並列化プラグイン機能を利用した並列有限要素法コード
の開発とその性能評価*奥田 洋司^{*1}, 柄谷 和輝^{*2}, 永田 雅喜^{*3}Development of the Parallel Finite Element Method Code Using the
Parallelized Plug-in Function and Performance EvaluationHiroshi OKUDA^{*4}, Kazuteru GARATANI and Masaki NAGATA^{**} Department of Quantum Engineering and Systems Science, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

In recent years, more large-scale and complicated numerical analysis came to be performed in connection with fast development of technology. Such a large-scale problem is carrying out parallel computing, and has realized a short time and highly precise analysis. However, since the knowledge of parallel computing and experience are needed for development of a parallel analysis program, for a common user, it becomes the work which spends a remarkable labor and time. Then, in this study, the parallel FEM analysis code was developed using the plug-in function of GeoFEM. The plug-in function here is for developing parallel FEM analysis code without the knowledge of parallel computing. And parallel performance evaluation of the developed program was performed, and it checked that the high parallel performance of 80% or more of the rates of CPU usage came out also under a low communication performance like PC cluster.

Key Words: GeoFEM, Plug-In, Parallel, PC-Cluster, Speed-Up, Parallel Efficiency

1. 緒 言

近年, 科学技術の飛躍的な発達に伴い, より大規模で複雑な数値解析が求められるようになった。しかしながら単体の計算機における大規模解析には, ハードウェアにより取扱える問題に制限があるため, 解析領域をいくつかの領域に分割し, 並列計算機上で並列に処理を行う方法が取られている。この並列計算で使用する並列計算機には大きく分類すると共有メモリ型と分散メモリ型がある。この内, 共有メモリ型並列計算機に関しては, OpenMP⁽¹⁾やHPF⁽²⁾(High Performance Fortran)などの並列化支援言語などの技術により, 既存の逐次解析コードの状態でも, ある程度の並列計算が可能になってきている。一方で, 分散メモリ型の並列計算機においては, PE(Processing Element)間のデータ転送にMPI⁽³⁾(Message Passing Interface)やPVM⁽⁴⁾(Parallel Virtual Machine)など

のライブラリを用いた明示的な記述が必要になる。さらに, より計算効率を上げるためには, 並列化に適したデータ構造とアルゴリズムを採用したコードに書き換える作業が必要になる。しかし, この作業には並列計算の知識, 経験が必要とされるため, 逐次解析コードを並列化することは, 一般のユーザにとってかなりの労力, 時間を要する作業といえる。そのため, 逐次コードからの並列化を支援する仕組みが広く望まれている。

また, 最近ではPC(Personal Computer)およびネットワーク周辺機器の高性能低価格化が進んだことと, UNIX互換のフリーOSとして知られる「Linux」の登場により, 安価で高速なパーソナルコンピュータを寄せ集めたPCクラスと呼ばれる仮想分散メモリ型並列計算機が容易に構築できるようになった。これにより並列解析コードさえあれば, 大型並列計算機を使用しなくとも並列解析が身近で実行できるようになった。今後もPCクラスのような分散メモリ型並列計算の計算機環境が増大すると予想されることから, このような分散メモリ型に対応した並列解析のコード開発期間の短縮と, 並列性能は非常に重要な問題になる。

以上のことを背景に本研究では, 並列有限要素法解

* 原稿受付 2001年8月9日。

^{*1} 正員, 東京大学大学院工学系研究科(☎113-8656 東京都文京区本郷7-3-1)。^{*2} 正員, CRCソリューションズ(☎136-8581 東京都江東区南砂2-7-5)。^{*3} 横浜国立大学大学院(☎240-0067 横浜市保土ヶ谷区常盤台79-5)。

E-mail: okuda@q.t.u-tokyo.ac.jp

析プラットフォームとして知られる GeoFEM⁽⁵⁾ の逐次解析コードを並列化するプラグイン機能を利用し、並列有限要素解析コードの開発を行った。そして種々の計算機環境で、そのコードの並列性能評価を行った。

2. 並列化プラグイン機能

2.1 GeoFEM について GeoFEM とは超高速並列計算機システム(地球シミュレータ)の応用ソフトウェアのひとつであり、(財)高度情報科学技術研究機構(RIST)を中核拠点に開発が進められている多目的並列有限要素法解析システムである。固体地球シミュレーションでは、そのスケールの大きさと現象の複雑さから問題解決に大規模並列計算は不可欠となるため、より効率的に並列計算を行う必要がある。それには、そのハードウェアに合わせたアルゴリズムと適切なプログラミングが不可欠となる。並列有限要素法では、その基礎となるデータ構造から並列化に十分留意する必要がある。既存のプログラムを単純に並列化しただけでは、効率の良い解析はできない。GeoFEM はこのような観点から、まず並列計算機に適した解法を検討・選択し、並列化に適したデータ構造とアルゴリズムを採用し、有限要素法により精度と効率の良いシミュレーションを可能としたシステムである。特徴のひとつとして GeoFEM は高度な並列性能を有する並列プラットフォームとして利用することができ、一種のテンプレートとしてさまざまな並列解析機能の追加・開発が可能なプラグイン機能を有している。本研究では、このプラグイン機能を利用して並列有限要素法コード開発を行った。

2.2 プラグイン機能概略⁽⁶⁾⁽⁷⁾ 一般に並列解析コードを開発する場合、まず逐次解析コードを開発し、その後並列化する方法が取られる。このような逐次解析コードの並列化を行う場合、まず考慮すべき点は、

入力データの領域分割、通信テーブルの作成、解析部の並列化、並列の入出力などである。並列計算の知識、経験のない一般のユーザにとっては難解な作業であるが GeoFEM のプラグイン機能を利用することで上記問題は解消される。プラグイン機能を利用することは、ユーザの作成した逐次解析コードに GeoFEM 本体からサポートされている並列入出力、並列線形ソルバや可視化機能を組み込み並列化することである。

図 1 に一般的な有限要素法解析コードを GeoFEM にプラグインする場合のイメージ図を示す。GeoFEM がプラットフォーム化している斜線部分を、ユーザの逐次解析コードと各処理部分と交換することをプラグイン作業と呼ぶ。この作業は、並列処理が集約された並列線形ソルバを利用しているため、ユーザが「並列」を意識せずに逐次解析コードを並列化できる。これがプラグイン機能の利用することの最大の利点であるといえる。また各ユーザ解析コードにとってオリジナル部分である剛性行列作成部(Make element matrix)および求解後の後処理部(Processing after the solution)は、GeoFEM が局所的なデータ処理を行っているため並列化プラグイン作業においてコードを変更する必要はない。次に GeoFEM がこのような並列化プラグイン機能を実現している仕組みについて説明する。

2.3 通信テーブルを含む局所化データ構造⁽⁸⁾⁽⁹⁾

2.3.1 領域分割 有限要素法を分割して並列計算を実施する場合に、基本的には単一プロセッサを使用していた場合と全く同じデータ構造を使用して、同じアルゴリズムによって計算が可能である。その際、要素、節点番号等も局所的なものが使用される。ただ、系全体としての保存則や整合性を保つための通信が必要となるため、そのためのデータが必要となる。GeoFEM では有限要素領域を分割する場合に、図 2 に示すような節点単位での分割を行い、境界上では要

Fig. 1 GeoFEM plug-in function system

Fig. 2 GeoFEM Partitioning

素がオーバーラップする構造を採用している。本研究ではオーバーラップ領域を袖領域と呼び、図2の領域分割はオーバーラップしている領域が1層のため、袖領域深さ1という。

2.3.2 局所データ構造 節点単位で領域分割された場合の局所データ情報は：

- ・その領域に割り当てられた節点
- ・割り当てられた節点を含む要素およびそれに含まれる節点すべて

を含むものとしている。さらに局所節点は、以下の3種類に分類される。

- ・内点：本来その領域に割り当てられた節点
- ・外点：本来その領域には属していないが袖領域の要素に含まれている節点(ここでは袖領域深さ1の場合を想定している)
- ・境界点：内点のうち隣接領域の外点となっている節点

節点の番号付けは、まず内点(境界点を含む)を先に、そのあと外点という順番で格納する。まず基本的なデータとして：

- NEIBPETOT 隣接 PE 数
- NEIBPE(*ipe*) 隣接 PE 番号 *ipe*=1, NEIBPE
- NODTOT PE における節点数(外点を含む)
- INTNODTOT PE における節点数(内点)

がある。図3に示した PE #0 において：

- 内点 1, 2, 3, 4, 5, 6, 7
- 外点 8, 9, 10, 11, 12, 13, 14, 15
- 境界点 2, 4, 5, 6, 7

である。

2.3.3 外点の記述 境界点は境界点ではない内点と区別する必要はないが、外点は同じ隣接領域に属する節点同士に区別する必要がある。つまり外点は隣接領域番号の順番に連続した番号付けになっている必要がある。すなわち：

Fig. 3 Local data information

STACK_IMPORT(*ipe*) *ipe*=1, NEIBPETOT
+1

STACK_IMPORT(1)=INTNODTOT+1

各隣接節点のうち最初の点の局所節点番号

STACK_IMPORT(NEIBPETOT+1)

=NODTOT+1

とした場合、図3に示す領域 PE #0 では：

NEIBPETOT=3

NEIBPE(1)=1, NEIBPE(2)=3, NEIBPE
(3)=2

NODTOT=15, INTNODTOT=7

STACK_IMPORT(1)=8, STACK_

IMPORT(2)=11

STACK_IMPORT(3)=13, STACK_
IMPORT(4)=16

STACK_IMPORT(*ipe*)~STACK_IMPORT
(*ipe*+1)-1 が領域 *ipe* に属する。すなわち：

1 番めの隣接領域(PE #1) 8, 9, 10

2 番めの隣接領域(PE #2) 11, 12

3 番めの隣接領域(PE #3) 13, 14, 15

という構成になる。

2.3.4 境界点の記述 境界点に関しては、

STACK_EXPORT(*ipe*) *ipe*=1, NEIBPETOT
+1

各隣接領域へ送られる境界点の最初の点の配列 NOD_
EXPORT へのポインタ

NOD_EXPORT(in)

境界節点のらび in=STACK_EXPORT(*ipe*)~
STACK_EXPORT(*ipe*+1)-1 の節点が *ipe* 番めの
隣接領域へ送られる。

という2種類の配列を使用する。図3に示す PE #0
では：

NEIBPETOT=3

NEIBPE(1)=1, NEIBPE(2)=3, NEIBPE
(3)=2

NODTOT=15, INTNODTOT=7

STACK_EXPORT(1)=1, STACK_EXPORT
(2)=5,

STACK_EXPORT(3)=6, STACK_EXPORT
(4)=9

となり、それぞれの隣接領域へ送られる節点 NOD_
EXPORT 0 は：

1 番めの隣接領域(PE #1) 2, 4, 6, 7

2 番めの隣接領域(PE #2) 7

3 番めの隣接領域(PE #3) 5, 6, 7

となる。

NOD_EXPORT 0 の順番は隣接領域において受信されるのと同じ順番になってなければいけない。例えば PE#2 上の内点 4, 5, 6 は PE#0 上では外点 15, 14, 13 となる。受信における(すなわち外点としての)番号付けを優先するため、この場合 PE#2 からは配列 NOD_EXPORT 0 に 6, 5, 4 という順番で格納されていなければならない。

2.4 ソルバへの通信の集約⁽⁸⁾⁽⁹⁾ 領域間の通信はこれまでに述べた境界領域の情報をもとにソルバ内にてユーザには完全に隠ぺいされた状態で実施される。その手順としては：

Step 1: 従属変数ベクトル X の中身を転送(送信)ベクトル WS に代入

Step 2: n 番めの隣接領域に対して：

istart=STACK_EXPORT($n-1$)+1

iend=STACK_EXPORT(n)

とすると、WS(istart)から WS(iend)までの連続したデータが n 番めの領域に転送される(図 4 参照)。

Step 3: Step 2 とは逆の手順で転送(受信)ベクトル WR を受け取る。 n 番めの隣接領域に対して：

istart=STACK_IMPORT($n-1$)+1

iend=STACK_IMPORT(n)

とすると、WR(istart)から WR(iend)までの連続したデータが n 番めの領域から転送される(図 5 参照)。

Spep 4: 従属変数ベクトル X に転送(受信)ベクトル

WR の中身を代入する。

3. 並列化プラグイン機能の利用

ここでは、著者がプラグインの際に行った具体的な作業(コーディング)に関して説明する。まずプラグイン作業を行う前に逐次解析コードを用意した。用意したコードは三次元有限要素法による非定常熱伝導解析コード(コード名: Thermal)を用いた。以下にそのコードの特徴およびプラグイン作業の手順を説明する。

- 三次元有限要素法

- 使用要素：六面体一次要素

- 境界条件：節点温度規定、熱伝達境界

- 定常・非定常熱伝導解析

- 陰解オイラー時間積分

- 前処理付 CG ソルバ

- Fortran 90

STEP 1 GeoFEM 形式の入力データ読み込み部接続

GeoFEM 形式の入力データ(節点座標、節点要素情報、境界条件など)を読み込むために、ユーザの解析モジュールの入力データ読み込み部分を削除し、GeoFEM でサポートしているライブラリ(GeoUtil)の入力データ読み込みルーチン(geofem_get_input_datum)を呼ぶ。そして、その読み込んだデータをユーザの解析モジュールで使用される変数にポインタ代入でデータを渡す。これにより並列データ入出力等が可能になる。

Fig. 4 Send boundary point information to NEIBPE

Fig. 5 Receive external point information from NEIBPE

Fig. 6 Flowchart of Thermal

STEP 2 並列線形ソルバの接続

ユーザの解析モジュールで呼び出していたソルバを GeoFEM の並列線形ソルバを呼ぶように引数も含めて変更する。それに伴う新しい変数の設定(配列確保)も行う。また MPI の初期化を解析モジュールの初期に行うため、以下に示す 3 行を解析部初期に挿入する。

```
call MPI_INIT(ierr)
call MPI_COMM_RANK(MPI_COMM_
  WORLD, myid, ierr)
call MPI_COMM_SIZE(MPI_COMM_
  WORLD, petot, ierr)
```

STEP 3 並列動作確認

並列線形ソルバを組み込みが終了したこの時点で並列解析が可能となるため、並列計算の動作確認を行う。並列計算用の領域分割データを作成し、並列計算を行い、逐次解析と同様の解が出ることを確認する。

STEP 4 解析結果の接続

ユーザの解析モジュールから出力される解析結果(今回は熱伝導であるため節点温度のみ)を可視化器へ渡すため可視化器(geofem_put_to_visual)を呼び出す。同時に解析結果のための変数領域やインターフェイスを作成する。GeoFEM の可視化器には「ファイル渡し」および「メモリ渡し」2 種類の選択が可能であるが、今回はファイル渡しによる可視化を選択した。ファイル渡しでは三次元可視化ツール MicroAVS⁽¹⁰⁾ の UCD データ形式の出力をサポートしている。

以上がプラグイン作業の概略である。これにより逐次解析コードが並列化されたことになる。図 6 に今回プラグインしたコード全体の流れを示す。ユーザの逐次解析コード部分は、モジュール化した analyze に相当している。これ以外のコード部分(GeoFEM プラットフォーム部分に相当)は、ほぼ他の解析モジュール

(GeoFEM ダウンロード時に付属しているプラグイン済みの線形弾性解析モジュールなどの)コードをコピーして、Thermal 用に編集したものである。まず、呼び出している(use している)各モジュールの初期化を行う。init_analyzer においては geofem_get_input_datum により入力データの読み込みを行う。そして解析部 analyze で呼び出している並列ソルバにより並列計算が実施され、その解析結果が visualizer に渡される。これが全体の流れである。プラグイン作業で生じたコード変更量を表 1 に示す。

表 1 上部にコード全体の増加量、下部に増加したコードの内訳を示す。全体としては約 3 割コード量が増加したが、前述にもあるように 559 行のコード量増加のうち、ユーザの解析部以外で増加したコードの大半は、他の解析モジュールのコードをコピーしてきて Thermal 用に編集したものである。入力データ部分の増加量が少ないが、これは逐次解析コード作成の段階で GeoFEM へのプラグインを想定し最初から GeoFEM データ構造を採用したため、少量の変更で抑えることができた。ユーザ解析部で変更した点は境界条件を負荷する箇所である。領域ごとに負荷した境界条件を区別して読み込めるように変更した。ソルバ部に関しては、逐次処理ソルバの変更作業と同程度の変更量だと思われる。

ここでは並列プラグイン機能を利用した逐次解析コードの並列化手順を説明した。プラグイン作業中に並列を意識したのは MPI 初期化に必要な 3 行を挿入した箇所だけであった(STEP 2)。また 3 割程度のコード変更量からみても、プログラミング、数値解析(有限要素法)の知識さえあれば、並列計算の知識・経験がなくても十分可能な範囲の作業であり、プラグイン機能を利用せずに逐次解析コードを並列化するのに比べ

Table 1 Plug-in work conclusion

	Total number of lines	
Original code	1,918	
Plug-in code	2,477 (+559 lines)	
Items	Rate (%)	# of lines
Control part code	32	178
User part code	11	62
Input part code	4	21
Solver part code	12	66
Visualize part code	41	232

Fig. 7 FE mesh of Cube (4 domains partitioned)

Fig. 8 FE mesh of Crank (4 domains partitioned)

効率的であると思われる。このようにユーザが「並列」を意識せずに比較的容易に逐次解析コードを並列化することができることがこの機能を利用することの最大の利点である。

4. 並列有限要素解析による 並列性能評価

4・1 問題設定 次に開発した非定常熱伝導解析コードを用いた並列有限要素解析による並列性能評価を行った。使用した解析モデルは、立方体モデル(Cube)および実用形状部品であるクランク(Crank)を用いた。4領域分割例をそれぞれ図7, 8に示す。

Table 2 Model size

	Node	Element
Cube-Large	323,680	309,741
Cube-Small	77,616	72,283
Crank-Large	323,469	279,352
Crank-Small	79,043	62,968

Table 3 Specification of used computers

	Pentium II - cluster	Alpha- cluster	SR2201
OS	Redhat- Linux-6.1	Kondara MNU/Linux	HI-UX/MPP
CPU	Pentium II -266MHz	Alpha21264 -667MHz	HARP-IE- 150MHz
Memory/PE	128MB	1GB	256MB
Network Speed	10Mbps	100Mbps	300MBps
# of PEs	4	11	1024

Table 4 Elapsed time (Cube-Small) [s]

PE	1	2	4	8
PC-cluster	89.3	48.5	29.9	N.A.
Alpha-cluster	45.2	24.2	12.8	6.4
SR2201	357.7	194.0	104.7	52.7

Table 5 Elapsed time (Crank-Small) [s]

PE	1	2	4	8
PC-cluster	190.4	94.7	50.1	N.A.
Alpha-cluster	83.5	42.6	21.4	10.9
SR2201	756.0	379.6	196.4	100.7

Table 6 Elapsed time by 4 PE analysis using Large mesh

	Cube-Large	Crank-Large
PC-cluster	162.4	4872.6
Alpha-cluster	71.8	1896.6
SR2201	594.2	17691.2
Iteration	78	3,391

解析モデルの規模はCubeとCrankでほぼ同じ規模(節点数)とし、それぞれLarge, Smallと二つの規模について並列解析を行った。その規模を表2に示す。使用ハードウェアは異なる通信性能をもつPentium II-cluster, Alpha-cluster, Hitachi-SR 2201の3機種を使用した。各計算機の主ハードウェア性能を表3に示す。また、入力データはGeoFEMから提供されているパーティショニングツール(part-GeoFEM)を使用し領域分割を行った。その際、領域分割手法にはMetis⁽¹⁾を用いた。

4・2 定常熱伝導解析 本研究では、Cube, Crankのそれぞれのモデルについて定常熱伝導解析を行った。Cubeの境界条件は、Z=0, 1の面に温度規定条件を負荷し、Crankについては、クランクとコンロッドの結合部4箇所と、クランク軸の左右両端2箇所に温度規定条件を負荷した。両解析ともに熱伝導率は一定とした。前処理およびソルバには、不完全コレスキー分解および共役こう配法(CG)を用いて解析し、収束判定条件を 10×10^{-7} とした。

4・3 並列性能評価 並列化したコードの並列性能を評価するため、定常熱伝導解析の入出力部分を除いた各処理の計算時間を測定し、諸々の並列性能評価を行った。

(a) 処理時間, スピードアップ, 並列化効率⁽⁵⁾

表4, 5に各解析の処理時間(入出力を除く)を示す。

なお、PC-cluster, 8 PEによる解析は4台という台数制限のため解析は行っていない。

また、Largeモデルを用いた解析は、PC-clusterの記憶容量・台数の関係上4 PEのみによる解析を実施した。表6にその処理時間および各モデルの反復回数

Fig. 9 Speed-up of Cube

を示す。ここで表6より、ほぼ同じ節点数の Cube, Crank を用いた解析にもかかわらず処理時間が約30倍もの差が生じているが、これは反復回数からもわかるように収束するのに多くの時間を費やしたためである。

一般に並列性能の評価には、以下の式で用いられるスピードアップ S_n 、並列化効率 P_n が用いられる。

$$S_n = \frac{T_1}{T_n}, \quad P_n = \frac{S_n}{n} \dots\dots\dots (1)$$

n : No. of PEs

T_1 : Elapsed time at 1 PE

T_n : Elapsed time at n PE

式(1)と表4, 5の処理時間より算出した Small メッシュ解析の Cube, Crank のスピードアップを図9, 10に、並列化効率を図11, 12に示す。

図9, 10より全体的に効率の著しい低下が見られず、理論値(Ideal)に近いスピードアップが得られている

Fig. 10 Speed-up of Crank

Fig. 11 Parallel efficiency of Cube

ことがわかる。また図11, 12より Crank の解析は PE 数が増加しても高い効率を維持しているのに対して、Cube の効率が低下している。なかでも、PC-cluster の効率の低下が大きい。これは Crank メッシュに比べて Cube メッシュのほうが領域境界面積が大きい(図7, 8に示す Cube および Crank の4領域分割図参照)、つまり通信量が多いため解析時間における通信時間の占める割合が多くなることが原因のひとつとして考えられる。また通信性能のカatalog値(表3参照)を PC:Alpha:SR 2201 として比較すると約1:10:300であり、PC-cluster に関してはスイッチング機能なしの HUB を用いていることからネットワークが飽和し、Cube の中でも PC-cluster の並列性能低下に影響したと思われる。

(b) CPU 利用率

次に CPU 利用率(Rate of CPU Usage)の評価を行った。これは各 PE が有効な演算を行った時間の比率

Table 7 Rate of CPU usage of Cube-Small

PE	1	2	4	8
PC-cluster	100.0	94.7	82.0	N.A.
Alpha-cluster	100.0	98.9	97.4	94.9
SR2201	100.0	99.7	99.4	98.2

Table 8 Rate of CPU usage of Crank-Small

PE	1	2	4	8
PC-cluster	100.0	96.2	89.3	N.A.
Alpha-cluster	100.0	99.3	96.0	92.8
SR2201	100.0	99.8	97.2	96.2

Table 9 Rate of CPU usage of Large model (4 PE)

	Cube-Large	Crank-Large
PC-cluster	87.0	92.2
Alpha-cluster	98.4	97.6
SR2201	99.7	99.5

Fig. 12 Parallel efficiency of Crank

であり、以下の式により定義される。

$$W_n = \frac{\sum(\text{Working Time})}{\sum\{(\text{Working Time}) + (\text{Idling Time})\}} \dots\dots\dots (2)$$

ここで $\{(\text{Working Time}) + (\text{Idling Time})\}$ は各 PE での処理時間、 (Idling Time) は通信時間と待ち時間の和としてそれぞれ計測することができる。表 7, 8 に各 Small モデルの CPU 利用率を示す。また、表 9 に Large モデルの CPU 利用率を示す。

表 7, 8 より全体的に高い効率が得られているが、通信性能の高い Alpha-cluster および SR 2201 は、PE 数が増加しても CPU 利用率が大きく低下せず、効率の良い並列計算がなされていることがわかる。表 7 の 4 PE, PC-cluster の CPU 利用率が低いのは、スピードアップ、並列化効率の評価の際述べた理由と同じで、通信量の増大によりネットワークが飽和し通信時間が増加したことが原因である。また、表 9 より同じ 4 PE, Small モデルの解析の CPU 利用率に比べて、規模の増加に伴い効率が高くなっていることもわかる。

5. 結 言

本論文では、GeoFEM プラグイン機能を利用した逐次解析コードの並列化方法と、その開発したコードの並列性能評価について述べた。開発した並列有限要素解析コードは高い並列性能が得られ、効率の良い演算が可能であることがわかった。以下に本研究の主な成果を示す。

(1) 並列計算の知識・経験を必要としない逐次解析コードを容易に並列化する一つの手法を示し、並列有限要素法解析コードを開発した。コードの増加量は逐次解析コードの約 3 割(559 行)であった。

(2) スピードアップの並列性能評価では、異なる計算機上でほぼ理論値に近い性能が得られるなどハードに左右されない高い並列性能を示した。例えば Alpha-cluster, Crank-Small 解析で、1 PE では 83.5 秒の解析が 8 PE だと 10.9 秒($S_n=7.6$)で解析が終了した。

(3) PC クラスタのような低通信性能環境下でも、CPU 利用率で 80%以上は得られるが、ネットワークの飽和により通信性能が急激に低下した。

本研究で用いた Alpha クラスタは、Compaq より東京大学 ADVENTURE プロジェクトに貸与いただいたものである。記して謝意を表する。

文 献

- (1) OpenMP: A Proposed Industry Standard API for Shared Memory Programming, (1997-10).
- (2) <http://www.crpc.rice.edu/HPFF/home.html>
- (3) MPI フォーラム, MPI: メッセージ通信インターフェイス標準(日本語訳ドラフト), (1999).
- (4) Geist, G. A., Beguelin, A. L., Dongarra, J. J., Jiang, W., Manchek, R. J. and Sunderam, V. S., "PVM: Paralled Virtual Machine-A Users Guide and Tutorial for Network Parallel Computing", (1994), MIT Press.
- (5) <http://geofem.tokyo.rist.or.jp>
- (6) 第 1 回 GeoFEM セミナー, セミナー資料, (財)高度情報科学技術研究機構(RIST), (1999).
- (7) 第 2 回 GeoFEM セミナー, セミナー資料, (財)高度情報科学技術研究機構(RIST), (2000).
- (8) 奥田洋司, GeoFEM/Tiger: 固体地球シミュレータ(第 1 回)ー並列有限要素法プラットフォームー, 計算工学, 4-3 (1999), 152-161.
- (9) 奥田洋司, GeoFEM/Tiger: 固体地球シミュレータ(第 2 回)ー並列有限要素法プラットフォームー, 計算工学, 4-4 (1999), 28-36.
- (10) MicroAVS 2.0 ユーザーズマニュアル, (1997) (株)ケー・ジー・ティー.
- (11) <http://www-users.cs.umn.edu/karypis/metis/>