

- 23 動的緩和法を用いた有限要素解析のための  
グリッドコンピューティングの適用

The Application of Grid Computing for Finite Element Analysis  
by using Dynamic Relaxation Method

竹原 和夫<sup>1</sup> 宮辻和宏<sup>2</sup> 三上 市藏<sup>3</sup> 大浜 拓也<sup>4</sup>

Kazuo TAKEHARA, Kazuhiro MIYATSUJI, Ichizou MIKAMI and Takuya OHAMA

抄録：グリッドコンピューティングとは地理的に分散したコンピュータ資源をネットワークで接続して活用することで、大規模な処理能力を実現するための技術である。著者らは研究室の LAN 内に Linux と Windows の PC を対象に、ミドルウェアとして Globus Toolkit を適用した仮想グリッドコンピューティング環境を構築してきた。本研究では収束計算に動的緩和法を適用した有限要素法による並列計算プログラムを開発し、その妥当性を検証した。次に本プログラムをグリッド環境に実装して、並列化効率を検討した。また WindowsPC をクライアントに含めた計算結果から、本手法によって WindowsPC をグリッドコンピューティングの計算資源として利用可能であることがわかった。

**Abstract** : Grid computing is innovative technology that obtains computing power by using many computers connected by network. A virtual grid computing system which consists of one Linux PC for server and several linux PCs and windows PCs for computing nodes has been constructed in our laboratory by using Globus Toolkit which is the grid middleware. In this paper, the parallel computation of finite element analysis using dynamic relaxation as iterative technique was achieved and accuracy of this method was evaluated. By using this technique, it was investigated to succeed the parallel computation in grid computing. From the numerical results, both the parallel efficiency and the availability on use of windows PC for computing nodes were discussed.

キーワード：グリッドコンピューティング, 動的緩和法, 有限要素法, Globus Toolkit, MPICH-G2  
Keywords : Grid Computing, Dynamic Relaxation Method, Finite Element Method, Globus Toolkit, MPICH-G2

1. はじめに

グリッドコンピューティング<sup>1)</sup>とはネットワークで接続された多数のコンピュータ資源を活用することで、地理的に分散したコンピュータ資源を共有して利用する技術である。近年のネットワーク技術の向上により実現可能な技術となり、HPC の新たな手法として注目されてきている<sup>2)</sup>。グリッドコンピューティングの中でもスーパーコンピュータに匹敵するような、大規模な演算処理を行うことを目的としたコンピューティンググリッドは、最も古くから研究されているグリッドコンピューティングであり<sup>3)</sup>、医療分野や生物化学の分野で多くの研究が行われている。ま

た、インターネットを介してデータを取得し、PC の遊休中に取得したデータの解析を行い、結果をインターネットでサーバへ送付するといったインターネットを利用したプロジェクトベースの研究も進められている<sup>4)</sup>。

グリッドコンピューティング環境の構築には Linux などの UNIX 系 OS をプラットフォームとするのが都合がよいが、現在の市場を考慮すると既存の WindowsPC を遊休計算資源<sup>5)</sup>として活用したい。さらに Web 閲覧や事務処理など通常の Windows における作業を中断することなく、バックグラウンドで遊休計算資源として使うことができれば、大規模な数値計算などに非常に有効な手段となる。

著者らは既に研究室の LAN 内に仮想的にグリッドコン

<sup>1</sup> 正会員 工修 JIP テクノサイエンス(株)大阪テクノセンタ 解析技術部

(〒 532-0011 大阪市淀川区西中島 2-12-11 TEL 06-6307-5462 E-mail kazuo\_takehara@cm.jip-ts.co.jp)

<sup>2</sup> 正生員 工修 みずほ情報総研(株)第一事業部 法人ビジネス第 1 部 (〒 112-0001 東京都文京区白山 5-16-6 Tel:03-5978-7092)

<sup>3</sup> フェロー 工博 関西大学 工学部都市環境工学科 教授 (〒 564-8680 大阪府吹田市山手町 3-35-35 Tel:06-6368-1121)

<sup>4</sup> 非会員 関西大学 工学部都市環境工学科

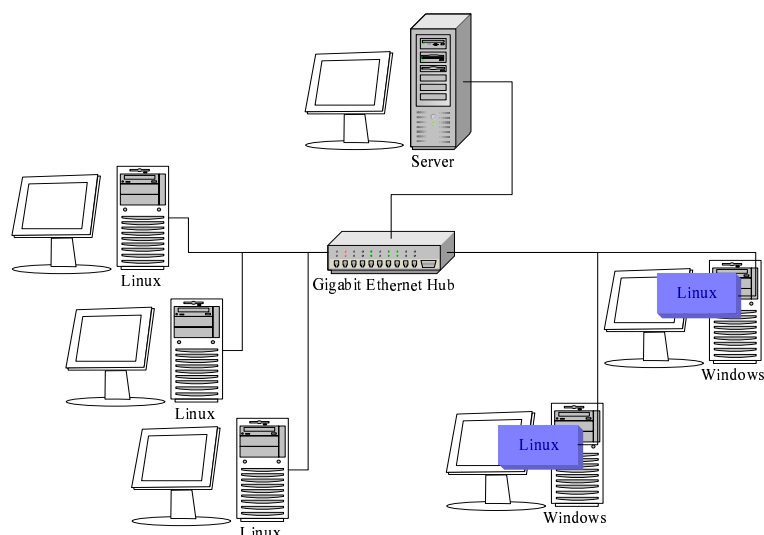


図-1 グリッドコンピューティング環境のシステム構成図

表-1 グリッドコンピューティング環境のスペック

	Server	Client	
CPU	PentiumIV 3.4GHz	PentiumIV 2.8GHz	
RAM	2GB	1GB	
HDD	160GB	80GB	
OS	Redhat Enterprise ES	Redhat Professional WS	Windows XP Professional
emulator		×	VMware Workstaion 4.5
Guest OS		×	Redhat Professional WS
Network	Gigabit Ethernet		

コンピューティングの環境を構築している<sup>6)</sup>。本研究では収束計算に動的緩和法を適用した有限要素法による並列計算プログラムを開発し、その妥当性を検証する。次に本プログラムを構築したグリッド環境に実装して、その並列化効率を検討する。さらに WindowsPC をクライアントとした場合についての試算を行い、WindowsPC のグリッドコンピューティングにおける計算資源としての適用性について検討する。

## 2. グリッドコンピューティング環境<sup>6)</sup>

### (1) ハードウェア

研究室のイントラネット内に構築した仮想グリッドコンピューティング環境のシステム構築図を図-1に、スペックを表-1に示す。認証を行うサーバに Linux を用い、5 台のクライアントのうち Linux のクライアントを 3 台、Windows のクライアントを 2 台とした。

### (2) ミドルウェア

グリッド環境を構築するために The Globus Project が開発した Globus Toolkit<sup>8)</sup>をミドルウェアとして用いた。Globus

Toolkit はグリッドコンピューティングのためのセキュリティの確保、資源管理機能、通信ライブラリなどグリッドコンピューティングの環境構築に必要な基本機能を備えており、無償で配布されている。また並列計算ツールには Globus Toolkit の管理機能やセキュリティ機能を使用して、MPI(Message Passing Interface) アプリケーションを実行させることができる MPICH-G2<sup>9)</sup>を用いた。なお Globus Toolkit のバージョンについて現在 Ver.4.0 が公開されているが、ここでは MPICH-G2 を利用できればよいだけであるので、現在の最新安定版とされているバージョン 2.4.3 を採用した。

### (3) 仮想エミュレータ

Globus Toolkit は Linux と Windows の両方に対応しているとされているが、Windows に対しては Web ベースのサービスに限定され、telnet などによるコマンドベースの実行に対応していない。そこで本研究では、Windows クライアントに仮想エミュレータ VMware Workstation を導入し、ゲスト OS として Linux を用いた仮想コンピュータをクライアントとして利用した。

異なる OS を同時に動作させる仮想化技術には coLinux などのカーネルエミュレータや PC エミュレータ、VMware

などの仮想マシンモニタの方法があるが、本研究では Globus Toolkit の GRAM(Globus Resource Allocation Manager) が動作できるように、ブリッジ接続の可能な VMware を採用した。

なお VMware では Linux クライアントで使用するメモリの容量とハードディスクの容量を前もって決定しておく必要があるため、ゲスト OS で利用する RAM の容量はゲスト OS が最適なパフォーマンスを示すことのできるように 1GB の RAM のうち 412MB をゲスト OS に割り振ることとした。

#### (4) 環境設定<sup>7)</sup>

MPICH-G2 を用いて並列計算を行うためには、まず Globus Toolkit のインストールされた複数台の計算機群を用いてグリッドコンピューティングの環境を整える必要がある。そのためにはまず CA(認証局) から個人の証明書を取得する必要がある。本研究では Globus Toolkit に添付されている Simple CA を使用してサーバ上に CA を構築し、証明書を取得しユーザの認証を行った。さらに取得したユーザ証明書とサーバ内の CA 証明書を使用し、登録の確認を行った。

#### (5) シングルサインオン

シングルサインオンとは複数の計算機に対して、一回の認証で使用可能となる仕組みである。以下のコマンドを実行し、パスフレーズを入力するとシングルサインオンを実現することができる。

```
% grid-proxy-init
```

grid-proxy-init の有効期限はデフォルトで 12 時間であるが、計算時間がこれを超える場合には正しく計算が行われない。このような場合はオプション -hours をつけてあらかじめシングルサインオンの有効期間を実行しておくといよい。例えば 48 時間に延長する場合は以下のようにする。

```
% grid-proxy-init -hours 48
```

#### (6) コンパイル

MPI によって並列化の記述を施したプログラムを並列計算プログラム用のコンパイラを用いてソースプログラムをコンパイルする。本研究における並列計算プログラムは fortran をベースとし、さらにメモリの動的割振り機能<sup>12)</sup>を用いることから、コンパイラには mpif90 を用いた。

#### (7) ジョブの実行

まずクライアントを登録するために、使用するクライアント名を示した以下のような "machines" ファイルを作成する。

```
“grid01.civil.kansai-u.ac.jp”1
“grid02.civil.kansai-u.ac.jp”2
```

ここでクライアントの後の数字は使用可能なプロセス数を示す。

Globus Toolkit を用いたグリッドコンピューティングでは RSL(Resource Specification Language)<sup>13)</sup> で記述された RSL ファイルというスクリプトを作成し、これを実行することで並列処理を行うことができる。実行モジュールを a.out、RSL ファイル名を test.rsl、使用するクライアント台数を N とすると、以下のコマンドを使用することで基本となる RSL ファイルを作成する。

```
% mpirun -np N -dumprsl a.out > test.rsl
```

すなわち Globus を用いた並列計算は、このスクリプトを以下のように実行することにより行われる。

```
% globusrun -w -f test.rsl
```

ここで、作成された RSL ファイルは記述されている設定がすべてサーバと同じになっているので、サーバとクライアントとで Globus Toolkit のインストールディレクトリが異なる場合や入出力ファイルが異なる場合はそれぞれのクライアントごとに設定を修正する必要がある。

#### (8) グリッドコンピューティングの終了

計算が終了してこれ以上グリッドにアクセスしない場合は、以下のコマンドでグリッドコンピューティングを終了する。

```
% grid-proxy-destroy
```

### 3. 数値計算手法

#### (1) 並列計算を適用した動的緩和法

要素は 8 節点 6 面体要素とし、解析対象を図-3 に示すように添字  $m$  を用いて  $\Omega_m$  と  $\Omega_{m+1}$  に分割する。節点  $n$  における  $x, y, z$  方向の変位を  $\{\delta\}_n = \{\delta_x, \delta_y, \delta_z\}_n$ 、作用外力を  $\{f_{g,ext}\}_n = \{f_x, f_y, f_z\}_n$  とする。時間は間隔  $\Delta t$  で分割する。

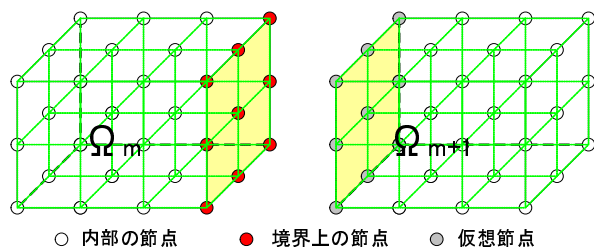


図-3 領域の分割

まず要素について有限要素法の考え方<sup>14)</sup>から要素剛性マトリクス  $[K_e]$  を求める。

$$[K_e] = \iiint [B]^T [D] [B] dV \dots\dots\dots (1)$$

時刻  $(p-1/2)\Delta t$  において  $[K_e]$  に  $\{\delta\}_n$  を乗じ各要素毎の節点力  $\{f_{e,int}\}_n$  を求める。

$$\{f_{e,int}\}_{n,p-1/2} = [K_e]\{\delta\}_{n,p-1/2} \dots\dots\dots (2)$$

次に並列計算を行うため、部分領域  $\Omega_m$  に属する要素に式 (2) を適用して、 $\Omega_m$  の節点力ベクトル  $\{f_{g,int}\}_{n,\Omega_m}$  を得る。

$$\left\{ \begin{array}{l} \{f_{g,int}\}_{n,p-1/2} \Big|_{\Omega_m} \\ \{f_{g,int}\}_{n,bou,p-1/2} \Big|_{\Omega_m} \end{array} \right\} = \sum_{\Omega_m} \{f_{e,int}\}_{n,p-1/2} \dots\dots\dots (3)$$

これは  $\Omega_m$  において要素剛性行列を重ね合わせることと同義である。ここで添字  $n, bou$  は、境界上の節点を表す。

続いて  $\Omega_m$  に隣接する部分領域  $\Omega_{m+1}$  について考える。 $\Omega_{m+1}$  についても  $\Omega_m$  と同様に式 (3) を適用すればよいが、 $\Omega_m$  との境界部の節点に対応する節点力ベクトルは既に  $\Omega_m$  で使用されているので、 $\Omega_{m+1}$  の計算でこれらを利用することができない。そこで図-3 に示すように境界上に仮想節点を生成し、 $\Omega_{m+1}$  側の繰返計算のために用いる。すなわち剛性行列は既存の要素構成節点から算出され、領域分割を考慮した繰返計算は仮想節点を含めた節点で行われることになる。ここで  $\Omega_{m+1}$  の節点力ベクトル  $\{f_{g,int}\}_{n,\Omega_{m+1}}$  は次式のように表される。

$$\left\{ \begin{array}{l} \{f_{g,int}\}_{n,p-1/2} \Big|_{\Omega_{m+1}} \\ \{f_{g,int}\}_{n,dblc,p-1/2} \Big|_{\Omega_{m+1}} \end{array} \right\} = \sum_{\Omega_{m+1}} \{f_{e,int}\}_{n,p-1/2} \dots\dots\dots (4)$$

ここで添字  $n, dble$  は境界上の仮想節点を表す。

次に節点  $n$  のつりあいに慣性項と減衰項を付加すると、以下の動的つりあい式が得られる。これは部分領域  $\Omega_m, \Omega_{m+1}$  でそれぞれ成り立つ。

$$\left[ m \frac{\partial \{\delta\}_{n,p-1/2}}{\partial t} + C \{\delta\}_{n,p-1/2} + \{f_{g,int}\}_{n,p-1/2} = \{f_{g,ext}\}_{n,p-1/2} \right]_{\Omega_m, \Omega_{m+1}} \dots\dots\dots (5)$$

$m$  は質量、 $C$  は減衰係数である。ここで式 (5) の偏微分項を時刻  $(p-1/2)\Delta t$  において差分間隔  $\Delta t/2$  で中央差分表示すると、次式が得られる。

$$\frac{\partial \{\delta\}_{n,p-1/2}}{\partial t} = \frac{\{\delta\}_{n,p} - \{\delta\}_{n,p-1}}{\Delta t} \dots\dots\dots (6)$$

また時刻  $(p-1/2)\Delta t$  における変位速度  $\{\dot{\delta}\}_n$  を、時刻  $p\Delta t$  と  $(p-1)\Delta t$  での変位速度を用いて次式のように表す。

$$\{\dot{\delta}\}_{n,p-1/2} = \frac{\{\delta\}_{n,p} + \{\delta\}_{n,p-1}}{2} \dots\dots\dots (7)$$

式 (6) と式 (7) を式 (5) に代入して整理すると、次の運動方程式が得られる。

$$\left[ \{\dot{\delta}\}_{n,p} = \frac{1}{1 + 0.5C \frac{\Delta t}{m}} \cdot \left[ (1 - 0.5C \frac{\Delta t}{m}) \{\dot{\delta}\}_{n,p-1/2} + \frac{\Delta t}{m} (\{f_{g,ext}\}_{n,p-1/2} - \{f_{g,int}\}_{n,p-1/2}) \right] \right]_{\Omega_m, \Omega_{m+1}} \dots\dots\dots (8)$$

ここに  $m$  は質量、 $C$  は減衰係数である。

次に結合方程式を求める。変位速度  $\{\dot{\delta}\}_n$  を時刻  $p$  において中央差分表示すると次式のようなになる。

$$\{\dot{\delta}\}_{n,p} = \frac{\partial \{\delta\}_{n,p}}{\partial t} = \frac{\{\delta\}_{n,p+1/2} - \{\delta\}_{n,p-1/2}}{\Delta t} \dots\dots\dots (9)$$

式 (9) を変形すると次式が求まる。これも  $\Omega_m, \Omega_{m+1}$  でそれぞれ成り立つ。

$$\left[ \{\delta\}_{n,p+1/2} = \{\delta\}_{n,p-1/2} + \Delta t \{\dot{\delta}\}_{n,p} \right]_{\Omega_m, \Omega_{m+1}} \dots\dots\dots (10)$$

境界条件について、拘束されている自由度に関する変位および変位速度は、反復計算の過程において次式のように常に 0 とする。

$$\{\delta\}_{n,p} = \{\dot{\delta}\}_{n,p} = 0 \dots\dots\dots (11)$$

最後に境界上の節点変位  $\{\delta\}_{n,bou}$  と  $\{\delta\}_{n,dblc}$  の更新を行う。更新は境界上の節点変位のみを通信し、両者の平均を用いる方法とする。

$$\{\delta\}_{n,p+1/2} = \frac{\{\delta\}_{n,bou,\Omega_m} + \{\delta\}_{n,dblc,\Omega_{m+1}}}{2} \dots\dots\dots (12)$$

計算は式 (2) (3) (4) (8) (10) (11) (12) を繰返し、解が収束するまで行われる。

## (2) 数値計算手順

数値計算手順を以下に示す。

- 1) 荷重として力もしくは強制変位を与える。
- 2) 時刻 0 における変位  $\{\delta\}_n$ 、節点力ベクトル  $\{f_{g,int}\}_n$  と全変位  $\{\delta\}_n$  を仮定する。
- 3) 式 (1) を用いて要素剛性行列を作成する。
- 4) 式 (2) を用いて要素毎の節点力ベクトルを作成する。
- 5) 式 (4) を用いて、各部分領域の節点力ベクトルを作成する。
- 6) 式 (8) を用いて  $\Delta t$  後の変位速度を計算する。
- 7) 式 (10) を用いて  $\Delta t$  後の変位を計算する。
- 8) 式 (11) を用いて境界条件の更新を行う。
- 9) 式 (12) を用いて境界部の通信を行う。
- 10) 得られた変位とその前に求めた変位の差が、許容誤差  $\Delta \epsilon_d$  以内に収まるまで 3) ~ 9) を繰り返す。ただし  $\Delta \epsilon_d$  は収束判定のための許容値であり、 $x, y, z$  各方向それぞれ

それぞれに対して行う．ここでは最大相対誤差が  $\Delta\epsilon_d$  以内に収まった場合を収束したものとする．

- 11) 許容誤差  $\Delta\epsilon_d$  以内に収まったならば，変位，反力，ひずみ，応力および主ひずみ，主応力を計算し，出力する．

mpi サブルーチン<sup>12)</sup>については，式 (10) で各部分領域の節点変位が求まった後，mpi の 1 対 1 非ブロッキング通信サブルーチン mpi\_isend, mpi\_irecv, mpi\_wait により節点変位の通信を行い，式 (12) により変位の更新を行う．

#### 4. 解析プログラムの検証

##### (1) 弾性解析における結果の検証

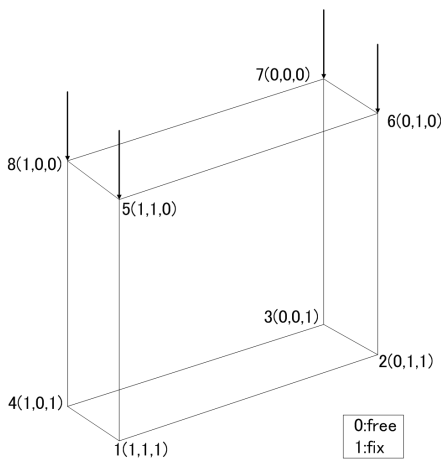


図-4 テストモデル

図-4 に示す 1 要素モデルで線形弾性解析を行って，解析プログラムの検証を行った．モデルの寸法は文献<sup>15)</sup>に準じて  $200 \times 200 \times 50$  とし，図-4 に示す対称条件を適用して  $1/8$  領域をモデル化した．静弾性係数は  $3.1e4(N/mm^2)$ ，ポアソン比は  $0.167$  とし，1 軸圧縮状態になるように荷重を設定した．数値積分は  $2 \times 2 \times 2$  の gauss 積分とした．

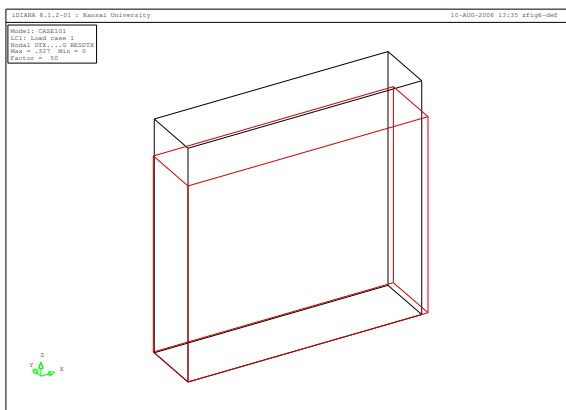


図-5 汎用プログラム<sup>16)</sup>による変形図

図-5 に汎用プログラム DIANA<sup>16)</sup>による変形図を示す．分布荷重  $100(N/mm^2)$  を要素の上面から作用させたときの載荷点における鉛直方向変位  $\delta_z$  は  $-3.226e-1(mm)$  である．

##### (2) 減衰係数

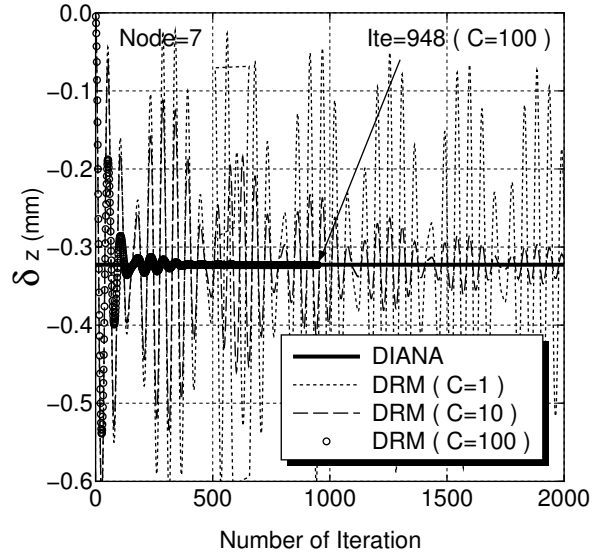


図-6 減衰係数の違いによる解の収束 (その 1)

本解法において反復回数と載荷点の鉛直方向変位の関係を図-6 に示す． $m$  は 1， $\Delta t$  は  $1.0e-4$ ， $\Delta\epsilon_d$  を  $1.0e-6$  と仮定して，減衰係数を 1 から 100 まで変化させた．図より減衰係数を小さくすると振動が生じ，収束するまでに多くの反復回数を必要とすることがわかる．ここでは減衰係数が 100 の場合において最も早く収束しており，その回数は 948 である．

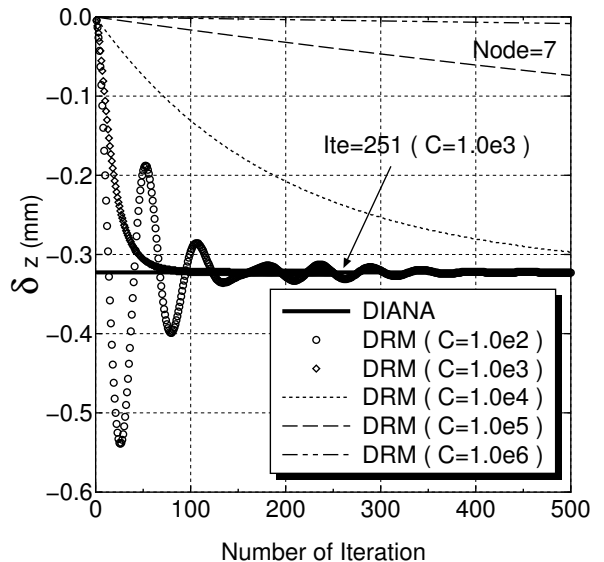


図-7 減衰係数の違いによる解の収束 (その 2)

図-7 に減衰係数が図-6 よりも比較的大きい場合を示す．図より減衰係数を大きくすると振動しなくなり，多くの反復回数を必要とすることがわかる．ここでは減衰係数が 1000 の場合において最も早く収束しており，その回数は

251 である。

これらのことから解を効率よく得るためには、適切な減衰係数を設定する必要があるといえる。

(3) 時間間隔

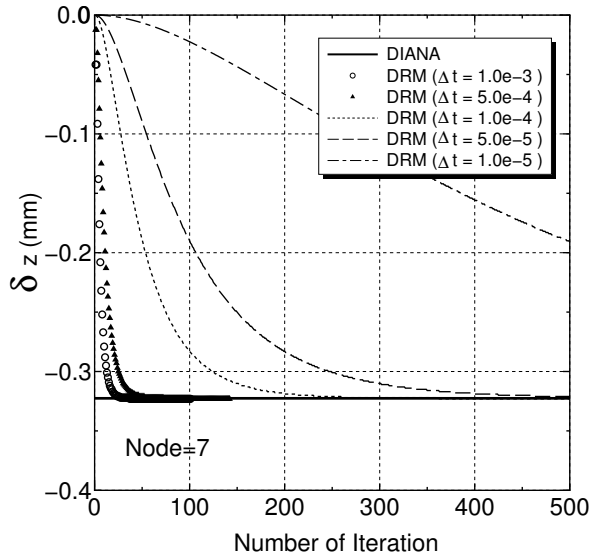


図-8 時間間隔の違いによる反復回数と z 軸方向変位

次に時間間隔  $\Delta t$  が解の収束に及ぼす影響を検討する。図-8 に反復回数と載荷点の鉛直方向変位の関係を示す。ここで時間間隔  $\Delta t$  を  $1.0e-3(s)$  から  $1.0e-5$  まで変化させた。図より時間間隔を大きい値とすれば収束に至るまでの反復回数が少なくなることがわかる。逆に小さい値の場合は収束までに多大な反復回数が必要となり、解析時間の長大化をもたらす。また、極端に大きい時間間隔は解の発散を生じさせる場合もあるので、解析を有効に行うためには適当な時間間隔を決定して用いる必要がある。

(4) 収束判定値

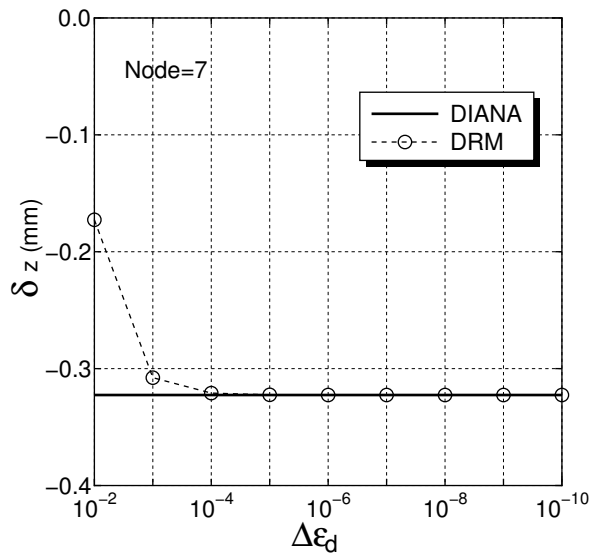


図-9 収束判定値と z 軸方向変位

図-9 に収束判定値  $\Delta \epsilon_d$  と載荷点の鉛直方向の変位の関係を示す。収束判定値は  $\Delta \epsilon_d = 1.0e-2$  から  $\Delta \epsilon_d = 1.0e-10$  まで変化させた。図より  $\Delta \epsilon_d$  が大きい場合は正しい解が得られているとはいえ、適切な許容値は  $1.0e-5$  よりも小さくすべきであることがわかる。

5. 数値解析結果

(1) 解析モデル

実構造物への適用例を示す。図-10 に示す解析モデルについて、寸法は長さ  $L=79000$ 、高さ  $h=3500$  で、高さ方向には 10 分割とした。節点数は 1760、要素数は 790 で、簡単のため奥行き方向には 1 要素とした。物性値および動的緩和法のパラメータは 4 と同じとした。境界条件および荷重条件は図-10 に示す通りで、荷重は変位制御とした。並列計算のための領域分割は長手方向に分割する方法とした。

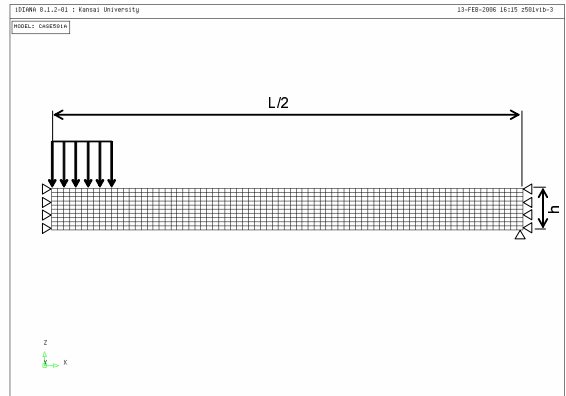


図-10 解析モデル

(2) 解の検証

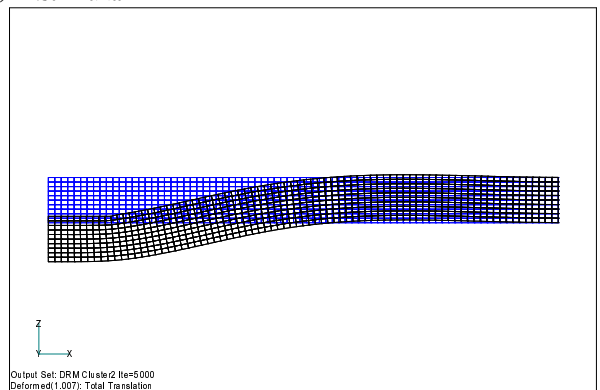


図-11 本解析手法による反復回数 5000 時の変形図

図-11 に本解析手法による反復回数 5000 の変形図を示すが、荷重載荷位置近傍はほとんど変位していないことがわかる。また図-11 からさらに反復を重ねていくと図-12 の変形図となり、全ての節点で図-13 の DIANA<sup>16)</sup> の変形図と一致することを確認した。なおこのときの反復回数は

87791であった。

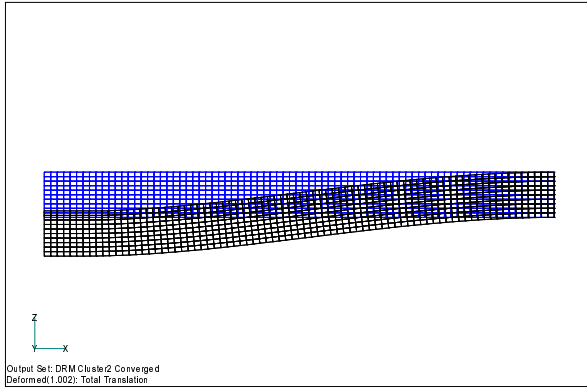


図-12 本解析手法による収束時の変形図

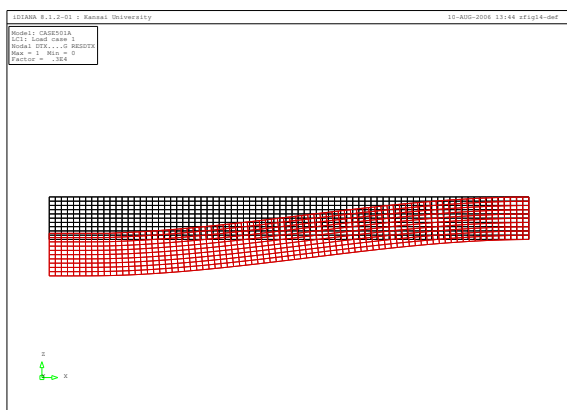


図-13 DIANA による変形図

### (3) 並列化効率

グリッド環境においてLinuxをクライアントにした場合の台数と解析時間の関係を図-14に示す。参考のため図-14に同じ条件のもとでSCore型PCクラスタ型並列計算機による結果も併記した。SCore型PCクラスタ型並列計算機はシステムソフトウェアにSCore5.6.1を採用し、スペックは表-2に示したものをを使用した。

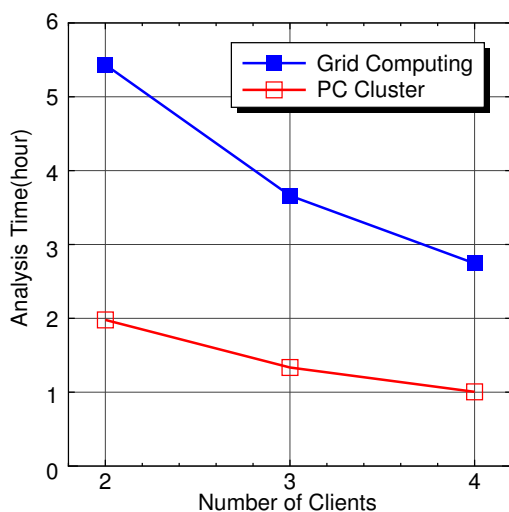


図-14 解析時間 (その1)

表-2 PC クラスタ スペック

	Server	Client
CPU	PentiumIII(1.16GHz)	PentiumIII(1.00GHz)
RAM	2GB	1GB
HDD	70GB	10GB
OS	Redhat Linux 7.3	
Network	Ethernet 100MBase TX	

図-14よりグリッド環境においても台数増加に伴って解析時間が短縮されていることがわかる。またここでのグリッドはLAN環境下で行われるため、専用通信回線を用いるPCクラスタに比べ解析時間が長くなっている。

### (4) Windows OS への適用

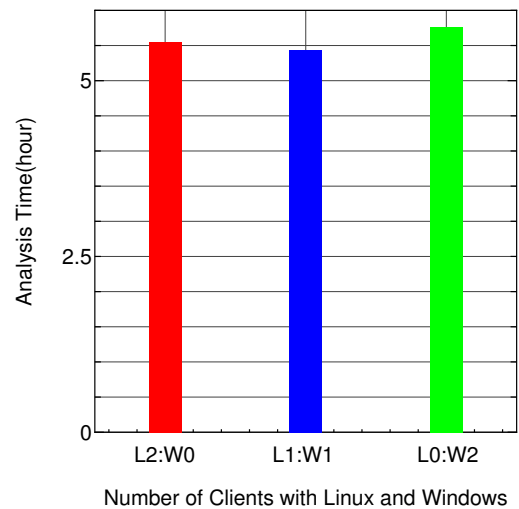


図-15 解析時間 (その2)

図-15にLinuxとWindowsのクライアントを用いて、2台の各組合せにおける解析時間を示す。図よりどのような組合せでもほとんど解析時間に差はみられないことがわかる。これより、本手法によればWindowsをクライアントに用いても計算資源として利用できそうである。

次にMS-WordなどWindowsアプリケーションの実行中において、本解析プログラムによる解析時間への影響を検討する。図-16はいずれも2台のWindowsクライアントによる解析結果であるが、Case1は図-15の解析時間を、Case2はWindowsアプリケーション実行中における解析時間を示す。図よりCase2はCase1に比べ解析時間が約1.25倍になっていることがわかる。これはVMwareメモリ量やWindowsアプリケーションの内容にも依存するが、並列プログラム実行中のWindows使用は、解析時間の長大化を招く可能性があることを考慮しておく必要がある。

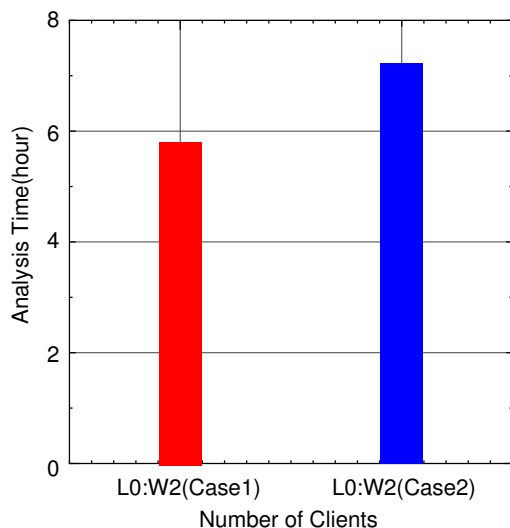


図-16 解析時間 (その3)

## 6. あとがき

Globus Toolkit を用いたグリッドコンピューティング環境に、動的緩和法を適用した有限要素法による並列計算プログラムを実装した。本手法によると WindowsPC をグリッドコンピューティングの計算資源として利用可能であることがわかった。またグリッドコンピューティングにおいてはクライアント停止などの危機管理の対応が懸念されるが、これについては今後の検討課題とする。

### 参考文献

- 1) 日本アイ・ピー・エムシステムズエンジニアリング (株) : グリッドコンピューティングとは何か, Globus Toolkit ではじめるグリッドの基礎, ソフトバンクパブリッシング, 2004.4.
- 2) 土木工学における計算力学手法の新展開 (SACOMCE) 講習会テキスト, 土木学会応用力学委員会, 同計算力学小委員会, 2005.8.
- 3) 溝口文雄 : グリッドコンピューティング 情報処理の新しい基盤技術, 岩波書店, 2005.1.

- 4) 伊達進, 下條真司, 奥村利幸, 松田秀雄, 秋山豊和, 中村春木 : バイオグリッドプロジェクト「スーパーコンピュータネットワークの構築」, 情報処理, 情報処理学会, Vol.44 No.6, pp601-607, 2003.6.
- 5) 柴田良一, 林憲一, 山方和昭, 澁谷寿夫 : Sun ONE Grid Engine と KNOPPIX を用いた教育用 PC 群による大規模グリッド構築に関する基礎的研究, 情報処理学会第 66 回全国大会, 情報処理学会, pp.69-72, 2004.3.
- 6) 三上市藏, 宮辻和宏, 竹原和夫, 丹羽量久 : 大規模 FEM 解析を目指した Linux と Windows 混在環境におけるグリッドコンピューティング, 土木情報利用技術講演集 Vol.30, pp57-60, 2005.10.
- 7) 西村直志 : グリッドコンピューティングと並列計算, 計算工学チュートリアル, 日本計算工学会, Vol.10 No.3, 2005.7.
- 8) The Globus Alliance  
<http://www.globus.org/>
- 9) MPICH-G2  
<http://www3.niu.edu/mpi/>
- 10) MPICH Home Page  
<http://www-unix.mcs.anl.gov/mpi/mpich/>
- 11) The Message Passing Interface (MPI) standard  
<http://www-unix.mcs.anl.gov/mpi/>
- 12) 青山幸也 : 並列プログラミング入門 MPI 版, 理化学研究所 情報基盤センター, 2004.3.
- 13) The Globus Resource Specification Language RSL  
[http://www-fp.globus.org/gram/rsl\\_spec1.html](http://www-fp.globus.org/gram/rsl_spec1.html)
- 14) 鷲津久一郎, 宮本博, 山田嘉昭, 山本善之, 川井忠彦 : 有限要素法ハンドブック I 基礎編, 培風館, 1981.9.
- 15) W.F.Chen : Plasticity in Reinforced Concrete, McGraw-Hill, 1982.
- 16) TNO Building and Construction Research, DIANA User's Manual Release 8.1, 2002.

(2006.5.19受付)