

有限要素解析の次元削減モデル構築における Empirical Cubature Method の並列化

Parallelization of Empirical Cubature Method
for Model Order Reduction of Finite Element Method

平野皓大¹⁾ 金子栄樹²⁾ 三目直登³⁾ 新館京平⁴⁾ 森田直樹⁵⁾

Akihiro Hirano, Shigeki Kaneko, Naoto Mitsume, Kyohei Shintate, and Naoki Morita

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: s2420891@u.tsukuba.ac.jp）

²⁾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）

³⁾博(工)筑波大学 システム情報系 助教（〒 305-0003 茨城県つくば市天王台 1-1-1, E-mail: mitsume@kz.tsukuba.ac.jp）

⁴⁾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）

⁵⁾博(環境) 筑波大学 システム情報系 助教（〒 305-0003 茨城県つくば市天王台 1-1-1, E-mail: nmorita@kz.tsukuba.ac.jp）

Finite Element Method (FEM) is widely used in manufacturing for design optimization, but it is impractical to cover every possible parameter combination. This limitation has led to growing interest in Reduced Order Model (ROM), which based on projection method. However, the projection process often becomes a major bottleneck in ROM analysis. To address this issue, Hernandez et al. proposed the Empirical Cubature Method (ECM), a type of hyper-reduction method. Furthermore, Farhat et al. developed an ECM approach for domain decomposition. However, the computational cost can increase because the number of selected elements grows with the number of domains. In this study, we propose an algorithm that efficiently reduces the number of selected elements by applying the ECM in a hierarchical scheme. We also evaluate its impact on computational time, accuracy, and the number of selected elements.

Key Words : Finite Element Method, Reduced Order Model, Hyper-reduction, Domain Decomposition Method, Empirical Cubature Method

1. 序論

製品設計の現場では、有限要素法に基づくシミュレーションが広く利用されている。そこでは各設計変数が製品のプロトタイプに与える影響を評価するために、繰り返しシミュレーションを行うことで設計を最適化するパラメトリックスタディが行われる。しかし、その設計変数の組み合わせを網羅的に計算することは計算時間の面で非現実的であるため、次元削減モデル (Reduced Order Models, ROM) を用いた効率的な計算が注目されている。今日、様々な次元削減手法が提案されているが、本研究では、固有直交分解 (Proper Orthogonal Decomposition, POD) を用いた射影ベースの ROM[1] に注目し、本手法を POD-Galerkin 法と呼称する。POD-Galerkin 法は非線形問題への適用において、次元削減モデルを構築するための射影演算を複数回行う必要があることから計算コストに課題があったが、Hernandez らの ECM (Empirical Cubature Method) [2] はその射影演算において効率的な積分法を導入することで計算コストの削減を実現した。

ECM とは、あらかじめデータ駆動的にサンプリングされた少数の重み付き要素の情報を用いて Gauss 積分を近似計算し、POD-Galerkin 法の射影演算を行うことで、次元削減モデル構築部分の計算時間を削減する手法である。ここでの要素とは、有限要素法における要素を指す。しかし、本手法の重み付きサンプリ

ングのアルゴリズムは逐次的にメッシュ全体へのアクセスが必要であるため、大規模問題のような分散メモリ型の並列計算が必要とされる環境では分散メモリ間での通信量が大きくなり、効率的な計算が行えない。そこで Farhat らは、分散メモリ環境で領域分割法に基づいた分散メモリ間での通信を必要としない近似的な重み付きサンプリングのアルゴリズム [3] を提案した。

しかし、本手法は各局所領域ごとに独立してサンプリングを行い、各局所領域ごとにサンプリングされた要素の総和を大域的なサンプリング要素とするため、大域的な要素数は領域分割数に応じて大きくなってしまう。重み付きサンプリングによって選定される要素数の増加は計算量の増加に直結するので、領域分割数を必要とする大規模並列問題に本手法を適用した場合、大域的なサンプリング要素数の増加による射影演算の総計算量増加が無視できない。

そこで本研究では、少量の通信を追加し、階層的なサンプリングを行う並列化 ECM アルゴリズムを提案する。これにより分散メモリ型並列計算環境でも領域分割を実行しない場合と同程度の大域的な重み付きサンプリング要素の取得を図る。本提案手法は片持ち梁モデルを用いた簡易的な構造解析の数値例を用いてその計算性能を評価する。

2. ECM を用いた ROM による次元削減モデル

(1) 非線形構造解析の ROM

本研究では陰解法の一種である Newmark- β 法を用いた非線形構造解析を考える。詳細モデルにおける支配方程式は式(1)で与えられる。

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_{t+\Delta t} + \boldsymbol{C}\dot{\boldsymbol{u}}_{t+\Delta t} + \boldsymbol{f}^{\text{int}}(\boldsymbol{u}_{t+\Delta t}) - \boldsymbol{f}^{\text{ext}}(t) = \boldsymbol{0} \quad (1)$$

$\boldsymbol{M} \in \mathbb{R}^{n \times n}$ は質量行列、 $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ は減衰行列、 $\boldsymbol{f}^{\text{int}}, \boldsymbol{f}^{\text{ext}} \in \mathbb{R}^n$ はそれぞれ内力ベクトル、外力ベクトルである。また、 $\boldsymbol{u}_{t+\Delta t}, \dot{\boldsymbol{u}}_{t+\Delta t}, \ddot{\boldsymbol{u}}_{t+\Delta t} \in \mathbb{R}^n$ はそれぞれ時刻 $t + \Delta t$ で得られる変位ベクトル、速度ベクトル、加速度ベクトルである。ここで、 n は詳細モデルの自由度、 t は時間を表す。

POD-Galerkin 法による次元削減モデルを用いた解析を本研究のような陰解法の時間発展問題に適用する場合、まず最初にいくつかの適当な時間ステップの解を snapshot として取得し、それらの固有直交分解によって snapshot を十分に表現可能な基底 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_k] \in \mathbb{R}^{n \times k}$ を取得する。次に詳細モデルでの解 $\boldsymbol{u}_{t+\Delta t}$ を得られた基底 \boldsymbol{V} を用いて式(2)のように近似する。

$$\boldsymbol{u}_{t+\Delta t} \approx \boldsymbol{V}\boldsymbol{a}_{t+\Delta t} \quad (2)$$

ここで、 $\boldsymbol{a}_{t+\Delta t} \in \mathbb{R}^k$ は次元削減モデルにおける解ベクトルである。 k は基底本数であり、次元削減後の低次元空間の自由度である。これより、次元削減モデルにおける非線形解析の支配方程式は式(3)で与えられる。

$$\boldsymbol{V}^T \boldsymbol{M} \boldsymbol{V} \ddot{\boldsymbol{a}}_{t+\Delta t} + \boldsymbol{V}^T \boldsymbol{C} \boldsymbol{V} \dot{\boldsymbol{a}}_{t+\Delta t} + \boldsymbol{V}^T \boldsymbol{f}^{\text{int}}(\boldsymbol{V}\boldsymbol{a}_{t+\Delta t}) - \boldsymbol{V}^T \boldsymbol{f}^{\text{ext}}(t) = \boldsymbol{0} \quad (3)$$

なお、非線形構造解析では Newton-Raphson 法によって非線形項の評価を行うが、次元削減モデルにおける Newton-Raphson 法では式(4)の線形方程式を反復的に解くことで式(3)の解を得る。

$$\boldsymbol{V}^T \boldsymbol{K}(\boldsymbol{V}\boldsymbol{a}_{t+\Delta t}^i) \boldsymbol{V} \Delta \boldsymbol{a}^{i+1} = \boldsymbol{V}^T \boldsymbol{\psi}(\boldsymbol{V}\boldsymbol{a}_{t+\Delta t}^i) \quad (4)$$

ここで、上付き添え字 $i, i+1$ は Newton-Raphson 法における反復回数を表し、 $\boldsymbol{K}(\cdot) \in \mathbb{R}^{n \times n}$ は接線剛性行列である。また、 $\Delta \boldsymbol{a}^{i+1} \in \mathbb{R}^k$ は次元削減モデルにおける Newton-Raphson 法の $i+1$ ステップ目の解の修正量である。加えて $\boldsymbol{\psi}(\cdot) \in \mathbb{R}^n$ は残差ベクトルである。

式(3)、式(4)における $\boldsymbol{V}^T \boldsymbol{f}^{\text{int}}(\boldsymbol{V}\boldsymbol{a})$ 、 $\boldsymbol{V}^T \boldsymbol{K}(\boldsymbol{V}\boldsymbol{a}) \boldsymbol{V}$ は、詳細モデルでの未知数に依存するため、Newton-Raphson 法の反復ごとに射影演算が必要となり、非線形解析では次元削減モデル適用における計算コスト削減の効果が小さくなってしまう。

(2) ECM (Empirical Cubature Method)

式(3)、式(4)における $\boldsymbol{V}^T \boldsymbol{f}^{\text{int}}(\boldsymbol{V}\boldsymbol{a})$ 、 $\boldsymbol{V}^T \boldsymbol{K}(\boldsymbol{V}\boldsymbol{a}) \boldsymbol{V}$ について、式(5)、式(6)のように要素アセンブリの形で考える。

$$\boldsymbol{V}^T \boldsymbol{f}^{\text{int}} = \sum_{e=1}^{N_e} \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e^{\text{int}} \quad (5)$$

$$\boldsymbol{V}^T \boldsymbol{K} \boldsymbol{V} = \sum_{e=1}^{N_e} \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{K}_e \boldsymbol{L}_e \boldsymbol{V} \quad (6)$$

Algorithm 1 greedy NNLS

```

1:  $E \leftarrow \emptyset$ 
2:  $w \leftarrow \mathbf{0}$ 
3:  $r \leftarrow b$ 
4: while  $\|\mathbf{G}w - b\|_2 > \tau \|b\|_2$  do
5:    $\mu \leftarrow \mathbf{G}^T r$ 
6:    $E \leftarrow E \cup \{\text{max\_value\_index } (\mu)\}$ 
7:    $w \leftarrow \text{NNLS } (G_E, b)$ 
8:    $r \leftarrow b - G_E w$ 
9: end while

```

$\boldsymbol{f}_e^{\text{int}} \in \mathbb{R}^{n_u^e}$ 、 $\boldsymbol{K}_e \in \mathbb{R}^{n_u^e \times n_u^e}$ はそれぞれ、要素内力ベクトル、要素接線剛性行列であり、 n_u^e は要素接線剛性行列の大きさを表す。また、 $\boldsymbol{L}_e \in \mathbb{R}^{n \times n_u^e}$ は要素 e の全体領域への寄与を表すブール行列である。ここで、各要素にスペースな重みベクトル $w = [w_1, w_2, \dots, w_{N_e}]^T \in \mathbb{R}^{N_e}$ を付与することで、式(7)、式(8)のように少ない要素のみで次元削減モデルの構築を行う。すなわち、重みベクトル w における 0 でない要素の数がサンプリングによって選定された要素の数となり、0 でない重みとそれに対応する要素のみの情報が次元削減モデル構築に用いられることになる。

$$\boldsymbol{V}^T \boldsymbol{f}^{\text{int}} \approx \sum_{e=1}^{N_e} w_e \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e^{\text{int}} \quad (7)$$

$$\boldsymbol{V}^T \boldsymbol{K} \boldsymbol{V} \approx \sum_{e=1}^{N_e} w_e \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{K}_e \boldsymbol{L}_e \boldsymbol{V} \quad (8)$$

各要素の重み w_e は Farhat らによって提案されたアクティビセット法の一種である greedy NNLS[3] により経験的に決定される。最初に、縮約した内力の要素寄与度を表すベクトル $\boldsymbol{g}_{ie} \in \mathbb{R}^k$ と、その要素ごとの和である $\boldsymbol{b}_i \in \mathbb{R}^k$ を式(9)で定義する。なお、 N_s は、POD 基底構築に使用する snapshot 数である。

$$\boldsymbol{g}_{ie} = \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e^{\text{int}} (\boldsymbol{L}_e^T \boldsymbol{V} \boldsymbol{a}_i), \quad \boldsymbol{b}_i = \sum_{e=1}^{N_e} \boldsymbol{g}_{ie}, i \in \{1, 2, \dots, N_s\} \quad (9)$$

さらに、式(9)を用いてブロック行列 $\boldsymbol{G} \in \mathbb{R}^{kN_s \times N_e}$ 、ベクトル $\boldsymbol{b} \in \mathbb{R}^{kN_s}$ が式(10)で定義される。

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_{11} & \cdots & \boldsymbol{g}_{1N_e} \\ \vdots & \ddots & \vdots \\ \boldsymbol{g}_{N_s 1} & \cdots & \boldsymbol{g}_{N_s N_e} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_{N_s} \end{bmatrix} \quad (10)$$

以上を用いて NNLS (Non-Negative Least Square) [4] が式(11)で定義される。

$$\boldsymbol{w} = \arg \min_{\alpha \geq 0} \|\boldsymbol{G}\boldsymbol{\alpha} - \boldsymbol{b}\|_2^2 \quad (11)$$

ここで、 $\boldsymbol{\alpha} \in \mathbb{R}^{N_e}$ は係数ベクトルである。本研究における ECM の重みは、algorithm 1 で表される、NNLS を用いたアクティビセット法である greedy NNLS によって決定される。なお、algorithm 1 の τ は greedy NNLS における閾値である。例として立方体有限要素モデル

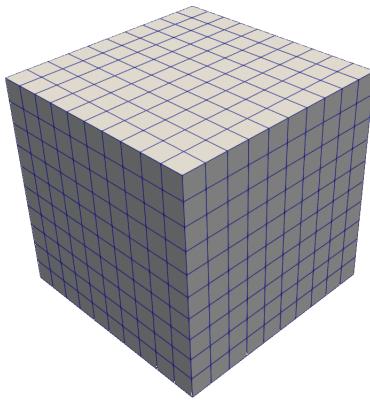


図-1 立方体モデルの有限要素メッシュ

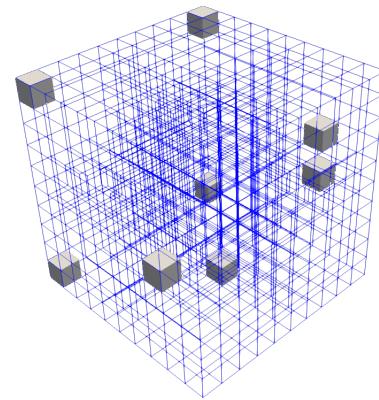


図-3 立方体モデルのサンプリング要素

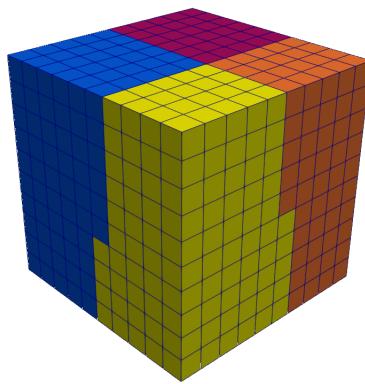


図-2 立方体モデルの有限要素メッシュ(4領域分割時)

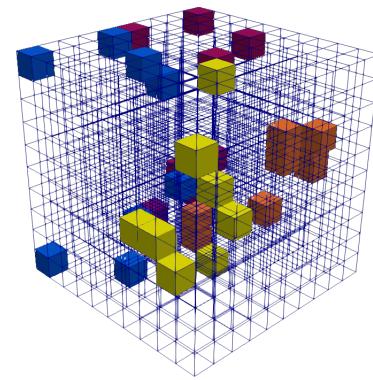


図-4 立方体モデルのサンプリング要素(4領域分割時)

図-1を考える。そこに本手法 ECM の greedy NNLS によって決定される重みを適用した場合、次元削減モデル構築に必要な要素は図-3 のようになる。

(3) 領域分割法における ECM

ECM の領域分割法に基づいた並列環境への適用においては、局所領域ごとに greedy NNLS を解くことで通信を必要としない並列化 ECM[3] が Farhat らによって提案されている。

ここで、対象とする解析領域 Ω が式 (12) のように各局所領域 $\Omega^{(i)}$ に重複なく分割されているとする。 N_{dom} は領域分割数である。

$$\Omega = \bigcup_i^{N_{\text{dom}}} \Omega^{(i)} \quad (12)$$

そこで、式 (13) のようにブロック行列 \mathbf{G} 、ベクトル \mathbf{b} を分割領域毎に定義し、局所的な NNLS 式 (14) を構築する。

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^{(1)} & \mathbf{G}^{(2)} & \dots & \mathbf{G}^{(N_{\text{dom}})} \end{bmatrix}, \quad \mathbf{b} = \begin{cases} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{b}^{(N_{\text{dom}})} \end{cases} \quad (13)$$

$$\mathbf{w} = \sum_{i=1}^{N_{\text{dom}}} \mathbf{L}^{T(i)} \mathbf{w}^{(i)}, \quad \mathbf{w}^{(i)} = \arg \min_{\alpha^{(i)} \geq 0} \|\mathbf{G}^{(i)} \boldsymbol{\alpha}^{(i)} - \mathbf{b}^{(i)}\|_2^2 \quad (14)$$

ここで、 $\mathbf{L}^{T(i)} \in \mathbb{R}^{n \times n_{\text{loc}}}$ は局所領域ごとの重みベクトルを全体領域サイズへ変換するブール行列であり、 n_{loc} は局所領域ごとの要素数である。これら局所的な NNLS によって領域ごとに Algorithm 1 に基づいた greedy NNLS を解き、局所領域ごと重みベクトル $\mathbf{w}^{(i)}$ を得る。また、これら局所領域毎の greedy NNLS には式 (15) で表される領域毎の閾値 $\tau^{(i)}$ が用いられる。

$$\sum_{i=1}^{N_{\text{dom}}} \tau^{(i)} \|\mathbf{b}^{(i)}\|_2 \approx \tau \|\mathbf{b}\|_2 \Rightarrow \tau^{(i)} = \left(\frac{\|\mathbf{b}\|_2}{N_{\text{dom}} \|\mathbf{b}^{(i)}\|_2} \right) \tau \quad (15)$$

2.(2) と同様に図を用いた適用例を示す。立方体モデル図-1 の立方体モデルについて、領域分割法(4領域)を

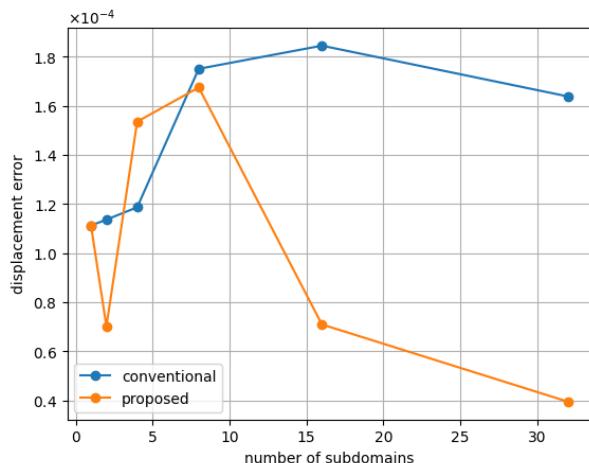


図-5 領域分割数と変位解の誤差の関係

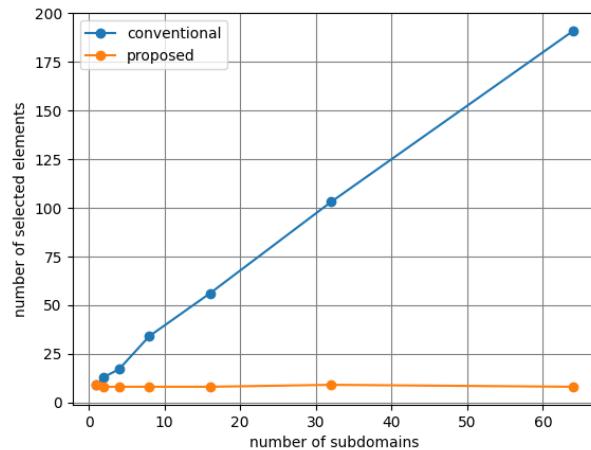


図-7 領域分割数と大域的なサンプリング要素数の関係

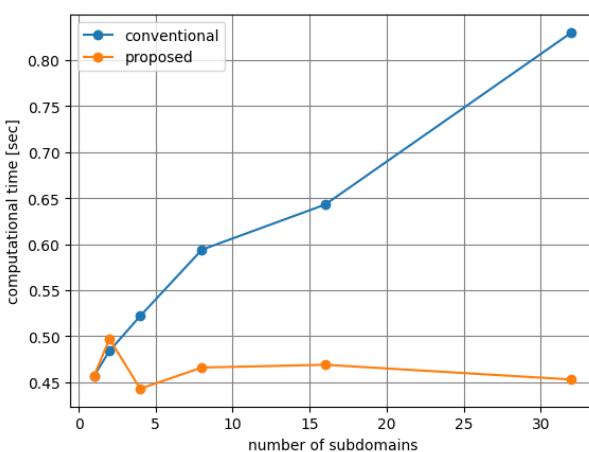


図-6 領域分割数と online 計算時間 [sec] の関係

適用した時のメッシュを図-2に示す。式(13)の並列化 ECM を立方体モデル図-2に適用した場合、図-4に示すように領域ごとに要素が選択される。

3. 提案手法：階層的サンプリングを用いた ECM

2.(3)の手法では、通信コストが発生せず、greedy NNLS の計算において良好な並列化効率を実現する。しかし、選択される大域的な要素数は領域分割数と共に増大してしまい、大規模な並列計算においては本来の目的である射影演算の計算コスト削減効果を小さくしてしまう。そのため本研究では、階層的サンプリングによる重みベクトル取得を行うことで、領域分割数に関わらず領域分割を行わない時と同等レベルの要素削減が行われた重みベクトルを作成する greedy NNLS アルゴリズムを提案する。具体的には、図-2、図-4のように一段階目で Farhat らの従来手法 [3] を適用し領域ごとに独立に要素をサンプリングした後、領域ごとのサンプリング要素の情報を MPI における gatherv 演算によって一つの計算ノードに統合し、統合された要素のみをアクティブセットとした状態で再度 greedy NNLS を解くことで、図-3程度まで要素の削減を行う。これ

表-1 モデル諸元

解析モデル	片持ち梁
片持ち梁大きさ [m]	(100.0 × 1.0 × 1.0)
要素数	4,816
節点数	2,700
ヤング率 [GPa]	206
ポアソン比	0.3
質量密度 [kg/m ³]	7,870

表-2 解析条件

外力 (自由端に付与)	$\sin(2\pi \cdot t/T)$
時間刻み	0.5 sec
継続時間	50.0 sec
計算ステップ数	100
offline 計算ステップ数	30
時間積分法	Newmark- β 法
offline 線形ソルバ	共役勾配法 (CG 法)
online 線形ソルバ	LU 分解による直接法
CG 法収束判定閾値	1.0×10^{-8}

により領域分割法による分散メモリ型並列環境での計算時においても領域分割を行わない時と同等の要素削減を実現する。

本提案手法は 2.(3) の Farhat らの手法 [3] と比較すると greedy NNLS アルゴリズム部分に追加で通信コストをかけることになるが、その後の計算コスト削減効果を鑑みて許容するものとする。

4. 数値例

領域分割法による片持ち梁の曲げ問題の並列解析を対象に、3. で述べた提案手法と 2.(3) の領域ごとに独立にサンプリングを行う Farhat らの従来手法を比較し、提案手法の性能評価を行った。なお、本研究ではグラフ構造を用いた領域分割法に基づく並列有限要素法 [5] を研

究基盤として利用した。モデル諸元を表-1に、解析条件を表-2に示す。また、計算機についてはLinuxサーバー(CPU : Intel Xeon Gold 6248、メモリ : 786 GB、OS : Ubuntu 20.04.3 LTS)を使用した。なお、片持ち梁には自由端に集中荷重を付与することで曲げ解析を行った。その際、時間と外力の関係は周期関数(sin関数)を用いた。なお、表-2におけるt, Tはそれぞれ現時刻と周期を表す。集中荷重の大きさの最大値は10Nとした。また、図-3の誤差指標としては、式(16)に示す通常の有限要素法による数値解とのL2ノルム誤差 ε を使用した。

$$\varepsilon = \frac{\|\mathbf{u}_{\text{FEM}} - \mathbf{u}_{\text{ROM}}\|_2}{\|\mathbf{u}_{\text{FEM}}\|_2} \quad (16)$$

ここで、 \mathbf{u}_{FEM} は通常の有限要素法で解いた各時間ステップにおける変位解の平均とし、 \mathbf{u}_{ROM} は提案手法を用いて解いた各時間ステップにおける変位解の平均とした。

式(16)で示した誤差指標を用い、各領域分割数(1, 2, 4, 8, 16, 32)でFarhatらによる並列ECMを用いた従来手法(conventional)と本研究で提案する階層的なサンプリングを行ったECMを用いたROM解析による求解を行う提案手法(proposed)の比較を図-5で行った。なお、領域分割数1とは、領域分割を行わない手法を指す。図-5の縦軸は 10^{-4} オーダーであり、両手法とも実用的な誤差の範囲に収まっていることが分かる。

また、図-6は従来手法と提案手法のonline過程での計算時間と領域分割数の関係を示し、図-7は従来手法と提案手法の縮小メッシュの要素数と領域分割数の関係を示す。ここでonline過程とは、POD-Galerkin法におけるsnapshot取得後の次元削減モデルによる計算を行う時間ステップの集合を指す。ROM解析では一般に、snapshotの収集過程をoffline過程、ROM適用の過程をonline過程と呼称する。(3)で述べたように、従来手法は領域分割数が増加するごとに大域的なサンプリング要素数が増え、それにより計算時間も増加している。対して、本研究の提案手法では領域数分割に関わらず領域分割を行わない時と同程度の要素数まで削減出来ている。また、計算時間についても領域数增加に関わらずほぼ一定の値を示している。これらから、提案手法では領域分割数増加に伴う大域的なサンプリング要素数の増加が抑えられ、またそれにより計算時間増加も抑えられていることが分かる。

5. 結論

本研究では、パラメータスタディの計算コスト削減に重要なROMにおいて、ボトルネックとなる射影演算コスト削減を実現するために重要なECMに対し、分散メモリ型並列計算環境での計算性能向上を目的とした階層的なサンプリングアルゴリズムを提案した。基礎検討としての片持ち梁の曲げ解析において、提案手法は従来手法と比較して良好な精度を維持したまま最大で約49%の計算時間削減を実現した。しかし、より32

領域分割よりも大きな領域分割数において、またより複雑な形状を持つモデルにおいての適用は本提案手法の有効性は未確認である。従って今後は、大規模並列問題や複雑な形状を持つモデル、あるいは他の数値解析手法への適用に向けた有効性の評価・改良を課題とする。

謝辞: 本研究は、JST創発的研究支援事業JP-MJFR21S、JSPS科研費23K16891並びに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。ここに記して敬意を表する。

参考文献

- [1] Hijazi, Saddam, et al. "Data-driven POD-Galerkin reduced order model for turbulent flows." Journal of Computational Physics 416 (2020): 109513.
- [2] Hernandez, Joaquin Alberto, Manuel Alejandro Caicedo, and Alex Ferrer. "Dimensional hyper-reduction of nonlinear finite element models via empirical cubature." Computer methods in applied mechanics and engineering 313 (2017): 687-722.
- [3] Farhat, Charbel, et al. "Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency." International Journal for Numerical Methods in Engineering 98.9 (2014): 625-662.
- [4] Lawson, Charles L., and Richard J. Hanson. Solving least squares problems. Society for Industrial and Applied Mathematics, 1995.
- [5] 森田直樹, et al. "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): 20241008-20241008.