計算工学講演会論文集 Vol. 30 (2025 年 6 月)
日本計算工学会

階層型 local POD に対する 適応的基底選択法とその負荷分散

Hierarchical Local POD with Adaptive Basis Selection and Load Balancing

新舘京平 [1] 森田直樹 [2] 金子栄樹 [3] 三目直登 [4]
Kyohei Shintate, Naoki Morita, Shigeki Kaneko and Naoto Mitsume

[1]筑波大学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: shintate.kyohei.qm@alumni.tsukuba.ac.jp)
[2]博 (環境) 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, E-mail: nmorita@kz.tsukuba.ac.jp)
[3]博 (工) 名古屋工業大学大学院工学研究科 助教
（〒 466-0061 愛知県名古屋市昭和区御器所町, E-mail: kaneko.shigeki@nitech.ac.jp)
[4]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, E-mail: mitsume@kz.tsukuba.ac.jp)

POD (proper orthogonal decomposition) is a method for obtaining a basis for representing physical phenomena and has been used in reduced order models for fast analysis. To apply POD to large-scale problems, distributed memory parallel computing, in which parallel processes are allocated to each partitioned subdomain, is effective. On the other hand, Local POD, which obtains a basis for each decomposed subdomain, has been proposed to improve the computational efficiency of POD. Against this background, we propose a method to independently set subdomains where the basis is acquired and subdomains where parallel processes are allocated. In this study, the selection of the appropriate number of bases for each subdomain and the extension to load balancing will be implemented.

Key Words : *Parallel Computing, Domain Decomposition, Graph Structure, Reduced Order Modeling*

1. 序論

工学分野において，偏微分方程式に含まれるパラメータを様々に変化させ，繰り返し解析を行うパラメトリックスタディが重要な役割を担っている．しかし，全ての解析を有限要素法をはじめとする高精度な解析システムで行うことは容易ではない．そこで近年では，高精度な数値解析に対する高速な代替モデルとして，ROM (reduced order model) [1] の研究が盛んに行われている．ROM は数値計算における高次元の未知数を，基底と呼ばれる特徴的な構造で捉えて低次元化する解析モデルであり，高速な解析を可能にする．この基底を用いた ROM 解析は，オフラインフェーズとオンラインフェーズから構成される．オフラインフェーズでは，まず高精度な解析システムを用いて数値解を計算し，snapshot と呼ばれるトレーニングデータを収集する．次に，収集したデータから，snapshot POD (snapshot proper orthogonal decomposition) [2] に代表される手法を用いて，基底を抽出する．snapshot POD とは，収集したデータに特異値分解を施し，少数の基底を選定する手法である．オンラインフェーズでは，これらの基底を数値解析に導入することで未知数を大幅に削減し，高速に解析を行う．なかでも，snapshot POD にもとづく未知数の低次元表現と Galerkin projection を組み合わせた POD-Galerkin 法は，多彩な工学問題 [1] に応用されている．

精緻なメッシュ構造を必要とする複雑現象の数値解析においては，高精度な解析システムの自由度数が増大することに伴い，メモリ容量の制約が課題となる．そのような複雑現象に ROM 解析を適用する場合，大自由度解析の数値解をトレーニングデータとして使用する必要がある．このとき，トレーニングデータおよびトレーニングデータから取得される POD 基底は，低次元化前の解析の自由度数に対応する大規模なベクトルから構成されるため，メモリ容量の制約が課題となる．このような課題に対する有効な解決手段として，分散メモリ型並列計算機を利用した並列計算が挙げられる．その代表的な方法として，対象のセルや要素を幾何学的に分割し，解析に要するデータの分散と計算負荷分散を実現する領域分割法があり，有限要素法をはじめとする数値解析分野で広く用いられている [3,4]．しかしながら，POD を基にした ROM 解析においては，POD の特異値分解部分 (オフラインフェーズ) のみを対象に領域分割法による分散メモリ型並列計算を実施した前例がある [5,6] ものの，POD-Galerkin 法を含む ROM 解析のオンラインフェーズ全てを並列化した例はなく，低次元化後も依然として自由度数が大きい問題 [7,8] に対しては，決定的な方策が提唱されていない．

他方，前述の領域分割の考え方を，分散メモリ型並列計算ではなく，基底を分割領域毎に取得するという観点で利用した既存手法として，L-POD (local POD) [9] が提案されている．L-POD とは，はじめに分割領域毎に独立して特異値分解を行い，各分割領域の基底を連結した後に，Galerkin 法と組み合わせた近似式により解析システムを低次元化する手法である．分割領域毎に基底が取得されることで，基底取得にかかる計算時間が短縮され，基底の表現能力の向上によって計算精度が向上する利点を持つ．加えて，各分割領域で基底を取得するため物理場の複雑性を部分的に回避するこ

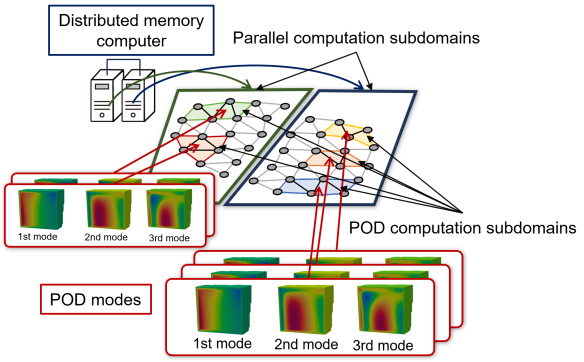

Fig.1: Schematic overview of the proposed method.

とができ [10]，マルチフィジックス問題 [11] など局所的に複雑な現象に対して活用されている．L-POD においても，次元削減後の問題が大規模になり得るが，領域分割による分散メモリ型並列化の前例がない．

このような背景から本研究では，L-POD に対し，オンラインフェーズも含めた包括的な領域分割型並列計算の実現を目的とする．同時に，領域分割型並列化を適用しつつも，POD 計算領域数を任意に調整可能とする方法論を提示する．このために，本研究ではまず，有限要素節点群・POD 計算領域群・並列計算領域群の全ての空間離散化構造に対し，それぞれのグラフ構造を定義する．その上で，2 階層の領域分割法を提案することで，「分散メモリ型並列化のための領域」とは独立な「POD 基底を取得する領域」の設定を可能とする．具体的には，Fig.1に示すように，まず有限要素のグラフを分割し POD 計算領域を定義した後に，その「POD 基底を取得する領域」の連結性を表すグラフ (メタグラフ) に対し再度領域分割を行うことで，「分散メモリ型並列化のための領域」を定義する．加えて，有限要素法における overlapping 型領域分割並列アルゴリズム [3] を拡張する形で，L-POD の領域間の相互作用を表す密なブロック行列に対する並列アルゴリズムを提案する．本論文では，POD 計算領域毎に基底数を可変にすることで生じる並列計算領域間の計算量のばらつきを解消し，計算負荷を均一化するための負荷分散手法について述べる．

2. Local Proper Orthogonal Decomposition (L-POD)

本章では，はじめに snapshot POD に基づいた ROM 解析手法について述べ，領域分割手法を導入した後に，分割領域毎に特異値分解を実行する解析手法である L-POD を概説する．

(1) Snapshot POD

ROM の構築に用いる基底選択の手法として snapshot POD [2] を用いる．snapshot POD は，高精度な解析システムで数値計算を行うことで，snapshot と呼ばれるトレーニングデータを収集し，形成される snapshot 行列に対して固有直交分解 (POD) を行う手法である．

n_{snap} 本の snapshot データ $\boldsymbol{d}_{(i)} \in \mathbb{R}^n$ $(i = 1, 2, \ldots, n_{\mathrm{snap}})$ からなる snapshot 行列 $\boldsymbol{S} \in \mathbb{R}^{n \times n_{\mathrm{snap}}}$ を次式に示す．

$$\boldsymbol{S} = \left[\boldsymbol{d}_{(1)}, \boldsymbol{d}_{(2)}, \ldots, \boldsymbol{d}_{(n_{\mathrm{snap}})} \right] \tag{1}$$

ここで，n は低次元化前の解析モデルの総自由度数を表す．

続いて，snapshot 行列 $\boldsymbol{S}$ の特徴を表現する k_{POD} 本の基底 $\boldsymbol{v}_i \in \mathbb{R}^n$ $(i = 1, 2, \ldots, k_{\mathrm{POD}})$ を取得することを考える．POD ではこの問題を，各 snapshot データと，基底によって張られる空間への各 snapshot データの正射影との誤差が最小になるよう，次の式に定式化する．

$$\boldsymbol{v}_i = \arg\min_{\boldsymbol{g}_i} \sum_{j=1}^{n_{\mathrm{snap}}} \left\| \boldsymbol{d}_{(j)} - \sum_{l=1}^{k_{\mathrm{POD}}} \boldsymbol{g}_l \boldsymbol{g}_l^{\mathrm{T}} \boldsymbol{d}_{(j)} \right\|^2$$
$$\text{with } \boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} = \boldsymbol{I},\ \boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_{k_{\mathrm{POD}}}] \in \mathbb{R}^{n \times k_{\mathrm{POD}}} \tag{2}$$

式 (2) は，snapshot 行列 $\boldsymbol{S}$ に対して特異値分解を行うことで求められる．

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^{\mathrm{T}} \tag{3}$$

ここで，$\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r] \in \mathbb{R}^{n \times r}$ $(r = \mathrm{rank}\, \boldsymbol{S})$ の各列は左特異ベクトル，$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{r \times r}$ $(i = 1, 2, \ldots, r)$ は特異値 σ_i を対角要素とする行列である．このとき，σ_i は i 番目の特異値を表し，添え字が小さいほど大きく，非負の値を持つ．$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_r] \in \mathbb{R}^{n_{\mathrm{snap}} \times r}$ の各列は右特異ベクトルである．

左特異ベクトルが並ぶ行列 $\boldsymbol{U}$ のうち，大きな特異値に対応する始めの k_{POD} 本を選定し，POD 基底 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{k_{\mathrm{POD}}}] \in \mathbb{R}^{n \times k_{\mathrm{POD}}}$ を得る．特異値 σ_i は i 番目の基底の寄与度を示すため，適切な閾値 $\varepsilon_{\mathrm{POD}}$ を用いて次式を満たすように基底の数 k_{POD} が決定される．

$$\frac{\sum_{i=1}^{k_{\mathrm{POD}}} \sigma_i}{\sum_{i=1}^{r} \sigma_i} > 1 - \varepsilon_{\mathrm{POD}} \tag{4}$$

(2) POD-Galerkin 法

本節では，解ベクトルを POD 基底で近似し，Galerkin 法によって高速に解を求める手法である POD-Galerkin 法について述べる．なお，本研究では節点における自由度数を 1 として定式化するものの，本研究にて提案する手法は，任意の自由度数に拡張可能である．

支配方程式として次式で表される非定常拡散方程式を用いる．

$$\frac{\partial u}{\partial t} - k\nabla^2 u = f \tag{5}$$

ここで，k は拡散係数，u は拡散する物理量を表す未知数であり，解析領域を 3 次元の有界領域 Ω とすると，f は解析領域 Ω で発生する物理量を表すソース項である．境界条件は次式で表される．

$$u = u_D \quad \text{on} \quad \Gamma_1 \tag{6}$$
$$\nabla u \cdot \boldsymbol{n} = q_N \quad \text{on} \quad \Gamma_2 \tag{7}$$

ここで，u_D は Dirichlet 境界 Γ_1 上での関数解 u の値であり，$\boldsymbol{n}$ は Neumann 境界 Γ_2 上の外向き法線ベクトル，q_N は境界 Γ_2 で与える物理量を表す関数である．式 (5), (6), (7) を有限要素法により離散化することで，離散化された式 (8) が得られる．

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{u} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{f} \tag{8}$$

ここで，$\boldsymbol{w} \in \mathbb{R}^n$ は重み関数に由来する任意のベクトル，n は有限要素法で離散化したときの自由度数を表す．ただし，重みベクトル $\boldsymbol{w}$ のベクトル要素 w_i は $w_i = 0\ (i \in \eta_g)$ を満たす．ここで，η_g は Dirichlet 境界条件が課された有限要素節点のインデックスの集合を表す．$\boldsymbol{K} \in \mathbb{R}^{n\times n}$ は有限要素法における係数行列，$\boldsymbol{u} \in \mathbb{R}^n$ は解ベクトル，$\boldsymbol{f} \in \mathbb{R}^n$ は右辺ベクトルである．

POD-Galerkin 法では，まず解ベクトル $\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{g}$ から Dirichlet 境界条件が課されたベクトル $\boldsymbol{g}$ を除いた $\bar{\boldsymbol{u}}$ の低次元化を行う．そのために，有限要素法の解ベクトル $\boldsymbol{u}$ から，Dirichlet 境界条件が課されたベクトル $\boldsymbol{g}$ の値を除いたベクトル $\bar{\boldsymbol{u}}$ を snapshot データとして収集する．このようにして収集された i 番目の snapshot データを $\boldsymbol{d}_{(i)} \in \mathbb{R}^n$ として，snapshot 行列 $\boldsymbol{S}$ を構築する．

続いて，snapshot 行列に特異値分解を施し，得られた POD 基底を用いて ROM を構築する．解ベクトル $\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{g}$ のうち，$\bar{\boldsymbol{u}}$ は snapshot 行列 $\boldsymbol{S}$ から得られた POD 基底 $\boldsymbol{V}$ によって近似されるとする．

$$\bar{\boldsymbol{u}} \approx \boldsymbol{V}\tilde{\boldsymbol{u}} \tag{9}$$

ここで，$\tilde{\boldsymbol{u}} \in \mathbb{R}^{k_{\mathrm{POD}}}$ は基底ベクトルの結合係数を表す未知ベクトルである．結果として，解ベクトル $\boldsymbol{u}$ に対する近似式 $\boldsymbol{u} \approx \boldsymbol{V}\tilde{\boldsymbol{u}} + \boldsymbol{g}$ を得る．式 (9) で表される近似式は Galerkin 法と組み合わせて用いられ，重み関数は次式に低次元化される．

$$\boldsymbol{w} \approx \boldsymbol{V}\tilde{\boldsymbol{w}} \tag{10}$$

ここで，$\tilde{\boldsymbol{w}} \in \mathbb{R}^{k_{\mathrm{POD}}}$ は任意のベクトルである．これら近似式 (9), (10) を用いて，式 (8) は次式に低次元化される．

$$\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{V}\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{g}) \tag{11}$$

式 (11) は任意のベクトル $\tilde{\boldsymbol{w}}$ に対して成り立つことから，次式を得る．

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{V}\tilde{\boldsymbol{u}} = \boldsymbol{V}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{g}) \tag{12}$$

式 (12) は，低次元化された係数行列 $\tilde{\boldsymbol{K}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{V} \in \mathbb{R}^{k_{\mathrm{POD}}\times k_{\mathrm{POD}}}$，低次元化された右辺ベクトル $\tilde{\boldsymbol{f}} = \boldsymbol{V}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{g}) \in \mathbb{R}^{k_{\mathrm{POD}}}$ を用いて，$\tilde{\boldsymbol{K}}\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}}$ と表される．

(3) Overlapping 型領域分割法

overlapping 型領域分割法では，まず以下の定義のもと，節点集合 X を N 個の節点集合 $X^{(i)}\ (i = 1, 2, \ldots, N)$ に重複なく分割する．

$$X^{(i)} \cap X^{(j)} = \emptyset \quad (i \neq j) \tag{13}$$

$$\bigcup_{i=1}^{N} X^{(i)} = X \tag{14}$$

式 (13), (14) を満たすように X を分割したとき，各領域に属する節点集合を内部節点集合とし，領域 i に属する内部節点数を $n^{(i)}$ で表す．オーバーラップ領域を加味した節点集合 $\bar{X}^{(i)}$ は次式で与えられる．

$$\bar{X}^{(i)} = X^{(i)} \cup X^{(i)}_{\mathrm{ovl}} \tag{15}$$

ここで，$X^{(i)}_{\mathrm{ovl}}$ は領域 i が隣接する節点からなる集合のうち，隣接する領域に属する節点に限定した集合である．なお，節点 a に定義される有限要素法の基底関数 $N_a(\hat{\boldsymbol{x}})$ に対して，節点 b の基底関数との積 $N_a(\hat{\boldsymbol{x}})N_b(\hat{\boldsymbol{x}})$ が非零となる座標 $\hat{\boldsymbol{x}} \in \mathbb{R}^3$ が存在するとき，すなわち，

$$\exists\, \hat{\boldsymbol{x}} \in \mathbb{R}^3,\ N_a(\hat{\boldsymbol{x}})N_b(\hat{\boldsymbol{x}}) \neq 0 \tag{16}$$

が成り立つ場合，節点 a は節点 b と隣接するとみなす．

次に，領域分割前の節点集合 X に対応する係数行列 $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ を，領域 i, j の内部節点集合 $X^{(i)}, X^{(j)}$ に対応する行列 $\boldsymbol{K}^{(i,j)} \in \mathbb{R}^{n^{(i)}\times n^{(j)}}$ を用いて次式で表す．

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^{(1,1)} & \boldsymbol{K}^{(1,2)} & \cdots & \boldsymbol{K}^{(1,N)} \\ \boldsymbol{K}^{(2,1)} & \boldsymbol{K}^{(2,2)} & \cdots & \boldsymbol{K}^{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}^{(N,1)} & \boldsymbol{K}^{(N,2)} & \cdots & \boldsymbol{K}^{(N,N)} \end{bmatrix} \tag{17}$$

ここで，$\boldsymbol{K}^{(i,j)}$ は領域 i が領域 j から受ける作用を表す行列であり，領域 i が領域 j と隣接するとき非零なブロック行列成分を持つ．

また，領域分割前の節点集合 X に対応し，各節点上にスカラ量が定義されたベクトル $\boldsymbol{x} \in \mathbb{R}^n$ は，領域 i の内部節点集合 $X^{(i)}$ に対応するベクトル $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n^{(i)}}$ を用いて次式で表される．

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(N)} \end{bmatrix} \tag{18}$$

(4) L-POD の定式化

L-POD [9] では，まず分割領域毎に独立して特異値分解を行い，各領域で基底を取り出し，各分割領域の基底を連結する．次に，Galerkin 法と組み合わせた近似式により高精度な解析システムを低次元化する手法である．

はじめに，分割領域 i の内部節点集合に対応する snapshot 行列 $\boldsymbol{S}^{(i)} \in \mathbb{R}^{n^{(i)}\times n_{\mathrm{snap}}}\ (i = 1, 2, \ldots, N_{\mathrm{POD}})$ に対して，特異値分解を行う．

$$\boldsymbol{S}^{(i)} = \boldsymbol{U}^{(i)}\boldsymbol{\Sigma}^{(i)}\boldsymbol{W}^{(i)\mathrm{T}} \tag{19}$$

ここで，N_{POD} は POD 基底を取得する分割領域の数である．式 (4) より，左特異ベクトルが並ぶ行列 $\boldsymbol{U}^{(i)} \in \mathbb{R}^{n^{(i)}\times r^{(i)}}$ ($r^{(i)} = \mathrm{rank}\, \boldsymbol{S}^{(i)}$) のうちはじめの $k^{(i)}_{\mathrm{POD}}$ 本を選定し，POD 基底 $\boldsymbol{V}^{(i)} \in \mathbb{R}^{n^{(i)}\times k^{(i)}_{\mathrm{POD}}}$ とする．各領域で独立に基底が取得されるため，全体領域での POD 基底 $\boldsymbol{V}$ は次式で与えられる．

$$\boldsymbol{V} = \mathrm{diag}\left(\boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)}, \ldots, \boldsymbol{V}^{(N_{\mathrm{POD}})}\right) \tag{20}$$

全体領域の係数ベクトル $\tilde{\boldsymbol{u}}$ は，分割された領域に対応する係数ベクトル $\tilde{\boldsymbol{u}}^{(i)} \in \mathbb{R}^{k^{(i)}_{\mathrm{POD}}}$ を用いて以下のように表される．

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \tilde{\boldsymbol{u}}^{(1)} \\ \tilde{\boldsymbol{u}}^{(2)} \\ \vdots \\ \tilde{\boldsymbol{u}}^{(N_{\mathrm{POD}})} \end{bmatrix} \tag{21}$$

L-POD においても，この $\tilde{\boldsymbol{u}}$ と $\boldsymbol{V}$ を用いて，従来の POD と同様に式 (9) で解ベクトルを近似する．重み関数に対しても，Galerkin 法を適用し，$\boldsymbol{V}$ を用いて式 (10) で近似することで，低次元化された連立一次方程式 (12) を得る．

このとき，低次元化された係数行列 $\tilde{\boldsymbol{K}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{V}$ は，次式で表される．

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \tilde{\boldsymbol{K}}^{(1,1)} & \tilde{\boldsymbol{K}}^{(1,2)} & \dots & \tilde{\boldsymbol{K}}^{(1,N_{\mathrm{POD}})} \\ \tilde{\boldsymbol{K}}^{(2,1)} & \tilde{\boldsymbol{K}}^{(2,2)} & \dots & \tilde{\boldsymbol{K}}^{(2,N_{\mathrm{POD}})} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{K}}^{(N_{\mathrm{POD}},1)} & \tilde{\boldsymbol{K}}^{(N_{\mathrm{POD}},2)} & \dots & \tilde{\boldsymbol{K}}^{(N_{\mathrm{POD}},N_{\mathrm{POD}})} \end{bmatrix} \tag{22}$$

ここで，低次元化された係数行列の各ブロック行列 $\tilde{\boldsymbol{K}}^{(i,j)} \in \mathbb{R}^{k_{\mathrm{POD}}{}^{(i)} \times k_{\mathrm{POD}}{}^{(j)}}$ は，対応する節点の POD 基底との積で表され，以下のように計算される．

$$\tilde{\boldsymbol{K}}^{(i,j)} = \boldsymbol{V}^{(i)\mathrm{T}}\boldsymbol{K}^{(i,j)}\boldsymbol{V}^{(j)} \tag{23}$$

領域 i と領域 j が隣接するとき，行列ベクトル積 $\boldsymbol{V}^{(i)\mathrm{T}}\boldsymbol{K}^{(i,j)}\boldsymbol{V}^{(j)}$ が非零になるため，式 (22) は式 (17) と同様なブロック行列の非零構造を持つ．

低次元化された右辺ベクトル $\tilde{\boldsymbol{f}} = \boldsymbol{V}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{g})$ は，対応する領域に属する POD 基底との積で計算される．

$$\tilde{\boldsymbol{f}} = \begin{bmatrix} \tilde{\boldsymbol{f}}^{(1)} \\ \tilde{\boldsymbol{f}}^{(2)} \\ \vdots \\ \tilde{\boldsymbol{f}}^{(N_{\mathrm{POD}})} \end{bmatrix} \tag{24}$$

ここで，$\tilde{\boldsymbol{f}}^{(i)} \in \mathbb{R}^{k_{\mathrm{POD}}^{(i)}}$ は領域 i での低次元化された右辺ベクトル，$\boldsymbol{f}^{(i)} \in \mathbb{R}^{n^{(i)}}$ は領域 i での右辺ベクトル，$\boldsymbol{g}^{(i)} \in \mathbb{R}^{n^{(i)}}$ は領域 i での Dirichlet 境界条件を満たし，Dirichlet 境界条件が指定されている $\boldsymbol{g}^{(i)}$ のベクトル要素以外は 0 のベクトルである．

(5) Empirical Cubature Method (ECM)

さらなる計算効率の向上を図るため，ECM (empirical cubature method) [12] を導入する．詳細は，紙面の都合により省略する．

3. L-POD に対する階層型並列化手法

本研究では，階層型グラフに基づいた L-POD の並列計算アルゴリズムを提案する．

はじめに，本研究で扱うグラフについて述べる．グラフ G は，ノード集合 $X = \{x_1, x_2, \dots, x_n\}$ とノードを結ぶエッジ集合 $E = \{e_{a,b} \mid x_a, x_b \in X\}$ を用いて，$G = (X, E)$ と定義されるデータ構造である．ここで，$e_{a,b}$ は 2 つのノード $x_a, x_b \in X$ が隣接していることを表す．また，n はノード数 $|X|$ である．本論文で定義されるグラフはセルフエッジ $e_{a,a}$ を含むものとし，向きのない無向グラフとして扱う．

(1) 計算点グラフとメタグラフ

計算点グラフとは，計算点間の相互作用を表現するグラフ構造であり，対象としている連立一次方程式の非零構造に対応する．例えば，有限要素メッシュの節点を計算点とする計算点グラフは，有限要素法により離散化して得られる連立一次方程式の非零構造に対応する．

グラフ構造を用いることで，ある隣接関係を持つ任意のノード集合に対して領域分割が定義できる．有限要素メッシュの節点を計算点とする計算点グラフを考えたとき，「有限要素メッシュに対する領域分割」が，「計算点グラフに対するグラフ分割」として一般化される．すわなち，ある隣接関係を持つ任意のグラフのノード集合 X に対して，式 (13), (14) を満たすように領域分割し，内部節点集合に対応するノード集合 $X^{(i)}$ $(i = 1, 2, \dots, N)$ を得た後，式 (15) によってオーバーラップ領域 $X_{\mathrm{ovl}}^{(i)}$ を含むノード集合 $\bar{X}^{(i)}$ が定義される．また，ノード集合 $\bar{X}^{(i)}$ から構成されるエッジ集合を $\bar{E}^{(i)}$ としたとき，部分グラフ $\bar{G}^{(i)} = (\bar{X}^{(i)}, \bar{E}^{(i)})$ が得られる．以降，計算点グラフに対するグラフ分割は，上述の操作に沿って部分グラフ $\bar{G}^{(i)}$ が定義されるグラフ分割とする．

次に，計算点グラフに対してグラフ分割を行ったとき，定義される領域の隣接関係について述べる．Fig.2の左側に示すように，計算点グラフを分割し得られる内部領域 $X^{(i)}$ に対し，それぞれ新たにメタノード $\tilde{x}_i$ を定義する．これらメタノードに対して，各領域の隣接関係を表すメタグラフ $\tilde{G} = (\tilde{X}, \tilde{E})$ が定義できる．ここで，ノード集合 $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$ は分割された領域を表す集合であり，エッジ集合 $\tilde{E} = \{\tilde{e}_{i,j} \mid \tilde{x}_i, \tilde{x}_j \in \tilde{X}\}$ は領域間の隣接関係を表す．このようなグラフは，領域間を跨ぐ計算点グラフのエッジを判別することで構築され，次式が成り立つとき領域 i は領域 j と隣接するとみなし，メタグラフのエッジ $\tilde{e}_{i,j}$ で表現する．

$$X^{(i)} \cap X_{\mathrm{ovl}}^{(j)} \neq \emptyset \tag{25}$$

続いて，計算点グラフおよびメタグラフに基づいて，グラフ分割前後の行列 $\boldsymbol{A}$，ベクトル $\boldsymbol{x}$ の対応関係を明示する．これらの対応関係は，3.2 節にて述べるグラフ構造に基づいた並列計算の定式化に利用する．具体例として，有限要素の節点を計算点とする計算点グラフを考えると，これに対応して係数行列 $\boldsymbol{K}$，解ベクトル $\boldsymbol{u}$，および右辺ベクトル $\boldsymbol{f}$ が定義される．これらは本章で扱う行列 $\boldsymbol{A}$ およびベクトル $\boldsymbol{x}$ の枠組みに含まれ，同様に扱うことが可能である．領域分割前のノード番号をグローバル番号 a_{G} とし，領域分割後の各領域におけるノード番号をローカル番号 a_{L} と定義する．分割前後の行列 $\boldsymbol{A}$ と $\boldsymbol{A}^{(i,j)}$ およびベクトル $\boldsymbol{x}$ と $\boldsymbol{x}^{(i)}$ の関係は，グローバル番号 a_{G} 上に定義されたベクトル要素を領域 i のローカル番号 a_{L} へ対応させる行列 $\boldsymbol{P}^{(i)} \in \mathbb{R}^{|X| \times |X^{(i)}|}$ を用いて，次式で表される．

$$\boldsymbol{A} = \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{P}^{(i)}\boldsymbol{A}^{(i,j)}\boldsymbol{P}^{(j)\mathrm{T}} \tag{26}$$

$$\boldsymbol{x} = \sum_{i=1}^{N} \boldsymbol{P}^{(i)}\boldsymbol{x}^{(i)} \tag{27}$$

ここで，領域 i と j が隣接しているとき，領域 i と j の相互作用を表すブロック行列 $\boldsymbol{A}^{(i,j)}$ は非零となる．メタグラフは領域間の隣接関係を表すため，Fig.2に示すよ

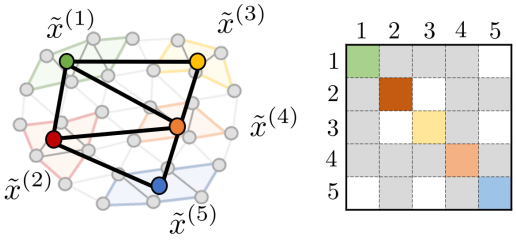

Fig.2: Relationships between metagraph and block matrix (non-zero block matrices are shown in color or gray).

うに行列 $\boldsymbol{A}$ のブロック行列 $\boldsymbol{A}^{(i,j)}$ に関する非零構造は，メタグラフを参照することで一意に定まる．領域 i のローカル番号 a_{L} を対応するグローバル番号に変換する関数 $a_{\mathrm{G}} = \mathrm{idx}(i, a_{\mathrm{L}})$ を考えると，$\boldsymbol{P}^{(i)}$ の行列成分は次式となる．

$$P^{(i)}_{a_{\mathrm{G}} a_{\mathrm{L}}} = \begin{cases} 1, & a_{\mathrm{G}} = \mathrm{idx}(i, a_{\mathrm{L}}) \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

(2) 分散メモリ型並列計算機に対する領域分割型並列化

本研究では，並列計算機の利用を前提として，反復法を採用する．反復法の並列計算は，そのアルゴリズムから，ベクトル和，ベクトル内積，行列ベクトル積の並列計算を実現すればよい．

グラフのノード集合 X を N_{par} 個のノード集合 $X^{(i)}$ に分割したとき，式 (26), (27) について，i 番目の分割領域を表すインデックス i に並列プロセスを割り当てる．インデックス i に関する並列プロセスは，部分グラフ $G^{(i)}$ が割り当てられ，ブロック行列 $\boldsymbol{A}^{(i,j)}$ $(i = 1, 2, \ldots, N_{\mathrm{par}})$ とベクトル $\boldsymbol{x}^{(i)}$ を保持する．ここで，N_{par} は並列プロセスを割り当てる領域数である．

行列ベクトル積 $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ の並列計算について述べる．行列ベクトル積の並列計算は，領域 i に関する計算 $\boldsymbol{A}^{(i,i)}\boldsymbol{x}^{(i)}$ と，領域 i とその隣接領域 j に関する計算 $\boldsymbol{A}^{(i,j)}\boldsymbol{x}^{(j)}$ に分けられ，式 (30) で表される．ここで，$\mathrm{nbhd}(\tilde{x}_i, \tilde{G})$ は，グラフ $\tilde{G} = (\tilde{X}, \tilde{E})$ に関してノード $\tilde{x}_i \in \tilde{X}$ と隣接するノードのインデックスをすべて返す関数である．

$$\begin{aligned} \boldsymbol{z} &= \sum_{i=1}^{N_{\mathrm{par}}} \boldsymbol{P}^{(i)} \boldsymbol{z}^{(i)} \\ &= \sum_{i=1}^{N_{\mathrm{par}}} \boldsymbol{P}^{(i)} \left\{ \boldsymbol{A}^{(i,i)}\boldsymbol{x}^{(i)} + \sum_{j \in \mathrm{nbhd}(\tilde{x}_i, \tilde{G})} \boldsymbol{A}^{(i,j)}\boldsymbol{x}^{(j)} \right\} \end{aligned} \tag{29}$$

ここで領域 i とその隣接領域 j に関する計算 $\boldsymbol{A}^{(i,j)}\boldsymbol{x}^{(j)}$ については，隣接領域 j に値が保持されたベクトル $\boldsymbol{x}^{(j)}$ $(j \in \mathrm{nbhd}(\tilde{x}_i, \tilde{G}))$ を MPI 通信によって取得し計算する．ここで行列 $\boldsymbol{A}^{(i,i)}$ は分割領域 i の内部領域 $X^{(i)}$ に対応する行列であり，分割領域ごとに独立に生成できる．また行列 $\boldsymbol{A}^{(i,j)}$ についても，分割領域 i のオーバーラップ領域 $X^{(i)}_{\mathrm{ovl}}$ に，行列 $\boldsymbol{A}^{(i,j)}$ の計算に必要なグラフを保持しているため，MPI 通信なく分割領域ごとに独立に生成できる．

(3) グラフ構造に基づく L-POD の階層型並列化

本研究では，2.4 節で紹介した L-POD における「POD 基底を取得する領域」と，3.2 節で紹介した「領域分割型並列化のための領域」を独立に定義し，L-POD の分散メモリ型並列計算を実現する方法として，メタグラフ構造を利用した階層型領域分割法を提唱する．なお，記述の簡潔性のために，これ以降は,「POD 基底を取得する領域」を POD 計算領域,「領域分割型並列化のための領域」を並列計算領域と呼称する．また，本研究では POD 計算領域数が並列計算領域数よりも大きい問題を対象とする．この階層型領域分割法の全体像を Fig.3 に示す．

まず，2.3 節で紹介したグラフと領域分割の定義を基に，有限要素の節点に対応するグラフに対して領域分割 (Fig.3 の Domain decomposition 1) を行い， POD 計算領域を生成する．このようにして生成された POD 計算領域は，L-POD によって低次元の基底関数が貼られた粗い有限要素のようなものと解釈できる．

次に，3.1 節にて紹介したメタグラフの定義に従い，POD 計算領域群に対応するメタグラフを生成する．なお，このメタグラフに対しても，有限要素の節点に対応するグラフと同様に，2.3 節で紹介した基底関数の積によるエッジの定義 (16) が適用できる．ここでの基底関数は，L-POD が生成する低次元化された基底関数である．通常，POD によって生成される基底関数は領域全体にわたって定義されるモードのようなものであり，そのグラフ構造は全対全連結 (密行列に対応) であるため領域分割型並列化は困難である．しかし，L-POD は領域が分割されており，上記のメタグラフに対応する疎なブロック行列構造を有するため，低次元化と領域分割型並列解析の両立が可能である．

上記のメタグラフに対し，再度領域分割 (Fig.3 の Domain decomposition 2) を行うことで，並列計算領域が定義できる．Fig.3 下部に図示されているように，各並列計算領域は特定の POD 計算領域群を内包しており，3.2 節の領域分割型並列計算を適用することで，分散メモリ型並列計算を実現する．また，この並列計算領域のメタグラフ (POD 計算領域メタグラフのメタグラフ) によって，並列計算時の通信テーブルが構築される．

(4) 計算点グラフに基づいた負荷分散

式 (4) に基づき POD 計算領域毎に基底数を可変にした場合，それらを内包する並列計算領域の計算量にばらつきが生じ，結果として全体の計算効率が低下する．そこで，本研究ではこの問題を解決するために負荷分散を導入する．

まず,「POD 計算領域を計算点としたグラフ」の i 番目のノードにノード重み $n^{(i)}_{\mathrm{weight}}$ を付与することで，重み付きグラフを得る．次に，この重み付きグラフに対してノード重みの総和を均一かつ，分割グラフを跨ぐエッジ数を最小にするように分割し部分グラフを得る．最後に，得られた部分グラフに基づき，3.3 節で述べた並列計算を行うことで負荷分散を実現する．

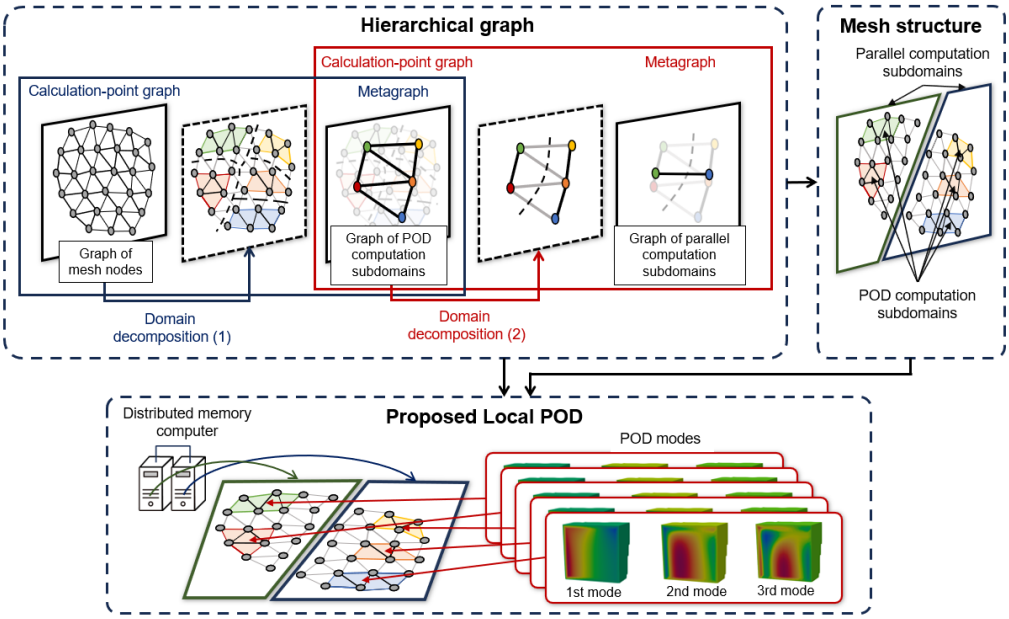

Fig.3: Schematic overview of the proposed method.

4. 数値実験

提案手法を用いた計算例は口頭発表にて紹介する．

5. 結論

本研究では，L-POD の領域分割型並列計算を目的として，階層型グラフ分割に基づいた並列計算アルゴリズムを提案した．加えて，グラフ構造を活用した負荷分散手法を新たに開発し，数値実験によって計算効率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，JSPS 科研費 23K24857，24K22288，24H00695 および学際大規模情報基盤共同利用・共同研究拠点 (JHPCN) jh240017 の支援を受けたものである．ここに記して謝意を表する．

参考文献

[1] Benner, P., Schilders, W., Grivet-Talocia, S., Quarteroni, A., Rozza, G. and Miguel Silveira, L., Volume 2: snapshot-based methods and algorithms. De Gruyter, Berlin, 2021.

[2] Sirovich, L., Turbulence and the dynamics of coherent structures. i - coherent structures. *Quarterly of Applied Mathematics*, **45**-3, 1987, pp. 561–571.

[3] Yagawa, G. and Shioya, R., Parallel finite elements on a massively parallel computer with domain decomposition. *Computing Systems in Engineering*, **4**-4, 1993, pp. 495–503.

[4] 森田直樹, 集路幸正, 田中克治, 馬込望, 新舘京平, 柴沼一樹, 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的ライブラリと多手法への展開. 日本計算工学会論文集, **2024**-1, 2024, p. 20241008.

[5] Cao, C., Nie, C., Pan, S., Cai, J. and Qu, K., A constrained reduced-order method for fast prediction of steady hypersonic flows. *Aerospace Science and Technology*, **91**, 2019, pp. 679–690.

[6] Wang, Z., McBee, B. and Iliescu, T., Approximate partitioned method of snapshots for POD. *Journal of Computational and Applied Mathematics*, **307**, 2016, pp. 374–384.

[7] Östh, J., Noack, B.R., Krajnović, S., Barros, D. and Borée, J., On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *Journal of Fluid Mechanics*, **747**, 2014, pp. 518–544.

[8] Peherstorfer, B., Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *SIAM Journal on Scientific Computing*, **42**-5, 2020, pp. A2803–A2836.

[9] Baiges, J., Codina, R. and Idelsohn, S., A domain decomposition strategy for reduced order models. application to the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **267**, 2013, pp. 23–42.

[10] Lucia, D.J., King, P.I. and Beran, P.S., Reduced order modeling of a two-dimensional flow with moving shocks. *Computers & Fluids*, **32**-7, 2003, pp. 917–938.

[11] Corigliano, A., Dossi, M. and Mariani, S., Domain decomposition and model order reduction methods applied to the simulation of multi-physics problems in MEMS. *Computers & Structures*, **122**, 2013, pp. 113–127.

[12] Hernandez, J.A., Caicedo, M.A. and Ferrer, A., Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, **313**, 2017, pp. 687–722.