階層化サブストラクチャ法による有限要素法の並列化

5F－7

福盛 秀雄　河野 洋一　西松 研　村岡 洋一
早稲田大学　理工学部

1　はじめに

本発表では、有限要素法求解の一手法であるサブストラクチャ法を階層化することによる有限要素法の並列化について述べる。本方法は基本的な演算効率と並列性の両方の向上を目指すものである。

直接法に基づく有限要素法並列化は従来、バンド行列を対象に並列化を実行しているものが主流であった。この方法はプロセッサ間のデータ分配の均等化が容易な反面、問題が大規模になるに従い、バンド中に存在する零要素による計算量および記憶領域の増大が顕著となってくる。

一方スカイライン法などを適用し、計算量と記憶領域を削減しようとする場合、プロセッサへの均等なデータ分配を実施するのは困難である。またこの場合でも、零要素への演算を完全に減らすことは難しい。

サブストラクチャ法は、有限要素法の解析領域をサブストラクチャと呼ばれる小領域に分割し、各サブストラクチャ間の境界上にある節点についての解を最初に求める。その後にサブストラクチャ内部の節点についての解を求めることにより、解析領域全体の解が求まる。内部の節点についての解を求める作業はデータ依存なしに実行可能なため、この部分では高い並列性が達成できる。さらに、サブストラクチャを複数の階層に分け、段階的に求解処理を実行することで、連立方程式の直接法による求解につきものの零要素に対する演算を減らすことができ、計算効率と並列化効率を同時に向上させることが可能となる。今回は富士通の並列計算機AP1000上に、この階層化されたサブストラクチャ法を実装し、評価を行なった。

2　階層化サブストラクチャ法

階層化サブストラクチャ法は次の手順により実行される。

1. 解析対象となる領域をサブストラクチャと呼ばれる小領域に分解する。第一階層のサブストラクチャは、内部に1節点を持つ、9節点より構成される小領域である。

2. 有限要素法の手順に従い、連立方程式 $A^{(k)}x^{(k)} = b^{(k)}$ を作成する。この方程式はサブストラクチャの内部節点と外部節点との関係を示す、次の式で表現することができる。

$$\begin{bmatrix} A_{ii}^{(1)} & A_{ib}^{(1)} \\ A_{bi}^{(1)} & A_{bb}^{(1)} \end{bmatrix} \begin{bmatrix} x_i^{(1)} \\ x_b^{(1)} \end{bmatrix} = \begin{bmatrix} b_i^{(1)} \\ b_b^{(1)} \end{bmatrix} \tag{1}$$

式(1)の上の行は、サブストラクチャ内部の節点についての式、下の行はサブストラクチャ境界部の節点についての式を示している。

3. 第k階層のサブストラクチャにおいて

(a) 式1で示される連立方程式の上の行 $A_{ii}^{(k)}x_i^{(k)} + A_{ib}^{(k)}x_b^{(k)} = b_i^{(k)}$ を $x_i^{(k)}$ について解き、下の行の式に代入することによって、内部節点についての未知数 $x_i^{(k)}$ に依存しない式

$$A_{bb}^{*(k)} x_b^{(k)} = b_b^{*(k)} \tag{2}$$

を作ることができる。ここで、

$$A_{bb}^{*(k)} = [A_{bb}^{(k)} - A_{bi}^{(k)}(A_{ii}^{(k)})^{-1}A_{ib}^{(k)}], \tag{3}$$
$$b_b^{*(k)} = [b_b^{(k)} - A_{bi}^{(k)}(A_{ii}^{(k)})^{-1}b_i^{(k)}] \tag{4}$$

である。

(b) 各サブストラクチャは、隣接するサブストラクチャとペアを作り、一段階上の階層を構成するサブストラクチャを新たに構成する。$A_{bb}^{*(k)}$ と $b_b^{*(k)}$ は、ペアの相手であるサブストラクチャのそれと足し合わされることで、新たに連立方程式 $A^{(k+1)}x^{(k+1)} = b^{(k+1)}$ を構成する。(図1)

4. サブストラクチャ数が1になるまで、上の手順3.(a)(b)を繰り返す。

5. サブストラクチャ数が1になったら、すなわち最上位サブストラクチャに到達したら、$x_i^{(n)}$ を求める。ここで n は最上位のサブストラクチャの階層値を示す。

6. 第$n-1$階層のサブストラクチャは、$x_i^{(n)}$ を利用して $x_i^{(n-1)}$ を求める。

7. 手順7.を、n=1になるまで、すなわち最下位のサブストラクチャに到達するまで繰り返す。

階層化サブストラクチャ法の特徴は次の通りである。

- 求解過程で扱われる行列はいずれも小さく（n×nの節点構成の解析領域において行列のサイズは9×9～n×n)、かついずれも密行列のため、逐次実行における計算効率に優れている。
- 各階層のサブストラクチャ間において、データの依存関係が存在しないことから、並列化効率の向上が期待できる。

3　並列計算機AP1000への実装

階層化サブストラクチャ法を富士通の並列計算機AP1000上に実装した。サブストラクチャ数がプロセッサと同数となる階層において、1サブストラクチャに対し1プロセッサを割り当て、これより下位に存在するサブストラクチャが同一プロセッサに存在するようにデータ割当を行なった。上位に存在するサブストラクチャに対する演算は、(全体のプロセッサ数)/(サブストラクチャ数)のプロセッサからなるクラスタを構成し、式(3)を並列実行することで並列化効率を向上させようとしている。

Parallelization of FEM with Multi-level Substructure Method
Hideo Fukumori, Yoichi Kono, Ken Nishimatsu and Yoichi Muraoka
Waseda University

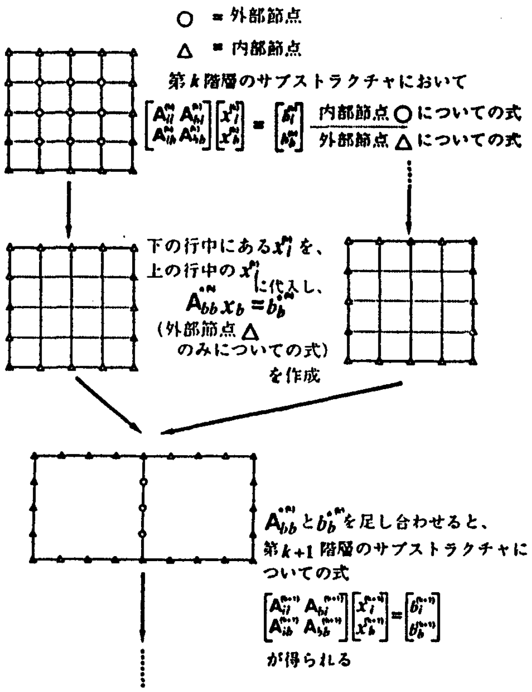

図 1: サブストラクチャの演算および足し合わせ

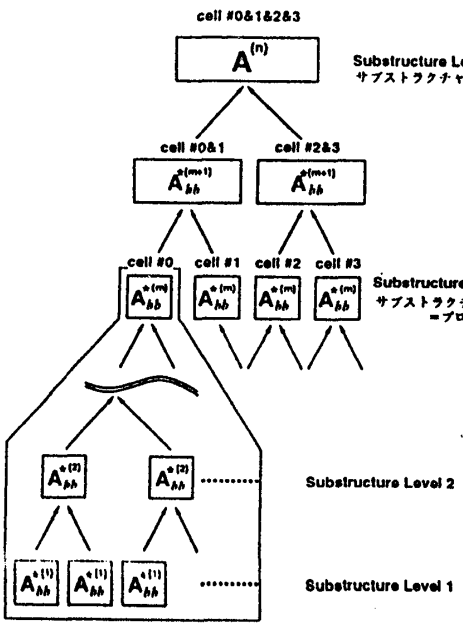

図 2: AP1000 上における並列化

4 性能評価

64×64 (節点数 4225, 三角形要素数 8192)、128×128 (節点数 16641, 三角形要素数 32768) に分割された正方形領域に対するラプラス問題について、性能評価を行なった。

各問題におけるサブストラクチャ階層数は、64×64 メッシュの場合 12、128×128 メッシュの場合 14 階層であった。プロセッサ台数がサブストラクチャ数を上回る階層においては、サブストラクチャ階層を 1 つ上げる際に多対多のプロセッサ間通信が発生する。この部分のオーバヘッドが、大きな台数構成における並列化効率の低下の原因となっているものと考えられる。

表 1: 64×64 メッシュにおける実行結果

台数	時間 (秒)	高速化率	並列化率 (%)
1	5.727	1.00	100.0
2	2.935	1.95	97.5
4	1.600	3.61	90.2
8	1.053	5.43	67.9
16	0.779	7.35	45.9
32	0.643	8.91	27.8
64	0.616	9.30	14.5

表 2: 128×128 メッシュにおける実行結果

台数	時間 (秒)	高速化率	並列化率 (対 4 台)(%)
4	10.273	1.00	100.0
8	6.794	1.51	75.5
16	4.662	2.20	55.0
32	3.536	2.91	36.4
64	3.034	3.39	21.2

5 まとめ

本発表では、階層化されたサブストラクチャ法による有限要素法の並列化を提案し、富士通の並列計算機 AP1000 上への実装結果を示した。以下にこれからの課題を述べる。

- 上位のサブストラクチャ階層におけるプロセッサ間通信の方法の見直し。現在は 1 対 1 通信の繰り返しで実行しているが、これをブロードキャスト中心の方法に改める。
- 本研究室で開発中の並列化要素分割プログラムと組み合わせることにより、並列計算機上に有限要素法システムを構築する。

参考文献

[1] Hideo Fukumori and Yoichi Muraoka: "Parallel FEM Solution Based on Substructure Method", PCW'93 Proceedings of Second Parallel Computing Workshop, P1-K, 1993

[2] I.St.Doltsinis and S.Nolting: "Studies on parallel processing for coupled field problems", Computer Methods in Applied Mechanics and Engineering vol.89, pp.497-521, 1991

[3] F.J.Peters: "Sparse Matrices and Substructures", Mathematisch Centrum, 1980