(1001), OMRL(1001)
2323: 1, GMSL(1001, 1001)
2324: COMMON/AL/AUSL(1001, 1001), APUSL(1001, 1001), PLL(1001, 1001)
2325: COMMON /G03/GO, GU, GPU
2326: DIMENSION XX(2), YY(2)
2327: C OPEN(6, FILE=' ', STATUS=' NEW' )
2328: NQ=7
2329: QQQ=-3.0
2330: TK=2600.0+273.15
2331: CALL GUPUOS
2332: C CALL GUPUOL
2333: DO 200 J=1, NQ
2334: QA=QQQ-3.0*DFLOAT(J-1)
2335: DO 300 K=1, NS
2336: DO 400 L=1, LS
2337: AA=PLS(K, L)
2338: DQ=AA-QA
2339: IF (DQ.LT.0.0) THEN
2340: XPU=DFLOAT(K-1)/200.0
2341: OM=2.1-DFLOAT(L-1)/1000.0
2342: WRITE(6, 100)QA, XPU, OM
2343: GO TO 300
2344: ELSE
2345: END IF
2346: 400 CONTINUE
2347: 300 CONTINUE
2348: 200 CONTINUE
2349: 100 FORMAT(20E20, 10)
2350: RETURN
2351: END
2352: C*****
2353: C*****
2354: C*****
2355: SUBROUTINE YPO
2356: IMPLICIT REAL*8 (A-H, O-Z)
2357: COMMON/DUM/D1, D2, D3, D4
2358: COMMON /G01/GO, GU, G1L, G2L
2359: COMMON/TZ/TK, XZ
2360: LS=0
2361: LOP=1
2362: IF (LOP.EQ.0) THEN
2363: READ(5, *) NN
2364: DO 300 K=1, NN
2365: READ(5, *) XPU, TC, OM, PLM
2366: TK=TC+273.15
2367: CALL MOXOPZ(XPU, OM, TK, PO2, PO, OXP, LS)
2368: PLC=DLOG10(PO2)
2369: DP=PLM-PLC
2370: C POL=DLOG10(PO)
2371: C WRITE(6, 100)TC, XPU, OM, PLM, PLC, DP
2372: 300 CONTINUE
2373: ELSE
2374: DO 650 L=1, 2
2375: IF (L.EQ.1) THEN
2376: TK=2400.0+273.15
2377: ELSE
2378: TK=1545.0+273.15
2379: END IF
2380: DO 750 J=1, 1
2381: XPU=1.0+0.005*DFLOAT(J-1)
2382: DO 700 K=1, 500
2383: OM=2.0-DFLOAT(K-1)*0.001
2384: CALL MOXOPZ(XPU, OM, TK, PO2, PO, OXP, LS)
2385: PO2L=DLOG10(PO2)
2386: WRITE(6, 100) XPU, TK, OM, PO2L
2387: 700 CONTINUE
2388: 750 CONTINUE
2389: 650 CONTINUE
2390: END IF
2391: 500 CONTINUE
2392: 100 FORMAT(10E20, 7)
2393: RETURN
2394: END
2395: C*****
2396: C*****
2397: C*****
2398: C*****
2399: SUBROUTINE PDAGAU1
2400: C CALCURATION OF AG-AU SYSTEM //01
2401: CCCCC
2402: C
2403: IMPLICIT REAL*8 (A-H, O-Z)

```

```

2404: PARAMETER (N=2)
2405: EXTERNAL FG1, FS1, FL1
2406: DIMENSION X(N), TS(2)
2407: COMMON /G01/G1S, G2S, G1L, G2L
2408: COMMON/TZ/TK, XZ
2409: COMMON/QAXY/QAXY
2410: C OPEN(6, FILE=' ', STATUS='NEW')
2411: READ(5, *) LLW, LQ2, TC1, TC2, DT
2412: READ(5, *) NC, ZA, CV
2413: WRITE(6, *) LLW, LQ2, TC1, TC2, DT
2414: WRITE(6, *) NC, ZA, CV
2415: XMAX=1.0
2416: XMIN=0.0
2417: C
2418: ITTN = INT((TC2-TC1+0.001)/DT)+1
2419: DO 1000 IT=1, ITTN
2420: TTC=TC1+(IT-1)*DT
2421: TK=TTC+273.15
2422: CALL SGAGS(TK, G1S)
2423: CALL SGAUS(TK, G2S)
2424: CALL SGAGL(TK, G1L)
2425: CALL SGAUL(TK, G2L)
2426: DO 2000 KZ=1, NC
2427: XZ=ZA+CV*(KZ-1)
2428: IF(XZ.GT.XMAX) GO TO 1000
2429: IF(NC.EQ.1) XZ=ZA
2430: CALL SSCN(FG1, XMAX, XMIN, X, XZ, AP, FMIN, ITR, JPH, LLW)
2431: IF(JPH.LE.0.AND.LQ2.EQ.2) GO TO 2000
2432: CALL CHK22(X, DCX, DCY)
2433: CPS=ETIME(TS)
2434: WRITE(6, 120) TTC, XZ, ITR, JPH, X(1), X(2), AP, FMIN, DCX, DCY, CPS
2435: 2000 CONTINUE
2436: 1000 CONTINUE
2437: 120 FORMAT(F12.1, F13.8, 2I5, 3F13.6, E30.18, 2E11.2, F13.6)
2438: RETURN
2439: END
2440: C*****
2441: SUBROUTINE CHK22(X, DCX, DCY)
2442: IMPLICIT REAL*8 (A-H, O-Z)
2443: COMMON /W2/WS, WL
2444: COMMON /G01/G1S, G2S, G1L, G2L
2445: COMMON/TZ/TK, XZ
2446: DIMENSION X(2)
2447: EX=1.0D-30
2448: RR=8.313608
2449: RT=RR*TK
2450: XS=X(1)
2451: XL=X(2)
2452: YS=1.0-XS
2453: YL=1.0-XL
2454: CXS=C1S+RT*DLOG(XS+EX)+WS+YS**2
2455: CXL=G1L+RT*DLOG(XL+EX)+WL*YL**2
2456: CYS=G2S+RT*DLOG(YS+EX)+WS*XS**2
2457: CYL=G2L+RT*DLOG(YL+EX)+WL*XL**2
2458: DCX=2.0*(CXS-CXL)/(CXS+CXL)
2459: DCY=2.0*(CYS-CYL)/(CYS+CYL)
2460: RETURN
2461: END
2462: C*****
2463: FUNCTION FS1(X)
2464: IMPLICIT REAL*8 (A-H, O-Z)
2465: COMMON /G01/G1S, G2S, G1L, G2L
2466: COMMON/TZ/TK, XZ
2467: COMMON /W2/W, WL
2468: RR=8.313608
2469: RT=RR*TK
2470: SX=-5.1194*X**4+12.353*X**3-10.377*X**2+3.1393*X+0.7
2471: W=-140000.0*SX
2472: EX=1.0D-50
2473: FS1=X*(G1S+RT*DLOG(X+EX))
2474: 1+(1.0-X)*(G2S+RT*DLOG(1.0-X+EX))
2475: 2+X*(1.0-X)*W
2476: C FS1=X*G1S+(1.0-X)*G2S
2477: RETURN
2478: END
2479: C*****
2480: C*****
2481: C*****
2482: FUNCTION FL1(X)
2483: IMPLICIT REAL*8 (A-H, O-Z)
2484: COMMON /G01/G1S, G2S, G1L, G2L
2485: COMMON/TZ/TK, XZ
2486: COMMON /W2/WS, W
2487: RR=8.313608
2488: RT=RR*TK
2489: SX=-5.1194*X**4+12.353*X**3-10.377*X**2+3.1393*X+0.7
2490: W=-100000.0*SX
2491: EX=1.0D-50
2492: FL1=X*(G1L+RT*DLOG(X+EX))

```

```

2493:      1 +(1. 0-X)*(G2L+RT*DLOG(1. 0-X+EX))
2494:      2 +X*(1. 0-X)*W
2495:      RETURN
2496:      END
2497: C*****
2498: C*****
2499: C*****
2500: C*****
2501: C*****
2502: C*****
2503:      FUNCTION FG1 (AP, X)
2504:      IMPLICIT REAL*8 (A-H, O-Z)
2505:      DIMENSION X(2)
2506:      COMMON /TZ/, TK, XZ
2507:      FG1=AP*FS1 (X(1)) +(1. 0-AP)*FL1 (X(2))
2508:      RETURN
2509:      END
2510: C*****
2511:      SUBROUTINE PDAGCU1
2512: C CALCURATION OF AG-CU SYSTEM //01
2513: CCCCC
2514: C
2515:      IMPLICIT REAL*8 (A-H, O-Z)
2516:      PARAMETER (N=2)
2517:      EXTERNAL FCG1, FCS1, FCL1
2518:      DIMENSION X(N), TS(2)
2519:      COMMON /G01/G1S, G2S, G1L, G2L
2520:      COMMON /TZ/, TK, XZ
2521: C      COMMON /QAXY/, QAXY
2522: C      OPEN (6, FILE=' ', STATUS='NEW')
2523: C      READ (5, *) LLW, LQ2, TC1, TC2, DT
2524: C      READ (5, *) NC, ZA, CV
2525: C      WRITE (6, *) LLW, LQ2, TC1, TC2, DT
2526: C      WRITE (6, *) NC, ZA, CV
2527: C      XMAX=1. 0
2528: C      XMIN=0. 0
2529: C
2530:      ITTN = INT((TC2-TC1+0. 001)/DT)+1
2531:      DO 1000 IT=1, ITTN
2532:      TTC=TC1+(IT-1)*DT
2533:      TK=TTC+273. 15
2534:      CALL SGAGS (TK, G1S)
2535:      CALL SGCUS (TK, G2S)
2536:      CALL SGAGL (TK, G1L)
2537:      CALL SGCUJ (TK, G2L)
2538:      DO 2000 KZ=1, NC
2539:      XZ=ZA+CV*(KZ-1)
2540:      IF (XZ. GT. XMAX) GO TO 1000
2541:      IF (NC. EQ. 1) XZ=ZA
2542:      CALL SSCN (FCG1, XMAX, XMIN, X, XZ, AP, FMIN, ITR, JPH, LLW)
2543:      IF (JPH. LE. 0. AND. LQ2. EQ. 2) GO TO 2000
2544:      CALL CHK22 (X, DCX, DCY)
2545:      CPS=ETIME (TS)
2546:      WRITE (6, 120) TTC, XZ, ITR, JPH, X(1), X(2), AP, FMIN, DCX, DCY, CPS
2547: 2000 CONTINUE
2548: 1000 CONTINUE
2549: 120 FORMAT (F12. 1, F13. 8, 2I5, 3F13. 6, E30, 18, 2E11. 2, F13. 6)
2550:      RETURN
2551:      END
2552: C*****
2553: C*****
2554:      FUNCTION FGS1 (X)
2555:      IMPLICIT REAL*8 (A-H, O-Z)
2556:      COMMON /G01/G1S, G2S, G1L, G2L
2557:      COMMON /TZ/, TK, XZ
2558:      COMMON /W2/, W, WL
2559:      RR=8. 313608
2560:      RT=RR*TK
2561:      SX=-1. 2*X*X+1. 2*X+0. 7
2562:      W=-150000. 0*SX
2563:      EX=1. 0D-50
2564:      FCS1=X*(G1S+RT*DLOG (X+EX))
2565:      1 +(1. 0-X)*(G2S+RT*DLOG (1. 0-X+EX))
2566:      2 +X*(1. 0-X)*W
2567: C      FS1=X*G1S+(1. 0-X)*G2S
2568:      RETURN
2569:      END
2570: C*****
2571: C*****
2572: C*****
2573:      FUNCTION FGL1 (X)
2574:      IMPLICIT REAL*8 (A-H, O-Z)
2575:      COMMON /G01/G1S, G2S, G1L, G2L
2576:      COMMON /TZ/, TK, XZ
2577:      COMMON /W2/, WS, W
2578:      RR=8. 313608
2579:      RT=RR*TK
2580:      SX=-1. 2*X*X+1. 2*X+0. 7
2581:      W=-100000. 0*SX

```

```

2582: EX=1. D-50
2583: FCL1=X*(G1L+RT*DLOG(X+EX))
2584: 1+(1.0-X)*(G2L+RT*DLOG(1.0-X+EX))
2585: 2+X*(1.0-X)*W
2586: RETURN
2587: END
2588:C*****
2589:C*****
2590:C*****
2591:C*****
2592:C*****
2593:C*****
2594: FUNCTION FCG1 (AP, X)
2595: IMPLICIT REAL*8 (A-H, O-Z)
2596: DIMENSION X(2)
2597: COMMON /TZ/ TK, XZ
2598: FCG1=AP*FCS1(X(1))+(1.0-AP)*FCL1(X(2))
2599: RETURN
2600: END
2601: CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2602: SUBROUTINE DBCAL2
2603: IMPLICIT REAL*8 (A-H, O-Z)
2604: COMMON /G01/G1S, G2S, G1L, G2L
2605: COMMON/TZ/TK, XZ
2606: TK=700.0+273.15
2607: LOM=201
2608: CALL SGAGS(TK, G1S)
2609: CALL SGAUS(TK, G2S)
2610: CALL SGAGL(TK, G1L)
2611: CALL SGAUL(TK, G2L)
2612: DO 1100 IOM=1, LOM
2613: X=0.0+0.005*(IOM-1)
2614: AA=FS1(X)
2615: BB=FL1(X)
2616: WRITE(6,100) TK, X, AA, BB
2617: 1100 CONTINUE
2618: 1050 CONTINUE
2619: 1000 CONTINUE
2620: 100 FORMAT(10E20.7)
2621: RETURN
2622: END
2623:C*****
2624:C*****
2625:C*****
2626: SUBROUTINE PHDNOX1
2627:C CALCURATION OF UO2-PUO2 IDEAL SOLUTION //01
2628: CCCCC
2629: C
2630: IMPLICIT REAL*8 (A-H, O-Z)
2631: PARAMETER (N=2)
2632: EXTERNAL FGO1, FMOXS, FMOXL
2633: DIMENSION X(N), TS(2)
2634: COMMON /G01/G1S, G2S, G1L, G2L
2635: COMMON/TZ/TK, XZ
2636: C OPEN(6, FILE=' ', STATUS='NEW')
2637: READ(5,*) LLW, LLO2, NC, TC1, TC2, ZA, CV
2638: WRITE(6,*) LLW, LLO2, NC, TC1, TC2, ZA, CV
2639: C LLW=2
2640: C TC1=2410.0
2641: C TC2=2410.0
2642: C DT=10.0
2643: C XMAX=1.0
2644: C XMIN=0.0
2645: C
2646: ITTN = INT((TC2-TC1+0.001)/DT)+1
2647: DO 1000 IT=1, ITTN
2648: TTC=TC1+(IT-1)*DT
2649: TK=TTC+273.15
2650: CALL GUO2S(TK, G1S, GUSF)
2651: CALL GPUO2S(TK, G2S, GPUSF)
2652: CALL GUO2L(TK, G1L, GULF)
2653: CALL GPUO2L(TK, G2L, GPULF)
2654: C WRITE(6,400) G1S, G2S, G1L, G2L
2655: C 400 FORMAT(4E25.8)
2656: C GO TO 1000
2657: C DO 2000 KZ=1, NC
2658: C XZ=ZA+CV*(KZ-1)
2659: C IF(NC.EQ.1) XZ=ZA
2660: C CALL SSCN(FGO1, XMAX, XMIN, X, XZ, AP, FMIN, ITR, JPH, LLW)
2661: C IF(JPH.LE.0.AND.LLO2.EQ.2) GO TO 2000
2662: C CPS=ETIME(TS)
2663: C WRITE(6,120) TTC, XZ, ITR, JPH, X(1), X(2), AP, FMIN, CPS
2664: 2000 CONTINUE
2665: 1000 CONTINUE
2666: 120 FORMAT(F12.1, F13.8, 2I5, 3F13.6, E30.18, F13.6)
2667: RETURN
2668: END
2669:C*****
2670:C*****

```

```

2671:C*****
2672:C*****
2673:    FUNCTION FMOXS(X)
2674:    IMPLICIT REAL*8 (A-H, O-Z)
2675:    COMMON /G01/G1S, G2S, G1L, G2L
2676:    COMMON/TZ/TK, XZ
2677:    RR=8. 313608
2678:    RT=RR*TK
2679:    W=0. 0
2680:    EX=1. 0D-15
2681:    IF (X. LT. 0. 0 ) X=0. 0
2682:    IF (X. GT. 1. 0 ) X=1. 0
2683:    FMOXS=X*(G1S+RT*DLOG(X+EX))
2684:    1 +(1. 0-X)*(G2S+RT*DLOG(1. 0-X+EX))
2685:    2 +X*(1. 0-X)*W
2686:    RETURN
2687:    END
2688:C*****
2689:C*****
2690:C*****
2691:    FUNCTION FMOXL(X)
2692:    IMPLICIT REAL*8 (A-H, O-Z)
2693:    COMMON /G01/G1S, G2S, G1L, G2L
2694:    COMMON/TZ/TK, XZ
2695:    RR=8. 313608
2696:    RT=RR*TK
2697:    W=0. 0
2698:    EX=1. D-15
2699:    IF (X. LT. 0. 0 ) X=0. 0
2700:    IF (X. GT. 1. 0 ) X=1. 0
2701:    FMOXL=X*(G1L+RT*DLOG(X+EX))
2702:    1 +(1. 0-X)*(G2L+RT*DLOG(1. 0-X+EX))
2703:    2 +X*(1. 0-X)*W
2704:    RETURN
2705:    END
2706:C*****
2707:C*****
2708:C*****
2709:C*****
2710:C*****
2711:C*****
2712:    FUNCTION FGO1(AP, X)
2713:    IMPLICIT REAL*8 (A-H, O-Z)
2714:    DIMENSION X(2)
2715:    COMMON /TZ/ TK, XZ
2716:    EX=1. 0D-30
2717:    FGO1=AP*FMOXS(X(1))
2718:    2 +(1. 0-AP)*FMOXL(X(2))
2719:    RETURN
2720:    END
2721:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2722:C*****
2723:C*****
2724:C*****
2725:C*****
2726:    SUBROUTINE SSON(FUNC, XMAX, XMIN, X, XZ, AP, FMIN, ITR, JPH, LLW)
2727:    CCC    SEARCH OF MIN. VALUE OF "FUNC" , PARAMETER X(2)
2728:    IMPLICIT REAL*8 (A-H, O-Z)
2729:    DIMENSION D(2), X(2), XX(2)
2730:    DIMENSION FLL(100)
2731:    C    COMMON/QAXY/QAXY
2732:    CCCC   CONVERGENCE CRITERIA  ETC
2733:    NR=20
2734:    NAP=11
2735:    PAI=3. 141592653589793
2736:    DRAD=2 0*PAI/DFLOAT(NR)
2737:    DRAP=PAI/DFLOAT(NAP-1)
2738:    EPS=1. 0D-10
2739:    EPH=1. 0D-13
2740:    EPSC=1. 0D-7
2741:    EPXX=1. 0D-30
2742:    EPG=1. 0D-15
2743:    ITRMAX=2000
2744:    QAP=0. 3141592
2745:    QAXY=10. 0
2746:    CCD=0. 9
2747:    FACT=0. 1
2748:    FACT2=0. 7
2749:    D(1)=0. 01
2750:    D(2)=D(1)*QAXY
2751:    KDC=0
2752:    LAPX=0
2753:    ITR=0
2754:    MMAX=7
2755:    LMAX=200
2756:    LAVE=5
2757:    CAP=1. 0D-15
2758:    CCCCCCCCCCCCCCCCCC
2759:    IOUT=100000

```



```

2760:CCC  SETTING OF INITIAL CONDITION
2761:      AP=0.5
2762:      X(1)=XZ
2763:      X(2)=X(1)+EPS
2764:      IF (XZ. LE. XMIN) XZ=XMIN
2765:      IF (XZ. GE. XMAX) XZ=XMAX
2766:CCCC
2767:CCCC  ITERATION START
2768:CCCC
2769:      IF (AP. GT. 0.4) THEN
2770:      KKAP=1
2771:      ELSE
2772:      KKAP=2
2773:      END IF
2774: 3000 CONTINUE
2775:      IF (LAPX. GT. LMAX) GO TO 800
2776:      ITR=ITR+1
2777:      LAPX=LAPX+1
2778:CCCCC
2779:CCCCC  X(1) & AP LOOP
2780:CCCCC
2781: 310 CONTINUE
2782:      CALL AX1 (XZ, AP, X, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
2783:      FFX1=FUNC (AP, X)
2784:      JJJ=101
2785:CCCCC  DEBUG OUT
2786:      IF (LLW. GE. 2) THEN
2787:      WRITE (6, 500) JJJ, ITR, LAPX, MTR, LRR, KXP, KKAP, KQ, D(1), X(1),
2788:      1 X(2), AP, FFX1
2789:      ELSE
2790:      END IF
2791:CCCCC
2792:      IF (ITR. GT. 1) THEN
2793:      MMAP=0
2794: 601 CONTINUE
2795:      MMAP=MMAP+1
2796:      LLLP=0
2797:      XX(1)=X(1)
2798:      XX(2)=X(2)
2799:      DO 650 J=1, NAP
2800:      DDD=D(2)*QAP
2801:      RA=DRAP*DFLOAT (J-1)
2802:      APX=AP+DDD*DCOS (RA)
2803:      CALL AX1 (XZ, APX, XX, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
2804:      FFR=FUNC (APX, XX)
2805:      IF (ITR. EQ. 100) THEN
2806:      WRITE (6, 810) J, KQ, D(1), D(2), APX, XX(1), XX(2), FFR
2807:      ELSE
2808:      END IF
2809:      IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
2810:      LLLP=J
2811:      FFX1=FFR
2812:      AP=APX
2813:      X(1)=XX(1)
2814:      X(2)=XX(2)
2815:      ELSE
2816:      END IF
2817: 650 CONTINUE
2818:      JJJ=111
2819:CCCCC  DEBUG OUT
2820:      IF (LLW. GE. 2) THEN
2821:      WRITE (6, 500) JJJ, ITR, LAPX, MTR, LLLP, KXP, KKAP, KQ, D(1), X(1),
2822:      1 X(2), AP, FFX1
2823:      ELSE
2824:      END IF
2825:      IF (MMAP. GT. MMAX) GO TO 602
2826:      IF (LLL. EQ. 1. OR. LLLP. EQ. NAP) GO TO 601
2827:CCCCCCCCCCCCCCCCCCCC
2828:CCCCCCCCCCCCCCCCCCCC
2829: 602 CONTINUE
2830:      LLLX=0
2831:      APX=AP
2832:      DO 660 J=1, NAP
2833:      DDD=D(1)*QAP
2834:      RA=DRAP*DFLOAT (J-1)
2835:      XX(1)=X(1)+DDD*DCOS (RA)
2836:      XX(2)=X(2)+DDD*DCOS (RA)
2837:      CALL AX1 (XZ, APX, XX, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
2838:      FFR=FUNC (APX, XX)
2839:      IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
2840:      LLLX=J
2841:      FFX1=FFR
2842:      AP=APX
2843:      X(1)=XX(1)
2844:      X(2)=XX(2)
2845:      ELSE
2846:      END IF
2847: 660 CONTINUE
2848:      JJJ=112

```

```

2849:CCCCC  DEBUG  OUT
2850:      IF (LLW. GE. 2) THEN
2851:      WRITE (6, 500) JJJ, ITR, LAPX, MTR, LRR, KXP, KKAP, KQ, D (1) , X (1) ,
2852:      1 X (2) , AP, FFX1
2853:      ELSE
2854:      END IF
2855:      ELSE
2856:      END IF
2857:CCCCCCCCCCCC
2858:      LRR=0
2859:      DO 330 JK=1, 2
2860:      DO 300 J=1, NR
2861:      RAD=DRAD*DFLOAT (J)
2862:      XX (1)=X (1) +D (1) *DSIN (RAD)
2863:      XX (2)=X (2) +D (1) *DSIN (RAD)
2864:      APX=AP+D (2) *DCOS (RAD)
2865:      CALL AX1 (XZ, APX, XX, D, XMAX, XMIN, EPXX, CAP, JK, KQ)
2866:      FFR=FUNC (APX, XX)
2867:C      IF (ITR. GE. 2) THEN
2868:C      WRITE (6, 810) J, KQ, APX, XX (1) , XX (2) , XZC, QXZ, FFR
2869:C      ELSE
2870:C      END IF
2871:C      DFGG=EPG*DABS (FFX1)
2872:C      FGG=FFX1-DFGG
2873:C      IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
2874:      KKAP=JK
2875:      LRR=J
2876:      FFX1=FFR
2877:      ELSE
2878:      END IF
2879: 300 CONTINUE
2880: 330 CONTINUE
2881:CCCCC
2882:      IF (LRR. NE. 0) THEN
2883:      KXP=0
2884:CCCCC
2885:      MTR=0
2886:      RAD=DRAD*DFLOAT (LRR)
2887:      X (1)=X (1) +D (1) *DSIN (RAD)
2888:      X (2)=X (2) +D (1) *DSIN (RAD)
2889:      AP=AP+D (2) *DCOS (RAD)
2890:      CALL AX1 (XZ, AP, X, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
2891:      FFX1=FUNC (AP, X)
2892:      JJJ=102
2893:CCCCC  DEBUG  OUT
2894:      IF (LLW. GE. 2) THEN
2895:      WRITE (6, 500) JJJ, ITR, LAPX, MTR, LRR, KXP, KKAP, KQ, D (1) , X (1) , X (2)
2896:      1 , AP, FFX1
2897:      ELSE
2898:      END IF
2899:CCCCC
2900:CCCCCCCCCCCC
2901: 303 CONTINUE
2902:      MTR=MTR+1
2903:      XX (1)=X (1) +D (1) *DSIN (RAD) *QAP
2904:      XX (2)=X (2) +D (1) *DSIN (RAD) *QAP
2905:      APX=AP+D (2) *DCOS (RAD) *QAP
2906:      CALL AX1 (XZ, APX, XX, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
2907:      FFR=FUNC (APX, XX)
2908:      IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
2909:      FFX1=FFR
2910:      X (1)=XX (1)
2911:      X (2)=XX (2)
2912:      AP=APX
2913:CCCCC  DEBUG  OUT
2914:      JJJ=103
2915:      IF (LLW. GE. 2) THEN
2916:      WRITE (6, 500) JJJ, ITR, LAPX, MTR, LRR, KXP, KKAP, KQ, D (1) , X (1)
2917:      1 , X (2) , AP, FFX1
2918:      ELSE
2919:      END IF
2920:      ELSE
2921:      GO TO 770
2922:      END IF
2923:CCCCCCCCCCCCCCCC
2924:CCCCCCCCCCCCCCCC
2925:      IF (MTR. GT. MMAX) THEN
2926:      GO TO 3000
2927:      ELSE
2928:      GO TO 303
2929:      END IF
2930:      ELSE
2931:      KXP=1
2932:      END IF
2933:      GO TO 800
2934: 770 CONTINUE
2935:      FLL (LAPX)=FFX1
2936: 800 CONTINUE
2937:CCCCC

```

```

2938:CCCC CONVERGENCE CHECK
2939:CCCC
2940: DMX=DMAX1(D(1),D(2))
2941: IF(DMX.LT.EPS) GO TO 2000
2942:CCCCCCC
2943:CCCCC REDUCTION OF D
2944:CCCCCCC
2945: IF(KXP.EQ.1) THEN
2946: LAPX=0
2947: KDC=KDC+1
2948: FDC=CCD/DFLOAT(KDC)
2949: IF(FDC.LT.0.1) FDC=0.1
2950: D(1)=D(1)*FDC
2951: D(2)=D(2)*FDC
2952: IF(LLW.GE.1) THEN
2953: WRITE(6,*)' REDUCTION OF D *****AAAA*****'
2954: ELSE
2955: END IF
2956:CCCCCCCCCCCCCCC
2957: ELSE
2958: IF(LAPX.GE.LAVE) THEN
2959: FLSUM=0.0
2960: DO 700 K=1,LAVE
2961: FLSUM=FLSUM+FLL(LAPX-K+1)
2962: 700 CONTINUE
2963: FLAVE=FLSUM/DFLOAT(LAVE)
2964: HENSA=DABS(FLAVE-FLL(LAPX))
2965: IF(HENSA.LE.EPH) THEN
2966: D(1)=D(1)*FACT
2967: D(2)=D(2)*FACT
2968: IF(LLW.GE.1) THEN
2969: WRITE(6,*)' REDUCTION OF D *****CCCC*****'
2970: ELSE
2971: END IF
2972: LAPX=0
2973: KDC=KDC+1
2974:CCCCC
2975: ELSE
2976: IF(LAPX.GE.LMAX) THEN
2977: LAPX=0
2978: KDC=KDC+1
2979: D(1)=D(1)*FACT2
2980: D(2)=D(2)*FACT2
2981: IF(LLW.GE.1) THEN
2982: WRITE(6,*)' REDUCTION OF D *****BBBB*****'
2983: ELSE
2984: END IF
2985: ELSE
2986: END IF
2987: END IF
2988: ELSE
2989: END IF
2990: END IF
2991:CCCCC
2992:CCCCC
2993: IF(ITR.GT.ITRMAX) GO TO 2900
2994: GO TO 3000
2995: 2000 CONTINUE
2996:CCCCC CHECK OF PHASE STATE*****NEAR 2PHASE BOUNDARY
2997: IF(AP.LT.EPSC) THEN
2998:CCCCC NO.2 SINGLE PHASE
2999: JPH=-2
3000: X(2)=XZ
3001: X(1)=0.0
3002: AP=0.0
3003: FMIN=FUNC(AP,X)
3004: ELSE
3005: APC=DABS(1.0-AP)
3006: IF(APC.LT.EPSC) THEN
3007:CCCCC NO.1 SINGLE PHASE
3008: JPH=-1
3009: X(1)=XZ
3010: X(2)=0.0
3011: AP=1.0
3012: FMIN=FUNC(AP,X)
3013: ELSE
3014:CCCCC 2 PHASE
3015: JPH=2
3016: X(2)=(XZ-AP*X(1))/(1.0-AP+EPXX)
3017: FMIN=FUNC(AP,X)
3018: END IF
3019: END IF
3020: GO TO 2700
3021:CCCCCCCCC CALCURATION DOSE NOT CONVERGED
3022: 2900 CONTINUE
3023: JPH=2900
3024: 2700 CONTINUE
3025: 500 FORMAT(8I5,E13.5,4E30.20)
3026: 810 FORMAT(2I5,6E30.20)

```

```

3027: RETURN
3028: END
3029: C*****
3030: CCCCCCCCCCCCCCCCCC
3031: SUBROUTINE AX1 (XZ, AP, X, D, XMAX, XMIN, EPXX, CAP, KKAP, KQ)
3032: IMPLICIT REAL*8 (A-H, O-Z)
3033: DIMENSION X(2), D(2), X1(2)
3034: EPS=1.0D-15
3035: IF (AP. GE. 1.0) AP=1.0-CAP*D(2)
3036: IF (AP. LE. 0.0) AP=CAP*D(2)
3037: CCCCCC
3038: IF (KKAP. EQ. 1) THEN
3039: IF (X(1). GT. XMAX) X(1)=XMAX
3040: IF (X(1). LT. XMIN) X(1)=XMIN
3041: X(2)=(XZ-AP*X(1))/(1.0-AP+EPXX)
3042: IF (X(2). GT. XMAX) X(2)=XMAX
3043: IF (X(2). LT. XMIN) X(2)=XMIN
3044: C AP=(XZ-X(2))/(X(1)-X(2))
3045: C IF (AP. GE. 1.0) AP=1.0-CAP*D(2)
3046: C IF (AP. LE. 0.0) AP=CAP*D(2)
3047: ELSE
3048: IF (X(2). GT. XMAX) X(2)=XMAX
3049: IF (X(2). LT. XMIN) X(2)=XMIN
3050: X(1)=(XZ-(1.0-AP)*X(2))/(AP+EPXX)
3051: IF (X(1). GT. XMAX) X(1)=XMAX
3052: IF (X(1). LT. XMIN) X(1)=XMIN
3053: C AP=(XZ-X(2))/(X(1)-X(2))
3054: C IF (AP. GE. 1.0) AP=1.0-CAP*D(2)
3055: C IF (AP. LE. 0.0) AP=CAP*D(2)
3056: END IF
3057: XZC=AP*X(1)+(1.0-AP)*X(2)
3058: QXZ=DABS((XZ-XZC)/(XZ+1.0))
3059: KQ=0
3060: IF (QXZ. GT. EPS) KQ=1
3061: RETURN
3062: END
3063: CCCCCCCCCCCCCC
3064: C*****
3065: SUBROUTINE SSCN3 (FUNC, XMAX, XMIN, YMAX, YMIN, SMAX, X, Y, XZ, YZ, AP, FMIN,
3066: 1 ITR, JPH, LLW)
3067: CCC SEARCH OF MIN. VALUE OF "FUNC", PARAMETER X(2)
3068: IMPLICIT REAL*8 (A-H, O-Z)
3069: EXTERNAL FUNC
3070: C COMMON/QAXY/QAXY
3071: DIMENSION D(2), X(2), XX(2), Y(2), YY(2)
3072: DIMENSION FLL(10000)
3073: DIMENSION BOX(2), BPU(2)
3074: CCCC CONVERGENCE CRITERIA ETC
3075: NR=20
3076: MR=11
3077: NAP=11
3078: PA1=3.141592653589793
3079: DRAD=2.0*PA1/DFLOAT(NR)
3080: DRADA=PA1/DFLOAT(MR-1)
3081: DRAP=PA1/DFLOAT(NAP-1)
3082: QXY=1.0
3083: QAP=1.0
3084: QAXY=10.0
3085: QH=1.0/3.0
3086: EPS=1.0D-10
3087: EPSC=1.0D-7
3088: EPH=1.0D-13
3089: EPXX=1.0D-30
3090: ITRMAX=2000
3091: ITOUT=1000000
3092: CGD=0.9
3093: FACT=0.1
3094: FACT2=0.7
3095: D(1)=0.002
3096: D(2)=D(1)*QAXY
3097: KDC=0
3098: LAPX=0
3099: ITR=0
3100: MMAX=7
3101: LMAX=200
3102: LAVE=5
3103: CAP=1.0D-30
3104: CCC SETTING OF INITIAL CONDITION
3105: AP=0.5
3106: X(1)=XZ
3107: X(2)=X(1)
3108: Y(1)=YZ
3109: Y(2)=Y(1)
3110: IF (XZ. LE. XMIN) XZ=XMIN
3111: IF (XZ. GE. XMAX) XZ=XMAX
3112: IF (YZ. LE. YMIN) YZ=YMIN
3113: IF (YZ. GE. YMAX) YZ=YMAX
3114: IF (AP. GT. 0.4) THEN
3115: KKAP=1

```

```

3116: ELSE
3117: KKAP=2
3118: END IF
3119: CCCC
3120: CCCC ITERATION START
3121: CCCC
3122: 3000 CONTINUE
3123: IF (LAPX. GT. LMAX) GO TO 800
3124: ITR=ITR+1
3125: LAPX=LAPX+1
3126: CCCCC
3127: CCCCC X(1) & AP LOOP
3128: CCCCC
3129: 310 CONTINUE
3130: CALL AX3(XZ, YZ, AP, X, Y, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3131: 1 EPXX, CAP, KKAP, KQ)
3132: FFX1=FUNC(AP, X, Y)
3133: CCC
3134: CCC DEBUG OUT
3135: LLL=101
3136: IF (LLW. GE. 3) THEN
3137: CALL OXTR(X, Y, BOX, BPU)
3138: WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), BOX(1)
3139: 1 , BOX(2), BPU(1), BPU(2), AP, FFX1
3140: ELSE
3141: IF (LLW. EQ. 2) THEN
3142: WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), X(1)
3143: 1 , X(2), Y(1), Y(2), AP, FFX1
3144: ELSE
3145: FND IF
3146: END IF
3147: CCCCCCCCCCCCCCCCCC
3148: G GO TO 777
3149: IF (ITR. GT. 1) THEN
3150: MMAP=0
3151: 601 CONTINUE
3152: MMAP=MMAP+1
3153: LLLP=0
3154: XX(1)=X(1)
3155: YY(1)=Y(1)
3156: XX(2)=X(2)
3157: YY(2)=Y(2)
3158: DO 650 J=1, NAP
3159: DDD=D(2)*QAP
3160: RA=DRAP*DFLOAT(J-1)
3161: APX=AP+DDD*DCOS(RA)
3162: CALL AX3(XZ, YZ, APX, XX, YY, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3163: 1 EPXX, CAP, KKAP, KQ)
3164: FFR=FUNC(APX, XX, YY)
3165: IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
3166: LLLP=J
3167: FFX1=FFR
3168: AP=APX
3169: X(1)=XX(1)
3170: Y(1)=YY(1)
3171: X(2)=XX(2)
3172: Y(2)=YY(2)
3173: ELSE
3174: END IF
3175: 650 CONTINUE
3176: LLL=111
3177: IF (LLW. GE. 3) THEN
3178: CALL OXTR(X, Y, BOX, BPU)
3179: WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), BOX(1)
3180: 1 , BOX(2), BPU(1), BPU(2), AP, FFX1
3181: ELSE
3182: IF (LLW. EQ. 2) THEN
3183: WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), X(1)
3184: 1 , X(2), Y(1), Y(2), AP, FFX1
3185: ELSE
3186: END IF
3187: END IF
3188: IF (MMAP. GT. MMAX) GO TO 602
3189: IF (LLLP. EQ. 1. OR. LLLP. EQ. NAP) GO TO 601
3190: CCCCCCCCCCCCCCCCCC
3191: 602 CONTINUE
3192: LLLX=0
3193: YY(1)=Y(1)
3194: YY(2)=Y(2)
3195: APX=AP
3196: DO 660 J=1, NAP
3197: DDD=D(1)*QXY
3198: RX=DRAP*DFLOAT(J-1)
3199: XX(1)=X(1)+DDD*DCOS(RX)
3200: XX(2)=X(2)+DDD*DCOS(RX)
3201: CALL AX3(XZ, YZ, APX, XX, YY, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3202: 1 EPXX, CAP, KKAP, KQ)
3203: FFR=FUNC(APX, XX, YY)
3204: IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN

```

```

3205:      LLLX=J
3206:      FFX1=FFR
3207:      AP=APX
3208:      X(1)=XX(1)
3209:      Y(1)=YY(1)
3210:      X(2)=XX(2)
3211:      Y(2)=YY(2)
3212:      ELSE
3213:      END IF
3214: 660 CONTINUE
3215:      LLL=112
3216:      IF(LLW. GE. 3) THEN
3217:          CALL OXTR(X, Y, BOX, BPU)
3218:          WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), BOX(1)
3219:          1, BOX(2), BPU(1), BPU(2), AP, FFX1
3220:      ELSE
3221:      IF(LLW. EQ. 2) THEN
3222:          WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), X(1)
3223:          1, X(2), Y(1), Y(2), AP, FFX1
3224:      ELSE
3225:      END IF
3226:      END IF
3227: C      IF(LLLX. EQ. 1. OR. LLLX. EQ. NAP) GO TO 602
3228: CCCCCCCCCCCCCCCCCC
3229: 603 CONTINUE
3230:      LLLY=0
3231:      XX(1)=X(1)
3232:      XX(2)=X(2)
3233:      APX=AP
3234:      DO 670 J=1, NAP
3235:          DDD=D(1)*QXY
3236:          RX=DRAP*DFLOAT(J-1)
3237:          YY(1)=Y(1)+DDD*DCOS(RX)
3238:          YY(2)=Y(2)+DDD*DCOS(RX)
3239:          CALL AX3(XZ, YZ, APX, XX, YY, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3240:          1 EPXX, CAP, KKAP, KQ)
3241:          FFR=FUNC(APX, XX, YY)
3242:          IF(FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
3243:              LLLY=J
3244:              FFX1=FFR
3245:              AP=APX
3246:              X(1)=XX(1)
3247:              Y(1)=YY(1)
3248:              X(2)=XX(2)
3249:              Y(2)=YY(2)
3250:              ELSE
3251:              END IF
3252: 670 CONTINUE
3253:      LLL=113
3254:      IF(LLW. GE. 3) THEN
3255:          CALL OXTR(X, Y, BOX, BPU)
3256:          WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), BOX(1)
3257:          1, BOX(2), BPU(1), BPU(2), AP, FFX1
3258:      ELSE
3259:      IF(LLW. EQ. 2) THEN
3260:          WRITE(6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D(1), X(1)
3261:          1, X(2), Y(1), Y(2), AP, FFX1
3262:      ELSE
3263:      END IF
3264:      END IF
3265: C      IF(LLLLY. EQ. 1. OR. LLLLLY. EQ. NAP) GO TO 603
3266:      ELSE
3267:      ENDIF
3268: CCCCCCCCCCCCCC
3269: 777 CONTINUE
3270:      LRR=0
3271:      KRR=0
3272:      DO 450 JK=1, 2
3273:          DO 400 JJ=1, MR
3274:              RADA=DRADA*DFLOAT(JJ-1)
3275:              APX=AP+D(2)*DCOS(RADA)
3276:              DO 300 J=1, NR
3277:                  RAD=DRAD*DFLOAT(J)
3278:                  XX(1)=X(1)+D(1)*DSIN(RAD)
3279:                  YY(1)=Y(1)+D(1)*DCOS(RAD)
3280:                  XX(2)=X(2)+D(1)*DSIN(RAD)
3281:                  YY(2)=Y(2)+D(1)*DCOS(RAD)
3282:                  CALL AX3(XZ, YZ, APX, XX, YY, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3283:                  1 EPXX, CAP, JK, KQ)
3284:                  FFR=FUNC(APX, XX, YY)
3285:                  IF(FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
3286:                      KKAP=JK
3287:                      LRR=J
3288:                      KRR=JJ-1
3289:                      FFX1=FFR
3290:                      ELSE
3291:                      END IF
3292: 300 CONTINUE
3293: 400 CONTINUE

```

```

3294: 450 CONTINUE
3295: CCCCC
3296: IF (LRR. NE. 0) THEN
3297: KXP=0
3298: CCCCC
3299: CCCCC
3300: CCCCC
3301: MTR=0
3302: RAD=DRAD*DFLOAT (LRR)
3303: X (1)=X (1)+D (1)*DSIN (RAD)
3304: Y (1)=Y (1)+D (1)*DCOS (RAD)
3305: X (2)=X (2)+D (1)*DSIN (RAD)
3306: Y (2)=Y (2)+D (1)*DCOS (RAD)
3307: RADA=DRADA*DFLOAT (KRR)
3308: AP=AP+D (2)*DCOS (RADA)
3309: CALL AX3 (XZ, YZ, AP, X, Y, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3310: 1 EPXX, CAP, KKAP, KQ)
3311: FFX1=FUNC (AP, X, Y)
3312: CCC DEBUG OUT
3313: LLL=102
3314: IF (LLW. GE. 3) THEN
3315: CALL OXTR (X, Y, BOX, BPU)
3316: WRITE (6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D (1), BOX (1)
3317: 1, BOX (2), BPU (1), BPU (2), AP, FFX1
3318: ELSE
3319: IF (LLW. EQ. 2) THEN
3320: WRITE (6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D (1), X (1)
3321: 1, X (2), Y (1), Y (2), AP, FFX1
3322: ELSE
3323: END IF
3324: END IF
3325: CCCCCCCCCC
3326: 303 CONTINUE
3327: MTR=MTR+1
3328: XX (1)=X (1)+D (1)*DSIN (RAD)*QH
3329: YY (1)=Y (1)+D (1)*DCOS (RAD)*QH
3330: XX (2)=X (2)+D (1)*DSIN (RAD)*QH
3331: YY (2)=Y (2)+D (1)*DCOS (RAD)*QH
3332: APX=AP+D (2)*DCOS (RADA)*QH
3333: CALL AX3 (XZ, YZ, APX, XX, YY, D, XMAX, XMIN, YMAX, YMIN, SMAX,
3334: 1 EPXX, CAP, KKAP, KQ)
3335: FFR=FUNC (APX, XX, YY)
3336: IF (FFR. LT. FFX1. AND. KQ. EQ. 0) THEN
3337: FFX1=FFR
3338: AP=APX
3339: X (1)=XX (1)
3340: Y (1)=YY (1)
3341: X (2)=XX (2)
3342: Y (2)=YY (2)
3343: CCCCCCCC
3344: CCC DEBUG OUT
3345: LLL=103
3346: IF (LLW. GE. 3) THEN
3347: CALL OXTR (X, Y, BOX, BPU)
3348: WRITE (6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D (1), BOX (1)
3349: 1, BOX (2), BPU (1), BPU (2), AP, FFX1
3350: ELSE
3351: IF (LLW. EQ. 2) THEN
3352: WRITE (6, 500) LLL, ITR, LAPX, MTR, LRR, KRR, KXP, KKAP, KQ, D (1), X (1)
3353: 1, X (2), Y (1), Y (2), AP, FFX1
3354: ELSE
3355: END IF
3356: END IF
3357: CCCCCCCC
3358: ELSE
3359: GO TO 770
3360: END IF
3361: CCCCCCCCCCCCCC
3362: CC
3363: CCCCCCCCCCCCCC
3364: IF (MTR. GT. MMAX) THEN
3365: GO TO 3000
3366: ELSE
3367: GO TO 303
3368: END IF
3369: ELSE
3370: KXP=1
3371: END IF
3372: GO TO 800
3373: 770 CONTINUE
3374: FLL (LAPX)=FFX1
3375: 800 CONTINUE
3376: CCCCC
3377: CCCCC CONVERGENCE CHECK
3378: CCCCC
3379: DMX=DMAX1 (D (1), D (2))
3380: IF (DMX. LT. EPS) GO TO 2000
3381: CCCCCC
3382: CCCCC REDUCTION OF D

```

```

3383: CCCCCC
3384:   IF (KXP. EQ. 1) THEN
3385:     LAPX=0
3386:     KDC=KDC+1
3387:     FDC=CCD/DFLOAT (KDC)
3388:     IF (FDC. LT. 0. 1) FDC=0. 1
3389: c   FDC=0. 8
3390:   D (1)=D (1)*FDC
3391:   D (2)=D (2)*FDC
3392:   IF (LLW. GE. 1) THEN
3393:     WRITE (6, *) ' REDUCTION OF D          *****AAAA***** '
3394:   ELSE
3395:     END IF
3396: CCCCCCCCCCCCCC
3397:   ELSE
3398:     IF (LAPX. GE. LAVE) THEN
3399:       FLSUM=0. 0
3400:       DO 700 K=1, LAVE
3401:         FLSUM=FLSUM+FLL (LAPX-K+1)
3402:       700 CONTINUE
3403:       FLAVE=FLSUM/DFLOAT (LAVE)
3404:       HENSA=DABS (FLAVE-FLL (LAVE))
3405:       IF (HENSA. LE. EPH) THEN
3406:         D (1)=D (1)*FACT
3407:         D (2)=D (2)*FACT
3408:         IF (LLW. GE. 1) THEN
3409:           WRITE (6, *) ' REDUCTION OF D          *****CCCC***** '
3410:         ELSE
3411:           END IF
3412:         LAPX=0
3413:         KDC=KDC+1
3414: CCCCC
3415:       ELSE
3416:         IF (LAPX. GE. LMAX) THEN
3417:           LAPX=0
3418:           KDC=KDC+1
3419:           D (1)=D (1)*FACT2
3420:           D (2)=D (2)*FACT2
3421:           IF (LLW. GE. 1) THEN
3422:             WRITE (6, *) ' REDUCTION OF D          *****BBBB***** '
3423:           ELSE
3424:             END IF
3425:           ELSE
3426:             END IF
3427:           END IF
3428:         ELSE
3429:           END IF
3430:         END IF
3431: CCCCC
3432: CCCCC
3433:   IF (ITR. GT. ITRMAX) GO TO 2900
3434:   GO TO 3000
3435: 2000 CONTINUE
3436: CCCCC CHECK OF PHASE STATE*****NEAR 2PHASE BOUNDARY
3437:   IF (AP. LT. EPSC) THEN
3438: CCCCC NO. 2 SINGLE PHASE
3439:     JPH=-2
3440:     X (2)=XZ
3441:     Y (2)=YZ
3442:     X (1)=0. 0
3443:     Y (1)=0. 0
3444:     AP=0. 0
3445:     FMIN=FUNC (AP, X, Y)
3446:   ELSE
3447:     APC=DABS (1. 0-AP)
3448:     IF (APC. LT. EPSC) THEN
3449: CCCCC NO. 1 SINGLE PHASE
3450:       JPH=-1
3451:       X (1)=XZ
3452:       X (2)=0. 0
3453:       Y (1)=YZ
3454:       Y (2)=0. 0
3455:       AP=1. 0
3456:       FMIN=FUNC (AP, X, Y)
3457:     ELSE
3458: CCCCC 2 PHASE
3459:       JPH=2
3460:       X (2)=(XZ-AP*X (1))/(1. 0-AP+EPXX)
3461:       Y (2)=(YZ-AP*Y (1))/(1. 0-AP+EPXX)
3462:       FMIN=FUNC (AP, X, Y)
3463:     END IF
3464:   END IF
3465:   GO TO 2700
3466: CCCCCCCC CALCURATION DOSE NOT CONVERGED
3467: 2900 CONTINUE
3468:   JPH=2900
3469: 2700 CONTINUE
3470: 500 FORMAT (9I4, E13. 5, 5E17. 9, E24. 15)
3471: 820 FORMAT (2I5, 5E17. 9, 2E24. 15)

```



```

3472: 830 FORMAT(215,5E17,9,E24,15)
3473: RETURN
3474: END
3475: C*****
3476: SUBROUTINE OXTR(X,Y,OM,XPU)
3477: IMPLICIT REAL*8(A-H,O-Z)
3478: DIMENSION X(2),Y(2),OM(2),XPU(2)
3479: EPS=1.0D-30
3480: DO 100 J=1,2
3481: OM(J)=X(J)/(1.0-X(J)+EPS)
3482: XPU(J)=Y(J)/(1.0-X(J)+EPS)
3483: 100 CONTINUE
3484: RETURN
3485: END
3486: CCCCC*****
3487: SUBROUTINE LAG(N,X,Y,XX,A)
3488: IMPLICIT REAL*8(A-H,O-Z)
3489: DIMENSION X(4),Y(4),B(4)
3490: CCCCC LAGRANGE INTERPOLATION (N-1) ORDER POLYNOMIAL
3491: A=0.0
3492: DO 25 K=1,N
3493: B(K)=1.0
3494: DO 20 J=1,N
3495: IF(J.NE.K) THEN
3496: B(K)=B(K)*(XX-X(J))/(X(K)-X(J))
3497: ELSE
3498: END IF
3499: 20 CONTINUE
3500: A=A+B(K)*Y(K)
3501: 25 CONTINUE
3502: RETURN
3503: END
3504: CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
3505: SUBROUTINE AX3(XZ,YZ,AP,X,Y,D,XMAX,XMIN,YMAX,YMIN,SMAX,
3506: 1 FPXX,CAP,KKAP,KQ)
3507: IMPLICIT REAL*8(A-H,O-Z)
3508: DIMENSION X(2),D(2),Y(2)
3509: CCCC
3510: CCCC ORIGINAL SET OF X(1),Y(1) & AP
3511: CCCC
3512: C MD=110
3513: C WRITE(6,*)MD
3514: EPS=1.0D-15
3515: CCA=0.0
3516: IF(AP.GE.1.0) AP=1.0-CAP*D(2)
3517: IF(AP.LE.0.0) AP=CAP*D(2)
3518: CCCCCCCCCC
3519: CCCCCC KKAP=1 CASE
3520: IF(KKAP.EQ.1) THEN
3521: IF(X(1).GT.XMAX) X(1)=XMAX-CCA*D(1)
3522: IF(X(1).LT.XMIN) X(1)=XMIN+CCA*D(1)
3523: IF(Y(1).GT.YMAX) Y(1)=YMAX-CCA*D(1)
3524: IF(Y(1).LT.YMIN) Y(1)=YMIN+CCA*D(1)
3525: SXY=X(1)+Y(1)
3526: IF(SXY.GT.SMAX) THEN
3527: C KQ=1
3528: C RETURN
3529: X(1)=X(1)/SXY
3530: Y(1)=Y(1)/SXY
3531: ELSE
3532: END IF
3533: X(2)=(XZ-AP*X(1))/(1.0-AP+EPXX)
3534: Y(2)=(YZ-AP*Y(1))/(1.0-AP+EPXX)
3535: IF(X(2).GT.XMAX) X(2)=XMAX-CCA*D(1)
3536: IF(X(2).LT.XMIN) X(2)=XMIN+CCA*D(1)
3537: IF(Y(2).GT.YMAX) Y(2)=YMAX-CCA*D(1)
3538: IF(Y(2).LT.YMIN) Y(2)=YMIN+CCA*D(1)
3539: SXY=X(2)+Y(2)
3540: IF(SXY.GT.SMAX) THEN
3541: C KQ=1
3542: C RETURN
3543: X(2)=X(2)/SXY
3544: Y(2)=Y(2)/SXY
3545: ELSE
3546: END IF
3547: X(1)=(XZ-(1.0-AP)*X(2))/(AP+EPXX)
3548: Y(1)=(YZ-(1.0-AP)*Y(2))/(AP+EPXX)
3549: CCCCCC KKAP=2 CASE
3550: ELSE
3551: IF(X(2).GT.XMAX) X(2)=XMAX-CCA*D(1)
3552: IF(X(2).LT.XMIN) X(2)=XMIN+CCA*D(1)
3553: IF(Y(2).GT.YMAX) Y(2)=YMAX-CCA*D(1)
3554: IF(Y(2).LT.YMIN) Y(2)=YMIN+CCA*D(1)
3555: SXY=X(2)+Y(2)
3556: IF(SXY.GT.SMAX) THEN
3557: C KQ=1
3558: C RETURN
3559: X(2)=X(2)/SXY
3560: Y(2)=Y(2)/SXY

```

```

3561: ELSE
3562: END IF
3563: X(1)=(XZ-(1.0-AP)*X(2))/(AP+EPXX)
3564: Y(1)=(YZ-(1.0-AP)*Y(2))/(AP+EPXX)
3565: IF(X(1).GT.XMAX) X(1)=XMAX-CCA*D(1)
3566: IF(X(1).LT.XMIN) X(1)=XMIN+CCA*D(1)
3567: IF(Y(1).GT.YMAX) Y(1)=YMAX-CCA*D(1)
3568: IF(Y(1).LT.YMIN) Y(1)=YMIN+CCA*D(1)
3569: SXY=X(1)+Y(1)
3570: IF(SXY.GT.SMAX) THEN
3571:c   KQ=1
3572:c   RETURN
3573: X(1)=X(1)/SXY
3574: Y(1)=Y(1)/SXY
3575: ELSE
3576: END IF
3577: X(2)=(XZ-AP*X(1))/(1.0-AP+EPXX)
3578: Y(2)=(YZ-AP*Y(1))/(1.0-AP+EPXX)
3579: END IF
3580: XZC=AP*X(1)+(1.0-AP)*X(2)
3581: YZC=AP*Y(1)+(1.0-AP)*Y(2)
3582: QXZ=DABS((XZ-XZC)/(XZ+1.0))
3583: QYZ=DABS((YZ-YZC)/(YZ+1.0))
3584: KQ=0
3585: IF(QXZ.GT.EPS.OR.QYZ.GT.EPS) KQ=1
3586:C   MD=190
3587:C   WRITE(6,*)MD
3588: RETURN
3589: END
3590:C*****
3591:C*****
3592: SUBROUTINE GGO2(T,G,GF)
3593: IMPLICIT REAL*8 (A-H,O-Z)
3594:CCCC  FREE ENERGY FUNCTION OF O2 (OXYGEN MOLECULE)
3595:CCCC  EVALUATED IHSAN BARIN(VCH WEINHEIM)
3596:CCCC  EFFECTIVE RT=5000K
3597: A7=0.75875808D-20
3598: A6=-0.16347702D-15
3599: A5=0.14838428D-11
3600: A4=-0.74653182D-08
3601: A3=0.23531162D-04
3602: A2=-0.56017358D-01
3603: A1=-0.17882230D03
3604: A0=-0.34057422D04
3605: G=A7*T**7+A6*T**6+A5*T**5+A4*T**4
3606: 1  +A3*T**3+A2*T**2+A1*T+A0
3607: GF=0.0
3608: RETURN
3609: END
3610:C*****
3611:C*****
3612: SUBROUTINE GOXA(T,G,GF)
3613: IMPLICIT REAL*8 (A-H,O-Z)
3614:CCCC  FREE ENERGY FUNCTION OF ATOMIC OXYGEN
3615:CCCC  EVALUATED IHSAN BARIN(VCH WEINHEIM)
3616:CCCC  EFFECTIVE RT=5000K
3617:CCCC  GF—FREE ENERGY OF FORMATION OF ATOMIC OXYGEN/G ATOM
3618: CJ=4.184
3619: HF298=59553.8
3620: A4=3.3404D-15
3621: A3=-1.0072D-10
3622: A2=9.8543D-7
3623: A1=-5.0412D-3
3624: A0=-36.929
3625: FEF=A4*T**4+A3*T**3+A2*T**2+A1*T+A0
3626: G=FEF*T+HF298
3627: G=G+CJ
3628: CALL GGO2(T,G02,GF02)
3629: GF=G-0.5*G02
3630: RETURN
3631: END
3632:C*****
3633:C*****
3634:C*****
3635: SUBROUTINE GUMET(T,G,GF)
3636: IMPLICIT REAL*8 (A-H,O-Z)
3637:CCCC  FREE ENERGY FUNCTION OF U (SOLID & LIQUID)
3638:CCCC  EVALUATED IHSAN BARIN(VCH WEINHEIM)
3639:CCCC  EFFECTIVE RT=4402K(BOILING POINT)
3640: CJ=4.184
3641: A7=4.4043D-25
3642: A6=-2.0913D-20
3643: A5=2.9421D-16
3644: A4=-1.9405D-12
3645: A3=6.7123D-9
3646: A2=-1.1373D-5
3647: A1=2.1223D-3
3648: A0=-11.778
3649: FEF=A7*T**7+A6*T**6+A5*T**5+A4*T**4

```

```

3650: 1      +A3*T**3+A2*T**2+A1*T+A0
3651:      G=FEF*T*CJ
3652:      GF=0.0
3653:      RETURN
3654:      END
3655: C*****
3656: C*****
3657: C*****
3658:      SUBROUTINE GPUMET (T, G, GF)
3659:      IMPLICIT REAL*8 (A-H, O-Z)
3660: CCCC   FREE ENERGY FUNCTION OF PU (SOLID & LIQUID)
3661: CCCC   EVALUATED IHSAN BARIN (VGH WEINHEIM)
3662: CCCC   EFFECTIVE RT~3498K (BOILING POINT)
3663:      CJ=4.184
3664:      A7=2.1892D-23
3665:      A6=-4.6924D-19
3666:      A5=3.8702D-15
3667:      A4=-1.5973D-11
3668:      A3=3.4098D-8
3669:      A2=-2.8631D-5
3670:      A1=-3.1951D-2
3671:      A0=-37.308
3672:      FEF=A7*T**7+A6*T**6+A5*T**5+A4*T**4
3673: 1      +A3*T**3+A2*T**2+A1*T+A0
3674:      G=FEF*T
3675:      GF=0.0
3676:      RETURN
3677:      END
3678: C*****
3679: C*****
3680: C*****
3681: C*****
3682: C*****
3683:      SUBROUTINE GUO2S (T, G, GF)
3684:      IMPLICIT REAL*8 (A-H, O-Z)
3685: CCCC   FREE ENERGY OF UO2 (SOLID)
3686: CCCC   CALCULATION OF HEAT CAPACITY, ENTALPY, ENTROPY,
3687: CCCC   AND GIBBS FREE ENERGY OF UO2 AS AFUNCTION OF
3688: CCCC   TEMPERATURE (CAL/MOLE)
3689: CCCC   RT~2000K   EVALUATED DATA BY CORDFUNKE ET AL
3690: CCCC   ABOVE 2000K HEIN ENTHALPY DATA
3691: CCCC   HFUO2 IS HEAT OF FORMATION OF UO2 (CAL) AT 298.15K
3692: CCCC   BASED ON BARIN (-1084.899 KJ/MOLE)
3693:      HFUO2=-259297.1
3694:      A9=-2.1652E-28
3695:      A8=4.0868E-24
3696:      A7=-3.0604E-20
3697:      A6=1.1877E-16
3698:      A5=-2.5920E-13
3699:      A4=3.1859E-10
3700:      A3=-1.9831E-7
3701:      A2=3.0521E-5
3702:      A1=2.9522E-2
3703:      A0=6.9774
3704:      TS=298.15
3705:      S298=18.4106
3706:      CJ=4.184
3707:      H=A9*TS**10/10.0+A8*TS**9/9.0+A7*TS**8/8.0+A6*TS**7/7.0
3708:      H=H+A5*TS**6/6.0+A4*TS**5/5.0+A3*TS**4/4.0
3709:      H=H+A2*TS**3/3.0+A1*TS**2/2.0+A0*TS
3710:      S=A9*TS**9/9.0+A8*TS**8/8.0+A7*TS**7/7.0+A6*TS**6/6.0
3711:      S=S+A5*TS**5/5.0+A4*TS**4/4.0+A3*TS**3/3.0+A2*TS**2/2.0
3712:      S=S+A1*TS+A0*DLOG (TS)
3713:      HS=H
3714:      SS=S
3715: CCCCCCCCCCCC
3716:      CP=A9*T**9+A8*T**8+A7*T**7+A6*T**6
3717:      CP=CP+A5*T**5+A4*T**4+A3*T**3+A2*T**2+A1*T+A0
3718:      H=A9*T**10/10.0+A8*T**9/9.0+A7*T**8/8.0+A6*T**7/7.0
3719:      H=H+A5*T**6/6.0+A4*T**5/5.0+A3*T**4/4.0
3720:      H=H+A2*T**3/3.0+A1*T**2/2.0+A0*T
3721:      S=A9*TS**9/9.0+A8*TS**8/8.0+A7*TS**7/7.0+A6*TS**6/6.0
3722:      S=S+A5*TS**5/5.0+A4*TS**4/4.0+A3*TS**3/3.0+A2*TS**2/2.0
3723:      S=S+A1*TS+A0*DLOG (T)
3724:      H=H-HS
3725:      S=S-SS
3726:      S=S+S298
3727:      FEF=(H-T*S)/T
3728:      G=FEF*T+HFUO2
3729:      G=G*CJ
3730: CCCC   FREE ENERGY OF FORMATION OF UO2
3731:      CALL GUMET (T, GU, GUF)
3732:      CALL GGO2 (T, GO2, GO2F)
3733:      GF=G-GU-GO2
3734: C      WRITE (6, *) T, H, S, G
3735:      RETURN
3736:      END
3737: C*****
3738: C*****

```

```

3739:C*****
3740: SUBROUTINE GPU02S(T, G, GF)
3741: IMPLICIT REAL*8 (A-H, O-Z)
3742:CCCC FREE ENERGY OF PU02 (SOLID)
3743:CCCC EVALUATED BY CORDFUNKE ET AL
3744:CCCC HF IS HEAT OF FORMATION OF PU02 (J/MOLE) AT 298.15K
3745:CCCC BASED ON BARIN
3746: HF=-1055832.0
3747: A7=0.28198363D-18
3748: A6=-0.35071008D-14
3749: A5=0.18820356D-10
3750: A4=-0.57858952D-7
3751: A3=0.11670506D-03
3752: A2=-0.19070767
3753: A1=0.21069693D2
3754: A0=-0.11959743D5
3755: GG=A7*T**7+A6*T**6+A5*T**5+A4*T**4+A3*T**3
3756: 1 +A2*T**2+A1*T+A0
3757: G=GG+HF
3758:C FREE ENERGY OF FORMATION OF PU02(S)
3759: CALL GPUMET(T, GPU, GPUF)
3760: CALL GGO2(T, GO2, GO2F)
3761: GF=G-GPU-GO2
3762: RETURN
3763: END
3764:C*****
3765:C*****
3766:C*****
3767: SUBROUTINE GPU02L(T, G, GF)
3768: IMPLICIT REAL*8 (A-H, O-Z)
3769:CCCC FREE ENERGY OF PU02 (LIQUID)
3770:CCCC EVALUATED BY CORDFUNKE ET AL
3771: CJ=4.184
3772: HF=-1055832.0/CJ
3773: TM=2663.0
3774: DHF=16000.0
3775:c***** Epstein DHF
3776:c DHF=16800.0
3777: CP=31.31
3778: HSMP=54389.3
3779: SSMP=63.003
3780: H=HSMP+DHF+CP*(T-TM)
3781: S=SSMP+DHF/TM+CP*DLOG(T/TM)
3782: G=H-T*S+HF
3783: G=G*CJ
3784: CALL GPUMET(T, GPU, GPUF)
3785: CALL GGO2(T, GO2, GO2F)
3786: GF=G-GPU-GO2
3787: RETURN
3788: END
3789:C*****
3790:C*****
3791:C*****
3792: SUBROUTINE GU02L(T, G, GF)
3793: IMPLICIT REAL*8 (A-H, O-Z)
3794:CCCC FREE ENERGY OF UO2 (LIQUID)
3795:CCCC BASED ON L. LEIBOWITZ ET AL (J. NUCL. MAT. 39, 115, 1971)
3796: CJ=4.184
3797: HFU02=-259297.1
3798: TM=3133.15
3799: DHF=17700.0
3800:C***** EPSTEIN DHF
3801:c DHF=21200.0
3802: CP=32.457
3803: HSMP=72656.3877
3804: SSMP=69.93273825
3805: H=HSMP+DHF+CP*(T-TM)
3806: S=SSMP+DHF/TM+CP*DLOG(T/TM)
3807: G=H-T*S+HFU02
3808: G=G*CJ
3809: CALL GUMET(T, GU, GUF)
3810: CALL GGO2(T, GO2, GO2F)
3811: GF=G-GU-GO2
3812: RETURN
3813: END
3814:C*****
3815: SUBROUTINE SGAGS(T, G)
3816: IMPLICIT REAL*8 (A-H, O-Z)
3817:CCCC FREE ENERGY OF SOLID AG NEAR MP (NOT STRICT)
3818:CCCC UNIT JOUL & K
3819: TMP=1233.95
3820: CP=30.0
3821: HMP=26412.0
3822: SMP=81.73
3823: H=HMP+CP*(T-TMP)
3824: S=SMP+CP*DLOG(T/TMP)
3825: G=H-T*S
3826: RETURN
3827: END

```

```

3828:C*****
3829:C*****
3830:C*****
3831:      SUBROUTINE SGAGL(T,G)
3832:      IMPLICIT REAL*8 (A-H,O-Z)
3833:CCCC FREE ENERGY OF LIQUID AG NEAR MP (NOT STRICT)
3834:CCCC UNIT JOUL & K
3835:      TMP=1233.95
3836:      CP=33.5
3837:      HMP=37709.0
3838:      SMP=90.885
3839:      H=HMP+CP*(T-TMP)
3840:      S=SMP+CP*DLOG(T/TMP)
3841:      G=H-T*S
3842:      RETURN
3843:      END
3844:C*****
3845:C*****
3846:      SUBROUTINE SGAUS(T,G)
3847:      IMPLICIT REAL*8 (A-H,O-Z)
3848:CCCC FREE ENERGY OF SOLID AU NEAR MP (NOT STRICT)
3849:CCCC UNIT JOUL & K
3850:      TMP=1337.58
3851:      CP=28.0
3852:      HMP=29099.0
3853:      SMP=88.429
3854:      H=HMP+CP*(T-TMP)
3855:      S=SMP+CP*DLOG(T/TMP)
3856:      G=H-T*S
3857:      RETURN
3858:      END
3859:C*****
3860:C*****
3861:      SUBROUTINE SGAUL(T,G)
3862:      IMPLICIT REAL*8 (A-H,O-Z)
3863:CCCC FREE ENERGY OF LIQUID AU NEAR MP (NOT STRICT)
3864:CCCC UNIT JOUL & K
3865:      TMP=1337.58
3866:      CP=30.962
3867:      HMP=41651.0
3868:      SMP=97.813
3869:      H=HMP+CP*(T-TMP)
3870:      S=SMP+CP*DLOG(T/TMP)
3871:      G=H-T*S
3872:      RETURN
3873:      END
3874:C*****
3875:C*****
3876:      SUBROUTINE SGCUS(T,G)
3877:      IMPLICIT REAL*8 (A-H,O-Z)
3878:CCCC FREE ENERGY OF SOLID CU NEAR MP (NOT STRICT)
3879:CCCC UNIT JOUL & K
3880:      TMP=1358.0
3881:      CP=28.0
3882:      HMP=29658.0
3883:      SMP=74.291
3884:      H=HMP+CP*(T-TMP)
3885:      S=SMP+CP*DLOG(T/TMP)
3886:      G=H-T*S
3887:      RETURN
3888:      END
3889:C*****
3890:      SUBROUTINE SGCUL(T,G)
3891:      IMPLICIT REAL*8 (A-H,O-Z)
3892:CCCC FREE ENERGY OF LIQUID CU NEAR MP (NOT STRICT)
3893:CCCC UNIT JOUL & K
3894:      TMP=1358.0
3895:      CP=32.844
3896:      HMP=42796.0
3897:      SMP=83.966
3898:      H=HMP+CP*(T-TMP)
3899:      S=SMP+CP*DLOG(T/TMP)
3900:      G=H-T*S
3901:      RETURN
3902:      END
3903:C*****
3904:      SUBROUTINE MOXOPZ(XPU,OM,TK,PO2,PO,OX,LS)
3905:C      OXYGEN POTENTIAL CALCURATION BY BLACKBURN MODEL
3906:C      FEB.05 1999 NAGAI
3907:      IMPLICIT REAL*8 (A-H,O-Z)
3908:      COMMON/FL11/FL11,FL21,FL22,FL31
3909:      DATA HK1, HK2, HK3, HK4/-79419.0, -16350.0, -54931.0, -92500.0/
3910:      DATA SK1, SK2, SK3, SK4/14.338, 4.96, 13.37, 21.3/
3911:      DATA QA, QB/-1.25E5, 2.006/
3912:      DATA HU5, SU5/11000.0, -12.75/
3913:      DATA HPU5, SPU5/0.0, 4.3/
3914:      RG=8.313608
3915:      TH=3133.15
3916:      FL21=-0.35

```

```

3917:c      f121=- (TK-2663.15)*1.2D-2
3918:      FL22=2.0+(2743.0-TK)*8.89D-3
3919:c      f122=0.0
3920:C**    FOR SENSITIVITY TEST
3921:c      FL21=-1.0
3922:c      FL22=0.0
3923:C**    FOR SENSITIVITY TEST END
3924:      FL31=0.0
3925:      FL11=0.75+(3073.0-TK)*6.25D-4*1.7
3926:c      f122=f111
3927:      CJ=4.184
3928:      CONVC=1.0D-10
3929:C      HASAMIUCHI HOU
3930:C      IN---XPU,OM,TK      OUT---P02&OXP
3931:      BL1=-50.0
3932:      BL2=100.0
3933:c      BB=DEXP(BL1)
3934:      LL=0
3935:      TT=TK
3936: 2000 CONTINUE
3937:C      AK1=DEXP(HK1/TT+SK1)
3938:c      IF(LS.EQ.0) THEN
3939:      GAK1=-2.046423D10/TT**3+4.6312905D7/TT**2
3940: 1 -1.91151D5/TT+35.49
3941:c      GAK1=-158838.1/TT+28.676
3942:c      AK2=DEXP(HK2/TT+SK2)
3943:c      ELSE
3944:c      GAK1=-79854.0017/TT+15.454481
3945:c      GAK1=-83897.787/TT+16.8939
3946:c      GAK1=-1.69146196D13/TT**3+1.8030446D10/TT**2
3947:c 1 -6.483023D6/TT+773.1501
3948:C      GAK1=GAK1*2.0
3949:c      END IF
3950:      AK1=DEXP(GAK1/2.0)
3951:      FLM=FL31*XPU*(1.0-XPU)
3952:      IF(LS.EQ.0) AK1=AK1*DEXP(FLM)
3953:      IF(LS.EQ.1) AK1=AK1*DEXP(FL11)
3954:      AK2=DEXP(HK2/TT+SK2)
3955:      IF(TT.GE.2000.0) QZF=0.0
3956:      IF(TT.LE.1000.0) QZF=6.6
3957:      IF(TT.GT.1000.0.AND.TT.LT.2000.0) QZF=13.2*(2000.0-TT)/2000.0
3958:      ZFK2=QZF*XPU
3959:      FFK2=10**ZFK2
3960:      AK2=AK2*FFK2
3961:      AK3=DEXP(HK3/TT+SK3)
3962:c      FLM=FL31*XPU*(1.0-XPU)
3963:c      IF(LS.EQ.0) AK3=AK3*DEXP(FLM)
3964:      IF(LS.EQ.1) AK3=AK3*DEXP(FL21)
3965:      AK4=DEXP(HK4/TT+SK4)
3966:      IF(LS.EQ.1) AK4=AK4*DEXP(FL22)
3967:      XFC=667.0*(OM-2.003)
3968:      XFD=(1.0-DTANH(XFC))/2.0
3969:      AK5=DEXP(HU5/TT+SU5)*XFD
3970:C**** this factor is not effective
3971:c      if(ls.eq.1) ak5=ak5*dexp(f111)
3972:      AK6=DEXP(HPU5/TT+SPU5)
3973:      IF(LS.EQ.1) AK6=AK6*DEXP(FL21)
3974:      CA=OM-2.0
3975:      IF(CA.LT.0.0) GO TO 100
3976:      CC=CA
3977: 200 CONTINUE
3978:      ZA=DEXP(108.0*CC**2)
3979:      ZB=DSQRT(ZA)
3980:      AK2=AK2*ZB
3981: 100 CONTINUE
3982:C      CALCURATION OF OXYGEN PRESSURE BY NEWTON/RAPSON MEATHOD
3983:      BB15=1.0/(BB**1.5)
3984:      BB2=BB**2
3985:      AK36=DSQRT(AK3*AK6)
3986:      XK1=1.0+AK1*BB+1.0/(AK2*BB)+DSQRT(AK5/AK2/BB)
3987:      XK2=2.0+AK1*BB+3.0/(AK2*BB)+2.5*DSQRT(AK5/AK2/BB)
3988:      XK3=1.0+DSQRT(AK4*BB)+1.0/DSQRT(AK3*BB)
3989:      XK3=XK3+1.0/AK36/BB
3990:      XK4=1.5+2.0/DSQRT(AK3*BB)+DSQRT(AK4*BB)
3991:      XK4=XK4+2.5/AK36/BB
3992:      FB=(1.0-XPU)*XK2/XK1+XPU*XK4/XK3
3993:      FB=FB-OM
3994:C      ADDITION FOR ISOPRESSURE CALC.
3995:      IF(LL.NE.0) GO TO 2001
3996:      FB1=FB
3997:      LL=1
3998:      BB=DEXP(BL2)
3999:      GO TO 100
4000: 2001 CONTINUE
4001:      IF(LL.GT.1) GO TO 2002
4002:      FB2=FB
4003:      FBX=FB1*FB2
4004:      IF(FBX.GT.0.0) GO TO 2900
4005:      BL3=(BL1+BL2)/2

```

```
4006:    BB=DEXP (BL3)
4007:    LL=2
4008:    GO TO 100
4009: 2002 CONTINUE
4010:    BLD=DABS (BL1-BL2)
4011:    IF (BLD. LE. CONV) GO TO 2800
4012:    FB3=FB
4013:    FBY=FB1*FB3
4014:    IF (FBY. GT. 0. 0) GO TO 2003
4015:    BL2=BL3
4016:    FB2=FB3
4017:    BL3=(BL1+BL3)/2
4018:    BB=DEXP (BL3)
4019:    GO TO 100
4020: 2003 CONTINUE
4021:    BL1=BL3
4022:    FB1=FB3
4023:    BL3=(BL2+BL3)/2
4024:    BB=DEXP (BL3)
4025:    GO TO 100
4026: 2800 CONTINUE
4027:    BBBL=(BL1+BL2+BL3)/3
4028:    BB=DEXP (BBBL)
4029:    PO2=(OM/BB)**2
4030:    CALL GOXA (TK, GO, GFO)
4031:    PO=DSQRT (PO2)*DEXP (-GFO/RG/TK)
4032: C    WRITE (6, *) BBBL, BB, PO2, PO
4033:    GO TO 2950
4034: 2900 CONTINUE
4035:    PO2=1. 0D50
4036: C    WRITE (6, *) FBX, PO2
4037: 2950 CONTINUE
4038: C    OXP=RG*TT*DLOG (PO2)
4039:    RETURN
4040:    END
```