

# 数値シミュレーションの基礎 (5)

青木 孝

神奈川大学 理学部 情報科学科

Introduction to Numerical Simulation(5)

Takashi Aoki

Department of Information Science, Kanagawa University

**Abstract.** I compared with PC(Pentium4+WindowsXP and Core2 Duo+WindowsVista Business) and Fortran Compiler(Fujitsu and Intel) for numerical simulation CPU time.

## 1 はじめに

行列方程式のソルバ：数値解法(直接法:ガウスと反復法)のCPU時間が、

「Pentium4 クロック 3.20GHz、2GB メモリ、システムバスクロック 800MHz、  
2次キャッシュ2MB、OS：WindowsXP」と、

「Intel Core2 Duo クロック 2.40GHz、2GB メモリ、システムバスクロック 800MHz、  
2次キャッシュ3MB、OS：WindowsVista Business」

のPCで、どのように差がでるのか関心があり調べた。

Fortran コンパイラは、

「富士通 Fortran V4.0L10」と、

「Intel Fortran 10.1」

があり、

「Pentium4 + WindowsXP」には「富士通 Fortran V4.0L10」を

「Intel Core2 Duo + WindowsVista Business」には「富士通 Fortran V4.0L10」と「Intel Fortran 10.1」の両方をインストールして調べた。

比較に使う行列方程式は、まず、熱方程式 ( $u = u(x, y)$ : 板の温度分布、 $f(x, y)$ : 熱源分布):

$$(1.1) \quad \operatorname{div}(-\kappa \nabla u) = f$$

を使う。この式 (1.1) の拡散係数である熱伝導率  $\kappa(x, y)$  が、場所  $(x, y)$  に依存する場合に、熱分布  $u(x, y)$  を、CV(Control Volume)法あるいは、いわゆる台形法(クランク-ニコルソン)で解くとする。

この熱方程式 (1.1) を、横 ( $x$  方向)11cm × 縦 ( $y$  方向)10cm の場で解き、「数値シミュレーションの基礎 (4)」で使った CV 法(対称離散化行列)とクランク・ニコルソン法(不均等あみ目では非対称離散化行列になる。)により離散化して得た行列方程式を利用する。

ソルバは、直接法ガウスとして: GLUR(行ガウス)、GLU(列ガウス: Fortran では有利)、BGLU(非対称帯ガウス)、また反復法として: ICCG12(対称用)、BCGSTB13(非対称用) を比較する。各ソルバで、「PC + コンパイラ」を比較する際の行列元数等のシミュレーション条件は、Table 1.1 の通り。

Table 1.1 Simulation condition for each SOLVER

solver	MJ1	MJ	N
GLUR(Sym)	6	6	3900
GLU(Sym)	6	6	3900
BGLU(NonSym)	35	8	57720
ICCG12(Sym)	109	109	1305820
BCGSTB13(NonSym)	164	41	1309950

この Table 1.1 で、MJ1=MJ は均等あみ目、MJ1≠MJ(MJ= $\frac{1}{4}$ MJ1) は不均等あみ目を表す。CV 法では、あみ目が均等であろうが不均等であろうが離散化行列は対称となる。一方、クランク・ニコルソン法では、均等あみ目ならば離散化行列は CV 法と同じ対称行列になるが、不均等あみ目では離散化行列が非対称行列となる。反復法では、非対称行列のソルバは対称行列のソルバに比べ約 3 倍余計に CPU 時間がかかるので、クランク・ニコルソン法の離散化方式は不利となる。以後、非対称用のソルバの CPU 比較実験をする際には、不均等あみ目をクランク・ニコルソン法により離散化した非対称行列を利用した。

## 2 CPU 比較実験

各 PC と各 Fortran コンパイラの組による CPU 時間の比較は次のようになる。Intel コンパイラは、通常の最適化モード「ifort」と、ループが自動並列化する「ifort/Qparallel」オプション、ループがベクトル化する「ifort/fast」オプションがあり、3 種のモードに変えて実験した。結論から言えば、「/Qparallel」と「/fast」オプションは、「Intel Core2 Duo + Windows Vista Business」の PC にはほぼ効かない。

Table 2.1 CPU time(sec) for each PC and Fortran Compiler

solver	N	Pen4XP+FjFort	Core2Vist+FjFort	Core2Vist+iFort(/Qpar)[/fast]
GLUR	3900	827	297	304(296)[288]
GLU	3900	73	84	84(85)[83]
BGLU	57720	9	6	5(5)[3]
ICCG12	1305820	51	50	49(53)[51]
BCGSTB13	1309950	167	157	154(130)[215]

Table 2.1 によれば、PC「Intel Core2 Duo」において、富士通 Fortran(FjFort) と Intel Fortran(iFort) コンパイラによる CPU 時間の差は、ほぼないことが分かる。同じ FjFort コンパイラで、PC を「Pen4XP」と「Core2Vist」に変えた時には、BGLU、ICCG12、BCGSTB13 のソルバによる CPU 時間の差はほぼない。しかし、GLU と GLUR ソルバにおいては、PC のハードによる差が見られる。GLU による CPU 時間は両 PC でほぼ変わらないが、 $\frac{GLUR}{GLU}$  の比が、「Pen4XP」では 11.5 倍、「Core2Vist」では 3.5 倍となる。通常は 3 倍程度が正しい。この  $\frac{GLUR}{GLU}$  の CPU 時間比の差は、CPU のハードの 1 次キャッシュのサイズに起因していると考えられる。Pen4 は、1 次キャッシュ 16KB(2 次キャッシュ 2MB)、Core2 Duo では、

この2倍の1次キャッシュ32KB(2次キャッシュ3MB)となっている。したがって、Pen4では、1cmをMJ分割してあみ目の精しさを表す、あみ目の分割数：MJ=4で、全節点数(行列の元数N)がN=1720となり、この配列N語長(8×N Byte)がちょうど14KBにあたり、1次キャッシュに1列分が入ることになり、このシミュレーション条件がほぼ計算効率の境界となる。一方、Core2 Duoでは、1次キャッシュが2倍なので、MJ=6で元数N=3900、配列N語長が32KBになるシミュレーション条件で、ほぼ計算効率の境界が現れる。実際にTable 2.1によれば、Core2 DuoではFortranコンパイラに依らず、MJ=6までは $\frac{GLUR}{GLU}$ のCPU時間比は3倍であるが、MJ=7ではほぼ5倍になる。一方、Pen4では、MJ=5までは、 $\frac{GLUR}{GLU}$ のCPU時間比が6倍程度だったものが、MJ=6以上になると、このCPU時間比が10倍を超えてくる。

Table 2.2  $\frac{GLUR}{GLU}$  CPU time(sec) ratio for each PC and Compiler

solver	MJ	N(KB)	Pen4XP+FjFort	Core2Vist+FjFort	Core2Vist+iFort
GLUR	4	1720(14)	45(sec)	22(sec)	22(sec)
GLUR	5	2700(22)	151(sec)	87(sec)	87(sec)
GLUR	6	3900(32)	827(sec)	297(sec)	304(sec)
GLUR	7	5320(43)	3520(sec)	1007(sec)	998(sec)
GLU	4	1720(14)	7(sec)	7(sec)	7(sec)
GLU	5	2700(22)	25(sec)	28(sec)	28(sec)
GLU	6	3900(32)	73(sec)	84(sec)	84(sec)
GLU	7	5320(43)	184(sec)	216(sec)	216(sec)
$\frac{GLUR}{GLU}$	4	1720(14)	6.4	3.1	3.1
$\frac{GLUR}{GLU}$	5	2700(22)	6.0	3.1	3.1
$\frac{GLUR}{GLU}$	6	3900(32)	11	3.5	3.6
$\frac{GLUR}{GLU}$	7	5320(43)	19	4.7	4.6

また、1つのプロセス(計算処理)にA(sec)のCPU時間がかかった時に、ウィンドウを2つ開き、同時に同じ計算を並行して行なったとする。この時、計算終了後の2つの計算プロセスのCPU時間が、それぞれC(sec)、D(sec)かかったとすれば、「C + D = 2A(sec)」が理想である。ワークステーションでは、この理想の結果になるが、特にPCにOSとしてWindowsを選んだ場合には、並行プロセスの多重処理がOSとして不完全であるので、「C + D > 2A(sec)」となってしまう。「C + D」と「2A」の時間を比べることによって、OSの多重処理の良し悪しを判断できることになる。

Table 2.3 CPU time(A→C+D) and multi Job ratio( $\frac{C+D}{A} : 2$ ) for each PC and Compiler

solver	N	Pen4XP+FjFort	Core2Vist+FjFort	Core2Vist+iFort
GLUR	3900	827→3200+3152(7.7)	297→410+401(2.7)	292→423+439(2.9)
GLU	3900	73→146+148(4.0)	84→171+170(4.0)	84→171+169(4.0)
BGLU	57720	9→26+26(5.8)	6→7+7(2.3)	5→6+6(2.4)
ICCG12	1305820	51→95+91(3.6)	50→89+88(3.5)	49→90+89(3.6)
BCGSTB13	1309950	167→296+299(3.6)	157→280+278(3.5)	154→264+263(3.4)

これまで測定したシミュレーション条件で、この多重処理の良し悪しをチェックしてみた。Table 2.3 では、「 $\frac{C+D}{A}$ 」を ( ) 内に示す。( ) 内の数値が「2」であれば、理想で、多重処理が悪ければ2を超える(「 $> 2$ 」)。概して、「Core2 Duo+Vista」は、「 $C+D \approx 3$ 」程度であり、「Pentium4+XP」よりも直接法においてはだいぶ改善されているのが見て取れる。反復法においては両OSで差はない。

これまで見たように、単体の反復法:ICCG12(対称用)、BCGSTB13(非対称用)に関しては、比較した2つのPCのハードとOSとFortranコンパイラに対してほぼ等しく変わりはない、という結論を得た。2つの計算を並行して行なってOSの多重処理の良し悪しを見ても、単体の反復法では、2つのPCでほぼ等しく変わりはないという結論を得た。そこで、1つの実務的な計算において、ここではn-MOSのデバイスシミュレーションを取り上げるが、2つのPCで差がでるのかどうか調べてみた。

### 3 n-MOSデバイスシミュレーションによる比較

MOSトランジスタの動作をシミュレーションするための標準的なモデルは、ドリフト拡散モデルと呼ばれ、電位 $\psi$ 、電子密度 $n$ 、正孔密度 $p$ についての連立系である(ポアソン方程式と $n, p$ の移流拡散方程式の連立系、ただし定常解用)。

$$(3.1) \quad \text{div}[-\epsilon \nabla \psi] = e(p - n + C), \quad C = -N_a + N_d$$

$$(3.2) \quad \text{div}[-D_n \nabla n + (\mu_n \nabla \psi)n] = GR \quad \mathbf{J}_n = -e[-D_n \nabla n + (\mu_n \nabla \psi)n]$$

$$(3.3) \quad \text{div}[-D_p \nabla p - (\mu_p \nabla \psi)p] = GR \quad \mathbf{J}_p = +e[-D_p \nabla p - (\mu_p \nabla \psi)p]$$

ここに、 $e$ は電荷定数、 $C$ はドーピング量で場所の既知関数、 $\mu_{n,p}$ は電子、正孔のモビリティで、場所と電位の関数であるが様々なモデルが提案されている。拡散係数 $D_{n,p}$ とは、Einsteinの関係:  $\mu/D = e/k_B T = \nu$  (Einstein定数)で結びつく。従来は、 $T = T_l$  (格子温度)としたが、ホットキャリア用の(HS)モデルでは、このEinstein定数がキャリア温度(電子 $T_n$ 、正孔 $T_p$ )の関数だとして、局所的な場でEinsteinの関係:

$$\nu_{n,p} = \frac{e}{k_B T_{n,p}}$$

が成り立つとする。キャリア生成結合項(右辺のGR項)は、ドレイン電圧 $V_D$ に対するドレイン電子電流 $I_d$ の特性に大きな影響を与え、様々なモデルがある。この方程式(3.1)(3.2)(3.3)の3本を、n-MOSの場に対して可変不均等あみ目上で、CV法により(元の積分形で)離散化し、計算可能な反復スキーム(ガンメル反復)を作る。その際、移流拡散方程式(3.2)(3.3)の離散化には、指数法(ベルヌーイ関数使用)が必須で、また、反復スキーム中のポアソン方程式(3.1)には、ガンメルの線形化と呼ばれる式の変形を行わないと解が収束しない。

全体の反復計算スキームは、3本の方程式を、 $\psi^{(k)}, n^{(k)}, p^{(k)}$ に対して、それぞれ逐次解いていく。全体の収束判定は、解いた $\psi^{(k+1)}, n^{(k+1)}, p^{(k+1)}$ をポアソン方程式に代入した残差で行い、収束判定値EPSGを $0.16 \cdot 10^{-4}$ として、その値以下になったら終了とする。これ

らの計算により、ドレイン電子電流値などの特性を得たいわけである。CV法の特質として、積分保存則を満足するように離散化するので、(計算した)電子電流 $J_n$ をドレイン電極に沿って積分して得たドレイン電流 $I_d$ は、基板正孔電流 $I_{sub}$ とソース電子電流 $I_s$ との和と比べ、0.0024%の差で一致する。ポアソン方程式(3.1)はICCG12(対称行列用)で解き、 $n,p$ の移流拡散方程式(3.2)(3.3)はBCGSTB13で解く。

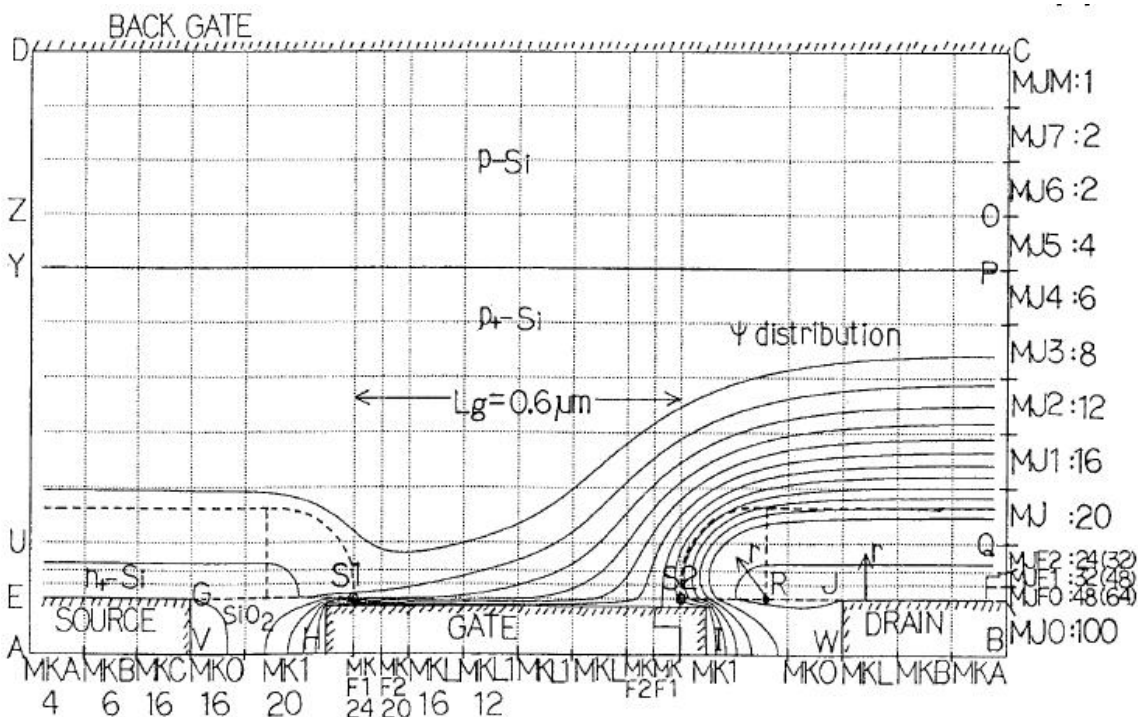


Fig.1 Simulation Field and Mesh step

実務上は、メモリや計算時間の節約から、n-MOS チャネル部分など大事な所を特に精しく他を粗くして不均等あみ目で計算を行なう。一番精しい所の1cm(Fig.1上の1cmは、実際のスケールでは $10^{-5}cm$ に当たる)を48分割した48系と、90分割した90系でシミュレーションした結果を示す。前の熱分布の場合：Table 2.1に対応する結果をTable 3.1に示す。

Table 3.1 CPU time(sec) for each PC and Fortran Compiler(n-MOS)

MJ1	N	Pen4XP+FjFort	Core2Vist+FjFort	Core2Vist+iFort(/Qpar)[/fast]
48	44457	24	14	12(11)[10]
90	112367	106	79	72(74)[94]

全体のガムル反復は、40系、90系とも約40回反復する。単体の反復法ICCG12、BCGSTB13とを組み合わせた、実際のn-MOSデバイスシミュレーションにおいては、Core2 Duo + VistaBusinessの方がPentium4 + WindowsXPに比べ、約3割程度CPU時間が速くなる(単体ではほぼ変わらなかった)。

同一2計算を並行処理にして、OSの違いによる多重処理について比較すると次のようになる。

Table 3.2 CPU time(A→C+D) and multi Job ratio( $\frac{C+D}{A} : 2$ ) for each PC and Compiler(n-MOS)

MJ1	N	Pen4XP+FjFort	Core2Vist+FjFort	Core2Vist+iFort
48	44457	24→35+34(2.9)	14→19+22(2.9)	12→14+22(3.0)
90	112367	106→162+163(3.1)	79→128+132(3.3)	72→127+132(3.6)

n-MOS デバイスシミュレーションでも、多重処理において OS の違いによる差は、ほぼない。単体でも反復法においては差がなかったので妥当な結果となる。

## 4 おわりに

結論として、n-MOS デバイスシミュレーションにとっては、Core2 Duo + VistaBusiness の方が Pentium4 + WindowsXP に比べ、いくぶん計算が速いという利点がある。

直接法等の単体シミュレーションを通して、PC のハード + OS + コンパイラのチェックを行なうと、Fortran では不利な GLUR で、Core2 Duo + VistBusines の利点があげられるが、総合的にみて大きな差はない。

最後に、参考として過去のデータを紹介する。デバイスシミュレーションで、1994 年に、元数 N=50141 元で、64MB メモリ、ISP2 付きの汎用機「HITAC M680d」と H(HP)712/100、192MB メモリ、クロック 100MHz の日立ワークステーションで比較を行なったところ、

Table 4.1 n-MOS simulation CPU time(sec) for M680d(IAP2) and H712/100

N	M680d+VOS3	H712/100+hFort
50141	2107	1964

これにより、M680d(IAP2) と H712/100 が、ほぼ同等の計算性能を持つことが分かる。

この 1994 年から、15 年間で富士通 Fortran コンパイラも最適化 (ワークステーション HP712/100 は日立 Fortran) してきた。PC のハードと OS とコンパイラによって、計算性能 (CPU 時間) がどのように推移してきたかをみた表を Table 4.2 に示す。元数は N=11118。

Table 4.2 n-MOS simulation CPU time(sec) for each PC (N=11118)

PC	OS	clock	FORver	CPU	ratio	memory
H712/100(WS)	hp-ux9.05	100MHz	V01-05-/C(H)	439	1.0	192MB
Pen2	Win98	400MHz	V2.0L10	131	0.3	128MB
Pen3	Win98	933MHz	V2.1L10	92	0.2	256MB
Pen4Mnote	Win2000	2.0GHz	V3.0L10	43	0.1	1GB
Pen4Mnote	Win2000	2.0GHz	V4.0L10	33	0.08	1GB
Pen4(DG3)	Win2000	2.66BGHz	V4.0L10	22	0.05	512MB
Pen4(DG5)	WinXP	3.20EGHz	V4.0L10	16	0.04	1GB
Pen4(HX1)	WinXPsp2	3.20GHz	V4.0L10	6	0.014	2GB
Coro2 DuoMnote	WinVista Busi	2.40GHz	V4.0L10	4	0.009	2GB

富士通 Fortran V4.0L10 は、内側の DO ループの最適化が良くなって、計算性能が上がった。コンパイラ V4.0L10 は、同じ Pentium4 ノート PC で、前のバージョン V3.0L10 より CPU 時間で約 3 割向上している。測定データは、n-MOS デバイスシミュレーションで、元数  $N=11118$  の場合である。HP700/100 のデータも比較のために載せている。Table 4.2 によれば、H712/100 に比べ、Pen4(HX1) 3.20GHz は、クロックが 32 倍になり、CPU は約  $\frac{1}{70}$  になっている。

また、いくつかの PC において、2 計算並行処理のデータを示す。計算時間が短いと不正確なので、元数を増やしてある。最近のデータは、前掲のデータを使った。

Table 4.3 CPU time(A→C+D) and multi Job ratio( $\frac{C+D}{A} : 2$ ) for each PC and Compiler recently(n-MOS)

PC	N	OS	clock	FORver	CPU
H712/100(WS)	11118	hp-ux9.05	100MHz	V01-05-/C(H)	439→441+441(2.0)
Pen4(DG3)	11118	Win2000	2.66BGHz	V4.0L10	22→45+28(3.3)
Pen4(DG5)	11118	WinXP	3.20EGHz	V4.0L10	16→21+20(2.6)
Pen4(HX1)	44457	WinXPsp2	3.20GHz	V4.0L10	24→35+34(2.9)
Coro2 DuoMnote	44457	WinVista Busi	2.40GHz	V4.0L10	14→19+22(2.9)

ワークステーション H712/100(Unix 系 : hp-ux9.05) に比べ、Windows 系 OS は、計算の多重処理が向上したとはいえ、まだ不満が残る。並列に処理を行なった場合には、Windows 系 OS では、単独計算 CPU 時間の約 1.5 倍多くかかってしまうことが分かる。

## 参考文献

- [1] 小国 力編著, 村田 健郎、三好 俊郎、ドンガラ J,J、長谷川 秀彦著、行列計算ソフトウェア (WS、スーパーコン、並列計算機), 丸善, pp.252-275, Nov.1991
- [2] 村田 健郎, CV 法と、手作り数値シミュレーションシステムの奨め, 日本計算工学会, 計算工学 vol.3, No.1, pp.44-53, 1998
- [3] 村田 健郎, CV 法と、手作り数値シミュレーションシステムの奨め (第 2 回)-非線形純拡散問題と割線反復法-, 日本計算工学会, 計算工学 vol.3, No.3
- [4] 村田 健郎, CV 法と、手作り数値シミュレーションシステムの奨め (第 3 回)-移流拡散系の離散化 : 特に指数法について-, 日本計算工学会, 計算工学 vol.3, No.4, pp.246-253, Dec.1998
- [5] 村田 健郎, CV 法と、手作り数値シミュレーションシステムの奨め (第 4 回)-連立非線形移流拡散系 : 半導体デバイス解析の場合-, 日本計算工学会, 計算工学 vol.4, No.2, 1999
- [6] 村田 健郎, 「BASIC 数学」連載 : 線形数値計算法とその応用, 現代数学社, 1992
- [7] 村田 健郎, 線形代数と線形計算法序節, サイエンス社, 1986
- [8] 小国 力著, MATLAB と利用の実際 [第 2 版], サイエンス社, 2001
- [9] 村田 健郎, 名取 亮, 唐木 幸比古著, 岩波書店, pp.56-137, 1990
- [10] 砂川重信著, 精講物理, 学生社, pp.342-344, Apr.1975
- [11] 松平升, 大槻義彦, 和田正信共著, 理工教養 物理学 I, 培風館, pp.103-105, Dec.1975