ハイパフォーマンス
コンピューティング 49－3
（1993．10．14）

ベクトル計算機上でのスカイライン法の高速ソルバ

長谷川　里美*，　田村　正義*，　後　保範*，　安達　斉**，　江口　義之**

＊　(株)日立製作所　ソフトウェア開発本部

＊＊日立東北ソフトウェア株式会社

ベクトル計算機を用いた応用分野の１つに構造解析がある。構造解析では，大規模なスカイライン行列を係数とする連立１次方程式を解く必要がある。連立１次方程式の解法には，スカイライン行列の構造に着目した手法である１行１列内積型スカイライン法を使うのが一般的である。

われわれは，ベクトル計算機上でのスカイライン法の高速化の１手法としてα行１列スカイライン法ソルバを開発し，スーパーコンピュータ HITAC S-3600 でいくつかのテスト問題を用いて性能評価を行った。

その結果，現在一般的な１行１列スカイライン法と比較して，３倍以上の高速性能を得た。

High Performance Skyline Methods on Supercomputers with Vector Processors

Satomi Hasegawa*, Masayoshi Tamura*, Yasunori Ushiro*,
Hitoshi Adachi**, Yoshiyuki Eguchi**

*　Software Development Center Hitachi,Ltd.

**　Hitachi Tohoku Software　Co., Ltd.

The structure analysis is an important application on supercomputers with vector processors. Large scale linear equation systems, which have sparse skyline matrices as coefficient matrices, should be solved in the field.

For that purpose, inner product type element-by-element skyline methods are popular now. The methods focus on the structure of non-zero elements in the skyline matrices.

A modified skyline method is proposed in this paper. Since α elements are processed partially simultaneously in the method, the methods must be faster than the conventional ones.

It is confirmed that the proposed method is more than 3 times faster than the element-by-element skyline method through numerical experiments on the HITAC S-3600 supercomputer.

1．はじめに

ベクトル計算機を用いた応用分野の１つに構造解析がある。構造解析では，偏微分方程式を有限要素法で離散化して，節点変位を求めるため，スカイライン行列を係数とする連立１次方程式を解く。スカイライン行列は，多くの成分がゼロ要素で，非ゼロ要素が対角付近に集まった疎な形をしている。

スカイライン行列を係数とする連立１次方程式の解法で，スカイライン行列の構造に着目した手法がスカイライン法である。

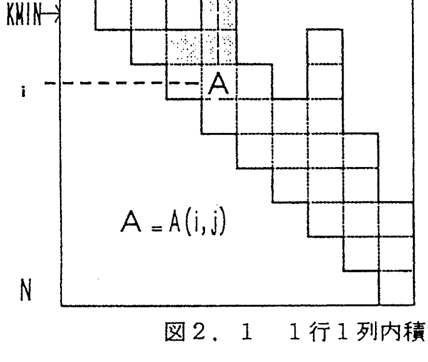
本報告では，２章で現在一般的な１行１列内積型スカイライン法について述べ，それに対する高速化である，（１）α行１列スカイライン法，と（２）期待できる効果について報告する。また，３章では，数値計算ライブラリを製品として開発している立場から，スカイライン法ソルバが満たすべき要件についても議論する。さらに，４章では得られた性能，特に，ユーザプログラムのソルバと比較したときのＣＰＵ時間と性能向上比について述べる。最後に，５章でα行１列スカイライン法に関する結論について述べる。

なお，本研究の成果は，ＨＩＴＡＣ　Ｓシリーズ及び，Ｍシリーズで稼働するＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２及び，ＭＡＴＲＩＸ／Ｍ／ＳＳＳ　Ｅ２として製品化している。

２．スカイライン法の高速化

２．１　１行１列内積型スカイライン法

１行１列内積型スカイライン法は，現在一般的なスカイライン法である。改訂コレスキー分解を用いてスカイライン行列を三角行列に分解後，前進代入，後退代入を行なって解を求める。このとき，スカイライン行列の列ごとの非ゼロ要素だけを１次元配列に詰めて記憶しておく。また，どの要素が１次元配列中の何番目にあたるのかを示したインデックスを用いて，１要素づつ参照しながら処理する。一般的なアルゴリズムは図２．１のようになっており，最内側ループでは内積演算が使われている。１行１列内積型スカイライン法では，内積演算でｓという同一の領域に足されるため，ベクトルロード，ストアと積和演算が並列に実行されても最後の結果を１つ得るたびに同期をとる必要がある。そのため，ベクトル演算とスカラ演算が並列に実行できるスーパーコンピュータ HITAC S-3000シリーズでもベクトル演算とスカラ演算を同時には実行できないため，高速化がされにくい。

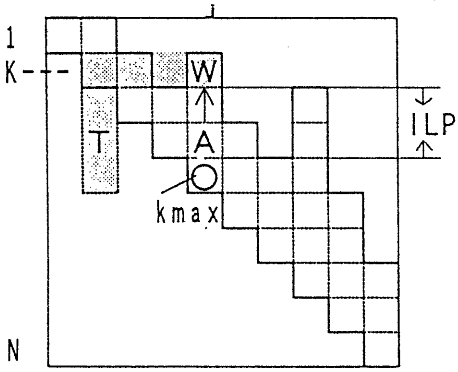



```
j = 1, N
   i = 1, j − 1
      s = 0
         k = kmin, i
         s = s + A(i,k) * A(k,j)
      A(i,j) = A(i,j) − s
```

図２．１　１行１列内積型スカイライン法の処理

2．2　1行1列ストア型スカイライン法

ベクトル演算とスカラ演算を同時に実行できるよう，最内側ループで内積演算を使わないようにしたのが図2．2に示す1行1列ストア型スカイライン法である。ここでは同期をとる必要がないため，ベクトル演算器が休むことなく稼働することが期待される。本報告では，この1行1列ストア型スカイライン法に対する高速化を行なう。



K＝1，n
　j＝k＋1，kmax
　　i＝1，ILP
　　　A＝A＋T＊W

図2．2　1行1列ストア型スカイライン法の処理

2．3　3行3列スカイライン法

3行3列スカイライン法は，スカイライン法の高速化の手法の一つとして文献1）にも示されている。3行3列スカイライン法は，分解処理の計算で3行3列を行列単位としてまとめて処理する方式を採用している。ＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２はこの手法をストア型スカイライン法で処理するという工夫を行った。この方式では，ベクトルレジスタ上のデータに対してより多くの演算が1度に実行されるため，スーパーコンピュータ HITAC S-3000シリーズの特質を活かした高速な処理ができる。しかし，3行3列スカイライン法はスカイライン行列が次の条件を満たしている場合にだけ適用できる。

（1）次元数が3の倍数

（2）非ゼロ要素の高さが3列づつそろっている

したがって，構造解析の対象である構造を分割したときの節点あたりの自由度が3，もしくは3の倍数であるときならばよい。特定方向の自由度だけを拘束するような条件がなければ，この条件を満足する。

2．4　α行1列スカイライン法

ＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２で提案した独自の高速化方式である。（1）α行1列スカイライン法ソルバの概要と（2）期待できる効果について述べる。

（1）α行1列スカイライン法の概要

α行1列スカイライン法は，α行を同時に処理する算法である。この方法は，3行3列スカイライン法が適用できない問題にも適用できる。また分解処理の単位であるα行の値は，プログラム中で最適な値を決める。

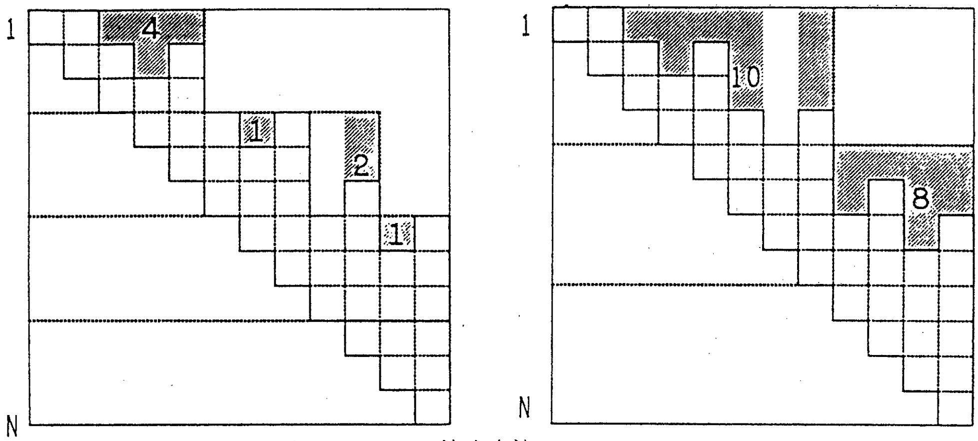


図２．３　αの決定方法

αを決定する処理の流れを図２．３に示す。ここでは$\alpha = 3$または４としている。

（ａ）図２．３に示すように，与えられたスカイライン行列を３行，４行のずつ区切る。分解処理の際，ゼロ要素なのに参照されてしまう無駄な要素数(図２．３のあみかけ部分)を数える。

３行で区切った場合：　８

４行で区切った場合：１８

（ｂ）各々の要素数をそれぞれの行数（３または４）で割る。

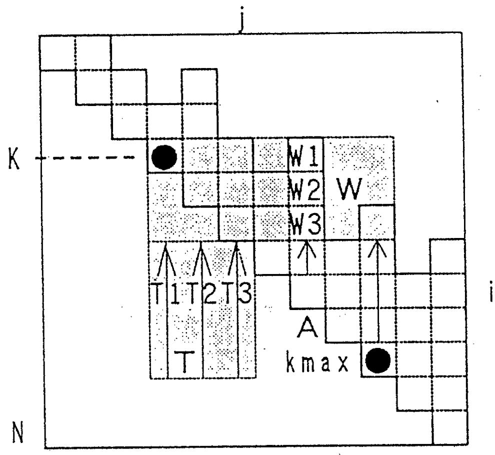
３行で区切った場合：　８／３＝２．６６

４行で区切った場合：１８／４＝４．５

（ｃ）結果が小さくなるときの行数を最適なαとして採用する。両方が同じ値の場合はαの大きいほう（$\alpha = 4$）を採用する。

$\alpha = 3$

αが決定した後のα行１列の処理を図２．４に示す。



Ｔの要素はＷの要素を対角で割って転置したもの

図２．４　α行１列スカイライン法の分解処理

図２．５にα行１列の分解処理の主消去部分を示す。

K＝1，N，3
　J＝K＋4，ｋｍａｘ
　　I＝1，ILP
　　　A＝A＋T1＊W1＋T2＊W2＋T3＊W3　①

図2．5　α行1列スカイライン法の主消去部分

図2．5の①では，Aに，3積和演算が一度に作用している。

（2）期待できる効果

スーパーコンピュータ HITAC S-3000シリーズでは，積和演算に対するベクトルロード，ストアの数を減らすことが高速化の鍵となる。そこで，積和演算に対するベクトルロード，ストアの数に着目して，表2．1に1行1列スカイライン法と4行1列スカイライン法（α＝4）の分解処理の主処理の部分の比較と命令数を示す。

表2．1　1行1列形式と一般形式の主処理部分の比較

<table>
<tr><th colspan="4">1行1列の処理</th></tr>
<tr><td colspan="4">

```
   DO 10 I=1,ILP
      A(I)=A(I)+S1*T1(I)
10 CONTINUE
```

</td></tr>
<tr><td rowspan="3">命令数</td><td>ロード</td><td>2</td><td rowspan="2">3</td></tr>
<tr><td>ストア</td><td>1</td></tr>
<tr><td>積和演算</td><td>1</td><td>1</td></tr>
</table>

<table>
<tr><th colspan="4">4行1列の処理</th></tr>
<tr><td colspan="4">

```
   DO 10 I=1,ILP
      A(a+I)=A(a+I)+S1*T1(t+I)
                   +S2*T2(t+I)
                   +S3*T3(t+I)
                   +S4*T4(t+I)
10 CONTINUE
```

</td></tr>
<tr><td rowspan="3">命令数</td><td>ロード</td><td>5</td><td rowspan="2">6</td></tr>
<tr><td>ストア</td><td>1</td></tr>
<tr><td>積和演算</td><td>4</td><td>4</td></tr>
</table>

表2．1より，1行1列スカイライン法ではベクトルロード，ストア：積和演算は3：1であり，4行1列では3：2になっている。一方，全体としては積和演算の総数は同じなので，4行1列スカイライン法でのベクトルロード，ストアの総数は1行1列スカイライン法の1／2になることが分かる。命令数の比較を図2．6に示す。

スーパーコンピュータS-3000シリーズでは積和演算とベクトルロード，ストアの並列実行が可能であることから，図2．6の命令数では1行1列スカイライン法が3に対し，4行1列スカイライン法は1．5となる。これを処理速度に直すと約2倍速くなることが期待できる。同様に，3行1列スカイライン法では，約1．8倍速くなることが期待できる。

以上より，α行1列スカイライン法の実行性能目標を1行1列型スカイライン法の2倍と設定した。

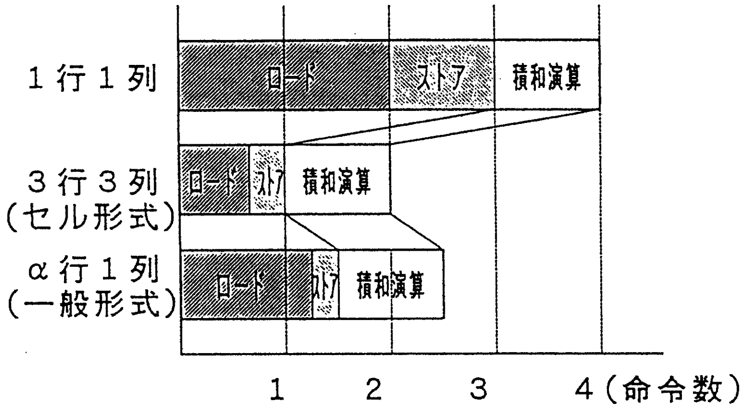


図2．6　命令数の比較

3．製品としてのスカイライン法ソルバ

製品としてのスカイライン法ソルバは，ユーザが使用している様々な構造解析プログラムのソルバ部分を置換できることが重要である。そのために重要な要件を表3．1に示す。

表3．1　製品としてのスカイライン法ソルバの要件

項番	要　　件	目　　的
1	入出力機能付き	大規模問題へ対応 メモリ所要量の削減
2	アドレス付け 対角から行列の縁へ（外） 行列の縁から対角へ（内）	既存の様々な構造解析プログラムに対応するため
3	スカイライン行列の要素もち方 対角を含む 対角を別にもつ	

4．実測と評価

4．1　各手法の比較

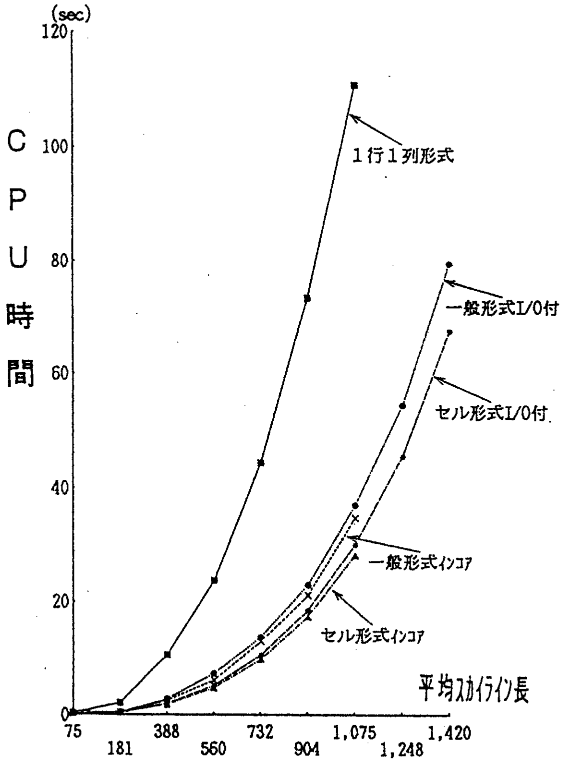
MATRIX／HAP／SSS　E2と，1行1列内積型スカイライン法プログラム[2)]の性能比較を行う。テスト問題として，表4．1に示す問題9例を用いた。

表4．1　問題データ

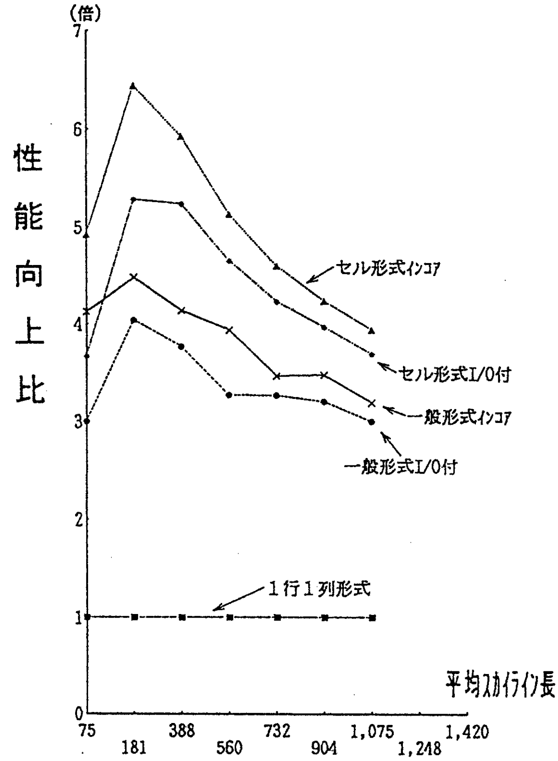
データ 項　目	問題1	問題2	問題3	問題4	問題5	問題6	問題7	問題8	問題9
次元数	1800	4500	9900	14400	18900	23400	27900	32400	36900
平均スカイライン長	75	181	388	560	732	904	1075	1247	1419

(注) 平均スカイライン長：全非ゼロ要素数／次元数

図4．1にCPU時間の比較を示す。1行1列形式は文献2)の1行1列内積型スカイライン法のソルバを示している。一般形式I/0付は入出力機能付きのα行1列スカイライン法，一般形式インコアは入出力機能なしのα行1列スカイライン法をそれぞれ示している。同様にセル形式I/0付は入出力機能付き3行3列スカイライン法，セル形式インコアは入出力機能なしの3行3列スカイライン法を示している。



(1) CPU時間



(2) 性能向上比

図4．1　1行1列内積型スカイライン法と各手法のソルバとの比較

測定条件は	・計算機	S－3600	・実行形態	マルチラン
	・オペレーティングシステム	VOS3／AS	・主記憶	256Mバイト

図4．1の(1)より，次のことが分かる。

(a) 1行1列内積型スカイライン法に比べてMATRIX／HAP／SSS　E2は高性能で解いていることが認められる。

(b) インコアで解けない大規模な問題（平均スカイライン長1248,1420）が入出力機能を使用することにより，解決できることが確認できる。

(c) 一般に3行3列スカイライン法（セル形式）はα行1列スカイライン法（一般形式）よりも性能が良い。しかし，その性能に大差はない。

(d) 拡張記憶を有効に使用したことにより，入出力機能の使用による性能劣化は少ない。

図４．１の（２）より，１行１列内積型スカイライン法に比べてＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２は３倍以上の性能向上比を示していることが認められる。セル形式は４～６倍、一般形式は３～４倍性能が向上している。この値は，一般形式について設定した目標値，２倍を達成している。

ここでの比較の対象は，現在一般的な１行１列内積型スカイライン法を用いている。目標性能はより高速なストア型スカイライン法との比較で決めたので，目標の達成は当然ともいえる。

図４．１の（２）で平均スカイライン長が１８１，３８８と比較的短い問題では、非常に高い性能向上比を示している。これはＳ－３０００シリーズではスカラセットアップとベクトル演算命令を並行実行していることと，セットアップの時間がベクトル演算量に比べて短くなっているためと考えられる。しかし，次元数，平均スカイライン長が大きくなると，しだいに理論的に求めた倍率に近づいてゆくことが予想できる。

４．２　構造解析プログラムのソルバ部分の置換

３章で示した要件をＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２として実現し，ユーザの使用している構造解析プログラムのソルバ部分を置換し，スーパーコンピュータ HITAC S-3600上で評価した結果，

（ａ）容易に置換可能であった

（ｂ）インコアで解けない大規模問題が解けるようになった

（ｃ）入出力機能による性能劣化は少ない

という成果が得られた。

５．むすび

ＭＡＴＲＩＸ／ＨＡＰ／ＳＳＳ　Ｅ２で導入したα行１列スカイライン法に関して次の結論を得た。

（１）現在一般的な１行１列内積型スカイライン法と比較して，約３倍以上の高速性能を得た。

（２）α行１列スカイライン法は高速に実行できる３行３列スカイライン法と比べて大差なく高速に実行できた。

（３）製品としてのスカイライン法ソルバの要件として入出力機能，アドレス付け，対角要素の持ち方に対する考慮が必要であることを明らかにした。

（４）構造解析プログラムのソルバ部分の置換を容易に行え，期待通りの効果が上がることを確た。

参考文献

１）村田健郎ほか３名著，工学における数値シミュレーション，丸善，１９８８．

２）小国　力ほか著，行列計算ソフトウェア，丸善，１９９１．