

## Fill-in 付き (M)ICCG 法のベクトル計算機上の性能

藤野 清次 (計算流体力学研究所)

竹内 敏己 (花王文理科学研究所)

横川 三津夫 (日本原子力研究所)

### Abstract

(M)ICCG 法は Fill-in を考慮して前処理を行うと収束性が格段に向上してくることは多くの論文で報告されている。スカラー計算機では収束性の向上がそのまま効率向上に直接結び付くことが多い。しかし汎用のベクトル計算機では収束性の他にベクトル長や特にメモリー衝突などにも十分注意を払わないと逆に効率低下を招くことがある。そこで、本論文では最初にベクトル計算機でのメモリー衝突が原因による効率低下の例を示す。次に超平面法でベクトル化した (M)ICCG 法は fill-in のタイプによってメモリーへのアクセス間隔が異なり、それを念頭に置いて格子点数を選ばないとメモリー衝突 (memory bank conflict) の影響を大きく受けることを指摘する。最後に、メモリー衝突の影響を最少限にするための指針を与える。この指針を守ることによって初めて fill-in 付き (M)ICCG 法の収束性が論じられることにも言及する。

### Performance of (M)ICCG method with fill-in on vector computers

Seiji Fujino

Institute of Computational Fluid Dynamics  
2-1-4, Haramachi, Meguro-ku, Tokyo 152, Japan

Toshiki Takeuchi

Institute of mathematical Science of KAO Corporation  
2-1-3, Bunka, Sumida-ku, Tokyo 131, Japan

Mitsuo Yokokawa

Japan Atomic Energy Research Institute  
Tokai-mura, Naga-gun, Ibaraki-ken 319-11, Japan

### Abstract

We consider performance of (M)ICCG method with fill-in on some vector computers. We show the deterioration of performance due to the effect of memory bank conflict. The performance of (M)ICCG method with fill-in depends on the convergence rate, vector length and especially memory bank conflict. We give a recipe how to choose the number of the gridpoints in the  $x$  direction with the least effect of memory bank conflict for gaining a high efficiency of (M)ICCG method with fill-in.

# 1 はじめに

(M)ICCG (Modified Incomplete Cholesky decomposition Conjugate Gradient method) 法は、fill-in を考慮する割合を増やす程反復回数が減少することは多くの報告で知られている [1],[2]。また (M)ICCG 法はベクトル計算機上では一般的に超平面法でベクトル化して使われることが多い。汎用のベクトル計算機上の効率を左右する要因として挙げられるものに、解法自体が持つ収束性の他にベクトル長の大きさや間接指標の有無がある。更に文献 [3] によって超平面法を使うときは特にメモリー衝突 (memory bank conflict) の影響が大きいことが分かってきた。例えば、2次元長方形領域で辞書式に格子の番号付けをしたとき、5点差分近似式を超平面法でベクトル化するとメモリーへアクセスする間隔は  $N_x - 1$  ( $N_x$  は  $X$  方向の格子点数とする) になる。そのとき  $N_x - 1$  が奇数になるように選ばないと機種によっては計算時間が2倍以上違ってくるということがわかっている [3]。また (M)ICCG 法は fill-in の割合を増やすほど逆にベクトル長は短くなり、必ずしも反復回数の減少が計算効率の向上に結びつかないことがわかってきた [4]。

そこで、本論文ではメモリーへアクセスする間隔によるメモリー衝突の影響を最少にするあるいは出来るだけ少なくすることを考える。その立場に立って収束性の向上した、しかし逆にベクトル長が短くなった fill-in 付き (M)ICCG 法のベクトル計算機上の性能を議論する。

## 2 Fill-in 付き (M)ICCG 法

2次元長方形領域を考え、格子点  $(i, j)$  の番号を辞書式、すなわち  $x(i)$  方向、 $y(j)$  方向の順に付けるものとする。また本論文では連立1次方程式を解くとき境界条件によらずすべての格子点を未知数の扱いにする。また  $x$  方向の格子点数を  $N_x$ 、 $y$  方向のそれを  $N_y$  と付ける。このとき不完全コレスキー分解における非零要素がある場所の集合の違いにより (M)ICCG 法は次の3つのタイプに分けられ、その集合を  $S_{1,1}, S_{1,2}, S_{1,3}$  とすると次のように表せる [1]。

$$(M)ICCG(1,1) \text{ 法: } S_{1,1} = \{(i, j) \mid j = i, i \pm 1, i \pm N_x\}$$

$$(M)ICCG(1,2) \text{ 法: } S_{1,2} = \{(i, j) \mid j = i, i \pm 1, i \pm N_x, i \pm (N_x - 1)\}$$

$$(M)ICCG(1,3) \text{ 法: } S_{1,3} = \{(i, j) \mid j = i, i \pm 1, i \pm N_x, i \pm (N_x - 1), i \pm (N_x - 2)\}$$

また超平面法を用いてベクトル化したときの各 MICCG 法の主な性質をまとめると表1のようになる [4]。

| 解法           | 指標間の関係    | 超平面の数            | 平均ベクトル長<br>( $N_x = N_y$ のとき) | アクセス間隔    |
|--------------|-----------|------------------|-------------------------------|-----------|
| (M)ICCG(1,1) | $i+j=一定$  | $N_x + N_y - 1$  | $O(N_x/2)$                    | $N_x - 1$ |
| (M)ICCG(1,2) | $i+2j=一定$ | $N_x + 2N_y - 2$ | $O(N_x/3)$                    | $N_x - 2$ |
| (M)ICCG(1,3) | $i+3j=一定$ | $N_x + 3N_y - 3$ | $O(N_x/4)$                    | $N_x - 3$ |

表1 (M)ICCG(1,1), (1,2), (1,3) 法の性質

この表からわかるように、Fill-in の割合を増やしていくにつれて (M)ICCG 法はベクトル長が短くなり、しかもメモリーへのアクセス間隔が1だけ変化する。このアクセス間隔のずれが (M)ICCG 法の効率評価に重大な影響を及ぼす。このことは第4章で指摘する。

### 3 アクセス間隔に対するメモリー衝突の影響

ここではアクセス間隔の違いによる計算効率の違いをベクトル乗算と加算だけからなる図1のプログラムで調べた。

```

N=10000
KANKAK=256
CALL TIMER(TS)
DO 100 K=1, KANKAK
DO 100 I=1, N*KANKAK, K
A(I)=B(I)+C(I)*D(I)
100 CONTINUE
CALL TIMER(TE)
CPU=TE-TS
    
```

図1 アクセス間隔と計算速度の関係を調べるプログラム

サブプログラム TIMER は CPU 計測のルーチンである。計算は倍精度で行った。間隔が1 (連続アクセスに相当) から 256 まで調べた。実際の計測は図1のプログラム中の文番号 100 の2重の DO ループを 500 回繰り返すその合計時間から計算速度を求めた。各アクセス間隔の計算速度の平均を Mflops 単位で表2に示す。本論文で数値実験のために使用したベクトル計算機は、S-820/80、SX-2、VP-200、VP-2400(FORTRAN77/VP V10L30, OS:MSP E20)そして日本原子力研究所のVP-2600/10(FORTRAN77 EX/VP V11L10 OPT(E)) である。各計算機の主な性能を表2に示す。

| 製造元  | 機種         | クロック数  | 最高速度       | バンク数 | 実装メモリー |
|------|------------|--------|------------|------|--------|
| 日立   | S-820/80   | 4.0 ns | 2.0 Gflops | 128  | 512 MB |
| 日本電気 | SX-2       | 6.0 ns | 1.3 Gflops | 512  | 256 MB |
| 富士通  | VP-2400    | 4.0 ns | 2.0 Gflops | 256  | 512 MB |
| 富士通  | VP-2600/10 | 3.2 ns | 5.0 Gflops | 512  | 512 MB |
| 富士通  | VP-200     | 7.0 ns | .57 Gflops | 256  | 256 MB |

表2 ベクトル計算機の主な性能

メモリーへアクセスする間隔が奇数 (1,3,5 ...)、偶数 (2,6,10,14 ...)、4の倍数 (4,12,20 ...)、8の倍数 (8,24,40 ...)、そして16の倍数 (16,32,48 ...) のときの計算速度 (Mflops) をプロットしたものを図2 (a), ... (e) に示す。またそれぞれの計算速度の平均値を表3に示す。以下本論文では、特に断わらない限りは計算速度はすべて Mflops 単位で表すものとする。

| No | アクセス間隔 | S-820/80 | SX-2 | VP-2400 | VP-2600/10 | VP-200 |
|----|--------|----------|------|---------|------------|--------|
| 1  | 奇数     | 543      | 294  | 250     | 1474       | 142    |
| 2  | 偶数     | 369      | 293  | 249     | 746        | 74     |
| 3  | 4の倍数   | 197      | 273  | 128     | 560        | 37     |
| 4  | 8の倍数   | 100      | 151  | 65      | 377        | 37     |
| 5  | 16の倍数  | 34       | 52   | 44      | 158        | 27     |
|    | 連続アクセス | 774      | 468  | 385     | 2500       | 250    |

表3 アクセス間隔ごとの計算速度の平均値

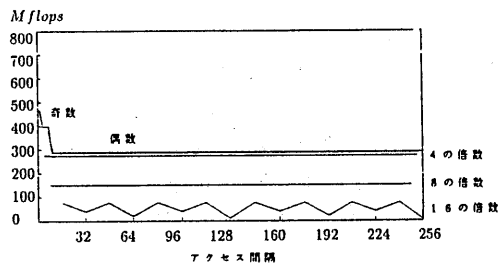
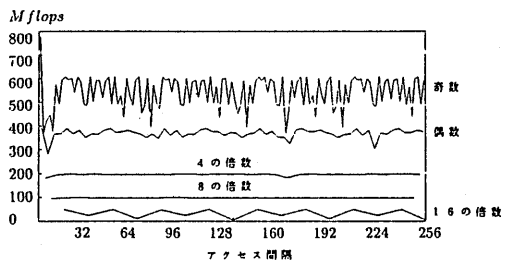
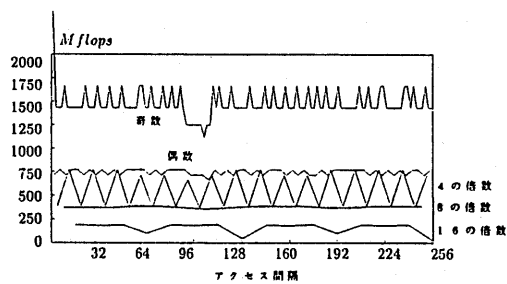
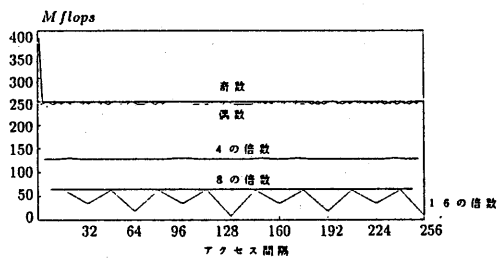
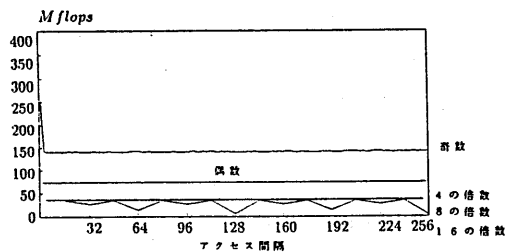


図2 (a) S-820/80 上のアクセス間隔毎の計算速度 (b) SX-2 上のアクセス間隔毎の計算速度



(c) VP-2400 上のアクセス間隔毎の計算速度 (d) VP-2600 上のアクセス間隔毎の計算速度



(e) VP-200 上のアクセス間隔毎の計算速度

この結果からメモリー衝突の影響が非常に大きいこと、そのため我々はメモリーへアクセスするとき4や8の倍数の間隔は極力避け、出来るだけ間隔が奇数になるように工夫すべきだということがわかる。さらに Fill-in 付き (M)ICCG 法のようにその Fill-in のタイプによってアクセス間隔が必然的に異なってくるときに、格子点数を同じにして各解法の効率を比較するのはメモリー衝突の影響が収束性の問題か判別するのが難しくなることに気付く。

## 4 数値実験

### 4.1 基礎方程式と計算条件

ここでは Fill-in 付き (M)ICCG 法の実際の計算効率を調べるために単位正方形領域で定義された次の基礎方程式を考える。

$$(k_1 P_x)_x + (k_2 P_y)_y = -f \quad (1)$$

物理定数のうち  $k_1$  は1の場合と25の場合について調べ、 $k_2$  は1に固定した。右辺項  $f$  は100とした。また境界条件は Dirichlet 条件で、 $x=0$  と  $x=1$  上では  $P=100$  そして  $y=0$  と  $y=1$  上では  $P=0$  とした。方程式を2次精度の中心差分で離散化すると5点差分近似式になりこれを MICCG 法で解いた。ベクトル化は簡便のためリストベクトル(間接指標)を用いた超平面法で行なった。各 (M)ICCG 法の収束判定条件は相対残差2乗ノルムで  $10^{-8}$ 、反復計算の初期値はすべて0とした。格子幅は、 $\Delta x = 1/(N_x - 1)$ ,  $\Delta y = 1/(N_y - 1)$ ,  $N_x = N_y$  とし、5通り ( $N_x, N_y = 249, 250, 251, 252, 253$ ) の格子数を選んで効率を調べた。演算は全て倍精度で行った。各格子点数の場合の (M)ICCG 法のメモリーへアクセスする間隔を参考までに表4に示す。

| 解法           | 格子点数 ( $N_x \times N_y$ ) |           |           |           |           |
|--------------|---------------------------|-----------|-----------|-----------|-----------|
|              | 249 × 249                 | 250 × 250 | 251 × 251 | 252 × 252 | 253 × 253 |
| (M)ICCG(1,1) | 248(8の倍数)                 | 249       | 250(偶数)   | 251       | 252(4の倍数) |
| (M)ICCG(1,2) | 247                       | 248(8の倍数) | 249       | 250(偶数)   | 251       |
| (M)ICCG(1,3) | 246(偶数)                   | 247       | 248(8の倍数) | 249       | 250(偶数)   |

表4 各 (M)ICCG 法のアクセス間隔

### 4.2 格子点数と (M)ICCG 法の性能評価基準に対する考察

さて我々は第3章で効率の面からアクセス間隔は出来るだけ奇数にすべきだということを学んだ。しかしアクセス間隔が1ずつ違う3つのタイプの (M)ICCG 法のアクセス間隔を同時に全て奇数にすることは明らかに不可能である。表5に数値実験で用いたアクセス間隔を示す。

| アクセス間隔    | S-820/80 | SX-2 | VP-2400 | VP-2600 | VP-200 |
|-----------|----------|------|---------|---------|--------|
| 247       | 432      | 290  | 250     | 1429    | 143    |
| 248(8の倍数) | 99       | 151  | 65      | 385     | 37     |
| 249       | 608      | 290  | 250     | 1429    | 141    |
| 250(偶数)   | 384      | 290  | 250     | 714     | 74     |
| 251       | 608      | 290  | 244     | 1667    | 143    |
| 252(4の倍数) | 198      | 273  | 128     | 714     | 37     |

表5 数値実験で用いたアクセス間隔の計算速度

この表からわかるように、Fill-in 付き (M)ICCG 法の効率比較を行うための最善の手は、たとえば (M)ICCG(1,1) 法と (M)ICCG(1,3) 法は格子点数:252<sup>2</sup>で実験を行い、アクセス間隔:  $N_x - 1$  と  $N_x - 3$  を奇数にする。一方、(M)ICCG(1,2) 法は格子点数がそれとは1つだけずれた 253<sup>2</sup>または 251<sup>2</sup>で実験を行い、アクセス間隔  $N_x - 2$  を奇数にする。これが各解法に一番公平であろう。このようにすればメモリー衝突による影響を最少限に防げる。(S-820/80 についてはもう少し細かい配慮が必要になる。) 次善の手として奇数間隔と偶数間隔の効率の差がほとんどない機種たとえば SX-2, VP-2400 では、格子点数 252<sup>2</sup> のような選びかたも考えられるだろう。

以上の考察を裏付けるために3つのタイプの (M)ICCG 法の格子点数ごとの計算速度 (Mflops) を図3 (a), ... (d) に表す。機種によって違いがあるが図2で示した結果と同様に、リストベクトルを使った超平面法による (M)ICCG 法のアクセス間隔と計算速度との間には強い相関関係があることが分かる。また S-820/80 上での各 (M)ICCG 法が収束するまでの CPU 時間 (msec) と格子点数との関係を図4 (a), (b) に表す。格子点数の僅かの違いで収束までの計算時間が大きく変化することがわかる。

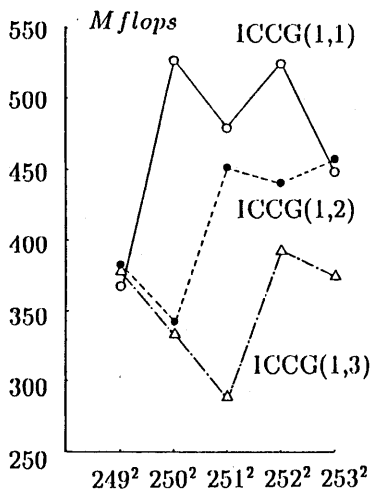
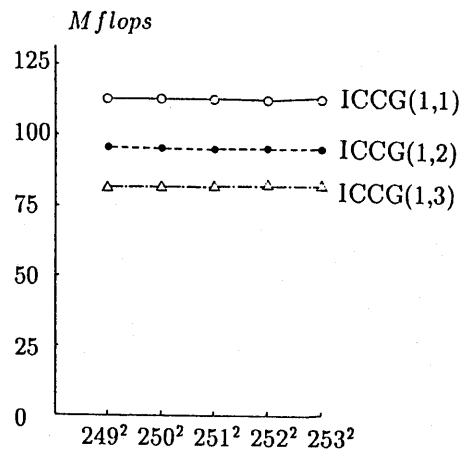
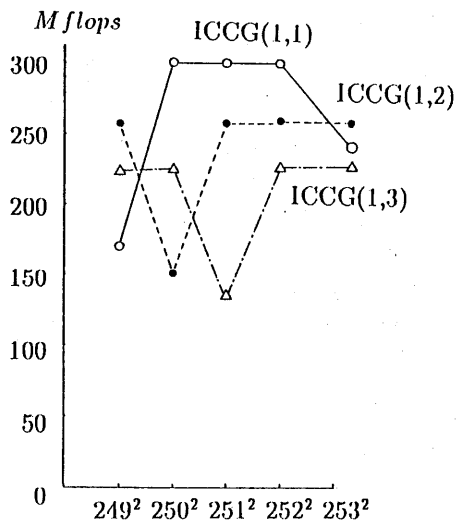


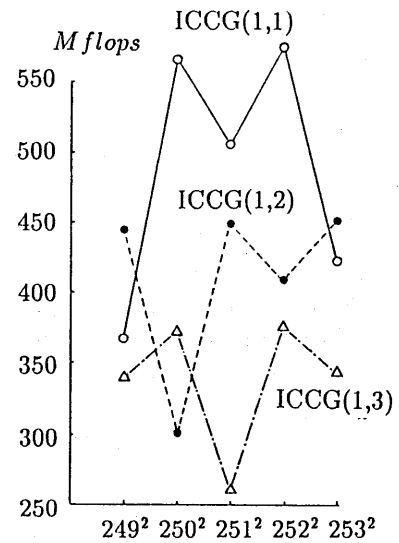
図3 (a) 格子点数ごとの計算速度 (on S-820/80)



(b) 格子点数ごとの計算速度 (on SX-2)



(c) 格子点数ごとの計算速度 (on VP-2400)



(d) 格子点数ごとの計算速度 (on VP-2600)

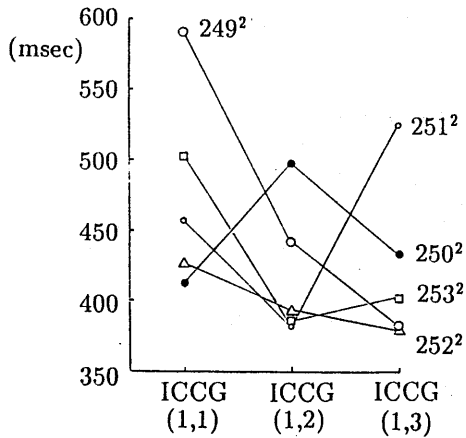
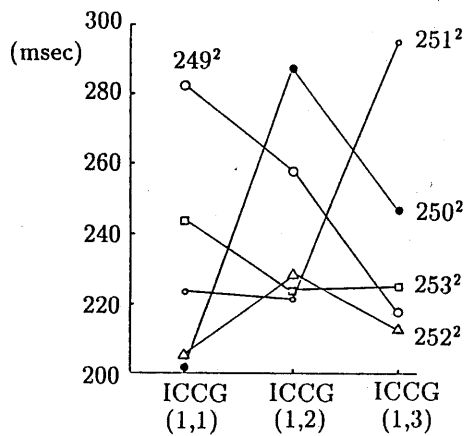


図4 (a) CPU時間と格子点数との関係 ( $k_1 = k_2 = 1$  のとき)



(b) CPU時間と格子点数との関係 ( $k_1 = 25, k_2 = 1$  のとき)

各(M)ICCG法が収束するまでに要した反復回数を表6に示す。S-820/80での(M)ICCG(1,3)法の反復回数だけが他の計算機の結果と違ったので括弧内に示す。また物理定数 $k_1 = k_2 = 1$ のときの各解法の計算時間を表7に、同様に物理定数 $k_1 = 25, k_2 = 1$ のときの各解法の計算時間を表8に示す。格子点数は第4章の考察結果をもとに、(M)ICCG(1,1)法と(M)ICCG(1,3)法は格子点数 $25^2$ で実験を行い、アクセス間隔 $N_x - 1$ と $N_x - 3$ を奇数にした。一方、(M)ICCG(1,2)法は格子点数 $25^2$ で実験を行い、アクセス間隔 $N_x - 2$ を奇数にした。物理定数 $k_1 = k_2 = 1$ のときFill-inを考慮すると(M)ICCG法は反復回数が大幅に減少することがわかる。一方、物理定数 $k_1$ を $k_2$ の25倍まで大きくするとFill-inを考慮したMICCG(1,2),(1,3)法の収束性はもとのMICCG(1,1)法に比べてほとんど改善されないことがわかる。

| 解法         | $k_1=k_2=1$ のとき | $k_1=25, k_2=1$ のとき |
|------------|-----------------|---------------------|
| ICCG(1,1)  | 110             | 41                  |
| ICCG(1,2)  | 76              | 39                  |
| ICCG(1,3)  | 59(58)          | 40(39)              |
| MICCG(1,1) | 53              | 25                  |
| MICCG(1,2) | 44              | 24                  |
| MICCG(1,3) | 41(33)          | 27(18)              |

表6 各(M)ICCG法の反復回数(括弧内の数字はS-820/80)

| 解法         | $N_x$ | アクセス間隔        | S-820/80 | SX-2 | VP-2400 | VP-2600 | VP-200 |
|------------|-------|---------------|----------|------|---------|---------|--------|
| ICCG(1,1)  | 252   | $N_x - 1=251$ | 426      | 1910 | 741     | 391     | 1360   |
| ICCG(1,2)  | 253   | $N_x - 2=251$ | 383      | 1850 | 682     | 388     | 1260   |
| ICCG(1,3)  | 252   | $N_x - 3=249$ | 380      | 1790 | 652     | 394     | 1240   |
| MICCG(1,1) | 252   | $N_x - 1=251$ | 205      | 925  | 358     | 189     | 657    |
| MICCG(1,2) | 253   | $N_x - 2=251$ | 222      | 1070 | 395     | 224     | 731    |
| MICCG(1,3) | 252   | $N_x - 2=249$ | 213      | 1270 | 461     | 278     | 881    |

表7 物理定数 $k_1=k_2=1$ のときの各解法の計算時間(単位:msec)

| 解法         | $N_x$ | アクセス間隔        | S-820/80 | SX-2 | VP-2400 | VP-2600 | VP-200 |
|------------|-------|---------------|----------|------|---------|---------|--------|
| ICCG(1,1)  | 252   | $N_x - 1=251$ | 159      | 744  | 277     | 147     | 509    |
| ICCG(1,2)  | 253   | $N_x - 2=251$ | 198      | 950  | 351     | 199     | 648    |
| ICCG(1,3)  | 252   | $N_x - 3=249$ | 251      | 1240 | 450     | 271     | 859    |
| MICCG(1,1) | 252   | $N_x - 1=251$ | 98       | 456  | 169     | 89      | 311    |
| MICCG(1,2) | 253   | $N_x - 2=251$ | 121      | 585  | 216     | 123     | 399    |
| MICCG(1,3) | 252   | $N_x - 3=249$ | 116      | 866  | 304     | 183     | 580    |

表8 物理定数 $k_1=25, k_2=1$ のときの各解法の計算時間(単位:msec)



## 5 まとめ

本論文では Fill-in 付き MICCG 法のベクトル計算機上の性能を判断するための基準を提案した。Fill-in 付き MICCG 法は Fill-in のタイプによってメモリーへのアクセス間隔が異なるので各解法の効率評価を行うとき収束性やベクトル長のほかにメモリー衝突にも十分注意を払う必要があることがわかった。またベクトル化のために超平面法を使用したときメモリー衝突の頻度はメモリーへのアクセス間隔の大きさとそれが起こる回数でその性能が決まるため、アクセス間隔ができるだけ奇数になるように  $x$  方向の格子点数を設定することが大切であることがわかった。以上の考察の結果、メモリー衝突の影響を最少限にした上で Fill-in 付き MICCG 法の効率評価をおこなえば得られた結果に対する信頼性が最も高いことがわかった。

## 6 謝辞

本論文をまとめることが出来たのは、図書館情報大学 長谷川秀彦氏に最新の研究成果 [4] を教授していただき効率評価の示唆を受けたお蔭である。それが本研究を進める上で理論的な裏付けになるとともに数値実験を行う上で非常に参考になった。慎んで感謝の意を表す。

## 7 参考文献

- [1] 村田健郎、名取 亮、唐木幸比古、「大型数値シミュレーション」、岩波書店、1990
- [2] 長谷川秀彦、春日里美、「対称正定値疎行列  $A$  と不完全コレスキー分解  $U^T U$  の固有値分布」、図書館情報大学研究報告、第 8 巻 1 号、1989
- [3] S.Fujino, M.Mori, T.Takeuchi, 'Performance of Hyperplane Ordering on Vector Computers', J. Comp. Applied Math. (submitted)
- [4] 長谷川秀彦、「対称正定値疎行列に対する PCG 法のプログラム」、図書館情報大学研究報告、1991 (to appear)