研究速報

流れ場シミュレーション・プログラムにおける計算機性能の比較

Comparison of Computer Performance on the Programs for Flow Simulation

谷口伸行*
Nobuyuki TANIGUCHI

1. はじめに

数値シミュレーションに関する最近のコンピュータ環境をみると，ベクトル演算プロセッサーを用いたスーパーコンピュータ開発から，RISCチップによる単体CPUの演算性能向上，さらに，並列型の超スーパーコンピュータへと，技術開発がめまぐるしく行われてきた．その結果，ユーザーが利用できるコンピュータ資源は，数年前に比べて，その大きさも種類も格段に向上しており，当然ながらコンピュータ性能に対する評価も変えねばならない．実際にコンピュータ・プログラムを製作，実行するユーザーにとって，適切なコンピュータ環境を選んでいるか，また，使用したコンピュータが十分に性能を発揮しているかは，きわめて重要な問題である．しかし，コンピュータ性能を評価するデータにはコンピュータ開発者の立場から示されたものが多く，数値シミュレーションなどのためのアプリケーション・プログラム実行性能を推定するには必ずしも利用しやすい情報とはいえない．

本速報では，流れ場数値シミュレーション研究において実際に利用しているコンピュータ・プログラムをチューニングする目的で，各種のパーソナルコンピュータ，ワークステーション，スーパーコンピュータにおいて実行テストを実施した結果について報告する．ただし，これらの実行テストは少数の特定プログラムのみを対象に行ったもので，また，実際のプログラム実行環境（マルチユーザー，マルチジョブ環境）のままで実施したため，計測値の一般性や再現性は必ずしも保証できない．また，本報告に記載されていない機種の詳細な仕様や，運用されているソフトウェアーによっても結果が変わる恐れがあることを，あらかじめ指摘しておく．本報告のデータを利用する際には，これらの点に十分に注意されたい．

*東京大学生産技術研究所　第2部

2. サンプルプログラムの内容

流れ場数値シミュレーションは大規模な数値計算を必要とする代表的な事例であり，また，その主要部分である大規模配列の反復演算は，コンピュータの得意とする処理の一つであることもあって，実用的な流れ場シミュレーションプログラムにおいても最大演算性能に比較的近い効率での実行が可能である．よって，流れ場シミュレーションは汎用コンピュータの実数演算性能を評価するのに適した課題の一つと考えられる．

本報告では，乱流の数値シミュレーション法であるラージ・エディ・シミュレーション（LES）のために開発した2種のプログラムをサンプルとして用いた．以下に，本計算での数値解析の概要を示す．解析法の詳細は文献［1］，［2］を，解析内容については文献［3］を参照されたい．

非圧縮性乱流場LESの基礎方程式は以下の通りである。

運動方程式；

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\nu_t\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \quad (1)$$

$$\nu_t=(Cs\Delta)^2\sqrt{\frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)}$$

圧力方程式；

$$\frac{\partial^2 p}{\partial x_i\partial x_i}=-\frac{\partial D}{\partial t}+\frac{\partial^2}{\partial x_i\partial x_j}\left(\nu_t\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \quad (2)$$

$$-\frac{\partial^2 u_i u_j}{\partial x_i\partial x_j},\quad D=\frac{\partial^2 u_i}{\partial x_i}$$

このうち，運動方程式(1)は各座標方向3成分それぞれについて与えられる．式(1)，(2)は非圧縮性流れ場の一般的な基礎式であり，乱流粘性ν_tを算出する部分がLES（本

研　究　速　報

計算では，Smagorinsky SGS モデルを適用）特有の計算となる．LES では式(1)および式(2)から流れ場，すなわち，速度 u_iと圧力 pの時間変化を得る．ここで，式(1)に 2 次精度 Adams-Bashforth 法，式(2)に 1 次精度 Euler 法を用いて時間微分について時間刻み δtで離散化すると，

$$u_i^n - u_i^{n-1} = \delta t\{1.5\, f_i(u^{n-1}, p^{n-1}) - 0.5\, f_i(u^{n-2}, p^{n-2})\} \quad (3)$$

$$D^{n+1} - D^n = \delta t\, \{\nabla^2 p^n + g_i(u^n)\} \quad (4)$$

を得る．ここで，上付き添え字 $n-2$, $n-1$, n は離散化した時間ステップをあらわす．式(3)は速度成分 u_i^nの 3 成分について，式(4)は連続の式の条件

$$D^{n+1} = 0 \quad (5)$$

を代入したのち圧力 p^nについての偏微分方程式とみなして，これらを順次計算することで時間ステップ nの数値解を得る．以上の数値解法は MAC 法と呼ばれ，非圧縮性流れ場シミュレーションに広く用いられている方法である．

(3)，(4)式の右辺 f_i, g_iは空間微分のみを含む．本計算では，偏微分方程式の数値解析手法として一般的な有限差分法（finite defference method; FDM）と有限要素法（finite element method; FEM）を用いた 2 種類の計算プログラムを使用した．それぞれの計算手法を表 1 にまとめて示した．

表 1　サンプルプログラムの解析法

プログラム名	LES_FDM	LES_FEM
離散化法	有限差分法	有限要素法
計算ｱﾙｺﾞﾘｽﾞﾑ	MAC 法	MAC 法
時間離散化ｽｷｰﾑ	2 次精度 Adams-Bashforth	2 次精度 Adams-Bashforth
計算格子分割・空間離散化ｽｷｰﾑ	直交直線座標格子 2 次精度中心差分	6 面体双 1 次要素 Galarkin 法
圧力ﾎﾟｱｿﾝ式の行列解法	ICCGS 法	CG 法
LES モデル	Smagorinski モデル	Smagorinski モデル

3. 実行テスト結果

FDM および FEM を適用した LES 乱流解析プログラムによるチャンネル乱流シミュレーションをサンプルとして実行テストを行った．初期条件には前もって同じプログラムで計算した乱流場の数値解を代入し，少数回の時間ステップだけ計算を進行した（実際の解析では約10000ステップを実行する）．実行したサンプル計算を表 2 に，使

表 2　実行テストの計算条件

計算ケース	プログラム	計算点（節点）数	使用ﾒﾓﾘ	時間ｽﾃｯﾌﾟ数
FDM-0	LES_FDM	32x64x32=65,536	27MB	10
FDM-1	〃	〃	〃	30
FDM-2	〃	〃	〃	100
FEM-0	LES_FEM	49,060	22MB	1
FEM-1	〃	〃	〃	4
FEM-2	〃	〃	〃	10

表 3　使用した計算機とその仕様

No.	ｺﾝﾋﾟｭｰﾀ機種名（共同利用場所、IP 名）	CPU	主ﾒﾓﾘ	OS	ｺﾝﾊﾟｲﾗｰ
1	DEC Alpha 600-5/333 (生研、beta)	DEC 21164 (333MHz)	384MB	OFS 1v3.2c	DEC Fortran
2	Hitachi 9000 735/125 (生研、eta)	PA-Risc (125MHz)	400MB	HP-UX 9.05	Fortran
3	IBM RS/6000 590 (生研、mu)	IBM Power2 (66MHz)	128MB	AIX 3.2.5	Fratran
4	SiliconGraphics Indy	MIPS R4600 (100MHz)	64MB	IRIX 5.3	Fortran77 4.02
5	Fujitsu S-4/10	Super SPARC10	64MB	Solaris 2.1	Fujitsu Fortran90v1
6	Hewlett-Packard 9000 735	PA-Risc (125MHz)	64MB	HP-UX 9.0	Fortran
7	SiliconGraphics Cray Origin 2000	MIPS R10000 (200MHz)	1GB	IRIX 6.2	MIPS Power Fortran 7.0
8	Kubota Computer Titan 3000	MIPS R4000	96MB	Titan OS 4.2	Fortran
9	Cray EL94		512MB	UNICOS 9.0	CF90 V2.0
10	Dynus GAIA275AXP	DEC 321064 (275MHz)	128MB	Windows NT 4.0	DEC Fortran 1.1
11	Visual Technology Alpha300AXP(kagiki)	DEC 221164 (300MHz)	128MB	Windows NT 4.0	DEC Fortran 1.1
12	Fujitsu VX (生研、fuji)		2 GB	UXP/V	VP-Fortran
13a	Hitachi S3800/480 (大型、svos)		2 GB	VOS 3 /FS	HAP Fortran90
13b	〃 (大型、sunix)	〃	〃	HI-OFS/1MJ	HAP Fortran90

＊機種名は導入時の名称

＊共同利用～ 生研: 生産技術研究所計算機室　IP 名.iis.u-tokyo.ac.jp

大型: 大型計算機センター　IP 名.cc.u-tokyo.ac.jp

用した計算機およびその仕様を表 3 に示す．ここで用いた計算機は東京大学に設置されているもので，大型計算機センター，生産技術研究所共同利用計算機室，あるいは，著者の所属する研究室にて運用されている．実行テストは，実際のプログラム実行環境を想定して，通常のマルチユーザー/マルチジョブ環境で実施した．そのため，計測値の一般性や再現性は必ずしも保証できないが，いくつかのテストから計算時間の測定誤差は数％程度と推定している．また，ここに記載していない機種の詳細な仕様や，運用されているソフトウェアーによっても結果が変わる恐れがあることを指摘しておく．なお，Fortran コンパーラーのオプションについては，以下に説明するものを除いては，メーカーの標準推奨値を用いている．コンパイラーの自動最適化による効果は小さくないこともあるので（No. 7, 自動最適化を行わない場合（no-Opt）を参照），使用機種の解説書に従うことをお勧めする．

表 4 に各計算機による計算時間を，図 2 には機種 No. 1 (DEC600 500/333) を基準とした換算性能を示した．今回用いた機種のいくつか（No. 3, 7, 8, 9, 12, 13）は複数の CPU で構成されており並列計算が可能であるが，実行テストはすべて単体 CPU で実施した．

各手法のプログラム上の特徴として，FDM では各計算点の長い配列に対する計算が最内側 DO ループとなるが，FEM では要素毎の節点（本プログラムでは 8 点）に対する計算が最内側 DO ループとなることが多い．ベクトル演

修正前のオリジナルプログラム

```
      DO 10 JY=1,LY
      DO 20 KZ=1,LZ
      DO 30 IX=1,LX
       I=1 +(IX-1) +LX*(JY-1) +LX*LY*(KZ-1)
       P(I)= D(I) * ( R(I) - A(I,1) * P(I-1)
     *                    - A(I,2) * P(I-M1)
     *                    - A(I,3) * P(I-M2) )
     *      - D(I) * (  A(I,5)*P(I-MP1)
     *                + A(I,6)*P(I-MP2)
     *                + A(I,7)*P(I-MP3) )
 30   CONTINUE
 20   CONTINUE
 10   CONTINUE
```

最内側 DO ﾙｰﾌﾟ（IX=1,LX）において、
右辺代入値 P(I)と
左辺参照値 P(I-1)、P(I-M1)、…で同じ配列ｱﾄﾞﾚｽ に ｱｸｾｽ する。
↓
最内側 DO ﾙｰﾌﾟ がﾍﾞｸﾄﾙ化しない

ベクトル化修正後のプログラム

```
      DO 10 JY=1,LY
      DO 20 IH=1,LX+LZ-1
      IS=1 + IH      +LX*(JY-1) -LX*LY
      ID=      -1                +LX*LY
      KZ1=max(1,IH-LX+1)
      KZ2=min(LZ,IH)
      IS1=IS+ID*KZ1
      IS2=IS+ID*KZ2
< directive option >                    ←
      DO 30 I=IS1,IS2,ID
       P(I)= D(I) * ( R(I) - A(I,1) * P(I-1)
     *                    - A(I,2) * P(I-M1)
     *                    - A(I,3) * P(I-M2) )
     *      - D(I) * (  A(I,5)*P(I-MP1)
     *                + A(I,6)*P(I-MP2)
     *                + A(I,7)*P(I-MP3) )
 30   continue
 20   continue
 10   continue
```

ｺﾝﾊﾟｲﾗへ以下を明示するｵﾌﾟｼｮﾝ
最内側 DO ﾙｰﾌﾟ（I=IS1,IS2,ID）において、
右辺代入値 P(I)と
左辺参照値 P(I-1)、P(I-M1)、…で同じ配列ｱﾄﾞﾚｽ に ｱｸｾｽ しない。

＊機種による ＜directive option＞の表記　（参考文献[5],[6],[7]）
Cray EL94、Titan：　CDIR$ IVDEP
Hitachi S3800　　：　*VOPTION VEC
Fujitsu VX　　：　!OCL NOVREC

図 1　演算順序の入れ替えによるプログラムのベクトル化修正の事例（サンプルプログラム LES-FDM の行列反復解法（ICCGS 法）の部分.）

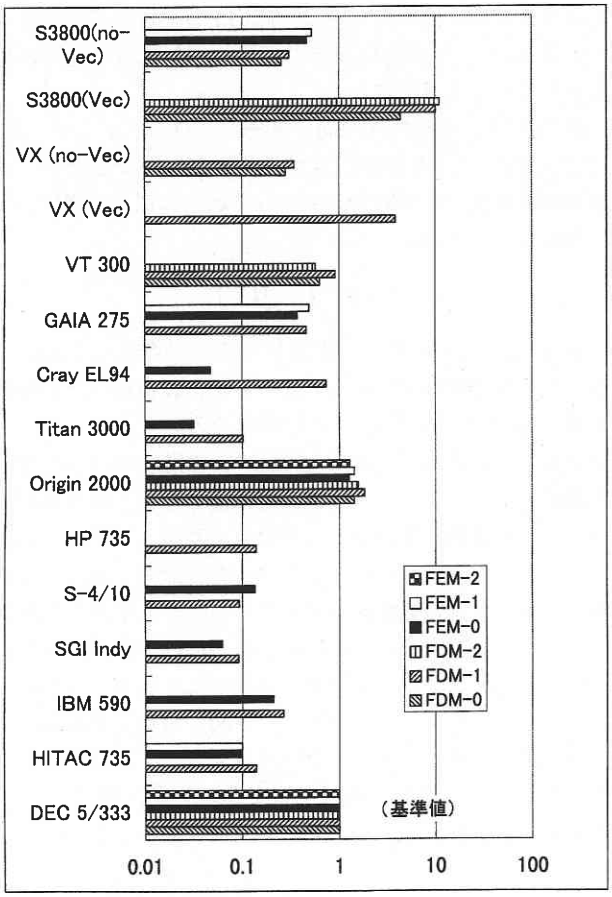

図 2　各計算機演算速度の相対比較（DEC 5/333を基準として計算時間比の逆数で表示した.）

表 4　実行テスト結果

No.	Computer System	FDM-0	FDM-1	FDM-2	FEM-0	FEM-1	FEM-2
1	DEC 600-5/333	114	385	1164	111	412	992
2	Hitachi 9000-735/125		2762		1139	4135	
3	IBM 590		1452		533		
4	Silicon Graphics　Indy		4234		1792		
5	Fujitsu S-4/10		4138		821		
6	HP 9000-735/125		2781				
7	SG-Cray Origin 2000 (no-Opt) (no-Vec)	80	210 663 125	742	88	288	772
8	Kubota Titan3000/V		3804		3495		
9	Cray EL94		528		2343		
10	Dynus GAIA275AXP		840		300	840	
11	VT Alpha 300AXP	180	420	2040			
12	FujitsuVX (no-Vec)	405	100 1115				
13a	Hitachi S3800/VOS	26	38	105			
13b	〃　/unix (no-Vec)	450	1266		238	789	

＊数値は、time ｺﾏﾝﾄﾞなどで表示される全 CPU 時間(sec)、空欄は実施せず。
＊no-Vec は、ベクトル化修正をしないプログラム（図 1）を使用。
＊no-Opt は、コンパイル時に自動最適化を用いない場合。

算プロセッサーを持たない機種においては，FDM および FEM プログラムの演算速度はほぼ同じように変化することから，本実行テストの規模の計算では，これらの点に関してプログラム・チューニングをする必要はないと考えられる．一方，ベクトル演算プロセッサーを持つ機種（No. 8, 9, 12, 13）では，FEM プログラムについてコンパイラーによる自動ベクトル化（最適化）が十分に行われておらず演算速度が大きく劣っている．一方，FDM においては行列反復解法の一部で計算アルゴリズムを修正することで（図 1，文献 [4]），ベクトル演算プロセッサー利用比（＝ベクトル演算プロセッサー実行時間/全 CPU 実行時間）約85％と高性能が得られているが，これらを行わない場合（No. 12, No. 13, no-Vecと表記）は FEM の場合と同程度かより劣る結果となる．なお，この修正はベクトル演算プロセッサーを持たない計算機では不用であり，むしろ，計算時間の増加をまねくが，本実行テストでは主に計算機ハードウェアーの性能を判定する目的のため全ての計算機で修

正プログラムを使用している．修正を行わないプログラム（No. 7, no-Vec と表記）の結果では約60％に計算時間が短縮される場合もみられる．

また，CPU 演算性能が高いスーパーコンピュータ（No. 12, No. 13）では入出力に要する時間が無視できない．これらの計算機で計算時間がステップ数に比例して増加しない理由の一つと考えられる．

4. おわりに

本報告では，著者らが開発した流れ場数値シミュレーションの典型的なプログラムについて，13機種の計算機を単体 CPU で用いた場合の実行性能を示した．しかし，これらの計算機のいくつかは複数 CPU で構成され，並列計算機能をもつ．よって，並列計算機能を考慮した同様の実行テストを計画している．並列計算を効率よく実行するにはプログラム修正が必要となるが，並列計算を指示するためのプログラム言語が統一されていないことが大きな障害となっている．Fortran の最新仕様 Fortran 90[8] においても並列化プログラムが明確に定義されていないことは，コンピュータ・ユーザーにとって不満が残る点である．

今回用いたサンプルプログラムは Fortran 77に準拠して開発された．しかし，他のプログラム言語，C(C＋＋)，Pascal，Java などは，インターフェースやネットワークとの関連性では優れた面もあり，高い演算速度が得られるならば数値シミュレーションのプログラム言語として受け入れられる可能性も高い．これらの言語によるシミュレーション・プログラムの実行テストも進めていきたい．

なお，本実行テストに使用したプログラムの入手については著者（ntani@iis.u-tokyo.ac.jp）へ連絡されたい．

（1997年 5 月30日受理）

参 考 文 献

1) 谷口他，第 9 回 NST シンポジウム論文集（1994），49-52
2) M. Tsubokura 他，ASME FED 215 (ASME-JSME Joint Conf. Fluid Eng., 1995), 81-87
3) 谷口他，生産研究49-1（1997），11-18
4) J. J. Dongara 他（小国訳），コンピューターによる連立一次方程式の解法，丸善（1993），155-162
5) スーパーコンピュータ HITAC の S-3800の効率的利用法(1), (2), (3)，センターニュース Vol. 26-5（1994）51-79，Vol. 26-6（1994）68-102，Vol. 27-1（1995）48-64
6) Fujitsu VX 使用手引，東京大学生産技術研究所電子計算機室 U-97-A-031（1997）14-15
7) CF90™ Commands and Directives Reference Manual, Cray Research Inc. SR-3901 2.0（1995）79-91
8) M. Metcalf 他（西村他訳），詳解 Fortran 90（bit 別冊），共立出版（1993）