

ADVENTUREシステムを用いた1億自由度規模 構造解析の実用性

塩谷隆二* , 荻野正雄†

1 緒言

CPUの演算処理速度向上やマルチコア化,及び搭載メインメモリサイズの大規模化によって,大規模自由度モデルを用いた計算機シミュレーションに対する需要が高まってきている.これに対し,著者らは大規模解析を目的とした汎用並列有限要素法解析システムADVENTURE[1]の開発を進めてきた.その成果はオープンソースソフトウェアとして無償公開しており,多様な並列計算機環境で広く利用されてきている.特に,3次元弾塑性解析モジュールADVENTURE.Solidは,高い並列効率を得られる階層型領域分割法(Hierarchical Domain Decomposition Method: HDDM) [2][3]を採用しており,1億自由度規模の解析実績を持つ[2][4].また,実用的な大規模解析として,2億自由度規模モデルの動弾性解析にも成功している[5].しかし,1億自由度を超える大規模解析は1,000プロセッサ以上で構成された地球シミュレータなどのスパコン上においての成果であり,一般ユーザが容易に利用できる成果ではない.そこで,ADVENTUREシステムを汎用計算機Intel Itanium2のクラスターで構成されたFujitsu PRIMEQUEST 580に移植することで,1億自由度規模モデル解析の実現性と実用性について検証する.

大規模問題に対しては共役勾配(Conjugate Gradient: CG)法などの反復法を用いるのが一般的であるが,大規模問題は悪条件問題となりやすいために反復法による求解が困難となる.よって,安定かつ高速な収束性を持つ前処理法が重要となる.著者らはこれまで,マルチグリッド的な前処理法を持つバランシング領域分割(Balancing Domain Decomposition: BDD)法[6]に関する研究を進めてきた[7][8].BDD法は高い収束性を持ち,悪条件問題に対しても有効な手法である.その前処理は,Neumann-Neumann前処理[9]に対してコースグリッド修正(Coarse grid correction)を加えたアルゴリズムとなっている.さらに,大規模問題におけるBDD法の並列効率を改善するために,著者らによってコースグリッド修正を近似的に行う不完全バランシング領域分割(Incomplete Balancing Domain Decomposition with DIAGonal-scaling: IBDD-DIAG)法が提案された[10].IBDD-DIAG法はBDD法と同様にスケーラブルな手法であることが示されている[11].

本稿では,2章で領域分割法(Domain Decomposition Method: DDM)を述べ,3章でIBDD-DIAG法の前処理アルゴリズムを述べる.4章ではIBDD-DIAG法に必要な演算量を推測し,5章において1億自由度規模モデルを用いた数値実験例を示す.

*東洋大学工学部, E-mail: shioya@toyonet.toyo.ac.jp

†九州大学大学院工学研究院, E-mail: ogino@mech.kyushu-u.ac.jp

2 領域分割法 (DDM)

IBDD-DIAG 法は DDM に対する前処理として表現でき, DDM は解析領域を重ねりのない部分領域に分割し, 静的縮約を行った領域間境界自由度を反復法で解くアルゴリズムを基本としている. 次の連立一次方程式を考える.

$$Ku = f \quad (1)$$

ただし, K は剛性行列, u は未知変数ベクトル, 及び f は外力ベクトルである. 次に, 解析領域 Ω を重ねりのない N 個の部分領域 $\Omega_1, \dots, \Omega_N$ に分割し, 領域間境界を $\Gamma = \cup_{i=1}^N \partial\Omega_i$ とする. ここで, R_i を全体領域から部分領域 Ω_i に自由度を制限する 0-1 行列とする. このとき, u_i を Ω_i における未知ベクトルとすると, $u_i = R_i u$ により得られる. 剛性行列 K も同様に, $K = \sum_{i=1}^N R_i^T K_i R_i$ となる.

次に, u_i を領域内部自由度に関する u_{Ii} と領域間境界上自由度に関する u_{Bi} に分ける. これにより, 領域 Ω_i に関する K_i , ローカル右辺ベクトル f_i , R_i も同様に次式で表される.

$$u_i = \begin{bmatrix} u_{Ii} \\ u_{Bi} \end{bmatrix}, \quad K_i = \begin{bmatrix} K_{IIi} & K_{IBi} \\ K_{IBi}^T & K_{BBi} \end{bmatrix}, \quad f_i = \begin{bmatrix} f_{Ii} \\ f_{Bi} \end{bmatrix}, \quad R_i = \begin{bmatrix} R_{Ii} & 0 \\ 0 & R_{Bi} \end{bmatrix} \quad (2)$$

これより, 式 (1) は次のように表される.

$$\begin{bmatrix} K_{II1} & 0 & \cdots & 0 & K_{IB1}R_{B1} \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & K_{IIN} & K_{IBN}R_{BN} \\ R_{B1}^T K_{IB1}^T & \cdots & \cdots & R_{BN}^T K_{IBN}^T & K_{BB} \end{bmatrix} \begin{bmatrix} u_{I1} \\ \vdots \\ \vdots \\ u_{IN} \\ u_B \end{bmatrix} = \begin{bmatrix} f_{I1} \\ \vdots \\ \vdots \\ f_{IN} \\ f_B \end{bmatrix} \quad (3)$$

ただし,

$$K_{BB} = \sum_{i=1}^N R_{Bi}^T K_{BBi} R_{Bi}, \quad f_B = \sum_{i=1}^N R_{Bi}^T f_{Bi} \quad (4)$$

式 (3) より, 部分領域内部の自由度を静的縮約することで部分領域間自由度に関する次式が得られる.

$$Su_B = g \quad (5)$$

ただし,

$$S = \sum_{i=1}^N R_{Bi}^T S_i R_{Bi}, \quad S_i = K_{BBi} - K_{IBi}^T K_{IIi}^{-1} K_{IBi} \quad (6)$$

$$g = \sum_{i=1}^N R_{Bi}^T (f_{Bi} - K_{IBi}^T K_{IIi}^{-1} f_{Ii}) \quad (7)$$

であり, S はシュアコンプリメント行列と呼ばれる. 式 (5) を解くことで, 領域間境界上自由度の解 u_B が得られる. 領域内部自由度の解については, u_B を用いて次式により得られる.

$$u_{Ii} = K_{IIi}^{-1} (f_{Ii} - K_{IBi} R_{Bi} u_B), \quad i = 1, \dots, N \quad (8)$$

本稿で取り扱う構造解析では一般的に S は対称正定値となるため、式 (5) に前処理付き共役勾配 (Preconditioned Conjugate Gradient: PCG) 法を適用する。

ここで、PCG 法のアルゴリズムでは係数行列とベクトルの積を求める必要があるが、シュアコンプリメント行列 S は式 (6) のような形をしていることから、 S を陽に作成するのは困難となる。しかし、

$$y = Sp \quad (9)$$

に対して、

$$\begin{bmatrix} x_i \\ \cdot \end{bmatrix} = \begin{bmatrix} K_{IIi} & K_{IBi} \\ K_{IBi}^T & K_{BBi} \end{bmatrix} \begin{bmatrix} 0 \\ -R_{Bi}p \end{bmatrix}, i = 1, \dots, N \quad (10)$$

$$\begin{bmatrix} w_i \\ \cdot \end{bmatrix} = \begin{bmatrix} K_{IIi} & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} x_i \\ \cdot \end{bmatrix}, i = 1, \dots, N \quad (11)$$

$$\begin{bmatrix} \cdot \\ y_i \end{bmatrix} = \begin{bmatrix} K_{IIi} & K_{IBi} \\ K_{IBi}^T & K_{BBi} \end{bmatrix} \begin{bmatrix} w_i \\ R_{Bi}p \end{bmatrix}, i = 1, \dots, N \quad (12)$$

$$y = \sum_{i=1}^N R_{Bi}^T y_i \quad (13)$$

は等価な処理であるため、 S を陽に作成することなく、同様の演算結果を得ることが可能である。

3 不完全ランシング領域分割 (IBDD-DIAG) 法

式 (5) に対する IBDD-DIAG 法の前処理行列 M は次式で表される [10]。

$$M^{-1} = \tilde{Q} + (I - \tilde{Q}S) T_{DIAG} (I - S\tilde{Q}) \quad (14)$$

ただし、 T_{DIAG} は対角スケーリング前処理、および R_0 は領域分割の重みを考慮したコーススペースへの自由度制限行列である。また、 $\tilde{Q} = R_0^T \tilde{S}_0^{-1} R_0$ 、 \tilde{S}_0^{-1} は不完全ランシングオペレータであり、 $S_0 = R_0 S R_0^T$ で表されるコース行列と呼ばれる対称行列を不完全コレスキー分解したものである。Mandel による BDD 法 [6] と比較すると、Neumann-Neumann 前処理の代わりに対角スケーリング前処理、コースグリッド修正を不完全に実施することが異なっている。なお、コースグリッド修正を行うコーススペースについては、解析すべき対象によって異なるものとなるが、3次元静弾性問題においては剛体運動を用いて構築する [7]。このとき、全ての部分領域が剛体モード 6 自由度を有しているとする、コーススペースは $6N$ の自由度を持つこととなる。

ここで、式 (14) の実装方法について考える。シュアコンプリメント行列 S については、式 (10)-(13) によって計算可能である。一方、不完全ランシングオペレータ \tilde{S}_0^{-1} については、次元数 $6N$ である次式の近似解として計算可能である。

$$S_0 d_0 = R_0 \tilde{r} \quad (15)$$

\tilde{r} は領域間境界上のベクトルである。式 (15) はコース問題と呼ばれ、IBDD-DIAG 法では不完全コレスキー分解を行った \tilde{S}_0^{-1} を用いて d_0 を近似することとなる。

4 IBDD-DIAG 法の実装

本システムは HDDM[2] に基づく負荷分散を行う。HDDM では、部分領域群 $\Omega_1, \dots, \Omega_N$ に 2 段階の階層構造を持たせることでロバストな負荷分散を実現している。親階層は部分 (Part) と呼ばれ、解析領域を並列計算機に応じた負荷分散となるよう領域分割を行う。次に、子階層は部分領域 (Subdomain) であり、DDM が計算機環境に応じて持つ最適領域分割数となるように各部分をさらに領域分割する。よって、各計算機は最初に計算機性能に応じて静的に負荷分散された複数の部分領域を担当することとなる。ホモジニアスな並列計算機環境において、モデル自由度数 nf 、最適領域分割数 N_{dom} 、使用プロセッサ数 N_p で解析する場合、各プロセッサには (N_{dom}/N_p) 個の部分領域が割り当てられ、各部分領域は (nf/N_{dom}) の自由度を持つこととなる。

ここで、IBDD-DIAG 法において重要となる 3 つの処理について考える。1 つ目は、式 (10) 及び式 (12) に表れるローカル剛性行列 K_i とのベクトル積である。対称性とスパース性を考慮した次元数 (nf/N_{dom}) の行列-ベクトル積の計算量が必要となる。またそれは、PCG 法反復 1 回につき 6 回必要である。

次に 2 つ目は、式 (11) 及び式 (8) に表れる $K_{II_i}^{-1}$ の演算である。次元数は (nf/N_{dom}) に比べて小さいものとなるが、対称スパース行列に対するコレスキー分解、前進代入および後退代入の計算量が必要となる。コレスキー分解は PCG 法反復 1 回目に 1 回、前進代入および後退代入は PCG 反復 1 回につき 3 回である。

次に 3 つ目は、式 (15) で表されるコース問題である。次元数 $6N_{dom}$ の対称スパース行列に対する並列不完全コレスキー分解の計算量が必要となる。本稿では、HDDM の親階層負荷分散に基づく列ブロック分割を行い、フィルインの棄却ルールは並列効率を考慮して分割された列ブロックの最初のいくつかを棄却するものとした。計算量は、PCG 法反復 1 回目に対称スパース行列に対する不完全コレスキー分解を 1 回、PCG 反復 1 回につき前進代入および後退代入を 2 回である。

よって、これらの計算量と、 K_i 、 $K_{II_i}^{-1}$ 、および \tilde{S}_0^{-1} をメインメモリ上に記憶することとなる。

5 数値実験例

1 億自由度規模モデルの実用性の検証を行う。Fig.1 に示す簡易原子炉压力容器モデルに対し、境界条件として底面を固定し、水平方向に自重による体積力を与えた。4 面体 2 次要素を用いて有限要素メッシュを作成し、Table 1 より、総自由度数は 1 億自由度規模となった。IBDD-DIAG 法による静弾性応力解析ソルバとして ADVENTURE_Solid Ver.1.2 を使い、階層型領域分割には ADVENTURE_Metis Ver.1.1 を用いた。使用計算機は九州大学情報基盤研究開発センターの Fujitsu PRIMEQUEST 580 (Intel Itanium2 1.6GHz dual core) である。

比較のために、IBDD-DIAG 法と対角スケーリング前処理 (DIAG) の収束性を Fig.2 に示す。図は相対残差ノルムが 3 桁まで小さくなった時点を示しているが、対角スケーリング前処理では収束の見込みが得られないのに対して、IBDD-DIAG 法を用いることで 461 回で収束解が得ることに成功した。これより、大規模解析に対する IBDD-DIAG 法の有用性が示された。

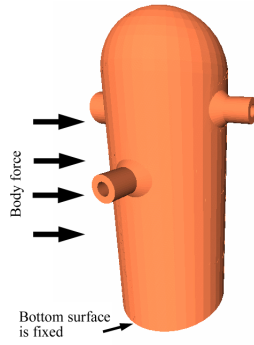


Fig. 1: A simplified pressure vessel model with 100 million DOFs mesh.

Table 1: The mesh information of 100 million DOFs model.

Elements	25,083,456
Nodes	34,772,634
DOFs	104,195,502

次に、IBDD-DIAG法を用いた1億自由度規模モデル解析の性能をTable 2に示す。表において、スピードアップ率はコア数192を基準とした速度比を表す。領域分割数は経験に基づいて準最適な値となるように決定している。また、反復回数が使用コア数によって変化しているのは、コース行列 S_0 の並列不完全コレスキー分解過程において、フィルイン棄却ルールが並列プロセッサ数に依存しているため、式(15)の近似精度が異なるためである。Table 2より、本システムは70%以上の高い並列効率が得られており、512コア使用時において約41分と実用時間内に解析を終えることに成功した。また、192コア使用、つまり3ノードの使用においても約85分で解析に成功しており、本システムの有用性が示された。

一方、192コア使用時には1コア辺りの使用メモリサイズが736MBに達し、さらに少ないコア数では1億自由度規模の解析を行うことができなかった。4章で示した通り、IBDD-DIAG法では K_i 、 $K_{II_i}^{-1}$ 、および \tilde{S}_0^{-1} をメインメモリ上に記憶している。前処理法に起因する \tilde{S}_0^{-1} に要したメモリサイズは各コアにつき50-200MB程度であり、改善の余地はあるものの、本来のDDMアルゴリズムに起因する K_i および $K_{II_i}^{-1}$ によるところが大きいことが分かる。よって、計算時間を大きく損なうことなく、式(10)や式(11)の演算を省メモリで実現することで、1億自由度規模モデルの実用性をさらに高めることが可能になると言える。

Table 2: Runtime performances of IBDD-DIAG method on the PRIMEQUEST 580.

Cores	Subdomains	Iter.	Time(sec.)	Speed-up	Memory(GB)	Mem./core(MB)
192	173,760	461	5,108	1.00	138	736
256	173,824	528	4,041	1.26	135	540
512	174,080	613	2,471	2.07	132	264

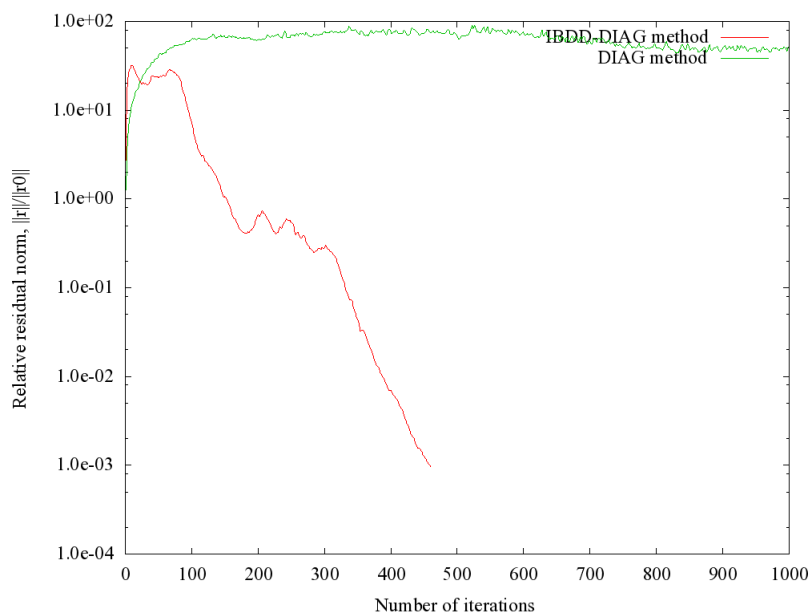


Fig. 2: CG convergence history of 100 million DOFs model.

6 結言

大規模問題に対して有効な IBDD-DIAG 法を汎用計算機で構築された Fujitsu PRIME-QUEST 580 に実装し、1 億自由度規模モデルの静弾性応力解析を行った。コア数 192、ノード数 3 の構成によって、約 85 分で解析に成功し、汎用並列計算機上においても 1 億自由度規模モデルの実用性が高いことが示された。

参考文献

- [1] ADVENTURE Project, (<http://adventure.q.t.u-tokyo.ac.jp/>), (accessed 2008-04-12)
- [2] 塩谷隆二, 矢川元基 : 階層型領域分割法による 1 億自由度並列有限要素解析, 計算工学講演会論文集, Vol.3, pp.201-206, 2001
- [3] Miyamura, T. et al. : Elastic-plastic analysis of nuclear structures with millions of DOFs using the hierarchical domain decomposition method, *Nuclear Engineering and Design*, Vol.212, pp.335-355, 2002
- [4] Ogino, M. et al. : Seismic response analysis of nuclear pressure vessel model with ADVENTURE system on the Earth Simulator, *Journal of The Earth Simulator*, Vol.2, pp.41-54, 2005

- [5] Shioya, R. et al. : An earthquake-proof analysis of a BWR pressure vessel using the next-generation computational solid mechanics simulator, *Annual Report of the Earth Simulator Center April 2006 - March 2007*, pp.163-166, 2007
- [6] Mandel, J. : Balancing domain decomposition, *Communications on Numerical Methods in Engineering.*, Vol.9, pp.233-241, 1993
- [7] 塩谷隆二, 他 : バランシング領域分割法による 3 次元大規模構造解析, 日本計算工学会論文集, Vol.2, pp.139-144, 2000
- [8] 荻野正雄, 他 : バランシング領域分割法による並列弾性有限要素解析, 日本機械学会論文集 A 集, Vol.69, No.685, pp.36-43, 2003
- [9] DeRoeck, Y.H. and LeTallec, P. : Analysis and test of a local domain decomposition preconditioner, *Proceedings of the Fourth International Symposium on Domain Decomposition Methods*, pp.112-128, 1991
- [10] Ogino, M. et al. : An inexact balancing preconditioner for large-scale structural analysis, *Journal of Computational Science and Technology*, 2008 (to be printed)
- [11] Mukaddes, A.M.M. et al. : A scalable balancing domain decomposition based preconditioner for large scale heat transfer problems, *JSME International Journal Series B*, Vol.49, No.2, pp.533-540, 2006