

SSCのIHXモデルとDRACSモデルの改良

1989年4月

動力炉・核燃料開発事業団
大洗工学センター

複製又はこの資料の入手については、下記にお問い合わせください。

〒311-13 茨城県東茨城郡大洗町成田町4002

動力炉・核燃料開発事業団

大洗工学センター システム開発推進部・技術管理室

Enquires about copyright and reproduction should be addressed to: Technology Management Section O-arai Engineering Center, Power Reactor and Nuclear Fuel Development Corporation 4002 Narita-cho, O-arai-machi, Higashi-Ibaraki, Ibaraki-ken, 311-13, Japan

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

SSCのIHXモデルとDRACSモデルの改良

山口 彰* 田嶋雄次**

要 旨

ループ型高速炉プラント動特性解析コードSSC-Lとプール型高速炉プラント動特性解析コードSSC-PのIHX(中間熱交換器)モデルとDRACS(直接炉心冷却系)モデルについて実証炉の仕様に合致する改良及び組み込みを行った。本作業により実証炉の各種事象解析が可能となった。

実証炉の設計仕様によれば伝熱管内が1次系、胴側が2次系のIHXが採用されているが、現状のSSC-L及びSSC-Pでは伝熱管内が2次系、胴側が1次系としてモデル化されている。そこで今回管内1次、胴側2次のモデルを作成しSSC-L及びSSC-Pに組み込んだ。

同様に実証炉の設計仕様によればDRACSモデルは浸漬型が有望視されている。現状のSSC-Lは貫通型であるため、今回これを改造して浸漬型のモデルをSSC-L及びSSC-Pに組み込んだ。SSC-Lには、炉心上部プレナムにDRACS熱交換器を浸漬したモデル、SSC-Pにはホットプール及びIHX入口プレナムにそれぞれDRACS熱交換器を浸漬した2モデルを組み込んだ。(後者を特にPRACSと呼ぶ)。

なお今回作成した新モデルと従来の旧モデルは入力オプションにより選択が可能である。

* 動燃事業団大洗工学センター安全工学部原子炉工学室

** 株式会社 ワイズシステム研究所

Modification of the IHX model and the DRACS model in SSC

Akira Yamaguchi* Yuji Tajima**

Abstract

SSC - L and SSC - P are computer codes for the thermohydraulic transient simulation in a loop - type and a pool - type liquid metal fast breeder reactors (LMFBRs), respectively.

The purpose of this work is to modify the intermediate heat exchanger (IHX) model and the direct reactor auxiliary cooling system (DRACS) model of SSC accordance with the design of a large - scale demonstration LMFBRs. As a result of this work, both codes are to be used for thermohydraulic of various accidents in the demonstration LMFBRs.

In the IHX design of the demonstration LMFBRs, the primary coolant flows inside the heat transfer tube and the secondary coolant is the outside. On the other hand, SSC models the primary flow outside and the secondary flow inside the heat transfer tube. In this work, the IHX model has been modified so that SSC can simulate the thermohydraulics in the demonstration LMFBR.

In the DRACS model of SSC, the heat transport piping runs from the upper plenum and returns to the lower plenum. Then immersed - type DRACS model has been newly developed in this work. For the SSC - P code, PRACS model also has been developed. Every model for the DRACS can be chosen by user input option.

* Reactor Engineering Section, O-arai Engineering Center

** WISE SYSEEM LABORATORY CO., LTD.

目 次

要 旨

図一覧表

表一覧表

1.	序 文	1
2.	モデルの説明	2
	2. 1 IHX モデル	2
	2. 2 DRACS モデル	4
	2. 3 その他のコード修正	8
3.	テスト計算の結果	10
4.	結論及び今後の課題	58
5.	謝 辞	59
6.	参考文献	60
	付録A. 入力マニュアル	61
	付録B. テスト計算の入力データ	63
	付録C. コレクションセット	65
	付録D. 本作業で使⽤したデータセット	100

図 一 覧 表

図3-1 SSC-L旧IHXによるテスト計算結果	12
図3-1a SSC-L旧IHXによるIHX温度分布図	15
図3-2 SSC-L新IHXによるテスト計算結果	19
図3-2a SSC-L新IHXによるIHX温度分布図	22
図3-3 SSC-L旧DRACSによるテスト計算結果	26
図3-4 SSC-L新DRACSによるテスト計算結果	31
図3-5 SSC-P旧IHXによるテスト計算結果	36
図3-5a SSC-P旧IHXによるIHX温度分布図	39
図3-6 SSC-P新IHXによるテスト計算結果	42
図3-6a SSC-P新IHXによるIHX温度分布図	45
図3-7 SSC-P DRACSモデルのテスト計算結果	48
図3-8 SSC-P PRACSモデルのテスト計算結果	53

表 一 覧 表

表C-1 SSC-L IHXモデルの変更のコレクションセット	67
表C-2 SSC-L DRACSモデルの改良のコレクションセット	70
表C-3 SSC-L IHX温度分布図作成用コレクションセット	73
表C-4 SSC-P IHXモデルの変更のコレクションセット	74
表C-5 SSC-P DRACSモデルのコレクションセット	77
表C-6 SSC-P IHX温度分布図作成用コレクションセット	96
表C-7 SSC-P外部バイパスモデルの改良のコレクションセット	97
表C-8 SSC-Pチェックバルブモデルの改良のコレクションセット	98
表C-9 SSC-Pコールドプール圧損モデルのコレクションセット	99

1. 序 文

ループ型及び、プール型実証炉の設計仕様によれば、中間熱交換器（IHX）モデルは伝熱管内が1次系、胴側が2次系のものが採用されている。また崩壊熱除去系としては、浸漬型のDRACSが有望視されている。

高速増殖炉のプラント動特性解析コードであるSSC-L及びSSC-Pの現状のIHXモデルは伝熱管内が2次系、胴側が1次系である。またSSC-Lの現状のDRACSモデルは冷却材を炉心上部プレナムから取り出し、下部プレナムに戻すという貫通型である。（SSC-PにはDRACSモデルはまだ組み込まれていない）

今回既成のモデルを改良して実証炉の設計仕様に合致するように、以下のモデルを新たに組み込んだ。

- (1) SSC-LのIHXモデルを管内1次、胴側2次とする
- (2) SSC-LのDRACSモデルを炉心上部プレナムに浸漬するタイプとする
- (3) SSC-PのIHXモデルを管内1次、胴側2次とする
- (4) SSC-PのDRACSモデルをホットプールに浸漬するタイプとする
- (5) SSC-PのPRACSモデルをIHX入口プレナムに浸漬するタイプとする

本作業の結果、IHXモデルについてはSSC-L、SSC-P両コード共、管内を1次にするか2次にするかが入力オプションによって選べるようになった。

崩壊熱除去系については、SSC-Lは冷却材を上部プレナムから取り出し、下部プレナムに戻すという旧モデルと、上部プレナムにDRACS熱交換器を浸漬するという新モデルの2者が入力オプションによって選べるようになった。

またSSC-PのDRACSモデルは、DRACS熱交換器をホットプール及びIHX入力プレナムに浸漬するという2モデルが入力オプションによって選べるようになった。

今回追加した新モデルは、テスト計算により正常に動作することを確認した。

本報告書は、2章に新モデルの説明、3章にテスト計算の結果、4章に結論及び今後の課題について記したものである。

また付録として、本作業により追加された入力マニュアル、テスト計算の入力データの説明、コレクションセットの全リスト、本作業で使用したデータセットの内容について記した。

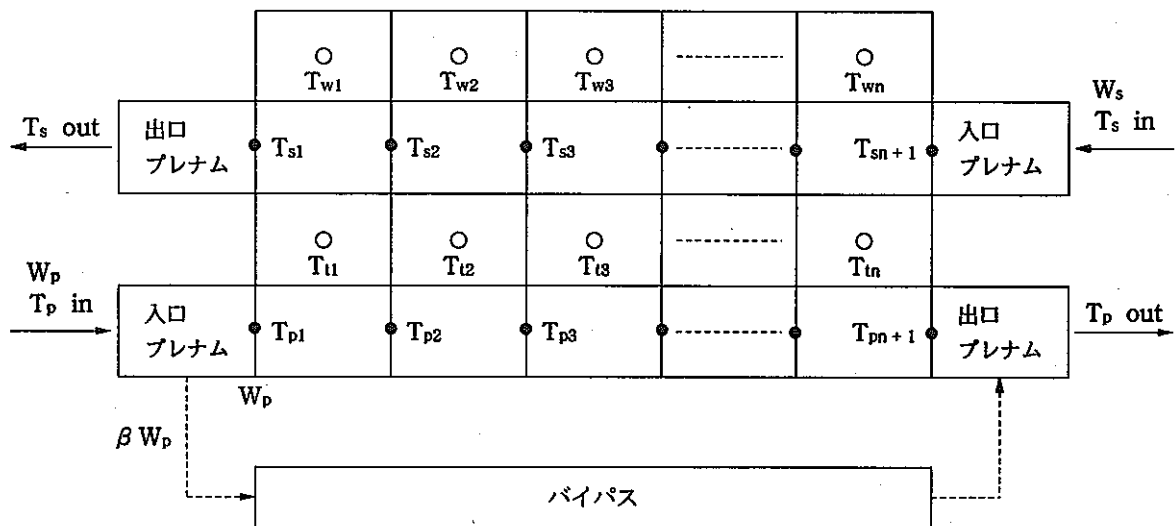
2. モデルの説明

今回追加したモデルは既に組み込まれているモデルを改良して作成したものであり、本章では改良した部分の説明を記す。モデルの詳細については文献〔1〕を参照されたい。

2.1 IHXモデル

SSC-L、SSC-P両コード共、変更した点は1次系と2次系の流路の交換である。すなわち1次系は伝熱チューブとのみ熱交換を行ない、2次系は伝熱チューブ及びシェル壁と熱交換を行なう。

下図にモデルの概略を示す。



(上から順にシェル壁、2次系、伝熱チューブ、1次系となっている)

上記の図を用いて、エネルギー保存則の差分式を示す。

(1次系冷却材)

$$(\rho V)_p (E_{pi+1}^{n+1} - E_{pi+1}^n) = \Delta t [W_p (E_{pi}^{n+1} - E_{pi+1}^{n+1}) - Q_{pt}^{n+1}]$$

(2次系冷却材)

$$(\rho V)_s (E_{si+1}^n - E_{si+1}^{n+1}) = \Delta t [W_s (E_{si+1}^{n+1} - E_{si+1}^n) + Q_{ts}^{n+1} - Q_{sw}^{n+1}]$$

(伝熱チューブ)

$$(MC)_t (T_{ti}^{n+1} - T_{ti}^n) = Q_{pt}^{n+1} - Q_{ts}^{n+1}$$

(シェル壁)

$$(MC)_w (T_{wi}^{n+1} - T_{wi}^n) = Q_{sw}^{n+1}$$

(1次系から伝熱チューブへの伝熱量)

$$Q_{pt}^{n+1} = (UA)_{pt}^{n+1} (T_{pi}^{n+1} - T_{ti}^{n+1})$$

(2次系から伝熱チューブへの伝熱量)

$$Q_{st}^{n+1} = (UA)_{st}^{n+1} (T_{si}^{n+1} - T_{ti}^{n+1})$$

(2次系からシェル壁への伝熱量)

$$Q_{sw}^{n+1} = (UA)_{sw}^{n+1} (T_{si}^{n+1} - T_{wi}^{n+1})$$

(エンタルピーと温度の関係式)

$$E = f(T)$$

ただし変数の意味は

ρV 要素内冷却材の質量

E エンタルピー

Δt タイムステップ

Q 単位時間あたりの伝熱量

W 冷却材流量

MC 構造材の熱容量

T 温度

冷却材の要素平均温度

U 熱伝達率

A 伝熱面積

であり、添字は以下の意味を持つ

p 1次側

s 2次側

t 伝熱管

w シェル壁

i 空間ノード

n 時間ノード

要素平均温度については、従来モデルと同様に対数平均温度差の考え方に基づいて決定している。

また伝熱に寄与しないバイパス流についても、従来モデルと同様に1次系にのみ考慮している。流路交換に伴ない、エネルギー式だけでなく、運動方程式に使用する流体慣性や流力等価直径

等も変化する。これらの修正も合わせて行なった。

なお従来モデルと新モデルは入力オプションにより選択が可能である。

入力マニュアル、コレクションセットについては本報告書の付録を参照されたい。

2. 2 DRACSモデル

現在SSC-Lにモデル化されているDRACSは、炉心上部プレナムから冷却材を取り出し、下部プレナムに戻すというものである。

今回は新しいタイプとして以下のモデルを追加した。

- DRACS熱交換器を上部プレナムに浸漬する (SSC-L)
- DRACS熱交換器をホットプールに浸漬する (SSC-P)
- DRACS熱交換器をIHX入力プレナムに浸漬する (SSC-P)

3番目のタイプを特にPRACSと呼んでいる。

従来モデルからの変更点は以下の3点である。

- 1次系配管と機器を取り除く
- DRACS1次系出入口圧力
- DRACS1次系出入口でのエネルギーバランス

質量保存則については、今回のモデルはすべて浸漬型であるため、特に考慮する必要はない。ただし、上部プレナム2次元モデルと連結する場合には考慮すべきである。今回は上部プレナムは1点あるいは2点モデルとした。(正確には2点モデルの場合もマスバランスは考慮すべきであるが、今回は2領域全体でバランスすればよいという立場をとった。これは今後の課題である)

- DRACS1次系の運動方程式 (全モデル共通)

$$\left(\frac{Z}{A}\right) \frac{dw}{dt} = \Delta P + \rho g Z - \frac{W^2}{A^2} (1/\rho_{out} - 1/\rho_{in}) - \frac{fZw|w|}{2\rho DA^2} - \frac{cw|w|}{2\rho}$$

右辺 第1項は圧力勾配

第2項は重力圧損

第3項は加速圧損

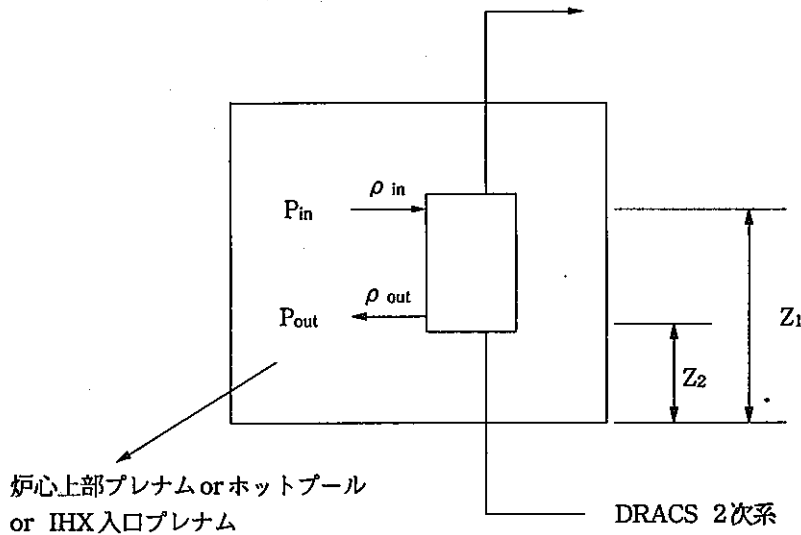
第4項は摩擦圧損

第5項は形状圧損

であり、形状圧損係数cは入力あるいは定常計算において定格条件と合致するように決められる。

また $Z, \Delta P$, は下図の記号を用いて次式で定義する

$$\begin{cases} Z = Z_1 - Z_2 \\ \Delta P = P_{in} - P_{out} = -\rho \text{ing}Z \\ \rho = \frac{1}{2} (\rho_{in} + \rho_{out}) \end{cases}$$



前ページの記号の意味は以下のようになっている

- Z DRACS1 次系の出入口エレベーション差
- Z_1 DRACS1 次系の入口高さ
- Z_2 DRACS1 次系の出口高さ
- ρ DRACS1 次系の平均密度
- ρ_{in} DRACS1 次系の入口の密度
- ρ_{out} DRACS1 次系の出口の密度
- P_{in} DRACS1 次系の入口圧力
- P_{out} DRACS1 次系の出口圧力
- W DRACS1 次系の流量
- A DRACS1 次系の流路断面積
- D DRACS1 次系の等価水力直径
- f 摩擦圧損係数
- c 形状圧損係数
- g 重力加速度

エレベーションについてはZとZ₁のみを入力とする。

圧力についてはΔPを直接計算し、P_{in} P_{out}は特に計算しない。

ΔPの計算は ρ_{in}の流体で満たされていると仮定し静水圧の差とした。

なお、この運動方程式は他の方程式(1, 2次系の流動や核計算等)

と共に5次の予測子・修正子法により解かれる。

○ DRACS1次系出入口でのエネルギー保存則

SSC-Lの炉心上部プレナム及びSSC-Pホットプールに浸漬するタイプの場合、下図の記号を用いて

$$(\rho V) \frac{dE}{dt} = W_C(E_C - E) + W_B(E_B - E) + \sum A_i U_i (T_i - T) + |W_D| (E_D - E)$$

と表わされる。

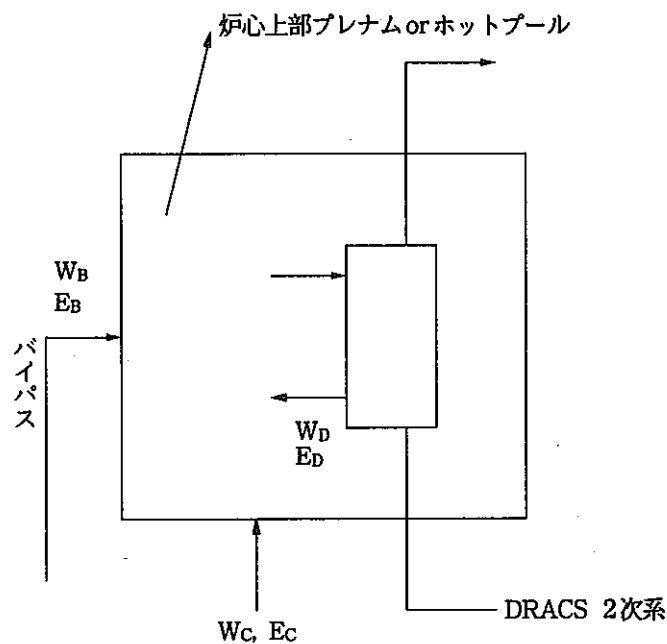
右辺 第1項は炉心からの熱輸送量

第2項はバイパスチャンネルからの熱輸送量

第3項は構造材やカバーガスとの熱伝達量

第4項はDRACSからの熱輸送量

であり、2領域モデルの場合は上式右辺の第4項が2領域境界とDRACS出口の位置関係により上部、下部どちらかの領域のエネルギー式に加算される。



前ページの記号の意味は以下の様になっている

- E 炉心上部プレナム (or ホットプール: 以下略) のエンタルピー
- ρV 炉心上部プレナムの冷却材質量
- E_C 炉心出口冷却材のエンタルピー
- W_C 炉心出口冷却材流量
- E_B バイパスチャネルのエンタルピー
- W_B バイパスチャネル流量
- A_i 炉心上部プレナムと構造材等の伝熱面積
- V_i " " 熱伝達率
- T_i 構造材等の温度
- E_D DRACS1次系出口エンタルピー ($W_D < 0$ の時は入口エンタルピー)
- W_D DRACS1次系の流量
- T 炉心上部プレナムの温度 ($T = T(E)$)

なお、このエネルギー保存則は他の方程式と共に5次の予測子・修正子法により解かれる。

SSC-PのIHX入口プレナムに浸漬するタイプ (PRACS) の場合、次ページの図の記号を用いて

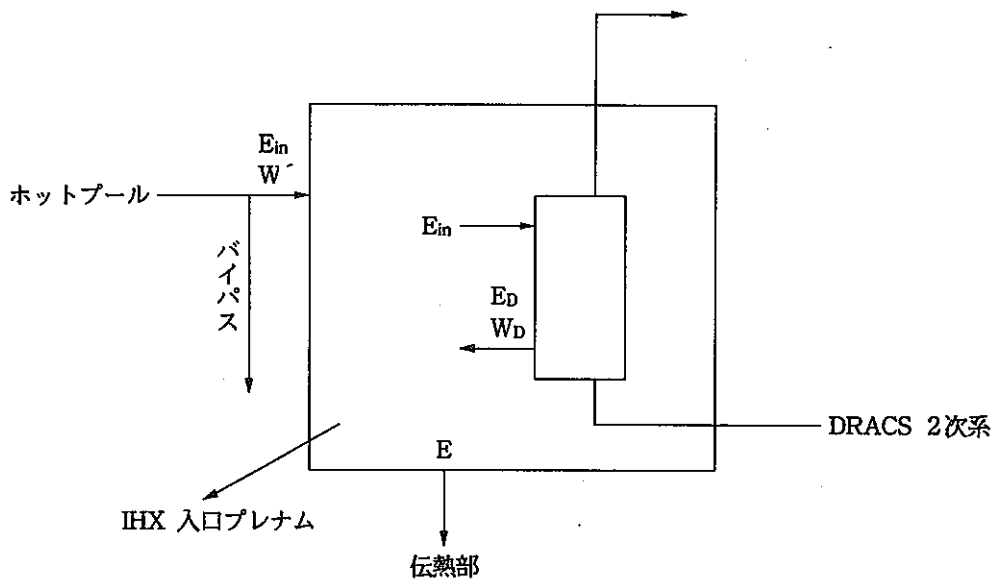
$$(\rho V) \frac{dE}{dt} = W' (E_{in} - E) + |W_D| (E_D - E)$$

と表わされる。記号の意味は以下の様になっている

- E IHX入口プレナムのエンタルピー
- ρV IHX入口プレナム冷却材の質量
- E_{in} IHX入口エンタルピー
- W' IHX伝熱部の流量 (バイパス流を引いたもの)
- E_D DRACS1次系出口エンタルピー
- W_D DRACS1次系流量

なお上式はIHX伝熱計算の前に以下の差分式により直接解かれる。

$$(\rho V) \frac{E_{n+1} - E_n}{\Delta t} = W' (E_{in} - E^{n+1}) + |W_D| (E_D - E^{n+1})$$



以上のDRACSモデルの追加によりSSC-Lについては従来の貫通型と上部プレナム浸漬型、SSC-Pについてはホットプール浸漬型とPRACSがそれぞれ入力オプションによって選択できるようになった。

入力マニュアルとコレクションセットについては本報告書の付録を参照されたい。

(注) DRACSの入力は従来とほとんど変わらないが、新モデルの1次系配管については最低1ノードの分割数をホットレグ、コールドレグ共に入力する必要がある。

(配管でのエネルギー式は解いていないが、変数名は従来そのまま使用している部分があるため。1次系の配管や機器についての入力データは従来のもをそのまま利用すれば問題はない)

2.3 その他のコード修正

(1) SSC-L

- IHX 胴側を流れる系のパイプ内径を計算する式に誤りがあったので修正した。
(入力データで0を入力した場合に本式を利用するが、今まではすべて0以外の値を入力しているので問題はなかった)
- 上部プレナム2領域モデルの改良に誤りがあったので修正した。(ただし本修正の結果もやや問題があり、2領域モデルについては再調査の必要がある)

(2) SSC - P

- 炉心外部バイパス流量が0の時、現コードには誤りがあったのでそれを修正した。
- 炉心核計算に現コードではプロンプトジャンプ法が使用できなかったのを修正した。
- コールドプールの液位がトランジェント計算時に激しく振動することがあったのでコールドプールの圧損を設定した。
- チェックバルブの圧損係数を流量の折れ線近似としていたが、これを直線近似に変更した。
- ホットプールのモデルが入力データに関係なくいつも2領域モデルであったのでこれを修正した。

これらの他に IHX 内の温度分布のプロット図を書くために温度分布をプロットファイルに出力するコレクションセットも作成した。

本章で述べた修正内容とコレクションセットの対応については本報告書の付録を参照されたい。

なお SSC - L の修正は SSC - L の CY42 に対して行なったものである。

3. テスト計算の結果

本作業で作成した新モデルの確認のためSSC-L,SSC-Pそれぞれ以下の4ケースのテスト計算を行なった。

事象は自然循環を仮定した除熱源喪失事故解析 (DRACSについては2次系以降は強制循環) であり、全ケース共cpuは90分であった。

ケース名	コード名	モデル	解析時間 (秒)
L-1	SSC-L	旧 IHX モデル	14000
L-2	SSC-L	新 IHX モデル	14000
L-3	SSC-L	旧 DRACS モデル	25000
L-4	SSC-L	新 DRACS モデル	25000
P-5	SSC-P	旧 IHX モデル	14000
P-6	SSC-P	新 IHX モデル	14000
P-7	SSC-P	ホットプール浸漬型DRACS	14000
P-8	SSC-P	PRACS モデル	14000

ケースL-3とL-4が他と比べて計算が速いのは、炉心チャンネル分割数とIHXノード分割数が少ないためである。

本報告書の付録に入力データの説明を記したので参照されたい。

解析結果のプロット図を図3-1から図3-8に示す。

ケース名の番号と図の番号が対応している。

また図3-1aの様に文字aを付けたものはIHXの温度分布図である。IHXの温度分布図は定常時、過渡時、最終時について示した。図中の番号は

- 1 1次系冷却材温度
- 2 2次系冷却材温度
- 3 伝熱管温度
- 4 シェル壁温度

を表わしており、スタッガードメッシュを使用しているため3と4は両端が切れていることが分

かる。

これらをふまえて図3-1aと図3-2a及び図3-5aと図3-6aを比較すればIHX1次2次交換のモデルは正常に動作していることが分かる。すなわちシェルの温度は2次系の温度に近づいている。

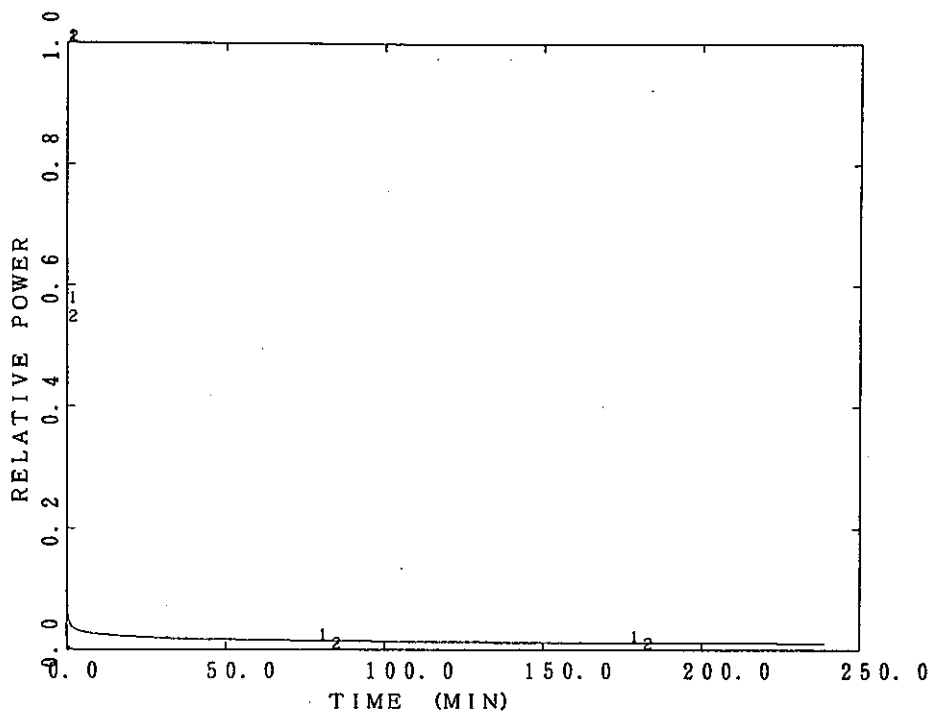
SSC-LのDRACSモデルについては図3-3と図3-4を比較して炉容器出口温度は新モデルの方がやや低くなっている。これは新モデルの方がDRACS1次系の流量が多いためこのためDRACSによる除熱量が多くなっている。(新モデルのDRACS定格条件は未知のため旧モデルと同じデータを使用した)

また、炉容器の入口温度はDRACSが起動(1800秒で起動)すると直ちに上昇して出口温度に接近している。これは新モデルでは、IHXの1次側入口温度がDRACS起動により急に低くなり自然循環力が旧モデルと比べて大きくなる。このため1次系の流量が多くなりIHXにより除熱が少なくなる。(これは旧モデルと、比較しての意味であって実際に流量が増加しているわけではない。実際には流量減少が旧モデルより少なくなりこのような結果になる。)

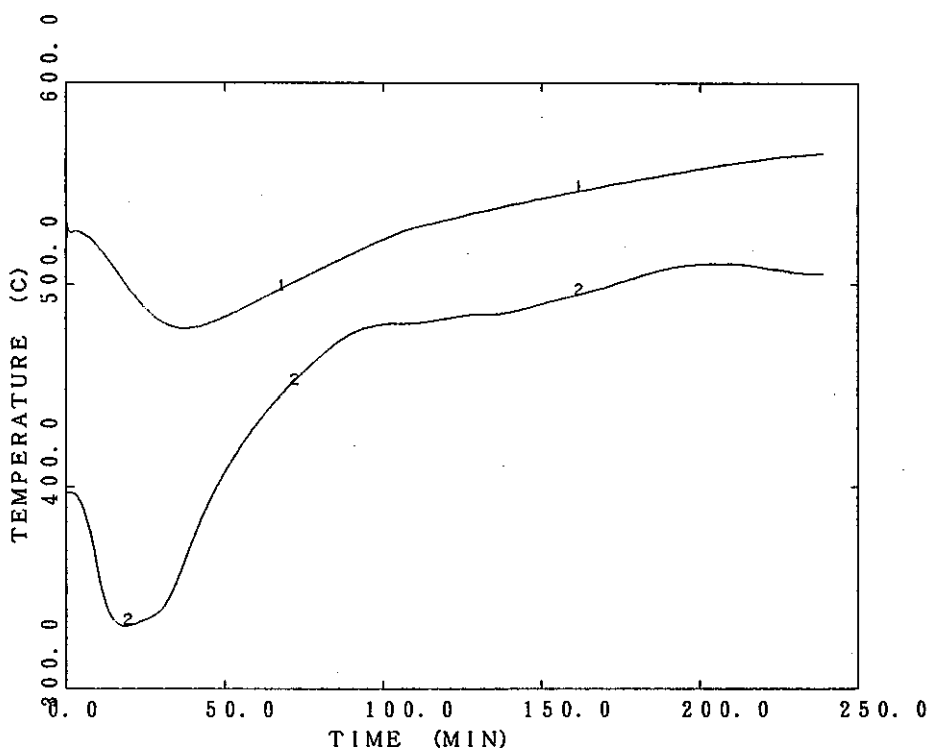
最後にSSC-PのDRACSモデルについて図3-6~図3-8を比較する。図3-6はDRACSが起動されない時の結果であり、(IHXモデルはケースP-5以外はすべて新モデルを用いた)これと比較すると、DRACS or PRACSを起動したケースはホットプール温度が低くなっており、SSC-Pの両モデルも正常に動作していることが分かる。図3-7と図3-8のIHX1次系の流量を比較するとDRACS起動直後(1800秒)図3-7では減少し、図3-8では増加している。図3-8(PRACS)で増加している理由は入口プレナムの温度が低くなり自然循環力が増加したためである。

一方図3-7の場合はホットプールが冷やされるためIHX入口圧力が減少する。(IHX入口はホットプール液面と炉心出口の中間に位置しているためIHX入口圧は密度変化により変化する)この場合もIHXでの自然循環力は増加するが、圧力減少の方が大きいためこのような結果になった。このようにSSC-PではIHX1次系の流量はホットプールとコールドプールの液位差も支配的要因であり1次系全体の自然循環力を正しく評価するためには両プールの詳細な温度分布が必要となる。これは今後の課題である。

なおSSC-Pのモデル及び入力マニュアルについては文献[2]及び[3]を参照されたい。

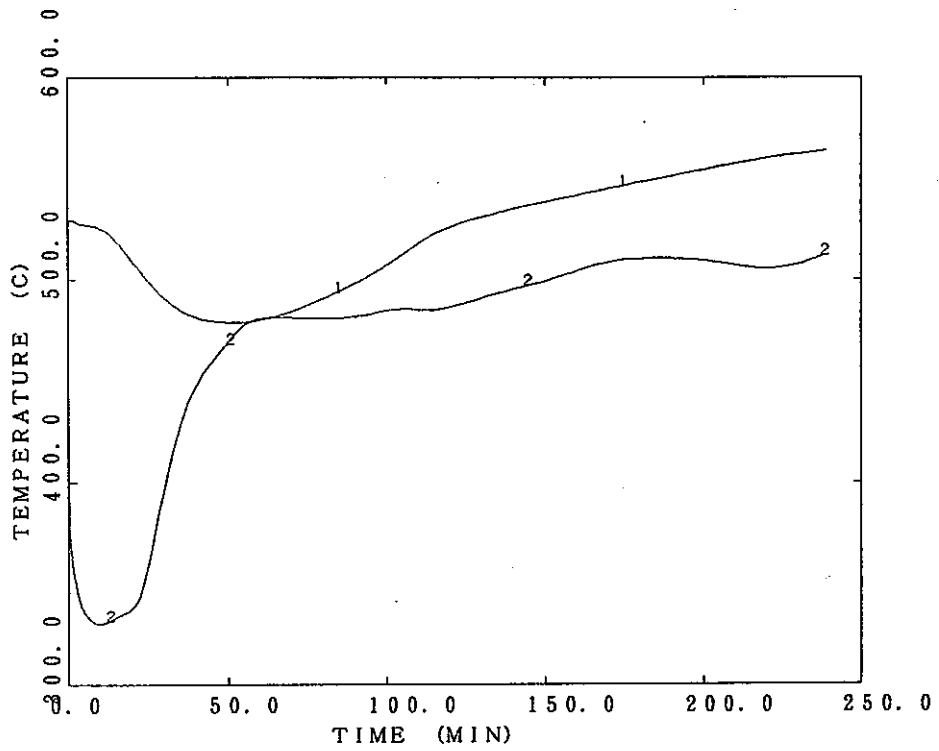


原子炉出力 (1全出力 2核分裂による出力)

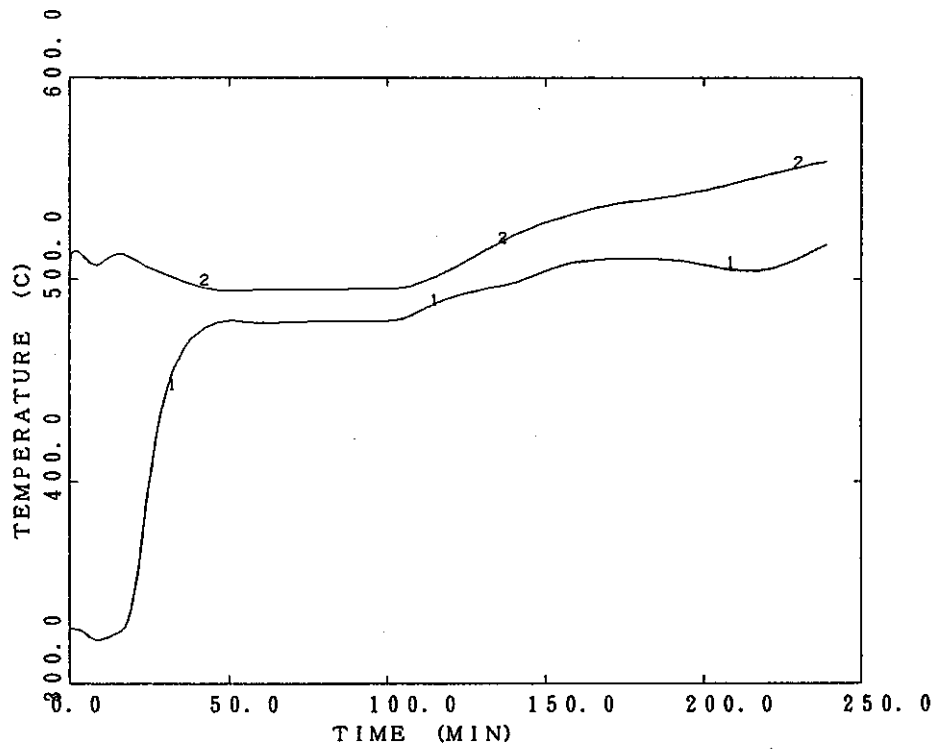


炉容器出入口温度 (1出口 2入口)

図3-1 ケースL-1のプロット出力

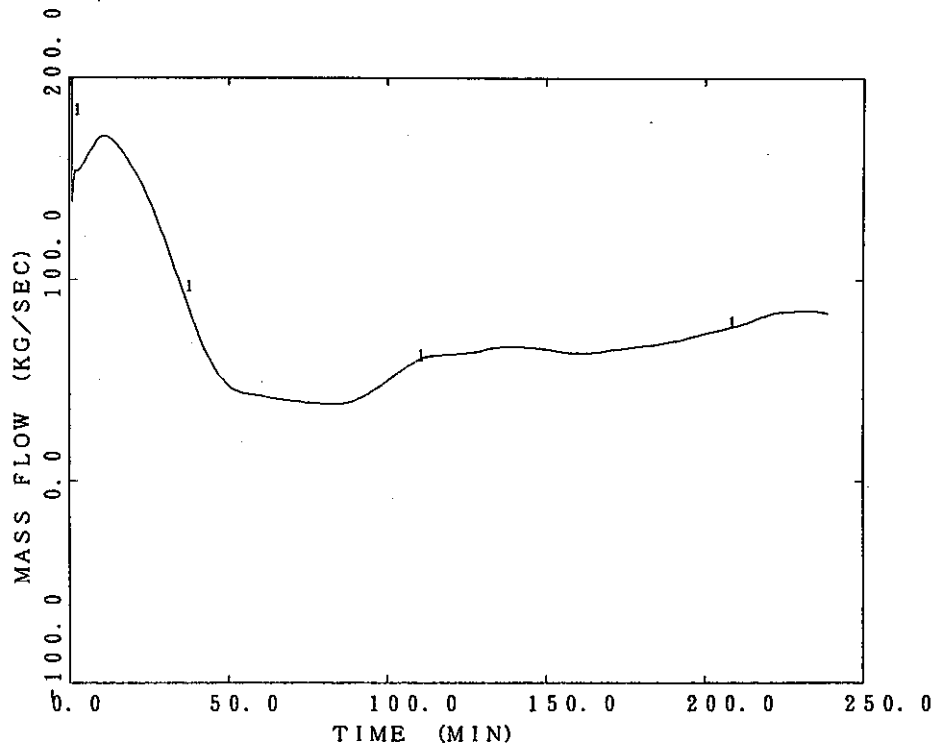


IHX1次系出入口温度 (1入口 2出口)

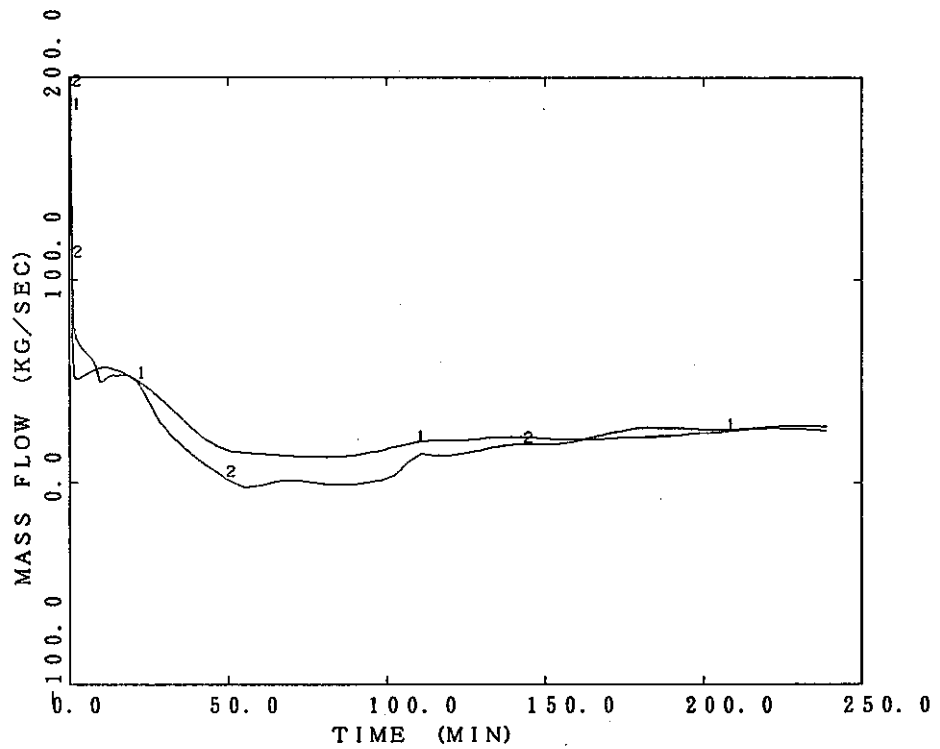


IHX2次系出入口温度 (1入口 2出口)

図3-1 ケースL-1のプロット出力 (つづき)



全炉心流量



IHX1系と2次系の流量

図3-1 ケースL-1のプロット出力 (つづき)

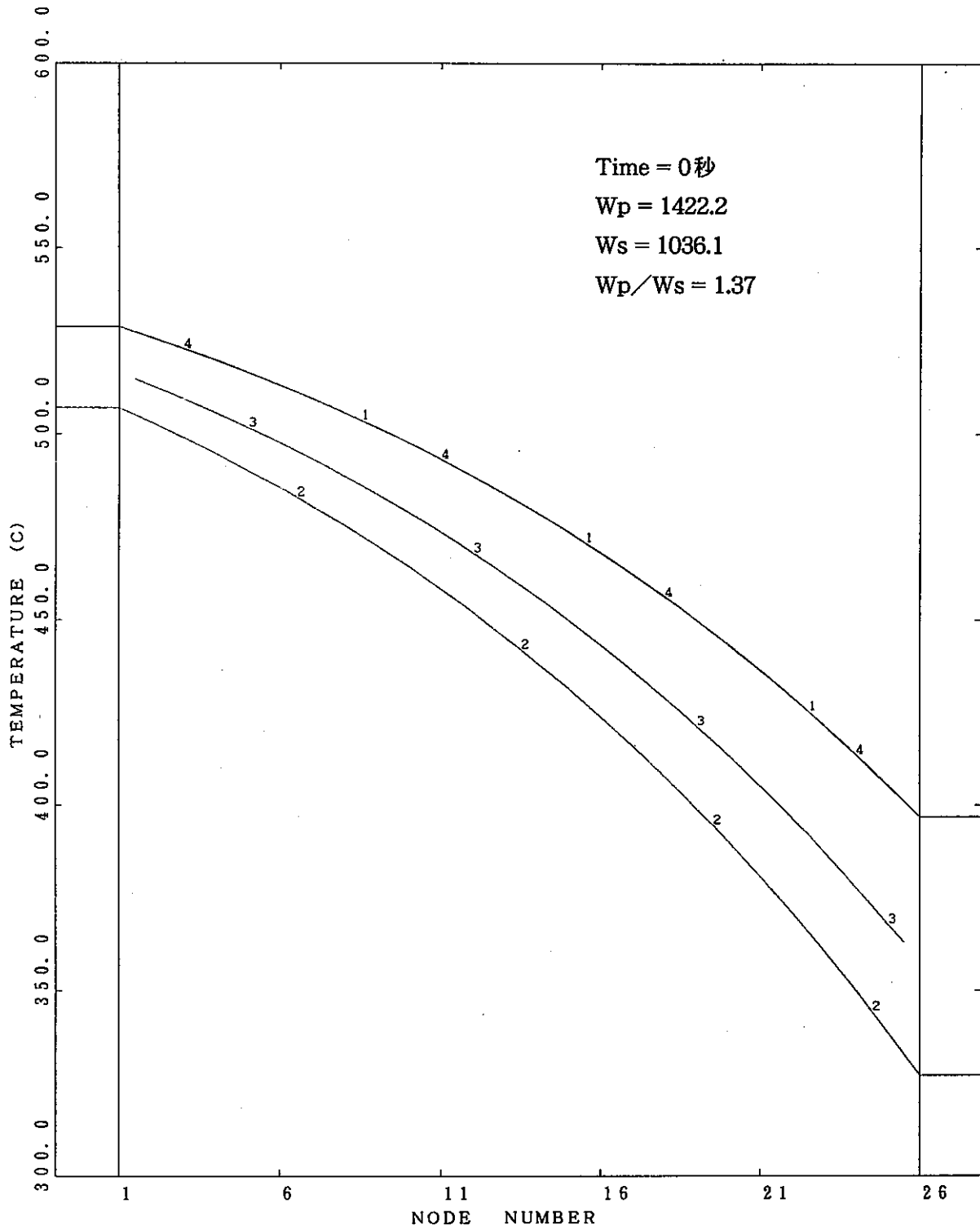


図3-1a ケースL-1のHX温度分布図

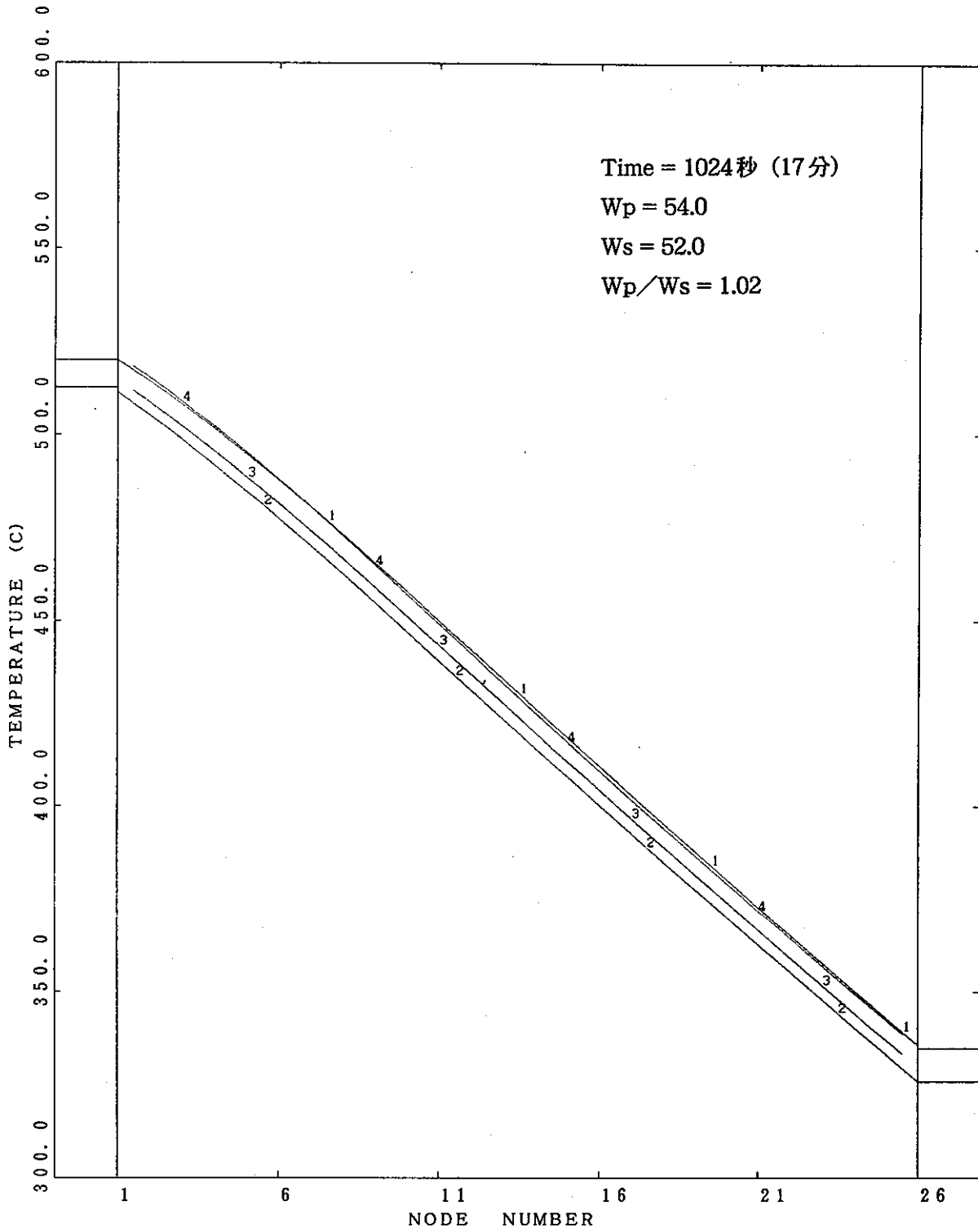


図3-1a ケースL-1のIHX温度分布図 (つづき)

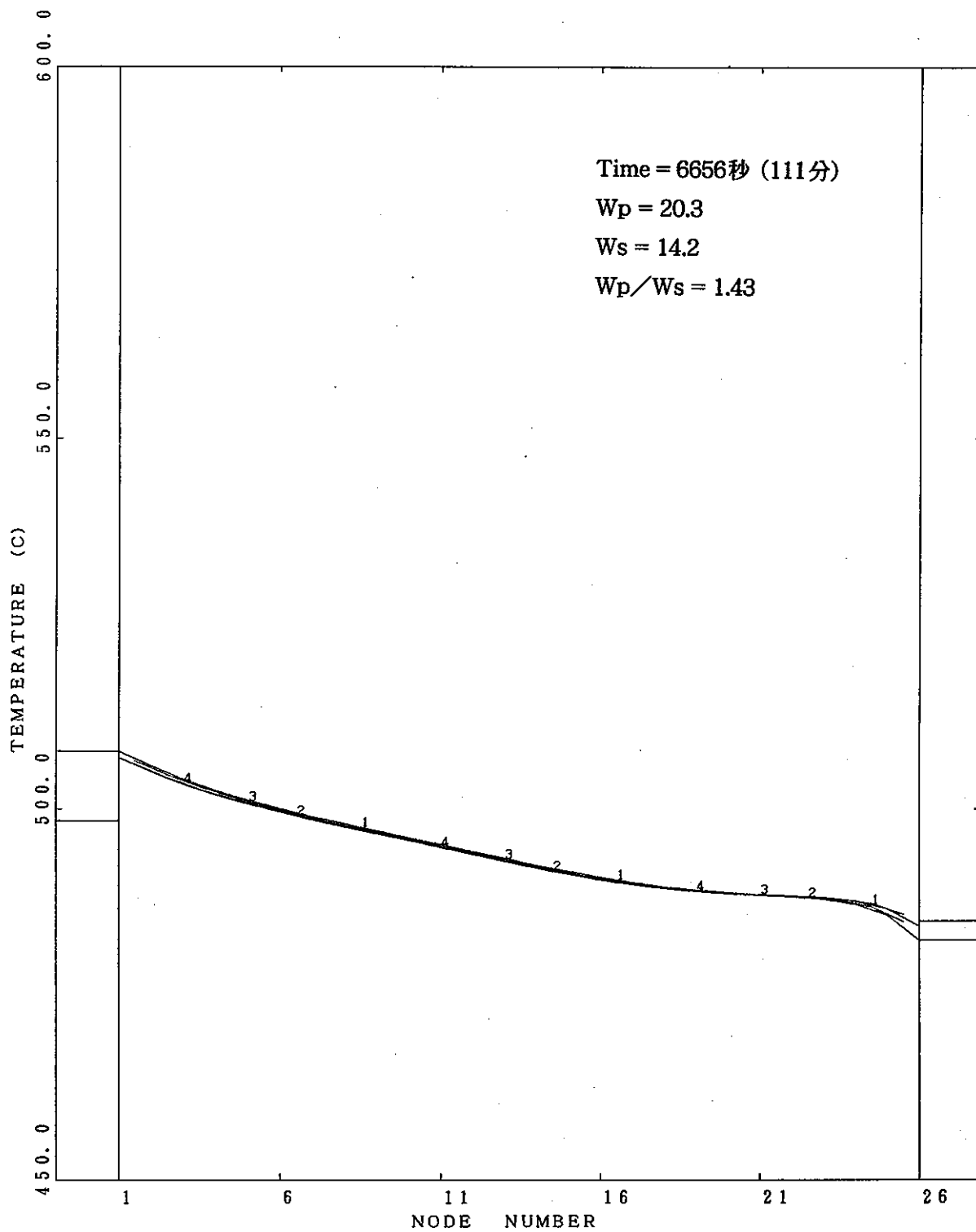


図3-1a ケースL-1のHX温度分布図 (つづき)

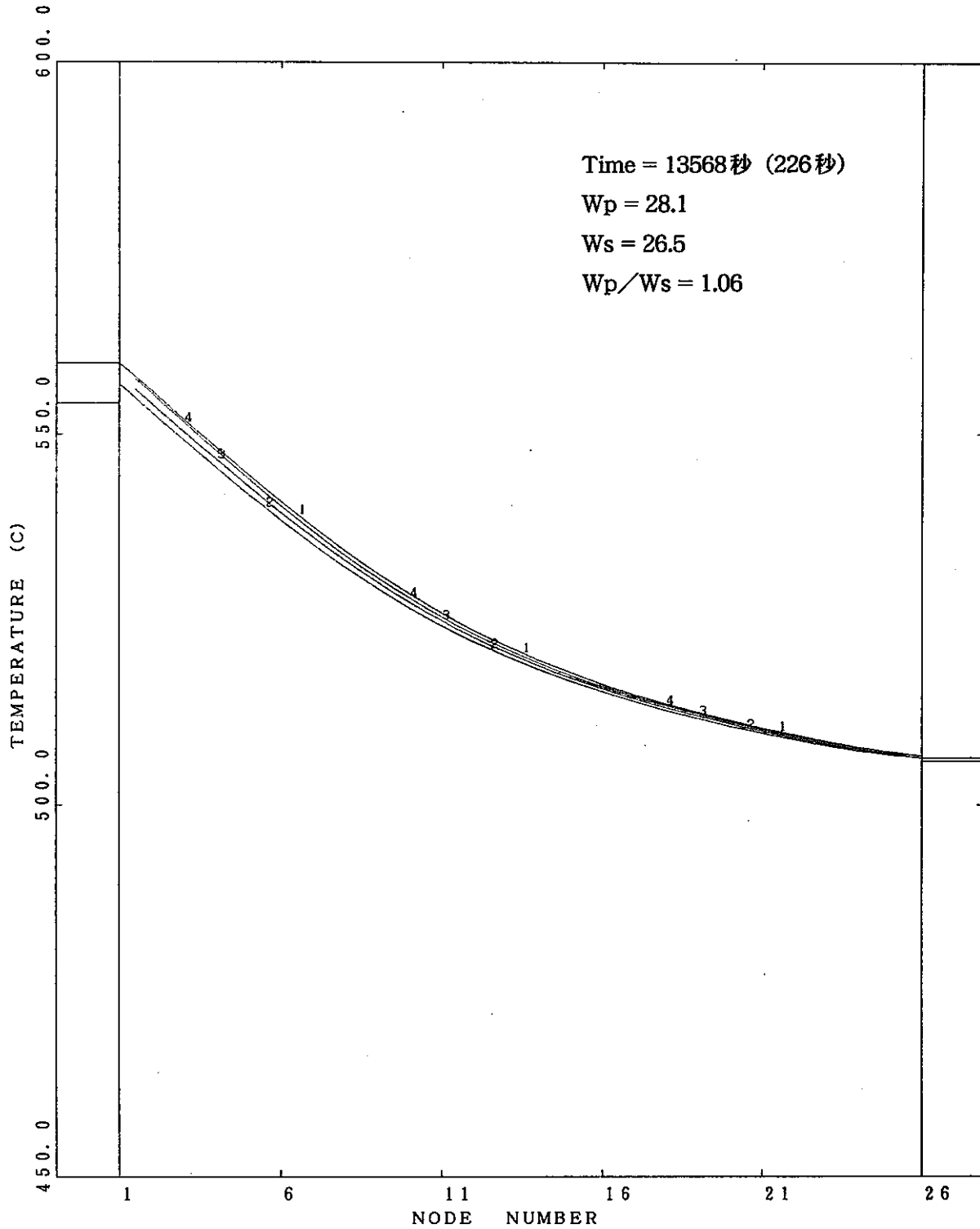
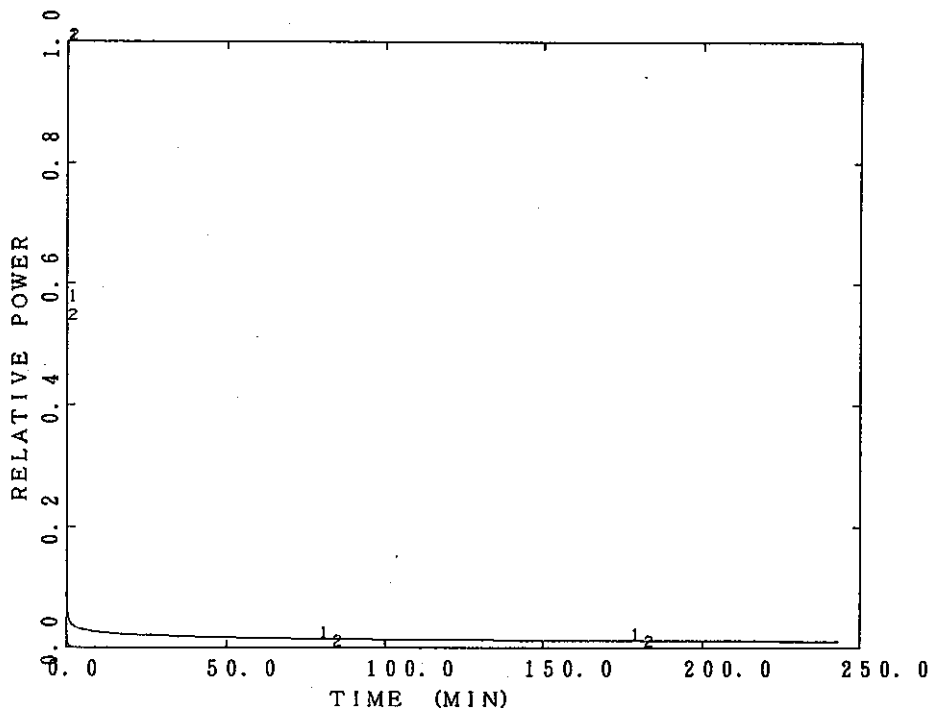
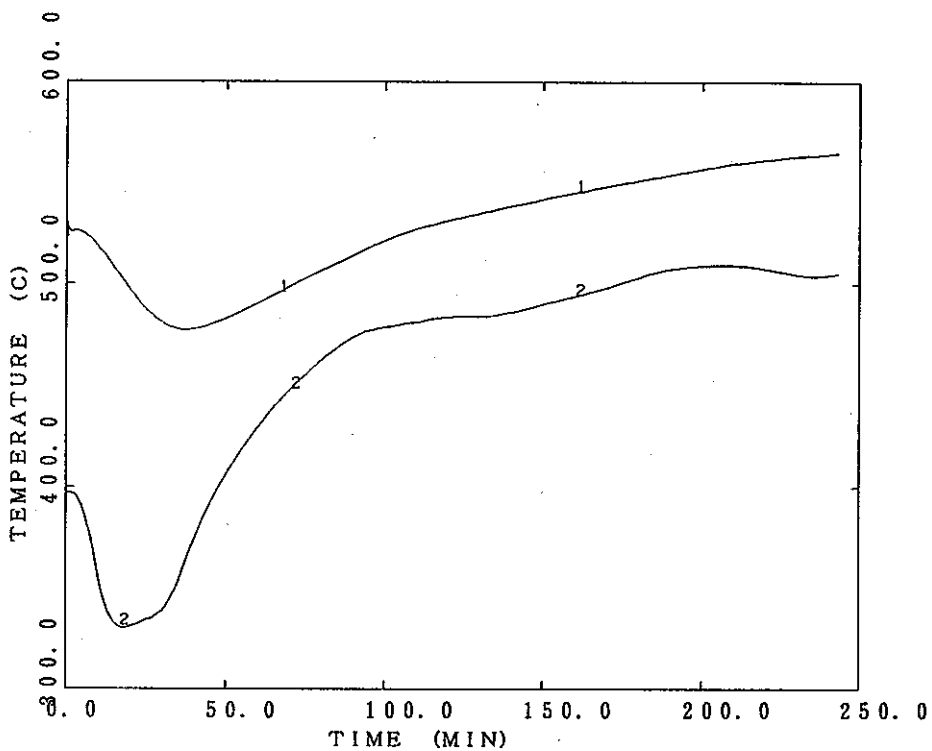


図3-1a ケースL-1のIHX温度分布図 (つづき)

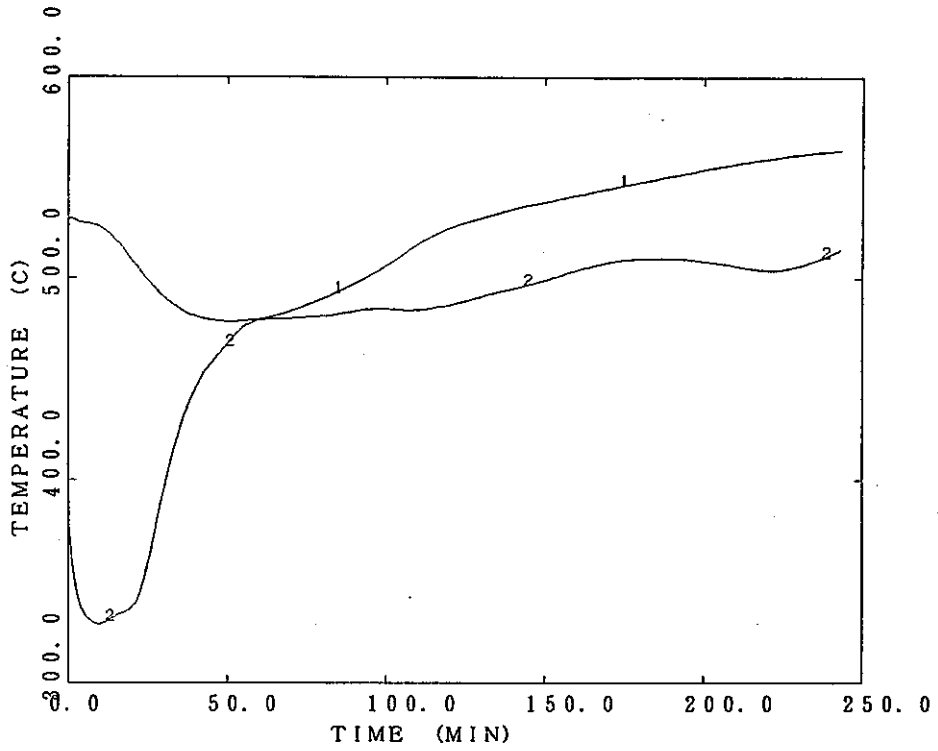


原子炉出力 (1全出力 2核分裂による出力)

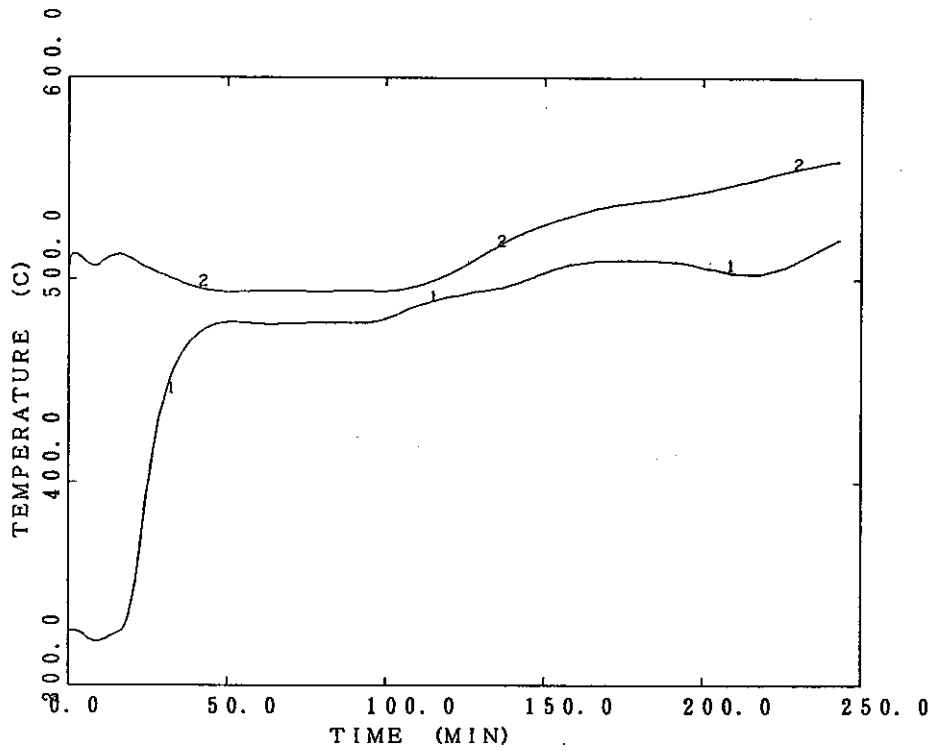


炉容器出入口温度 (1出口 2入口)

図3-2 ケースL-2のプロット出力

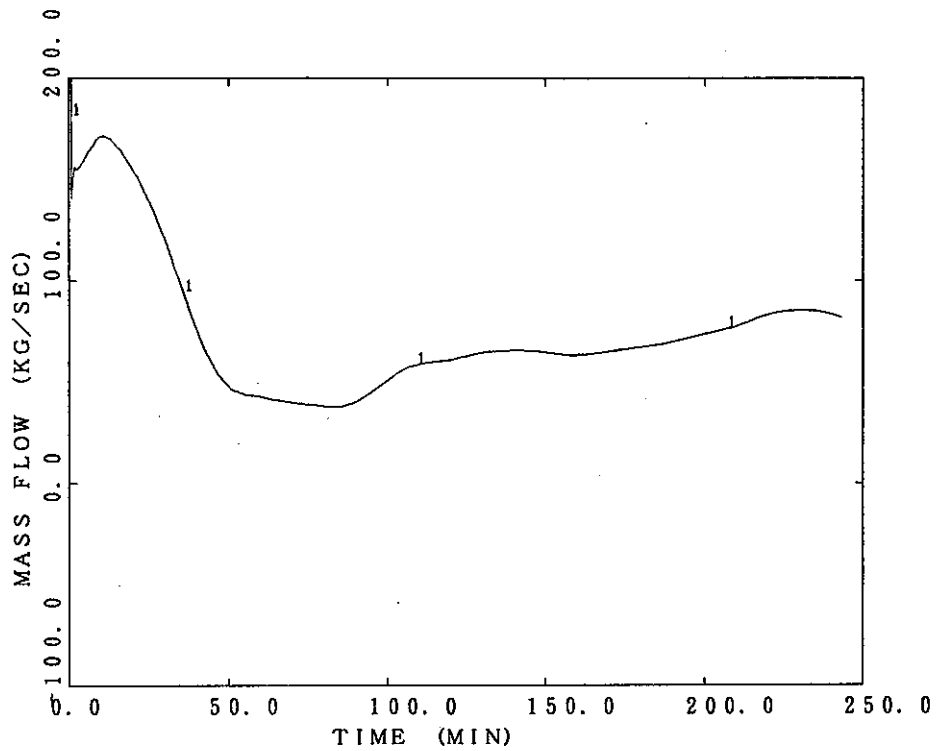


IHX1次系出入口温度 (1入口 2出口)

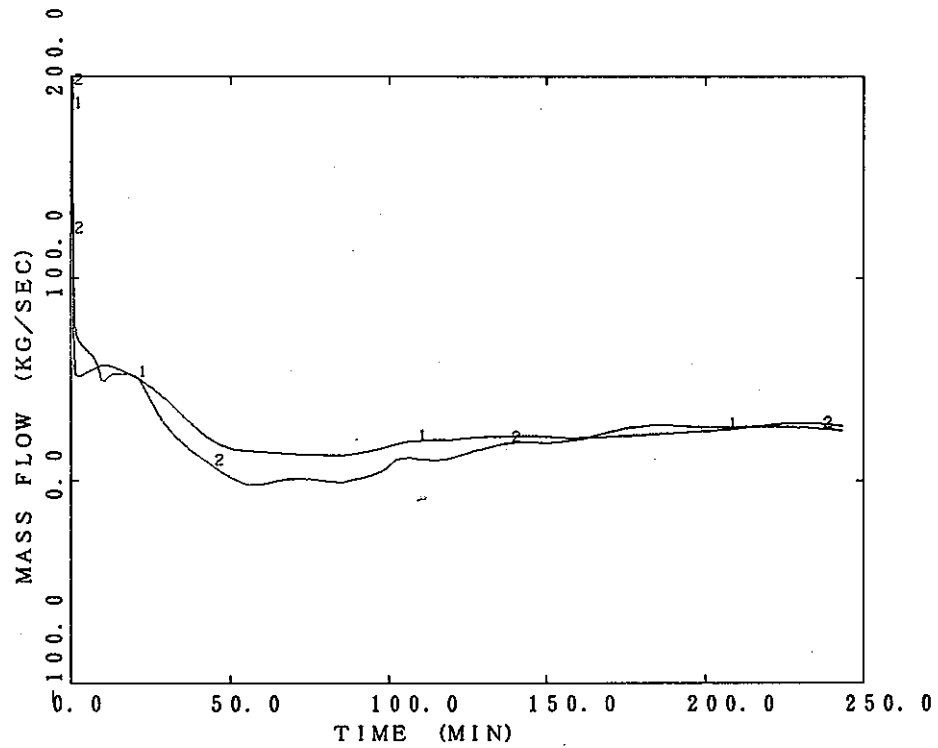


IHX2次系出入口温度 (1入口 2出口)

図3-2 ケースL-2のプロット出力 (つづき)



全炉心流量



IHX1 次系と2次計の流量

図3-2 ケースL-2のプロット出力 (つづき)

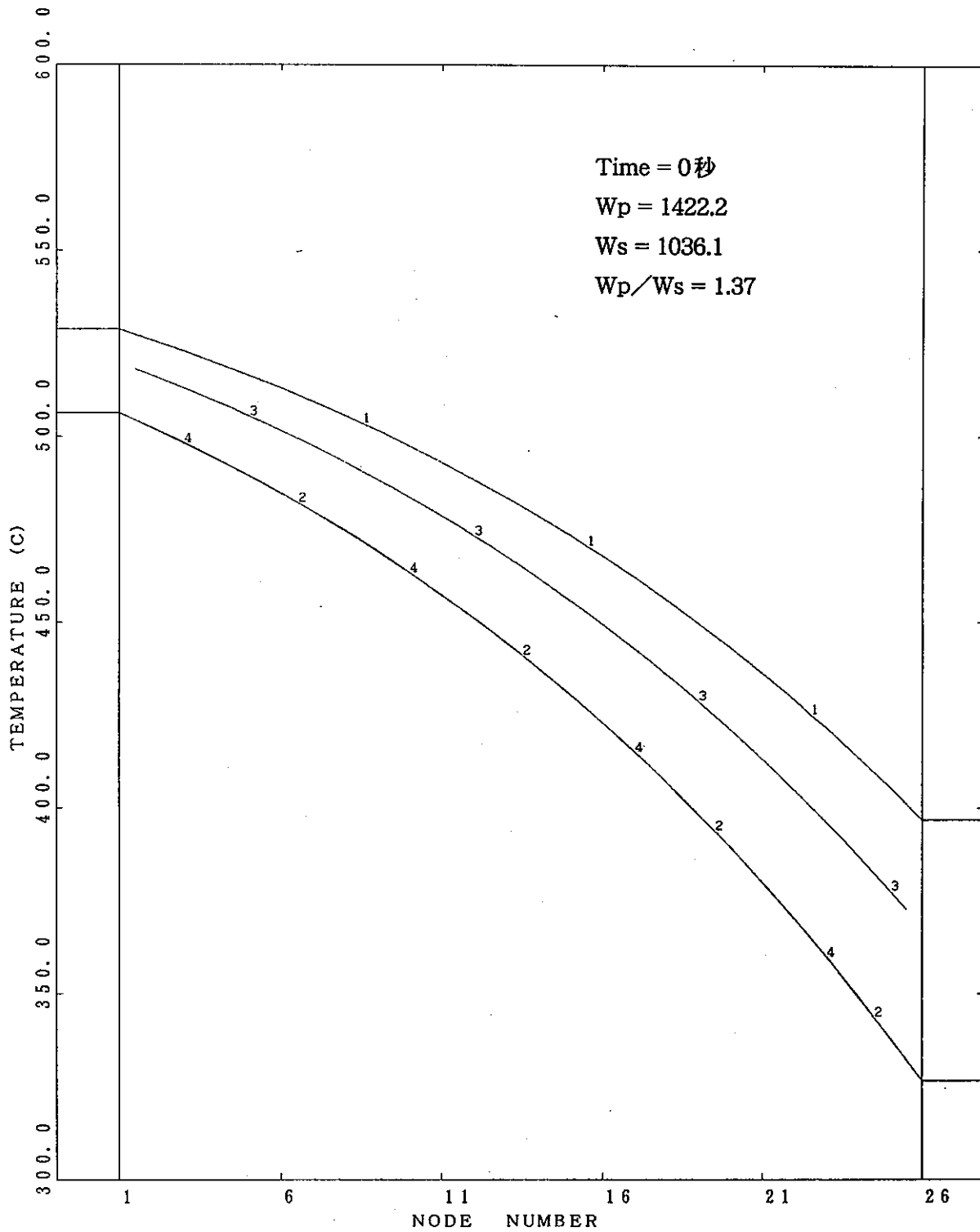


図3-2a ケースL-2のIHX温度分布図

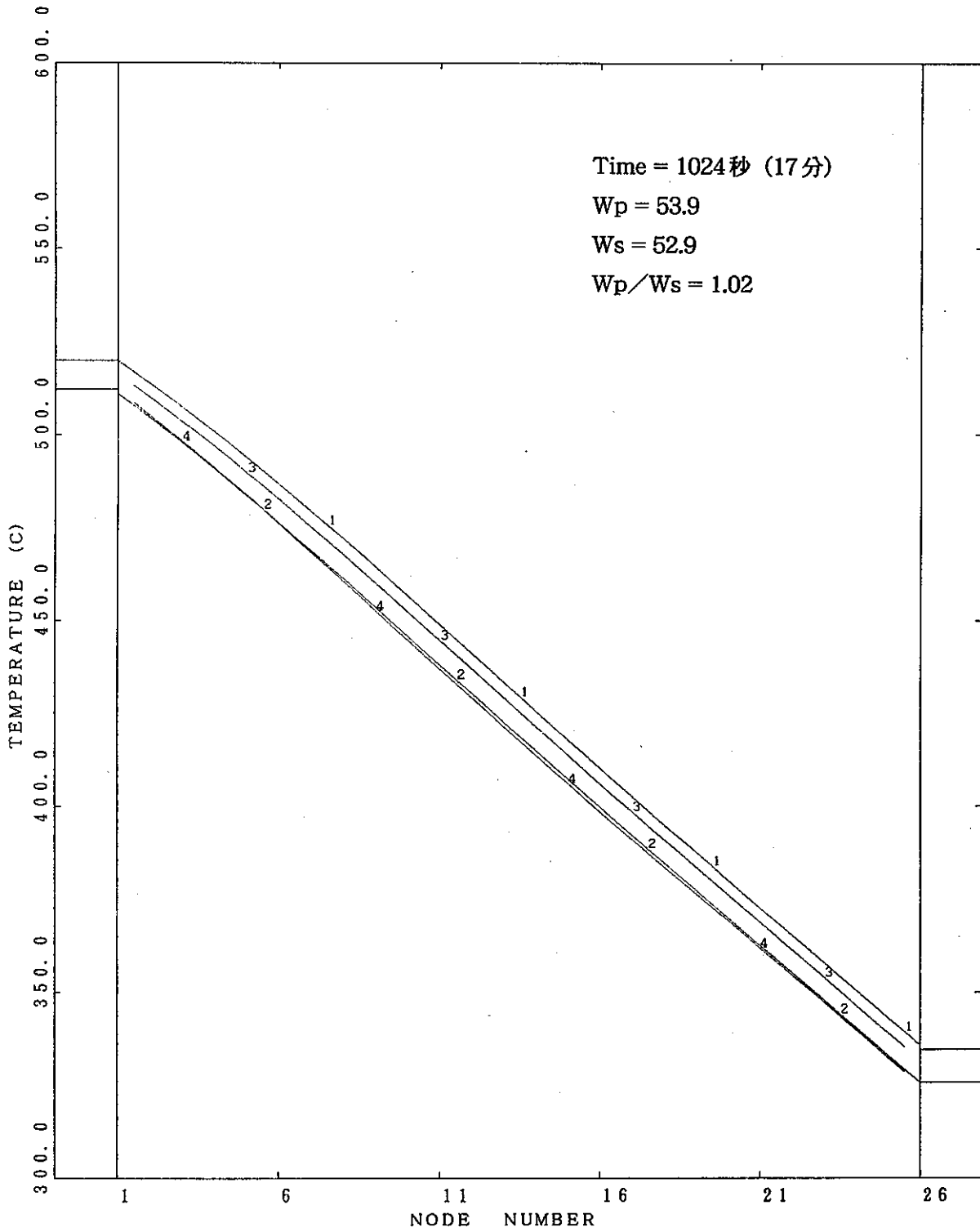


図3-2a ケースL-2のIHX温度分布図 (つづき)

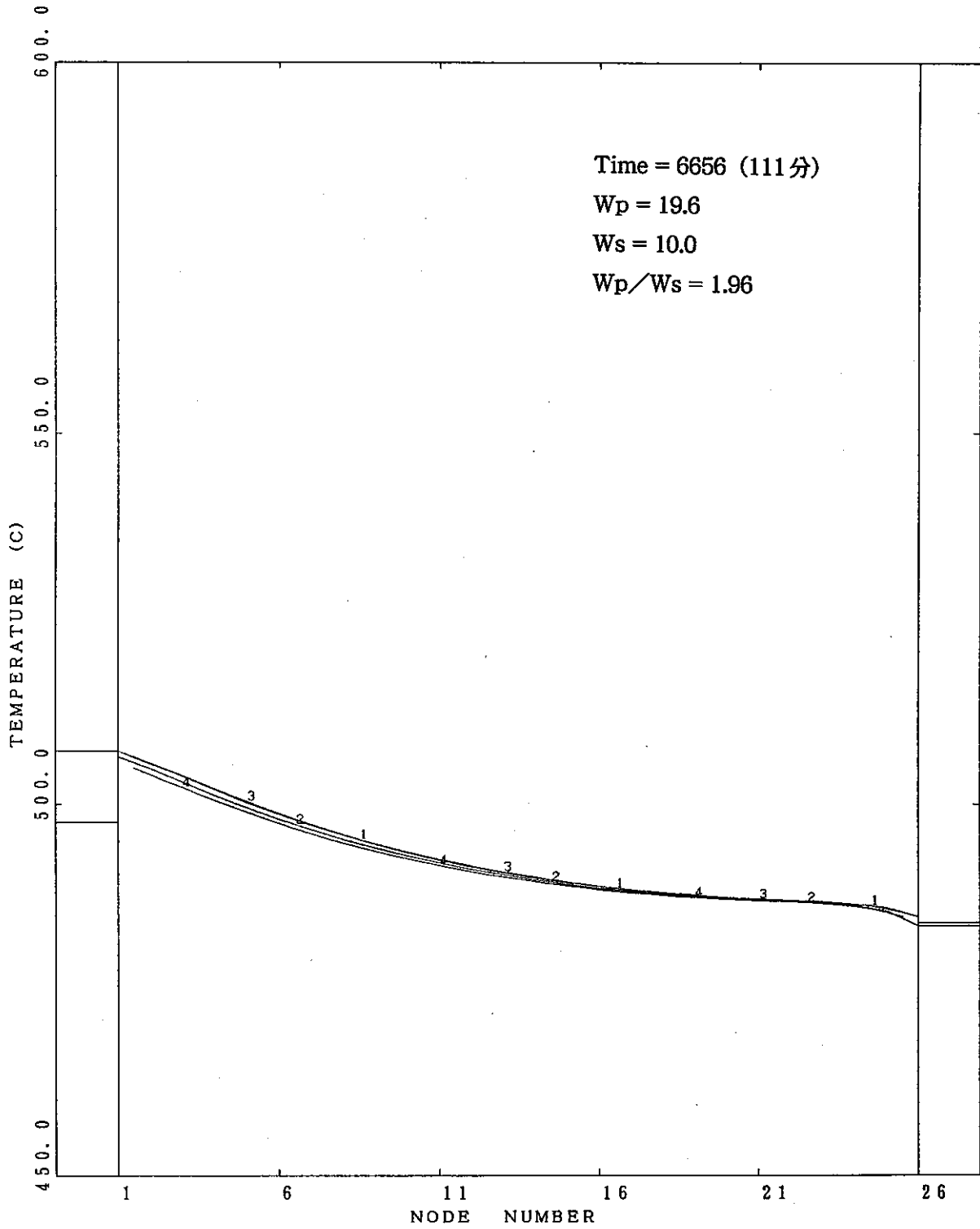


図3-2a ケースL-2のIHX温度分布図 (つづき)

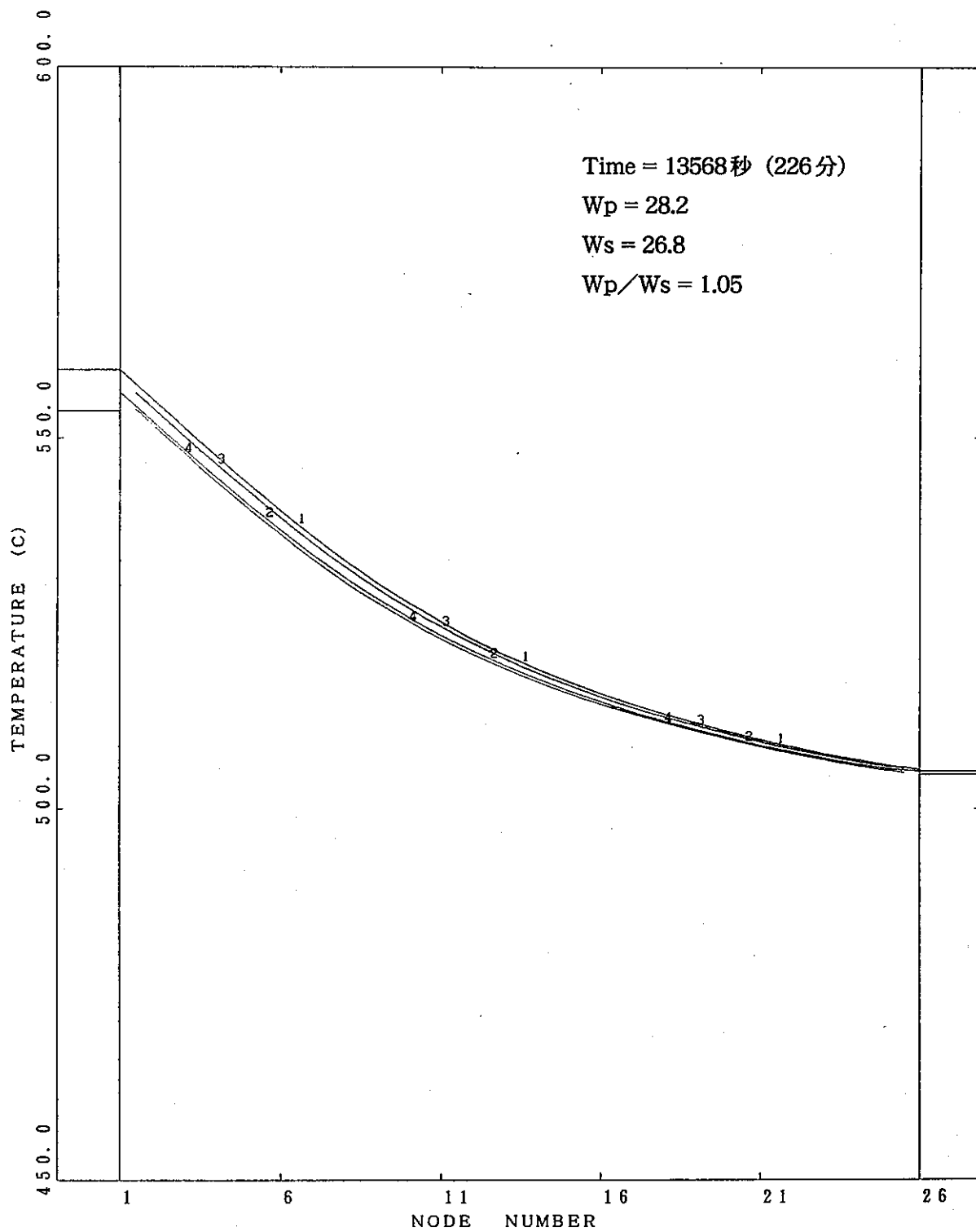
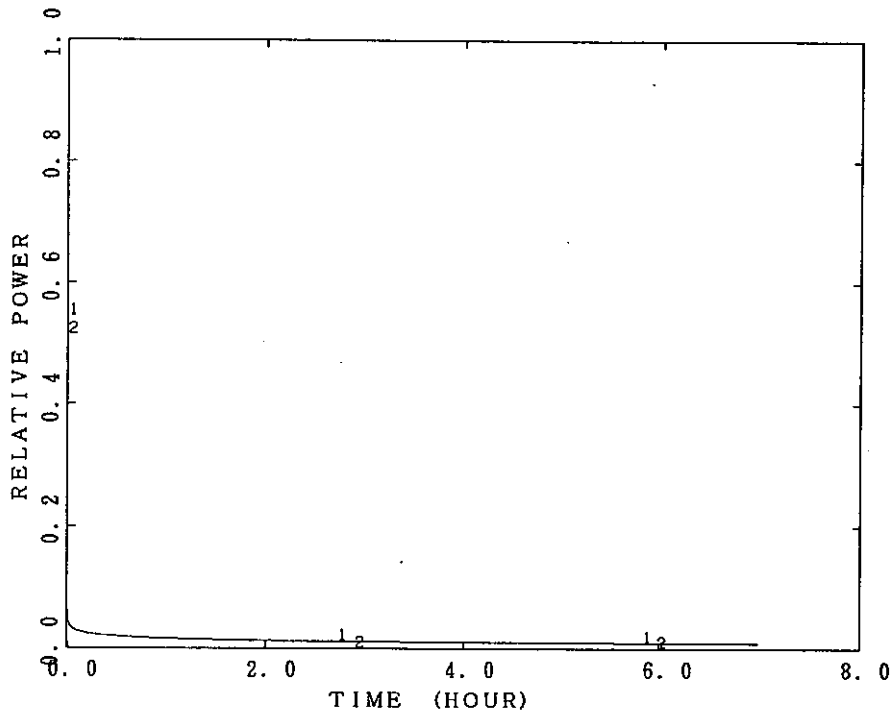
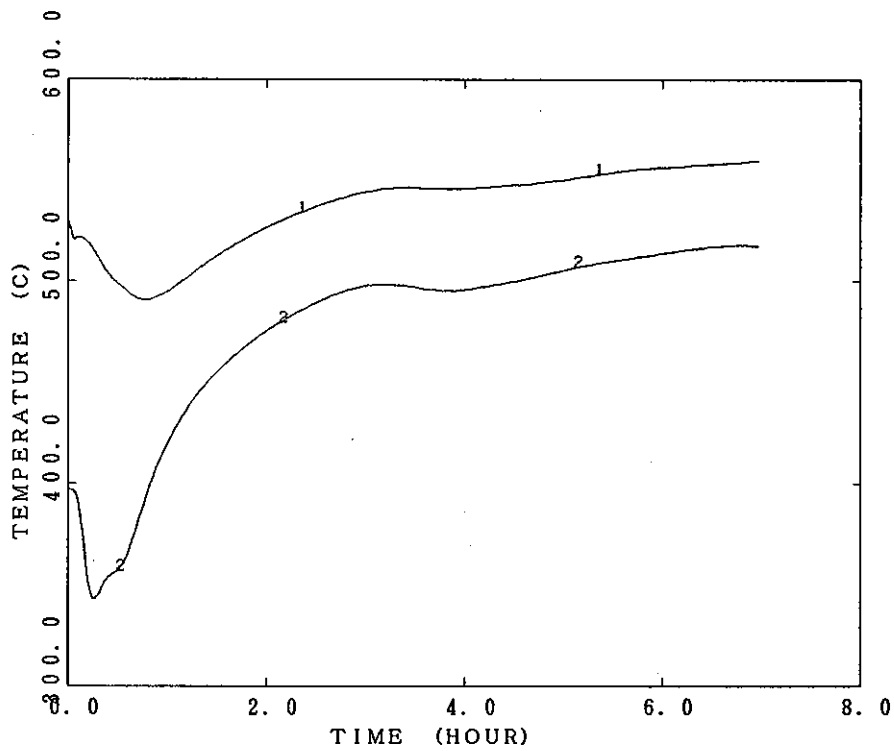


図3-2a ケースL-2のIHX温度分布図

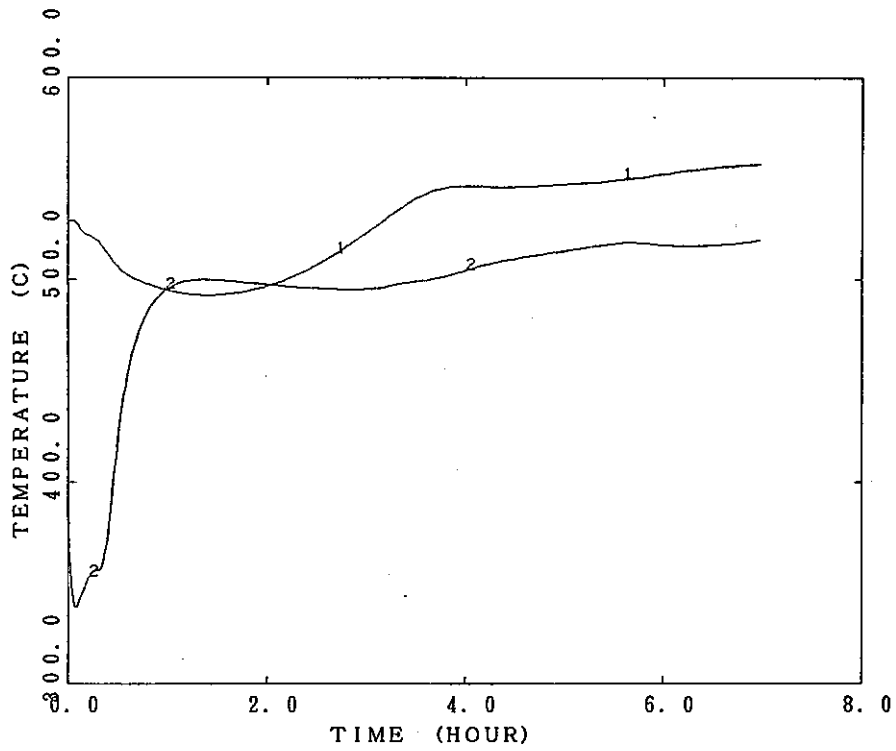


原子炉出力 (1全出力 2核分裂による出力)

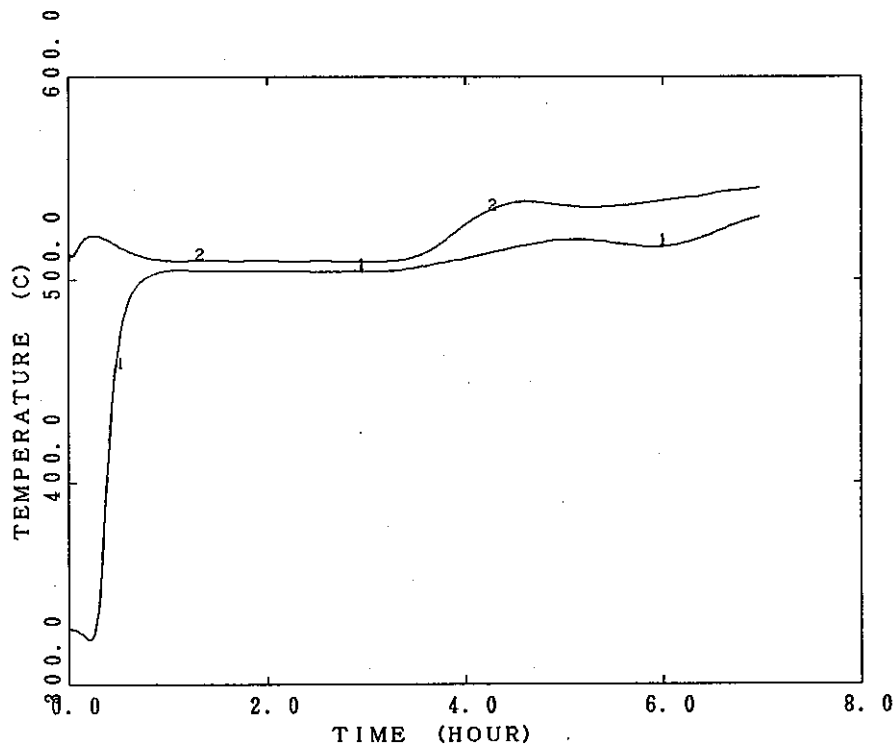


炉容器出入口温度 (1出口 2入口)

図3-3 ケースL-3のプロット出力

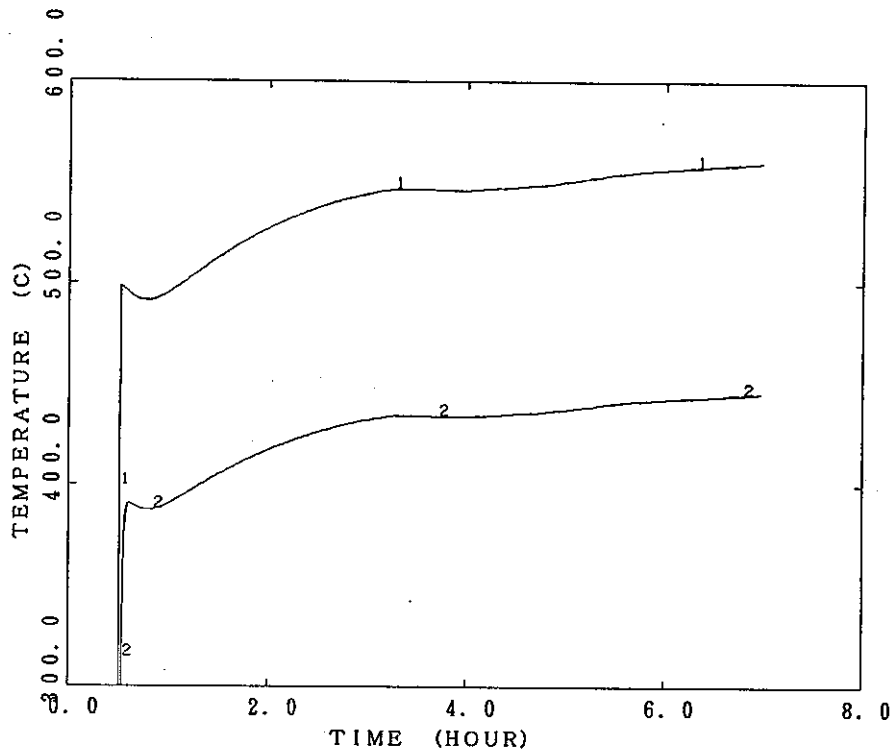


IHX1次系出入口温度 (1入口 2出口)

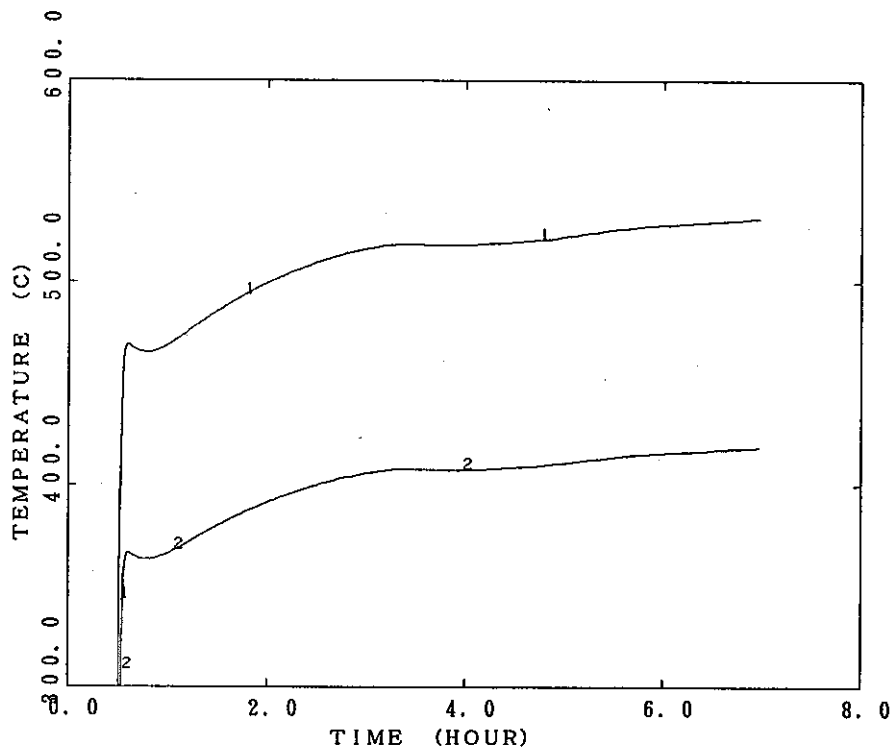


IHX2次系出入口温度 (1入口 2出口)

図3-3 ケースL-3のプロット出力 (つづき)

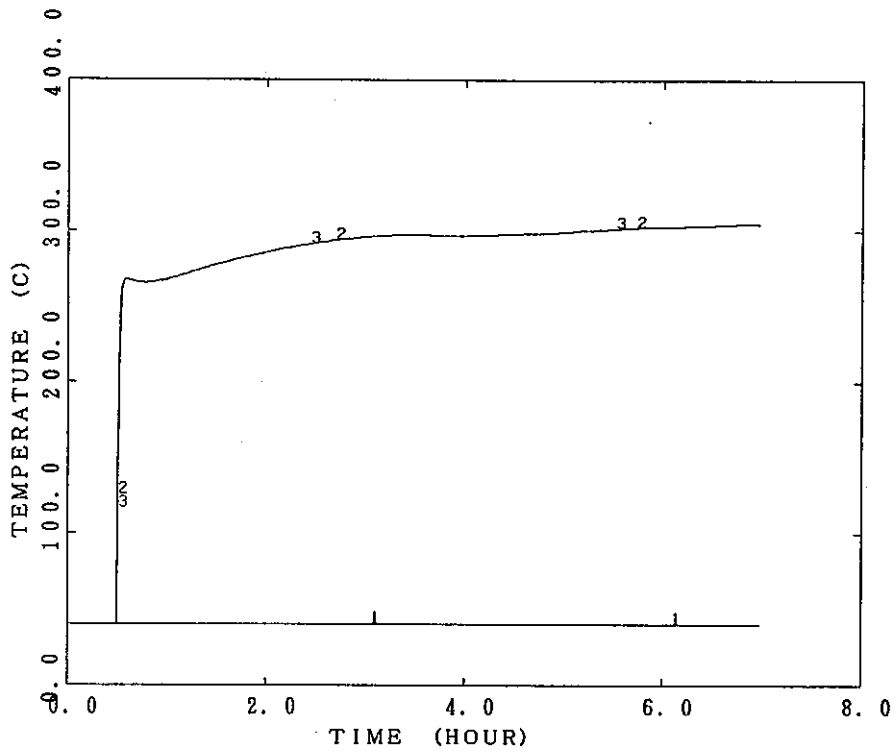


DRACS1次系出入口温度 (1入口 2出口)

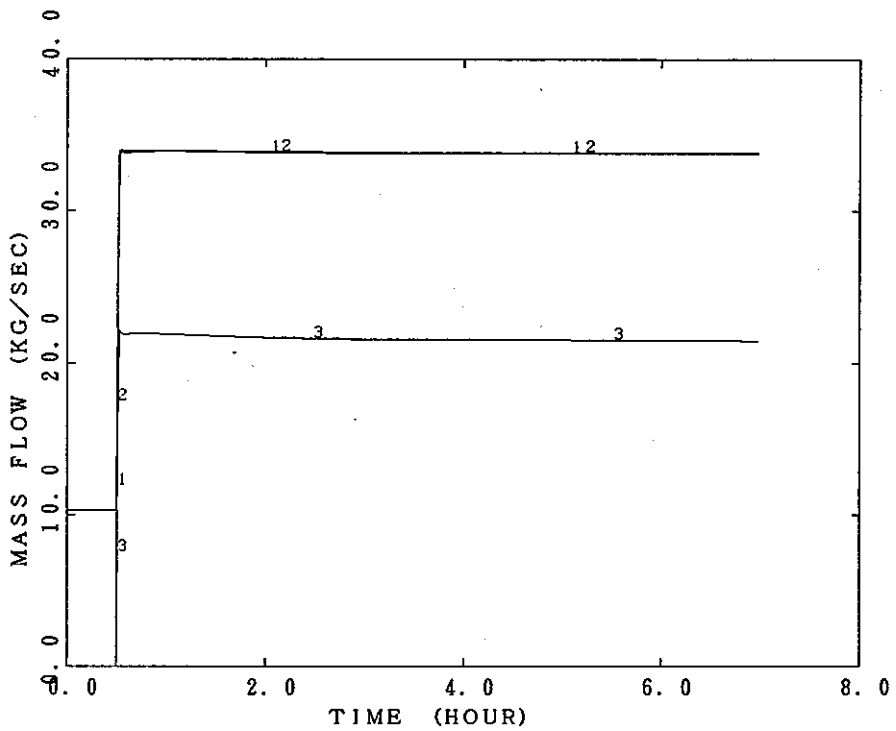


DRACS2次系出入口温度 (1出口 2入口)

図3-3 ケースL-3のプロット出力 (つづき)

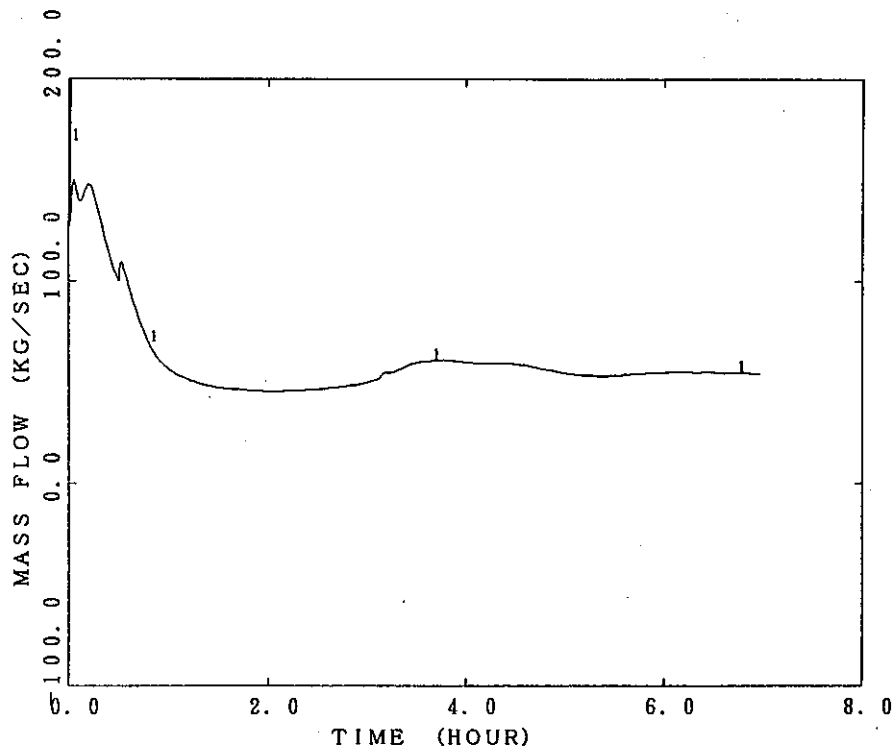


DRACS 空気冷却器空気温度 (1入口 2出口 3スタッフ)

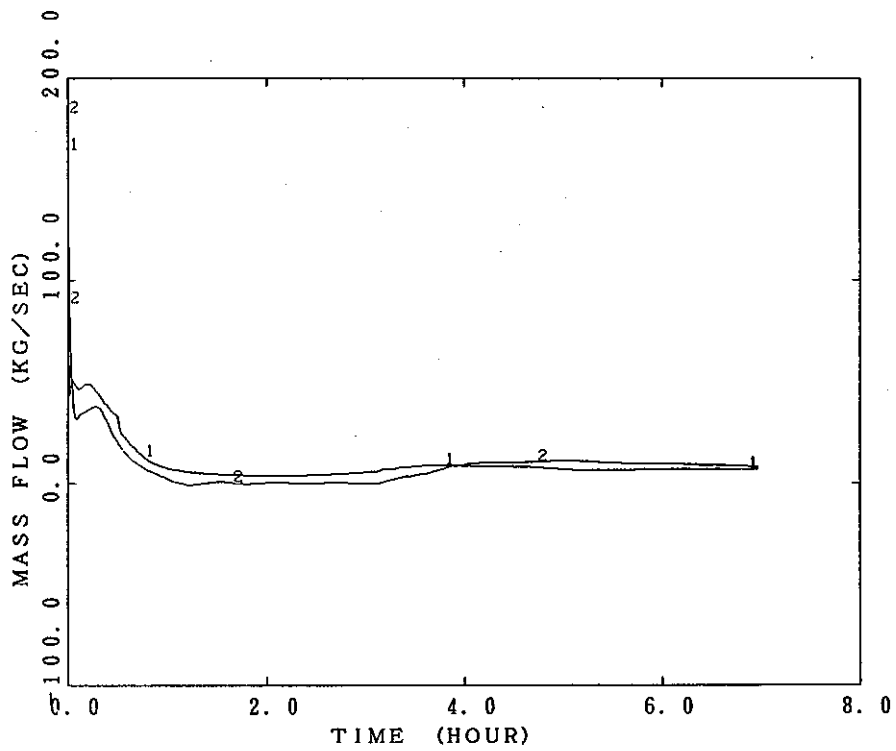


DRACS 流量 (1 1次系 2 2次系 3空気)

図3-3 ケースL-3のプロット出力 (つづき)

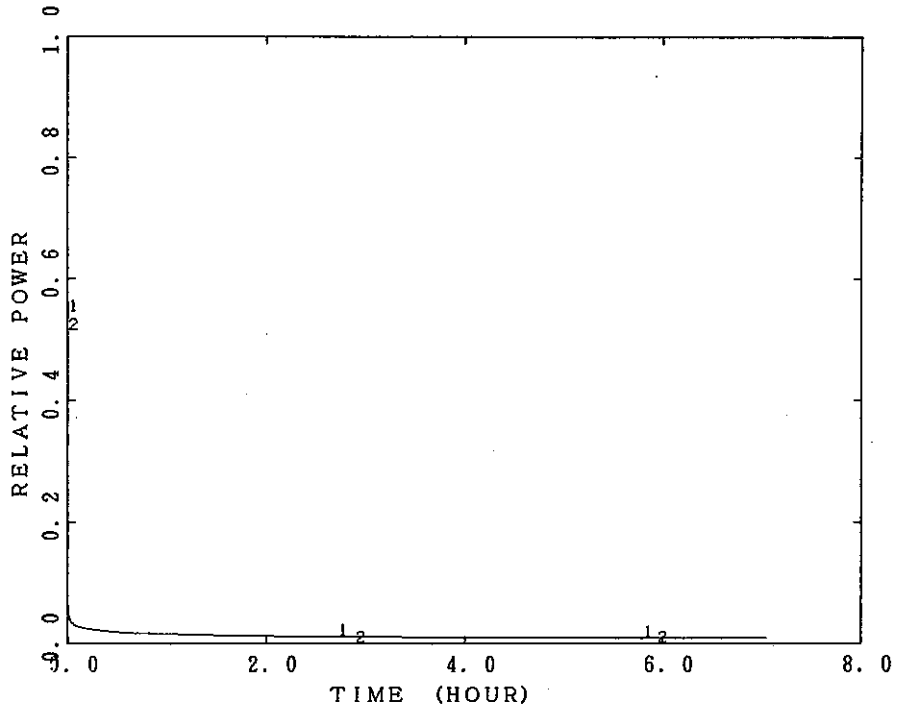


全炉心流量

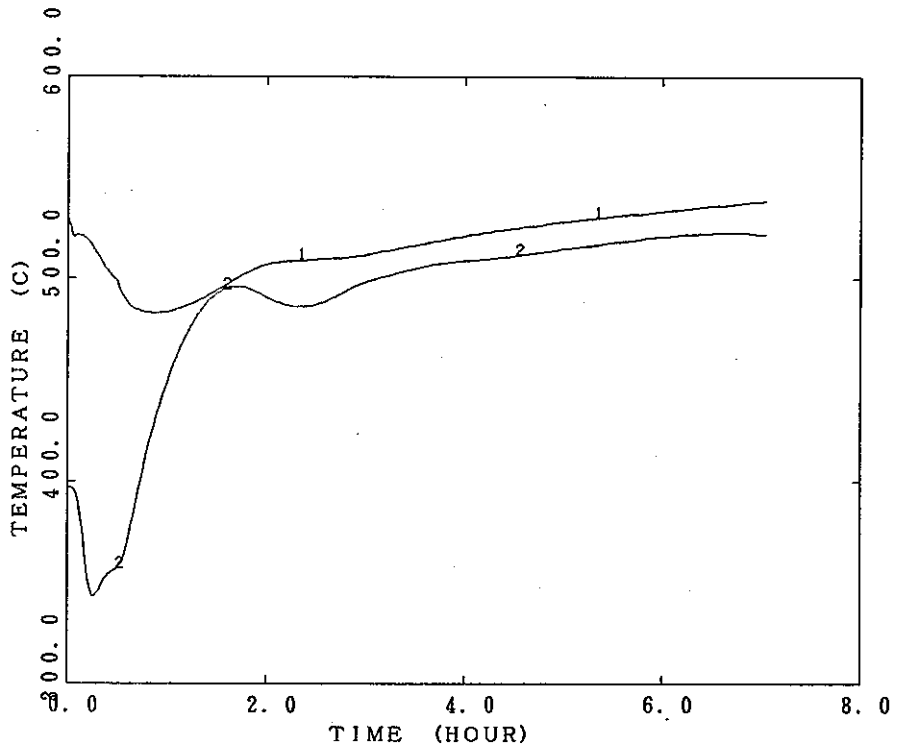


IHX1次系と2次系の流量

図3-3ケースL-3のプロット出力 (つづき)

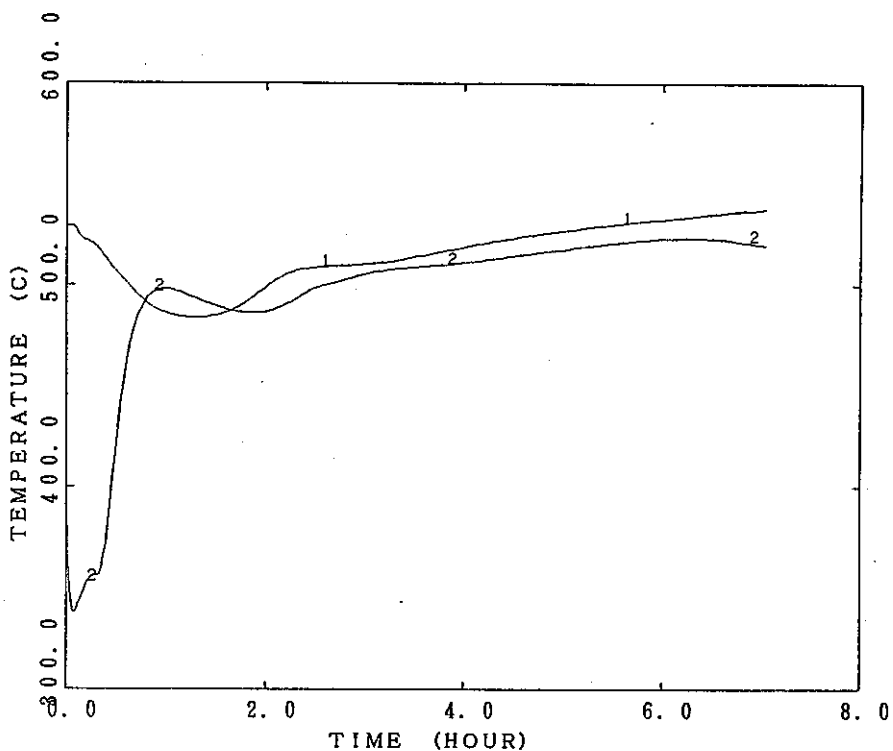


原子炉出力 (1全出力 2核分裂による出力)

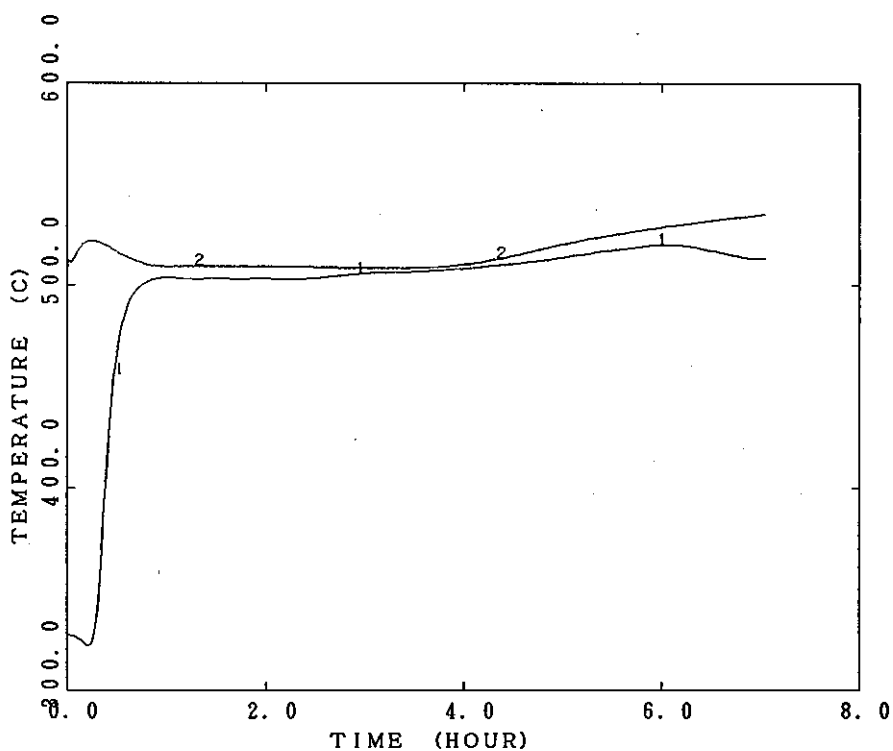


炉容器出入口温度 (1出口 2入口)

図3-4 ケースL-4のプロット出力

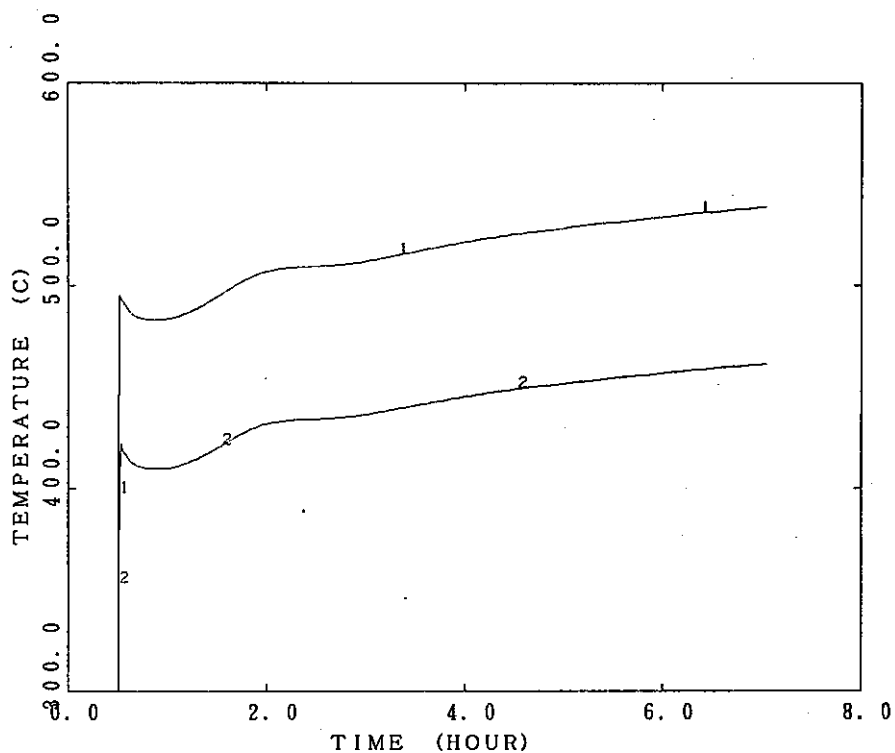


IHX1次系出入口温度 (1入口 2出口)

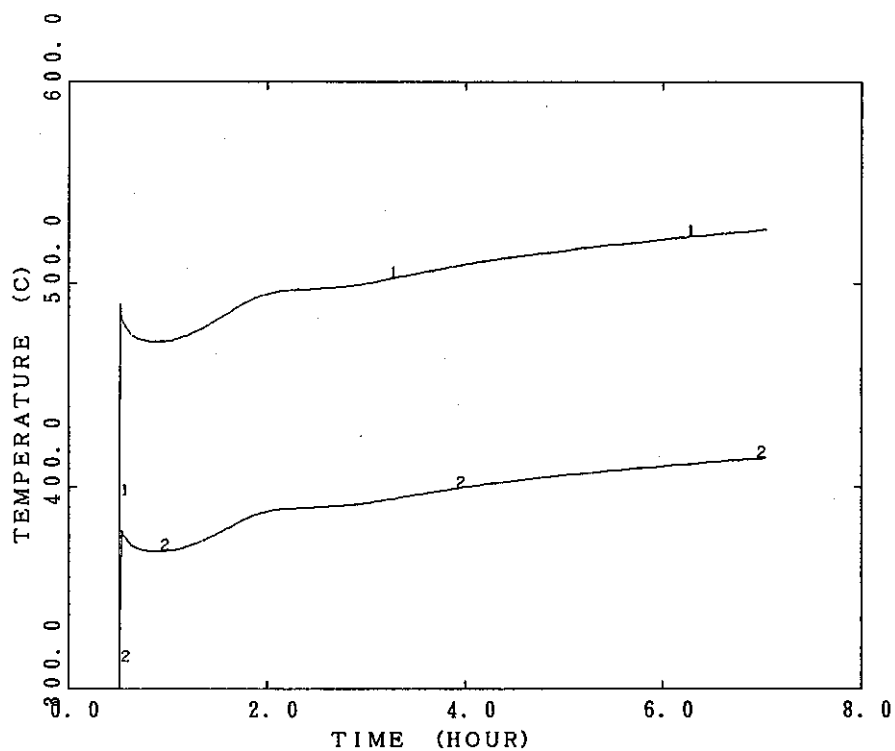


IHX2次系出入口温度 (1入口 2出口)

図3-4 ケースL-4のプロット出力 (つづき)

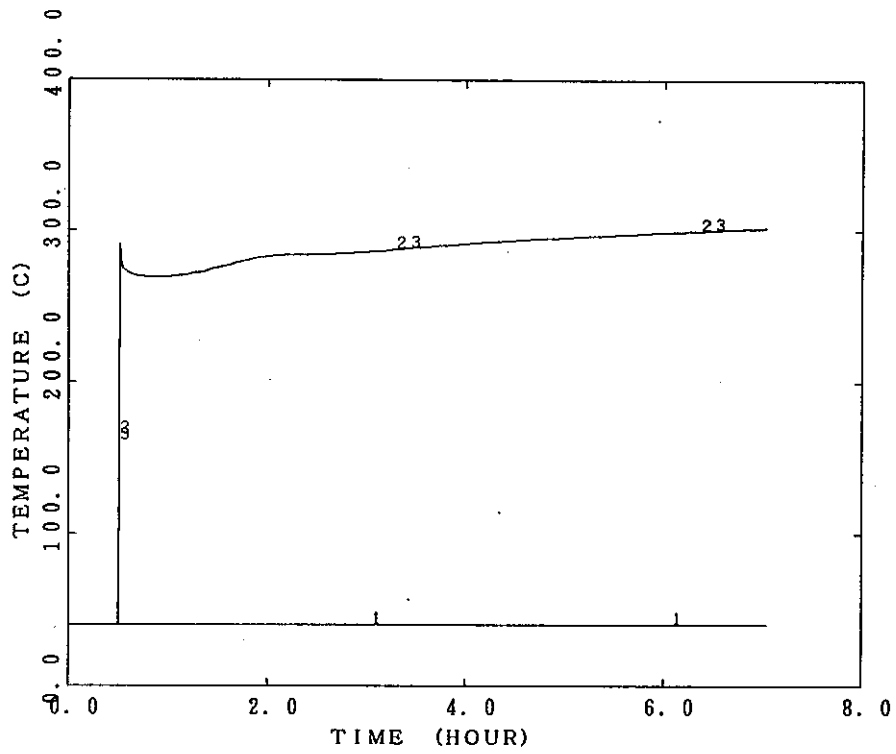


DRACS 1次系出入口温度 (1入口 2出口)

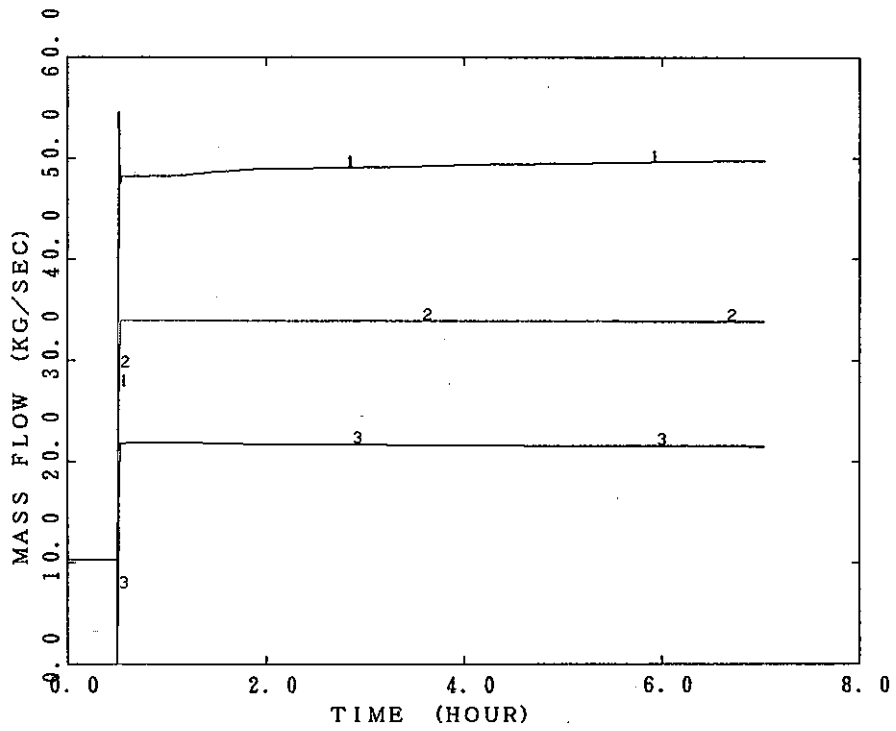


DRACS 2次系出入口温度 (1出口 2入口)

図3-4 ケースL-4のプロット出力 (つづき)

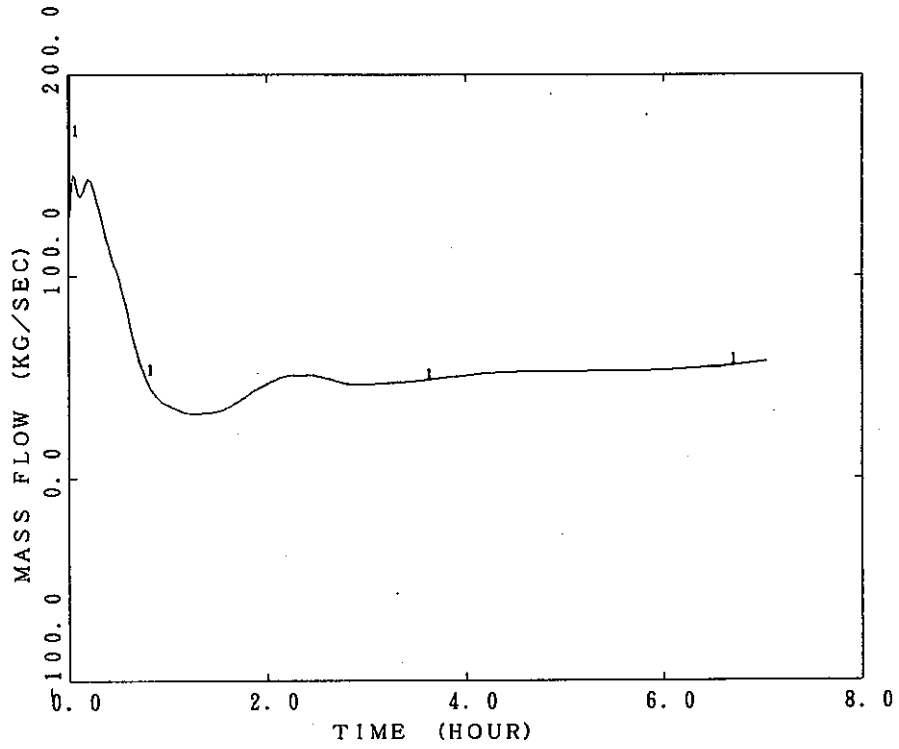


DRACS 空気冷却器空気温度 (1入口 2出口 3スタッフ)

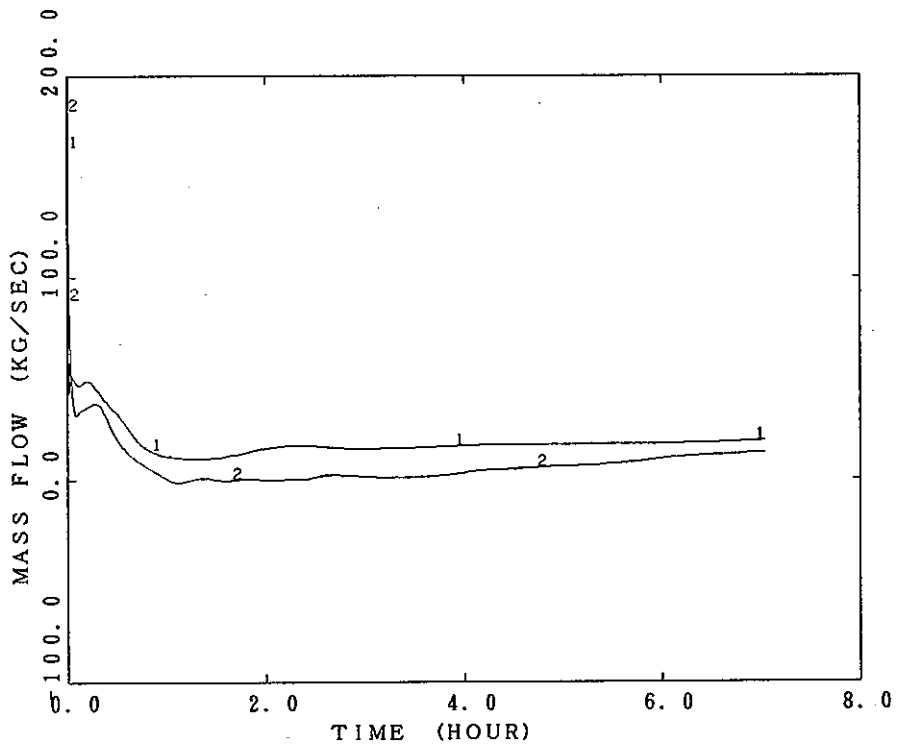


DRACS 流量 (1 1次系 2 2次系 3空気)

図3-4 ケースL-4のプロット出力 (つづき)

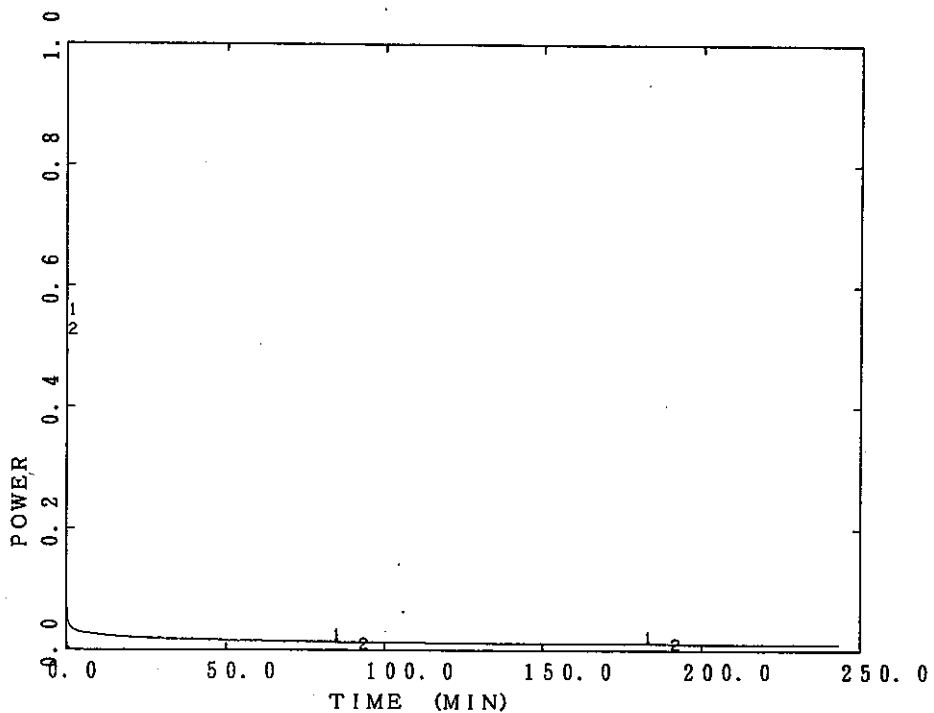


全炉心流量

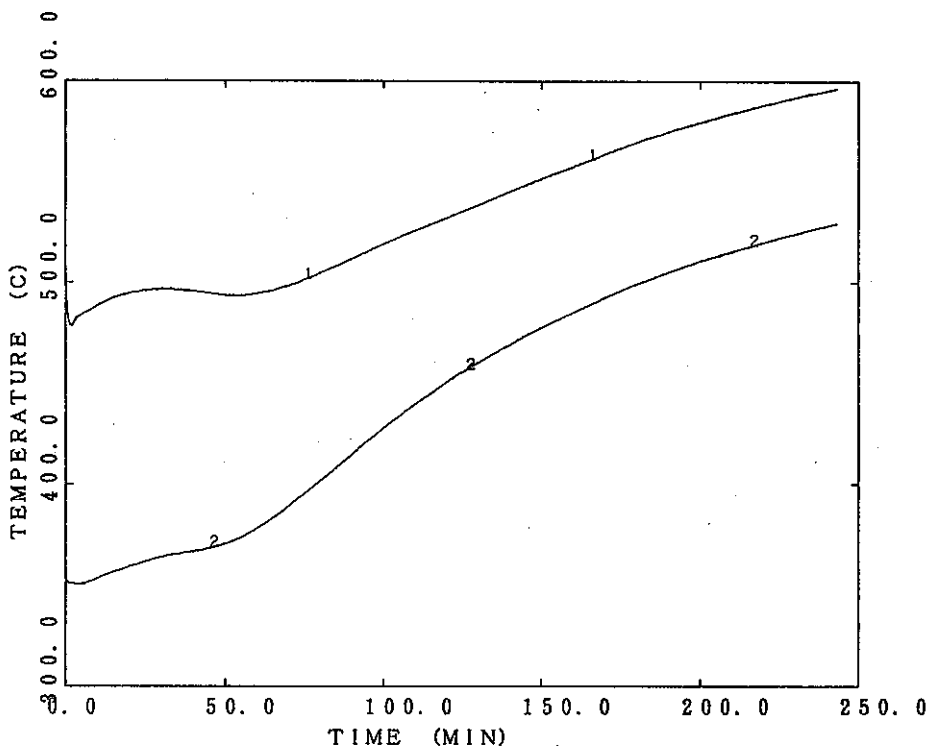


IHX1 次系と 2 次系の流量

図3-4 ケースL-4のプロット出力 (つづき)

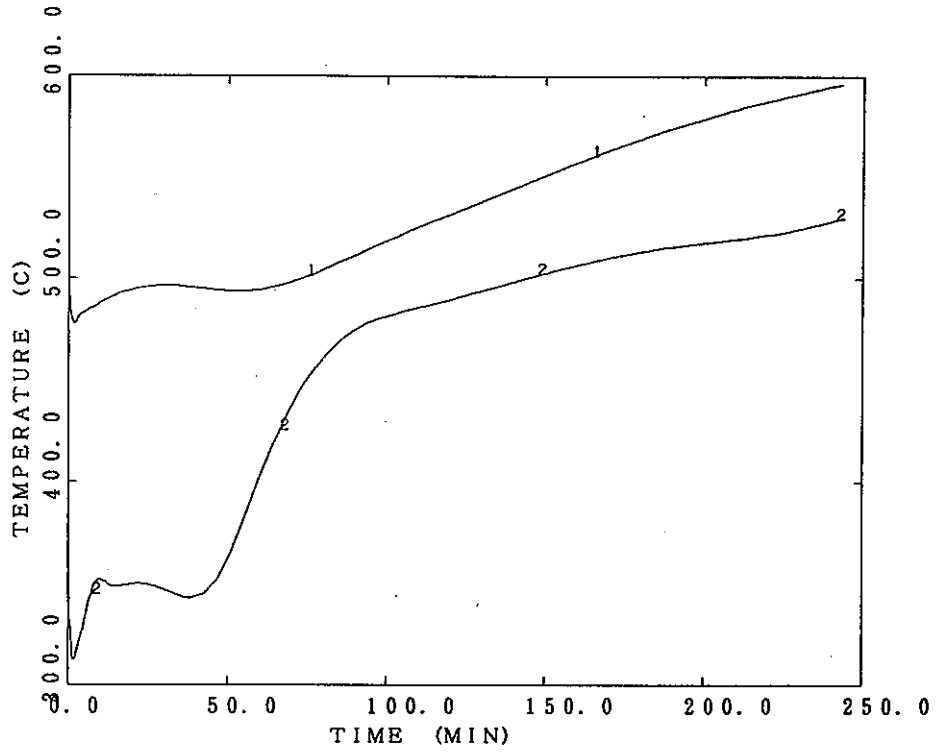


原子炉出力 (1全出力 2核分裂による出力)

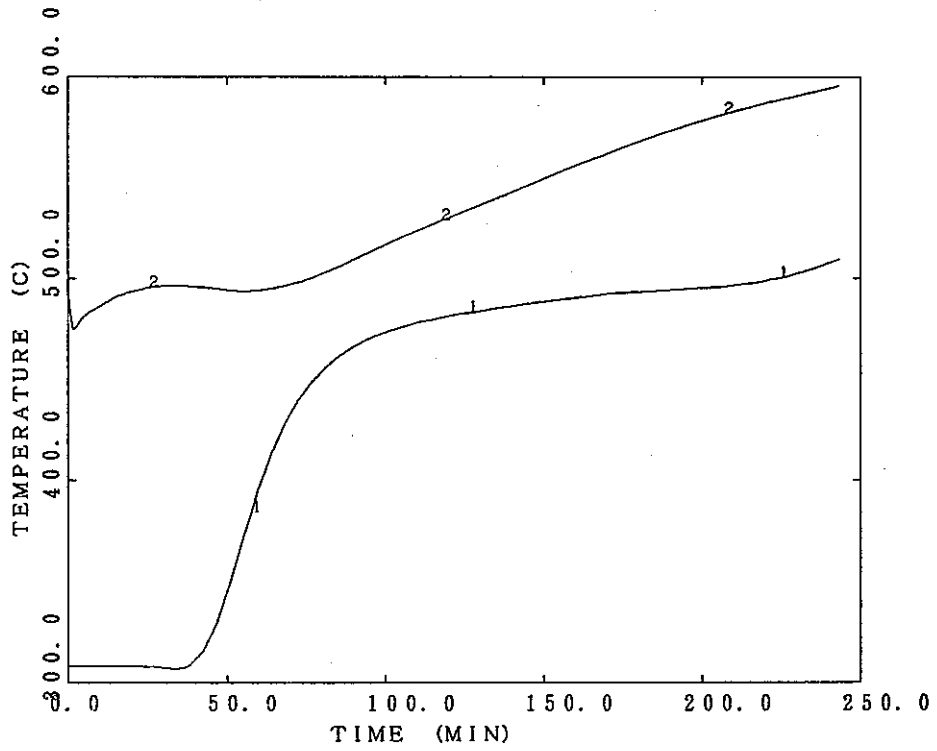


ポットプール (1)、コールドプール (2) の温度

図3-5 ケースP-5のプロット出力

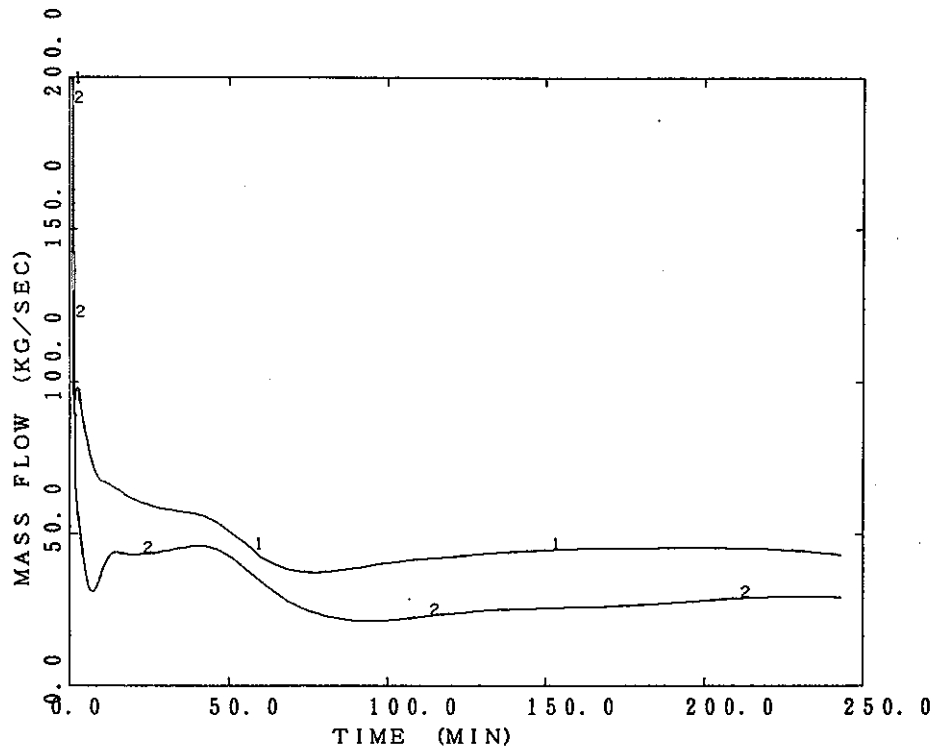


IHX1次系出入口温度 (1入口 2出口)

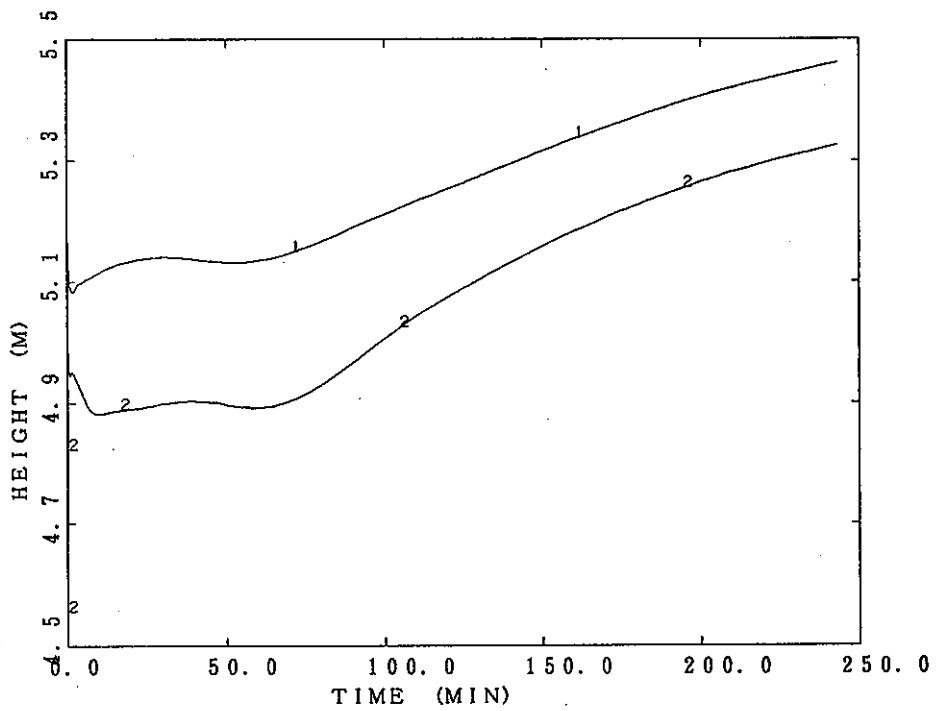


IHX2次系出入口温度 (1入口 2出口)

図3-5 ケースP-5のプロット出力 (つづき)



IHX1 次系と 2 次系の流量



ホットプール (1)、コールドプール (2) の液位

図3-5 ケースP-5のプロット出力 (つづき)

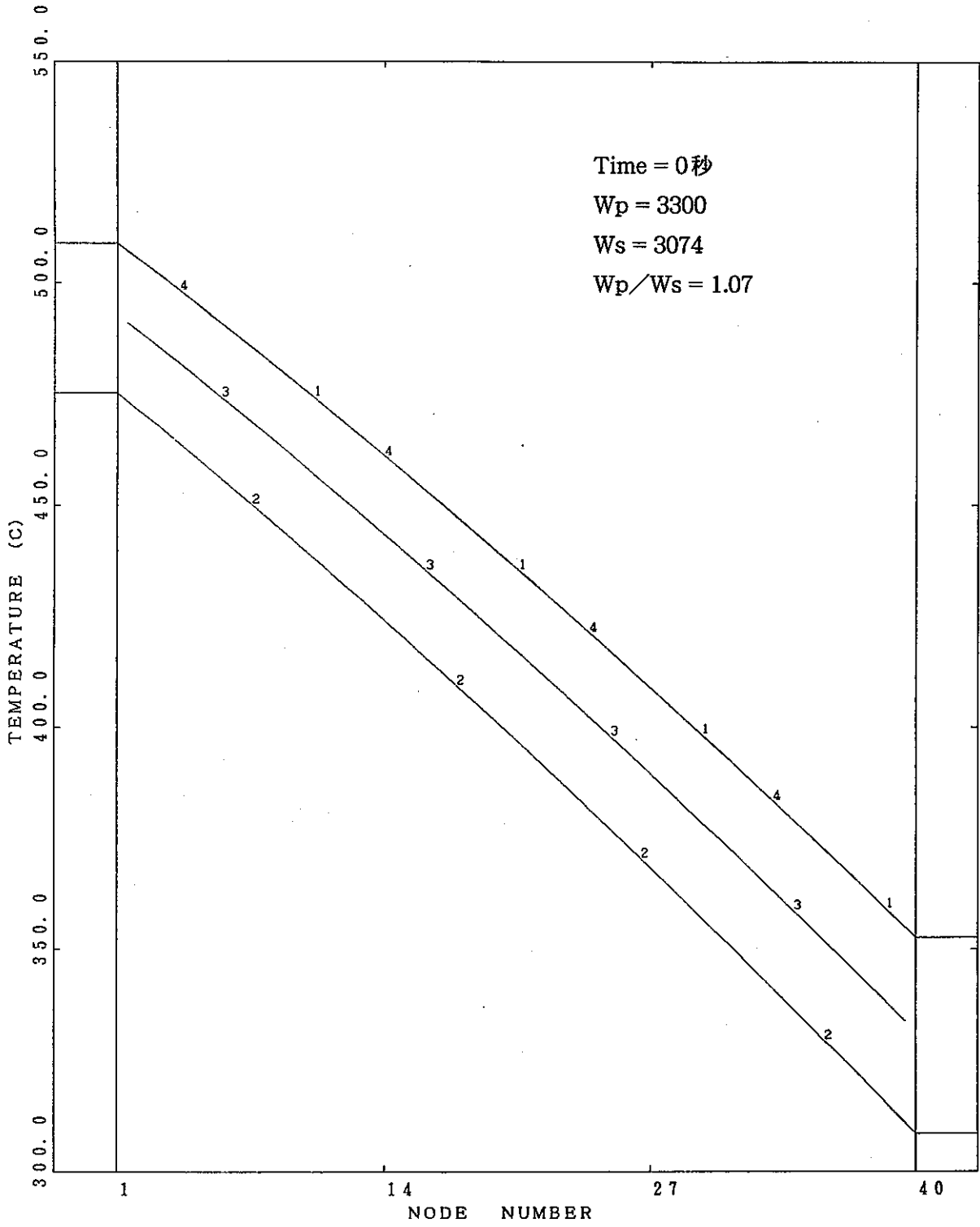


図3-5a ケースP-5のIHX温度分布図

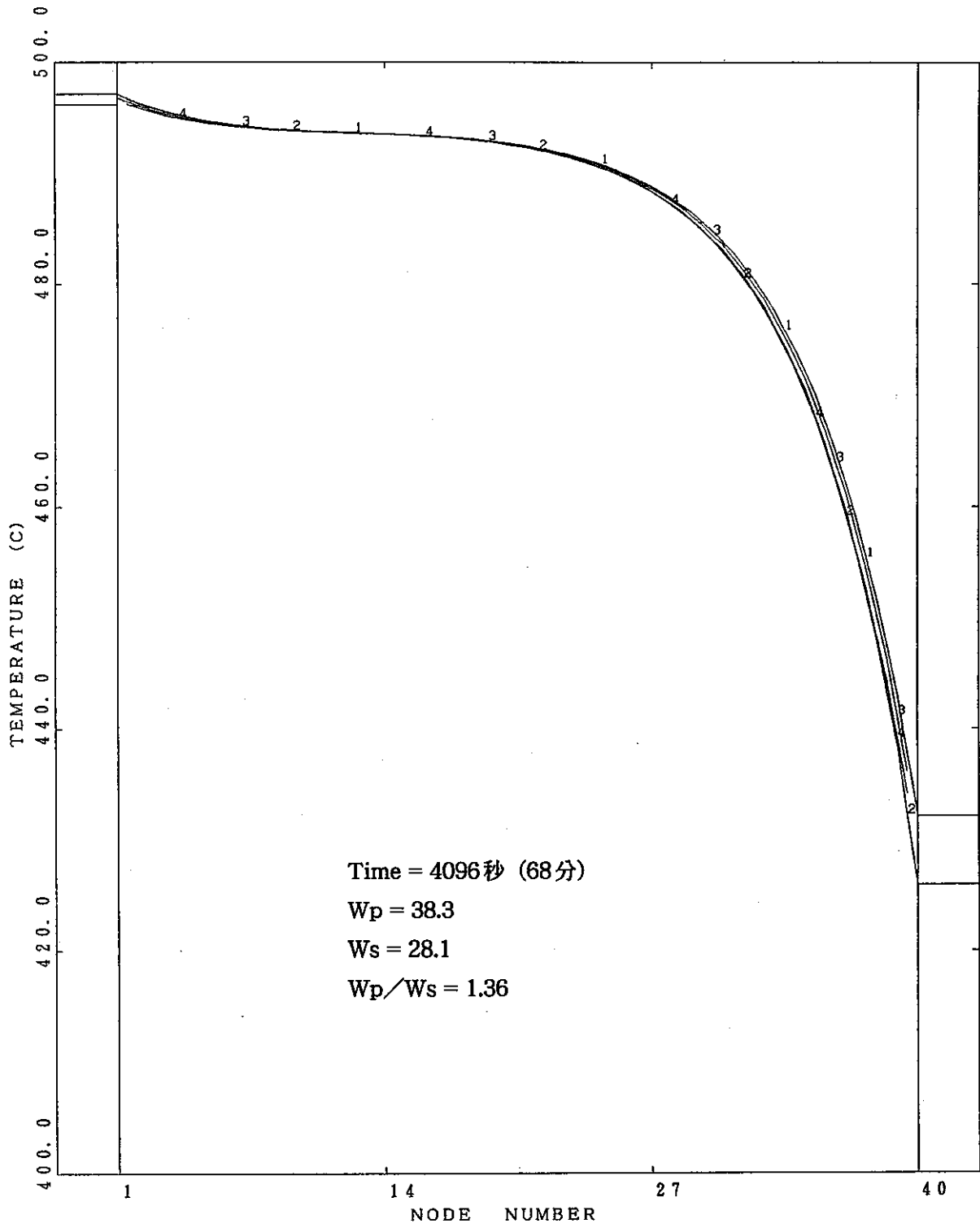


図3-5a ケースP-5のIHX温度分布図

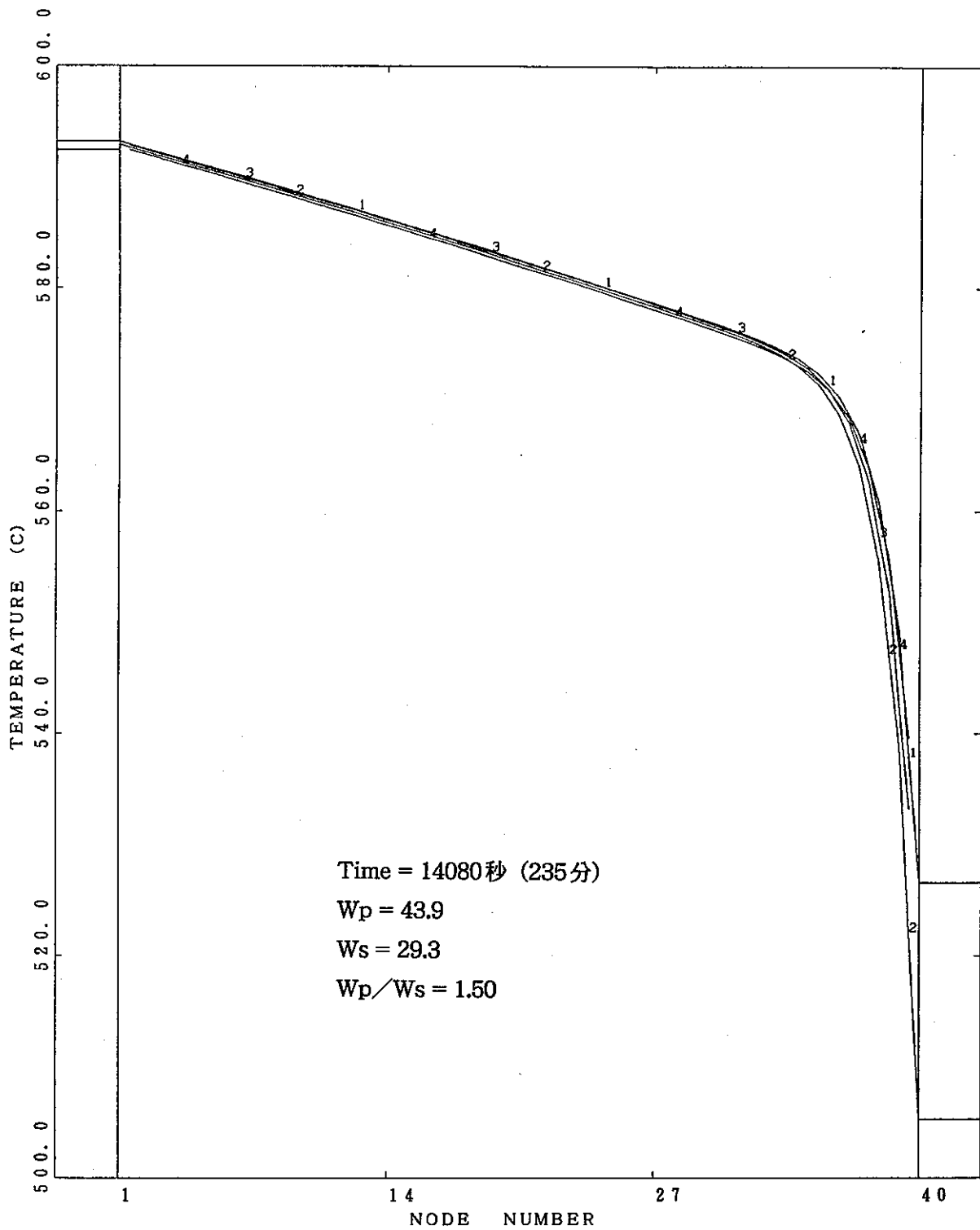
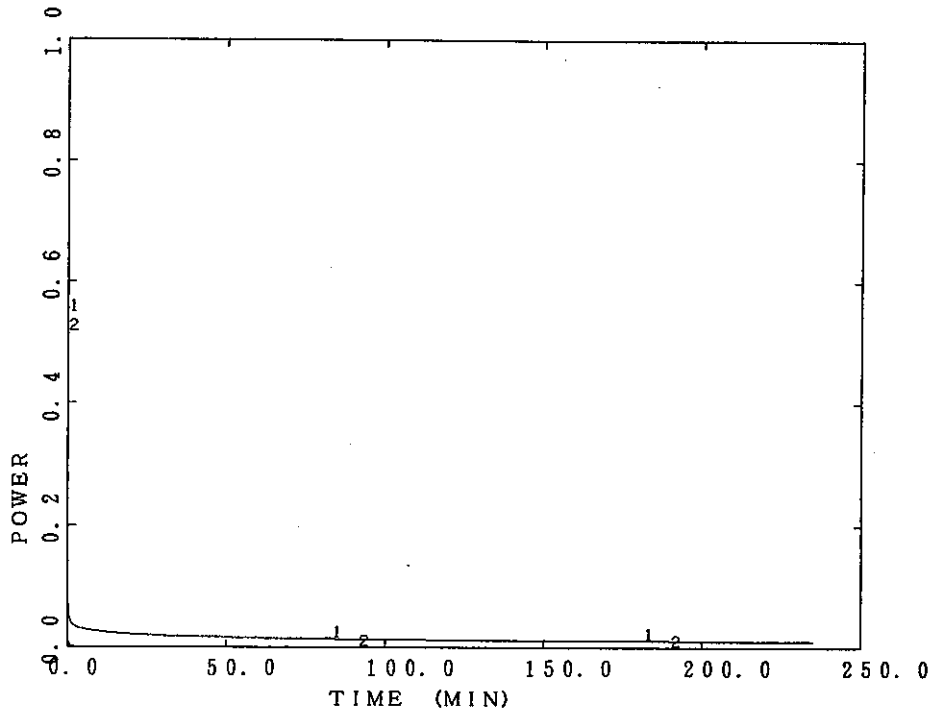
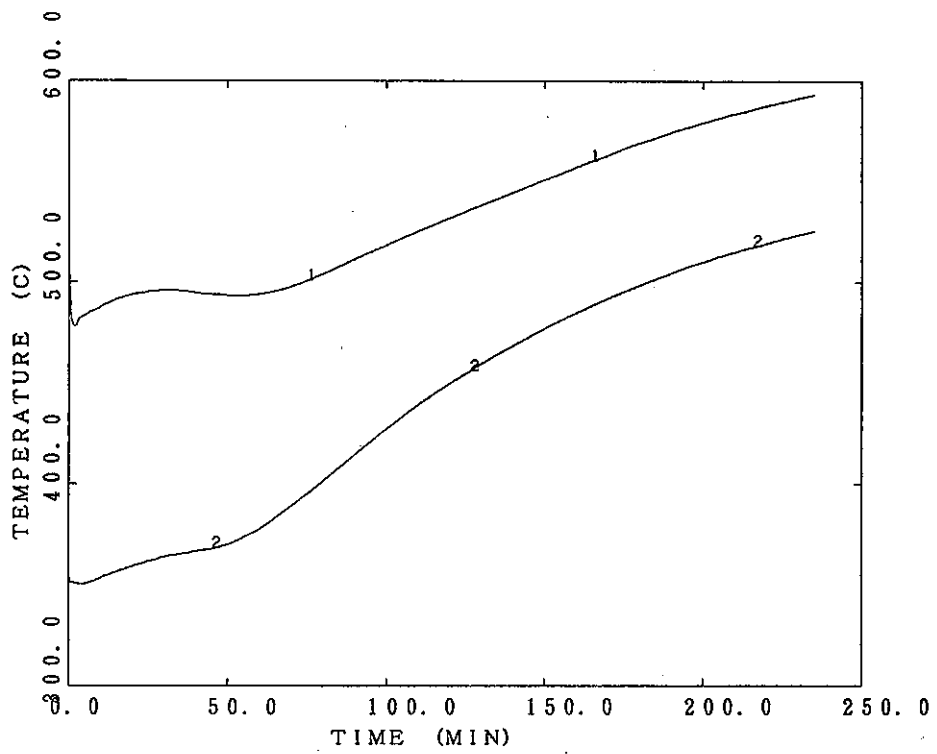


図3-5a ケースP-5のIHX温度分布図 (つづき)

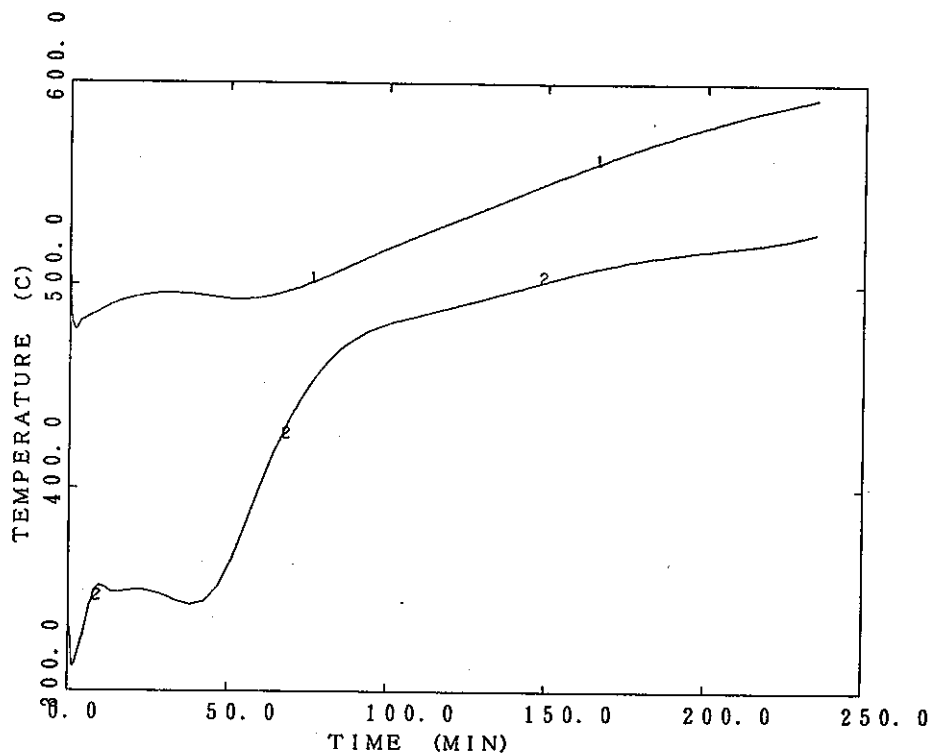


原子炉出力 (1全出力 2核分裂による出力)

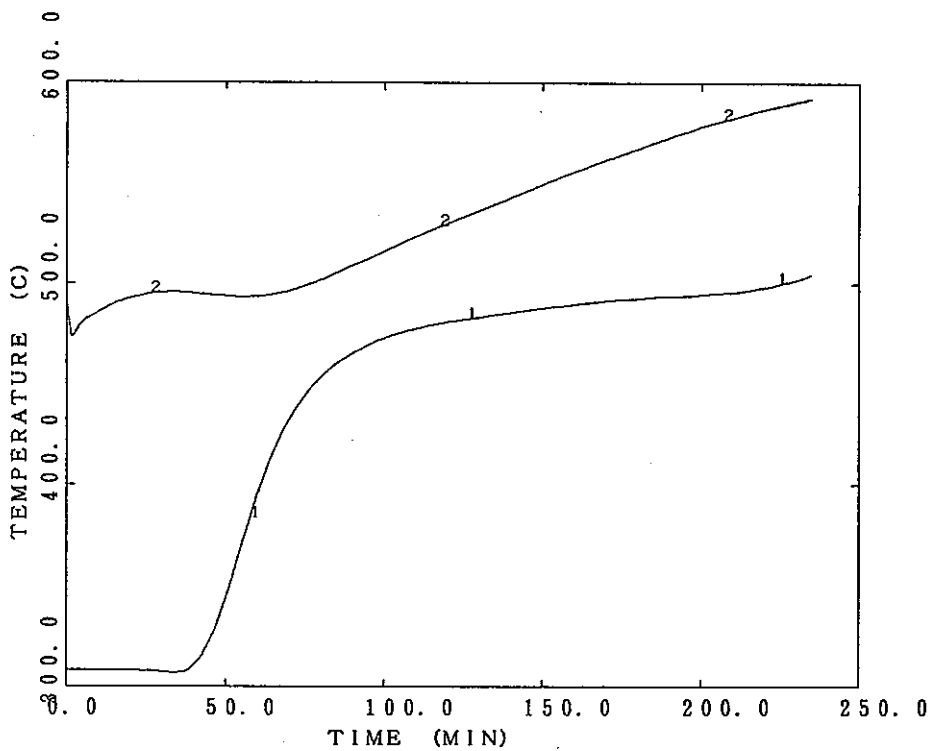


ホットプール (1)、コールドプール (2) の温度

図3-6 ケースP-6のプロット出力

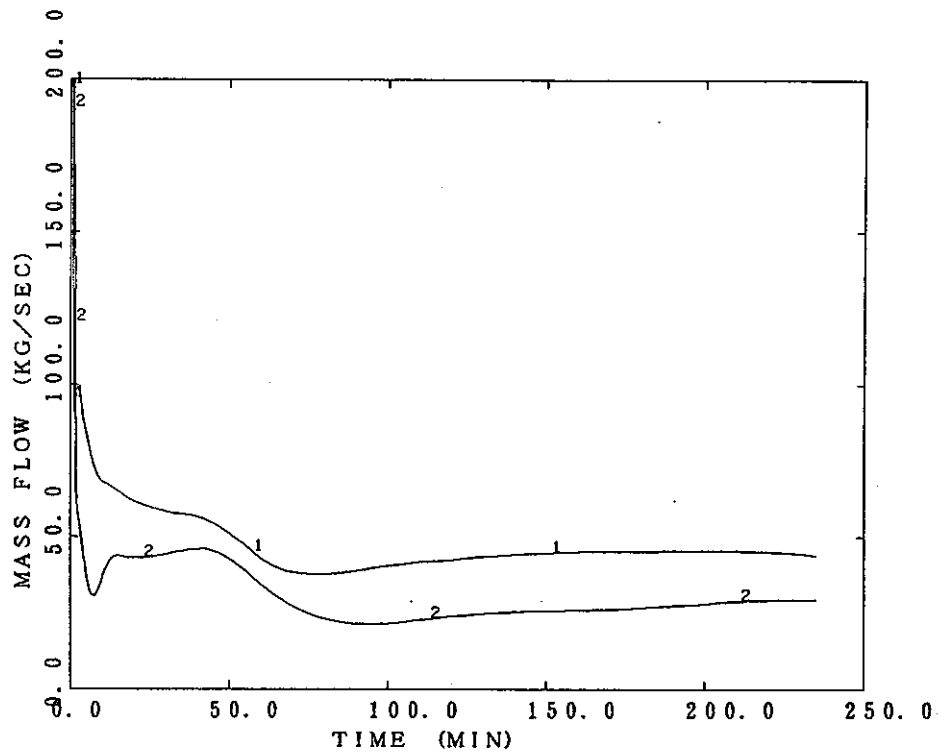


IHX1次系出入口温度 (1入口 2出口)

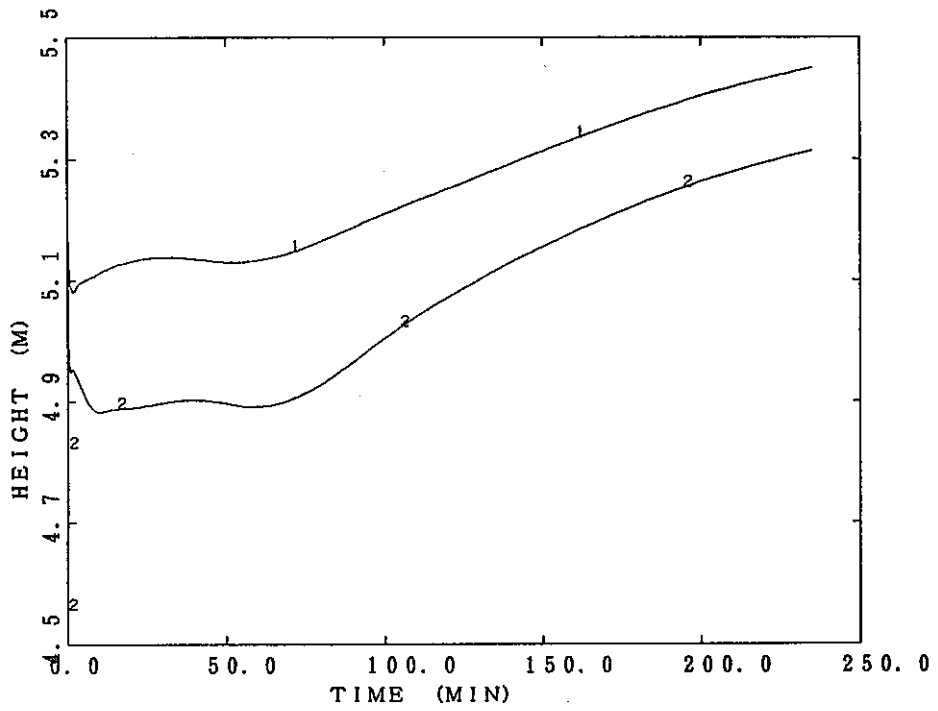


IHX2次系出入口温度 (1入口 2出口)

図3-6 ケースP-6のプロット出力 (つづき)



IHX1 次系と 2 次系の流量



ホットプール (1)、コールドプール (2) の液位

図3-6 ケースP-6のプロット出力 (つづき)

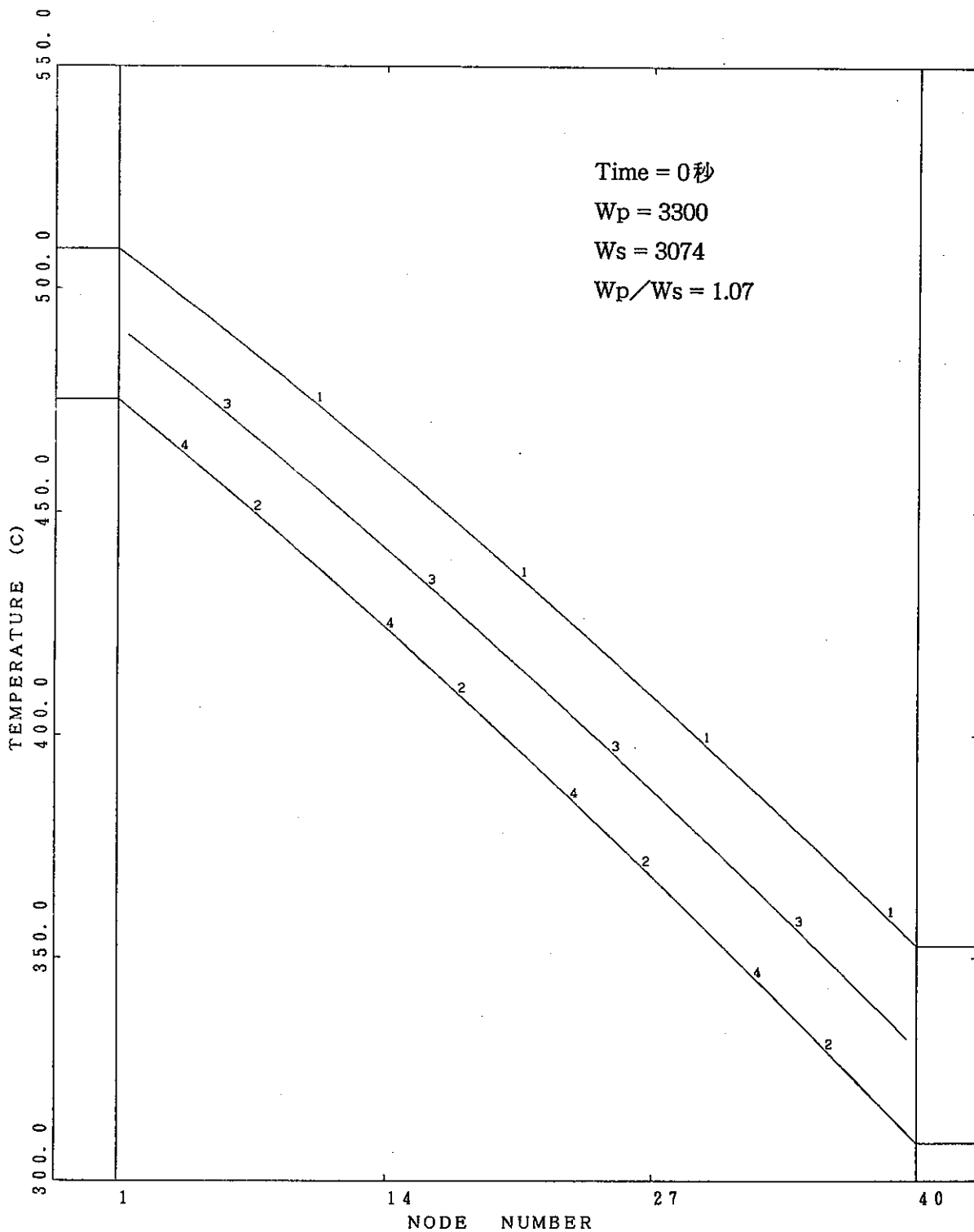


図3-6a ケースP-6のIHX温度分布図

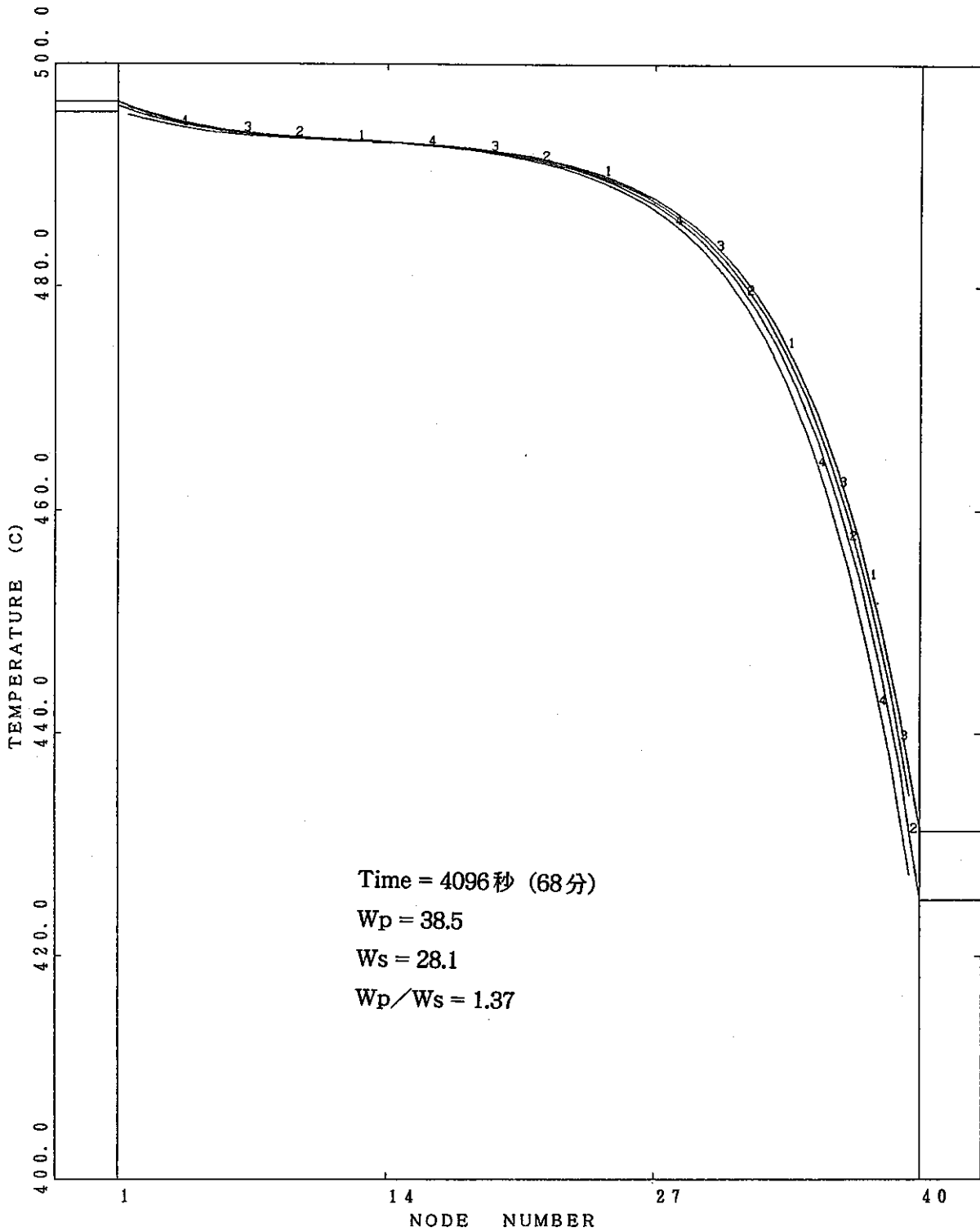


図3-6a ケースP-6のIHX温度分布図 (つづき)

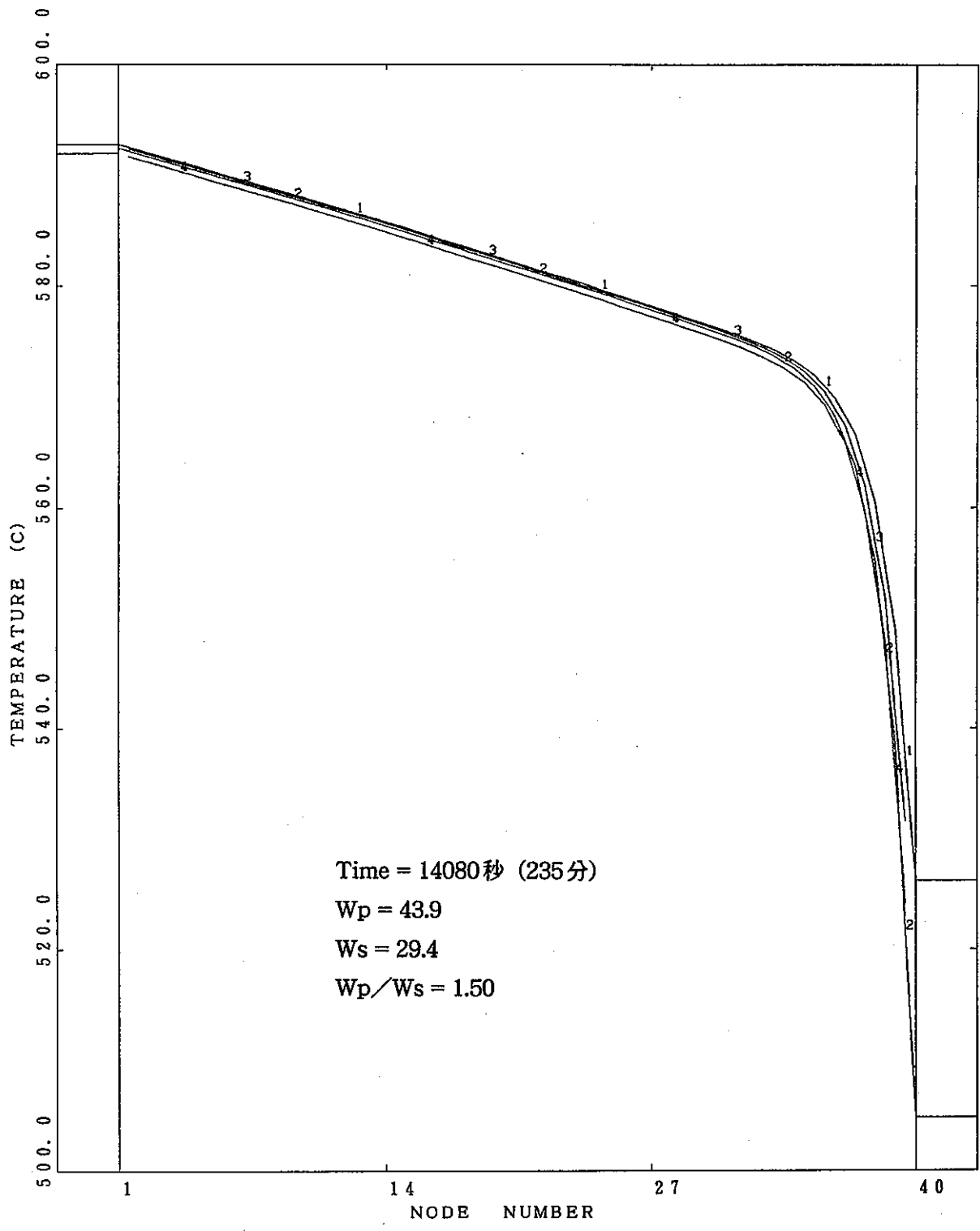
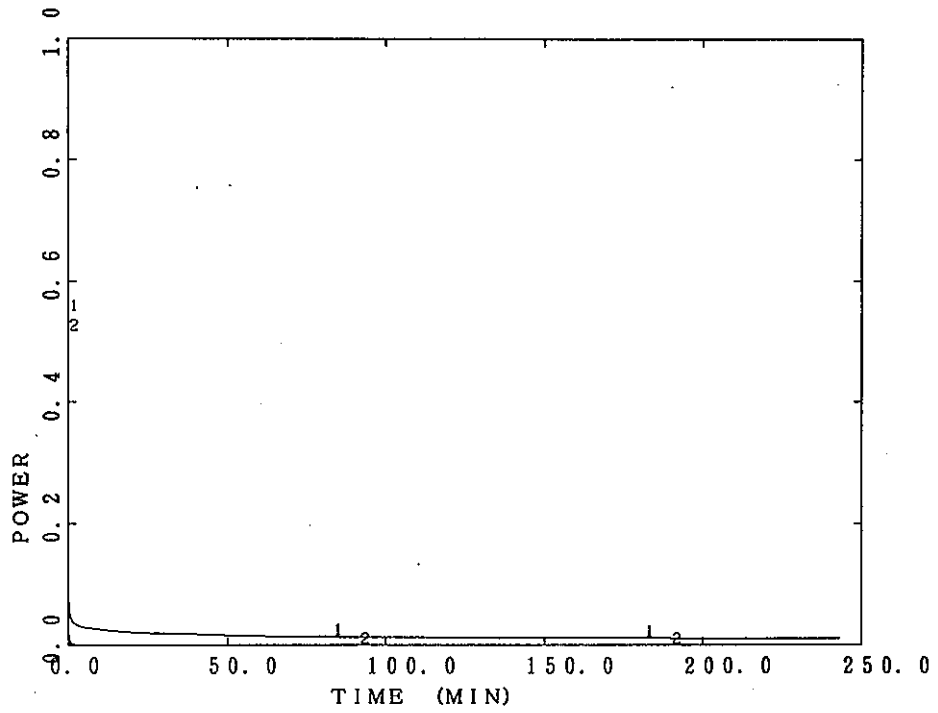
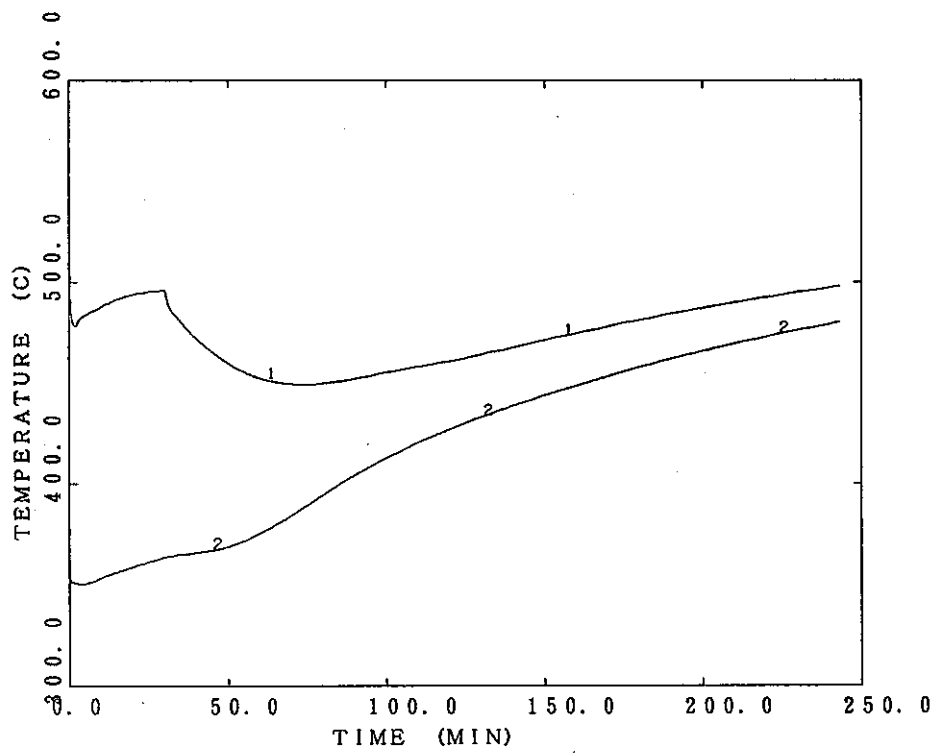


図3-6a ケースP-6のHX温度分布図 (つづき)

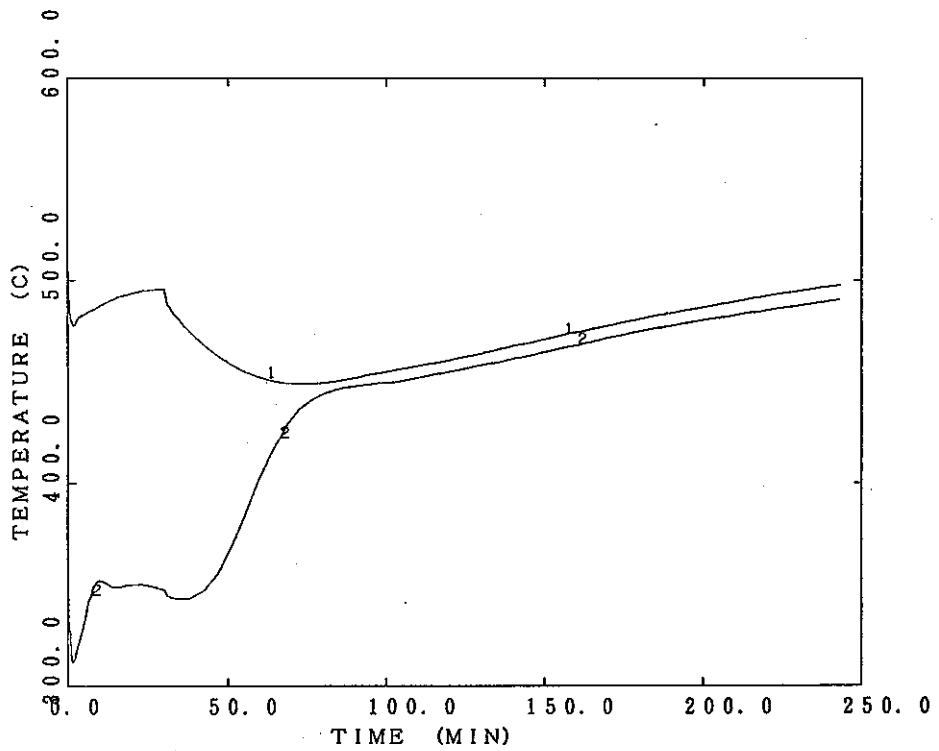


原子炉出力 (1全出力 2核分裂による出力)

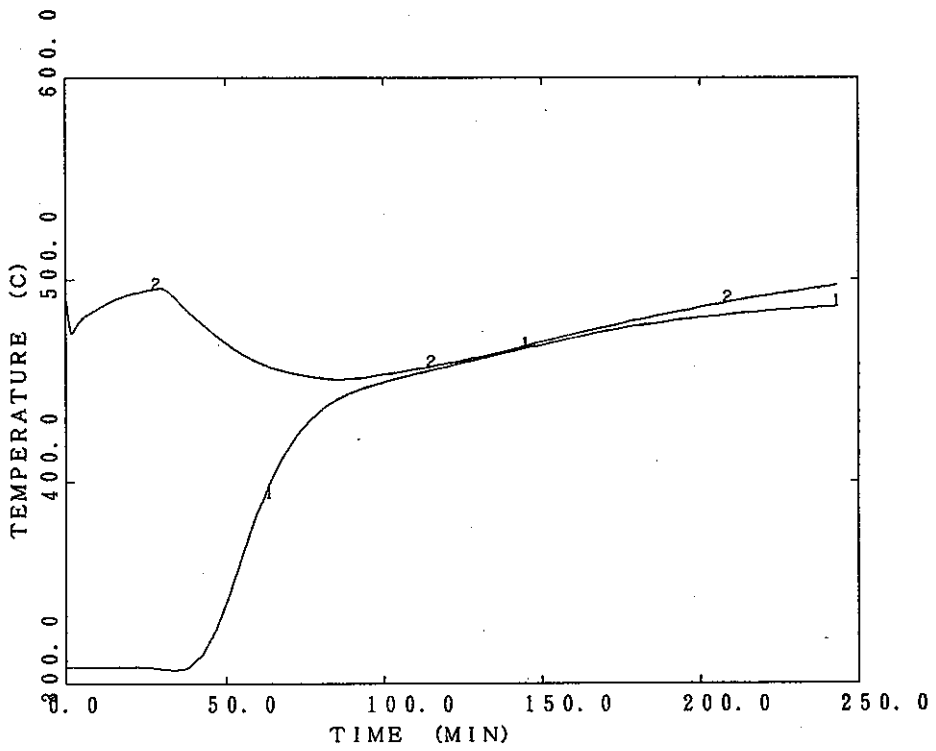


ホットプール (1)、コールドプール (2) の温度

図3-7 ケースP-7のプロット出力

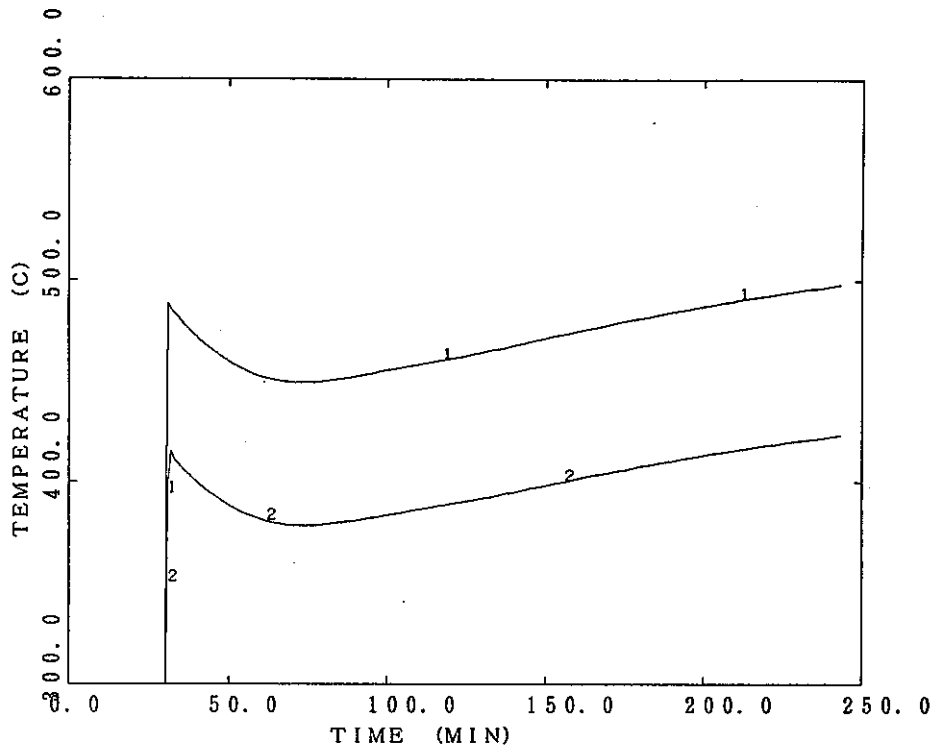


IHX1次系出入口温度 (1入口 2出口)

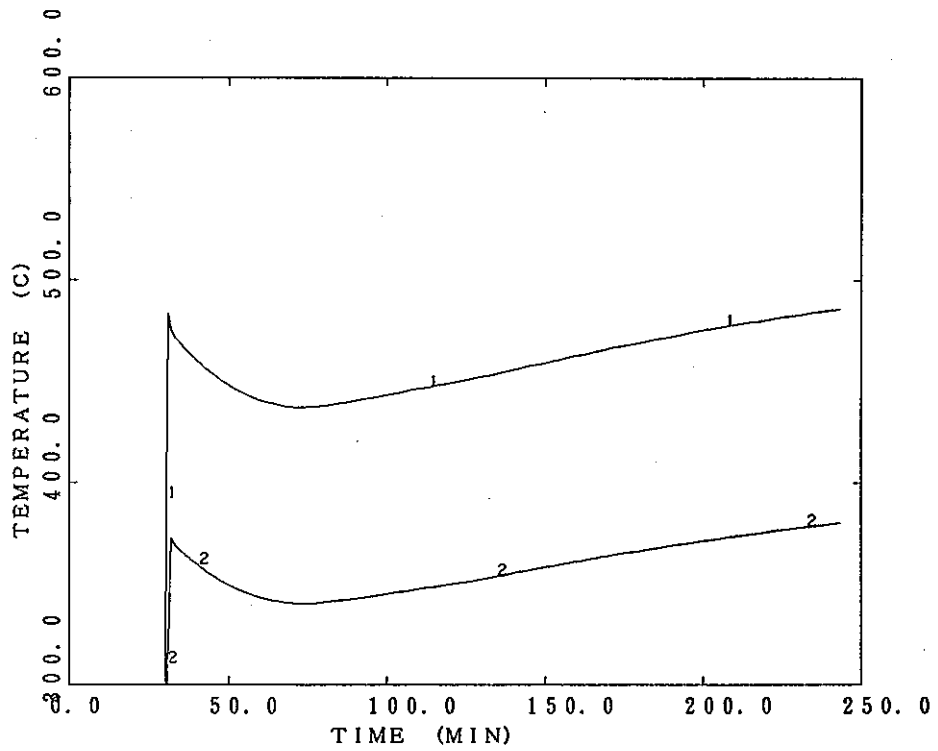


IHX2次系出入口温度 (1入口 2出口)

図3-7 ケースP-7のプロット出力 (つづき)

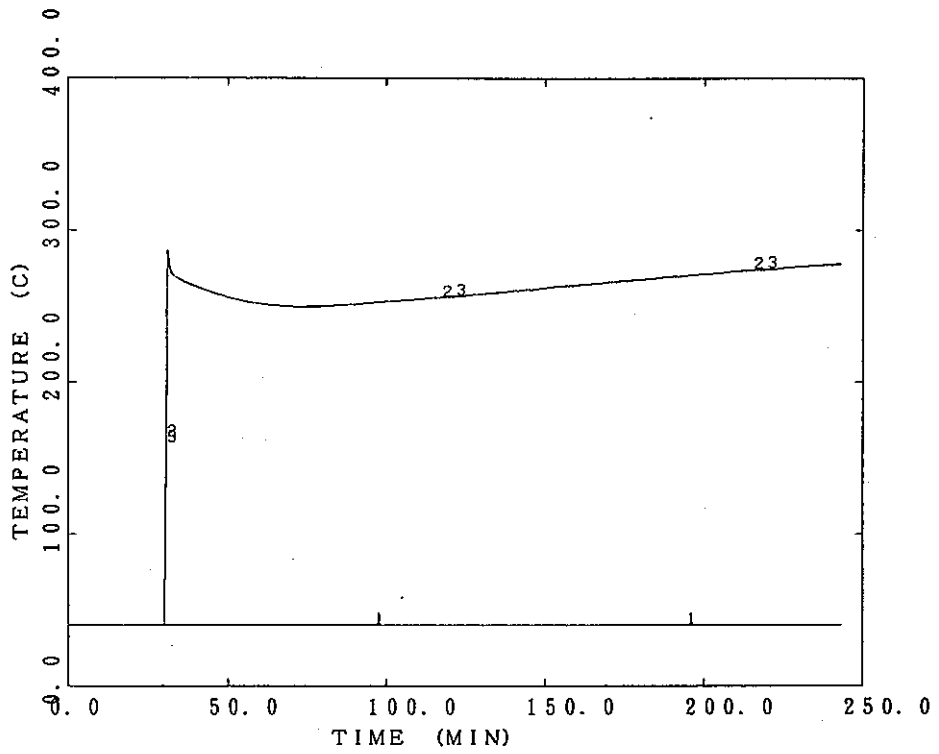


DRACS 1次系出入口温度 (1入口 2出口)

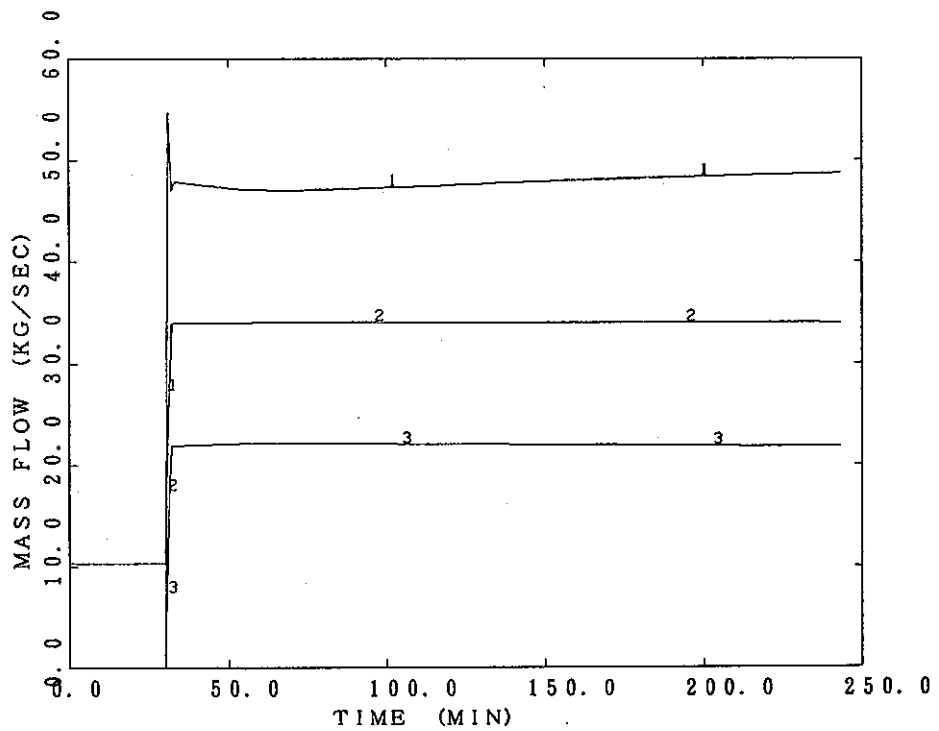


DRACS 2次系出入口温度 (1出口 2入口)

図3-7 ケースP-7のプロット出力 (つづき)

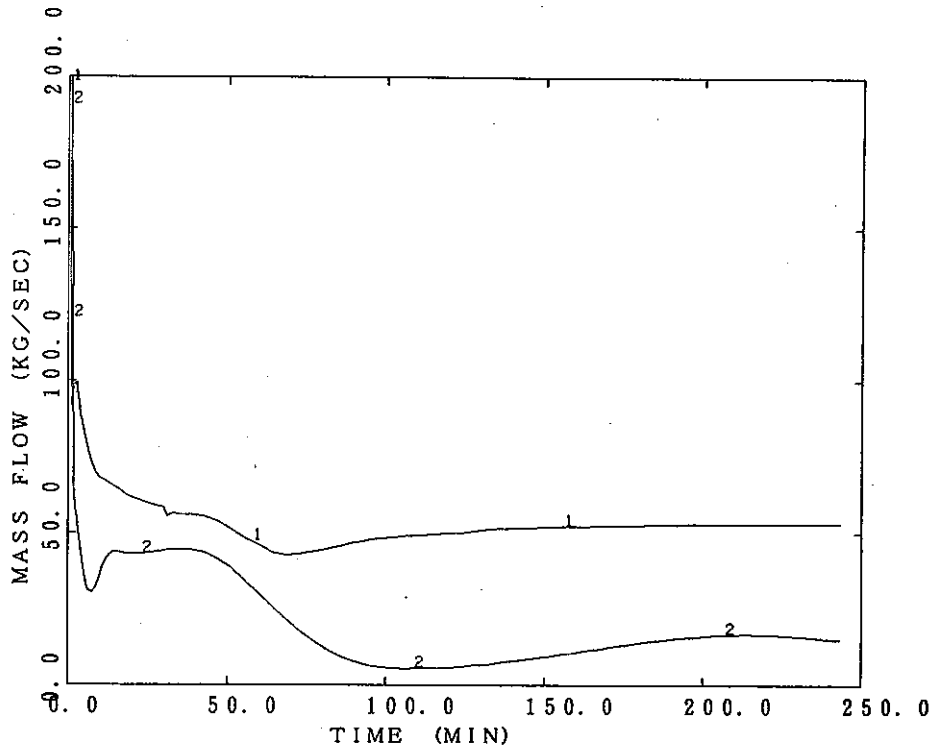


DRACS 空気温度 (1入口 2出口 3スタッフ)

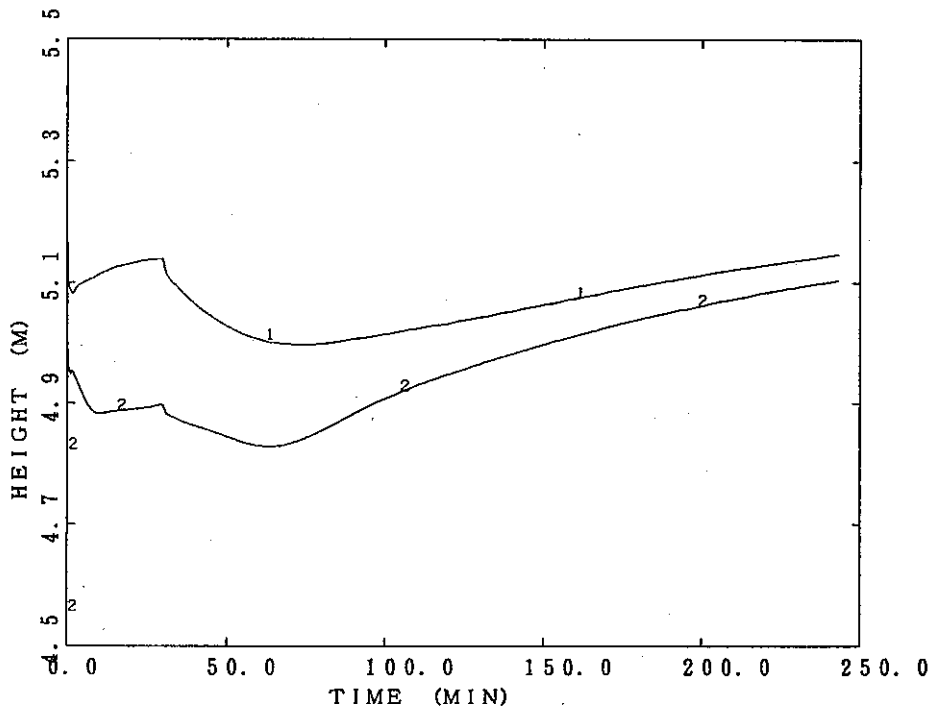


DRAC 流量 (1 1次系 2 2次系 3空気)

図3-7 ケースP-7のプロット出力 (つづき)



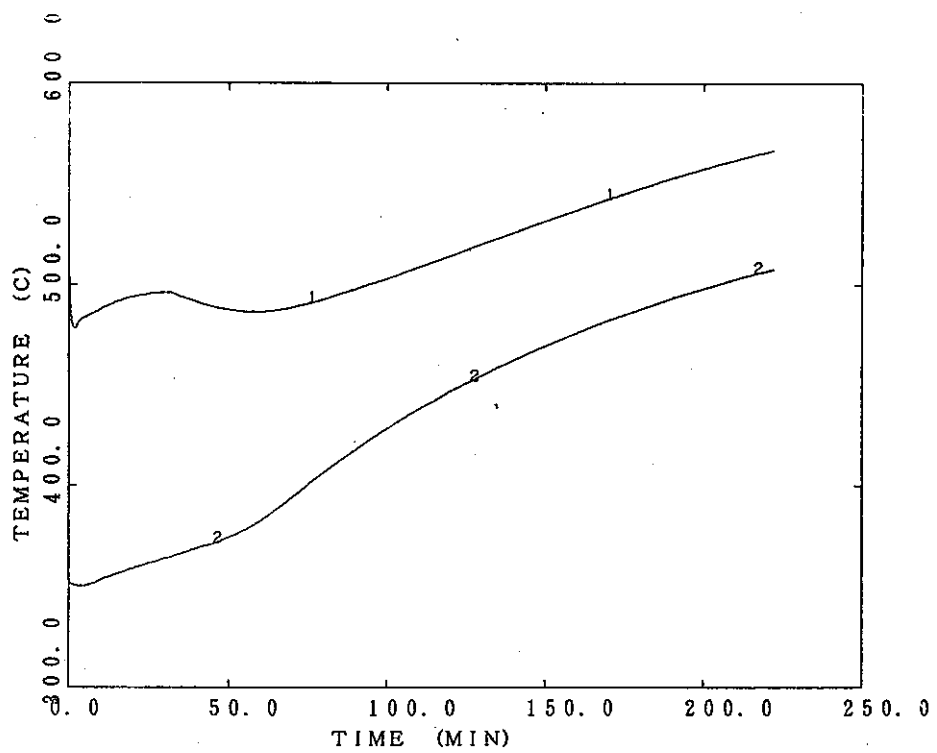
IHX 1次系と2次系の流量



ホットプール (1)、コールドプール (2) の液位

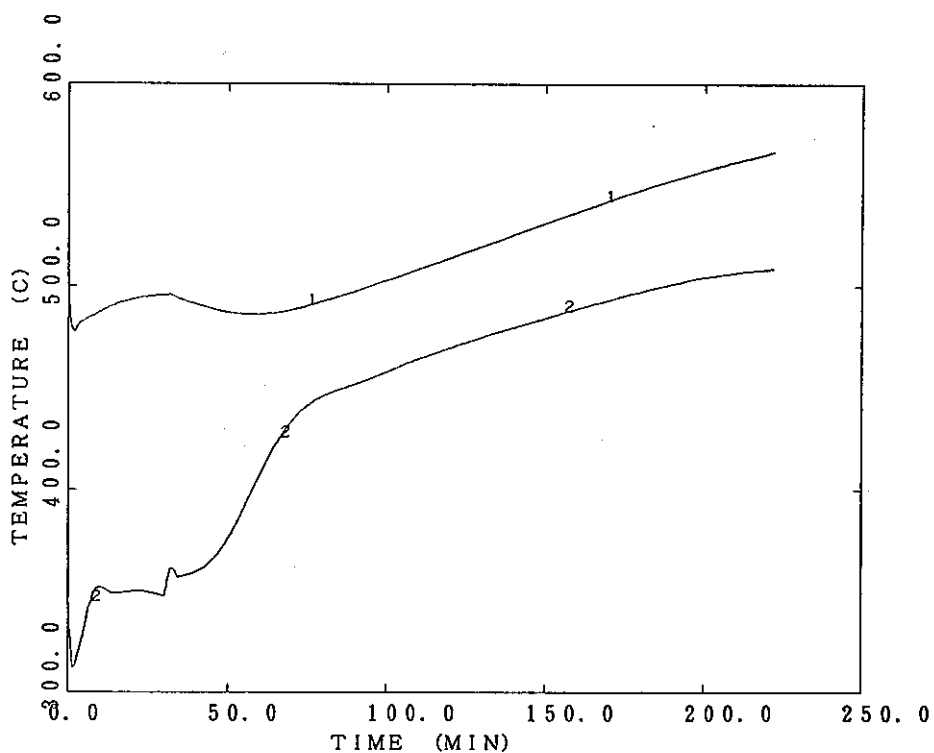
図3-7 ケースP-7のプロット出力 (つづき)

原子炉出力 (1 全出力 2 核分裂による出力)

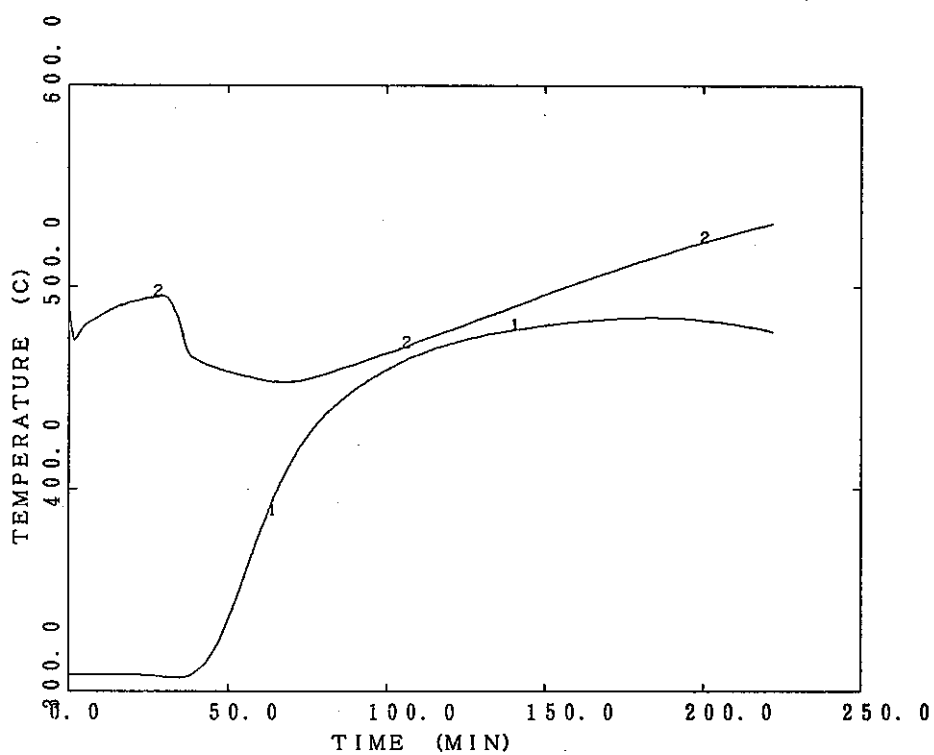


ホットプール (1)、コールドプール (2) の温度

図3-8 ケースP-8のプロット出力

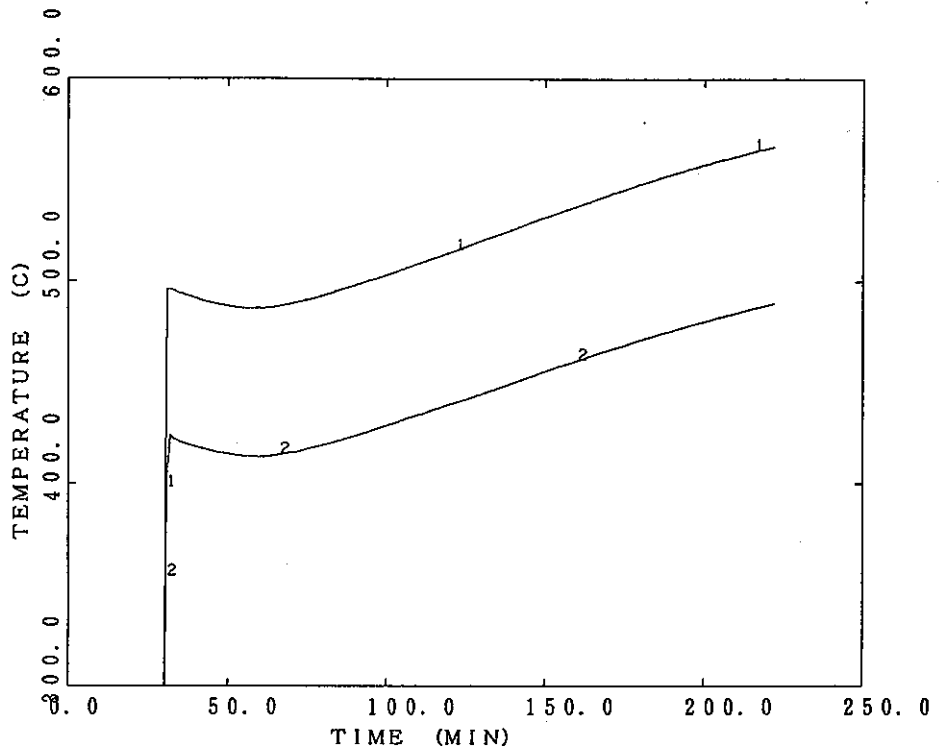


IHX1次系出入口温度 (1入口 2出口)

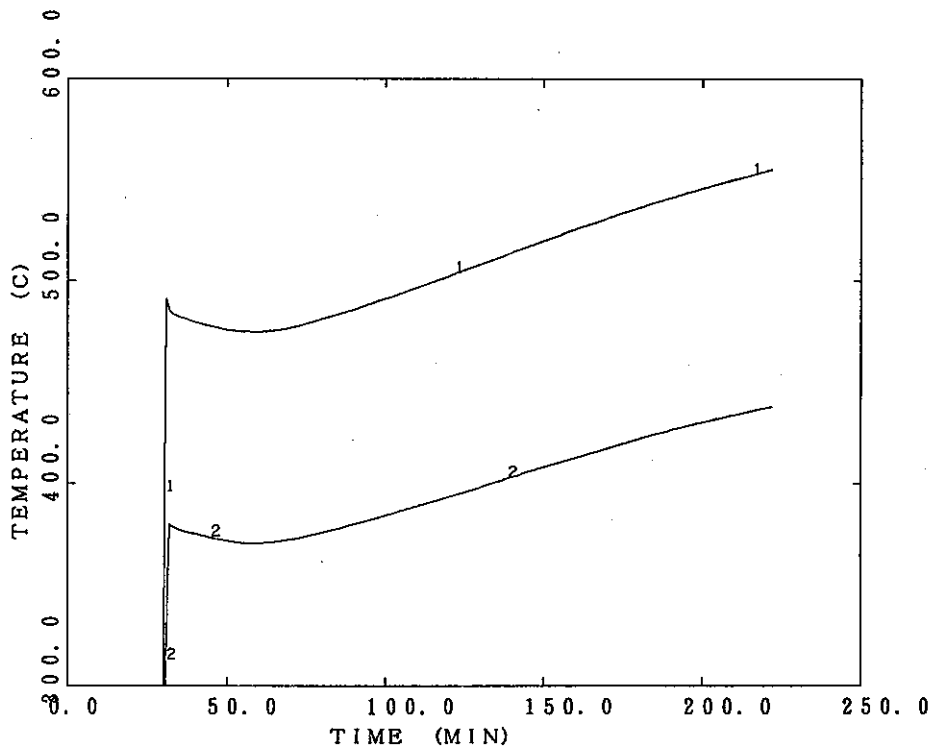


IHX2次系出入口温度 (1入口 2出口)

図3-8 ケースP-8のプロット出力 (つづき)

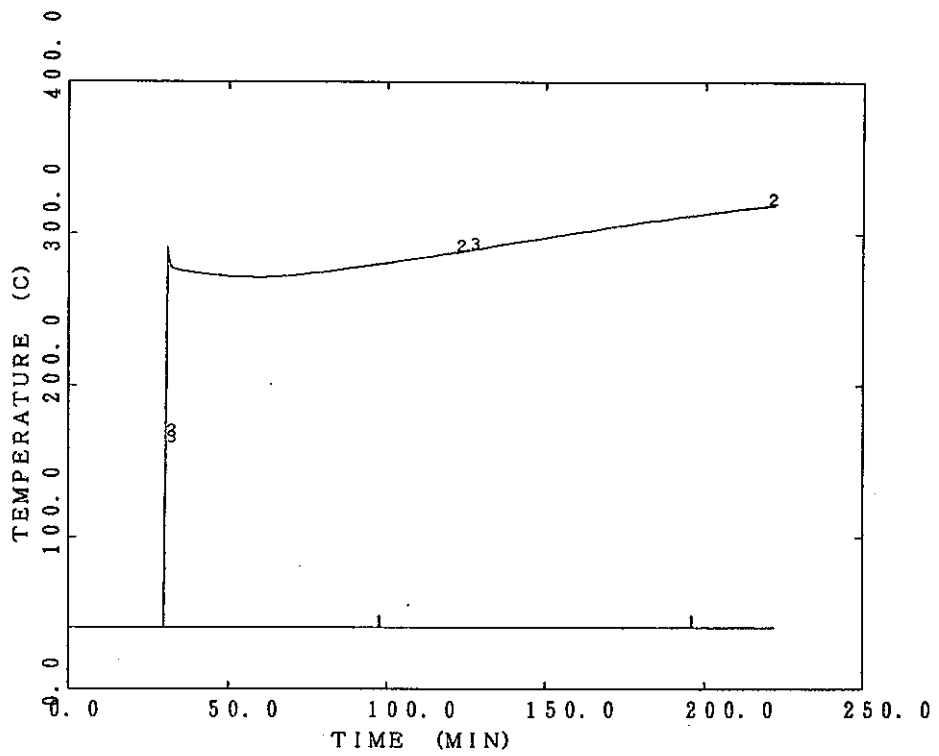


DRASC 1次系出入口温度 (1入口 2出口)

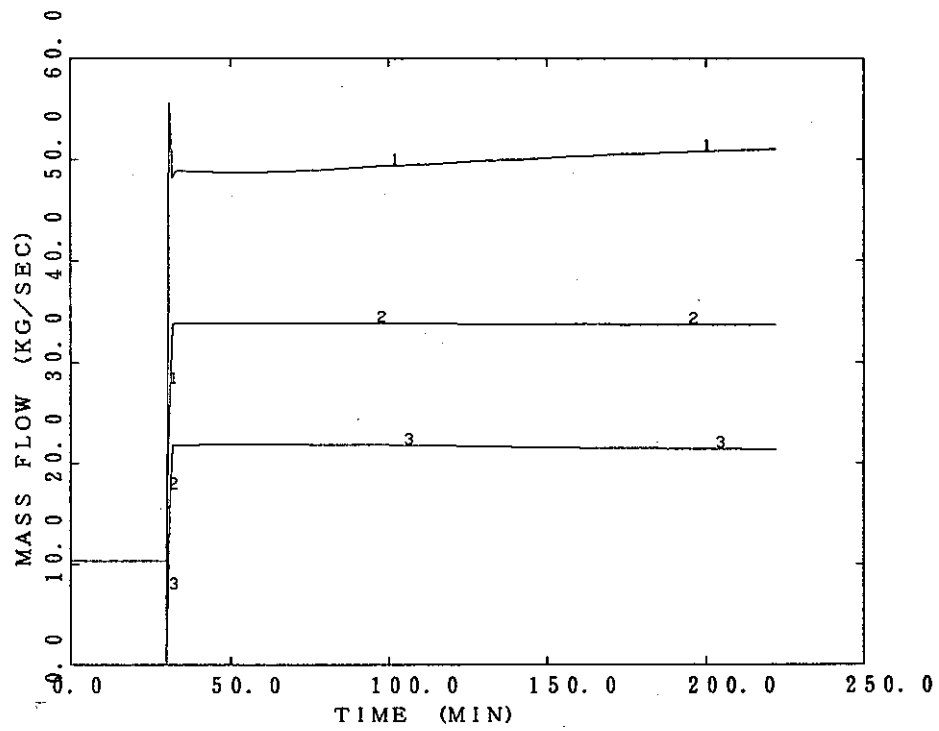


DRASC 2次系出入口温度 (1出口 2入口)

図3-8 ケースP-8のプロット出力 (つづき)

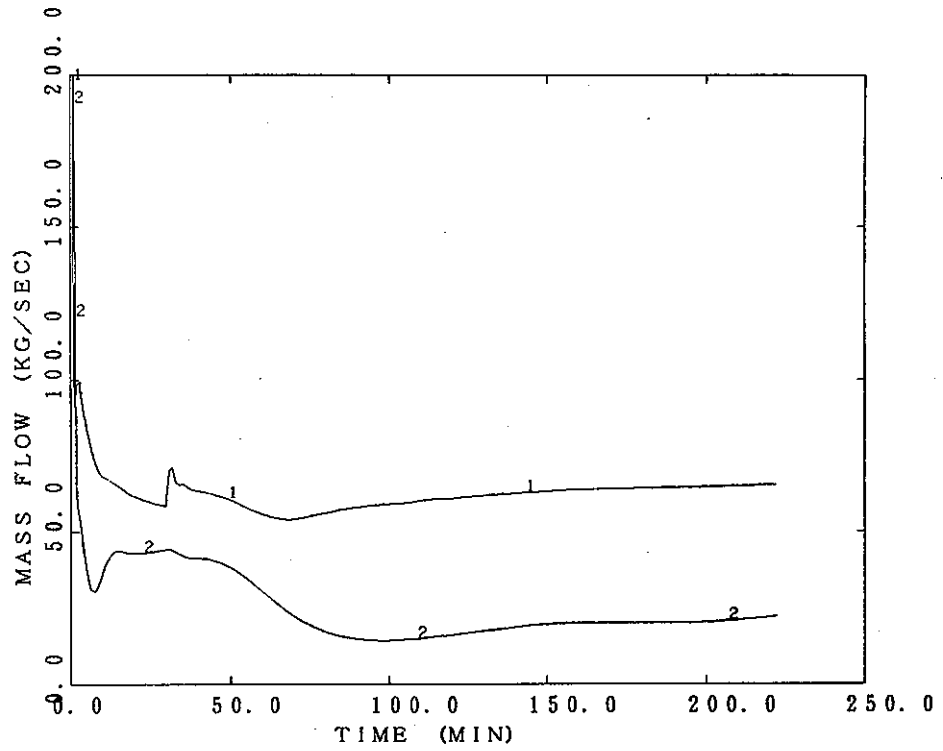


DRACS 空気温度 (1入口 2出口 3スタッフ)

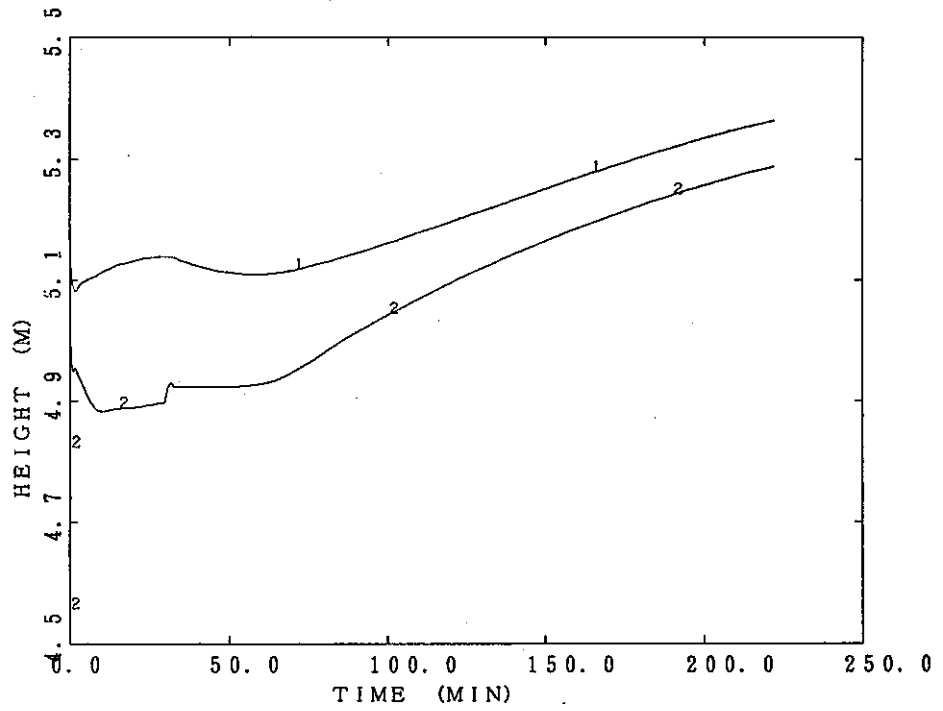


DRACCS 流量 (1 1次系 2 2次系 3空気)

図3-8 ケースP-8のプロット出力 (つづき)



IHX1次系と2次系の流量



ホットプール (1)、コールドプール (2) の液位

図3-8 ケースP-8のプロット出力 (つづき)

4. 結論及び今後の課題

実証炉の設計仕様に合致すべく、FBRプラントシステムコードSSC-L及びSSC-PのIHXモデルとDRACSモデルの改良を実施した。

改良点はIHXモデルについては現状の胴側1次、伝熱管側2次の流路を交換すること、すなわち胴側2次、管側1次とした。

DRACSモデルについては、現在SSC-Lに組み込まれている炉心上部プレナムより冷却材を取り出し下部プレナムに戻すという貫通型のモデルを炉心上部プレナム及びIHX入口プレナムにそれぞれ浸漬するタイプのモデルに変更した。

新モデルのチェックのため自然循環を仮定した除熱源喪失事故解析のテスト計算を実施し、妥当な結果を得た。

本作業により実証炉の各種事象解析が可能となった。

また、新モデルと旧モデルは入力オプションにより選択できるので、各種原子炉の基本仕様選定に有益な技術情報を提供することが可能となった。

今後の課題として、1次系全体の自然循環力を正しく評価するためには複雑な構造を持つ炉心上部プレナム内の詳細な温度分布や圧力分布を知ることが必要となる。現状の上部プレナム2領域モデルや2次元モデルには若干の不備が認められている。また今後追加した上部プレナムに浸漬するDRACSモデルは上部プレナム2次元モデルとは融合していない。今後これらの改良も含めて炉心上部プレナムモデルを整備していくことが必要である。

SSC-Pについては各種事故事象解析への適用例が少ないため、今後は各種事故事象解析を数多く実施するとともに、SSC-Lによって得られた知見を反映していく必要がある。

6. 参考文献

- [1] 山口彰他、SSC-L：ループ型高速炉システムコードモデル開発・改良と利用マニュアル、
PNC N9410 87-143、Oct.1987.
- [2] 山口彰、田嶋雄次、長谷川俊行、高速増殖大型炉の設計主要目に関する研究(1) プール型
高速増殖炉システムコードSSC-P改良整備報告書および使用説明書
PNC N9410 86-026, Apr.1987.
- [3] 山口彰、田嶋雄次、プール型高速炉動特性解析コードSSC-Pの改良と熱輸送系過渡事象
の解析
内部資料 1988年3月

付録 A 入力マニュアル

1. SSC - L

基本的に今回追加した入力は4番ファイルすなわちネームリスト入力のみである。5番ファイルは変更の必要がない。

○ネームリスト名 IHXOP

IOPIHX 0の時 IHX旧モデル (管内2次、胴側1次)

0以外の時新モデル (管内1次、胴側2次)

Y2PIHX 2次側パイプの内径 (0を入力するとコード内で計算する)

V2IHX IHX伝熱部の2次側の体積 (0を入力するとコード内で計算する)

AA2IHX IHX2次側の流路断面積 (0を入力するとコード内で計算する)

○ネームリスト名 DRACS (既に存在している)

IOPDRC 0の時 DRACS旧モデル、0以外の時新モデル

他の変数の意味は今までと同様である。新モデルを使用すると実際には使われない変数があるが、同じデータを入力して問題はない。(例えば1次系の配管や機器のデータ) SSC-Lの入力マニュアルについては文献 [1] を参照されたい。

(注) DRACS新モデルの場合 DRACS 出口高さを示す変数 (Z1LPL) は使用されないで次式により求める

出口高さ = 入口高さ (Z1UPL) - 出入口高低差 (Z1DHX)

2. SSC - P

ネームリスト IHXOPについてはSSC-Lと同様なのでネームリスト DRACS についてのみ説明する。

IOPDRC 0の時ホットプールに浸漬する DRACSモデル使用0以外の時PRACSモデル

Z1UPL IOPDRC = 0の時は DRACS 入口の炉心出口からの高さを表わし、IOPDRC = 0以外の時は PRACS 入口の IHX 入口プレナム下端からの高さを表す。

他は前ページの (注) も含めて SSC-L と同様である。

SSC-Pの入力マニュアルについては文献 [2] と [3] を参照されたい。

なお今回 DRACS モデル組み込みに伴ってプロット項目も追加された。

変更したのはプロットファイル第3レコード（文献 [2],[3] で大分類3に相当）で、その内容を以下に示す。

- 1 炉心下部プレナム冷却材温度
- 2 ホットプールA領域温度
- 3 ホットプールB領域温度
- 4 ホットプール温度
- 5 コールドプール温度
- 6 DRACS1次系NHX入口温度
- 7 " NHX出口温度
- 8 DRACS2次系NHX出口温度
- 9 " ACS入口温度
- 10 " ACS出口温度
- 11 " NHX入口温度
- 12 " ACS空気側入口温度
- 13 " ACS空気側出口温度
- 14 " ACS空気側スタック温度
- 15 DRACS1次系流量（1ループ当り）
- 16 " （全ループ合計）
- 17 DRACS2次系流量（1ループ当り）
- 18 DRACS空気流量（1ループ当り）

付録B テスト計算の入力データ

今回実施したテスト計算の入力データはすべてデータセットとして保存している。SSC-LのデータはPA340.SSCL88.DATA SSC-PのデータはPA340.SSCP88.DATAとしている。本報告書の3章で使ったテスト計算のケース名に対応する入力データのメンバー名を以下に示す。

ケース名	5番ファイル	4番ファイル
L-1	MIHX	NIHX
L-2	MIHX	NIHXNEW
L-3	MDRCS	NDRCS
L-4	MDRCS	NDRCNEW
P-5	MTRIP	NIHX
P-6	MTRIP	NIHXNEW
P-7	MTRIP	NDRCS
P-8	MTRIP	NPRCS

MIHX SSC-Lの5番ファイル入力データで、4チャンネル1ループでありIHXは26ノードに分割されている。

MDRCS SSC-Lの5番ファイル入力データで、1チャンネル1ループでありIHXは6ノードに分割されている。

MTRIP SSC-Pの5番ファイル入力データで、5チャンネル2ループでありIHXは

40

ノードに分割されている。

なおSSC-Pの5番ファイル入力データは文献〔3〕で実施したプール型炉の設計データ（日本原子力発電株式会社より入手）に基づく実プラントの過渡解析で使用したデータを一部修正したものである。

以下に変更点を示す。

- 1次系のエレベーション
- 1次系、2次系ポンプのイナーシャ

○炉心チャンネル分割数

○ホットプール、コールドプールの面積

○崩壊熱曲線

これらのデータは上記設計データを参考にして作成した。

付録 C コレクションセット

今回作成したコレクションセットのリストを表C-1～表C-9に示す。これらコレクションセットはSSC-LはPA340.SSCL88.HiST.DATA SSC-PはPA340.SSCP88.HiST.DATAとしてデータセットに保存している。

以下、各メンバーの内容を簡単に記す。なお、SSC-Lの修正はCY42に対して実施したものである。

(1) SSC-L

〈CHIHX〉

- IHXの流路を管側1次、胴側2次となるモデルを作成し旧モデルとの選択も可能とした。
- 胴側流路のみかけのパイプ内径を計算する式に誤りがあったので修正した。

〈CHDRCS〉

- 上部プレナムに浸漬するタイプのDRACSモデルを作成し旧モデルとの選択も可能とした。
- 旧DRACSモデルの定常計算でDRACS出口圧の計算式に誤りがあったので修正した。

〈PLTIHX〉

- IHXの温度分布図を作成するためIHXの1次、2次、チューブ、シェルの各ノードの温度をプロットファイルに出力した。

〈UPLEBG〉

- 上部プレナム2領域モデルの改良に誤りがあったので修正した。(ただしこの修正をしても結果がおかしいので他にも問題がありそうである。今回のテスト計算ではこのコレクションセットは使用していない)

(2) SSC-P

〈CHIHX〉

- IHXの流路を管側1次、胴側2次となるモデルを作成し旧モデルとの選択も可能とした。

〈DRACS〉

- ホットプールに浸漬するタイプのDRACSとIHX入口プレナムに浸漬するPRACSモデルを作成し、両者の選択が可能とした。

〈PLTIHX〉

- IHXの温度分布図を作成するため1次、2次、チューブ、シェルの各ノードの温度をプロットファイルに出力した。

〈BPZERO〉

- 炉心外部バイパス流量が&の時もコードが正常に動作するようにした。

〈CHECK〉

- 炉心上部プレナム1領域モデルが選択できるようにした。
(今までは1領域を選択しても2領域として計算していた)
- チェックバルブの圧損係数を流量の直線近似とした。
(今までは折れ線近似)

〈PDPOOL〉

- コールドプールの流動圧損をモデル化した

表 C - 1 SSC - L IHX モデルの変更

```

*/ *****
*ID CHIX
*/ *****
*/      DEFINE NEW COMMON DECK
*/ *****
*CD IHXOPT
      COMMON /IHXOP1/ IOPIHX
      COMMON /IHXOP2/ Y2PIHX,V2IHX,AA2IHX
*/ *****
*/      CHANGE OF IHX ( PRIMARY <--> SECONDARY )
*/ *****
*BEFORE BLKDAT.208
*CALL IHXOPT
      DATA IOPIHX / 0 /
*BEFORE CRDR9R.19
*CALL IHXOPT
      NAMELIST / IHXOP / IOPIHX,Y2PIHX,V2IHX,AA2IHX
*BEFORE CRDR9R.128
      READ (4,IHXOP)
      WRITE(6,IHXOP)
*/ *****
*/      FOR RESTART FILE
*/ *****
*INSERT SVCNTP.26
      COMMON /IHXOP1/ IIXHOP(1)
      COMMON /IHXOP2/ RIHXOP(3)
*BEFORE SAVE9T.30
      WRITE(NSAVE) IIXHOP , RIHXOP
*BEFORE REST9T.48
      READ (NSAVE) IIXHOP , RIHXOP
*/ *****
*/      SUBROUTINE CALC1R
*/ *****
*BEFORE CALC1R.44
*CALL IHXOPT
*INSERT CALC1R.361
C----- SECONDARY=SHEL
      IF (IOPIHX.NE.0) THEN
        EPS=1.0E-5
        A2IHX=AA2IHX
        Y1PIPE(LL)=Y1TUB1
        A1IHX=FLOAT(N1ACTV(IN1ACT+1))*C9PI*Y1TUB1*Y1TUB1/4.
        V1IHX=A1IHX*X1PIPE(IX1PIP+L1IHX(IL1IHX+1))
        IF (A2IHX.GT.EPS) GO TO 90
        IF (Y2PIHX.GT.EPS) GO TO 80
        A2IHX = Y1TUB2*F1PD*(2.*SQRT(3.0)*F1POD**2/C9PI - 1.0)
70      Y2PIHX = A2IHX/F1PD
        GO TO 10
80      A2IHX = Y2PIHX*F1PD
        GO TO 10
90      IF (Y2PIHX.GT.EPS) GO TO 10
        GO TO 70
10     CONTINUE
        IF (V2IHX.LT.EPS) V2IHX=A2IHX*X1PIPE(IX1PIP+L1IHX(IL1IHX+1))
        GOTO 1000
      END IF
C-----
      Y2PIHX=Y1TUB1
*DELETE CALC1R.364
      A1IHX = Y1TUB2*F1PD*(2.*SQRT(3.0)*F1POD**2/C9PI - 1.0)
*BEFORE CALC1R.417
      IF (IOPIHX.EQ.0) THEN
*INSERT CALC1R.417
      END IF
*/ *****
*/      SUBROUTINE INX1S
*/ *****
*INSERT INX1S.28

```



```

*CALL IHXOPT
*DELETE IHX1S.89,90
  IF (IOPIHX.EQ.0) THEN
    A2 = Y1TUB2*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
    A1 = Y1TUB1*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
  ELSE
    A1 = Y1TUB2*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
    A2 = Y1TUB1*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
  END IF
*DELETE IHX1S.94,95
  IF (IOPIHX.EQ.0) THEN
    FP = .5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1 + Y1TUB2))
    FS = .5*Y1TUB1*ALOG((Y1TUB1 + Y1TUB2)/(2.*Y1TUB1))
  ELSE
    FP = .5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1 + Y1TUB2))
    FS = .5*Y1TUB1*ALOG((Y1TUB1 + Y1TUB2)/(2.*Y1TUB1))
  END IF
*DELETE IHX1S.100
  APES = W2REF(KW2REF)*Y2PIHX/A2IHX
*DELETE IHX1S.153
  USUBST = 1./(Y2PIHX/(ANUSS*AKS) + FS/AKT + H1FLS)
*/ *****
*/ SUBROUTINE HYDR1S
*/ *****
*INSERT HYDR1S.25
*CALL IHXOPT
*DELETE HYDR1S.135
  FLOW = W2REF(IW2REF+1)*Y2PIHX/A2IHX
*DELETE HYDR1S.169
  PDFRIC = .5*(SIMP1U(FOVERD(IFOVER+1),NIHX,DX)/Y2PIHX +FRIC(REDC,
*/ *****
*/ SUBROUTINE HYDR1T
*/ *****
*INSERT HYDR1T.56
*CALL IHXOPT
*DELETE HYDR1T.159
  FLOW=W*Y2PIHX/A2IHX
*DELETE HYDR1T.192
  PDFRIC=.5*WMODW/Y2PIHX/A2*
*/ *****
*/ SUBROUTINE PRET1T
*/ *****
*BEFORE PRET1T.27
*CALL IHXOPT
*DELETE PRET1T.125,126
  IF (IOPIHX.EQ.0) THEN
    APT= C9PI*Y1TUB2*DELX*FLOAT(N1ACTV(IN1ACT+K))
    AST= C9PI*Y1TUB1*DELX*FLOAT(N1ACTV(IN1ACT+K))
  ELSE
    AST= C9PI*Y1TUB2*DELX*FLOAT(N1ACTV(IN1ACT+K))
    APT= C9PI*Y1TUB1*DELX*FLOAT(N1ACTV(IN1ACT+K))
  END IF
*DELETE PRET1T.131,132
  IF (IOPIHX.EQ.0) THEN
    V1NA = V1IHX/FLOAT(NM1)
    V2NA = A2IHX*DELX
  ELSE
    V2NA = V2IHX/FLOAT(NM1)
    V1NA = A1IHX*DELX
  END IF
*DELETE PRET1T.134,135
  IF (IOPIHX.EQ.0) THEN
    FP = .5*Y1TUB2*ALOG(2.*Y1TUB2/YADD)
    FS = .5*Y1TUB1*ALOG(YADD/(2.*Y1TUB1))
  ELSE
    FS = .5*Y1TUB2*ALOG(2.*Y1TUB2/YADD)
    FP = .5*Y1TUB1*ALOG(YADD/(2.*Y1TUB1))
  END IF
*INSERT PRET1T.143
  IF (IOPIHX.NE.0) ISH = NIHX*(K-1)
*DELETE PRET1T.147
  IF (IOPIHX.EQ.0) THEN
    T1SHEL(IISH)=0.5*(T1NA(IT1NA+ISH+I)+T1NA(IT1NA+ISH+IH1))

```

```

ELSE
  T1SHEL(IISH)=0.5*(T1NAS(IT1NAS+ISH+I)+T1NAS(IT1NAS+ISH+IM1))
END IF
*/ *****
*/      SUBROUTINE IHX1T
*/ *****
*INSERT IHX1T.48
*CALL IHXOPT
*DELETE IHX.34,35
  IF (IOPIHX.EQ.0) THEN
    FP=.5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1+Y1TUB2))
    FS=.5*Y1TUB1*ALOG((Y1TUB1+Y1TUB2)/(2.*Y1TUB1))
  ELSE
    FS=.5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1+Y1TUB2))
    FP=.5*Y1TUB1*ALOG((Y1TUB1+Y1TUB2)/(2.*Y1TUB1))
  END IF
*DELETE IHX1T.123
  APES= WS*Y2PIHX/A2IHX
*DELETE IHX.121
  1 1./(Y2PIHX/(ANUSS*AKS)+FS/AKT+H1FLS)
*DELETE IHX1T.370
  1      1./(Y2PIHX/(ANUSS*AKS) + FS/AKT + H1FLS)
*INSERT IHX1T.198
  IF (IOPIHX.NE.0) USBPSH=0.0
*INSERT IHX1T.217
  IF (IOPIHX.NE.0) GOTO 50
*BEFORE IHX1T.371
  TSH=T1PSH(IT1PSH+IS)
  CSH=HCAP7C(L1STRC,TSH)
  USBSSH=1./(Y2PIHX/(ANUSS*AKS))
  IF (IOPIHX.EQ.0) USBSSH=0.0
  ZI=H*APSH*USBSSH/RHO/V2NA
*DELETE IHX1T.375
  TAU = RHO*V2NA/(0.5*USUBST(IUSUBS+IS)*AST/CS
  1      +0.5*USBSSH*APSH/CS)
*INSERT IHX.227
  TSTS = TSBAR - TSH
  IF (TSTS.LT.Z9MIN) TSTS = 0.0
*DELETE IHX.230
  2      -ZI*TSTS + EXX*E1PNAS(IE1PNS+IIS))/(1.0+EX+EXX)
*INSERT IHX.234
  TSTS = TSBAR - TSH
  IF (TSTS.LT.Z9MIN) TSTS = 0.0
*INSERT IHX.236
  1      -ZI*TSTS
*INSERT IHX.283
  IF (IOPIHX.EQ.0) GOTO 100
  EX= H*APSH*USBSSH/BSHEL/CSH
  TNEW=(T1SHEL(ISHELL+IS)+EX*TSBAR)/(1.+EX)
  FDEV=ABS(T1PSH(IT1PSH+IS)-TNEW)
  CHNG=ABS(T1SHEL(ISHELL+IS)-TNEW)
  T1PSH(IT1PSH+IS)=(1.+ACCL)*TNEW-ACCL*T1PSH(IT1PSH+IS)
  IF (CHNGMX.LT.CHNG) CHNGMX = CHNG
  IF (FDEVMX.LT.FDEV) FDEVMX = FDEV

```

表 C - 2 SSC - L DRACS モデルの改良

```

*/ *****
*/ ID CHDRCS
*/ *****
*/ UPDATE COMMON DECK
*/ *****
*/ DELETE MCS.10
      1      L1FLAG , L1FLSP , L1DRAC , N9DRCS , IOPDRC
*/ *****
*/ CHANGE OF DRACS
*/ *****
*/ BEFORE BLKDAT.20B
*/ CALL /BCORC/
      DATA IOPDRC / 0 /
*/ ----- FOR RESTART
*/ DELETE MCS.2
      COMMON /VAR11/ IVAR11(5)
*/ ----- FOR NAMELIST
*/ INSERT MCS.29
      G      , IOPDRC
*/ *****
*/ SUBROUTINE FLNA1S
*/ *****
*/ INSERT FLNA1S.20
C
C ----- DRACS CHANGE
C
      IF (IOPDRC.EQ.0) THEN
*/ DELETE FLNA1S.47
      PDRCOU = PDRGIN + XAV * (Z1NAC+Z1DHX-Z1NAH) * C1GRAV
*/ INSERT FLNA1S.51
C
C ----- IOPDRC=1
C
      ELSE
      CNA = W1NAR * W1NAR * (1./X1 - 1./X2) / (A1DHX*A1DHX)
      RE3 = W1NAR * X1DHSL / (A1DHX*YAV)
      F3 = FRIC(RE3,0.0)
      FDHX = F3 * Z1DHX / (X1DHSL * XAV * A1DHX * A1DHX)
      FKNA = 0.5 * W1NAR * ABS(W1NAR) * FDHX
      BYNA = C1GRAV*XAV*Z1DHX
      ZGDRCI = Z6NALV - Z1UPL
      PDRGIN = P6CGAS + X1*ZGDRCI*C1GRAV
      PDRCOU = PDRGIN + X1*C1GRAV*Z1DHX
      PDRAC = PDRGIN-PDRCOU
      END IF
*/ *****
*/ SUBROUTINE VESL1T
*/ *****
*/ DELETE MCS.370
      IF (IOPDRC.EQ.0) W6VOUT = W6VOUT + W1DHX
*/ DELETE MCS.377
      IF (IOPDRC.EQ.0) W6TOT = W6TOT + W1DHX
*/ BEFORE MCS.379
      IF (IOPDRC.EQ.0) THEN
*/ INSERT MCS.379
      ELSE
      PINV=(PINV+P6TERM)/DENOM
      END IF
*/ *****
*/ SUBROUTINE COOL6T
*/ *****
*/ DELETE MCS.100
      IF (IOPDRC.EQ.0) W6VOUT = W6VOUT + W1DHX
*/ *****
*/ SUBROUTINE LPLN6T
*/ *****
*/ DELETE MCS.151
      IF (W1DHX.LE.0.0 .OR. IOPDRC.NE.0) GO TO 120
*/ *****

```

```

*/      SUBROUTINE DCLP1T
*/ *****
*BEFORE DCLP1T.12
*CALL /T62/
*INSERT DCLP1T.13
C
C----- DRACS CHANGE
C
      IF (IOPDRC.EQ.0) GO TO 50
      IF (IDENT.EQ.1) THEN
        T1NAI=T6NAB
        IF (Z6JET.LT.Z1UPL) T1NAI=T6NAA
        T1DRCS(L1DRCS)=T1NAI
        E1DRCS(L1DRCS)=ENTH1H(T1NAI)
      END IF
      IF (IDENT.EQ.2) THEN
        T1DRCS(N1DRCS)=T1NAO
        E1DRCS(N1DRCS)=ENTH1H(T1NAO)
      END IF
      RETURN
50 CONTINUE
C
*/ *****
*/      SUBROUTINE FLNA1T
*/ *****
*BEFORE MCS.304
C
C----- DRACS CHANGE
C
      IF (IOPDRC.EQ.0) THEN
*INSERT MCS.328
C
C----- IOPDRC=1
C
      ELSE
      ZANA = Z1DHX/A1DHX
      CNA=W1NA*W1NA*(1./X1-1./X2)/(A1DHX*A1DHX)
      RE3 = W1NA * X1DHSL / (A1DHX*YAV)
      F3 = FRIC(RE3,0.0)
      FDHX = F3 * Z1DHX / (X1DHSL * XAV * A1DHX * A1DHX)
      FKNA = 0.5 * W1NA * ABS(W1NA) * (FDHX + F1DRK/XAV)
      BYNA = C1GRAV*XAV*Z1DHX
      ZGDRCI=Z6NALV-Z1UPL
      PDRGIN=P6CGAS+X1*ZGDRCI*C1GRAV
      PFCDDN = 0.0
      PDRAC = -X1*C1GRAV*Z1DHX
      END IF
*/ *****
*/      SUBROUTINE UPLN6T
*/ *****
*BEFORE MCS.156
C
C----- DRACS CHANGE
C
      IF (IOPDRC.EQ.0) THEN
*INSERT MCS.159
C
C----- IOPDRC=1
C
      ELSE
      Z1=Z1UPL
      Z2=Z1-Z1DHX
      IF (W1DHX.LT.0.0) GO TO 4500
      IF (Z6JET.GE.Z2) THEN
        EOUT=ENTH1H(T1NAO)
        EBOUT=EBOUT-W1DHX*(EOUT-E6NAB)
      ELSE
        EOUT=ENTH1H(T1NAO)
        EAOUT=EAOUT-W1DHX*(EOUT-E6NAA)
      END IF
      GO TO 4700
4500 CONTINUE
      IF (Z6JET.GE.Z1) THEN

```

```
      EOUT=ENTH1H<T1NAI)
      EBOUT=EBOUT+W1DHX*(EOUT-E6NAB)
    ELSE
      EOUT=ENTH1H<T1NAI)
      EAOUT=EAOUT+W1DHX*(EOUT-E6NAA)
    END IF
  END IF
4700 CONTINUE
```

表 C - 3 SSC - L IHX 温度分布图作成用

```

*ID PLTIHX
*INSERT YMGPLT.5
  WRITE(74) S1FLOW
  WRITE(91,*) '**** M.C.T = ',S1FLOW,' SEC. '
*BEFORE PRNT1T.25
*CALL DAT12T
*CALL //1T
*INSERT PRNT1T.82
  WRITE(74) ((T1NA(I)),I=IST1,LGT1)
  WRITE(91,*) ((T1NA(I)),I=IST1,LGT1)
*INSERT PRNT1T.84
  WRITE(74) T10UHX(IT10UH+K)
  WRITE(91,*) T10UHX(IT10UH+K)
*INSERT PRNT1T.116
  WRITE(74) T20UHX(IT20UH+K)
  WRITE(91,*) T20UHX(IT20UH+K)
*INSERT PRNT1T.121
  WRITE(74) ((T1NAS(I)),I=IST,LGT)
  WRITE(91,*) ((T1NAS(I)),I=IST,LGT)
  ITB1=IT1TUB+(K-1)*LT1TLP+1
  ITS1=IT1SHE+(K-1)*LT1SLP+1
  ITB2=ITB1+NIHX-1
  ITS2=ITS1+NIHX-1
  WRITE(74) (T1TUBE(I),I=ITB1,ITB2)
  WRITE(91,*) (T1TUBE(I),I=ITB1,ITB2)
  WRITE(74) (T1SHEL(I),I=ITS1,ITS2)
  WRITE(91,*) (T1SHEL(I),I=ITS1,ITS2)

```

表 C - 4 SSC - P IHX モデルの変更

```

*/ *****
*ID CHIX
*/ *****
*/      DEFINE NEW COMMON DECK
*/ *****
*CD IHXOPT
COMMON /IHXOP1/ IOPIHX
COMMON /IHXOP2/ Y2PIHX,V2IHX,AA2IHX
*/ *****
*/      CHANGE OF IHX ( PRIMARY <--> SECONDARY )
*/ *****
*BEFORE BLOCKD.544
*CALL IHXOPT
DATA IOPIHX / 0 /
*INSERT CRDR9R.67
*CALL IHXOPT
NAMELIST / IHXOP / IOPIHX,Y2PIHX,V2IHX,AA2IHX
*BEFORE CRDR9R.167
READ (4,IHXOP)
WRITE(6,IHXOP)
*/ *****
*/      FOR RESTART FILE
*/ *****
*BEFORE SAVE9T.68
COMMON /IHXOP1/ IIXHOP(1)
COMMON /IHXOP2/ RIHXOP(3)
*BEFORE REST9T.83
COMMON /IHXOP1/ IIXHOP(1)
COMMON /IHXOP2/ RIHXOP(3)
*BEFORE SAVE9T.101
WRITE(NSAVE) IIXHOP , RIHXOP
*BEFORE REST9T.128
READ (NSAVE) IIXHOP , RIHXOP
*/ *****
*/      SUBROUTINE CALC1R
*/ *****
*BEFORE CALC1R.281
*CALL IHXOPT
*INSERT CALC1R.605
C----- SECONDARY=SHEL
IF (IOPIHX.NE.0) THEN
  EPS=1.0E-5
  A2IHX=AA2IHX
  Y1PIPE(LL)=Y1TUB1
  A1IHX=FLOAT(N1ACTV(IN1ACT+1))*C9PI*Y1TUB1*Y1TUB1/4.
  V1IHX=A1IHX*X1PIPE(IX1PIP+L1IHX(IL1IHX+1))
  IF (A2IHX.GT.EPS) GO TO 90
  IF (Y2PIHX.GT.EPS) GO TO 80
  A2IHX = Y1TUB2*F1PD*(2.*SQRT(3.0)*F1POD**2/C9PI - 1.0)
70  Y2PIHX = A2IHX/F1PD
  GO TO 10
80  A2IHX = Y2PIHX*F1PD
  GO TO 10
90  IF (Y2PIHX.GT.EPS) GO TO 10
  GO TO 70
10  CONTINUE
  IF (V2IHX.LT.EPS) V2IHX=A2IHX*X1PIPE(IX1PIP+L1IHX(IL1IHX+1))
  GOTO 1000
END IF
C-----
Y2PIHX=Y1TUB1
*BEFORE CALC1R.661
IF (IOPIHX.EQ.0) THEN
*INSERT CALC1R.661
END IF

```

```

*/ *****
*/      SUBROUTINE IHX1S
*/ *****
*INSERT IHX1S.106
*CALL IHXOPT
*DELETE IHX1S.197,198
      IF (IOPIHX.EQ.0) THEN
          A2 = Y1TUB2*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
          A1 = Y1TUB1*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
      ELSE
          A1 = Y1TUB2*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
          A2 = Y1TUB1*C9PI*DELX*FLOAT(N1ACTV(IN1ACT+1))
      END IF
*DELETE IHX1S.202,203
      IF (IOPIHX.EQ.0) THEN
          FP = .5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1 + Y1TUB2))
          FS = .5*Y1TUB1*ALOG((Y1TUB1 + Y1TUB2)/(2.*Y1TUB1))
      ELSE
          FS = .5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1 + Y1TUB2))
          FP = .5*Y1TUB1*ALOG((Y1TUB1 + Y1TUB2)/(2.*Y1TUB1))
      END IF
*DELETE IHX1S.208
      APES = W2IHX(IW2IHX+1)*Y2PIHX/A2IHX
*DELETE IHX1S.261
      USUBST = 1./(Y2PIHX/(ANUSS*AKS) + FS/AKT + H1FLS)
*/ *****
*/      SUBROUTINE HYDR1S
*/ *****
*INSERT HYDR1S.113
*CALL IHXOPT
*DELETE HYDR1S.252
      FLOW = W2IHX(IW2IHX+1)*Y2PIHX/A2IHX
*DELETE HYDR1S.286
      PDFRIC = .5*(SIMP1U(FOVERD(IFOVER+1),NIHX,DX)/(Y2PIHX*A2)+
*/ *****
*/      SUBROUTINE HYDR1T
*/ *****
*INSERT HYDR1T.177
*CALL IHXOPT
*DELETE HYDR1T.293
      FLOW=W*Y2PIHX/A2IHX
*DELETE HYDR1T.326
      PDFRIC=.5*WMODW/Y2PIHX/A2*
*/ *****
*/      SUBROUTINE PRET1T
*/ *****
*BEFORE PRET1T.193
*CALL IHXOPT
*DELETE PRET1T.291,292
      IF (IOPIHX.EQ.0) THEN
          APT= C9PI*Y1TUB2*DELX*FLOAT(N1ACTV(IN1ACT+K))
          AST= C9PI*Y1TUB1*DELX*FLOAT(N1ACTV(IN1ACT+K))
      ELSE
          AST= C9PI*Y1TUB2*DELX*FLOAT(N1ACTV(IN1ACT+K))
          APT= C9PI*Y1TUB1*DELX*FLOAT(N1ACTV(IN1ACT+K))
      END IF
*DELETE PRET1T.297,298
      IF (IOPIHX.EQ.0) THEN
          V1NA = V1IHX/FLOAT(NM1)
          V2NA = A2IHX*DELX
      ELSE
          V2NA = V2IHX/FLOAT(NM1)
          V1NA = A1IHX*DELX
      END IF
*DELETE PRET1T.300,301
      IF (IOPIHX.EQ.0) THEN
          FP = .5*Y1TUB2*ALOG(2.*Y1TUB2/YADD)
          FS = .5*Y1TUB1*ALOG(YADD/(2.*Y1TUB1))
      ELSE
          FS = .5*Y1TUB2*ALOG(2.*Y1TUB2/YADD)
          FP = .5*Y1TUB1*ALOG(YADD/(2.*Y1TUB1))
      END IF
*INSERT PRET1T.309

```



```

        IF (IOPIHX.NE.O) ISH = NIHX*(K-1)
*DELETE PRET1T.313
        IF (IOPIHX.EQ.O) THEN
            T1SHEL(IISH)=0.5*(T1NA(IT1NA+ISH+I)+T1NA(IT1NA+ISH+IM1))
        ELSE
            T1SHEL(IISH)=0.5*(T1NAS(IT1NAS+ISH+I)+T1NAS(IT1NAS+ISH+IM1))
        END IF
*/ *****
*/      SUBROUTINE IHX1T
*/ *****
*INSERT IHX1T.48
*CALL IHXOPT
*DELETE IHX.44,45
        IF (IOPIHX.EQ.O) THEN
            FP=.5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1+Y1TUB2))
            FS=.5*Y1TUB1*ALOG((Y1TUB1+Y1TUB2)/(2.*Y1TUB1))
        ELSE
            FS=.5*Y1TUB2*ALOG(2.*Y1TUB2/(Y1TUB1+Y1TUB2))
            FP=.5*Y1TUB1*ALOG((Y1TUB1+Y1TUB2)/(2.*Y1TUB1))
        END IF
*DELETE IHX1T.296
        APES= WS*Y2PIHX/A2IHX
*DELETE IHX.134
        1 1./(Y2PIHX/(ANUSS*AKS))+FS/AKT+H1FLS)
*DELETE IHX1T.543
        1      1./(Y2PIHX/(ANUSS*AKS) + FS/AKT + H1FLS)
*INSERT IHX1T.371
        IF (IOPIHX.NE.O) USBPSH=0.0
*INSERT IHX1T.390
        IF (IOPIHX.NE.O) GOTO 50
*BEFORE IHX1T.544
        TSH=T1PSH(IT1PSH+IS)
        CSH=HCAP7C(L1STRC,TSH)
        USBSSH=1./(Y2PIHX/(ANUSS*AKS))
        IF (IOPIHX.EQ.O) USBSSH=0.0
        ZI=H*APSH*USBSSH/RHO/V2NA
*DELETE IHX1T.548
        TAU = RHO*V2NA/(0.5*USUBST(IUSUBS+IS)*AST/CS
        1      +0.5*USBSSH*APSH/CS)
*INSERT IHX.240
        TSTS = TSBAR - TSH
        IF (TSTS.LT.Z9MIN) TSTS = 0.0
*DELETE IHX.243
        2      -ZI*TSTS + EXX*E1PNAS(IE1PNS+IIS))/(1.0+EX+EXX)
*INSERT IHX.247
        TSTS = TSBAR - TSH
        IF (TSTS.LT.Z9MIN) TSTS = 0.0
*INSERT IHX.249
        1      -ZI*TSTS
*INSERT IHX.296
        IF (IOPIHX.EQ.O) GOTO 100
        EX= H*APSH*USBSSH/BSHEL/CSH
        TNEW=(T1SHEL(IISHELL+IS)+EX*TSBAR)/(1.+EX)
        FDEV=ABS(T1PSH(IT1PSH+IS)-TNEW)
        CHNG=ABS(T1SHEL(IISHELL+IS)-TNEW)
        T1PSH(IT1PSH+IS)=(1.+ACCL)*TNEW-ACCL*T1PSH(IT1PSH+IS)
        IF (CHNGMX.LT.CHNG) CHNGMX = CHNG
        IF (FDEVMX.LT.FDEV) FDEVMX = FDEV
    
```

表 C - 5 SSC - P DRACS モデル

```

*ID MCS
*/ *****
*/          DRAC1S INTERFACE
*/ *****
*INSERT INIT6T.510
      CALL FLNA1S
      CALL FLNK1S
      CALL STAK1S
*/ *****
*/          SUBROUTINE FLNA1S
*/ *****
*DK FLNA1S
      SUBROUTINE FLNA1S
*CALL Y09V
*CALL DATC9V
*CALL DATC9I
*CALL DAT26V
      DIMENSION  Y6HYDR(1),Y6HYOZ(1),Z6AFUL(1),Z6LFGP(1),Z6UBLK(1),
1  F6TPOW(1),F6PWD(1), F6NPWA(1),F6LSA(1), F6FLOW(1),Z6LBLK(1),
2  A6ROD(1), F6PD(1),  Z6UFGP(1),Z6CHAN(1),L6CLAD(1),L6STRC(1),
3  X6SUM(1), P6FINZ(1),F6ZINZ(1),Z6INZ(1), Z6OUZ(1)
      EQUIVALENCE( C9VDIM(1),Y6HYDR(1),Y6HYOZ(1),Z6AFUL(1),Z6LFGP(1),
1  Z6UBLK(1),F6TPOW(1),F6PWD(1), F6NPWA(1),F6LSA(1), F6FLOW(1),
2  Z6LBLK(1),A6ROD(1), F6PD(1),  Z6UFGP(1),Z6CHAN(1),L6CLAD(1),
3  L6STRC(1),X6SUM(1), P6FINZ(1),F6ZINZ(1),Z6INZ(1), Z6OUZ(1) )
*CALL DAT26P
*CALL DAT26L
*CALL DAIN6V
*CALL DAIN6I
*CALL DATT6V
*CALL REF1V
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
*CALL /YMGSTK/
      C1GRAV=C9GRAV
      T1 = T1NAIR
      X1 = DENS1D(T1)
      Y1 = VISC1N(T1)
      T2 = T1NAOR
      X2 = DENS1D(T2)
      Y2 = VISC1N(T2)
      XAV = (X1+X2)/2.0
      YAV = (Y1+Y2) / 2.0
C
      CNA = W1NAR*W1NAR*(1./X1-1./X2)/(A1DHX*A1DHX)
C
      RE3 = W1NAR * X1DHSL / (A1DHX*YAV)
      F3 = FRIC(RE3,0.0)
      FDHX = F3 * Z1DHX / (X1DHSL * XAV * A1DHX * A1DHX)
      FKNA = 0.5 * W1NAR * ABS(W1NAR) * FDHX
C
      BYNA = C1GRAV * XAV*Z1DHX
C
      PDRGIN = 0.0
      PDRCOU = X1*Z1DHX*C1GRAV
      PDRAC = PDRGIN - PDRCOU
C
      F1DRK IS KNOWN
C
      IF (F1DRK.GT.0.0) GO TO 500
C
      F1DRK IS UNKNOWN
C
      IF (F1DRK.EQ.0.0) THEN
          F1DRK = 2.0 * XAV * (CNA + PDRAC + BYNA - FKNA)
          / W1NAR / ABS(W1NAR)
          GO TO 500
      END IF

```

```

C
C      PRESSURE LOSS IS KNOWN
C
      IF (F1DRK.LT.0.0) THEN
          F1DRK = - F1DRK
          F1DRK = 2.0 * XAV * F1DRK
1         / WINAR / ABS(WINAR)
      END IF
C
500  CONTINUE
      CALL PAGE9U
      WRITE(6,9000) CNA , BYNA , FKNA , PORCIN , PORCOU , P1FCO ,
1         P1REFF , WINAR , F1DRK
9000  FORMAT (////' ***** DRACS STEADY STATE OUTPUT *****' /
1         ' ----- SUBROUTINE FLNA1S -----' /
2         ' ACCELELATION PRESSURE LOSS          =',E12.4,' N/M**2' /
3         ' GRAVITY PRESSURE LOSS                =',E12.4,' N/M**2' /
4         ' FRICTION PRESSURE LOSS              =',E12.4,' N/M**2' /
5         ' DRACS LOOP INLET PRESSURE           =',E12.4,' N/M**2' /
6         ' DRACS LOOP OUTLET PRESSURE          =',E12.4,' N/M**2' /
7         ' PRESSURE LOSS AT FCD                 =',E12.4,' N/M**2' /
8         ' PUMP PRESSURE RISE                   =',E12.4,' N/M**2' /
9         ' REFERENCE MASS FLOW RATE             =',E12.4,' KG/SEC' /
A         ' OTHER PRESSURE LOSS COEFFICIENT     =',E12.4)
      RETURN
      END
*/ *****
*/      SUBROUTINE FLNK1S
*/ *****
*DK FLNK1S
      SUBROUTINE FLNK1S
*CALL DATC9V
*CALL DATC9I
*CALL REF1V
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
*CALL /YMGSTK/
C
      C1GRAV=C9GRAV
C
C
      T1=T1NKHR
      X1=DENK1D(T1)
      Y1=VISCNK(X1,T1)
C
      T2=T1NCCR
      X2=DENK1D(T2)
      Y2=VISCNK(X2,T2)
C
      XAV=(X1+X2)/2.0
      YAV=(Y1+Y2)/2.0
C
      CNK = WINKR * WINKR * (1./X1 - 1./X2) *
1         (1./A1DHTB/A1DHTB - 1./A1NHTB/A1NHTB)
      BYNK=C1GRAV*(-X1*Z1NKH+XAV*Z1NHX+X2*Z1NKC-XAV*Z1DHX)
C
C
      RE1 = WINKR * X1NK / (A1NK*Y1)
      F1 = FRDRAC(RE1)
      FNKH = F1*Y1NKH/(X1NK*X1*A1NK**2)
C
      RE2 = WINKR * X1NK / (A1NK*Y2)
      F2 = FRDRAC(RE2)
      FNKC = F2 * Y1NKC / (X1NK*X2*A1NK**2)
C
      RE3 = WINKR * Y1KNI/(A1DHTB*YAV)
      F3 = FRDRAC(RE3)
      FDHX = F3 * Y1DHTB / (Y1KNI*XAV*A1DHTB**2)
C
      RE4 = WINKR * X1NHTB / (A1NHTB*YAV)
      F4 = FRDRAC(RE4)
      FNHX = F4 * Y1NHTB / (X1NHTB*XAV*A1NHTB**2)
C

```

```

C
C   F1DRNK IS KNOWN
C
IF (F1DRNK.GT.0.0) THEN
  FNK = 0.5 * W1NKR * W1NKR * (FNKH+FNKC+FDHX+FNHX+F1DRNK/XAV)
  P2REFP = F1DRNK/2.0/XAV*W1NKR*ABS(W1NKR)-CNK-BYNK+FNK
  GO TO 500
END IF

C
C   F1DRNK IS UNKNOWN
C
IF (F1DRNK.EQ.0.0) THEN
  FNK = 0.5 * W1NKR * W1NKR * (FNKH+FNKC+FDHX+FNHX)
  F1DRNK = 2.0*XAV * (P2REFP+CNK+BYNK-FNK) / W1NKR / ABS(W1NKR)
  GO TO 500
END IF

C
C   PRESSURE LOSS IS KNOWN
C
IF (F1DRNK.LT.0.0) THEN
  FNK = 0.5 * W1NKR * W1NKR * (FNKH+FNKC+FDHX+FNHX)
  F1DRNK = - F1DRNK
  P2REFP = F1DRNK-CNK-BYNK+FNK
  F1DRNK = 2.0 * XAV * F1DRNK
1 / W1NKR / ABS(W1NKR)
END IF

C
C
C   500 WRITE(6,9000) CNK , BYNK , FNK , P2REFP , W1NKR , F1DRNK
C
C   SET INITIAL CONDITION
C
IF (W1NK.NE.0.0) THEN
  T1=T1NKH
  X1=DENK1D(T1)
  Y1=VISCNK(X1,T1)

C
  T2=T1NKC
  X2=DENK1D(T2)
  Y2=VISCNK(X2,T2)

C
  XAV=(X1+X2)/2.0
  YAV=(Y1+Y2)/2.0

C
  CNK = W1NK * W1NK * (1./X1 - 1./X2) *
1 (1./A1DHTB/A1DHTB - 1./A1NHTB/A1NHTB)
  BYNK=C1GRAV*(-X1*Z1NKH+XAV*Z1NHX+X2*Z1NKC-XAV*Z1DHX)

C
  RE1 = W1NK * X1NK / (A1NK*Y1)
  F1 = FRDRAC(RE1)
  FNKH = F1*Y1NKH/(X1NK*X1*A1NK**2)

C
  RE2 = W1NK * X1NK / (A1NK*Y2)
  F2 = FRDRAC(RE2)
  FNKC = F2 * Y1NKC / (X1NK*X2*A1NK**2)

C
  RE3 = W1NK * Y1KNI/(A1DHTB*YAV)
  F3 = FRDRAC(RE3)
  FDHX = F3 * Y1DHTB / (Y1KNI*XAV*A1DHTB**2)

C
  RE4 = W1NK * X1NHTB / (A1NHTB*YAV)
  F4 = FRDRAC(RE4)
  FNHX = F4 * Y1NHTB / (X1NHTB*XAV*A1NHTB**2)
  FNK = 0.5 * W1NK * W1NK * (FNKH+FNKC+FDHX+FNHX+F1DRNK/XAV)
  P2INIT = F1DRNK/2.0/XAV*W1NK*ABS(W1NK)-CNK-BYNK+FNK
  F2INIT = P2INIT / P2REFP
  DO 600 K = 1 , 25
    IF (P2EMPD(K).LT.F2INIT) THEN
      P2EMPD(K) = F2INIT
    ELSE
      GO TO 700
    END IF

```

```

600     CONTINUE
      END IF
700     WRITE(6,9100) CNK , BYNK , FNK , P2INIT , W1NK
      RETURN
9000    FORMAT (//////' ***** DRACS STEADY STATE OUTPUT *****'/
1       '      ----- SUBROUTINE FLNK1S -----'/
1       ' ACCELELATION PRESSURE LOSS      =',E12.4,' N/M**2'/
1       ' GRAVITY PRESSURE LOSS           =',E12.4,' N/M**2'/
1       ' FRICTION PRESSURE LOSS         =',E12.4,' N/M**2'/
1       ' PUMP PRESSURE RISE              =',E12.4,' N/M**2'/
1       ' REFERENCE MASS FLOW RATE        =',E12.4,' KG/SEC'/
1       ' OTHER PRESSURE LOSS COEFFICIENT =',E12.4)
9100    FORMAT (//////' ***** DRACS INITIAL STATE OUTPUT *****'/
1       '      ----- SUBROUTINE FLNK1S -----'/
1       ' ACCELELATION PRESSURE LOSS      =',E12.4,' N/M**2'/
1       ' GRAVITY PRESSURE LOSS           =',E12.4,' N/M**2'/
1       ' FRICTION PRESSURE LOSS         =',E12.4,' N/M**2'/
1       ' PUMP PRESSURE RISE              =',E12.4,' N/M**2'/
1       ' INITIAL MASS FLOW RATE         =',E12.4,' KG/SEC')
C
      END
*/ *****
*/ SUBROUTINE STAK1S
*/ *****
*DK STAK1S
      SUBROUTINE STAK1S
*CALL DATC9V
*CALL DATC9I
*CALL REF1V
*CALL VAR1I
*CALL VAR1V
*CALL /YMGORC/
*CALL /YMGSTK/
C
C
      PATH = P9ATH
      DNI = PATH / (Q1AIR*T1AIR)
      DNO = PATM / (Q1AIR*T1AOR)
      DAV = 0.5 * (DNI + DNO)
      CONST = 0.5 * W1AR**2 / DAV / A1STAK**2
      P1STR=C1AIR*Z1STAK*(1./T1AIR-1./T1AOR)
C
C      PRESSURE LOSS AT DAMPER
C
C      IF (N1DAMP.EQ.0) GO TO 3
C      FSTD1 = FLOAT(N1DCOS) / FLOAT(N1DAMP) * (1.0-R1AREA) *
C      1      COS(0.5*C9PI*R1DCOS)
C      FSTD2 = FLOAT(N1DFIN) / FLOAT(N1DAMP) * (1.0-R1AREA) *
C      1      COS(0.5*C9PI*R1DFIN)
C      FSTD = 1.0 / (1.0 - FSTD1 - FSTD2)
C      FSTD = FSTD * FSTD - 1
C      GO TO 4
C      3 FSTD = 0.0
C      4 CONTINUE
C      PDDMP = FSTD * CONST
C
C      IN THIS CASE, PRESSURE LOSS ACROSS THE AIR COOLER IS KNOWN
C
C      IF (F1STAK.LT.0.0) THEN
          PDACS = - F1STAK
          A = PATM - P1STR
          P2FANO = - P1STR + PDACS
          B = PATM + P2FANO
          F1STAK = PDACS / CONST
      END IF
C
C      IN THIS CASE, F1STAK IS KNOWN
C
C      IF (F1STAK.GT.0.0) THEN
          PDACS = F1STAK * CONST
          P2FANO = P1STR + PDACS
          A = PATM - P1STR
          B = PATM + P2FANO

```

```

END IF
C
C   IN THIS CASE, P2FANO IS KNOWN
C
IF (F1STAK.EQ.0.0) THEN
  A = PATH - P1STR
  B = PATH + P2FANO
  PDACS = P1STR + P2FANO
  F1STAK = PDACS / CONST
END IF
C
C   CALCULATE TIME CONSTANT IN STACK
C
S1TAUO = Z1STAK * A1STAK * DAV / W1AR
C
C
WRITE(6,9000) PDACS , P2FANO , F1STAK
9000 FORMAT (/////'***** DRACS STEADY STATE OUTPUT *****'/
1        '----- SUBROUTINE STAK1S -----'/
1        ' PDACS =',E12.4,' N/M**2'/
1        ' P2FANO =',E12.4,' N/M**2'/
1        ' F1STAK =',E12.4)
RETURN
END
*/ *****
*/ *****
*/
*/          SSC-L DRACS INTERFACE
*/
*/ *****
*/ *****
*/ *****
*/          FOR RE-START
*/ *****
*BEFORE SAVE9T.68
COMMON /REF1V/ RREF1V(13)
COMMON /VAR1I/ IVAR1I(5)
COMMON /VAR1V/ RVAR1V(22)
COMMON /DVDT1/ RDVDT1(6)
COMMON /YMGDRC/ RYMGDR(384)
COMMON /YMGDRI/ IYMGDR(6)
COMMON /YMGSTK/ RYMGST(64)
COMMON /YMGSTI/ IYMGST(4)
*BEFORE REST9T.83
COMMON /REF1V/ RREF1V(13)
COMMON /VAR1I/ IVAR1I(5)
COMMON /VAR1V/ RVAR1V(22)
COMMON /DVDT1/ RDVDT1(6)
COMMON /YMGDRC/ RYMGDR(384)
COMMON /YMGDRI/ IYMGDR(6)
COMMON /YMGSTK/ RYMGST(64)
COMMON /YMGSTI/ IYMGST(4)
*BEFORE SAVE9T.101
WRITE(NSAVE) RREF1V , IVAR1I , RVAR1V , RDVDT1 ,
1          RYMGDR , IYMGDR , RYMGST , IYMGST
*BEFORE REST9T.128
READ(NSAVE) RREF1V , IVAR1I , RVAR1V , RDVDT1 ,
1          RYMGDR , IYMGDR , RYMGST , IYMGST
*/ *****
*/          BLOCK DATA
*/ *****
*BEFORE BLOCKD.544
*CALL VAR1I
DATA IOPDRC / 0 /
*/ *****
*/          NAMELIST STATEMENTS
*/ *****
*INSERT CRDR9R.67
*CALL /YMGDRC/
*CALL /YMGSTK/
*CALL VAR1I

```

```

*CALL VAR1V
  NAMELIST /DRACS/ L1DRAC , N9DRCS , F1DLOP , S1DRCS , Y1KNI ,
1     Y1KNO , C1NTB , STAUN , STAUK , S1TAU , Z1UPL ,
2     Z1LPL , Z1DHX , Y1NAH , Y1NAC , Z1NAH , Z1NAC ,
3     X1NA , X1DHSL , N1EMPD , T1EMPD , P1EMPD , P1REFP ,
4     N2EMPD , P2EMPD , T2EMPD , P2REFP , A1DHX , P1FCD ,
5     P2FANO , F1DRNK ,
6     F1DRK , F1STAK , Z1STAK , A1STAK , N1DAMP , N1DCOS ,
8     N1DFIN , R1AREA , R1DFIN , R1DCOS , N1FANH , P1FANH ,
9     T1FANH , Y1DHTB , Z1NKH , Z1NKC , Z1NHX , Y1NKC ,
A     Y1NKH , Y1NHTB , X1NK , X1NHTB , A1NHTB ,
B     W1NAR , W1NA , T1NAIR , T1NAOR , T1NAI , T1NAO ,
C     W1NKR , W1NK , T1NKHR , T1NKCR , T1NKH , T1NKC ,
D     W1AR , W1A , T1AIR , T1AOR , T1AI ,
E     T1AO , T1ST ,
F     G1DRCS , G2DRCS , N1DRCS , N2DRCS , L1DRCS , L2DRCS
G     , IOPDRC

*BEFORE CRDR9R.167
  READ (4,DRACS)
  WRITE(6,DRACS)
  IF (N1DRCS.GT.25. OR .N2DRCS.GT.25) THEN
    WRITE(6,9001) N1DRCS , N2DRCS
    CALL EXIT9U(91001,8HCRDR9R )
9001  FORMAT(1H1,'***** ERROR *****'/
1     ' NODE NUMBER IN PRIMARY DRACS LOOP IS ',I4,/
2     ' NODE NUMBER IN SECONDARY DRACS LOOP IS ',I4,/
3     ' THEY MUST BE LESS THAN 25')
  END IF

*/ *****
*/
*/   SUBROUTINE CALC1R
*/
*/ *****
*INSERT CALC1R.45
*CALL VAR1I
*CALL VAR1V
*INSERT CALC1R.294
  IF (L1DRAC.EQ.1) N1FEQT = N1FEQT + 2
*/ *****
*/
*/   SUBROUTINE EQIV1T
*/
*/ *****
*BEFORE EQIV1T.229
*CALL DVDT1T
*CALL VAR1I
*CALL VAR1V
*CALL DATD5V
*CALL DATD5I
*/ ----- POOL VERSION CORDING MISS ( BY TAJ)
*DELETE EQIV1T.240
  KK=IY1+N1EQ-N1HPEQ
*DELETE EQIV1T.301
  KK=IY1+N1EQ-N1HPEQ
*DELETE EQIV1T.398
  KK=IY1DYD+N1EQ-N1HPEQ
*/ ----- DRACS AND PROMPT JUMP
*INSERT EQIV1T.259
  JJ=0
  IF (L5POPT.GT.0) GO TO 120
  JJ=N5DNGP
  KK=N1FEQT-JJ-2
  DO 110 K=1,JJ
    Y1(IY1+KK+K)=1.0
110  CONTINUE
120  CONTINUE
  IF (L1DRAC.EQ.1) THEN
    KBC = IY1 + LY1 - JJ - 3
    Y1(KBC) = W1NA
    KBC = KBC + 1
    Y1(KBC) = W1NK
  .END IF
*INSERT EQIV1T.339

```

```

JJ=0
IF (LSPQPT.GT.0) GO TO 2120
  JJ=NSDNGP
  KK=N1FEQT-JJ-2
  DO 2110 K=1,JJ
    FSPREC(K)=Y1(IY1+KK+K)
2110 CONTINUE
2120 CONTINUE
  IF (L1DRAC.EQ.1) THEN
    KBC = IY1 + LY1 - JJ - 3
    W1NA = Y1(KBC)
    KBC = KBC + 1
    W1NK = Y1(KBC)
  END IF
*INSERT EQIV1T.417
  JJ=0
  IF (LSPQPT.GT.0) GO TO 3120
    JJ=NSDNGP
    KK=N1FEQT-JJ-2
    DO 3110 K=1,JJ
      Y1DYDT(IY1DYD+KK+K)=F5PCDT(K)
3110 CONTINUE
3120 CONTINUE
  IF (L1DRAC.EQ.1) THEN
    KBCC = IY1DYD + LY1DYD - NSDNGP - 3
    Y1DYDT(KBCC) = W1NAP
    KBCC = KBCC + 1
    Y1DYDT(KBCC) = W1NKP
  END IF
*/ *****
*/
*/          SUBROUTINE FLOW1T
*/
*/ *****
*/BEFORE FLOW1T.138
*CALL VAR1I
*CALL VAR1V
C
*/BEFORE FLOW1T.166
  IF (L1DRAC.EQ.1) THEN
    CALL FLNA1T
C
    IF (L1FLSP.GT.-1) GO TO 10.
C
    CALL FLNK1T
  CONTINUE
10  END IF
*/ *****
*/
*/          INITIALIZATION          ****INIT1T****
*/
*/ *****
*/BEFORE INIT1T.350
*CALL DATT6V
*CALL DVDT1T
*CALL VAR1I
*CALL VAR1V
*INSERT INIT1T.599
C  W1NA = 0.0
  T1=T1NAB
  IF (IOPDRC.NE.0) T1=T1INH*(IT1INH+1)
  IF (T1NAI.LT.0.0) T1NAI = T1
  IF (T1NAO.LT.0.0) T1NAO = T1
  CALL DCLP1S
  CALL DCLP2S
  L1FLSP = 1
  W1NAP = 0.0
  W1NKP = 0.0
*/ *****
*/
*/          SUBROUTINE INIT6T
*/
*/ *****
*/INSERT INIT6T.212

```



```

*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRG/
*CALL /YMGSTK/
*INSERT INIT6T.424
C
C   INITIALIZE HEAT REMOVAL SYSTEM
C
      W1DHX = F1DLOP * W1NA
      P1DHX = 0.0
      X1DHX = 0.0
C
      Q1AIR = C9RGAS
      C1AIR = C9GRAV * P9ATM / C9RGAS
      A1LPL = A6LPLF
      A1DHTB = C1NTB * (Y1KNO*0.5)**2 * C9PI
      A1NA = (X1NA*0.5)**2 * C9PI
      A1NK = (X1NK*0.5)**2 * C9PI
      Y1KNSI = (A1DHX + A1DHTB) / C9PI
      Y1KNSI = SQRT(Y1KNSI) * 2.0
C
*/ *****
*/
*/           SUBROUTINE LOOP1T
*/
*/ *****
*INSERT LOOP1T.161
*CALL VAR1I
*CALL VAR1V
C
*INSERT LOOP1T.186
      IF (L1DRAC.EQ.1) THEN
          IF(L1FLSP.GT.-1) GO TO 20
          CALL DRAC1T
20      CONTINUE
      END IF
*/ *****
*/           SUBROUTINE HTPL1T
*/ *****
*INSERT HTPL1T.95
*CALL VAR1I
*CALL VAR1V
C
      EAOUT=0.0
      EBOUT=0.0
      IF (IOPDRG.NE.0) GO TO 50
      Z1=Z1UPL
      Z2=Z1-Z1DHX
      IF (W1DHX.LT.0.0) GO TO 20
      IF (Z1JET.GE.Z2) THEN
          EOUT=ENTH1H(T1NAD)
          EBOUT=EBOUT-W1DHX*(EOUT-E1NAB)
          IF (V1NAB.EQ.0.0) EAOUT=EBOUT
      ELSE
          EOUT=ENTH1H(T1NAD)
          EAOUT=EAOUT-W1DHX*(EOUT-E1NAA)
          IF (V1NAA.EQ.0.0) EBOUT=EAOUT
      END IF
      GO TO 50
C
20 CONTINUE
      IF (Z1JET.GE.Z1) THEN
          EOUT=ENTH1H(T1NAI)
          EBOUT=EBOUT+W1DHX*(EOUT-E1NAB)
          IF (V1NAB.EQ.0.0) EAOUT=EBOUT
      ELSE
          EOUT=ENTH1H(T1NAI)
          EAOUT=EAOUT+W1DHX*(EOUT-E1NAA)
          IF (V1NAA.EQ.0.0) EBOUT=EAOUT
      END IF
      50 CONTINUE
*INSERT HTPL1T.147
      *   EBOUT -

```

```

*INSERT HTPL1T.154
  *   EAOUT -
*INSERT HTPL1T.182
  *   EBOUT -
*INSERT HTPL1T.200
  *   EAOUT -
*/ *****
*/          SUBROUTINE IHX1T
*/ *****
*INSERT IHX1T.201
*CALL VAR1I
*CALL VAR1V
*INSERT IHX1T.281
  WPRAC=0.0
  IF (IOPDRC.NE.0) WPRAC=W1NA
*INSERT IHX1T.311
  EOUT=0.0
  IF (IOPDRC.EQ.0) GO TO 5000
  IF (WPRAC.LT.0.0) THEN
    EOUT=ENTH1H(T1NAI)
  ELSE
    EOUT=ENTH1H(T1NAO)
  END IF
  5000 CONTINUE
*INSERT IHX1T.313
  EXX = H*ABS(WPRAC)/(DENS1D(TBAR)*V1PLEN(1))
*DELETE IHX1T.315
  1   (E1NA(IE1NA+I1+1) + EX*E1PIN + EXX*EOUT)/(1.+EX+EXX)
*INSERT IHX.186
  EOUT=0.0
  IF (IOPDRC.EQ.0) GO TO 5001
  IF (WPRAC.LT.0.0) THEN
    EOUT=ENTH1H(T1NAI)
  ELSE
    EOUT=ENTH1H(T1NAO)
  END IF
  5001 CONTINUE
*INSERT IHX.188
  EX1 = H*ABS(WPRAC)/(DENS1D(TBAR)*V1PLEN(1))
*DELETE IHX.192
  1   EX1*EOUT + EXX*E1PNA(IE1PNA+2))/(1.+EX+EX1+EXX)
*INSERT IHX.386
  EOUT=0.0
  IF (IOPDRC.EQ.0) GO TO 6000
  IF (WPRAC.LT.0.0) THEN
    EOUT=ENTH1H(T1NAI)
  ELSE
    EOUT=ENTH1H(T1NAO)
  END IF
  6000 CONTINUE
*INSERT IHX.388
  EXX=H*ABS(WPRAC)/(DENS1D(TBAR)*V1PLEN(1))
*DELETE IHX.390
  1   EX*E1PNA(IE1PNA+1)+EXX*EOUT)/(1.+EX+EXX)
*/ *****
*/ *****
*/
*/          MONJU DRACS LOOP
*/
*/ *****
*/ *****
*INSERT DENK1D.20
  DENK1D = DNA
  RETURN
*INSERT VISCNK.2
  VISCNK = VISC1N(Y)
  RETURN
*/ *****
*/
*/          SUBROUTINE DRAC1T
*/
*/ *****
*INSERT DRAC1T.77

```

```

C
C      DRACS PRIMARY LOOP THERMAL CALCULATION (BEFORE IHX)
      CALL DCLP1T(S1DRCT,1)
*INSERT DRAC1T.78
C
C      DRACS PRIMARY LOOP THERMAL CALCULATION (AFTER IHX)
      CALL DCLP1T(S1DRCT,2)
C
C      DRACS SECONDARY LOOP THERMAL CALCULATION (AFTER IHX)
      CALL DCLP2T(S1DRCT,1)
*INSERT DRAC1T.79
C
C      DRACS SECONDARY LOOP THERMAL CALCULATION (AFTER IHX)
      CALL DCLP2T(S1DRCT,2)
*/ *****
*/
*/      SUBROUTINE HXKN1T
*/
*/ *****
*DELETE HXKN1T.36,42
*INSERT HXKN1T.43
*CALL /YMGORC/
*CALL /YMGSTK/
*DELETE HXKN1T.78,166
C
C
C
C      IF (L1FLAG.GT.1) GO TO 2
C
      T=(T1NKHR+T1NKCR)/2.0
      TC=T-273.15
      D1NKR=DENK1D(T)
      E1CPKR = HCAP1C(T)
      CONDKR = COND1K(T)
C
      T=(T1NAIR+T1NAOR)/2.0
      D1NAR=DENS1D(T)
      E1CPNR=HCAP1C(T)
      CONDNR=COND1K(T)
C
      T=0.25*((T1NKHR+T1NKCR)+(T1NAIR+T1NAOR))
      CONDWR=COND7K(1,T)
C
C
C
      C=T1NAIR-T1NKHR
      D=T1NAOR-T1NKCR
      IF (ABS(C-D).GT.Z9MIN .AND. C*D.GT.0.0) THEN
      T1LMNR=(C-D)/ALOG(C/D)
      ELSE
      T1LMNR = 0.5 * (C + D)
      END IF
      F1NKR=(T1NKHR-T1NKCR)/T1LMNR
      F1NAR=(T1NAIR-T1NAOR)/T1LMNR
C
      G1NKR=4.0*W1NKR/(C9PI*C1NTB*Y1KNI**2)
      H1NKR=0.625*(CONDKR/Y1KNI)**0.6*(G1NKR*E1CPKR)**0.4
      Y1EQLR=(Y1KNSI**2-C1NTB*Y1KNO**2)/(C1NTB*Y1KNO)
      G1NAR=4.0*W1NAR/(C9PI*(Y1KNSI**2-C1NTB*Y1KNO**2))
      H1NAR=0.625*(CONDNR/Y1EQLR)**0.6*(G1NAR*E1CPNR)**0.4
C
      RQNKNR=1./H1NKR+Y1KNI/(H1NAR*Y1KNO)+(Y1KNO-Y1KNI)*Y1KNI/
      1 (CONDWR*(Y1KNO+Y1KNI))
      Q1NKNR=1./RQNKNR
C
C
C
      RETURN
2      T = (T1NKH + T2DRCS(N2DRCS))/2.0
      TC=T-273.15
      D1NK=DENK1D(T)
      E1CPK = HCAP1C(T)

```

```

CONDK = COND1K(T)
C
T = (T1DRCS(L1DRCS) + T1NAO) / 2.0
D1NA=DENS1D(T)
E1CPN=HCAP1C(T)
CONDN=COND1K(T)
C
T = 0.25 * ((T1NKH+T2DRCS(N2DRCS)) + (T1DRCS(L1DRCS)+T1NAO))
CONDW=COND7K(1,T)
IF(T1NKH .GE. (T1DRCS(L1DRCS)-0.001))
1 T1NKH = T1DRCS(L1DRCS) - 0.001
IF(T1NAO .LE. (T2DRCS(N2DRCS)+0.001))
1 T1NAO = T2DRCS(N2DRCS) + 0.001
A = T1DRCS(L1DRCS) - T1NKH
B = T1NAO - T2DRCS(N2DRCS)
IF (ABS(A-B).GT.Z9MIN .AND. A*B.GT.0.0) THEN
T1LMN=(A-B)/ALOG(A/B)
ELSE
T1LMN = 0.5 * (A + B)
END IF
C
G1NK=4.0*W1NK/(C9PI*C1NTB*Y1KNI**2)
G1NK=ABS(G1NK)
H1NK=0.625*(CONDK/Y1KNI)**0.6*(G1NK*E1CPK)**0.4
Y1EQL=(Y1KNSI**2-C1NTB*Y1KNO**2)/(C1NTB*Y1KNO)
G1NA=4.0*W1NA/(C9PI*(Y1KNSI**2-C1NTB*Y1KNO**2))
G1NA=ABS(G1NA)
H1NA=0.625*(CONDN/Y1EQL)**0.6*(G1NA*E1CPN)**0.4
C
IF (H1NK.GT.Z9MIN. AND .H1NA.GT.Z9MIN) THEN
RQNK=1./H1NK+Y1KNI/(H1NA*Y1KNO)+(Y1KNO-Y1KNI)*Y1KNI/
1 (CONDW*(Y1KNO+Y1KNI))
Q1NKN=1./RQNK
ELSE
Q1NKN = 0.0
END IF
C
YY1 = F1NAR * Q1NKN * E1CPNR / (Q1NKNR * E1CPN)
YY2 = F1NKR * Q1NKN * E1CPKR / (Q1NKNR * E1CPK)
C
S1TAUN = STAUN * D1NA / D1NAR
S1TAUK = STAUK * D1NK / D1NKR
C
T1NAOP = (W1NA/W1NAR*(T1DRCS(L1DRCS)-T1NAO) - YY1*T1LMN) / S1TAUN
T1NKHP = (W1NK/W1NKR*(T2DRCS(N2DRCS)-T1NKH) + YY2*T1LMN) / S1TAUK
C
C
C
RETURN
END
*/ *****
*/
*/ SUBROUTINE HXKA1T
*/
*/ *****
*INSERT HXKA1T.21
*CALL DATC9V
*CALL DATC9I
*CALL /YMGDRC/
*CALL /YMGSTK/
*DELETE HXKA1T.23,26
*DELETE HXKA1T.40,99
C
C
C
IF (L1FLAG.GT.1) GO TO 2
C
T=(T1NKHR+T1NKCR)/2.0
D1NKR=DENK1D(T)
E1CPKR = HCAP1C(T)
C

```

```

C=T1NKHR-T1AOR
D=T1NKCR-T1AIR
IF (ABS(C-D).GT.Z9MIN. AND .C*D.GT.0.0) THEN
T1LMAR=(C-D)/ALOG(C/D)
ELSE
F1LMAR = 0.5 * (C + D)
END IF
F1AR=(T1AOR-T1AIR)/T1LMAR
F1NKR=(T1NKHR-T1NKCR)/T1LMAR
C
C
C   WRITE(6,50) L1FLAG,T,D1NKR,E1CPKR,T1LMAR,F1AR,F1NKR
50  FORMAT (1H ,I3,6(1X,1PE12.4))
C
C
RETURN
2   T = (T2DRCS(L2DRCS) + T1NKC)
    D1NK=DENK1D(T)
    E1CPK = HCAP1C(T)
C
C   T1AI = T1AIR
    WNK=ABS(W1NK)
    WA=ABS(W1A)
    TAU = S1TAU * D1NK / D1NKR
    TR=(T1AO+T1AI)/(T1AOR+T1AIR)
    XX1IN = (WA/W1AR)**.3 * TR**(-.21) / F1AR
    XX2 = (F1NKR*E1CPKR)/E1CPK*(TR**.21)*(WA/W1AR)**.7
C
C   C = T2DRCS(L2DRCS) - T1AO
    D=T1NKC-T1AI
    IF (ABS(C-D).GT.Z9MIN. AND .C*D.GT.0.0) THEN
T1LMA=(C-D)/ALOG(C/D)
ELSE
T1LMA = 0.5 * (C + D)
END IF
C
C   T1NKCP = ((T2DRCS(L2DRCS)-T1NKC)*WNK/W1NKR - XX2*T1LMA)/TAU
C
C
C   X2 = (T1NKC-T1AI) / (T2DRCS(L2DRCS)-T1AI)
    X = (T1AO-T1AI) / (T2DRCS(L2DRCS)-T1AI)
    DO 10 I=1,5
    IF ((1.0-X).GT.Z9MIN) THEN
        Y = ALOG((1.0-X)/X2)
        C = 1.0 - X - X2
    ELSE
        Y = 0.0
        C = 0.0
    END IF
    IF (ABS(Y).GT.Z9MIN) THEN
        Z = C / Y
        X = X - (X*XX1IN - Z) /
1      (XX1IN + ((1.0 - Z)/(1.0-X))/Y)
    ELSE
        IF (ABS(C).LT.Z9MIN) THEN
            X = X - (X*XX1IN) / (XX1IN + 1.0)
        ELSE
            WRITE (6,9000)
            STOP
        END IF
    END IF
    IF (X.GT.1.0) X = 1.0
10  CONTINUE
    T1AO = T1AI + X*(T2DRCS(L2DRCS)-T1AI)
C
C
C
C   RETURN
9000 FORMAT (1H1,///'*****'/
1     ,'* *'/
2     ,'*   INVALID TEMPERATURES IN HXKA1T *'/
3     ,'*   T1NKC =',F10.2,' *'/

```

```

4          ,'*          T1NKH =',F10.2,'          */
5          ,'*          T1AI  =',F10.2,'          */
6          ,'*          T1AO  =',F10.2,'          */
7          ,'*          X     =',E12.3,'          */
8          ,'*          X2    =',E12.3,'          */
9          ,'*                                     */
A          ,'*****')
END
*/ *****
*/
*/          SUBROUTINE STAK1T
*/
*/ *****
*INSERT STAK1T.3
*CALL VD9V
*CALL INTG9I
*CALL INTG9V
      .DIMENSION      S9PINT(1)
      EQUIVALENCE( C9VDIM(1),S9PINT(1) )
*CALL DATC9V
*CALL DATC9I
*CALL /YMGSTK/
*DELETE STAK1T.22,54
C
C
C          WRITE(6,1)
1  FORMAT(1H ,16H STAK1T CALLED .)
C
C          IF (L1FLAG.GT.1) GO TO 2
C
C          RETURN
C
2  CONTINUE
      T1AI=T1AIR
      TAU=S1TAU0*T1ST/T1AI
      T1ST=W1A*(T1AO-T1ST)/(TAU*W1AR)
      P1ST=C1AIR*21STAK*(1./T1AI-1./T1ST)
      TMFAN = S9MSTR - S1DRCS
      IF (TMFAN.LT.0.0) TMFAN = 0.0
      CALL INTP9U (P1FANH , T1FANH , N1FANH , TMFAN , PMFAN)
      PMFAN = P2FANO * PMFAN
      PATM = P9ATM
      A=PATM-P1ST
      B = PATM + PMFAN
      IF (N1DAMP.EQ.0) GO TO 3
      FSTD1 = FLOAT(N1DCOS) / FLOAT(N1DAMP) * (1.0-R1AREA) *
1      COS(0.5*C9PI*R1DCOS)
      FSTD2 = FLOAT(N1DFIN) / FLOAT(N1DAMP) * (1.0-R1AREA) *
1      COS(0.5*C9PI*R1DFIN)
      FSTD = 1.0 / (1.0 - FSTD1 - FSTD2)
      FSTD = FSTD * FSTD - 1
      GO TO 4
3  FSTD = 0.0
4  WRAT = W1A / W1AR
      TRAT = T1AO / T1AOR
      IF (WRAT.LE.0.0) THEN
          FSTAK = 0.0
      ELSE
          FSTAK = (F1STAK+FSTD) * WRAT**(-0.32) * TRAT**0.23
      END IF
      DNI=PATH/(Q1AIR*T1AI)
      DNO=PATH/(Q1AIR*T1AO)
      D=1./DNO-1./DNI
      F1 = B * DNI
      R = B / A
      T = T1AO / T1AI
      IF (ABS(R-1.0).LE.Z9MIN) THEN
          FACTR = 0.0
      ELSE
          F2 = ALOG(R*T)
          F3 = ALOG(R)

```

```

      FACTR = F1 * (1.0-1.0/R/R/T) / (1.0+F2/F3) / (F2+0.5*FSTAK)
      END IF
      W1A = A1STAK * SQRT(FACTR)
C
C
C
      RETURN
      END
*/ *****
*/
*/      SUBROUTINE FLNA1T
*/ *****
*INSERT FLNA1T.68
*CALL /YMGDR/
*DELETE FLNA1T.70,75
*DELETE FLNA1T.82,141
C
C
      WRITE(6,1)
1  FORMAT(1H ,16H FLNA1T CALLED .)
C
      C1GRAV=C9GRAV
C
      T1=T1NAI
      X1=DENS1D(T1)
      Y1 = VISC1N(T1)
      T2=T1NAO
      X2=DENS1D(T2)
      Y2 = VISC1N(T2)
      XAV=(X1+X2)/2.0
      YAV = (Y1+Y2) / 2.0
      ZANA = Z1DHX/A1DHX
C
      CNA=W1NA*W1NA*(1./X1-1./X2)/(A1DHX*A1DHX)
C
      RE3 = W1NA * X1DHSL / (A1DHX*YAV)
      F3 = FRIC(RE3,0.0)
      FDHX = F3 * Z1DHX / (X1DHSL * XAV * A1DHX * A1DHX)
C
      FKNA = 0.5 * W1NA * ABS(W1NA) * (FDHX + F1DRK/XAV)
C
      BYNA = C1GRAV*XAV*Z1DHX
C
      PDRAC = -X1*C1GRAV*Z1DHX
C
      IF (L1FLSP.LT.1) GO TO 10
      IF (S9MSTR.GT.39.0) GO TO 10
C
C
      DRACS START CONDITION
C
      IF (S9MSTR.GT.S1DRCS) GO TO 10
      P1DHX=0.
      W1DHX=0.
      X1DHX=0.
      L1FLSP=1
      RETURN
10 CONTINUE
C
      W1NAP=(CNA+PDRAC-FKNA+BYNA)/ZANA
C
C
      W1DHX = F1DLOP * W1NA
      X1DHX = F1DLOP / ZANA
      L1FLSP=L1FLSP-1
C
      RETURN
      END
*/ *****
*/
*/      SUBROUTINE FLNK1T
*/

```

```

*/ *****
*INSERT FLNK1T.25
*CALL VD9V
*CALL INTG9I
*CALL INTG9V
      DIMENSION      S9PINT(1)
      EQUIVALENCE( C9VDIN(1),S9PINT(1) )
*CALL /YMGDRC/
*DELETE FLNK1T.27,85
C
C
      C1GRAV=C9GRAV
C
C
      T1=T1NKH
      X1=DENK1D(T1)
      Y1=VISCNK(X1,T1)
C
      T2=T1NKC
      X2=DENK1D(T2)
      Y2=VISCNK(X2,T2)
C
      XAV=(X1+X2)/2.0
      YAV=(Y1+Y2)/2.0
C
      CNK = W1NK * W1NK * (1./X1 - 1./X2) *
1      (1./A1DHTB/A1DHTB - 1./A1NHTB/A1NHTB)
      BYNK=C1GRAV*(-X1*Z1NKH+XAV*Z1NHX+X2*Z1NKC-XAV*Z1DHX)
      YANK = Y1NKH/A1NK + Y1NKC/A1NK + Y1NHTB/A1NHTB + Y1DHTB/A1DHTB
C
C
C
      RE1=W1NK*X1NK/(A1NK*Y1)
      F1=FRDRAC(RE1)
      FNKH=F1*Y1NKH/(X1NK*X1*A1NK**2)
C
      RE2=W1NK*X1NK/(A1NK*Y2)
      F2=FRDRAC(RE2)
      FNKC=F2*Y1NKC/(X1NK*X2*A1NK**2)
C
      RE3 = W1NK * Y1KNI / (A1DHTB*YAV)
      F3=FRDRAC(RE3)
      FDHX = F3 * Y1DHTB / (Y1KNI*XAV*A1DHTB**2)
C
      RE4=W1NK*X1NHTB/(A1NHTB*YAV)
      F4=FRDRAC(RE4)
      FNHX=F4*Y1NHTB/(X1NHTB*XAV*A1NHTB**2)
C
      FNK=0.5*W1NK*W1NK*(FNKH+FNKC+FDHX+FNHX+F1DRNK/XAV)
C
      TMPUMP = S9MSTR - S1DRCS
      IF (TMPUMP.LT.0.0) THPUMP = 0.0
      CALL INTP9U (P2EMPD , T2EMPD , N2EMPD , TMPUMP , DPPUMP)
      DPPUMP = DPPUMP * P2REFP
      W1NKP = (DPPUMP+CNK+BYNK-FNK)/YANK
C
C
      WRITE(6,50) RE1,F1,FNKH,RE2,F2,FNKC,RE3,F3,FNKH
      50 FORMAT(9(1X,1PE12.4))
      60 FORMAT(6(2X,1PE12.4))
C
C
      RETURN
      END
*/ *****
*/
*/          OUTPUT DRACS RESULTS
*/
*/ *****
*INSERT WRIT1T.278
      WRITE(L9OUT,220) W1NA
220  FORMAT(/50X,27HPRIMARY DRACS FLOW RATE  =,E15.4,8X,6H(KG/S))
      WRITE(L9OUT,221) W1DHX

```



```

221  FORMAT(50X, 27H          TOTAL          =,E15.4,8X,6H(KG/S))
      WRITE(L9OUT,222) W1NK
222  FORMAT(/50X,27HSECONDARY DRACS FLOW RATE =,E15.4,8X,6H(KG/S))
      WRITE(L9OUT,223) W1A
223  FORMAT(/50X,27HHHX AIR FLOW RATE      =,E15.4,8X,6H(KG/S))
*INSERT WRIT1T.144
*CALL REF1V
*CALL VAR1I
*CALL VAR1V
*CALL DVDT1T
*INSERT PRNT6T.185
*CALL REF1V
*CALL VAR1I
*CALL VAR1V
*CALL DVDT1T
*CALL /YMGORC/
      DATA LMCSO /81/
*INSERT PRNT6T.205
C      WRITE(LMCSO,50500) S6COOL
50500 FORMAT(1H1,///, ' MASTER CLOCK =',1PE12.4)
      WRITE(L9OUT,51000) W1NA , W1DHX , T1NAI , T1NAO
C      WRITE(LMCSO,51000) W1NA , W1DHX , T1NAI , T1NAO
51000 FORMAT(/,15X,31H***** PRIMARY DRACS LOOP ***** ,/,
1      13X,6HW1NA ,6X,6HW1DHX ,6X,6HT1NAI ,6X,6HT1NAO ,/,
2      10X,1P4E12.4,/)
      WRITE(L9OUT,52000) W1NK , T1NKC , T1NKH
C      WRITE(LMCSO,52000) W1NK , T1NKC , T1NKH
52000 FORMAT(/,15X,33H***** SECONDARY DRACS LOOP ***** ,/,
1      13X,6HW1NK ,6X,6HT1NKC ,6X,6HT1NKH ,/,
2      10X,1P3E12.4,/)
      WRITE(L9OUT,53000) W1A , T1AI , T1AO , T1ST
C      WRITE(LMCSO,53000) W1A , T1AI , T1AO , T1ST
53000 FORMAT(/,15X,41H***** NATURAL DRAFT HEAT EXCHANGER ***** ,/,
1      13X,6HW1A ,6X,6HT1AI ,6X,6HT1AO ,6X,6HT1ST ,/,
2      10X,1P4E12.4,/)
      WRITE(L9OUT,53500)
C      WRITE(LMCSO,53500)
53500 FORMAT(/,15X,'***** PRIMARY DRACS PIPE TEMPERATURE*****',/)
      WRITE(L9OUT,54000) (T1DRCS(KKK),KKK=1,L1DRCS)
C      WRITE(LMCSO,54000) (T1DRCS(KKK),KKK=1,L1DRCS)
54000 FORMAT(13X,7HT1DRCS=,1P8E12.4/)
      WRITE(L9OUT,54100)
C      WRITE(LMCSO,54100)
54100 FORMAT(13X,'          HEAT EXCHANGER EXISTS.//)
      WRITE(L9OUT,54200) (T1DRCS(KKK),KKK=L1DRCS+1,N1DRCS)
C      WRITE(LMCSO,54200) (T1DRCS(KKK),KKK=L1DRCS+1,N1DRCS)
54200 FORMAT(13X,7HT1DRCS=,1P8E12.4,/)
      WRITE(L9OUT,54300)
C      WRITE(LMCSO,54300)
54300 FORMAT(/,15X,'***** SECONDARY DRACS PIPE TEMPERATURE*****',/)
      WRITE(L9OUT,54400) (T2DRCS(KKK),KKK=1,L2DRCS)
C      WRITE(LMCSO,54400) (T2DRCS(KKK),KKK=1,L2DRCS)
54400 FORMAT(13X,7HT2DRCS=,1P8E12.4/)
      WRITE(L9OUT,54500)
C      WRITE(LMCSO,54500)
54500 FORMAT(13X,'          HEAT EXCHANGER EXISTS.//)
      WRITE(L9OUT,54600) (T2DRCS(KKK),KKK=L2DRCS+1,N2DRCS)
C      WRITE(LMCSO,54600) (T2DRCS(KKK),KKK=L2DRCS+1,N2DRCS)
54600 FORMAT(13X,7HT2DRCS=,1P8E12.4,/)
*/ ----- PLOT -----
*DELETE TAJPLOT.5,6
1
1          T1NKH , T2DRCS(L2DRCS),
2          T1NKC , T2DRCS(N2DRCS) , T1AI , T1AO , T1ST ,
*/ -----IRACS MODIFY
*DELETE TAJPLOT.62,65
1          W2THRE(IW2THR+K) , W2ACS(IW2ACS+K) ,
2          W2SG (IW2SG +K) , W2AIR(IW2AIR+K)
*/ *****
*/
*/          COMMON
*/
*/ *****
*DELETE VAR1I.2,3

```

```

COMMON /VAR1I/
  1 L1FLAG , L1FLSP , L1DRAC , N9DRCS , IOPORC
*INSERT VAR1V.5
  4 , F1OLOP , S1DRCS
*COMDECK /YMGDRC/
C
C   COMMON FOR PRIMARY DRACS LOOP
C
COMMON /YMGDRC/
  1 A1NA , Y1NAH , Y1NAC , Z1NAH , Z1NAC , X1NA , X1DHSL ,
  2 P1EMPD(25) , T1EMPD(25) , P2EMPD(25) , T2EMPD(25) ,
  2 P1REFF , P2REFF ,
  2 A1LPL , Z1NKH , Z1NKC , Z1NHX , Y1NKC , Y1NKH , Y1NHTB ,
  3 Y1DHTB , X1NK , X1NHTB , A1NK , A1DHTB , A1NHTB , F1DRNK ,
  4 T1NKCR , T1NKHR , T1NAIR , T1NAOR , W1NKR , Y1KNI , Y1KNO ,
  5 Y1KNSI , C1NTB , STAUN , STAU
  6 ,G1DRCS(3,25) , G2DRCS(3,25) , T1DRCS(25) , T2DRCS(25) ,
  7 E1DRCS(25) , E2DRCS(25)
COMMON /YMGDR1/
  1 N1EMPD , N2EMPD
  2 ,N1DRCS , N2DRCS , L1DRCS , L2DRCS
*COMDECK /YMGSTK/
C
C   DATA FOR PRIMARY DRACS LOOP
C
COMMON /YMGSTK/
  1 S1TAUO , Z1STAK , C1AIR , A1STAK , Q1AIR , W1AR , T1AIR ,
  2 T1AOR , F1STAK , R1AREA , R1DFIN , R1DCOS , S1TAU , P2FANO ,
  3 P1FANH(25) , T1FANH(25)
COMMON /YMGSTI/
  1 N1DAMP , N1DCOS , N1DFIN , N1FANH
*DK DCLP1S
SUBROUTINE DCLP1S
C
C
C.....
C   A. YAMAGUCHI 1986
C
C / *CALL ***
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
C
C
T1INDC = T1NAI
T1OUDC = T1NAO
DO 100 K = 1 , L1DRCS
T1ORCS(K) = T1INDC
E1DRCS(K) = ENTH1H(T1ORCS(K))
100 CONTINUE
DO 200 K = L1DRCS+1 , N1DRCS
T1ORCS(K) = T1OUDC
E1DRCS(K) = ENTH1H(T1ORCS(K))
200 CONTINUE
RETURN
END
*DK DCLP2S
SUBROUTINE DCLP2S
C
C
C.....
C   A. YAMAGUCHI 1986
C
C / *CALL ***
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
C
C
T2INDC = T1NKH
T2OUDC = T1NKC
DO 100 K = 1 , L2DRCS
T2ORCS(K) = T2INDC
E2DRCS(K) = ENTH1H(T2ORCS(K))

```

```

100 CONTINUE
    DO 200 K = L2DRCS+1 , N2DRCS
        T2DRCS(K) = T20UDC
        E2DRCS(K) = ENTH1H(T2DRCS(K))
200 CONTINUE
    RETURN
    END
*DK DCLP1T
    SUBROUTINE DCLP1T(H,IDENT)
C
C
C.....
C    A. YAMAGUCHI 1986
C
C/ *CALL ***
*CALL VD9V
*CALL DATT6V
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
*CALL PCOEFV
*CALL DATX1I
*CALL DATX1V
*CALL DATX1P
*CALL DATX1L
    DIMENSION      F1BETA(1), T1NAS(1), P1PDHX(1), T1TUBE(1),
+ T1OUHX(1), Q1LOSS(1), T1INHX(1), T1BYP(1), N1ACTV(1),
+ E1INHX(1), E1OUHX(1), E1NAS(1), E1BYP(1), T1IN(1),
+ T1OUT(1), T2IN(1), T2OUT(1)
    EQUIVALENCE ( C9VDIM(1), F1BETA(1), T1NAS(1), P1PDHX(1),
+ T1TUBE(1), T1OUHX(1), Q1LOSS(1), T1INHX(1), T1BYP(1),
+ N1ACTV(1), E1INHX(1), E1OUHX(1), E1NAS(1), E1BYP(1),
+ T1IN(1), T1OUT(1), T2IN(1), T2OUT(1) )
C
    IF (IOPDRC.NE.0) GO TO 50
    IF (IDENT.EQ.1) THEN
        T1NAI=T1NAB
        IF (Z1JET.LT.Z1UPL) T1NAI=T1NAA
        T1DRCS(L1DRCS)=T1NAI
        E1DRCS(L1DRCS)=ENTH1H(T1NAI)
    END IF
    IF (IDENT.EQ.2) THEN
        T1DRCS(N1DRCS)=T1NAO
        E1DRCS(N1DRCS)=ENTH1H(T1NAO)
    END IF
    RETURN
C
50 CONTINUE
    IF (IDENT.EQ.1) THEN
        T1NAI=T1INHX(IT1INH+1)
        T1DRCS(L1DRCS)=T1NAI
        E1DRCS(L1DRCS)=ENTH1H(T1NAI)
    END IF
    IF (IDENT.EQ.2) THEN
        T1DRCS(N1DRCS)=T1NAO
        E1DRCS(N1DRCS)=ENTH1H(T1NAO)
    END IF
    RETURN
    END
*DK DCLP2T
    SUBROUTINE DCLP2T(H,IDENT)
C
C
C.....
C    A. YAMAGUCHI 1986
C
C/ *CALL ***
*CALL VAR1I
*CALL VAR1V
*CALL /YMGDRC/
C
C
    IF (IDENT.EQ.1) THEN
C

```

```

C   FOR THE FIRST NODE
C
      J = 1
      AREA = G2DRCS(1,J)
      DELTX = G2DRCS(2,J)
      TAV = 0.5 * (T1NKH + T2DRCS(J))
      RHOAV = DENS1D(TAV)
      EX = H * W1NK / RHOAV / AREA / DELTX
      E2DRCS(J) = (E2DRCS(J) + EX * ENTH1H(T1NKH)) / (1.0 + EX)
      T2DRCS(J) = TEMP1T(E2DRCS(J))
C
C   PRIMARY LOOP BEFORE HEAT EXCHANGER
C
      JFIRST = J + 1
      DO 100 J = JFIRST , L2DRCS
      AREA = G2DRCS(1,J)
      DELTX = G2DRCS(2,J)
      TAV = 0.5 * (T2DRCS(J-1) + T2DRCS(J))
      RHOAV = DENS1D(TAV)
      EX = H * W1NK / RHOAV / AREA / DELTX
      E2DRCS(J) = (E2DRCS(J) + EX * E2DRCS(J-1)) / (1.0 + EX)
      T2DRCS(J) = TEMP1T(E2DRCS(J))
100  CONTINUE
      END IF
      IF (IDENT.EQ.2) THEN
C
C   FOR THE FIRST NODE
C
      J = L2DRCS + 1
      AREA = G2DRCS(1,J)
      DELTX = G2DRCS(2,J)
      TAV = 0.5 * (T1NKC + T2DRCS(J))
      RHOAV = DENS1D(TAV)
      EX = H * W1NK / RHOAV / AREA / DELTX
      E2DRCS(J) = (E2DRCS(J) + EX * ENTH1H(T1NKC)) / (1.0 + EX)
      T2DRCS(J) = TEMP1T(E2DRCS(J))
C
C   PRIMARY LOOP BEFORE HEAT EXCHANGER
C
      JFIRST = J + 1
      DO 200 J = JFIRST , N2DRCS
      AREA = G2DRCS(1,J)
      DELTX = G2DRCS(2,J)
      TAV = 0.5 * (T2DRCS(J-1) + T2DRCS(J))
      RHOAV = DENS1D(TAV)
      EX = H * W1NK / RHOAV / AREA / DELTX
      E2DRCS(J) = (E2DRCS(J) + EX * E2DRCS(J-1)) / (1.0 + EX)
      T2DRCS(J) = TEMP1T(E2DRCS(J))
200  CONTINUE
      END IF
      RETURN
      END

```

表 C - 6 SSC - P IHX 温度分布图作成用

```

*IO PLTIHX
*BEFORE PRNT9T.250
  WRITE(74) S1FLOW
  WRITE(91,*) '**** M.C.T = ',S1FLOW,' SEC. '
*BEFORE PRNT1T.156
*CALL DTT12P
*CALL DTT12L
  DIMENSION      T1DOWN(1),  E1DOWN(1),  T1SHEL(1),  A1WALL(1),
+   B1WALL(1),  V1OLNA(1),  F1THIK(1),  T1WALL(1),  A2WALL(1),
+   B2WALL(1),  V2OLNA(1),  F2THIK(1),  T2WALL(1)
  EQUIVALENCE    ( C9VDIM(1), T1DOWN(1),  E1DOWN(1),  T1SHEL(1),
+   A1WALL(1),  B1WALL(1),  V1OLNA(1),  F1THIK(1),  T1WALL(1),
+   A2WALL(1),  B2WALL(1),  V2OLNA(1),  F2THIK(1),  T2WALL(1) )
*CALL SCRH1V
*CALL SCRH1I
*CALL SCRH1P
*CALL SCRH1L
  DIMENSION      USUBPT(1),  USUBST(1),  EPDOWN(1),  TPDOWN(1),  E1PNA(1),
+   T1PNA(1),  T1PSH(1),  T1PTUB(1),  T1PWAL(1),  T2PWAL(1),  E2PNA(1),
+   E1PNAS(1),  T1PNAS(1)
  EQUIVALENCE    (
+   C9VDIM(1),  USUBPT(1),  USUBST(1),  EPDOWN(1),  TPDOWN(1),  E1PNA(1),
+   T1PNA(1),  T1PSH(1),  T1PTUB(1),  T1PWAL(1),  T2PWAL(1),  E2PNA(1),
+   E1PNAS(1),  T1PNAS(1) )
*INSERT PRNT1T.213
  WRITE(74) ((T1NA(I)),I=1ST1,LGT1)
  WRITE(91,*) ((T1NA(I)),I=1ST1,LGT1)
*INSERT PRNT1T.215
  WRITE(74) T10UHX(IT10UH+K)
  WRITE(91,*) T10UHX(IT10UH+K)
*INSERT PRNT1T.247
  WRITE(74) T20UHX(IT20UH+K)
  WRITE(91,*) T20UHX(IT20UH+K)
*INSERT PRNT1T.252
  WRITE(74) ((T1NAS(I)),I=1ST,LGT)
  WRITE(91,*) ((T1NAS(I)),I=1ST,LGT)
  ITB1=IT1TUB+(K-1)*LT1TLP+1
  ITS1=IT1SHE+(K-1)*LT1SLP+1
  ITB2=ITB1+NIHX-1
  ITS2=ITS1+NIHX-1
  WRITE(74) (T1TUBE(I),I=ITB1,ITB2)
  WRITE(91,*) (T1TUBE(I),I=ITB1,ITB2)
  WRITE(74) (T1SHEL(I),I=ITS1,ITS2)
  WRITE(91,*) (T1SHEL(I),I=ITS1,ITS2)

```

表 C - 7 SSC - P 外部バイパスモデルの改良

```

*ID BPZERO
*/ *****
*/      SUBROUTINE BPAS1S
*/ *****
*DELETE BPAS1S.99,100
CN      F1LSBP = ((DPBP - RHO * C9GRAV * Z1BPOU) * RHO) / (W1BPAS * ABS(
CN      X      W1BPAS))
*/ *****
*/      SUBROUTINE BPAS1T
*/ *****
*DELETE PFIX.2
CN      IF(W1BPAS.LE.0.0.AND.F1WBPS.LT.0.0) F1WBPS=0.0
          F1WBPS=0.0
          W1BPAS=0.0
*/ *****
*/      SUBROUTINE VESL1T
*/ *****
*DELETE VESL1T.313,314
CN      PINV = PINV + (P1BPOU+P1BPG+P1BPWW) / X1BPAS
CN      DENOM = DENOM + 1. / X1BPAS
    
```

表 C - 8 SSC - P チェックバルブモデルの改良

```

*ID CHECK
*/ *****
*/      SUBROUTINE ZONE1T
*/ *****
*INSERT ZONE1T.77
C***** YMG DEBUG
        IF(L6NIX.EQ.1) GO TO 1
C***** YMG DEBUG
*/ *****
*/      SUBROUTINE CVAL1T
*/ *****
*DELETE CVAL1T.212,215
C*****
C  YMG REVISED
C  FOLLOWING FOUR LINES ARE SKIPPED
C  IF(WNORM.GT.0.5)  GOTO 50
C  C = F1CVL(2) + F1CVL(3)*WNORM
C  IF(WNORM.GT.0.08) GOTO 50
C  C = F1CVL(4) + F1CVL(5)*WNORM
    
```

表 C - 9 SSC - P コールドプール圧損モデル

```

*ID PDPOOL
*/ *****
*/ SUBROUTINE PRES1T
*/ *****
*INSERT PRES1T.161
  WDIF=0.0
  DO 5000 K=1,N1PATH
    IF (J1BREK(IJ1BRE+K).EQ.0) THEN
      WDIF=WDIF+W1IH(X(IW1IH+K))*F1LMPX(IF1LMX+K)
      -W1PUMP(IW1PUM+K)*F1LMPP(IF1LMP+K)
    1 ELSE
      WDIF=WDIF+W1IH(X(IW1IH+K))*F1LMPX(IF1LMX+K)
      -W1DOBK(IW1DOB+K)*F1LMPP(IF1LMP+K)
    1 END IF
  5000 CONTINUE
*INSERT PRES1T.166
  1 + 200.0*WDIF*ABS(WDIF)/A1CPG/A1CPG/D1COPL
*INSERT PFX1.114
  1 + 200.0*WDIF*ABS(WDIF)/A1CPG/A1CPG/D1COPL
*/ *****
*/ SUBROUTINE VESL1T
*/ *****
*INSERT VESL1T.240
  WDIF=0.0
  DO 5000 K=1,N1PATH
    IF (J1BREK(IJ1BRE+K).EQ.0) THEN
      WDIF=WDIF+W1IH(X(IW1IH+K))*F1LMPX(IF1LMX+K)
      -W1PUMP(IW1PUM+K)*F1LMPP(IF1LMP+K)
    1 ELSE
      WDIF=WDIF+W1IH(X(IW1IH+K))*F1LMPX(IF1LMX+K)
      -W1DOBK(IW1DOB+K)*F1LMPP(IF1LMP+K)
    1 END IF
  5000 CONTINUE
*INSERT VESL1T.246
  1 + 200.0*WDIF*ABS(WDIF)/A1CPG/A1CPG/D1COPL

```


付録D 本作業で使用したデータセット

本作業で使用したデータセットのメンバーと内容を以下に示す。

(ID名: PA340 はすべて省略した)

1. SSCL88. CNTL

SSC-Lに関するジョブコン

JDBG サブルーチン単位でソースを修正しロードモジュールを作る
JHCL ヒストリアン、コンパイル、ロードモジュール作成
JIHW IHXの温度分布図に使用する流量出力のロードモジュール作成
JPLTLINK IHXの温度分布図作成プログラムのロードモジュール作成
J1,J2 SSC-L実行ジョブコン

2. SSCL88. DATA

SSC-Lの入力データ

GIHX IHX温度分布図作成用入力データ
MDRCS 付録B参照
MIHX "
MRST リスタート用入力データ
NDRCNEW 付録B参照
NDRCS "
NIHX "
NIHXNEW "
PDRCS DRACSテスト計算プロット入力データ
PIHX IHX "

3. SSCL88. HIST.DATA

SSC-Lに関するコレクションセット、内容は付録Cを参照されたい。

4. SSCL88. CLIST

SSC - Lに関するCLIST

G1 SSC - L実行CLIST

G2 SSC - Lリスタート実行CLIST

IHXW IHX 温度分布図に使用する流量出力プログラムの実行CLIST

P SSC - Lプロットプログラム実行CLIST

PG IHX 温度分布図作成プログラム実行CLIST

5. SSCL88. LOAD

SSC - Lロードモジュール

SSCL88 今回の最新ロードモジュール

6. SSCP88. CNTL

SSC - Pに関するジョブコン

JDBG サブルーチン単位でソースを修正し、ロードモジュールを作る

JHCL ヒストリアン、コンパイル、ロードモジュール作成

JIHXW IHX の温度分布図に使用する流量出力のロードモジュール作成

JPLTLNK IHX の温度分布図作成プログラムのロードモジュール作成

7. SSCP88. DATA

SSC - Pの入力データ

GIHX IHX 温度分布図作成用入力データ

MRST リスタート用入力データ

MTRIP 付録B参照

NDRCS "

NIHX "

NIHXNEW "

NPRCS "

PDRCS DRACS テスト計算プロット入力データ

PIHX IHX "

8. SSCP88. HIST.DATA

SSC - Pに関するコレクションセット、内容は付録Cを参照されたい。

9. SSCP88. CLIST

SSC - Pに関するCLIST

G1 SSC - P実行CLIST

G2 SSC - Pリスタート実行CLIST

IHXW IHX温度分布図に使用する流量出力プログラムの実行CLIST

P SSC - Pプロットプログラム実行CLIST

PG IHX温度分布図作成プログラム実行CLIST

10. SSCP88. LOAD

SSC - Pロードモジュール

SSCP88 今回の最新ロードモジュール

11. GRAPH.FORT

グラフィック関係のプログラムソース

LIHXW SSC - L IHX温度分布図に使用する流量出力

LLIB SSC - L IHX温度分布図作成

LMAIN "

PIHXW SSC - P IHX温度分布図に使用する流量出力

PLIB SSC - P IHX温度分布図作成

PMAIN "

12. GRAPH.LOAD

グラフィック関係のロードモジュール

LIHX SSC - L IHX温度分布図作成

LIHXW SSC - L IHX温度分布図に使用する流量出力

PIHX SSC - P IHX温度分布図作成

PIHXW SSC - P IHX温度分布図に使用する流量出力