

流れ問題の行列非記憶な有限要素解法

広島大学 藤間 昌一 (Shoichi FUJIMA)
田端 正久 (Masahisa TABATA)

はじめに

流れ問題の有限要素解法では解析対象領域を有限要素に分割し、通常は、方程式に対応する要素剛性行列を要素ごとに組み立てて全体に足し合わせるにより全体剛性行列を作成し、連立一次方程式のソルバーを用いる。実際、非定常計算で右辺のベクトルだけが違う連立一次方程式を時間ステップごとに解く場合、全体剛性行列を作成する方法は最小の計算量を達成する。しかし最近登場しているベクトル型やパラレル型のスーパーコンピュータ環境では、必ずしも計算量と計算時間は比例しないので、計算量が最小の方法が最良とは言えず、従来の方法を再検討する必要がある[1]。

この論文では、流れ問題の混合型有限要素解法で全体剛性行列を組み立てずに計算する方法を提案する。計算機の特性に適合した工夫を行えば、記憶容量と計算速度の両方のパフォーマンスが向上できる可能性があることを示す。

第1章ではこの論文で考慮する偏微分方程式とその有限要素解法を示す。第2章では混合型有限要素近似の代表的な基底関数の組み合わせについて、自由度と行列のサイズの見積りを行い、計算実行上の第1の問題点<記憶量の増大>を提示する。第3章では、全体剛性行列を記憶しない有限要素解法を提案する。しかし、この方法では、第2の問題点<計算量の増大>があることを示す。第4章では、実際にスーパーコンピュータ上で、有限要素計算に現れる行列演算のベンチマークテストを実施し、工夫によりこれらの2つの問題点を同時に解決することが可能であることを示す。

1. 方程式と有限要素解法

R^m の有界領域 Ω で偏微分方程式,

$$cu + \text{grad } p = r, \quad x \in \Omega \quad (1a)$$

$$\text{div } u = 0, \quad x \in \Omega \quad (1b)$$

を考える。ここで、 $u(x) = (u_1(x), \dots, u_m(x)) : \Omega \rightarrow R^m$ と $p(x) : \Omega \rightarrow R$ は流速と圧力を表わす未知関数、 c は正定数、 $r = (r_1(x), \dots, r_m(x)) : \Omega \rightarrow R^m$ は既知関数である。方程式(1)は流れ問題などを解く際によく現れる方程式である。実際例を付録に示す。

(1)式に標準的な混合型有限要素近似の手続き[2]を施すことを考える。

初めに、(1)式を弱形式に変換する。次に、流速を近似する関数空間の基底関数を $\phi_\beta(x)e_j, j=1, \dots, m, \beta=1, \dots, Nu$ ($\phi_\beta(x) : \Omega \rightarrow R$ はスカラー関数、 $e_j \in R^m$ は x_j 方向の単位ベクトル)、圧力を近似する関数空間の基底関数を $\psi_\lambda(x), \lambda=1, \dots, Np$ として、

$$u \approx \sum_{j=1}^m \sum_{\beta=1}^{Nu} u_{\beta} \phi_{\beta}(x) \mathbf{e}_j$$

$$p \approx \sum_{\lambda=1}^{Np} p_{\lambda} \psi_{\lambda}(x)$$

のように近似を行う。NuとNpは領域の要素分割に対応して決まる流速と圧力の節点数で、 $u_{\beta} \in R^m$ 、 $p_{\lambda} \in R$ は流速と圧力の節点値である。弱形式に現れる重み関数にも基底関数と同じ形状のものを使うと（ガレルキン型近似），行列形の有限要素方程式，

$$Mu_h + B^T p_h = r_h, \quad (2a)$$

$$Bu_h = 0 \quad (2b)$$

が得られる。ここで、 $u_h \in R^{mNu}$ と $p_h \in R^{Np}$ は節点値からなるベクトルで、Mは質量行列，

$$[M]_{\alpha\beta}^{ij} \equiv c \int_{\Omega} \phi_{\alpha} \mathbf{e}_i \cdot \phi_{\beta} \mathbf{e}_j dx = c \delta_{ij} \int_{\Omega} \phi_{\alpha} \phi_{\beta} dx$$

Bは発散行列，

$$[B]_{\kappa\beta}^j \equiv - \int_{\Omega} \psi_{\kappa} \operatorname{div}(\phi_{\beta} \mathbf{e}_j) dx = - \int_{\Omega} \psi_{\kappa} \frac{\partial \phi_{\beta}}{\partial x_j} dx$$

r_h は既知ベクトル，

$$[r_h]_i^{\alpha} \equiv \int_{\Omega} r \cdot (\phi_{\alpha} \mathbf{e}_i) dx = \int_{\Omega} r_i \phi_{\alpha} dx$$

である。

連立一次方程式(2)は流速と圧力が連成していてサイズが大きい。そこで、以下のような適合離散化圧力ポアソン方程式[3]（以下ではPPEと略すことがある）を経て解かれる。

まず、質量行列Mを対角行列である集中質量行列 \bar{M} ，

$$[\bar{M}]_{\alpha\beta}^{ij} = \delta_{ij} \delta_{\alpha\beta} \sum_{\beta} [M]_{\alpha\beta}^{ij}$$

で近似する。 $B\bar{M}^{-1}$ を(2a)式に乘じれば，

$$(B\bar{M}^{-1}B^T)p_h = B\bar{M}^{-1}r_h \quad (3)$$

が得られる。左辺の行列はラプラス作用素の近似になっており、サイズが小さく、性質の良い連立一次方程式を与えている。得られた p_h を(2a)式に代入することにより u_h を求めることができる。

2. 記憶量に関する考察

有限要素方程式(2)を計算速度、記憶量（必要な計算機のメモリの量）の観点から効率的に解く方法を考える。実用上、2次元($m=2$)と3次元($m=3$)の場合を考えれば十分である。

2.1 自由度

初めに、正方形領域($m=2$)、立方体領域($m=3$)のフレドリクス・ケラー型の m -単体 ($m=2$ のとき3角形、 $m=3$ のとき4面体) による有限要素分割を考える(図1)。

各辺の分割数を n とすれば、得られる分割の要素数 N_e は、 $N_e = 2n^2$ ($m=2$), $6n^3$ ($m=3$)である。また、頂点の数 N_p は $m=2$ のとき $n^2 = \frac{1}{2}N_e$, $m=3$ のとき $n^3 = \frac{1}{6}N_e$ と見積られる。一般の領域の m -単体分割でも $2N_p \cong N_e$ ($m=2$), $6N_p \cong N_e$ ($m=3$)の比率から遠くはずれるのは特殊な場合だけである。従って、基準となる領域に対して見積りを行い、結果を N_p に対する比で表示すれば、一般領域の分割に対する見積りとしても役立つ。

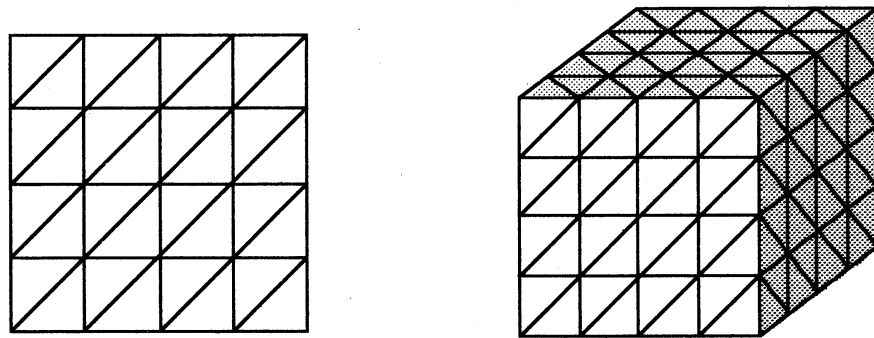


図1. フレドリクス・ケラー型の要素分割 ($m=2,3, n=4$)

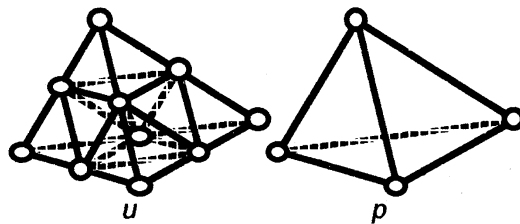


図2. 3次元P2同相P1 / P1要素

次に、基底関数の組み合わせの代表例としてP2同相P1/P1要素[4]を考える。これは圧力の基底関数 $\psi_x(x) \in C(\bar{\Omega})$ が要素上1次関数で、流速の基底関数 $\phi_a(x) \in C(\bar{\Omega})$ が要素を 2^m 個に分割した各々の小さい m -単体上で1次関数であるものである(図2)。この組み合わせは、Inf-Sup条件を満たしており、(3)式の一意可解性が保証される[5]。

P2同相P1/P1要素における節点数は、圧力の節点数 N_p に対して、流速の節点数は $4N_p$ ($m=2$), $8N_p$ ($m=3$)ある。流速ベクトルの自由度は、その m 倍になる。また、計算機上でよく使われる倍精度の実数は記憶に8バイト必要である。これらをまとめると表1のようになる。

いくつかの実際的な N_p の値に関する記憶量も併せて表1に示してある。現在の計算機の性能を考えれば、 $n=100$ 程度ならば大規模な分割、 $n=10$ ならば小規模な分割と言うのが目安であろう。すなわち、2次元ならば、 $N_p=100$ 程度が小規模分割、 $N_p=10^4$ 程度が大規模分割であり、3次元ならば、 $N_p=1000$ 程度が小規模分割、

$Np = 10^6$ 程度が大規模分割となる。

表1. p_h と u_h に関する自由度と記憶量 (P2同相P1/P1要素)

	m=2		m=3	
	ph	uh	ph	uh
Number of Nodal Points	Np	$4Np$	Np	$8Np$
Degree of Freedom	Np	$8Np$	Np	$24Np$
Memory Size [bytes]	$8Np$	$64Np$	$8Np$	$192Np$
$Np=100$	0.8KB	6.4KB		
$Np=1000$	8KB	64KB	8KB	190KB
$Np=10000$	80KB	640KB	80KB	1.9MB
$Np=100000$			800KB	19MB
$Np=1000000$			8MB	190MB

2.2 行列のサイズ

発散行列 B のサイズを見積る。これは疎行列であるので、非ゼロ成分だけを圧縮して記憶するのが有力な方法である。 $[B]_{\kappa\beta}^j$ が非ゼロとなる必要条件是、 $\psi_\kappa(x)$ と $\phi_\beta(x)$ の台に重なりがあることである。フレドリクス・ケラー型の分割では、 $\psi_\kappa(x)$ の節点が領域内部にあれば、そのような $\phi_\beta(x)$ は19個($m=2$)又は65個($m=3$)ある(図3)。従って、行列 B の非ゼロ成分数は、 $m=2$ では $Np \times 19m = 38Np$ 、 $m=3$ では、 $Np \times 65m = 195Np$ となる。

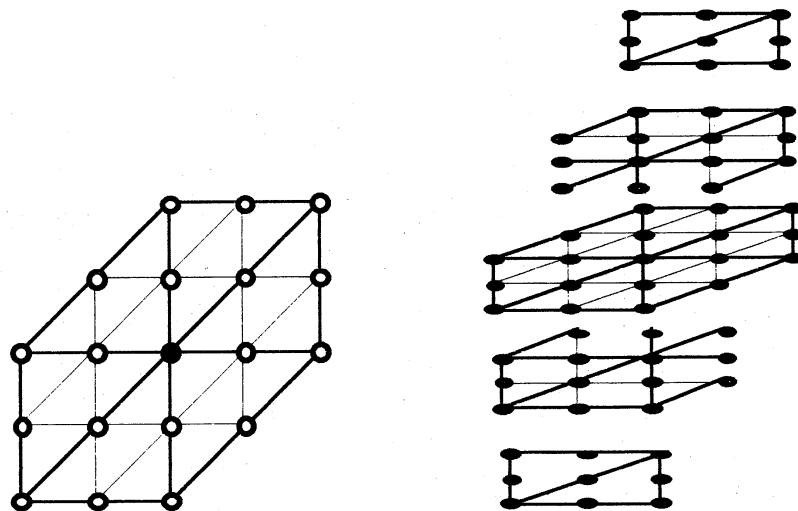


図3. $\psi_\kappa(x)$ と $\phi_\beta(x)$ の台に重なりのある $\phi_\beta(x)$ の節点($m=2,3$)

次に、適合離散化圧力ポアソン方程式のサイズを見積る。 $[BM^{-1}B^T]_{\alpha\alpha}$ が非ゼロとなる必要条件是、ある $\phi_\alpha(x)$ が存在してその台が $\psi_\kappa(x)$ と $\psi_\lambda(x)$ のそれぞれの台と重な

りをもつことである。より感覚的な言葉で言えば、各圧力の節点に関して2つ隣の節点まで非ゼロ成分が現れうる。結局、疎な対称帯行列となり、連立一次方程式に直接法と反復法のどちらを使うかによって記憶法が異なってくる。

反復法を用いるために非ゼロ成分だけを圧縮して記憶する方法では、1行当たり19個($m=2$)又は65個($m=3$)の非ゼロ成分があるので、対称性を考慮すればその数は、 $m=2$ では $Np \times 19/2 \cong 10Np$ 、 $m=3$ では $Np \times 65/2 \cong 33Np$ となる。

直接法を用いるために帯行列として記憶する方法では、半帯幅は「2つ隣り」の節点のうちの最も離れた節点番号との差になるため、 $2Np^{(m-1)/m}$ となり、必要な記憶量は $Np \times 2Np^{(m-1)/m} = 2Np^{(2m-1)/m}$ となる。表2にこれらを具体的な Np の値と共に示す。

表2. 行列の記憶量 (P2同相P1/P1要素)

	m=2			m=3		
	B	PPE(疎)	PPE(帯)	B	PPE(疎)	PPE(帯)
words	38Np	10Np	$2Np^{(3/2)}$	195Np	33Np	$2Np^{(5/3)}$
bytes	304Np	80Np	$16Np^{(3/2)}$	1560Np	264Np	$16Np^{(5/3)}$
$Np=100$	30KB	8KB	16KB			
1000	300KB	80KB	500KB	1.6MB	260KB	1.6MB
10000	3MB	800KB	16MB	16MB	2.6MB	74MB
100000				160MB	26MB	3.4GB
1000000				1.6GB	260MB	160GB

(注) 「疎」は反復法の、「帯」は直接法のための記憶法。

2.3 考察

現在の計算機の利用可能なメモリ量は、100MB~1GB程度が目安である[6]。

表2から、2次元問題に関しては、大規模な要素分割に対しても、今考えている有限要素計算をメモリ上で実行することが可能なことがわかる。

これに対して3次元問題では、記憶量の制約を受けるため直接法は実用的でなく、反復法を利用することになる。しかし、中規模な要素分割においても行列 B が大きくなるほか、PPEの行列や u_h のサイズが大きくなり、全体がメモリ上に収まらなくなる恐れがある。磁気ディスク装置を利用した仮想記憶方式等により実行可能であっても、計算速度は極端に低下することになる。

今、第一の制約となっているのは、行列に必要な記憶量である。そこで以降では、3次元問題について、行列を記憶しないで有限要素法計算を進める方法を提案する。

3. 全体剛性行列を組み立てずに有限要素法計算を進めるには

PPEの行列 $\overline{B}M^{-1}B^T$ は性質が良い。そこで連立一次方程式の解法にCG法を用いることを考える。CG法の手順の中で行列のデータを使う演算は、行列とベクトルの掛け

算だけである。演算 $(\overline{B}M^{-1}B^T)q_h$ を (q_h は p_h と同じサイズのあるベクトルである) 行列 B と \overline{M}^{-1} のデータを利用して,

$$B\{\overline{M}^{-1}(B^T q_h)\}$$

のように行うことにより, PPE行列は構成しなくても済む。但し, 演算量は増加する。

結局, 行列とベクトルの積,

$$Bv_h \quad (4a)$$

$$B^T q_h \quad (4b)$$

(v_h は u_h と同じサイズのあるベクトルである) を求めることができれば, 有限要素法計算が進行できることがわかる。この型の演算は, (3)式の右辺の計算と u_h を求める際の(2a)式の計算にも現れることに注意する。

そこで, 行列 B を記憶する代わりに, Bv_h , $B^T q_h$ の計算をそれぞれ専用の手続き (FORTRAN言語における副プログラム) を用意する方法を提案する。要素剛性行列の足し合わせにより全体剛性行列が組み立てられる有限要素法の特長を利用すれば, 例えば演算 $B^T q_h$ は,

$$B^T q_h = \left(\sum_e B^{eT} \right) q_h = \sum_e (B^{eT} q_h)$$

により行うことができる。ここで, B^e は行列 B に対応する要素剛性行列であり, \sum_e は全要素について和を取る。

従ってFORTRAN言語のプログラムでは, $B^T q_h$ は,

```
!-----
      vh=0          !      配列代入
      DO ie=1,Ne
!      次の文をj=1,2,3,β=1,2,...,10について書き下す(DOループではない)
          vh(j,NDEβ(ie))=vh(j,NDEβ(ie))+
              ( BjNDE1(e),NDEβ(e) *qh(NDE1(ie)) &
                + BjNDE2(e),NDEβ(e) *qh(NDE2(ie)) &
                + BjNDE3(e),NDEβ(e) *qh(NDE3(ie)) &
                + BjNDE4(e),NDEβ(e) *qh(NDE4(ie)))
      END DO
!-----
```

により書くことができる。ここで, 配列,

REAL vh(3,Nu),qh(Np)

は v_h, q_h に対応するものであり, 整数配列,

INTEGER NDE1(Ne),NDE2(Ne),...,NDE10(Ne)

は要素番号に対してその要素に属する節点の番号を与える表である。 B^e の各成分の計算式が上のDOループ内の対応する場所書き込まれ, DOループ実行中に計算されることとなる。

具体的には, 4面体要素では B^e の各成分は要素の4頂点の座標から計算可能である。

しかし、1つの成分を求めるためには数10回の乗除算が必要であり、行列を記憶しておく方法に比べて、必要な計算量が数10倍に増えてしまっていることになる。

結局、従来の計算量が計算速度に比例する計算機環境では、このような記憶量の削減と計算の高速化は相反する要求である。しかし、最近登場しているコンピュータ（ベクトル型スーパーコンピュータや並列計算機など）ではこの比例関係が成立しないものが多くなってきている。これらの環境では、計算方法と計算機の設計との調和度により計算速度が決まる。例えばベクトル型のスーパーコンピュータでは、同じ内容の独立な計算を多く繰り返すときに極めて高速な計算が可能である。その意味で、上述のプログラムはベクトル型スーパーコンピュータの高速性を活かす可能性をもっている。

4. 総合的なパフォーマンス向上のための試み

全体剛性行列を構成しない有限要素法計算について、記憶量と計算速度の両面でパフォーマンスが向上する可能性を、P2同相P1/P1要素に対して調べる。

4.1 中間的計算量の保存

座標から計算される中間的な計算値を予め計算して記憶しておけば、計算量を極端に増大させずに記憶量を削減することが可能になる。

P2同相P1/P1要素では、行列 B^e の計算に必要な流速の基底関数の微分は小4面体上で区分的に定数である。それらの定数は元の4面体の重心座標の微分から単純な式（微分の整数倍またはそれらを加えたもの）で計算が可能である。予めそれらの微分を求めて記憶しておけば、成分毎の計算量が大幅に削減可能になる。

重心座標の微分は、要素ごとに12個のデータになり、領域全体では $12Ne = 72Np$ 語のデータ量になる。行列 B の大きさが $195Np$ 語であったのに比べれば、必要な記憶量は小さいと言える。

4.2 要素番号の付け直し

ベクトル型スーパーコンピュータの高速性を活かすには、ループの演算の独立性を保証する必要がある。例えば、前頁に挙げたプログラムのDOループに関しては、固定の $\beta(1 \leq \beta \leq 10)$ に対する節点番号列 $NDE\beta(ie), ie=1, \dots, Ne$ がスーパーコンピュータ用語の「リストベクトル」に相当している。リストベクトルにはデータの重複があってはならないので、 $NDE\beta(ie), ie=1, \dots, Ne$ には同じ節点番号が2度現れてはならないことになる。しかし4面体要素分割では、この条件を満たすことは不可能である。実際、 $Ne \cong 6Np$ から、同じ節点番号が平均して6度も現れることがわかる。

代表的なスーパーコンピュータであるHITAC S-3800では、実際にはリストベクトルの条件が緩く、同じデータが現れる間隔が512以上離れていれば問題がない。従って、要素番号を適切に付け直して、この条件をみたすようにすれば、スーパーコン

コンピュータの高速性を活かすことが可能になる。

図3にこの条件を満たす要素番号の付け直しを行った例を示す。縦方向に同じ節点番号が現れる要素番号の最小差は1263である。

Element Number	Nodal Point Number			
	NDE1()	NDE2()	NDE3()	NDE4()
697 :	1101	1542	1122	1543
2031 :	1102	1543	1101	1122
3365 :	1543	1122	1102	1123
4699 :	1122	1563	1543	1542
6033 :	1543	1122	1564	1563
7367 :	1122	1123	1543	1564
10035 :	1543	1984	1542	1563
11369 :	1984	1563	1543	1564
16638 :	1564	1123	1543	1124
16708 :	1521	1542	1080	1543
17971 :	1565	1124	1564	1543
18041 :	1522	1543	1521	1080
19304 :	1124	1543	1565	1544
19374 :	1543	1080	1522	1081
20637 :	1543	1102	1124	1123
20707 :	1080	1101	1543	1542
21970 :	1124	1543	1103	1102
22040 :	1543	1080	1102	1101
23303 :	1543	1544	1124	1103
23373 :	1080	1081	1543	1102
28635 :	1984	1543	1565	1564
29968 :	1565	1984	1544	1543
30038 :	1984	1521	1543	1542
31371 :	1521	1522	1984	1543
32637 :	1102	1081	1543	1082
33970 :	1103	1082	1102	1543
35303 :	1082	1543	1103	1544
36636 :	1543	1522	1082	1081
37969 :	1082	1543	1523	1522
39302 :	1543	1544	1082	1523
40635 :	1543	1522	1984	1523
41968 :	1544	1523	1543	1984

図3. 要素番号付け直し後の要素節点对応表の例, $N_e = 48000$, $N_p = 9261$
1543番の節点をもつ要素だけを表示した。

4.3 ベンチマークテスト

ここで述べた方法と従来の全体剛性行列 B の非ゼロ成分だけを記憶する方法の計算時間の比較を表3に示す。これらは図3に示した要素分割のデータを用いて、行列演算(4)を1回行う時間を測定したものである。'vector'はHITAC S-3800の自動ベクトル化コンパイラによる計算時間を、'scalar'は同じ計算機でベクトル化を行わない場合の計算時間を示す。'scalar'の計算時間からもわかる通り、新しい方法は要素剛性行列

の各成分を必要な度に再計算するので計算量は非常に増加している。しかしベクトル計算機では、ベクトル計算機の高い加速率が得られるため計算時間が約1/16に短縮されることがわかった。

表3. 計算時間の比較 (単位: 秒)

		Present	Conventional	Ratio
Bvh	Vector	0.004	0.061	0.06
	Scalar	0.409	0.124	3.3
tBqh	Vector	0.006	0.106	0.06
	Scalar	0.576	0.12	4.8

おわりに

流れ問題の混合型有限要素解法で、全体剛性行列を組み立てずに行う方法を提案し、ベクトル型スーパーコンピュータ上のベンチマークテストにより、記憶量と計算速度の両面でパフォーマンスが向上することを示した。

スーパーコンピュータは現在、大規模並列型へと遷移しつつある。並列計算機では、メモリがプロセッサごとに分散して配置されている型が多い。そのような環境では、全体剛性行列を構成するのは不可能であり、ここで述べた手法を含む何らかの手法を検討する必要がある。

参考文献

- [1] N. Kikuchi, T. Torigaki, K. Suzuki and J. M. Guedes, Recent development of finite element methods in the high-speed computing environment, *Super-Computing - Applications, Algorithms and Architectures*, J. Kondo(ed.), Springer-Verlag, Tokyo, 20-71, (1991).
- [2] F. Thomasset, *Implementation of the Finite Element Methods for the Navier-Stokes Equations*, Springer, 1981.
- [3] P. M. Gresho, S. T. Chan, R. L. Lee and C. D. Upson, A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations, Part 1: Theory, *Int. J. Num. Meth. Fluids*, 4, 557-598, (1984).
- [4] M. Bercovier and O. Pironneau, Error estimates for finite element method solution of the Stokes problem in the primitive variables, *Numer. Math.* 33, 211-224(1979).
- [5] M. Tabata and S. Fujima, An upwind finite element scheme for high Reynolds number flows, *Int. J. Num. Meth. Fluids*, 12, 305-322(1991).
- [6] 例えば, 東京大学大型計算機センターニュース, 1995.1.
- [7] S. Fujima, M. Tabata and Y. Fukasawa, Extension to three-dimensional problems of the upwind finite element scheme based on the choice of up- and downwind points, *Comp.*

Meth. Appl. Mech. Engrg., 112, 109-131, (1994).

- [8] C. B. Jiang, M. Kawahara, K. Hatanaka and K. Kashiya, A three-step finite element method for convection dominated incompressible flow, *Comp. Fluid Dyn. Journal*, 1, 443-462, (1993).

付録 方程式(1)に帰着する解法の実例

Ω を m 次元 ($m=2,3$) の有界領域とする。非定常非圧縮Navier-Stokes問題:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \cdot \text{grad} u - \frac{1}{Re} \Delta u + \text{grad} p &= f, & x \in \Omega, t > 0 \\ \text{div} u &= 0, & x \in \Omega, t > 0 \end{aligned}$$

を考える。ここに、 $u(x,t): \Omega \times (0, +\infty) \rightarrow R^m$ と $p(x,t): \Omega \times (0, +\infty) \rightarrow R$ は流速と圧力を表わす未知関数、 Re は Reynolds 数、 $f(x,t): \Omega \times (0, +\infty) \rightarrow R^m$ は外力を表わす既知関数である。

この問題に関して時間方向に前進差分近似を考える[5, 7]。 τ を時間刻み、 u^n, p^n を時刻 $n\tau$ における関数 u, p の値とすると、差分方程式は、

$$\begin{aligned} \frac{1}{\tau} u^{n+1} + \text{grad} p^n &= \frac{1}{\tau} u^n - u^n \cdot \text{grad} u^n + \frac{1}{Re} \Delta u^n + f^n & x \in \Omega, n = 0, 1, 2, \dots \\ \text{div} u^{n+1} &= 0 & x \in \Omega, n = 0, 1, 2, \dots \end{aligned}$$

となる。この方程式は、流速 u^n が与えられたときに、圧力 p^n と流速 u^{n+1} を求める問題になっている。これは、式(1)の型の問題である。

他にも時間方向3段階陽的離散化法[8]など多くの解法で(1)の形の方程式が現れている。