2D-2 領域分割法を用いた並列化境界要素法による内部界面運動の解析

櫛田 雅弘 (阿南工業高等専門学校・一般教科)
杉野 隆三郎 (阿南工業高等専門学校・制御情報工学科)
宮本 陽生 (阿南工業高等専門学校・一般教科)

1. はじめに

数値シミュレーションにおいては, 高い精度が要求され, 多大な計算時間とメモリ空間を要求されることがある. そこで, クラスタコンピューティングの手法を用いた, メモリ分散による並列処理が注目されている.

領域分割法 (DDM) を用いた並列化有限要素法 (FEM) の解析手法は連続的な成功を収めており様々な非定常現象の解析コードが開発されている [1] が, 境界要素法 (BEM) をベースにする並列化 BEM はいまだ基礎的段階にある. そこで, 本研究では,DDM を用いた並列化 BEM による流体内部に存在する界面運動の解析コード [2] を開発し, PVM を用いたクラスターシステムを構築する. そして, 領域分割の方法, 自由度の設定などの違いによる並列化効率や, 各種計算量の数値精度について検討する.

2. 領域分割 – 境界要素法

領域分割法による並列化境界要素法を用いた内部界面運動の数値シミュレーションのアルゴリズムをは次のとおりである.

STEP 1:ラグランジュ乗数に初期値を与える

$$\lambda^0 = \widehat{\lambda}(:\text{constant}) \tag{1}$$

STEP 2:各領域において BEM 計算をし, ノイマンデータの拘束条件を与えて次の汎関数の停留値問題を構成する

$$\sum_{i=1}^{2}\left[\int_{\Gamma_{I+W}^i}\phi_i\frac{\partial\omega_i^*}{\partial \boldsymbol{n}}d\Gamma-\int_{\Gamma_{I+W}^i}\frac{\partial\phi_i}{\partial \boldsymbol{n}}\omega_i^* d\Gamma\right]+\int_{\Gamma_I}\tau^n\lambda^n d\Gamma=0 \tag{2}$$

次に付帯条件としてノイマンデータの連続条件を導入する

$$\tau^n=\frac{\partial\phi_1}{\partial \boldsymbol{n}}+\frac{\partial\phi_2}{\partial \boldsymbol{n}} \qquad on \quad \Gamma_I \tag{3}$$

STEP 3:システムに次の条件を与えて境界値問題を解く

$$\frac{\partial\phi_i}{\partial \boldsymbol{n}}=0 \qquad on \quad \Gamma_W^i \tag{4}$$

$$\phi_1=\lambda^n, \quad \phi_2=\alpha\lambda^n+\beta \qquad on \quad \Gamma_I^i \tag{5}$$

STEP 4:ラグランジュ乗数を UZAWA の方法で収束方向へ更新する

$$\lambda^{n+1}=\lambda^n+\omega\tau^n \tag{6}$$

STEP 5:界面上の付帯条件の相対誤差により収束判定を行う

$$\int_{\Gamma_I}\frac{\tau^n\cdot\tau^n}{\tau^0\cdot\tau^0}d\Gamma\le\varepsilon \tag{7}$$

3. 内部界面計算と並列化

3.1 内部界面運動

内部界面移動については, $\frac{\partial\phi_1}{\partial x}$ と $\frac{\partial\phi_2}{\partial x}$, $\frac{\partial\phi_1}{\partial y}$ と $\frac{\partial\phi_2}{\partial y}$, を加重平均して得られる値を流速成分として, 次の時間ステップにおける移動境界のラグランジュ座標を計算する. この手順を繰り返すことで任意の時刻までの界面の運動をシミュレートすることができる.

3.2 PVM を用いた並列化

本研究では,PVM を用いたクラスタシステムの並列化効率を調べるため, 以下の 3 タイプの並列計算コードを構築した.

Type 1: すべての計算を first-machine で実行する直列型.

Type 2: プロセス間通信をコントロールするマスタープロセスと領域 1 の BEM 計算のスレーブプロセスを first-machine で, 残りの領域 2 の BEM 計算のスレイブプロセスを second-machine で実行する 2 台並列型.

Type 3: first-machine でマスタープロセス, second-machine で領域 1 のスレイブプロセス, third-machine で領域 2 のスレイブプロセスを実行する 3 台並列型.

4. 数値計算例

2 次元矩形領域内に密度差を持つ 2 流体が垂直方向に層を成し, 内部界面を形成しているモデルを取り上げる. 密度の違う 2 つの部分領域 $\Omega_{1,2}$ は界面 (移動境界)$\Gamma_I^{1,2}$ および固定壁 $\Gamma_W^{1,2}$ で囲まれているものとする. 横幅は無次元化長さ $L=0.02$, 高さ $H=0.06$ とし, 下の流体 1 の底面から上の流体 2 までの高さは $h=0.03$ とする. この界面に重力の影響で発生する不安定現象のシミュレーションをテスト問題とする.

'Analysis for Motion of Internal Interface' by Parallel BEM with DDM

Masahiro Kushida (General Education),Ryuzaburo Sugino (System and Control Engineering),Haruo Miyamoto (General Education),Anan College of Technology

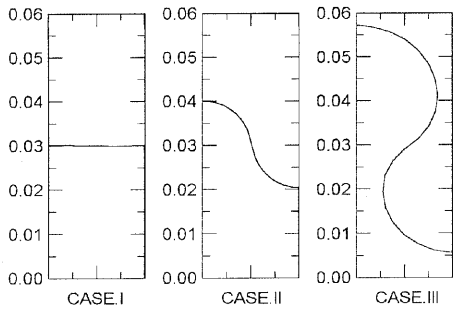

図 1: 内部界面の移動

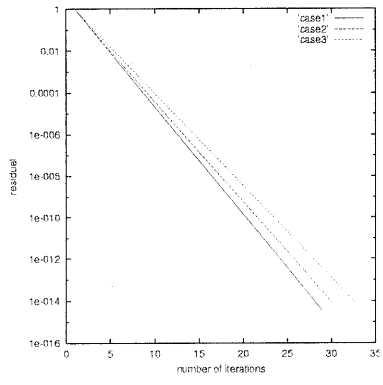

図 2: 界面形状の複雑さよる収束率の変化

5. 計算結果

図.1 は, 直列型での内部界面の移動, 図.2 は各界面形状における UZAWA 法の収束率を示している. これから内部界面の変形が大きくなるほど内部界面の結合条件を満足することが厳しくなることがわかった. 2 台並列型, 3 台並列型においてもこれと同じ結果を得た. これは, 並列化の種類により誤差などは生じず, 各データに影響はないことを示している. 表.1 は, 各計算コードのメモリの大きさの比を表したものである. メモリが分散され 1 台が持つメモリ量が減少している. 表.2 は, 各計算コードによる全体の処理時間, 図.3 は 3 種類の並列計算コードによる時間ステップ毎の実行時間の変化を示している. 各並列計算の全体の実行時間は計算機が増えるほど長くなり, 時間が進むほど各時間ステップでの実行時間が長くなっている. 並列計算のセッションに加わるマシンの台数が増える毎に実行時間が長くなるのはシミュレーション全体の処理時間に対するプロセッサ間の通信に必要な時間と BEM 計算を行う時間のバランスによるものである. すなわち本実験では,BEM 計算の処理時間が短いため, 通信時間の間に発生するプロセスの待機時間が相対的に大きな割合をしめていると考えられる. しかし, これは分割する領域の数や領域における要素数などの自由度が増大するにつれ, 待機時間の問題は解消され, 全体の処理プロセスの実行時間の効率は向上するものと考えられる.

表 1: メモリの分散比

計算コード	MASTER	SLAVE
直列型	1	-
2 台並列型	0.788	0.548
3 台並列型	0.779	0.548

表 2: 各並列計算の全体の処理時間

	直列型	2 台並列型	3 台並列型
時間 (sec)	14.975	22.299	26.991

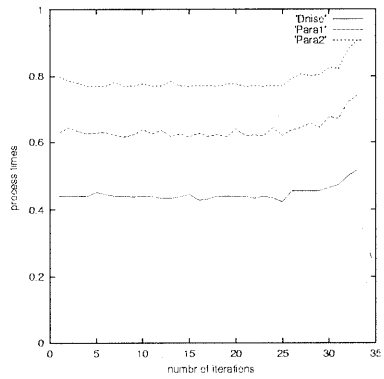

図 3: 時間ステップによる実行時間の変化

6. まとめ

単一プロセッサで実行していた処理を複数のプロセッサにメモリを分散することが可能である並列計算は有効だと考えらる. 分割する領域の数や領域における要素数などの自由度を増加させることで実行時間の効率が向上することを示し, 大規模数値計算を可能にするための並列計算コードを構築することが今後の課題である.

参考文献

[1] 矢川　元基, パラレルコンピューティング, 培風館,1991

[2] Sugino,R. Imai,I. and Tosaka,N., Boundary Element Analysis of Multi-phase Problems with Free-boundary, *Mathematical Science and Applications* Vol.14 *Free Boundary Probrems: Theory and Applications*, Gakuto,pp.431-439,2000