

本資料は 年 月 日付けで登録区分、
変更する。 2001. 6. -6

[技術情報室]

社内一般

地層中核種移行計算コード「FEMWASTE-PNC」の開発

1990年3月

動力炉・核燃料開発事業団
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大字村松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)
2001



地層中核種移行計算コード 「FEMWASTE-PNC」の開発

*畑中耕一郎, *梅木 博之, *大井 貴夫

*根山 敦史, *仲島 邦彦

要 旨

目 的

ORNL で開発, 公開されている地層中核種移行計算コード「FEMWASTE」の改良を行い, 機能を拡張する。また, 改良したコードの検証を行い, その妥当性を確かめる。

方 法

「FEMWASTE」で, その数理モデルに考慮していない湧き出し及び崩壊系列の効果を新たに組み込み, コーディングを行い, 他のコードや解析解との比較計算を行う。

結 果

「FEMWASTE」に対して, 他のコードや解析解の同一条件での計算の比較を実施すると同時に, 境界条件の経時変化, 地表面に放出される核種量の計算, 非定常計算のリスタートも取り扱えるようにした。さらに入力データガイドも作成し, 入力データ作成時に参照出来るようにした。検証に関しては, 伝熱計算コード「TRUMP」で湧き出しの計算, また, Bateman の式の解析解で崩壊系列の計算の妥当性を確かめる為に比較計算を行った。結果は両ケースとも良く一致した。

結 論

地層処分システムの性能評価をより現実的に行う事が可能となった。しかし, 計算領域が2次元である, 線型吸着しか取り扱えない等の点については今後改良の予定である。なお, 今回開発したコードに「FEMWASTE-PNC」と命名した。

目 次

1. 序 言	1
2. 「FEMWASTE」の概念	3
2.1 数理モデル	3
2.2 初期条件	5
2.3 境界条件	5
2.3.1 Dirichlet型境界条件	6
2.3.2 Neumann型境界条件	6
2.3.3 Cauchy型境界条件	6
2.4 有限要素近似	7
2.4.1 空間依存の場合	7
2.4.2 時間依存の場合	9
2.4.3 基本関数と上流法	9
2.4.4 数値積分	11
2.5 モジュールの構成	13
3. 「FEMWASTE-PNC」の概念	17
3.1 機能追加	17
3.2 数理モデル	18
3.3 初期条件	19
3.4 境界条件	20
3.4.1 Dirichlet型境界条件	20
3.4.2 Neumann型境界条件	20
3.4.3 Cauchy型境界条件	20
3.5 有限要素近似	21
3.6 モジュールの構成	23
4. モジュールの詳細	26
4.1 Program MAIN	26
4.2 Subroutine GM2DXZ	26
4.3 Subroutine DATAIN	30
4.4 Subroutine AFABTA	31
4.5 Subroutine FLUX	32
4.6 Subroutine Q4D	33

4.7	Subroutine ASEMBL	34
4.8	Subroutine Q4	35
4.9	Subroutine SHAPE	36
4.10	Subroutine BC	36
4.11	Subroutine Q4SP	37
4.12	Subroutine SOLVE	38
4.13	Subroutine SFLOW	39
4.14	Subroutine Q4R	40
4.15	Subroutine PRINTT	41
4.16	Subroutine STORE	42
4.17	Subroutine QINT	43
4.18	Subroutine BFLUX	44
4.19	Subroutine AROSR	45
4.20	Subroutine SRFLUX	46
4.21	Subroutine MASS	47
4.22	Subroutine TOTC	48
5.	ファイルの構成	49
5.1	「FEMWASTE-PNC」の関連ファイル	49
5.2	「FEMWASTE-PNC」用の入力データファイル	51
5.3	幾何学データ及び流れ場のデータファイル	51
5.4	リスタート用の入力ファイル及び出力ファイル	52
5.5	標準出力のデータファイル	53
5.6	ポスト処理用のデータファイル	53
6.	検 証	55
6.1	湧き出し	55
6.1.1	計算モデル	55
6.1.2	初期条件	55
6.1.3	境界条件	55
6.1.4	その他の計算条件	56
6.1.5	計算結果	56
6.2	崩壊系列	58
6.2.1	Bateman の方程式の解析解	58
6.2.2	計算モデル	58
6.2.3	初期条件	58

6.2.4	境界条件	59
6.2.5	その他の計算条件	59
6.2.6	計算結果	60
7.	結 言	64
APPENDIX A	要素の有限要素式の誘導	65
APPENDIX B	入力仕様	71
APPENDIX C	ソースプログラムリスト	87

地層中核種移行計算コード
「FEMWASTE-PNC」の開発

1. 序 言

多孔質媒体中を移行する溶解物質の輸送の2次元非定常計算コード「FEMWASTE」はアメリカのオークリッジ国立研究所(ORNL)で開発され、公開されている。本コードは現在までに拡張と修正が繰り返し行われて来ており、その機能も豊富なものとなっている。定式化に用いられている数値モデルは移流、分散の項を含んだ移流分散方程式に、吸着、崩壊の項を付加したものである。吸着に関しては線形吸着(ヘンリー型)を採用しており、崩壊に関しては、崩壊系列は考慮しておらず、崩壊した核種は全て消滅してしまうと仮定している。

数値モデルの定式化にあたって、空間領域に関しては、有限要素法が用いられており、使用要素は一次のアイソパラメトリック要素である。近似法はガラーキン法と上流法とを取り扱っている。時間領域に関しては、差分法が用いられており、クランクニコルソン法、後退差分法、中間差分法を取り扱っている。「FEMWASTE」は、このように種々の解法及び機能を有するが、地層処分システムの性能評価計算に使用する場合、次に示すような不備な点が見られる。

- ① 崩壊系列を考慮していない(一回の崩壊のみ)。
- ② 湧き出し(ソースターム)が考慮されていない。
- ③ 境界条件の経時変化が取り扱えない。
- ④ 地表面に放出される核種の量を計算していない。
- ⑤ リスタート機能を有しない。

これらの点を整備する為に東海事業所環境技術開発部地層処分開発室では、「FEMWASTE-PNC」の開発及び検証を実施した。図-1-1は「FEMWASTE-PNC」と入力情報及び出力情報の関係を示している。

本報告書では「FEMWASTE」に導入されている数値モデル、境界条件及び機能について述べた後、今回、開発を行った「FEMWASTE-PNC」の数値モデル、機能、「FEMWASTE」の改良項目、改良内容、検証について述べる。また、APPENDIX A, B, Cに、それぞれ、要素の有限要素式の誘導、「FEMWASTE-PNC」の入力仕様、「FEMWASTE-PNC」のソースプログラムリストを記述する。

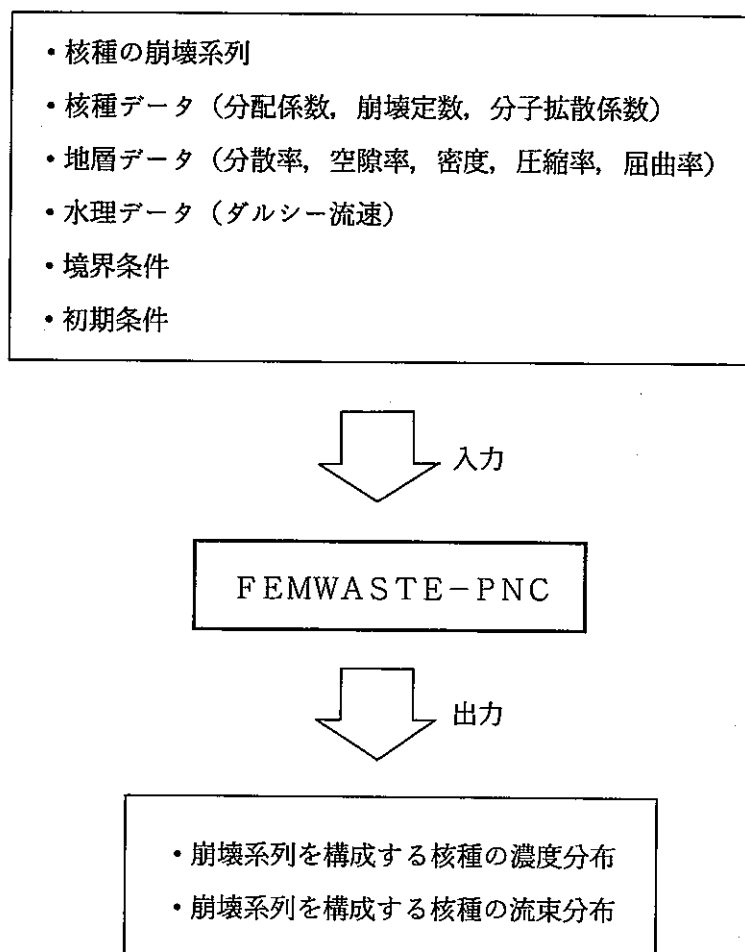


図-1-1 FEMWASTE-PNC の入出力情報

2. 「FEMWASTE」の概念

2.1 数理モデル

「FEMWASTE」に使用されている、地表面下の多孔質媒体中を移行する溶解物質を表現する方程式は、二次元領域において、移流、分散、吸着、崩壊の影響を考慮して、質量保存則から誘導されている。この方程式は (2.1) 式のように書かれる。

$$\begin{aligned} \frac{\partial}{\partial t} (\theta c + \rho s) + (\theta c + \rho s) \alpha' \frac{\partial h}{\partial t} = & - \left(\frac{\partial V_x c}{\partial x} + \frac{\partial V_z c}{\partial z} \right) \\ & + \left[\frac{\partial}{\partial x} (\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z}) + \frac{\partial}{\partial z} (\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z}) \right] \\ & - \lambda (\theta c + \rho s) \end{aligned} \quad (2.1)$$

- ここに、 θ : 体積含水率 (-)
 c : 液相中に溶解している核種の濃度 (M/L³)
 ρ : 固相の見かけ密度 (M/L³)
 s : 固相に吸着した核種の濃度 (M/M)
 α' : 媒体の修正圧縮率 (1/L)
 h : 地下水の圧力水頭 (L)
 $D_{xx}, D_{xz}, D_{zx}, D_{zz}$: 分散係数 (L²/T)
 V_x, V_z : x方向, z方向のダルシー流速 (L/T)
 λ : 核種の崩壊定数 (1/T)
 x, z : 空間座標 (L)
 t : 時間座標 (T)

(2.1) 式に関して、左辺第一項は、単位体積中の全質量の時間変化、左辺第二項は、圧力で体積が変化する事による全質量の時間変化を表している。また、右辺第一項と第二項は、移流と分散により単位時間、単位体積中に流入あるいは流出するフラックスの変化量を表しており、右辺第三項は核種の崩壊による質量減少を表している。(2.1) 式中的変数 θ , h , V_x , V_z の値は地下水理の計算結果から得られる。

分散係数は流れの場と媒体特性に依存している量であり、次式のように示される。

$$\theta D_{xx} = a_T V + \frac{(a_L - a_T) V_x^2}{V} + D_m \cdot T \quad (2.2 a)$$

$$\theta D_{zz} = \theta D_{zx} = \frac{(a_L - a_T) V_x V_z}{V} \quad (2.2 b)$$

$$\theta D_{zz} = a_T V + \frac{(a_L - a_T) V_z^2}{V} + D_m \cdot T \quad (2.2 c)$$

$$V = \sqrt{V_x^2 + V_z^2} \quad (2.2 d)$$

ここに, a_T : 流れと垂直方向の分散率 (L)

a_L : 流れと平行方向の分散率 (L)

T : 媒体の屈曲率 (—)

D_m : 地下水中の核種の分子拡散係数 (L^2/T)

(2.1) 式において, x, z, t は独立変数であり, c, s は従属変数である。よって, (2.1) 式のままだでは1つの方程式に対して2つの従属変数を有する事になり, 解析的にあるいは数値的には解く事ができない。よって, (2.1) 式が解をもつ為にはもう一つの式が必要である。そこで, 液相に溶解している核種と固相に吸着している核種の平衡状態が瞬時に生じると仮定し, これを線型の方程式で表現すると, (2.3) 式に示すような線型吸着式 (ヘンリー型) になる。

$$s = Kd \cdot c \quad (2.3)$$

ここに, Kd : 核種の分配係数 (L^3/M)

この仮定は, 流速, 拡散係数が非常に小さく, 核種の溶解した液体が, 固相と接触して平衡反応に達する時間が十分にあることが考えられるときに妥当である。

(2.3) 式を (2.1) 式に代入すると従属変数が c のみになった核種の移行方程式 (2.4) が得られる。

$$\begin{aligned} \theta R d \frac{\partial c}{\partial t} = & - \left(\frac{\partial V_x c}{\partial x} + \frac{\partial V_z c}{\partial z} \right) \\ & + \left[\frac{\partial}{\partial x} (\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z}) + \frac{\partial}{\partial z} (\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z}) \right] \\ & - \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \theta R d \right) c \end{aligned} \quad (2.4)$$

$$R d = 1 + \frac{\rho K d}{\theta} \quad (2.5)$$

$R d$ は遅延係数であり, 地下水中に溶解した核種の移行の遅れを示す指標である。

2.2 初期条件

核種移行の計算は通常、非定常で行われるので、計算に際して、初期条件を設定しなければならない。いま、図-2-1に示すように、 R を計算対象領域とする。 R は2次元の一般的な閉領域を示しており、図-2-1中の B_1 、 B_2 、 B_3 は境界条件を指定する境界領域である。 B_1 、 B_2 、 B_3 は後述するように、それぞれDirichlet型、Neumann型、Cauchy型の境界条件を指定する領域である。 R において(2.4)式が成り立って仮定すると、その初期条件は(2.6)式で示される。

$$c = c_0(x, z, t=0) \quad \text{in } R \quad (2.6)$$

c_0 は R における核種濃度の初期値である。「FEMWASTE」では通常、初期条件として領域の濃度を0として計算を開始する。

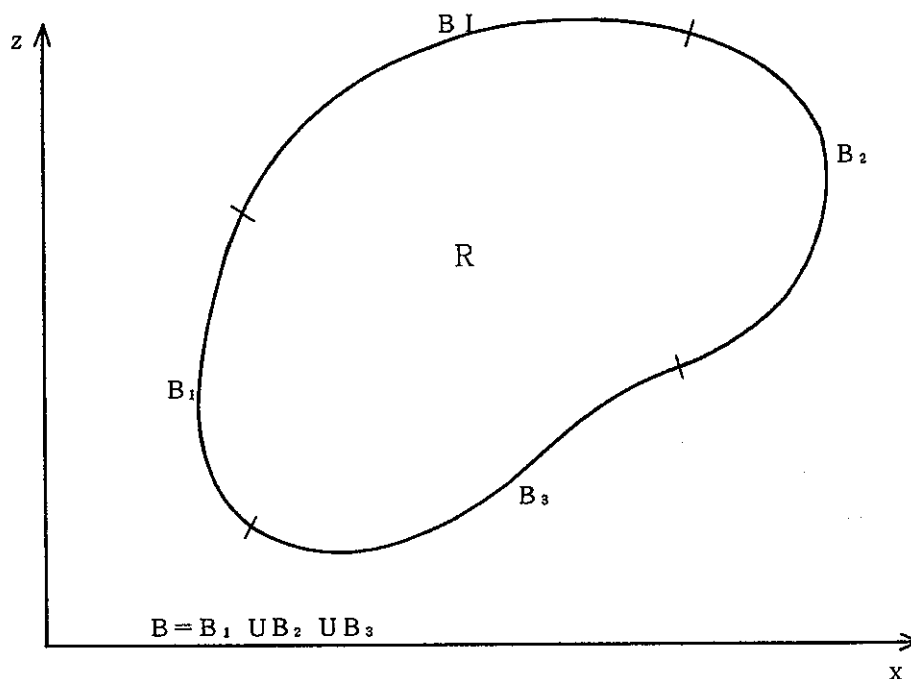


図-2-1 計算対象領域の概念図

2.3 境界条件

領域 R で式(2.4)が成り立つ時、この領域の核種濃度分布を解析的、あるいは、数値的に解く場合、初期条件と同時に境界条件を設定しなければならない。「FEMWASTE」の場合、境界条件として、次の三条件を指定する事ができる。

- ① Dirichlet型境界条件 (図-2-1中の B_1)
- ② Neumann型境界条件 (図-2-1中の B_2)

③ Cauchy 型境界条件 (図-2-1中のB₃)

図-2-1におけるBは全境界を示しており、全境界は必ず①～③の境界条件で設定されなければならない。なお、「FEMWASTE」では、これらの境界条件の時間変化は取り扱えない。

2.3.1 Dirichlet 型境界条件

本境界条件は、対象となる境界において核種濃度を固定するものである。式で表すと(2.7)式のようになる。

$$c = c_1(x, z) \quad \text{on } B_1 \quad (2.7)$$

B₁は全境界Bの一部である。また、c₁は濃度の固定値であり、場所の関数である。

「FEMWASTE」では境界節点番号とその節点に対応する濃度固定値を指定するようになっている。

2.3.2 Neumann 型境界条件

本境界条件は、対象となる境界の法線方向から、計算領域へ分散により、流入あるいは流出する核種のフラックスを規定するものである。式で表すと(2.8)式のようになる。

$$\begin{aligned} & -(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z} - V_x c) n_x - (\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z} - V_z c) n_z \\ & = q_2(x, z) + (V_x n_x + V_z n_z) c \quad \text{on } B_2 \quad (2.8) \end{aligned}$$

B₂は全境界Bの一部であり、n_x、n_zはB₂上の法線ベクトルの方向余弦である。q₂は分散により、流入あるいは流出する核種フラックスの値であり、場所の関数である。「FEMWASTE」では、核種は常に境界において法線方向の流速のみにより運ばれ、濃度勾配によっては核種は運ばれないと考えており、このように仮定できる境界要素の番号と節点番号を指定するようになっている。

2.3.3 Cauchy 型境界条件

本境界条件は対象となる境界の法線方向から、計算領域へ、移流、分散により、流入あるいは流出する核種のフラックスを規定するものである。式で表すと(2.9)式のようになる。

$$\begin{aligned} & -(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z} - V_x c) n_x - (\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z} - V_z c) n_z \\ & = q_3(x, z) \quad \text{on } B_3 \quad (2.9) \end{aligned}$$

B₃は全境界Bの一部であり、n_x、n_zはB₃上の法線ベクトルの方向余弦である。q₃は移流、分散により流入、流出する核種フラックスの値であり場所の関数である。「FEMWASTE」では、境界要素を構成する節点番号とその節点に対応する核種のフラックスを指定する。

2.4 有限要素近似

「FEMWASTE」は、数理モデルの定式化にあたって、空間領域に関しては、有限要素法が用いられており、時間領域に関しては有限差分法が用いられている。本節では有限要素法による定式化の手順及び時間方向の差分の方法について簡単に述べる。

2.4.1 空間依存の場合

図-2-1における空間領域Rを構成する要素において(2.1)式が成り立つと仮定する。いま、(2.4)式の右辺の項を全て左辺に移項すると、(2.10)式のようになる。

$$\begin{aligned}
 L(c) = & \theta R d \frac{\partial c}{\partial t} + \left(\frac{\partial V_x c}{\partial x} + \frac{\partial V_z c}{\partial z} \right) \\
 & - \left[\frac{\partial}{\partial x} \left(\theta D_{xx} \frac{\partial c}{\partial x} + \theta D_{xz} \frac{\partial c}{\partial z} \right) + \frac{\partial}{\partial z} \left(\theta D_{zx} \frac{\partial c}{\partial x} + \theta D_{zz} \frac{\partial c}{\partial z} \right) \right] \\
 & + \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \theta R d \right) c = 0 \quad (2.10)
 \end{aligned}$$

また、「FEMWASTE」では、有限要素の離散化に際して、線型の基本関数を仮定した四角形要素を採用している。よって1つの四角形要素における厳密解cは(2.11)式のように近似解 \hat{c} で近似されることになる。

$$c \cong \hat{c} = \sum_{j=1}^4 c_j(t) \cdot N_j = [N] \{c(t)\} \quad (2.11)$$

ここに、[N]は節点濃度と要素内部濃度を結びつける基本関数マトリックスであり、{c(t)}は時刻tにおける要素の節点ベクトルである。

(2.10)式のcを \hat{c} と置き換え、(2.11)式を代入した式に、[W]を重み関数マトリックスとして、重みつき残差法を適用すると(2.12)式を得る。

$$\int_{R_e} [W]^T [L(\hat{c})] dR = 0 \quad (2.12)$$

(2.12)式を整理し、まとめると要素の有限要素式(2.13)になる。

$$[m] \{\dot{c}\} + [s] \{c\} + \{q\} = 0 \quad (2.13)$$

ここに [m] : 要素質量マトリックス

[s] : 要素剛性マトリックス

{q} : 要素境界の流束ベクトル

{c} : 要素節点濃度ベクトル

[m], [s], {q}, {c} はそれぞれ(2.14)～(2.17)で示される。

$$[m] = \int_{R_e} \theta R d [W]^T [N] dR \quad (2.14)$$

$$\begin{aligned} [s] = \int_{R_e} & \left[\frac{\partial [W]}{\partial x} \left(\theta D_{xx} \frac{\partial [N]}{\partial x} + \theta D_{xz} \frac{\partial [N]}{\partial z} \right) + \frac{\partial [W]^T}{\partial z} \left(\theta D_{zx} \frac{\partial [N]}{\partial x} + \theta D_{zz} \frac{\partial [N]}{\partial z} \right) \right. \\ & - \left(\frac{\partial [W]^T}{\partial x} V_x [N] + \frac{\partial [W]^T}{\partial z} V_z [N] \right) \\ & \left. + [W]^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \cdot \theta R d \right) [N] \right] dR \end{aligned} \quad (2.15)$$

$$\begin{aligned} [q] = \int_{B_e} & [W]^T \left[\left(-\theta D_{xx} \frac{\partial c}{\partial x} - \theta D_{xz} \frac{\partial c}{\partial z} + V_x c \right) n_x \right. \\ & \left. + \left(-\theta D_{zx} \frac{\partial c}{\partial x} - \theta D_{zz} \frac{\partial c}{\partial z} + V_z c \right) n_z \right] dB \end{aligned} \quad (2.16)$$

$$\{\dot{c}\} = \frac{\partial \{c(t)\}}{\partial t} \quad (2.17)$$

詳しい誘導は APPENDIX A に記述されている。

(2.13) 式を全体の空間領域 R について加え合わせると全体の有限要素式 (2.18) が得られる。

$$[M] \{\dot{C}\} + [S] \{C\} + \{Q\} = 0 \quad (2.18)$$

ここに、[M] : 全体の質量マトリックス

[S] : 全体の剛性マトリックス

{Q} : 全体境界の流束ベクトル

{C} : 全体の節点濃度ベクトル

(2.18) 式においては、内部境界 (要素間境界) の積分の寄与は隣接する要素どうしで打ち消し合う。これは、積分を反時計回りに定義したとき、隣接する要素の共有する辺を互いに逆方向に積分する事による (図-2-2 参照)。この為に、内部境界から生じる境界項は (2.18) 式に含まれない。

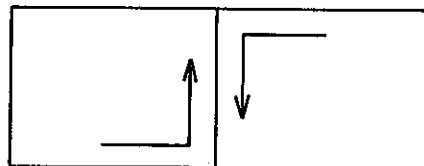


図-2-2 領域内要素境界上での積分

2.4.2 時間依存の場合

時間領域において用いられている近似解法は有限差分法である。「FEMWASTE」では、時間領域の差分において、三種類解法が採用されている。これらは、中心差分法（クラנקニコルソン法）、後退差分法、中間差分法である。

i) 中心差分法（クラークニコルソン法）

$$[M] \frac{(\{c(t+\Delta t)\} - \{c(t)\})}{\Delta t} + \frac{1}{2} \cdot [S] (\{c(t+\Delta t)\} + \{c(t)\}) + \{Q\} = 0 \quad (2.19)$$

ここに、[M], [S], {Q} は $t = t + \frac{\Delta t}{2}$ において評価される。

ii) 後退差分法

$$[M] \frac{(\{c(t+\Delta t)\} - \{c(t)\})}{\Delta t} + [S] \{c(t+\Delta t)\} + \{Q\} = 0 \quad (2.20)$$

ここに、[M], [S], {Q} は $t = t + \Delta t$ において評価される。

iii) 中間差分法

$$\left(2 \frac{[M]}{\Delta t} + [S]\right) \{c(t + \frac{\Delta t}{2})\} - \frac{2}{\Delta t} [M] \{c(t)\} + \{Q\} = 0 \quad (2.21)$$

$$\{c(t + \Delta t)\} = 2 \{c(t + \frac{\Delta t}{2})\} - \{c(t)\} \quad (2.22)$$

ここに、[M], [S], {Q} は $t = t + \frac{\Delta t}{2}$ において評価される。

この計算法は未知数ベクトル $\{c(t + \Delta t)\}$ が時間増分 Δt の間に時間とともに、線形的に変化すると仮定している。

2.4.3 基本関数と上流法

「FEMWASTE」の使用要素は、4節点をもった双一次のアイソパラメトリック四角形要素である。i番目の節点における基本関数は、正規化された直交座標系で(2.23)式のように書ける。

$$N_i = \frac{1}{4} (1 + \xi_i \xi) (1 + \eta_i \eta) \quad i = 1, 2, 3, 4. \quad (2.23)$$

ここに、 ξ_i と η_i は図-2-3に示されるように、節点番号が反時計方向に1~4までつけられている要素の局所座標である。

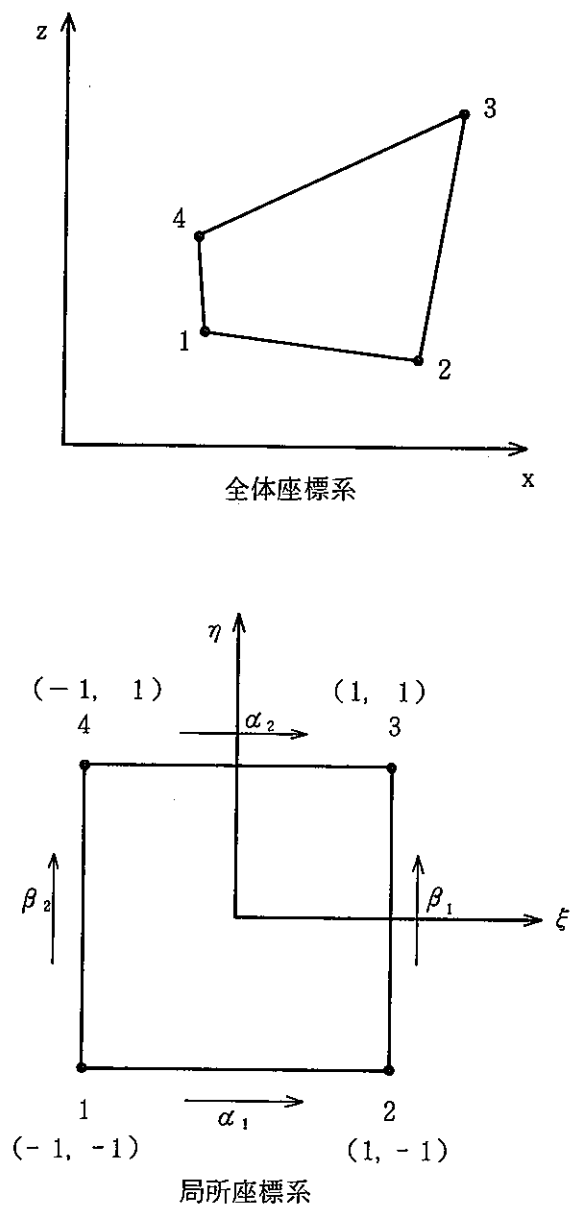


図-2-3 全体座標系と局所座標系における有限要素

(2.4) 式における移流項が分散項と同様に影響を及ぼすようになると、数値的に解が収束しにくくなる。この場合、基本関数と異なる重み関数を使用する事により、数値的な困難は最小限に抑えられる。「FEMWASTE」では重み関数の選択に関して、ガラーキン法と上流法が取り扱えるが、移流の影響による数値的な困難を最小限に抑える為には上流法が有効である。

ガラーキン法は重み関数 W_i に基本関数を用いて解を求める方法である。

$$[W] = [N] \quad (2.24)$$

上流法は重み関数を (2.25) 式のように定義して解を求める方法である。

$$\begin{aligned} W_1 &= \frac{1}{16} [(1+\eta)(3\beta_2\eta - 3\beta_2 - 2) + 4] [(1+\xi)(3\alpha_1\xi - 3\alpha_1 - 2) + 4] \\ W_2 &= \frac{1}{16} [(1+\eta)(3\beta_1\eta - 3\beta_1 - 2) + 4] [(1+\xi)(-3\alpha_1\xi + 3\alpha_1 + 2)] \\ W_3 &= \frac{1}{16} [(1+\eta)(-3\beta_1\eta + 3\beta_1 + 2)] [(1+\xi)(-3\alpha_2\xi + 3\alpha_2 + 2)] \\ W_4 &= \frac{1}{16} [(1+\eta)(-3\beta_2\eta + 3\beta_2 + 2)] [(1+\xi)(3\alpha_2\xi - 3\alpha_2 - 2) + 4] \end{aligned} \quad (2.25)$$

これらの重み関数は図-2-3に示されるように、それぞれ、要素の各辺、12, 43, 23, 14 に対して割り当てられる。

(2.4) 式に対する厳密解は、(2.26) 式が適用される時に得られる。

$$\alpha = \coth \left[\frac{uL}{2D} \right] - \frac{2D}{uL} \quad (2.26)$$

ここに、 α : $\alpha_1, \alpha_2, \beta_1, \beta_2$ に対応

u : 流速 (L/T)

L : 代表長さ (L)

D : 分散係数 (L²/T)

2.4.4 数値積分

全体座標系における要素は様々な形状の四角形であるが、これらの四角形内での積分を容易に行うために、これらの四角形を局所座標系の (1, 1), (-1, 1), (-1, -1), (1, -1) で定義される正方形領域に写像を行う。全体座標系と局所座標系における座標は、(2.27 a), (2.27 b) 式に示される関係にある。

$$x = \sum_{j=1}^4 x_j N_j \quad (2.27 a)$$

$$z = \sum_{j=1}^4 z_j N_j \quad (2.27 b)$$

ここに, x_j, z_j : 節点の全体座標

N_j : 局所座標 ξ, η に依存する形状関数

「FEMWASTE」では, このように要素の形状を表す形状関数と要素内の未知数の変化を表す基本関数とを同一の関数で表現する要素, すなわち, アイソパラメトリック要素を採用している。

全体座標系から局所座標系へ座標変換を行う為のヤコビアン行列は(2.28)式で表される。

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} \quad (2.28)$$

[J] の行列式を J とすると,

$$J = \det [J] = \left(x_j \frac{\partial N_j}{\partial \xi} \right) \cdot \left(z_k \frac{\partial N_k}{\partial \eta} \right) - \left(z_j \frac{\partial N_j}{\partial \xi} \right) \cdot \left(x_k \frac{\partial N_k}{\partial \eta} \right) \quad (2.29)$$

よって, (2.14), (2.15) 式を局所座標系に座標変換を行うと, ヤコビアン行列の行列式を用いて, (2.30), (2.31) 式のようになる。

$$\begin{aligned} [S] = \int_{-1}^1 \int_{-1}^1 & \left[\frac{\partial [W]^T}{\partial x} \left(\theta D_{xx} \frac{\partial [N]}{\partial x} + \theta D_{xz} \frac{\partial [N]}{\partial z} \right) + \frac{\partial [W]^T}{\partial z} \left(\theta D_{zx} \frac{\partial [N]}{\partial x} + \theta D_{zz} \frac{\partial [N]}{\partial z} \right) \right. \\ & - \left. \left(\frac{\partial [W]^T}{\partial x} V_x [N] + \frac{\partial [W]^T}{\partial z} V_z [N] \right) \right. \\ & \left. + [W]^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta Rd \frac{\partial h}{\partial t} + \lambda \cdot \theta Rd \right) [N] \right] \det [J] d\xi d\eta \quad (2.30) \end{aligned}$$

$$[m] = \int_{-1}^1 \int_{-1}^1 \theta Rd [W]^T [N] \det [J] d\xi d\eta \quad (2.31)$$

この (2.30), (2.31) 式は数値積分を用いて容易に積分を行う事が出来る。「FEMWASTE」では, ガウスの求積法を用いて積分を行っている。

マトリックス [S] を評価する為には, 形状関数と重み関数の微分値が必要である。(2.27), (2.28) 式から全体座標系と局所座標系における微分形の関係は (2.32) 式で示される。

$$\begin{Bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial z} \end{Bmatrix} \quad (2.32)$$

(2.32) の左から逆行列を掛けると (2.33) 式が得られる。

$$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial z} \end{Bmatrix} = \frac{1}{\det [J]} \begin{bmatrix} \frac{\partial z}{\partial \eta} & -\frac{\partial z}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix} \quad (2.33)$$

よって (2.33) 式から、形状関数、重み関数の微分形は (2.34) ~ (2.37) 式で示される。これらの式を F (2.30) 式に代入する事により数値積分に適した形になる。

$$\frac{\partial [N]}{\partial x} = \frac{1}{\det [J]} \left[[z]^T \frac{\partial [N]}{\partial \eta} \cdot \frac{\partial [N]}{\partial \xi} - [z]^T \frac{\partial [N]}{\partial \xi} \cdot \frac{\partial [N]}{\partial \eta} \right] \quad (2.34)$$

$$\frac{\partial [W]}{\partial x} = \frac{1}{\det [J]} \left[[z]^T \frac{\partial [N]}{\partial \eta} \cdot \frac{\partial [W]}{\partial \xi} - [z]^T \frac{\partial [N]}{\partial \xi} \cdot \frac{\partial [W]}{\partial \eta} \right] \quad (2.35)$$

$$\frac{\partial [N]}{\partial z} = \frac{1}{\det [J]} \left[[x]^T \frac{\partial [N]}{\partial \eta} \cdot \frac{\partial [N]}{\partial \xi} - [x]^T \frac{\partial [N]}{\partial \xi} \cdot \frac{\partial [N]}{\partial \eta} \right] \quad (2.36)$$

$$\frac{\partial [W]}{\partial z} = \frac{1}{\det [J]} \left[[x]^T \frac{\partial [N]}{\partial \eta} \cdot \frac{\partial [W]}{\partial \xi} - [x]^T \frac{\partial [N]}{\partial \xi} \cdot \frac{\partial [W]}{\partial \eta} \right] \quad (2.37)$$

2.5 モジュールの構成

「FEMWASTE」はメインプログラムを含む合計16個のモジュールから構成されている。これらのモジュールは「FEMWASTE」のオリジナルバージョンに比べてかなりの改良と機能追加が行われて来ており、それらの機能をまとめたものが表-2-1であり、関係を示したものが図-2-4である。また、タイムループにおける主要モジュールとそれらのつながりを示したものが図-2-5である。計算の手順は以下ようになる。

- ① 「MAIN」で変数の配列宣言，初期値設定を行う。
- ② 「DATAIN」で入力データ，流れの場を読み込む。
- ③ 「AFABTA」で要素の各辺における重みファクターを計算する。
- ④ 「Q4」で要素マトリックスを作成する。
- ⑤ 「ASEMBL」で要素マトリックスを加え合わせ全体マトリックスを作成する。
- ⑥ 「BC」で全体マトリックスの境界に相当する部分に境界条件を組み込む。
- ⑦ 「SOLVE」で境界条件を導入した全体マトリックスから全節点の濃度を計算する。
- ⑧ 「FLUX」で全節点におけるフラックスを計算する。
- ⑨ 「SFLOW」で境界におけるフラックスの合計値を計算する。
- ⑩ 「PRINTT」で計算結果をプリントする。

⑩ 「STORE」で計算結果をディスクに書き込む。

⑪ ③～⑩を各タイムステップ毎に繰り返す。

なお、②～⑪の流れは、モジュール「GM2DXZ」内で行われる。

表-2-1 「FEMWASTE」を構成するモジュールの機能

モジュール名	機 能
MAIN	<ul style="list-style-type: none"> ・メインプログラム ・全ての配列の大きさの設定と初期化を行う。
GM2DXZ	<ul style="list-style-type: none"> ・定常計算及び非定常計算を行う。 ・MAINとGM2DXZ以外の全てのモジュールを呼び出す。
DATAIN	<ul style="list-style-type: none"> ・入力データの読み込み、チェック、エコーを行う。 ・解析対象の幾何学的情報、流れの場の情報を読み込む。 ・ソースタームを読み込む。
AFABTA	<ul style="list-style-type: none"> ・要素の各辺の重みファクターを計算する。
FLUX	<ul style="list-style-type: none"> ・要素の各点のx方向、z方向のフラックスを計算する。
Q4D	<ul style="list-style-type: none"> ・x方向、z方向のフラックスを計算する為に、要素内で積分を行い要素マトリックスを作成する。
ASEMBL	<ul style="list-style-type: none"> ・要素マトリックスを加え合わせて全体マトリックスを作成する。
Q4	<ul style="list-style-type: none"> ・要素の各点の濃度を計算する為に、要素内で積分を行い、要素マトリックスを作成する。
SHAPE	<ul style="list-style-type: none"> ・局所座標系における基本関数と重み関数の値を計算する。
BC	<ul style="list-style-type: none"> ・Dirichlet型境界条件、Neumann型境界条件、Cauchy型境界条件を全体マトリックスの中に組み込む。
Q4SP	<ul style="list-style-type: none"> ・境界上における浸出フラックスの積分を行う。
SOLVE	<ul style="list-style-type: none"> ・ガウスの消去法と後退代入法により、マトリックス方程式を解く。
SFLOW	<ul style="list-style-type: none"> ・各時間間隔における境界のフラックス、流量、全体の流れの合計値を計算する。
Q4R	<ul style="list-style-type: none"> ・要素内で濃度の積分を行う。
PRINTT	<ul style="list-style-type: none"> ・流量、濃度、フラックス等の計算結果をプリントする。
STORE	<ul style="list-style-type: none"> ・ポスト処理を行う為に計算結果をディスクに書き込む。

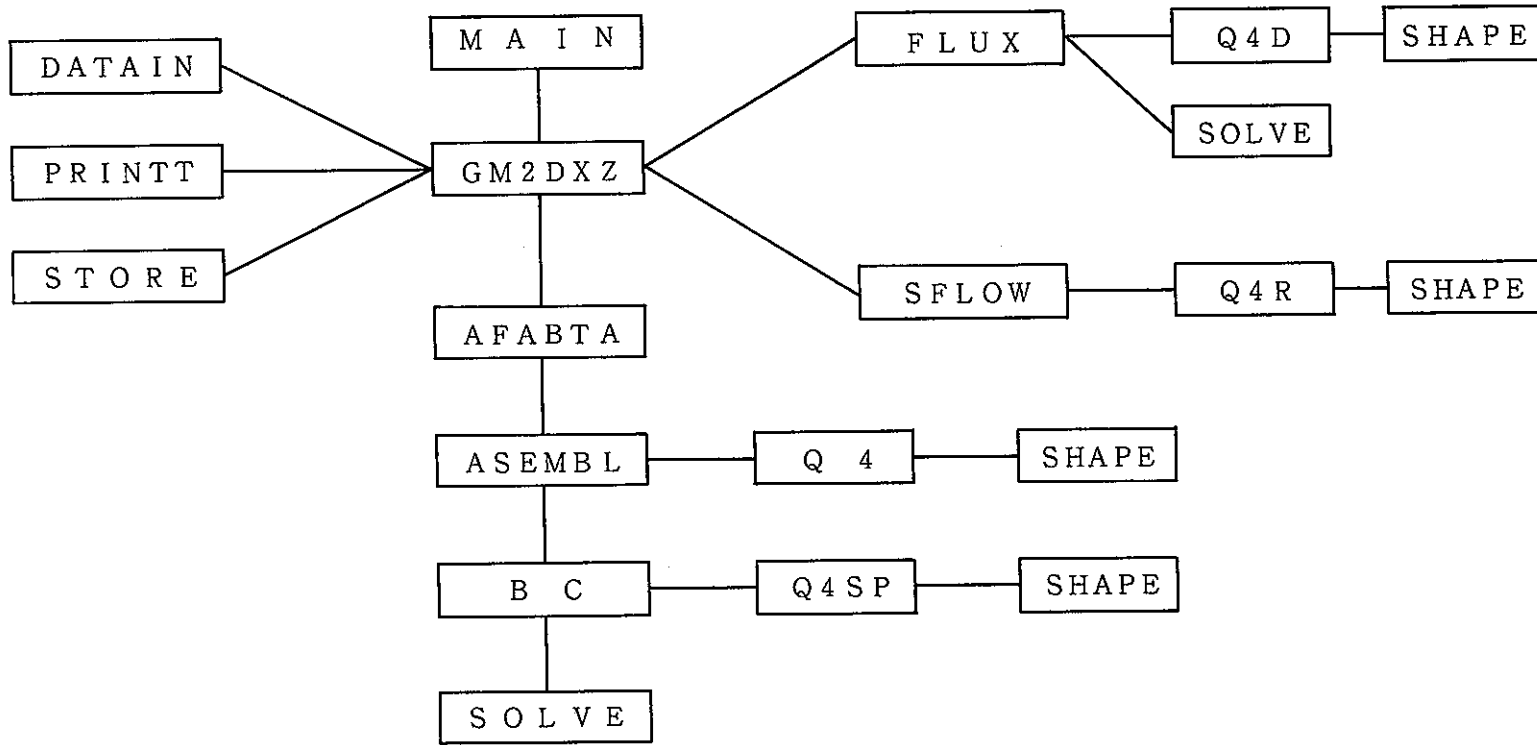


図-2-4 「FEMWASTE」のモジュールの構成

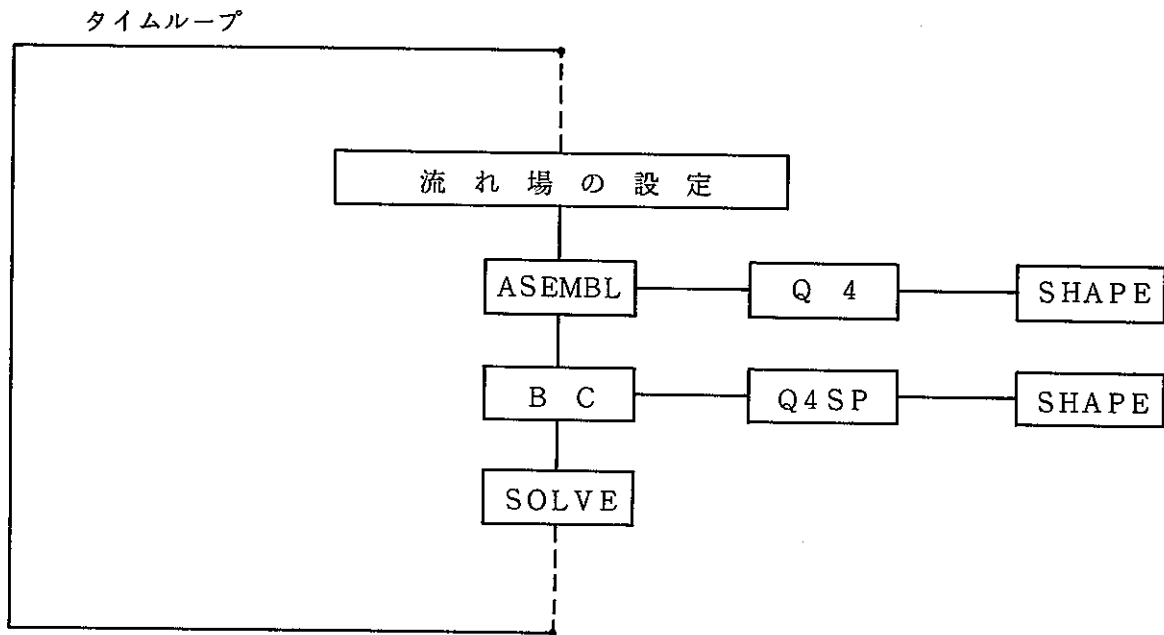


図-2-5 「FEMWASTE」のタイムループ内での
モジュールの構成

3. 「FEMWASTE-PNC」の概念

3.1 機能追加

「FEMWASTE」は2章で述べた様な機能を有するが、地層処分システムの性能評価計算に使用する場合、いくつかの不備な点が見られる。「FEMWASTE-PNC」では、これらの不備な点をピックアップし、以下の機能を追加して、「FEMWASTER」の改良を行った。

- ① 崩壊系列（3メンバーまで考慮）
- ② 湧き出し（ソースターム）
- ③ 境界条件の経時変化
- ④ 地表面に放出される核種量の計算
- ⑤ リスタート機能

これらの機能を追加する事にした理由は以下の通りである。

① 崩壊系列（3メンバーまで考慮）

「FEMWASTE」は崩壊に関しては、一回の崩壊による核種の消滅しか考慮していない。しかしながら、多くの放射性核種は崩壊系列のメンバーになっており、崩壊による核種の消滅とともに生成の効果も考慮する事はより現実的である。また、崩壊系列を構成する全ての核種の移行計算を同時に行う事により、核種特性による濃度分布、被ばく線量の比較を直接核種毎に行う事ができる。

② 湧き出し（ソースターム）

「FEMWASTE」は湧き出しは考慮していない。このために処分場からの核種の放出を考えると、処分場の占有する空間は計算領域とせず、処分場の境界を計算領域の一部として考え、この境界で濃度固定か、フラックス一定の境界条件でしか与える事はできない。よって、湧き出しを考慮する事により、処分場の空間も計算領域とみなす事ができ、より現実に近い状態で計算が行える。また、処分場から放出される核種の量の経時変化の取り扱いも容易になる。

③ 境界条件の経時変化

「FEMWASTE」で指定する境界条件はDirichlet型、Neumann型、Cauchy型があるが、いづれも、境界条件の経時変化を取り扱えない。よって、処分場からの核種の放出量を境界条件で与える場合、境界条件の経時変化を考慮しないと、核種の放出量の経時変化に対応できない。

④ 地表面に放出される核種量の計算

「FEMWASTE」では指定した境界から放出される時間あたりの核種の量の計算を行う事ができない。よって地表面に放出される核種の放出量を計算する事ができず、被ばく線量に換算する事もできない。

⑤ リスタート機能

長半減期核種の移行計算を行う場合、その核種の減衰の影響を見る為には、数百万年～数千万年までの計算を行わなければならない。これらの核種に対して、移行計算を「FEMWASTE」を用いて行くと、同じ時間増分で計算し続ける為に計算時間は非常に長くなる。よって、適切な時間まで計算を行い、その時点での濃度分布をファイルに書きだし、そのファイルに書き出された濃度分布を初期値として時間増分を大きくして新たに計算を行う事により、全体の計算時間は大幅に短縮できる。リスタート機能はこのように計算時間の短縮に役立つ。

3.2 数理モデル

「FEMWASTE-PNC」の数理モデルは、(2.1)式で示される「FEMWASTE」の数理モデルに崩壊系列、湧き出しの効果を導入して、(3.1)式のように示される。

$$\begin{aligned} & \frac{\partial}{\partial t}(\theta c_i + \rho s_i) + (\theta c_i + \rho s_i) \alpha' \frac{\partial h}{\partial t} = - \left(\frac{\partial V_x c_i}{\partial x} + \frac{\partial V_z c_i}{\partial z} \right) \\ & + \left[\frac{\partial}{\partial x} (\theta D_{xxi} \frac{\partial c_i}{\partial x} + \theta D_{xzi} \frac{\partial c_i}{\partial z}) + \frac{\partial}{\partial z} (\theta D_{zxi} \frac{\partial c_i}{\partial x} + \theta D_{zzi} \frac{\partial c_i}{\partial z}) \right] \\ & + \lambda_{i-1} (\theta c_{i-1} + \rho s_{i-1}) - \lambda_i (\theta c_i + \rho s_i) + Q_i(t) \end{aligned} \quad (3.1)$$

湧き出し項は時間の関数となっている。

- ここに、
- θ : 体積含水率
 - c_i : 液相中に溶解している核種 i の濃度 (M/L^3)
 - ρ : 固相の見かけ密度 (M/L^3)
 - s_i : 固相に吸着した核種 i の濃度 (M/M)
 - α' : 媒体の修正圧縮率 ($1/L$)
 - h : 地下水の圧力水頭 (L)
 - $D_{xxi}, D_{xzi}, D_{zxi}, D_{zzi}$: 核種 i の分散係数 (L^2/T)
 - V_x, V_z : x 方向, z 方向のダルシー流速 (L/T)
 - λ_i : 核種 i の崩壊定数
 - x, z : 空間座標 (L)
 - t : 時間座標 (T)
 - $Q_i(t)$: 核種 i の湧き出し ($M/L^3 T$)

分散係数については次式のようなになる。

$$\theta D_{xxi} = a_i V_x + \frac{(a_i - a_T) V_x^2}{V} + D_{mi} \cdot T \quad (3.2 a)$$

$$\theta D_{xxi} = \theta D_{zxi} = \frac{(a_L - a_T) V_x V_z}{V} \quad (3.2 b)$$

$$\theta D_{zzi} = a_T V + \frac{(a_L - a_T) V_z^2}{V} + D_{mi} \cdot T \quad (3.2 c)$$

$$V = \sqrt{V_x^2 + V_z^2}$$

ここに、 a_T : 流れと垂直方向の分散率 (L)

a_L : 流れと平衡方向の分散率 (L)

T : 媒体の屈曲率 (—)

D_{mi} : 地下水中の核種 i の分子拡散係数 (L^2/T)

ここで、液相に溶解している核種 i と固相に吸着している核種 i の平衡状態が瞬時に生じると仮定し、これを線型の方程式で表現すると、(3.3) 式で示すような線型吸着式 (ヘンリー型) になる。

$$s_i = K_{di} c_i \quad (3.3)$$

ここに、 K_{di} : 核種 i の分配係数 (L^3/M)

(3.3) 式を (3.1) 式に代入すると核種 i の移行方程式 (3.4) が得られる。

$$\begin{aligned} \theta R_{di} \frac{\partial c_i}{\partial t} = & - \left(\frac{\partial V_x c_i}{\partial x} + \frac{\partial V_z c_i}{\partial z} \right) \\ & + \left[\frac{\partial}{\partial x} \left(\theta D_{xxi} \frac{\partial c_i}{\partial x} + \theta D_{xz} \frac{\partial c_i}{\partial z} \right) + \frac{\partial}{\partial z} \left(\theta D_{zx} \frac{\partial c_i}{\partial x} + \theta D_{zzi} \frac{\partial c_i}{\partial z} \right) \right] \\ & - \left(\frac{\partial \theta}{\partial t} + \alpha' \theta \cdot R_{di} \frac{\partial h}{\partial t} \right) \\ & + \lambda_{i-1} \theta R_{d_{i-1}} c_{i-1} - \lambda_i \theta R_{di} c_i + Q_i(t) \end{aligned} \quad (3.4)$$

$$R_{di} = 1 + \frac{\rho K_{di}}{\theta} \quad (3.5)$$

R_{di} は核種 i の遅延係数である。

3.3 初期条件

核種移行の計算は通常、非定常で行われているので、計算に際して初期条件を設定しなければならない。初期条件は (2.6) 式同様 (3.6) 式で示される。

$$c_i = c_{oi}(x, z, t=0) \quad \text{in } R \quad (3.6)$$

c_{oi} は R における核種 i の濃度の初期値である。

「FEMWASTE-PNC」はリスタート機能を有しており、核種移行計算の終了後、計算終了時点の濃度分布を初期条件として時間増分を大きくし、再びその核種移行計算を継続して行う事ができる。この時の初期条件は(3.7)式のようになる。

$$c_i = c_{Ti}(x, z, t = T) \quad \text{in } R \quad (3.7)$$

c_{Ti} はRにおける時間Tでの核種iの濃度分布である。

このリスタート機能は超長半減期を有する核種の移行計算を、その核種が減衰するまで行う時に有利である。

3.4 境界条件

3.4.1 Dirichlet 型境界条件

本境界条件は対象となる境界において核種iの濃度を固定するものである。式で表すと(3.8)式のようになる。

$$c_i = c_{i1}(x, z) \quad \text{on } B_1 \quad (3.8)$$

B_1 は全境界Bの一部である。また、 c_{i1} は核種iの濃度の固定値であり、場所の関数である。「FEMWASTE-PNC」では境界節点番号とその節点に対応する濃度固定値を指定するようになっている。

3.4.2 Neumann 型境界条件

本境界条件は、対象となる境界の法線方向から、計算領域へ分散により、流入あるいは流出する核種iのフラックスを規定するものである。式で表すと(3.9)式のようになる。

$$\begin{aligned} & - \left(\theta D_{xxi} \frac{\partial c_i}{\partial x} + \theta D_{zz} \frac{\partial c_i}{\partial z} - V_x c_i \right) n_x - \left(\theta D_{zxi} \frac{\partial c_i}{\partial x} + \theta D_{zz1} \frac{\partial c_i}{\partial z} - V_z c_i \right) n_z \\ & = q_{2i}(x, z) + (V_x n_x + V_z n_z) c_i \quad \text{on } B_2 \quad (3.9) \end{aligned}$$

B_2 は全境界Bの一部であり、 n_x 、 n_z は B_2 上の法線ベクトルの方向余弦である。 q_{2i} は分散により、流入あるいは流出する核種iのフラックスの値であり、場所の関数である。「FEMWASTE-PNC」では核種は常に境界において、法線方向の流速により運ばれ、濃度勾配によっては核種は運ばれないと考えられており、このように仮定できる境界要素の番号と節点番号を指定するようになっている。

3.4.3 Cauchy 型境界条件

本境界条件は対象となる境界の法線方向から、計算領域へ、移流、分散により、流入あるいは流出する核種iのフラックスを規定するものである。式で表すと(3.10)式のようになる。

$$\begin{aligned} & - \left(\theta D_{xxi} \frac{\partial c_i}{\partial x} + \theta D_{zxi} \frac{\partial c_i}{\partial z} - V_x c_i \right) n_x - \left(\theta D_{zx} \frac{\partial c_i}{\partial x} + \theta D_{zz} \frac{\partial c_i}{\partial z} - V_z c_i \right) n_z \\ & = q_{3i}(x, z, t) \quad \text{on } B_3 \quad (3.10) \end{aligned}$$

B_3 は全境界 B の一部であり、 n_x 、 n_z は B_3 上の法線ベクトルの方向余弦である。 q_{3i} は移流、分散により流入、流出する核種 i のフラックスの値であり、場所と時間の関数である。

「FEMWASTE-PNC」では、境界要素を構成する節点番号とその節点に対応する核種 i のフラックスを指定する。また、フラックスの経時変化を取り扱う事もでき、これはあらかじめ作成しておいたファイルから逐次読み込まれる。

3.5 有限要素近似

「FEMWASTE-PNC」の有限要素近似は空間依存の場を除いて時間依存の場、基本関数、形状関数、数値積分等については「FEMWASTE」と同一である。空間依存の場の有限要素近似が異なる理由は、崩壊系列と湧き出しの効果を考慮に入れたことによる。よって本節では空間依存の場における有限要素近似について述べる。その他の近似手法については、2.4節を参照されたい。

いま、(3.4) 式の右辺の項を全て左辺に移項すると核種 i の移行方程式は (3.11) 式のようになる。

$$\begin{aligned} L(c_i) = & \theta R d_i \frac{\partial c_i}{\partial t} + \left(\frac{\partial V_x c_i}{\partial x} + \frac{\partial V_z c_i}{\partial z} \right) \\ & - \left[\frac{\partial}{\partial x} \left(\theta D_{xxi} \frac{\partial c_i}{\partial x} + \theta D_{xz} \frac{\partial c_i}{\partial z} \right) + \frac{\partial}{\partial z} \left(\theta D_{zxi} \frac{\partial c_i}{\partial x} + \theta D_{zzi} \frac{\partial c_i}{\partial z} \right) \right] \\ & + \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d_i \frac{\partial h}{\partial t} \right) \\ & - \lambda_{i-1} \theta R d_{i-1} c_{i-1} + \lambda_i \theta R d_i c_i - Q_i(t) = 0 \end{aligned} \quad (3.11)$$

「FEMWASTE-PNC」では、有限要素の離散化に際して、線型の基本関数を仮定した四角形要素を採用している。よって1つの四角形要素における厳密解 c_i は (3.12) 式のように、近似解 \hat{c} で近似される事になる。

$$c_i \cong \hat{c}_i = \sum_{j=1}^4 c_{ij}(t) \cdot N_j = [N] \{c_i(t)\} \quad (3.12)$$

ここに、 $[N]$ は、核種 i の節点濃度と要素内部濃度を結びつける基本関数マトリックスであり、 $\{c_i(t)\}$ は時刻 t における核種 i に対する要素の節点ベクトルである。(3.11) 式の c_i を \hat{c} と置き換え、(3.12) 式を代入した式に $[W]$ を重み関数マトリックスとして、重みつき残差法を適用すると (3.13) 式を得る。

$$\int_{Re} [W]^T [L(\hat{c}_i)] dR = 0 \quad (3.13)$$

(3.13) 式を整理し、まとめると要素の有限要素式 (3.14) 式になる。

$$[m_i]\{\dot{c}_i\} + [s_i]\{c_i\} + \{d_i\} + \{q_i\} = 0 \quad (3.14)$$

ここに、
 $[m_i]$: 要素質量マトリックス
 $[s_i]$: 要素剛性マトリックス
 $\{d_i\}$: 要素内のソースタームベクトル
 $\{q_i\}$: 要素境界の流束ベクトル
 $\{c_i\}$: 要素節点濃度ベクトル

$[m_i]$, $[s_i]$, $\{d_i\}$, $\{q_i\}$, $\{c_i\}$ はそれぞれ (3.15) ~ (3.18) で示される。

$$[m_i] = \int_{Re} \theta R d_i (W)^T [N] dR \quad (3.15)$$

$$\begin{aligned} [s_i] = \int_{Re} & \left[\frac{\partial (W)^T}{\partial x} \left(\theta D_{xxi} \frac{\partial [N]}{\partial x} + \theta D_{xzi} \frac{\partial [N]}{\partial z} \right) + \frac{\partial (W)^T}{\partial z} \left(\theta D_{zxi} \frac{\partial [N]}{\partial x} + \theta D_{zzi} \frac{\partial [N]}{\partial z} \right) \right. \\ & - \left(\frac{\partial (W)^T}{\partial x} V_x [N] + \frac{\partial (W)^T}{\partial z} V_z [N] \right) \\ & \left. + (W)^T \left(\frac{\partial \theta}{\partial x} + \alpha' \theta R d_i \frac{\partial h}{\partial t} + \lambda_i \cdot \theta R d_i \right) [N] dR \right] \end{aligned} \quad (3.16)$$

$$\{d_i\} = \int_{Re} - (W)^T [M] dR \quad (3.17)$$

$$\begin{aligned} \{q_i\} = \int_{Re} & (W)^T \left[\left(-\theta D_{xxi} \frac{\partial c_i}{\partial x} - \theta D_{xzi} \frac{\partial c_i}{\partial z} + V_x c_i \right) n_x \right. \\ & \left. + \left(-\theta D_{zxi} \frac{\partial c_i}{\partial x} - \theta D_{zzi} \frac{\partial c_i}{\partial z} + V_z c_i \right) n_z \right] dB \end{aligned}$$

$$\{\dot{c}_i\} = \frac{\partial \{c_i(t)\}}{\partial t} \quad (3.18)$$

(3.17) 式の $[M]$ は、親核種からの崩壊した核種がソースタームとして娘核種に与えられる事 (及びその要素における核種の湧き出し) を示している。

(3.14) 式を全体の空間領域 R について加え合わせると核種 i に対する全体の有限要素式 (3.19) が得られる。

$$[M_i]\{\dot{C}_i\} + [S_i]\{C_i\} + \{D_i\} + \{Q_i\} = 0 \quad (3.19)$$

ここに、
 $[M_i]$: 全体の質量マトリックス
 $[S_i]$: 全体の剛性マトリックス

- [D_i] : 全体のソースタームベクトル
- [Q_i] : 全体境界の流束ベクトル
- [C_i] : 全体の節点温度ベクトル

3.6 モジュールの構成

「FEMWASTE-PNC」は、「FEMWASTE」を構成する16個のモジュールに新たに追加した6個のモジュールの合計22個のモジュールから構成されている。「FEMWASTE」を構成する16個のモジュールに関しては、機能拡張に伴う修正及び改良を行った。しかし、それらの基本的な機能及びつながりについては、もとのままである。これらのモジュールと新しく追加を行ったモジュールのつながりを図-3-1に示す。表-3-1は新モジュールの機能をまとめたものである。また、タイムループ内における主要モジュールとそれらのつながりを示したものが図-3-2である。

計算手順は2.5節の①～⑩に示す通りであるが、崩壊系列の計算を行う時はタイムループ内でさらに、④～⑩を繰り返す。

表-3-1 「FEMWASTE-PNC」用の新モジュールの機能

モジュール名	機能
AROSR	・湧き出しを考慮する要素（処分場の位置）として指定した領域の面積を計算する。
SRFLUX	・地表面において流入あるいは流出する核種の量の合計値を計算する。
QINT	・処分場からの経時的に変化する核種の放出量を内挿する。
BFLUX	・指定した境界に、経時的に変化するフラックスの内挿値を与える。
TOTC	・領域全体に存在する核種の総量の計算を行う。
MASS	・要素内の核種の量を計算する。

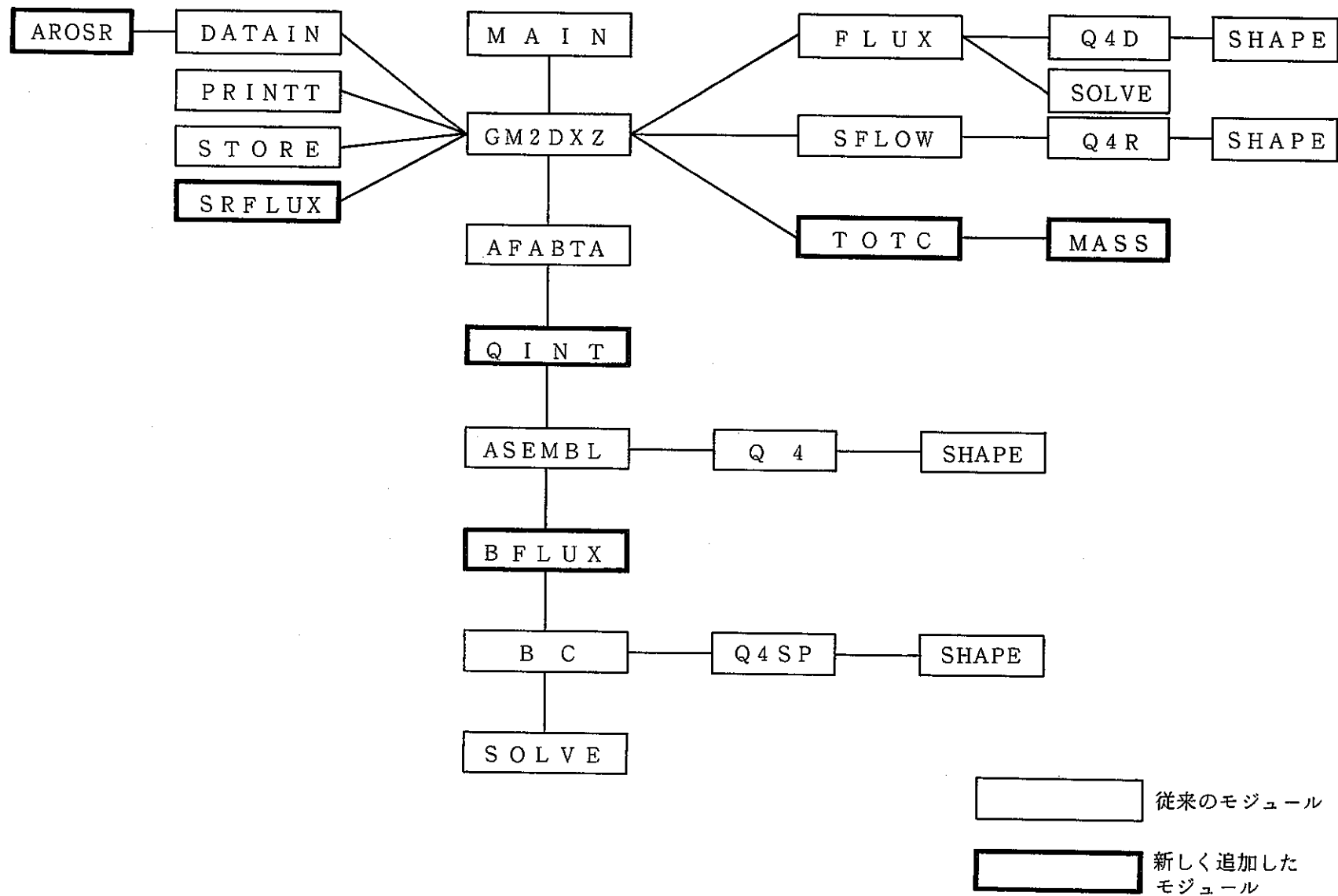


図-3-1 「FEMWASTE-PNC」のモジュールの構成

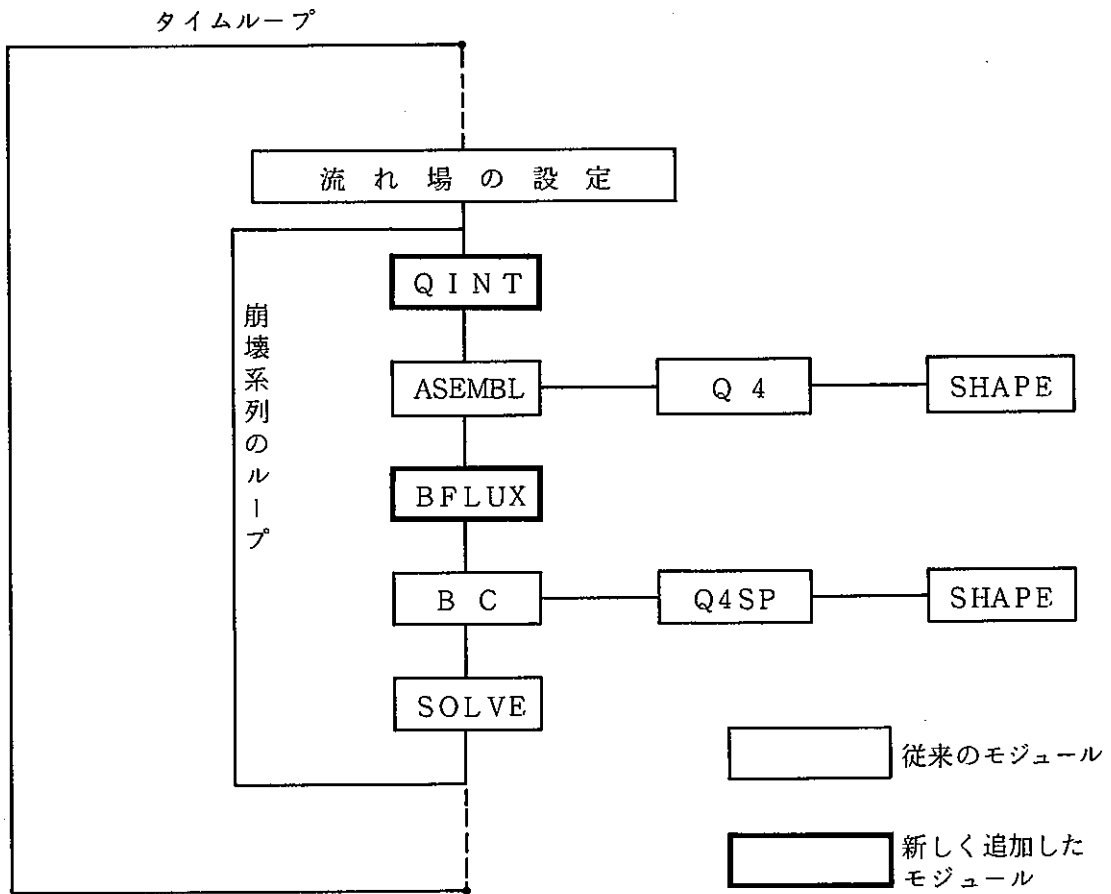


図-3-2 「FEMWASTE-PNC」のタイプループ内でのモジュールの構成

4. モジュールの詳細

4.1 Program MAIN

本モジュールは「FEMWASTE-PNC」のメインプログラムであり、プログラム全体で使用している変数の配列の大きさと初期値設定を行っている。

本モジュールの処理の流れを図-4-1に示す。

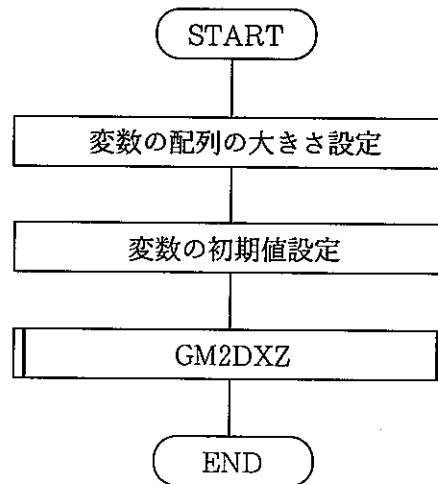
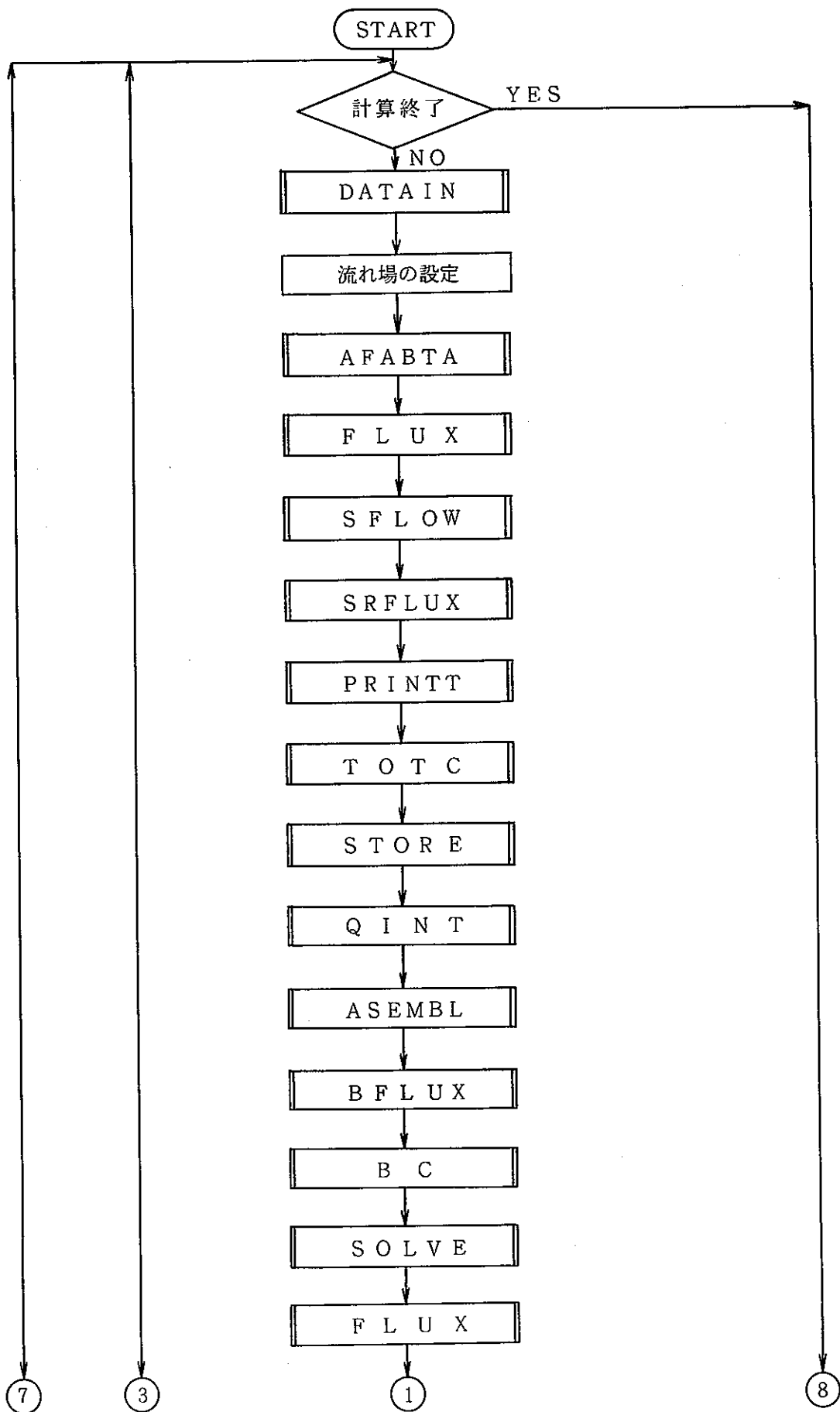


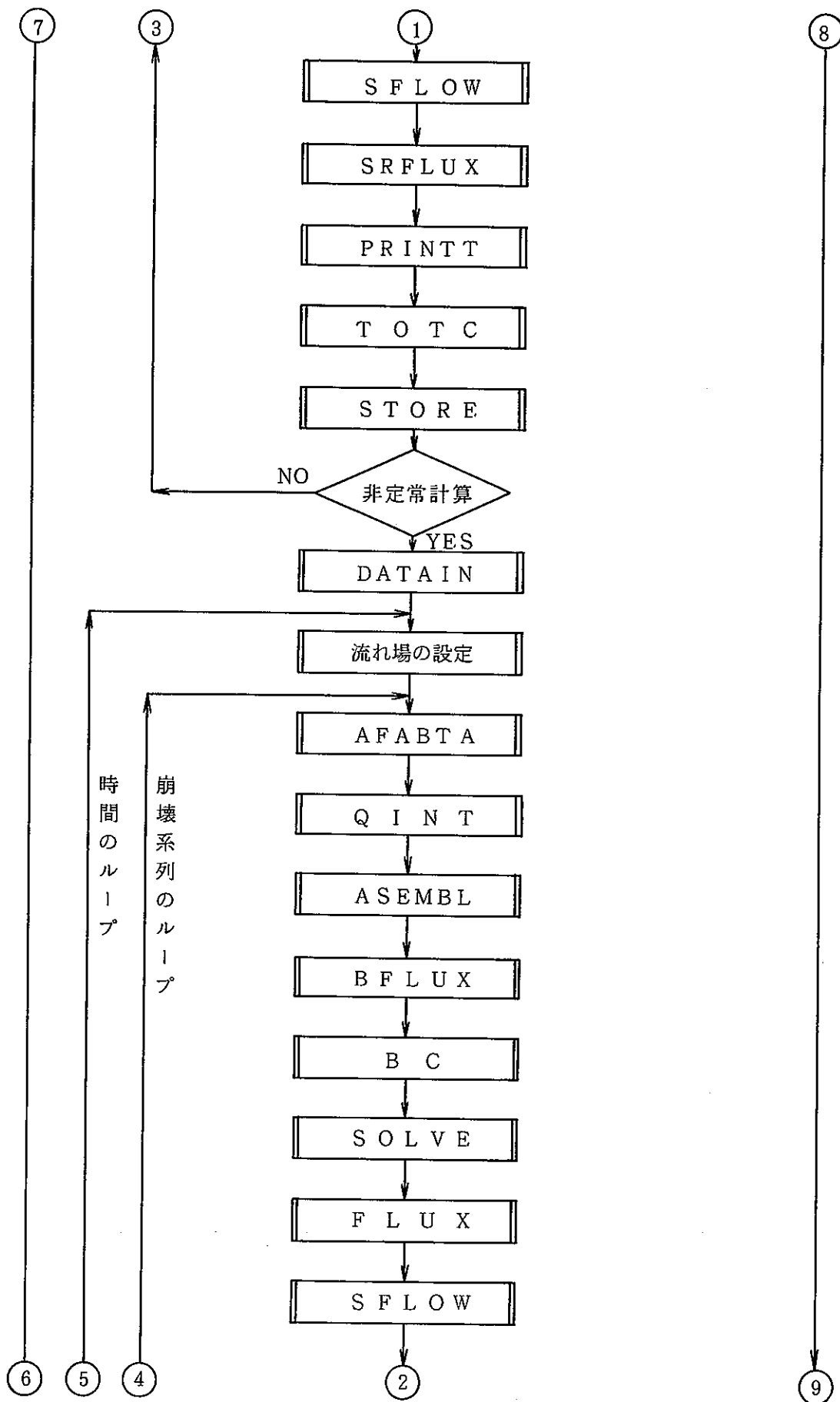
図-4-1 Program Main の処理の流れ

4.2 Subroutine GM2DXZ

本モジュールは「FEMWASTE-PNC」の実質的なメインプログラムであり、定常計算、非定常計算を行う。殆どのモジュールが本モジュールから呼び出される。

本モジュールの処理の流れを図-4-2に示す。





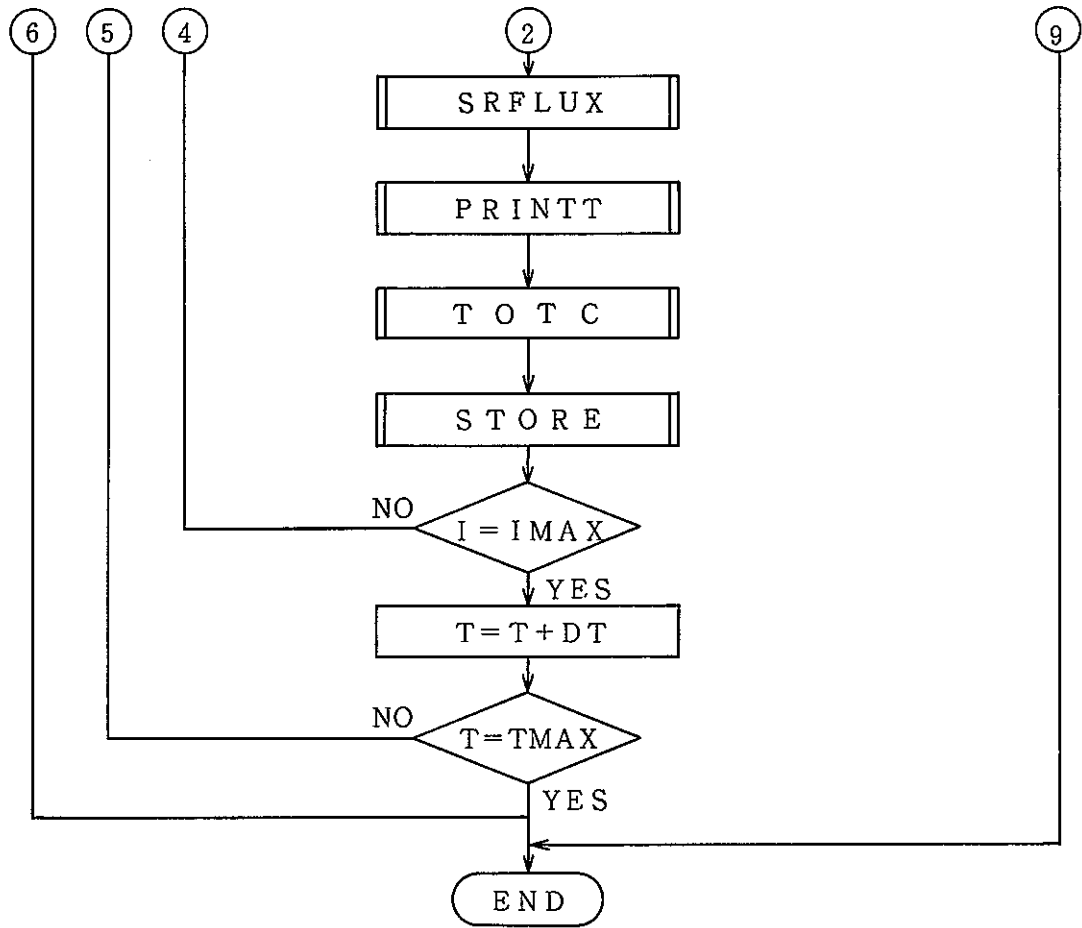


図-4-2 Subroutine GM2DXZの処理の流れ

4.3 Subroutine DATAIN

本モジュールは「FEMWASTE-PNC」用の入力データ、解析対象の幾何学的情報、及びソースタームの経時変化を読み込み、データチェック及びエコーを行う。また、節点番号、要素番号が自動作成機能を用いて指定されている時の番号の作成及びこれらに付随する数値の補間を行う。
本モジュールの処理の流れを図-4-3に示す。

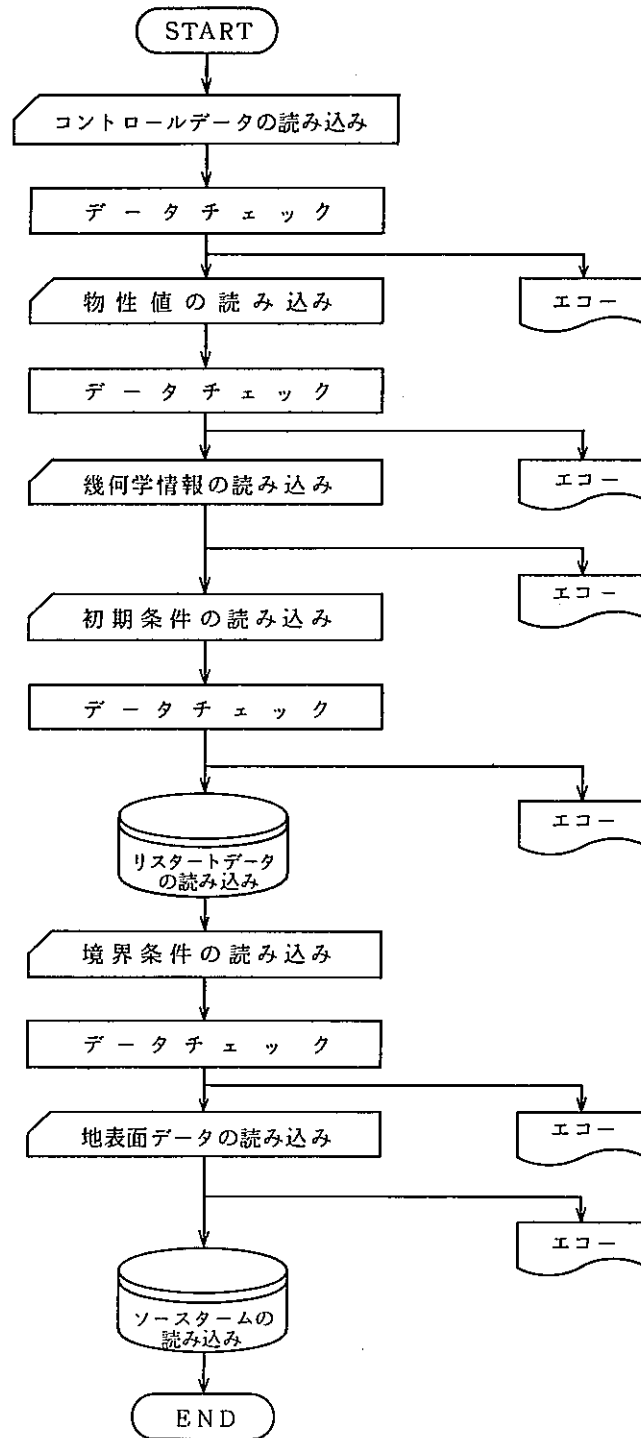


図-4-3 Subroutine DATAINの処理の流れ

4.4 Subroutine AFABTA

本モジュールは要素の各々の辺における重み係数の計算を行う。この重み係数は、上流法で計算を行う時の、重み関数の計算に用いる。

本モジュールの処理の流れを図-4-4に示す。

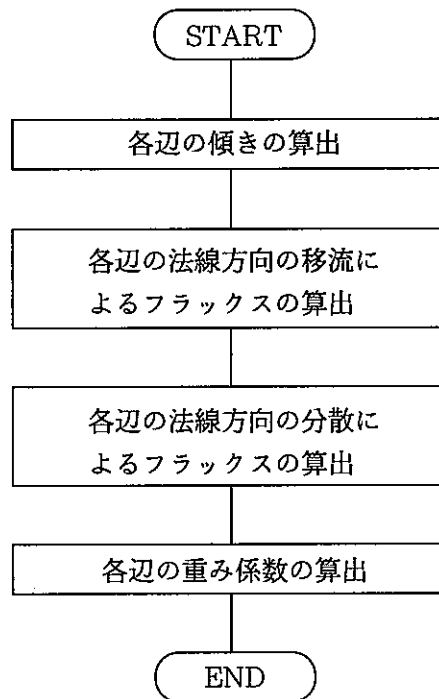


図-4-4 Subroutine AFABTA の処理の流れ

4.5 Subroutine FLUX

本モジュールは要素の各節点における、移流と分散により運ばれる x 方向, z 方向の核種のフラックスを計算する。これは行列の式を解くことにより解が求められる。

本モジュールの処理の流れを図-4-5 に示す。

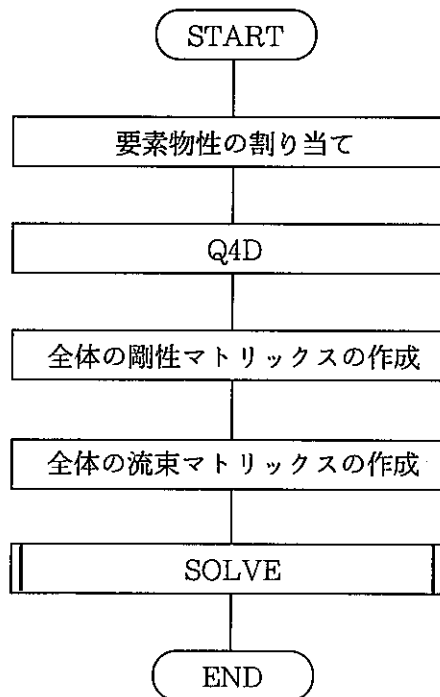


図-4-5 Subroutine FLUX の処理の流れ

4.6 Subroutine Q4D

本モジュールは、要素の各節点における、x方向、z方向の核種のフラックスを計算する為に、要素内で積分を行い、要素剛性マトリックス及び要素の流束ベクトルを作成する。積分はガラーキン法に基づいて行われる。

本モジュールの処理の流れを図-4-6に示す。

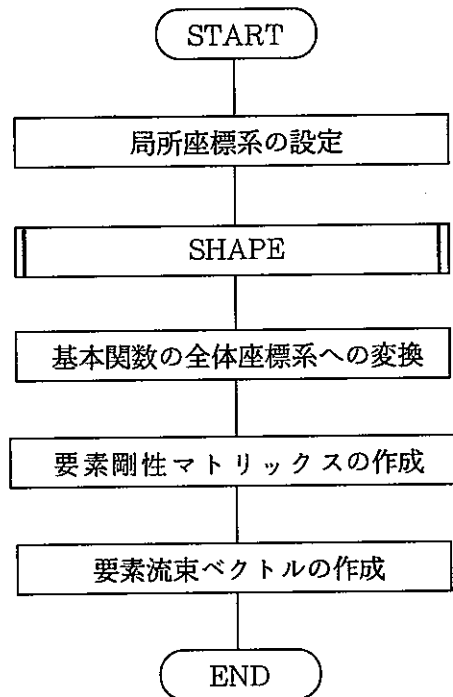


図-4-6 Subroutine Q4D の処理の流れ

4.7 Subroutine ASEMBL

本モジュールは核種の濃度を計算する為に要素剛性マトリックス、要素流束ベクトル及び要素内のソースタームベクトルを加え合わせ、全体の剛性マトリックス、全体の流束ベクトル、全体のソースタームベクトルを作成する。

本モジュールの処理の流れを図-4-7に示す。

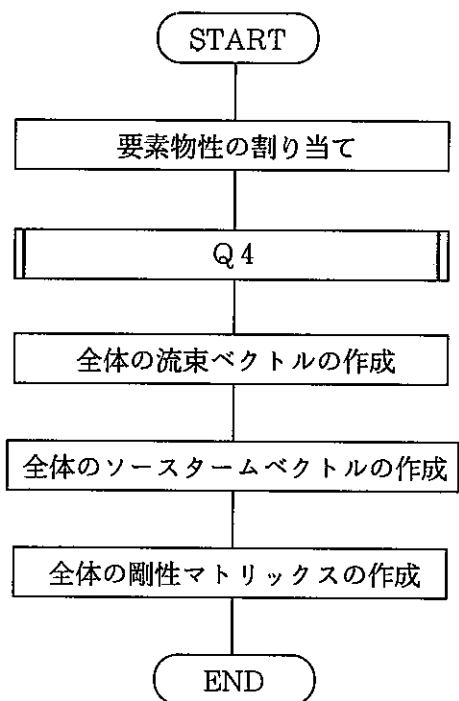


図-4-7 Subroutine ASEMBL の処理の流れ

4.8 Subroutine Q4

本モジュールは、要素の各節点における核種濃度を計算する為に要素内で積分を行い、要素剛性マトリックス、要素の流束ベクトル、要素のソースタームベクトルを作成する。

本モジュールの処理の流れを図-4-8に示す。

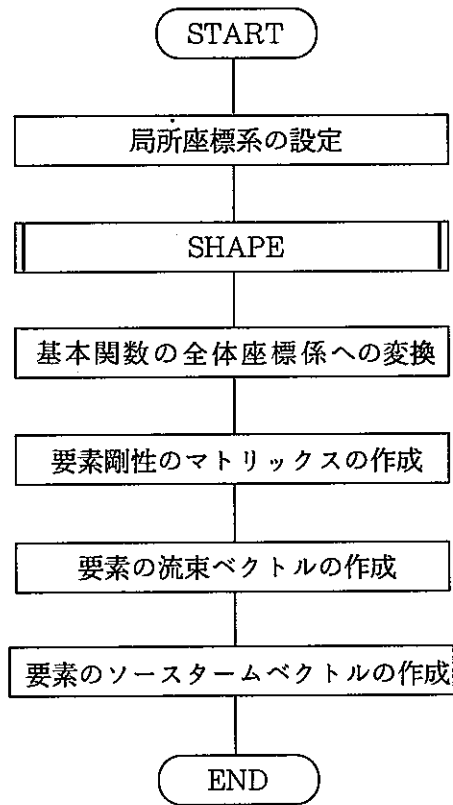


図-4-8 Subroutine Q4 の処理の流れ

4.9 Subroutine SHAPE

本モジュールは、局所座標系に変換された要素内の各節点における基本関数あるいは重み関数の計算を行う。

本モジュールの処理の流れを図-4-9に示す。

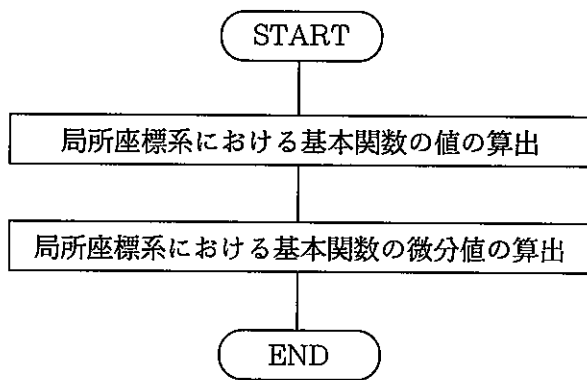


図-4-9 Subroutine SHAPE の処理の流れ

4.10 Subroutine BC

本モジュールは、有限要素法により解を求める為に、Dirichlet 型、Neumann 型、Cauchy 型の境界条件を入力データによる指定に従って全体の剛性マトリックス、及び全体の流束ベクトルの中に組み込む。

本モジュールの処理の流れを図-4-10に示す。

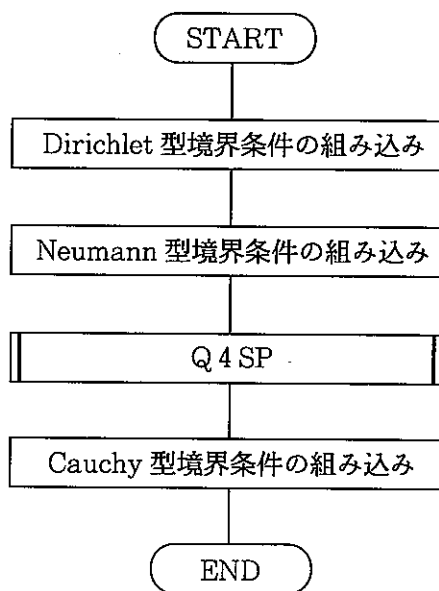


図-4-10 Subroutine BC の処理の流れ

4.11 Subroutine Q4SP

本モジュールは、ユーザーが指定した Neumann 型境界条件の範囲に属する要素内での移流、分散によるフラックスを評価する為に積分を行い、要素剛性マトリックスと要素内の流束ベクトルを作成する。

本モジュールの処理の流れを図-4-11 に示す。

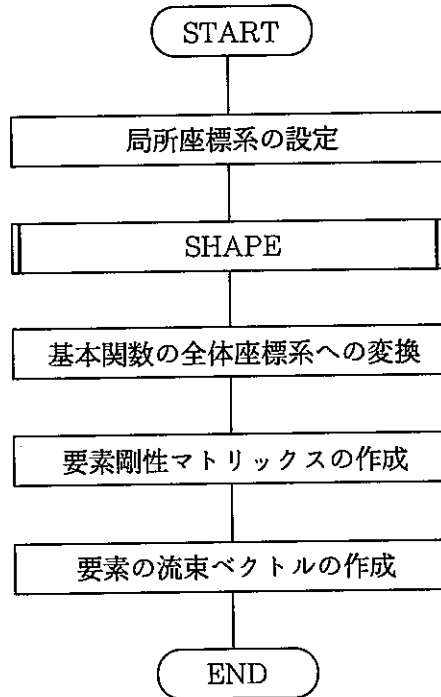


図-4-11 Subroutine Q4SP の処理の流れ

4.12 Subroutine SOLVE

本モジュールは、マトリックスの方程式を解く為に、全体の剛性マトリックスの三角分解と後退代入を行い、各節点における核種濃度を計算する。

本モジュールの処理の流れを図-4-12に示す。

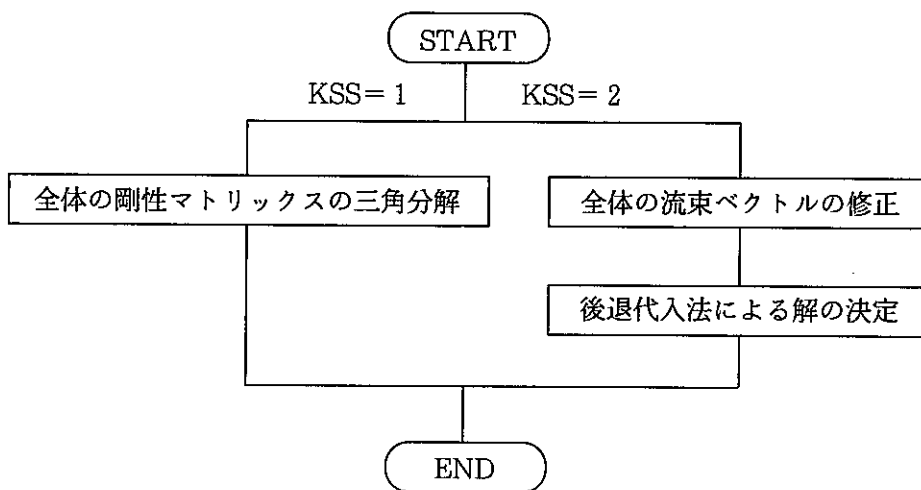


図-4-12 Subroutine SOLVE の処理の流れ

4.13 Subroutine SFLOW

本モジュールは、時間増分当りに生ずる、境界から放出される全質量、その割合、時間0からの全放出量、及び領域内での核種の増加量、減少量を計算する。

本モジュールの処理の流れを図-4-13に示す。

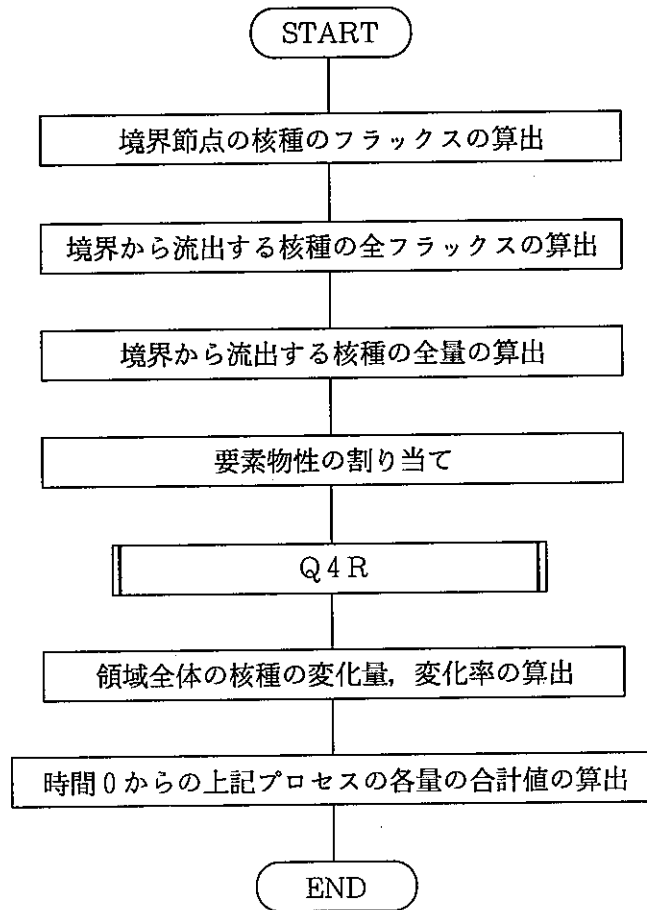


図-4-13 Subroutine SFLOW の処理の流れ

4.14 Subroutine Q4R

本モジュールは、全体領域での核種の変化量を計算する為に要素内で積分を行い、要素内での核種の変化を求める。

本モジュールの処理の流れを図-4-14に示す。

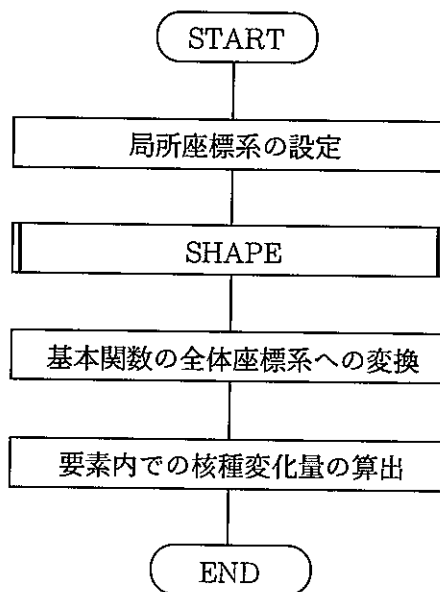


図-4-14 Subroutine Q4R の処理の流れ

4.15 Subroutine PRINTT

本モジュールは計算結果（核種濃度、核種フラックス）及び境界から流出していく、あるいは領域内での核種の量の変化量及び変化率を標準出力する。

本モジュールの処理の流れを図-4-15に示す。

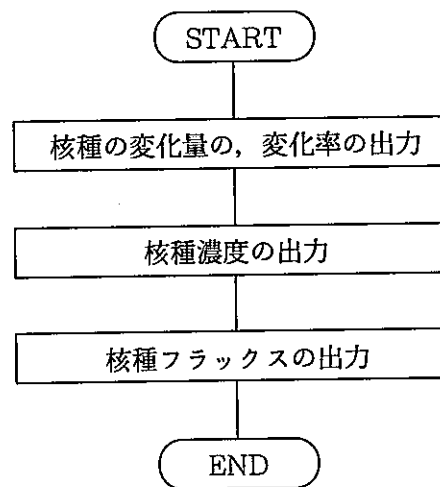


図-4-15 Subroutine PRINTT の処理の流れ

4.16 Subroutine STORE

本モジュールは、計算終了後、ポスト処理を行う為に、解析対象の幾何的情報及び指定した各時間ステップの核種濃度と核種フラックスの分布を書式なしでファイルに書き込む。

本モジュールの処理の流れを図-4-16に示す。

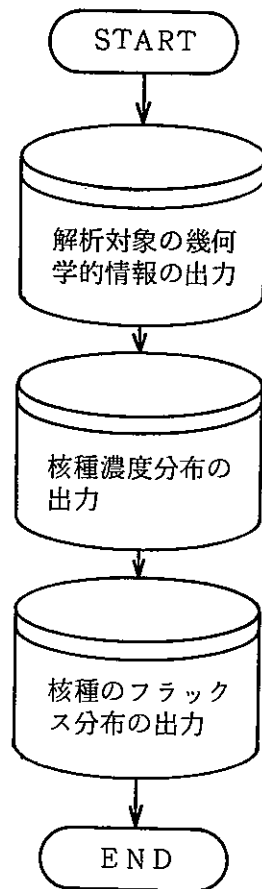


図-4-16 Subroutine STORE の処理の流れ

4.17 Subroutine QINT

本モジュールは、ニアフィールドからの核種の放出量として読み込んだ単位時間当りの核種量の経時変化を補間して、計算中の時間ステップの単位時間当りの核種量を求める。さらに、これをソースタームを考慮する領域における単位時間、単位体積当りの核種量に換算する。

本モジュールの処理の流れを図-4-17に示す。

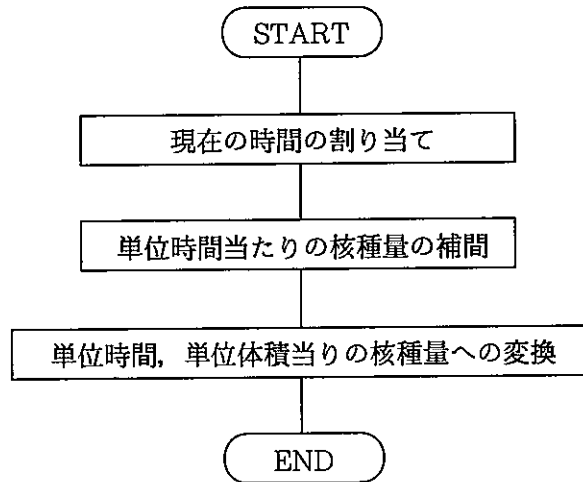


図-4-17 Subroutine QINT の処理の流れ

4.18 Subroutine BFLUX

本モジュールはニアフィールドからの核種放出量として読み込んだ単位時間当りの核種量の経時変化を補間して、計算中の時間ステップの単位時間当りの核種量を求める。さらに、これを核種のフラックスとして Cauchy 型境界条件で与える領域における単位時間、単位面積当りの核種量に換算する。

本モジュールの処理の流れを図-4-18に示す。

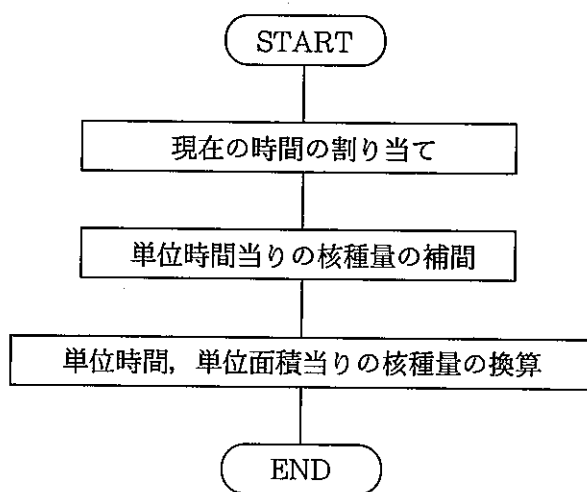


図-4-18 Subroutine BFLUX の処理の流れ

4.19 Subroutine AROSR

本モジュールは、ソースタームとしての処分場を考慮する領域の面積を、これを構成する要素の面積の合計として算出する。

本モジュールの処理の流れを図-4-19に示す。

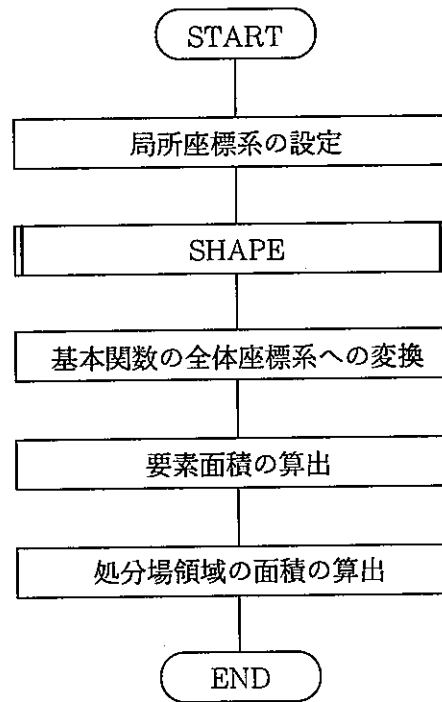


図-4-19 Subroutine AROSR の処理の流れ

4.20 Subroutine SRFLUX

本モジュールは、境界領域の地表面に相当する部分から放出される単位時間当りの核種量を算出する。

本モジュールの処理の流れを図-4-20に示す。

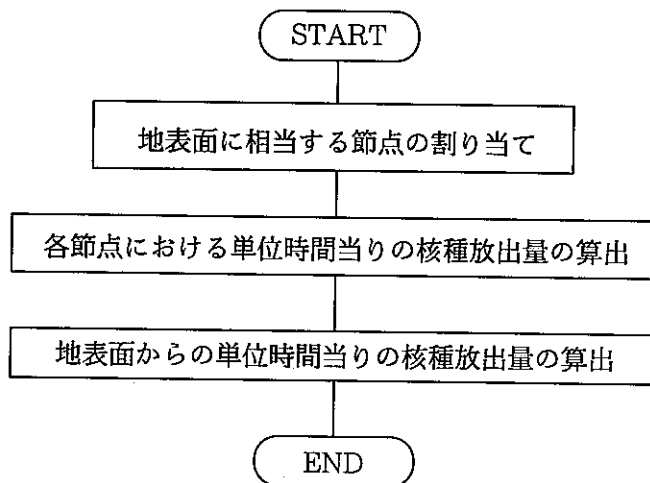


図-4-20 Subroutine SRFLUX の処理の流れ

4.21 Subroutine MASS

本モジュールは計算領域を構成する各要素内に存在する核種の量を算出する。
本モジュールの処理の流れを図-4-21に示す。

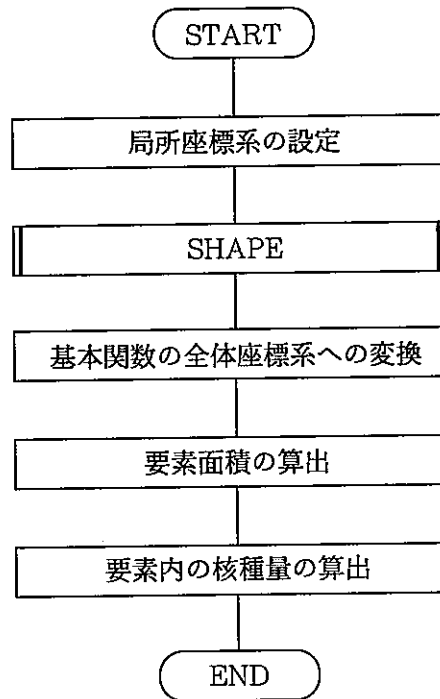


図-4-21 Subroutine MASS の処理の流れ

4.22 Subroutine TOTC

本モジュールは、計算領域を構成する各要素毎に求められた核種量を合計し、全領域に存在する核種量を算出する。

本モジュールの処理の流れを図-4-22に示す。

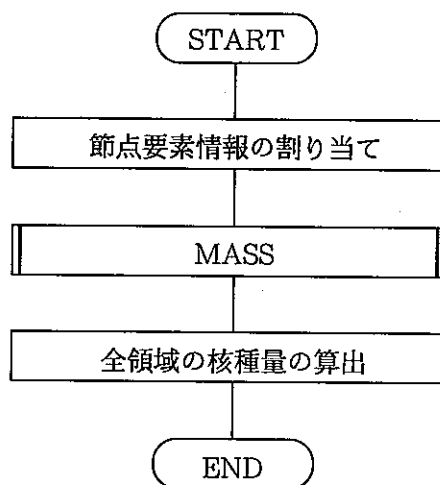


図-4-22 Subroutine TOTC の処理の流れ

5. ファイルの構成

5.1 「FEMWASTE-PNC」の関連ファイル

「FEMWASTE-PNC」は合計7個の外部ファイルと結びつけられており、これらのファイルを用いて計算に必要なデータの読み込み及び計算結果の書き出しを行っている。これらのファイルは以下に示されるような入力用ファイルと出力用ファイルに分けられる。

1) 入力用ファイル

- ① 「FEMWASTE-PNC」用の入力データファイル
- ② 計算領域の幾何学的データと流れの場のデータが記録されているファイル（「FEMWATER」のポスト処理用出力ファイル）
- ③ 「FEMWASTE-PNC」のリスタート用入力ファイル

2) 出力用ファイル

- ① 「FEMWASTE-PNC」による計算結果の標準出力用ファイル
- ② 「FEMWASTE-PNC」による計算結果（濃度、フラックス）のポスト処理用出力ファイル
- ③ 「FEMWASTE-PNC」による計算結果（放出量）のポスト処理用出力ファイル
- ④ 「FEMWASTE-PNC」のリスタート用出力ファイル

図-5-1はこれらのファイルと「FEMWASTE-PNC」用の関係を示している。図中のデバイス番号は「FEMWASTE-PNC」で使用されている、READ文及びWRITE文中の論理装置指定番号に対応している。

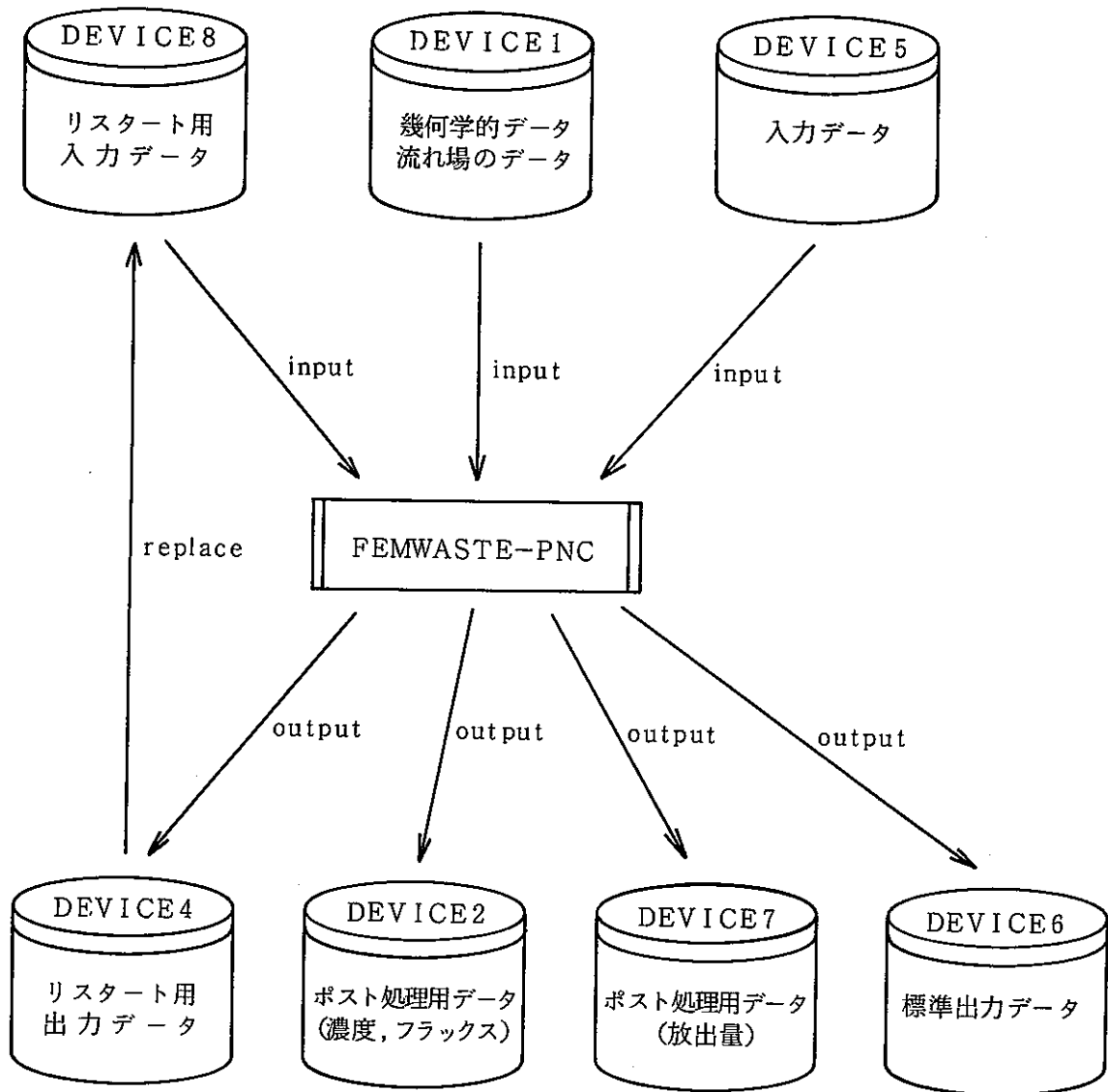


図-5-1 「FEMWASTE-PNC」ファイルの関係

5.2 「FEMWASTE-PNC」用の入力データファイル

「FEMWASTE-PNC」用に入力データファイルは、「FEMWASTE-PNC」に関連する7個のファイルの内、唯一ユーザーが編集を行うファイルであり、計算条件の指定を行う。このファイルは、カードイメージで作成するようになっており、次に示すカードから構成される。

- ① タイトカード
- ② コントロールカード
- ③ 物性カード
- ④ 初期条件カード
- ⑤ 境界条件カード

入力仕様は APPENDIX B に示されている。

計算領域の幾何学的情報及び流れ場の情報は、「FEMWASTE-PNC」の対となっている飽和・不飽和浸透流計算コード「FEMWATER」によって提供される。

5.3 幾何学的データ及び流れ場のデータファイル

幾何学的データと流れ場のデータは同一のファイルに、幾何学的データ、流れ場のデータの順で収納されており、本データファイルは通常「FEMWATER」によって提供され、その内容は書式なし WRITE 文で書かれている。

幾何学的データ及び流れ場のデータは以下に示すデータから構成されている。

- 幾何学的データ：
 - ① 全節点
 - ② 全要素
 - ③ 境界節点
 - ④ 境界要素
- 流れ場のデータ：
 - ① 時間
 - ② 圧力水頭
 - ③ 全水頭
 - ④ 体積含水率
 - ⑤ 流速

また、幾何学的データ及び流れ場のデータは以下に示す変数の並びで「FEMWASTE-PNC」に読み込まれる。

- 幾何学的データ： { READ(1) (TITLE(I),I=1,9), NPROBM,NNP,NEL,NBN,NBEL,NTIM
 READ(1) (X(NP),NP=1,NNP), (Z(NP),NP=1,NNP), ((IE(M,IQ), M=1, NEL),
 IQ=1, 5), (DLB(M),M=1, NBEL), (DCOSXB(M),M=1, NBEL), (DCOSZB(M),M=
 1, NBEL), (NBE(M),M=1, NBEL), ((ISB(M,IQ),M=1, NBEL),IQ=1, 4), (NPB(NP),
 NP=1, NBN)

• 流れ場のデータ :

READ(1) TIMEM, (H(NP),NP=1, NNP), (HT(NP),NP=1, NNP), ((TH(M,IQ),
M=1, NEL),IQ=1, 4), (VX(NP),NP=1, NNP), (VZ(NP),NP=1, NNP)

ここに, X : 節点の x 座標 (L)
 Z : 節点の z 座標 (L)
 IE : 要素を構成する節点番号及び材質番号 (-)
 DLB : 境界を構成する要素の境界面の長さ (L)
 DCOSXB : DLB の x 方向の方向余弦 (-)
 DCOSZB : DLB の z 方向の方向余弦 (-)
 NBE : 境界を構成する要素番号 (-)
 ISB : 境界要素における境界面を構成する節点 (-)
 NPB : 境界を構成する節点番号 (-)
 NNP : 全節点数 (-)
 NEL : 全要素数 (-)
 NBEL : 全境界要素数 (-)
 NBU : 全境界節点数 (-)
 TIMEM : 時間 (T)
 H : 圧力水頭 (L)
 HT : 全水頭 (L)
 TH : 体積含水率 (-)
 VX : x 方向の流速 (L/T)
 VZ : z 方向の流速 (L/T)

5.4 リスタート用入力ファイル及び出力ファイル

「FEMWASTE-PNC」のリスタート用入力ファイルは、前に行った「FEMWASTE-PNC」の実行終了時点における核種の濃度分布が収納されているファイルである。「FEMWASTE-PNC」はこのリスタート用入力ファイルを用いて、計算を継続的に行う事ができるようになる。よって、実行終了時点以降の濃度分布が知りたい場合、「FEMWASTE-PNC」においては、再び時間0から計算を始める必要はなくリスタートファイル内に収納されている前の計算終了時の濃度分布を初期条件として、計算を行い、その時間以降の核種の濃度分布を知る事ができる。

「FEMWASTE-PNC」は常にリスタートが行えるように計算終了時点における濃度分布を収納したファイルを作成している。ユーザーはリスタートを行いたい場合、このリスタート用出力ファイルをリスタート用入力ファイルにリプレースを行い、入力データにおいてリスタートを行うためのフラグを立てる事により、リスタートを行う事ができる。

リスタート用ファイルは書式なしでデータが収納されており、以下に示すような並びで「FEMWASTE-PNC」に読み込まれる。

```
READ(8) TIMR, ((RP(I,J),I=1,NNP), J=1,NRAM)
```

ここに、TIMR : 計算終了時点における時間 (T)

RP : 核種濃度 (M/L^3)

NNP : 全節点数 (-)

NRAM : 崩壊系列に含まれる核種の数 (-)

5.5 標準出力データファイル

「FEMWASTE-PNC」用の標準出力データファイルは、プリンタ出力用にデータが編集されて収納されているファイルである。その内容は以下の通りである。

- ① 入力データのエコー
- ② 核種の変化量, 変化率
- ③ 核種濃度
- ④ 核種フラックス

5.6 ポスト処理用データファイル

「FEMWASTE-PNC」のポスト処理用のデータファイルは、計算終了後に図形処理を行い、核種の濃度や放出量の場所的、時間的变化を視覚的に把える為に作成される。このファイルは核種濃度、フラックス用と核種放出量用の2つのポストファイルに分れている。その内容は以下の通りである。

- 核種濃度, フラックス用 : ① 時間
 - ② 核種濃度分布
 - ③ 核種フラックス分布
- 核種放出量用 : ① 時間
 - ② 地表面からの核種放出量

また、これらのポストファイル内のデータは以下に示す変数の並びで「FEMWASTE-PNC」から書き出される。

- 核種濃度, フラックス用 :

```
WRITE(2) IRAM, IRAM, TIME, (R(NP),NP=1,NNP), (FX(NP),NP=1,NNP),  
(FZ(NP),NP=1,NNP)
```

- 核種放出量用 :

```
WRITE(7) TIME, (TGFLOW(II),II=1,NRAM)
```

ここに、 TIME : 時間 (T)
R : 核種濃度 (M/L^3)
FX : x方向の核種フラックス ($M/L^2 T$)
FZ : z方向の核種フラックス ($M/L^2 T$)
TGFLOW : 核種放出量 (M/T)
IRAM : 核種の ID 番号 (-)
NNP : 全節点数 (-)
NRAM : 崩壊系列に含まれる核種の数 (-)

6. 検 証

「FEMWASTE-PNC」は、より現実的に地層処分システムの評価を行う為に、「FEMWASTE」に対して、3.1節に示すような機能を追加したコードである。その数理モデルに対しては湧き出し、崩壊系列の効果を考慮しているが、本章ではその機能の検証を行った結果を述べる。

6.1 湧き出し

湧き出しすなわちソースタームを考慮した場合の計算が正しく行われているか否を確認する為に、有限差分法により定式化が行われている非定常熱輸送計算コード「TRUMP」との比較を行った。比較を行う為のコードとしてTRUMPを選んだ理由は、両コードの定式化が行われている微分方程式が同型である事による。

6.1.1 計算モデル

本検証に対しては、図-6-1に示すような1次元モデルを考えた。図-6-1は「FEMWASTE-PNC」に対応した計算モデルである。

図中のQが湧き出し（単位時間、単位体積当りに発生する量）を表しており、モデル左端から195 mの位置にあると仮定している。

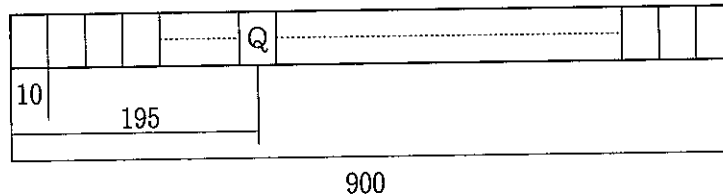


図-6-1 計算モデルの概念図

6.1.2 初期条件

計算開始時においては計算領域の濃度を0とする。図-6-2は初期条件を示す。

$$c(t=0) = 0$$

図-6-2 初期条件

6.1.3 境界条件

本計算を行う上での境界条件を図-6-3に示す。

$$\frac{\partial c}{\partial n} = 0 \quad \left[\text{---} \right] \quad c = 0$$

$$\frac{\partial c}{\partial n} = 0$$

図-6-3 境界条件

6.1.4 その他の計算条件

計算は上で述べた初期条件，境界条件の下で，移流の効果が無視でき，拡散のみで湧き出した物質が運ばれる系を考えた。

この系を与える計算条件を表-6-1に示す。

表-6-1 計算条件

湧き出し($g/m^3 \cdot y$)	1.0×10^1 ($t = 0 \sim 10y$, $x = 195 \text{ m}$)
拡散係数(m^2/y)	1.0×10^1
流速(m/y)	1.0×10^{-12}
計算時間(y)	1.0×10^3

6.1.5 計算結果

図-6-4に「FEMWASTE-PNC」と「TRUMP」の計算結果を示す。 $t = 100 \text{ y}$, 500 y , 1000 y の場合の濃度分布である。「FEMWASTE-PNC」は節点，「TRUMP」は要素中心で濃度の計算を行っている為， x 座標のプロット位置は異なっているが，各点を結ぶ曲線を考えると両者はほとんど一致していると思なす事ができる。よって「FEMWASTE-PNC」の湧き出しの計算は正常に行っていると考えられる。

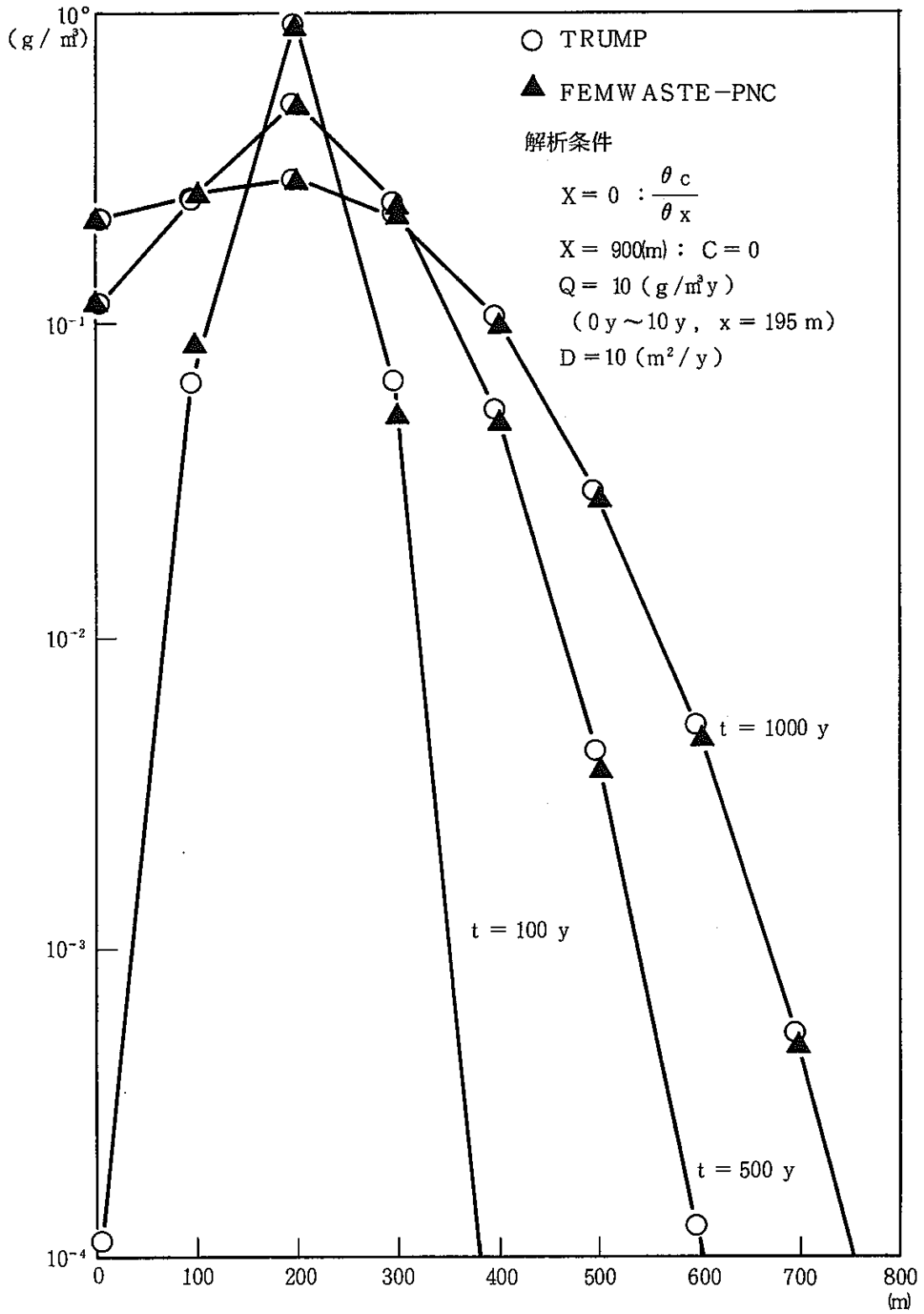


図-6-4 「FEMWASTE-PNC」と「TRUMP」の比較

6.2 崩壊系列

3メンバーを有する崩壊系列の計算が正しく行われているか否かを確認する為に Bateman の方程式の解析解との比較を行った。

Bateman の方程式は次元をもたない全体の量としての比較しか行えない為、「FEMWASTE-PNC」の計算においては、計算モデルからできるだけ核種が外に出て行かないように流速及び拡散係数を小さくとり、計算モデル内に存在する核種の全量で評価を行った。

6.2.1 Bateman の方程式の解析解

3メンバーを有する崩壊系列における Bateman の方程式の解析解は次式で示される。

1) 親核種 (member 1)

$$N_1 = N_1^0 e^{-\lambda_1 t} \quad (6.1)$$

2) 娘核種 (member 2)

$$N_2 = \lambda_1 N_1^0 \left(\frac{e^{-\lambda_1 t}}{\lambda_2 - \lambda_1} + \frac{e^{-\lambda_2 t}}{\lambda_1 - \lambda_2} \right) \quad (6.2)$$

3) 孫娘核種 (member 3)

$$N_3 = \lambda_1 \lambda_2 N_1^0 \left\{ \frac{e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} + \frac{e^{-\lambda_2 t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{e^{-\lambda_3 t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} \right\} \quad (6.3)$$

ここに、 N_1 : 時間 t における親核種 (member 1) の原子数 (-)

N_2 : 時間 t における娘核種 (member 2) の原子数 (-)

N_3 : 時間 t における孫娘核種 (member 3) の原子数 (-)

N_1^0 : 時間 $t = 0$ における親核種 (member 1) の原子数 (-)

λ_1 : 親核種 (member 1) の崩壊定数 (1/T)

λ_2 : 娘核種 (member 2) の崩壊定数 (1/T)

λ_3 : 孫娘核種 (member 3) の崩壊定数 (1/T)

t : 時間 (T)

6.2.2 計算モデル

「FEMWASTE-PNC」の崩壊系列の計算に用いたモデルは、湧き出しの計算に用いたモデルと同一であり、図-6-1に示される。

6.2.3 初期条件

計算開始時においては、各 member の濃度を 0 とする。図-6-5に初期条件を示す。

・親核種 (member 1)

$$c_1 (t = 0)$$

- 娘核種 (member 2)

$$c_2 (t = 0)$$

- 孫娘核種 (member 3)

$$c_3 (t = 0)$$

図-6-5 初期条件

6.2.4 境界条件

本計算を行う上での境界条件を図-6-6に示す。

- 親核種 (member 1)

$$\frac{\partial c_1}{\partial n} = 0 \quad \frac{\partial c_1}{\partial n} = 0 \quad c_1 = 0$$

- 娘核種 (member 2)

$$\frac{\partial c_2}{\partial n} = 0 \quad \frac{\partial c_2}{\partial n} = 0 \quad c_2 = 0$$

- 孫娘核種 (member 3)

$$\frac{\partial c_3}{\partial n} = 0 \quad \frac{\partial c_3}{\partial n} = 0 \quad c_3 = 0$$

図-6-6 境界条件

6.2.5 その他の計算条件

計算は上で述べた初期条件、境界条件の下で、移流及び拡散により運ばれる核種ができるだけ外部に出て行かないで、計算モデルの中で質量が保存される系を考えた。計算は各メンバーの半減期が短い場合、長い場合の2ケース考えた。

核種の半減期の短い場合と長い場合で用いた計算条件をそれぞれ表-6-2、表-6-3に示す。

表-6-2 核種の半減期が短い場合の計算条件

	member 1	member 2	member 3
湧き出し (g/m ³ y)	1.0×10 ¹ (t=1~10y)	0	0
拡散係数 (m ² /y)	1.0 × 10 ⁻⁵	1.0 × 10 ⁻⁵	1.0 × 10 ⁻⁵
流 速 (m/y)	1.0 × 10 ⁻²⁰	1.0 × 10 ⁻²⁰	1.0 × 10 ⁻²⁰
計算時間 (y)	1.0 × 10 ³	1.0 × 10 ³	1.0 × 10 ³
半 減 期 (y)	1.0 × 10 ³	1.0 × 10 ⁴	∞
崩壊定数 (1/y)	6.39×10 ⁻⁹	6.39×10 ⁻⁵	0

表-6-3 核種の半減期が長い場合の計算条件

	member 1	member 2	member 3
湧き出し (g/m ³ y)	1.0×10 ⁻¹ (t=1~1000y)	0	0
拡散係数 (m ² /y)	1.0 × 10 ⁻¹⁰	1.0 × 10 ⁻¹⁰	1.0 × 10 ⁻¹⁰
流 速 (m/y)	1.0 × 10 ⁻²⁰	1.0 × 10 ⁻²⁰	1.0 × 10 ⁻²⁰
計算時間 (y)	1.0 × 10 ⁶	1.0 × 10 ⁶	1.0 × 10 ⁶
半 減 期 (y)	1.0 × 10 ⁸	1.0 × 10 ⁵	∞
崩壊定数 (1/y)	6.39×10 ⁻⁹	6.39×10 ⁻⁶	0

また、湧き出しは親核種 (member 1) しか考慮しておらず、各核種の遅延はないものとしている。

6.2.6 計算結果

表-6-2, 表-6-3の計算条件により行った「FEMWASTE-PNC」とBatemanの式の解析解の計算結果を図-6-7, 図-6-8に示す。

また、これらの結果を数値で比較したのが表-6-4, 表-6-5である。これらの計算結果より、核種の半減期が短い場合、長い場合ともに「FEMWASTE-PNC」とBatemanの方程式の解はよく一致している。よって「FEMWASTE-PNC」の崩壊系列の計算は正常に行っていると考えられる。

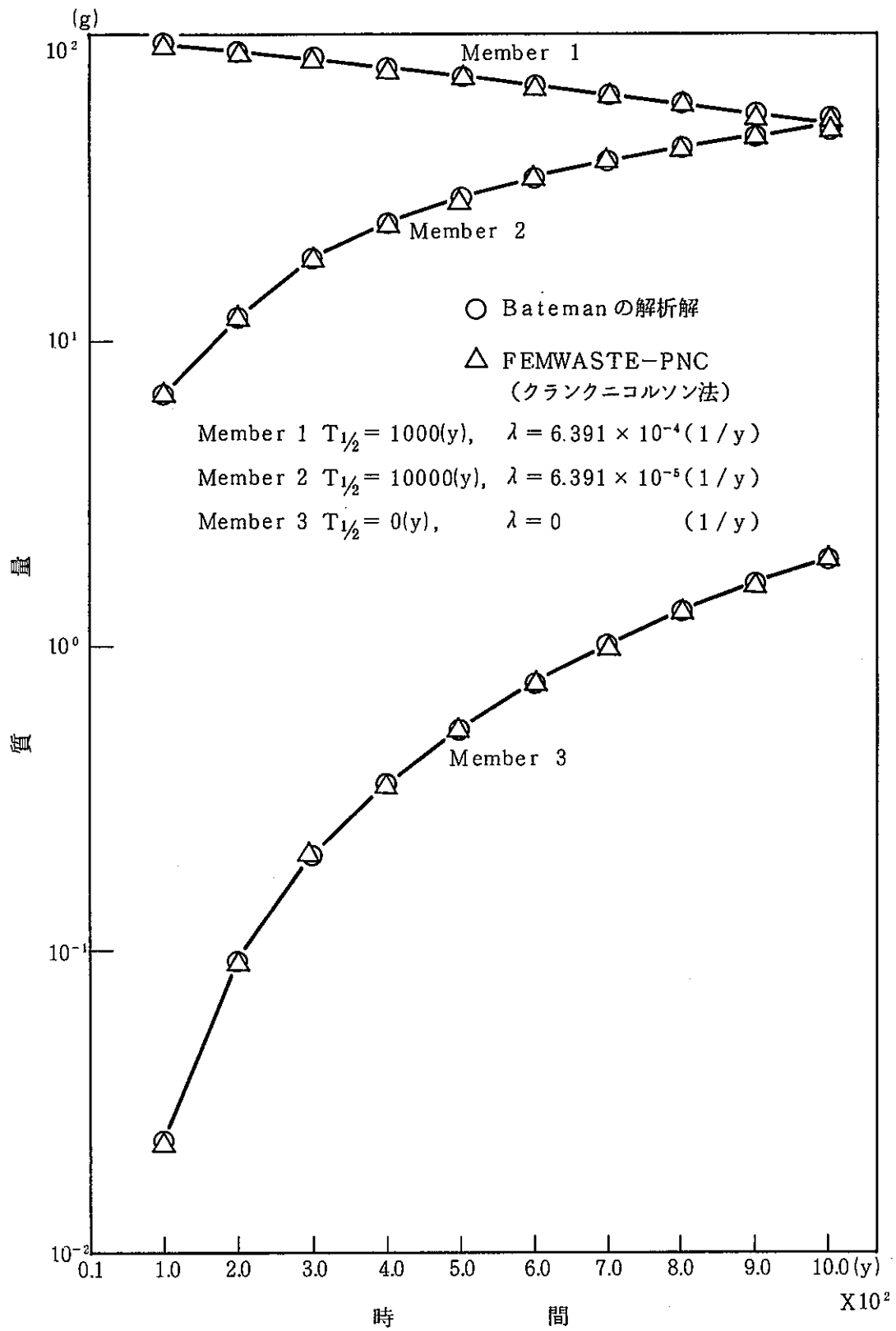


図-6-7 「FEMWASTE-PNC」と解析解の比較
 (核種の半減期が短い場合)

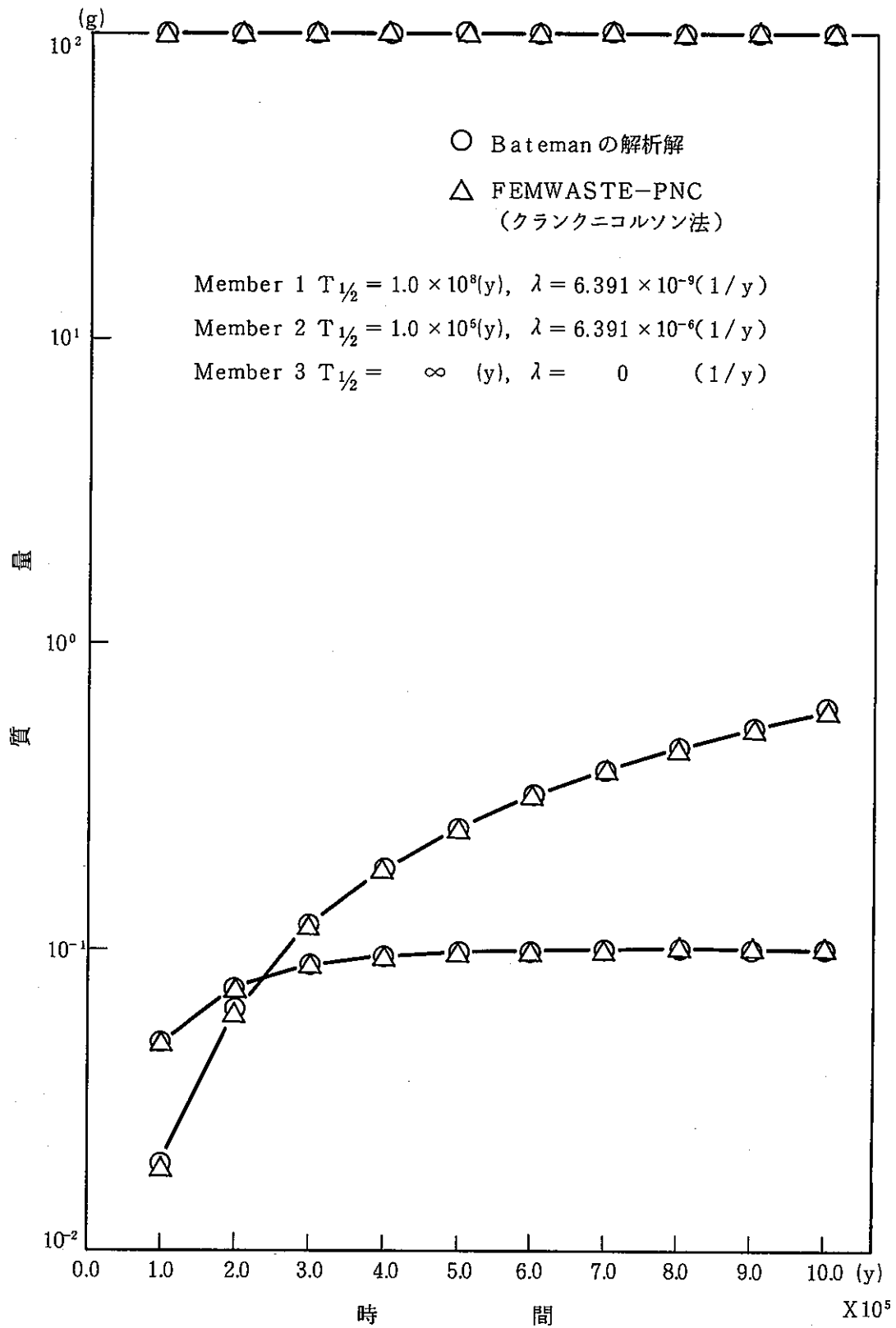


図-6-8 「FEMWASTE-PNC」と解析解の比較
 (核種の半減期が長い場合)

表-6-4 数値による「FEMWASTE-PNC」と解析解の比較
(核種の半減期が短い場合)

time(y)	member 1			member 2			member 3		
	Bateman	FEMWASTE	error(%)	Bateman	FEMWASTE	error(%)	Bateman	FEMWASTE	error(%)
100	9.3303E+01	9.3386E+01	0.0354	6.6733E+01	6.6086E+01	0.9695	2.3423E-02	2.2730E-02	2.9586
200	8.7055E+01	8.7086E+01	0.0356	1.2854E+01	1.2793E+01	0.4746	9.1372E-02	9.0028E-02	1.4709
300	8.1225E+01	8.1255E+01	0.0369	1.8574E+01	1.8518E+01	0.3015	2.0055E-01	1.9860E-01	0.9723
400	7.5786E+01	7.5813E+01	0.0356	2.3866E+01	2.3814E+01	0.2179	3.4788E-01	3.4535E-01	0.7273
500	7.0711E+01	7.0737E+01	0.0368	2.8759E+01	2.8701E+01	0.2017	5.3048E-01	5.2743E-01	0.5750
600	6.5975E+01	6.6000E+01	0.0379	3.3279E+01	3.3234E+01	0.1352	7.4570E-01	7.4215E-01	0.4760
700	6.1557E+01	6.1581E+01	0.0390	3.7452E+01	3.7410E+01	0.1121	9.9102E-01	9.8701E-01	0.4046
800	5.7435E+01	5.7457E+01	0.0383	4.1301E+01	4.1262E+01	0.0944	1.2641E+00	1.2596E+00	0.3560
900	5.3589E+01	5.3710E+01	0.0392	4.4848E+01	4.4812E+01	0.0803	1.5629E+00	1.5580E+00	0.3135
1000	5.0000E+01	5.0020E+01	0.040	4.8115E+01	4.8081E+01	0.0707	1.8852E+00	1.8800E+00	0.2758

表-6-5 数値による「FEMWASTE-PNC」と解析解の比較
(核種の半減期が長い場合)

time(y)	member 1			member 2			member 3		
	Bateman	FEMWASTE	error(%)	Bateman	FEMWASTE	error(%)	Bateman	FEMWASTE	error(%)
1.0×10^5	9.9931E+01	9.9931E+01	0.0	4.9981E-02	4.9631E-02	0.7003	1.9310E-02	1.8790E-02	2.6929
2.0×10^5	9.9861E+01	9.9862E+01	0.0010	7.4936E-02	7.4761E-02	0.2335	6.3597E-02	6.2812E-02	1.2343
3.0×10^5	9.9792E+01	9.9793E+01	0.0010	8.7380E-02	8.7292E-02	0.1007	1.2035E-01	1.1943E-01	0.7644
4.0×10^5	9.9723E+01	9.9724E+01	0.0010	9.3567E-02	9.3523E-02	0.0470	1.8331E-01	1.8232E-01	0.5401
5.0×10^5	9.9654E+01	9.9654E+01	0.0	9.6626E-02	9.6604E-02	0.0228	2.4935E-01	2.4832E-01	0.4131
6.0×10^5	9.9585E+01	9.9585E+01	0.0	9.8121E-02	9.8110E-02	0.0112	3.1690E-01	3.1586E-01	0.3282
7.0×10^5	9.9516E+01	9.9516E+01	0.0	9.8834E-02	9.8829E-02	0.0051	3.8519E-01	3.8413E-01	0.2752
8.0×10^5	9.9447E+01	9.9447E+01	0.0	9.9156E-02	9.9153E-02	0.0030	4.5383E-01	4.5276E-01	0.2358
9.0×10^5	9.9378E+01	9.9379E+01	0.0010	9.9282E-02	9.9281E-02	0.0010	5.2261E-01	5.2153E-01	0.2067
1.0×10^6	9.9309E+01	9.9310E+01	0.0010	9.9311E-02	9.9311E-02	0.0	5.9144E-01	5.9036E-01	0.1826

7. 結 言

本報告書は、今回開発を行った地層中核種移行計算コード「FEMWASTE-PNC」の概念、有限要素法による定式化、検証について述べたものである。オリジナルの「FEMWASTE」はアメリカのオークリッジ国立研究所（ORNL）で開発され、公開されている。このコードは核種の湧き出しや崩壊系列を考慮していない為に、地層処分システムの性能評価計算に使用するコードとしては不十分であった。そこで新たに、これらの機能を付加してより現実的な現象に対応できるようにしたのが「FEMWASTE-PNC」である。また、「FEMWASTE-PNC」を開発した事により、このコードを地下水移行シナリオに沿った、処分場におけるガラス固化体からの核種の放出から、地表での人間の被ばくに至る一連の性能評価モデルを記述する1つのプロセスとして位置づける事ができた。

しかしながら、「FEMWASTE-PNC」に関するモデル上及び計算上の問題点も多数残されている。これらの問題点を列挙すると以下ようになる。

- ① 崩壊系列が3メンバーまでしか取り扱えない。
- ② 線型吸着だけを考慮している。
- ③ 2次元モデルである。
- ④ 大量の計算領域が必要である。
- ⑤ 多大な計算時間を有する。

①に関しては、核種濃度を格納している配列の次元数を増やすだけで、比較的簡単に解決できるが、②、③に関しては、新たなモデルの組み込みを行い、プログラム全体を修正する必要があるため、非常に時間を要する事が予想される。④、⑤に関しては、プログラム自体の問題としてよりも使用している計算機の性能に大きく左右されるので、これらの問題を解決する為には、大量の計算領域を有する高速コンピュータの導入が必要となろう。

今後、これらの問題点を徐々に解決しつつ、現段階で最も適切と思われる計算条件を設定し、性能評価計算を行っていきたい。

APPENDIX A

要素の有限要素式の誘導

$$\begin{aligned}
& \int_{\text{Re}} \{W\}^T \{L(\hat{c})\} dR \\
&= \int_{\text{Re}} \{W\}^T \left\{ \theta R d \frac{\partial \hat{c}}{\partial t} + \left(V_x \frac{\partial \hat{c}}{\partial x} + V_z \frac{\partial \hat{c}}{\partial z} \right) \right. \\
&\quad \left. - \left[\theta D_{xx} \frac{\partial^2 \hat{c}}{\partial x^2} + (\theta D_{xz} + \theta D_{zx}) \frac{\partial^2 \hat{c}}{\partial x \partial z} + \theta D_{zz} \frac{\partial^2 \hat{c}}{\partial z^2} \right] \right. \\
&\quad \left. + \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \theta R d \right) \hat{c} \right\} dR = 0
\end{aligned}$$

$$\hat{c} = \{N\} \{c(t)\} \text{ を代入}$$

時間項：

$$\begin{aligned}
& \int_{\text{Re}} \{W\}^T \left[\theta R d \frac{\partial \hat{c}}{\partial t} \right] dR \\
&= \int_{\text{Re}} \theta R d \{W\}^T \{N\} dR \cdot \frac{\partial \{c(t)\}}{\partial t}
\end{aligned}$$

移流項：

$$\begin{aligned}
& \int_{\text{Re}} \{W\}^T \left[V_x \frac{\partial \hat{c}}{\partial x} + V_z \frac{\partial \hat{c}}{\partial z} \right] dR \\
&= - \int_{\text{Re}} \left[\frac{\partial \{W\}^T}{\partial x} \cdot V_x \{N\} + \frac{\partial \{W\}^T}{\partial z} V_z \{N\} \right] dR \cdot \{c(t)\} \\
&\quad + \int_{\text{Be}} \{W\}^T [V_x c_B + V_z c_B] dB
\end{aligned}$$

分散項：

$$\begin{aligned}
& \int_{\text{Re}} \{W\}^T \left[\theta D_{xx} \frac{\partial^2 \hat{c}}{\partial x^2} + (\theta D_{xz} + \theta D_{zx}) \frac{\partial^2 \hat{c}}{\partial x \partial z} + \theta D_{zz} \frac{\partial^2 \hat{c}}{\partial z^2} \right] dR \\
&= - \int_{\text{Re}} \left[\theta D_{xx} \frac{\partial \{W\}^T}{\partial x} \frac{\partial \{N\}}{\partial x} + \theta D_{xz} \frac{\partial \{W\}^T}{\partial x} \frac{\partial \{N\}}{\partial z} + \right. \\
&\quad \left. + \theta D_{zx} \frac{\partial \{W\}^T}{\partial z} \frac{\partial \{N\}}{\partial x} + \theta D_{zz} \frac{\partial \{W\}^T}{\partial z} \frac{\partial \{N\}}{\partial z} \right] dR \cdot \{c(t)\} \\
&\quad + \int_{\text{Be}} \{W\}^T \left[\theta D_{xx} \frac{\partial c_B}{\partial x} + \theta D_{xz} \frac{\partial c_B}{\partial z} + \theta D_{zx} \frac{\partial c_B}{\partial x} + \theta \frac{\partial c_B}{\partial z} \right] dR
\end{aligned}$$

その他の項：

$$\begin{aligned}
& \int_{\text{Re}} \{W\}^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \theta R d \right) \hat{c} dR \\
&= \int_{\text{Re}} \{W\}^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \theta R d \right) \{N\} dR \cdot \{c(t)\}
\end{aligned}$$

$$\begin{aligned}
& \int_{\text{Re}} \theta \text{Rd} \{W\}^T \{N\} \text{dR} \cdot \frac{\partial \{c(t)\}}{\partial t} \\
& - \int_{\text{Re}} \left[\frac{\partial \{W\}^T}{\partial x} V_x \{N\} + \frac{\partial \{W\}^T}{\partial z} V_z \{N\} \right] \text{dR} \cdot \{c(t)\} \\
& + \int_{\text{Be}} \{W\}^T \{V_x c_B n_x + V_z c_B n_z\} \cdot \text{dB} \\
& + \int_{\text{Re}} \left[\theta D_{xx} \frac{\partial \{W\}^T}{\partial x} \frac{\partial \{N\}}{\partial x} + \theta D_{xz} \frac{\partial \{W\}^T}{\partial x} \frac{\partial \{N\}}{\partial z} \right. \\
& \left. + \theta D_{zx} \frac{\partial \{W\}^T}{\partial z} \frac{\partial \{N\}}{\partial x} + \theta D_{zz} \frac{\partial \{W\}^T}{\partial z} \frac{\partial \{N\}}{\partial z} \right] \text{dR} \cdot \{c(t)\} \\
& + \int_{\text{Be}} \{W\}^T \left[\left(\theta D_{xx} \frac{\partial c_B}{\partial x} + \theta D_{xz} \frac{\partial c_B}{\partial z} \right) \cdot n_x + \left(\theta D_{zx} \frac{\partial c_B}{\partial x} + \theta D_{zz} \frac{\partial c_B}{\partial z} \right) \cdot n_z \right] \cdot \text{dB} \\
& + \int_{\text{Re}} \{W\}^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta \text{Rd} \frac{\partial h}{\partial t} + \lambda \cdot \theta \text{Rd} \right) \{N\} \text{dR} \cdot \{c(t)\} = 0
\end{aligned}$$

$$\begin{aligned}
& \int_{\text{Re}} \theta \text{Rd} \{W\}^T \{N\} \text{dR} \cdot \frac{\partial \{c(t)\}}{\partial t} \\
& + \int_{\text{Re}} \left[\frac{\partial \{W\}^T}{\partial x} \left(\theta D_{xx} \frac{\partial \{N\}}{\partial x} + \theta D_{xz} \frac{\partial \{N\}}{\partial z} \right) + \frac{\partial \{W\}^T}{\partial z} \left(\theta D_{zx} \frac{\partial \{N\}}{\partial x} + \theta D_{zz} \frac{\partial \{N\}}{\partial z} \right) \right. \\
& \left. - \left(\frac{\partial \{W\}^T}{\partial x} V_x \{N\} + \frac{\partial \{W\}^T}{\partial z} V_z \{N\} \right) \right. \\
& \left. + \{W\}^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta \text{Rd} \frac{\partial h}{\partial t} + \lambda \cdot \theta \text{Rd} \right) \{N\} \right] \text{dR} \cdot \{c(t)\} \\
& + \int_{\text{Be}} \{W\}^T \left[\left(-\theta D_{xx} \frac{\partial c_B}{\partial x} - \theta D_{xz} \frac{\partial c_B}{\partial z} + V_x c_B \right) n_x \right. \\
& \left. + \left(-\theta D_{zx} \frac{\partial c_B}{\partial x} - \theta D_{zz} \frac{\partial c_B}{\partial z} + V_z c_B \right) n_z \right] \text{dB} = 0
\end{aligned}$$

ここで

$$\{ n \} = \int_{Re} \theta R d \{ W \}^T \{ N \} dR$$

$$\begin{aligned} \{ s \} = \int_{Re} & \left[\frac{\partial \{ W \}^T}{\partial x} \left(\theta D_{xx} \frac{\partial \{ N \}}{\partial x} + \theta D_{xz} \frac{\partial \{ N \}}{\partial z} \right) + \frac{\partial \{ W \}^T}{\partial z} \left(\theta D_{zx} \frac{\partial \{ N \}}{\partial x} + \theta D_{zz} \frac{\partial \{ N \}}{\partial z} \right) \right. \\ & - \left(\frac{\partial \{ W \}^T}{\partial x} V_x \{ N \} + \frac{\partial \{ W \}^T}{\partial z} V_z \{ N \} \right) \\ & \left. + \{ W \}^T \left(\frac{\partial \theta}{\partial t} + \alpha' \theta R d \frac{\partial h}{\partial t} + \lambda \cdot \theta R d \right) \{ N \} \right] dR \end{aligned}$$

$$\begin{aligned} \{ q \} = \int_{Be} & \{ W \}^T \left[\left(-\theta D_{xx} \frac{\partial c_B}{\partial x} - \theta D_{xz} \frac{\partial c_B}{\partial z} + V_x c_B \right) n_x \right. \\ & \left. + \left(-\theta D_{zx} \frac{\partial c_B}{\partial x} - \theta D_{zz} \frac{\partial c_B}{\partial z} + V_z c_B \right) n_z \right] dB \end{aligned}$$

$$\{ \dot{c} \} = \frac{\partial \{ c(t) \}}{\partial t}$$

$$\therefore \{ m \} \{ \dot{c} \} + \{ s \} \{ c \} + \{ q \} = 0$$

APPENDIX B

入 力 仕 様

0 = 記憶しない

1 = 論理装置 2 に結果を記憶する

KSS : 定常状態コントロール

0 = 最終的に得られる解 ($NTI=0$) として、あるいは非定常解 ($NTI>0$) を得る
 為の初期条件として定常解が求められる

1 = 入力データに従って初期条件を下に非定常解が求められる ($NTI>0$ でなければ
 ならない)

NMPPM : 1つの材質に対して定義される物性値の数

9 = 現在の「FEMWASTE-PNC」のモデルで定義されている数

IWET : 重み関数コントロール

0 = ガラーキン重み関数

1 = 上流重み関数

ILUMP : 質量マトリックスランピングコントロール

0 = 質量ランピングを考えない

1 = 質量ランピングを考える

IMID : 中間差分法による時間積分

0 = 中間差分法を適用しない

1 = 中間差分法を適用する

ISW1 : 湧き出し (ソースターム) の経時変化を与えるスイッチ

0 = 経時変化を考えない

1 = 経時変化を考える

ISW2 : 境界におけるフラックス (Cauchy 型境界条件) の経時変化を与えるスイッチ

0 = 経時変化を考えない

1 = 経時変化を考える

IRES : リスタートコントロール

0 = リスタートではない通常的时间 0 からの計算を行う

1 = リスタートファイルを読み込む事によりリスタートを行う。

3. 基本的な実数パラメータ

1つの問題に1枚必要である。E型, D型, F型のいずれのタイプでも入力フィールドに書き
 込める。

DELT	CHNG	DELMAX	TMAX	W	
10	20	30	40	50	80

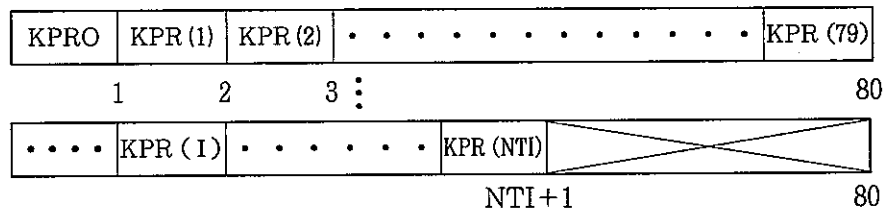
- DELT : 初期の時間スラップサイズ (T)
 CHNG : 次式による DELT の決定に使用されている乗数 (-)
 $DELT = DELT * (1.0 + CHNG)$
 DELMAX : 最大の時間ステップサイズ (T)
 TMAX : 最大の計算時間 (T)
 W : 時間微分に対する重みファクター

4 a. プリンターへの出力コントロール

必要なカードの枚数は時間ステップ数 NTI に依存しており、次式によって計算される。

$$[(NTI + 1) / 80 + 1]$$

各カードは次に示すように 80 カラムを有するカードの 1 カラムが 1 タイムステップに相当する。



KPRO : 定常状態あるいは初期条件に関するプリンタコントロール

- 0 = プリントしない
- 1 = すべての境界から流入あるいは流出する流量及び蓄積量をプリントする
- 2 = 1 の場合に加えて、濃度分布をプリントする
- 3 = 2 の場合に加えて、フラックス分布をプリントする

KPR(I) : 時間ステップ上における非定常解のプリンタコントロール

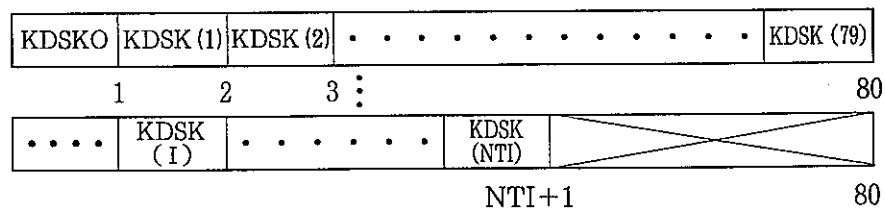
- 0 = プリントしない
- 1 = すべての境界から流入あるいは流出する流量及び蓄積量をプリントする
- 2 = 1 の場合に加えて、濃度分布をプリントする
- 3 = 2 の場合に加えて、フラックス分布をプリントする

4 b. 補助記憶コントロール

必要なカードの枚数は時間ステップ数 NTI に依存しており、次式によって計算される。

$$[(NTI + 1) / 80 + 1]$$

各カードは次に示すように 80 カラムを有するカードの 1 カラムが 1 タイムステップに相当する。



KDSKO : 定常状態あるいは初期条件に関する補助記憶コントロール

0 = 補助記憶を行わない

1 = 論理装置 2 に補助記憶を行う

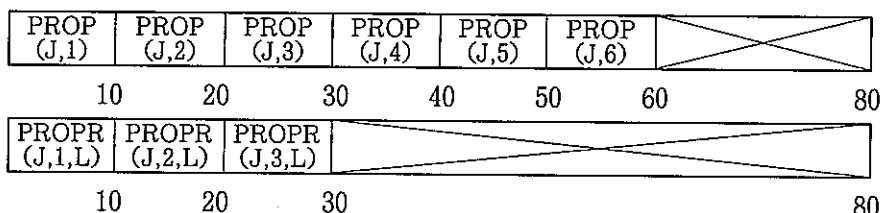
KDSK(I) : 時間ステップ上における非定常解の補助記憶コントロール

0 = 補助記憶を行わない

1 = 論理装置 2 に補助記憶を行う

5. 材質特性

全体で NMAT グループ分のカードが必要である。NMAT 個の材質に関して、それぞれグループに分けられている。各グループのカードはさらに輸送媒体と核種に関するデータに分けられる。現在の「FEMWASTE-PNC」のモデルでは1つの材質は、輸送媒体に関して6個、核種に関して3個の合計9個のデータから構成される。核種データのカードは各材質毎に NRAM 回繰り返す。



PROP(J,I) : J 番目の材質の輸送媒体の I 番目の材質特性

PROPR(J,K,L) : J 番目の材質の核種 L の K 番目の材質特性

PROP(J,1) : 輸送媒体 J の見かけ密度 (M/L^3)

PROP(J,2) : 輸送媒体 J の縦方向分散率 (L)

PROP(J,3) : 輸送媒体 J の横方向分散率 (L)

PROP(J,4) : 輸送媒体 J の空隙率 (-)

PROP(J,5) : 輸送媒体 J の圧縮率 ($1/L$)

PROP(J,6) : 輸送媒体 J の屈曲率 (-)

PROPR(J,1, L) : 輸送媒体 J における核種 L の分配係数 (L^3/M)

PROPR(J,2, L) : 輸送媒体 J における核種 L の崩壊定数 ($1/L$)

PROPR(J,3, L) : 輸送媒体 J における核種 L の分子拡散係数 (L^2/T)

6. 幾何学的データ

幾何学的データはフリーフォーマットで論理装置 1 から読み込まれる。本データは、節点座標、要素構成、境界面に関する情報から構成されている。カードイメージの入力は必要はないが、こ

のデータファイルは論理装置 1 に割り当てられていなければならない。

入力は Subroutine DATAIN で次のような順序で行われる。詳しくは 5.3 節を参照のこと。

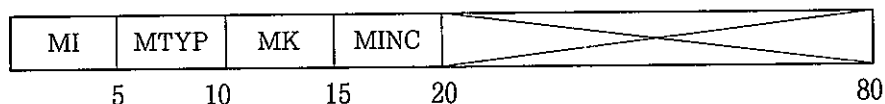
REWIND 1

```

READ(1) (TITLEM(I),I=1,9), NPROBM, NNP, NEL, NBN, NBEL, NTIM
READ(1) (X(NP),NP=1,NNP), (Z(NP),NP=1,NNP), ((IE(M,IQ),M=1,NEL),
IQ=1,5), (DLB(M),M=1,NBEL), (DCOSXB(M),M=1,NBEL), (DCOSZB(M),
M=1,NBEL), (NBE(M),M=1,NBEL), ((ISB(M,IQ),M=1,NBEL),IQ=1,4),
(NPB(NP),NP=1,NBN)
    
```

7. 材質の修正

本データは $NCM > 0$ の時にのみ必要である。通常、1つの材質の修正に対して1枚のカードが必要である。しかしながら、影響を受ける要素の数が増分 MINC で下限 MI から上限 MK に変わる場合、自動修正機能が使用される。MK と MINC のフィールドに関して、自動修正機能使用されない時は、それらのフィールドはブランクとすればよい。



MI : I 番目の材質の修正を行う要素番号の下限

MTYP : 材質 I の修正を行う材質のタイプ

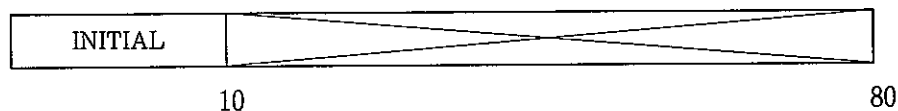
MK : 自動修正機能を用いて材質修正を行う時の要素番号の上限

MINC : 自動修正機能を用いて材質修正を行う時の要素番号の増分

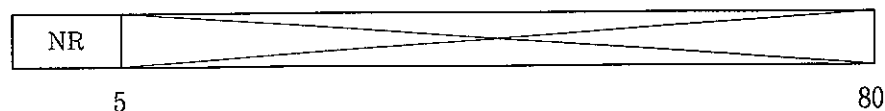
8. 初期条件

初期条件は、崩壊系列を構成する核種毎に指定を行う。最も一般的な場合、核種毎に、1つの節点当りに、1つのカードが必要である。すなわち1つの核種に対して、全節点数 NNP 分のカードが必要である。

初期条件を与えるデータセットは、初期条件を識別するカード、核種を識別するカード、及びその核種の初期濃度を示すカードから構成される。

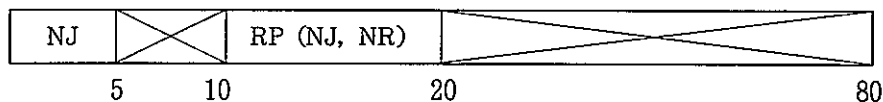


INITIAL : 初期条件のデータセットである事を識別する文字列



NR : 核種を識別する番号

CARD J:

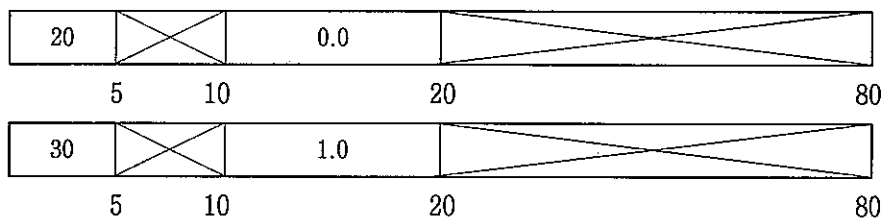


NJ : NJ 番目の節点番号

RP (NJ, NR) : 核種 NR の節点 NJ における初期濃度 (M/L³)

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードとその核種の初期濃度を示すカードからなるデータを NRAM 回繰り返す。

核種の初期濃度を示すカードにおいて、初期濃度 RP (NJ, NR) はしばしば、隣り合う接点どうし同じ値をとる。よって「FEMWATER-PNC」では、このような場合の入力として、節点番号の並びにギャップが存在するときには、初期濃度はギャップ内の節点に対して、ギャップの始まる前の節点濃度と同じ値が設定される。つまり、隣り合う接点どうしが次のようなギャップがある場合、



節点 21 から節 29 の初期濃度は RP=0.0 に割り当てられる。

・初期条件に関する注意

非定常計算に関する初期条件は2つの異なる方法で与えられる。1つはカード入力からのものであり、他1つは一連の境界条件と流れの場を使った定常計算から与えられるものである。後者に関しては、前初期条件の入力が必要である。

・定常あるいは非定常の入力に関する注意

定常状態のオプションは非定常計算におけるシステムの最終状態かあるいは初期条件を与える為に使用される。前者に関しては KSS=0 かつ NTI=0 である。一方、後者に関しては KSS=0 かつ NTI>0 である。KSS≠0 の時は、定常計算ではなく、以下に説明があるデータセット 9~13 が入力データとして考慮される。初期条件としての定常解を使用する時は、非定常計算に使用したのに加えて、一連の境界条件と流れ場が必要となる。入力データセット 9~13 は定常状態の境界条件 (KSS=0) あるいは非定常状態の境界条件 (KSS≠0) を定義する。

・リスタートデータに関する注意

リスタートを行うに際しても初期条件データセットは必要である。この場合、RP の値は計算上無視されるので、リスタートの初期条件を与える節点のみを指定すればよい。また、

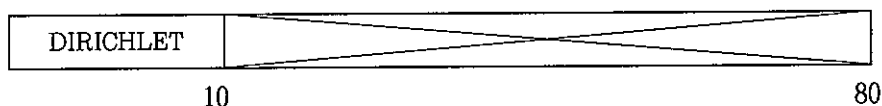
各核種の RP の値はリスタートファイルから読み込まれる。

9. Dirichlet 型境界条件 [(KSS=0, NTI=0), (KSS=1, NTI>0)]

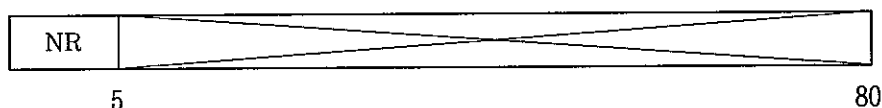
Dirichlet 型境界条件を与えるデータセットは、Dirichlet 型境界条件を識別するカード、核種を識別するカード、及び核種に対して Dirichlet 型境界条件を与えるカードから構成される。

境界条件を与えるカードは、NBC>0 の時にのみ必要である。自動作成機能を使用されない場合、NBC 個のカードが必要である。

カードのフォーマットを以下に示す。

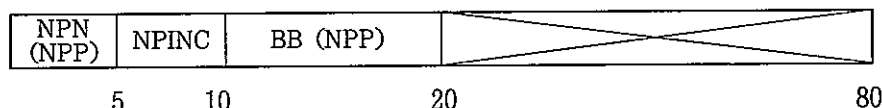


DIRICHLET : Dirichlet 型境界条件のデータセットであることを識別する文字列



NR : 核種を識別する番号

CARD J :



NPN(NPP) : NPP 番目の Dirichlet 型境界条件を指定する節点番号

NPINC : 節点 NPN までの節点の自動生成を行う場合の増分

BB(NPP) : 節点 NPN(NPP)における固定濃度 (M/L³)

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

NPINC>0 の時、自動生成機能はデータセット 8 に述べられているのと同じ方法で使用される。すなわち、指定したカードの直前のカードの節点番号から、増分に従って順次番号づけが行われる。

各節点の指定濃度は境界条件を与えるカードの BB の値が与えられる。

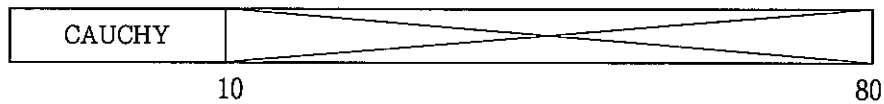
10. Cauchy 型境界条件 [(KSS=0, NTI=0), (KSS=1, NTI>0)]

Cauchy 型境界条件を与えるデータセットは、Cauchy 型境界条件を識別するカード、核種を識別するカード、及び核種に対して Cauchy 型境界条件を与えるカードから構成される。

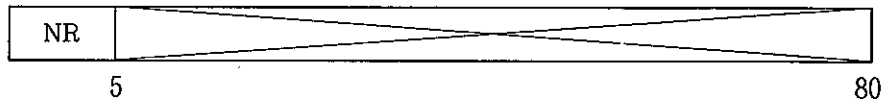
境界条件を与えるカードは、NST>0 の時にのみ必要である。通常 NST に等しい数のカードが使用される。しかしながら、KINC に 0 より大きい数を指定すると、「FEMWASTE-PNC」の内部で自動生成機能により、その指定に従って境界条件を与えるカードが生成されるので、

NST 個のカードは必要でない。

カードのフォーマットを以下に示す。

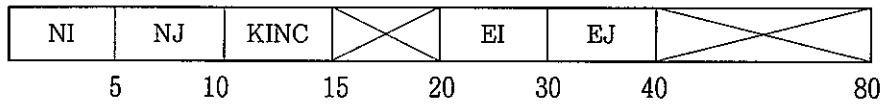


CAUCHY : Cauchy 型境界条件のデータセットである事を識別する文字列



NR : 核種を識別する番号

CARD MPP :



NI : MPP 番目の境界に接する要素の辺の 1 つの節点番号

NJ : MPP 番目の境界に接する要素の辺の他 1 つの節点番号

KINC : NI と NJ を生成する, 自動生成機能を使用する為の増分

EI : MPP 番目の境界に接する要素の辺上の節点 NI における法線方向のフラックス (M/LT)

EJ : MPP 番目の境界に接する要素の辺上の節点 NJ における法線方向のフラックス (M/LT)

崩壊系列が NRAM 個の核種で構成されている場合, 核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

KINC > 0 の時, 節点の増分は直前のカードの NI と NJ から形成される。

$$NPINC = NJ - NI$$

とすると, 自動生成機能を使用する場合 NI と NJ は次のように変化する。

$$NI + NPINC, NI + 2 * NPINC, \dots\dots$$

$$NJ + NPINC, NJ + 2 * NPINC, \dots\dots$$

これらの値は最も大きい整数が現在の NI と NJ の値より大きくならなように逐次設定される。

• Cauchy 型境界条件の時間変化に関する注意

Cauchy 型の境界条件においては, 境界から流入あるいは流出するフラックスの時間変化を取り扱う事ができる。この場合, EI と EJ に相当するフラックスは, 外部のファイルから読み込まれるようになっている。

よって, Cauchy 型境界条件の時間変化を考慮する場合は, 境界の接点番号と増分すなわち NI, NJ, KINC のみを指定しておけばよい。

11. Neumann 型境界条件 [(KSS = 0, NTI = 0), (KSS = 1, NTI > 0)]

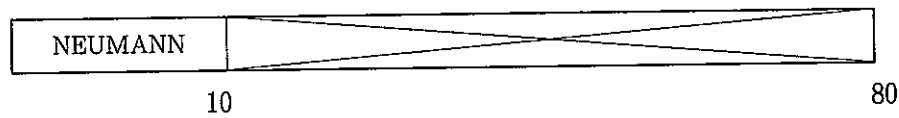
Neumann 型境界条件を与えるデータセットは, Neumann 型境界条件を識別するカード, 核

種を識別するカード及び核種に対して Neumann 型境界条件を与えるカードから構成される。

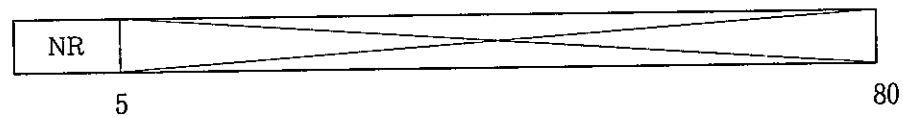
境界条件を与えるデータセットは、NRSEL>0の時にのみ、必要である。本データセットは降雨浸透の境界条件のようなものである。流れが領域外に流出する場合は、その境界は濃度依存している Neumann 境界であり、そのように適用できる。流れが領域内に流入する場合は、その境界はフラックスが0である Cauchy 型境界条件となるが、入力データとして指定する必要はない。

よって本データセットは Neumann 型境界条件と呼ばれる。基本的には、境界に接する要素の辺の数だけのカードが必要である。

カードのフォーマットを以下に示す。

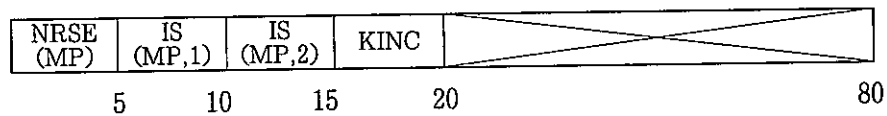


NEUMANN : Neumann 型境界条件のデータセットであることを識別する文字列



NR : 核種を識別する番号

CARD MP :



NRSE(MP) : MP 番目の Neumann 型境界を有する要素番号

IS(MP, 1) : MP 番目の Neumann 型境界を有する要素の境界における第1節点の番号

IS(MP, 2) : MP 番目の Neumann 型境界を有する要素の境界における第2節点の番号

KINC : NRSE と IS を自動生成する為の増分

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

KINC>0の時、自動生成機能により、節点番号と要素番号の増分は現カードの直前のカードの情報から生成される。

$$NPINC = IS(MP, 2) - IS(MP, 1)$$

$$MINC = NPINC - 1$$

要素番号は次のようにして得られる。

$$M = NRSE(MP) \text{ (前のカード)}$$

$$NRSE(MP+1) = M + MINC$$

$$NRSE(MP+2) = M + 2 * MINC$$

⋮

これは、最も大きい要素番号が現カードの NRSE に最も近づくまで続けられる。節点番号もまた、同様に次のようにして得られる。

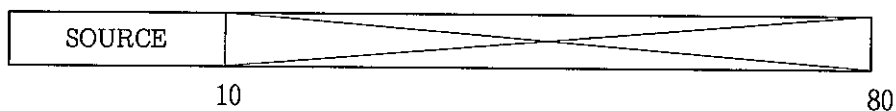
$$\begin{array}{ll}
 \text{NI}=\text{IS}(\text{MP}, 1) \text{ (前のカード)} & \text{NJ}=\text{IS}(\text{MP}, 2) \text{ (前のカード)} \\
 \text{IS}(\text{MP}+1)=\text{NI}+\text{NPINC} & \text{IS}(\text{MP}+1, 2)=\text{NJ}+\text{NPINC} \\
 \text{IS}(\text{MP}+2, 1)=\text{NI}+2 * \text{NPINC} & \text{IS}(\text{MP}+2, 2)=\text{NJ}+2 * \text{NPINC} \\
 \vdots & \vdots
 \end{array}$$

12. ソースターム

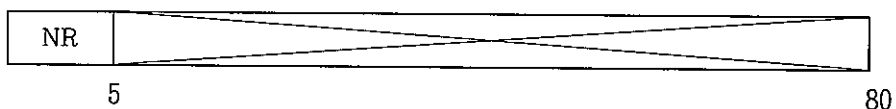
本データセットは、ソースタームを与えるデータである事を識別するカード、核種を識別するカード及びソースタームを与えるカードから構成される。

本データセットは、NSRC>0の時にのみ必要である。ソースタームを与えるカードはNSRCに等しいカードが各核種毎に必要なである。

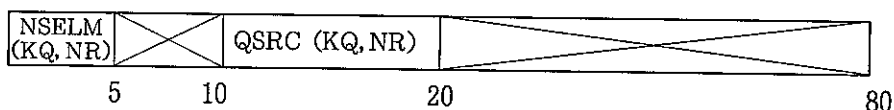
カードのフォーマットを以下に示す。



SOURCE : ソースタームを与えるデータセットである事を識別する文字列



NR : 核種を識別する番号



NSELM (KQ, NR) : KQ 番目のソースタームを与える要素番号

QSRC (KQ, NR) : 指定した要素におけるソース量 (M/L³T)

• ソースタームの時間変化に関する注意

ソースタームのデータセットの指定に関しては、指定した要素におけるソースタームの時間変化を取り扱う事ができる。

この場合、QSRC に相当する量は、外部のファイルから読み込まれるようになっている。よってソースタームの時間変化を考慮する場合は、ソースタームを考慮する要素番号すなわち NSELM のみを指定しておけばよい。

13. 初期あるいは定常状態の流れ変数

本データセットは論理装置 1 により書式なしで読み込まれる。データセットの内容は、時間 TIMEM, 圧力水頭 H, 全水頭 HT, ダルシー流速 VX, VZ で構成されている。詳しくは APPENDIX C の Subroutine GM2DXZ を参照されたい。

```

READ(1) TIMEM, (H(NP), NP=1, NNP), (HT(NP), NP=1, NNP),
((TH(M, IQ), M=1, NEL), IQ=1, 4), (VX(NP), NP=1, NNP)
(VZ(NP), NP=1, NNP)
    
```

KVI=1 の時は、これらの流れの場の変数は非定常流れについても使われる。

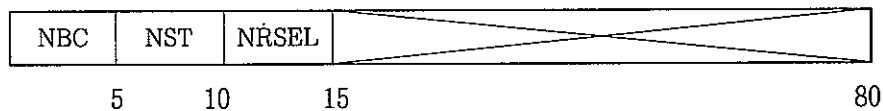
・ 非定常入力に関する注意

KSS=0, NTI>0, の時のみ、データセット 15~18 が必要となる。これらのデータセットは定常計算 (KSS=0, NTI=1), 非定常計算 (KSS=1, NTI>0) を定義するために使用されるデータセット 9~13 と同じである。そこでのほとんどの自動生成機能、記号の定義、他の入力に関する注意は、データセット 15~18 でも同じように適用される。カードの説明は以下に行われているが、これらは、NTI>0, KSS=0 のときにのみ必要となる。

KSS=1, NTI>0 のとき、非定常計算は初期状態としての定常解を得ることなしに、進められる。よって、データセット 15~18 において定義してあるすべての変数はデータセット 9~13 に定義してある変数と同じである。

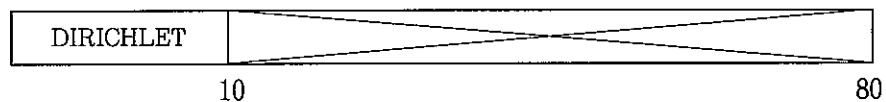
14. 非定常計算 (KSS=0, NTI>0) に関する整数パラメータ

1 つの問題に 1 枚必要

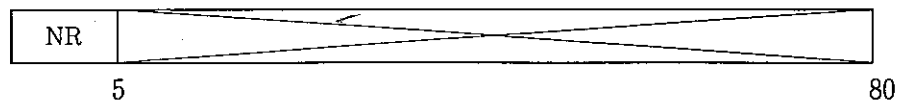


- NBC : Dirichlet 型境界条件を設定する節点数
- NST : Cauchy 型境界条件を設定する要素境界数
- NRSEL : Neumann 型境界条件を設定する要素境界数

15. Dirichle 型境界条件 (KSS=0, NTI>0)

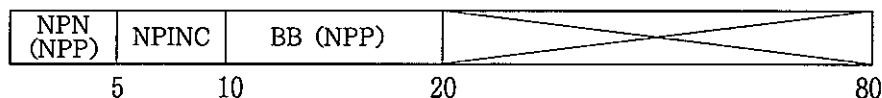


DIRICHLET : Dirichlet 型境界条件のデータセットであることを識別する文字列



NR : 核種を識別する番号

CARD J:



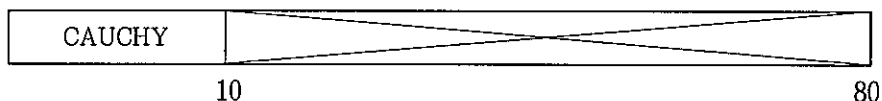
NPN(NPP) : NPP 番目の Dirichlet 型境界条件を指定する節点番号

NPINC : 節点 NPN までの節点の自動生成を行う場合の増分

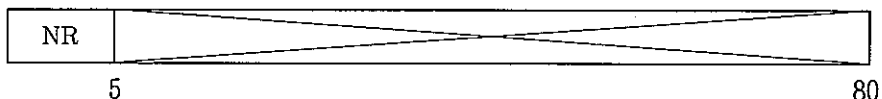
BB(NPP) : 節点 NPN(NPP)における固定濃度 (M/L³)

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

16. Cauchy 型境界条件 (KSS = 0, NTI > 0)

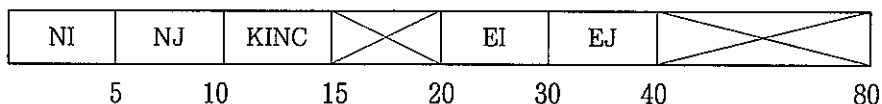


CAUCHY : Cauchy 型境界条件のデータセットである事を識別する文字列



NR : 核種を識別する番号

CARD MPP :



NI : MPP 番目の境界に接する要素の辺の 1 つの節点番号

NJ : MPP 番目の境界に接する要素の辺の他 1 つの節点番号

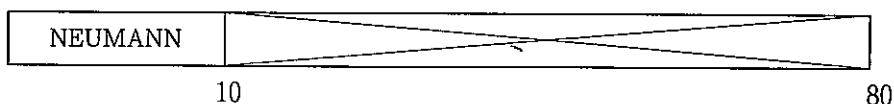
KINC : NI と NJ を生成する、自動生成機能を使用する為の増分

EI : MPP 番目の境界に接する要素の辺上の節点 NI における法線方向のフラックス (M/LT)

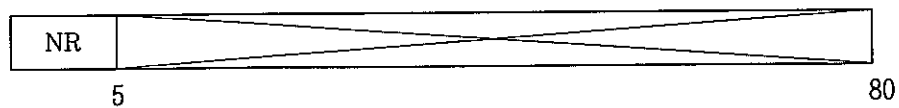
EJ : MPP 番目の境界に接する要素の辺上の節点 NJ における法線方向のフラックス (M/LT)

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

17. Neumann 型境界条件 (KSS = 0, NTI > 0)

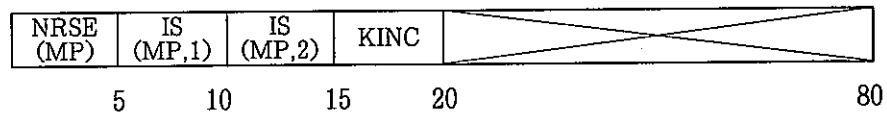


NEUMANN : Neumann 型境界条件のデータセットである事を識別する文字列



NR : 核種を識別する文字列

CARD MP :



NRSE(MP) : MP 番目の Neumann 型境界を有する要素番号

IS(MP,1) : MP 番目の Neumann 型境界を有する要素番号における第1 節点の番号

IS(MP,2) : MP 番目の Neumann 型境界を有する要素番号における第2 節点の番号

KINC : NRSE と IS を自動生成する為の増分

崩壊系列が NRAM 個の核種で構成されている場合、核種を識別するカードと境界条件を与えるカードからなるデータを NRAM 回繰り返す。

18. 非定常の流れ変数

本データセットは KVI=2 の時のみ必要であり、論理装置 1 により読み込まれる。本データセットに含まれる変数はデータセット 14 に述べられている内容と同じであり、同様のフォーマットで読み込まれる。詳しくは APPENDIX C のサブルーチン GM2DX2 を参照されたい。

APPENDIX C

ソースプログラムリスト

```

C
C *****
C *
C *
C *
C *
C *****

```

ADD S BY HATA

```

C THIS COMPUTER CODE IS REVISED FOR THREE MEMBER DECAY CHAIN OF
C RADIOACTIVE NUCLIDE BY HATA IN JULY 1989.
C THE CONTENT OF REVISION IS FOLLOWING :
C 1) ADDED CALCULATION OF DECAY CHAIN.
C 2) CHANGED BOUNDARY CONDITION FOR FLUX.
C 3) ADDED FUNCTION IN CONNECTING 'RELEASE' CODE ,WHICH CALCULATE
C NEAR FIELD RADIOACTIVE NUCLIDE TRANSPORT IN BUFFER MATERIAL.

```

ADD E

```

C THIS COMPUTER CODE IS CONTAINED IN THE FOLLOWING REPORT:
C YEH, G. T. AND D. S. WARD, 1980. #FEMWASTE: A FINITE-ELEMENT
C MODEL OF WASTE TRANSPORT THROUGH SATURATED-UNSATURATED POROUS
C MEDIA,# ORNL-5601, OAK RIDGE NATIONAL LABORATORY,
C OAK RIDGE, TN. 37830

```

```

C THIS VERSION OF THE FEMWASTE CODE IS MODIFIED TO OPERATE ON CDC
C EQUIPMENT. NEW OR CHANGED STATEMENTS FOR THIS PURPOSE ARE MARKED
C BY CDC BEGINNING IN COLUMN 75.

```

```

C A SLIGHTLY UPDATED VERSION IS CONTAINED IN:
C YEH, G. T. AND R. H. STRAND, 1981. #FECWASTE: USER'S MANUAL OF A
C FINITE ELEMENT CODE FOR SIMULATING WASTE TRANSPORT THROUGH
C SATURATED-UNSATURATED POROUS MEDIA,# ORNL-TM- , OAK RIDGE
C NATIONAL LABORATORY, OAK RIDGE, TN. 37830

```

FOR ANY QUESTION, PLEASE CONTACT DR. G. T. YEH AT (615) 574-7285

```

C ADDITIONAL REFERENCES IS:
C DUGUID, J. AND M. REEVES, 1976. #MATERIAL TRANSPORT THROUGH
C SATURATED-UNSATURATED POROUS MEDIA: A GALERKIN FINITE ELEMENT MODEL
C ORNL 4928, OAK RIDGE NATIONAL LABORATORY, OAK RIDGE, TENNESSEE 37830

```

MAIN PROGRAM

PROGRAM FEMWAS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2

```

C IMPLICIT REAL*8(A-H,O-Z)
C REAL*4 PMAT

```

CHANGE S ARRAY SIZE BY HATA

```

C DIMENSION X(595),Z(595),IE(528,5)
C DIMENSION C(595,29),R(595),RP(595),WETAB(528,4)
C DIMENSION FX(595),FZ(595)
CC
C DIMENSION VX(595),VXP(595),VZ(595),VZP(595),H(595),HP(595),

```

```

C   > DH(595),HT(595),TH(528,4),THP(528,4),DTH(528,4)
CC
C   DIMENSION DLB(199),DCOSXB(199),DCOSZB(199),BFLX(200),BFLXP(200),
C   > NBE(199),ISB(199,4),NPB(200)
C
C   DIMENSION DL(99),DCOSX(99),DCOSZ(99),NRSE(99),IS(99,4),NPRS(100)
C   DIMENSION NPN(30),BB(30),NPST(40),DP(40)
C   DIMENSION PMAT(3,9),PROP(3,9)
C   DIMENSION KPR(500),KDSK(500)
C   DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
C
C   MEM : ARRAY SIZE TO MEMBER IN A DECAY CHAIN
C
C   PARAMETER(MEM=3)
C   DIMENSION X(1100),Z(1100),IE(1100,5)
C   DIMENSION C(1100,80),R(1100),RP(1100,MEM),WETAB(1100,4,MEM)
C   DIMENSION FX(1100),FZ(1100)
C
C   DIMENSION VX(1100),VXP(1100),VZ(1100),VZP(1100),H(1100),HP(1100),
C   > DH(1100),HT(1100),TH(1100,4),THP(1100,4),DTH(1100,4)
C
C   DIMENSION DLB(400),DCOSXB(400),DCOSZB(400),BFLX(400,MEM),
C   > BFLXP(400,MEM),NBE(400),ISB(400,4),NPB(400)
C
C   DIMENSION DL(400),DCOSX(400),DCOSZ(400)
C   DIMENSION NRSE(400,MEM),IS(400,4,MEM),NPRS(400,MEM)
C   DIMENSION NPN(400,MEM),BB(400,MEM),NPST(400,MEM),DP(400,MEM)
C   DIMENSION PMAT(3,9),PROP(10,9)
C   DIMENSION PROPR(10,9,MEM),RDEC(1100,2),RR(1100)
C   DIMENSION KPR(5000),KDSK(5000)
C   DIMENSION FRATE(10),FLOW(10),TFLOW(10,MEM)
C
C   DIMENSION NSELM(400,MEM),QSRC(400,MEM),QTER(1100,MEM),
C   > QTIM(1100),ELA1(400,MEM),ELA2(400,MEM),TOTA(MEM),
C   > AEL(MEM),NPGW(400,2),BFLXW(400),NPSS(400),
C   > TGMOUN(MEM),TGFLOW(MEM)
C
CHANGE E
COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
COMMON /PARM/ DELT,CHNG,DELMAX,TMAX,DELTO
COMMON /MTL/ NMAT,NMPPM,NCM
COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
CHANGE S DATA STATEMENT BY HATA
C
C   DATA MAXNP,MAXEL,MAXBW,MAXNTI /595,528,29,500/
C   DATA MAXMAT,MXMPPM /3,9/
C   DATA MAXBEL,MAXBNP /199,200/
C   DATA MXRSEL,MXRSNP /99,100/
C   DATA MXSTNP,MAXBCN/40,30/
C
C   DATA MAXNP,MAXEL,MAXBW,MAXNTI /1100,1100,80,5000/
C   DATA MAXMAT,MXMPPM /10,9/
C   DATA MAXBEL,MAXBNP /400,400/
C   DATA MXRSEL,MXRSNP /400,400/

```

```

DATA MXSTNP,MAXBCN/400,400/
C
C   MAXRAM : MAXIMUM NUMBER OF MEMBER IN A DECAY CHAIN
C
DATA MAXRAM,MAXSRC,MAXRED/3,400,1100/
C
CHANGE E
C
CHANGE S DATA STATEMENT BY HATA
C
DATA PMAT/4H      ,4H  KD,4H      ,4H      ,4HRHOB,4H      ,4H      /
C > 4H  AL,4H      ,4H      ,4H  AT,4H      ,4H      L,4HAMB,4HA      ,4H      /
C > 4HPOR,4H      ,4H      ,4HALP,4H      ,4H      ,4H  AM,4H      ,4H      /
C > 4H  TAU,4H      /
C
DATA PMAT/4H      ,4HRHOB,4H      ,4H      ,4H  AL,4H      ,4H      /
C > 4H  AT,4H      ,4H      ,4HPOR,4H      ,4H      ,4HALP,4H      ,4H      /
C > 4H  TAU,4H      ,4H      ,4H  KD,4H      ,4H      L,4HAMB,4HA      ,4H      /
C > 4H  AM,4H      /
C
CHANGE E
C
C ----- INITIATE ARRAYS FOR NODAL POINTS
C
DO 100 NP=1,MAXNP
X(NP)=0.0
Z(NP)=0.0
R(NP)=0.0
C
CHANGE S DIMENSION BY HATA
C
RP(I) ---> RP(I,J)
C
RP(NP)=0.0
C
DO 101 IR=1,MAXRAM
101 RP(NP,IR)=0.0
C
CHANGE E
C
C
VX(NP)=0.0
VXP(NP)=0.0
VZ(NP)=0.0
VZP(NP)=0.0
H(NP)=0.0
HP(NP)=0.0
DH(NP)=0.0
HT(NP)=0.0
FX(NP)=0.0
FZ(NP)=0.0
DO 100 IB=1,MAXBW
100 C(NP,IB)=0.0
C
C ----- INITIATE ARRAYS FOR ELEMENTS
C
DO 150 MP=1,MAXEL

```

```

C
      DO 120 IQ=1,5
120      IE(MP,IQ)=0.
C
      DO 140 IQ=1,4
      TH(MP,IQ)=0.0
      THP(MP,IQ)=0.0
C
CHANGE S DIMENSION BY HATA
C
      WETAB(I,J) ----> WETAB(I,J,K)
C
      WETAB(MP,IQ)=0.0
C
      DO 141 IR=1,MAXRAM
141      WETAB(MP,IQ,IR)=0.0
C
CHANGE E
C
140      DTH(MP,IQ)=0.0
C
150      CONTINUE
C
C
C ----- INITIATE ARRAYS FOR BOUNDARY ELEMENTS
C
      DO 200 MP=1,MAXBEL
      DLB(MP)=0.0
      DCOSXB(MP)=0.0
      DCOSZB(MP)=0.0
      NBE(MP)=0
      DO 200 IQ=1,4
      ISB(MP,IQ)=0
200      CONTINUE
C
C
C ----- INITIATE ARRAYS FOR BOUNDARY NODAL POINTS
C
CHANGE S DIMENSION BY HATA
C
      BFLX(I) ----> BFLX(I,J)
      BFLXP(I) ----> BFLXP(I,J)
C
      DO 250 NP=1,MAXBNP
      BFLX(NP)=0.0
      BFLXP(NP)=0.0
C 250      NPB(NP)=0
C
      DO 250 NP=1,MAXBNP
      DO 251 NR=1,MAXRAM
      BFLX(NP,NR)=0.0
      BFLXP(NP,NR)=0.0
251      CONTINUE
      NPB(NP)=0
250      CONTINUE
C
C

```

```

C ----- INITIATE ARRAYS FOR INFLUX-OUTFLUX BOUNDARY ELEMENTS
C
C      DO 300 MP=1,MXRSEL
C          DL(MP)=0.0
C          DCOSX(MP)=0.0
C          DCOSZ(MP)=0.0
CHANGE S DIMENSION BY HATA
C
C      NRSE(I) ---> NRSE(I,J)
C      IS(I,J) ---> IS(I,J,K)
C
C      NRSE(MP)=0
C      DO 300 IQ=1,4
C          IS(MP,IQ)=0
C 300    CONTINUE
C
C      DO 300 IR=1,MAXRAM
C          NRSE(MP,IR)=0
C      DO 300 IQ=1,4
C          IS(MP,IQ,IR)=0
C 300    CONTINUE
C
CHANGE E
C
C ----- INITIATE ARRAYS FOR INFLUX-OUTFLUX BOUNDARY NODAL POINTS
CHANGE S DIMENSION BY HATA
C
C      NPRS(I) ---> NPRS(I,J)
C
C      DO 350 NP=1,MXRSNP
C 350    NPRS(NP)=0
C
C      DO 350 NP=1,MXRSNP
C      DO 350 IR=1,MAXRAM
C 350    NPRS(NP,IR)=0
C
CHANGE E
C
C ----- INITIATE ARRAYS FOR SURFACCE TERM POINT FLUX
CHANGE S DIMENSION BY HATA
C
C      NPST(I) ---> NPST(I,J)
C      DP(I) ---> DP(I,J)
C
C      DO 500 NP=1,MXSTNP
C          NPST(NP)=0
C 500    DP(NP)=0.0
C
C      DO 500 NP=1,MXSTNP
C      DO 500 IR=1,MAXRAM
C          NPST(NP,IR)=0
C 500    DP(NP,IR)=0.0
C
CHANGE E

```

```

C
C
C ----- INITIATE ARRAYS FOR DIRICHLENT BOUNDARY CONDITIONS
C
CHANGE S DIMENSION BY HATA
C
C      BB(I)  ---> B(I,J)
C      NPN(I) ---> NPN(I,J)
C
C      DO 510 NP=1,MAXBCN
C          BB(NP)=0.0
C 510     NPN(NP)=0
C
C          DO 510 NP=1,MAXBCN
C          DO 510 IR=1,MAXRAM
C              BB(NP,IR)=0.0
C 510     NPN(NP,IR)=0
C
C
CHANGE E
C
C ----- INITIATE ARRAYS FOR MATERIAL PROPERTIES
C
C ADD S DIMENSION BY HATA
C
C      PROPR(I,J,K) --- RADIOACTIVE NUCLIDE PROPERTIES
C
C      DO 650 I=1,MAXMAT
C          DO 610 J=1,MXMPPM
C 610     PROP(I,J)=0.0
C 650     CONTINUE
C
C          DO 650 I=1,MAXMAT
C              DO 610 J=1,MXMPPM
C                  PROP(I,J)=0.0
C              DO 610 K=1,MAXRAM
C                  PROPR(I,J,K)=0.0
C 610     CONTINUE
C 650     CONTINUE
C
C
C ADD E
C
C
C ADD S  INITIATE ARRAY FOR TOTAL OUTFLOW MASS
C
C      DO 710 I=1,MAXRAM
C          TGMOUN(I)=0.0
C          TGFLOW(I)=0.0
C 710 CONTINUE
C
C
C ADD E
C
C ----- INITIATE ARRAYS FOR FLOW THROUGH VARIOUS TYPES OF BOUNDARIES
C
C      DO 700 I=1,10
C          FRATE(I)=0.0
C          FLOW(I)=0.0

```



```

C
CHANGE S DIMENSION BY HATA
C
C 700 TFLOW(I)=0.0
C
      DO 701 NR=1,MAXRAM
        TFLOW(I,NR)=0.0
      701 CONTINUE
      700 CONTINUE

C
C
C ----- INITIATE ARRAYS FOR PRINT AND WRITE ON DISK CONTROL
C
      DO 800 NP=1,MAXNTI
        KPR(NP)=0
      800 KDSK(NP)=0

C
C ADD S DO LOOP BY HATA
C
      DO 810 NQ=1,MAXSRC
        DO 810 JQ=1,MAXRAM
          NSELM(NQ,JQ)=0
      810 QSRC(NQ,JQ)=0.0

C
      DO 820 NRR=1,MAXRED
        QTIM(NRR)=0.0
      DO 820 NRJ=1,MAXRAM
      820 QTER(NRR,NRJ)=0.0

C
C ADD E
C
C
C
C ----- PASS THE PROGRAM TO GM2DXZ
C
C ADD ACTUAL ARGUMENTS OF CALL STATEMENT BY HATA
C
      CALL GM2DXZ( --- ,MAXRAM,MAXSRC,MAXRED,PROPR,RDEC,RR,
                  NSELM,QSRC,QTER,QTIM,ELA1,ELA2,TOTA,AEL,NPGW,BFLXW)

C
      CALL GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,TH,THP
> DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,DCOSX,
> DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLOW,
> PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
> MAXBCN,MAXNTI,MAXMAT,MXMPPM)

C
      CALL GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,TH,THP
> DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,DCOSX,
> DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLOW,
> PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
> MAXBCN,MAXNTI,MAXMAT,MXMPPM,MAXRAM,MAXSRC,MAXRED,PROPR,RDEC,RR,
> NSELM,QSRC,QTER,QTIM,ELA1,ELA2,TOTA,AEL,NPGW,BFLXW,NPSS,TGMOUN,
> TGFLOW)

C
C ADD E
C
      STOP
      END

```

```

SUBROUTINE GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,
> TH,THP,DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,
> DCOSX,DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLO
> ,PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
> MAXBCN,MAXNTI,MAXMAT,MXMPPM,MAXRAM,MAXSRC,MAXRED,PROPR,RDEC,RR,
> NSELM,QSRC,QTER,QTIM,ELA1,ELA2,TOTA,AEL,NPGW,BFLXW,NPSS,TGMOUN,
> TGFLOW)

C
C ADD S DUMMY ARGUMENTS OF SUBROUTINE STATEMENT BY HATA
C
C SUBROUTINE GM2DXZ( ---,MAXRAM,MAXSRC,PROPR,RDEC,RR,
C NSELM,QSRC,QTER,QTIM,ELA1,ELA2,NPGW,BFLXW)
C
C SUBROUTINE GM2DXZ(X,Z,IE,WETAB, C,R,RP,VX,VXP,VZ,VZP,H,HP,DH,HT,
C > TH,THP,DTH,FX,FZ, DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,BFLX,BFLXP, DL,
C > DCOSX,DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,KPR,KDSK,FRATE,FLOW,TFLO
C > ,PMAT,PROP,MAXNP,MAXEL,MAXBW,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,
C > MAXBCN,MAXNTI,MAXMAT,MXMPPM)
C
C ADD E
C
C IMPLICIT REAL*8(A-H,O-Z)
C REAL*4 PMAT,REAL4
C
C CHANGE S DIMENSION BY HATA
C
C DIMENSION TITLE(9)
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),WETAB(MAXEL,4)
C DIMENSION FX(MAXNP),FZ(MAXNP)
CC
C DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP),H(MAXNP),
C > HP(MAXNP),DH(MAXNP),HT(MAXNP),TH(MAXEL,4),THP(MAXEL,4),
C > DTH(MAXEL,4)
CC
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP)
C > BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
CC
C DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
C > IS(MXRSEL,4),NPRS(MXRSNP)
CC
C DIMENSION NPN(MAXBCN),BB(MAXBCN),NPST(MXSTNP),DP(MXSTNP)
CC
C DIMENSION PMAT(3,MXMPPM),PROP(MAXMAT,MXMPPM)
C DIMENSION KPR(MAXNTI),KDSK(MAXNTI)
CC
C DIMENSION FRATE(10),FLOW(10),TFLOW(10)
CC
C DIMENSION TITLE(9)
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP)
C DIMENSION RP(MAXNP,MAXRAM),WETAB(MAXEL,4,MAXRAM)
C DIMENSION FX(MAXNP),FZ(MAXNP)
C
C DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP),H(MAXNP),
C > HP(MAXNP),DH(MAXNP),HT(MAXNP),TH(MAXEL,4),THP(MAXEL,4),
C > DTH(MAXEL,4)
C

```

```

DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL)
DIMENSION BFLX(MAXBNP,MAXRAM),BFLXP(MAXBNP,MAXRAM)
DIMENSION NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
C
DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),
>NRSE(MXRSEL,MAXRAM),IS(MXRSEL,4,MAXRAM),NPRS(MXRSNP,MAXRAM)
C
DIMENSION NPN(MAXBCN,MAXRAM),BB(MAXBCN,MAXRAM)
DIMENSION NPST(MXSTNP,MAXRAM),DP(MXSTNP,MAXRAM)
C
DIMENSION PMAT(3,MXMPPM),PROP(MAXMAT,MXMPPM)
DIMENSION PROPR(MAXMAT,MXMPPM,MAXRAM),RDEC(MAXNP,2),RR(MAXNP)
DIMENSION KPR(MAXNTI),KDSK(MAXNTI)
C
DIMENSION FRATE(10),FLOW(10),TFLOW(10,MAXRAM)
C
DIMENSION NSELM(MAXSRC,MAXRAM),QSRC(MAXSRC,MAXRAM),
> QTER(MAXRED,MAXRAM),QTIM(MAXRED),ELA1(MXSTNP,MAXRAM),
> ELA2(MXSTNP,MAXRAM),TOTA(MAXRAM),AEL(MAXRAM),NPGW(MAXBNP,2)
> BFLXW(MAXBNP),NPSS(MAXBNP),TGMOUN(MAXRAM),TGFLOW(MAXRAM)
C
CDBG BY HATA
C DIMENSION TREL(3)
CDBG
CHANGE E
COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
COMMON /PARM/ DELT,CHNG,DELMAX,TMAX,DELTO
COMMON /MTL/ NMAT,NMPPM,NCM
COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C
C INPUT PROBLEM IDENTIFICATION AND DESCRIPTION
C
10 READ(5,10000)NPROB,(TITLE(I),I=1,9)
IF (NPROB.LE.0) GO TO 250
WRITE(6,10100)NPROB,(TITLE(I),I=1,9)
C
C READ AND PRINT INPUT DATA
C
KOUT=0
KDIG=0
C
C ADD S BY HATA
C
NRED=0
NBRED=0
TGRATE=0.0
C
C ADD E
C
C -----
KSS=1
C -----
C
C ADD ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C

```

```

C      CALL DATAIN( --- ,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,QTIM,NSRC,
C      MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,TOTA,NPGW,NSUR)
C
C      CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
C      > NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,MAXEL,
C      > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXNTI,
C      > MAXDIF,W,ISTOP, 1)
C
C      CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
C      > NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,MAXEL,
C      > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXNTI,
C      > MAXDIF,W,ISTOP, 1,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,QTIM,NSRC,
C      > MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,TOTA,NPGW,NSUR,NPSS,IRES,
C      > TIMR)
CDBG BY HATA
C      WRITE(6,*) ' DEBUG A DATAIN '
C      WRITE(6,*) '   NRAM ',NRAM
CDBG
C      REAL4=DELT
C      KR=LOG10(REAL4)
C
C ADD E
C
C -----
C ----- APPLY DIRICHLET BOUNDARY SPECIFICATIONS TO INITIAL CONDITION
C
C      IF(NBC.EQ.0) GO TO 16
C
C CHANGE S DIMENSION BY HATA
C
C      NPN(I) ----> NPN(I,J)
C      RP(I) ----> RP(I,J)
C      BB(I) ----> BB(I,J)
C
C      DO 14 NPP=1,NBC
C      NP=NPN(NPP)
C      14 RP(NP)=BB(NPP)
C
C      DO 14 IPP=1,NRAM
C      DO 14 NPP=1,NBC
C      NP=NPN(NPP,IPP)
C      RP(NP,IPP)=BB(NPP,IPP)
CDBG BY HATA
C      WRITE(6,*) ' DEBUG IRAM,NP,RP ',IRAM,NP,RP(NP,IPP)
CDBG
C      14 CONTINUE
C
C CHANGE E
C
C      16 CONTINUE
C -----
C      IF (ISTOP.GT.0) GO TO 240
C
C COMPUTE BAND-WIDTH VARIABLES
C
C      IHALFB=MAXDIF
C      IBAND=2*IHALFB+1

```

```

      IHBP=IHALFB+1
      IF (IBAND.GT.MAXBW) GO TO 230
C
C   PREPARE INITIAL VARIABLES
C
C
      TIME=0.
C
C   READ INITIAL VELOCITIES, PRESSURES, AND WATER CONTENTS, IF NECESSARY
C
      DO 20 M=1,NEL
        MTYP=IE(M,5)
        POR=PROP(MTYP,4)
        DO 20 IQ=1,4
          TH(M,IQ)=POR
      20   CONTINUE
C
      60 READ(1) TIMEM,(H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),((TH(M,IQ), M=1,
        > NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
C
      DO 95 NP=1,NNP
        VXP(NP)=VX(NP)
      95   VZP(NP)=VZ(NP)
C
C   ADD S DO LOOP 91 BY HATA
C
      DO 91 IRAM=1,NRAM
        :
        :
      91 CONTINUE
C
CDBG BY HATA
C
      WRITE(6,*) ' DEBUG GM2DXZ '
      WRITE(6,*) ' NRAM ',NRAM
      WRITE(6,*) ' PROPR 1,2,3 ',PROPR(1,1,1),PROPR(1,2,1),
        > PROPR(1,3,1)
C
CDBG
C
C   ADD S BY HATA
C
      DO 293 NP=1,NNP
        RDEC(NP,1)=0.0
        RDEC(NP,2)=0.0
      293 CONTINUE
C
C
C   ADD E
C
      DO 91 IRAM=1,NRAM
C
C   ADD S ACTUAL ARGUMENTS OF CALL STATEMENT BY HATA
C
      CALL AFABTA( --- ,PROPR,IRAM,MAXRAM)
C
      CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
        > MAXMAT,MXMPPM, NEL, W, KSS)

```

```

C
CDBG BY HATA
C WRITE(6,*) ' DEBUG B AFABTA '
C WRITE(6,*) '   NRAM IRAM ',NRAM,IRAM
CDBG
CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
> MAXMAT,MXMPPM, NEL, W, KSS,PROPR,IRAM,MAXRAM)
C
CDBG BY HATA
C WRITE(6,*) ' DEBUG A AFABTA '
C WRITE(6,*) '   NRAM IRAM ',NRAM,IRAM
CDBG
C ADD E
C
CHANGE S WRITE STATEMENT BY HATA
C
C WRITE(6,10600)
C
C WRITE(6,10600) IRAM
C
CHANGE E
C
KLINE=0
DO 100 MP=1,NEL,2
  NJMN=MP
  NJMX=MINO(MP+1,NEL)
CHANGE S WRITE STATEMENT BY HATA
C
C WRITE(6,10700)(NJ,(WETAB(NJ,IQ),IQ=1,4),NJ=NJMN,NJMX)
C
C WRITE(6,10700)(NJ,(WETAB(NJ,IQ,IRAM),IQ=1,4),NJ=NJMN,NJMX)
C
CHANGE E
C
KLINE=KLINE+1
IF(MOD(KLINE,50).EQ.0) WRITE(6,10600)
100 CONTINUE
C
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT AND DO STATEMENT BY HATA
C
C CALL FLUX( --- ,RR, --- ,MAXRAM,IRAM,PROPR)
C
C DO LOOP 101
C
C CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
> MAXMAT,MXMPPM)
C
C DO 111 KK=1,NNP
  RR(KK)= RP(KK,IRAM)
111 CONTINUE
C
C CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RR,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
> MAXMAT,MXMPPM,MAXRAM,IRAM,PROPR)
C
C ADD E
C
KFLOW=-1

```

```

C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
C CALL SFLOW( --- ,RR, --- ,PROPR,MAXRAM,IRAM)
C
C CALL MRFLUX( --- )
C
C CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
C > ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
C > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH
C > KFLOW)
C
C CALL SFLOW(X,Z,IE,WETAB,RR,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
C > ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
C > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH.
C > KFLOW,PROPR,MAXRAM,IRAM)
C
CDBG BY HATA
C WRITE(6,*) ' DEBUG B SRFLUX '
C WRITE(6,*) ' NRAM IRAM ',NRAM,IRAM
CDBG
C CALL SRFLUX(BFLX,BFLXP,NPSS,NPB,TGRATE,TGFLOW,NSUR,IRAM,
C > MAXBNP,MAXRAM)
CDBG BY HATA
C WRITE(6,*) ' DEBUG A SRFLUX '
C WRITE(6,*) ' NRAM IRAM ',NRAM,IRAM
CDBG
C
C ADD E
C
C PRINT INITIAL VARIABLES
C
C DO 99 I=1,9
C FLOW(I)=0.0
C IF(I.EQ.6 .OR. I.EQ.7) FRATE(I)=0.0
C
C CHANGE S DIMENSIN BY HATA
C
C TFLOW(I) ---> TFLOW(I,J)
C
C IF(I.LE.5 .OR. I.GE.8) TFLOW(I)=0.0
C
C IF(I.LE.5 .OR. I.GE.8) TFLOW(I,IRAM)=0.0
C 99 CONTINUE
C
C CHANGE E
C
C KDIAG=-2
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
C CALL PRINTT(RR, --- ,IRAM,MAXRAM,NSUR)
C
C CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG,
C > MAXNP,NNP,IBAND, -1)
C

```

```

CDBG BY HATA
C WRITE(6,*) ' SUB PRINTT 1 '
CDBG
CALL PRINTT(RR,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG,
> MAXNP,NNP,IBAND, -1,TGRATE,TGFLOW,TGMOUN,IRAM,MAXRAM,NSUR,NRAM)
CDBG BY HATA
C WRITE(9,*) ' TIME= ',TIME
CALL TOTC(X,Z,RR,IE,MAXNP,MAXEL,NEL,IRAM)
CDBG
C
C ADD E
C
C
C CHANGE S DIMENSION BY HATA
C
C TFLOW(I) ---> TFLOW(I,J)
C
C TFLOW(6)=0.0
C TFLOW(7)=0.0
C
C TFLOW(6,IRAM)=0.0
C TFLOW(7,IRAM)=0.0
C
C CHANGE E
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
C CALL STORE( --- ,RR, --- ,IRAM)
C
C IF(KSTR.EQ.1 .AND. KDSKO.EQ.1 .AND. KSS.EQ.1) CALL STORE(X,Z,IE,
C > RP,FX,FZ,TITLE, NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C IF(KSTR.EQ.1 .AND. KDSKO.EQ.1 .AND. KSS.EQ.1) CALL STORE(X,Z,IE,
C > RR,FX,FZ,TITLE, NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME,IRAM)
C
C ADD E
C
C 91 CONTINUE
C
C ADD E END OF DO LOOP 91
C
C-----
C IF(KSS.EQ.1) GO TO 109
C
C
C ----- PERFORM STEADY STATE COMPUTATION
C
DO 101 NP=1,NNP
101 HP(NP)=H(NP)
DO 102 M=1,NEL
DO 102 IQ=1,4
102 THP(M,IQ)=TH(M,IQ)
DO 103 NP=1,NNP
103 DH(NP)=(H(NP)-HP(NP))/DELT
DO 104 M=1,NEL
DO 104 IQ=1,4

```



```

104 DTH(M,IQ)=(TH(M,IQ)-THP(M,IQ))/DELT
C
C ADD S DO LOOP 108 BY HATA
C
C     DO 108 I=1,NRAM
C         :
C         :
C     108 CONTINUE
C
C     DO 108 IRAM=1,NRAM
C
C ADD S ACTUAL ARGUMENTS OF CALL STATEMENT BY HATA
C
C     IF(ISW1.EQ.1) CALL QINT( --- )
C
C     CALL ASEMBL( --- ,MAXRAM,MAXSRC,IRAM,PROPR,RDEC,NSELM,QSRC,NSRC
C
C     CALL ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
C > PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
C     IF(ISW1.EQ.1) CALL QINT(QSRC,TIME,QTER,QTIM,TOTA,MAXSRC,MAXRAM,
C > MAXRED,IRAM,NRED,NSRC)
C
C     CALL ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
C > PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM,MAXRAM,MAXSRC,IRAM,PROPR
C > PROPR,RDEC,NSELM,QSRC,NSRC)
C
C ADD E
C
C ADD S ACTUAL ARGUMENTS OF CALL STATEMENT BY HATA
C
C     IF(ISW2.EQ.1) CALL BFLUX( --- )
C
C     CALL BC( --- ,MAXRAM,IRAM)
C
C     CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,DL,DCOS
C > DCOSZ,NRSE,IS,W,MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,MXRSEL)
C
C     IF(ISW2.EQ.1) CALL BFLUX(TIME,DP,ELA1,QTER,QTIM,EBA,AEL,NPST,IRAM
C > NBRED,MXSTNP,MAXRAM,MAXRED)
C
C     CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,DL,DCOS
C > DCOSZ,NRSE,IS,W,MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,MXRSEL,
C > MAXRAM,IRAM)
C
C ADD E
C
C     CALL SOLVE(1,C,R,NNP,IHALFB,MAXNP,MAXBW)
C     CALL SOLVE(2,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C CHANGE S DIMENSION BY HATA
C
C     RP(I) ---> RP(I,J)
C
C     DO 105 NP=1,NNP
C 105 RP(NP)=R(NP)
C

```

```

DO 105 NP=1,NNP
RR(NP)      =R(NP)
105 RP(NP,IRAM)=R(NP)
C
CHANGE E
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL FLUX( --- ,RR, --- ,MAXRAM,IRAM,PROPR)
C
C      CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RP,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
C      > MAXMAT,MXMPPM)
C
C      CALL FLUX(X,Z,IE,WETAB,C,FX,FZ,RR,VX,VZ,PROP,MAXNP,MAXEL,MAXBW,
C      > MAXMAT,MXMPPM,MAXRAM,IRAM,PROPR)
C
C ADD E
C
C      KFLOW=-1
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL SFLOW( --- ,RR, --- ,PROPR,MAXRAM,IRAM)
C
C      CALL SRFLUX( --- )
C
C      CALL SFLOW(X,Z,IE,WETAB,RP,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
C      > ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
C      > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH
C      > KFLOW)
C
C      CALL SFLOW(X,Z,IE,WETAB,RR,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE,
C      > ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXEL,
C      > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT,DH
C      > KFLOW,PROPR,MAXRAM,IRAM)
C
C      CALL SRFLUX(BFLX,BFLXP,NPSS,NPB,TGRATE,TGFLOW,NSUR,IRAM,
C      > MAXBNP,MAXRAM)
C
C ADD E
C
C      DO 107 I=1,9
C      FLOW(I)=0.0
C      IF(I.EQ.6 .OR. I.EQ.7) FRATE(I)=0.0
C
CHANGE S  DIMENSION  BY HATA
C
C      TFLOW(I) ---> TFLOW(I,J)
C
C      IF(I.LE.5 .OR. I.GE.8) TFLOW(I)=0.0
C
C      IF(I.LE.5 .OR. I.GE.8) TFLOW(I,IRAM)=0.0
C
CHANGE E
C
107 CONTINUE
C
C ----- PRINT STEADY STATE VARIABLE

```

```

C
C      KDIAG=-1
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL PRINTT(RR, --- ,IRAM,MAXRAM,NSUR)
C
C      CALL PRINTT(RP,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG,
C > MAXNP,NNP,IBAND, 0)
C
CDBG  BY HATA
C      WRITE(6,*) ' SUB PRINTT 2 '
CDBG
C      CALL PRINTT(RR,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPRO,KOUT,KDIAG,
C > MAXNP,NNP,IBAND, 0,TGRATE,TGFLOW,TGMOUN,IRAM,MAXRAM,NSUR,NRAM)
CDBG  BY HATA
C      WRITE(9,*) ' TIME= ',TIME
C      CALL TOTC(X,Z,RR,IE,MAXNP,MAXEL,NEL,IRAM)
CDBG
C
C ADD E
C
C CHANGE S  DIMENSION  BY HATA
C
C      TFLOW(I) ----> TFLOW(I,J)
C
C      TFLOW(6)=0.0
C      TFLOW(7)=0.0
C
C      TFLOW(6,IRAM)=0.0
C      TFLOW(7,IRAM)=0.0
C
C CHANGE E
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL STORE( --- ,RR, --- ,IRAM)
C
C      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE,
C > RP,FX,FZ,TITLE, NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
C      IF(KSTR.EQ.1 .AND. KDSKO.EQ.1) CALL STORE(X,Z,IE,
C > RR,FX,FZ,TITLE, NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME,IRAM)
C
C ADD E
C
C 108 CONTINUE
C
C ADD E  END OF DO LOOP 108
C
C
C      IF(NTI.EQ.0) GO TO 10
C
C ----- READ TRANSIENT BOUNDARY CONDITIONS
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA

```

```

C
C   CALL DATAIN( --- ,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,QTIM,NSRC,
C               MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,NPGW,NSUR)
C
C   CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
C   > NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,MAXEL,
C   > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXNTI,
C   > MAXDIF,W,ISTOP, 2)
C
C   CALL DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX,DCOSZ,
C   > NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,MAXEL,
C   > MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,MAXNTI,
C   > MAXDIF,W,ISTOP, 2,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,QTIM,NSRC,
C   > MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,TOTA,NPGW,NSUR,NPSS,IRES,
C   > TIMR)
C
C ADD E
C
C       KSS=1
C     109 CONTINUE
C-----
C
C
C PERFORM TRANSIENT-STATE CALCULATION
C
C       TIME=DELT
C       W1=W
C       W2=1.-W
C       KFLOW=1
C       KDIAG=0
CDBG BY HATA
C
C       TIME=TIMR
C
C       DO 219 KK=1,NRAM
C         TREL(KK)=0.0
C 219 CONTINUE
CDBG
C
C READ TIME-DEPENDENT VELOCITIES, AS REQUIRED
C
C   DO 220 ITM=1,NTI
C     DO 110 NP=1,NNP
C       VXP(NP)=VX(NP)
C       VZP(NP)=VZ(NP)
C 110   HP(NP)=H(NP)
C     DO 130 M=1,NEL
C       DO 130 IQ=1,4
C 130   THP(M,IQ)=TH(M,IQ)
C       IF(KVI.NE.2) GO TO 170
C
C     READ(1) TIMEM,(H(NP),NP=1,NNP),(HT(NP),NP=1,NNP),((TH(M,IQ),
C   >   1,NEL),IQ=1,4),(VX(NP),NP=1,NNP),(VZ(NP),NP=1,NNP)
C
C ASSEMBLE COEFFICIENT MATRICES A, B, AND C, AND CONSTRUCT
C LOAD VECTOR R
C
C 170 DO 180 NP=1,NNP

```

```

180      DH(NP)=(H(NP)-HP(NP))/DELT
      DO 200 M=1,NEL
      DO 200 IQ=1,4
200      DTH(M,IQ)=(TH(M,IQ)-THP(M,IQ))/DELT
C
C ADD S DO LOOP 202,203
C
C      DO 202 IRAM=1,NRAM
C          :
C          :
C      202 CONTINUE
C
C      DO 203 NP=1,NNP
C      DO 203 KP=1,2
C      203 RDEC(NP,KP)=0.0
C
C      DO 203 NP=1,NNP
C          RDEC(NP,1)=0.0
C          RDEC(NP,2)=0.0
203 CONTINUE
C
C      DO 202 IRAM=1,NRAM
C
C      IF(KVI.NE.2) GO TO 201
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
C      CALL AFABTA( --- ,PROPR,IRAM,MAXRAM)
C
C      CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
C >          MAXMAT,MXMPPM, NEL, W,KSS)
C
C      CALL AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
C >          MAXMAT,MXMPPM, NEL, W,KSS,PROPR,IRAM,MAXRAM)
C
C ADD E
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT
C
C      IF(ISW1.EQ.1) CALL QINT( --- )
C
C      CALL ASEMBL( --- ,MAXRAM,IRAM,PROPR,RDEC)
C
C      201 CONTINUE
C
C 201 CALL ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
C >          PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
C 201 CONTINUE
C
C      IF(ISW1.EQ.1) CALL QINT(QSRC,TIME,QTER,QTIM,TOTA,MAXSRC,MAXRAM
C >          MAXRED,IRAM,NRED,NSRC)
C
C DBG BY HATA
C      WRITE(9,*) ' B CALL ASEMBL'

```

```

C      WRITE(9,*) ' X , Z '
C      WRITE(9,1112) (X(I),Z(I),I=1,NNP)
C      WRITE(9,*) ' C STIFF MATRIX '
C      WRITE(9,1112) ((C(I,J),J=1,IBAND),I=1,NNP)
C      WRITE(9,*) ' LOAD MATRIX '
C      WRITE(9,1112) (R(I),I=1,NNP)
C      WRITE(9,*) ' VX , VXP '
C      WRITE(9,1112) (VX(I),VXP(I),I=1,NNP)
C      WRITE(9,*) ' VZ , VZP '
C      WRITE(9,1112) (VZ(I),VZP(I),I=1,NNP)
C      WRITE(9,*) ' MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM,MAXRAM,MAXSRC,IRA
C      WRITE(9,*) MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM,MAXRAM,MAXSRC,IRA
CDBG
      CALL ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DTH,
>          PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM,MAXRAM,MAXSRC,IRAM,
>          PROPR,RDEC,NSELM,QSRC,NSRC)
CDBG BY HATA
C      WRITE(9,*) ' A CALL ASEMBL '
C      WRITE(9,1111)
C      WRITE(9,1112) ((C(I,J),J=1,IBAND),I=1,NNP)
C      WRITE(9,1113)
C      WRITE(9,1112) (R(I),I=1,NNP)
C1111  FORMAT(1X,' STIFF MATRIX ')
C1112  FORMAT(1X,6(2X,E10.3))
C1113  FORMAT(1X,' LOAD MATRIX')
CDBG
C
C ADD E
C
C APPLY BOUNDARY CONDITIONS
C
C
C ADD S ACTUAL SRGUMENT OF CALL STATEMENT BY HATA
C
C      IF(ISW2.EQ.1) CALL BFLUX( --- )
C
C      CALL BC( --- ,MAXRAM,IRAM)
C
C      CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
C      >      DL,DCOSX,DCOSZ,NRSE,IS, W, MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
C      >      MXRSEL)
C
C      IF(ISW2.EQ.1) CALL BFLUX(TIME,DP,ELA1,QTER,QTIM,EBA,AEL,NPST,
C      >      IRAM,NBRED,MXSTNP,MAXRAM,MAXRED)
C
C      CALL BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
C      >      DL,DCOSX,DCOSZ,NRSE,IS, W, MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
C      >      MXRSEL,MAXRAM,IRAM)
C
C ADD E
C
C
C TRIANGULARIZE C MATRIX
C
C      CALL SOLVE(1,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C BACK SUBSTITUTE
C

```

```

      CALL SOLVE(2,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
      IF(IMID.EQ.0) GO TO 208
C
CHANGE S  DIMENSION  BY HATA
C
      RP(I)  ---> RP(I,J)
C
      NPN(I) ---> NPN(I,J)
C
      BB(I)  ---> BB(I,J)
C
C
      DO 205 NP=1,NNP
C 205      R(NP)=2.0DO*R(NP)-RP(NP)
C
      DO 206 NPP=1,NBC
C
      NI=NPN(NPP)
C 206      R(NI)=BB(NPP)
C
C
      DO 205 NP=1,NNP
      205      R(NP)=2.0DO*R(NP)-RP(NP,IRAM)
      DO 206 NPP=1,NBC
      NI=NPN(NPP,IRAM)
      206      R(NI)=BB(NPP,IRAM)
C
CHANGE E
C
      208      CONTINUE
C
C  CALCULATE MATERIAL FLUX FX(NP) AND FZ(NP)
C
C
C  ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL FLUX( --- ,MAXRAM,IRAM,PROPR)
C
C      CALL FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ, PROP, MAXNP,MAXEL,
C  >      MAXBW,MAXMAT,MXMPPM)
C
C      CALL FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ, PROP, MAXNP,MAXEL,
C  >      MAXBW,MAXMAT,MXMPPM,MAXRAM,IRAM,PROPR)
C
C  ADD E
C
C  DETERMINE BOUNDARY FLOWS
C
C
C  ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
C      CALL SFLOW( --- ,PROPR,MAXRAM,IRAM)
C
C      CALL SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE
C  >      ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXE
C  >      MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT
C  >      DH, KFLOW)
C
C      CALL SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,NBE
C  >      ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,MAXE
C  >      MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,DELT
C  >      DH, KFLOW,PROPR,MAXRAM,IRAM)

```

```

C
      CALL SRFLUX(BFLX,BFLXP,NPSS,NPB,TGRATE,TGFLOW,NSUR,IRAM,
>              MAXBNP,MAXRAM)
C
      TGMOUN(IRAM)=TGMOUN(IRAM)+TGFLOW(IRAM)
C
C ADD E
C
C PRINT VARIABLES AT CURRENT TIME STEP
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
      CALL PRINTT( --- ,IRAM,MAXRAM,NSUR)
C
      CALL PRINTT(R,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPR(ITM),KOUT,
>      KDIAG,MAXNP,NNP,IBAND, ITM)
C
CDBG BY HATA
C      WRITE(6,*) ' SUB PRINTT 3 '
C      TREL(IRAM)=TREL(IRAM)+TGRATE
CDBG
      CALL PRINTT(R,FX,FZ,FRATE,FLOW,TFLOW,TIME,DELT,KPR(ITM),KOUT,
>KDIAG,MAXNP,NNP,IBAND, ITM,TGRATE,TGFLOW,TGMOUN,IRAM,MAXRAM,NSUR
>NRAM)
CDBG BY HATA
C      WRITE(9,*) ' TIME= ',TIME
      CALL TOTC(X,Z,R,IE,MAXNP,MAXEL,NEL,IRAM)
CDBG
C
C ADD E
C
C ADD S ACTUAL ARGUMENT OF CALL STATEMENT BY HATA
C
      CALL STORE( --- ,IRAM)
C
      IF(KSTR.EQ.1 .AND. KDSK(ITM).EQ.1) CALL STORE(X,Z,IE,R,FX,FZ,
C      1 TITLE,NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME)
C
      IF(KSTR.EQ.1 .AND. KDSK(ITM).EQ.1) CALL STORE(X,Z,IE,R,FX,FZ,
C      1 TITLE,NPROB,NNP,NEL,NTI,MAXNP,MAXEL,TIME,IRAM)
C
C ADD E
C
C MOVE S BY HATA
C
      MOVE DO LOOP 210 IN DO LOOP 202 AND CHANGE DIMENSION RP
C
      DO 210 NP=1,NNP
210      RP(NP,IRAM)=R(NP)
C
C MOVE E
C
      202 CONTINUE
C
C ADD E DO LOOP 202

```



```

C
C
C   PREPARE FOR NEXT TIME STEP
C
C       IF (KSS.EQ.0) GO TO 10
C       IF (TIME.GE.TMAX) GO TO 10
C
C   ADD S   BY HATA
C
C       REAL4=TIME
C       KN=LOG10(REAL4)
C       IF(KR.NE.KN) THEN
C         DELT=DELT*10
C         KR=KN
C       ENDIF
C
C   ADD E
C
C       DELT=DELT*(1.+CHNG)
C       DELT=DMIN1(DELT,DELMAX)
C       TIME=TIME+DELT
C   MOVE S   BY HATA
C
C       DO 210 NP=1,NNP
C 210      RP(NP)=R(NP)
C
C   MOVE E
C 220      CONTINUE
C         GO TO 10
C 230      WRITE(6,10400)IBAND, MAXBW
C 240      WRITE(6,10500)ISTOP
C 250      CONTINUE
C
C   ADD S   WRITE RESTART FILE   BY HATA
C
C       WRITE(4) TIME,((RP(NP,IRAM),NP=1,NNP),IRAM=1,NRAM)
C
C   ADD E
C
C   CDBG   BY HATA
C         WRITE(9,*) ' TREL ',TREL
C         WRITE(9,*) ' TGMOUN ',TGMOUN
C   CDBG
C         RETURN
C
C 10000   FORMAT(I5,9A8)
C 10100   FORMAT(/8H1PROBLEM,I5,3H.. ,9A8/)
C 10200   FORMAT(8F10.0)
C 10300   FORMAT(////35H1ERROR IN VELOCITY INPUT AT ELEMENT, I5//)
C 10400   FORMAT(////12H BANDWIDTH =,I4,25H EXCEEDS MAX. ALLOWABLE =,I4//)
C 10500   FORMAT(1H0,4X,'ISTOP = ',I5)
C
C   CHANGE S   FORMAT STATEMENT   BY HATA
C
C       10600   FORMAT(1H1,I2,49H1TH RADIOACTIVE NUCLIDE OF CONSIDERING
C              DECAY CHAIN//1H , 'TABLE OF --- )
C
C 10600   FORMAT(1H1,'TABLE OF WEIGHTING FACTORS OF EVERY ELEMENTS'//)

```

```
C
10600 FORMAT(1H1,I2,49HTH RADIOACTIVE NUCLIDE OF CONSIDERING DECAY CHA
> //1H , 'TABLE OF WEIGHTING FACTORS OF EVERY ELEMENTS'//)
C
CHANGE E
C
10700 FORMAT(1H,1P ,I5,4E12.4,5X,I5,4E12.4)
END
```

```

SUBROUTINE DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX
> DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,
> MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
> MAXNTI,MAXDIF,W,ISTOP, IPASS,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,
> QTIM,NSRC,MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,TOTA,NPGW,NSUR
> NPSS,IRES,TIMR)
C
C ADD S DUMMY ARGUMENTS OF SUBROUTINE STATEMENT BY HATA
C
C SUBROUTINE DATAIN( --- ,MAXRAM,NRAM,PROPR,NSELM,QSRC,QTER,QTIM,
C NSRC,MAXSRC,MAXRED,ISW1,ISW2,AEL,ELA1,ELA2,TOTA,NPGW,NSUR)
C
C SUBROUTINE DATAIN(X,Z,IE,DLB,DCOSXB,DCOSZB,NBE,ISB,NPB,DL,DCOSX
C > DCOSZ,NRSE,IS,NPRS,NPN,BB,NPST,DP,RP,PMAT,PROP,KPR,KDSK,MAXNP,
C > MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
C > MAXNTI,MAXDIF,W,ISTOP, IPASS)
C
C ADD E
C
C
C FUNCTION OF SUBROUTINE--TO READ, PRINT, AND CHECK VARIABLES
C PERTAINING TO SIMULATION TIME, GEOMETRY OF THE SYSTEM, BOUNDARY-
C INITIAL CONDITIONS, AND PROPERTIES OF BOTH THE MATERIAL BEING
C TRANSPORTED AND THE POROUS MEDIA.
C
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*4 PMAT
C
C CHANGE S DIMENSION BY HATA
C
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
CC
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL)
C > ISB(MAXBEL,4),NPB(MAXBNP)
CC
C DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL),
C > IS(MXRSEL,4),NPRS(MXRSNP)
CC
C DIMENSION NPN(MAXBCN),BB(MAXBCN),DP(MXSTNP),NPST(MXSTNP)
C DIMENSION RP(MAXNP),PMAT(3,MXMPPM),PROP(MAXMAT,MXMPPM)
C DIMENSION KPR(MAXNTI),KDSK(MAXNTI)
C DIMENSION TITLEM(9)
CC
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5)
C
C DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),NBE(MAXBEL)
C > ISB(MAXBEL,4),NPB(MAXBNP)
C
C DIMENSION DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),
C > NRSE(MXRSEL,MAXRAM),IS(MXRSEL,4,MAXRAM),NPRS(MXRSNP,MAXRAM)
C
C DIMENSION NPN(MAXBCN,MAXRAM),BB(MAXBCN,MAXRAM)
C DIMENSION DP(MXSTNP,MAXRAM),NPST(MXSTNP,MAXRAM)
C DIMENSION RP(MAXNP,MAXRAM),PMAT(3,MXMPPM),PROP(MAXMAT,MXMPPM)

```

```

DIMENSION PROPR(MAXMAT, MXMPPM, MAXRAM)
DIMENSION KPR(MAXNTI), KDSK(MAXNTI)
DIMENSION TITLEM(9)
C
DIMENSION NSELM(MAXSRC, MAXRAM), QSRC(MAXSRC, MAXRAM),
> QTER(MAXRED, MAXRAM), QTIM(MAXRED), ELA1(MXSTNP, MAXRAM),
> ELA2(MXSTNP, MAXRAM), TOTA(MAXRAM), AEL(MAXRAM), NPGW(MAXBNP, 2)
> NPSS(MAXBNP)
CHARACTER*10 INITL, DIRIC, NEUMN, CAUCH, SOURC, GSURF, ICHA
DIMENSION XQ(4), ZQ(4)
C
CHANGE E
C
COMMON /GEOM/ SNFE, CSFE, NNP, NEL, IBAND
COMMON /BDY/ NBEL, NBN, NRSEL, NRSN, NBC, NSTN(5), NST
COMMON /CONTRL/ NTI, NSTR, KSTR, KPRO, KDSKO, KSS, KVI
COMMON /PARM/ DELT, CHNG, DELMAX, TMAX, DELTO
COMMON /MTL/ NMAT, NMPPM, NCM
COMMON /WET/ APHA1, APHA2, BETA1, BETA2, IWET, ILUMP, IMID
C
C ADD COMMON AND DATA STATEMENT BY HATA
C
COMMON /STERM/NODR
C
DATA INITL, DIRIC, NEUMN, CAUCH, SOURC, GSURF/
> 'INITIAL ', 'DIRICHLET ', 'NEUMANN ',
> 'CAUCHY ', 'SOURCE ', 'SURFACE '/
C
C ADD E
C
ISTOP=0
C
C ADD S ASSIGNMENT STATEMENT BY HATA
C
NODR=1
C
C ADD E
C
-----
IF(KSS.EQ.0) GO TO 445
-----
C
C ADD S INPUT LIST OF READ STATEMENT BY HATA
C
READ(5, 10900) --- , NRAM
C
NRAM : NUMBER OF MEMBER IN CONSIDERING DECAY CHAIN.
C
READ(5, 10900) NNP, NEL, NMAT, NCM, NTI, NBC, NST, NRSEL, KVI, KSTR, KSS, NMF
C 1 , IWET, ILUMP, IMID
C
READ(5, 10900) NNP, NEL, NMAT, NCM, NTI, NBC, NST, NRSEL, NSUR, NSRC, NRAM,
C 1 KVI, KSTR, KSS, NMPPM, IWET
C
READ(5, 10920) ILUMP, IMID, ISW1, ISW2, IRES
C
C ADD E
C

```

```

READ(5,11000)DELT,CHNG,DELMAX,TMAX,W
DELTO=DELT
READ(5,11100) KPRO,(KPR(ITM),ITM=1,NTI)
READ(5,11100)KDSKO,(KDSK(ITM),ITM=1,NTI)
IF(TMAX.LE.0.0) TMAX=1.0E38
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10000) ---
C WRITE(6,10010) ---
C
C WRITE(6,10000)NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,KVI,KSTR,KSS,
C > NMPPM,IWET,ILUMP,IMID
C
C WRITE(6,10000)NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,NSUR,NSRC,NRAM,
C > KVI,KSTR,KSS,NMPPM,IWET
C WRITE(6,10010)ILUMP,IMID,ISW1,ISW2,IRES
CDBG
C WRITE(6,*)' NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,NSRC,NRAM '
C WRITE(6,*)NNP,NEL,NMAT,NCM,NTI,NBC,NST,NRSEL,NSRC,NRAM
C WRITE(6,*)' KVI,KSTR,KSS,NMPPM,IWET,ILUMP,IMID,ISW1,ISW2 '
C WRITE(6,*) KVI,KSTR,KSS,NMPPM,IWET,ILUMP,IMID,ISW1,ISW2
CDBG
C
C ADD E
C
C WRITE(6,10001)DELT,CHNG,DELMAX,TMAX,W
C WRITE(6,10100)
C WRITE(6,11200)KPRO,(KPR(ITM),ITM=1,NTI)
C WRITE(6,10100)
C WRITE(6,11200)KDSKO,(KDSK(ITM),ITM=1,NTI)
C
C
C READ AND PRINT MATERIAL PROPERTIES
C
C 100 IF (NMPPM.LE.0) GO TO 120
C
C CHANGE S WRITE STATEMENT BY HATA
C
C NMPPM ---> 6
C
C WRITE(6,10200)((PMAT(I,J),I=1,3),J=1,NMPPM)
C
C WRITE(6,10200)((PMAT(I,J),I=1,3),J=1,6)
C
C CHANGE E
C
C CHANGE S READ AND WRITE STATEMENT BY HATA
C
C PROP : PROPERTY OF POROUS MEDIUM
C PROPR : PROPERTY OF RADIOACTIVE NUCLIDE
C
C DO 110 I=1,NMAT
C READ(5,11000)(PROP(I,J),J=1,NMPPM)
C 110 WRITE(6,11300)I,(PROP(I,J),J=1,NMPPM)
C
C DO 110 I=1,NMAT
C READ(5,11000)(PROP(I,J),J=1,6)

```

```

        WRITE(6,11300)I,(PROP(I,J),J=1,6)
        DO 111 I1=1,NRAM
C ADD S WRITE STATEMENT BY HATA
C
        WRITE(6,10201) ((PMAT(K,J),K=1,3),J=7,9)
C
C ADD E
C
        READ(5,11010) (PROPR(I,J,I1),J=1,3)
        WRITE(6,11310)I,I1,(PROPR(I,J,I1),J=1,3)
111 CONTINUE
110 CONTINUE
C
CHANGE E
C
120 CONTINUE
C
C READ NODAL-POINT AND ELEMENT DATA FROM AUXILIARY STORAGE AND PRINT,
C IF NECESSARY
C
CDBG
C WRITE(6,*) ' DATA FROM FEMWATER '
CDBG
REWIND 1
READ(1) (TITLEM(I),I=1,9),NPROBM,NNP,NEL,NBN,NBEL,NTIM
C
CHANGE S READ STATEMENT BY HATA
C
C READ(1) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,IQ),M=1,NEL), I
C > 1,4),(DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),(DCOSZB(M),M=1,NBEL
C > (NBE(M),M=1,NBEL),((ISB(M,IQ),M=1,NBEL),IQ=1,4),(NPB(NP),NP=1,
C > NBN)
C
C READ(1) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,IQ),M=1,NEL), I
C > 1,5),(DLB(M),M=1,NBEL),(DCOSXB(M),M=1,NBEL),(DCOSZB(M),M=1,NBEL
C > (NBE(M),M=1,NBEL),((ISB(M,IQ),M=1,NBEL),IQ=1,4),(NPB(NP),NP=1,
C > NBN)
C
CHANGE E
C
WRITE(6,10300)
KLINE=-1
DO 130 NP=1,NNP,3
    NJMN=NP
    NJMX=MINO(NP+2,NNP)
    KLINE=KLINE+1
    IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10300)
130 WRITE(6,11500)(NJ,X(NJ),Z(NJ),NJ=NJMN,NJMX)
WRITE(6,10400)
INPTAB=4
KLINE=-1
MAXDIF=0
DO 150 M=1,NEL
C
CHANGE S BY HATA
C
C IE(M,5)=1
C

```

```

      IE(M,5)=IE(M,5)
C
CHANGE E
      MND=0
      DO 140 IQ=1,3
        IQ1 = IQ + 1
        DO 140 JQ=IQ1,4
          ND = IABS(IE(M,IQ)-IE(M,JQ))
          MND = MAXO(ND,MND)
140      MAXDIF = MAXO(ND,MAXDIF)
      KLINE=KLINE+1
      IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10400)
      WRITE(6,11600)M,(IE(M,I),I=1,5),MND
150      CONTINUE
C
C
C  MODIFY MATERIAL TYPES FOR SELECTED ELEMENTS, IF NECESSARY
C
      IF (NCM.LE.0) GO TO 370
      WRITE(6,10500)
      L=0
340  READ(5,10900)MI,MTYP,MK,MINC
      IE(M,5) = MTYP
      WRITE(6,11700)MI,IE(MI,5)
      L = L + 1
      IF (MK.LE.MI) GO TO 360
      IF (MINC.LE.0) MINC = 1
      MI = MI + MINC
      DO 350 MJ=MI,MK,MINC
        IE(MJ,5) = MTYP
        WRITE(6,11700)MJ,IE(MJ,5)
350  L = L + 1
360  IF (L.LT.NCM) GO TO 340
C
C  CHECK MATERIAL TYPES FOR EACH ELEMENT
C
370  DO 380 M=1,NEL
      MTYP=IE(M,5)
      IF (MTYP.GT.0.AND.MTYP.LE.NMAT) GO TO 380
      WRITE(6,14100)M
      ISTOP=ISTOP+1
380  CONTINUE
C
C  READ INITIAL CONDITIONS
C
CHANGE S  GO TO STATEMENT AND ADD READ STATEMENT  BY HATA
C
C  GO TO 390  ---> GO TO 391
C
C  READ(5,10910) NR
C
C  NR : RADIOACTIVE NUCLIDE NUMBER
C
C  IF (ISTOP.EQ.0) GO TO 390
C  WRITE(6,13600)ISTOP
C 392  READ(5,11330) ICHA
C  IF(ICHA.EQ.INITL) GO TO 391

```

```

C      ISTOP=ISTOP+1
C      WRITE(6,13600)ISTOP
C      GO TO 830
C 390 NI=0
C
CDBG
C      WRITE(6,*) ' SUB DATAIN 392 '
CDBG
      IF (ISTOP.EQ.0) GO TO 392
      WRITE(6,13600)ISTOP
392 READ(5,11330) ICHA
CDBG
C      WRITE(6,*) ' INITL , ICHA '
C      WRITE(6,11330) INITL,ICHA
CDBG
      IF(ICHA.EQ.INITL) GO TO 391
      ISTOP=ISTOP+1
      WRITE(6,13600)ISTOP
      GO TO 830
391 READ(5,10910) NR
CDBG
C      WRITE(6,*) ' DEBUG '
C      WRITE(6,*) ' NR =',NR
CDBG
      NI=0
C
CHANGE E
C
      NJ=0
400 IF (NJ.EQ.NNP) GO TO 440
C
CHANGE S DIMENSION BY HATA
C
C      RP(I) ---> RP(I,J)
C
C      READ(5,11800)NJ,RP(NJ)
C
C      READ(5,11800)NJ,RP(NJ,NR)
CDBG
C      WRITE(6,*) ' DEBUG '
C      WRITE(6,*) ' NJ RP(NJ,NR) ',NJ,RP(NJ,NR)
CDBG
C
CHANGE E
C
410 NI=NI+1
CDBG
C      WRITE(6,*) ' DEBUG '
C      WRITE(6,*) ' NI= ',NI,' NJ=',NJ
CDBG
      IF (NI.GT.1) GO TO 420
      IF (NJ.EQ.1) GO TO 420
      WRITE(6,13500)NJ
      ISTOP=ISTOP+1
      GO TO 820
420 IF (NJ.EQ.NI) GO TO 400
      IF (NJ.GT.NI) GO TO 430
      WRITE(6,13500)NJ

```



```

        ISTOP=ISTOP+1
        GO TO 820
C
CHANGE S DIMENSION BY HATA
C
C      RP(I) ---> RP(I,J)
C
C 430 RP(NI)=RP(NI-1)
C
C      430 RP(NI,NR)=RP(NI-1,NR)
CDBG
C      WRITE(6,*) ' DEBUG '
C      WRITE(6,*) ' RP(NI,NJ) ',RP(NI,NJ)
CDBG
C
CHANGE E
C
C      GO TO 410
C
C      440 CONTINUE
C
C ADD S IF STATEMENT BY HATA
C
C      IF(NR.EQ.NRAM) GO TO 445
C      GO TO 391
C
C ADD E
C
C -----
C      445 CONTINUE
C
C ADD S READ FROM RESTART FILE BY HATA
C
C      TIMR=0.0
C      IF(IRES.EQ.1) READ(8) TIMR,((RP(I,J),I=1,NNP),J=1,NRAM)
C
C ADD E
C
C      IF(IPASS.GT.1) READ(5,10900)NBC,NST,NRSEL
C      INPTAB=INPTAB+1
C
CHANGE S IF STATEMENT BY HATA
C
C      IF(KSS.EQ.0.AND.IPASS.EQ.1) WRITE(6,10510)INPTAB,NBC,NST,NRSEL
C      IF(KSS.EQ.0.AND.IPASS.EQ.2) WRITE(6,10520)INPTAB,NBC,NST,NRSEL
C      IF(KSS.EQ.1) WRITE(6,10520) INPTAB,NBC,NST,NRSEL
C
C      IF(KSS.EQ.0.AND.IPASS.EQ.1)WRITE(6,10510)INPTAB,NBC,NST,NRSEL,NS
C      IF(KSS.EQ.0.AND.IPASS.EQ.2)WRITE(6,10520)INPTAB,NBC,NST,NRSEL,NS
C      IF(KSS.EQ.1) WRITE(6,10520)INPTAB,NBC,NST,NRSEL,NSRC
C
CHANGE E
C
C -----
C
CHANGE S DIMENSION BY HATA
C
C      DP(I) ---> DP(I,J)

```

```

C
C      DO 450 NP=1,MXSTNP
C 450   DP(NP)=0.
C
C      DO 450 IRAM=1,NRAM
C      DO 450 NP=1,MXSTNP
C 450   DP(NP,IRAM)=0.
C
CHANGE E
C
C      IF (NBC.EQ.0) GO TO 550
C
C READ CONSTANT-CONCENTRATION DIRICHLET CONDITIONS BB(NPP) TO BE
C APPLIED AT NODES NPN(NPP)
C
C ADD S READ STATEMENT BY HATA
C
C      READ(5,10910) NR
C
C      NR : RADIOACTIVE NUCLIDE NUMBER
C
C      READ(5,11330) ICHA
C
C      IF(ICA.EQ.DIRIC) GO TO 461
C      ISTOP=ISTOP+1
C      WRITE(6,13600) ISTOP
C      GO TO 830
C
C 461 READ(5,10910) NR
C
C ADD E
C
C      INPTAB=INPTAB+1
C      NPP=0
C 460 IF (NPP.EQ.NBC) GO TO 520
C      IF (NPP.LT.NBC) GO TO 470
C      WRITE(6,12900)NBC
C      ISTOP=ISTOP+1
C      GO TO 520
C 470 READ(5,11900)NI,NPINC,BBI
C      IF (NPINC.GT.0) GO TO 490
C 480 NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
C      NPN(I) ---> NPN(I,J)
C      BB(I) ---> BB(I,J)
C
C      NPN(NPP)=NI
C      BB(NPP)=BBI
C
C      NPN(NPP,NR)=NI
C      BB(NPP,NR)=BBI
C
CHANGE E
C
C      GO TO 460
C 490 IF (NPP.GT.0) GO TO 500

```

```

      ISTOP=ISTOP+1
      WRITE(6,13900)
C
CHANGE S DIMENSION BY HATA
C
C      NPP(I) ----> NPP(I,J)
C      BB(I) ----> BB(I,J)
C
C 500 NJ=NPN(NPP)+NPINC
C      BBJ=BB(NPP)
C
C 500 NJ=NPN(NPP,NR)+NPINC
C      BBJ=BB(NPP,NR)
C
CHANGE E
C
      NK=NI-1
C
CHANGE S DIMENSION BY HATA
C
C      NPN(I) ----> NPN(I,J)
C      BB(I) ----> BB(I,J)
C
C      DO 510 NP=NJ,NK,NPINC
C          NPP=NPP+1
C          NPN(NPP)=NP
C 510      BBJ=BB(NPP)
C
C          DO 510 NP=NJ,NK,NPINC
C              NPP=NPP+1
C              NPN(NPP,NR)=NP
C 510      BBJ=BB(NPP,NR)
C
CHANGE E
C
      GO TO 480
      520 CONTINUE
C
C ADD S WRITE STATEMENT BY HATA
C
      WRITE(6,10990) NR
C
C ADD E
C
      IF(KSS.EQ.0 .AND. IPASS.EQ.1) WRITE(6,10610)INPTAB
      IF(KSS.EQ.0 .AND. IPASS.EQ.2) WRITE(6,10620)INPTAB
      IF(KSS.EQ.1) WRITE(6,10620)INPTAB
C
CHANGE S DIMENSION BY HATA
C
C      NPN(I) ----> NPN(I,J)
C      BB(I) ----> BB(I,J)
C
C      DO 530 NPP=1,NBC
C 530      WRITE(6,12000)NPN(NPP),BB(NPP)
C
C          DO 530 NPP=1,NBC
C 530      WRITE(6,12000)NPN(NPP,NR),BB(NPP,NR)

```

```

C
CHANGE E
C
C ADD S IF STATEMENT BY HATA
C
      IF(NR.EQ.NRAM) GO TO 550
      GO TO 461
C
C ADD E
C
      550 IF (NST.LE.0) GO TO 650
C
C READ SURFACE-TERM FLUXES EI AND EJ TO BE APPLIED AT BOUNDARY
C NODES NI AND NJ, RESPECTIVELY
C
C ADD S READ STATEMENT BY HATA
C
      READ(5,11330) ICHA
      READ(5,10910) NR
C
      AEL = 0.0
C
      READ(5,11330) ICHA
C
      IF(ICHA.EQ.CAUCH) GO TO 561
      ISTOP=ISTOP+1
      WRITE(6,13600) ISTOP
      GO TO 830
C
      561 READ(5,10910) NR
C
      AEL(NR)=0.0
C
C ADD E
C
      INPTAB=INPTAB+1
      NPP=0
      MP=0
C
C ADD S WRITE STATEMENT BY HATA
C
      WRITE(6,10990) NR
C
C ADD E
C
      IF(KSS.EQ.0 .AND. IPASS.EQ.1) WRITE(6,10710)INPTAB
      IF(KSS.EQ.0 .AND. IPASS.EQ.2) WRITE(6,10720)INPTAB
      IF(KSS.EQ.1) WRITE(6,10720)INPTAB
      560 IF (MP.EQ.NST) GO TO 610
      READ(5,12100)NI,NJ,KINC,EI,EJ
      IF (KINC.GT.0) GO TO 580
      570 MP=MP+1
      DX=X(NI)-X(NJ)
      DZ=Z(NI)-Z(NJ)
      EL= SQRT(DX*DX+DZ*DZ)
C
C ADD S ASSIGN STATEMENT NY HATA
C

```

```

C     ELA1(MP)=EL
C
C     AEL=AEL+EL
C
C     ELA1(MP,NR)=EL
C     AEL(NR)=AEL(NR)+EL
C
C ADD E
C
C     IF(MP.GT.1) GO TO 571
C     NPP=NPP+1
C
C CHANGE S DIMENSION BY HATA
C
C     NPST(I) ----> NPST(I,J)
C
C     NPST(NPP)=NI
C
C     NPST(NPP,NR)=NI
C
C CHANGE E
C
C     NII=NPP
C     NPP=NPP+1
C
C CHANGE S DIMENSION BY HATA
C
C     NPST(I) ----> NPST(I,J)
C
C     NPST(NPP)=NJ
C
C     NPST(NPP,NR)=NJ
C
C CHANGE E
C
C     NJJ=NPP
C     GO TO 578
C     571 DO 572 I=1,NPP
C
C CHANGE S DIMENSION BY HATA
C
C     NPST(I) ----> NPST(I,J)
C
C     IJ=NPST(I)
C
C     IJ=NPST(I,NR)
C
C CHANGE E
C
C     IF(IJ.EQ.NI) GO TO 573
C     572 CONTINUE
C     NPP=NPP+1
C
C CHANGE S DIMENSION BY HATA
C
C     NPST(I) ----> NPST(I,J)
C
C     NPST(NPP)=NI

```

```

C
      NPST(NPP, NR)=NI
C
CHANGE E
C
      NII=NPP
      GO TO 574
573 NII=I
574 DO 575 J=1, NPP
C
CHANGE S DIMENSION BY HATA
C
      NPST(I) ----> NPST(I, J)
C
      IJ=NPST(J)
C
      IJ=NPST(J, NR)
C
CHANGE E
C
      IF(IJ.EQ.NJ) GO TO 576
575 CONTINUE
      NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
      NPST(I) ----> NPST(I, J)
C
      NPST(NPP)=NJ
C
      NPST(NPP, NR)=NJ
C
CHANGE E
C
      NJJ=NPP
      GO TO 578
576 NJJ=J
C
CHANGE S DIMENSION BY HATA
C
      DP(I) ----> DP(I, J)
C
578 DP(NII)=DP(NII)+EI*EL/3.0+EJ*EL/6.0
C
      DP(NJJ)=DP(NJJ)+EI*EL/6.0+EJ*EL/3.0
C
578 DP(NII, NR)=DP(NII, NR)+EI*EL/3.0+EJ*EL/6.0
      DP(NJJ, NR)=DP(NJJ, NR)+EI*EL/6.0+EJ*EL/3.0
C
CHANGE E
C
      EK=EJ
      WRITE(6, 12200) NI, NJ, EI, EJ
      GO TO 560
580 IF (MP.GT.0) GO TO 590
      ISTOP=ISTOP+1
      WRITE(6, 14000)
590 NPINC=IABS(NJ-NI)
C

```

```

CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C NPMIN=MAXO(NPST(NPP),NPST(NPP-1))
C
C NPMIN=MAXO(NPST(NPP,NR),NPST(NPP-1,NR))
C
CHANGE
C
C NPMAX=MINO(NI,NJ)-1
C DO 600 NK=NPMIN,NPMAX,NPINC
C NL=NK+NPINC
C WRITE(6,12200)NK,NL,EK,EK
C MP=MP+1
C DX=X(NK)-X(NL)
C DZ=Z(NK)-Z(NL)
C EL= SQRT(DX*DX+DZ*DZ)
C
C ADD S ASSIGN STATEMENT BY HATA
C
C ELA1(MP)=EL
C
C AEL=AEL+EL
C
C ELA1(MP,NR)=EL
C AEL(NR)=AEL(NR)+EL
C
C ADD E
C
C IF(MP.GT.1) GO TO 591
C NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C NPST(NPP)=NK
C
C NPST(NPP,NR)=NK
C
CHANGE E
C
C NKK=NPP
C NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C NPST(NPP)=NL
C
C NPST(NPP,NR)=NL
C
CHANGE E
C
C NLL=NPP
C GO TO 598

```

```

591 DO 592 K=1,NPP
C
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C KL=NPST(K)
C
C KL=NPST(K,NR)
C
CHANGE E
C
IF(KL.EQ.NK) GO TO 593
592 CONTINUE
NPP=NPP+1
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C NPST(NPP)=NK
C
C NPST(NPP,NR)=NK
C
CHANGE E
C
NKK=NPP
GO TO 594
593 NKK=K
594 DO 595 L=1,NPP
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C KL=NPST(L)
C
C KL=NPST(L,NR)
C
CHANGE E
C
IF(KL.EQ.NL) GO TO 596
595 CONTINUE
NPP=NPP+1
CHANGE S DIMENSION BY HATA
C
C NPST(I) ----> NPST(I,J)
C
C NPST(NPP)=NL
C
C NPST(NPP,NR)=NL
C
CHANGE E
C
NLL=NPP
GO TO 598
596 NLL=L
C
CHANGE S DIMENSION BY HATA
C

```



```

C      DP(I)  ---> DP(I,J)
C
C 598      DP(NKK)=DP(NKK)+EK*EL/2.0
C          DP(NLL)=DP(NLL)+EK*EL/2.0
C
C 598      DP(NKK,NR)=DP(NKK,NR)+EK*EL/2.0
C          DP(NLL,NR)=DP(NLL,NR)+EK*EL/2.0
C
C CHANGE E
C
C 600      CONTINUE
C
C CHANGE S  ARITHMETIC ASSIGNMENT STATEMENT  BY HATA
C
C      NSTN  ---> NSTN(I)
C
C 610 NSTN=NPP
C
C      GO TO 560
C
C 610 NSTN(NR)=NPP
C
C CHANGE E
C
C ADD S  IF STATEMENT  BY HATA
C
CDBG  BY HATA
C      WRITE(9,*) ' DEBUG SUB DATAIN '
C      WRITE(9,*) '  ELA1 '
C      WRITE(9,1221) (ELA1(KK,NR),KK=1,NST)
C      WRITE(9,*) '  AEL(NR) ', AEL(NR)
C      WRITE(9,*) '  DP '
C      WRITE(9,*) (DP(J,NR),J=1,NST)
C1221  FORMAT(4(3X,E10.3))
CDBG
C      IF(NR.EQ.NRAM) GO TO 650
C      GO TO 561
C
C ADD E
C
C READ NUMBERS OF ELEMENTS AND SIDES TO WHICH OPEN BOUNDARU CONDITION
C ARE TO BE APPLIED
C
C 650 IF(NRSEL .LE.0) GO TO 850
C
C ADD S  READ  STATEMENT  BY HATA
C
C      READ(5,11330) ICHA
C      READ(5,10910) NR
C
C      READ(5,11330) ICHA
C
C      IF(ICHA.EQ.NEUMN) GO TO 661
C      ISTOP=ISTOP+1
C      WRITE(6,13600) ISTOP
C      GO TO 830
C
C 661 READ(5,10910) NR

```

```

C
C ADD E
C
      INPTAB=INPTAB+1
      NPP=0
      MPI=0
660 IF(MPI.EQ.NRSEL) GO TO 710
      READ(5,10900)MI,IS1,IS2,KINC
      IF (KINC.GT.0) GO TO 680
670 MPI=MPI+1
C
CHANGE S DIMENSION BY HATA
C
      NRSE(I) ----> NRSE(I,J)
      IS(I,J) ----> IS(I,J,K)
C
      NRSE(MPI)=MI
      IS(MPI,1)=IS1
      IS(MPI,2)=IS2
C
      NRSE(MPI,NR)=MI
      IS(MPI,1,NR)=IS1
      IS(MPI,2,NR)=IS2
C
CHANGE E
C
      IF(MPI.GT.1) GO TO 671
      NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
      NPRS(I) ----> NPRS(I,J)
C
      NPRS(NPP)=IS1
C
      NPRS(NPP,NR)=IS1
C
CHANGE E
C
      NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
      NPRS(I) ----> NPRS(I,J)
C
      NPRS(NPP)=IS2
C
      NPRS(NPP,NR)=IS2
C
CHANGE E
C
      GO TO 678
671 DO 672 I=1,NPP
C
CHANGE S DIMENSION BY HATA
C
      NPRS(I) ----> NPRS(I,J)
C

```

```

C      IJ=NPRS(I)
C
C      IJ=NPRS(I, NR)
C
CHANGE E
C
      IF(IJ.EQ.IS1) GO TO 673
672   CONTINUE
      NPP=NPP+1
C
CHANGE S  DIMENSION  BY HATA
C
C      NPRS(I) ---> NPRS(I, J)
C
C      NPRS(NPP)=IS1
C
C      NPRS(NPP, NR)=IS1
C
CHANGE E
C
      GO TO 674
673   CONTINUE
674   DO 675 J=1, NPP
C
CHANGE S  DIMENSION  BY HATA
C
C      NPRS(I) ---> NPRS(I, J)
C
C      IJ=NPRS(J)
C
C      IJ=NPRS(J, NR)
C
CHANGE E
C
      IF(IJ.EQ.NJ) GO TO 676
675   CONTINUE
      NPP=NPP+1
C
CHANGE S  DIMENSION  BY HATA
C
C      NPRS(I) ---> NPRS(I, J)
C
C      NPRS(NPP)=IS2
C
C      NPRS(NPP, NR)=IS2
C
CHANGE E
C
      GO TO 678
676   CONTINUE
678   CONTINUE
      GO TO 660
680   IF (MPI.GT.0) GO TO 690
      ISTOP=ISTOP+1
      WRITE(6,13700)
C
CHANGE S  DIMENSION  BY HATA
C

```

```

C      IS(I,J) ----> IS(I,J,K)
C
C 690 NPINC=IS(MPI,2)-IS(MPI,1)
C
C      690 NPINC=IS(MPI,2,NR)-IS(MPI,1,NR)
C
CHANGE E
C
C      MINC=IABS(NPINC)-1
C
CHANGE S DIMENSION BY HATA
C
C      NRSE(I) ----> NRSE(I,J)
C
C      MJ=NRSE(MPI)+MINC
C
C      MJ=NRSE(MPI,NR)+MINC
C
CHANGE E
C
C      MK=MI-1
C      DO 700 M=MJ,MK,MINC
C          MPJ=MPI
C          MPI=MPI+1
C
CHANGE S DIMENSION BY HATA
C
C      NRSE(I) ----> NRSE(I,J)
C      IS(I,J) ----> IS(I,J,K)
C
C      NRSE(MPI)=M
C      IS(MPI,1)=IS(MPJ,2)
C      NK=IS(MPI,1)
C      IS(MPI,2)=IS(MPI,1)+NPINC
C      NL=IS(MPI,2)
C
C      NRSE(MPI,NR)=M
C      IS(MPI,1,NR)=IS(MPJ,2,NR)
C      NK=IS(MPI,1,NR)
C      IS(MPI,2,NR)=IS(MPI,1,NR)+NPINC
C      NL=IS(MPI,2,NR)
C
CHANGE E
C
C      IF(MPI.GT.1) GO TO 691
C      NPP=NPP+1
C
CHANGE S DIMENSION BY HATA
C
C      NPRS(I) ----> NPRS(I,J)
C
C      NPRS(NPP)=NK
C
C      NPRS(NPP,NR)=NK
C
CHANGE E
C
C      NPP=NPP+1

```

```

C
CHANGE S  DIMENSION  BY HATA
C
C          NPRS(I) ----> NPRS(I,J)
C
C          NPRS(NPP)=NL
C
C          NPRS(NPP,NR)=NL
C
CHANGE E
C
C          GO TO 698
691      DO 692 K=1,NPP
C
CHANGE S  DIMENSION  BY HATA
C
C          NPRS(I) ----> NPRS(I,J)
C
C          KL=NPRS(K)
C
C          KL=NPRS(K,NR)
C
CHANGE E
C
C          IF(KL.EQ.NK) GO TO 693
692      CONTINUE
C          NPP=NPP+1
C
CHANGE S  DIMENSION  BY HATA
C
C          NPRS(I) ----> NPRS(I,J)
C
C          NPRS(NPP)=NK
C
C          NPRS(NPP,NR)=NK
C
CHANGE E
C
C          GO TO 694
693      CONTINUE
694      DO 695 L=1,NPP
C
CHANGE S  DIMENSION  BY HATA
C
C          NPRS(I) ----> NPRS(I,J)
C
C          KL=NPRS(L)
C
C          KL=NPRS(L,NR)
C
CHANGE E
C
C          IF(KL.EQ.NL) GO TO 696
695      CONTINUE
C          NPP=NPP+1
C
CHANGE S  DIMENSION  BY HATA
C

```

```

C      NPRS(I) ---> NPRS(I,J)
C
C      NPRS(NPP)=NL
C
C      NPRS(NPP,NR)=NL
C
CHANGE E
C
      GO TO 698
696     CONTINUE
698     CONTINUE
700     CONTINUE
      GO TO 670
710     NRSN=NPP
C
C ADD S  WRITE STATEMENT  BY HATA
C
      WRITE(6,10990) NR
C
C ADD E
C
      IF(KSS.EQ.0 .AND. IPASS.EQ.1) WRITE(6,10810)INPTAB
      IF(KSS.EQ.0 .AND. IPASS.EQ.2) WRITE(6,10820)INPTAB
      IF(KSS.EQ.1) WRITE(6,10820)INPTAB
C
CHANGE S  DIMENSION  BY HATA
C
C      NRSE(I) ---> NRSE(I,J)
C      IS(I,J) ---> IS(I,J,K)
C
C      DO 720 MP=1,NRSEL
C          M=NRSE(MP)
C 720     WRITE(6,11600)M,IS(MP,1),IS(MP,2)
C
C          DO 720 MP=1,NRSEL
C              M=NRSE(MP,NR)
C 720     WRITE(6,11600)M,IS(MP,1,NR),IS(MP,2,NR)
C
CHANGE E
C
C DETERMINE DIRECTION COSINES DCOSX(MP) AND DCOSZ(MP) FOR THE
C OPEN BOUNDARY SIDES
C
      DO 810 MPI=1,NRSEL
C
CHANGE S  DIMENSION  BY HATA
C
C      NRSE(I) ---> NRSE(I,J)
C
C      MI=NRSE(MPI)
C
C      MI=NRSE(MPI,NR)
C
CHANGE E
C
      DO 800 MPJ=1,NBEL
          MJ=NBE(MPJ)

```

```

                IF (MJ.NE.MI) GO TO 800
C
CHANGE S  DIMENSION  BY HATA
C
C          IS(I,J)  ---> IS(I,J,K)
C
C          IF (ISB(MPJ,1).EQ.IS(MPI,1).AND.ISB(MPJ,2).EQ.IS(MPI,2)) GO
C >          TO 780
C          IF (ISB(MPJ,1).EQ.IS(MPI,2).AND.ISB(MPJ,2).EQ.IS(MPI,1)) GO
C >          TO 780
C
C          IF (ISB(MPJ,1).EQ.IS(MPI,1,NR).AND.
C >          ISB(MPJ,2).EQ.IS(MPI,2,NR)) GO TO 780
C          IF (ISB(MPJ,1).EQ.IS(MPI,2,NR).AND.
C >          ISB(MPJ,2).EQ.IS(MPI,1,NR)) GO TO 780
C
CHANGE E
C
C          GO TO 800
C          DO 790 J=1,4
C
CHANGE S  DIMENSION  BY HATA
C
C          IS(I,J)  ---> IS(I,J,K)
C
C          IS(MPI,J)=ISB(MPJ,J)
C
C          IS(MPI,J,NR)=ISB(MPJ,J)
C
CHANGE E
C
C          DL(MPI)=DLB(MPJ)
C          DCOSX(MPI)=DCOSXB(MPJ)
C          DCOSZ(MPI)=DCOSZB(MPJ)
C          GO TO 810
C          800  CONTINUE
C          ISTOP=ISTOP+1
C          WRITE(6,13800)MI
C          810 CONTINUE
C
ADD S  IF STATEMENT  BY HATA
C
C          IF(NR.EQ.NRAM) GO TO 850
C          GO TO 661
C
ADD E
C
ADD S  GOURUND SURFACE NODE
C
C          850 IF (NSUR.LE.0) GO TO 890
C
READ GOUND SURFACE NODE TO CALCULATE FLUX
C
C          READ(5,11330) ICHA
C
C          IF(ICA.EQ.GSURF) GO TO 855
C          ISTOP=ISTOP+1

```

```

        WRITE(6,13600) ISTOP
        GO TO 830
C
C ADD E
C
      855 CONTINUE
        INPTAB=INPTAB+1
        NPP=0
      864 IF (NPP.EQ.NSUR) GO TO 861
        IF (NPP.LT.NSUR) GO TO 862
        WRITE(6,12900)NSUR
        ISTOP=ISTOP+1
        GO TO 861
      862 READ(5,12100)NI,NJ,NPINC
        IF(NPINC.GT.0) GO TO 863
      866 NPP=NPP+1
C
        NPGW(NPP,1)=NI
        NPGW(NPP,2)=NJ
C
        GO TO 864
      863 IF (NPP.GT.0) GO TO 867
        ISTOP=ISTOP+1
        WRITE(6,13900)
C
C
      867 NII=NPGW(NPP,1)+NPINC
        NJJ=NPGW(NPP,2)+NPINC
C
        NKI=NI-1
        NKJ=NJ-1
C
C
        NPPA=NPP
C
        DO 868 NP=NII,NKI,NPINC
          NPPA=NPPA+1
      868   NPGW(NPPA,1)=NP
C
        NPPA=NPP
C
        DO 871 NP=NJJ,NKJ,NPINC
          NPPA=NPPA+1
      871   NPGW(NPPA,2)=NP
C
        NPP=NPPA
C
        GO TO 866
      861 CONTINUE
CDBG BY HATA
C   WRITE(9,*) ' NPGW '
C   DO 1999 I=1,NPP
C     WRITE(9,1222) NPGW(I,1),NPGW(I,2)
C1999 CONTINUE
C1222 FORMAT(2(2X,I5))
CDBG
C
        WRITE(6,10850) INPTAB

```



```

      DO 869 NPP=1,NSUR
        WRITE(6,12010) NPGW(NPP,1),NPGW(NPP,2)
869  CONTINUE
C
      CALL ARDOS(NPGW,NPSS,NSUR,MAXBNP)
CDBG  BY HATA
C      WRITE(9,*) ' NPSS '
C      WRITE(9,1888) (NPSS(I),I=1,NSUR)
C1888  FORMAT(6(2X,I5))
C
C  ADD E
C
C  ADD S  READ STATEMENT  BY HATA
C
C      READ SOURCE TERM INFOMATION
C
C 890  IF(NSRC.LE.0) GO TO 960
C
C      READ(5,11330) ICHA
C
C      IF(ICHA.EQ.SOURC) GO TO 901
      ISTOP=ISTOP+1
      WRITE(6,13600) ISTOP
      GO TO 830
C
C 901  READ(5,10910) NR
C
C      INPTAB=INPTAB+1
C
C  ADD S  WRITE STATEMENT  BY HATA
C
C      WRITE(6,10990) NR
C
C  ADD E
C
C      IF(KSS.EQ.0 .AND. IPASS.EQ.1) WRITE(6,10950)INPTAB
      IF(KSS.EQ.0 .AND. IPASS.EQ.2) WRITE(6,10960)INPTAB
      IF(KSS.EQ.1) WRITE(6,10960)INPTAB
C
C      DO 910 KQ=1,NSRC
      READ(5,11320) NSELM(KQ,NR),QSRC(KQ,NR)
      WRITE(6,10970) NSELM(KQ,NR),QSRC(KQ,NR)
910  CONTINUE
C
C      IF(NR.EQ.NRAM) GO TO 950
      GO TO 901
950  CONTINUE
C
C  CALCULATE ELEMENT AREA
C
C      DO 919 IR=1,NRAM
      DO 920 II=1,NSRC
      ML=NSELM(II,IR)
      DO 930 IQ=1,4
      NP=IE(ML,IQ)
      XQ(IQ)=X(NP)

```

```

      ZQ(IQ)=Z(NP)
930 CONTINUE
C
      CALL AROSR(XQ,ZQ,AREB)
C
      ELA2(II,IR)=AREB
C
920 CONTINUE
919 CONTINUE
C
      DO 939 IR=1,NRAM
        TOTA(IR)=0.0
      DO 940 II=1,NSRC
        TOTA(IR)=TOTA(IR)+ELA2(II,IR)
940 CONTINUE
C      WRITE(9,*) ' IR  TOTA ',IR,TOTA
939 CONTINUE
C
960 CONTINUE
C
      IF(ISW1.EQ.1.OR.ISW2.EQ.1) GO TO 1000
      GO TO 999
C
1000 CONTINUE
C      READ(3,END=999) QTIM(NODR),(QTER(NODR,JJ),JJ=1,NRAM)
      READ(3,1009,END=999) QTIM(NODR),(QTER(NODR,JJ),JJ=1,NRAM)
CDBG BY HATA
C      WRITE(9,*) ' QTIM , QTER '
C      WRITE(9,*) QTIM(NODR),(QTER(NODR,JJ),JJ=1,NRAM)
CDBG
      NODR=NODR+1
      GO TO 1000
C
999 CONTINUE
      NODR=NODR-1
C
C      DO 1010 IRAM=1,NRAM
C      NBRED=1
C      CALL BFLUX(0.0,DP,ELA1,QTER,QTIM,EBA,AEL,NPST,IRAM,NBRED,
C      >          MXSTNP,MAXRAM,MAXRED)
C      EBA=EBA
C1010 CONTINUE
C
C ADD E
C
820 IF(ISTOP.EQ.0) GO TO 830
      WRITE(6,13600)ISTOP
830 CONTINUE
CDBG BY HATA
C      WRITE(6,*) ' DEBUG SUB DATAIN '
C      WRITE(6,*) '  NRAM ',NRAM
CDBG
      RETURN
C
C ADD S  FORMAT LIST BY HATA
C
C      10000 FORMAT( --- I5/5X,
C

```

```

C
C           :
C           40H NUMBER OF MEMBER IN DECAY CHAIN. . . .,I5/ 5X)
C
C10000 FORMAT(35H0INPUT TABLE 1.. BASIC PARAMETERS // 5X,
C > 40H NUMBER OF NODAL POINTS. . . . .,I5/ 5X,
C > 40H NUMBER OF ELEMENTS. . . . .,I5/ 5X,
C > 40H NUMBER OF DIFFERENT MATERIALS . . . .,I5/ 5X,
C > 40H NUMBER OF CORRECTION MATERIALS. . . .,I5/ 5X,
C > 40H NUMBER OF TIME INCREMENTS . . . .,I5/ 5X,
C > 40H NUMBER OF BOUNDARY CONDITIONS . . . .,I5/ 5X,
C > 40H NUMBER OF SURFACE TERMS . . . .,I5/ 5X,
C > 40H NUMBER OF SEEPAGE SURFACE TERMS . . . .,I5/ 5X,
C > 40H VELOCITY INPUT CONTROL. . . . .,I5/ 5X,
C > 40H AUXILIARY STORAGE CONTROL . . . . .,I5/ 5X,
C > 40H STEADY-STATE CONTROL. . . . .,I5/ 5X,
C > 40H NO. OF MATERIAL PROPERTY PER MATERIAL .I5/ 5X,
C > 40H UPSTREAM WEIGHTING INDICATOR, IWET. . .,I5/ 5X,
C > 40H LUMPING INDICATOR, ILUMP. . . . .,I5/ 5X,
C > 40H TIME-DIFFERENCE INDICATOR, IMID . . . .,I5)
C
10000 FORMAT(35H0INPUT TABLE 1.. BASIC PARAMETERS // 5X,
> 40H NUMBER OF NODAL POINTS. . . . .,I5/ 5X,
> 40H NUMBER OF ELEMENTS. . . . .,I5/ 5X,
> 40H NUMBER OF DIFFERENT MATERIALS . . . .,I5/ 5X,
> 40H NUMBER OF CORRECTION MATERIALS. . . .,I5/ 5X,
> 40H NUMBER OF TIME INCREMENTS . . . .,I5/ 5X,
> 40H NUMBER OF BOUNDARY CONDITIONS . . . .,I5/ 5X,
> 40H NUMBER OF SURFACE TERMS . . . .,I5/ 5X,
> 40H NUMBER OF SEEPAGE SURFACE TERMS . . . .,I5/ 5X,
> 40H NUMBER OF GROUND SURFACE TERMS . . . .,I5/ 5X,
> 40H NUMBER OF SOURCE TERMS. . . . .,I5/ 5X,
> 40H NUMBER OF MEMBER IN DECAY CHAIN . . . .,I5/ 5X,
> 40H VELOCITY INPUT CONTROL. . . . .,I5/ 5X,
> 40H AUXILIARY STORAGE CONTROL . . . . .,I5/ 5X,
> 40H STEADY-STATE CONTROL. . . . .,I5/ 5X,
> 40H NO. OF MATERIAL PROPERTY PER MATERIAL .I5/ 5X,
> 40H UPSTREAM WEIGHTING INDICATOR, IWET. . .,I5)
10010 FORMAT(/5X,
> 40H LUMPING INDICATOR, ILUMP. . . . .,I5/ 5X,
> 40H TIME-DIFFERENCE INDICATOR, IMID . . . .,I5/ 5X,
> 40H TIME-DEPENDENT SOURCE INDICATOR, ISW1 .I5/ 5X,
> 40H TIME-DEPENDENT FLUX INDICATOR, ISW2 . .I5/ 5X,
> 40H A FLAG OF RESTART FILE . . . . .,I5)
C
C ADD E
C
10001 FORMAT(1H ,5X,
> 40H TIME INCREMENT. . . . .,F10.6/ 5X,
> 40H MULTIPLIER FOR INCREASING DELT. . . .,F10.6/ 5X,
> 40H MAXIMUM VALUE OF DELT . . . . .,1PE10.3/ 5X,
> 40H MAXIMUM VALUE OF TIME . . . . .,1PE10.3/ 5X,
> 40H TIME-INTEGRATION PARAMETER. . . . .,0P,F10.6)
10100 FORMAT(/6X,14HOUTPUT CONTROL)
C
CHANGE S FORMAT STATEMENT BY HATA
C
C 10200 FORMAT( -- 9H MAT. NO. , 9(3A4)) ---> ( -- 9H MAT. NO. ,6(3A4)
C

```

```

C10200 FORMAT(36H1INPUT TABLE 2.. MATERIAL PROPERTIES// 9H MAT. NO., 9
C   > 3A4))
C
10200 FORMAT(36H1INPUT TABLE 2.. MATERIAL PROPERTIES// 9H MAT. NO., 6(
   > 3A4))
C
CHANGE E
C
C ADD S  FORMAT STATEMENT  BY HATA
C
10201 FORMAT(9X,3(3A4))
C
C ADD E
C
10300 FORMAT(34H1INPUT TABLE 3.. NODAL POINT DATA // 7X,4HNODE, 8X,1HX
   > 14X,1HZ,13X,4HNODE,8X,1HX,14X,1HZ,13X,4HNODE,8X,1HX,14X,1HZ)
10400 FORMAT(34H1INPUT TABLE 4.. ELEMENT DATA // 11X,
   > 31HGLOBAL INDICES OF ELEMENT NODES/7X,7HELEMENT, 3X,1H1,7X,1H2,
   > 7X,1H3,7X,1H4,6X,8HMATERIAL,6X, 10HNODE DIFF. )
10500 FORMAT(/52H CORRECTIONS TO MATERIAL TYPES FOR SELECTED ELEMENTS
C
CHANGE S  FORMAT STATEMENT  BY HATA
C
C   10510 FORMAT( --- , 'NO. OF SOURCE TERM ELEMENT, NSRC .....', I5)
C   10520 FORMAT( --- , 'NO. OF SOURCE TERM ELEMENT, NSRC .....', I5)
C
C10510FORMAT(1H1, 'INPUT TABLE', I3, '  STEADY STATE B. C. PARAMETERS'//5
C   > 'NO. OF DIRICHLET B. C. NODES ..... ', I5/5X,
C   > 'NO. OF CAUCHY B. C. ELEMENT-SIDES, NST ..... ', I5/5X,
C   > 'NO. OF NEUMANN B. C. ELEMENT-SIDES, NRSEL ..... ', I5)
C10520 FORMAT(1H1, 'INPUT TABLE', I3, '  TRANSIENT B. C. PARAMTERS'//5X,
C   > 'NO. OF DIRICHLET B. C. NODES ..... ', I5/5X,
C   > 'NO. OF CAUCHY B. C. ELEMENT-SIDES, NST ..... ', I5/5X,
C   > 'NO. OF NEUMANN B. C. ELEMENT-SIDES, NRSEL ..... ', I5)
C10610FORMAT(1H1, 'INPUT TABLE', I3, '  STEADY STATE B. C. OF FORM R=BB'//
C   > 6H  NODE,7X,2HBB/)
C10620 FORMAT(1H1, 'INPUT TABLE', I3, '  TRANSIENT B. C. OF FORM R=BB'//
C   > 6H  NODE,7X,2HBB/)
C10710 FORMAT(1H1, 'INPUT TABLE', I3, '    STEADY STATE CAUCHY B. C.  E=E
C   > AT NODE NI, E=EJ AT NODE NJ'//3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/
C10720 FORMAT(1H1, 'INPUT TABLE', I3, '  TRANSIENT STATE CAUCHY B. C.  E=E
C   > AT NODE NI, E=EJ AT NODE NJ'//3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/
C10810FORMAT(1H1, 'INPUT TABLE', I3, '    STEADY STATE NEUMANN B. C. INFO
C   > ATION'//5X,14HELEMENT NODE 1,2X,6HNODE 2)
C10820FORMAT(1H1, 'INPUT TABLE', I3, '  TRANSIENT STATE NEUMANN B. C. INFO
C   > ATION'//5X,14HELEMENT NODE 1,2X,6HNODE 2)
C
10510 FORMAT(1H1, 'INPUT TABLE', I3, '  STEADY STATE B. C. PARAMETERS'//5
   > 'NO. OF DIRICHLET B. C. NODES ..... ', I5/5X,
   > 'NO. OF CAUCHY B. C. ELEMENT-SIDES, NST ..... ', I5/5X,
   > 'NO. OF NEUMANN B. C. ELEMENT-SIDES, NRSEL ..... ', I5/5X,
   > 'NO. OF SOURCE TERM ELEMENTS , NSRC ..... ', I5)
10520 FORMAT(1H1, 'INPUT TABLE', I3, '  TRANSIENT B. C. PARAMTERS'//5X,
   > 'NO. OF DIRICHLET B. C. NODES ..... ', I5/5X,
   > 'NO. OF CAUCHY B. C. ELEMENT-SIDES, NST ..... ', I5/5X,
   > 'NO. OF NEUMANN B. C. ELEMENT-SIDES, NRSEL ..... ', I5/5X,
   > 'NO. OF SOURCE TERM ELEMENTS , NSRC ..... ', I5)
10610 FORMAT(1H , 'INPUT TABLE', I3, '  STEADY STATE B. C. OF FORM R=BB'//

```

```

    > 6H NODE,7X,2HBB/)
10620 FORMAT(1H , 'INPUT TABLE',I3,' TRANSIENT B. C. OF FORM R=BB'//
    > 6H NODE,7X,2HBB/)
10710 FORMAT(1H , 'INPUT TABLE',I3,' STEADY STATE CAUCHY B. C. E=EI
    >AT NODE NI, E=EJ AT NODE NJ'//3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/)
10720 FORMAT(1H , 'INPUT TABLE',I3,' TRANSIENT STATE CAUCHY B. C. E=EI
    >AT NODE NI, E=EJ AT NODE NJ'//3X,2HNI,3X,2HNJ,6X,2HEI,13X,2HEJ/)
10810 FORMAT(1H , 'INPUT TABLE',I3,' STEADY STATE NEUMANN B. C. INFO
    >ATION'//5X,14HELEMENT NODE 1,2X,6HNODE 2)
10820 FORMAT(1H , 'INPUT TABLE',I3,' TRANSIENT STATE NEUMANN B. C. INFO
    >ATION'//5X,14HELEMENT NODE 1,2X,6HNODE 2)
10850 FORMAT(
    >1H1,/'INPUT TABLE',I3,' GROUND SURFACE NODE INFORMATION'//
    > 20H NODE1 NODE2 /)
C
CHANGE E
C
10900 FORMAT(16I5)
C
C ADD S FORMAT STATEMENT BY HATA
C
C 10910 FORMAT(I5)
C
10910 FORMAT(I5)
10920 FORMAT(5I5)
10950 FORMAT(1H , 'INPUT TABLE',I3,' STEADY STATE SOURCE TERM INFORM
    >ION'//5X,10HELEMENT ,2X,6H Q )
10960 FORMAT(1H , 'INPUT TABLE',I3,' TRANSIENT STATE SOURCE TERM INFORM
    >ION'//5X,10HELEMENT ,2X,6H Q )
10970 FORMAT(8X,I4,3X,1PE12.4)
10990 FORMAT(1H1,I2,42HTH RADIONUCLIDE OF CONSIDERING DECAY CHAIN/)
C
C ADD E
C
CHANGE S FORMAT STATEMENT BY HATA
C
C11000 FORMAT(8F10.0)
C11100 FORMAT(80I1)
C11200 FORMAT(10X,40I2)
C11300 FORMAT(I8,1P9E12.4)
C
11000 FORMAT(6F10.0)
11010 FORMAT(3F10.0)
11100 FORMAT(80I1)
11200 FORMAT(10X,40I2)
11300 FORMAT(I8,1P6E12.4)
C
CHANGE E
C
C ADD S FORMAT STATEMENT BY HATA
C
C 11310 FORMAT(I8,1P3E12.4)
C 11320 FORMAT(I5,5X,F10.0)
C
11310 FORMAT(I8,I8,1P3E12.4)
11320 FORMAT(I5,5X,F10.0)
11330 FORMAT(A10)
1009 FORMAT(5X,E10.3,3(2X,E12.5))

```

```

C
C ADD E
C
11400 FORMAT(I5,2F10.3)
11500 FORMAT(1P,I10,2E15.4,I11,2E15.4,I11,2E15.4)
11600 FORMAT(I10,4I8,I10,I15)
11700 FORMAT(I10,32X,I10)
11800 FORMAT(I5,5X,F10.0)
11900 FORMAT(2I5,F10.0)
C
C ADD S  FORMAT STATEMENT  BY HATA
C
11910 FORMAT(2I5)
C
C ADD E
C
12000 FORMAT(I5,E15.4)
12010 FORMAT(2X,I5,4X,I5)
12100 FORMAT(3I5,5X,2F10.0)
12200 FORMAT(2I5,2(1PE15.4))
129000 FORMAT(////36H CHECK BOUNDARY CONDITIONS, MAXIMUM=,I5//)
13300 FORMAT(////30H ERROR IN NODAL-POINT CARD NO.,I5//)
13400 FORMAT(////26H ERROR IN ELEMENT CARD NO.,I5//)
13500 FORMAT(////36H ERROR IN INITIAL-CONDITION CARD NO.,I5//)
13600 FORMAT(////45H ASSEMBLY AND SOLUTION WILL NOT BE PERFORMED,,I5,
> 19H FATAL CARD ERRORS//)
13700 FORMAT(////38H ERROR IN FIRST TRANSIENT-SURFACE CARD//)
13800 FORMAT(////44H ERROR IN TRANSIENT-SURFACE CARD FOR ELEMENT,I5//)
13900 FORMAT(////49H ERROR IN FIRST R=BB TYPE BOUNDARY-CONDITION CARD /,
> /)
14000 FORMAT(////33H ERROR IN FIRST SURFACE-TERM CARD//)
14100 FORMAT(////40H ERROR IN MATERIAL TYPE CODE FOR ELEMENT ,I5//)
END

```

```

        SUBROUTINE AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
        > MAXMAT,MXMPPM,NEL,W,KSS,PROPR,IRAM,MAXRAM)
C
C ADD S DUMMY ARGUMENT OF SUBROUTINE ARGUMENT BY HATA
C
C SUBROUTINE AFABTA( --- ,PROPR,IRAM,MAXRAM)
C
C SUBROUTINE AFABTA(X,Z,IE,WETAB, VX,VXP,VZ,VZP, PROP, MAXNP,MAXEL,
C ) MAXMAT,MXMPPM,NEL,W,KSS)
C
C ADD E
C
C
C FUNCTION OF THE SUBROUTINE-TO CALCULATE THE WEIGHTING FACTORS
C ON FOUR SIDES OF EACH OF THE ELEMENTS
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C CHANGE S DIMENSION BY HATA
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
C DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP)
C DIMENSION PROP(MAXMAT,MXMPPM)
C DIMENSION APHA(4),BETA(4),MDIM(4)
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4,MAXRAM)
C DIMENSION VX(MAXNP),VXP(MAXNP),VZ(MAXNP),VZP(MAXNP)
C DIMENSION PROP(MAXMAT,MXMPPM),PROPR(MAXMAT,MXMPPM,MAXRAM)
C DIMENSION APHA(4),BETA(4),MDIM(4)
C
C CHANGE E
C
C W1=W
C W2=1.0-W
C IF(KSS.EQ.0) GO TO 100
C W1=1.0
C W2=0.0
100 CONTINUE
C DO 500 M=1,NEL
C MTYP=IE(M,5)
C
C CHANGE S VARIABLE FUNCTION BY HATA
C
C AL=PROP(MTYP,3)
C AT=PROP(MTYP,4)
C AM=PROP(MTYP,8)
C TAU=PROP(MTYP,9)
C
C AL=PROP(MTYP,2)
C AT=PROP(MTYP,3)
C AM=PROPR(MTYP,3,IRAM)
C TAU=PROP(MTYP,6)
C
C CHANGE E
C
C DD=TAU*AM
CDBG BY HATA
C WRITE(6,*) ' DEBUG AFABTA '

```

```

C      WRITE(6,*) '          IRAM      ',IRAM
C      WRITE(6,*) '  AL,AT,AM,TAU,DD ',AL,AT,AM,TAU,DD
CDBG
C
      M1=IE(M,1)
      M2=IE(M,2)
      M3=IE(M,3)
      M4=IE(M,4)
C
      A=Z(M2)-Z(M1)
      B=X(M2)-X(M1)
      BETA(1)= ATAN2(A,B)
      A=Z(M3)-Z(M2)
      B=X(M3)-X(M2)
      BETA(2)= ATAN2(A,B)
      A=Z(M3)-Z(M4)
      B=X(M3)-X(M4)
      BETA(3)= ATAN2(A,B)
      A=Z(M4)-Z(M1)
      B=X(M4)-X(M1)
      BETA(4)= ATAN2(A,B)
      MDIM(1)=M1
      MDIM(2)=M2
      MDIM(3)=M3
      MDIM(4)=M4
C
      DO 400 J=1,4
      JQ=J
      JQP=JQ+1
      IF(JQ.EQ.4) JQP=1
      M1=MDIM(J)
      IF(J-3) 200,200,210
200  M2=MDIM(J+1)
      GO TO 300
210  M2=MDIM(1)
300  CONTINUE
      ALENG= SQRT((X(M2)-X(M1))**2+(Z(M2)-Z(M1))**2)
      CBETA= COS(BETA(J))
      SBETA= SIN(BETA(J))
      VXX=0.5*((VX(M1)+VX(M2))*W1+(VXP(M1)+VXP(M2))*W2)
      VZZ=0.5*((VZ(M1)+VZ(M2))*W1+(VZP(M1)+VZP(M2))*W2)
      VAL=VXX*CBETA+VZZ*SBETA
      VV= SQRT(VXX*VXX+VZZ*VZZ)
      IF(VV.EQ.0.0) GO TO 330
      VVI=1.0/VV
      DLL=(AL*VXX*VXX + AT*VZZ*VZZ)*VVI + DD
      DLT=(AL-AT)*VXX*VZZ*VVI
      DTT=(AL*VZZ*VZZ + AT*VXX*VXX)*VVI + DD
      DAL= ABS(DLL*CBETA*CBETA+2.0E0*CBETA*SBETA*DLT+DTT*SBETA*SBETA)
      VEL=VAL*ALENG
      DISP=2.0*DAL
      IF (DISP.EQ.0.0.OR.VEL.EQ.0.0) GO TO 320
      APHA(J)=1.0/ TANH(VEL/DISP)- DISP/VEL
      IF(APHA(J).LT.0.0) APHA(J)=0.0
      GO TO 400
320  IF(VEL.EQ.0.0) APHA(J)=0.0
      IF(VEL.NE.0.0) APHA(J)=1.0
      GO TO 400

```


330 APHA(J)=0.0
400 CONTINUE

C
CHANGE S DIMENSION BY HATA
C
C WETAB(I,J) ---> WETAB(I,J,K)
C
C WETAB(M,1)=APHA(1)
C WETAB(M,2)=APHA(2)
C WETAB(M,3)=APHA(3)
C WETAB(M,4)=APHA(4)
C
C WETAB(M,1,IRAM)=APHA(1)
C WETAB(M,2,IRAM)=APHA(2)
C WETAB(M,3,IRAM)=APHA(3)
C WETAB(M,4,IRAM)=APHA(4)
C
CHANGE E
C
500 CONTINUE
RETURN
END

```

      SUBROUTINE FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ,PROP,
> MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM,MAXRAM,IRAM,PROPR)
C
C ADD S DUMMY ARGUMENT OF SUBROUTINE STATEMENT BY HATA
C
C SUBROUTINE FLUX( --- ,MAXRAM,IRAM)
C
C SUBROUTINE FLUX(X,Z,IE,WETAB, C,FX,FZ, R,VX,VZ,PROP,
> MAXNP,MAXEL,MAXBW,MAXMAT,MXMPPM)
C
C ADD E
C
C FUNCTION OF SUBROUTINE TO COMPUTE DARCY MATERIAL FLUX FX AND FZ,
C RESPECTIVELY
C
      IMPLICIT REAL*8(A-H,O-Z)
C
CHANGE S DIMENSION BY HATA
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION C(MAXNP,MAXBW),FX(MAXNP),FZ(MAXNP)
      DIMENSION R(MAXNP),VX(MAXNP),VZ(MAXNP),PROP(MAXMAT,MXMPPM)
      DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4,MAXRAM)
      DIMENSION C(MAXNP,MAXBW),FX(MAXNP),FZ(MAXNP)
      DIMENSION R(MAXNP),VX(MAXNP),VZ(MAXNP)
      DIMENSION PROP(MAXMAT,MXMPPM),PROPR(MAXMAT,MXMPPM,MAXRAM)
      DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C
CHANGE E
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C INITIALIZE THE FLUX FX(NP) AND FZ(NP)
C
      DO 100 NP=1,NNP
        FX(NP)=0.0
100    FZ(NP)=0.0
C
C COMPUTE THE FLUX COMPONENTS BY APPLYING THE FINITE ELEMENT METHOD
C TO THE DISPERSION TERMS
C
      IHALFB=(IBAND-1)/2
      IHBP=IHALFB+1
      DO 300 IXZ=1,2
        DO 110 NP=1,NNP
          DO 110 IB=1,IBAND
110    C(NP,IB)=0.0
        DO 180 M=1,NEL
C
CHANGE S DIMENSION BY HATA
C
      WETAB(I,J) ----> WETAB(I,J,K)
C
      APHA1=WETAB(M,1)
      APHA2=WETAB(M,3)

```

```

C          BETA1=WETAB(M,2)
C          BETA2=WETAB(M,4)
C
C          APHA1=WETAB(M,1,IRAM)
C          APHA2=WETAB(M,3,IRAM)
C          BETA1=WETAB(M,2,IRAM)
C          BETA2=WETAB(M,4,IRAM)
C
C CHANGE E
C          MTYP=IE(M,5)
C
C CHANGE S VARIABLE FUNCTION BY HATA
C
C          AL=PROP(MTYP,3)
C          AT=PROP(MTYP,4)
C          AM=PROP(MTYP,8)
C          TAU=PROP(MTYP,9)
C
C          AL=PROP(MTYP,2)
C          AT=PROP(MTYP,3)
C          AM=PROPR(MTYP,3,IRAM)
C          TAU=PROP(MTYP,6)
C
C CHANGE E
C
C          DD=AM*TAU
C
CDBG
C          WRITE(6,*) ' DEBUG FLUX '
C          WRITE(6,*) ' IRAM ',IRAM
C          WRITE(6,*) ' AL,AT,AM,TAU,DD ',AL,AT,AM,TAU,DD
CDBG
C          DO 120 IQ=1,4
C              NP=IE(M,IQ)
C              XQ(IQ)=X(NP)
C              ZQ(IQ)=Z(NP)
C              CRQ(IQ)=R(NP)
C              VXQ(IQ)=VX(NP)
C              VZQ(IQ)=VZ(NP)
120          CONTINUE
C
C          COMPUTE THE ELEMENT MATRIX QB(IQ,JQ) AND QQ(IQ)
C
C          CALL Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DD,AL,AT,SNFE,CSFE,IXZ)
C
C          ASSEMBLE QB(IQ,JQ) INTO THE GLOBAL MATRIX C(NP,IB) AND
C          FORM THE LOAD VECTOR FX(NP) OR FZ(NP)
C
C          DO 140 IQ=1,4
C              NI=IE(M,IQ)
C              DO 130 JQ=1,4
C                  NJ=IE(M,JQ)
C                  IB=NJ-NI+IHBP
C                  C(NI,IB)=C(NI,IB)+QB(IQ,JQ)
130          CONTINUE
C
C          IF(IXZ.EQ.2) GO TO 135
C          FX(NI)=FX(NI)+RQ(IQ)

```

```
                GO TO 140
135             FZ(NI)=FZ(NI)+RQ(IQ)
140             CONTINUE
180             CONTINUE
C
C SOLVE THE MATRIX EQUATION CX=B
C
                IF(IXZ.EQ.2) GO TO 200
                CALL SOLVE(1,C,FX,NNP,IHALFB,MAXNP,MAXBW)
                CALL SOLVE(2,C,FX,NNP,IHALFB,MAXNP,MAXBW)
                GO TO 300
200             CALL SOLVE(1,C,FZ,NNP,IHALFB,MAXNP,MAXBW)
                CALL SOLVE(2,C,FZ,NNP,IHALFB,MAXNP,MAXBW)
300             CONTINUE
                RETURN
                END
```

```

SUBROUTINE Q4D(QB,RQ,XQ,ZQ,VXQ,VZQ,CRQ,DD,AL,AT,SNFE,CSFE,IND)
C
C FUNCTION OF SUBROUTINE-TO EVALUATE THE MATRIX QUADRATURE OVER THE
C AREA OF ONE ELEMENT. THESE INTEGRALS ARISE THROUGH THE
C APPLICATION OF THE GALERKIN INTEGRATION SCHEME
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   REAL*8 N(4)
C   DIMENSION QB(4,4),RQ(4),XQ(4),ZQ(4),CRQ(4),VXQ(4),VZQ(4)
C   DIMENSION S(4),T(4),W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
C   DIMENSION PJAB(2,2),DNSS(4),DNNT(4),DWSS(4),DWTT(4)
C
C   DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,--
C   > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C   INITIALIZE MATRICES QB(IQ,JQ) AND QQ(IQ)
C
C   ADD S BY HATA
C
C   SNFE=SNFE
C   CSFE=CSFE
C
C   ADD E
C
C   DO 100 IQ=1,4
C     RQ(IQ)=0.0
C     DO 100 JQ=1,4
100       QB(IQ,JQ)=0.0
C
C   SUMMATION OF THE INTEGRAND OVER THE GAUSSIAN POINTS
C
C   DO 400 KG=1,4
C
C   DETERMINE LOCAL COORDINATE (SS,TT) OF
C   GAUSS-INTEGRATION POINT KG
C
C   SS=P*S(KG)
C   TT=P*T(KG)
C
C   CALCULATE THE VALUES OF BASIS FUNCTIONS, N(IQ), AND WEIGHTING
C   W(IQ) AND THEIR DERIVATIVES WITH RESPECTIVE TO X AND Z, RESPECTIVEL
C   AT THE GAUSSIAN POINT KG
C
C   CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWTT,SS,TT)
C   DO 110 I=1,2
C     DO 110 J=1,2
110       PJAB(I,J)=0.0
C   DO 120 I=1,4
C     PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNNT(I)
C     PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
C     PJAB(2,1)=PJAB(2,1)-XQ(I)*DNNT(I)
120       PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
C     DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
C     DJACI=1.0/DJAC
C     DO 130 I=1,2
C       DO 130 J=1,2
130         PJAB(I,J)=PJAB(I,J)*DJACI
C     DO 140 I=1,4

```

```

          DNX(I)=DNSS(I)*PJAB(1,1)+DN TT(I)*PJAB(1,2)
          DNZ(I)=DNSS(I)*PJAB(2,1)+DN TT(I)*PJAB(2,2)
          DWX(I)=DWSS(I)*PJAB(1,1)+DW TT(I)*PJAB(1,2)
140      DWZ(I)=DWSS(I)*PJAB(2,1)+DW TT(I)*PJAB(2,2)
C
C INTERPOLATE WITH THE BASIS FUNCTIONS N(IQ) TO OBTAIN VALUES OF
C VELOCITY COMPONENTS VXK AND VZK AT GAUSSIAN POINT
C
          VXK=0.0
          VZK=0.0
          DO 150 IQ=1,4
              VXK=VXK+VXQ(IQ)*N(IQ)
              VZK=VZK+VZQ(IQ)*N(IQ)
150      CONTINUE
C
          VK= SQRT(VXK*VXK+VZK*VZK)
          VKI=1.0/VK
          AKXK=(AL*VXK*VXK+AT*VZK*VZK)*VKI + DD
          AKZK=(AL*VZK*VZK+AT*VXK*VXK)*VKI + DD
          AKXZK=(AL-AT)*VXK*VZK*VKI
C
C ACCUMULATE THE SUMS TO OBTAIN THE MATRIX INTEGRALS QB(IQ,JQ)
C AND QQ(IQ)
C
          DO 300 IQ=1,4
              DO 300 JQ=1,4
                  QB(IQ,JQ)=QB(IQ,JQ)+W(IQ)*N(JQ)*DJAC
                  IF(IND.EQ.2) GO TO 200
                  RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXK*DNX(JQ)+AKXZK*DNZ(JQ)
>                  DJAC + W(IQ)*CRQ(JQ)*VXK*N(JQ)*DJAC
                  GO TO 300
200      RQ(IQ)=RQ(IQ)-W(IQ)*CRQ(JQ)*(AKXZK*DNX(JQ)+AKZK*DNZ(JQ)
>                  DJAC + W(IQ)*CRQ(JQ)*VZK*N(JQ)*DJAC
C
300      CONTINUE
400      CONTINUE
          RETURN
          END

```

```

SUBROUTINE ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DT
> PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MAXMPPM,MAXRAM,MAXSRC,IRAM,PROPR
> RDEC,NSELM,QSRC,NSRC)
C
C ADD S DUMMY ARGUMENT OF SUBROUTINE STATEMENT BY HATA
C
C SUBROUTINE ASEMBL( --- ,MAXRAM,MAXSRC,IRAM,PROPR,RDEC,
C NSELM,QSRC,NSRC)
C
C SUBROUTINE ASEMBL(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,TH,THP,DH,DT
C > PROP,W,MAXNP,MAXEL,MAXBW,MAXMAT,MAXMPPM)
C
C ADD E
C
C FUNCTION OF SUBROUTINE--TO ASSEMBLE THE GLOBAL COEFFICIENT MATRIX
C C(NP,IB) AND LOAD VECTOR R(NP) FROM THE ELEMENT MATRICES QA(IQ,JQ)
C AND QB(IQ,JQ).
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 KD,LAMBDA
C
C CHANGE S DIMENSION BY HATA
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),VX(MAXNP),VXP(MAXNP)
C > VZ(MAXNP),VZP(MAXNP),TH(MAXEL,4),THP(MAXEL,4),DH(MAXNP),
C > DTH(MAXEL,4),PROP(MAXMAT,MAXMPPM)
C
C DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
C > DTHQ(4),THQ(4)
C
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4,MAXRAM)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP),VX(MAXNP),VXP(MAXNP),
C > RP(MAXNP,MAXRAM),RDEC(MAXNP,2),
C > VZ(MAXNP),VZP(MAXNP),TH(MAXEL,4),THP(MAXEL,4),DH(MAXNP),
C > DTH(MAXEL,4),PROP(MAXMAT,MAXMPPM),PROPR(MAXMAT,MAXMPPM,MAXRAM)
C
C DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
C > DTHQ(4),THQ(4),QRR(4,4),QRS(4)
C
C DIMENSION QRR(4)
C
C DIMENSION NSELM(MAXSRC,MAXRAM),QSRC(MAXSRC,MAXRAM)
C
C CHANGE E
C
C COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
C COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
C COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
C COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C CDBG
C WRITE(9,*) ' SUBROUTINE ASEMBL '
C CDBG
C IHALFB=(IBAND-1)/2
C IHBP=IHALFB+1
C

```

```

        DELTI=1./DELT
        W1=W
        W2=1.-W
        IF (KSS.NE.0) GO TO 10
        DELTI=0.
        W1=1.
        W2=0.
C
C   INITIALIZE MATRICES C(NP,IB) AND R(NP)
C
    10 DO 20 NP=1,NNP
        R(NP)=0.
        DO 20 IB=1,IBAND
    20     C(NP,IB)=0.0
C
C   ADD S DO LOOP 21 BY HATA
C
        DO 21 NP=1,NNP
            RDEC(NP,1)=RDEC(NP,2)
            RDEC(NP,2)=0.0
    21 CONTINUE
CDBG BY HATA
        TTT=0.
        DO 22 NP=1,NNP
            TTT=TTT+RDEC(NP,1)
    22 CONTINUE
C   WRITE(9,*) ' TTT ',TTT
C   WRITE(6,*) ' RDEC(NP,1) '
C   WRITE(6,111) (RDEC(NP,1),NP=1,NNP)
C 111 FORMAT(1X,5(2X,E12.4))
C   TLAMB=0.0
C   TLAMA=0.0
CDBG
C
C   ADD E
C
C
C   COMPUTE MATRICES QA(IQ,JQ) AND QB(IQ,JQ) FOR EACH ELEMENT M
C
        DO 50 M=1,NEL
C
C   CHANGE S DIMENSION BY HATA
C
C       WETAB(I,J) ---> WETAB(I,J,K)
C
C       APHA1=WETAB(M,1)
C       APHA2=WETAB(M,3)
C       BETA1=WETAB(M,2)
C       BETA2=WETAB(M,4)
C
C       APHA1=WETAB(M,1,IRAM)
C       APHA2=WETAB(M,3,IRAM)
C       BETA1=WETAB(M,2,IRAM)
C       BETA2=WETAB(M,4,IRAM)
CDBG BY HATA
C       WRITE(9,*) ' APHA1,APHA2,BETA1,BETA2 '
C       WRITE(9,1111) APHA1,APHA2,BETA1,BETA2
C 1111 FORMAT(4(3X,E10.3))

```



```

CDBG
C
CHANGE E
C
      MTYP=IE(M,5)
C
CHANGE S  VARIABLE FUNCTION  BY HATA
C
      KD=PROP(MTYP,1)
      RHOB=PROP(MTYP,2)
      AL=PROP(MTYP,3)
      AT=PROP(MTYP,4)
      LAMBDA=PROP(MTYP,5)
      POR=PROP(MTYP,6)
      ALP=PROP(MTYP,7)
      AM=PROP(MTYP,8)
      TAU=PROP(MTYP,9)
C
      KD=PROPR(MTYP,1,IRAM)
      RHOB=PROP(MTYP,1)
      AL=PROP(MTYP,2)
      AT=PROP(MTYP,3)
      LAMBDA=PROPR(MTYP,2,IRAM)
      POR=PROP(MTYP,4)
      ALP=PROP(MTYP,5)
      AM=PROPR(MTYP,3,IRAM)
      TAU=PROP(MTYP,6)
CDBG  BY HATA
C      WRITE(6,*) ' KD,RHOB,AL,AT,LAMBDA,POR,ALP,AM,TAU '
C      WRITE(6,*)  KD,RHOB,AL,AT,LAMBDA,POR,ALP,AM,TAU
CDBG
C
CHANGE E
C
      DO 30 IQ=1,4
      NP=IE(M,IQ)
      XQ(IQ)=X(NP)
      ZQ(IQ)=Z(NP)
      DHQ(IQ)=DH(NP)
      THQ(IQ)=TH(M,IQ)*W1+THP(M,IQ)*W2
      DTHQ(IQ)=DTH(M,IQ)
      VXQ(IQ)=VX(NP)*W1+VXP(NP)*W2
      VZQ(IQ)=VZ(NP)*W1+VZP(NP)*W2
      30  CONTINUE
C
C ADD S  ACTUAL ARGUMENT OF CALL STATEMENT  BY HATA
C
      CALL Q4( --- ,QRR,M,IRAM,MAXRAM,MAXSRC,NSELM,QSRC,NSRC)
C
      CALL Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,      DTHQ,DHQ,KD,RHOB,AL,AT,
C  >  LAMBDA,POR,ALP,AM,TAU)
C
      MELM=M
C
CDBG  BY HATA
C      WRITE(9,*) ' B CALL Q4 '
C      WRITE(9,*) ' XQ , ZQ '
C      WRITE(9,1111) (XQ(KK),ZQ(KK),KK=1,4)

```

```

C      WRITE(9,*) ' VZQ , VZQ '
C      WRITE(9,1111) (VXQ(KK),VZQ(KK),KK=1,4)
C      WRITE(9,*) ' THQ '
C      WRITE(9,1111) (THQ(KK),KK=1,4)
C      WRITE(9,*) ' DTHQ '
C      WRITE(9,1111) (DTHQ(KK),KK=1,4)
C      WRITE(9,*) ' DHQ '
C      WRITE(9,1111) (DHQ(KK),KK=1,4)
C      WRITE(9,*) ' QA '
C      WRITE(9,1111) ((QA(KK,LL),LL=1,4),KK=1,4)
C      WRITE(9,*) ' QB '
C      WRITE(9,1111) ((QB(KK,LL),LL=1,4),KK=1,4)
CDBG
      CALL Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,DTHQ,DHQ,KD,RHOB,AL,AT,LAMBDA
      > POR,ALP,AM,TAU,QRR,MELM,IRAM,MAXRAM,MAXSRC,NSERM,NSRC,QSRC,QR
      > RSRCC)
CDBG BY HATA
C      WRITE(9,*) ' B CALL Q4 '
C      WRITE(9,*) ' QA '
C      WRITE(9,1111) ((QA(KK,LL),LL=1,4),KK=1,4)
C      WRITE(9,*) ' QB '
C      WRITE(9,1111) ((QB(KK,LL),LL=1,4),KK=1,4)
C      WRITE(9,*) ' QRR '
C      WRITE(9,1111) ((QRR(KK,LL),LL=1,4),KK=1,4)
CDBG
C
C ADD E
C
C ASSEMBLE QA(IQ,JQ) AND QB(IQ,JQ) INTO THE GLOBAL MATRIX
C  $C(NP,IB) = W1*B + A/DELTA$  AND FORM THE LOAD VECTOR
C  $R(NP) = (A/DELTA - W2*B)*RP$ . MATRIX C IS ASYMMETRIC DUE TO
C THE ADVECTION TERM.
C
      IF(IMID.EQ.1) GO TO 41
C
      DO 40 IQ=1,4
        NI=IE(M,IQ)
        DO 45 JQ=1,4
          NJ=IE(M,JQ)
          QA(IQ,JQ)=QA(IQ,JQ)*DELTA
C
C CHANGE S DIMENSION AND ADD VARIABLE
C
C      RP(I) ---> RP(I,J)
C
C      R(NI)=R(NI)+ --- +RDEC(NI,1)
C      RDEC(NI,2)=-QRR(IQ,JQ)*RP(NJ,IRAM)
C
C      R(NI)=R(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*RP(NJ)
C
C      R(NI)=R(NI)+(QA(IQ,JQ)-W2*QB(IQ,JQ))*RP(NJ,IRAM)
C      RDEC(NI,2)=RDEC(NI,2)+QRR(IQ,JQ)*RP(NJ,IRAM)
CDBG BY HATA
C      WRITE(9,*) ' R(NI) ',R(NI)
C      WRITE(9,*) ' RDEC(NI,2) ',RDEC(NI,2)
CDBG
C
CHANGE E

```

```

C
      IB=NJ-NI+IHBP
C
C ADD S DECAF EFFECT BY HATA
C
      C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)
C
      C(NI,IB)=C(NI,IB)+QA(IQ,JQ)+W1*QB(IQ,JQ)
C
C ADD E
C
      45      CONTINUE
C            RDEC(NI,2)=RDEC(NI,2)+QRR(IQ)*RP(NI,IRAM)
C            R(NI)=R(NI)+QRS(IQ)
      40      CONTINUE
C
C
      GO TO 50
C
C
      41      DO 43 IQ=1,4
C            NI=IE(M,IQ)
C            DO 46 JQ=1,4
C              NJ=IE(M,JQ)
C              QA(IQ,JQ)=2.0D0*QA(IQ,JQ)*DELTI
C
C CHANGE S DIMENSION AND ADD VARIABLE
C
      RP(I) ---> RP(I,J)
C
      R(NI)=R(NI)+ --- +RDEC(NI,1)
C
      R(NI)=R(NI) + QA(IQ,JQ)*RP(NJ)
C
      R(NI)=R(NI) + QA(IQ,JQ)*RP(NJ,IRAM)
      RDEC(NI,2)=RDEC(NI,2)+QRR(IQ,JQ)*RP(NJ,IRAM)
C
C CHANGE E
C
      IB=NJ-NI+IHBP
C
C ADD S DECAF EFFECT BY HATA
C
      C(NI,IB)=C(NI,IB) + QA(IQ,JQ) + QB(IQ,JQ)
C
      C(NI,IB)=C(NI,IB) + QA(IQ,JQ) + QB(IQ,JQ)
      46      CONTINUE
C            RDEC(NI,2)=RDEC(NI,2)+QRR(IQ)*RP(NI,IRAM)
C            R(NI)=R(NI) + QRS(IQ)
      43      CONTINUE
C
      50      CONTINUE
CDBG
C      WRITE(9,*) ' STIFF MATRIX'
C      WRITE(9,1111) ((C(I,J),J=1,IBAND),I=1,NNP)
CDBG
C
C ADD S BY HATA

```

```
C
      DO 24 NP=1,NNP
        R(NP)=R(NP)+RDEC(NP,1)
24    CONTINUE
CDBG  BY HATA
      DO 25 NP=1,NNP
        TLAMA=TLAMA+RDEC(NP,2)
25    CONTINUE
C      WRITE(9,*) ' TLAMA ',TLAMA
CDBG
C
C ADD E
C
      RETURN
      END
```

```

        SUBROUTINE Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,DTHQ,DHQ,KD,RHOB,AL,AT,
    >    LAMBDA,POR,ALP,AM,TAU,QRR,MELM,IRAM,MAXRAM,MAXSRC,NSELM,NSRC
    >    QSRC,QRS)
C
C CHANGE S DUMMY ARGUMENT BY HATA
C
C    SUBROUTINE Q4( --- ,QRR,MELM,IRAM,MAXRAM,MAXSRC,NSELM,NSRC,QSRC)
C
C    SUBROUTINE Q4(QA,QB,XQ,ZQ,VXQ,VZQ,THQ,      DTHQ,DHQ,KD,RHOB,AL,AT
C    >    LAMBDA,POR,ALP,AM,TAU)
C
C CHANGE E
C
C    FUNCTION OF SUBROUTINE--TO EVALUATE THE MATRIX QUADRATURES QA(IQ,JQ
C    AND QB(IQ,JQ) OVER THE AREA OF ONE ELEMENT.
C
C
C    IMPLICIT REAL*8 (A-H,O-Z)
C    REAL*8 N(4),KD,LAMBDA
C
C
C    DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
C    DIMENSION PJAB(2,2),DNSS(4),DNST(4),DWSS(4),DWTT(4)
C    DIMENSION QA(4,4),QB(4,4),VXQ(4),VZQ(4),XQ(4),ZQ(4),DHQ(4),
C    > DTHQ(4),THQ(4)
C    DIMENSION S(4),T(4)
C
C ADD S ARRAY FOR DECAY TERM
C
C    DIMENSION QRR(4,4),NSELM(MAXSRC,MAXRAM),QSRC(MAXSRC,MAXRAM),
C    >    QRS(4)
C    DIMENSION QRR(4)
C
C ADD E
C
C
C    COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C    DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00,-
C    > 1.0D+00 /, T / -1.0D+00,-1.0D+00, 1.0D+00, 1.0D+00 /
C
C    DD=AM*TAU
C
C INITIALIZE MATRICES QA(IQ,JQ) AND QB(IQ,JQ)
C
C
C ADD S ARRAY FOR DECAY TERM QRR
C
C    POR=POR
C
C    DO 10 IQ=1,4
C        DO 10 JQ=1,4
C            QA(IQ,JQ)=0.0
C    10    QB(IQ,JQ)=0.0
C
C    POR=POR
C
C    DO 10 IQ=1,4

```

```

C          QRR(IQ)=0.0
          QRS(IQ)=0.0
DO 10 JQ=1,4
          QA(IQ,JQ)=0.0
          QB(IQ,JQ)=0.0
          QRR(IQ,JQ)=0.0
10 CONTINUE
C
C ADD E
C
C          AREA=0.000
C
C          DO 40 KG=1,4
C
C DETERMINE THE LOCAL COORDINATES (SS,TT) AND EVALUATE THE JACOBIAN A
C EACH GAUSS-INTEGRATION POINT KG
C
C          SS = P*S(KG)
          TT = P*T(KG)
C
C CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND W(IQ)
C AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT TO X AND Z,
C RESPECTIVELY, AT THE GAUSS POINT KG
C
C          CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN
C PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=- (DZ/DS)/DJAC,
C PJAB(2,1)=- (DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC, WHERE
C DJAC IS THE DETERMINANT OF THE JACOBIAN
C
C          DO 11 I=1,2
          DO 11 J=1,2
            PJAB(I,J)=0.0
11          CONTINUE
C
C          DO 16 I=1,4
          PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNTT(I)
          PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
          PJAB(2,1)=PJAB(2,1) - XQ(I)*DNTT(I)
          PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
16          CONTINUE
C
C COMPUTE THE DETERMINANT OF THE JACOBIAN
C
C          DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
          DJACI=1.000/DJAC
          AREA=AREA + DJAC
C
C          DO 17 I=1,2
          DO 17 J=1,2
17          PJAB(I,J)=PJAB(I,J)*DJACI
C
C          DO 18 I=1,4
          DNX(I)=PJAB(1,1)*DNSS(I) + PJAB(1,2)*DNTT(I)
          DNZ(I)=PJAB(2,1)*DNSS(I) + PJAB(2,2)*DNTT(I)
          DWX(I)=PJAB(1,1)*DWSS(I) + PJAB(1,2)*DWTT(I)

```

```

18      DWZ(I)=PJAB(2,1)*DWSS(I) + PJAB(2,2)*DWTT(I)
C
C INTERPOLATE WITH THE BASIS-INTERPOLATION FUNCTIONS N(IQ) TO OBTAIN
C THE ADVECTIVE VELOCITY AT EACH GAUSS INTEGRATION POINT
C
      DHK=0.
      THK=0.
      DTHK=0.
      VXK=0.
      VZK=0.
      DO 20 IQ=1,4
          DHK=DHK+N(IQ)*DHQ(IQ)
          THK=THK+N(IQ)*THQ(IQ)
          DTHK=DTHK+N(IQ)*DTHQ(IQ)
          VXK=VXK+N(IQ)*VXQ(IQ)
          VZK=VZK+N(IQ)*VZQ(IQ)
20
C
C ADD S ASSIGN STATEMENT BY HATA
C
CDBG
C      WRITE(6,*) ' N ',(N(I),I=1,4)
C      WRITE(6,*) ' RDECQ ',(RDECQ(I),I=1,4)
CDBG
C
C ADD E
C
      VK= SQRT(VXK*VXK+VZK*VZK)
      VKI=1./VK
      A=DJAC*(THK+RHOB*KD)
      DXX=DJAC*((AL*VXK*VXK+AT*VZK*VZK)*VKI+DD)
      DZZ=DJAC*((AL*VZK*VZK+AT*VXK*VXK)*VKI+DD)
      DXZ=DJAC*(AL-AT)*VXK*VZK*VKI
      C=DJAC*(DTHK+ALP*(THK+RHOB*KD)*DHK+LAMBDA*(THK+RHOB*KD))
C
      VXK=VXK*DJAC
      VZK=VZK*DJAC
C
C ADD S SOURCE TERM BY HATA
C
      SRC=0.
C
      DO 25 IL=1,NSRC
          NSE=NSELM(IL,IRAM)
          IF(MELM.EQ.NSE) SRC=QSRC(IL,IRAM)
CDBG BY HATA
C      WRITE(6,*) 'IRAM, NSE ,SRC ',IRAM,NSE,SRC
CDBG
25      CONTINUE
C
C ADD E
C
C
C ADD S CONSTANT FOR DECAY TERM CR
C
      CR=DJAC*LAMBDA*(THK+RHOB*KD)
      SRC=DJAC*SRC
CDBG BY HATA
C      WRITE(6,*) ' SUB Q4 '

```

```

C          WRITE(6,*) ' CR = ',CR
CDBG
C
C ADD E
C
C
C ACCUMULATE THE SUMS TO EVALUATE THE MATRIX INTEGRALS QA(IQ,JQ)
C AND QB(IQ,JQ)
C
C          DO 30 IQ=1,4
C            ZW=W(IQ)
C            DO 35 JQ=1,4
C              WN=W(IQ)*N(JQ)
C              DWXDNX=DWX(IQ)*DNX(JQ)
C              DWZDNZ=DWZ(IQ)*DNZ(JQ)
C              DWXN=DWX(IQ)*N(JQ)
C              DWZN=DWZ(IQ)*N(JQ)
C              DWZDNX=DWZ(IQ)*DNX(JQ)
C              DWXDNZ=DWX(IQ)*DNZ(JQ)
C
C ADD S ARRAY FOR DECAY TERM QRR
C
C          QA(IQ,JQ)=QA(IQ,JQ)+A*WN
C 30        QB(IQ,JQ)=QB(IQ,JQ)+DWXDNX*DXX+DXZ*(DWXDNZ+DWZDNX)+DZZ*
C >          DWZDNZ + C*WN - (VXK*DWXN+VZK*DWZN)
C
C          QA(IQ,JQ)=QA(IQ,JQ)+A*WN
C          QB(IQ,JQ)=QB(IQ,JQ)+DWXDNX*DXX+DXZ*(DWXDNZ+DWZDNX)+DZZ*
C >          DWZDNZ + C*WN - (VXK*DWXN+VZK*DWZN)
C          QRR(IQ,JQ)=QRR(IQ,JQ)+WN*CR
CDBG BY HATA
C          WRITE(6,*) ' IQ,JQ,QRR(IQ,JQ) ',IQ,JQ,QRR(IQ,JQ)
CDBG
C 35        CONTINUE
C
C          QRR(IQ)=QRR(IQ)+ZW*CR
C          QRS(IQ)=QRS(IQ)+ZW*SRC
C
C 30        CONTINUE
C
C ADD E
C
C 40        CONTINUE
C
C          IF(ILUMP.NE.0) GO TO 50
C          RETURN
C
C 50        CONTINUE
C          DO 52 I=1,4
C            SUM=0.0
C            DO 52 J=1,4
C              SUM=SUM+QA(I,J)
C 51          QA(I,J)=0.0
C            QA(I,I)=SUM
C 52        CONTINUE
C          RETURN
C          END

```



```

      SUBROUTINE SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C  FUNCTION OF THE SUBROUTINE-TO COMPUTE THE VALUES OF BASIS FUNCTIONS
C  AND WEIGHTING FUNCTIONS IN LOCAL COORDINATES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 N(4)
C
      DIMENSION DNSS(4),DNTT(4),DWSS(4),DWTT(4),W(4)
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
      SM=1.0-SS
      SP=1.0+SS
      TM=1.0-TT
      TP=1.0+TT
      N(1)=0.25*SM*TM
      N(2)=0.25*SP*TM
      N(3)=0.25*SP*TP
      N(4)=0.25*SM*TP
      DNSS(1)=-0.25*TM
      DNSS(2)=0.25*TM
      DNSS(3)=0.25*TP
      DNSS(4)=-0.25*TP
      DNTT(1)=-0.25*SM
      DNTT(2)=-0.25*SP
      DNTT(3)=0.25*SP
      DNTT(4)=0.25*SM
C
      IF(IWET.EQ.0) GO TO 100
C
      W(1)=0.0625*(TP*(3.0*BETA2*(-TM)-2.0)+4.0)*(SP*(3.0*APHA1*(-SM)-
1      2.0)+4.0)
      W(2)=0.0625*(TP*(3.0*BETA1*(-TM)-2.0)+4.0)*(SP*(3.0*APHA1*SM+2.0):
      W(3)=0.0625*(TP*(3.0*BETA1*TM+2.0))*(SP*(3.0*APHA2*SM+2.0))
      W(4)=0.0625*(TP*(3.0*BETA2*TM+2.0))*(SP*(3.0*APHA2*(-SM)-2.0)+4.0)
      DWSS(1)=0.125*(TP*(3.0*BETA2*(-TM)-2.0)+4.0)*(3.0*APHA1*SS-1.0)
      DWSS(2)=-0.125*(TP*(3.0*BETA1*(-TM)-2.0)+4.0)*(3.0*APHA1*SS-1.0)
      DWSS(3)=-0.125*(TP*(3.0*BETA1*TM+2.0))*(3.0*APHA2*SS-1.0)
      DWSS(4)=0.125*(TP*(3.0*BETA2*TM+2.0))*(3.0*APHA2*SS-1.0)
      DWTT(1)=0.125*(SP*(3.0*APHA1*(-SM)-2.0)+4.0)*(3.0*BETA2*TT-1.0)
      DWTT(2)=0.125*(SP*(3.0*APHA1*SM+2.0))*(3.0*BETA1*TT-1.0)
      DWTT(3)=-0.125*(SP*(3.0*APHA2*SM+2.0))*(3.0*BETA1*TT-1.0)
      DWTT(4)=-0.125*(SP*(3.0*APHA2*(-SM)-2.0)+4.0)*(3.0*BETA2*TT-1.0)
      RETURN
C
100 DO 200 I=1,4
      W(I)=N(I)
      DWSS(I)=DNSS(I)
200 DWTT(I)=DNTT(I)
      RETURN
      END

```

```

SUBROUTINE BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
> DL,DCOSX,DCOSZ,NRSE,IS,W,MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
> MXRSEL,MAXRAM,IRAM)
C
CHANGE S DUMMY ARGUMENT OF SUBROUTINE STATEMENT BY HATA
C
SUBROUTINE BC( --- ,MAXRAM,IRAM)
C
SUBROUTINE BC(X,Z,IE,WETAB,C,R,RP,VX,VXP,VZ,VZP,NPN,BB,NPST,DP,
C > DL,DCOSX,DCOSZ,NRSE,IS,W,MAXNP,MAXEL,MAXBW,MAXBCN,MXSTNP,
C > MXRSEL)
C
CHANGE E
C
C FUNCTION OF SUBROUTINE--TO APPLY CONSTANT-CONCENTRATION DIRICHLET
C CONDITIONS AND BOTH CONSTANT AND TRANSIENT NEUMANN SURFACE
C BOUNDARY CONDITIONS.
C
IMPLICIT REAL*8 (A-H,O-Z)
C
CHANGE S COMMENT BY HATA
C
REAL*8 KD
C
CHANGE E
C
CHANGE S DIMENSION BY HATA
C
DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP),RP(MAXNP),VX(MAXNP),VXP(MAXNP)
C > VZ(MAXNP),VZP(MAXNP),NPN(MAXBCN),BB(MAXBCN),NPST(MXSTNP),
C > DP(MXSTNP),DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),NRSE(MXRSEL)
C > IS(MXRSEL,4)
CC
C DIMENSION KQ(2),VXQ(4),VZQ(4),XQ(4),ZQ(4),DFLXQ(2,2)
C
DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4,MAXRAM)
C DIMENSION C(MAXNP,MAXBW),R(MAXNP),VX(MAXNP),VXP(MAXNP),
C > RP(MAXNP,MAXRAM),NPN(MAXBCN,MAXRAM),BB(MAXBCN,MAXRAM),
C > VZ(MAXNP),VZP(MAXNP),NPST(MXSTNP,MAXRAM),DP(MXSTNP,MAXRAM),
C > DL(MXRSEL),DCOSX(MXRSEL),DCOSZ(MXRSEL),
C > NRSE(MXRSEL,MAXRAM),IS(MXRSEL,4,MAXRAM)
C
DIMENSION KQ(2),VXQ(4),VZQ(4),XQ(4),ZQ(4),DFLXQ(2,2)
C
CHANGE E
C
COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
COMMON /CONTRL/ NTI,NSTR,KSTR,KPRO,KDSKO,KSS,KVI
COMMON /PARM/ DELT,CHNG,DELMAX,TMAX
COMMON /WET/ APHA1,APHA2,BETA1,BETA2,IWET,ILUMP,IMID
C
IHALFB=(IBAND-1)/2
IHBP=IHALFB+1
C

```

```

C  APPLY CONSTANT-CONCENTRATION DIRICHLET BOUNDARY CONDITIONS
C
      IF (NBC.EQ.0) GO TO 100
      DO 90 NPP=1,NBC
C
C  MODIFY ROW NPN(NPP) OF MATRIX C(NP,IB)
C
C  CHANGE S  DIMENSION  BY HATA
C
C      NPN(I) ----> NPN(I,J)
C
C      NI=NPN(NPP)
C
C      NI=NPN(NPP,IRAM)
C
C  CHANGE E
C
      DO 10 IB=1,IBAND
      10      C(NI,IB)=0.0
            C(NI,IHBP)=1.0
C
C  CHANGE S  DIMENSION  BY HATA
C
C      BB(I)  ----> BB(I,J)
C
C      R(NI)=BB(NPP)
C
C      R(NI)=BB(NPP,IRAM)
C
C  CHANGE E
C
C  MODIFY LOAD VECTOR R(NP) FOR NON-ZERO BB(NPP)
C
C  CHANGE S  DIMENSION  BY HATA
C
C      BB(I)  ----> BB(I,J)
C
C      IF (BB(NPP).EQ.0.0) GO TO 50
C
C      IF (BB(NPP,IRAM).EQ.0.0) GO TO 50
C
C  CHANGE E
C
      DO 20 IB=1,IHALFB
      NJ=NI-IB
      IF (NJ.LT.1) GO TO 30
      JB=IHBP+IB
C  CHANGE S  DIMENSION  BY HATA
C
C      BB(I)  ----> BB(I,J)
C
C      20      R(NJ)=R(NJ)-BB(NPP)*C(NJ,JB)
C
C      20      R(NJ)=R(NJ)-BB(NPP,IRAM)*C(NJ,JB)
C

```

CHANGE E

```
C
  30   DO 40 IB=1,IHALFB
        NJ=NI+IB
        IF (NJ.GT.NNP) GO TO 50
        JB=IHBP-IB
```

CHANGE S DIMENSION BY HATA

```
C
  BB(I) ---> B(I,J)
C
  40   R(NJ)=R(NJ)-BB(NPP)*C(NJ,JB)
C
  40   R(NJ)=R(NJ)-BB(NPP,IRAM)*C(NJ,JB)
```

CHANGE E

```
C
C
C ZERO COLUMN NPN(NPP) OF MATRIX C(NP,IB)
```

```
C
  50   DO 60 IB=1,IHALFB
        NJ=NI-IB
        IF (NJ.LT.1) GO TO 70
        JB=IHBP+IB
  60   C(NJ,JB)=0.0
  70   DO 80 IB=1,IHALFB
        NJ=NI+IB
        IF (NJ.GT.NNP) GO TO 90
        JB=IHBP-IB
  80   C(NJ,JB)=0.0
  90   CONTINUE
```

```
C
  100 IF(NRSEL.LE.0) GO TO 140
```

```
C
C ENTER SEEPAGE TERMS IN MATRIX C(NP,IB)
```

```
C
  W1=W
  W2=1.-W
  IF (KSS.NE.0) GO TO 110
  W1=1.
  W2=0.
```

```
  110 DO 130 MP=1,NRSEL
```

CHANGE S DIMENSION BY HATA

```
C
  NRSE(I) ---> NRSE(I,J)
C
  WETAB(I,J) ---> WETAB(I,J,K)
C
  IS(I,J) ---> IS(I,J,K)
```

```
C
  M=NRSE(MP)
C
  APHA1=WETAB(M,1)
C
  APHA2=WETAB(M,3)
C
  BETA1=WETAB(M,2)
C
  BETA2=WETAB(M,4)
C
  NI=IS(MP,1)
C
  NJ=IS(MP,2)
C
  KQ(1)=IS(MP,3)
C
  KQ(2)=IS(MP,4)
```

```

C
      M=NRSE(MP,IRAM)
      APHA1=WETAB(M,1,IRAM)
      APHA2=WETAB(M,3,IRAM)
      BETA1=WETAB(M,2,IRAM)
      BETA2=WETAB(M,4,IRAM)
      NI=IS(MP,1,IRAM)
      NJ=IS(MP,2,IRAM)
      KQ(1)=IS(MP,3,IRAM)
      KQ(2)=IS(MP,4,IRAM)
C
CHANGE E
C
      DO 120 IQ=1,4
        NP=IE(M,IQ)
        XQ(IQ)=X(NP)
        ZQ(IQ)=Z(NP)
        VXQ(IQ)=VX(NP)*W1+VXP(NP)*W2
        VZQ(IQ)=VZ(NP)*W1+VZP(NP)*W2
120    CONTINUE
C
      CALL Q4SP(DFLXQ,KQ,DL(MP),DCOSX(MP),DCOSZ(MP),XQ,ZQ,VXQ,VZQ)
C
      IB=IHBP
      JB=IHBP+(NJ-NI)
C
      IF(IMID.NE.0) GO TO 121
C
      C(NI,IB)=C(NI,IB)+W1*DFLXQ(1,1)
      C(NI,JB)=C(NI,JB)+W1*DFLXQ(1,2)
C
CHANGE S  DIMENSION  BY HATA
C
      RP(I)  ---> RP(I,J)
C
      R(NI)=R(NI)-W2*(DFLXQ(1,1)*RP(NI)+DFLXQ(1,2)*RP(NJ))
C
      R(NI)=R(NI)-W2*(DFLXQ(1,1)*RP(NI,IRAM)+DFLXQ(1,2)*RP(NJ,IRAM))
C
CHANGE E
C
      GO TO 123
121    C(NI,IB)=C(NI,IB) + DFLXQ(1,1)
      C(NI,JB)=C(NI,JB) + DFLXQ(1,2)
C
123    JB=IHBP
      IB=IHBP+(NI-NJ)
      IF(IMID.NE.0) GO TO 125
C
      C(NJ,JB)=C(NJ,JB)+W1*DFLXQ(2,2)
      C(NJ,IB)=C(NJ,IB)+W1*DFLXQ(2,1)
C
CHANGE S  DIMENSION  BY HATA
C
      RP(I)  ---> RP(I,J)
C
      R(NJ)=R(NJ)-W2*(DFLXQ(2,1)*RP(NI)+DFLXQ(2,2)*RP(NJ))
C

```

```

                R(NJ)=R(NJ)-W2*(DFLXQ(2,1)*RP(NI,IRAM)+DFLXQ(2,2)*RP(NJ,IRAM)
C
CHANGE E
C
                GO TO 130
125          C(NJ,JB)=C(NJ,JB) + DFLXQ(2,2)
            C(NJ,IB)=C(NJ,IB) + DFLXQ(2,1)
130          CONTINUE
C
140 IF (NST.LE.0) GO TO 160
C
C  MODIFY LOAD VECTOR FOR SURFACE TERMS OF THE FORM DR/DN=C
C
CHANGE S  DIMENSION  BY HATA
C
C  NPST(I) ---> NPST(I,J)
C  DP(I)   ---> DP(I,J)
C
C  DO 150 NPP=1,NSTN
C      NP=NPST(NPP)
C 150  R(NP)=R(NP)+DP(NPP)
C
C      DO 150 NPP=1,NSTN(IRAM)
C          NP=NPST(NPP,IRAM)
150    R(NP)=R(NP)+DP(NPP,IRAM)
C
CHANGE E
C
160 RETURN
    END

```

```

SUBROUTINE Q4SP(DFLXQ,KQ,DL,DCOSXQ,DCOSZQ,XQ,ZQ,VXQ,VZQ)
C
C
C FUNCTION OF SUBROUTINE--TO EVALUATE THE SEEPAGE-FLUX INTEGRALS
C ALONG THE BOUNDARY LINE EXTENDING FROM NODE LQ TO NODE MQ.
C
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   REAL*8 N(4)
C
C   DIMENSION W(4),DNX(4),DNZ(4),DWX(4),DWZ(4)
C   DIMENSION PJAB(2,2),DNSS(4),DNNT(4),DWSS(4),DWTT(4)
C   DIMENSION S(4,4),T(4,4),DFLXQ(2,2),KQ(2),SSA(2),TTA(2),
C > XQ(4),ZQ(4),VXQ(4),VZQ(4)
C
C   DATA S/0.DO,-.57735D0,0.DO,-1.DO,.57735D0,0.DO,1.DO,0.DO, 0.DO,
C > 1.DO,0.DO,-.57735D0,-1.DO,0.DO,.57735D0,0.DO/, T/0.DO,-1.DO,0.DO
C > -.57735D0,-1.DO,0.DO,-.57735D0,0.DO,0.DO, .57735D0,0.DO,1.DO,
C > .57735D0,0.DO,1.DO,0.DO/
C
C INITIALIZE NODAL COMPONENTS OF LINE INTEGRAL
C
C   DO 10 JQ=1,2
C     DO 10 IQ=1,2
C 10     DFLXQ(IQ,JQ)=0.
C
C DETERMINE LOCAL COORDINATES OF GAUSS-INTEGRATION POINTS KG
C
C   LQ=KQ(1)
C   MQ=KQ(2)
C   SSA(1)=S(LQ,MQ)
C   TTA(1)=T(LQ,MQ)
C   SSA(2)=S(MQ,LQ)
C   TTA(2)=T(MQ,LQ)
C   DO 40 KG=1,2
C     SS = SSA(KG)
C     TT = TTA(KG)
C
C CALCULATE VALUES OF BASIS AND WEIGHTING FUNCTIONS N(IQ) AND W(IQ)
C AND THEIR DERIVATIVES DWX DNX AND DWZ DNZ WITH RESPECT TO X AND Z
C RESPECTIVELY, AT THE GAUSS POINT KG
C
C   CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWTT,SS,TT)
C
C COMPUTE THE INVERSE TRANSFORMATION MATRIX I.E. THE INVERSE JACOBIAN
C PJAB(1,1)=(DZ/DT)/DJAC, PJAB(1,2)=-(DZ/DS)/DJAC,
C PJAB(2,1)=-(DX/DT)/DJAC, PJAB(2,2)=(DX/DS)/DJAC, WHERE
C DJAC IS THE DETERMINANT OF THE JACOBIAN
C
C   DO 11 I=1,2
C     DO 11 J=1,2
C       PJAB(I,J)=0.0
C 11     CONTINUE
C
C   DO 16 I=1,4
C     PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNNT(I)
C     PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)

```

```

        PJAB(2,1)=PJAB(2,1) - XQ(I)*DN TT(I)
        PJAB(2,2)=PJAB(2,2) + XQ(I)*DN SS(I)
16      CONTINUE
C
C
C      COMPUTE THE DETERMINANT OF THE JACOBIAN
C
        DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
        DJACI=1.000/DJAC
C
        DO 17 I=1,2
          DO 17 J=1,2
17      PJAB(I,J)=PJAB(I,J)*DJACI
C
        DO 18 I=1,4
          DNX(I)=PJAB(1,1)*DN SS(I) + PJAB(1,2)*DN TT(I)
          DNZ(I)=PJAB(2,1)*DN SS(I) + PJAB(2,2)*DN TT(I)
          DWX(I)=PJAB(1,1)*DW SS(I) + PJAB(1,2)*DW TT(I)
18      DWZ(I)=PJAB(2,1)*DW SS(I) + PJAB(2,2)*DW TT(I)
C
C      INTERPOLATE WITH FUNCTIONS N(IQ), DNX(IQ), DNZ(IQ) TO OBTAIN
C      VALUES OF DARCY VELOCITIES VXQP AND VZQP
C
        VVK=0.
        VVK=0.
        DO 20 IQ=1,4
          VVK=VVK+VXQ(IQ)*N(IQ)
20      VVK=VVK+VZQ(IQ)*N(IQ)
C
C      EVALUATE THE NORMAL DARCY VELOCITY AT THE GAUSS POINT AND ACCUMULATE
C      THE INTEGRAL SUMS
C
        VNK=VVK*DCOSXQ+VZK*DCOSZQ
        IF(VNK.LE.0.0) GO TO 40
        DO 30 JQ=1,2
          MQ=KQ(JQ)
          DO 30 IQ=1,2
            LQ=KQ(IQ)
30      DFLXQ(IQ,JQ)=DFLXQ(IQ,JQ)+W(LQ)*VVK*N(MQ)
40      CONTINUE
        DO 50 JQ=1,2
          DO 50 IQ=1,2
50      DFLXQ(IQ,JQ)=.5*DL*DFLXQ(IQ,JQ)
        RETURN
        END

```



```

SUBROUTINE SOLVE(KKK,C,R,NNP,IHALFB,MAXNP,MAXBW)
C
C
C FUNCTION OF SUBROUTINE--TO SOLVE THE MATRIX EQUATION CX = R,
C RETURNING THE SOLUTION X IN R. IT IS ASSUMED THAT THE ARRAY C(NP,IB)
C CONTAINS THE FULL BAND OF AN ASYMMETRIC MATRIX.
C
C
C   IMPLICIT REAL*8(A-H,O-Z)
C
C   DIMENSION C(MAXNP,MAXBW),R(MAXNP)
C
C   IHBP=IHALFB+1
C
C IF KKK = 1, THEN TRIANGULARIZE THE BAND MATRIX C(NP,IB), BUT
C IF KKK = 2, THEN SIMPLY SOLVE WITH THE RIGHT-HAND SIDE R(NP)
C
C   IF (KKK.EQ.2) GO TO 50
C
C TRIANGULARIZE MATRIX C(NP,IB)
C
C   NU=NNP-IHALFB
C   DO 20 NI=1,NU
C     PIVOTI=1./C(NI,IHBP)
C     NJ=NI+1
C     IB=IHBP
C     NK=NI+IHALFB
C     DO 10 NL=NJ,NK
C       IB=IB-1
C       A=-C(NL,IB)*PIVOTI
C       C(NL,IB)=A
C       JB=IB+1
C       KB=IB+IHALFB
C       LB=IHBP-IB
C       DO 10 MB=JB,KB
C         NB=LB+MB
C         C(NL,MB)=C(NL,MB)+A*C(NI,NB)
10
20     CONTINUE
C     NR=NU+1
C     NU=NNP-1
C     NK=NNP
C     DO 40 NI=NR,NU
C       PIVOTI=1./C(NI,IHBP)
C       NJ=NI+1
C       IB=IHBP
C       DO 30 NL=NJ,NK
C         IB=IB-1
C         A=-C(NL,IB)*PIVOTI
C         C(NL,IB)=A
C         JB=IB+1
C         KB=IB+IHALFB
C         LB=IHBP-IB
C         DO 30 MB=JB,KB
C           NB=LB+MB
30           C(NL,MB)=C(NL,MB)+A*C(NI,NB)
40     CONTINUE
C   RETURN
C

```

```

C  MODIFY LOAD VECTOR R(NP)
C
50  NU=NNP+1
    IBAND=2*IHALFB+1
    DO 70 NI=2,IHBP
        IB=IHBP-NI+1
        NJ=1
        SUM=0.0
        DO 60 JB=IB,IHALFB
            SUM=SUM+C(NI,JB)*R(NJ)
60      NJ=NJ+1
70      R(NI)=R(NI)+SUM
        IB=1
        NL=IHBP+1
        DO 90 NI=NL,NNP
            NJ=NI-IHBP+1
            SUM=0.0
            DO 80 JB=IB,IHALFB
                SUM=SUM+C(NI,JB)*R(NJ)
80      NJ=NJ+1
90      R(NI)=R(NI)+SUM
C
C  BACK SOLVE
C
    R(NNP)=R(NNP)/C(NNP,IHBP)
    DO 110 IB=2,IHBP
        NI=NU-IB
        NJ=NI
        MB=IHALFB+IB
        SUM=0.0
        DO 100 JB=NL,MB
            NJ=NJ+1
100     SUM=SUM+C(NI,JB)*R(NJ)
110     R(NI)=(R(NI)-SUM)/C(NI,IHBP)
        MB=IBAND
        DO 130 IB=NL,NNP
            NI=NU-IB
            NJ=NI
            SUM=0.0
            DO 120 JB=NL,MB
                NJ=NJ+1
120     SUM=SUM+C(NI,JB)*R(NJ)
130     R(NI)=(R(NI)-SUM)/C(NI,IHBP)
    RETURN
    END

```

```

      SUBROUTINE SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,
> NBE,ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,
> MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
> DELT,DH, KFLOW,PROPR,MAXRAM,IRAM)
C
C ADD S DUMMY ARGUMENT OF SUBROUTINE BY HATA
C
C SUBROUTINE SFLOW( --- ,PROPR,MAXRAM,IRAM)
C
C SUBROUTINE SFLOW(X,Z,IE,WETAB,R,FX,FZ,TH,PROP,DLB,DCOSXB,DCOSZB,
C > NBE,ISB,NPB,BFLX,BFLXP,NPRS,NPST,NPN,FRATE,FLOW,TFLOW,MAXNP,
C > MAXEL,MAXBNP,MAXBEL,MXRSNP,MXRSEL,MXSTNP,MAXBCN,MAXMAT,MXMPPM,
C > DELT,DH, KFLOW)
C
C ADD E
C
C FUNCTION OF SUBROUTINE--TO COMPUTE BOUNDARY FLUXES, FLOW RATES,
C INCREMENTAL FLOWS OCCURRING DURING TIME DELT, TOTAL FLOWS SINCE
C TIME ZERO, AND THE CHANGE IN MOISTURE CONTENT FOR THE ENTIRE
C SYSTEM DURING TIME DELT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KD,LAMBDA
C
C CHANGE S DIMENSION BY HATA
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4)
      DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP),DH(MAXNP),TH(MAXEL,4)
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL),BFLX(MAXBNP),
C > BFLXP(MAXBNP),NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
      DIMENSION NPRS(MXRSNP),NPST(MXSTNP),NPN(MAXBCN)
      DIMENSION PROP(MAXMAT,MXMPPM)
      DIMENSION FRATE(10),FLOW(10),TFLOW(10)
CC
      DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THRQ(4),THQ(4)
C
      DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),WETAB(MAXEL,4,MAXRAM)
      DIMENSION FX(MAXNP),FZ(MAXNP),DH(MAXNP),TH(MAXEL,4)
      DIMENSION DLB(MAXBEL),DCOSXB(MAXBEL),DCOSZB(MAXBEL)
      DIMENSION BFLX(MAXBNP,MAXRAM),BFLXP(MAXBNP,MAXRAM)
      DIMENSION NBE(MAXBEL),ISB(MAXBEL,4),NPB(MAXBNP)
      DIMENSION NPRS(MXRSNP,MAXRAM),NPST(MXSTNP,MAXRAM)
      DIMENSION NPN(MAXBCN,MAXRAM),R(MAXNP)
      DIMENSION PROP(MAXMAT,MXMPPM),PROPR(MAXMAT,MXMPPM,MAXRAM)
      DIMENSION FRATE(10),FLOW(10),TFLOW(10,MAXRAM)
C
      DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THRQ(4),THQ(4)
      DIMENSION QR(5),QD(5),QL(5),QDH(5),QRP(5),QDP(5),QLP(5),QDHP(5)
C
C CHANGE E
C
      COMMON /GEOM/ SNFE,CSFE,NNP,NEL,IBAND
      COMMON /BDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
      COMMON /WET/ APHA1,APHA2,BETA1,BETA2, IWET,ILUMP,IMID
C
C CHANGE S DATA STATEMENT BY HATA
C
      DATA QR,QD,QL,QDH /0.0D0,0.0D0,0.0D0,0.0D0/

```

```

C
  DATA QR /0.000,0.000,0.000,0.000,0.000/
  DATA QD /0.000,0.000,0.000,0.000,0.000/
  DATA QL /0.000,0.000,0.000,0.000,0.000/
  DATA QDH /0.000,0.000,0.000,0.000,0.000/
C
CHANGE E
C
C
C ADD S BY HATA
C
  MXRSEL=MXRSEL
C
C ADD E
C
C
C -----
CDBG BY HATA
C
C WRITE(6,*) ' DEBUG SFLOW '
C WRITE(6,*) ' 1. QR,QD,QL ',QR,QD,QL
C
CDBG
C
C ADD S ASSIGN STATEMENT BY HATA
C
C QR=0.0
C QD=0.0
C QL=0.0
C
C ADD E
C
CDBG BY HATA
C
C WRITE(6,*) ' DEBUG SFLOW '
C WRITE(6,*) ' 1.1 QR,QD,QL ',QR,QD,QL
C
CDBG
  IF(KFLOW.GT.0) GO TO 9
C
CHANGE S ASSIGNMENT STATEMENT BY HATA
C
C QR=0.0
C QD=0.0
C QL=0.0
C
  QR(IRAM)=0.0
  QD(IRAM)=0.0
  QL(IRAM)=0.0
C
CHANGE E
C
  9 CONTINUE
C -----
C
C CALCULATE NODAL FLOW RATES
C
C ADD S IF STATEMENT BY HATA

```

```

C
C      IF(IRAM.EQ.1) GO TO 200
C      GO TO 210
C 200 CONTINUE
C
C ADD E
C
CHANGE S  DIMENSION  BY HATA
C
C      BFLX(I)  ---> BFLX(I,J)
C      BFLXP(I) ---> BFLXP(I,J)
C
C      DO 10 NP=1,NBN
C      BFLXP(NP)=BFLX(NP)
C 10    BFLX(NP)=0.
C
C      DO 10 NP=1,NBN
C      BFLXP(NP,IRAM)=BFLX(NP,IRAM)
C 10    BFLX(NP,IRAM)=0.
C
CHANGE E
C
C      DO 30 MP=1,NBEL
C      M=NBE(MP)
C      NI=ISB(MP,1)
C      NJ=ISB(MP,2)
C      DO 20 I=1,NBN
C      IJ=NPB(I)
C      IF(IJ.NE.NI) GO TO 20
C      NII=I
C      GO TO 22
C 20    CONTINUE
C 22    DO 25 J=1,NBN
C      IJ=NPB(J)
C      IF(IJ.NE.NJ) GO TO 25
C      NJJ=J
C      GO TO 27
C 25    CONTINUE
C 27    CONTINUE
C      FNI=(FX(NI)*DCOSXB(MP)+FZ(NI)*DCOSZB(MP))*DLB(MP)
C      FNJ=(FX(NJ)*DCOSXB(MP)+FZ(NJ)*DCOSZB(MP))*DLB(MP)
C
CHANGE S  DIMENSION  BY HATA
C
C      BFLX(I)  ---> BFLX(I,J)
C      BFLXP(I) ---> BFLXP(I,J)
C
C      BFLX(NII)=BFLX(NII)+FNI/3.0+FNJ/6.0
C      BFLX(NJJ)=BFLX(NJJ)+FNJ/3.0+FNI/6.0
C
C      BFLX(NII,IRAM)=BFLX(NII,IRAM)+FNI/3.0+FNJ/6.0
C      BFLX(NJJ,IRAM)=BFLX(NJJ,IRAM)+FNJ/3.0+FNI/6.0
C
CHANGE E
C
CDBG  BY HATA
C
C      WRITE(6,*) '  DEBUG SFLOW '

```

```

C          WRITE(6,*) '      BFLX      '
C          WRITE(6,*) ' FNI,FNJ,BFLX(NII),BFLX(NJJ) ', FNI,FNJ,BFLX(NII),
C      >          BFLX(NJJ)
C
CDBG
C      30      CONTINUE
C
C ADD S      CONTINUE STATEMENT BY HATA
C
C 210      CONTINUE
C
C ADD E
C
C
C DETERMINE FLOWS AND FLOW RATES THRU THE VARIOUS
C TYPES OF BOUNDARY NODES, STARTING WITH THE
C NET FLOWS THROUGH ALL BOUNDARY NODES
C
C          S=0.
C          SP=0.
C
C CHANGE S  DIMENSION BY HATA
C
C          BFLX(I) ----> BFLX(I,J)
C          BFLXP(I) ----> BFLXP(I,J)
C
C      DO 40 NP=1,NBN
C          S=S+BFLX(NP)
C 40      SP=SP+BFLXP(NP)
C
C          DO 40 NP=1,NBN
C              S=S+BFLX(NP,IRAM)
C          40      SP=SP+BFLXP(NP,IRAM)
C
C CHANGE E
C
CDBG BY HATA
C
C          WRITE(6,*) ' NO.5  S, SP ',S,SP
C
CDBG
C          FRATE(5)=S
C          FLOW(5)=.5*(S+SP)*DELT
C
C CONSTANT DIRICHLET BOUNDARY NODES
C
C          FRATE(1)=0.
C          FLOW(1)=0.
C          IF (NBC.LE.0) GO TO 60
C          S=0.
C          SP=0.
C          DO 50 NPP=1,NBC
C
C CHANGE S  DIMENSION BY HATA
C
C          NPN(I) ----> NPN(I,J)
C
C          NP=NPN(NPP)

```

```

C
      NP=NPN(NPP,IRAM)
C
CHANGE E
C
      DO 45 I=1,NBN
        IJ=NPB(I)
        IF(IJ.NE.NP) GO TO 45
        NII=I
        GO TO 46
45      CONTINUE
46      CONTINUE
C
CHANGE S DIMENSION BY HATA
C
      BFLX(I) ---> BFLX(I,J)
      BFLXP(I) ---> BFLXP(I,J)
C
      S=S+BFLX(NII)
C 50      SP=SP+BFLXP(NII)
C
      S=S+BFLX(NII,IRAM)
C 50      SP=SP+BFLXP(NII,IRAM)
C
CHANGE E
C
CDBG BY HATA
C
      WRITE(6,*) ' DEBUG NO.1 S ,SP ',S,SP
C
CDBG
      FRATE(1)=S
      FLOW(1)=.5*(S+SP)*DELT
C
C CONSTANT NEUMANN BOUNDARY NODES
C
60 FRATE(2)=0.
      FLOW(2)=0.
      IF (NST.LE.0) GO TO 80
      S=0.
      SP=0.
      DO 70 NPP=1,NSTN(IRAM)
C
CHANGE S DIMENSION BY HATA
C
      NPST(I) ---> NPST(I,J)
C
      NP=NPST(NPP)
C
      NP=NPST(NPP,IRAM)
C
CHANGE E
C
      DO 65 I=1,NBN
        IJ=NPB(I)
        IF(IJ.NE.NP) GO TO 65
        NII=I
        GO TO 66

```

```

        65      CONTINUE
        66      CONTINUE
C
CHANGE S  DIMENSION  BY HATA
C
C          BFLX(I)  ---> BFLX(I,J)
C          BFLXP(I) ---> BFLXP(I,J)
C
C          S=S+BFLX(NII)
C  70      SP=SP+BFLXP(NII)
C
C          S=S+BFLX(NII,IRAM)
C  70      SP=SP+BFLXP(NII,IRAM)
C
CHANGE E
C
CDBG  BY HATA
C
C      WRITE(6,*) '  DEBUG  NO.2 ,S,SP ',S,SP
C
CDBG
      FRATE(2)=S
      FLOW(2)=.5*(S+SP)*DELT
C
C  TRANSIENT SEEPAGE BOUNDARY NODES
C
C  80 FRATE(3)=0.
C     FLOW(3)=0.
C     IF(NRSEL.LE.0) GO TO 100
C     S=0.
C     SP=0.
C     DO 90 NPP=1,NRSN
C
CHANGE S  DIMENSION  BY HATA
C
C          NPRS(I)  ---> NPRS(I,J)
C
C          NP=NPRS(NPP)
C
C          NP=NPRS(NPP,IRAM)
C
CHANGE E
C
      DO 85 I=1,NBN
C          IJ=NPB(I)
C          IF(IJ.NE.NP) GO TO 85
C          NII=I
C          GO TO 86
C  85      CONTINUE
C  86      CONTINUE
C
CHANGE S  DIMENSION  BY HATA
C
C          BFLX(I)  ---> BFLX(I,J)
C          BFLXP(I) ---> BFLXP(I,J)
C
C          S=S+BFLX(NII)
C  90      SP=SP+BFLXP(NII)

```



```

C
      S=S+BFLX(NII,IRAM)
      90    SP=SP+BFLXP(NII,IRAM)
C
CHANGE E
C
CDBG BY HATA
C
C    WRITE(6,*) ' DEBUG NO.3 S, SP ',S,SP
C
CDBG
      FRATE(3)=S
      FLOW(3)=.5*(S+SP)*DELT
C
C NUMERICAL FLOW THROUGH UNSPECIFIED BOUNDARY NODES
C
      100 S=0.
          SP=0.
          DO 110 I=1,3
              S=S+FRATE(I)
          110    SP=SP+FLOW(I)
CDBG BY HATA
C
C    WRITE(6,*) ' DEBUG NO.4 S, SP ',S,SP
C
CDBG
      FRATE(4)=FRATE(5)-S
      FLOW(4)=FLOW(5)-SP
C
C CALCULATE THE INCREASES IN THE INTEGRATED MATERIAL CONTENTS FOR THE
C FLUID AND THE SOLID PHASES AND DETERMINE LOSSES DUE TO RADIOACTIVE
C DECAY
C
CDBG BY HATA
C
C    WRITE(6,*) ' DEBUG '
C    WRITE(6,*) ' 2. QR,QD,QL,QDH ',QR,QD,QL,QDH
C    WRITE(6,*) ' QRP,QDP,QLP,QDHP ',QRP,QDP,QLP,QDHP
C
CDBG
CHANGE S ASSIGN STATEMENT BY HATA
C
C    QRP=QR
C    QDP=QD
C    QLP=QL
C    QDHP=QDH
C    QR=0.
C    QD=0.
C    QL=0.
C    QDH=0.0
C
      QRP(IRAM)=QR(IRAM)
      QDP(IRAM)=QD(IRAM)
      QLP(IRAM)=QL(IRAM)
      QDHP(IRAM)=QDH(IRAM)
      QR(IRAM)=0.
      QD(IRAM)=0.

```

```

        QL(IRAM)=0.
        QDH(IRAM)=0.0
C
CHANGE E
C
        DO 130 M=1,NEL
C
CHANGE S  DIMENSION  BY HATA
C
        WETAB(I,J) ----> WETAB(I,J,K)
C
        APHA1=WETAB(M,1)
        APHA2=WETAB(M,3)
        BETA1=WETAB(M,2)
        BETA2=WETAB(M,4)
C
        APHA1=WETAB(M,1,IRAM)
        APHA2=WETAB(M,3,IRAM)
        BETA1=WETAB(M,2,IRAM)
        BETA2=WETAB(M,4,IRAM)
C
CHANGE E
C
        MTYP=IE(M,5)
C
CHANGE S  VARIABLE FUNCTION  BY HATA
C
        PROP(I,1) ----> PROPR(I,1,K)
        PROP(I,2) ----> PROP(I,1)
        PROP(I,5) ----> PROPR(I,2,K)
        PROP(I,6) ----> PROP(I,4)
        PROP(I,7) ----> PROP(I,5)
C
        KD=PROP(MTYP,1)
        RHOB=PROP(MTYP,2)
        LAMBDA=PROP(MTYP,5)
        POR=PROP(MTYP,6)
        ALP=PROP(MTYP,7)
C
        KD=PROPR(MTYP,1,IRAM)
        RHOB=PROP(MTYP,1)
        LAMBDA=PROPR(MTYP,2,IRAM)
        POR=PROP(MTYP,4)
        ALP=PROP(MTYP,5)
C
CHANGE E
C
CDBG  BY HATA
C
        WRITE(6,*) '  DEBUG  SFLOW  '
        WRITE(6,*) '  IRAM  ',IRAM
        WRITE(6,*) '  KD,RHOB,LAMBDA,POR,ALP  ',KD,RHOB,LAMBDA,POR,ALP
C
CDBG
        DO 120 IQ=1,4
            NP=IE(M,IQ)
            XQ(IQ)=X(NP)
            ZQ(IQ)=Z(NP)

```

```

                DHQ(IQ)=DH(NP)
                THQ(IQ)=TH(M,IQ)
                RQ(IQ)=R(NP)
120            THRQ(IQ)=THQ(IQ)*RQ(IQ)
C
C
                CALL Q4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THRQ,THQ,RHOB,KD)
C
CHANGE S      ASSIGNMENT STATEMENT  BY HATA
C
                QR=QR+QRM
                QD=QD+QDM*RHOB*KD
                QLM=QRM+QDM
                QL=QL+LAMBDA*QLM
                QDH=QDH+ALP*QDHM
C
                QR(IRAM)=QR(IRAM)+QRM
                QD(IRAM)=QD(IRAM)+QDM*RHOB*KD
                QLM=QRM+QDM
                QL(IRAM)=QL(IRAM)+LAMBDA*QLM
                QDH(IRAM)=QDH(IRAM)+ALP*QDHM
130            CONTINUE
C
CDBG BY HATA
C
                WRITE(6,*) '  DEBUG  3.  QR, QRP ',QR,QRP
                WRITE(6,*) '  DEBUG  4.  QR, QRP ',QD,QDP
                WRITE(6,*) '  DEBUG  5.  QL, QLP ',QL,QLP
                WRITE(6,*) '  DEBUG  6.  QDH      ',QDH
C
CDBG
C
CHANGE S      ASSIGNMENT STATEMENT  BY HATA
C
                LOW(6)=QR-QRP
                FRATE(6)=FLOW(6)/DELT
                FLOW(7)=QD-QDP
                FRATE(7)=FLOW(7)/DELT
                FRATE(8)=.5*(QL+QLP)
C -----
C                IF(KFLOW .LE.0) FRATE(8)=QL
C -----
C
                FLOW(6)=QR(IRAM)-QRP(IRAM)
                FRATE(6)=FLOW(6)/DELT
                FLOW(7)=QD(IRAM)-QDP(IRAM)
                FRATE(7)=FLOW(7)/DELT
                FRATE(8)=.5*(QL(IRAM)+QLP(IRAM))
C -----
C                IF(KFLOW .LE.0) FRATE(8)=QL(IRAM)
C -----
C
CHANGE E
C
                FLOW(8)=DELT*FRATE(8)
C
CHANGE S      ASSIGNMENT STATEMENT  BY HATA
C

```

```
C      FRATE(9)=QDH
C
C      FRATE(9)=QDH(IRAM)
C
CHANGE E
C
C      FLOW(9)=DELT*FRATE(9)
C
C
CHANGE S  DIMENSION  BY HATA
C
C      TFLOW(I) ----> TFLOW(I,J)
C
C      DO 140 I=1,9
C 140    TFLOW(I)=TFLOW(I)+FLOW(I)
C
C      DO 140 I=1,9
C 140    TFLOW(I,IRAM)=TFLOW(I,IRAM)+FLOW(I)
C
CHANGE E
C
C
C ADD S  ZERO CLEAR STATEMENT  BY HATA
C
C      QRP=0.0
C      QDP=0.0
C      QLP=0.0
C      QDHP=0.0
C
C ADD E
C
C      RETURN
C      END
```

```

SUBROUTINE Q4R(QRM,QDM,QDHM,XQ,ZQ,RQ,DHQ,THRQ,THQ,RHOB,KD)
C
C
C FUNCTION OF SUBROUTINE--TO EVALUATE THE CONCENTRATION INTEGRAL
C OVER THE AREA OF ONE ELEMENT.
C
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   REAL*8 N(4),KD
C
C   DIMENSION XQ(4),ZQ(4),RQ(4),DHQ(4),THRQ(4),THQ(4)
C   DIMENSION S(4),T(4)
C   DIMENSION W(4)
C   DIMENSION PJAB(2,2),DNSS(4),DNNT(4),DWSS(4),DWTT(4)
C
C   DATA P / 0.577350269189626 /, S / -1.0D+00, 1.0D+00, 1.0D+00, -
C > 1.0D+00 /, T / -1.0D+00, -1.0D+00, 1.0D+00, 1.0D+00 /
C
C EVALUATE QUANTITIES FOR USE IN THE JACOBIAN DJAC, BELOW, NECESSARY
C FOR TRANSFORMATION FROM GLOBAL TO LOCAL COORDINATES
C
C   QRM=0.
C   QDM=0.0
C   QDHM=0.0
C   DO 20 KG=1,4
C
C DETERMINE LOCAL COORDINATES (SS,TT) OF GAUSS-INTEGRATION POINT KG
C
C   SS = P*S(KG)
C   TT = P*T(KG)
C
C EVALUATE THE JACOBIAN DJAC
C
C   CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWTT,SS,TT)
C
C   DO 11 I=1,2
C     DO 11 J=1,2
C       PJAB(I,J)=0.0
C 11   CONTINUE
C   DO 16 I=1,4
C     PJAB(1,1)=PJAB(1,1) + ZQ(I)*DNNT(I)
C     PJAB(1,2)=PJAB(1,2) - ZQ(I)*DNSS(I)
C     PJAB(2,1)=PJAB(2,1) - XQ(I)*DNNT(I)
C     PJAB(2,2)=PJAB(2,2) + XQ(I)*DNSS(I)
C 16   CONTINUE
C     DJAC=PJAB(2,2)*PJAB(1,1) - PJAB(1,2)*PJAB(2,1)
C
C INTERPOLATE TO OBTAIN THE CONCENTRATION RQP AT THE GAUSS POINT KG
C
C   RQP=0.
C   DHQP=0.0
C   THRQP=0.0
C   THQP=0.0
C   DO 10 IQ=1,4
C     RQP=RQP+RQ(IQ)*N(IQ)
C     DHQP=DHQP+DHQ(IQ)*N(IQ)
C     THQP=THQP+THQ(IQ)*N(IQ)
C 10   THRQP=THRQP+THRQ(IQ)*N(IQ)

```

```
C  
C ACCUMULATE THE SUM TO EVALUATE THE INTEGRAL QRM  
C  
      QRM=QRM+THRQP*DJAC  
      QDM=QDM+RQP*DJAC  
      QDHM=QDHM+(THQP+RHOB*KD)*DHQP*RQP*DJAC  
20    CONTINUE  
      RETURN  
      END
```

```

SUBROUTINE PRINTT(R,FX,FZ, FRATE,FLOW,TFLOW, TIME,DELT,KPR,KOUT,
> KDIAG,MAXNP,NNP,IBAND,ITIM,TGRATE,TGFLOW,TGMOUN,IRAM,MAXRAM,NSUR
> NRAM)
C
C ADD S DUMMY ARGUMENT BY HATA
C
C SUBROUTINE PRINTT( --- ,IRAM,MAXRAM,NSUR)
C
C SUBROUTINE PRINTT(R,FX,FZ, FRATE,FLOW,TFLOW, TIME,DELT,KPR,KOUT,
> KDIAG,MAXNP,NNP,IBAND, ITIM)
C
C ADD E
C
C FUNCTION OF SUBROUTINE--TO OUTPUT FLOWS, CONCENTRATIONS, MATERIAL
C FLUXES, WATER CONTENTS, DARCY VELOCITIES, PRESSURE HEADS, AND
C TOTAL HEADS AS SPECIFIED BY THE PARAMETER KPR.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C DIMENSION R(MAXNP),FX(MAXNP),FZ(MAXNP)
C
C CHANGE S DIMENSION BY HATA
C
C TFLOW(I) ---> TFLOW(I,J)
C
C DIMENSION FRATE(10),FLOW(10),TFLOW(10)
C
C DIMENSION FRATE(10),FLOW(10),TFLOW(10,MAXRAM),TGMOUN(MAXRAM),
> TGFLOW(MAXRAM)
C
C CHANGE E
C
C ADD S BY HATA
C
C IF(NSUR.GT.O.AND.IRAM.EQ.NRAM)
> WRITE(7) TIME,(TGFLOW(II),II=1,NRAM)
C
C ADD E
C
C IF (KPR.EQ.O) RETURN
C
C PRINT FLOW SYSTEM PARAMETERS AND MASS BALANCE
C
C KDIAG=KDIAG+1
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10600) IRAM,KDIAG, ---
C WRITE(6,10500) (FRATE(I), ---
C
C WRITE(6,10600)KDIAG,TIME,DELT,ITIM
C WRITE(6,10500)(FRATE(I),FLOW(I),TFLOW(I),I=1,9)
C
C WRITE(6,10600) IRAM,KDIAG,TIME,DELT,ITIM
C WRITE(6,10500) (FRATE(I),FLOW(I),TFLOW(I,IRAM),I=1,9)
C IF(NSUR.GT.O)
> WRITE(6,10510) TGRATE,TGFLOW(IRAM),TGMOUN(IRAM)

```

```

C ADD E
C
C 10 IF (KPR.EQ.1) RETURN
C
C PRINT CONCENTRATIONS
C
C KOUT=KOUT+1
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10000) IRAM,KOUT, ---
C
C WRITE(6,10000) KOUT,TIME,DELT,IBAND, ITIM
C
C WRITE(6,10000) IRAM,KOUT,TIME,DELT,IBAND, ITIM
C
C ADD E
C
C KLINE=-1
C DO 20 NI=1,NNP,8
C NJMN=NI
C NJMX=MINO(NI+7,NNP)
C KLINE=KLINE+1
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10000) IRAM,KOUT, ---
C WRITE(6,10100) NI, ---
C
C IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10000)KOUT,
C > TIME,DELT,IBAND, ITIM
C
C 20 WRITE(6,10100)NI,(R(NJ),NJ=NJMN,NJMX)
C
C IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10000) IRAM,
C > KOUT,TIME,DELT,IBAND, ITIM
C
C 20 WRITE(6,10100) NI,(R(NJ),NJ=NJMN,NJMX)
C
C ADD E
C
C IF (KPR.EQ.2) RETURN
C
C PRINT MATERIAL FLUX
C
C KOUT=KOUT+1
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10200) IRAM,KOUT, ---
C
C WRITE(6,10200)KOUT,TIME,DELT,IBAND, ITIM
C
C WRITE(6,10200) IRAM,KOUT,TIME,DELT,IBAND, ITIM
C
C ADD E
C
C KLINE=-1

```



```

DO 30 NI=1,NNP,4
  NJMN=NI
  NJMX=MINO(NI+3,NNP)
  KLINE=KLINE+1
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C WRITE(6,10200) IRAM,KOUT, ---
C WRITE(6,10300) (NJ,FX(NJ), ---
C
C IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10200)KOUT,
C > TIME,DELT,IBAND, ITIM
C 30 WRITE(6,10300)(NJ,FX(NJ),FZ(NJ),NJ=NJMN,NJMX)
C
C IF(MOD(KLINE,50).EQ.0 .AND. KLINE.GE.1) WRITE(6,10200) IRAM,
C > KOUT,TIME,DELT,IBAND, ITIM
C 30 WRITE(6,10300) (NJ,FX(NJ),FZ(NJ),NJ=NJMN,NJMX)
C
C ADD E
C
C RETURN
C
C ADD S FORMAT LIST BY HATA
C
C 10000 FORMAT(1H1,I2,42HTH RADIONUCLIDE OF CONSIDERING
C > DECAY CHAIN//13H OUTPUT --- )
C 10200 FORMAT(1H1,I2,42HTH RADIONUCLIDE OF CONSIDERING
C > DECAY CHAIN//13H OUTPUT --- )
C 10500 FORMAT(1H0,/13H TYPE OF FLOW --- )
C 10600 FORMAT( --- ***/1H0,I2,42HTH RADIONUCLIDE OF CONSIDERING
C > DECAY CHAIN///18H SYSTEM-FLOW TABLE --- )
C
C10000 FORMAT(13H1OUTPUT TABLE,I4,27H.. CONCENTRATIONS AT TIME =,
C > 1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//5X,
C > 12H *** ITIME =,I10//
C > 7H NODE I,5X,36HCONCENTRATION AT NODES I,I+1,...,I+7/)
C10100 FORMAT(I7,8(1PD15.4))
C10200 FORMAT(13H1OUTPUT TABLE,I4,26H.. MATERIAL FLUX AT TIME =, 1PD12
C > ,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//5X,
C > 12H *** ITIME =,I10//1X,
C > 4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,
C > 2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ/)
C10300 FORMAT(4(I5,2E11.3,5X))
C10500 FORMAT(/5X,13H TYPE OF FLOW,35X,4HRATE,8X,9HINC. FLOW,7X,
C > 10HTOTAL FLOW/5X,40H CONSTANT-CONCENTRATION NODE FLOW. . . .,3(
C > E12.4,5X)/5X,40H CONSTANT-FLUX-NODE FLOW . . . . .,3(E12.
C > 5X)/5X,40H SEEPAGE FLUX-NODE FLOW. . . . .,3(E12.4,5X)
C > 5X,40H NUMERICAL LOSSES. . . . .,3(E12.4,5X)/5X,
C > 40H NET FLOW. . . . .,3(E12.4,5X)/5X,
C > 40H INCREASE IN MATERIAL CONTENT (LIQUID) .,3(E12.4,5X)/5X,
C > 40H INCREASE IN MATERIAL CONTENT (SOLID). .,3(E12.4,5X)/5X,
C > 40H RADIOACTIVE LOSSES (LIQUID AND SOLID) .,3(E12.4,5X)/5X,
C > 40H INCREASE DUE TO COMP. OF SKELTON . . . . .,3(E12.4,5X))
C10600 FORMAT(1H1,52H*****
C > 62H*****
C > 5H*****//18H SYSTEM-FLOW TABLE,I4,12H.. AT TIME =,1PD12.4,
C > 9H ,(DELT =,1PD12.4,1H),6H ITIM=,I10)
C

```

```
10000 FORMAT(1H1,I2,42HTH RADIONUCLIDE OF CONSIDERING DECAY CHAIN//
> 13H OUTPUT TABLE,I4,27H.. CONCENTRATIONS AT TIME =,
> 1PD12.4,9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//5X,
> 12H *** ITIME =,I10//
> 7H NODE I,5X,36HCONCENTRATION AT NODES I,I+1,...,I+7/)
10100 FORMAT(I7,8(1PD15.4))
10200 FORMAT(1H1,I2,42HTH RADIONUCLIDE OF CONSIDERING DECAY CHAIN//
> 13H OUTPUT TABLE,I4,26H.. MATERIAL FLUX AT TIME =, 1PD12.4,
> 9H ,(DELT =,1PD12.4,15H),(BAND WIDTH =,I4,1H)//5X,
> 12H *** ITIME =,I10//1X,
> 4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ,6X,4HNODE,9X,
> 2HFX,9X,2HFZ,6X,4HNODE,9X,2HFX,9X,2HFZ/)
10300 FORMAT(4(I5,2E11.3,5X))
10500 FORMAT(1H0,/
> 5X,13H TYPE OF FLOW,35X,4HRATE,8X,9HINC. FLOW,7X,
> 10HTOTAL FLOW/5X,40H CONSTANT-CONCENTRATION NODE FLOW. . . .,3(
> E12.4,5X)/5X,40H CONSTANT-FLUX-NODE FLOW . . . . .,3(E12.
> 5X)/5X,40H SEEPAGE FLUX-NODE FLOW. . . . .,3(E12.4,5X)
> 5X,40H NUMERICAL LOSSES. . . . .,3(E12.4,5X)/5X,
> 40H NET FLOW. . . . .,3(E12.4,5X)/5X,
> 40H INCREASE IN MATERIAL CONTENT (LIQUID) ..,3(E12.4,5X)/5X,
> 40H INCREASE IN MATERIAL CONTENT (SOLID). ..,3(E12.4,5X)/5X,
> 40H RADIOACTIVE LOSSES (LIQUID AND SOLID) ..,3(E12.4,5X)/5X,
> 40H INCREASE DUE TO COMP. OF SKELTON ..... ,3(E12.4,5X))
10510 FORMAT(5X,40H TOTAL FLUX OF GROUND SURFACE ..... ,E12.4,5X,
> E12.4,5X,E12.4)
10600 FORMAT(1H1,52H*****
> 62H*****
> 5H*****/1H0,I2,42HTH RADIONUCLIDE OF CONSIDERING DECAY CHAIN//
> /18H SYSTEM-FLOW TABLE,I4,12H.. AT TIME =,1PD12.4,
> 9H ,(DELT =,1PD12.4,1H),6H ITIM=,I10)
```

```
C
C ADD E
C
      END
```

```

        SUBROUTINE STORE(X,Z,IE,R,FX,FZ,TITLE,NPROB,NNP,NEL,NTI,
        > MAXNP,MAXEL,TIME,IRAM)
C
C ADD S DUMMY ARGUMENT BY HATA
C
C SUBROUTINE STORE( --- ,IRAM)
C
C SUBROUTINE STORE(X,Z,IE,R,FX,FZ,TITLE,NPROB,NNP,NEL,NTI,
C > MAXNP,MAXEL,TIME)
C
C ADD E
C
C
C FUNCTION OF SUBROUTINE--TO STORE PERTINENT QUANTITIES ON AUXILIARY
C DEVICE FOR FUTURE USE, E.G. FOR PLOTTING. WHAT DEVICE IS TO BE
C USED MUST BE SPECIFIED BY APPROPRIATE JOB-CONTROL CARDS.
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C DIMENSION TITLE(9)
C DIMENSION X(MAXNP),Z(MAXNP),IE(MAXEL,5),R(MAXNP),FX(MAXNP),
C > FZ(MAXNP)
C
C DATA NPPROB/-1/
C
C IF (NPPROB.EQ.(-1)) REWIND 2
C IF (NPPROB.EQ.NPROB) GO TO 10
C WRITE(2) (TITLE(I),I=1,9),NPROB,NNP,NEL,NTI
C WRITE(2) (X(NP),NP=1,NNP),(Z(NP),NP=1,NNP),((IE(M,IQ),M=1,NEL),
C > IQ=1,5)
C NPPROB=NPROB
C
C ADD S OUTPUT LIST OF WRITE STATEMENT BY HATA
C
C 10 WRITE(2) IRAM,IRAM,TIME, ---
C
C 10 WRITE(2) TIME,(R(NP),NP=1,NNP),(FX(NP),NP=1,NNP),(FZ(NP),NP=1,NN
C
C 10 WRITE(2) IRAM,IRAM,TIME,(R(NP),NP=1,NNP),(FX(NP),NP=1,NNP),
C - (FZ(NP),NP=1,NNP)
C
C ADD E
C
C RETURN
C END

```

```

      SUBROUTINE QINT(QSRC,TIME,QTER,QTIM,TOTA,MAXSRC,MAXRAM,
>                   MAXRED,IRAM,NRED,NSRC)
C
C   THIS SUBROUTINE WAS CREATED BY HATA ON JULY,1989.
C
C   FUNCTION OF SUBROUTINE--TO INTERPOLATE SOURCE TERM AT PRESENT
C   TIME FROM POST FILE OF RELEASE CODE
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION QSRC(MAXSRC,MAXRAM),QTER(MAXRED,MAXRAM),QTIM(MAXRED),
>              TOTA(MAXRAM)
C
      COMMON/STERM/NODR
C
      BTIME=QTIM(NODR)
CDBG BY HATA
C   WRITE(9,*) ' DEBUG SUB QINT '
C   WRITE(9,*) ' NODR ',NODR
C   WRITE(9,*) ' TIME , BTIME ',TIME,BTIME
C   WRITE(9,*) ' QSRC'
C   WRITE(9,*) ( QSRC(I,IRAM),I=1,NSRC)
CDBG
      IF(TIME.GT.BTIME) THEN
          SRINT=0.0
          GO TO 50
      ENDIF
C
      10 CONTINUE
C
          TI1=QTIM(NRED)
          TI2=QTIM(NRED+1)
C
          IF(TIME.GE.TI1.AND.TIME.LT.TI2) GO TO 20
C
          NRED=NRED+1
          GO TO 10
C
      20 CONTINUE
C
          QT1=QTER(NRED,IRAM)
          QT2=QTER(NRED+1,IRAM)
          GRAD=(QT2-QT1)/(TI2-TI1)
          SRINT=GRAD*(TIME-TI1)+QT1
CDBG BY HATA
C   WRITE(9,*) ' NRED ',NRED
C   WRITE(9,*) ' TI1 , TI2 , QT1 , QT2 '
C   WRITE(9,1111) TI1,TI2,QT1,QT2
C   WRITE(9,*) ' TIME , GRAD , SRINT '
C   WRITE(9,1111) TIME,GRAD,SRINT
C1111 FORMAT(4(3X,E10.3))
C   WRITE(9,*) ' TOTA(IRAM) '
C   WRITE(9,*) TOTA(IRAM)
CDBG
          SRINT=SRINT/TOTA(IRAM)
C
      50 CONTINUE

```

```
C      DO 40 I=1,NSRC
      QSRC(I,IRAM)=SRINT
40     CONTINUE
C      RETURN
      END
```

```

        SUBROUTINE BFLUX(TIME,DP,ELA1,QTER,QTIM,EBA,AEL,NPST,IRAM,NBRED,
        >                MXSTNP,MAXRAM,MAXRED)
C
C   THIS SUBROUTINE WAS CREATED BY HATA ON JULY.1989.
C
C   FUNCTION OF SUBROUTINE--TO INTERPOLATE TIME VARIABLE BOUNDARY FLUX.
C
C       IMPLICIT REAL*8(A-H,O-Z)
C
C       DIMENSION DP(MXSTNP,MAXRAM),ELA1(MXSTNP,MAXRAM),AEL(MAXRAM),
        >                QTER(MAXRED,MAXRAM),QTIM(MAXRED),NPST(MXSTNP,MAXRAM)
C
C       COMMON/STERM/ NODR
C       COMMON/BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
C
C       BTIME=QTIM(NODR)
CDBG BY HATA
C       WRITE(9,*) ' DEBUG  SUB BFLUX '
C       WRITE(9,*) ' NODR ',NODR
C       WRITE(9,*) ' TIME , BTIME ',TIME,BTIME
C       WRITE(9,*) ' IRAM EBA '
C       WRITE(9,*)  IRAM,EBA
CDBG
C       IF(TIME.GT.BTIME) THEN
C           DO 81 JJ=1,MXSTNP
C               DP(JJ,IRAM)=0.0
81      CONTINUE
C           GO TO 50
C       ENDIF
C
C   10 CONTINUE
C
C       TI1=QTIM(NBRED)
C       TI2=QTIM(NBRED+1)
C
C       IF(TIME.GE.TI1.AND.TIME.LT.TI2) GO TO 20
C
C       NBRED=NBRED+1
C       GO TO 10
C
C   20 CONTINUE
C
C       QT1=QTER(NBRED,IRAM)
C       QT2=QTER(NBRED+1,IRAM)
C       GRAD=(QT2-QT1)/(TI2-TI1)
C       EBA=GRAD*(TIME-TI1)+QT1
CDBG BY HATA
C       WRITE(9,*) ' DEBUG  SUB BFLUX '
C       WRITE(9,*) ' NBRED ',NBRED
C       WRITE(9,*) ' TI1 , TI2 , QT1 , QT2 '
C       WRITE(9,1111) TI1,TI2,QT1,QT2
C       WRITE(9,*) ' TIME , GRAD , EBA '
C       WRITE(9,1111) TIME,GRAD,EBA
C1111  FORMAT(4(3X,E10.3))
CDBG
C       EBA=EBA/AEL(IRAM)
C

```

```

C
C READ SURFACE-TERM FLUXES EBA TO BE APPLIED AT BOUNDARY NODES NI AND
C AND NJ, RESPECTIVELY
C
      DO 80 JJ=1,MXSTNP
        DP(JJ,IRAM)=0.0
      80 CONTINUE
C
      NPP=0
      MP=0
C
560 IF(MP.EQ.NST) GO TO 50
C
      MP=MP+1
C
      EL=ELA1(MP,IRAM)
C
      NI=NPST(MP,IRAM)
      NJ=NPST(MP,IRAM)
C
      IF(MP.GT.1) GO TO 571
C
      NPP=NPP+1
      NII=NPP
      NPP=NPP+1
      NJJ=NPP
C
      GO TO 578
571 DO 572 I=1,NPP
C
      IJ=NPST(I,IRAM)
C
      IF(IJ.EQ.NI) GO TO 573
572 CONTINUE
      NPP=NPP+1
C
      NII=NPP
      GO TO 574
573 NII=I
574 DO 575 J=1,NPP
C
      IJ=NPST(J,IRAM)
C
      IF(IJ.EQ.NJ) GO TO 576
575 CONTINUE
      NPP=NPP+1
C
      NJJ=NPP
      GO TO 578
576 NJJ=J
C
578 DP(NII,IRAM)=DP(NII,IRAM)+EBA*EL/3.0+EBA*EL/6.0
      DP(NJJ,IRAM)=DP(NJJ,IRAM)+EBA*EL/6.0+EBA*EL/3.0
C
C
      GO TO 560
C
50 CONTINUE

```

```
CDBG BY HATA
C WRITE(9,*) ' DP '
C WRITE(9,*) (DP(I,IRAM),I=1,NST)
CDBG
      RETURN
      END
```



```

SUBROUTINE AROSR(XQ,ZQ,AREA)
C
C THIS SUBROUTINE CREATED BY HATA ON JULY,1989.
C
C FUNCTION OF SUBROUTINE--TO CALICULATE THE AREA OF SOUCE ELEMENT.
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   REAL*8 N(4)
C
C   DIMENSION XQ(4),ZQ(4),PJAB(2,2),DNSS(4),DNTT(4),DWSS(4),DWTT(4)
C   DIMENSION S(4),T(4),W(4)
C
C   DATA P / 0.577350269189626/,S/-1.0D+00,1.0D+00,1.0D+00,-1.0D+00/
C   >      ,T/-1.0D+00,-1.0D+00,1.0D+00,1.0D+00/
C
C   AREA=0.0D0
C
C   DO 10 KG=1,4
C     SS=P*S(KG)
C     TT=P*T(KG)
C
C     CALL SHAPE(N,W,DNSS,DNTT,DWSS,DWTT,SS,TT)
C
C     DO 20 I=1,2
C       DO 20 J=1,2
C         PJAB(I,J)=0.0
C 20    CONTINUE
C
C     DO 30 I=1,4
C       PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNTT(I)
C       PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
C       PJAB(2,1)=PJAB(2,1)-XQ(I)*DNTT(I)
C       PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
C 30    CONTINUE
C
C     DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
C     AREA=AREA+DJAC
C
C 10    CONTINUE
C
C     AREA=AREA
CDBG  BY HATA
C     WRITE(9,*) ' DEBUG  SUB AROSR '
C     WRITE(9,*) ' NUCLIDE NUMBER IR ',IR
C     WRITE(9,*) ' SOURCE ELEMENT AREA '
C     WRITE(9,*)   ELA2(ML,IR)
CDBG
C
C     RETURN
C     END

```

```

SUBROUTINE SRFLUX(BFLX,BFLXP,NPSS,NPB,TGRATE,TGFLOW,NSUR,IRAM,
> MAXBNP,MAXRAM)
C
C THIS SUBROUTINE CREATED BY HATA ON AUGUST,1989.
C
C FUNCTION OF SUBROUTINE--TO CALICULATE TOTAL FLUX OF SURFASE AREA.
C
C     IMPLICIT REAL*8 (A-H,O-Z)
C
C     DIMENSION BFLX(MAXBNP,MAXRAM),BFLXP(MAXBNP,MAXRAM),TGFLOW(MAXRAM)
C     DIMENSION NPSS(MAXBNP),NPB(MAXBNP)
C     COMMON /BNDY/ NBEL,NBN,NRSEL,NRSN,NBC,NSTN(5),NST
C     COMMON /NODO/ NVV
C
C CALCULATE NODAL FLOW RATES
C
C     NVV=NVV
C
C     TGRATE=0.
C     TGFLOW(IRAM)=0.0
C
C     IF (NSUR.LE.0) GO TO 60
C
C     S=0.
C     SP=0.
C
C CDBG BY HATA
C     WRITE(9,*) ' SRFLUX '
C CDBG
C     DO 50 NPP=1,NVV
C
C         NP=NPSS(NPP)
C
C         DO 45 I=1,NBN
C             IJ=NPB(I)
C             IF(IJ.NE.NP) GO TO 45
C             NII=I
C             GO TO 46
C 45          CONTINUE
C 46          CONTINUE
C
C             S=S+BFLX(NII,IRAM)
C 50          SP=SP+BFLXP(NII,IRAM)
C
C         TGRATE=S
C         TGFLOW(IRAM)=S
C
C 60 CONTINUE
C
C     RETURN
C     END

```

```

SUBROUTINE ARDOS(NPGW, NPSS, NSUR, MAXBNP)
C
C FUNCTION OF SUBROUTINE--TO CALCULATE SURFACE TOTAL FLUX
C
C DIMENSION NPGW(MAXBNP,2), NPSS(MAXBNP)
COMMON /NODO/ NVV
C
C MP=0
C NPP=0
C
C 560 IF (MP.EQ.NSUR) GO TO 610
C
C MP=MP+1
C NI=NPGW(MP,1)
C NJ=NPGW(MP,2)
C
C 570 CONTINUE
C
C IF(MP.GT.1) GO TO 571
C NPP=NPP+1
C
C NPSS(NPP)=NI
C
C NII=NPP
C NPP=NPP+1
C
C
C NPSS(NPP)=NJ
C
C NJJ=NPP
C GO TO 578
C 571 DO 572 I=1,NPP
C
C IJ=NPSS(I)
C
C IF(IJ.EQ.NI) GO TO 573
C 572 CONTINUE
C NPP=NPP+1
C
C NPSS(NPP)=NI
C
C NII=NPP
C GO TO 574
C 573 NII=I
C 574 DO 575 J=1,NPP
C
C IJ=NPSS(J)
C
C IF(IJ.EQ.NJ) GO TO 576
C 575 CONTINUE
C NPP=NPP+1
C
C NPSS(NPP)=NJ
C
C NJJ=NPP
C GO TO 578
C 576 NJJ=J
C

```

```
578 CONTINUE  
C  
    GO TO 560  
C  
610    CONTINUE  
C  
    NVV=NPP  
C  
    RETURN  
    END
```

```

SUBROUTINE MASS(XQ,ZQ,RQ,CRET,AREA)
C
C THIS SUBROUTINE CREATED BY HATA ON JULY,1989.
C
C FUNCTION OF SUBROUTINE--TO CALICULATE MASS OF THE ELEMENT.
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   REAL*8 N(4)
C
C   DIMENSION XQ(4),ZQ(4),PJAB(2,2),DNSS(4),DNNT(4),DWSS(4),DWTT(4)
C   DIMENSION S(4),T(4),W(4),RQ(4)
C
C   DATA P / 0.577350269189626/,S/-1.0D+00,1.0D+00,1.0D+00,-1.0D+00/
C   >       ,T/-1.0D+00,-1.0D+00,1.0D+00,1.0D+00/
C
C   AREA=0.0D0
C
C   DO 10 KG=1,4
C     SS=P*S(KG)
C     TT=P*T(KG)
C
C   CALL SHAPE(N,W,DNSS,DNNT,DWSS,DWTT,SS,TT)
C
C   DO 20 I=1,2
C     DO 20 J=1,2
C       PJAB(I,J)=0.0
C 20  CONTINUE
C
C   DO 30 I=1,4
C     PJAB(1,1)=PJAB(1,1)+ZQ(I)*DNNT(I)
C     PJAB(1,2)=PJAB(1,2)-ZQ(I)*DNSS(I)
C     PJAB(2,1)=PJAB(2,1)-XQ(I)*DNNT(I)
C     PJAB(2,2)=PJAB(2,2)+XQ(I)*DNSS(I)
C 30  CONTINUE
C
C     DJAC=PJAB(2,2)*PJAB(1,1)-PJAB(1,2)*PJAB(2,1)
C     AREA=AREA+DJAC
C
C 10  CONTINUE
C
C     AREA=AREA
C
C     CONC=0.0
C
C   DO 40 IQ=1,4
C     CONC=CONC+RQ(IQ)*N(IQ)*AREA
C 40  CONTINUE
CDBG BY HATA
C   WRITE(9,*) ' SHAPE FUNCTION N '
C   WRITE(9,*) (N(I) ,I=1,4)
C   WRITE(9,*) ' WEIGHTING FUNCTION W '
C   WRITE(9,*) (W(I) ,I=1,4)
C
C   CRET=CONC*AREA
C   CRET=CONC
C
C   RETURN
C   END

```

```

      SUBROUTINE TOTC(X,Z,RR,IE,MAXNP,MAXEL,NEL,IR)
C
C THIS SUBROUTINE CREATED BY HATA ON AUGAST,1989.
C
C FUNCTION OF SUBROUTINE--TO CALICULATE TOTAL MASS OF NUCLIDE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(MAXNP),Z(MAXNP),RR(MAXNP),IE(MAXEL,5)
      DIMENSION XQ(4),ZQ(4),RQ(4)
      DIMENSION CONC(1000),AR(1000)
C
      TOTM=0.0
      IRR=IR
C
      DO 10 ML=1,NEL
      DO 20 IQ=1,4
          NP=IE(ML,IQ)
          XQ(IQ)=X(NP)
          ZQ(IQ)=Z(NP)
          RQ(IQ)=RR(NP)
20 CONTINUE
C
      CALL MASS(XQ,ZQ,RQ,CRET,AREA)
C
C
      AR(ML) = AREA
      CONC(ML)=CRET
      TOTM=TOTM+CRET
C
10 CONTINUE
C
C WRITE(9,1000) IR
C WRITE(9,1100) (I,CONC(I),I=1,NEL)
C WRITE(9,1150)
C WRITE(9,1100) (I,AR(I),I=1,NEL)
C WRITE(9,1200) IR,TOTM
1000 FORMAT(2X,I2,' TH RADIONUCRIDE MEMBER '/2X,'MASS OF ELEMENT')
C1150 FORMAT(4X,' ELEMENT AREA ')
1100 FORMAT(2X,4(I3,2X,1PE12.5))
1200 FORMAT(/2X,' NO.',I2,' TOTAL MASS IN SYSTEM :',1PE12.5/)
C
      RETURN
      END

```