PNC SN941 85-90

配布限定

多次元伝熱流動解析コードの整備・改良(II)

タスク3：PCG法の検討

1985年5月

動力炉・核燃料開発事業団
大洗工学センター

この資料は，動燃事業団の開発業務を進めるため，限られた関係者だけに配布するものです。従って，その取扱いには十分注意を払って下さい。なお，この資料の供覧，複製，転載，引用等には事業団の承認が必要です。

配　布　限　定

PNC SN941 85−90

1985年 5 月

多次元伝熱流動解析コードの整備・改良(II)

タスク3：PCG法の検討

村松 寿晴[*], 前川 勇[*]

二ノ方 寿[*]

要　旨

現在使用している単相多次元伝熱流動解析コードCOMMIX−1A（Ver 12.0）の連立1次方程式解法はPSOR法であるが，計算セル数が5000を越えるような詳細問題では収束性が著しく悪い。そこで，理論的には有限回の反復で解を求めることができるCG法(Conjugate Gradient Method）にPreconditioning機能を追加したPCG法をオプションとして追加し，PCG法の特性の把握を行なった。得られた結果は，以下に示す通りである。

1）解くべき連立1次方程式の係数行列をバンド最小化することにより，反復回数が減少する。

2）スケーリングおよび不完全コレスキー分解によるPreconditioningの効果は大きく，反復回数を著しく減少させる。

3）過渡変化の激しい非定常問題ではPCG法がSOR法よりもCPU時間の面で有効であるが，逆に過渡変化の緩やかな問題に対してはPCG法は不利となる。

4）PCG法がベクトル計算機向きであることが確かめられ，またスカラー演算よりも有効であることが確認された。

* 大洗工学センター安全工学部高速増殖炉工学室

NOT FOR PUBLICATION
PNC SN941 85-90
May, 1985

Improvement and Validation of Three-Dimensional Thermal-Hydraulics Code (II)

Task 3 : Investigation of P.C.G. Method

Toshiharu Muramatsu*, Isamu Maekawa*, and Hisashi Ninokata*

Abstract

COMMIX-1A is a single-phase three-dimensional thermal-hydraulic analysis code with finite difference method developed at U.S. Argonne National Laboratory. In the original version, the scheme employs the P.S.O.R. method to solve the linear equation system. It has been pointed out that the convergence of mass balance becomes worse for detailed problems with computational cells of more than 5000. To eliminate this problem, a new solver has been added to the code. The scheme is based on the P.C.G. (Preconditioned Conjugate Gradient) method.

The performances of P.C.G. method are as follows:

1) Iteration number is decreased by reducing the band width of the coefficient matrix.
2) Iteration number is decreased by using a preconditioned coefficient matrix with scaling and/or incomplete Choleski decomposition processes.
3) P.C.G. method has been successfully applied to the severe transient problems. However it is found that the method is not efficient for slow transient analyses.
4) It has been confiremed that the P.C.G. method is suited for vector machine.

* FBR Engineering Section, Safety Engineering Div., OEC, PNC

目　　次

第1章　緒　　言 …… 1

第2章　PCG法と計算手法 …… 2

　2.1　PCG法 …… 2

　　2.1.1　CG法 …… 2

　　2.1.2　対角スケーリング法 …… 4

　　2.1.3　ICCG法 …… 4

　2.2　計算手法 …… 6

　　2.2.1　係数行列の作成 …… 6

　　2.2.2　計算の流れ …… 10

　　2.2.3　Hybrid Scheme …… 11

第3章　PCG法に関する各種検討 …… 12

　3.1　数値実験用解析体系 …… 12

　3.2　係数行列の作成とその効果 …… 12

　3.3　PCG法オプションによる条件数の改善 …… 14

　3.4　SOR法との比較 …… 14

　3.5　Hybrid Scheme の効果 …… 17

　3.6　PCG解法の選定の目安 …… 18

　3.7　ベクトル化の検討 …… 18

　3.8　タンク型炉炉上部プール解析 …… 18

　3.9　高速実験炉「常陽」MK－I 自然循環解析 …… 19

　3.10　今後の課題 …… 20

第4章　結　　言 …… 21

　　　参考文献 …… 22

付　録

　A　追加入力データ …… 23

　B　圧力方程式の離散化 …… 24

　C　エネルギー方程式の離散化 …… 26

　D　PCG計算パッケージ …… 28

　E　サブルーチン負荷率 …… 29

List of Table

Table E.1 Comparison of Sub-Program Occupation Ratio Between Original Version and Corrected Version

List of Figures

Figure 2.1 C.G. Algorithm for Symmetrical Matrix (Hestenes-Stiefel Version)

Figure 2.2 C.G. Algorithm for Unsymmetrical Matrix

Figure 2.3 I.C.C.G. Algorithm

Figure 2.4 Overall Flow Chart of COMMIX-1A

Figure 2.5 General Flow Chart of SUBROUTINE MOLOOP

Figure 2.6 General Flow Chart of SUBROUTINE ENLOOP

Figure 2.7 Overall Flow Chart of P.C.G. Solver

Figure 3.1 Mesh Arrangement for Fundamental Problems

Figure 3.2 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = C.G. Method)

Figure 3.3 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = Scaling Method)

Figure 3.4 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = I.C.C.G. Method)

Figure 3.5 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Scheme = C.G. Method)

Figure 3.6 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Scheme = Scaling Method)

Figure 3.7 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Scheme = I.C.C.G. Method)

Figure 3.8 Comparison of Matrix Condition of Each Scheme on 126 Cells Problem

Figure 3.9 Comparison of Matrix Condition of Each Scheme on 1400 Cells Problem

Figure 3.10 Comparison of Convergency of Each Scheme on 126 Cells Problem

Figure 3.11 Comparison of Mass Balance Convergency of Each Scheme about 126 Cells Problem on Scalar Machine (Time Step = 1)

Figure 3.12 Comparison of Mass Balance Convergency of Each Scheme about 126 Cells Problem on Scalar Machine (Time Step = 28)

Figure 3.13 Comparison of Convergency of Each Scheme on 1400 Cells Problem

Figure 3.14 Comparison of Mass Balance Convergency of Each Scheme about 1400 Cells Problem on Scalar Machine (Time Step = 1)

Figure 3.15 Comparison of Mass Balance Convergency of Each Scheme about 1400 Cells Problem on Scalar Machine (Time Step = 600)

Figure 3.16 Effect of Hybrid Schemes on 126 Cells Problem

Figure 3.17 Effect of Hybrid Schemes on 1400 Cells Problem

Figure 3.18 Velocity Transient at Inlet of 126 Cells Model

Figure 3.19 C.P.U. Time Ratio VS. Velocity Change Ratio

Figure 3.20 Time Step Size Δt Dependency of Matrix Condition on 126 Cells Problem

Figure 3.21 Comparison of Convergency of Each Scheme about 126 Cells Problem on Vector Machine

Figure 3.22 Comparison of Convergency of Each Scheme about 1400 Cells Problem on Vector Machine

Figure 3.23 Schematic Figure of Water Test Section

Figure 3.24 Computational Cell Partitioning of the Test Plenum

Figure 3.25 Input Data for Steady-State Run of Water Test

Figure 3.26 Input Data for Transient Run of Water Test

Figure 3.27 Velocity and Temperature Transient at Inlet of the Test Section

Figure 3.28 Comparison of Processing Time of Each Scheme on Scalar Machine

Figure 3.29 Comparison of Processing Time of Each Scheme on Vector Machine

Figure 3.30 Cell Partitioning for Two-Dimensional Analyses of JOYO MK-I Natural Circulation Test

Figure 3.31 Comparison of Convergency of Each Scheme about Steady-State Run of JOYO Natural Circulation Test

Figure D.1 Source List of P.C.G. Solver

Figure E.1 Comparison of Convergency Between Original Version and Corrected Version

Figure E.2 Effect of S. YMOMI Skip for C.P.U. Time of S. TIMSTP
Figure E.3 Effect of S. YMOMI Skip for C.P.U. Time of S. SOLVIT

第1章　緒　　言

昭和58年1月に米国NRCを通してANLから動燃事業団に導入された有限差分法による単相3次元汎用熱流動解析コード"COMMIX－1A（Ver. 12.0）[1)]"は，動燃事業団における高速炉の設計および伝熱流動安全評価解析を推進するための標準コードとして位置付けられようとしている。

有限差分法あるいは有限要素法を用いた解析計算に於ては，最も計算時間のかかる部分の1つに大型連立1次方程式を解くというプロセスがある。このプロセスの計算量の効率化を計ることは，解析コード全体の効率化に少なからぬ貢献をする。

連立1次方程式の解法には，大別すると次の様なものがある。

- 直接法
  - Cramer の公式
  - 消去法
    - 三角化法
      - Gauss 法
      - Crout 法
      - Choleski 法
    - 掃出し法（Gauss－Jordan法）
    - エスカレータ法
  - 直交化法
- 反復法
  - 線形反復法
    - Gauss－Seidel 法
    - 逐次過剰緩和法（SOR法）
    - Chebyshev 準反復法
  - 傾斜法
    - 最急降下法（SD法）
    - 共役傾斜法（CG法）

COMMIX－1Aに於ては，PSOR法（Point Successive Overrelaxation iterative method）を使用しており，現在まで各種計算に用いられてきている。しかし，詳細計算（計算セル数≳5000）での収束性が極端に悪いため，実際的な解析（3次元流れ問題等）が十分に行なえないことが明らかとなっている。そこで，理論的にはn次元連立1次方程式を高々n回の反復で解くことができるCG法にPreconditioning 機能を追加したPCG法をCOMMIX－1Aの解法の1つとして追加した。この解法は，各反復において（行列×ベクトル）の計算のみを必要とするだけで，消去法のように係数行列の全ての要素を配列として記憶する必要が全くない。いわば有限回の計算回数で厳密解が得られる直接法の性質と，行列とベクトルの積和計算のみを必要とする反復法の性質とを兼ね備えた解法である。

本報告書は，今回追加したPCG法の詳細と基本問題について従来のSOR法と比較検討した結果について記したものである。

第２章　Ｐ.Ｃ.Ｇ.法と計算手法

2.1　PCG法

2.1.1　ＣＧ法

ＣＧ法は，1952年にM. HestenesとE. Stiefelとによって発表された手法[2)]で，反復法でありながら有限回の反復で連立１次方程式を解くことができるという画期的なものである。さらに，ＣＧ法には，ＳＯＲ法に見られるような加速パラメータの選定の煩わしさがなく，直接法のように大量の記憶領域も必要としないのが特徴である。

解くべき連立１次方程式を

$$Ax = b$$

とする。ここで

Ａ：正値対称な疎係数行列＝$\begin{bmatrix} a_{11}, & a_{12} \cdots\cdots a_{1j} \\ & \cdots\cdots\cdots\cdots \\ a_{i1}, & a_{i2} \cdots\cdots a_{ij} \end{bmatrix}$

ｘ：未知数

ｂ：定数

である。

ＣＧ法の対称行列用アルゴリズムを以下に，またフローチャートをFigure 2.1に示す。

$$r_i^o = b_i - \sum_{j=1}^{n} a_{ij}\, x_j^o$$

$$P_i^o = r_i^o$$

$$\alpha^k = (r_i^k,\ r_i^k)/(AP_i^k, P_i^k)$$

$$x_i^{k+1} = x_i^k + \alpha^k p_i^k$$

$$r_i^{k+1} = r_i^k - \alpha^k Ap_i^k$$

$$\beta^k = (r_i^{k+1}, r_i^{k+1})/(r_i^k, r_i^k)$$

$$p_i^{k+1} = r_i^{k+1} + \beta^k p_i^k$$

$$i = 1,\ 2,\ \cdots\cdots,\ n$$

ここで，各記号は以下の意味である。

Press：未知数前timestep値

r　：残　差

a　：係数行列要素

p　：修正方向

α ：修正量
β ：残差率
y ：$a_{ij} \cdot p_j$
t ：time step
k ：iteration step
n ：次 数
i, j ：行列 index
EPS ：収束判定値

CG法の重要な特徴は，残差ベクトル γ_i について，

$$(\gamma_i \cdot \gamma_j) = 0 \qquad (i = j)$$

が成立することである。すなわち， i = j ならば γ_i と γ_j は直交する。したがって，異なった方向にn回の反復を行なえば，n次元ベクトル空間でn＋1個以上の直交基は存在しない故に，$\gamma^{n+1} = 0$ となり厳密解に到達する。詳しくは，文献2）参照のこと。

また，アルゴリズムを見て判かる様に，計算過程に於て行列とベクトルの乗算及びベクトルの内積・加算しかなく，ベクトル計算機向きであると言える。

一方，Aが非対称行列の場合には，

$$A \cdot A^T \text{ および } A^T \cdot A$$

が対称行列になる性質を利用して，Figure 2.1のアルゴリズムを修正する。すなわち，Aの現われる演算において，

$$\gamma_i^o = A^T \cdot (b_i - \sum_{j=1}^{n} a_{ij} x_j^o) \qquad i = 1, n$$

および

$$y_i^k = A^T \cdot (\sum_{j=1}^{n} a_{ij} p_j) \qquad i = 1, n$$

の形で計算する。この修正アルゴリズムを以下に，またフローチャートをFigure 2.2に示す。

$$\gamma_i^o = A^T (b_i - \sum_{j=1}^{n} a_{ij} x_j^o)$$

$$p_i^o = \gamma_i^o$$

$$\alpha^k = (\gamma_i^k, \gamma_i^k) / (A^T (A p_i^k), p_i^k)$$

$$x_i^{k+1} = x_i^k + \alpha^k p_i^k$$

$$\gamma_i^{k+1} = \gamma_i^k - \alpha^k A^T (A p_i^k)$$

$$\beta^k = (\gamma_i^{k+1}, \gamma_i^{k+1}) / (\gamma_i^k, \gamma_i^k)$$

$$p_i^{k+1} = r_i^{k+1} + \beta^k p_i^k$$

$$i = 1, 2, \cdots\cdots, n$$

2.1.2 対角スケーリング法[3)]

CG法の収束を有利にするPreconditioningの一つであり，最も簡単なものである。

連立1次方程式

$$\left\{\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots\cdots\cdots\cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots\cdots\cdots\cdots + a_{2n}x_n = b_2 \\ \quad\vdots \qquad\qquad\qquad\qquad \vdots \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots\cdots\cdots\cdots + a_{nn}x_n = b_n \end{array}\right.$$

をそのまま解くかわりに，第1，2，……n式の両辺を，それぞれ対角要素の平方根（$d_1, d_2, \cdots d_n$）で割り，また変数変換

$$y_1 = e_1 x_1 \qquad y_2 = e_2 x_2 \cdots\cdots\cdots \qquad y_n = e_n x_n$$

を行なって，

$$\left\{\begin{array}{l} a'_{11}y_1 + a'_{12}y_2 + \cdots\cdots\cdots\cdots + a'_{1n}y_n = b'_1 \\ a'_{21}y_1 + a'_{22}y_2 + \cdots\cdots\cdots\cdots + a'_{2n}y_n = b'_2 \\ \quad\vdots \qquad\qquad\qquad\qquad \vdots \qquad \vdots \\ a'_{n1}y_1 + a'_{n2}y_2 + \cdots\cdots\cdots\cdots + a'_{nn}y_n = b'_n \end{array}\right.$$

という形にして解くものである。ここで

$$a'_{ij} = a_{ij}/(d_i e_j)$$

$$b'_i = b_i/d_i$$

$$d_i = \sqrt{a_{ii}}$$

$$e_i = \sqrt{a_{ii}}$$

である。

2.1.3 ICCG法

ICCG法（Incomplete Choleski and Conjugate Gradient Method）[4),5)]は，Meijerink, van der Vorst らにより開発された。これは，直接法（$L \cdot L^T$分解）とCG法を混合した手法である。疎な係数行列に完全コレスキー分解（$L \cdot L^T$）を行なえば，分解行列Lはもはや疎ではなくなり密行列となる。そこで行列Lももとの係数行列と同程度に疎にするために不完全コレスキー分解を考えることになる。即ち，もとの係数行列において0が入っている場所については何も

計算せず0と置いてしまい，計算を続行することである。これにより作成される行列Lともとの係数行列Aとの関係は，

$$L \cdot L^T \approx A$$

となる。この関係を用い，

$$A \cdot x = b$$

を解くかわりに

$$(L^{-1} A L^{-T})\ (L^T x) = (L^{-1} b)$$

として，これにCG法を適用するものである。これは，$A \cdot x = b$ に対し変数変換を行なったことと同等で，

$$A' \cdot x' = b'$$

を解くことに帰着する。ここで，

$$A' = (L^{-1} A L^{-T})$$

$$x' = (L^T x)$$

$$b' = (L^{-1} b)$$

である。

ICCG法の対称行列用アルゴリズムを以下に，またフローチャートをFigure 2.3に示す。

$$r_i^o = b_i - \sum_{j=1}^{n} a_{ij} x_j^o$$

$$p_i^o = (LL^T)^{-1} r_i^o$$

$$\alpha^k = (r_i^k, (LL^T)^{-1} r_i^k) / (Ap_i^k, p_i^k)$$

$$x_i^{k+1} = x_i^k + \alpha^k p_i^k$$

$$r_i^{k+1} = r_i^k - \alpha_i^k A p_i^k$$

$$\beta^k = (r_i^{k+1}, (LL^T)^{-1} r_i^{k+1}) / (r_i^k, (LL^T)^{-1} r_i^k)$$

$$p_i^{k+1} = (LL^T)^{-1} r_i^{k+1} + \beta^k p_i^k$$

$$i = 1, 2, \cdots\cdots, n$$

アルゴリズム中の $(LL^T)^{-1} r_i$ は，以下の様な前進代入および後退代入による消去法により評価される。

$$L \cdot V = r_i$$
$$V = L^{-1} \cdot r_i$$
$$L^T \cdot u = V$$
$$u = L^{-T} V$$
$$= L^{-1} L^{-T} r_i$$

ここで,

L ： 下三角行列（既知）

L^T ： 上三角行列（既知）

r_i ： 残差（既知）

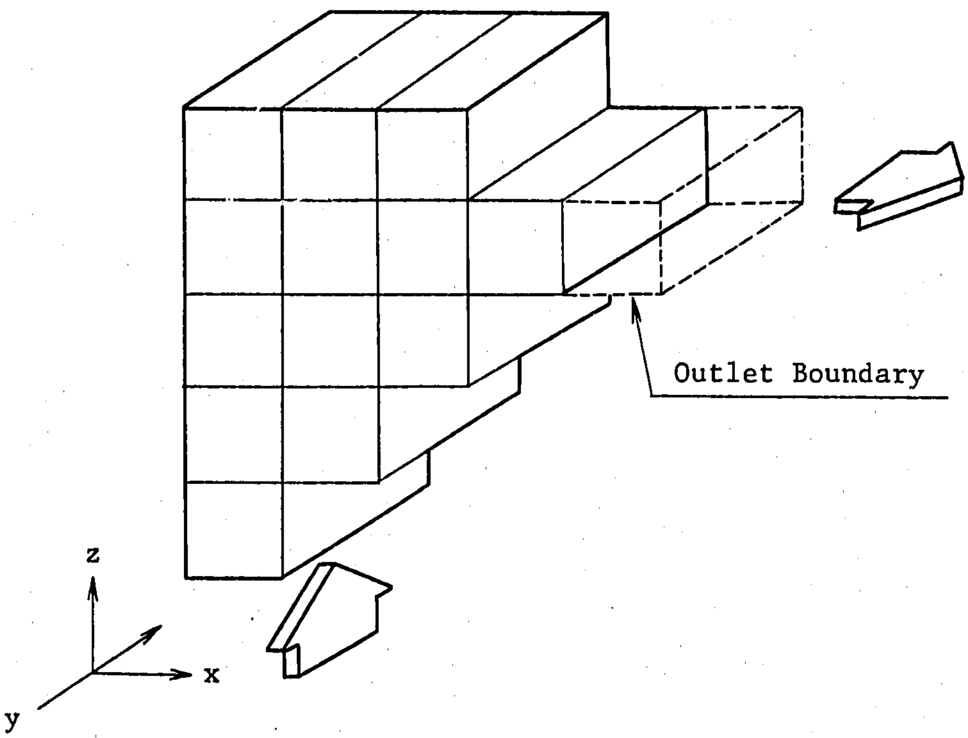
u ： $(L L^T)^{-1} \cdot r_i$

V ： 中間作業ベクトル

2.2 計算手法

2.2.1 係数行列の作成

係数行列の作成に当り，例題として以下の様な5×1×5の体系を考える。



COMMIX－1A では，出口境界セルについては圧力方程式を解かないため，係数行列からもこれを削除して考える。したがって，取り扱うべき1次方程式数は，解析体系内セル総数から出口境界セル数を引いた値となり，上記体系では13個となる。

COMMIX－1A で取り扱う圧力方程式は,

$$A_0 P_0 = \sum_{\ell=1}^{6} A_\ell P_\ell + B_0$$

で表わされ，右辺の第1項を左辺に移行して展開すると，

$$A_0 P_0 - A_1 P_1 - A_2 P_2 - A_3 P_3 - A_4 P_4 - A_5 P_5 - A_6 P_6 = B_0$$

となる。ここで，

A：圧力方程式係数

P：圧力

B：圧力方程式定数

である。また，式中の添字の意味は，

0：MO，注目セル

1：MIM，MOに接する－x 側セル

2：MIP，〃 ＋x 〃

3：MJM，〃 －y 〃

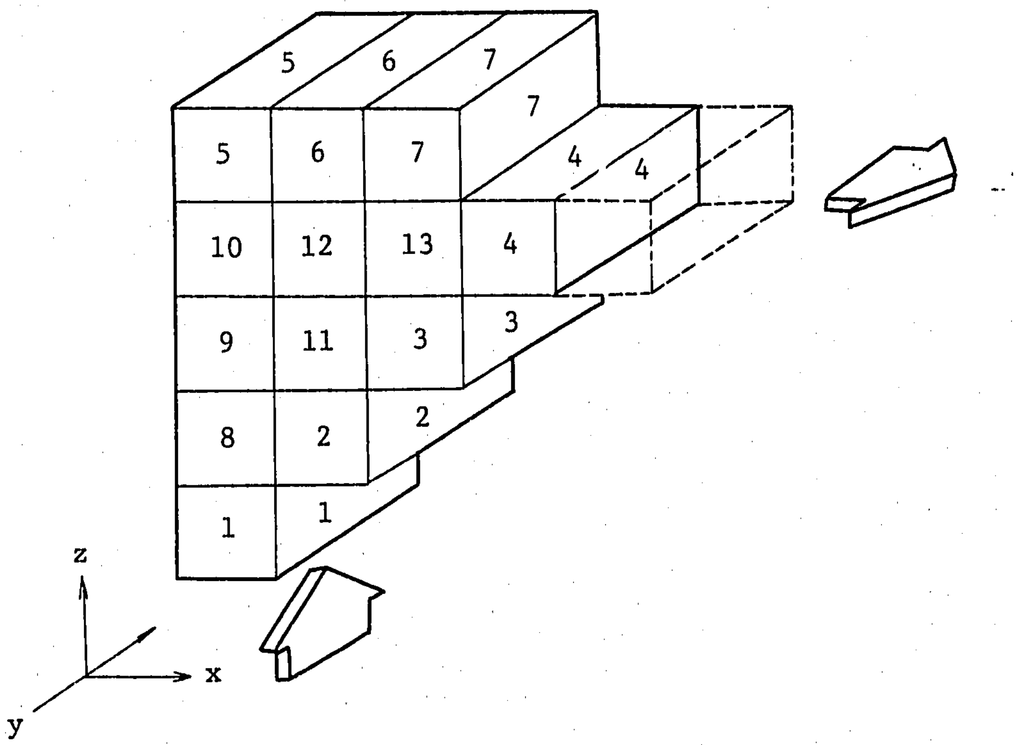
4：MJP，〃 ＋y 〃

5：MKM，〃 －z 〃

6：MKP，〃 ＋z 〃

である。

COMMIX－1A に於けるセルの番号付けは，境界表面の規定順序に依存し，概して連続した番号付けとはならない。COMMIX－1A に従って番号付けを行なうと以下の様になる。



上記のセル番号順に係数行列を作成すると，

$$
\begin{array}{cc}
 & \begin{array}{ccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{array} \\
\begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13 \end{array} &
\begin{bmatrix}
MO & & & & & & & MKP & & & & & \\
 & MO & & & & & & MIM & & & MKP & & \\
 & & MO & & & & & & & & MIM & & MKP \\
 & & & MO & & & & & & & & & MIM \\
 & & & & MO & MIP & & & & MKM & & & \\
 & & & & MIM & MO & MIP & & & & & MKM & \\
 & & & & & MIM & MO & & & & & & MKM \\
MKM & MIP & & & & & & MO & MKP & & & & \\
 & & & & & & & MKM & MO & MKP & MIM & & \\
 & & & & MKP & & & & MKM & MO & & MIP & \\
 & MKM & MIP & & & & & & MIM & & MO & MIP & \\
 & & & & & MKP & & & & MIM & MKM & MO & MIP \\
 & & MKM & MIP & & & MKP & & & & & MIM & MO
\end{bmatrix}
\end{array}
$$

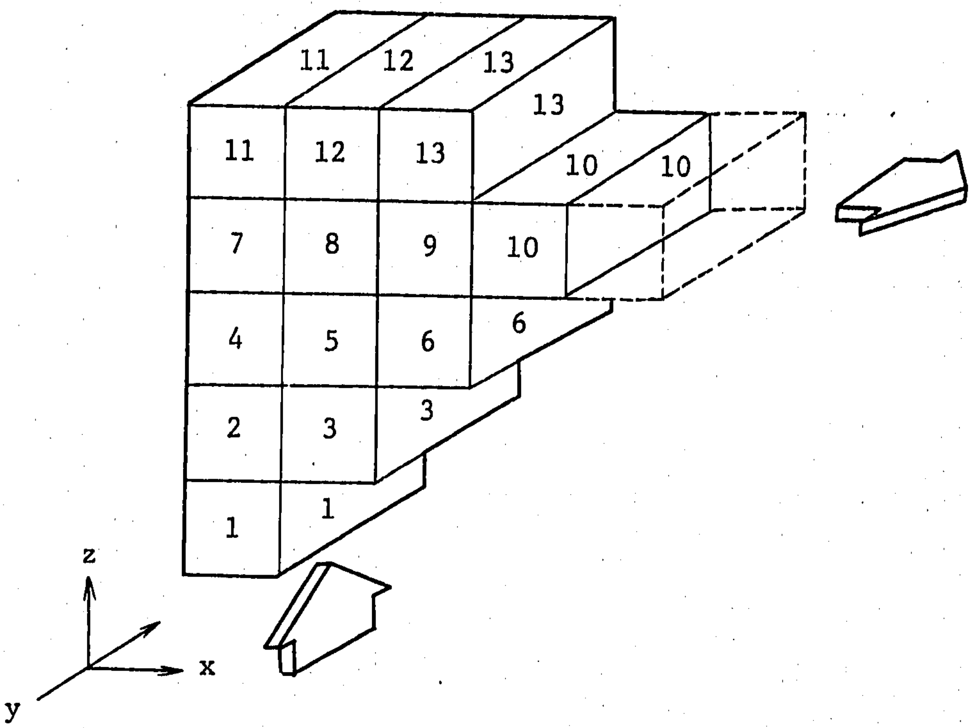
となる。ただし，空白位置には零が入る。したがって，解くべき連立１次方程式は，

$$
\begin{bmatrix}
a_{10} & & & & & & & -a_{16} & & & & & \\
 & a_{20} & & & & & & -a_{21} & & & -a_{26} & & \\
 & & a_{30} & & & & & & & & -a_{31} & & -a_{36} \\
 & & & a_{40} & & & & & & & & & -a_{41} \\
 & & & & a_{50} & -a_{52} & & & & -a_{55} & & & \\
 & & & & -a_{61} & a_{60} & -a_{62} & & & & & -a_{65} & \\
 & & & & & -a_{71} & a_{70} & & & & & & -a_{75} \\
-a_{85} & -a_{82} & & & & & & -a_{80} & -a_{86} & & & & \\
 & & & & & & & -a_{95} & a_{90} & -a_{96} & -a_{91} & & \\
 & & & & -a_{106} & & & & -a_{105} & a_{100} & & -a_{102} & \\
 & -a_{115} & -a_{112} & & & & & & -a_{111} & & a_{110} & -a_{112} & \\
 & & & & & -a_{126} & & & & -a_{121} & -a_{125} & a_{120} & -a_{122} \\
 & & -a_{135} & -a_{132} & & & -a_{136} & & & & & -a_{131} & a_{130}
\end{bmatrix}
\cdot
\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \\ P_{10} \\ P_{11} \\ P_{12} \\ P_{13} \end{bmatrix}
=
\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \\ B_{11} \\ B_{12} \\ B_{13} \end{bmatrix}
$$

である。ところで，ＣＧ法系統の解法に於ては，係数行列のバンド幅が小さい方が解への誤差の混入が少なく収束に有利* なため，ここでもセルの番号付けを変更することでバンド幅の最小化を試みる。

同様の体系で以下の番号付けを考える。

* 対角優位な係数行列が形成されるため。第 3. 2 節参照



同様にセルの番号順に係数行列を作成すると，

$$
\begin{array}{cc}
 & \begin{array}{ccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{array} \\
\begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13 \end{array} &
\begin{bmatrix}
\mathrm{MO} & \mathrm{MKP} & & & & & & & & & & & \\
\mathrm{MKM} & \mathrm{MO} & \mathrm{MIP} & \mathrm{MKP} & & & & & & & & & \\
 & \mathrm{MIM} & \mathrm{MO} & & \mathrm{MKP} & & & & & & & & \\
 & \mathrm{MKM} & & \mathrm{MO} & \mathrm{MIP} & & \mathrm{MKP} & & & & & & \\
 & & \mathrm{MKM} & \mathrm{MIM} & \mathrm{MO} & \mathrm{MIP} & & \mathrm{MKP} & & & & & \\
 & & & & \mathrm{MIM} & \mathrm{MO} & & & \mathrm{MKP} & & & & \\
 & & & \mathrm{MKM} & & & \mathrm{MO} & \mathrm{MIP} & & & \mathrm{MKP} & & \\
 & & & & \mathrm{MKM} & & \mathrm{MIM} & \mathrm{MO} & \mathrm{MIP} & & & \mathrm{MKP} & \\
 & & & & & \mathrm{MKM} & & \mathrm{MIM} & \mathrm{MO} & \mathrm{MIP} & & & \mathrm{MKP} \\
 & & & & & & & & \mathrm{MIP} & \mathrm{MO} & & & \\
 & & & & & & \mathrm{MKM} & & & & \mathrm{MO} & \mathrm{MIP} & \\
 & & & & & & & \mathrm{MKM} & & & \mathrm{MIM} & \mathrm{MO} & \mathrm{MIP} \\
 & & & & & & & & \mathrm{MKM} & & & \mathrm{MIM} & \mathrm{MO}
\end{bmatrix}
\end{array}
$$

となり，半帯幅５の準バンド行列となる。しかし，上記行列要素の内，約⅔が零要素であり，解析体系が大型化した場合の計算機記憶域の労費も無視できない。そこで，各行に格納される要素数が３次元解析体系の場合，注目セルを含めて最大７個であることを利用し，以下の様な圧縮格納法を採用した。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	MO	MKP						1	2					
2	MKM	MO	MIP	MKP				1	2	3	4			
3	MIM	MO	MKP					2	3	5				
4	MKM	MO	MIP	MKP				2	4	5	7			
5	MKM	MIM	MO	MIP	MKP			3	4	5	6	8		
6	MIM	MO	MKP					5	6	9				
7	MKM	MO	MIP	MKP				4	7	8	11			
8	MKM	MIM	MO	MIP	MKP			5	7	8	9	12		
9	MKM	MIM	MO	MIP	MKP			6	8	9	10	13		
10	MIP	MO						9	10					
11	MKM	MO	MIP					7	11	12				
12	MKM	MIM	MO	MIP				8	11	12	13			
13	MKM	MIM	MO					9	12	13				

ここで，1列から7列までが係数値（Content），また8列から14列までが格納位置（Index）である。この圧縮格納法を用いることにより，N^2（解析セル数の2乗）の配列を $N \times 14$ に縮小する。この格納法の効果は，解析体系が大型化するに従い指数関数的に現われる。

一方，エネルギー方程式については，出口境界セルも合わせて解くため，取り扱うべき1次方程式数は上記体系で14個となる。方程式形は，

$$A_0 V_0 = \sum_{\ell=1}^{6} A_\ell V_\ell + B_0$$

であり，係数行列の作成方法は圧力方程式の場合と同様である。ここで，

A： エネルギー方程式係数

V： エンタルピー

B： エネルギー方程式定数

である。

2.2.2 計算の流れ

Figure 2.4にCOMMIX－1A 全体のフロー・チャートを示す。コード内の時間制御は，SUBROUTINE TIMSTPで行なわれ，このSUBROUTINEの下位モジュールとして運動量式計算部（SUBROUTINE MOLOOP），k－ε乱流輸送式計算部（SUBROUTINE TKLOOPおよびSUBROUTINE TELOOP）およびエネルギー式計算部（SUBROUTINE ENLOOP)が接続されている。これらSUBROUTINEは，各時間ステップ毎に順次計算される。Figure 2.5にSUBROUTINE MOLOOPのフロー・チャートを示す。ここでは，運動量方程式の離散化式中の係数を計算（SUBROUTINE XMOMI，SUBROUTINE YMOMIおよびSUBROUTINE ZMOMI）し，また圧力方程式中の係数を計算（SUBROUTINE PEQN）し，ここで得られる圧力方程式係数を用いて新圧力（n+1ステップの圧力）をSUBROUTINE SOLVITで反復法で求める。従来この反復はSOR法を使用しており，今回PCG法をオプションとして追加した。ここで新圧力が求まった後，SUBROUTINE MOMENIで新流速を先に計算した運動方程式係

数を用いて更新し，SUBROUTINE TIMSTP に戻る。Figure 2.6 にSUBROUTINE ENLOOP のフロー・チャートを示す。ここでは，エネルギー式中の係数をSUBROUTINE ENERGI で計算し，この係数を用いSUBROUTINE SOLVENでSOR法により新エンタルピーを求める。今回このSUBROUTINE SOLVEN内にもオプションとしてPCG. Solver を追加した。ここで新エンタルピーが求まった後，SUBROUTINE TIMSTP に戻る。 k－ε 乱流輸送計算においてもSUBROUTINE SOLVENが用いられ，エネルギー計算時と同様にPCG. Solver がオプション選択できる。フロー・チャート等は，文献6）参照のこと。

Figure 2.7にSUBROUTINE SOLVITおよびSUBROUTINE SOLVENに組み込んだPCG. Solver のフロー・チャートを示す。このサブルーチンでは，後述する入力データIFPCG（SUBROUTINE SOLVIT）およびIFPCG2（SUBROUTINE SOLVEN）によりPCG解法の内の各種オプションが選択される。このオプションのうち，1，－3，－5 は，第2.1節で説明した手法であり，それぞれCG法，スケーリング法，ICCG法である。－1のオプションは，各時間ステップ毎に作られる係数行列の全ての固有値をヤコビ法により求めるものである。ここで計算された固有値は収束の速さに関する目安となり，$\lambda_{max}/\lambda_{min}$* で判定される。また，－2のオプションは，繰り返し中に修正方向 P^{k+1} $(=\gamma^{k+1}+\beta^k\cdot P^k,\ =(LL^T)^{-1}\gamma^{k+1}+\beta^k P^k)$，および残差 γ^k 中に累積される計算誤差（丸め誤差）を一掃し，残差多項式の重みを付け変える操作である。反復を続行した場合との違いは，

$$P^{k+1}=\gamma^{k+1}+\beta^k P^k \quad ,\ =(LL^T)^{-1}\gamma^{k+1}+\beta^k P^k$$

の替りに

$$P^{k+1}=\gamma^{k+1} \qquad ,\ =(LL^T)^{-1}\gamma^{k+1}$$

とすることである。この操作の頻度は，N（方程式数）である。

2.2.3 Hybrid Scheme

このHybrid Scheme とは，CG系統解法とSOR法との混在使用による解法の意味である。一般に激しい過渡領域においてSOR法により質量バランスを取るには相当の反復が必要であるが，CG系統の解法ではN回以下で求めることができる。一方，過渡領域が終了し安定領域に入った場合には，SOR法は数回の反復で解が求まる事が多いが，CG系統の解法ではやはりN回以下の反復が必要である。したがって，過渡変化の激しい時期にはCG系統解法を，また安定期にはSOR法を使用するといった混在使用が考えられる。この手法の切り換え時期は，各時間ステップに於ける体系内最大流速に対する最大流速変動量（$\Delta V_{max}=V^{n+1}-V^n$），すなわち $\Delta V_{max}/V_{max}$ を閾値とすることによりなされる。この閾値は入力データで定義される。

---

* 絶対値最大固有値／絶対値最小固有値：条件数≧10^4 以上では，収束が著しく悪くなる。

第3章　P.C.G.法に関する各種検討

3.1　数値実験用解析体系

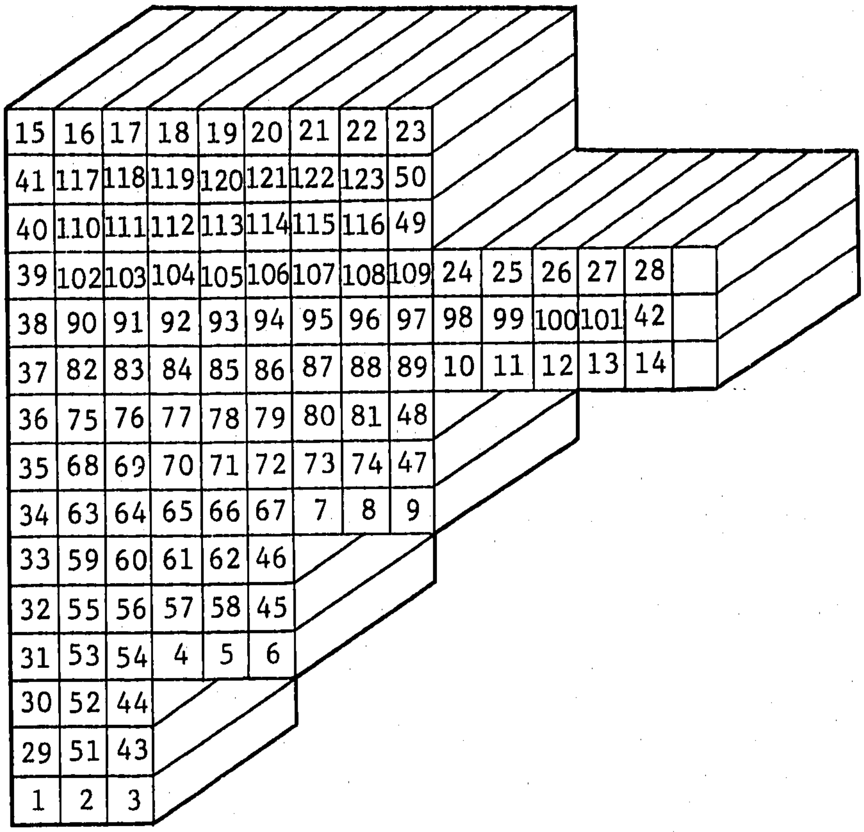
PCG法に関する検討に際し，Figure 3.1に示す2種類の基本問題体系を設定した。これらの体系の基本形状データを以下に示す。

Items		Problem 1	Problem 2
Geometry		Box	←
No. of Computational Cells		126	1400
No. of Cells in X-Direction		15	50
in Y-Direction		1	1
in Z-Direction		15	50
Calculational Cell Size	along X-Axis	0.03333	0.03333
	along Y-Axis	0.1	0.1
	along Z-Axis	0.03333	0.03333
Inlet Velocity		1.0 m/s	←
Inlet Temperature		20.0 °C	←
Boundary Condition		Free Slip with Adiabatic	

各問題において取り扱う圧力方程式数は，第2.2.1項で述べたように，それぞれ123個，1390個である。

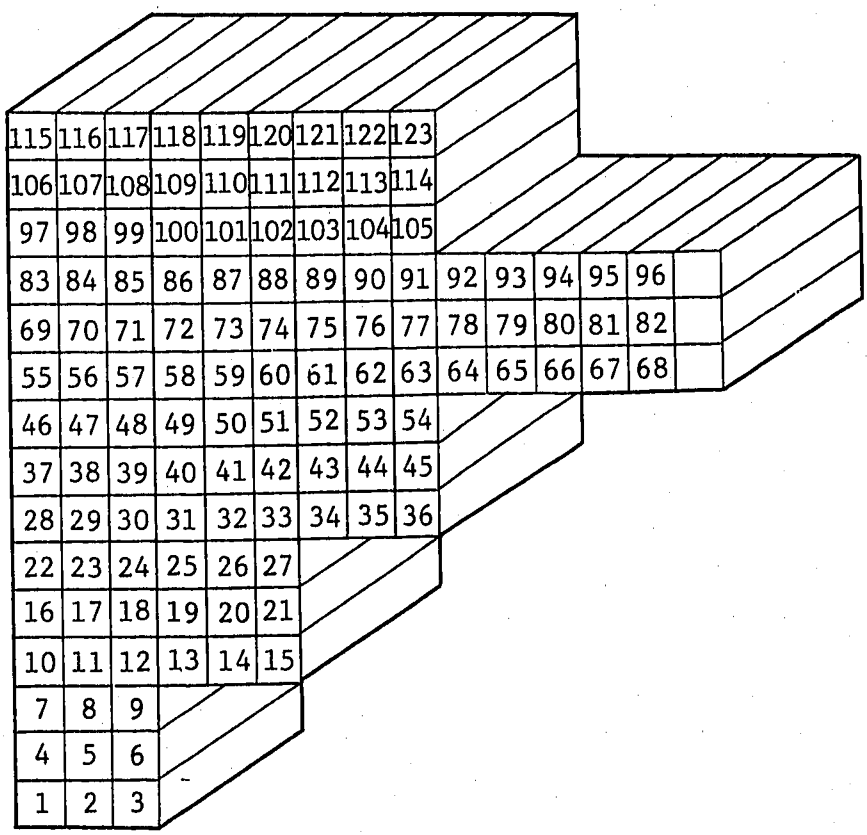
3.2　係数行列の作成とその効果

126セル問題について，オリジナル・プログラムに従ってセルの番号付けを行なうと，以下の様になる。



この番号付けは，Fingure 3.1 に示した境界表面規定順序に従って行なわれ，これにより作成される係数行列の半帯幅は 102* となる。

一方，バンド幅を小さくすることを目的として番号付けを行なうと，



* 123番セルを隣接セル情報で表わす場合，22番，50番，116番，122番のセル情報が必要である。したがって，半帯幅は（123－22）＋1となる。

となり，係数行列の半帯幅は 15 となる。

このバンド幅の最小化による反復回数の改善の比較をCG法，スケーリング法および ICCG 法についてFigure 3.2〜Figure 3.4に示す。各図中に於ける反復回数 123 回の直線は方程式数であり，第 2.1.1 項で述べた収束に対する理論的反復回数値である。結果より明らかなように，バンド幅が小さい係数行列を用いた場合の各時間ステップに於ける反復回数は，それを用いない場合の反復回数よりも少ない。この傾向は各反復手法について一様に見られ，反復回数の減少率は平均約 25％である。

1400 セル問題についても同様な処理により半帯幅を計算すると，オリジナル・プログラムでは 1313，バンド幅最小化処理では 50 となる。このバンド幅最小化による反復回数の改善の比較をCG法，スケーリング法および ICCG 法についてFigure 3.5〜Figure 3.7 に示す。 1400 セル問題に於ても 126 セル問題と同様に，バンド幅を小さくした係数行列を用いた場合の方が用いない場合に比べ，各時間ステップでの反復回数は少ない。この反復回数の減少率は，平均約52％である。

上記 2 種類の問題に於て，反復回数の減少率が違う理由は，次節で述べる係数行列の条件数の違いによるものである。いづれにしろ，係数行列のバンド幅最小化効果は反復回数の減少に明確に現われている。

3.3 PCG法オプションによる条件数の改善

第 2.2.2 項でも触れたように，係数行列が持つ固有値のうち，絶対値最大の固有値と絶対値最小の固有値の比（$\lambda_{max}/\lambda_{min}$）で定義される値（条件数）は，収束の速さに関する目安となる。これは，CG法に関する重要な性質の 1 つから導びかれる。この性質とは，

「係数行列 Aの相異なる固有値が m（<n）個ならば，CG法は m回の反復で正解に到達する。」（証明は文献 4）を参照）

である。すなわち，修正方向（P）を探索する際，固有ベクトルに沿って決定してゆくため，この固有値のばらつきが小さいほど計算誤差（丸め誤差）の影響を受けず速く収束する。一般に計算誤差の影響を受け易い問題は，悪条件（ill−Conditioned）であると言い，この閾値は文献 2）によると 10^4 程度である。

今回追加した PCG法オプションのうち，スケーリング法， ICCG 法の Precenditioning機能は，固有値のばらつきを小さくする働きをする。

Figure 3.8 および Figure 3.9 に各Preconditioning の効果を 126 セル問題および 1400セル問題についてそれぞれ示す。結果より判かる通り，いづれの問題についても，CG法の条件数を基準とした場合 ICCG 法による Preconditioningが最も効果がある。この Preconditioning による反復回数への効果は，Figure 3.2〜Figure 3.7 に明確に現われている。

3.4 SOR法との比較

SOR法との比較計算として，以下に示す 6 ケースを選定した。

1）SOR法(1)　（IT＝1，ITMAXP＝99）
2）SOR法(2)　（IT＝10，ITMAXP＝99）
3）SOR法(3)　（IT＝1，ITMAXP＝10000）
4）CG法　（ITMCG＝150，1500）
5）スケーリング法　（ITMCG＝150，1500）
6）ICCG法　（ITMCG＝150，1500）

1)，2)，3）のSOR法の3ケースは，反復制御パラメータを変えたものである。1）のSOR法のケースは，解析問題を定常計算問題としてとらえ，定常解を得ることを目的としたものである。この場合には，Figure 2.4に示した運動量計算部，乱流輸送量計算部およびエネルギー計算部を含む繰り返しループ（ITERMX）を1に設定し，かつFigure 2.5に示した圧力方程式の最大反復回数をデフォルト値の99回に設定する。すなわち，計算を開始してから定常状態に到達するまでの過程に於ては，質量バランスの厳密さは追求しないで時間ステップを進め，最終的な定常解で質量バランスが取れれば良いという考え方に立った計算である。2）のSOR法のケースは，定常解析問題を過渡計算問題の収束解としてとらえ，計算開始から指定時間到達までの過程に注目したものである。この場合には，各時間ステップで厳密に質量バランスを取りながら時間ステップを進めてゆく必要があるため，Figure 2.4内の運動量計算部，乱流輸送量計算部およびエネルギー計算部を含む繰り返しループを10に設定する。3）のSOR法のケースは，連立1次方程式を解く能力をCG系統解法の能力と比較するためのものである。すなわち，圧力方程式の係数（A_0，A_1～A_6，B_0）により形成される連立1次方程式に於てどの程度の反復で解が得られるかを比べるための計算である。このため，圧力方程式を解くための SORループをデフォルト値の99回から10000回に変更した。4)，5)，6）のCG系統解法での最大反復回数は，126セル問題の場合に150回，1400セル問題の場合に1500回に設定した。

Figure 3.10に126セル問題について定常状態到達までの各種解法のCPU時間（FACOM VP－100）の比較を，また以下に各解法が定常状態到達までに費したCPU時間を示す。

1）SOR法(1)　12.2 sec.
2）SOR法(2)　25.9 sec.
3）SOR法(3)　36.3 sec.
4）CG法　58.6 sec.
5）スケーリング法　16.0 sec.
6）ICCG法　16.9 sec.

各解法のうち，最も少ないCPU時間で定常状態に到達するのは，SOR法(1)である。これは，各時間ステップで厳密に質量バランスを取らないことによる。したがって，解析問題を定常計算問題としてとらえ，単に定常解を求めるだけであればSOR法のこの使い方がCG系統解法よりも有利である。一方，定常解析問題を過渡計算問題の収束解としてとらえたSOR法(2)では，SOR法(1)の約2倍のCPU時間が使用され，かつスケーリング法および ICCG法の約1.6倍となっている。このことから，解析問題が過渡計算問題の場合にはスケーリング法またはICCG法のCG系統解法が有利である。また，SOR法(3)の連立1次方程式を解く能力の比較では,CG法よりは速いがスケーリング法および ICCG法よりは遅く約2.2倍の CPU時間が必要となっ

ている。CG系統解法同志での比較については，Figure 3.2～Figure 3.4に見られる様に各時間ステップでの反復回数はCG法，スケーリング法，ICCG法の順に減少しているがCPU時間での比較ではこの順番となっていない。すなわち，ICCG法の平均反復回数はスケーリング法の約3割であるがCPU時間はほぼ同程度の値となっている。この原因は，1時間ステップを計算するCPU時間のうち，約7割が不完全コレスキー分解処理に費やされているためであり，これにより反復回数の有利さは相殺されている。この不完全コレスキー分解処理では，圧縮格納された係数行列中から処理に必要な要素を探し出すインデックス操作が頻繁に行なわれ，この処理に大部分のCPU時間が用いられている。したがって，この処理部の効率化を計ることによってCPU時間の大幅な短縮が期待できる。以下に1時間ステップに於けるS. SOLVIT内各処理の平均CPU時間負荷率を示す。

	CG法	スケーリング法	I.C.C.G法
スケーリング処理	――	12％	――
不完全コレスキー分解処理	――	――	68％
反　復　処　理	100％	88％	32％
CPU時間合計(sec.)	0.301	0.278	0.294
反復1回当りのCPU時間(msec.)	3.24	4.29	4.95

この結果より，ICCG法における不完全コレスキー分解処理の効率化がICCG法全体のCPU時間低減に大きく寄与するであろうことが明らかに言える。

Figure 3.11およびFigure 3.12に時間ステップ1での質量バランスの収束過程と時間ステップ28での同様の結果をそれぞれ示す。時間ステップ1に於ては，SOR法により質量バランスを取るために相当な時間を要し，CG系統解法の約22倍の時間を必要としている。一方，時間ステップ28（Figure 3.10の$\Delta V_{max}/V_{max}=10^{-2}$付近）に於ては，CG系統解法の約半分の時間で質量バランスが取られている。これは，第2.2.3項で述べた通り，激しい過渡状態（流動不安定状態）ではCG系統解法が有利であり，緩やかな過渡状態（流動安定状態）ではSOR法が有利であることを裏づけている。

Figure 3.13に1400セル問題について定常状態到達までの各解法のCPU時間（FACOM VP－100）の比較を，また以下に各解法が定常状態到達までに費したCPU時間を示す。

1）SOR (1)　4693.2 sec.
2）SOR (2)　5997.6 sec.
3）SOR (3)　7662.4 sec.
4）CG法　7814.2 sec.
5）スケーリング法　7504.4 sec.
6）ICCG法　7062.8 sec.

1400セル問題についても，最も少ないCPU時間で定常状態に到達するのはSOR法(1)である。また，過渡計算問題として取り扱ったSOR法(2)のCPU時間は，CG系統解法のいづれよりも速い。これは，126セル問題の場合と逆であり，計算セル数（係数行列次数）が増大したために1時間ステップのCPU時間が増大したことによる。以下に1時間ステップに於ける

S. SOLVIT 内容処理の平均 CPU時間負荷率を示す。

	CG 法	スケーリング法	ICCG 法
スケーリング処理	——	17%	——
不完全コレスキー分解処理	——	——	72%
反復処理	100%	83%	28%
CPU時間合計(sec)	12.63	13.02	14.12
反復1回当りの CPU時間(msec)	67.18	66.77	81.21

Figure 3.14 および Figure 3.15 に時間ステップ1での質量バランスの収束過程と時間ステップ600(Figure 3.13 の $\Delta V_{max}/V_{max}=10^{-2}$ 付近)での同様の結果をそれぞれ示す。これら両者の傾向は126セル問題と同様であり，過渡変化の度合いによる各解法のメリットが現われている。

3.5 Hybrid Scheme の効果

第3.4節に於いて，過渡変化の度合いによる各解法のメリットを述べた。すなわち，激しい過渡変化時にはCG系統解法を，また緩やかな過渡変化時に SOR 法を適用することにより最短経路を辿って定常状態に到達するものと思われる。この Hybrid Scheme による実行は,以下の2種類について行なった。

1) Hybrid (1)　　Scaling 法＋SOR 法(2)
2) Hybrid (2)　　ICCG法 ＋SOR 法(2)

解法の切り替えは入力データ DHYBR で行ない，その値はそれぞれ10^{-1}，10^{-2}とした。126セル問題について定常状態到達までの CPU時間(FACOM VP-100)の比較を Figure 3.16 に，また以下に各解法が定常状態到達までに費した CPU時間を示す。

1) Hybrid (1)　　16.2 sec.
2) Hybrid (2)　　16.9 sec.

両 Hybrid Scheme とも前節で示した SOR 法(2) の CPU時間の約6割で定常状態に到達している。しかし，スケーリング法のみおよび ICCG 法のみにより計算したケースの CPU時間とは有意な差は見られない。

Figure 3.17 に1400セル問題について定常状態到達までの CPU時間の比較を，また以下に各解法が定常状態到達までに費した CPU時間を示す。

1) Hybrid (1)　　4832.6 sec.
2) Hybrid (2)　　4264.4 sec.

この問題に対しては，SOR 法(2)の CPU時間の約7割で定常状態に到達している。また，スケーリング法のみおよび ICCG 法のみにより計算したケースに比べても速い。この問題に於ては Hybrid Scheme の効果が顕著に現われている。

3.6 PCG解法の選定の目安

過渡変化の度合いにより各解法を選定する目安を得るため，126セル問題体系を用い流速変化率を変えて計算を行なった。計算に用いる過渡現象はFigure 3.18に示す5種類とし，初期状態の1m/sから除々に流速を減少させる。Figure 3.19に計算結果を示す。図中の横軸は単位時間ステップに於ける流速変動率であり，縦軸はCPU時間の比（t_{ICCG}/t_{SOR}）である。この結果より，単位時間ステップでの流速変動率が大きくなるに従いICCG法がSOR法よりも優勢になるのが判かる。Figure 3.20に係数行列条件数のΔt依存性を示す。この結果では，Δtをしだいに大きくするに従い条件数も大きくなってゆく傾向が判かる。したがって，単純に流速変動率を大きく設定（すなわちΔtを大きく設定）しても，必ずしもICCG法がSOR法よりも優勢となる保証はなく，普遍定量的な解法選定の目安は得ることはできない。

3.7 ベクトル化の検討

第2.1.1項で述べたように，CGアルゴリズムは計算過程に於て行列とベクトルの乗算及びベクトルの内積・加算のみであり，ベクトル計算機向きである。そこで，FACOM VP-100システムを用いてベクトル演算* を行ない，CPU時間をスカラー演算のそれと比較を行なった。

Figure 3.21に126セル問題について定常状態到達までの各解法のCPU時間の比較を，また以下に各解法が定常状態到達までに費やしたCPU時間をスカラー演算の場合の値と共に併記した。

	ベクトル演算	スカラー演算
1）CG法	27.6 sec.	58.6 sec.
2）ICCG法	9.1 sec.	16.9 sec.

結果から，ベクトル演算により定常状態到達までのCPU時間がスカラー演算の2.1倍，1.9倍に加速されているのが判かる。

1400セル問題についての結果をFigure 3.22に，また定常状態到達までのCPU時間をスカラー演算のそれと共に以下に示す。

	ベクトル演算	スカラー演算
1）CG法	4842.6 sec.	7814.2 sec.
2）ICCG法	4211.1 sec.	7062.8 sec.

この結果では，ベクトル化によりそれぞれ1.6倍，1.7倍に加速している。

ただし，今回は単純にオリジナル・プログラムをベクトル化したのみで何の最適化チューリングも行なっていない。したがって，最適化を行なえばより高速化される可能性がある。

3.8 タンク型炉炉上部プール解析[7)]

標記実験は，日本に於ける大型タンク型LMFBRのFeasibility Study作業の一環として電力中央研究所（CRIEPI）が実施した水実験である。この実験の目的は，炉上部機構および中間熱交換器の上下位置が炉上部プール内での流動状況，自由液面変動およびサーマル・ストラテ

---

* SOR繰り返しループ内にはベクトル化対象部分が存在しないため，スカラー演算とした。

ィフィケーション現象に及ぼす影響を明らかにすることにある。試験装置はFigure 3.23 に示す通りであり，タンク型LMFBRの炉上部を模擬している。燃料集合体を模擬した炉上部機構下部からの水は，炉上部プールを経て内筒中央部のフロー・スリットを通り，アニュラス部下端の出口ノズルより流出する。

試験条件を以下に示す。

<定常試験（$Re = 9.6 \times 10^4$）>

流量　　16 ℓ/s

水温　　8℃

<過渡試験（$Ri = 2.36$）>

流量　　（16 ℓ/s → 3.84 ℓ/s）／10sec.

温度　　（62℃ → 9.6℃）／30sec.

過渡試験は，原子炉スクラム後の炉上部プール内ストラティフィケーション現象を模擬するもので，Ri 数を Super Phoenix の事象に合わせている。

解析のためのメッシュ図をFigure 3.24 に示す。境界条件は，流体についてFree Slip，温度については断熱とした。また，乱流モデルは使用していない。Figure 3.25 およびFigure 3.26 に入力データを示す。Figure 3.27 は，体系入口における流速と温度の過渡関数グラフである。

Figure 3.28 にシミュレーション時間30秒までに使用したCPU時間の積算を示す。このケースは，スカラー演算であり，CG系統解法は SOR 法よりも多くの時間を費している。Figure 3.29 は，ベクトル演算結果であり，これによりCG系統解法は SOR 法よりも若干速くなっている。このベクトル化演算により，ICCG 法はスカラー演算に対し約 7 %加速した。

3.9 高速実験炉「常陽」MK－I 自然循環解析

高速実験炉「常陽」の MK－I 炉心に於て実施された自然循環試験[8] のうち，Case －E を対象とした 2 次元解析問題* に CG系統解法を適用した。

解析体系のセル分割は Figure 3.30 に示す通りであり，解析条件等は全て文献 8）のものを使用した。Figure 3.31 に定常状態到達までのCPU時間の推移を各解法について示す。また以下に，各解法が定常状態到達までに費した CPU時間を示す。

SOR 法	1121.3 sec.
CG 法	定常解に未到達
スケーリング法	1481.4 sec.
ICCG 法	1540.8 sec.

ここで，SOR 法の実行に於ては，定常解を求める場合に通常用いられる反復制御データ（IT＝1，ITMAXP＝99）を使用し，かつ付録Eに示す改良プログラムを使用した。

各種解法のうち，最も少ない CPU時間で定常状態に到達するものは SOR 法であり，ICCG

* 試験開始前の 75MW定常状態を求める Steady−State Run

法の約73％である。この結果からも，第3.4節で述べたように定常問題に対してはSOR法が有利であるということが言える。

3.10 今後の課題

以上に述べてきたPCG法に関する各種検討によって，PCG法の特性および今後改良してゆくべき問題点が明らかとなった。得られたPCG法の特性は，

1）解くべき連立1次方程式の係数行列を最小バンド化して解くことにより反復回数が減少する（第3.2節）。

2）スケーリングおよび不完全コレスキー分解によるPreconditioningの効果は大きく，反復回数を著しく減少させる（第3.3節）。

3）過渡変化の激しい非定常問題ではPCG法がSOR法よりもCPU時間の面で有効であるが，過渡変化の緩やかな問題に対してはPCG法は不利となる（第3.4節）。

4）各時間ステップでの反復回数はICCG法の方がスケーリング法よりも常時少ないが，CPU時間での比較ではICCG法の有意性は一概には言えない（第3.4節）。

5）（スケーリング法＋SOR法）あるいは（ICCG法＋SOR法）のような異種解法の混在使用は非定常計算に有効であり，それぞれの解法を単独に使用した場合よりも一概に計算時間は少ない（第3.5節）。

6）係数行列の条件数は時間ステップ幅に依存し，3）の結果に基づいて時間ステップ幅を大きく設定しても必ずしもICCG法がSOR法よりも有利となる保証はない（第3.6節）

7）PCG法がベクトル計算機向きであることが確かめられ，スカラー演算よりも有効であることが確認された（第3.7節）。

である。

一方，改良あるいは明確にすべき点としては，

1）ICCG法における不完全コレスキー分解処理の効率化

2）PCG解法選定に関する普遍定量的目安の明確化

3）ベクトル最適化チューニングの実施

があげられる。

1）は第3.4節で述べられたものであり，この効率化は直接ICCG法全体の効率化に反映される。例えば対角要素抽出のためのインデックス操作を簡略化するために，あらかじめ係数行列内に対角要素格納用アドレスを設けるといった方法が考えられる。

2）は任意の解析問題が与えられた場合に，どの解法を選択すべきかを判断する材料を得ることである。この問題に対しては解析ケースの積み重ねが重要であり，その中から一般性を引き出してゆく作業が必要である。

3）はプログラムをベクトル化し易い様に変更するものであり，例えばDOループ内のIF文の低減，配列要素の入れ替え等が考えられる。

今後，上記改良に取りくむと共に解析ケースを増し，PCG解法の限界等を明確にしてゆく予定である。

第4章 結　　　言

単相3次元汎用熱流動解析コード"COMMIX－1A"にPCG法(Preconditioned Conjugate Gradient Method)をオプションとして追加し，オリジナルのPSOR法(Point Successive Overrelaxation Method)との比較・検討を行なった。得られたPCG法の特性を以下に示す。

1) 解くべき連立1次方程式の係数行列をバンド最小化することにより，反復回数が減少する。
2) スケーリングおよび不完全コレスキー分解によるPreconditioningの効果は大きく，反復回数を著しく減少させる。
3) 過渡変化の激しい非定常問題ではPCG法がSOR法よりもCPU時間の面で有効であるが，過渡変化が緩やかな問題に対してはPCG法は不利となる。
4) PCG法がベクトル計算機向きであることが確かめられ，またスカラー演算よりも有効であることが確認された。

今後，各計算部のプログラミング最適化を実施し，PCG法の限界を明確化する必要がある。

参 考 文 献

1) H.M. Domanus, "COMMIX-1A: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems", ANL Draft Report Sep., 1982

2) 一松 信, "共役勾配法", 教育出版, 1984

3) 戸川隼人, "マトリクスの数値計算", オーム社, 1979

4) 野寺 隆, "大型疎行列に対するＰＣＧ法",
Seminar on Mathematical Science No.7, 慶応大学, 1983

5) David S. Kershaw, "The Incomplete Choleski-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations", J. of Comp. Physics 26, 1978

6) 村松 他, "多次元伝熱流動解析コードの整備改良(Ⅱ), タスク１：COMMIX－１Ａへの ｋ－ε ２方程式乱流モデルの追加", ＰＮＣ資料 SN941 85－14, 1985

7) I. Maekawa, "Study on In-Vessel Thermal Hydraulics in Pool-Type LMFBRs, Phase 1 : Water Tests Analysis", PNC N941 84-84, 1984

8) M. Takahashi, "JOYO Mark-I Natural Circulation Analysis with COMMIX-1A", PNC SN941 84-99, 1984

付録A　追加入力データ

今回のPCG解法オプション追加により，新たに追加された入力データを以下に示す。

Namelist ／GEOM／

IFPCG　圧力方程式解法オプション・フラグ(0)

0：SOR法

1：CG法

2：CG法＋デバッグ

−1：CG法＋固有値計算

−2：CG法＋リスタート・オプション

−3：スケーリング法

−5：ICCG(0)法

−11：Hybrid (1)　ICCG → SOR

−12：Hybrid (2)　スケーリング法→SOR

−13：Hybrid (3)　CG → SOR

IFPCG2　エネルギー方程式解法オプション・フラグ(0)

IFPCGと同様（但し，−3，−5，−11，−12のオプションは除く）

Namelist ／DATA／

ITMCG　CG系統解法使用時の最大反復回数(N)*

DHYBR　Hybrid Scheme 使用時の切換値 $\left(\frac{\Delta V_{max}}{V_{max}}=0.1\right)$

*　係数行列次数

注）カッコ内はデフォルト値である。

付録B　圧力方程式の離散化

1）基礎式

$$\gamma_V \frac{\partial \rho}{\partial t} + \frac{\partial \gamma_{Ax} \rho u}{\partial x} + \frac{\partial \gamma_{Ay} \rho v}{\partial y} + \frac{\partial \gamma_{Az} \rho w}{\partial z} = D$$

ここで,

γ_V	:	Volume Porosity
γ_{Ax}, γ_{Ay}, γ_{Az}	:	Surface Permeability
ρ	:	密　度
u, v, w	:	流　速
t	:	時　間
x, y, z	:	座　標
D	:	Mass Residual

2）離散化

$$\begin{aligned}
&\Delta x\ \Delta y\ \Delta z\ \gamma_{V0} \left(\frac{\partial \rho}{\partial t}\right)_0 \\
&\quad + (\Delta y\ \Delta z\ \gamma_{Ax}\, u\, \rho)_2 - (\Delta y\ \Delta z\ \gamma_{Ax}\, u\, \rho)_1 \\
&\quad + (\Delta x\ \Delta z\ \gamma_{Ay}\, v\, \rho)_4 - (\Delta x\ \Delta z\ \gamma_{Ay}\, v\, \rho)_3 \\
&\quad + (\Delta x\ \Delta y\ \gamma_{Az}\, w\rho)_6 - (\Delta x\ \Delta y\ \gamma_{Az}\, w\rho)_5 = \Delta x\ \Delta y\ \Delta z \cdot D
\end{aligned}$$

ここで，ρはUpwind Schemeを使用している。また，文献1）でのMomentumの式は,

$$u_1 = u_1 - d_1\ (P_0 - P_1)$$

$$u_2 = u_2 - d_2\ (P_2 - P_0)$$

$$v_3 = v_3 - d_3\ (P_0 - P_3)$$

$$v_4 = v_4 - d_4\ (P_4 - P_0)$$

$$w_5 = w_5 - d_5\ (P_0 - P_5)$$

$$w_6 = w_6 - d_6\ (P_6 - P_0)$$

で表わされており，これらの式を先の離散化式中に代入すると,

$$
\begin{aligned}
V_0\,\gamma_{VO}\left(\frac{\partial\rho}{\partial t}\right)_0 & + A_2<\rho>_2\ \{u_2 - d_2\ (P_2 - P_0)\} \\
& - A_1<\rho>_1\ \{u_1 - d_1\ (P_0 - P_1)\} \\
& + A_4<\rho>_4\ \{v_4 - d_4\ (P_4 - P_0)\} \\
& - A_3<\rho>_3\ \{v_3 - d_3\ (P_0 - P_3)\} \\
& + A_6<\rho>_6\ \{w_6 - d_6\ (P_6 - P_0)\} \\
& - A_5<\rho>_5\ \{w_5 - d_5\ (P_0 - P_5)\}
\end{aligned}
$$

整理すると

$$
\begin{aligned}
& \{A_1<\rho>_1 d_1 + A_2<\rho>_2 d_2 + A_3<\rho>_3 d_3 + A_4<\rho>_4 d_4 + A_5<\rho>_5 d_5 + A_6 \\
& <\rho>_6 d_6\}P_0 \\
& \quad = A_1<\rho>_1 d_1 P_1 + A_2<\rho>_2 d_2 P_2 + A_3<\rho>_3 d_3 P_3 \\
& \qquad + A_4<\rho>_4 d_4 P_4 + A_5<\rho>_5 d_5 P_5 + A_6<\rho>_6 d_6 P_6 \\
& \qquad + A_1<\rho>_1 u_1 - A_2<\rho>_2 u_2 + A_3<\rho>_3 v_3 \\
& \qquad - A_4<\rho>_4 v_4 + A_5<\rho>_5 w_5 - A_6<\rho>_6 w_6 - V_0\,\gamma_{VO}\left(\frac{\partial\rho}{\partial t}\right)_0 + V_0 D
\end{aligned}
$$

まとめると,

$$
\sum_{\ell=1}^{6} A_\ell<\rho>_\ell d_\ell)\ P_0
$$

$$
= \sum_{\ell=1}^{6} A_\ell<\rho>_\ell d_\ell P_\ell + \sum_{\ell=1}^{6} (-1)^{\ell+1} A_\ell<\rho>_\ell u_\ell - V_0\,\gamma_{VO}\left(\frac{\partial\rho}{\partial t}\right)_0 + V_0 D
$$

ここで

$$
\begin{aligned}
a_\ell^P &= A_\ell<\rho>_\ell d_\ell \\
a_0^P &= \sum_{\ell=1}^{6} a_\ell^P \\
b_0^P &= \sum_{\ell=1}^{6} (-1)^{\ell+1} A_\ell<\rho>_\ell u_\ell - V_0\,\gamma_{VO}\left(\frac{\partial\rho}{\partial t}\right)_0 \\
\delta &= V_0 D
\end{aligned}
$$

と置くと,

$$
a_0^P P_0 = \sum_{\ell=1}^{6} a_\ell^P P_\ell + b_0^P
$$

となる。この式中の a_0^P, a_ℓ^P, b_0^P がSUBROUTINE PEQNで計算され,係数行列要素となる。

付録C エネルギー方程式の離散化

1) 基礎式

$$\gamma_V \rho \frac{\partial h}{\partial t} + \gamma_V h\left[\frac{\partial \rho}{\partial t} - \frac{D}{\gamma_V}\right] + \gamma_j \frac{\partial}{\partial x_j}(\rho u_j h)$$

$$= \gamma_V \frac{dP}{dt} + \gamma_j \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j}\right) + \gamma_V Q$$

$$+ \left(\gamma_j \frac{\partial}{\partial x_j}(\tau_{ji} u_j) + \gamma_V \rho g_j u_j + R_j u_j\right)$$

ここで

h ： エンタルピー

γ_j ： Surface Permeability

λ ： 熱伝導度

T ： 温度

Q ： 発熱量

τ ： Stress Tensor

g ： 加速度

R ： Distributed Resistance

2) 離散化

$$\left[\left\{\frac{\rho}{\Delta t} + \left(\frac{\partial \rho}{\partial t}\right)\right\} V_0 \gamma_{V0} + a_0^T\right] h_0$$

$$= a_1^T h_1 + a_2^T h_2 + a_3^T h_3 + a_4^T h_4 + a_5^T h_5 + a_6^T h_6$$

$$+ V_0 \gamma_V \frac{dP}{dt} + V_0 \gamma_V Q + V_0 \gamma_{V0} \frac{\rho}{\Delta t} h_0^n$$

ここで,

$$a_\ell^T = (-1)^{\ell+1} F_{\ell,\ 0} + D_\ell$$

$$a_0^T = \sum_{\ell=1}^{6} a_\ell^T$$

$$F_\ell = \Delta x_i\, \Delta x_j\, \gamma_j <\rho>_\ell u_\ell$$

$$D_\ell = \Delta x_i\, \Delta x_j \left(\frac{\Delta x/2}{\lambda_0} + \frac{\Delta x_\ell/2}{\lambda_\ell}\right)$$

である。SUBROUTINE ENERGI では, $\left[\left\{\frac{\rho}{\Delta t} + \left(\frac{\partial \rho}{\partial t}\right)\right\} V_0 \gamma_{V0} + a_0^T\right]$, a_ℓ^T および

$V_0\,\gamma_V\,\frac{dP}{dt}+V_0\,\gamma_V Q+V_0\,\gamma_{V0}\,\frac{\rho}{\Delta t}\,h_0^n$ をそれぞれ A_0, A_ℓ, B_0 の形で出力しSUBROUTINE SOLVENで作られる係数行列の要素となる。

付録D　P.C.G.計算パッケージ

今回追加したPCG計算ルーチンのうち，計算部を抜き出し一般に使用できるように汎用化した。このソース・リストをFigure D.1に示す。この計算パッケージはサブルーチン化されており，その引数は

```
SUBROUTINE PCGPACK (
      IFPCG, ITMCG, DCONV2, IMAT2
      AA, BB, AAT, BBT, P, BTO, N)
```

である。各引数の意味は，本文およびプログラム中のコメント文を参照されたい。

例題として，以下の連立1次方程式を考える。

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 = 7 \\ \quad\quad\;\; 3x_2 + \;\; x_3 = 4 \\ \;\; x_1 \quad\quad\quad + 8x_3 = 2 \end{cases}$$

この連立1次方程式により作られる圧縮係数行列は，

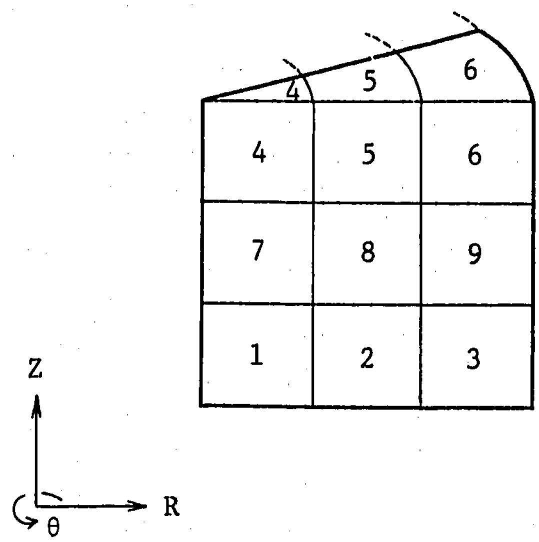
$$\left[\begin{array}{ccccccc:ccccccc} 3, & 5, & 2, & 0, & 0, & 0, & 0 & 1, & 2, & 3, & 0, & 0, & 0, & 0 \\ 3, & 1, & 0, & 0, & 0, & 0, & 0 & 2, & 3, & 0, & 0, & 0, & 0, & 0 \\ 1, & 8, & 0, & 0, & 0, & 0, & 0 & 1, & 3, & 0, & 0, & 0, & 0, & 0 \end{array}\right]$$

となる。現バージョンでは1方程式中の未知数は7個までしか扱えないがDimensionを変更することにより拡張が可能である。

付録E　サブルーチン負荷率

高速実験炉「常陽」自然循環解析問題について，サブルーチン負荷率の記述を始める前にプログラム改良の説明を行なう。

COMMIX－1AによるR－Z 2次元体系解析では，2次元体系でありながらθ方向流速が存在してしまう。これは，



の様な解析体系（$\theta = 2\pi$）を考えた場合，各セルの隣接セル情報が，

	<MO＝1>	<MO＝2>	<MO＝3>
MIM	＝－7	＝ 1	＝ 2
MIP	＝ 2	＝ 3	＝－10
MJM	＝ 1	＝ 2	＝ 3
MJP	＝ 1	＝ 2	＝ 3
MKM	＝－1	＝－2	＝－3
MKP	＝ 7	＝ 9	＝ 8

注）負の値は体系境界である。

となり，θ方向（J方向）については着目セル番号と同じ番号が係数行列中に格納されてしまうことによるものである。すなわち，圧力方程式は，

$$A_0 P_0 - A_1 P_1 - A_2 P_2 - A_3 P_0 - A_4 P_0 - A_5 P_5 - A_6 P_6 = B_0$$

という形になっている。そこで，体系が2次元であるということを利用し，$A_3 P_0$および$A_4 P_0$の項を右辺に移項し定数項に含めることとした。したがって圧力方程式は

$$A_0 P_0 - A_1 P_1 - A_2 P_2 - A_5 P_5 - A_6 P_6 = (B_0 + A_3 P_0 + A_4 P_0)$$

となり，θ方向情報が削除された形となる。

以上のプログラム改良による効果をサブルーチン負荷率により確認した。この負荷率測定には，富士通㈱提供の FORTUNE（Fortran Tuner）を使用した。

定常状態到達までに使用した各サブルーチン単位の負荷率を Table E. 1 に示す。この Table 中には，オリジナル・プログラムの結果とθ方向情報を削除（S. YMOMI Skip）した改良プログラムの結果の2種類を載せてある。オリジナル・プログラムの結果では，R－Z 2次元解析であるにもかかわらずθ方向の運動方程式係数を計算する S. YMOMI の負荷が一番大きく，全く意味の無い計算部分に時間を取られているのが判かる。このオリジナル・プログラムによる定常解算出には，FACOM VP－100（Scalar 演算）で 1594.8 秒の時間を要している。改良プログラムによる時間は第 3.9 節で述べた通りであり，プログラムの改良によりCPU時間の約3割が節約されている。Figure E. 1 に定常状態に到達するまでのCPU時間の推移を両プログラムについて示す。結果では，$\Delta V_{max}/V_{max}$が10^{-4}以下になったあたりから両者に顕著な差が現われ始め，定常解に到達した段階でその差は最大を示している。S. TIMSTP およびS. SOLVIT に於ける改良プログラムの効果をそれぞれ Figure E. 2, E. 3 に示す。

Table E.1 Comparison of Sub-Program Occupation Ratio Between Original Version and Corrected Version

S.P. Name	Occupation Ratio (%)		Note
	S. YMOMI is Calculated	S. YMOMI is Skipped	
YMOMI	26.1	——	
SOLVIT	22.1	25.3	
ZMOMI	14.6	21.2	
XMOMI	13.1	19.1	
PEQN	6.2	9.0	
GDCONV	6.0	8.7	
GETDL	2.7	3.7	
VISLIQ	2.7	3.9	
MOMENI	1.2	1.8	
REBAZG	1.1	1.5	
MOLOOP	0.8	1.1	
FILLM	0.2	0.2	
FORCES	0.2	0.3	
TIMSTP	0.2	0.3	
INPSTR	0.1	0.1	
IREBAL	0.1	0.1	
RSET2	0.1	0.1	
Total	97.5	96.4	
CPU Time on BEP1 (Sec.)	1594.8	1121.3	

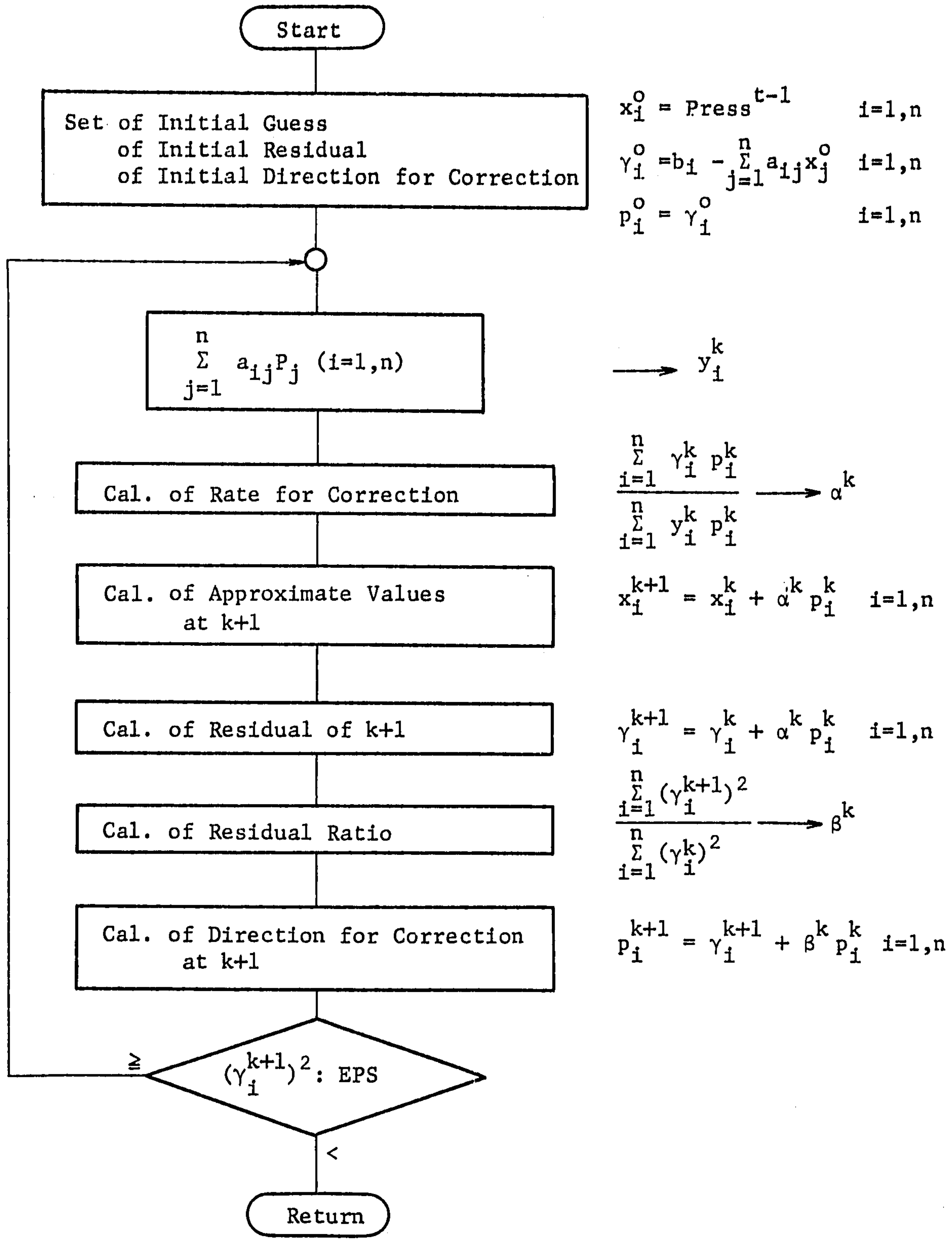


Fig. 2.1 C.G. Algorithm (Hestenes-Stiefel version) for Symmetrical Matrix

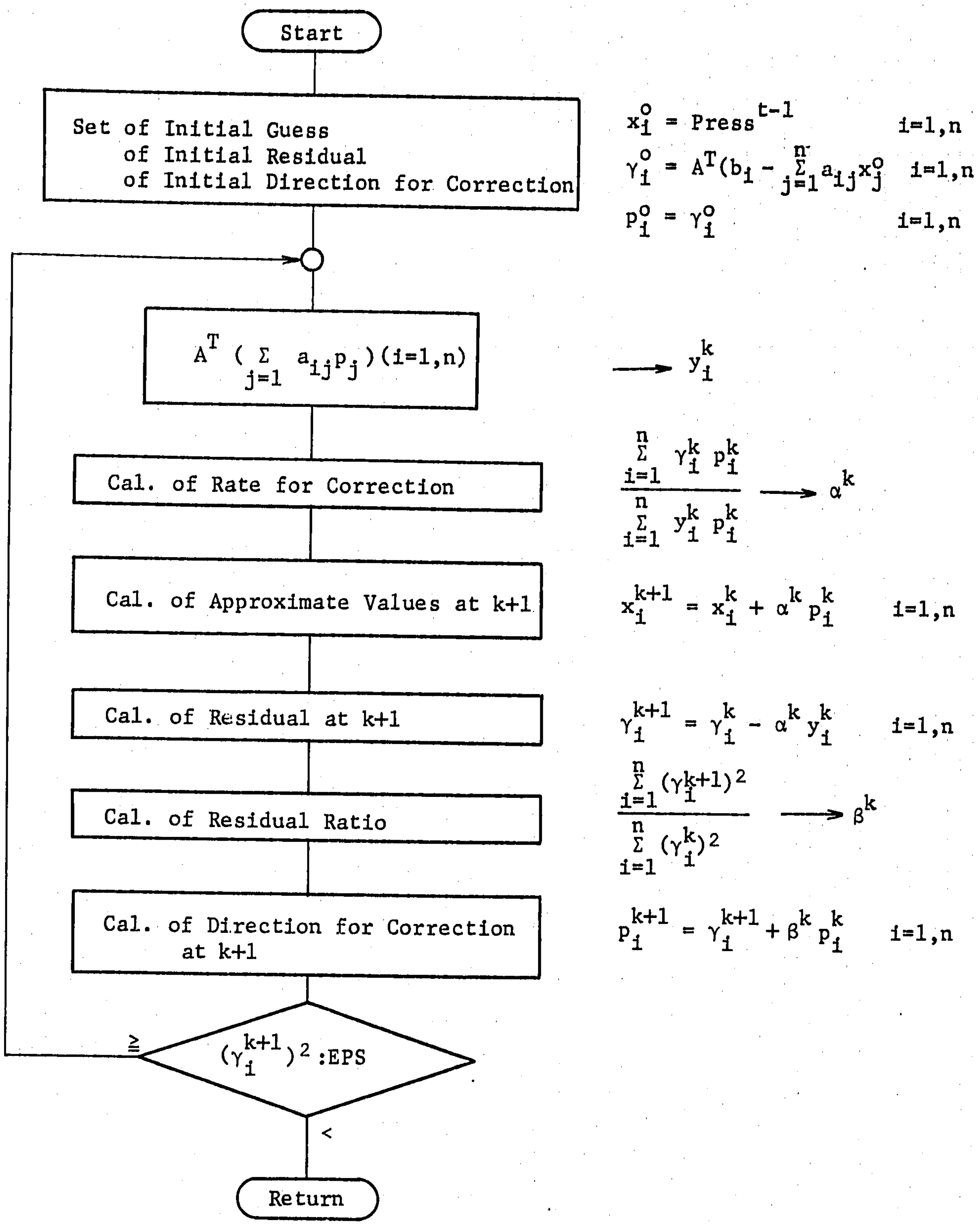
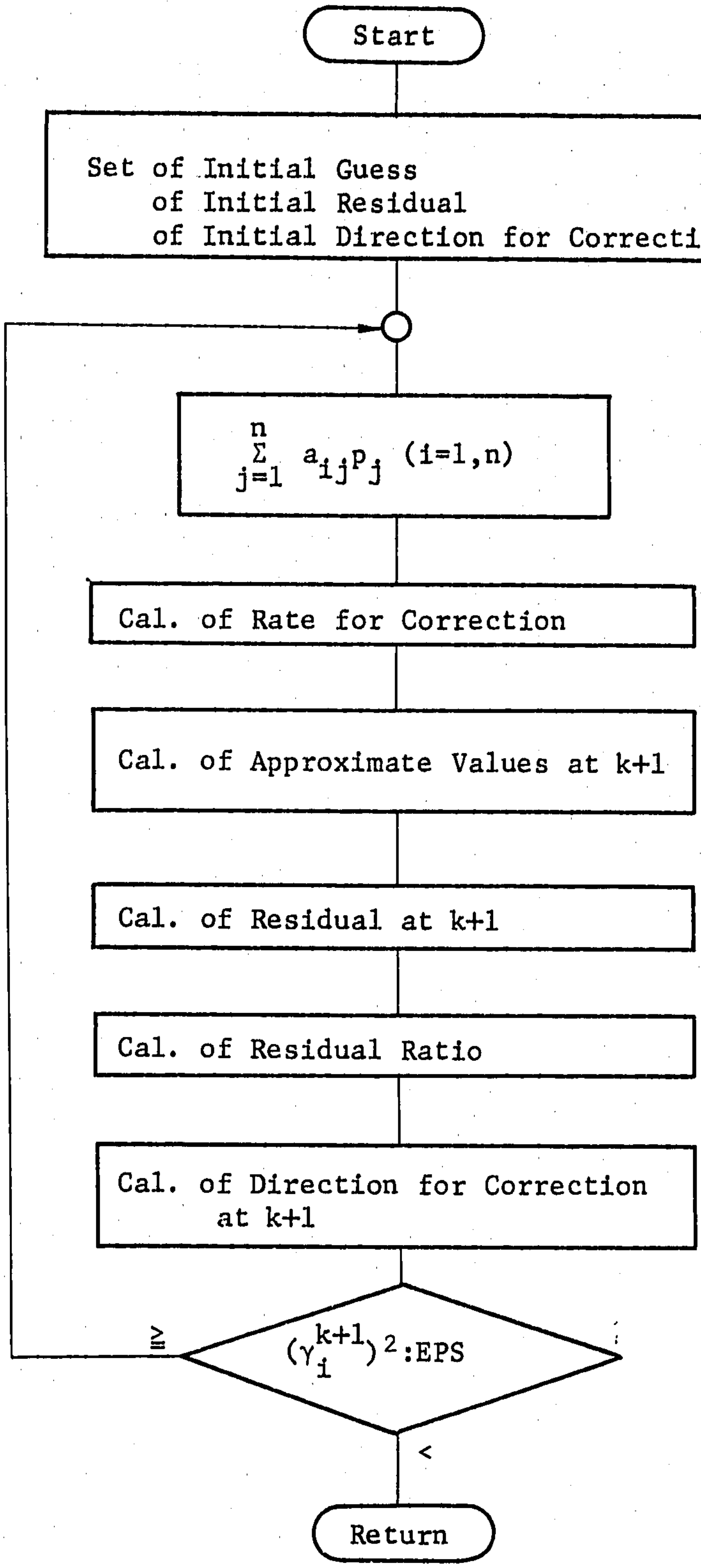


Fig. 2.2 C.G. Algorithm for Unsymmetrical Matrix



$$x_i^o = \mathrm{Press}^{t-1} \qquad i=1,n$$

$$\gamma_i^o = b_i = \sum_{j=1}^{n} a_{ij}x_j^o \qquad i=1,n$$

$$p_i^o = (LL^T)^{-1}\gamma_i^o \qquad i=1,n$$

$$\longrightarrow y_i^k$$

$$\frac{\sum_{i=1}^{n} \gamma_i^k (LL^T)^{-1}\gamma_i^k}{\sum_{i=1}^{n} y_i^k p_i^k} \longrightarrow \alpha^k$$

$$x_i^{k+1} = x_i^k + \alpha^k p_i^k \qquad i=1,n$$

$$\gamma_i^{k+1} = \gamma_i^k - \alpha^k p_i^k \qquad i=1,n$$

$$\frac{\sum_{i=1}^{n} (\gamma_i^{k+1}, (LL^T)^{-1}\gamma_i^{k+1})}{\sum_{i=1}^{n} (\gamma_i^k, (LL^T)^{-1}\gamma_i^k)} \longrightarrow \beta^k$$

$$p_i^{k+1} = (LL^T)^{-1}\gamma_i^{k+1} + \beta^k p_i^k \qquad i=1,n$$

Fig. 2.3 I.C.C.G. Algorithm

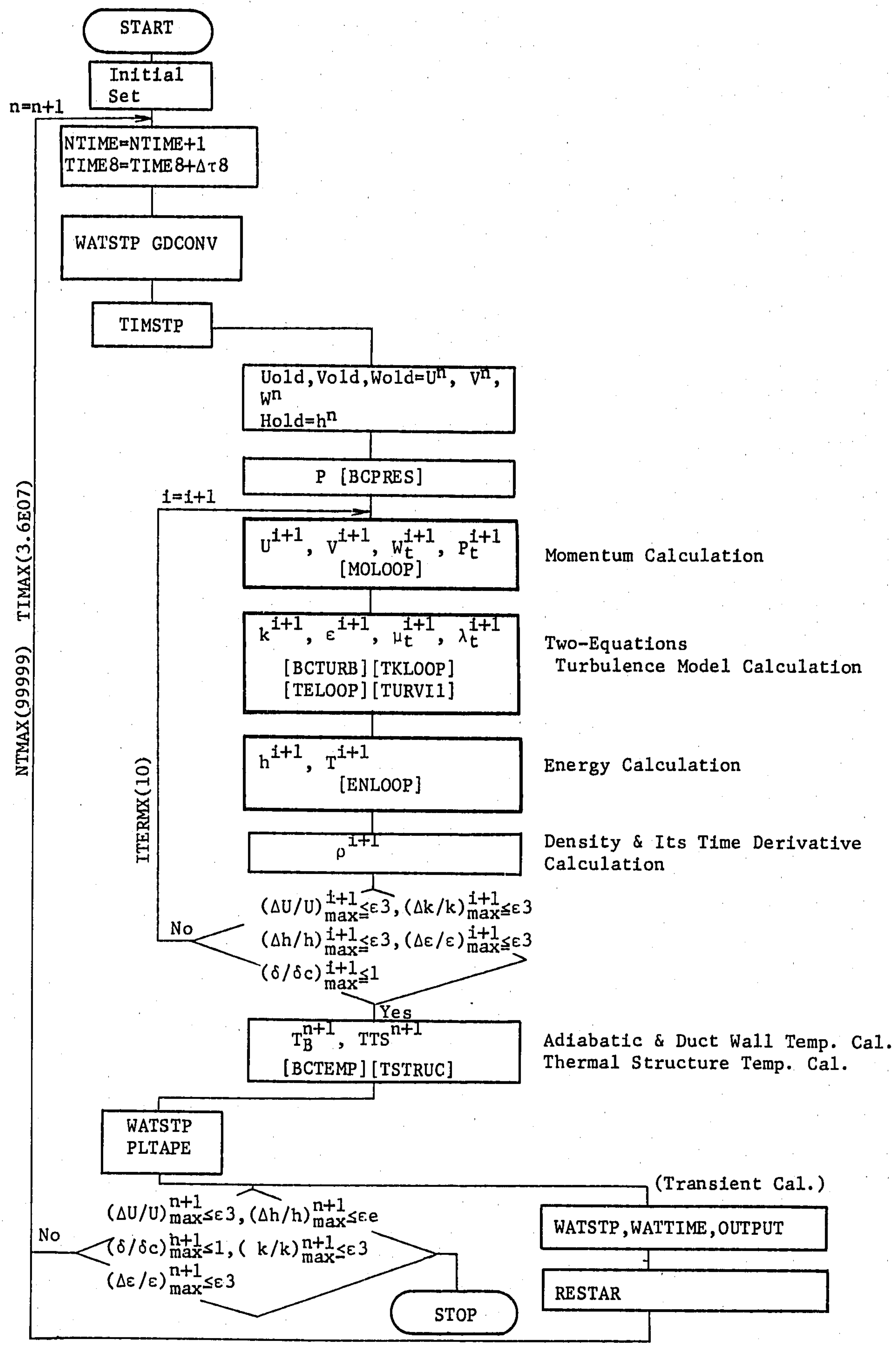


Fig. 2.4 Overall Flow Chart of COMMIX-1A

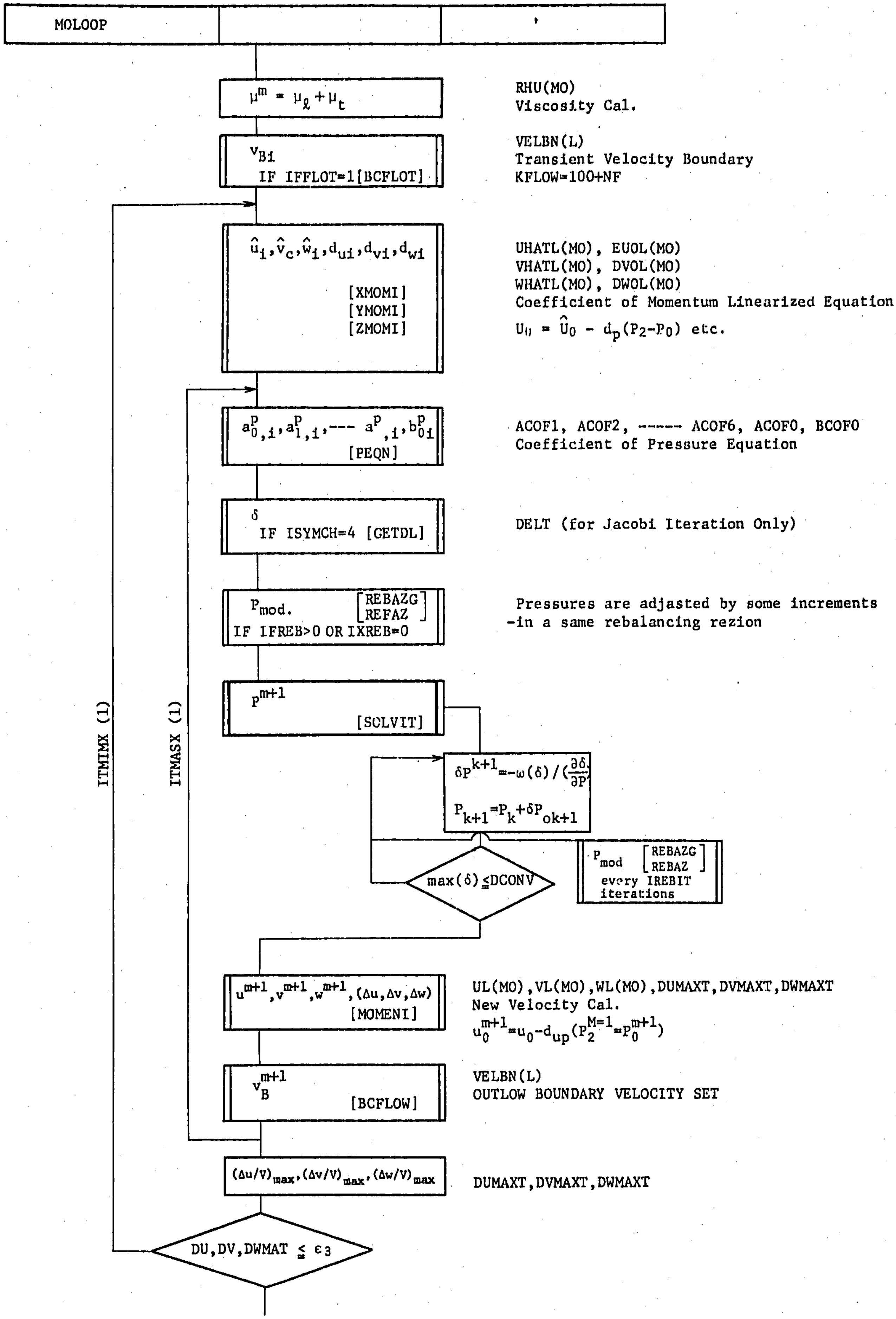


Fig. 2.5 General Flow Chart of SUBROUTINE MOLOOP

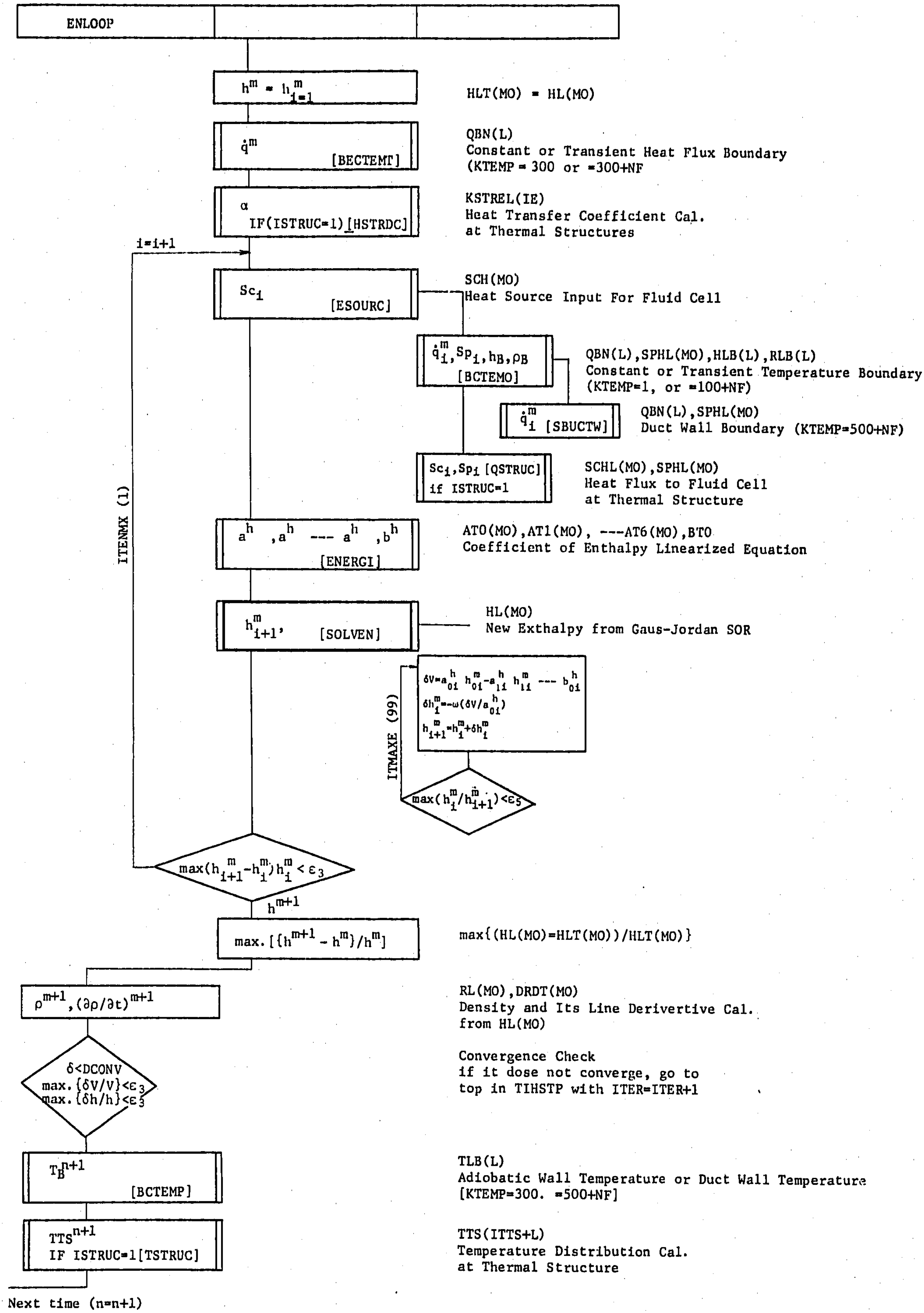


Fig. 2.6 General Flow Chart of SUBROUTINE ENLOOP

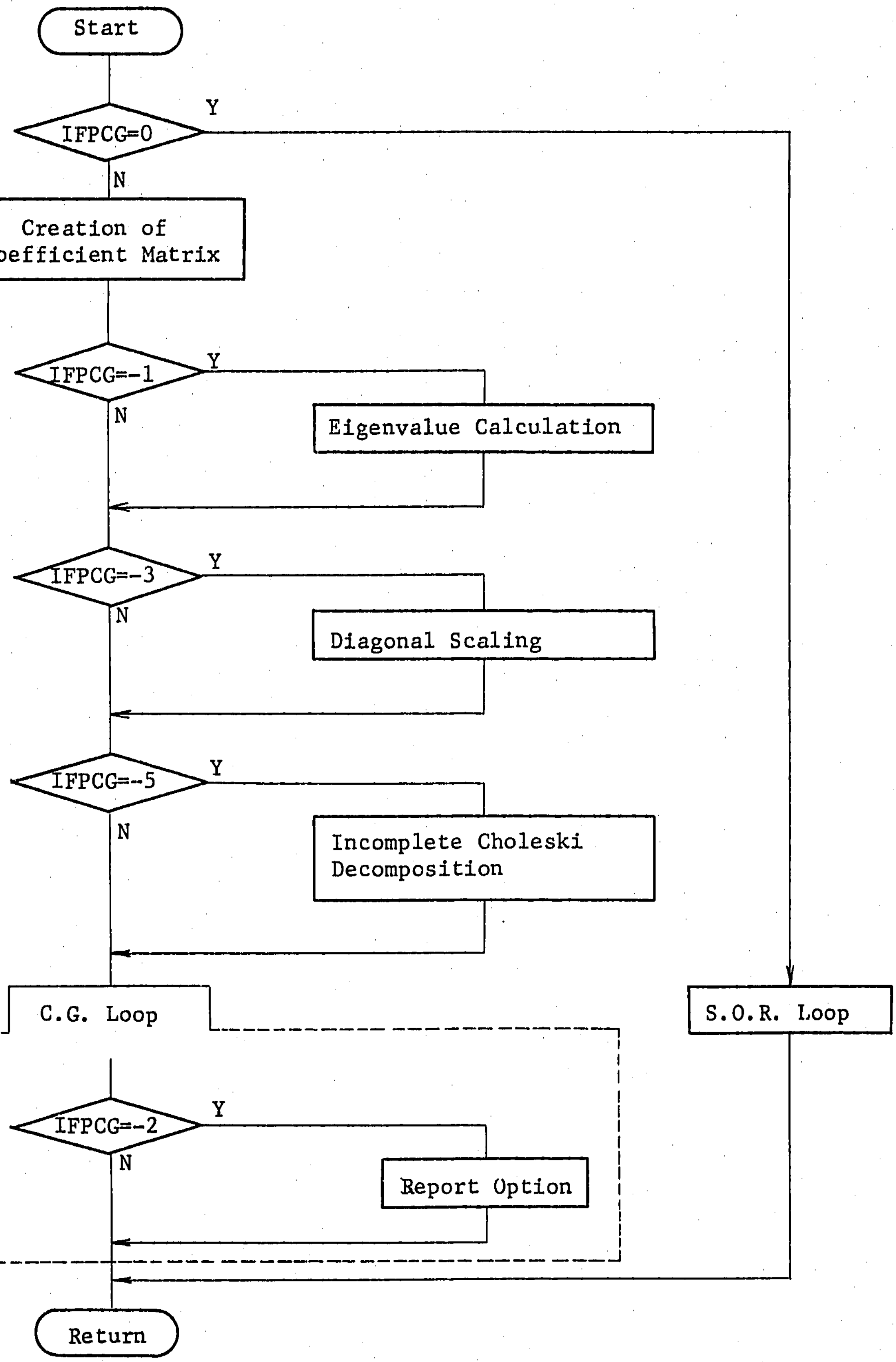


Fig. 2.7 Overall Flow Chart of P.C.G. Solver

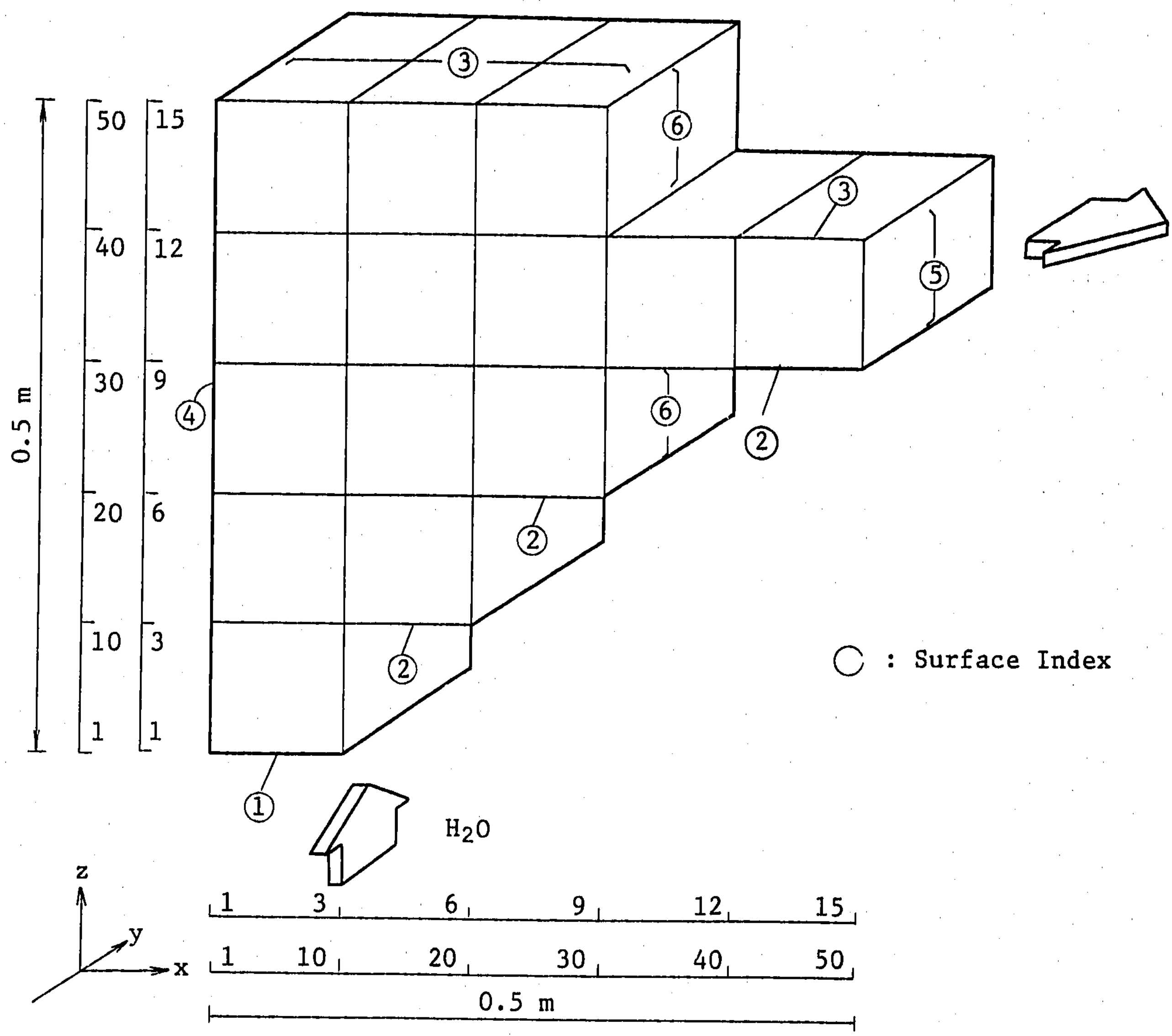


Figure 3.1 Mesh Arrangement for Fundamental Problems

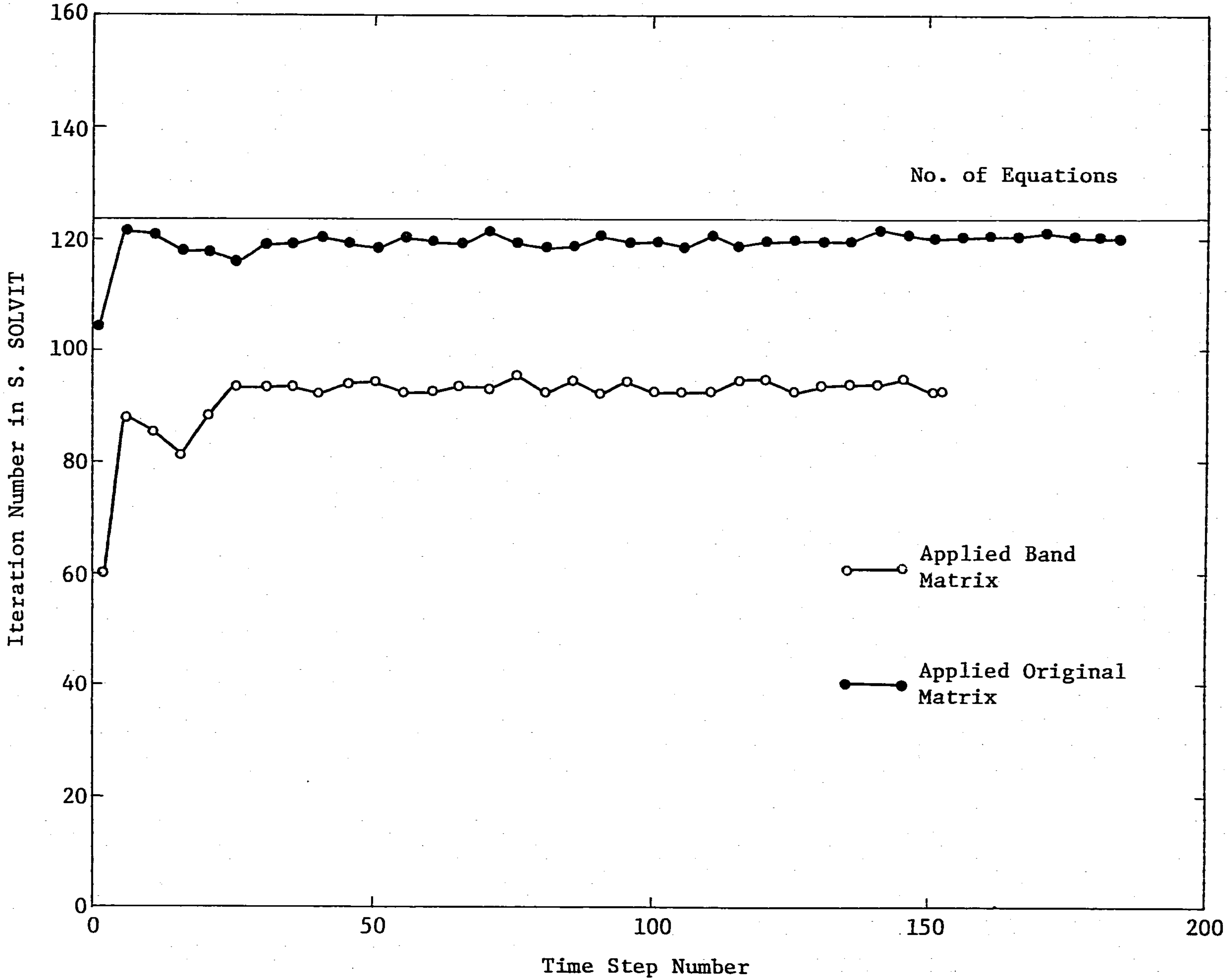


Figure 3.2 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = C.G. Method)

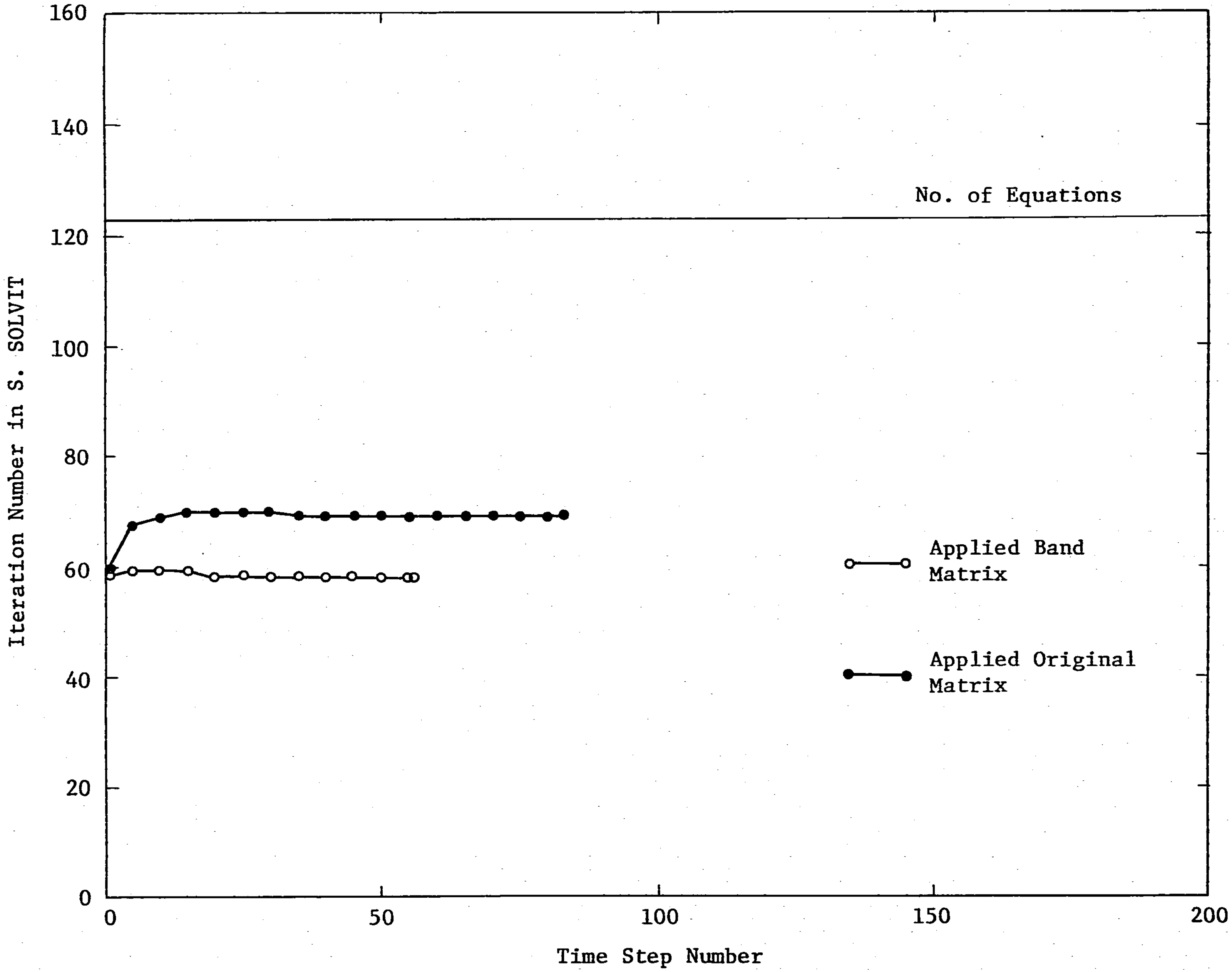


Figure 3.3 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = Scaling Method)

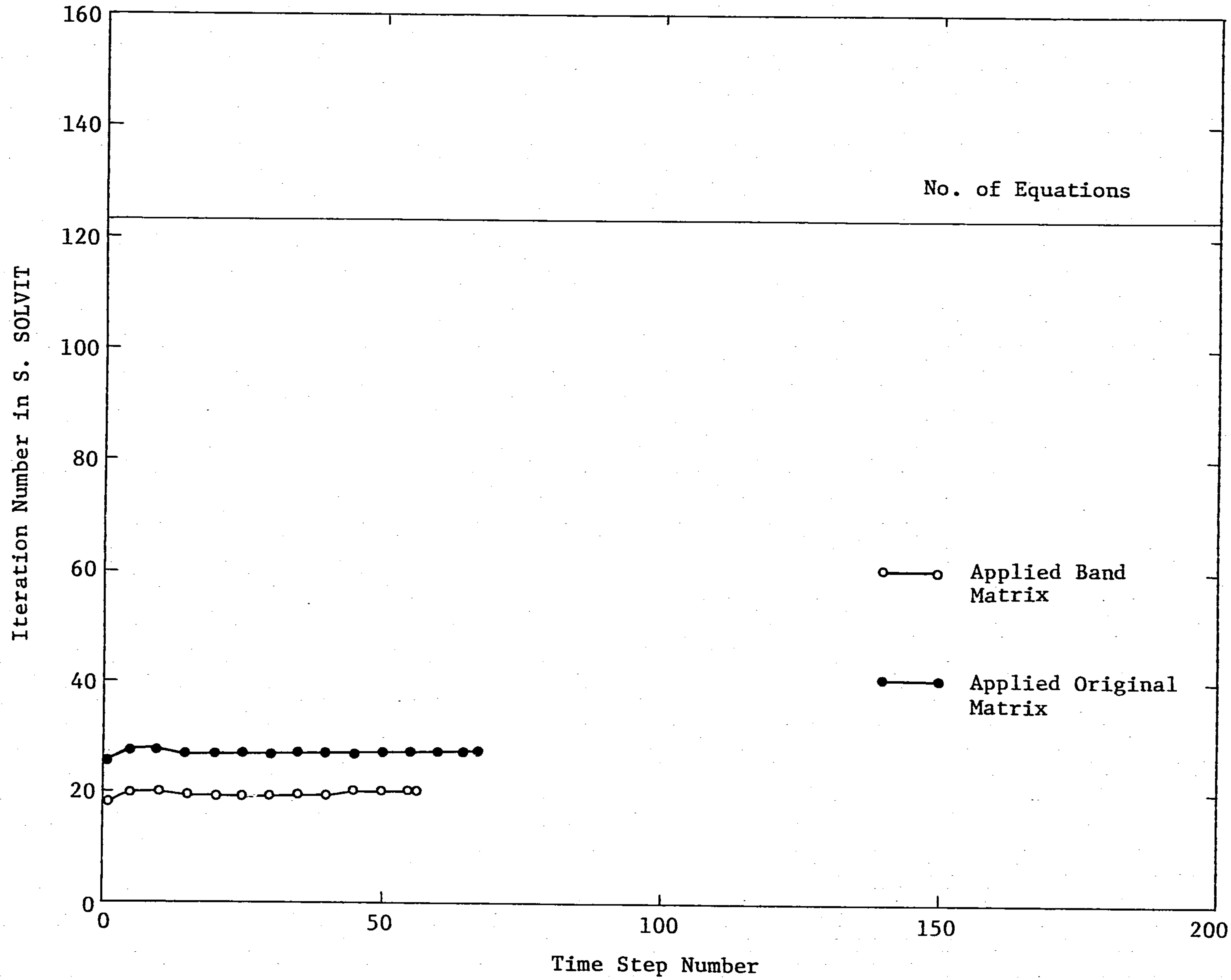


Figure 3.4 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 126 Cells Problem (Iteration Scheme = I.C.C.G. Method)

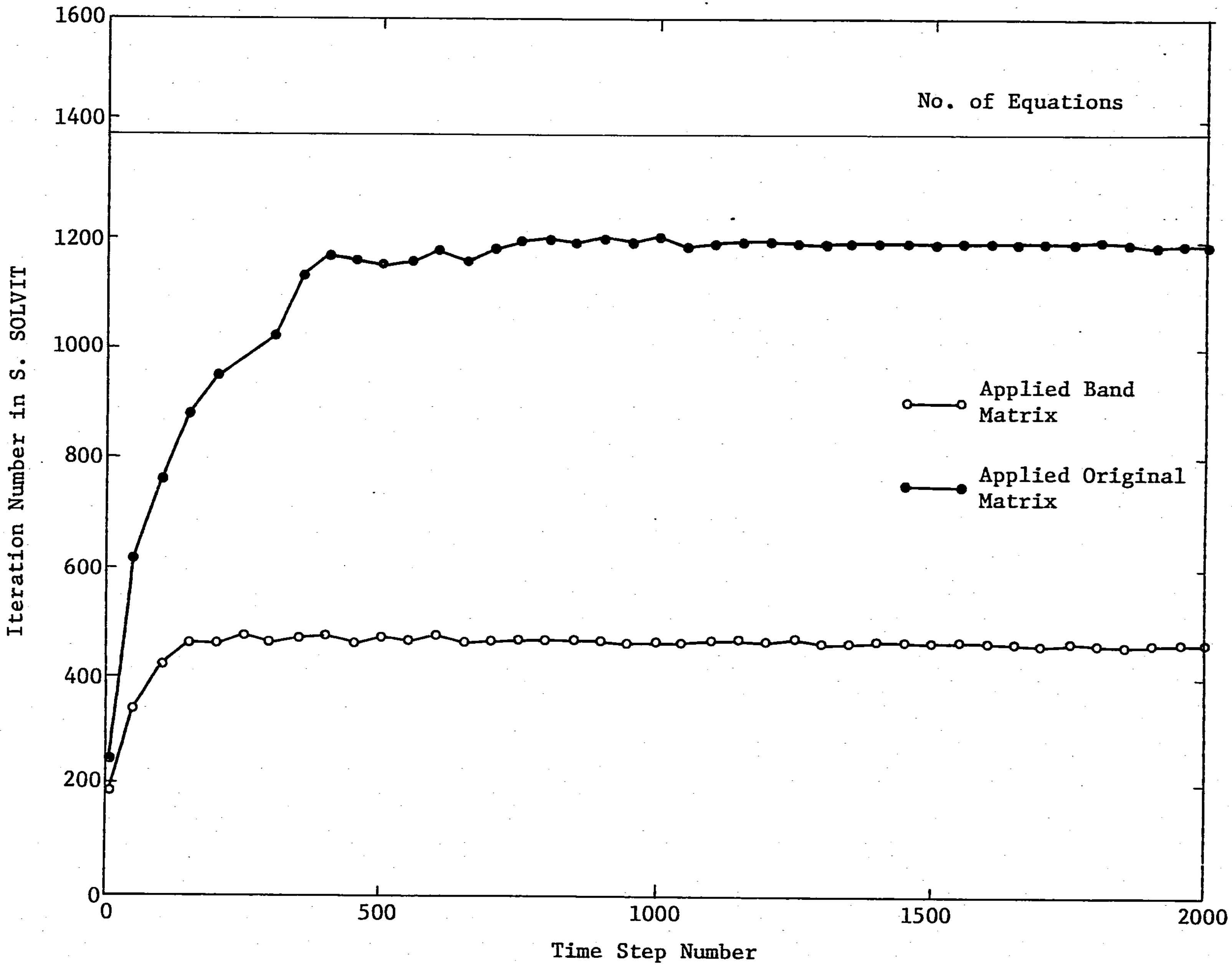


Figure 3.5　Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Soheme = C.A. Method)

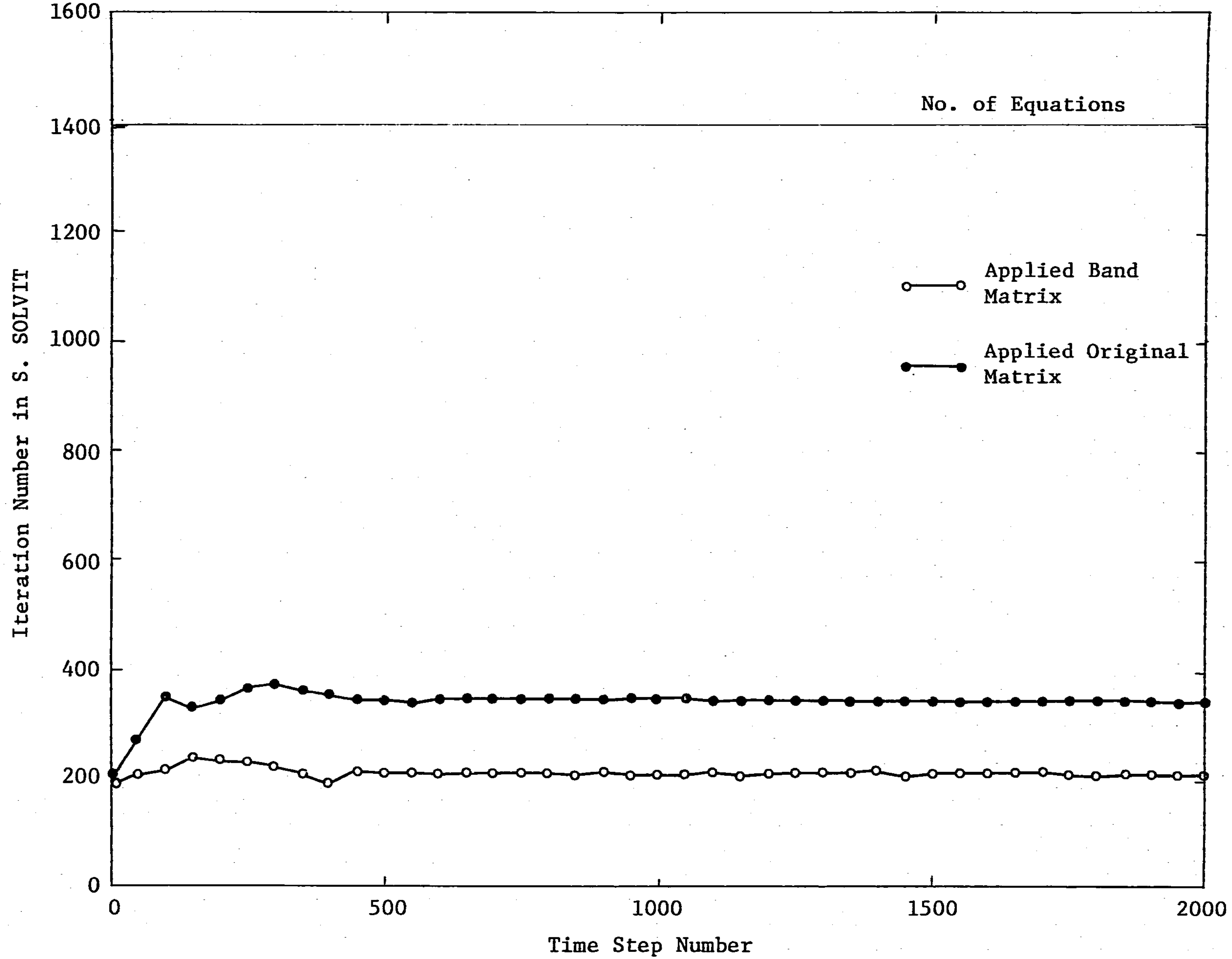


Figure 3.6 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Soheme = Scaling Method)

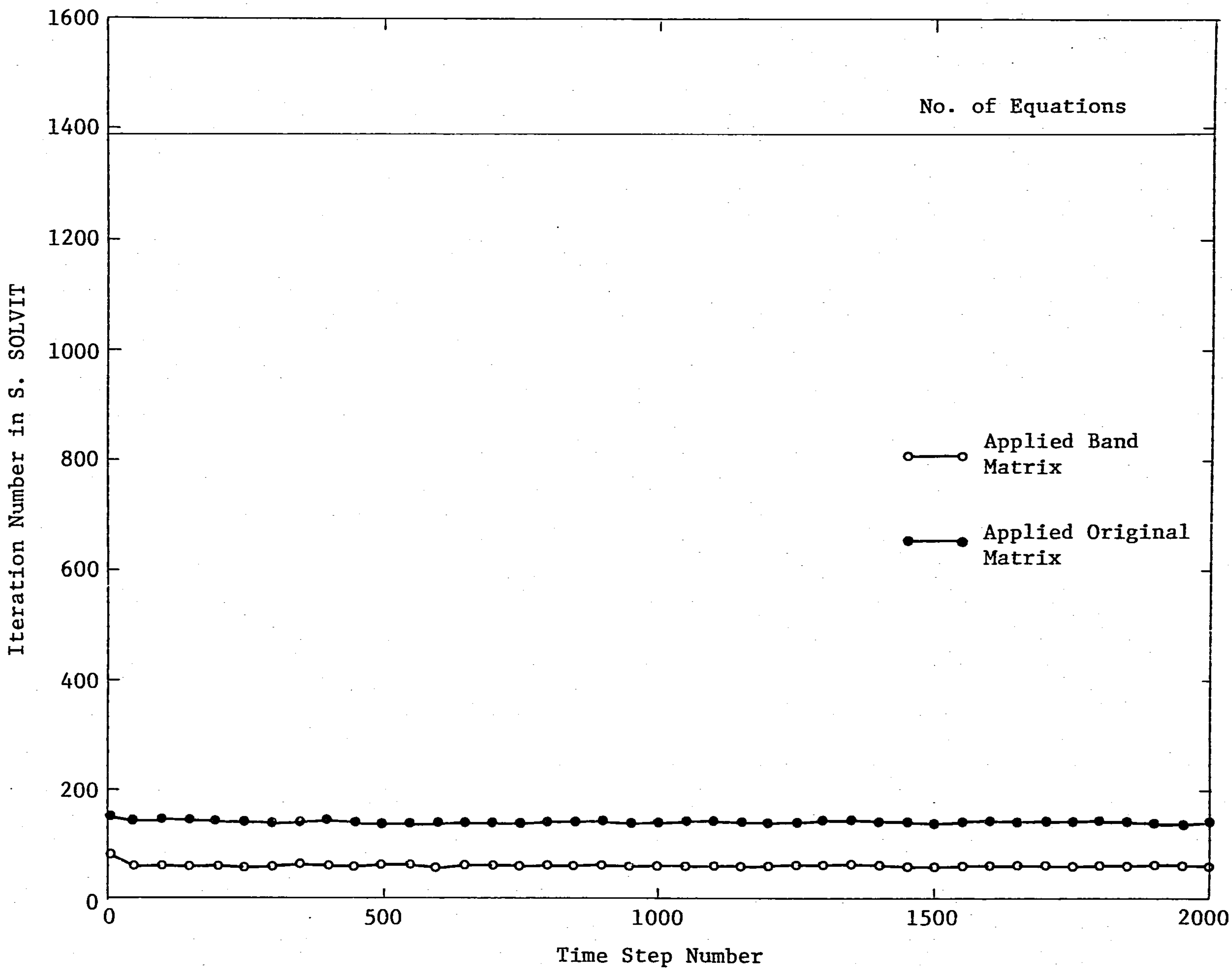


Figure 3.7 Effect of Band Matrix Width about Iteration Number in S. SOLVIT on 1400 Cells Problem (Iteration Scheme = I.C.C.G. Method)

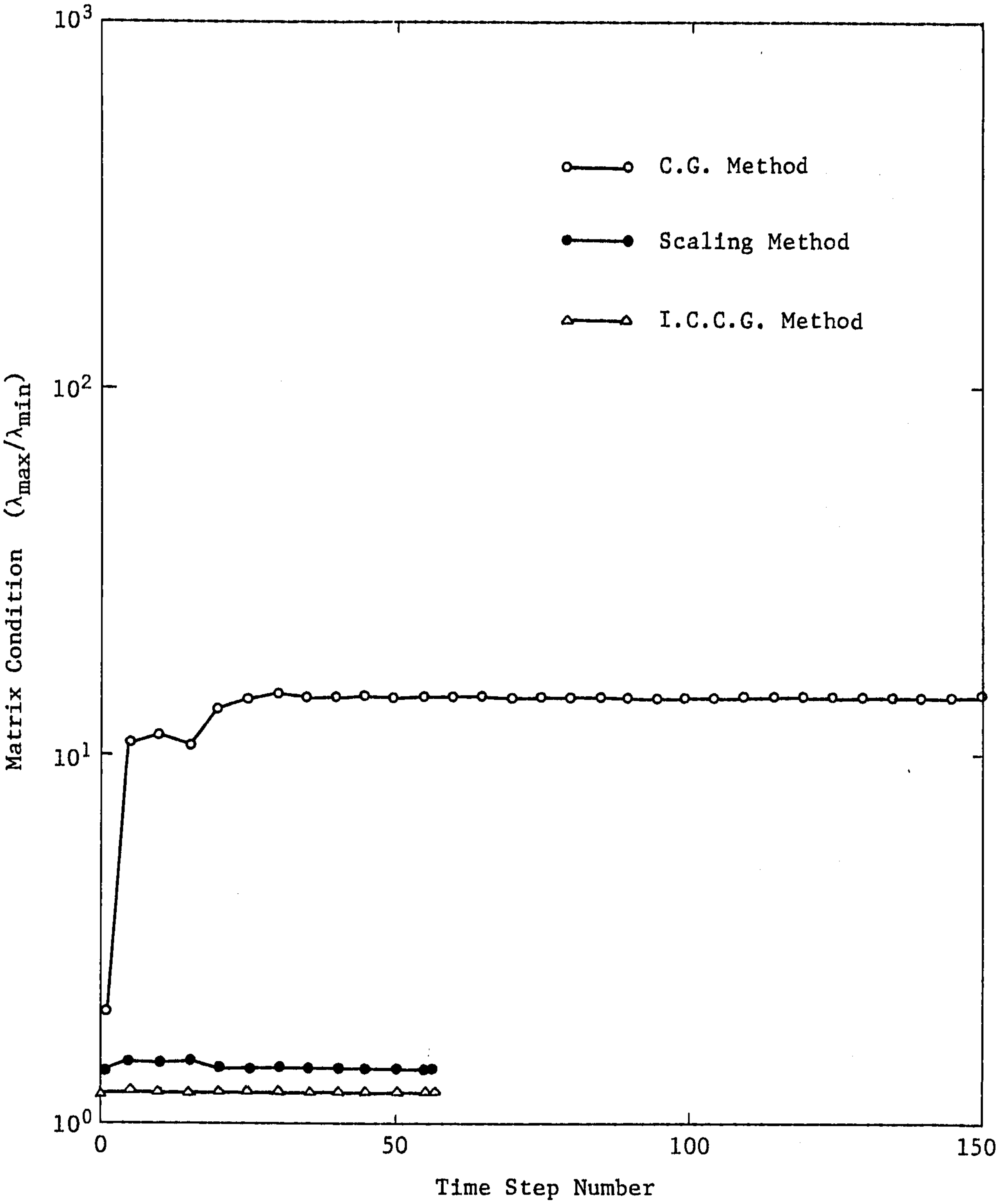


Figure 3.8 Comparison of Matrix Condition of Each Scheme on 126 Cells Problem

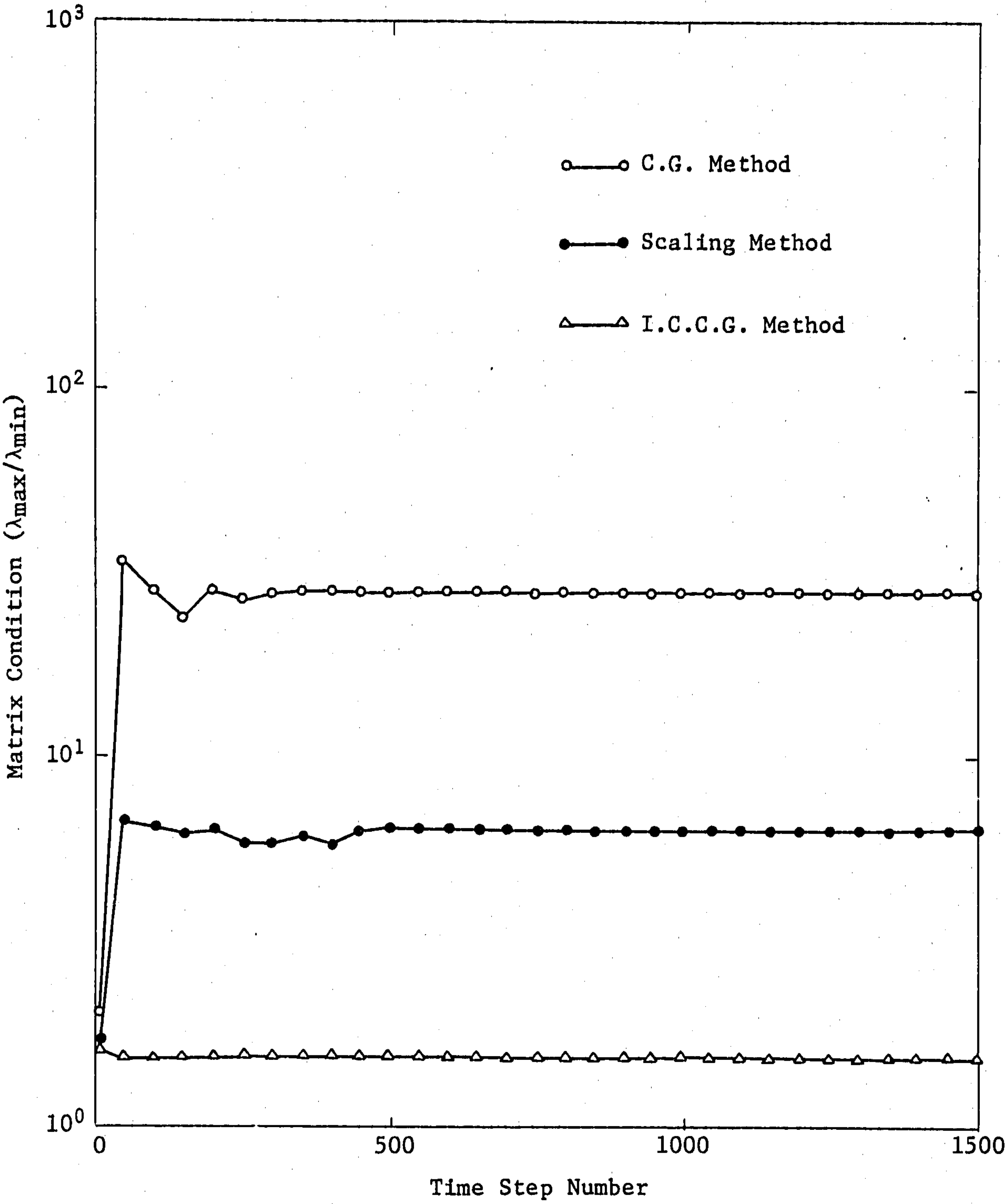


Figure 3.9 Comparison of Matrix Condition of Each Scheme on 1400 Cells Problem

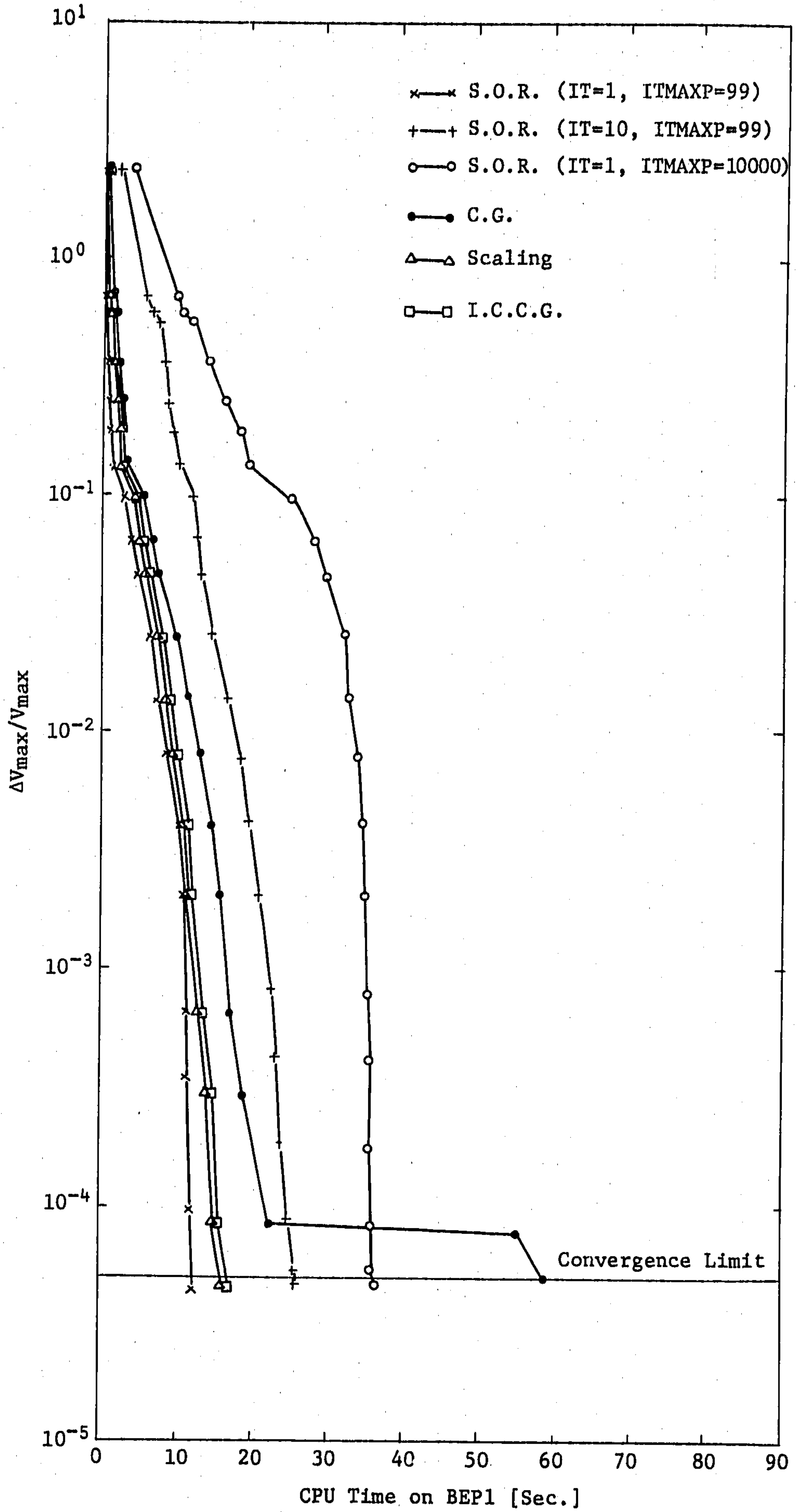


Figure 3.10 Compalison of Convergency of Each Scheme on 126 Cells Problem

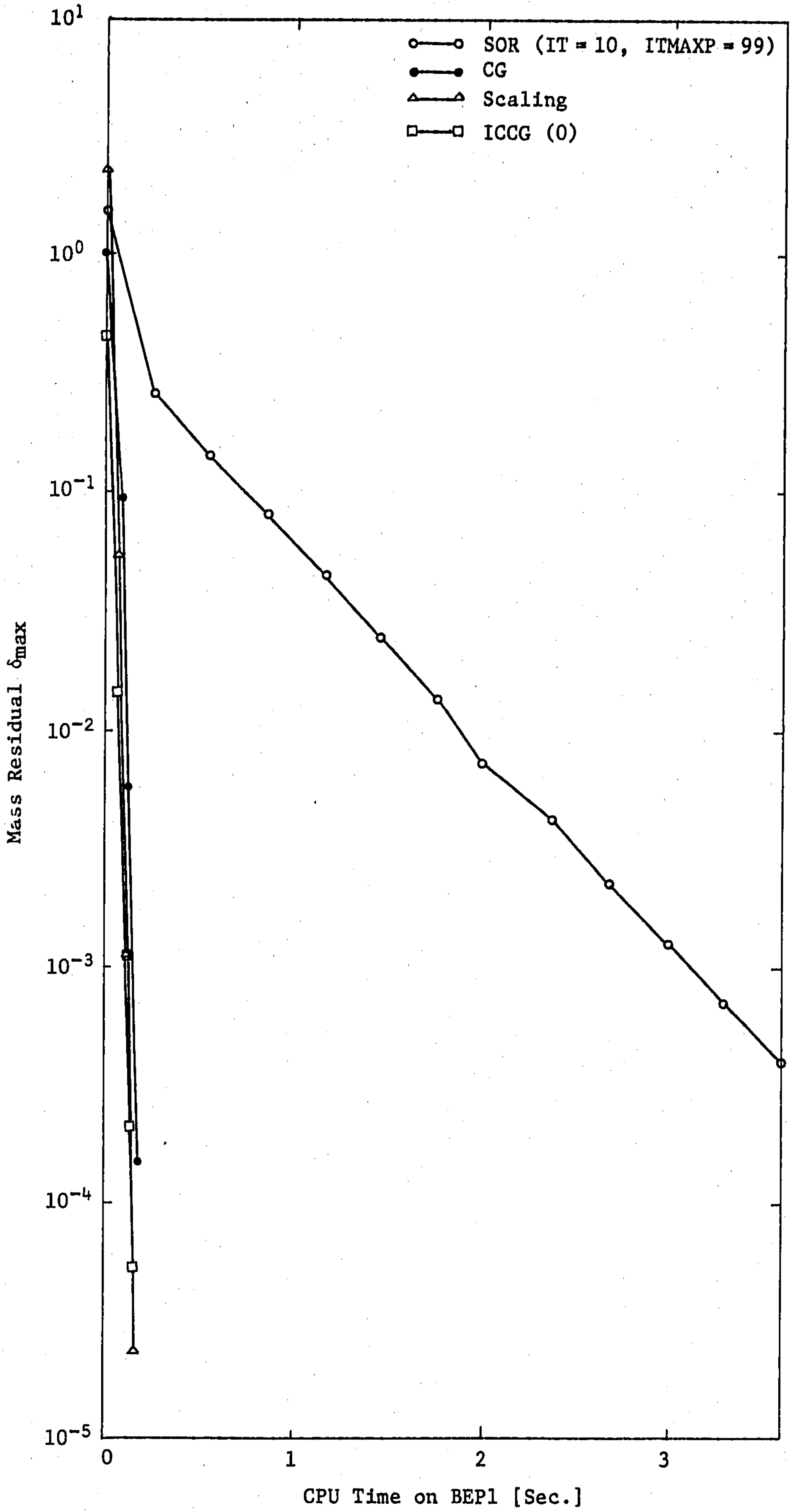


Figure 3.11 Comparison of Mass Balance Convergency of Each Scheme about 126 Cells Problem on Scalar Machine (Time Step = 1)

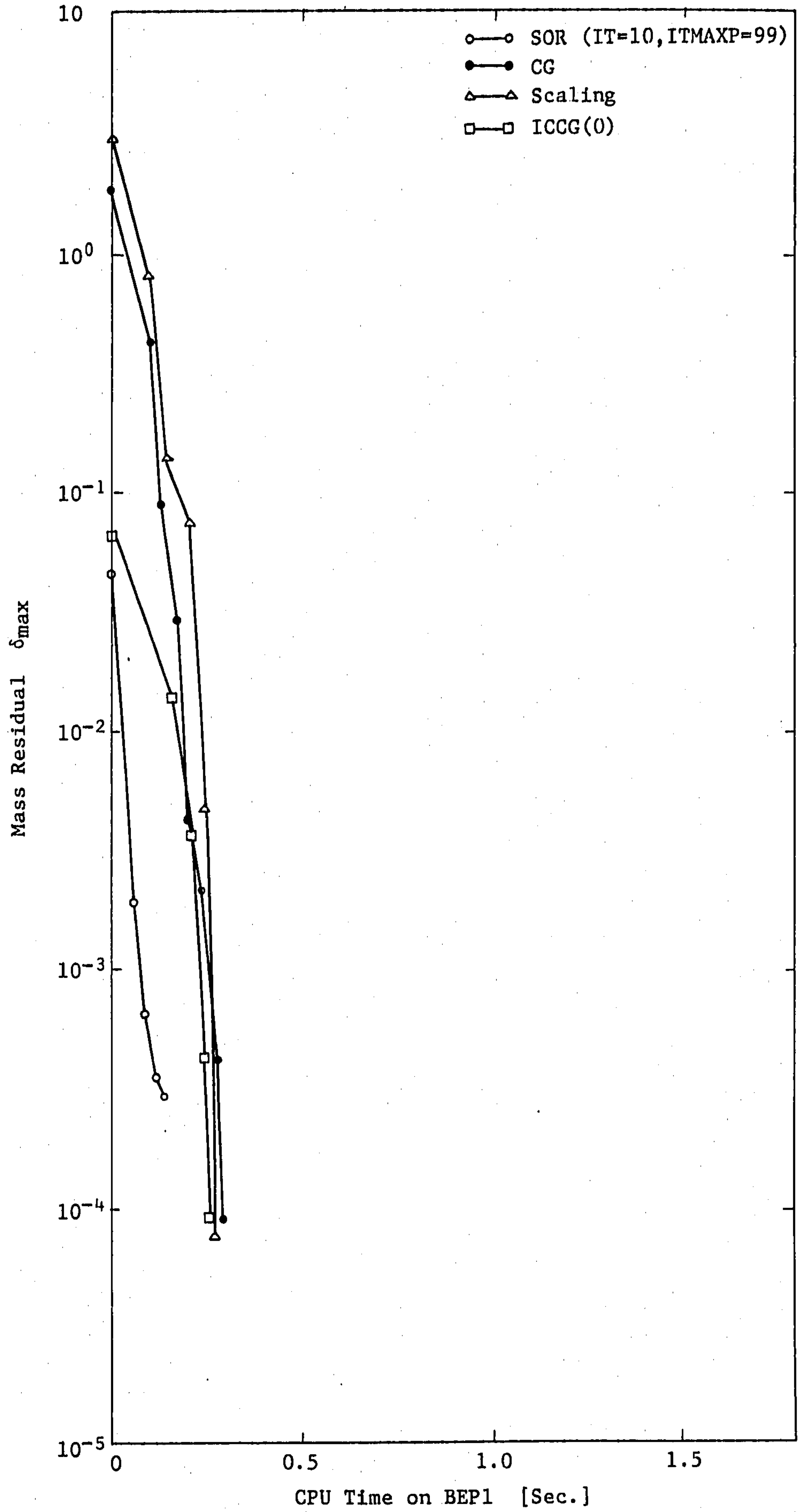


Figure 3.12 Comparison of Mass Balance Convergency of Each Scheme about 126 Cells Problem on Scalar Machine (Time Step = 28)

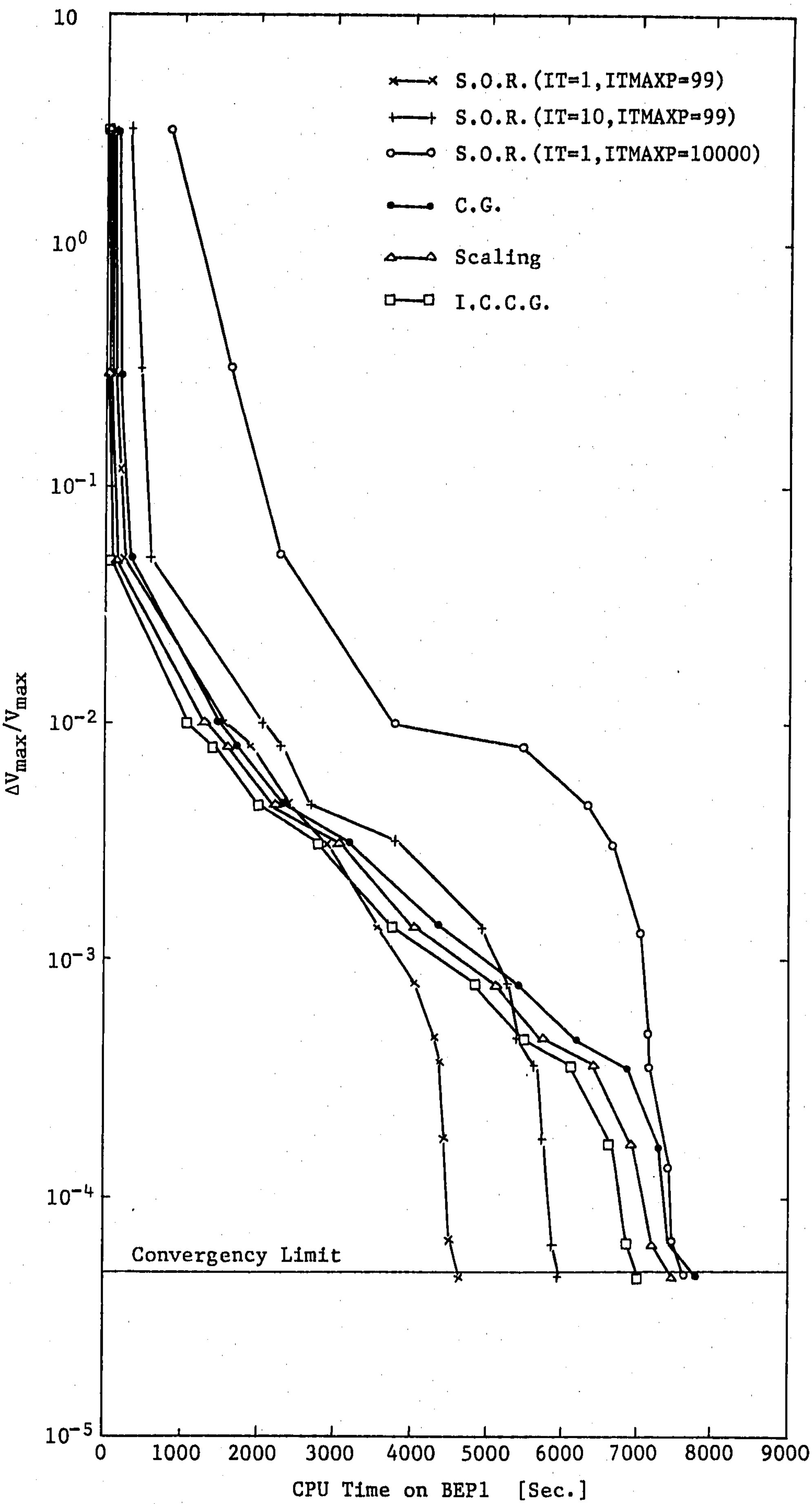


Figure 3.13 Comparison of Convergency of Each Scheme on 1400 Cells Problem

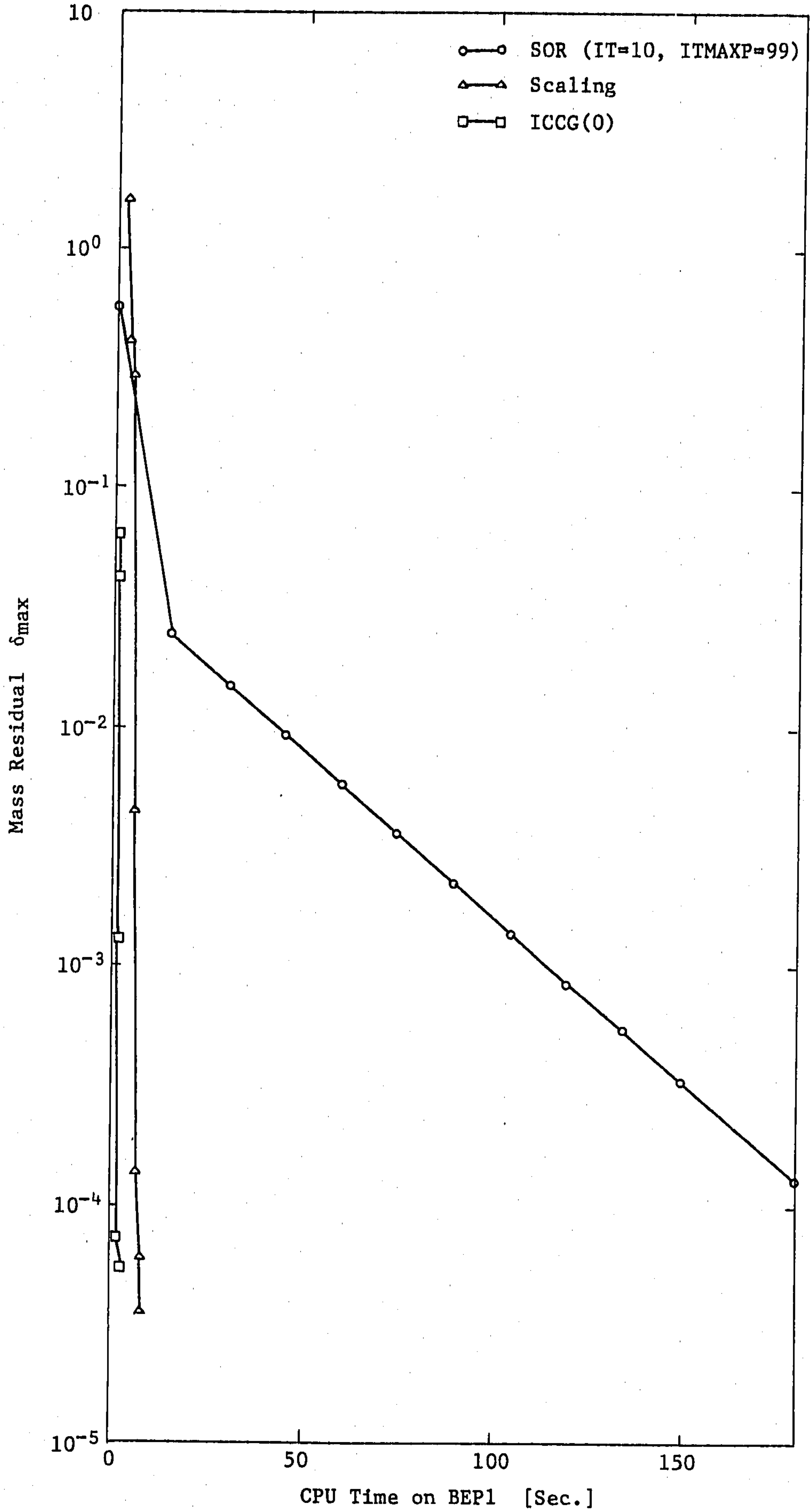


Figure 3.14 Comparison of Mass Balance Convergency of Each Scheme about 1400 Cells Problem on Scalar Machine (Time Step = 1)

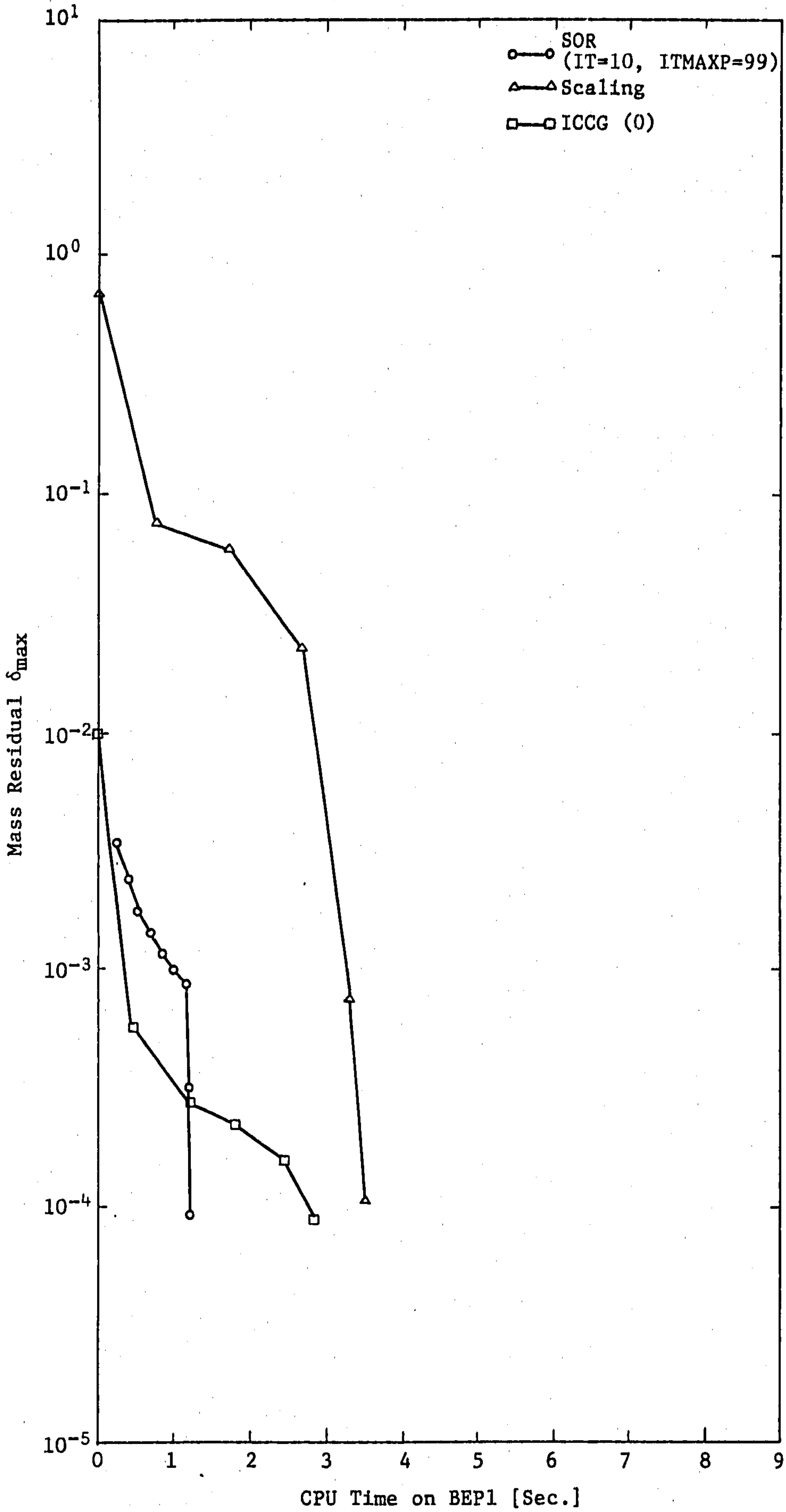


Figure 3.15 Comparison of Mass Balance Convergency of Each Scheme about 1400 Cells Problem on Scalar Machine (Time Step=600)

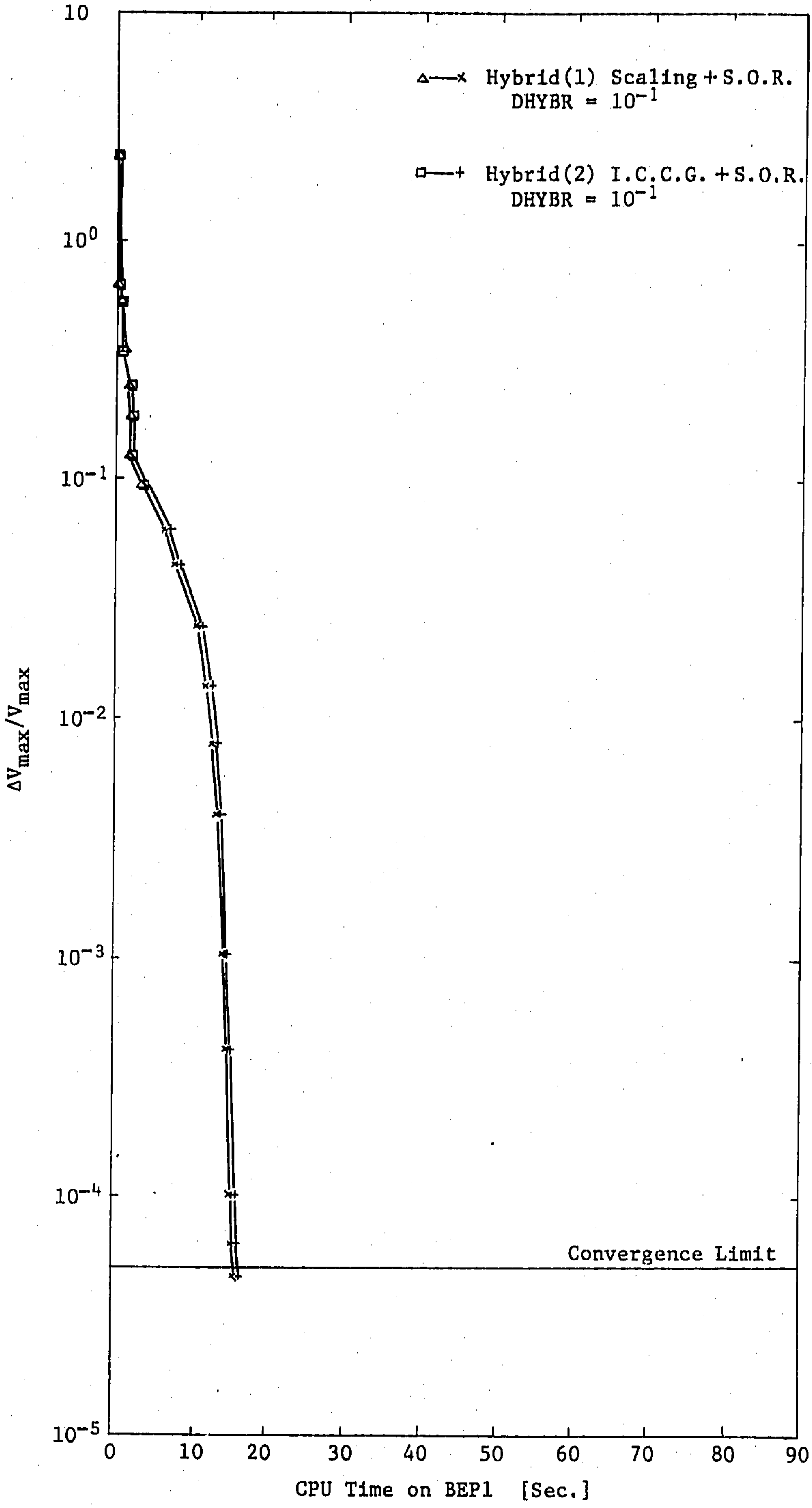


Figure 3.16 Effect of Hybrid Schemes on 126 Cells Problem

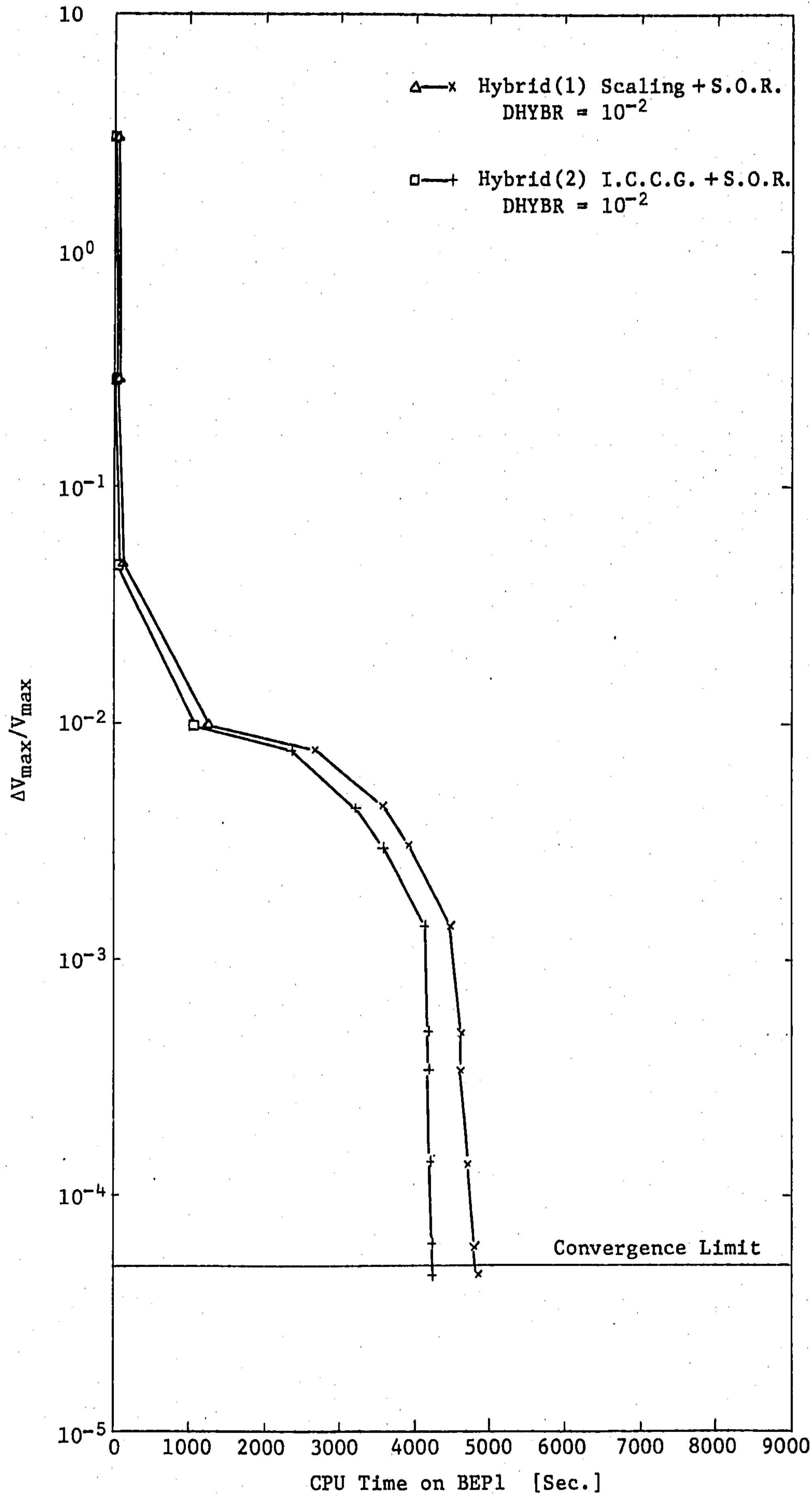


Figure 3.17 Effect of Hybrid Schemes on 1400 Cells Problem

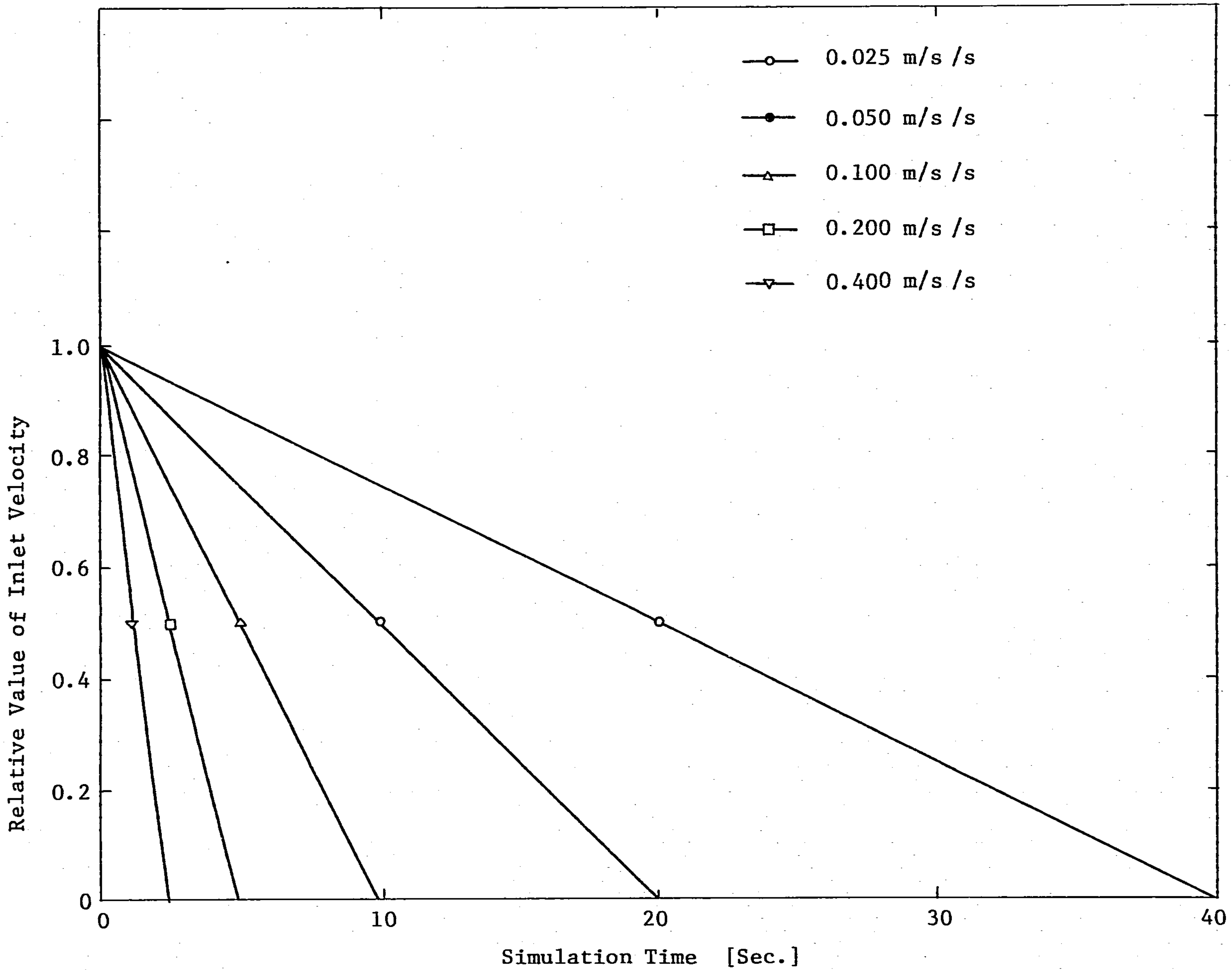


Figure 3.18 Velocity Transient at Inlet of 126 Cells Model

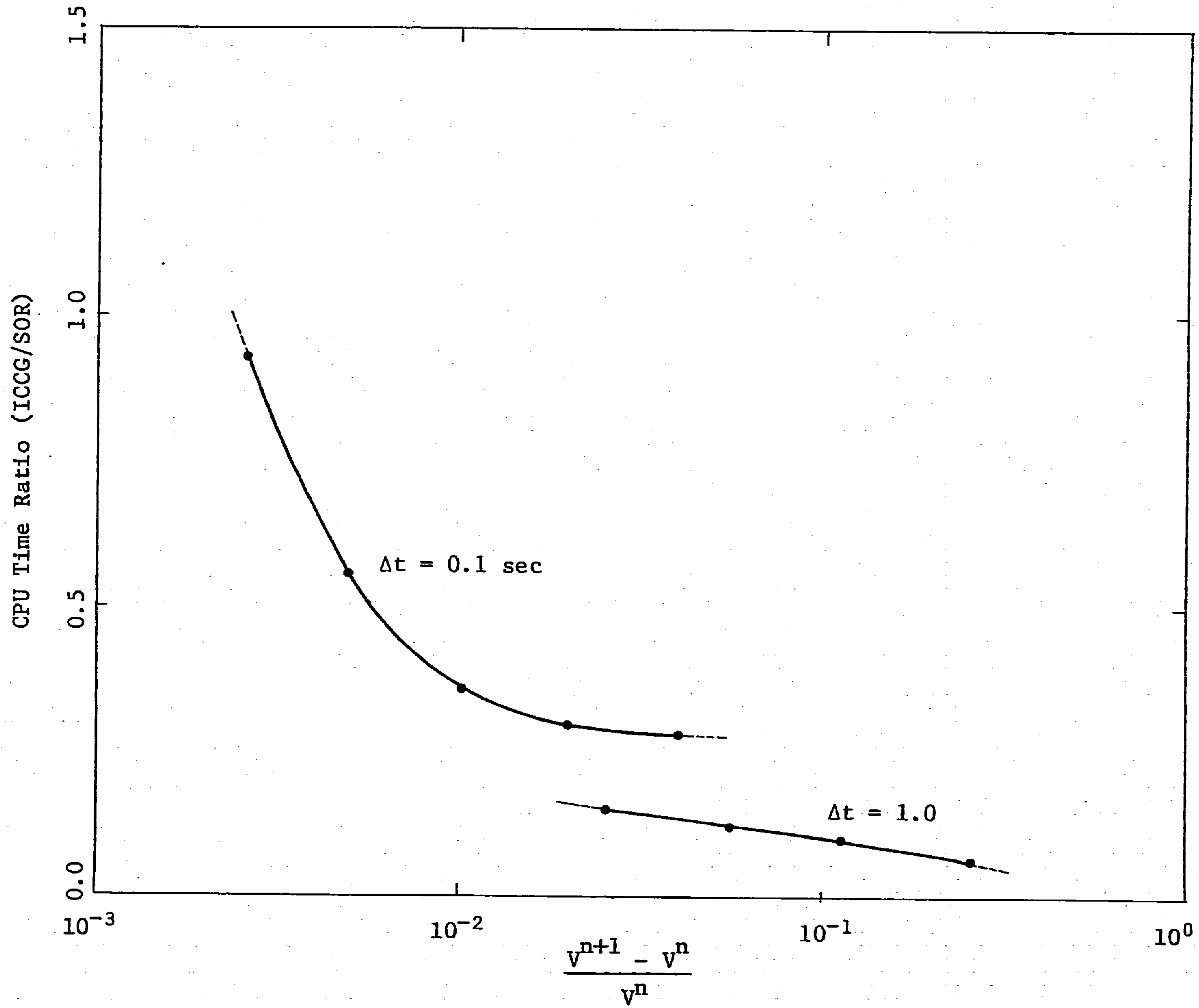


Fig. 3.19 CPU Time Ratio vs. Velocity Change Ratio

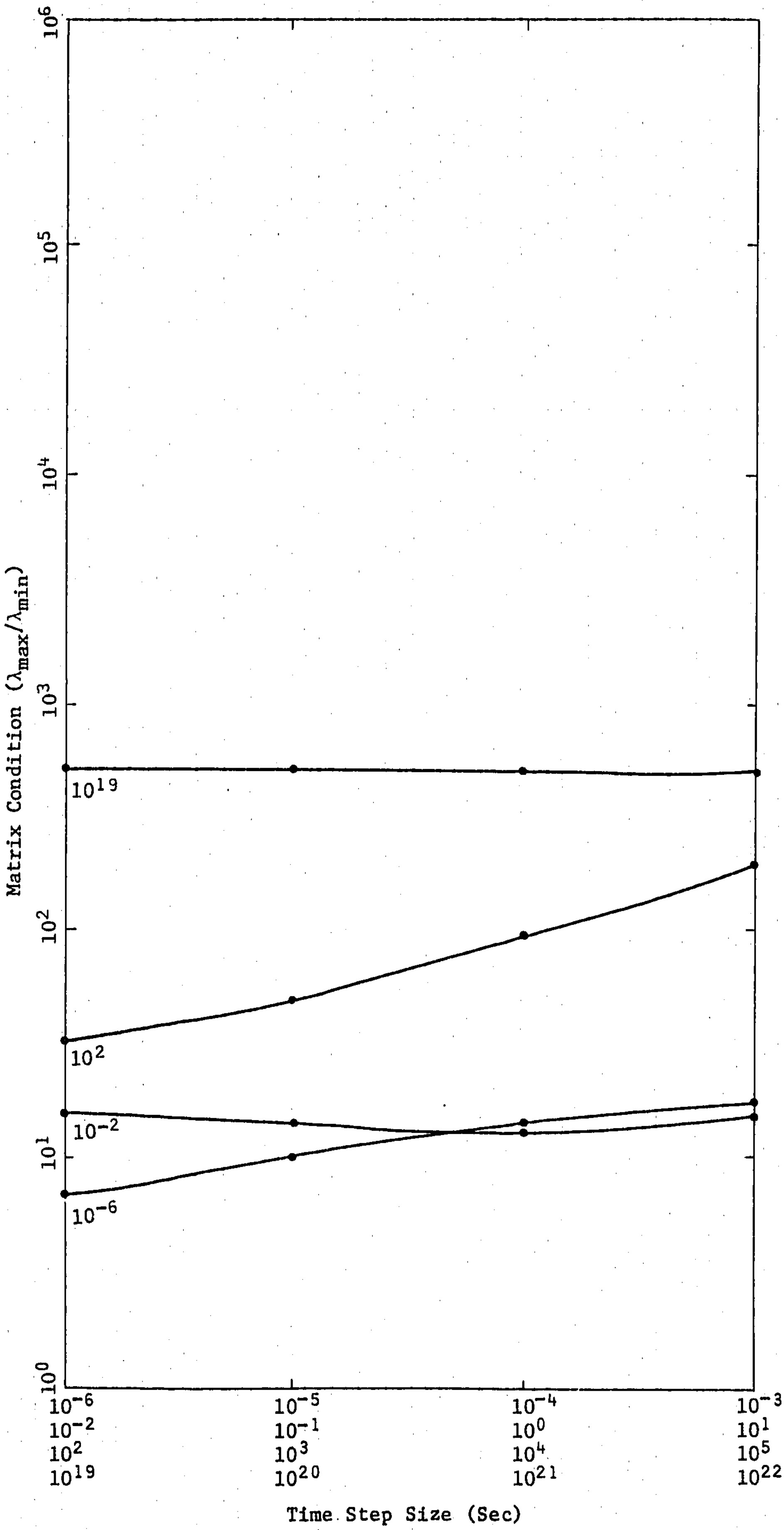


Fig. 3.20 Time Step Size Δt Dependency of Matrix Condition

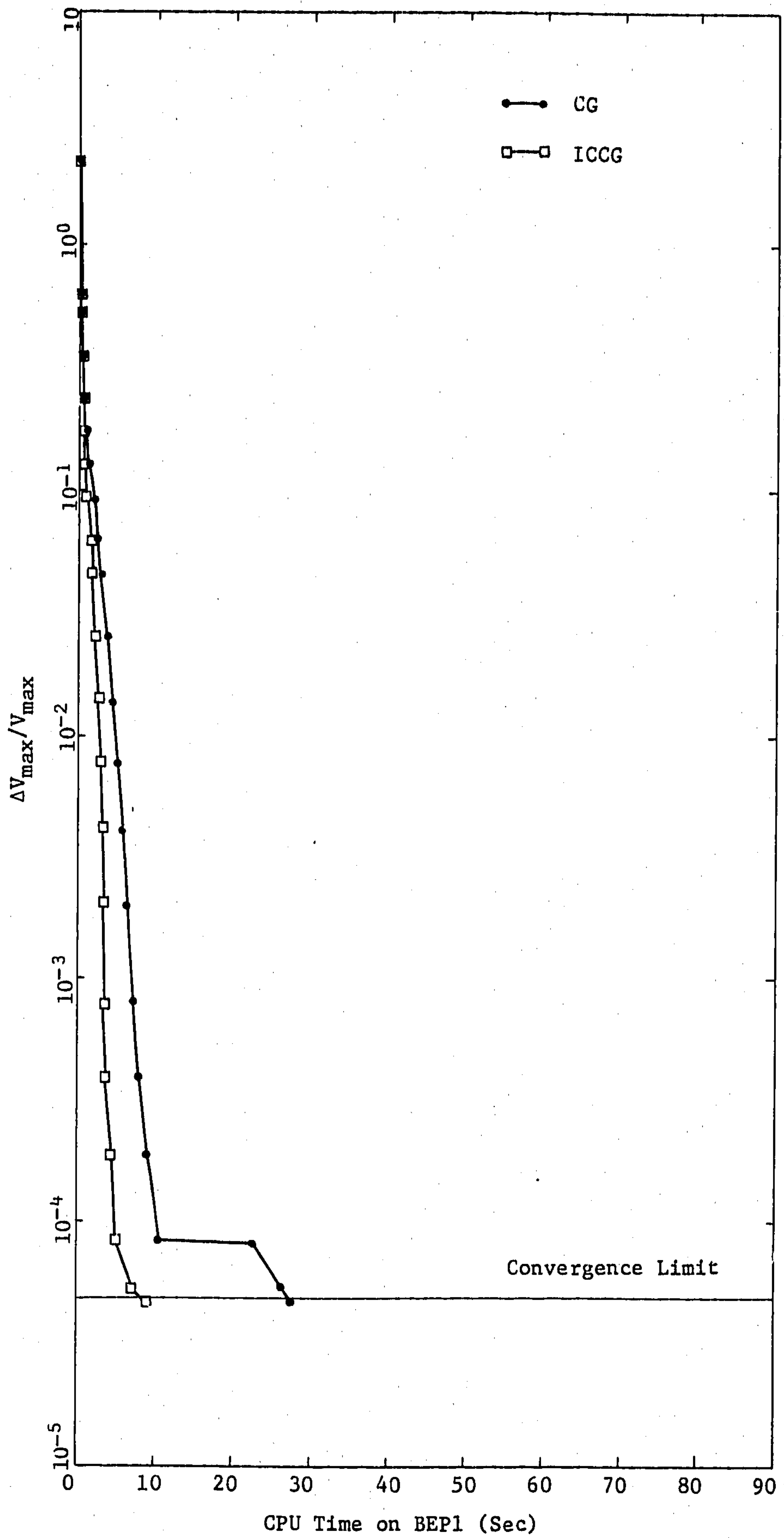


Fig. 3.21 Comparison of Convergency of Each Scheme about 120 Cells Problem on Vector Machine

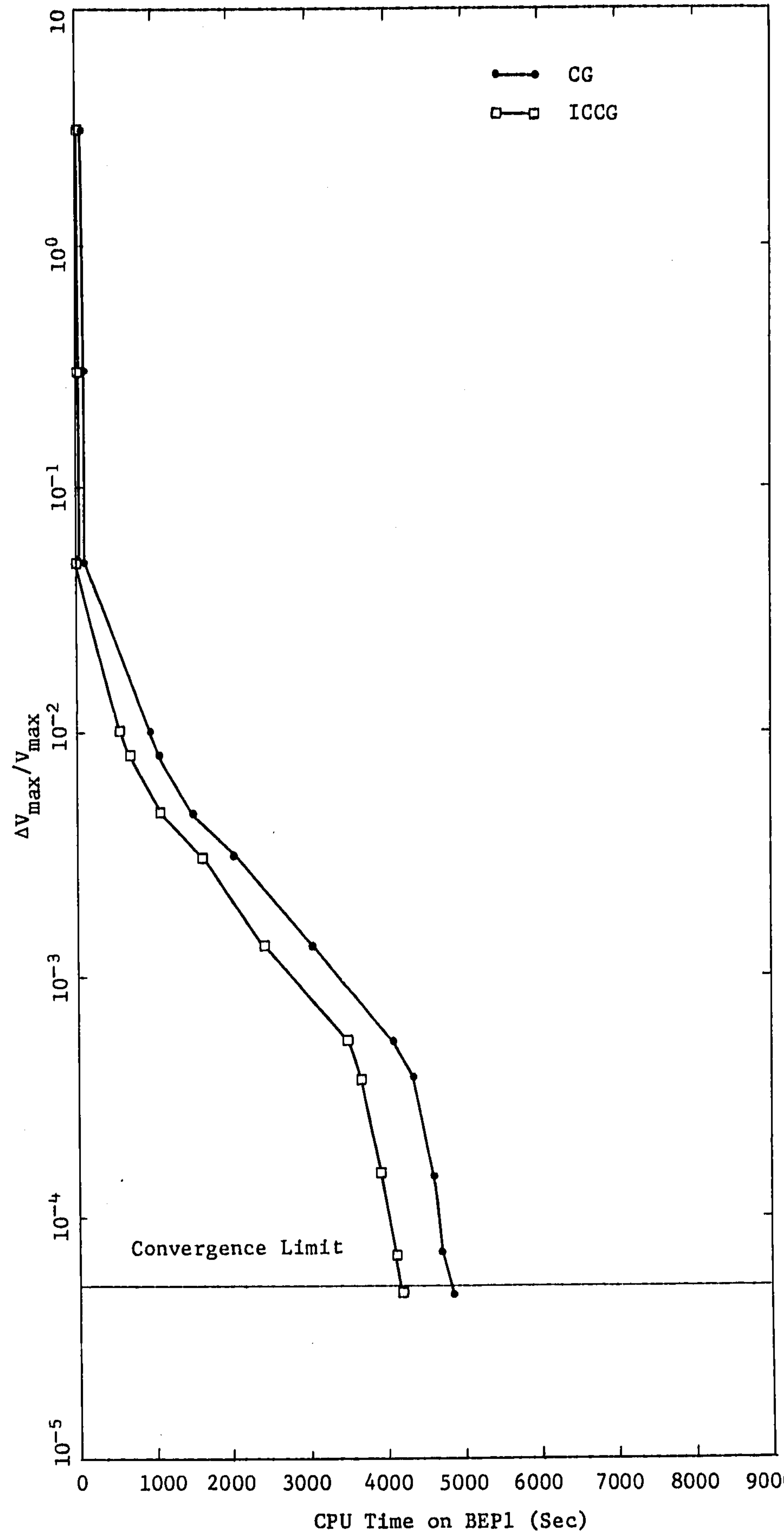


Fig. 3.22 Comparison of Convergency of Each Scheme about 1400 Cells Problem on Vector Machine

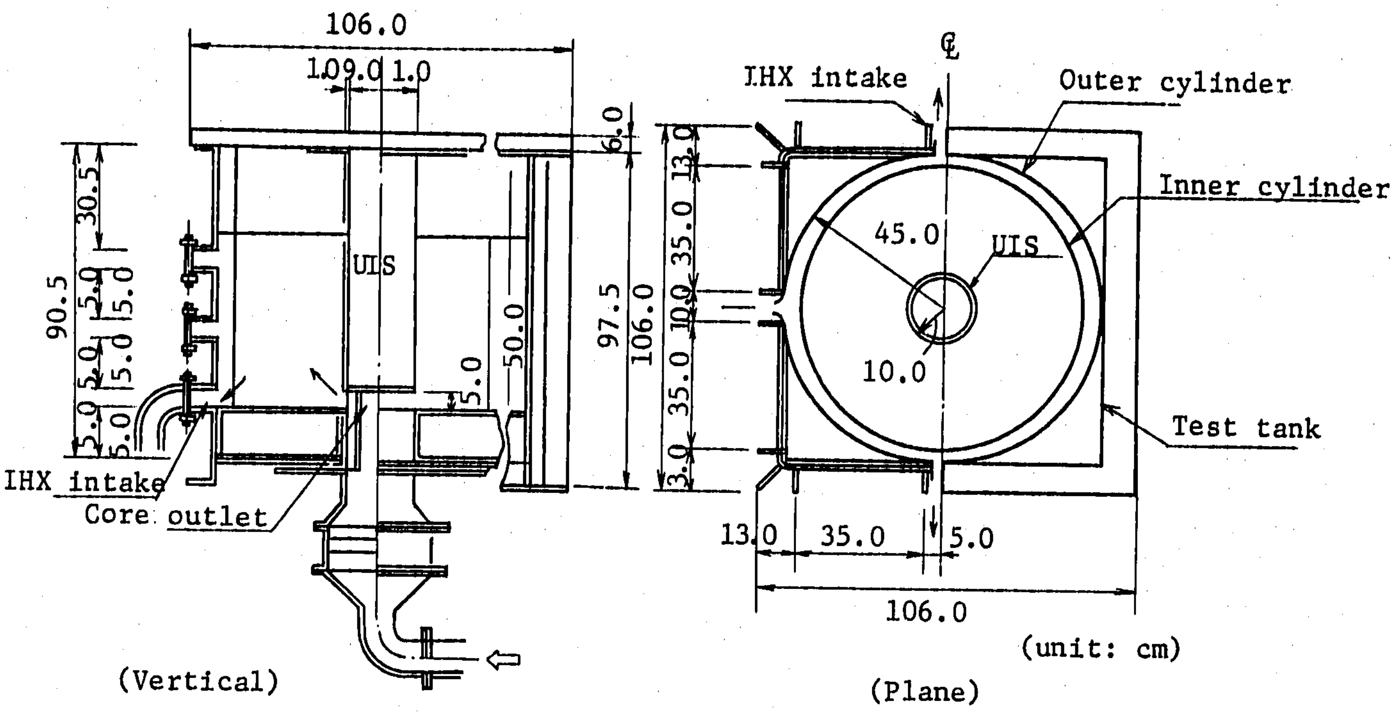


Fig. 3.23 Schematic Figure of Water Test Section

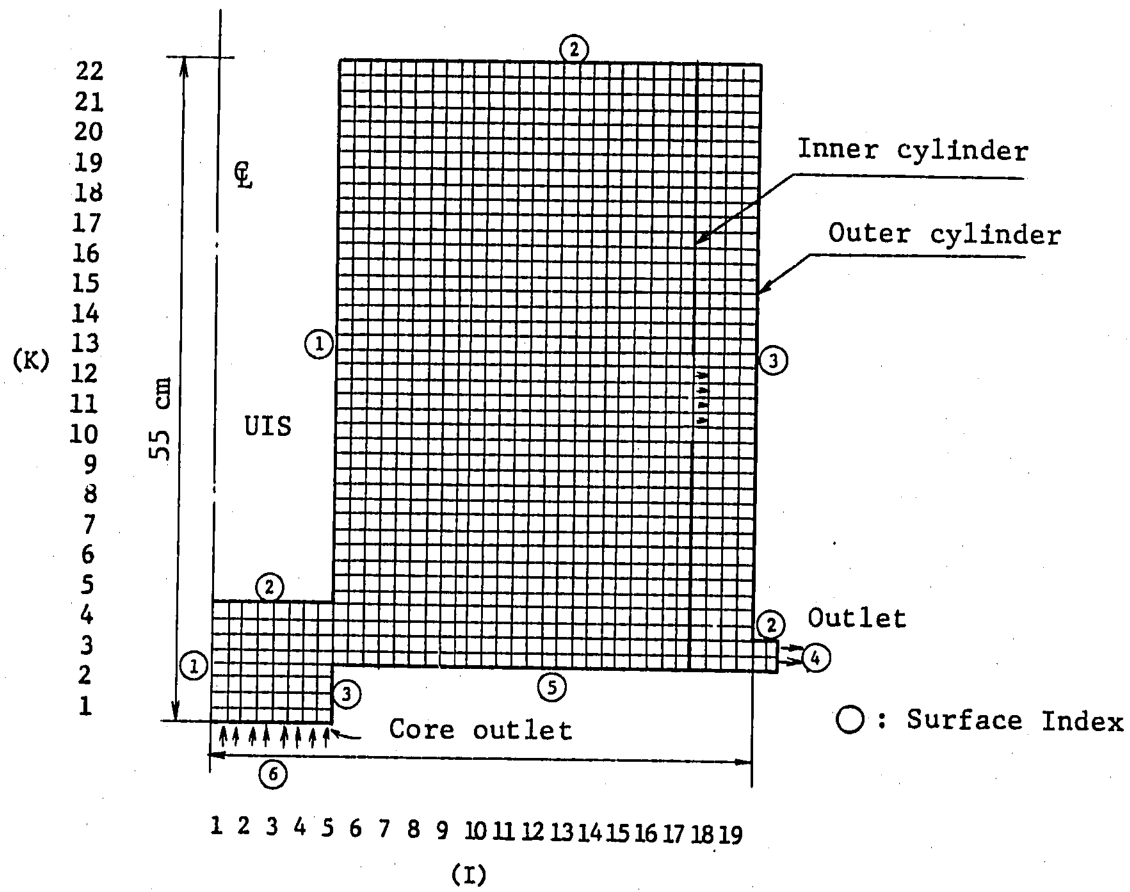


Fig. 3.24 Computational Cell Partitioning of the Test Plenum

```
***********************************************************************
*                                                                     *
* THIS DATA IS FOR RUN NO11  OF CRIEPI IN PRIMARY VESSEL WATER THERMAL *
*       HYDRAULIC EXPERIMENTS   THIS RUN IS 2 D TEST VESSEL STEADY     *
*                                                                     *
* << PRECONDITIONED CONJUGATE GRADIENT SCHEME FOR MOMENTUM & ENERGY >> *
*                                                                     *
*       ##### CR11  DETAIL POROSITIES AND PERMIABILITIES #####        *
*                                                                     *
*                                      CREATE   :   MAR.01,'85        *
*                                      MODIFY   :   MAR.01,'85        *
*                                      R U N    :   MAR.04.'85        *
*                                      D.S.N.   :   COMMIX1A.PCG2.LOAD *
*                                      M.N.     :   PCGMOEM4          *
*                                                                     *
***********************************************************************
 &GEOM   IGEOM=-1,NL1=84,  NM1=297,ISYMCH=3,IFITEN=3,  IFREB=594,
         IFRES=1,  IMAX=19,  JMAX=1,   KMAX=22,  NSURF=6,
         DX=19*0.025          ,DY=6.283210,DZ=22*0.025
        ,XNORML= 1.   ,0.   ,-1.   ,-1.  ,0.  ,0.
        ,YNORML= 0.   ,0.   ,0.    ,0.   ,0.  ,0.
        ,ZNORML= 0.   ,-1.  ,0.    ,0.   ,1.  ,1.,
        ITURKE= 0,IFPCG=-5,IFPCG2=0,LMPRNT=0,
 &END
REG -1.           1   1   1   1   1   4   1
REG -1.           5   5   1   1   5  22   1
REG -1.           1   4   1   1   4   4   2
REG -1.           5  18   1   1  22  22   2
REG -1.          19  19   1   1   3   3   2
REG -1.           4   4   1   1   1   2   3
REG -1.          18  18   1   1   4  22   3
REG -1.          19  19   1   1   3   3   4
REG -1.           5  19   1   1   3   3   5
REG -1.           1   2   1   1   1   1   6
REG 81.93    -4   3   3   1   1   1   1   6
REG 68.01    -4   4   4   1   1   1   1   6
END
 &DATA  IFENER=0 ,NTHCON=0,NTMAX=  2,IDTIME=0,DT(1)=1.0,IT=10,
        KFLOW=    -3,  -3,  -3,  -5,  -3, 1,
        KTEMP=   400, 400, 400, 400, 400, 1,
        TEMPO=     62.,GRAVZ= -9.807 ,TEMP(6)=62.,VELOC(6)=0.7003,
        NREBRT=3, ITKBUG=0,
        NREBM =   16, 240,  40,
        NREBX =    2,   2,   1,
        IREBIT=   5,
        ITMCG=300,ITMAXP=99,
        DCONV2=1.0E-5,DCONV3=1.0E-5,
        CINK1=6.62E-5,CINK2=6.62E-5,KEITER=1,NTPLOT=-10,
        CINE1=5617.69,CINE2=5617.69,
        NTPRNT=1,2,-9999,
        NTHPR =01201, 03201, 04201, 05201, 09201,
    &END
REBM   1   1   4   1   1   1   4
REBX   1   4   4   1   1   3   4
REBM   2   5  16   1   1   3  22
REBX   2  16  16   1   1  11  12
REBM   3  17  18   1   1   3  22
REBX   3  18  18   1   1   3   3
END
END
AL'  0.8345      3   3   1   1   1   4
ALZ  0.8345      3   3   1   1   1   3
ALX  0.8621      3   3   1   1   1   4
AL   0.4948      4   4   1   1   1   2
ALZ  0.4948      4   4   1   1   1   2
AL   0.9291      4   4   1   1   3   4
ALZ  0.9291      4   4   1   1   3   3
ALX  0.7931      2   2   1   1   1   4
ALX  0.         16  16   1   1   3  10
ALX  0.         16  16   1   1  13  22
END
```

Figure 3.25   Input Data for Steady-State Run of Water Test

```
THIS DATA IS FOR RUN NO11 OF CRIEPI IN PRIMARY VESSEL WATER THERMAL
HYDRAULIC EXPERIMENTS   THIS RUN IS 2 D TEST VESSEL TRANSIENT
##### CR11 DETAIL POROSITIES AND PERMIABILITIES #####
 &GEOM   IFRES=3,IFPCG=-5,IFPCG2= 1,
 &END
 &DATA  ISTATE=2,      NTMAX=1000,DT=.1,
        IFENER=1,IT=99,    TIMAX=3000.,
        KFLOW(6)=101,KTEMP(6)=102,
        TVAL=0.,2.,2.,3.,3.,10.,10.,300.,
             0.,1.,1.,2.,2.,3.,3.,6.,6.,30.,30.,300.,
        FVAL=1.,.639,.639,.569,.569,.24,.24,.24,
             1.,.40645,.40645,.2258,.2258,.1774,.1774,
             .16129,.16129,.15484,.15484,.15484,
        NEND=8,12,
        NTPLOT=-10,
        NTPRNT=1,50,100,160,200,240,270,
               300,600,1200,1800,2400,3000,
 &END
END
END
```

Figure 3.26   Input Data for Transient Run of Water Test

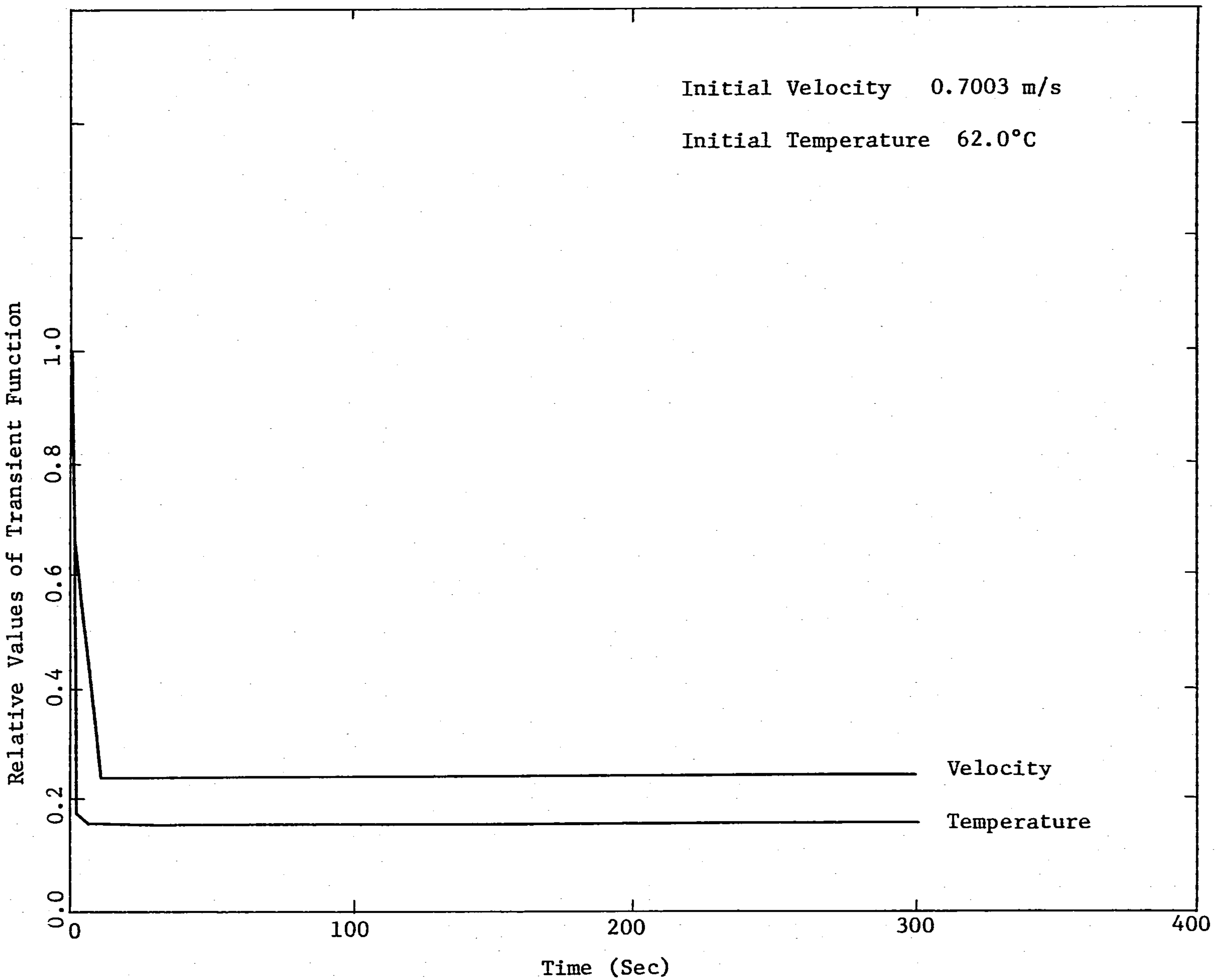


Fig. 3.27 Velocity and Temperature Transient at Inlet of the Test

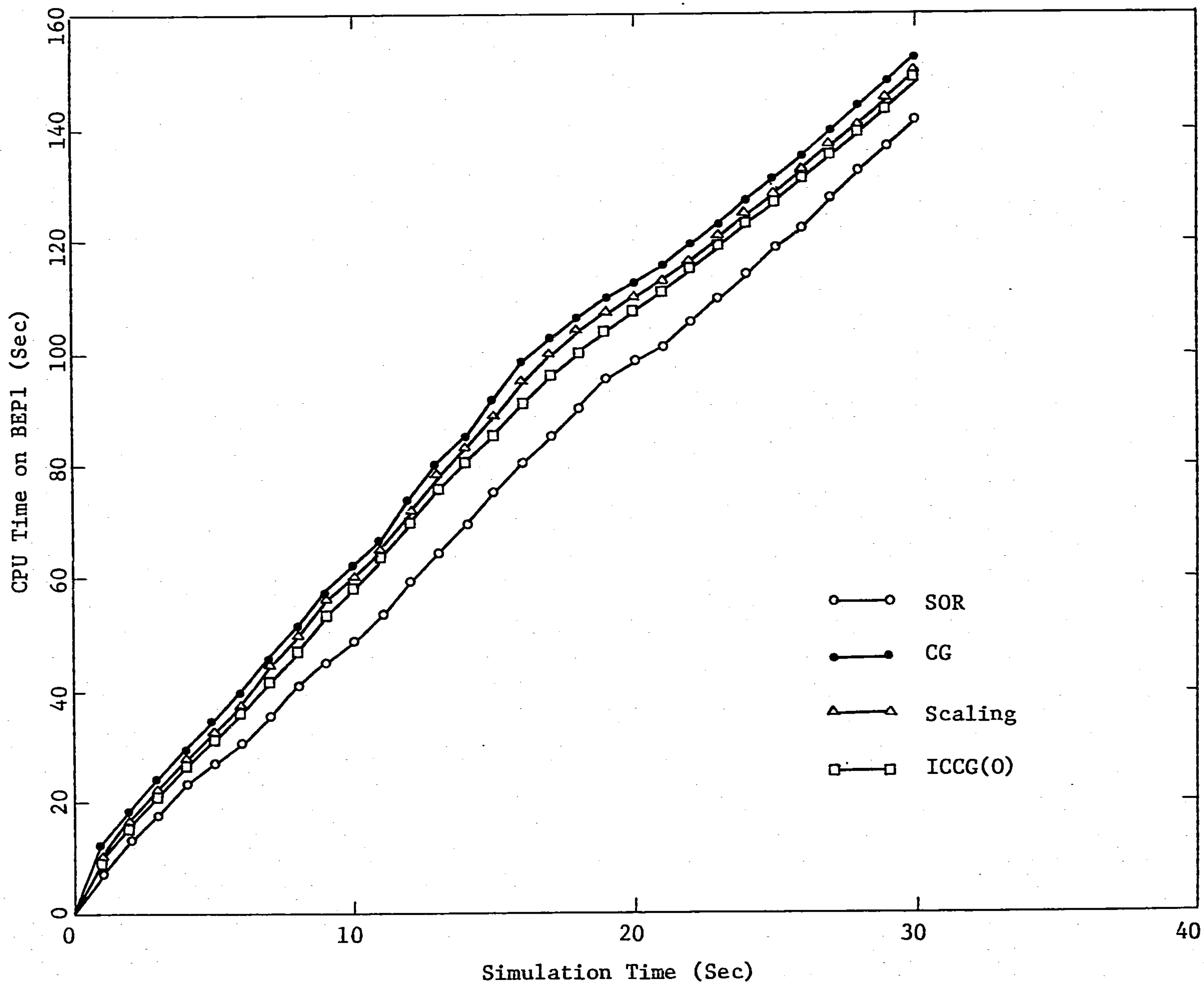


Fig. 3.28 Comparison of Processing Time of Each Scheme on Scalar Machine

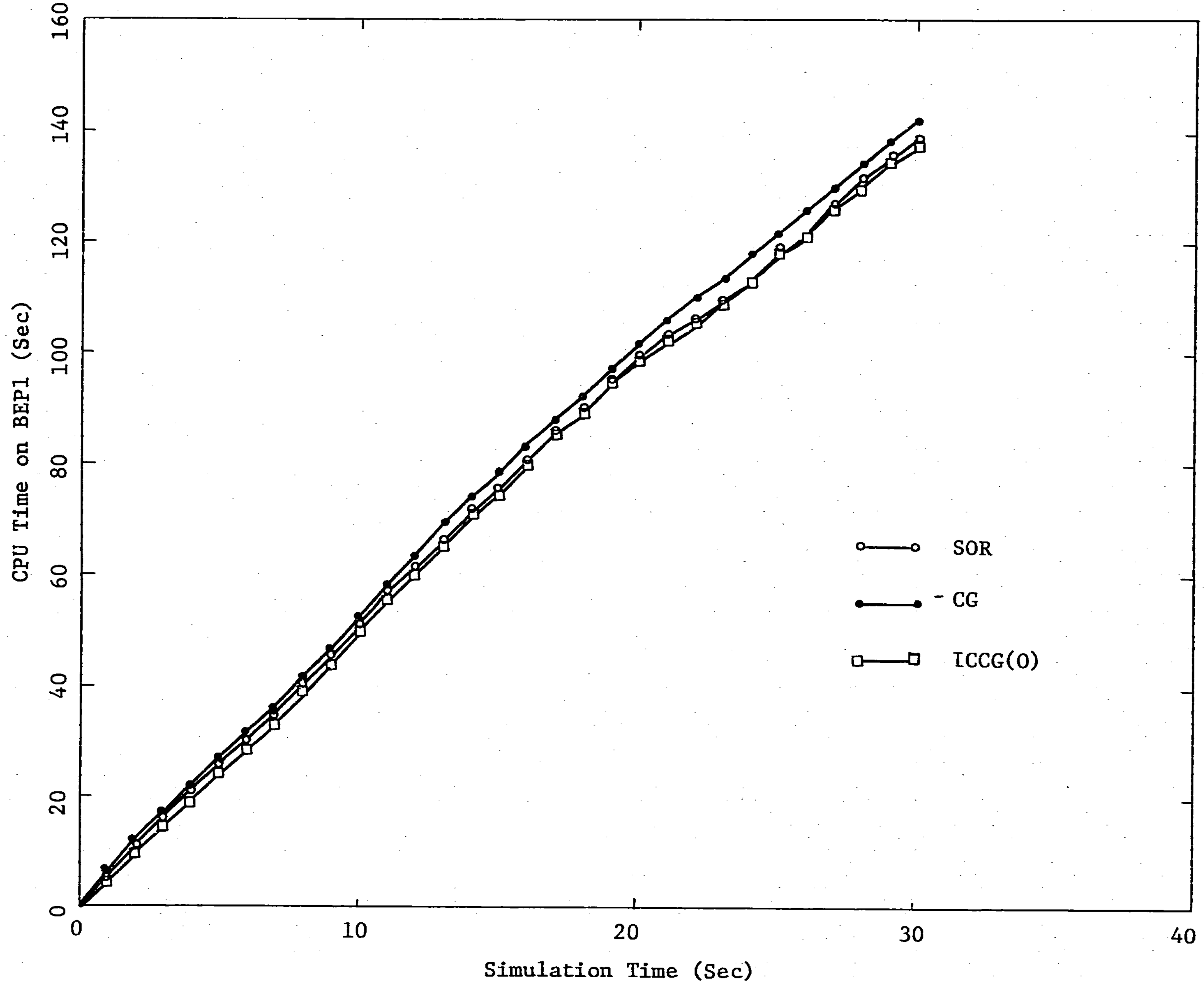


Fig. 3.29 Comparison of Processing Time of Each Scheme on Vector Machine

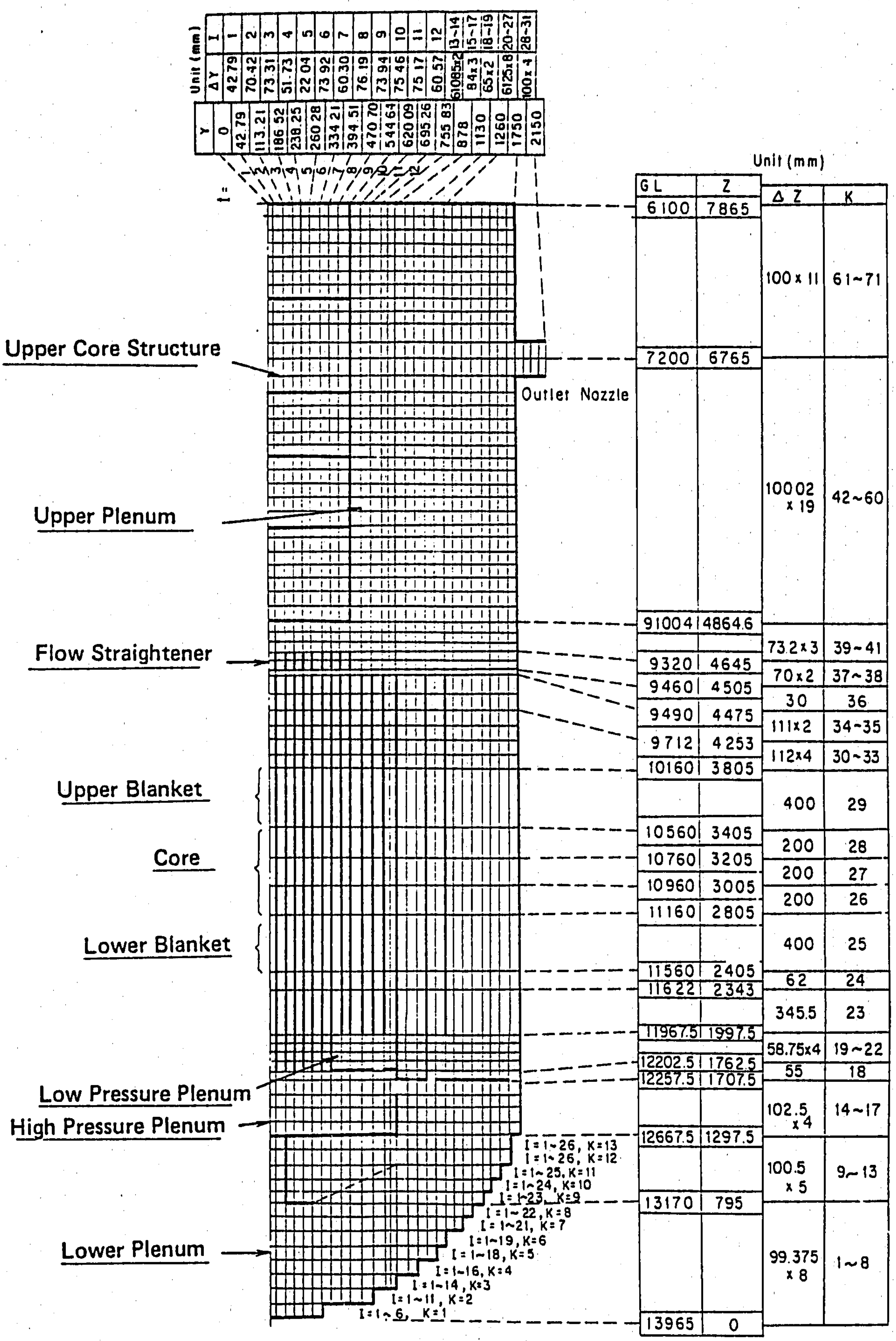


Figure 3.30 Cell Partitioning for Two-Dimensional Analyses of JOYO MK-I Natural Circulation Test

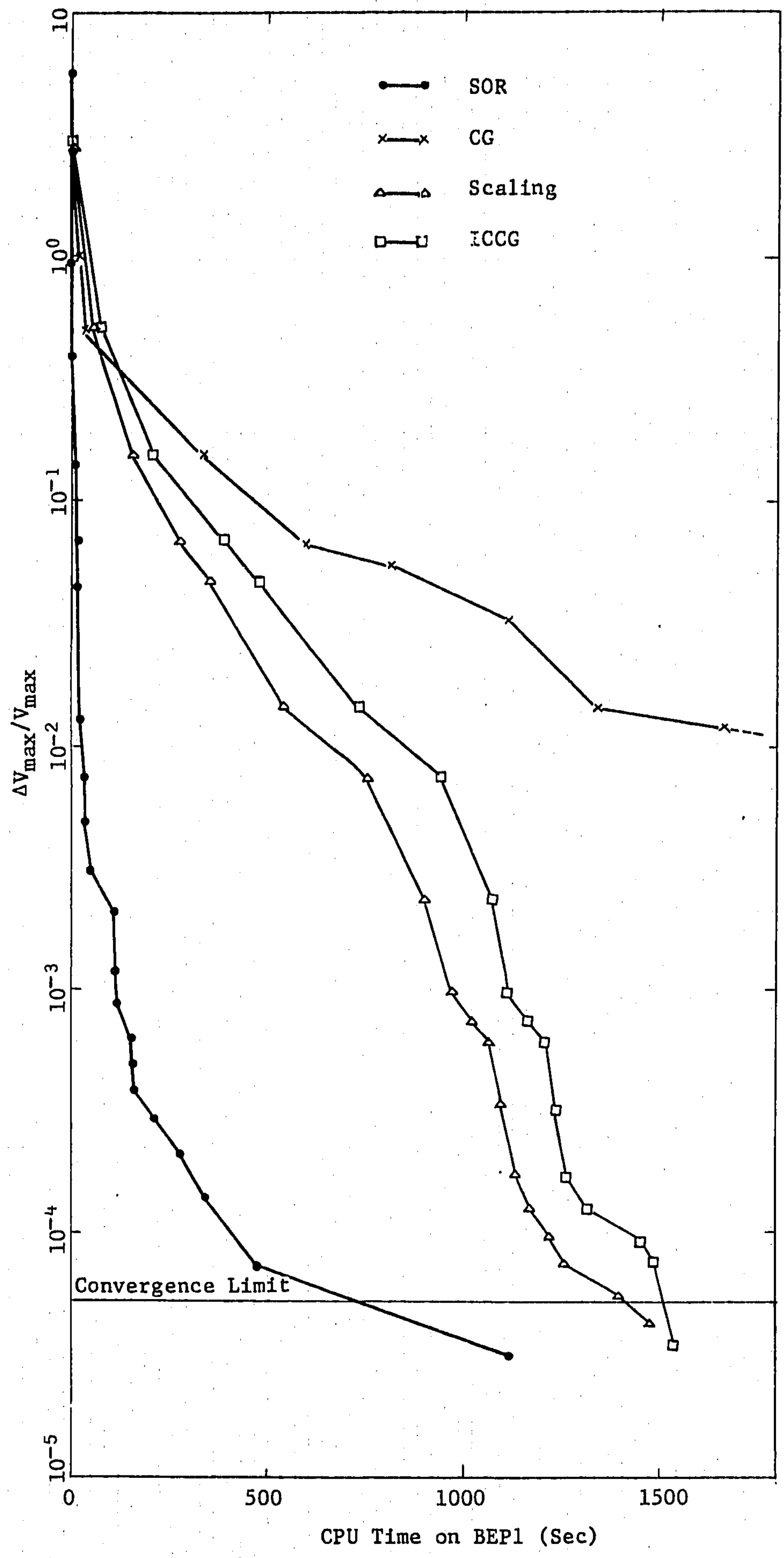


Fig. 3.31 Comparison of Convergency of Each Scheme about

```
C/
C/
C/
C/       P. C. G. ITERATION SCHEME
C/
C/                                       NOV.08,'84      T.MURAMATSU.
C/
C/ ---------------------------------------------------------------------
C/
C/          P.C.G. OPTION FLAG LIST (IFPCG IN NAMELIST /GEOM/)
C/
C/                     1.  S.O.R. SCHEME                 :   0
C/                     2.  C.G. SCHEME                   :   1
C/                     3.  DEBUGGING OPTION              :   2
C/                     4.  EIGENVALUE CALCULATION        :  -1
C/                     5.  RESTART OPTION                :  -2
C/                     6.  SCALING OPTION                :  -3
C/                     7.  L-D-U DECOMPOSITION OPTION    :  -4
C/                     8.  I.C.C.G. (0) SCHEME           :  -5
C/                     9.  M.I.C.C.G. (0) SCHEME         :  -6
C/                    10.  I.C.C.G. (1.1) SCHEME         :  -7
C/                    11.  I.C.C.G. (1.2) SCHEME         :  -8
C/                    12.  I.C.C.G. (1.3) SCHEME         :  -9
C/                    13.  I.C.C.G. (2.1) SCHEME         : -10
C/
C/ ---------------------------------------------------------------------
C/
C/          INPUT PARAMETER LIST IN NAMELIST /DATA/
C/
C/              ITHCG  :  MAXIMUM ITERATION NUMBER  (NN1)
C/              DCONV2 :  CONVERGENCE LIMIT  (1.0E-06)
C/              IMAT2  :  RESTART CYCLE (N)
C/
C/ ---------------------------------------------------------------------
C/
C/          ARRAYS LIST
C/
C/              AA    :   COEFFICIENT MATRIX
C/                          (N,14)
C/                           | |__ INDEX
C/                           |____ CONTENTS
C/              BB    :   STORAGE AREA FOR L
C/                          (N,14)
C/              AAT   :   STORAGE AREA FOR L(T)
C/                          (N,14)
C/              BBT   : WORK AREA
C/                          (N,14)
C/              XX    :   UNKNOWN VALUES
C/                          (N)
C/              S1    :   WORK AREA
C/                          (N)
C/              S2    :   AA * PP
C/                          (N)
C/              RR    :   RESIDUAL
C/                          (N)
C/              PP    :   DIRECTION FOR CORRECTION
C/                          (N)
C/              ALPH  :   PIVOT
C/                          (N)
C/              BETA  :   NOT USE
C/                          (N)
C/              BTO   :   CONSTANT VECTIOR
C/                          (N)
C/              P     :   UNKNOWN VALUE FOR MAIN PROGRAM
C/                          (N)
C/
C/ ---------------------------------------------------------------------
C/
C/          =====> T1077.COMMIX1A.CSET.FORT(PCGPACK)
C/
C/ ---------------------------------------------------------------------
C/
      SUBROUTINE PCGPACK(IFPCG,ITHCG,DCONV2,IMAT2,
     1                   AA,BB,AAT,BBT,P,BTO,N)
C
      DIMENSION AA(N,14),BB(N,14),AAT(N,14),BBT(N),XX(N),S1(N),S2(N),
     1          RR(N),PP(N),ALPH(N),BETA(N),BTO(N),P(N)
C
```

Figure D.1 Source List of P.C.G. Solver (1/9)

```
      IF(IFPCG.NE.0) WRITE(6,898) IFPCG
      IF(IFPCG.EQ.0) WRITE(6,897) IFPCG
  898 FORMAT(1H ,' .. SOLVER ==> PCG METHOD ( IFPCG= ',I5,' ) ..')
  897 FORMAT(1H ,' .. SOLVER ==> SOR METHOD ( IFPCG= ',I5,' ) ..')
      IF(IFPCG.EQ.0) GO TO 99
C
C-------------------------------- EIGENVALUE CALCULATION OPTION START --
      IF(IFPCG.NE.-1) GO TO 3074
C
      DO 9135 I=1,N
       DO 9135 J=1,14
 9135   BB(I,J)=AA(I,J)
C
      DO 9136 KAISU=1,50
       ELMAX=0.0
       DO 9137 I=1,N
        DO 9137 J=1,7
         IBUF=BB(I,J+7)
         IF(IBUF.LE.I) GO TO 9137
         IF(BB(I,J).LE.ELMAX) GO TO 9137
         ELMAX=BB(I,J)
         MAXI=I
         MAXJ=J
 9137  CONTINUE
C
       IF(ABS(BB(I,J)).LT.0.00001) GO TO 9138
C
       DO 9139 K=1,7
        IBUF=BB(I,K+7)
        IF(IBUF.EQ.I) GO TO 9140
 9139  CONTINUE
       BPP=0.0
       GO TO 9141
 9140  BPP=BB(I,K)
 9141  CONTINUE
       DO 9142 K=1,7
        IBUF=BB(J,K+7)
        IF(IBUF.EQ.J) GO TO 9143
 9142  CONTINUE
       BQQ=0.0
       GO TO 9144
 9143  BQQ=BB(J,K)
 9144  CONTINUE
C
       RESI=2.0*BB(I,J)/(BPP-BQQ)
       TEST=0.5*ATAN(RESI)
C
       SEST=SIN(TEST)
       CEST=COS(TEST)
C
       DO 9145 J2=1,7
        BPJ=BB(I,J2)
        BQJ=BB(J,J2)
        BB(I,J2)=BPJ*CEST+BQJ*SEST
        BB(J,J2)=-BPJ*SEST+BQJ*CEST
 9145  CONTINUE
C
       DO 9146 I2=1,N
        DO 9150 J2=1,7
         IBUF=BB(I2,J2+7)
         IF(IBUF.EQ.I) GO TO 9151
         IF(IBUF.EQ.J) GO TO 9151
 9150   CONTINUE
        GO TO 9146
 9151   CONTINUE
        BIP=BB(I2,I)
        BIQ=BB(I2,J)
        BB(I2,I)=BIP*CEST+BIQ*SEST
        BB(I2,J)=-BIP*CEST+BIQ*CEST
 9146  CONTINUE
C
 9136 CONTINUE
C
 9138 EIMAX=BB(1,1)
      EIMIN=BB(1,1)
      DO 9147 I3=1,N
       DO 9147 J3=1,7
        IBUF=BB(I3,J3+7)
```

Figure D.1 (Continued) (2/9)

```
          IF(IBUF.NE.I3) GO TO 9147
          WRITE(6,9149) BB(I3,J3)
          IF(BB(I3,J3).GT.EIMAX) EIMAX=BB(I3,J3)
          IF(BB(I3,J3).LT.EIMIN) EIMIN=BB(I3,J3)
 9147 CONTINUE
C
      CONDI=EIMAX/EIMIN
C
      WRITE(6,9148) CONDI
 9148 FORMAT('     CONDITION NUMBER OF COEFFICIENT MATRIX=',E16.6,/)
C
 9149 FORMAT('  EIGENVALUE=',E16.6,/)
C
 3074 CONTINUE
C-------------------------------- EIGENVALUE CALCULATION OPTION END ---
C
C-------------------------------- PIVOT SCALING OPTION START ----------
      IF(IFPCG.NE.-2.AND.IFPCG.NE.-3.AND.
     1   IFPCG.NE.-24.AND.IFPCG.NE.-34) GO TO 3060
C
C
C       PIVOT SCALING TO COEFFICIENT MATRIX & CONSTANT VECTOR
C
C                    AA : COEFFICIENT MATRIX
C                   BTO : CONSTANT VECTOR
C                  ALPH : STORE AREA FOR PIVOT
C
C             IFPCG = -3 : COMPLETE PIVOTING OPERATION
C
      BUF=0.0
      DO 3050 K=1,N
      DO 3052 I=1,7
       IF(AA(K,I).LE.0.0) GO TO 3052
       BUF=SQRT(ABS(AA(K,I)))
       GO TO 3053
 3052 CONTINUE
 3053 CONTINUE
C
      IF(BUF.EQ.0.0) WRITE(6,3081) K
 3081 FORMAT(' ***** PIVOT SELECTION ERROR ON ROW =',I6,' *****')
      IF(BUF.EQ.0.0) WRITE(6,3082) (AA(K,I),I=1,7)
 3082 FORMAT(1H ,5X,7E16.6)
C
      DO 3051 L=1,7
       AA(K,L)=AA(K,L)/BUF
       INDS=AA(K,L+7)
       DO 3076 J=1,7
        INDU=AA(INDS,J+7)
        IF(INDU.NE.K) GO TO 3076
        INDT=J
        GO TO 3077
 3076  CONTINUE
       GO TO 3051
 3077 CONTINUE
      AA(INDS,INDT)=AA(INDS,INDT)/BUF
 3051 CONTINUE
      BTO(K)=BTO(K)/BUF
 3050 ALPH(K)=BUF
 3060 CONTINUE
C-------------------------------- PIVOT SCALING OPTION END ----------
C
C-------------------------------- I.C.C.G. OPTION START -------------
      IF(IFPCG.NE.-5) GO TO 7075
C
      DO 7398 I=1,N
      DO 7398 J=1,14
       AAT(I,J)=0.0
 7398  BB(I,J)=AA(I,J)
C
      BB(1,1)=SQRT(BB(1,1))
C
      DO 7300 J=2,7
       IF(BB(1,J).EQ.0.0) GO TO 7300
       BB(1,J)=BB(1,J)/BB(1,1)
 7300 CONTINUE
C
      DO 7301 I=2,N
       DO 7307 I3=1,7
```

Figure D.1 (Continued) (3/9)

```
         IP=BB(I,I3+7)
         IF(IP.NE.I) GO TO 7307
         S=BB(I,I3)
         IS2=BB(I,I3+7)
C         WRITE(6,9127) I,S,IS2
 9127     FORMAT(' I=',I6,' S=',E16.6,' IS2=',I6)
         IF(IS2.NE.I) WRITE(6,9010) I,IS2
 9010    FORMAT(' --- PIVOT SELECTION ERROR  I=',I5,' IS2=',I5,' ---')
         IF(S.EQ.0.0) GO TO 7301
         GO TO 7308
 7307  CONTINUE
       IS2=9999
       WRITE(6,9010) I,IS2
 7308  CONTINUE
C
       II=I-1
       DO 7302 K=1,II
        DO 7309 K2=8,14
         IS3=BB(K,K2)
         IF(IS3.EQ.IS2) GO TO 7310
 7309   CONTINUE
C       WRITE(6,9015) I,K,K2
 9015   FORMAT(' ***** I=',I5,' K=',I5,' K2=',E16.6)
        GO TO 7302
 7310   S=S-BB(K,K2-7)**2
C        WRITE(6,9128) BB(K,K2-7)
 9128    FORMAT('        BB(K,K2-7)=',E16.6)
        IF(S.LE.0.0) WRITE(6,9011) K,IS3,S,BB(K,K2-7)
 9011  FORMAT(' -- K=',I5,' IS3=',I5,' S=',E16.6,' BB(K,K2-7)=',E16.6)
 7302  CONTINUE
       IF(S.LT.0.0) S=1.0
       BB(I,I3)=SQRT(S)
C
       IF(I.EQ.N) GO TO 9009
C
       III=I+1
       DO 7303 J=8,14
        IS4=BB(I,J)
        IF(IS4.LT.III) GO TO 7303
        SS2=BB(I,J-7)
        IS4=BB(I,J)
        DO 7304 K=1,II
         DO 7311 I2=8,14
          IBB1=BB(K,I2)
          IF(IBB1.NE.I) GO TO 7311
          BB1=BB(K,I2-7)
          GO TO 7312
 7311    CONTINUE
         BB1=0.0
 7312    CONTINUE
C
         DO 7313 J2=8,14
          IBB2=BB(K,J2)
          IF(IBB2.NE.IS4) GO TO 7313
          BB2=BB(K,J2-7)
          GO TO 7314
 7313    CONTINUE
         BB2=0.0
 7314    CONTINUE
C
         SS2=SS2-BB1*BB2
 7304    CONTINUE
C
         BB(I,J-7)=SS2/BB(I,I3)
 7303    CONTINUE
 7301 CONTINUE
 9009 CONTINUE
C
C
      DO 7315 I=1,N
       DO 7316 J=1,7
        IA=BB(I,J+7)
        IF(IA.GE.I) GO TO 7316
        BB(I,J)=0.0
        BB(I,J+7)=0.0
 7316  CONTINUE
 7315 CONTINUE
C                    ----- L(T) -----
```

Figure D.1 (Continued) (4/9)

```
      DO 7317 I=1,N
       DO 7318 J=1,7
        IA1=BB(I,J+7)
        IF(IA1.EQ.0) GO TO 7318
        DO 7319 J2=1,7
         IF(AAT(IA1,J2).EQ.0.0) GO TO 7320
 7319   CONTINUE
        GO TO 7318
 7320   CONTINUE
        AAT(IA1,J2)=BB(I,J)
        AAT(IA1,J2+7)=I
 7318  CONTINUE
 7317 CONTINUE
C                      ----- L -----
C
 8001 FORMAT(' ----- DECOMPOSED MATRIX (L-T) -----')
 8003 FORMAT(' ----- DECOMPOSED MATRIX (L) -----')
 7388 FORMAT(' ***** ORIGINAL COMPRESSED MATRIX *****')
 7384 FORMAT(' I=',I8)
 7385 FORMAT(1H ,5X,7E15.6)
 7075 CONTINUE
C
C------------------------------- I.C.C.G. OPTION END ---------------
C
C     ************************************************
C   *****                                            *****
C *****             CONJUGATE GRADIENT ROUTINE START      *****
C   *****                                            *****
C     ************************************************
C
      ITERCG=0
      IXX1=0
      DO 4000 IXX=1,N
       XX(IXX)=P(IXX)
       IF(IFPCG.EQ.-3) XX(IXX)=P(IXX)*ALPH(IXX)
 4000 CONTINUE
      DO 4001 I=1,N
       W=BTO(I)
      DO 4002 J=1,7
       ICEL=AA(I,J+7)
       IF(ICEL.EQ.0) GO TO 4001
       IF(AA(I,J).EQ.0.0.OR.XX(ICEL).EQ.0.0) GO TO 4002
       W=W-AA(I,J)*XX(ICEL)
 4002 CONTINUE
 4001  RR(I)=W
      DO 4003 I=1,N
 4003  PP(I)=RR(I)
C
      IF(IFPCG.NE.-5) GO TO 7023
                BBT(1)=RR(1)/AAT(1,1)
                DO 5001 I=2,N
                 DO 5002 J=1,7
                  J2=AAT(I,J+7)
                  IF(J2.EQ.I) GO TO 5003
 5002            CONTINUE
                 GO TO 5004
 5003            PIVOT=AAT(I,J)
 5004            IF(PIVOT.EQ.0.0) PIVOT=1.0
 9006            FORMAT(1H ,5X,E16.6)
                 W=RR(I)
                  DO 5006 K2=1,7
                   K3=AAT(I,K2+7)
                   IF(K3.GE.I) GO TO 5007
                   IF(AAT(I,K2).EQ.0.0.OR.BBT(K3).EQ.0.0) GO TO 5006
                   W=W-AAT(I,K2)*BBT(K3)
 5006              CONTINUE
 5007              W=W/PIVOT
                 BBT(I)=W
 5001           CONTINUE
C
C                          ----- L V = R -----
C
                DO 9190 JE=1,7
                 IBUF=BB(N,JE+7)
                 IF(IBUF.EQ.N) GO TO 9191
 9190           CONTINUE
                GO TO 9197
 9191           PP(N)=BBT(N)/BB(N,JE)
```

Figure D.1 (Continued) (5/9)

```
                DO 9192 IE=N-1,1,-1
                 DO 9193 JB=1,7
                  IBUF=BB(IE,JB+7)
                  IF(IBUF.EQ.IE) GO TO 9194
 9193            CONTINUE
                 GO TO 9195
 9194            PIVOT=BB(IE,JB)
 9195            IF(PIVOT.EQ.0.0) PIVOT=1.0
                 W=BBT(IE)
                 DO 9196 JE2=1,7
                  IBUF=BB(IE,JE2+7)
                  IF(IBUF.LE.IE) GO TO 9196
                  IF(BB(IE,JE2).EQ.0.0.OR.PP(IBUF).EQ.0.0) GO TO 9196
                  W=W-BB(IE,JE2)*PP(IBUF)
 9196            CONTINUE
                 W=W/PIVOT
                 PP(IE)=W
 9192            CONTINUE
 9197            CONTINUE
C
C                                ----- L(T) U = V -----
 7023 CONTINUE
C
C    ****************************************
C  *   CG ( CONJUGATE GRADIENT ) LOOP START   *
C    ****************************************
C
      DLMAX=0.0
 4100 CONTINUE
      ITERCG=ITERCG+1
      DO 4004 I=1,N
       W=0.0
      DO 4005 J=1,7
       ICEL=AA(I,J+7)
       IF(ICEL.EQ.0) GO TO 4004
       IF(AA(I,J).EQ.0.0.OR.PP(ICEL).EQ.0.0) GO TO 4005
       W=W+AA(I,J)*PP(ICEL)
 4005 CONTINUE
 4004 S2(I)=W
C
      IF(IFPCG.NE.-5) GO TO 7047
                BBT(1)=RR(1)/AAT(1,1)
                DO 5015 I=2,N
                 DO 5016 J=1,7
                  J2=AAT(I,J+7)
                  IF(J2.EQ.I) GO TO 5017
 5016            CONTINUE
                 GO TO 5018
 5017            PIVOT=AAT(I,J)
 5018            IF(PIVOT.EQ.0.0) PIVOT=1.0
                 W=RR(I)
                  DO 5020 K2=1,7
                   K3=AAT(I,K2+7)
                   IF(K3.GE.I) GO TO 5021
                   IF(AAT(I,K2).EQ.0.0.OR.BBT(K3).EQ.0.0) GO TO 5020
                   W=W-AAT(I,K2)*BBT(K3)
 5020              CONTINUE
 5021              W=W/PIVOT
                 BBT(I)=W
 5015           CONTINUE
C
C                                ----- L V = R -----
C
                DO 9198 JE=1,7
                 IBUF=BB(N,JE+7)
                 IF(IBUF.EQ.N) GO TO 9199
 9198           CONTINUE
                GO TO 9205
 9199           S1(N)=BBT(N)/BB(N,JE)
                DO 9200 IE=N-1,1,-1
                 DO 9201 JB=1,7
                  IBUF=BB(IE,JB+7)
                  IF(IBUF.EQ.IE) GO TO 9202
 9201            CONTINUE
                 GO TO 9203
 9202            PIVOT=BB(IE,JB)
 9203            IF(PIVOT.EQ.0.0) PIVOT=1.0
                 W=BBT(IE)
```

Figure D.1 (Continued) (6/9)

```
                  DO 9204 JE2=1,7
                   IBUF=BB(IE,JE2+7)
                   IF(IBUF.LE.IE) GO TO 9204
                   IF(BB(IE,JE2).EQ.0.0.OR.S1(IBUF).EQ.0.0) GO TO 9204
                   V=V-BB(IE,JE2)*S1(IBUF)
 9204             CONTINUE
                  V=V/PIVOT
                  S1(IE)=V
 9200             CONTINUE
 9205             CONTINUE
C
C                              ----- L(T) U = V -----
 7047 CONTINUE
C
      DENOMI=0.0
      DENOMJ=0.0
      DO 4006 I=1,N
       IF(IFPCG.NE.-5) DENOMJ=DENOMJ+RR(I)*PP(I)
C
       IF(IFPCG.EQ.-5) DENOMJ=DENOMJ+RR(I)*S1(I)
C
 4006  DENOMI=DENOMI+S2(I)*PP(I)
      ALPHA=DENOMJ/DENOMI
      DO 4007 I=1,N
 4007  XX(I)=XX(I)+ALPHA*PP(I)
C
      DENOMI=0.0
      DO 4008 I=1,N
       IF(IFPCG.NE.-5) DENOMI=DENOMI+RR(I)**2
C
 4008  IF(IFPCG.EQ.-5) DENOMI=DENOMI+RR(I)*S1(I)
C
      DO 4009 I=1,N
 4009  RR(I)=RR(I)-ALPHA*S2(I)
C
      IF(IFPCG.NE.-5) GO TO 7051
                 BBT(1)=RR(1)/AAT(1,1)
                 DO 5042 I=2,N
                  DO 5043 J=1,7
                   J2=AAT(I,J+7)
                   IF(J2.EQ.I) GO TO 5044
 5043             CONTINUE
                  GO TO 5045
 5044             PIVOT=AAT(I,J)
 5045             IF(PIVOT.EQ.0.0) PIVOT=1.0
                  V=RR(I)
                   DO 5047 K2=1,7
                    K3=AAT(I,K2+7)
                    IF(K3.GE.I) GO TO 5048
                    IF(AAT(I,K2).EQ.0.0.OR.BBT(K3).EQ.0.0) GO TO 5047
                    V=V-AAT(I,K2)*BBT(K3)
 5047               CONTINUE
 5048               V=V/PIVOT
                  BBT(I)=V
 5042            CONTINUE
C                              ----- L V = R -----
C
                 DO 9206 JE=1,7
                  IBUF=BB(N,JE+7)
                  IF(IBUF.EQ.N) GO TO 9207
 9206            CONTINUE
                 GO TO 9213
 9207            S1(N)=BBT(N)/BB(N,JE)
                 DO 9208 IE=N-1,1,-1
                  DO 9209 JB=1,7
                   IBUF=BB(IE,JB+7)
                   IF(IBUF.EQ.IE) GO TO 9210
 9209             CONTINUE
                  GO TO 9211
 9210             PIVOT=BB(IE,JB)
 9211             IF(PIVOT.EQ.0.0) PIVOT=1.0
                  V=BBT(IE)
                  DO 9212 JE2=1,7
                   IBUF=BB(IE,JE2+7)
                   IF(IBUF.LE.IE) GO TO 9212
                   IF(BB(IE,JE2).EQ.0.0.OR.S1(IBUF).EQ.0.0) GO TO 9212
                   V=V-BB(IE,JE2)*S1(IBUF)
 9212             CONTINUE
```

Figure D.1 (Continued) (7/9)

```
                 V=V/PIVOT
                 S1(IE)=V
 9208            CONTINUE
 9213            CONTINUE
C                                          ----- L(T) U = V -----
 7051 CONTINUE
C
      DENOMJ=0.0
      DO 4010 I=1,N
       IF(IFPCG.NE.-5) DENOMJ=DENOMJ+RR(I)**2
C
 4010  IF(IFPCG.EQ.-5) DENOMJ=DENOMJ+RR(I)*S1(I)
C
      BET=DENOMJ/DENOMI
C
      DO 4011 I=1,N
       IF(IFPCG.NE.-5) PP(I)=RR(I)+BET*PP(I)
C
 4011  IF(IFPCG.EQ.-5) PP(I)=S1(I)+BET*PP(I)
C
      I1=0
      DLMAX=1.E-39
      RRMAX=1.E-39
      DO 4501 III=1,N
       IF(ABS(RR(I1)).GT.RRMAX) RRMAX=ABS(RR(I1))
       GO TO 4504
 4505  CONTINUE
 4504  CONTINUE
 4501 CONTINUE
C
C
      IF(RRMAX.LT.DCONV2) GO TO 4012
      IF(ITERCG.GE.ITMCG) GO TO 4012
      IF(IFPCG.LT.1000.AND.IFPCG.NE.4.AND.IFPCG.NE.-24.
     1  AND.IFPCG.NE.-34) GO TO 4030
C------------- RESTART OPTION FOR ERROR CLEAR OF RR(I) -----------------
      IF(ITERCG.NE.NEXT) GO TO 4030
      WRITE(6,4400) ITERCG,IMAT2
 4400 FORMAT(' ***** RESTART OPTION ON : ITERCG=',I5,' INTERVAL=',I5,
     1' *****')
      NEXT=NEXT+IMAT2
      DO 4031 I=1,N
       PP(I)=RR(I)
 4031 CONTINUE
C-----------------------------------------------------------------------
 4030 CONTINUE
      IF(IFPCG.NE.3) GO TO 4027
      WRITE(6,4020) ITERCG,DCONV2,RRMAX
 4020 FORMAT(10X,' IN ITERATION LOOP : ITERCG=',I5,' DCONV2=',E15.6,
     1' RRMAX=',E15.6)
      DO 4025 II=1,N
       WRITE(6,4026) II,XX(II),RR(II),DENOMJ,BET
 4026  FORMAT(' II=',I3,' XX=',E12.5,' RR=',E12.5,
     1          ' DENOMJ=',E12.5,
     2          ' BET=',E12.5)
 4025 CONTINUE
 4027 CONTINUE
      GO TO 4100
C
C          ************************************
C          * CONJUGATE GRADIENT LOOP TERMINATING *
C          ************************************
C
 4012 CONTINUE
      WRITE(6,4021) ITMCG,ITERCG,DCONV2,RRMAX,DLMAX
 4021 FORMAT(5X,' >>> C.G. ITERATION COMPLETED : LIMIT=',I5,
     1' ITERCG=',I5,' DCONV2=',E15.6,' RRMAX=',E15.6,' DLMAX=',E16.6,
     2' <<<')
      IF(IFPCG.NE.3) GO TO 4029
      DO 4028 II=1,N
       WRITE(6,4026) II,XX(II),RR(II),DENOMJ,BET
 4028 CONTINUE
 4029 CONTINUE
      IF(IFPCG.NE.-3.AND.IFPCG.NE.-34) GO TO 3064
C
C         SCALE BACK OF SCALED MATRIX AA BY ALPH(I)
C
      DO 3063 I=1,N
```

Figure D.1 (Continued) (8/9)

```
 3063  XX(I)=XX(I)/ALPH(I)
 3064 CONTINUE
C
      I1=0
      DO 4013 MO=1,N
       P(MO)=XX(I1)
 4013 CONTINUE
      GO TO 660
C
   99 CONTINUE
C
C                   ---- S. O. R.  PROCESS ----
C                   I                         I
C                   I                         I
C                   I                         I
C                   I                         I
C                   I                         I
C                   ---------------------------
C
  660 CONTINUE
      RETURN
      END
```

Figure D.1 (Continued) (9/9)

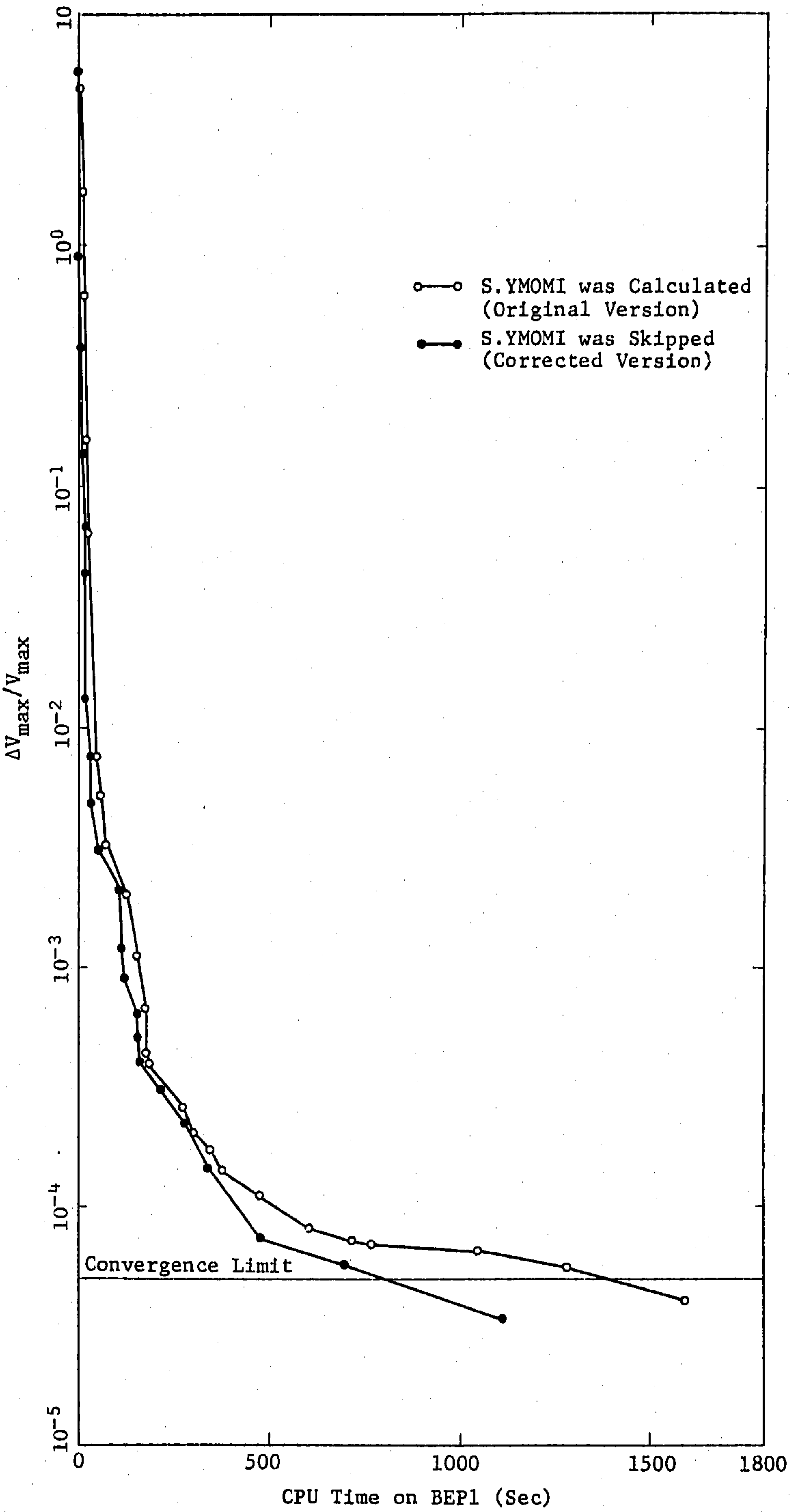


Fig. E.1 Comparison of Convergency Between Original Version

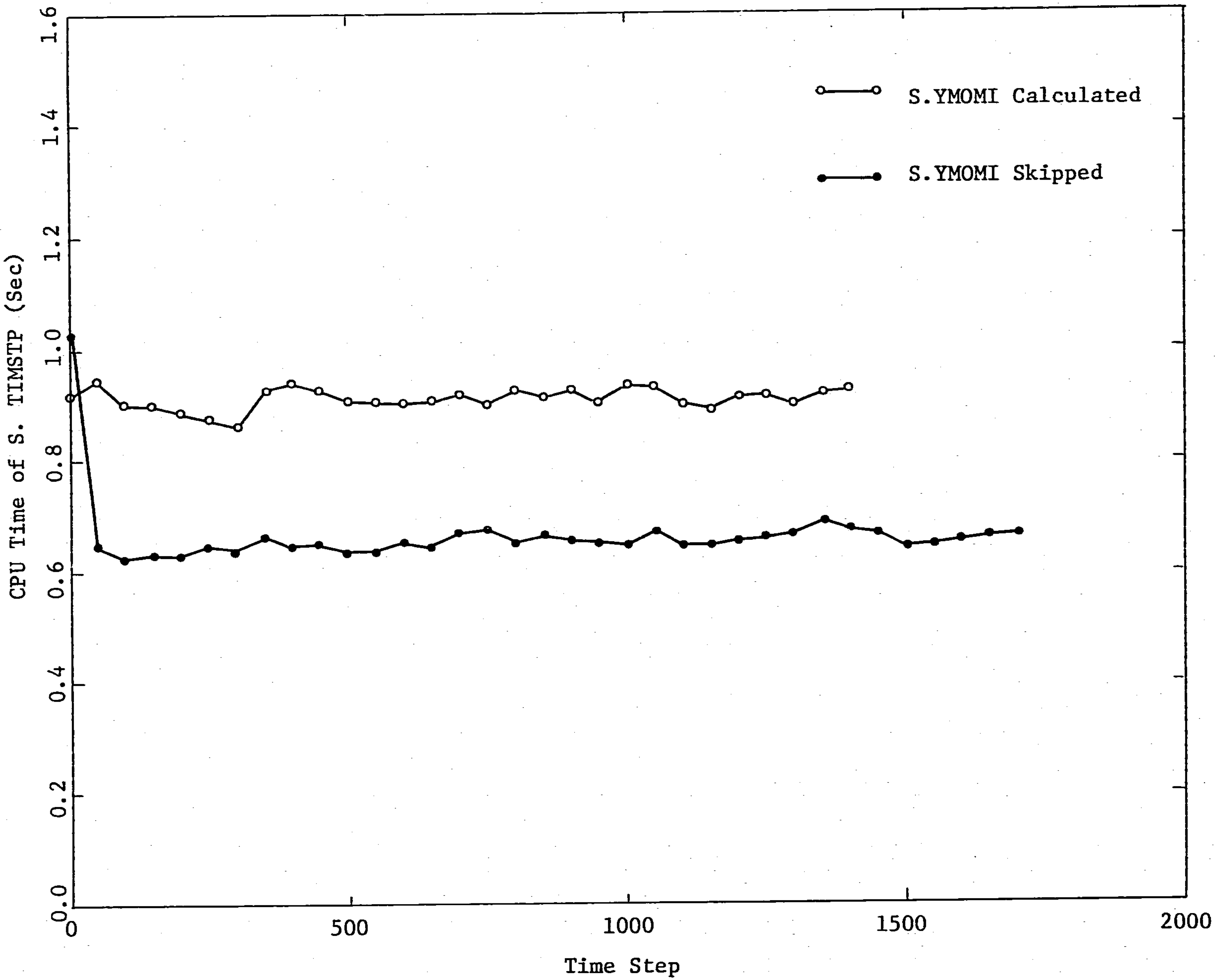


Fig. E.2 Effect of S.YMOMI Skip for CPU Time of S.TIMSTP

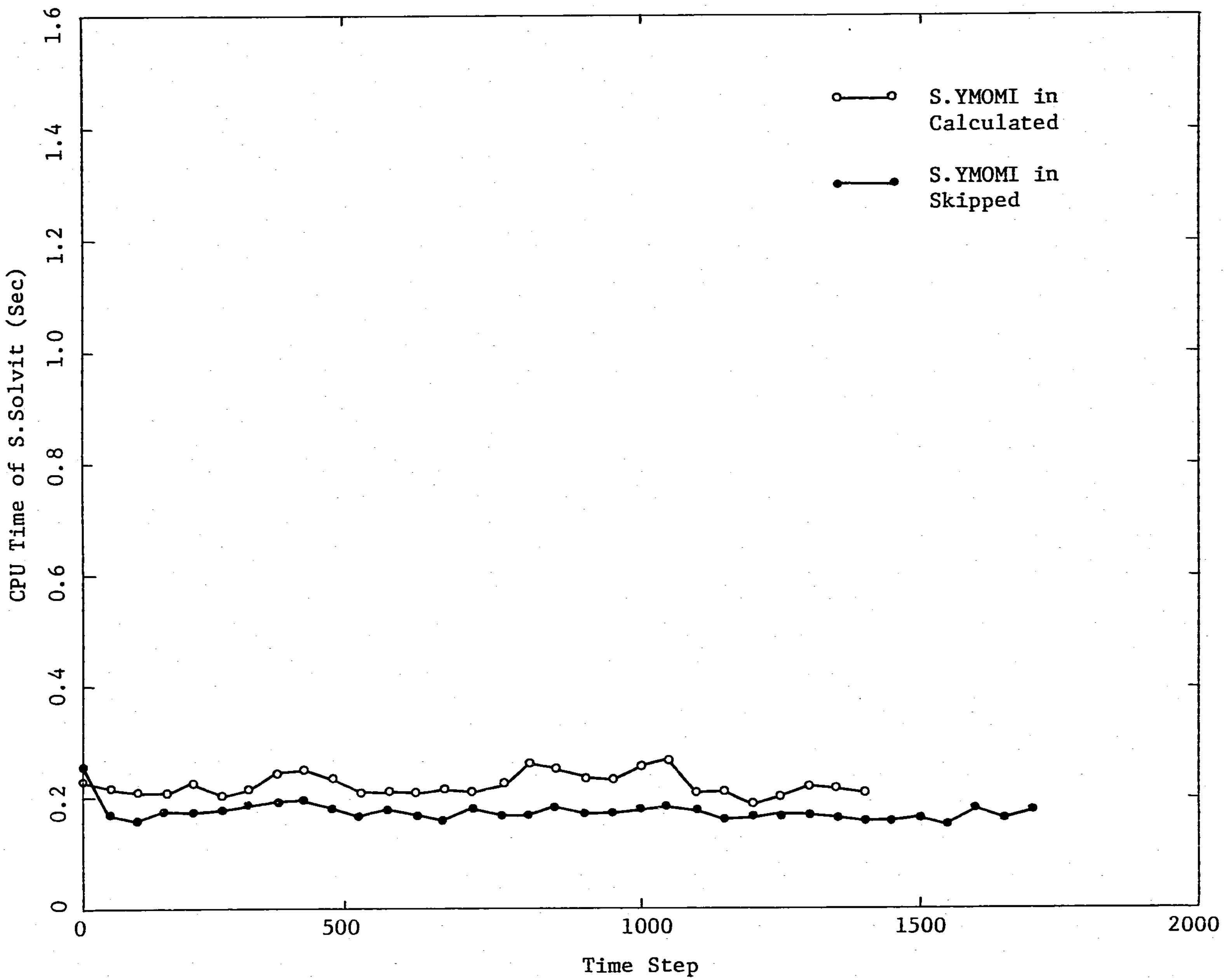


Fig. E.3 Effect of S.YMOMI Skip for CPU Time of S.SOLVIT