ADVENTURE_Magnetic による，
移動体を含む回転機の大規模並列有限要素解析

杉本振一郎

八戸工業大学 工学部

１．はじめに

ADVENTURE プロジェクト[1]では，数万ノード規模の超並列計算機環境において 1,000 億自由度規模の大規模電磁界解析を行うことを目的に，並列電磁界解析ソルバ ADVENTURE_Magnetic (AdvMag)の開発を進めている。AdvMag のターゲットアプリの一つとして，回転機の大規模並列解析に 2016 年度より取り組んでいる。回転子という移動体を含む回転機の非定常有限要素解析は並列環境での効率的な取り扱いが難しく，スーパーコンピュータを有効に活用できていない分野の一つである。そこで階層型領域分割法に新たな領域分割技術を導入し，並列数に応じて計算時間を短縮することのできるソルバを開発し，数億～数十億自由度の回転機の非定常有限要素解析を効率よく行えるようになることを目指している。本稿では，2018 年度の若手。女性利用課題に採択された研究の中で，主に周期境界条件を階層型領域分割法で効率的に扱えるようになった手法に焦点をあてて紹介する。

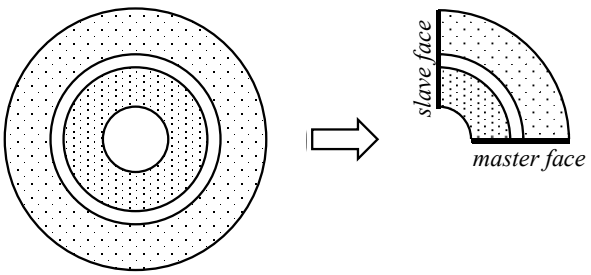

図 1. 周期境界条件による解析領域の削減。

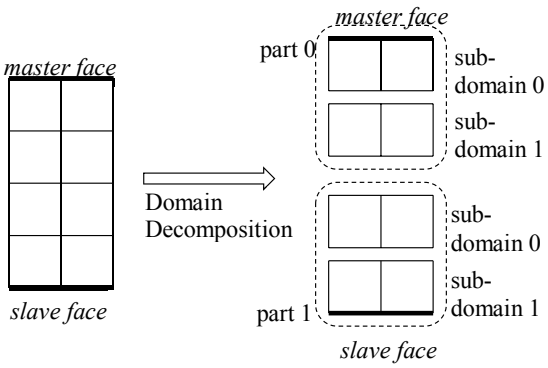

図 2. 階層型の領域分割。

回転機の数値解析では計算時間の短縮を目的として，形状および電磁界の周期性を利用して回転機の 1/2 や 1/4 などを解析対象として自由度を抑えるといったことがしばしば行われる(図 1)。このときマスター面およびスレーブ面では同じ値または-1 倍の値を取るという周期境界条件を考慮する。この条件は両面上の対応する自由度を同一のものとすることで扱うことができる。

並列環境において全体の有限要素行列を作り，それを並列反復法で解く一般的な並列手法であれば周期境界条件を考慮することは難しくないが，スーパーコンピュータで回転機の解析を効率よく行うのは難しい。一方，階層型領域分割法はスーパーコンピュータで回転機フルモデルの解析を効率よく行えるが，周期境界条件を考慮するのは難しい。これは，階層型領域分割法では解析領域を分割したデータを part ごとにコンピュータが分散して持ち(図 2)，part 内でさらに分割された subdomain ごとに小さな行列を作るために対応する自由度が同じ行列に含まれず，通常の方法では同一のものとして扱うことが難しいためである。

そこでまず動かない対象において階層型領域分割法で周期境界条件を効率よく考慮できる手法を提案した。周期境界条件を考慮しない場合と考慮した場合について，スーパーコンピュータ上でウィークスケーリングを行って計算時間を比較し，提案手法により周期境界条件を考慮しても，考慮しない場合と遜色ない性能が得られることがわかった。以下では提案手法の概要について述べ，数値計算例を示す。

２．階層型領域分割法での周期境界条件の考慮[2]

２．１．一般的な周期境界条件

階層型領域分割法を用いない通常の有限要素解析では，境界上の対応する自由度を同一の自由度と見なして要素係数行列の自由度を置き換えることで，周期境界条件を考慮できる[3]。

図 3 に上面をマスター面，下面をスレーブ面とする，周期境界条件を考慮する解析領域の例を示す。なお実際の解析では四面体要素を用いるが，本稿ではイメージしやすくするため，これらの図は 2 次元の四角形要素で描く。

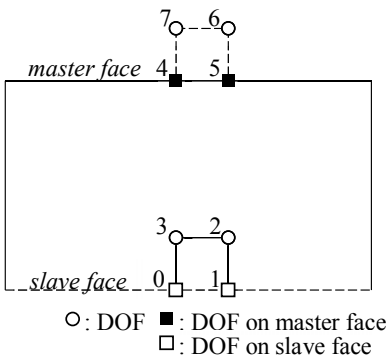

図 3. 周期境界条件を考慮する境界上の自由度。

要素分割も周期的に行うと仮定すれば，スレーブ面上の 0 番と 1 番の自由度を含む要素 0123 と合同で，マスター面上の 4 番と 5 番の自由度を含む要素 4567 が本来は存在すると考えられる。ここで，0 番と 4 番，1 番と 5 番の値はそれぞれ等しいか，-1 倍になる。

解くべき有限要素方程式を行列形式で表す。

$$Ku = f. \tag{1}$$

ここで，Kは係数行列，uは未知自由度，fは右辺ベクトルを表す。要素 0123 では，要素係数行列を式(2)のように書けるとする。

$$K_{e0}u_{e0} = f_{e0}. \tag{2}$$

ここで，

$$K_{e0} = \begin{bmatrix} K_{00} & K_{01} & K_{02} & K_{03} \\ K_{10} & K_{11} & K_{12} & K_{13} \\ K_{20} & K_{21} & K_{22} & K_{23} \\ K_{30} & K_{31} & K_{32} & K_{33} \end{bmatrix},\ u_{e0} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix},\ f_{e0} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

である。また，要素 4567 の要素係数行列を

$$K_{e1}u_{e1} = f_{e1}, \tag{3}$$

とすると，要素 0123 と要素 4567 は合同であるため，

$$K_{e0} = K_{e1}, \tag{4}$$

である。一方，周期境界条件より，

$$u_{e0} = cu_{e1}. \tag{5}$$

である。ここで，c はマスター面とスレーブ面の値が等しいときは 1，-1 倍になるときは-1 である。さらに，電流や永久磁石の磁化の方向は周期条件と同じように，マスター面とスレーブ面の値が等しいときは同じ向き，-1 倍になるときは逆向きになるため，

$$f_{e0} = cf_{e1}, \tag{6}$$

である。

メッシュも含めて周期的であると仮定するため，マスター面，スレーブ面のどちらか一方を解析領域外とする必要がある。ここではスレーブ面を解析領域外とする。その結果，0 番と 1 番の自由度は係数行列Kから消去しなければならないため，これを$u_0 = cu_4$, $u_1 = cu_5$で置き換える。ただし，未知自由度にc がかかっていると求解過程で不都合が生じる。式(2)の自由度を置き換えてそれぞれの行について展開すると以下のようになる。

$$cK_{00}u_4 + cK_{01}u_5 + K_{02}u_2 + K_{03}u_3 = f_0, \tag{7a}$$

$$cK_{10}u_4 + cK_{11}u_5 + K_{12}u_2 + K_{13}u_3 = f_1, \quad (7b)$$

$$cK_{20}u_4 + cK_{21}u_5 + K_{22}u_2 + K_{23}u_3 = f_2, \quad (7c)$$

$$cK_{30}u_4 + cK_{31}u_5 + K_{32}u_2 + K_{33}u_3 = f_3. \quad (7d)$$

ここで行列の対称性を維持するために式(7a), (7b)の両辺を$c = \pm 1$で割ると, 最終的に要素 0123 の要素係数行列は

$$\begin{bmatrix} K_{00} & K_{01} & K_{02}/c & K_{03}/c \\ K_{10} & K_{11} & K_{12}/c & K_{13}/c \\ cK_{20} & cK_{21} & K_{22} & K_{23} \\ cK_{30} & cK_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} u_4 \\ u_5 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_0/c \\ f_1/c \\ f_2 \\ f_3 \end{bmatrix}, \quad (8)$$

と書ける。これを 0 番と 1 番の自由度を消去した係数行列Kに足す。他のスレーブ面上の自由度についても，要素係数行列の該当する自由度をマスター面上の自由度に置き換えて係数行列Kに足しこんでいく。

２．２．階層型領域分割法

階層型領域分割法は領域分割法[4]-[6]を並列計算機環境に効率よく実装するための 1 手法である。大規模問題を効率よく数値計算することのできる手法としてよく知られており，分散メモリ環境で良好な並列効率を得られることが期待できる[7]。

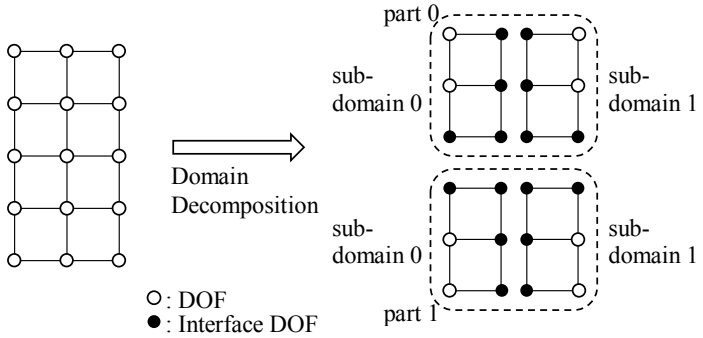

図 4. 階層型の領域分割。

階層型領域分割法では要素の重なりがないように解析領域を part と subdomain からなる階層構造に分割する(図 4)。

$$\Omega = \bigcup_{i=0}^{N_p-1} \bigcup_{j=0}^{N_s-1} \Omega^{(i,j)}. \quad (9)$$

ここで，$\Omega^{(i,j)}$は part iの中の subdomain jを表す。上付き添え字(i,j)は領域$\Omega^{(i,j)}$に関連する項目

であることを表す。N_p，N_sはそれぞれ part 数，part あたりの subdomain 数を表し，N_sは全 part で等しいとする。図中の黒丸は part，subdomain 間で共有される自由度であり，インターフェース自由度と呼ばれる。式(1)に階層型領域分割法を適用すると，次のように書ける。

$$\begin{bmatrix} K_{II} & K_{IB} \\ K_{IB}^T & K_{BB} \end{bmatrix} \begin{bmatrix} u_I \\ u_B \end{bmatrix} = \begin{bmatrix} f_I \\ f_B \end{bmatrix}. \tag{10}$$

ここで，u_Bはインターフェース自由度であり，

$$K_{II} = \begin{bmatrix} K_{II}^{(0,0)} & & & & & 0 \\ & \ddots & & & & \\ & & K_{II}^{(0,N_s-1)} & & & \\ & & & K_{II}^{(1,0)} & & \\ & & & & \ddots & \\ 0 & & & & & K_{II}^{(N_p-1,N_s-1)} \end{bmatrix},$$

$$K_{IB} = \begin{bmatrix} K_{IB}^{(0,0)} R_B^{(0,0)} \\ \vdots \\ K_{IB}^{(0,N_s-1)} R_B^{(0,N_s-1)} \\ K_{IB}^{(1,0)} R_B^{(1,0)} \\ \vdots \\ K_{IB}^{(N_p-1,N_s-1)} R_B^{(N_p-1,N_s-1)} \end{bmatrix},$$

$$K_{BB} = \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_s-1} R_B^{(i,j)T} K_{BB}^{(i,j)} R_B^{(i,j)},$$

$$f_i = \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_s-1} R_I^{(i,j)T} f_I^{(i,j)},$$

$$f_B = \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_s-1} R_B^{(i,j)T} f_B^{(i,j)},$$

である。下付添え字 I, Bはそれぞれ subdomain 内部の自由度，インターフェース自由度に関する項であり，$R_I^{(i,j)}$，$R_B^{(i,j)}$は自由度をそれぞれ subudomain 内部の自由度，またはインターフェース自由度へと制限する 0-1 行列である。式(10)より自由度をインターフェース自由度に静的縮約したインターフェース問題が以下のように得られる。

$$Su_B = g. \tag{11}$$

ただし，

$$S = \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_s-1} R_B^{(i,j)T} S^{(i,j)} R_B^{(i,j)},$$

$$S^{(i,j)} = K_{BB}^{(i,j)} - K_{IB}^{(i,j)T} \left(K_{II}^{(i,j)} \right)^{\dagger} K_{IB}^{(i,j)},$$

$$g = \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_s-1} R_B^{(i,j)T} \left(f_B^{(i,j)} - K_{IB}^{(i,j)T} \left(K_{II}^{(i,j)} \right)^{\dagger} f_I^{(i,j)} \right),$$

であり，Sはシュア補元行列，$S^{(i,j)}$は領域$\Omega^{(i,j)}$におけるローカルシュア補元行列である。ここで$K_{II}^{(i,j)}$は特異行列であるため，式中の$\left(K_{II}^{(i,j)}\right)^{\dagger}$は一般化逆行列である。階層型領域分割法では並列環境下で式(11)を共役勾配(Conjugate Gradient: CG)法や共役直交共役勾配(Conjugate Orthogonal Conjugate Gradient: COCG)法などの反復法で解き，得られたu_Bを Dirichlet 条件として式(12)を解くことで，全体の解を得る。

$$K_{II}^{(i,j)} u_I^{(i,j)} = f_I^{(i,j)} - K_{IB}^{(i,j)} R_B^{(i,j)} u_B \qquad i = 0, \ldots, N_p - 1, j = 0, \ldots, N_S - 1. \tag{12}$$

階層型領域分割法では，それぞれの MPI プロセスが 1 つの part を担当する。そのため，part 間で共有されるインターフェース自由度についてプロセス間通信でデータを送受信しなければならない。階層型領域分割法では part と MPI プロセスは 1 対 1 で対応するので，簡単のために本稿では 0 番目と 1 番目の MPI プロセスをそれぞれ part 0，part 1 と呼ぶ。

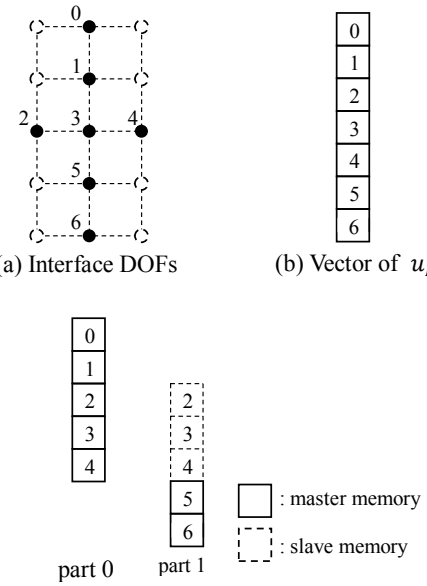

図 5. インターフェース自由度のメモリ空間。

図 4 のケースでは，7 つのインターフェース自由度(図 5 (a))があり，u_Bは 7 つの要素をもつベクトル(図 5 (b))である。しかし解析領域は分割されているため，ベクトルのデータはそれぞれの MPI プロセスに分散して記憶されている。さらに，part 間で共有されているインターフェース自由度はそれぞれの MPI プロセスにメモリが確保されている。ここで，同一のインターフェース自由度について複数のメモリが確保されているため，いずれか一つをマスターメモリとして定める。本稿では，同一のインターフェース自由度に対してメモリが確保されている part の中で，

part 番号の最も小さいものが持つメモリをマスターメモリとする。2〜4 番のインターフェース自由度は part 0 と part 1 に共有されているので，part 0 のメモリがマスターメモリとなり，part 1 のメモリはスレーブメモリとなる(図 5 (c))。それぞれの part では，マスターメモリ，スレーブメモリを持つインターフェース自由度をそれぞれ基本インターフェース自由度，従属インターフェース自由度と呼ぶ。part 0 ではすべてが基本インターフェース自由度である。一方，part 1 では 2〜4 が従属インターフェース自由度，5，6 が基本インターフェース自由度である。プロセス間通信は，シュア補元行列の行列ベクトル積を行う際に part 間で共有されているインターフェース自由度について行われる。まず従属インターフェース自由度の情報が，その基本インターフェース自由度を持つ part へと送られる。その後，それぞれの part で必要な処理をした結果を基本インターフェース自由度を持つ part からその従属インターフェース自由度を持つ part へ送り，スレーブメモリへコピーする。

プロセス間通信を行うため，通信テーブルを作る。まず，インターフェース自由度の part 内でのローカルな通し番号を，従属インターフェース自由度，基本インターフェース自由度の順につける(図 6 左)。このとき，part 0 の 2〜4，part 1 の 0〜2 について，part 0，part 1 が持つ通信テーブルはそれぞれ図 6 右上と右下のようになる。それぞれの通信テーブルにおいて，「part 0」「part 1」は通信相手である。また，「:」に続く数字は，その通信相手とデータを送受信する必要があるインターフェース自由度の数である。例えば，part 0 が part 0 と通信する必要はないため，「part 0:」の次の数字は「0」である。また，part 1 とは 3 つのインターフェース自由度について通信を行うので，「part 1:」の次の数字は「3」となる。これらの数字が 1 以上である場合には，次の行にデータを送受信するインターフェース自由度のローカルな通し番号を必要なだけ並べる。ここで，part 0 の 0 と 1 のように，同じ part 内の subdomain 間でしか共有していないインターフェース自由度は通信テーブルには含まれない。プロセス間通信はこれらの通信テーブルに基づいて行われる。

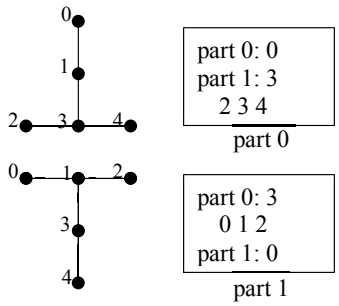

図 6. それぞれの part でのインターフェース自由度の通し番号と通信テーブル。

２．３．階層型領域分割法での周期境界条件

階層型領域分割法ではそれぞれの subdomain で係数行列$K_{II}^{(i,j)}$を作成する。そのため，図 7 のようにマスター面とスレーブ面の対応する自由度がそれぞれ別の subdomain に位置した場合，通常の方法では周期境界条件を考慮できない。階層型領域分割法では計算時間や使用メモリ量の観点から subdomain あたり 100 要素と細かく分割するが，本稿で対象とする規模は 100 万自由度(要素数 80 万)以上であるため，同じ subdomain 内に対応する自由度が位置することはまれであ

る。よって，別の手段で周期境界条件を考慮しなければならない。

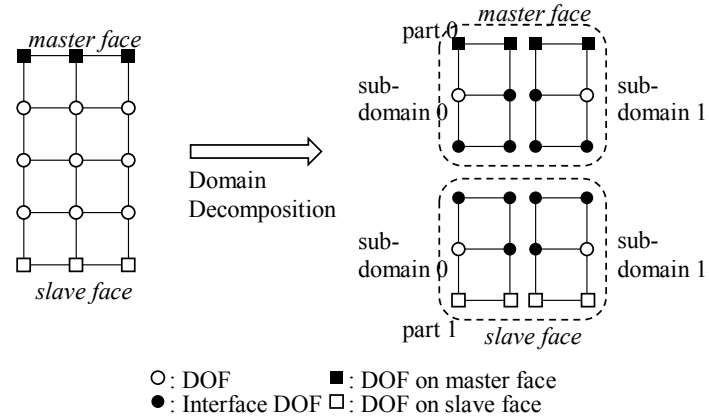

図 7. 階層型の領域分割と周期境界条件。

階層型領域分割法では subdomain 間で共有される自由度をインターフェース自由度と呼ぶ。これらはそれぞれの subdomain に分散されているが，対応するインターフェース自由度は領域分割前にはもともと同一の自由度である。そこで本稿では，周期境界条件を考慮する自由度もインターフェース自由度とすることで，対応する自由度を同一の自由度として扱う。

周期境界条件を考慮しない場合には，同一のインターフェース自由度に対してメモリが確保されている part の中で part 番号の最も小さいものが持つメモリをマスターメモリ，その part ではそのインターフェース自由度を基本インターフェース自由度，それ以外の part では従属インターフェース自由度としていた。そこでこれを踏襲するため，実際のマスター面，スレーブ面に関係なく，周期境界条件で同一として扱う自由度が所属している part の中で part 番号の最も小さい part においてその自由度を基本インターフェース自由度，それ以外の part では従属インターフェース自由度とする。図 7 の例ではマスター面上の自由度はすべて part 0，スレーブ面上の自由度はすべて part 1 に属しているため，そのままそれぞれ基本インターフェース自由度，従属インターフェース自由度となるが，仮に part 番号が逆になればインターフェース自由度の種類も逆になる。実際の解析ではより複雑な領域分割が行われるためマスター面，スレーブ面それぞれに基本インターフェース自由度，従属インターフェース自由度が入り乱れることになる。しかし大規模な解析の後処理は subdomain 単位で行われる。それぞれの subdomain に分散してデータを保持しているため，これらが入り乱れることにより支障が生じることはない。

周期境界条件を考慮する場合の通信テーブルについて述べる。インターフェース自由度の part 内でのローカルな通し番号は，周期境界条件を考慮しない場合と同様に従属インターフェース自由度，基本インターフェース自由度の順につける(図 8 左)。ここでは周期境界条件を考慮する自由度であるかどうかを考慮する必要はない。part 0 ではすべてが基本インターフェース自由度なので，左上から順に番号を付けている。part 1 では真ん中のインターフェース自由度(6 番)のみが基本インターフェース自由度で，それ以外が従属インターフェース自由度なので，真ん中の自由度をとばして左上から順に番号を付け，最後に真ん中の自由度に番号を付けている。この通し番号に基づいて作成した通信テーブルが図 8 右である。この通信テーブルを用いて，通常の階層

型領域分割法と同様にプロセス間通信を行う。

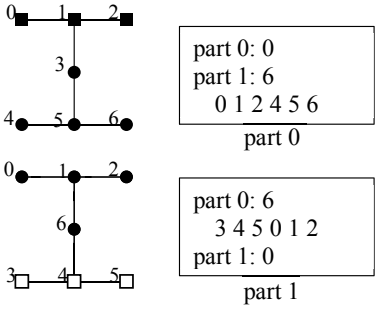

図 8. それぞれの part でのインターフェース自由度の通し番号と
周期境界条件を含む通信テーブル。

提案手法におけるc の取り扱いについて述べる。従属インターフェース自由度の情報をその基本インターフェース自由度を持つ part へと送る際, 基本インターフェース自由度の情報を受け取ってスレーブメモリへコピーする際に，スレーブメモリを持つ part がc をデータにかける。本稿の例では，part 1 が送信データにc をかけてから part 0 へ送り，part 0 から送られたデータを part 1 が受け取ってスレーブメモリにコピーする際にc を受信データにかける。

提案手法は通信テーブルの作成部分と，c をかける部分のみが通常の階層型領域分割法のアルゴリズムと異なる。そのため実装が簡便で，階層型領域分割法が持つ処理の分散性や演算性能をほぼそのまま引き継ぐことができると期待できる。

３．数値計算例[2]

時間調和渦電流問題に周期境界条件を考慮する階層型領域分割法を適用して，提案手法の性能を検証する。数値計算例として無限長ソレノイドコイルを用いた渦電流解析の精度検証問題[3]を用いる。導体部の半径は 0.1 m であるとする。磁気抵抗率νは解析領域全体で$1/4\pi \times 10^7$ m/H，導体部の導電率σは7.7×10^6 S/m，角周波数ωは$2\pi \times 60$ rad/s とする。コイルに流れる強制電流密度の実部，虚部の大きさは 50，0 [A/m^2]とする。問題の対称性を考慮し，中心角 20°，高さ 0.2 m の領域を解析対象のモデルとする。このモデルでは周期境界条件を与えなくても，与えても同じ解を得ることができる。そこで, 周期境界条件を与えない場合と与えた場合でそれぞれ解析し，性能を検証する。周期境界条件を与える場合には，このモデルのコイルの軸方向に直角な面に与える。

解析には東京大学情報基盤センターOakforest-PACS スーパーコンピュータ(OFP)[8]を用いる。OFP は 8,208 ノードの Fujitsu PRIMERGY CX600 M1 (68 コアの Intel Xeon Phi 7250，96GB memory 搭載)を Intel Omni-Path で接続したスーパーコンピュータである。本稿では最大 128 ノード(8,704 コア)を用いてウィークスケーリングを行う。計算量を一定にするため，インターフェース問題の求解に用いる COCG 法の反復回数を 200 回とする。また 1 ノードあたり 100 万自由度となるメッシュを用いる。そのため最小の自由度は 1 ノードのときに 100 万，最大の自由度は 128 ノードのときに 1.28 億となる。

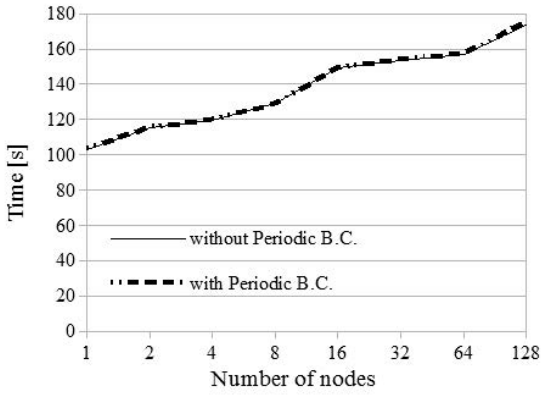

図 9. 計算時間。

図 9 に計算時間とノード数の関係を示す。一般的に，新たなアルゴリズムを追加すると計算時間は長くなる。しかし提案手法では，階層型領域分割法の主要なアルゴリズムには手を加えておらず，通信量がわずかに増える程度である。そのため，周期境界条件を考慮しない場合に比べて計算時間の増加が 1%未満に抑えられており，提案手法は周期境界条件を考慮しない従来の階層型領域分割法と遜色ない性能をスーパーコンピュータ上で得られることが確認できた。よって，提案手法と階層型領域分割法による回転機の並列解析手法を組み合わせることで，解析領域を減らした回転機モデルを効率よく解析でき，さらに計算時間を短縮できると期待できる。

4．おわりに

今後は階層型領域分割法で効率的に回転機を扱うことを可能とした並列解析手法と提案手法を組み合わせ，回転機の解析の計算時間をより短縮することを目指すとともに，リニアモータなど直進運動する機器へも適用範囲を広げていく。

謝　辞

本研究は東京大学情報基盤センター若手・女性利用者推薦, JSPS 科研費 17H02829, 17H03256, 一般財団法人 青森県工業技術教育振興会の助成を受けて実施された。

ここに記し，感謝の意を表す。

参　考　文　献

[1] ADVENTURE プロジェクト HP: http://adventure.sys.t.u-tokyo.ac.jp/jp/

[2] 杉本振一郎，階層型領域分割法での周期境界条件の効率的な考慮，電気学会論文誌 B, Vol.139，No.11，pp.637-642 (2019)

[3] 中田高義，高橋則雄：「電気工学の有限要素法　第 2 版」，森北出版 (1986)

[4] R. Glowinski, Q.V. Dinh and J. Periaux: “Domain Decomposition Methods for Nonlinear Problems in Fluid Dynamics”, Computer Methods in Applied Mechanics and Engineering, Vol.40, Issue 1, pp.27-109 (1983)

[5] A. Quarteroni and A. Valli: “Domain Decomposition Methods for Partial Differential Equations”, Clarendon Press, Oxford (1999)

[6] A. Toselli and O. Widlund: “Domain Decomposition Methods, Algorithms and Theory (Springer Series in Computational Mechanics)”, Springer (2004)

[7] R. Shioya and G. Yagawa: “Iterative Domain Decomposition FEM with Preconditioning Technique for Large Scale Problem”, ECM’99 Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour, pp.255-260 (1999)

[8] 東京大学情報基盤センター スーパーコンピューティング部門 HP：
https://www.cc.u-tokyo.ac.jp/