有限要素解析のための並列領域分割法ライブラリの開発と評価

Developing and Evaluation on A Parallel Domain Decomposition Method Library for Finite Element Analysis

屋 雄介・名古屋大学　　荻野 正雄・名古屋大学　　石井 克哉・名古屋大学
Yusuke Oku, Nagoya University　Masao Ogino, Nagoya University　Katsuya Ishii, Nagoya University

1 緒言

領域分割法(Domain Decomposition Method，DDM)はその高い並列効率などから大規模解析に有効な手法の一つとして知られており，オープンソースの汎用並列有限要素解析システム ADVENTURE[1]で採用されている．本研究で用いる DDM は，解くべき連立一次方程式に対する Schur 補元方程式[2]を構築し，反復法で解く手法である．ADVENTURE の DDM 実装は，高速化のため Schur 補元方程式を陰的に構築する方法が用いられている．しかし，プログラム構成が複雑なことから，反復法や前処理の組合せによる収束性の検討が困難となり，それらが十分に行われていない．以上の背景から，本研究では Schur 補元方程式を陽に構築し，反復法や前処理の組合せによる収束性の検討を容易に可能とするための領域分割法ライブラリに関する研究を行っている．今回は静弾性問題の有限要素解析に対して開発中のライブラリを適用し，反復法や前処理の組合せの検証，解の精度評価や並列実行時間の評価を行った．

2 DDM 概要

静弾性問題の有限要素解析で得られる解くべき方程式を次式で表す．

$$Ax = b \tag{1}$$

このとき，係数行列 A は $n \times n$ の正定値対称行列，x は $n \times 1$ の未知ベクトル，右辺ベクトル b は $n \times 1$ の既知ベクトルである．ここで，有限要素分割された解析対象を N 個の部分領域に分割する．このとき，領域分割によって生じる部分領域間境界上に関する項を下付き添え字 B で，部分領域内部に関する項を下付き添え字 I で表すと，式(1)は次式に並び替えられる．

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix} \begin{pmatrix} x_I \\ x_B \end{pmatrix} = \begin{pmatrix} b_I \\ b_B \end{pmatrix} \tag{2}$$

ただし，行列 A_{II} は $n_I \times n_I$，行列 A_{IB} は $n_I \times n_B$，行列 A_{BI} は $n_B \times n_I$，行列 A_{BB} は $n_B \times n_B$ であり，$n = n_I \times n_B$ である．このとき，Schur 補元行列 S と右辺ベクトル g は次式で表される．

$$S = A_{BB} - A_{BI}(A_{II})^{-1}A_{IB} \tag{3}$$

$$g = b_B - A_{BI}(A_{II})^{-1}A_{IB} \tag{4}$$

これにより，部分領域境界上の未知ベクトル x_B に関する次式が得られる．

$$Sx_B = g \tag{5}$$

式(5)を Schur 補元方程式と呼ぶ．

3. 並列領域分割法ライブラリ

現在開発中の DDM ライブラリのシステム概要を Fig.1 に示す．ライブラリへの入力として，解くべき式の係数行列 A，右辺ベクトル b，Schur 補元方程式構築に必要な領域分割メッシュを与える．次に，DDM ライブラリ内で，領域分割メッシュの情報を基に自由度の並び替え行列 P を作成する．その後，A，b，P を MUMPS[3]が持つ Schur 補元方程式構築関数に入力することで，Schur 補元方程式を陽に構築する．得られた Schur 補元方程式を反復法ライブラリ Lis[4]を用いて解く．最後に，得られた x_B を MUMPS に入力し，式(2)の x_I を求め，全体の解 x が得られる．

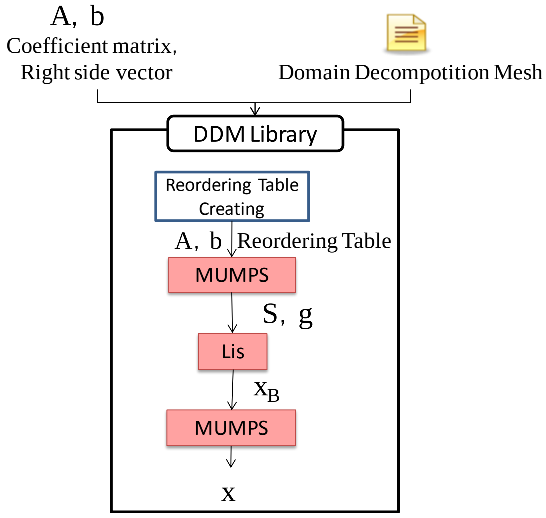

Fig.1 Flow chart of a DDM library

4. 並列領域分割法ライブラリの評価

開発中の並列領域分割ライブラリを用いて，静弾性問題を用いて性能評価を行った．ベンチマーク問題として，要素数が約 2.5 万，未知数が約 13 万，行列 A の非零要素が約 970 万の問題を用いた．また，領域数は 240 とした．

Lis により，反復法を CG 法に固定し，前処理を幾つか変更して $Sx_B = g$ を解いた際の反復回数の結果を Table.1 に示す. 反復法の収束判定は相対残差ノルムが 7 桁小さくなった時点とした．

Table 1 Iteration counts

	CG-None	CG-Jacobi	CG-SSOR[1.2]	CG-簡易Jacobi(ADVENTURE)
$Sx_B=g$	5,691	3,571	1,945	3,604

Schur 補元方程式を陽に構築したことにより，これまで ADVENTURE による CG 法簡易 Jacobi 前処理の検証しかできていなかったが，様々な反復法や前処理の組合せによる収束性の検証が容易に可能能となった．解の精度評価や並列処理時の計算時間などはポスター発表にて議論する．

参 考 文 献

[1] ADVENTURE Project Homepage : http://adventure.sys.t.u-tokyo.ac.jp/jp/，2012
[2] Zhang，The Schur Complement and Its Applications，Springer, 2005
[3] 直接法ライブラリ MUMPS:http://graal.ens-lyon.fr/MUMPS, 2011
[4] 反復法ライブラリ Lis(a Library of Iterative Solvers for liner systems)：http://www.ssisc.org/lis/index.ja.html ，2013