

熱伝導有限要素法解析プログラムの並列化条件*

原田 公一*¹, 磯部 俊夫*¹, 奥村 秀人*²

Massively Parallel Conditions of Heat Conduction Finite Element Analysis Program *

Koichi HARADA *¹, Toshio ISOBE *¹, Hidehito OKUMURA *²

ABSTRACT

This paper describes a heat conduction FEM (Finite Element Method) analysis program for an objective massively parallel computer system with processing elements to be arranged in a two-dimensional mesh based on the computer architecture presented by Harada et al..

Two conditions of parallel processing of the finite element analysis program were found to be effective for the parallel computer in the definite extent of FEM.

One was to set dummy elements outside the boundary of the analysis region. The other was to use a two-dimensional array (I, J) corresponding to the grid mesh position of processing elements for the numbering of elements and nodes.

Verification of the FEM program for parallel processing was carried out using the parallel computer "Paragon" with 16×21 nodes (processing elements).

Keywords: Massively parallel processing, Finite element method, Conjugate gradient method

概 要

原田らが提案している2次元的に演算器を配列した隣接結合型計算機を対象に、有限要素法による解析プログラムの超並列化の条件を求めた。

限られた範囲であるが有限要素法プログラムの並列処理において、2つの条件が極めて有効であることが分かった。

1つは解析領域の境界の外側にダミーの要素を配列させることであり、もう1つは格子点と演算器との対応づけを行い、それを2次元配列で番号づけして、全体マトリックスの非零成分のみを分散させることである。

本条件の妥当性を確認するために、熱伝導有限要素法解析プログラムを例題として、 16×21 台の演算器からなる超並列計算機 Paragon を用いて検証計算を行った。

1. はじめに

近年、電子計算機の高速化により科学技術分野における数値シミュレーション技術は目覚ましく進展している。また最近では並列計算機¹⁾により大規模な計算が行われ、複雑な物理現象の解明に本格的に歩み出してい

る。

しかし、数千～数万の超並列計算機が市販されるようになってきているが、実用的な使用はこれからと言える。

さらに、超並列計算機の差分法や有限要素法に代表される数値計算法による科学技術分野の数値シミュレーションへの適用にはまだ多くの研究課題がある。

* 平成9年9月10日受付 (received, 10 September 1997)

* 1 数理解析部 (Computational Sciences Division)

* 2 神奈川工科大学 (Kanagawa Institute of Technology)

本研究で対象としている科学技術分野の、特に航空宇宙関連の研究開発においては超音速機からスペースプレーン等の研究開発に超並列計算機を用いた数値シミュレーションが必須の状況になっている。これは全機モデルをターゲットとして熱流体構造の連成問題等の数値シミュレーションを行うことの重要性が高まっていることによる。しかし、数値解析方法すべてが並列処理により効率が向上するとは限らず、それぞれの解析方法について評価し、その技術を蓄積していくことが数値シミュレーションに限らず、今後の超並列計算機の開発においても重要なことになっている。

このようなことから本研究では原田ら²⁾の提案している演算器を2次的に配列した隣接結合型並列計算機を対象として、有限要素法による解析プログラムの超並列化についての研究を行うことを目的としている。

その第一段階として、超並列計算機として隣接結合型並列計算機、有限要素法解析プログラムとして二次元熱伝導有限要素法解析プログラム³⁾を基本とする範囲に限定することにより、超並列処理技術の基礎的研究を行

う。ここでは二次元熱伝導有限要素法解析プログラムをもとに要素分割、要素番号および節点番号のインデックスの付け方に差分法的な手法を取り入れ、要素に関するマトリックスの生成部分と全体マトリックスの合成、その連立方程式の解法である共役傾斜法の部分について、隣接結合型並列計算機の並列アルゴリズムを適用した。

さらに、この並列アルゴリズムにより開発したプログラムの妥当性を超並列計算機Paragon(米国インテル社製)で検証を行い、限定された範囲ではあるが有限要素法解析プログラムの並列化の条件を見いだすことができたので報告する。

2. 隣接結合型並列計算機のイメージ

図1にここで扱う隣接結合型並列計算機の構成を示す。これはインストラクションメモリ部、制御処理部、データ処理部、入出力処理部、スカラーデータメモリ部およびアレイデータメモリ部より成る。

また、アレイデータメモリ部はアレイデータメモリコントロールユニットと 128×128 個のメモリ要素から構

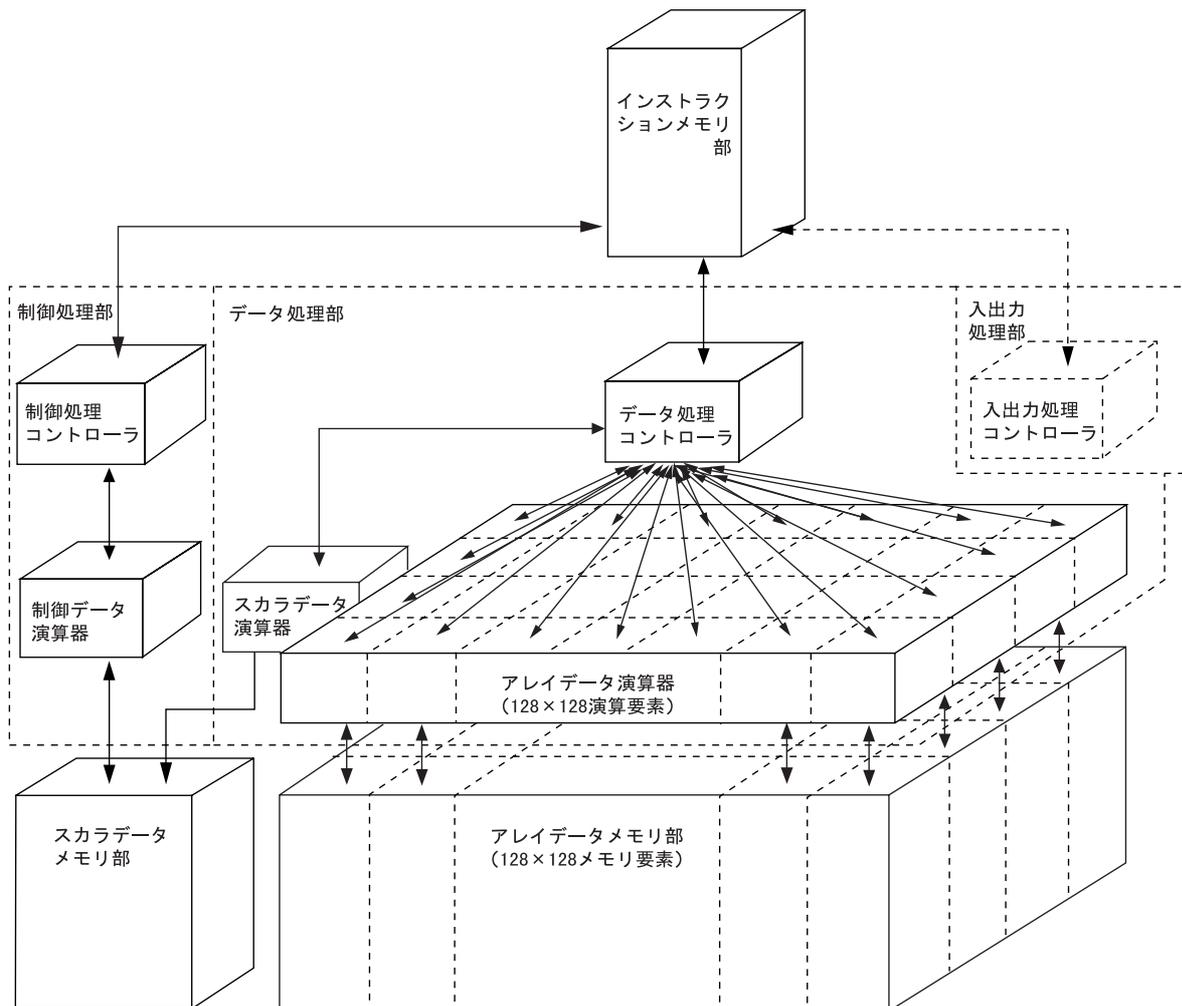


図1 並列計算機の構成

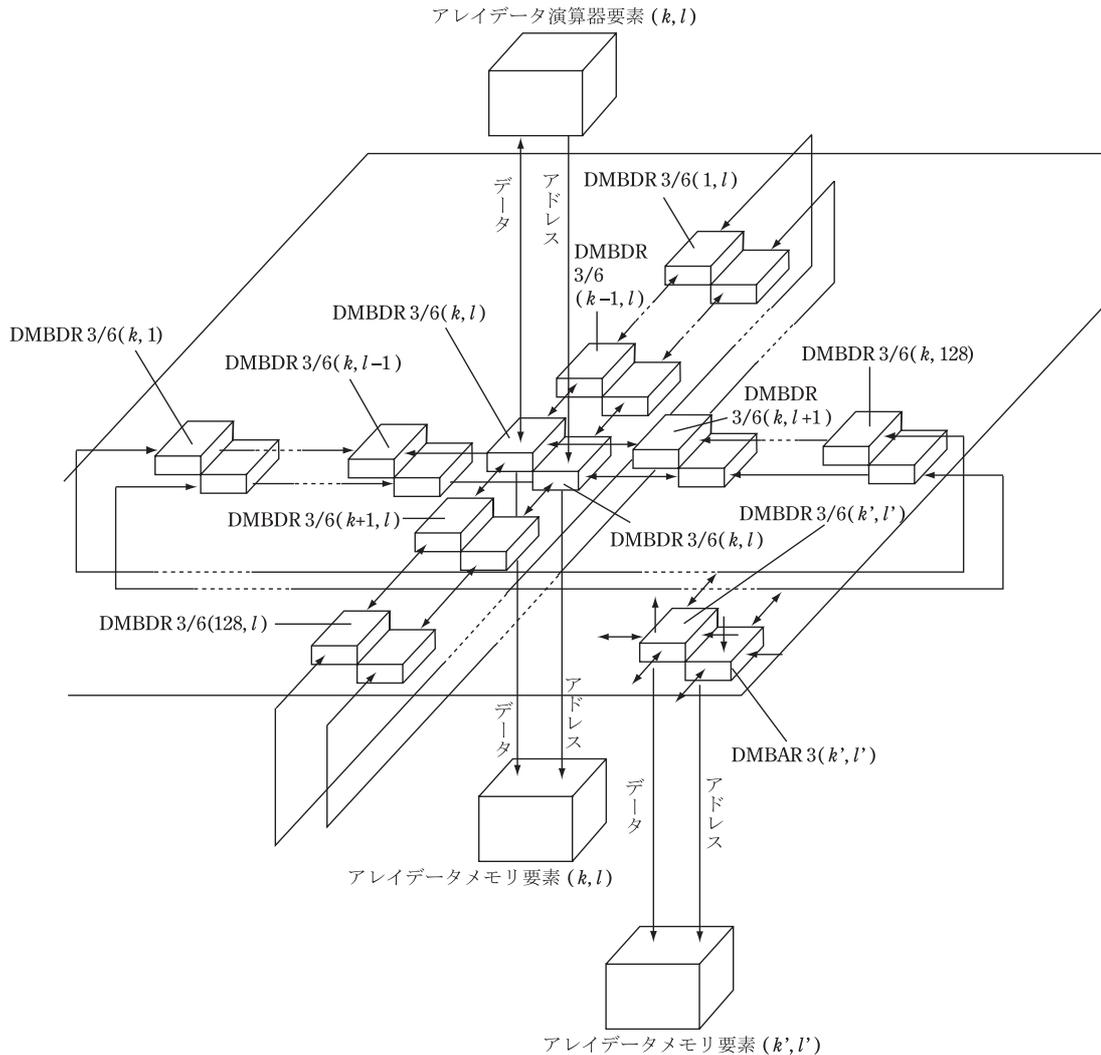


図2 アレイデータメモリコントロールユニットの構成

成されており、各メモリ要素は1語64ビットを想定した記憶装置である。

さらに、図2に示すようにこのアレイデータメモリコントロールユニットはアレイデータ演算器要素とアレイデータメモリ要素とを対応つける結合網である。ここでは、 128×128 対のデータおよびアドレスを保持するレジスタ (DMBDR3 (K, L) および DMBAR3 (K, L)) を行列各々隣接結合による上下左右のシフトレジスタで終端がリング結合しているトラス結合網を構成する。

3. 二次元非定常熱伝導解析プログラム THERM2

はじめに二次元問題を対象とした有限要素法解析プログラムの並列化を研究することにした。そこでは、計算が大規模になり、かつ全体マトリックスあるいは全体ベクトルを繰り返し再計算する動的応答問題、弾塑性問題および非定常熱伝導問題等の多くの問題で並列化の研究対象となる。ここでは非定常熱伝導問題の解析プログラム THERM2 を選択した。その理由は、ソースプログラ

ムが公開されていること、1節点あたりの自由度が最小の1であり、取り扱いが容易であることおよび航空宇宙分野で熱流体構造の連成問題の数値シミュレーションが急務であることによる。

このプログラムはモジュール化するのが容易なため、熱流体解析プログラムに組み込み、熱流体構造熱伝導連成問題の解析に適用され、その有効性が実証されている³⁾。しかし、実用的な問題では、さらに大規模な数値シミュレーションが必要となり、超並列計算機で実行することおよび有限要素法解析プログラムの超並列化が重要となっている。

プログラム THERM2 は図3に示すような構成になっている。またそれぞれのサブルーチンは表1の機能を有している。プログラムでは図4の4節点四角形アイソパラメトリック要素を使用し、連立方程式の解法は共役傾斜法を用いている。また時間積分法にはクランクニコルソン法を適用している。

プログラムサイズは約0.26MB (約2800ステップ)で

表 1 プログラム THERM2 の主要なサブルーチンの機能

CLTIME	スタート時刻, 終了時刻の表示。
CIMAGE	(NREAD) ファイルからの入力データをカードイメージで表示する。
PARMIN	コントロールパラメタの入力及び設定。
DATAIN	要素データ, 節点データ, 物性値データ及び境界条件データの入力と ガウス積分に関するデータの設定。
CINDEX	全体マトリックスの列方向のサイズの計算。
LINKID	LINDX, LGOAL 等のインデックスの作成。
TIMEDT	タイムステップの長さの入力, リスタート時の初期温度データの入力。
GHMATX	熱伝導マトリックス [K] と熱容量マトリックス [C] の計算。 及び拘束条件による見かけの熱荷重ベクトルの計算。
GHSURF	熱伝達境界 (S3) に関するマトリックスとベクトルの計算。
GQVECT	内部発熱及び熱流束規定境界に関する熱荷重ベクトルの計算。
QFLOAD	全体熱荷重ベクトルの作成。
CGSOLV	$[A]\{X\} = \{B\}$ の連立方程式を共役傾斜法により解く。
PRNOUT	温度の計算結果を出力する。
RESTAR	リスタート計算のために (NFILR) に最終ステップの温度分布をストア する。

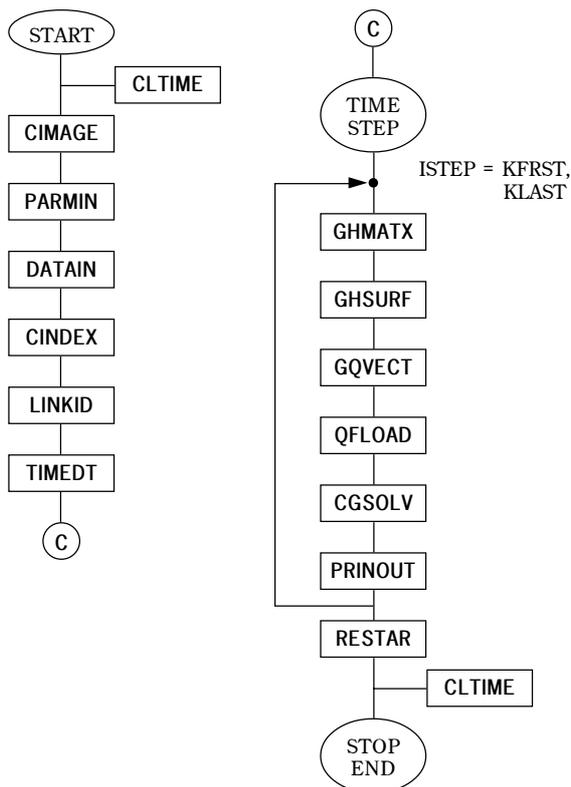


図 3 プログラム THERM2 のメインルーチンにおけるフローチャート

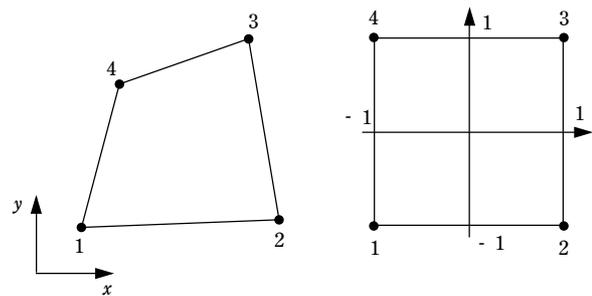


図 4 4 節点アイソパラメトリック要素

ある。また 3 カ所変更するのみで、パソコンからスーパーコンピュータまでどのコンピュータでも稼働することが可能である。さらに時間ステップについて途中から計算を再開できるリスタート機能を有している。

つぎに非線形問題を扱うことができるように物性値および熱特性に関する特性値の温度依存性あるいは時間依存性を任意にサブルーチンで定義できるようになっている。

4. 並列処理

プログラム THERM2 をそのまま並列計算機に投入しても逐次処理用になっているので高い性能はでない。そのため並列アルゴリズムを取り入れたプログラムに変更する。これを並列化プログラムと呼ぶことにする。

ここでは、解析領域を四角形の要素に分割する非定常熱伝導有限要素法の並列処理を示す。ただし、要素に関しては四角形要素を取り扱っているが、四角形要素を 2 つの三角形に分割することで三角形要素を用いたプログラムにも適応することが可能である。

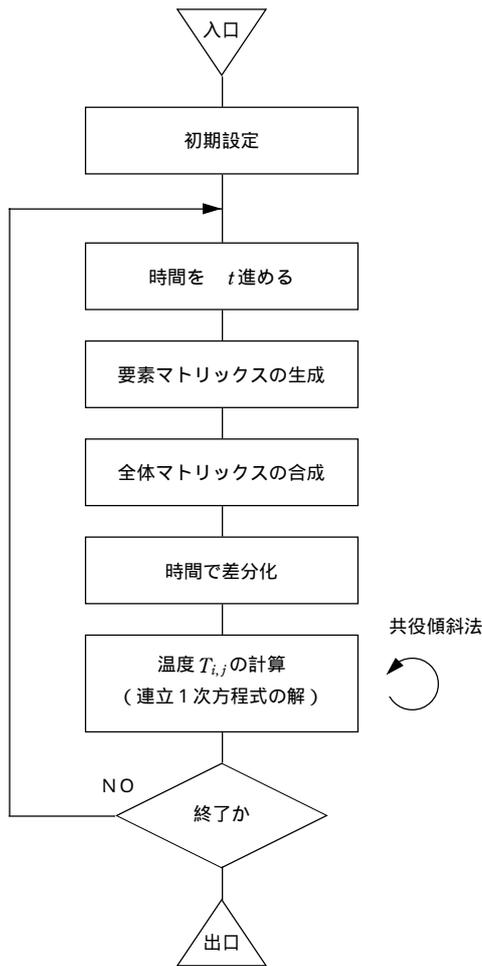


図5 処理手続

非定常の有限要素法解析の場合、図5に示すように要素マトリックスの生成、全体マトリックスの合成、連立方程式の解法の順に計算され、それらを繰り返す。ここで、並列処理のアルゴリズムを適用するのは要素マトリックスの生成部分、全体マトリックスの合成部分、そして連立方程式の解法の部分である。

(1) 図6のように解析領域A B C Dの周囲の境界に沿って、点線の四角形要素をダミーとして設定することを基本とし、空白の存在する場合には、その部分もダミー要素で埋めることにより、総数 $M \times N$ 個の四

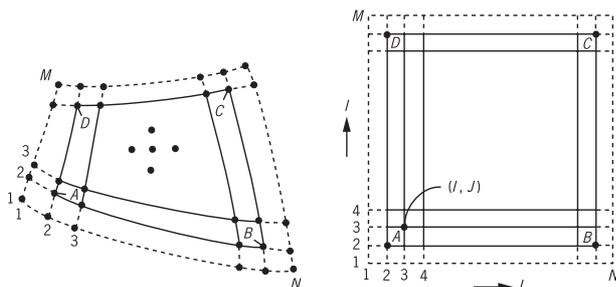


図6 有限要素分割とメモリの対応

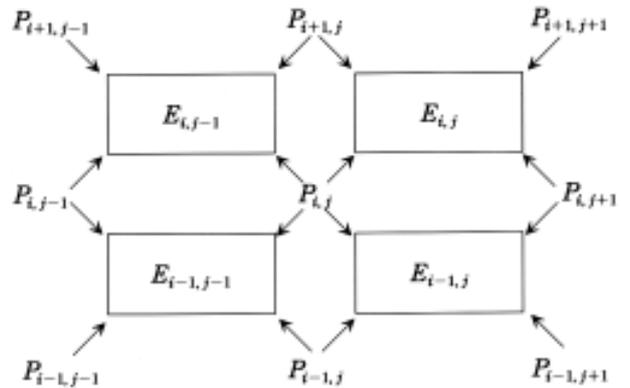


図7 要素マトリックスの生成

角形要素に分割する。ただし、ダミーの四角形要素に関する要素マトリックスの成分はすべて零とする。

- (2) 通常の有限要素法解析では、要素番号および節点番号は、それぞれ1つの整数型のインデックスによって示すが、本研究の並列処理アルゴリズムでは、図6に示すように2つのインデックスを用いて表示する。つまり (I, J) に対応するメモリには、要素番号 $(I - 1, J - 1)$ に関する要素マトリックスのデータを格納する(図7)。ここで $P_{i,j}$ は要素演算器・メモリを、 $E_{i,j}$ は要素マトリックスを示す。
- (3) また全体のマトリックスの合成では、通常計算された成分の値は、1つの配列に格納されるが、並列処理のアルゴリズムでは、メモリ要素 (I, J) には全体マトリックスの成分中で節点 (I, J) とその周囲の8個の節点についてのデータを格納させる。したがって本方法では、全体マトリックスを分解して格納することになる。
- (4) また連立方程式の部分における並列処理については、基本となった非定常熱伝導プログラムと並列処理を適応した並列化プログラムにおいて共に共役傾斜法を使用しており、変更点はなかった。具体的には共役傾斜法の演算で、マトリックスとベクトルの積においてマトリックス成分のうち非零の成分のみを計算するようにした(図8)。

4.1 問題の表現

T を温度、 N を重み関数(ここでは N として内挿関数を用いる)、 t を時間、および $x, y, c,$ をそれぞれ x, y 方向の熱伝導率、比熱、密度として熱伝導方程式

$$\frac{\partial}{\partial x} \left(\lambda_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial T}{\partial y} \right) - \rho c \frac{\partial T}{\partial t} = 0$$

に重み付き残差法を適用して

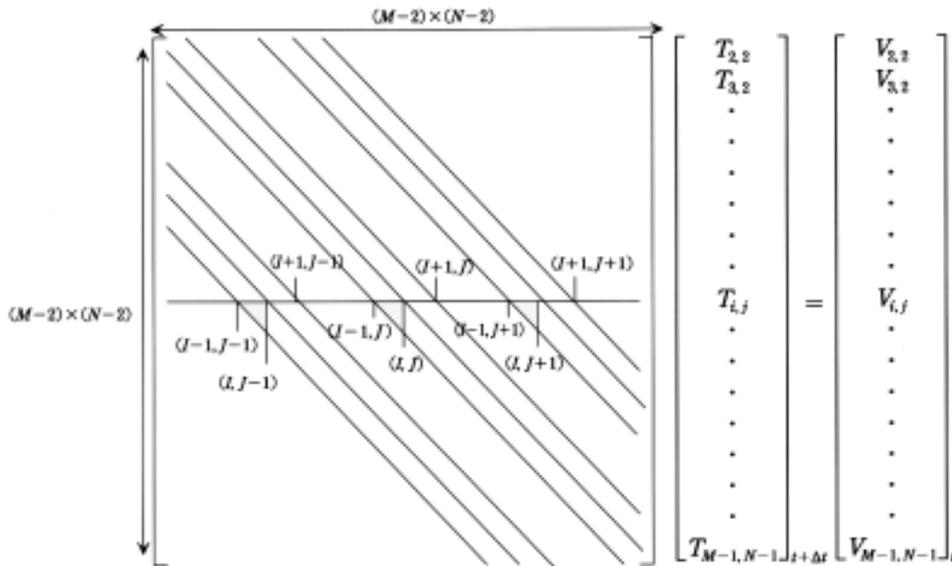


図8 全体マトリックスにおける非零成分の位置

$$\int_{V^e} \left\{ \frac{\partial}{\partial x} \left(\lambda_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial T}{\partial y} \right) - c_p \frac{\partial T}{\partial t} \right\} N dV^e = 0$$

を得る。ただし、ここでは重み関数 N は境界条件を満たし、部分積分可能な関数とする。これをグリーンの定理を用いて整理すると

$$\begin{aligned} \int_{V^e} c_p \frac{\partial T}{\partial t} N dV^e + \int_{V^e} \left(\lambda_x \frac{\partial T}{\partial x} \frac{\partial N}{\partial x} + \lambda_y \frac{\partial T}{\partial y} \frac{\partial N}{\partial y} \right) dV^e \\ = \int_{S^e} \left(\lambda_x \frac{\partial T}{\partial x} l_x + \lambda_y \frac{\partial T}{\partial y} l_y \right) N dS^e \end{aligned} \quad (4.1)$$

となる。図9のように上面、下面を断熱壁とする。 Q を熱流量、 α を熱伝導率として式(4.1)の右辺は

$$\int_{S_{top}} -\alpha(T - T_\infty) N dS + \int_{S_{bot}} -Q N dS \quad (4.2)$$

となり、整理すると

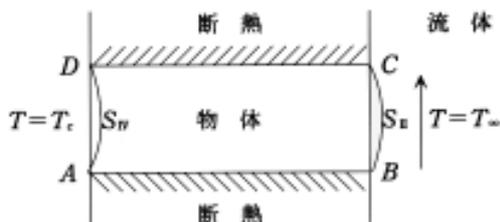


図9 例題

$$\begin{aligned} \int_{V^e} c_p \frac{\partial T}{\partial t} N dV^e + \int_{V^e} \left(\lambda_x \frac{\partial T}{\partial x} \frac{\partial N}{\partial x} + \lambda_y \frac{\partial T}{\partial y} \frac{\partial N}{\partial y} \right) dV^e \\ + \int_{S^e} \alpha T N dS = \int_{S_{top}} \alpha T_\infty N dS - \int_{S_{bot}} Q N dS \end{aligned} \quad (4.3)$$

となる。

4.2 要素マトリックスの生成

解析領域を四角形要素(2,2) ~ (M-1, N-1)に分けて要素マトリックスを生成する。 $E_{i,j}$ は要素を、 $P_{i,j}$ は節点を示し、節点 $P_{i,j}$ に演算器・メモリを対応させるとして

$$\int_{V^e} c_p \frac{\partial T}{\partial t} N dV^e$$

を計算する。

ここで要素 $E_{i-1,j-1}$ 内の温度分布を内挿関数 $N_{i,j}(x,y)$ により節点の温度を用いて表す。

$$\begin{aligned} T = N_{i-1,j-1} T_{i-1,j-1} + N_{i-1,j} T_{i-1,j} \\ + N_{i,j} T_{i,j} + N_{i,j-1} T_{i,j-1} \end{aligned} \quad (4.4)$$

とすると(図10)

$$\begin{aligned} \frac{\partial T}{\partial t} = N_{i-1,j-1} \frac{\partial T_{i-1,j-1}}{\partial t} + N_{i-1,j} \frac{\partial T_{i-1,j}}{\partial t} \\ + N_{i,j} \frac{\partial T_{i,j}}{\partial t} + N_{i,j-1} \frac{\partial T_{i,j-1}}{\partial t} \end{aligned} \quad (4.5)$$

ここで

$$N = N_{i-1,j-1}$$

とすると

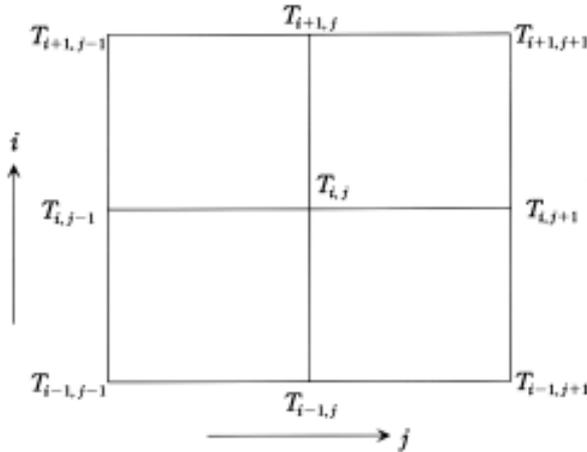


図 10 各節点の温度

以下同様に

$$N = N_{i-1,j}, N = N_{i,j}, N = N_{i,j-1} \quad (4.7)$$

とすると結果として要素 $E_{i-1,j-1}$ についてのマトリックスは図 11 のようになる。これを図 12 のように表すとする。

これらの計算は全要素に対して各々独立に求められるので並列処理が可能である。

同様の計算を

$$\int_{V^e} \left(\lambda_x \frac{\partial T}{\partial x} \frac{\partial N}{\partial x} + \lambda_y \frac{\partial T}{\partial y} \frac{\partial N}{\partial y} \right) dV^e, \int_{S^e} \alpha T N dS$$

についても行い全部の要素マトリックスを得ることができる。

$$\begin{aligned} \text{与式} = & \left[\int_{V^e} \rho \phi N_{i-1,j-1}^2 dV^e, \int_{V^e} \rho \phi N_{i-1,j-1} N_{i-1,j} dV^e, \right. \\ & \left. \int_{V^e} \rho \phi N_{i-1,j-1} N_{i,j} dV^e, \int_{V^e} \rho \phi N_{i-1,j-1} N_{i,j-1} dV^e \right] \\ & \times \frac{\partial}{\partial t} \begin{Bmatrix} T_{i-1,j-1} \\ T_{i-1,j} \\ T_{i,j} \\ T_{i,j-1} \end{Bmatrix} \quad (4.6) \end{aligned}$$

4.3 全体マトリックスの合成

節点 $P_{i,j}$ に注目する。

要素 $E_{i-1,j-1}, E_{i-1,j}, E_{i,j}, E_{i,j-1}$ の要素マトリックスの重み関数 $N_{i,j}$ に対応するマトリックス要素から全体マトリックスが合成できる。図 12 のように配列を観ると全体マトリックスは図 13 の各成分より合成できる。

ここで各列の値を加算すると全体マトリックスの合成が並列演算可能である。ただし、 $P_{i,j}$ が境界点で、対応

重み関数	$N_{i-1,j-1}$	$N_{i-1,j}$	$N_{i,j}$	$N_{i,j-1}$
$N_{i-1,j-1}$	$\int_{V^e} \rho \phi N_{i-1,j-1}^2 dV^e$	$\int_{V^e} \rho \phi N_{i-1,j-1} N_{i-1,j} dV^e$	$\int_{V^e} \rho \phi N_{i-1,j-1} N_{i,j} dV^e$	$\int_{V^e} \rho \phi N_{i-1,j-1} N_{i,j-1} dV^e$
$N_{i-1,j}$	$\int_{V^e} \rho \phi N_{i-1,j} N_{i-1,j-1} dV^e$	$\int_{V^e} \rho \phi N_{i-1,j}^2 dV^e$	$\int_{V^e} \rho \phi N_{i-1,j} N_{i,j} dV^e$	$\int_{V^e} \rho \phi N_{i-1,j} N_{i,j-1} dV^e$
$N_{i,j}$	$\int_{V^e} \rho \phi N_{i,j} N_{i-1,j-1} dV^e$	$\int_{V^e} \rho \phi N_{i,j} N_{i-1,j} dV^e$	$\int_{V^e} \rho \phi N_{i,j}^2 dV^e$	$\int_{V^e} \rho \phi N_{i,j} N_{i,j-1} dV^e$
$N_{i,j-1}$	$\int_{V^e} \rho \phi N_{i,j-1} N_{i-1,j-1} dV^e$	$\int_{V^e} \rho \phi N_{i,j-1} N_{i-1,j} dV^e$	$\int_{V^e} \rho \phi N_{i,j-1} N_{i,j} dV^e$	$\int_{V^e} \rho \phi N_{i,j-1}^2 dV^e$

図 11 要素マトリックス

	$N_{i-1,j-1}$	$N_{i-1,j}$	$N_{i,j}$	$N_{i,j-1}$
$N_{i-1,j-1}$	$(i-1, j-1)_1$	$(i-1, j-1)_2$	$(i-1, j-1)_3$	$(i-1, j-1)_4$
$N_{i-1,j}$	$(i-1, j-1)_1$	$(i-1, j-1)_2$	$(i-1, j-1)_3$	$(i-1, j-1)_4$
$N_{i,j}$	$(i-1, j-1)_1$	$(i-1, j-1)_2$	$(i-1, j-1)_3$	$(i-1, j-1)_4$
$N_{i,j-1}$	$(i-1, j-1)_1$	$(i-1, j-1)_2$	$(i-1, j-1)_3$	$(i-1, j-1)_4$

図 12 要素マトリックスの表現

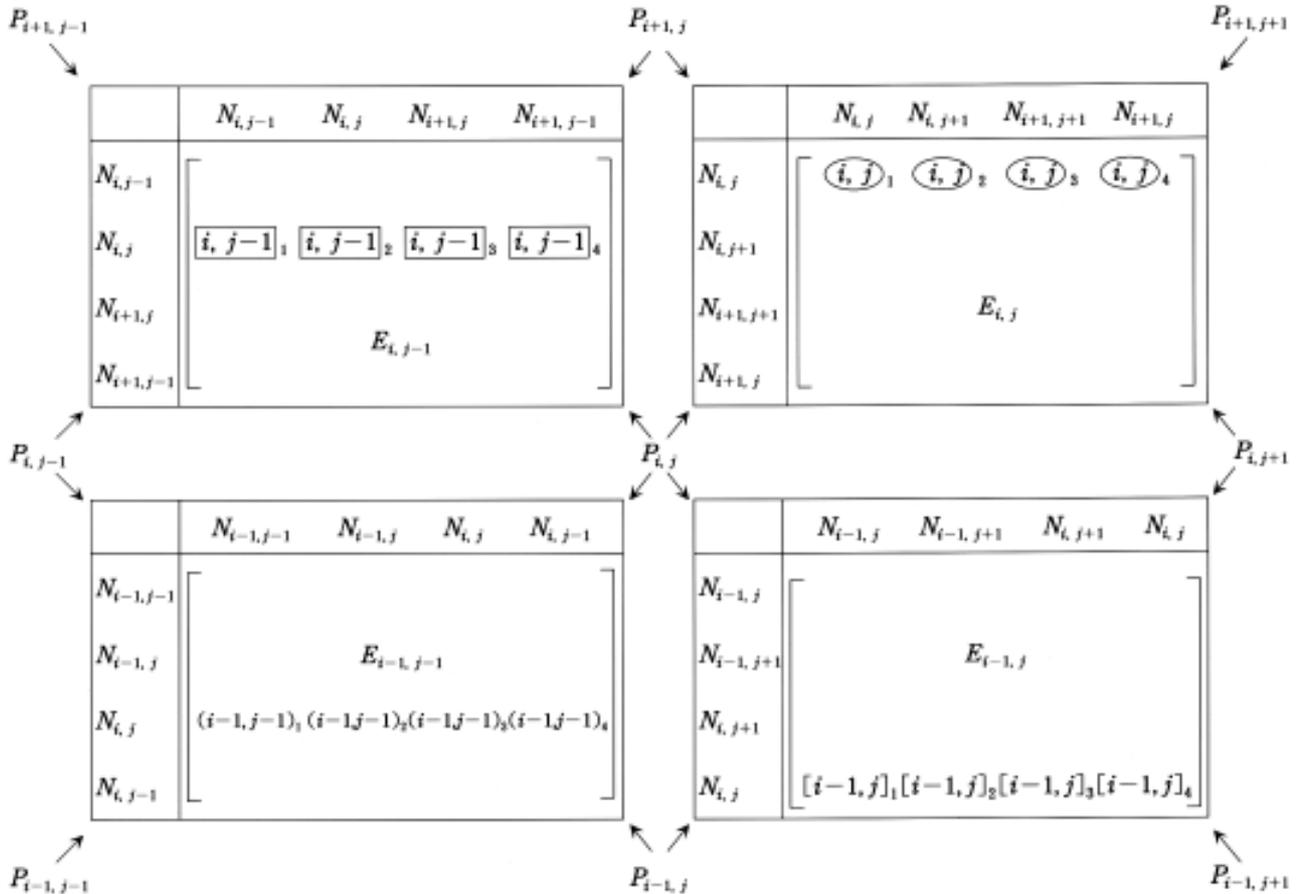


図 13 要素マトリックスの成分

する要素が存在しない場合にはダミーの要素を回りに配置するものとする。

また、全体マトリックスの合成に必要なデータは

- (i - 1, j - 1)
- (i, j - 1)
- (i + 1, j - 1)
- (i - 1, j)
- (i, j)
- (i + 1, j)
- (i - 1, j + 1)
- (i, j + 1)
- (i + 1, j + 1)

にあり、(i, j) から観ると全ては周囲 8 節点だけであり

データ転送、演算に要する時間が大幅に節約できる。また、通常計算される成分の値は 1 つの配列に格納されるが、ここでの並列処理では、メモリ (i, j) にそれ自身の節点と周囲の 8 個の節点についてのデータを格納し、全体マトリックスを分散する (図 14)。

4. 4 共役傾斜法による求解

連立 1 次方程式の解法では基本となった非定常熱伝導解析プログラム (THERM2) と同じ共役傾斜法を採用した。

それはマトリックスとベクトルの積の計算においてマトリックス成分中の非零成分のみを計算するからである。

$N_{i-1,j-1}$	$N_{i,j-1}$	$N_{i+1,j-1}$	$N_{i-1,j}$	$N_{i,j}$	$N_{i+1,j}$	$N_{i-1,j+1}$	$N_{i,j+1}$	$N_{i+1,j+1}$
$(i-1, j-1)_1$	$(i-1, j-1)_4$		$(i-1, j-1)_2$	$(i-1, j-1)_3$				
	$+ [i, j-1]_1$	$+ [i, j-1]_4$	$+ [i-1, j]_1$	$+ [i-1, j]_4$		$[i-1, j]_2$	$[i-1, j]_3$	
			$+ (i, j)_1$	$(i, j)_4$			$+ (i, j)_2$	$(i, j)_3$
			$+ [i, j-1]_2$	$+ [i, j-1]_3$				

図 14 全体マトリックスの合成

共役傾斜法で並列処理が可能なところは、マトリックスとベクトルの積をもとめるところおよびベクトルの内積の部分である。

ただし、内積を求めるところでは全体の合計が必要となり、それが性能の劣化をもたらす。

共役傾斜法ではその性能劣化が大きいプログラム全体としては、共役傾斜法による計算量が小さいため全体の性能は大きく低下することはない(5.2節参照)。

5. 並列処理プログラムのParagonにより検証

5.1 超並列計算機Paragon

本研究で開発した並列処理プログラムによって図9に示す単純な例題の計算を行った。その結果は2次元非定常熱伝導解析プログラム(THERM2)の計算結果と完全に一致した。この事は、本文で提案している超並列化の手法の正当性を示すものである。

本研究で仮想している隣接結合型超並列計算機は、その要素(演算器およびメモリ)が隣接する左右上下の4台の要素とのみ結合しているタイプのものであり各要素は各々格子状に配列されている。

並列化プログラムはM行N列の格子状に配置されている要素により計算を実行するものとしてコード化されている。

プログラムの構造は、1台の計算機で実行する時は

```
DO 100 I = 2, M - 1
DO 100 J = 2, N - 1
  処理
```

```
100 CONTINUE
```

となる。I, Jが2より始まりM - 1, N - 1で終わるのは周辺がダミーの要素マトリックスのためである。

想定している要素M × N台の隣接結合型の超並列計算機が存在すれば、この2重のDOループをとることができ、その制御処理が必要なくなり、その分性能が向上する。

理論上は使用する要素計算機の台数分だけ計算速度が高まるが、実際の並列計算機で処理した場合、台数分だけ性能が上がるかという疑問が当然出てくる。そこで現在稼働している並列計算機を使用して並列化プログラムを実行し、実験的に検証することを試みた。

本研究ではParagonを用いて検証実験を行った。

まず航技研で稼働している超並列計算機Paragon XP/S⁵⁾の概要について説明する。

この超並列計算機は米国インテル社製であり64bit RISCプロセッサi860XP CPUを演算ノードとし、演算ノード数が32から最大1024まで搭載可能なシステムとなっている。ここでは要素計算機をノードと呼ぶ。

ノードはMesh Routing Component (MRC)と呼ばれる通信機構と接続され、このMRCを介してほかのノードとのデータ転送を行う。また、32Mバイトのメモリと外部との入出力のためのポート等を持っている。

MRCは、上下左右の隣接した4ノードと結合されている。この結合方式は我々の想定している隣接結合型並列計算機と同様なものである。航技研のシステムは

16 × 21

合計336ノードで構成されている。

それぞれのノードは、32Mバイトのメモリをもつ独立した計算機で、ほかのノードとは完全に独立して動作する。すなわちParagonは独立した計算機同志をMRCで結合したシステムである。

Paragonを使用するときは、計算に必要とするノード数を指定する。そうすると指定した数のノードが確保されて計算が実行される。確保した各ノードには、0から始まる番号が付けられる。

プログラムは通常SPMD (Single Program Multiple Data)で実行される。すなわち確保した全ノードに同一のプログラムが格納されていて、各ノードは格納されたプログラムに従って独立に動作する。

Paragonによるプログラムの並列化はインテル社のNXライブラリと呼ばれる関数群を用いて行う。

本研究で用いたNXライブラリ関数⁶⁾は、次の通りである。

- ・mynode : 現在プログラムが実行されているノード番号を返す関数。
- ・csend : 任意のノードへ指定したデータを送信するための関数。受信関数crecvと同期してデータの転送を行う。
- ・crecv : 関数csendにより送られてきたデータを受信するための関数。送信関数csendと同期してデータの転送を行う。
- ・gdsum : 全ノードが持っている変数の値を加算するための関数。全ノードに加算された値が返される。
- ・dclock : システム起動時から現在までの時間を100n秒の分解能で取得する関数。

5.2 検証実験

まず、並列化プログラムを1ノードで実行した。この実行時間は、

3956.2秒

となった。

Paragonは336台の要素計算機から構成される超並列計算機である。今回の並列化プログラムの例題は

128 × 128

台の隣接結合型の要素計算機を使うものである。

検証実験ではParagonの128台の要素計算機を用いて行った。はじめに仮想計算機に合わせて128×128ノードを考えたが、ノード数が不足するため128×1とした。すなわち、隣接結合型計算機の1列128台の要素計算機をParagonの要素計算機1台に割り当てる。

使用する128台の要素計算機には

0 ~ 127

のノード番号が付けられる。

検証でのプログラムは、DOループが1重の

DO 100 J = 2, N

処理

100 CONTINUE

の形式となる。

並列化プログラムでは要素計算機の相互作用は

- ・上下左右のデータ交換
- ・各要素計算機のある変数の値の総和を計算の2つだけである。

検証実験では、上下のデータ交換は必要なく左右のデータ交換だけとなる。

超並列化プログラムでは要素計算機間でデータを相互に交換する変数は、

$X(M, N)$

の2次元配列で定義されている。検証実験ではこの変数は、

$X(N)$

の1次元配列となる。計算中には右左の要素計算機のデータを参照することがあるが、そのためのデータ格納領域として変数 X に対して右要素計算機の変数 X の値の格納領域

$XR(N)$

と、左要素計算機の変数 X の値の格納領域

$XL(N)$

を用意する。

これらの領域にはある処理が終了して、それぞれの要素計算機の変数 X の値が確定した時その値を左の要素計算機に対しては変数 XR へ、右の要素計算機に対しては変数 XL へデータを関数 $csend$ 、 $crecv$ を用いて転送する。

このような処理を追加することで隣接結合型の超並列計算機を模擬することができる。

仮想の隣接結合型超並列計算機では、このようなデータ転送の手続きは必要ない。従って検証実験では、このデータ転送処理は模擬のためのオーバーヘッドとなる。

全要素計算機のある変数の総計計算はParagonの持っている変数の総計計算のための関数 $gdsum$ を使用した。並列化プログラムから検証実験プログラムへの変換は上

記の点に考慮すれば機械的に行うことができる。

プログラムは、まずそれぞれのノード番号を取得する。取得したノード番号は、それぞれのノードが計算する列の番号となっている。

このようにして、並列化プログラムから書き換えた検証実験プログラムを128台の要素計算機で実行した。

128ノードは全く独立に計算を実行するが、左右のノードへデータを転送する時および変数の値の総和を計算する時に、それぞれのノードは同期がとられる。

計算時間の測定は関数 $dclock$ を用い、あるノードでの計算開始時刻と計算終了時刻から求める。

Paragon 128ノードによる計算時間は、

29.9秒

となった。

計算結果は最初に行なった1ノードの計算結果と完全に一致した。

以上の結果をまとめると表2のようになる。

表2 処理時間の比較

プログラム	計算時間	(a)/(b)
(a) 1ノードによる 検証プログラム	3956.2秒	
(b) 128ノードによる 検証プログラム	29.9秒	131.9

この計算では、データ転送のオーバーヘッドのため計算時間が128倍より小さくなると予想していたが、計算結果は128倍以上の131倍となった。

プログラムの構造は図5に示すようにマトリックスの生成部分と連立1次方程式の解法(共役傾斜法)に分けられる。共役傾斜法では解が収束するまで繰り返し計算される。繰り返し回数はこのプログラムでは4~9回である。したがって、ある時刻での計算時間は n を収束するまでの繰り返し回数、 ta をマトリックス生成時間および tb を共役傾斜法の1ループの時間とすると

$$ta + tb \times n$$

となる。

計算時間の実測の結果、1台のノードによるその計算時間は

$$8.4 + 0.35n$$

また128台では

$$0.0495 + 0.00588n$$

となった。

0.1秒から0.1秒おきに40.1秒までの401回の計算での平均の n は

$$4.08$$

でありこのときの計算時間の比は、

$$(8.4 + 0.35 \times 4.08) / (0.0495 + 0.00588 \times 4.08)$$

おおよそ 133.7 倍となる。

マトリックス生成部分の計算時間の比は、

$$8.4 / 0.0495 = 169.7$$

となり、また共役傾斜法の計算時間の比は

$$0.35 / 0.00588 = 59.5$$

となりマトリックス生成部分で効率を上げているのがわかる。128 より大きな値は一見不思議に思えるが、次の 2 点で説明できる。

(1) Paragon では 1 要素計算機あたり 16K バイトのキャッシュメモリを持っており 128 台の要素計算機を使うことにより各要素計算機の使用メモリ容量は 1/100 程度に軽減される。すなわちキャッシュメモリへのヒット率が向上することになる。

(2) 要素計算機を 1 列に配置すると 2 重の DO ループが 1 重の DO ループで済まされるため制御処理が軽減される。

以上、本文で提案した並列化の手法は隣接結合型並列計算機に対するものであるが、市販の並列計算機においても有効に機能することが分かった。

6. 結 論

科学技術計算分野において、有限要素法は汎用性があり複雑な形状や各種の境界条件を扱うことができるため、構造解析など多くの分野で利用されている。

また最近では大規模科学技術計算を超並列計算機で並列処理する試みが行われはじめています。

本研究では 2 次元非定常熱伝導有限要素法解析プログラムを基本とし、原田らの提案した隣接結合型並列計算機を対象としたアルゴリズムにより、超並列処理用の並列化プログラムを開発した。さらに、この並列化プログラムの検証を超並列計算機 Paragon によりおこなった。

その結果、次のような超並列化の条件を確認することができた。

- (1) 解析領域を $(M - 2) \times (N - 2)$ の格子に一对一の対応が可能な四角形要素で分割し、さらに解析領域の境界に沿ってダミーの四角形要素を配列する。したがって総要素数は $M \times N$ になる。これは解析領域内のすべての節点がそれぞれ周囲の 4 つの四角形要素を有するようにするためである。ただし、ダミーの四角形要素に関する要素マトリックスの成分は零に設定する。
- (2) 通常の有限要素プログラムにおいては、節点番号および要素番号をそれぞれ整数型の 1 つの指標を設定している。しかし、ここで用いた並列処理アルゴリズムでは、節点番号および要素番号をそれぞれ整数型の 2 つの指標で (I, J) のように表現する。それに

より並列計算機の (I, J) に対応した要素計算機のメモリに節点番号 (I, J) とその周囲の 8 つの節点に関するデータと要素番号 $(I - 1, J - 1)$ に関するデータを格納する。これにより通常の全体マトリックスの合成ではマトリックスの成分の値は各要素計算機で必要な 9 つの節点に関するデータを分散させて格納することになる。

格子状に分割したときの要素マトリックスの生成、全体マトリックスの合成および連立方程式の解法では各々並列処理が可能である。(1) および (2) の条件は対象とする問題の解析領域の形状や超並列計算機に依存する場合があると考えられるが、解析領域を部分領域に分割する方法あるいは指標の表示方法など今後の課題としたい。格子状に分割した効果は要素計算機を 128 台使用した時点で 131 倍を超えた。これにはキャッシュメモリの効果も含まれているが、それ以上に並列処理に適した格子生成の効果をもたらした結果である。

128 × 128 台の演算器からなる並列計算機を目標として有限要素法プログラムの超並列化をこころみた。結果は Paragon の 128 台の要素計算機で検証したが、並列化が有効に機能していることを示している。

そのためには構造格子に分割した各要素について要素マトリックスを得ることから始めるのが一番分かり易くそのように分割した。

超並列化したプログラムを超並列計算機 Paragon で実行し、演算器の台数に見合う性能向上を達成できたことはその妥当性を示すものと言える。

最後に本研究は航空宇宙技術研究所と神奈川工科大学との共同研究「有限要素法の有効性に関する研究」において行われたものであることをここに記す。

参考文献

- (1) 富田眞治, “並列コンピュータ工学”昭美堂, 1996年
- (2) 原田公一, 岩宮敏幸, “三次元 Skew 配列による並列計算機の性能”, 航空宇宙技術研究所報告 TR - 1211, 1993年
- (3) 山本行光, 奥村秀人, 大竹邦彦, “空力加熱と熱伝導のカップリング解析”, 官民特定共同研究「先進航空機の要素技術に関する研究」報告会要旨集, 1993年, pp.43-46
- (4) 原田公一, 奥村秀人, 磯部俊夫, “熱伝導有限要素法プログラムの超並列化条件”, 情報処理学会第 51 回全国大会, 1995年, pp.6-77 ~ 6-78
- (5) インテルジャパン(株), “パラゴン”平成 6 年 3 月, 超並列計算機システム利用説明会資料
- (6) インテルジャパン(株), “並列計算機利用の手引”, 平成 7 年 5 月