

## 並列計算機システム PPA を用いた 有限要素シミュレータの試作と評価†

土 肥 俊\*\*

本論文は、偏微分方程式解析の高速化を図るために 34 台のミニコンから成る一次元アレイ方式の並列計算機システム PPA (Parallel Processor Array) 上に試作した有限要素シミュレータの概要を紹介し、その性能評価について述べる。本シミュレータは熱伝導系を対象としているが、その入力データの形式は通常の有限要素解析システムにおける入力データの形式と本質的に変わらず、ユーザは並列演算ということをもっと意識する必要がない。性能評価では本シミュレータによる計算の速度が使用するプロセッサの台数にほぼ比例して速くなること、また十分な台数のプロセッサがあれば計算時間は問題の節点数にたかだか比例的に増加することを示す。さらに汎用大型計算機 HITAC M-200 H とその内蔵アレイプロセッサ IAP を用いた数種の計算法によるシミュレーションとの比較により、本シミュレータが計算速度の点で M-200 H と遜色がなく、また性能価格比ではそれをはるかに上まわっていることを示す。最後に有限要素シミュレーションに主眼をおいて PPA を拡張する場合、有限要素モデルの規模との関連でプロセッサ台数を実質的にどの程度まで増やせるか等について考察する。その結果節点数 20,000 点程度の問題を扱う場合、プロセッサ台数は 140 台くらいまで増やせること、またそのとき各プロセッサで必要なデータ領域のメモリ量は現在市販されているマイクロプロセッサでも十分実装可能な大きさであることなどが明らかになる。

### 1. ま え が き

偏微分方程式の数値解析は種々の科学技術計算において生じる重要な問題の一つであるが、一般に系が大規模になると莫大な計算時間がかかるため、計算の高速化が強く望まれている問題でもある。このような科学技術計算の高速化を目的としたいわゆるスーパーコンピュータは大別すると CRAY-1 に代表されるパイプライン方式のベクトル計算機と ILLIAC IV に代表されるアレイ方式の並列計算機があり、とくに前者は近年実用化の点で長足の進歩を遂げている。しかし一般に、パイプライン方式のみによる高速化には限界があり、より飛躍的な高速化のためにはアレイ方式を用いなければならないと考えられている<sup>1),2)</sup>。

偏微分方程式の数値解法の一つに有限要素法があるが、この手法は複雑かつ多様な形状・境界条件をもつ実際の系にきわめて近いモデルの解析を行うことができるなど、優れた特徴をもっている。このため有限要素法は近年多くの分野で用いられるようになってきた。

著者らは有限要素法による偏微分方程式の数値解析をアレイ方式の並列計算機システムを用いて高速に行うための並列計算手法を提案した<sup>8),9)</sup>。またその後熱

伝導系を例にとり、それらの手法を用いた有限要素シミュレータを 34 台のミニコンから成る 1 次元アレイ方式の並列計算機システム PPA (Parallel Processor Array)<sup>3),4)</sup> 上に試作し<sup>10),11)</sup>、さらにその性能評価を行ってきた。PPA は連立常微分方程式で表される連続系のシミュレーションの高速化を主目的として開発されたシステムであるため、有限要素シミュレーションに対して必ずしも理想的なシステムではない。しかしながら試作したシミュレータは汎用大型計算機 HITAC M-200 H によるシミュレーションと比較しても計算速度の点で遜色がなく、性能価格比では M-200 H をはるかに上まわっている。

本論文ではこの PPA 上に実現された有限要素シミュレータの概要を紹介し、その性能評価の結果を報告する。また有限要素シミュレーションに主眼をおいて PPA を拡張する場合、有限要素モデルの規模との関連でプロセッサ台数をどの程度まで増やすことができ、またその結果どのくらいの計算速度が得られるかを考察する。

### 2. 並列計算機システム PPA

PPA は北海道大学高速システムシミュレーション装置 HOSS の高速演算用サブシステムであり、連立常微分方程式で記述される連続系のシミュレーションを高速に行うことを主目的として開発された並列計算機システムである (図 1)<sup>3),4)</sup>。そのおもな特徴を以下

† Development and Performance Evaluation of a Finite Element Simulator Using the Parallel Processor Array (PPA) by SHUN DOI (Faculty of Engineering, Hokkaido University).

\*\* 北海道大学工学部精密工学科

\* 現在 日本電気(株)

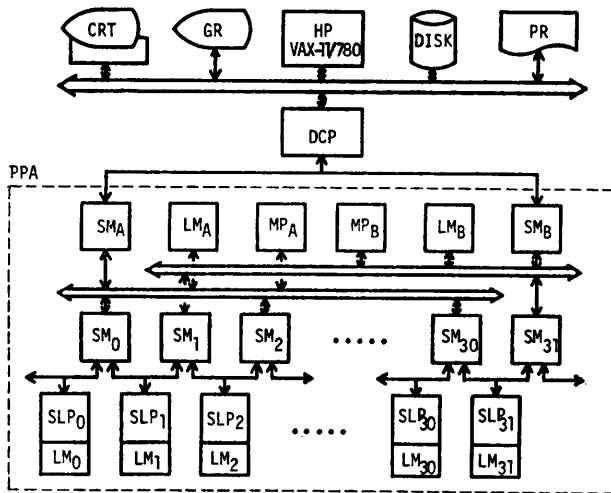


図1 並列計算機システム PPA のハードウェア構成  
Fig. 1 The hardware architecture of the Parallel Processor Array (PPA).

に示す。

(1) PPA は 32 台のスレーブ計算機 SLP とそれらを制御する 2 台のマスタ計算機 MP から成り、各計算機が独自にプログラムを実行する MIMD 型の並列計算機システムである。

(2) 32 台のスレーブ計算機 SLP<sub>i</sub> (i=0, ..., 31) は 1 次元の循環アレイを構成する。各 SLP<sub>i</sub> は共有メモリ SM<sub>i-1</sub>, SM<sub>i</sub> をアクセスすることができ、それによって隣接する 2 台のスレーブ計算機 SLP<sub>i-1</sub>, SLP<sub>i+1</sub> と直接交信する。

(3) マスタ計算機 MP-A, MP-B はそれぞれ偶数番, 奇数番の SM を自分のメモリとしてアクセスすることができ、さらにそれらを介して偶数番, 奇数番の SLP を制御する。マスタ計算機同士は専用のレジスタを介して交信する。

(4) 各 MP はそれぞれ支配下にある任意複数台の SLP に同時に起動をかけることができる。一方それら任意複数台の SLP すべてがあるレジスタに書き込みを行ったときそのことが MP に通知され (AND 機能), それと同時に書き込まれた内容の論理和が通知される (OR 機能)。また各 MP は支配下の全 SM に一斉に同一内容の情報を送るブロードキャスト機能をもつ。

(5) 各計算機 (DEC 社 PDP 11/34 相当) は 32 ビット浮動小数点データを高速に処理するハードウェアの高速浮動小数点演算機構 (FPP) とそれらを用いるための基本的な計算命令をもつ。

PPA に関する詳しい説明は文献 4) に述べられて

いる。

### 3. 有限要素モデル

本シミュレータが対象とする問題は 2 次元熱伝導方程式

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial \phi}{\partial y} \right) + Q$$

$$= \begin{cases} \mu \frac{\partial \phi}{\partial t} & (1) \\ 0 & (2) \end{cases}, (x, y) \in S$$

$\phi = \phi(x, y, t)$ : 温度,  $\mathbf{k} = [k_x, k_y]^T$ : 熱伝導率ベクトル,  $Q$ : 熱発生率,  $\mu$ : 熱容量

を, 境界条件

$$\phi(x, y, t) = \phi^*(x, y), (x, y) \in L_1 \quad (\text{固定境界}) \quad (3)$$

$$\mathbf{k}^T \frac{\partial \phi}{\partial \mathbf{n}} + q + \alpha \phi = 0, (x, y) \in L_2 \quad (\text{伝熱境界}) \quad (4)$$

$\mathbf{n} = [n_x, n_y]^T$ : 境界における法線ベクトル,  $q$ : 境界における熱流束,  $\alpha$ : 熱伝達率

および, 初期条件 (非定常熱伝導方程式 (1) の場合),

$$\phi(x, y, 0) = \phi_0(x, y) \quad (5)$$

の下で解く問題である。

領域  $S$  を分割する平面形状 (内部要素) として三角形を用い, その形状関数として一次関数を用いる。パラメータ  $k_x, k_y, Q, \mu$  は各内部要素内で, また  $q, \alpha$  は各境界要素 (境界  $L_2$  に含まれる内部要素の辺) 内でそれぞれ一様とし, 定数であるかまたは温度  $\phi$ , 時刻  $t$  いずれか一方の一次補間関数であるとする。温度  $\phi$  の一次補間関数は熱容量等の物性値が温度に依存する材料非線形問題を解析する場合に用いられる。また時刻  $t$  の一次補間関数は境界上を熱源が移動するというような時間依存型の問題において用いられる。

以上の問題を解くために式 (1), (4) に有限要素法を導入すると, 各内部要素および境界要素について, 要素節点方程式

$$K_e \phi_e + P_e \dot{\phi}_e - f_e = q_e \quad (6)$$

が得られる。  $K_e, P_e, f_e, q_e$  はそれぞれ要素熱伝導行列, 要素熱容量行列, 要素節点外部熱流束, 要素節点熱流束を表す。式 (6) を全要素について重ね合わせると, 全体節点方程式

$$K \phi + P \dot{\phi} - f = q \quad (7)$$

が得られる。  $K, P, f, q$  はそれぞれ全体熱伝導行列,

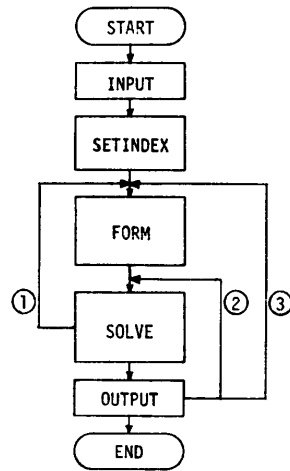


図2 有限要素解析の一般的な計算手順  
Fig. 2 A general flow of the finite element analysis.

全体熱容量行列, 全体節点外部熱流束, 全体節点熱流束である。また  $K, P$  をまとめてたんに全体行列と呼ぶ。

式(7)で, 式(3)により固定された境界節点に対応する  $\phi$  のベクトル要素は既知であり,  $q$  のベクトル要素は未知である。それ以外では  $\phi$  の要素は未知であり,  $q$  の要素は既知でかつ零である。式(7)は  $\phi$  の未知要素に関する連立常微分方程式になる。以下では説明の簡単化のために  $\phi$  のベクトル要素すべてが未知で, かつ  $q=0$  とする。

計算機による基本的な計算手順は線形定常問題, 非線形定常問題, 非定常問題それぞれ以下ようになる。

(1) 線形定常問題の場合,  $\dot{\phi}=0$  となり, 式(7)は連立一次方程式

$$K\phi=f \quad (8)$$

となる。処理の流れは図2で繰返し計算を行わない場合に相当する。FORM で  $K, f$  を計算し, SOLVE で連立一次方程式(8)を解く。全体行列  $K, P$  は一般に多くの零要素を含む。そこで実際にはこれらの零要素を除いた非零要素のみをベクトル状に圧縮して記憶し, メモリを節約する(圧縮行列形式, 4.3 節参照)。図2の SETINDEX はそのためのインデックスベクトルを計算する。

(2) 非線形定常問題の場合, すなわち式(2), (4)のパラメータが  $\phi$  の関数である場合,

$$\begin{aligned} K^{(v)}\phi^{(v+1)} &= f^{(v)} \\ K^{(v)} &= K(\phi^{(v)}), \quad f^{(v)} = f(\phi^{(v)}) \end{aligned} \quad (9)$$

によって定義される繰返し計算を図2のループ①で行い,  $\|\phi^{(v+1)} - \phi^{(v)}\| \leq \varepsilon$  となったとき収束したとみな

し, 計算を終了する。

(3) 非定常問題(非線形非定常問題を含む)の場合, まず時間微分を差分近似する。

$$\dot{\phi} = \frac{1}{\Delta t} \{\phi(t+\Delta t) - \phi(t)\} \quad (10)$$

$$\dot{\phi} = \frac{1}{2} \{\phi(t+\Delta t) + \phi(t)\}$$

これより全体節点方程式(7)は

$$\begin{aligned} A(t)\phi(t+\Delta t) &= g(t) \\ A(t) &= \left\{ P(t) + \frac{\Delta t}{2} K(t) \right\} \end{aligned} \quad (11)$$

$$g(t) = \left\{ P(t) - \frac{\Delta t}{2} K(t) \right\} \phi(t) + \Delta t \cdot f(t)$$

となり, 式(5)より得られる  $\phi(0)$  を初期値とする一連の連立一次方程式が得られる。すなわち, 指定された時間  $t$  ( $0 \leq t \leq t_{\text{end}}$ ) について一時刻ステップ前の解  $\phi(t)$  を用いて  $\phi(t+\Delta t)$  に関する連立一次方程式を解き, これを順次繰り返す(図2, ループ②)。パラメータが温度  $\phi$  または時刻  $t$  の関数であるとき, 各時刻ステップごとに FORM により  $K(t), P(t), f(t)$  を計算し直す(図2, ループ③)。

以上の説明からも明らかのように, 非線形解析, 非定常解析は線形定常解析を基礎とする繰返し計算を要し, 計算時間はその繰返し回数に伴って増大する。

本シミュレータでは図2の FORM, SOLVE に相当する計算を並列化し, 計算の高速化を図る。

## 4. 並列計算手法

### 4.1 並列計算の手順

基本的な並列計算の方法は文献(8)で提案したものと変わらない。すなわち, 入力データとなる有限要素モデルは前述の連立一次方程式(式(8), (9), (11))の係数行列が帯行列になるように節点番号付けされているものとする。ここで式(8), (9), (11)の連立一次方程式を改めて

$$A\phi=g \quad (A: N \times N) \quad (12)$$

とおく。このときこの  $N$  元連立一次方程式において係数行列  $A$  が帯半幅\*  $n_b$  の帯行列とする。また SLP 台数を一般的に  $m_p$  台とする。

まず式(12)の  $A, \phi, g$  を

$$m \leq \min(m_p, N/n_b) \quad (13)$$

を満たす  $m$  個の均等なブロックに分割する。すなわち,

\* 行列  $A=[a_{ij}]$  の帯半幅  $n_b$  を  

$$n_b = \max_{a_{ij} \neq 0} |i-j| + 1$$
 により定義する。

$$A = \begin{pmatrix} A_1 & L_1 & & & 0 \\ U_2 & A_2 & L_2 & & \\ & 0 & & \ddots & \\ & & & U_m & A_m \end{pmatrix}, \quad \phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_m \end{pmatrix}, \quad (14)$$

$$g = \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

ここで  $L_i$  は狭義下三角行列,  $U_i$  は狭義上三角行列である. 以下での説明のため  $D_i, E_i, F_i$  を  $A_i = D_i + E_i + F_i$  を満たす対角行列, 狭義下三角形行列, 狭義上三角行列とし, 各節点解ベクトル  $\phi_i (i=1, \dots, m)$  を

$$\phi_i = [\phi_{i1}, \dots, \phi_{in}]^T, \quad n = N/m$$

と表す. また  $m$  を並列度と呼ぶことにする.

このとき各節点解ベクトル  $\phi_i$  の計算をそれぞれ SLP<sub>*i*</sub> に割り当てる. また各内部要素および境界要素を, その要素を構成する節点が割り当てられている SLP に割り当てる. なおその要素を構成する節点が2台の SLP にまたがって割り当てられている場合は番号の若い SLP に割り当てる.

このような割当ての下で各 SLP は割り当てられた要素および節点について前章の手順で計算を行う. すなわち, 各 SLP は割り当てられた各要素の要素節点方程式(6)を計算し, それらを共有メモリ SM の一次元アレイ上に構成された全体節点方程式(7)に重ね合わせ, さらに割り当てられた節点について連立一次方程式(式(8), (9), (11))を次節に述べる並列 SOR 法により解く.

#### 4.2 並列 SOR 法

連立一次方程式(12)の分割(14)に対する並列 SOR 法は

$$\begin{aligned} \phi_i^{(k)} &= \phi_i^{(k-1)} + \omega D_i^{-1} r_i^{(k)} \\ r_i^{(k)} &= g_i - U_i \phi_{i-1}^{(k-1)} - (D_i + F_i) \phi_i^{(k-1)} - E_i \phi_{i+1}^{(k)} \\ &\quad - L_i \phi_{i+1}^{(k)} \end{aligned} \quad (i=1, \dots, m, k=1, 2, \dots) \quad (15)$$

により定義される<sup>8)</sup>.  $\phi_i^{(k)}$  は  $k$  回反復後の  $\phi_i$  の近似解であり,  $\phi_i^{(0)}$  はその初期値である.

並列 SOR 法は各 SLP<sub>*i*</sub> が他の SLP と同期をとりながら  $\phi_i^{(k)}$  のベクトル要素  $\phi_{i1}^{(k)}, \dots, \phi_{in}^{(k)}$  に対して逐次的に行う SOR 反復計算の過程を表している. PPA の共有メモリ方式では  $\phi_i$  を SM<sub>*i-1*</sub>, SM<sub>*i*</sub> 双方に割り当てることにより特別にデータ転送を行うことなく

式(15)を計算できる.

収束判定は, 各 SLP がそれぞれに割り当てられた節点について残差を用いて独自に行う一方, MP が2章(4)で述べた AND/OR 機能を用いて全体としての判定を行う. すなわち各 SLP は割り当てられた節点について収束条件が満たされない場合, AND/OR 機構のレジスタに零以外の書込みを行う. 収束条件を満足しない SLP が1台でもあれば, それがハードウェア的に MP に通知される. したがってその場合 MP は全 SLP に反復計算の継続を指示する.

式(12)で係数行列  $A$  がエルミート正定値行列でかつその対角要素がすべて正のとき, 式(15)の並列 SOR 法は  $0 < \omega < 2$  に対して収束する<sup>9)</sup>. したがって連立一次方程式(8), (9), (11)に対する並列 SOR 法は収束が保証される.

#### 4.3 圧縮行列形式

全体行列  $K, P$  および  $A$  は大きな記憶容量を必要とする. 有限要素法では二つの節点が内部要素または境界要素によって直接つながっているときに限り全体行列の対応する行列要素が非零となる. したがって全体行列そのものを実際に計算しなくとも, どの行列要素が非零であり以後の計算で必要であるかが容易にわかる. 本シミュレータでは各全体行列についてその非零要素のみをベクトル状に圧縮記憶する形式を用い, それによりメモリと計算時間の節約を図る. この形式を圧縮行列形式と呼び, その概略を式(12)の  $A$  を例にとり説明する.

係数行列  $A$  の帯状の部分行列  $[U_i A_i L_i] (i=1, \dots, m)$  の非零要素を行番号, 列番号の若い順にベクトル  $\alpha_i$  の要素に対応させる.  $\alpha_i$  の各ベクトル要素  $\alpha_{ij} (j=1, 2, \dots)$  を元の全体行列  $A$  に対応づけるために必要となるインデックスベクトルは有限要素モデルの節点間の接続関係を用いることによってホスト計算機上で前処理として計算され, モデルとともに各 SLP にロードされる. そのインデックスベクトルは次の二つの1次元配列である.

INDEX 1: SLP<sub>*i*</sub> の INDEX 1<sub>*i*</sub>( $k$ ) ( $k=1, \dots, n$ ) は  $[U_i A_i L_i]$  の第  $k$  行に含まれる非零要素数を表す.

INDEX 2: SLP<sub>*i*</sub> の INDEX 2<sub>*i*</sub>( $j$ ) ( $j=1, 2, \dots$ ) は  $\alpha_i$  のベクトル要素  $\alpha_{ij}$  の, 全体行列  $A$  における列番号を表す.

なおこの手法は疎マトリックス法<sup>6)</sup>の一種と見ることでもでき, 通常の計算機でも利用できる. 図2の流れ図では SETINDEX でこれらのインデックスを計算

する。

ここでこの圧縮行列形式を用いることによりどの程度メモリと計算時間が節約されるかを考える。

係数行列  $A$  の帯部分の行列要素数は約  $2n_b N$  である。一方、圧縮行列では有限要素モデルの要素分割が一樣な三角形メッシュの場合、 $A$  の各行の非零要素数が 7 (境界節点に対応する行では 7 以下) であり、全体の非零要素数は最大  $7N$  である。これよりインデックスベクトルのベクトル要素数は最大  $(1+7)N$  となり、これらの整数 1 データの大きさを実数 1 データの半分とすると圧縮行列形式では行列要素数にして最大  $11N$  個の領域を必要とする。なお要素分割が不均一な場合でも全体の非零要素数は上述の均一な場合と比べて大差ない。

節点数  $N$  と帯半幅  $n_b$  の比はモデルの要素分割のしかたに依存し一樣ではないが、次のように単純な形状からその一つの目安を得ることができる。

長方形領域を考え、これを  $n_1 \times n_2$  ( $n_1 \leq n_2$ ) のメッシュに分割する。このとき節点数  $N = n_1 \times n_2$  に対して帯半幅  $n_b = n_1 + 1 \approx n_1 \leq N^{1/2}$  となる。すなわち長方形領域を一樣なメッシュに分割する場合、帯半幅はたかだか節点数の平方根程度である。帯半幅はモデルの細部の形状の複雑さにはほとんど依存しない。したがって要素分割が極端に不均一な場合は別として、帯半幅  $n_b$  は  $N^{1/2}$  程度と考えることができる。

このとき係数行列  $A$  の帯部分のおおよその行列要素数  $2n_b N$  が  $2N^{3/2}$  となる。一方圧縮行列形式の場合に必要な領域の大きさは行列要素数にして最大  $11N$  であった。これより以下のことが結論できる。

(1) メモリ使用量は圧縮形式、帯形式それぞれ  $N, N^{3/2}$  に比例する。帯形式のメモリ使用量は  $N$  が大きくなるにつれて著しく増加する。実際にこれを 2, 3 の節点数について示したのが表 1 である。

表 1 圧縮行列形式と帯行列形式におけるメモリ使用量の比較

Table 1 A comparison of the required memory size for the proposed matrix form and for the usual band matrix form.

節 点 数	1,000	5,000	10,000
メモリ			
圧縮行列形式	11,000	55,000	110,000
使用量			
帯行列形式	64,000	710,000	2,000,000
比(圧縮形式/帯形式)	0.172	0.077	0.055

メモリ使用量の単位は実数データに換算したデータ個数である。

(2) 帯形式に必要なメモリ容量は 10,000 節点で倍精度にして 16M バイトとなり、現在の大型計算機のもつ実メモリと同程度の大きさになる。したがって帯形式の場合それ以上の節点数では連立一次方程式解析に 2 次記憶装置を用いなければならず、そのことが計算速度を著しく低下させる。

(3) SOR 反復計算の 1 反復ステップ当りの計算時間は係数行列の行列要素数にほぼ比例する。したがって帯形式の計算時間は  $N$  が大きくなるとメモリ使用量と同様に著しく増加する。圧縮形式の場合、配列要素は間接的に指示されるため 1 節点当りの計算時間が帯形式の場合に比べて長い。しかし実際のプログラムの比較からこれを 2 倍と見積もっても、圧縮形式の計算時間は節点数 1,000 程度で帯形式の場合の 1/5, 5,000 節点では 1/10 以下である。

## 5. 評 価

### 5.1 使用する SLP 台数と計算速度の関係

4.1 節で述べたアルゴリズムでは有限要素モデルが与えられたときそれを式(13)で示される上限以下の任意台数の SLP で解くことができる。実際のシミュレータでは各 SLP で扱える節点数メモリに容量によ

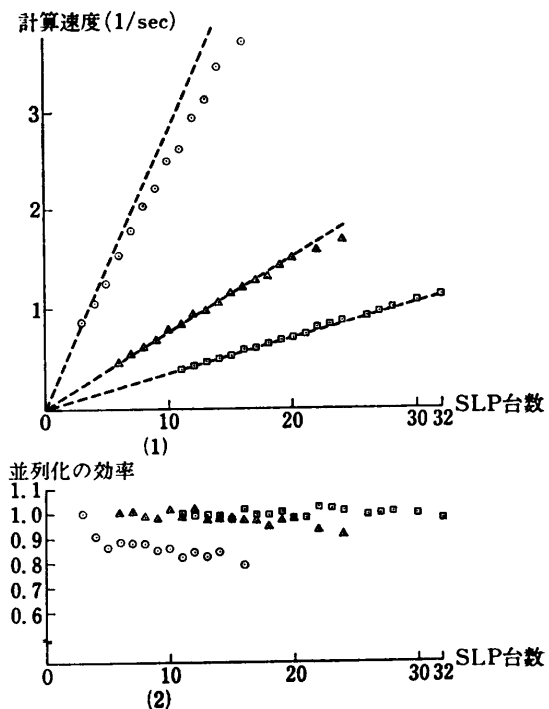


図 3 使用する SLP 台数と計算速度の関係  
Fig. 3 Relation between the number of using SLPs and the computational speed.

る制限があるため、使用できる SLP 台数には下限も存在する。この章では、使用する SLP 台数（並列度）を上述の下限と上限の間で変化させて同一の問題を解くことにより、並列度と計算速度の関係を調べる。

例題として付録(1), (2), (3)のモデルを用い、それぞれ使用する SLP 台数を3台から16台、6台から24台、11台から32台の間で変化させて各例題を解いたときの計算速度（計算時間の逆数）を図3(1)に示す。また図3(1)を正規化した結果を図3(2)に示す。図3(2)では最小の並列度  $n_0$  (それぞれ3, 6, 11)での計算速度  $v_0$  を基準にして、並列度  $n$  における計算速度  $v$  を  $(v/v_0)/(n/n_0)$  により正規化している。この値が1であればそれは SLP 台数に見合う高速化がなされていることを意味する。

図よりいずれの場合も並列度にほぼ比例して速くなり、SLP 台数に見合う高速化が得られていることがわかる。

4章のアルゴリズムでは隣接する SLP 以外のデータ交換が不要である。さらに試作したシミュレータは PPA の共有メモリ AND/OR 機構、ブロードキャスト機構等各機構を効果的に用いることにより、SLP を制御する MP の負荷を使用する SLP 台数に依存することなくほぼ一定にしている。数値実験の結果は PPA の各機構とそこで実現されたシミュレータとがよく適合していることを示す。

## 5.2 モデルの規模と計算速度の関係

ここではシステムが十分な台数の SLP をもっている場合を想定して、モデルの規模と計算速度の関係を調べる。すなわち、同一の偏微分方程式においてメッシュ分割の粗さのみを変化させることにより得られる数種の有限要素モデルをそれぞれ式(13)の  $N/n_b$  台の SLP を用いて解き、モデルの節点数と計算時間の関

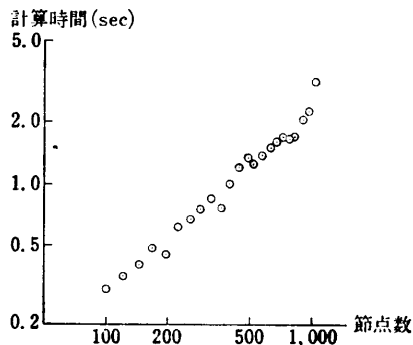


図4 有限要素モデルの規模と計算時間の関係  
Fig. 4 Relation between the scale of finite element problems and the computational time.

係を調べる。

実験の結果を図4に示す。例題には付録(4)においてメッシュ分割数  $n$  を10から32まで（節点数で100から1,024まで）変化させて得られるモデルを用い、各モデルをそれぞれ最大の並列度  $N/n_b$  (10から32)台の SLP を用いて解いた。なお SOR 反復計算はいずれの場合も残差の自乗平均が同一な一定値に達したとき打ち切った。

図より節点数と計算時間の関係はほぼ比例に近いことがわかる。すなわちシステムが十分な台数の SLP をもっていれば、計算時間は節点数にたかだか比例的に増加する。

筆者はほかにも多くの例題について同様の実験を行ったが、ほぼ同じ結果が得られた。

## 5.3 汎用大型計算機との計算速度の比較

本シミュレータの計算速度を北海道大学大型計算機センタの汎用大型計算機 HITAC M-200 H による以下の5通りの計算と比較する。

(1) 圧縮行列形式・点 SOR 法：係数行列は圧縮形式とし、連立一次方程式解法に点 SOR 法を用いる。計算の流れは図2で繰返し計算を行わない場合に相当する。

(2) 帯行列形式・点 SOR 法：圧縮形式と帯形式の比較を行うため、(1)のプログラムで行列形式を帯形式に変える。インデックスの計算が不要になる。

(3) IAP・帯行列形式・点 SOR 法：(2)のプログラムでより高速に計算を行うために、M-200 H の内蔵アレイプロセッサ IAP<sup>5)</sup> を用いる。

(4) 帯行列形式・ガウスの消去法：連立一次方程式解法として直接法のうちで最も一般的なガウスの消去法を用いる。

(5) IAP・帯行列形式・ガウスの消去法：(3)のプログラムを IAP を用いて実行する。

比較の結果を表2に示す。例題としては付録(5)の1,521節点の線形定常問題を用いる。前処理時間は圧縮形式の場合のインデックスベクトルの計算時間を表す。SOR 法の場合の打ち切り条件としては消去法による計算で求めた解の精度を用い、それにより SOR 法の計算結果で消去法と同程度の精度が得られるようにした。なおその結果 SOR 法による実際の反復回数は PPA, M-200 H いずれの場合も100回程度であった。表より次のことが結論できる。

(1) 表2の(1), (2)より、圧縮形式の計算時間は前処理を含めても帯形式の場合の1/6程度である。

表 2 試作したシミュレータと汎用大型計算機 HITAC M-200 H との計算時間の比較  
Table 2 A comparison of the computational time for the developed simulator and for the large scale computer HITAC M-200 H.

	計 算 方 法			計算時間 (sec)		解析時 間の比
	ハードウェア	行列 形式	連立一次方 程式解法	前処理	解 析	
(0)	PPA	圧縮	並列 SOR 法	2.38*	3.34	1.0
(1)	M-200 H	圧縮	点 SOR 法	0.22	1.73	0.52
(2)	M-200 H	帯	点 SOR 法	—	11.71	3.51
(3)	M-200 HIAP	帯	点 SOR 法	—	2.01	0.60
(4)	M-200 H	帯	消去法	—	4.66	1.40
(5)	M-200 HIAP	帯	消去法	—	1.44	0.43

\* VAX-11/780 による CPU 時間

この結果は 4.3 節(3)の推定によく合致する。

(2) 表の(1)の計算は SLP を 1 台にしたときの本シミュレータの計算と実質的に等価である。表の(0), (1)の解析時間を比較すると, PPA による計算時間は M-200 H の 2 倍までかからない。ハードウェアは PPA がはるかに安価であるが, その価格比を単純に CPU 部分の比として 1 対 10 程度とすると, 性能価格比は PPA による計算が M-200 H の 5 倍を超える。

(3) M-200 H による計算のなかで最も速かったのは(5)の IAP を使用した消去法であり, 本シミュレータはその 2 倍強の時間がかかる。しかし性能価格比は 5 倍近い。また実際に計算を始めてから解が出るまでの経過時間 (elapsed time) では本シミュレータのほうがむしろ速い。

(4) 本シミュレータではホスト計算機による前処理に解析時間と同程度の時間がかかっている。前処理の時間は有限要素モデルの空間節点数にのみ依存し, 非線形性の有無や定常・非定常の別には依存しない。非線形解析や非定常解析では繰返し計算を必要とするために線形定常解析に比べて解析時間が長い。したがって非線形解析や非定常解析では前処理時間の全解析時間に対する割合が線形定常解析に比べて小さくなる。

## 6. PPA に対する考察

5.1 節の数値実験では計算速度が SLP 台数にほぼ比例して速くなることを示した。そこでこの章では SLP 台数に主眼を置いて PPA を拡張しその性能の向上を図る場合, 有限要素モデルの規模との関連において SLP 台数を実際どの程度まで増やすことができ, 各 SLP にどのくらいのメモリが必要であるか,

またその結果どのくらいの計算速度が得られるかを考察する。

4 章のアルゴリズムでは, モデルに依存して使用できる SLP の台数 (並列度) に上限があり, それは節点数  $N$  と帯半幅  $n_b$  の比  $N/n_b$  で与えられる。4.3 節では長方形領域に対する簡単な考察から, 帯半幅  $n_b$  の一つの目安として  $N^{1/2}$  という値を得た。これを使うと並列度の上限  $N/n_b = N^{1/2}$  となる。すなわちシステムが対象とするモデルの最大節点数を  $N$  とすると, システム全体が有効に稼働できる最大の SLP 台数は  $N^{1/2}$  台程度となる。

4 章のアルゴリズムでは各 SLP のデータ領域の大きさによって各 SLP 当りの最大節点数が決まり, システム全体として扱えるモデルの規模が決まる。試作したシミュレータでは非線形・非定常を問わず 1 節点当りのメモリ使用量が約 150 バイト, 倍精度に拡張すると約 220 バイトになる。1 台の SLP に  $n$  節点を割り当てるとき必要なメモリ量は単精度, 倍精度それぞれ約  $150n$ ,  $220n$  バイトになる。

以上の関係をグラフにまとめると図 5 のようになる。図の縦軸, 横軸にそれぞれ節点数, SLP 台数をとり, それらを対数目盛で表している。いま, システムの SLP 台数を  $m_p$  とすると, 図の上では 3 本の直線となる

$$m \leq m_p \quad (16)$$

$$N \geq m^2 \quad (17)$$

$$N \leq N_1 m \quad (N_1: \text{SLP 1 台で扱える最大節点数}) \quad (18)$$

により囲まれた領域がそのシステムの扱えるモデルと SLP 台数の関係を表す。式(16), (17), (18)はそれぞれ, システムの SLP 台数による制限, モデルの帯半幅による制限, 各 SLP のデータ領域の大きさによる

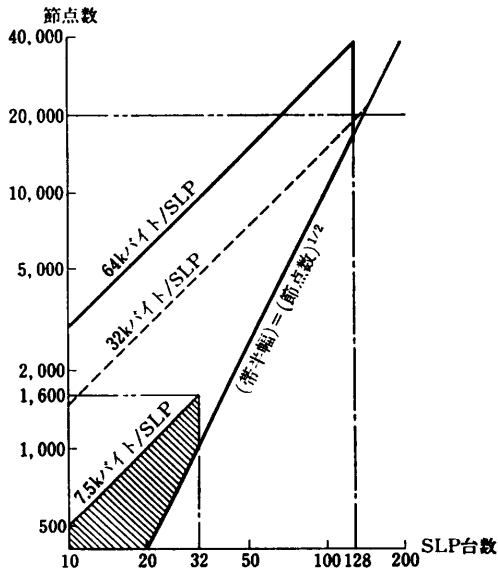


図5 システムの SLP 台数, メモリ容量とそのシステムで扱えるモデルの規模の関係

Fig. 5 Relation among the number of SLPs, the memory size of each SLP and the scale of the problems treated by the SLP system.

制限を表す。

たとえば, PPA 上に実現された試作シミュレータは SLP 台数  $m_p$  が 32 台, 各 SLP の扱える節点数  $N_1$  が 50 節点であり, 図の斜線部分が扱えるモデルと SLP 台数の関係を示す。図より最大 1,600 節点のモデルが扱え, そのとき使用する SLP 台数が 32 台であることがわかる。また 1,000 節点程度の問題は約 20 台から 32 台までの任意の台数で扱えることなどがわかる。なおこの SLP 台数の選択の幅が広いことはシステムのハードウェア障害に対するソフトウェアの融通性が高いことを意味する。

いま, システムが扱うモデルの最大節点数を 20,000 点とする。この値は現在ある大型の有限要素解析システムで扱える最大の節点数にほぼ一致する<sup>7)</sup>。このとき SLP 台数の上限が約 140 台であり, 各 SLP の必要最小限のデータ領域が 32 k バイトである。したがって実際には SLP 台数が 128 台, 各 SLP のデータ領域が 64 k バイト程度というシステム構成が妥当な線として考えられる。このシステムで扱える範囲は図の太線で示される。64 k バイトというメモリ容量は現在市販されているミニコンさらにはマイコンでも実装可能な大きさである。計算速度に関しては 5.2 節の結果から推定して, 現行の PPA における SLP を 128 台に拡張した場合, 20,000 節点程度の線形定常問

題を 1 分前後で解けることになる。

## 7. むすび

並列計算機システム PPA 上に試作した有限要素シミュレータの概要とその性能評価の結果を述べた。

試作したシミュレータは 2 次元の熱伝導系を対象とし, 要素は最も単純な三角形一次要素を用いた。本シミュレータで用いた基本的手法はこれらの場合に限られるものではなく, より広い分野のより複雑な要素にそのまま適用できる。

試作したシミュレータに入力される有限要素モデルの形式は通常の計算機における有限要素解析システムのモデルと本質的に変わらない。本シミュレータは入力された有限要素モデルの節点数と帯半幅の比, 使用可能な SLP 台数の関係から自動的に分割・割当てを行う。したがってユーザは並列演算ということをもまったく意識することなくシミュレーションを行うことができる。

PPA は既存の計算機をもとに構成されているため, ハードウェアの開発費が安くしかも信頼性は高い。有限要素解析が日常の業務になっているところでは PPA と同様の構成をもつ並列計算機システムを用いることで汎用計算機よりもはるかに安価な計算ができればよい。6 章の考察はそのシステムが対象とするモデルの規模との関連において妥当な SLP の台数とメモリ容量およびその結果得られる計算速度の目安を与えてくれる。

節点数 20,000 点程度のモデルに主眼をおくと SLP 台数は 140 台くらいまでが妥当な台数の範囲となる。そのとき各 SLP で必要なメモリ容量は市販のマイクロプロセッサでも十分実装可能な大きさである。

最近マイクロプロセッサの性能向上が著しい。また SLP 程度の機能をもつミニコンでも CPU が 1 チップ化される等, LSI 技術の進歩による大規模集積化が進み, これが価格の低下と同時に処理速度の向上をもたらしている。したがってこれらをもとに PPA と同様の構造をもつ並列計算機システムを構成すれば, 性能および性能価格比の点でさらに大幅な向上が得られよう。

本論文では例題として領域形状や境界条件が単純なモデルの線形定常解析のみを用いた。今後, 領域形状等がより複雑なモデルやそれらの非線形解析, 非定常解析についても評価を行い, 経験の蓄積を行ってゆく。



試作したシミュレータはさらに検討を加えるべき点をまだ多く残しているが、PPA による有限要素シミュレーションの有効性は十分立証されたといえる。

**謝辞** 本研究においてその動機付けと有益なご指導を与えてくださった故小山昭一先生、ならびにその後種々のご助言をくださった島公脩教授、汎用シミュレータ施設牧野圭二助手に心から感謝いたします。

### 参 考 文 献

- 1) 田中善一郎：科学技術用コンピュータにおける高速化手法，日経エレクトロニクス，No. 270，pp. 122-135 (1981)。
- 2) 田中善一郎：500 MFLOPS 級の商用コンピュータが稼働へ，日経エレクトロニクス，No. 314，pp. 108-126 (1983)。
- 3) 小山昭一他：パラレル・プロセッサ・アレイを用いた連続系シミュレータのアーキテクチャ，シミュレーション，Vol. 1，No. 1，pp. 42-49 (1981)。
- 4) 牧野圭二，三木信弘：PPA (パラレル・プロセッサ・アレイ)：汎用シミュレータ“HOSS”における並列演算サブシステム，信学技報，EC 82-27，pp. 57-68 (1982)。
- 5) 日立製作所：HITAC M-180/M-200 H，内蔵アレイプロセッサ解説書，HITAC マニュアル，8080-2-041。
- 6) ツィエンキーヴィッツ，O. C.：基礎工学におけるマトリックス有限要素法，培風館，東京 (1975)。
- 7) 田中 昇他：有限要素法による大型構造解析プログラムシステム (MISA)，三井造船技報，No. 93，pp. 1-39 (1976)。
- 8) 土肥 俊，小山昭一：有限要素法の並列処理手法，電子通信学会論文誌，Vol. J65-D，No. 4，pp. 464-470 (1982)。
- 9) 土肥 俊，小山昭一：連立一次方程式の並列処理向き反復解法の収束性と応用，情報処理学会論文誌，Vol. 23，No. 4，pp. 387-395 (1982)。
- 10) 土肥 俊：並列処理方式による有限要素解析システムの開発，情報処理学会第 25 回全国大会，

2L-2 (1982)。

- 11) 土肥 俊：並列計算機 PPA を用いた有限要素シミュレータ，第 3 回シミュレーション・テクノロジー・コンファレンス，pp. 43-46 (1983)。

### 付 録

例題(1) 偏微分方程式

$$\phi = \phi(x, y), \Delta\phi + 1.5 = 0 \quad (0 \leq x \leq 10, 0 \leq y \leq 20)$$

境界条件

$$\phi(x, 0) = \phi(0, y) = 0, \quad \frac{\partial\phi}{\partial x} \Big|_{x=10} = \frac{\partial\phi}{\partial y} \Big|_{y=20} = 0$$

要素分割：x 方向，y 方向それぞれ 8，17 の一様三角形メッシュに分割する。節点数は 136 点。

例題(2) 偏微分方程式，境界条件：例題(1)と同じ。要素分割：x 方向，y 方向それぞれ 12，25 の一様三角形メッシュに分割する。節点数は 300 点。

例題(3) 偏微分方程式，境界条件：例題(1)と同じ。要素分割：x 方向，y 方向それぞれ 16，33 の一様三角形メッシュに分割する。節点数は 528 点。

例題(4) 偏微分方程式

$$\phi = \phi(x, y), \Delta\phi + 1.5 = 0 \quad (0 \leq x, y \leq 10)$$

境界条件

$$\phi(x, 0) = \phi(0, y) = 0, \quad \frac{\partial\phi}{\partial x} \Big|_{x=10} = \frac{\partial\phi}{\partial y} \Big|_{y=10} = 0$$

要素分割：x 方向，y 方向ともに n の一様三角形メッシュに分割する。節点数は n<sup>2</sup> 点。

例題(5) 偏微分方程式：例題(4)と同じ。

境界条件

$$\phi(x, 0) = \phi(x, 10) = 0, \quad \frac{\partial\phi}{\partial x} \Big|_{x=0} = \frac{\partial\phi}{\partial x} \Big|_{x=10} = 0$$

要素分割：x 方向，y 方向ともに 39 の一様三角形メッシュに分割する。節点数は 1,521 点。

(昭和 58 年 5 月 13 日受付)

(昭和 58 年 10 月 11 日採録)