

# 気液二相流れの領域分割型並列有限要素解析

## Domain Decomposition Based Parallel Finite Element Analysis for Gas-Liquid Two-Phase Flow

佐久田朝行<sup>1)</sup> 川上幸亮<sup>2)</sup> 新館京平<sup>3)</sup> 馬込望<sup>4)</sup> 三目直登<sup>5)</sup>

Tomonori Sakuda, Ryosuke Kawakami, Kyohei Shintate, Nozomi Magome and Naoto Mitsume

<sup>1)</sup>筑波大学理工情報生命学術院システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail:s1911110@u.tsukuba.ac.jp)

<sup>2)</sup>宇宙航空研究開発機構 研究開発部門 (〒 305-8505 茨城県つくば市千現 2-1-1)

<sup>3)</sup>筑波大学理工情報生命学術院システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1)

<sup>4)</sup>筑波大学理工情報生命学術院システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1)

<sup>5)</sup>博(工) 筑波大学 システム情報系 助教 (〒 305-8573 茨城県つくば市天王台 1-1-1)

In this study, a domain decomposition-based parallel finite element analysis for gas-liquid two-phase flow was developed, and its computational accuracy and parallel performance were evaluated. First, a stabilized finite element solver based on the Level-set method was developed, and its accuracy was assessed using dam-break problem and rising bubble problem. The result of the dam-break problem under no-slip condition did not show good accuracy while the other experiments showed good agreement. One of the reason for less accuracy on the dam-break problem under slip condition is concluded as its bad measurement. Afterward, a parallel computing algorithm for a distributed memory environment was implemented into the developed solver, and its parallel performance was evaluated on the same benchmark problems. The results confirmed that parallelization reduced computation time while maintaining accuracy, demonstrating that applying parallel computing to interface-capturing finite element analysis for gas-liquid two-phase flow improves both efficiency and precision.

**Key Words** : *finite element method, level-set method, two-phase flow, parallel computation, domain decomposition method*

### 1. 序論

気液二相流れとは、液体と気体の界面が存在する流れのことである。気液二相流れは工学分野において広く見られる。従来、このような流れは実験と観測により分析されることが多かったが、近年では数値計算手法と計算機ハードウェア性能の向上に伴い、複雑な多流体システムの研究において数値実験が一般的になってきている。

気液二相流れの数値解析手法は、メッシュフリー法とメッシュベース法に大別できる。メッシュフリー法は粒子法 [1,2,3] に代表される。粒子法とは流体を多数の計算粒子の集合体として表現するラグランジュ的手法であり、メッシュのコントロールが不要な点と、自由表面を含む複雑な境界移動にも適用できる点が優れている。しかし、解像度を調整することが難しく、とくに境界条件の精度をよくすることは困難である。それに対し、有限体積法や有限要素法に代表される非構造格子系のメッシュベース法は解析空間における空間解像度の調整が可能であり、境界条件を容易かつ高精度に設定できる。

有限体積法は非構造格子を用いたメッシュベースの解析手法であり、分割した各セルについて一般化保存則を立て、差分近似することで定式化される。解析領域の任意の場所における解像度の調整が可能である利

点があり、また、その定式化の過程から、流体の質量保存に優れることが特徴として挙げられる。構造格子系でない場合は高精度化に使用される参照点が多く与えられている保証はなく、一般的に高精度化が難しいといえる。有限要素法は非構造格子を用いたメッシュベースの解析手法であり、各計算点に基底関数を定義し、変分原理に基づいて解を近似する手法である。有限体積法と同様に解析領域の解像度の調整も可能だが、基底関数の工夫次第で高精度化をすることが可能である。しかし、一般的に有限体積法のように保存則は満たさない特徴がある。本研究では、基底関数による高精度化の柔軟性や拡張性の幅広さに着目し、有限要素法を使用する。

非構造格子系の解析手法に基づいた気液二相流れにおいては、界面の移動に合わせて気相・液相の解析領域が変化するので、それをどのように表現するかが課題になる。そこで、ALE(Arbitrary Lagrangian-Eulerian)法 [4] に代表される界面追跡法が提案されている。界面追跡法は、移動する界面に合わせて解析メッシュも移動させる手法である。その結果、界面が複雑な形状になるときには解析メッシュに歪みが生じてしまい、解析メッシュの再生成が必要になる場合があり、界面追跡法は移動境界に対するロバスト性に欠ける点に問題がある。

界面追跡法に見られる問題を克服するため、

VOF (Volume of Fluid) 法 [5] や Level-set 法 [6] に代表される界面捕捉法が提案された．これらの手法では界面を陰的に表現する関数を導入することで，解析メッシュの移動や再生成をすることなく解析を進めることができる．特に Level-set 法は符号付き距離関数 (Signed Distance Function, SDF) が使用され，滑らかで連続的に界面を表現できる．これにより，表面張力の計算や界面の移動を安定的に求めることができる．しかし，界面捕捉法を用いて高精度な結果を得るためには界面追跡法に比べて詳細な解析メッシュが必要となる．

これに伴い，大規模な問題に対してはその計算コストの大きさが課題となる．大きな計算コストに対処するため，領域分割型並列有限要素解析が有効な手法として注目されている．領域分割型並列計算とは分散メモリ型並列環境を対象とした方法であり，解析領域を部分領域に分割しておき，各分割領域に計算ノードを割り当てることで並列に独立した計算処理を行うことで，計算時間を削減する．各計算ノードは割り当てられた領域のデータを保持し，大域的な値や他の計算ノードに割り当てられた値は計算ノード間の MPI (Message Passing Interface) 通信によって共有され，総合しながら解析が進められる．

本研究では，気液二相流れに対して領域分割型並列有限要素解析を適用し，その計算効率の向上を図る．まず，Level-set 法と有限要素法に基づいた気液二相流れ解析ソルバを開発し，精度検証する．この検証では，気液二相流れのベンチマーク問題として一般的なダムブレイク問題と気泡上昇流れ問題の解析し，既存研究のデータと比較する．さらに，分散メモリ型並列環境における並列計算アルゴリズムを開発し，同ソルバに適用，さらに並列性能検証を行う．

## 2. Level-set 法と有限要素法に基づいた気液二相流れ解析

本章では，流体の支配方程式である運動方程式と連続の式に対して有限要素法を用いた解析について述べる．その際，流体に特有の数値不安定性に対処するため，SUPG/PSPG (Streamline Upwind Petrov-Galerkin/Pressure Stabilizing Petrov-Galerkin) 法 [7] に基づく安定化手法を適用する．さらに，Level-set 法によって気相と液相の界面を表現し，二相間で不連続な物理量を平滑化することで，一つの連立一次方程式を導出する．また，Level-set 法と有限要素法に基づいた気液二相流れ解析に特有の数値的な問題について言及し，補正方法について述べる．また，最後に表面張力を導入する．

### (1) 有限要素法に基づいた流体解析

本研究では流れに非圧縮性を仮定する．この流れは境界  $\Gamma$  で囲われた三次元空間  $\Omega$  に広がっており，すべての境界で Dirichlet 境界条件または Neumann 境界が与えられているものとする．このとき，非圧縮性流体の支配方程式は Navier-Stokes 方程式と連続の式によってそれぞれ式 (1), (2) のように記述される．また，Dirichlet 境界条件と Neumann 境界がそれぞれ式 (3), (4) のよう

に成り立っているとする．

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \text{ in } \Omega \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \text{ in } \Omega \quad (2)$$

$$\mathbf{u} = \hat{\mathbf{u}}, \text{ on } \Gamma_g \quad (3)$$

$$(-p\mathbf{I} + \mu \nabla \mathbf{u}) \mathbf{n} = \hat{\mathbf{t}}, \text{ on } \Gamma_h \quad (4)$$

ここで  $\rho$  は流体の密度， $\mathbf{u}$  は流体の速度ベクトル， $p$  は流体の圧力， $\mu$  は流体の粘性係数， $\mathbf{f}$  は外力ベクトルをあらわす．また， $\Gamma_g$  は Dirichlet 境界， $\Gamma_h$  は Neumann 境界を指し， $\hat{\mathbf{u}}$ ， $\hat{\mathbf{t}}$  は定数ベクトルである．さらに， $\mathbf{n}$  は境界に対する外向き単位法線ベクトルである．

支配方程式 (1), (2) について重み付き残差法を用い，各変数と重み関数を形状関数で補間し，適切な式変形をすれば，式 (5) を得る．形状関数によって補間された重み関数の各節点値は任意の値をもつので，式 (5) は  $\mathbf{u}^h$ ， $p^h$  の各節点値についての連立方程式として解くことができる．

$$\begin{aligned} \int_{\Omega} \mathbf{w}^h \cdot \left( \rho \frac{\partial \mathbf{u}^h}{\partial t} + \rho(\mathbf{u}^h \cdot \nabla) \mathbf{u}^h \right) d\Omega + \int_{\Omega} \nabla \mathbf{w}^h : (-p^h \mathbf{I} + \mu \nabla \mathbf{u}^h) d\Omega \\ + \int_{\Omega} q^h (\nabla \cdot \mathbf{u}^h) d\Omega + \sum_{e=1}^M \int_{\Omega_e} \tau_m^e (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \cdot \mathbf{r} d\Omega \\ + \sum_{e=1}^M \int_{\Omega_e} \tau_m^e \nabla q^h \cdot \mathbf{r} d\Omega + \sum_{e=1}^M \int_{\Omega_e} \tau_c^e (\nabla \cdot \mathbf{w}^h) (\nabla \cdot \mathbf{u}^h) d\Omega \\ = \int_{\Gamma_h} \mathbf{w}^h \cdot \hat{\mathbf{t}} d\Gamma_h + \int_{\Omega} \mathbf{w} \cdot \mathbf{f} d\Omega \end{aligned} \quad (5)$$

$$\mathbf{r} = \rho \frac{\partial \mathbf{u}^h}{\partial t} + \rho(\mathbf{u}^h \cdot \nabla) \mathbf{u}^h + \nabla p^h - \mu \nabla^2 \mathbf{u}^h \quad (6)$$

ここで  $\mathbf{w}$ ， $q$  はそれぞれ重み関数であり，右上の添え字  $h$  はその変数が形状関数によって補間されていることを意味する．また，右上の添え字  $e$  は要素インデックスを表し， $M$  は全要素数を指す．式 (5) 左辺において第 4～6 項は流体解析に特有の数値不安定性に対処するために加えられる安定化項であり， $\tau_m^e$ ， $\tau_c^e$  は安定化パラメータである．第 4 項は SUPG 項と呼ばれ，移流の卓越に起因する解の数値不安定性を緩和させる効果がある．第 5 項は PSPG 項と呼ばれ，非圧縮性に起因する解の数値不安定性を緩和させる効果がある．第 6 項は衝撃捕捉項と呼ばれ，界面の不連続性に起因する数値不安定性を緩和させる効果がある．また， $\tau_m^e$ ， $\tau_c^e$  はそれぞれ式 (7)，(8) を用いて求められる [8]．

$$\tau_m^e = \left[ \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{u}^h\|}{h_e} \right)^2 + \left( \frac{4\mu}{\rho h_e^2} \right)^2 \right]^{-1/2} \quad (7)$$

$$\tau_c^e = \frac{h_e}{2} \|\mathbf{u}^h\| z(Re_u) \quad (8)$$

$$z(Re_u) = \begin{cases} Re_u/3 & (0 \leq Re_u \leq 3) \\ 1 & (3 \leq Re_u) \end{cases} \quad (9)$$

$$Re_u = \rho \frac{\|\mathbf{u}^h\| h_e}{2\mu} \quad (10)$$

ここで  $h_e$  は要素サイズ， $Re_u$  は要素 Reynolds 数を表す．

最後に、式 (5) に得られた非線形微分方程式について、Crank-Nicolson 法を用いて時間微分項を、Adams-Bashforth 法によって移流速度を線形近似することで、線形ソルバで解くことのできる連立一次方程式を得ることができる。

## (2) Level-set 法の適用

Level-set 関数  $\phi$  は界面からの符号付き距離を値にもつ関数である。ここでは液体領域で正、気体領域で負の値、界面上で 0 の値を持つものとして定義する。Level-set 関数  $\phi$  は式 (11) に示す Eikonal 方程式に対し、界面で値が 0 となる境界条件を与えた場合の解である。

$$|\nabla\phi| = 1 \quad (11)$$

ここで、式 (12) に示す近似 Heaviside 関数  $H_D(\phi)$  を導入する。近似 Heaviside 関数は  $[-0.5, 0.5]$  の値域をもつ滑らかな関数であり、これを用いて二相間の不連続な物性値の分布を平滑化することができる。具体的には、密度と粘性係数を (13), (14) のように平滑化する。これらを式 (5) ~ (10) に使用し、界面の位置に適応した解析を行うことができる。

$$H_D = 0.5 \cdot \max \left\{ -1.0, \min \left[ 1.0, \frac{\phi}{D} + \frac{1}{\pi} \sin \left( \frac{\pi\phi}{D} \right) \right] \right\} \quad (12)$$

$$\rho = 0.5(\rho_l + \rho_g) + H_D(\rho_l - \rho_g) \quad (13)$$

$$\mu = 0.5(\mu_l + \mu_g) + H_D(\mu_l - \mu_g) \quad (14)$$

ここで  $D$  は界面の平滑化を行う幅であり、格子幅の 1 ~ 5 倍の値をとる。また、添え字  $l, g$  はそれぞれ液体、気体の物性値を表す。

Level-set 関数は、式 (15) の移流方程式に従って時間発展する。

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = 0 \quad \text{in } \Omega \quad (15)$$

ここで  $\mathbf{u}$  は流速ベクトルである。式 (15) は移流方程式であるため数値不安定性があり、式 (16) のように SUPG 法に基づいた安定化有限要素法を適用して解く。

$$\begin{aligned} & \int_{\Omega} \psi^h \left( \frac{\partial\phi}{\partial t} + \mathbf{u}^h \cdot \nabla\phi^h \right) d\Omega \\ & + \sum_e \int_{\Omega_e} \tau_{\phi} \mathbf{u}^h \cdot \nabla\psi^h \left( \frac{\partial\phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla\phi^h \right) d\Omega \\ & + \sum_e \int_{\Omega_e} \tau_{LSIC} \nabla\psi^h \cdot \nabla\phi^h d\Omega = 0 \end{aligned} \quad (16)$$

ここで  $\tau_{\phi}$ ,  $\tau_{LSIC}$  はそれぞれ式 (17), (18) に示す SUPG 項と衝撃捕捉項の安定化パラメータである。

$$\tau_{\phi} = \left[ \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{u}^h\|}{h_e} \right)^2 \right]^{-1/2} \quad (17)$$

$$\tau_{LSIC} = \frac{h_e}{2} \|\mathbf{u}^h\| \quad (18)$$

さて、Level-set 法と有限要素法に基づいた気液二相流れ解析には特有の二つの数値的な問題がある。一つは式 (15) の移流項に対する安定化の影響で界面幅が広がり、Level-set 関数の傾きが変化する問題であり、もう一つは各相の体積に誤差が生じ、体積が保存されない問題である。

Level-set 関数の傾きが変化する問題に対処するため、本研究では、Sussman ら [6] が提案した再初期化方程式に安定化項  $\alpha \nabla^2 \phi$  を加えた式 (19)[9] を用いる。式 (19) は式 (11) の誤差を修正する効果をもつ。本研究では、式 (5) の時間ステップごとに式 (19) の計算を上限回数に達するまで繰り返す。

$$\frac{\partial\phi(x)}{\partial \tilde{t}} = S(\phi_0)(1 - |\nabla\phi(x)|) + \alpha \nabla^2 \phi \quad (19)$$

ここで  $\tilde{t}$  は、式 (15) の  $t$  とは異なる仮想的な時間であり、 $\alpha$  は安定化の程度を調整する係数である。 $S(\phi_0)$  は符号関数の近似値であり、式 (20) のように計算することで得られる。

$$S(\phi) = \frac{\phi(x)}{\sqrt{\phi(x)^2 + \delta^2}} \simeq \text{sgn}(\phi) \quad (20)$$

ここで  $\delta$  は  $\phi$  を正規化する際に分母が非零となることを保証するために加える微小値である。

続いて、気液相の体積に誤差が生じ、体積が保存されない問題に対処するため、次の方法で体積を補正する。まず、時刻  $t$  における気体の体積  $V_{gas}(t)$  を式 (21) のように計算する。

$$V_{gas}(t) = \int_{\Omega} (0.5 - H_D) d\Omega \quad (21)$$

初期の体積を  $V_{gas}(0)$  とすると、時刻  $t$  における体積の誤差  $V_{error}$  は式 (22) のように計算される。

$$V_{error}(t) = V_{gas}(t) - V_{gas}(0) \quad (22)$$

そこで、界面の総面積  $A(t)$  を用いて補正量  $L_{error}$  を式 (23) のように計算する。本来界面は面積分を用いて表されるが、式 (24) のように Dirac の Delta 関数を用いて体積積分で表現できる [10]。

$$L_{error}(t) = \frac{V_{error}(t)}{A(t)} \quad (23)$$

$$A(t) = \int_{\Omega} \delta_D d\Omega, \quad (24)$$

$\delta_D$  には式 (25) に定義される近似 Delta 関数を使用し、平滑化して計算する [11]。

$$\delta_D(\phi) = \begin{cases} \frac{1}{2D} \left[ 1 + \cos \frac{\pi\phi}{D} \right], & (|\phi| \leq D) \\ 0, & (D \leq |\phi|) \end{cases} \quad (25)$$

ここで  $D$  は界面厚さである。

## (3) 表面張力の導入

表面張力は、Level-set 関数を用いて、CSF (Continuum Surface Force) モデル [12] として導入することができ

る．CSF モデルでは，表面張力による体積力を以下のように表す．

$$f_s = \sigma \kappa \nabla \phi \quad (26)$$

ここで  $\sigma$  は表面張力係数， $\kappa$  は界面の曲率， $\phi$  は Level-set 関数である．曲率は式 (27) のように計算される．

$$\kappa = -\nabla \cdot \mathbf{n}_s \quad (27)$$

ここで  $\mathbf{n}_s$  は外向き単位法線ベクトルであり，式 (28) のように計算される．

$$\mathbf{n}_s = \frac{\nabla \phi}{|\nabla \phi|} \quad (28)$$

表面張力は界面に働くため，式 (25) の近似デルタ関数を用いて判定を行う．

### 3. 分散メモリ型並列計算環境における領域分割型並列計算

本研究では線形ソルバに安定化双共役勾配法 (Bi-CGStab 法) [13] を採用する．Bi-CGStab 法はベクトル和，ベクトル内積，行列ベクトル積の線形代数演算から構成されており，これらの線形代数演算を並列計算できれば，反復法全体の並列計算が実現可能になる．そこで，本章では，領域分割型の並列計算を導入し，分散メモリ型並列計算環境におけるこれらの線形代数演算の並列計算アルゴリズムを示す．領域分割型並列計算では，解析領域の節点と節点同士の隣接関係をグラフ構造と見なし，さらにそのグラフを部分グラフに分割し，それぞれの領域ごとに演算が行われる．図 1 にその概略図を示す．

解析領域に属するすべての節点の集合  $X = \{x_1, x_2, \dots, x_M\}$  をグラフ構造におけるノードとみなす．各節点上に定義される基底関数同士の積が非零となる節点は隣接しているとみなし，この隣接関係をエッジ集合  $E = \{e_{i,j} | x_i, x_j \in X\}$  で表現する．このとき，解析領域をグラフ構造  $G = (X, E)$  と見なすことができる．ここではグラフ  $G$  はセルフエッジ  $e_{i,i}$  を含むものとし，無向グラフとして扱う．

続いてグラフ  $G$  を  $N$  個に分割することを考える．並列計算を前提としている領域分割では，計算量を均等にするだけでなく，隣接領域間の通信量を小さく抑えることも重要な指標となる．そのような領域分割は，制約付き最小カット問題として定式化されている．最小カット問題では，ノード集合  $X$  は  $N$  個の部分集合  $X^{(i)} (i \in \{1, \dots, N\})$  に重複なく分割され，式 (29)，(30) が成り立つ．

$$X^{(k)} \cap X^{(l)} = \emptyset, \quad \text{for } k \neq l \text{ and } k, l \in \{1, \dots, N\}, \quad (29)$$

$$\bigcup_{k=1}^N X^{(k)} = X. \quad (30)$$

ここで部分ノード集合  $X^{(k)}$  におけるすべてのノードに隣接するノードのうち， $X^{(k)}$  に含まれないものの集合を  $\text{Adj}(X^{(k)}, G)$  と定義する． $\text{Adj}(X^{(k)}, G)$  は overlap 領域と呼ばれ，部分ノード集合間の接続情報を補完する．最

小カット分割によって得られた部分ノード集合に，それぞれの overlap 領域を追加した集合を  $W^{(k)}$  とする．

$$W^{(k)} = \text{Adj}(X^{(k)}, G) \cup X^{(k)} \quad (31)$$

ノード集合  $W^{(k)}$  に属するノード同士で構成されるすべてのエッジで構成されるエッジ集合をノード集合  $F^{(k)}$  とすると，分割グラフは  $G^{(k)} = (W^{(k)}, F^{(k)})$  で与えられる．

さて，本研究で解く方程式はすべて，最終的に式 (32) のような疎な係数行列を持つ連立一次方程式になる．

$$A\mathbf{x} = \mathbf{b} \quad (32)$$

ここで  $A$  は係数行列， $\mathbf{x}$  は解ベクトル， $\mathbf{b}$  は右辺ベクトルという．式 (32) の係数行列  $A$  と解ベクトル  $\mathbf{x}$  を，分割グラフ  $G^{(k)}$  の内部領域  $X^{(k)}$  に合わせて式 (33)，(34) のように分割する．

$$A = \begin{bmatrix} A^{(1,1)} & \dots & A^{(1,N)} \\ \vdots & \ddots & \vdots \\ A^{(N,1)} & \dots & A^{(N,N)} \end{bmatrix} \quad (33)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{x}^{(N)} \end{bmatrix} \quad (34)$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(N)} \end{bmatrix} \quad (35)$$

ここで  $A^{(i,j)}$  は内部領域  $X^{(i)}$  が内部領域  $X^{(j)}$  から受ける作用を表す行列であり， $X^{(i)}$  と  $X^{(j)}$  が隣接するときに非零の値を含む．また， $\mathbf{x}^{(i)}$ ， $\mathbf{b}^{(i)}$  はそれぞれ内部領域  $X^{(i)}$  に対応する解ベクトルと右辺ベクトルである．さらに，ノードの全体グラフと分割グラフにおける関係を紐づけることを考える．まず，各ノードに解析領域全体に定義されるグローバル番号  $i_G$  に対して，各ノードが内部領域に属する分割グラフ  $G^{(k)}$  上に定義されるローカル番号  $i_L$  を定義する．さらに，グラフ  $G$  と分割グラフ  $G^{(k)}$  間において，ローカル番号  $i_L$  をグローバル番号  $i_G$  にうつす写像  $\text{idx}$  を式 (36) のように定義する．

$$i_G = \text{idx}(k, i_L, G) \quad (36)$$

この写像を用いて，ローカル番号  $i_L$  をグローバル番号  $i_G$  に対応させる行列  $\mathbf{P}^{(k)}$  が式 (37) のように定義する．

$$\mathbf{P}_{i_G, i_L}^{(k)} = \begin{cases} 1, & i_G = \text{idx}(k, i_L, G) \\ 0, & \text{otherwise} \end{cases} \quad (37)$$

式 (37) を用いると，式 (38)，(39)，(40) が成り立つ．

$$A = \sum_{k=1}^N \sum_{l=1}^N \mathbf{P}^{(k)} A^{(k,l)} \mathbf{P}^{(l)T} \quad (38)$$

$$\mathbf{x} = \sum_{k=1}^N \mathbf{P}^{(k)} \mathbf{x}^{(k)} \quad (39)$$

$$\mathbf{b} = \sum_{k=1}^N \mathbf{P}^{(k)} \mathbf{b}^{(k)} \quad (40)$$

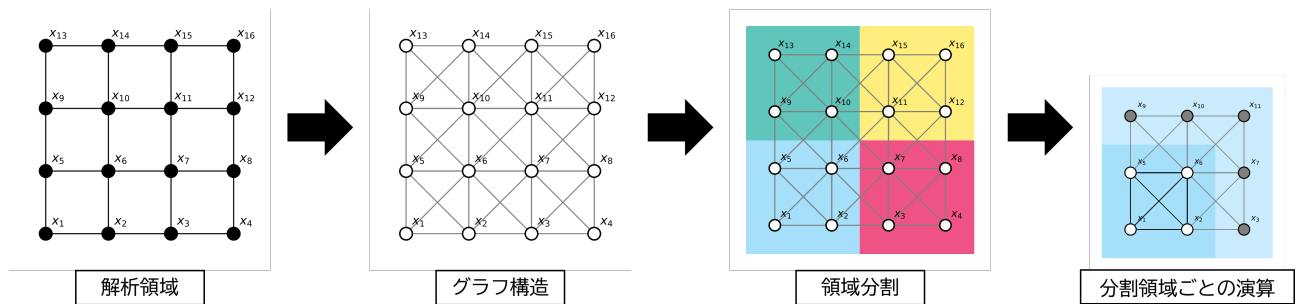


図-1 領域分割型並列計算

式 (38), (39), (40) の関係から, 全体領域の係数行列, 解ベクトル, 右辺ベクトルが分割した領域に対応する係数行列, 解ベクトル, 右辺ベクトルを用いて分解できることが示された. 続いて, この関係を用いれば, 全体領域における線形代数演算が分割グラフ  $G^{(k)}$  の内部領域における線形代数演算を用いて分解できることを示す. この計算において, 各分割グラフの内部領域に対応する行列やベクトルの値は, 並列メモリ型計算環境で用いる各計算ノードに保持され, 各領域における演算は各計算ノードごとに独立に行うことができる. ただし, overlap 領域  $\text{Adj}(X^{(k)}, G)$  については他の分割領域の値を参照する必要があり, 各計算ノード間で MPI (Message Passing Interface) 通信を行うことによって共有される.

ベクトル和  $z = x + y$  は式 (41) のように定義される. ベクトル和は overlap 領域の値を参照する必要がないため, 計算ノード間の通信は必要ない.

$$z = \sum_{k=1}^N P^{(k)} \{x^{(k)} + y^{(k)}\} \quad (41)$$

ベクトル内積  $\alpha = x^T y$  は, 式 (42) のように定義される.

$$\alpha = \sum_{k=1}^N \alpha^{(k)} = \sum_{k=1}^N x^{(k)T} y^{(k)} \quad (42)$$

ここで  $\alpha^{(k)}$  は分割領域  $X^{(k)}$  が担当するベクトル内積の値である. ベクトル内積は各計算ノードの担当する内積の結果  $\alpha^{(k)}$  を足し合わせることで求められ, 各計算ノードはそれを参照するが, その参照には計算ノード間の通信が必要である.

行列ベクトル積  $y = Ax$  は, 式 (44) のように定義される.

$$y = \sum_{k=1}^N P^{(k)} y^{(k)} \quad (43)$$

$$= \sum_{k=1}^N P^{(k)} A^{(k,k)} x^{(k)} + \sum_{k=1}^N \sum_{x_l \in \text{Adj}(X^{(k)}, G)} P^{(k)} A^{(k,l)} x^{(l)} \quad (44)$$

式 (44) の右辺第 2 項は overlap 領域の寄与を表す. 行列ベクトル積は, 図 2 のように overlap 領域の値を他の計算ノードから参照しなければならない, 計算ノード間の通信が必要である. 以上の手順に従い, 線形ソルバの分散メモリ型並列計算環境における領域分割型並列計算プロセスが確立された.

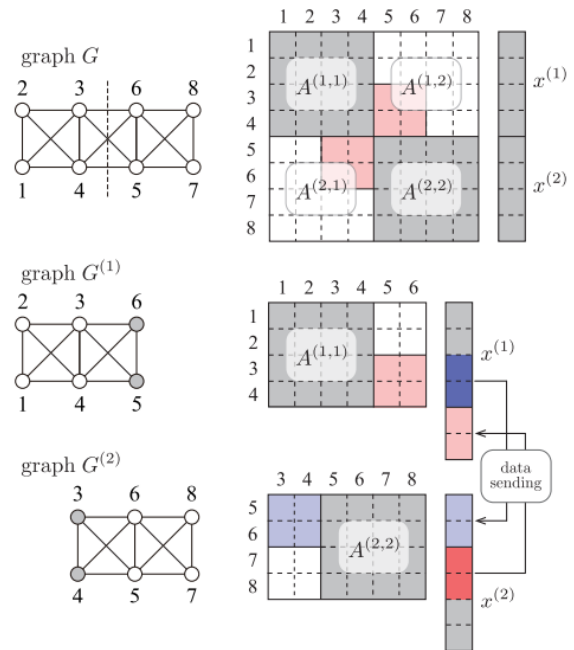


図-2 overlapping 分割で得られた分割グラフと行列ベクトル積の概略図 [14]

#### 4. 検証

提案手法の検証および計算例は, 口頭発表にて紹介する.

#### 5. 結論

本研究では, Level-set 法を用いた界面捕捉型の有限要素法に基づき, 気液二相流れの解析ソルバの開発および検証を行った. さらに, 同ソルバに応用する分散メモリ型並列計算環境における並列計算アルゴリズムの開発および検証を行った. この検証を通して, 提案手法による気液二相流れの有限要素解析が分割数に応じて計算速度が向上することが確認された.

謝辞: 本研究は, JST 創発的研究支援事業 JPMJFR215S および JSPS 科研費 23K26356 の支援を受けたものである. ここに記して謝意を表する. なお, 本研究は宇宙航空研究開発機構との共同研究ではない.

## 参考文献

- [1] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 12 1977.
- [2] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 12 1977.
- [3] S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123(3):421–434, 1996.
- [4] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary lagrangian–eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.
- [5] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.
- [6] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions for incompressible two-phase flow. *J. Comput.*, 144:146–159, 1994.
- [7] T.E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1991.
- [8] L.P. Franca and S.L. Frey. Stabilized finite element methods: II. the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Eng.*, 99:209–233, 1992.
- [9] P. Pimenta. Level set implementation using fenics, 2018. Accessed: 2025-01-21.
- [10] B. Engquist, A.K. Tornberg, and R. Tsai. Discretization of dirac delta functions in level set methods. *Journal of Computational Physics*, 207(1):28–51, 2005.
- [11] S. Nagrath, K.E. Jansen, and R.T. Lahey. Three dimensional simulation of incompressible two-phase flows using a stabilized finite element method and a level set approach. *Preprint submitted to Elsevier Science*, 2003.
- [12] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.
- [13] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.
- [14] 森田直樹, 集路幸正, 田中克治, 馬込望, 新館京平, 柴沼一樹, and 三目直登. 領域分割型並列シミュレーションのためのグラフ構造に基づく統一ライブラリと多手法への展開. 日本計算工学会論文集, 2024:20241008, 2024.