Vol. 26 No. 4 情報処理学会論文誌 July 1985

並列計算機 PAX による2次元弾性問題の有限要素解析†

佐藤善行†† 上村 健†† 星野 力†††

並列計算機 PAX を用いて，2次元弾性問題の有限要素解析を行った．与えられた有限要素モデルをプロセッサアレイ上へ直接投影し，各プロセッサは割り当てられた領域についての剛性マトリックス，変位，応力およびひずみを並列に計算する．これにより従来問題とされていた剛性マトリックスの帯状化のための節点番号付けを行う必要がなくなる．PAX でプログラムを実行し，実測した計算時間を解析し，節点数とプロセッサ台数の関数として表される計算効率を求めた．その結果，要素が各 PU へ均等に割り当てられる場合は，ほぼプロセッサ台数に比例した処理速度向上が得られることがわかった．

1. まえがき

科学技術計算は一般にきわめて膨大な演算時間と記憶容量を必要とするが，現存する計算機の能力には限界があり，ユーザは限られた環境の範囲内で規模を縮小して解析を行っているのが現状である．しかし近年，計算機性能のめざましい向上に伴い，科学技術計算は複雑化，大規模化の一途をたどっており，これがまた高速処理へのニーズを高めている．

計算処理を並列的に行う並列処理型計算機の研究，開発は古くから行われてきたが，最近 VLSI 技術の急速な進歩を土台としてさまざまな並列処理計算機のアーキテクチャの研究が進んでおり，並列処理計算機が将来の高速計算機の主流になるであろうといわれている．一方大規模な計算量を必要とする原子力，航空，気象などおもに偏微分方程式系モデルを計算する分野においては，有限差分法あるいは有限要素法による解析がおもに行われてきた．有限差分法の並列処理は，その高速処理に対する有効性が文献1),2)により明らかにされている．これは有限差分法が多くの同時処理可能な計算を含んでいるためであるが，同様のことが有限要素法についてもいえる．そこで高速処理を目的とした有限要素法の並列処理に関する研究もいくつか報告されている[3)~7)]．

土肥ら[3)]は一次元アレイ方式の並列処理システムを想定した並列処理アルゴリズムを提案している．これは帯状の列に分割された有限要素モデルを各プロセッサへ割り当て，逐次過剰緩和 (SOR) 法を用いた解析を行うものである．また Straasli ら[4)]は2次元アレイ方式の有限要素解析専用計算機を提案し，4台のプロセッサを用いて試作した．この報告では多色 SOR 法，共役勾配 (CG) 法を用いて2次元弾性問題の解析を行い，1台のプロセッサに対して約3倍の高速化を実現している．しかしプロセッサ台数が少なく，処理効率のプロセッサ台数依存性について検討していないので高並列計算機の有効性を実証するには至っていない．

本論文では128台のプロセッサを結合した並列計算機 PAX-128 で，CG 法を用いて2次元弾性問題の解析を行い，プロセッサ台数をパラメータとして含む処理効率の理論式を外挿することにより，高並列計算機が基本的な有限要素解析に対して有効であることが実証できたことを報告する．有限要素解析では2次元弾性問題は基本的なものであり，また実用上もこれに類似する問題は多く，この問題に対する並列計算機の有効性を実証することは大きな意義がある．ここでは有限要素モデルのプロセッサへの割り当てを，従来どおり2次元プロセッサアレイ上へ直接投影することにより行った．これにより自然現象のもつ近接作用は計算機システム上では，データ転送が近接するプロセッサ間に限られることを意味する．しかも従来問題とされていた剛性マトリクスの帯状化のための節点番号付けを行う必要がなくなる．

以下，2章では PAX-128 のシステム概要を示し，3章では有限要素法の並列処理手法を述べ，4章では並列処理効率を定義し PAX-128 の性能評価を行う．5章では大規模問題を解析できるような拡張された PAX システムへの性能外挿を行う．

2. 並列計算機 PAX[2),8),9)]

PAX は偏微分方程式で表される流体や場（連続体）

† The Finite Element Analysis of 2-dimensional Elastic Problem by Parallel Computer PAX by YOSHIYUKI SATO, TAKESHI KAMIMURA (Master's Program in Scientific Technology, University of Tsukuba) and TSUTOMU HOSHINO (Institute of Engineering Mechanics, University of Tsukuba).

†† 筑波大学理工学研究科
††† 筑波大学構造工学系

表 1 PAX-128 の基本性能
Table 1 Fundamental performance of PAX-128.

(a) プログラム言語 SPLM[8] による基本的な演算の所要時間

	演算	実行時間* (μs)
基本演算	$A(n)=1.$	63
	$A(n)=B(n)$	88
	$A(n)=B(n)+10.$	143
	$A(n)=B(n)+C(n)$	160
	$A(n)=B(n)*10.$	137
	$A(n)=B(n)*C(n)$	154
	$A(n)=\mathrm{SIN}(B(n))$	1,184
	$A(n)=B(n)*C(n)+D(n)$	241
	$A=B+C$	86
	$A=B*C$	84

	プロシジャ	機能	実行時間 (μs)
プロシジャ	SYNC	全 PU の同期	50
	BRDCST	1PU から全 PU へのデータ転送	160·L+311**
	CSUM	全 PU にわたっての和を求める	1,200

* 32ビット浮動小数点演算
** L は放送を行う浮動小数点数データの個数

(b) ハードウェアの速度

名称	マシンサイクル (ns)	演算実行時間 (μs) 加減算	乗算
マイクロプロセッサ (MC 68 B00)	500	2.5***	—
算術演算ユニット (Am 9511 A-4)	250	15*	37*
メモリ IC (16 kbit static RAM)	150**	—	—

* 32ビット浮動小数点演算
** メモリアクセス時間
*** 8ビット固定小数点演算

の数値シミュレーションを高速に行うことを目的として設計されている．このような連続体の系の著しい特徴は「近接作用」であり，空間のあらゆる点で近傍の点と作用を通じて物理量の更新が同時進行している．そこで多数のプロセッシングユニット (PU) を隣接結合し，各 PU が隣接の PU と通信しながら同時並列に計算処理することにより，全処理の高速化を狙ったのが PAX システムである．しかし連続体系のシミュレーション以外でも，PAX が有効に並列処理できる問題の存在が確かめられている[10)~12)]．PAX-128 のシステム概要を**図1**に，演算速度のハードウェア，ソフトウェア諸元を**表1**に示す．

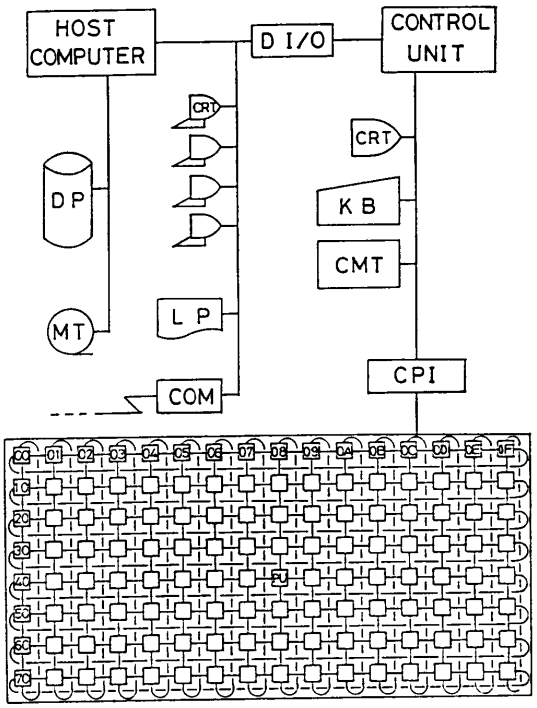
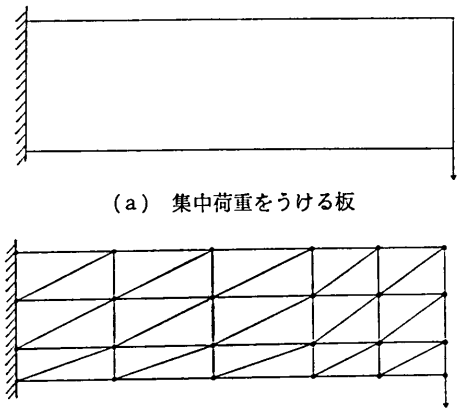


図 1 並列計算機システム PAX-128
Fig. 1 PAX-128 parallel computer system.
PU: Processing Unit, CPI: CU-PU Interface.



(a) 集中荷重をうける板

(b) 有限要素モデルへの離散化

図 2 弾性体の有限要素分割
Fig. 2 Finite element discretization of elastic plate.

3. 有限要素法の並列処理アルゴリズム

基本的な例題として，一端を固定し他端に荷重を加えた長方形弾性体の構造解析をとりあげる．これは構造解析の例として代表的なものである．**図2**に有限要素モデルへの離散化の一例を，**図3**にアルゴリズムを示す．以下に並列化の手法を述べる．

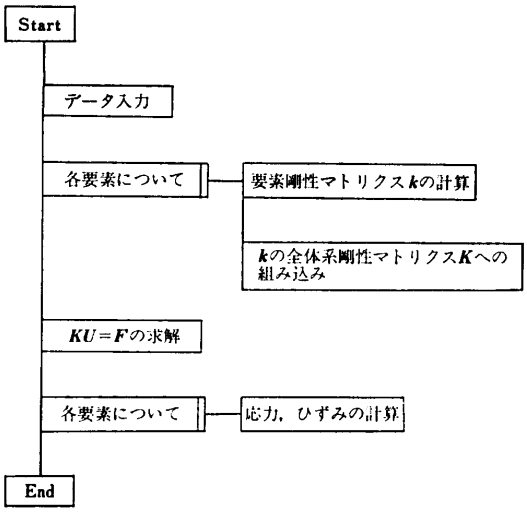


図 3 2次元弾性問題の有限要素解析アルゴリズム
Fig. 3 Algorithm of finite element analysis for 2-dimensional elastic problem.
U: 節点変位ベクトル，F: 節点荷重ベクトル

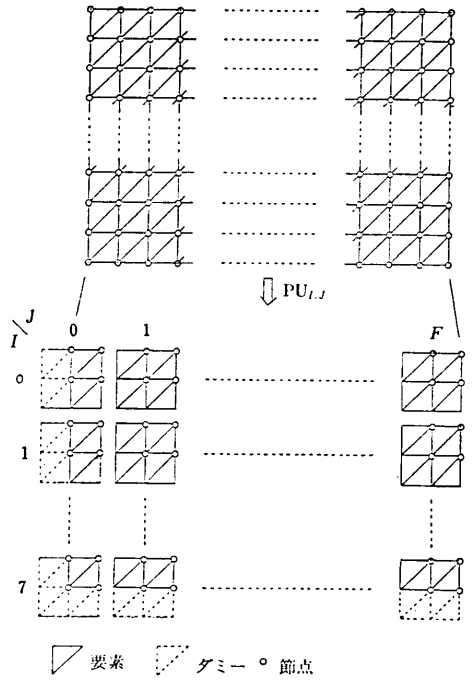


図 4 要素，節点の PU 割り当て
Fig. 4 Assignment of triangular elements and nodes to the PU array.

3.1 要素，節点の PU アレイへの割り当て

要素，節点の PU アレイへのマッピング例を**図4**に示す．このように全要素数，全節点数を PU 台数で割った数（割り切れるとする）だけの要素，節点を各 PU へ均等に割り当てる．このとき 1PU 当りの担当要素数は担当節点数の2倍になる．ただし PU

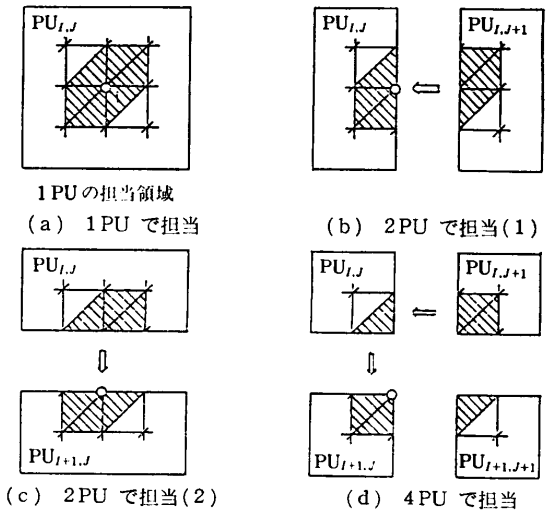


図 5 要素の PU 割り当て
Fig. 5 Assignment of elements to the PU array.
⇨: データ転送

アレイの0列目，7行目は担当要素数が他よりも少ない．ここで便宜上，N 個の節点に番号 i ($i=1,2,\cdots,N$) を付け，節点 i の x,y 方向の変位，荷重をそれぞれ (u_i,v_i), (f_i,g_i) とする．そして $2N$ 個の節点変位に関する $2N$ 元連立一次方程式 $\boldsymbol{KU}=\boldsymbol{F}$ において，$\boldsymbol{U}=(u_1,v_1,u_2,v_2,\cdots,u_N,v_N)^{\mathrm{T}}$，$\boldsymbol{F}=(f_1,g_1,f_2,g_2,\cdots,f_N,g_N)^{\mathrm{T}}$ とする．このとき $\boldsymbol{KU}=\boldsymbol{F}$ の PU アレイへのデータ割り当ては，節点 i を担当する PU が $\boldsymbol{K},\boldsymbol{U},\boldsymbol{F}$ の第 $(2i-1)$ 行，$(2i)$ 行成分を担当することにより行う．

3.2 全体系剛性マトリクス $\boldsymbol{K}$ の組み立て（処理1）

各 PU はまず全担当要素の要素剛性マトリクス $\boldsymbol{k}$ を逐次計算する．$\boldsymbol{K}$ の組み立ては，実際には担当節点に対応する $\boldsymbol{K}$ の行成分の非零成分を計算することにより行われる．この計算にはその節点を含む全要素に関する $\boldsymbol{k}$ が必要である．このとき必要な $\boldsymbol{k}$ がすべて同一 PU 内にある場合（**図5**(a)）は，ただちに非零成分を計算できる．しかし必要な $\boldsymbol{k}$ の一部が他の PU 内にある場合（図5(b),(c),(d)）は，PU 間のデータ転送（図中の→）が必要となる．このデータ転送は，図5(b),(c)では隣接 PU 間の通信用メモリ (CM) への1回の書き込みで，図5(d)では2回の書き込みで実現される．

3.3 節点変位求解の並列処理（処理2）

連立一次方程式の求解アルゴリズムには，ガウス消去法をはじめとする直接法と，CG 法，SOR 法等の反復法がある．ここで

① $\boldsymbol{K}$ は一般にかなり疎な規則正しい正定値対称

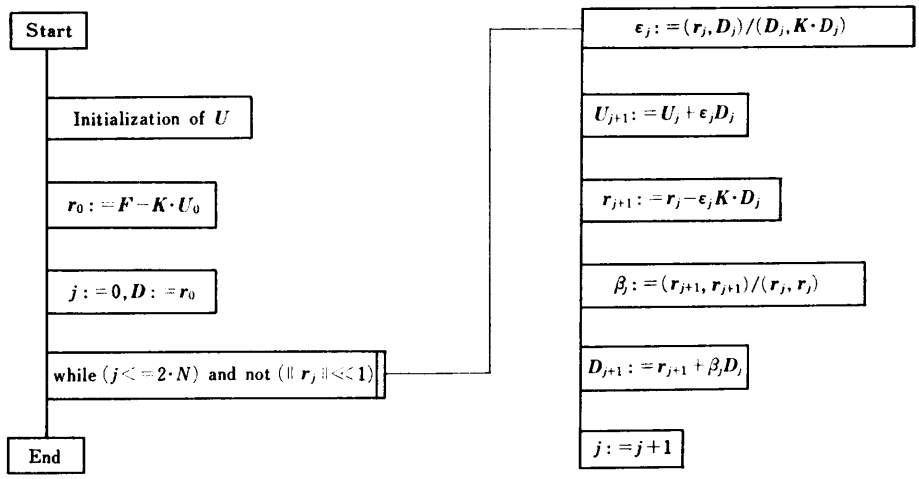
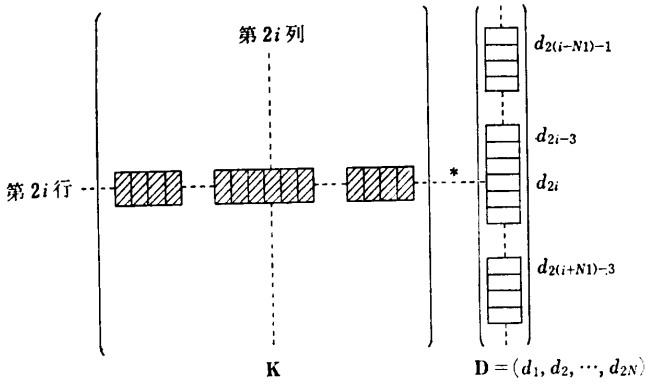
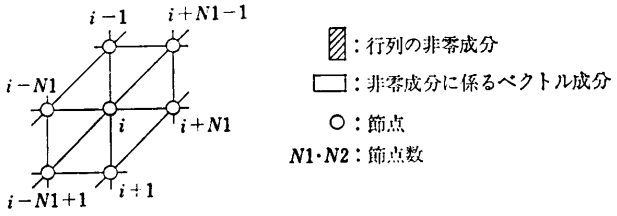


図 6 $2N$ 元連立一次方程式 $KU=F$ での共役勾配法のアルゴリズム
Fig. 6 Algorithm of Conjugate Gradient method for solving the system of equations $KU=F$.



(a) 行列とベクトルの積



(b) 物理空間 (=PU 空間)

図 7 ベクトル成分と節点の対応
Fig. 7 Correspondence between vector elements and nodes.

行列である.

② ガウス消去法は逐次処理的な演算が多く並列処理には向かない.

③ ガウス・ジョルダン法は並列処理に向くが, "fill-in" の問題が残る.

などの理由で K の疎性をいかす並列処理向きアルゴリズムとして反復法が有効となる. 本研究では多色 SOR 法より収束の速い CG 法を用いた. そのアルゴリズムを**図6**に示す. CG 法の演算の大部分は, 次の二つの演算によって占められている.

(1) ベクトルの内積

(2) 行列とベクトルの積

これらの演算の並列化を次のように行う.

3.3.1 ベクトルの内積

各 PU はまず担当する部分ベクトルについての内積を求める. そしてカスケード加算[11]によって全体での内積を求める.

3.3.2 行列とベクトルの積

節点 i に対応する K の行成分のうち, 第$(2i)$行成分の非零成分に係る列ベクトル成分を, 物理空間(これは PU 空間と対応している)で考えてみる. この列ベクトル成分に対応している節点は節点 i と直接結合されている(**図7**). よってこの演算には, 周囲よりベクトル成分をとり非零成分とかければよい. これは K の第$(2i-1)$行成分についても同様である. このときも節点の位置により, 図5に示したような PU 間のデータ転送が必要となる.

3.4 要素応力, ひずみ計算の並列処理 (処理3)

求められた節点変位をもとに, 各 PU は担当要素の応力, ひずみを計算する. このときも同様に図5に示したようなデータ転送が必要となる.

4. 有限要素法の並列処理効率

4.1 並列処理効率の定義

並列処理型計算機では, 逐次処理型計算機にとっては不必要な仕事や待ち時間が生じることがある. この仕事をオーバヘッドと呼ぶ. 具体的には PU 間のデー

タ転送，PU の待ち時間がある．並列処理効率 α を次のように定義する．

$$\alpha \equiv T_s/(P \cdot T_p) \quad (1)$$

ただし T_s は1台の PU による逐次処理時間，P は PU 台数，T_p は並列処理時間で $P \cdot T_p = T_n + T_0$，ただし T_n は全正味計算時間（各 PU の正味計算時間の和），T_0 は全オーバヘッド時間（各 PU のオーバヘッド時間の和）である．また本研究のように，T_n が T_s にほぼ等しいとき，α は次のように近似できる．

$$\alpha \fallingdotseq 1 - T_0/(P \cdot T_p) \quad (2)$$

4.2 有限要素法の並列処理時間と効率

プログラムの分析より，並列処理時間の理論式が節点数 $N_1 \cdot N_2$，PU 台数 $P_1 \cdot P_2$ $(=P)$ および CG 法の反復回数 h をパラメータとして得られる．並列処理時間は正味計算時間とオーバヘッド時間の和である．正味計算時間は前章で述べたとおり，各処理段階とも 1PU 当りの担当節点数 $N_1 \cdot N_2/(P_1 \cdot P_2)$ に比例する．オーバヘッド時間のうち PU の待ち時間は，各 PU ともほぼ同じ処理をするので無視できるほど小さい．PU 間のデータ転送時間のうち，図5のような $\boldsymbol{k}$（処理1），CG 法の修正方向ベクトル $\boldsymbol{D}$（処理2），$\boldsymbol{U}$（処理3）の転送時間は，1PU 当りの周囲節点数 $N_1/P_1 + N_2/P_2$ に比例する．また CG 法のカスケード加算に必要なデータ転送時間は PU アレイの周囲長 $P_1 + P_2 - 2$ に比例する．ただし節点変位求解の正味計算時間とオーバヘッド時間は h にも比例する．**表2**は並列処理時間中の支配的な項を各パラメータで表したものである．表2の各項を加えることにより並列処理時間，全オーバヘッド時間および並列処理効率の理論式が次のように得られる．

$$T_p = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \quad (3)$$

$$T_0 = (t_4 + t_5 + t_6) \cdot P \quad (4)$$

$$\alpha = 1 - T_0/(P \cdot T_p) = (t_1 + t_2 + t_3)/T_p \quad (5)$$

ここで $N_1/P_1 = N_2/P_2 = l$ として 1PU 当りの担当領域を1辺 l 節点の正方形領域とすると，T_p のうち $(t_4 + t_5 + t_6)$ は $(t_1 + t_2 + t_3)$ に対して無視できるので，α は

$$\alpha \fallingdotseq 1 - C_1 \cdot l/l^2 = 1 - C_1/l \quad (6)$$

（C_1 は定数）

で表される．

PAX で実際に有限要素解析の並列処理プログラムを実行し，さまざまな節点数に対する処理時間の実測を行った．これを式(3)を用いて解析した．その結果，式(3)中の係数を実測値より決定することができ，プログラムの処理時間と効率を N_1, N_2, P_1, P_2 および h の関数として表した「スケーリング則」を得ることができた．表2中の係数は PAX-128 の場合以下のとおりである．

$a_1 = 0.183$, $a_2 = 0.011$, $a_3 = 0.201$,
$b_1 = 0.008$, $b_2 = 0.102\text{E}-2$, $b_3 = 0.018\text{E}-2$,
$b_4 = 0.001$

表2 並列処理時間のスケーリング則（主要項のみ）
Table 2 Scaling law of parallel processing times (major terms only).

処理	処理時間の内訳	
	正味計算時間 (s)	オーバヘッド時間 (s)
$\boldsymbol{K}$ の組み立て	$t_1 \fallingdotseq a_1 \frac{N_1 \cdot N_2}{P_1 \cdot P_2}$	$t_4 \fallingdotseq b_1 \left(\frac{N_1}{P_1} + \frac{N_2}{P_2}\right)$
変位求解	$t_2 \fallingdotseq a_2 \frac{N_1 \cdot N_2}{P_1 \cdot P_2} h$	$t_5 \fallingdotseq \left\{b_2\left(\frac{N_1}{P_1} + \frac{N_2}{P_2}\right) + b_3(P_1 + P_2 - 2)\right\} h$
応力，ひずみ計算	$t_3 \fallingdotseq a_3 \frac{N_1 \cdot N_2}{P_1 \cdot P_2}$	$t_6 \fallingdotseq b_4 \left(\frac{N_1}{P_1} + \frac{N_2}{P_2}\right)$

$N_1 \cdot N_2$：節点数，$P_1 \cdot P_2$：PU 台数，h：CG 法の反復回数，$a_1, a_2, a_3, b_1, b_2, b_3, b_4$ は定数

表3 性能測定結果
Table 3 Measured performance of parallel processing.

	節点数（反復回数）		$\boldsymbol{K}$の組み立て	変位求解	応力，ひずみ計算	合計
実測値	8*16 (122)	処理時間 (s)	0.2254	2.5572	0.2073	2.9899
		効率(%)	92.9	66.2	81.4	69.3
	32*64 (400)	処理時間 (s)	3.0149	77.7190	3.2030	83.9369
		効率(%)	98.3	91.1	95.5	91.9
予測値	8*16 (122)	処理時間 (s)	0.2258	2.2057	0.2093	2.6408
		効率(%)	92.9	67.1	81.1	70.1
	32*64 (400)	処理時間 (s)	3.0238	76.1780	3.2400	82.4418
		効率(%)	97.9	92.6	95.1	92.9

4.3 測定結果

図2のような弾性体モデルの構造解析を行った．PU 台数は $P_1 = 8$, $P_2 = 16$ に固定し，節点数が $N_1 = 8$, $N_2 = 16$ のときと，$N_1 = 32$, $N_2 = 64$ のときの処理時間と効率の実測値，および式(3)，(5)より得られる予測値を**表3**に示す．これから，全体の計算量に対する変位計算の占める割合が大きく，変位計算での効

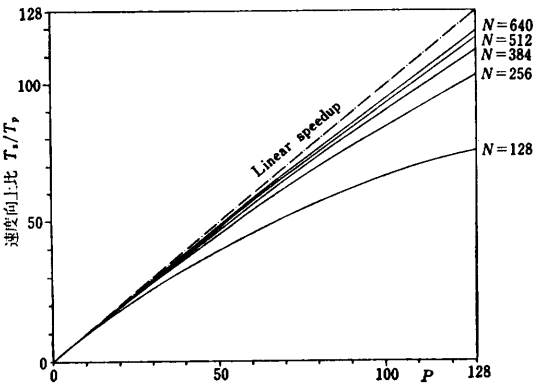


図 8 節点数が一定の場合の速度向上比
Fig. 8 Speedup ratios vs. number of PUs with constant number of nodes.
P: PU 台数, N: 節点数, T_s: 1台の PU での処理時間, T_p: 並列処理時間

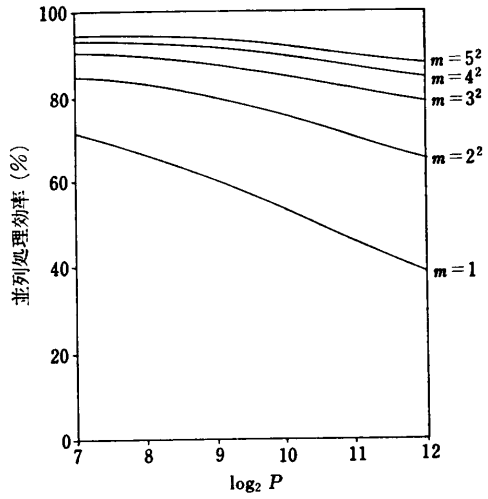


図 9 1PU 当りの節点数が一定の場合の並列処理効率の外挿
Fig. 9 Extrapolation of efficiency of parallel processing with constant number of nodes per PU.
P: PU 台数, m: 1PU 当りの節点数

率が全体の効率を大きく左右していることがわかる.

図8は式(3)を用いて, 節点数を固定したときの PU 台数と速度向上比の関係を表したものである. 理想的には PU 台数と速度向上比は比例する. これは図中の1点鎖線で表される. しかし実際にはオーバヘッドのために効率が低下し, 理想的な速度向上比を若干下回る. ただし PU 台数を固定したとき, 1PU 当りの担当節点数が多くなればその速度向上比は理想的な値に近づく. これは表2で示したように, 正味の計算量が担当領域の面積に比例するのに対して, オーバヘッドがその周囲長に比例するために相対的に効率がよくなるからである.

5. 並列処理効率の外挿

前章までで PAX-128 を用いた有限要素解析が, 1PU 当りの担当節点数を適当に多くすれば効率よく行えることを示した. しかし大規模問題が扱えるように PU 台数を多くした将来の PAX においても, 高い効率を達成するかどうかを検討しなければならない.

前章で求められた並列処理効率の理論式(5)を PU 台数 P の関数として扱い, PU が増設されたときの効率を予想する. その際次のような仮定を設ける.

① PU アレイは一辺が $\sqrt{P}$ の正方形アレイである.

② 節点数を N とし, 節点を一辺 $\sqrt{N}$ の格子状に配置する.

③ 1台の PU のハードウェア性能は PAX-128 のそれと同じである.

PU アレイへの節点割り当てを均等に行い, 1PU 当りの担当節点数 $m=N/P$ を一定として P を増加させたときの効率を**図9**に示す. このとき P が増加するにつれて効率が低下する. これは m が一定であるために正味の計算量と, オーバヘッドのうち節点変位等のデータ転送量が一定であるのに対して, カスケード加算に必要なデータ転送によるオーバヘッドが $\sqrt{P}$ に比例して増加するためである. しかし m が増加すればオーバヘッドは相対的に小さくなり効率の低下はゆるやかになる. 実用レベルの節点数 (たとえば $P=4,000$ に対して $N=80,000$) では m を大きくとるので, 90%以上の効率を保つことができ, PU を増設しても PAX は有限要素解析に有効であるといえる.

6. 結　　論

偏微分方程式のシミュレーション手法としてさまざまな分野に応用されている有限要素法をとりあげ, 並列計算機 PAX-128 を用いて2次元弾性問題の解析を行った. そして PAX の有限要素解析における性能評価を行うために, 節点数と PU 台数の関数として表される計算効率 (スケーリング則) を求めた.

このスケーリング則によれば PAX-128 では PU 台数が固定されるので, 要素が各 PU に均等に割り当てられる場合, 有限要素解析の並列処理効率 α は 1PU 当りの担当要素数が多いほど1に近づくことが

わかった.

またスケーリング則の PU 台数依存性，すなわち並列処理効率を PU 台数の関数としてみなすと，1PU 当りの担当要素数が多ければ，ほぼ PU 台数に比例した速度向上が得られることがわかった．たとえば将来の VLSI 技術により PU 1台分の演算速度が現在の100倍 (3 MFLOPS) に高速化されるとし，PU 台数を4,000台としたとき，節点数が80,000，要素数が160,000の問題の解析には，CG 法の反復回数を10,000とすると，約30秒を要するものと推定される．

本研究により，PAX システムが2次元弾性問題の有限要素解析において有効であることが示されたものといえよう．今後の課題としては，PU アレイへの要素，節点割り当てが不均等な場合のアルゴリズムの研究がある．

謝辞 本研究において本学構造工学系講師渡部修，白川友紀両氏にはご指導，ご協力いただきここに深く感謝いたします．

参 考 文 献

1) Hoshino, T. et al.: Highly Parallel Processor "PAX" for Wide Scientific Applications, International Conference on Parallel Processing IEEE, pp. 95-105 (1983).
2) Hoshino, T. et al.: PACS: A Parallel Microprocessor Array for Scientific Calculations, *ACM Trans. Comput. Syst.*, Vol. 1, No. 3, pp. 195-221 (1983).
3) 土肥 俊他：有限要素法の並列処理手法，電子通信学会論文誌，Vol. J 65-D, No. 4, pp. 464-470 (1982).
4) Straasli, O.O. et al.: *The Finite Element Machine: An Experiment in Parallel Processing*, *Research in Struct. & Solid Mechanics*, pp. 201-217, NASA Conf. Pub. 2245, Washington, D.C. (1982).
5) 小畑正貴他：ブロードキャストメモリ結合形並列計算機による行列計算の試み，計算機アーキテクチャ研究会資料，48-1 (1983).
6) 天野英晴他：一般疎行列専用計算機SM-square，計算機アーキテクチャ研究会資料，50-1 (1983).
7) 大澤 暁他：有限要素法専用の並列計算機のための立方格子状計算機とマッピングアルゴリズム，第28回情報処理学会全国大会講演論文集，pp. 113, 114 (1984).
8) 白川友紀他：並列計算機 PAX-128，電子通信学会論文誌，Vol. J 67-D, No. 8, pp. 853-860 (1984).
9) 影山隆久他：アレイ型並列計算機 PAX-128，第27回情報処理学会全国大会講演論文集，pp. 61, 62 (1983).
10) 真島澄子他：並列計算機 PACS-32 による第2種ボルテラ型積分方程式の処理，情報処理学会論文誌，Vol. 25, No. 3, pp. 499-505 (1984).
11) Hoshino, T. et al.: Processing of Molecular Dynamics Model by the Parallel Computer "PAX", *Comput. Phys. Commun.*, Vol. 31, No. 4, pp. 287-296 (1984).
12) 上村 健他：並列計算機 PACS による連立一次方程式の求解，第26回情報処理学会全国大会講演論文集，pp. 203, 204 (1983).

(昭和59年 5 月 14 日受付)
(昭和59年12月20日採録)