フリーメッシュ法を用いた弾塑性大変形解析の 分散並列処理とリメッシュアルゴリズム*

渡 部 修^{*1}, 伊 藤 久 雄^{*2}

Distributed Parallel Computing and Remesh Algorithm in Elasto-Plastic Large Deformation Problem using Free Mesh Method

Osamu WATANABE^{*3} and Hisao ITO

^{*3} University of Tsukuba, Inst. of Engineering Mechanics and Systems,
Tsukuba, Ibaraki, 305-8573 Japan

Recently, meshless method, which needs nodal data only as input data, is widely noticed of its significance. The authors have recently applied one of meshless methods, Free Mesh Method (abbreviated hereafter by FMM), to the elastic-plastic large deformation problem. FMM proposed by Yagawa et al. has some similarities to the conventional FEM, where the tentative triangular element are created around the node. This paper describes the formulation and implementation for FMM in parallel computational environment using PVM (Parallel Virtual Machine) communication architecture designed for PC clusters. Furthermore, FMM can utilize the remesh algorithm, since FMM is based on Node-by-Node concept. The present paper will confirm effects of remeshing algorithm to improve accuracy of numerical solutions as an example of largely deformed shear block.

Key Words: Free Mesh Method, Finite Element Method, Remeshing, Distributed Parallel Computing, Domain Decomposition Method, Elasto-Plastic Large Deformation Problem

1. 緒 言

計算機の性能向上に伴い、数値解析の対象が大規模で複雑な形状に向かっている。計算機技術の進歩により、これまでは扱えなかった問題が解けるようになってきたが、解析モデル作成のプレプロセスに要する労力が無視できない問題となっている。また、三次元物体では、対象により自動要素分割が必ずしも遂行できないという状況も指摘できる。このような状況を改善する方法として現在入力情報として要素-節点コネクティビティ情報を必要としないメッシュレス有限要素法⁽¹⁾や、三次元物体をさいの目状に分割するボクセル有限要素法⁽²⁾の研究が行われている。両手法とも、CADなどで作成したデータに基づき解析モデルを作成し、引き続き解析作業に入れるため、大規模解析では作業の省力化に大きく貢献できると考えられる。

これまでに、著者ら⁽³⁾はメッシュレス法の一つであるフリーメッシュ法⁽⁴⁾(Free Mesh Method, FMM)を弾塑性大変形問題に適用し、解析モデルの作成や数値

解析の挙動について従来の有限要素法と比較・検討を行ってきた。節点ごとの処理を基本とするフリーメッシュ法は、節点の剛性を節点周りに一時的に作成される三角形要素に基づいて評価する。このため、節点での情報の集約化の過程で解の平均化が起り精度が多少劣化するが、局所変形を追跡できるぐらいの節点がある場合には、フリーメッシュ法は有限要素法と同程度の精度があり、有限要素法で見られる不規則要素に依存した解の局所化などの現象は防止できることを示した。

本研究では、フリーメッシュ法を大規模な弾塑性大変形問題に適用することを目標として、解析モデルの作成から大変形解析に至るシームレスな解析システムの構築を行うことを目的としている。複数のパーソナルコンピュータをネットワーク接続したPCクラスタ形の並列計算機を用いて、フリーメッシュ法の50万自由度程度の大規模問題における実用性や、従来の有限要素法との比較・検討することを目的とする。またフリーメッシュ法では節点ごとの処理を基本としているため、節点周りの要素配置を動的に変更するリメッシュが容易に実現できる。本研究では、このリメッシュが解の精度に及ぼす影響を検討することも目的としている。

* 原稿受付 1999年8月23日。

^{*1} 正員, 筑波大学機能工学系(〒305-8573 つくば市天王台1-1-1)。

^{*2} 准員, 筑波大学大学院。

E-mail: watanabe@kz.tsukuba.ac.jp

2. フリーメッシュ法

2・1 基礎概念 メッシュレス法の中でも、矢川らの提案する FMM⁽⁴⁾ は、従来の FEM と同様に剛性評価のための空間領域単位として要素を用いており、FEM を直接メッシュレス化したものとみなすことができる。FMM では図 1 に示すように、各節点を中心節点としてその近傍に位置する衛星節点と呼ばれる他の節点群を用いて、一時的に中心節点に関するローカルメッシュと呼ばれる三角形要素群を作成する。一時的に生成される三角形要素について従来の FEM の手法をそのまま適用して、節点の剛性行列を計算する。この剛性評価をすべての節点について実施することにより、構造物の全体剛性行列を作成する。このローカルメッシュを作成する手段としては、次に述べる二つの手法が提案されている。

2・2 ローカルメッシュの作成 ローカルメッシュを作成する手法として、対角線分の長さを比較する対角線比較法とデローニー分割を利用する方法が現在提案されており、どちらの手法を用いても生成されるローカルメッシュはほぼ同一であることが示されている⁽³⁾。対角線比較法はプログラムが比較的容易であり、二次元形状では一意的なメッシュを生成する。本研究では三次元への拡張を考慮して、デローニー分割法⁽⁵⁾を局所的に適用することでローカルメッシュを作成する。デローニー分割はこれまで二次元、三次元有限要素法のメッシュジェネレータに使われており、合理的かつ確立した手法である。デローニー分割法を局所的に適用するための手順を、図 2 を参考にしながら、以下に示す。

(a) 検索半径内にある候補節点(点 $m, n, o, p, \dots$)を集め、この検索円を内接する仮想三角形 ijk を設定する。

(b) 中心節点 i でこの仮想三角形を三つの三角形($\triangle ijl, \triangle ijk, \triangle kil$)に分割する。

(c) 候補節点 m を内部に含む三角形($\triangle kil$ が相

当)を探し、この三角形をさらに三つの小三角形($\triangle ilm, \triangle kml, \triangle kim$)に分割する。

(d) この m を頂点にもつ三つの小三角形それぞれについて、外側に隣接する三角形とで四角形を作成する。例えば $\triangle ilm$ の場合、隣接する $\triangle ijl$ と合わせて四角形 $mijl$ を作成する。

Fig. 2 Generation of local mesh by Delaunay method

Fig. 1 Local mesh around central node

(e) 隣接三角形 ijl の外接円内に点 m が含まれているかどうかを調べる。図 2(e)では、 m は $\triangle ijl$ の外接円内に含まれている。

(f) 四角形 $mijl$ におけるもとの共有辺 il を消して、 jm でこの四角形を二つの三角形に分割する。このように隣接三角形の外接円内にもう一方の三角形の対向する頂点が含まれるかどうかを判定し、もし含まれていれば四角形の対角線を入れ替える操作をスワッピングと呼ぶ。 $\triangle ilm$ と $\triangle kml$ を合わせた四角形 $ilkml$ 、 $\triangle ilm$ と $\triangle kim$ を合わせた四角形 $ilmk$ でも同様の評価を行う。また残り二つの $\triangle kml$ 、 $\triangle kim$ についても、隣接三角形の外接円が点 m を含むかどうか判定し、結果に応じて対角線を入れ替えるスワッピングを行う。

(g) 2 番めの候補節点 n を含む三角形を探し(ここでは $\triangle kml$)、この三角形を三つの小三角形に分割する。以下、先程と同じように隣接三角形とで四角形を作成し、順次スワッピングを実行する。

(h) この操作をすべての候補節点に関して行う。途中経過の様子を図 2(h)に示す。

(i) 最終的にすべての候補節点で上記の操作が終了したら、仮想三角形の頂点 i, j, k を含む小三角形を消去し、中心節点 l を頂点に含む小三角形を抽出すると中心節点まわりの偏平のないローカルメッシュが一意に決定する。

この操作はすべて節点単位で行われ、必要な情報は検索半径内にある候補節点情報だけなので、並列化にも容易に対応できる。

2.3 剛性評価法 FMM では、ローカルメッシュの要素剛性行列を全体剛性行列に足し込み、これをすべての節点で繰返すことにより、全体剛性行列を得る。図 3 に示すように、三角形 IJK で I, J, K のいずれを中心節点としても、一時的なメッシュとして三角形 IJK が選ばれるため、FMM の全体剛性行列は FEM と比べて 3 倍の値を持つことになる。また、要素での応力などのデータは、図 3(a)に示すように節点でのデータに次のように変換される⁽³⁾。

Fig. 3 Conversion of data between node and element

$$(\sigma_{ij})_{I\text{-node}} = \sum_i \left\{ \frac{W_i}{W} (\sigma_{ij})_{i\text{-element}} \right\} \dots\dots\dots (1)$$

ここで、 W は節点 I 周りの重み W_i の和である。逆に要素の応力を評価するのに、図 3(b)に示すように節点に蓄積していたデータを基礎に次のように変換する。

$$(\sigma_{ij})_{i\text{-element}} = \frac{1}{3} \{ (\sigma_{ij})_{I\text{-node}} + (\sigma_{ij})_{J\text{-node}} + (\sigma_{ij})_{K\text{-node}} \} \dots\dots\dots (2)$$

3. 並列化手法

有限要素法を並列化する手法として、図 4 に示すように解析モデルを複数の領域に分割し、各領域の解析を複数のプロセッサに分担させる領域分割法⁽⁶⁾⁽⁷⁾(DDM, Domain Decomposition Method)が広く用いられている。FMM では剛性を評価する空間領域単位として従来の有限要素法と同様に要素という概念を用いているため、基本的に同じ手法の適用が可能である。

本研究では領域ごとに内部自由度を消去するサブストラクチャ法⁽⁸⁾(Sub-Structure method)と、全自由度を共役こう配法(CG法, Conjugate Gradient Method)で反復収束させる次の 2 種類を扱うことにする。

3.1 領域分割形サブストラクチャ法

サブストラクチャ法(Substructure method, SS)は、本来メモリーを節約して大規模問題を解く手法として古くから用いられている。サブストラクチャ法ではまず、解析モデルを複数の部分領域に分割し、部分領域(k)における剛性方程式を内部自由度 $\{U_s^{(k)}\}$ と領域間境界自由度 $\{U_i^{(k)}\}$ に分けて、

$$\begin{pmatrix} K_{ss}^{(k)} & K_{si}^{(k)} \\ K_{is}^{(k)} & K_{ii}^{(k)} \end{pmatrix} \begin{Bmatrix} U_s^{(k)} \\ U_i^{(k)} \end{Bmatrix} = \begin{Bmatrix} F_s^{(k)} \\ F_i^{(k)} \end{Bmatrix} \dots\dots\dots (3)$$

という形で構成する。ここで上添字の k は領域番号、下添字の s は領域内部自由度、 i は領域間境界自由度を表している。このとき、全体領域剛性マトリックスは図 5 のような矢印形の分布になるが、図 5 の太枠の

Fig. 4 Domain decomposition method

部分のみメモリーに読み込むことにより記憶容量を節約する。この式からコレスキー分解などの直接法を利用して内部自由度 $\{U_s^{(k)}\}$ を消去すると次式が求まる。

$$[A^{(k)}]\{U_i^{(k)}\} = \{b^{(k)}\} \quad \dots\dots\dots (4)$$

ここで

$$[A^{(k)}] = [K_{ii}^{(k)}] - [K_{is}^{(k)}][K_{ss}^{(k)}]^{-1}[K_{si}^{(k)}] \quad \dots\dots (5\cdot a)$$

$$\{b^{(k)}\} = \{F_i^{(k)}\} - [K_{is}^{(k)}][K_{ss}^{(k)}]^{-1}\{F_s^{(k)}\} \quad \dots\dots (5\cdot b)$$

式(5・b)はシュール・コンプリメント・マトリックス (Schur complement matrix) と呼ばれるマトリックスである。従来のサブストラクチャ法では、この式(5・a)を全領域に対して全体化し、直接法で解いていた。しかし、ここで直接法ではなくCG法などの反復法を用いることで、シュール・コンプリメント・マトリックス $[A^{(k)}]$ を直接全体化した $[A]$ を求めることなく式(4)の全体化が可能である。CG法で最も計算負荷のかかる部分はマトリックス・ベクトル積 $[A]\{p\}_n$ で、これは以下のように表すことができる。

$$[A]\{p\}_n = \sum_{i=1}^N [A^{(k)}]\{p\}_n \quad \dots\dots\dots (6)$$

ここで N は全領域の総数である。すなわち、式(6)の右辺のように、各 slave プロセスが並列に計算したシュール・コンプリメント・マトリックス $[A^{(k)}]$ と探索方向ベクトル $\{p\}_n$ の積を、master プロセスに集め

Fig. 5 Local stiffness matrix in substructure method

Fig. 6 Globalized local vectors with communications

て1本の全体化されたベクトルにまとめることで(図6), 直接 $[A]$ を求めなくても式(6)の左辺の計算が可能である。一次元のベクトル量の送受信だけで式(4)の全体化が図れ、内部自由度を消去したことによりベクトルの大きさは大幅に縮小されているため、高効率の並列化が行える。最終的に求められた $\{U_i^{(k)}\}$ を、各領域を担当する slave プロセスに送り、次のように後退代入により領域内部自由度 $\{U_s^{(k)}\}$ を計算する。

$$\{U_s^{(k)}\} = [K_{ss}^{(k)}]^{-1}\{F_s^{(k)}\} - [K_{ss}^{(k)}]^{-1}[K_{si}^{(k)}]\{U_i^{(k)}\} \quad \dots\dots\dots (7)$$

3.2 領域分割形共役こう配法 一方、3.1節のようにシュール・コンプリメント・マトリックスを作成して領域内部自由度を消去せずに、領域内の全自由度に対して共役こう配法を適用することも可能である。すなわち、式(3)のように全自由度を内部自由度と境界自由度に分離せずに、通常FEMのような一定のバンド幅を持つ形で剛性マトリックス $[K^{(k)}]$ を構成し、式(6)のシュール・コンプリメント・マトリックス $[A^{(k)}]$ をそのまま $[K^{(k)}]$ に置き換えて解く方法である。

これを領域分割形共役こう配法と呼ぶことにする。この方法は、プログラミングは容易であるが、通信するベクトルサイズは全自由度数分になるため、クラスター形の通信性能の低い環境では不利になると考えられる。

3.3 弾塑性解析の流れ FMMによる弾塑性解

Fig. 7 Flow chart of Elasto-Plastic FMM analysis

析の流れを図7に示す。領域分割形サブストラクチャ法と領域分割形共役こう配法の主な違いは点線で示したシュー・コンプリメント・マトリックス作成の有無である。解析モデルの領域分割、グローバル・ローカル節点番号、自由度インデックスの変換等はプログラムが自動的に処理するようにしているため、解析モデルは通常1台の計算機で実行する場合と同じものを用意すれば良い。

FMMの基本的な入力情報は節点情報データだけなので、領域分割作業は従来のFEMのような要素群の分割ではなく、節点群の分割となる。各slaveプロセスは自分に割り当てられた節点群について、領域間境界で不整合を起さないようにローカルメッシュを作成する。

FMMでは、剛性マトリックスを作成する際に近傍にある他の節点群を用いてローカルメッシュを作成するため、FEMの要素分割プロセスに相当する部分も並列処理できる。これは並列処理可能部分の拡大をもたらす、大規模・複雑形状の解析では有効であると考えられる。本プログラムでは、節点生成と解析結果表示(可視化ファイル作成)以外はすべて並列環境上で実行されている。

4. リメッシングアルゴリズムの効果

大変形を伴う解析では、初期に形状の良いメッシュを作成するように節点を配置しても、変形の進行に伴いメッシュがゆがんでくる。そこで、移動した節点を元に、デローニー分割法を用いて中心節点周りのローカルメッシュを改めて作り直すリメッシュの効果を検討する。従来、誤差解析を組合わせて要素の節点を移動する r 法や節点を増加させる h 法などが知られているが、ここでは、節点周りの要素の再配置を行うものをリメッシュと呼んでいる。

4.1 解析モデル 平面ひずみ状態の長方形平面ブロックにせん断変形を与える。図8のように、左右

Fig. 8 Boundary condition and geometry of shear deformation model

辺の拘束を取り除いた一般せん断問題にフリーメッシュ法を適用する。解析モデルは $a=200$ mm, $b=100$ mmとしている。荷重は水平方向に強制変位 e に相当する変位速度 $\dot{e}=200$ mm/sを与える。拘束条件としては、上辺の x 方向、底辺の x, y 方向を固定し、上辺が x 方向にのみ移動できる。時間増分は0.001 s, 増分総数は1000回である。

4.2 相当塑性ひずみ分布 1000節点と4000節点の2種類の解析モデルで、通常のFMM解析と50ステップごとによりメッシングを行ったFMM解析での相当塑性ひずみ分布図を図9に示す。100%以上のひずみを黒とし、その間を10等分割している。後で示す細かい分割によるひずみ分布図と比べると、リメッシュを行う1000節点の場合は図9に近いが、4000節点の程度の分割になると、リメッシュの有無による違いはなくなる。

4.3 偏平率 図10にこのときのローカル要素分割を示す。よりわかりやすくするため、メッシュの形状の良否を示す指標として、偏平率 H を三角形 IJK の角度を用いて次のように定義する⁽⁶⁾。

$$H = \left| \frac{\pi}{3} - \angle I \right| + \left| \frac{\pi}{3} - \angle J \right| + \left| \frac{\pi}{3} - \angle K \right| \dots (8)$$

正三角形では偏平率 H は零であり、三角形が完全に

(a) $N=1000$

(b) $N=4000$

Fig. 9 Effect of remesh in plastic strain contour at $e/a=1.0$

つぶれた直線状態では偏平率 H は最大値 $4/3\pi$ となる。図 11 にこの間を 10 等分割した偏平率の分布を示す。図 11 から偏平率は、1 000, 4 000 節点ともリメッシュにより大きく改善されていることがわかる。

5. 各種アルゴリズムの比較

5.1 計算機環境 本研究では表 1 に示す基本スペックを持つ PC クラスタを使用した。各 PC は 2 個の CPU を搭載しており、総計 10 台(最大 20 CPU)、メモリーは 3.4 Gbyte 使用できる。ネットワーク通信には、スイッチングハブを介して Fast Ethernet で接続している。OS には PC-UNIX である Linux (ver. 2.0.36)、通信ライブラリーにはメッセージパッシング形の PVM(ver. 3.4)を搭載している。購入した時期が異なるため、Type 1 と Type 2 の二つのパーソナルコンピュータ群による PC クラスタ構成となっている。

5.2 用いたアルゴリズム 解析モデルは表 2 に示すように、節点数の異なる 4 種類の長方形板モデルを作成した。Ex. 1~4 の各モデルでは、FMM 用(節点情報データのみ)と FEM 用(節点と要素情報データ)の 2 種類があるため、合計 8 種類のモデルデータを用意した。要素単位で処理を行う従来形の FEM と、メッシュ生成から一貫して節点単位の処理を行う

FMM の 2 種類の解析手法を比較する。また、並列化アルゴリズムとして内部自由度を消去し領域境界自由度のみに共役こう配法を適用する領域分割形サブストラクチャ法(DDM-SS)と、全自由度に対して共役こう配法を適用する領域分割形共役こう配法(DDM-CG)がある。よって、その組合せにより次の四つの解析アルゴリズムを比較する。

- (1) FEM・DDM-SS
- (2) FMM・DDM-SS
- (3) FEM・DDM-CG
- (4) FMM・DDM-CG

5.3 計算時間 領域分割形サブストラクチャ法では、解析に使用する計算機台数と領域分割数が全体

Table 1 Composition of PC clusters

	Processor	Memory
Type.1	Pentium II 350MHz × 8	256 Mbytes/CPU
Type.2	Pentium II 266MHz × 12	112 Mbytes/CPU

Network : 100Base-TX, Switching Hub

Table 2 Rectangular plate models

Model	Number of nodes	Number of D.O.F.
Ex.1	20,000	39,666
Ex.2	50,000	99,498
Ex.3	100,000	199,412
Ex.4	300,000	599,012

Fig. 10 Mesh distortion at $e/a=1.0$

Fig. 11 Distorsion index H at $e/a=1.0$

の解析時間を決定するパラメータとなる。Ex. 1~3のモデルで、プロセッサ数を8個(Type 1)に固定し、領域分割数を変化させたときの1タイムステップに要する計算時間を図12(a)に示す。共役こう配法における残差の収束判定条件は $\varepsilon=1.0 \times 10^{-10}$ とし、ローカルメッシュ生成に要する計算時間は候補節点を集める際の検索半径の大きさに依存するため、ここでは含めないことにする。図12(a)に示すように、FMMおよびFEMともに解析に要する時間はほぼ同じであり、どちらも計算時間を最短にするための最適な領域分割数が存在することがわかる。また、図12(b)に示すように、領域分割形共役こう配法の解析時間は、ほぼ並列台数とネットワーク飽和の有無に依存することが知られており⁽⁸⁾、領域分割数はDDM-SSほど大きな影響を与えない。

(a) DDM-SS analysis

(b) DDM-CG analysis

Fig. 12 Cpu time for 1 incremental time step analysis

Fig. 13 Details of cpu time in DDM-SS analysis

二次元三角形要素を用いたFMMでは、剛性マトリックス作成や応力・ひずみの計算に要する演算量は通常のFEMに比べて約3倍増加する。しかし図13に示すように全体の計算時間の約8割はソルバが占めており、この演算量の増加分が計算時間に及ぼす影響はほぼないと言って良い。よってメインプロセスでは両者はほぼ同等の性能を持っていると判断できる。

それぞれ4種類の解析モデルにおける、最適な分割

Table 3 Comparison of cpu time for DDM-SS and DDM-CG for 1 incremental time step

Model	DDM-SS		DDM-CG	
	FEM	FMM	FEM	FMM
Ex.1	21 sec	23 sec	110 sec	110 sec
Ex.2	66 sec	77 sec	453 sec	454 sec
Ex.3	202 sec	184 sec	1,244 sec	1,233 sec
Ex.4	903 sec	999 sec	6,069 sec	6,791 sec

System: Ex.1~3 Type.1(8CPU), Ex.4 Type1+Type.2(20CPU)

Table 4 Parallelization efficiency in Ex. 1 model

	DDM-SS		DDM-CG	
	FEM	FMM	FEM	FMM
16 division	85.6%	85.0%	57.3%	56.1%
64 division	63.1%	65.2%	38.0%	37.8%

(a) DDM-SS analysis

(b) DDM-CG analysis

Fig. 14 Convergence of Residual in FMM and FEM using DDM-CG

Fig. 15 Comparison of FMM and FEM in plastic strain contour at $e/a=1.0$ for $N=50\,000$ problem

数に設定したDDM-SS法とDDM-CG法の計算時間を表3に示す。使用した解析システムはEx. 1~3がType 1(8 CPU), Ex. 4がType 1とType 2(20 CPU)である。クラスタ形の通信性能の低い並列環境では、分割数を最適な値に設定したDDM-SS法のほうが計算時間ははるかに短い。

また、Ex. 1モデルを用いて、1台での計算時間を基準にした4台のマシンでの領域分割数に対する並列化効率の比較を表4に示す。領域分割数が増えるほどCG法を適用する領域間境界自由度が増加し、その結果通信量が増えるため、並列化効率は低下する。計算時間を短縮することと並列化効率を上げることは必ずしも同一ではなく、解析時間を最短にするためには領域分割数などの並列パラメータを最適に設定することが重要である。

5.4 共役こう配法の収束性 一般に反復法は直接法とは異なり、問題の性質により解を得るまでの演算数が増えるため、その収束性を評価することは重要である。39 666自由度のEx. 1モデルをDDM-SS法とDDM-CG法で解いたときの残差の収束状況をそれぞれ図14に示す。領域分割数はどちらも64分割としており、DDM-SSでは内部自由度を消去することにより、解くべき自由度数は4 236自由度となり、DDM-CG法に比べて収束回数はおおよそ1/4になっている。また、変形が小さいうちは、FEMおよびFMMともにほぼ同様の収束性を示しているが、変形が大きくなるにつれて若干収束性に差異が表れてくる。

図15に、Ex. 2の $N=50\,000$ 節点の解析結果を示す。変形が $e/a=1.0$ のときの塑性ひずみ分布であり、両者ともほぼ一致した分布形となっているが、FMM

の分布がより滑らかである。

6. 結 言

フリーメッシュ法を数十万自由度規模の問題に適用するため、PCクラスタを用いた並列環境で、解析モデルの作成から弾塑性大変形解析に至るシームレスな解析システムを構築した。局所的にデローニー法を適用することにより、解析途中での動的なリメッシングが可能となった。大変形を伴う解析では解析途中でリメッシングを行うことにより、変形の著しいメッシュが修正され、節点数が少ないとき、すなわちモデルと比較してメッシュサイズが大きい場合には精度改善の効果が高いことを確認した。

FEMで用いられている領域分割法をFMMに適用し、並列計算の高速化と大規模記憶容量の確保を図り、数十万自由度程度の解析が十分実用的な時間内で行えることを実証した。メインプロセスでのFEMとFMMの計算時間はほぼ同等であるが、プレプロセスの簡素化並びにメッシュ生成の分散並列化を考慮すれば、解析作業全体の省力化としてFMMは有効であると考えられる。

本研究の一部は情報処理振興事業協会「領域分割フリーメッシュ法による高度設計支援基盤ソフトウェアの開発」(代表: 矢川元基東京大学教授)より援助を得た。委員会において貴重なご意見をいただいた委員各位に対し、謝意を表する。

文 献

- (1) 奥田洋二, 計算工学におけるメッシュレス解法, 計算工学, 1-3(1996), 151-162.
- (2) 鈴木克洋・寺田賢二郎・大坪英臣・米里直樹, ボクセル情報を用いたソリッド構造の解析法, 日本造船学会論文集, 第182号, (1997), 595-600.
- (3) 渡部修・瀬川冬樹, フリーメッシュ法の弾塑性大変形問題への適用, 機論, 65-630, A(1999), 268-276.
- (4) 矢川元基・細川孝之, フリーメッシュ法(一種のメッシュレス法)の三次元問題への適用, 機論, 63-614, A(1997), 3263-3268.
- (5) 谷口健男, FEMのための要素自動分割(デローニー三角分割の利用), (1992), 森北出版.
- (6) 矢川元基・塩屋隆二, 超並列有限要素解析, (1998), 朝倉書店.
- (7) 堀江知義, 手軽にできる並列有限要素解析(第3回), 計算工学, 2-3(1997), 177-183.
- (8) Farhat, C. and Wilson, E., A new finite element concurrent computer program architecture, *J. Num. Meth. Eng.*, 24(1987), 1771-1792.
- (9) 堀江知義・倉前宏行, ネットワーク上のワークステーションによる並列有限要素解析および性能評価, 機論, 61-584, A(1995), 861-868.