

# Wavelet 変換による反復前処理法の 高速多重極境界要素法への応用

WAVELET TRANSFORM PRECONDITIONING OF ITERATIVE SOLVER  
FOR FAST MULTIPOLE BOUNDARY ELEMENT METHOD  
IN TWO-DIMENSIONAL POTENTIAL PROBLEMS

福井卓雄<sup>1)</sup>, 藪田順子<sup>2)</sup>

Takuo FUKUI and Junko YABUTA

<sup>1)</sup> 福井大学工学部建築建設工学科 (〒 910-8507 福井市文京 3-9-1, E-mail: tak@taku.anc-d.fukui-u.ac.jp)

<sup>2)</sup> 福井大学工学部環境設計工学科 (〒 910-8507 福井市文京 3-9-1, E-mail: yabu@taku.anc-d.fukui-u.ac.jp)

A preconditioning method of iterative solvers for the fast multipole boundary element method in two-dimensional potential problems is proposed here. The method uses the diagonal information of the Haar wavelet transformed coefficient matrix of the boundary element method. To obtain the information, Haar wavelet transform on the quad tree is introduced, and computation is performed on the tree by using the multipole expansion and the local expansion coefficients of the Green function. Numerical test shows that the iteration time proposes  $N^{1/4}$  for  $N$  element problem.

**Key Words :** *Preconditioning, Iterative solver, Haar wavelet transform, Fast multipole boundary element method, Two-dimensional potential problem*

## 1. はじめに

本論文では、高速多重極境界要素法における反復法の効率を上げることが目的として、前論文<sup>1)</sup>で提案した wavelet 変換を利用する前処理法を高速多重極境界要素法に適用するための計算法を提案する。

文献<sup>1)</sup>では、2次元ポテンシャル問題を例として、代表的な境界値問題における境界要素反復解法の収束効率を数値的に検討し、Dirichlet 問題および混合境界値問題において、とくに、収束効率が悪くなること、wavelet 変換を利用した前処理法により、これらの問題における収束効率を改善することができることを示した。また、文献<sup>2)</sup>においては、波動散乱問題を Burton-Miller 方程式を使って解析する場合について、種々の前処理法の効果を数値的に検証し、この場合でも、wavelet 変換を利用した前処理法が規模の大きな問題に対して効果的であることが示唆された。

ここでは、2次元ポテンシャル問題を対象として、文献<sup>1)</sup>で提案された wavelet 変換を利用する前処理法を高速多重極境界要素法に適用するための計算法を提案する。問題は、wavelet 変換後の係数行列の情報を、行列を計算することなく、いかにして得るかである。ここでは、4分木上の wavelet 変換を導入し、多重極展開を利用してこの問題を解決する。以下、2. で文献<sup>1)</sup>の方法をまとめ、3. で4分木上での計算法を示し、4. で簡単な数値例により手法の確認を行なう。

## 2. Haar wavelet 変換と反復法の前処理

ここでは、文献<sup>1)</sup>において用いた、Haar wavelet 変換による前処理法について概説する。

### 2.1 境界要素法の離散化方程式

2次元ポテンシャル問題を対象とする。よく知られているように、境界要素法の離散化代数方程式は

$$\mathbf{A}\mathbf{s} = \mathbf{B}\mathbf{u} \quad (1)$$

となる。ここに、 $\mathbf{u}$ ,  $\mathbf{s}$  は、境界上の Dirichlet および Neumann データを表すベクトルであり、係数行列  $\mathbf{A}$ ,  $\mathbf{B}$  の成分は、一定要素近似および選点法を使うとき、

$$\{\mathbf{A}\}_{ij} = \frac{1}{2\pi} \int_{E_j} \log \frac{1}{|\mathbf{x}_i - \mathbf{y}|} ds_y \quad (2)$$

$$\{\mathbf{B}\}_{ij} = \frac{\delta_{ij}}{2} + \frac{1}{2\pi} \int_{E_j} \frac{\mathbf{n}_y \cdot (\mathbf{x}_i - \mathbf{y})}{|\mathbf{x}_i - \mathbf{y}|^2} ds_y \quad (3)$$

となる。ここに、 $E_j$  は  $j$  番目の要素を表す。

### 2.2 反復法と前処理

例として、Dirichlet 問題について考えよう。(1)において  $\mathbf{x} = \mathbf{s}$ ,  $\mathbf{b} = \mathbf{B}\mathbf{u}$  とおけば、線形方程式

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (4)$$

が得られる。高速多重極境界要素法では、係数行列  $\mathbf{A}$  を直接に計算せずに、 $\mathbf{A}\mathbf{x}$  ( $\mathbf{B}\mathbf{u}$  も同様) を高速多重極法で計算して (4) を解くので、反復法を用いるよりほ

かでない。さらに、一般には境界要素法の方程式の係数行列は非対称となるので、GMRES や Bi-CGSTAB, GPBi-CG などの非対称行列に適用可能な反復法を用いなければならない<sup>3)</sup>。

さて、一般に、線形方程式を反復法で解く場合に、係数行列に適当な処理を加えて反復の収束性を高めることが行なわれており、これを前処理と呼ぶ。方程式 (4) の場合であれば、係数行列  $\mathbf{A}$  に左右から行列  $\mathbf{M}_1^{-1}$ ,  $\mathbf{M}_2^{-1}$  をかけて

$$\tilde{\mathbf{A}} = \mathbf{M}_1^{-1} \mathbf{A} \mathbf{M}_2^{-1}, \quad \tilde{\mathbf{x}} = \mathbf{M}_2 \mathbf{x}, \quad \tilde{\mathbf{b}} = \mathbf{M}_1^{-1} \mathbf{b} \quad (5)$$

として、(4) を係数行列  $\tilde{\mathbf{A}}$  を持つ方程式

$$\tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad (6)$$

に変換する。行列  $\tilde{\mathbf{A}}$  の性質が良ければ、方程式 (6) は (4) よりも早く収束する。

前処理を実行するのに、行列  $\tilde{\mathbf{A}}$  を直接に作る必要はない。たとえば、前処理付きの Bi-CGSTAB<sup>4)</sup> のアルゴリズムは

- 初期値  $\mathbf{x}_0$  を与える
- $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$ ,  $\mathbf{p}_0 = \mathbf{r}_0$ ,  $c_1 = (\mathbf{r}_0, \mathbf{r}_0)$
- $k = 0, \dots$  について、 $\|\mathbf{r}_k\| \leq \epsilon \|\mathbf{b}\|$  となるまで以下を繰り返す。
  - $\mathbf{q} = \mathbf{A} \mathbf{M}^{-1} \mathbf{p}_k$ ,  $c_2 = (\mathbf{r}_0, \mathbf{q})$ ,  $\alpha_k = c_1 / c_2$
  - $\mathbf{e} = \mathbf{r}_k - \alpha_k \mathbf{q}$ ,  $\mathbf{v} = \mathbf{A} \mathbf{M}^{-1} \mathbf{e}$
  - $c_3 = (\mathbf{e}, \mathbf{v}) / (\mathbf{v}, \mathbf{v})$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{M}^{-1} \mathbf{p}_k + c_3 \mathbf{M}^{-1} \mathbf{e}$
  - $\mathbf{r}_{k+1} = \mathbf{e} - c_3 \mathbf{v}$
  - $c_1 = (\mathbf{r}_0, \mathbf{r}_{k+1})$ ,  $\beta_k = c_1 / (c_2 c_3)$
  - $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k (\mathbf{p}_k - c_3 \mathbf{q})$

となる。ここに、 $\mathbf{M} = \mathbf{M}_1 \mathbf{M}_2$  である。すなわち、前処理行列  $\mathbf{M}^{-1}$  とベクトルとの積の計算 (方程式  $\mathbf{M} \mathbf{y} = \mathbf{c}$  を解く操作) を各反復ごとに 2 回づつ行なえば良い。

### 2.3 Haar wavelet 変換

Haar wavelet 変換<sup>5)</sup>は Fig.1 に示すような、基底の変換である。その一つの機能は、Fourier 解析のように、もとの近似関数を解像度の異なる近似関数群に分割することである。この性質は、高速多重極法で利用する木の構造の特性とも適合するものである。

ある区間が  $2^n$  個の区間に分割されているとする。 $\phi$  をある分割レベルの基底であるとする、 $\psi$  を操作することにより、一段階細分された基底を生成することができる。たとえば、 $j$  レベルの基底により関数  $f_j(x)$  が

$$f_j(x) = \sum_k c_k^{(j)} \phi(2^j x - k) \quad (7)$$

で表されているとすると、関数

$$g_j(x) = \sum_k d_k^{(j)} \psi(2^j x - k) \quad (8)$$

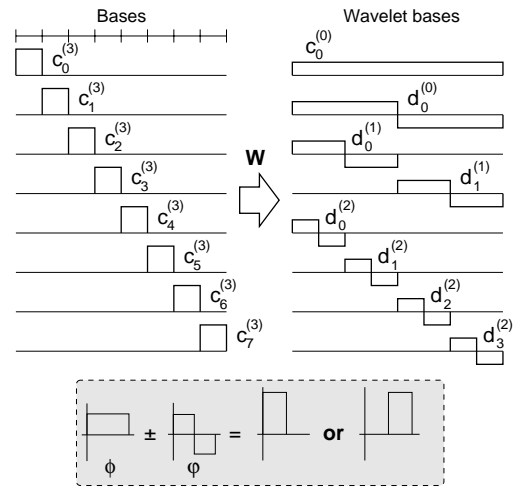


Fig.1 Haar wavelet transform

を使って、 $f_j(x)$  を一意的に分解して、

$$f_j(x) = f_{j-1}(x) + g_{j-1}(x) \quad (9)$$

と表すことができる。 $j-1$  は一つ上のレベルを表す。ここで、係数  $c_k^{(j-1)}$ ,  $d_k^{(j-1)}$  は

$$\begin{aligned} c_k^{(j-1)} &= \frac{1}{\sqrt{2}} (c_{2k}^{(j)} + c_{2k+1}^{(j)}) \\ d_k^{(j-1)} &= \frac{1}{\sqrt{2}} (c_{2k}^{(j)} - c_{2k+1}^{(j)}) \end{aligned} \quad (10)$$

によって決められる。この式は分解アルゴリズムと呼ばれる。(9) を再帰的に繰り返せば、結局、 $f_j$  は

$$f_j(x) = f_0(x) + g_0(x) + \dots + g_{j-1}(x) \quad (11)$$

と書けることになる。また、逆に  $f_{j-1}$ ,  $g_{j-1}$  が与えられていれば、(9) から  $f_j$  を決定でき、そのときの係数は

$$\begin{aligned} c_{2k}^{(j)} &= \frac{1}{\sqrt{2}} (c_k^{(j-1)} + d_k^{(j-1)}) \\ c_{2k+1}^{(j)} &= \frac{1}{\sqrt{2}} (c_k^{(j-1)} - d_k^{(j-1)}) \end{aligned} \quad (12)$$

により決まる。この式は再構成アルゴリズムと呼ばれる。式 (10), (12) において、係数  $1/\sqrt{2}$  は変換を正規化するためのものである。

さて、(10) を (7) の係数から (11) の係数への変換操作とみなせば、この操作を使ってベクトル間の変換

$$\mathbf{d} = \mathbf{W} \mathbf{c} \quad (13)$$

を定義することができる。行列  $\mathbf{W}$  が直交行列であることは容易に証明することができる。以下では、この変換を使って行列の前処理をすることを考える。

### 2.4 Haar wavelet 変換による前処理

変換  $\mathbf{W}$  を係数行列  $\mathbf{A}$  に作用させた方程式

$$(\mathbf{W} \mathbf{A} \mathbf{W}^T)(\mathbf{W} \mathbf{x}) = \mathbf{W} \mathbf{b} \quad (14)$$

について考えよう。正方形領域における係数行列  $\mathbf{WAW}^T$  の例を Fig.2 に示す。要素数は 256 である。上図は、係数行列  $\mathbf{A}$  の成分を絶対値のレベルで濃淡表示したもの、下図は変換後の行列のそれを示している。変換行列においては、行列成分の情報が階層化され、対角近傍成分および左上の方に行列全体の情報が集まっている。

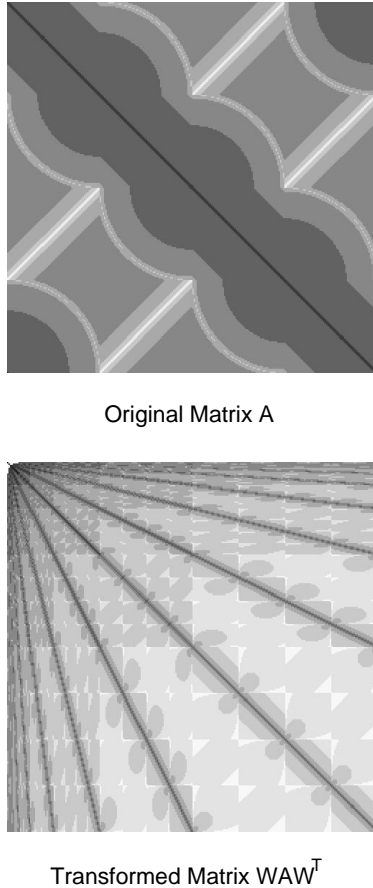


Fig.2 Coefficient matrix  $\mathbf{A}$  and its transformed matrix  $\mathbf{WAW}^T$ : square region

さて、行列  $\mathbf{A}$  を前処理するのに (14) を直接使うことはできない。なぜならば、 $\mathbf{M}^{-1} = \mathbf{W}\mathbf{W}^T = \mathbf{I}$  であるから、もとの方程式に  $\mathbf{I}$  を作用させたのと同じになってしまうからである。しかし、変換後の行列  $\mathbf{WAW}^T$  は対角付近に情報が集約されているものと考えられるので、これを使って効率の良い前処理行列を生成することは可能であろう。ここでは、最も簡単な例として、(14) に Jacobi 法を適用してみよう。 $\mathbf{WAW}^T$  の対角成分から作った行列を  $\mathbf{S}$  とし、

$$\mathbf{M}_1^{-1} = \mathbf{S}^{-1}\mathbf{W}, \quad \mathbf{M}_2^{-1} = \mathbf{W}^T \quad (15)$$

とすると、

$$\mathbf{M}^{-1} = \mathbf{W}^T\mathbf{S}^{-1}\mathbf{W} \quad (16)$$

となる。要素数が比較的小さいときのこの前処理法の効果については文献<sup>1),2)</sup>で示されたとおりである。

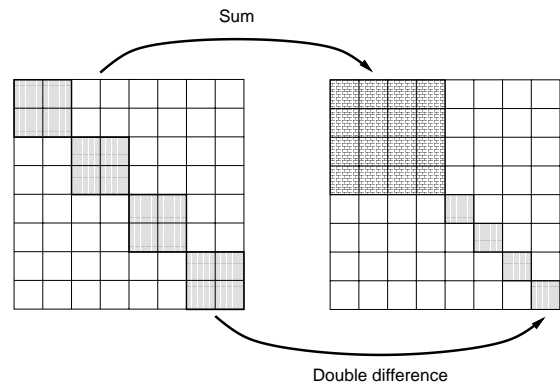


Fig.3 Calculation of the diagonal components of the wavelet transformed matrix

変換行列の対角成分の計算 上で述べた前処理を実行するためには  $\mathbf{WAW}^T$  の対角成分を求める必要がある。Fig.1 を参照すれば、これを、分解アルゴリズム (10) を使って、Fig.3 のように計算することができる。まず、行列全体を  $2 \times 2$  の小行列に区切り、一段階の変換によって得られる左上  $1/4$  部分行列の成分  $c_{kl}^{(j-1)}$  と右下  $1/4$  部分行列の対角成分  $d_{kk}^{(j-1)}$  とを

$$\begin{aligned} c_{kl}^{(j-1)} &= \frac{1}{2} \left( c_{2k,2l}^{(j)} + c_{2k,2l+1}^{(j)} + c_{2k+1,2l}^{(j)} + c_{2k+1,2l+1}^{(j)} \right) \\ d_{kk}^{(j-1)} &= \frac{1}{2} \left( c_{2k,2k}^{(j)} - c_{2k,2k+1}^{(j)} \right. \\ &\quad \left. - c_{2k+1,2k}^{(j)} + c_{2k+1,2k+1}^{(j)} \right) \end{aligned} \quad (17)$$

により計算する。すなわち、 $c_{kl}^{(j-1)}$  は対応する  $2 \times 2$  小行列の成分の総和となり、 $d_{kk}^{(j-1)}$  は対角部分の小行列の行と列に (10)<sub>2</sub> を作用させたもの (2 重差分) である。この操作を左上の  $1/4$  の行列について再帰的に繰り返せばすべての対角成分が得られる。

### 3. 高速多重極境界要素法への応用

前節で述べた前処理法を高速多重極境界要素法に適用するためには、行列の成分を計算することなく  $\mathbf{WAW}^T$  の対角成分を求める必要がある。すなわち、(17) において、 $d_{kk}^{(j-1)}$  の計算に必要な  $c_{kl}^{(j)}$  を、できるだけ行列の成分を使わずに計算することが必要である。幸いにして、Haar wavelet 変換は 2 分木との相性が良い。また、係数の部分行列の成分の総和は多重極表現を使って計算することができる。したがって、これらを組み合わせれば、必要な  $c_{kl}^{(j)}$  だけを計算して、対角成分を求めることが可能である。

#### 3.1 2分木上の Haar wavelet 変換

2 分木の上の Haar wavelet 変換を構成しよう。分解 (10) と再構成 (12) とを 2 分木の上で実行することを考える。分解は個々の成分を集めさせる操作であるから、2 分木の葉に要素 (あるいは選点) を割り当てるとすると、木のそれぞれの節に wavelet 基底が割り当てられる (Fig.4)。分解の操作は葉から根への計算により実現さ

れる。Fig.1 の場合には基底の並びはレベル順になったが、この場合には基底が並ぶ順序は木のたどり方により決まる。たとえば、Fig.4 に示すように、根から右側へ矢印の経路をたどり、帰りがけ順に計算を行えば、節ごとに Haar 係数が作られる。このとき、係数の並び順はレベルの異なるものが混在することになり、後ろから並べると、 $\{c_0^{(0)}, d_0^{(0)}, d_0^{(1)}, d_0^{(2)}, d_1^{(2)}, d_1^{(1)}, d_2^{(1)}\}$  となる。逆に、再構成は、矢印を逆向きに行きがけ順にたどる根から葉への計算となる。

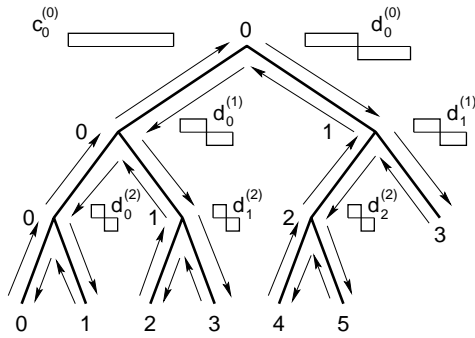


Fig.4 Haar wavelet bases and search path on a binary tree

図に示すように、この変換は要素数が  $2^n$  でなくても適用できることは明らかである。この場合でも、正規化のためのスケールリングを行なった変換 (10) を使えば、変換に対応する行列  $\mathbf{W}$  は直交行列となり、(12) は  $\mathbf{W}^T$  と等しくなる。ただし、この場合には、深さによって葉の基底 (要素あるいは選点に対するもの) の尺度が変わっていることに注意が必要である。

**2 分木変換行列の対角成分の計算** 2 分木上の wavelet 変換を使った場合でも、変換行列  $\mathbf{WAW}^T$  の対角成分は、(17) を使って Fig.3 に示したのと同様の考え方で計算することができる。ただし、この場合には、wavelet 基底の並びはレベル順ではないので、計算は Fig.3 のような左上方向への再帰計算にはならず、木の上での再帰計算となる。また、当然ではあるが、この場合も、行列成分の総和を計算するので、計算量は行列の次元  $N$  に対して  $O(N^2)$  である。

### 3.2 4 分木上の Haar wavelet 変換

4 分木の上の Haar wavelet 変換は、4 分木の 1 段が 2 段の 2 分木で表せることから容易に導かれる。4 分木の葉のセルに含まれる要素集合については 2 分木の Haar wavelet を使って変換する。4 分木上の Haar wavelet 変換では、要素数が  $2^n$  でなくても適用できることがとくに重要である。なぜならば、自然な分割から得られる葉に含まれる要素数は一般には  $2^n$  にはならないし、境界を覆う 4 分木では 4 つの子セルがそろそろこともまれであるからである。同様の考え方により、3 次元問題の場合には 8 分木の上の Haar wavelet 変換を利用することができる。Fig.5 に 4 分木の上の Haar wavelet 変換により生成される行列  $\mathbf{WAW}^T$  を示す。

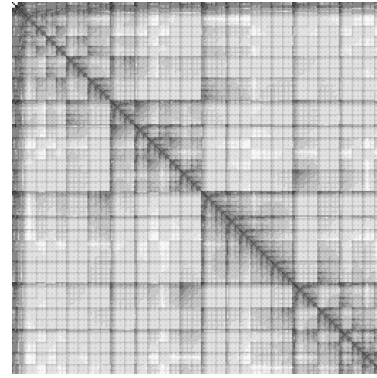


Fig.5 Wavelet transformed matrix on quad tree

### 3.3 4 分木上の $\mathbf{WAW}^T$ の対角成分

#### (1) 非正規変換の導入

4 分木の上で  $\mathbf{WAW}^T$  の対角成分を計算するにあたって、非正規な分解アルゴリズム

$$c_k^{(j-1)} = c_{2k}^{(j)} + c_{2k+1}^{(j)}, \quad d_k^{(j-1)} = c_{2k}^{(j)} - c_{2k+1}^{(j)} \quad (18)$$

を導入しよう。対応する再構成アルゴリズムは

$$c_{2k}^{(j)} = \frac{1}{2} (c_k^{(j-1)} + d_k^{(j-1)})$$

$$c_{2k+1}^{(j)} = \frac{1}{2} (c_k^{(j-1)} - d_k^{(j-1)}) \quad (19)$$

である。変換 (18) を  $\tilde{\mathbf{W}}$  で表すことにする。当然ながら、再構成は  $\tilde{\mathbf{W}}^{-1} \neq \tilde{\mathbf{W}}^T$  である。

#### (2) $\tilde{\mathbf{W}}\mathbf{A}\tilde{\mathbf{W}}^T$ の対角成分の計算

変換  $\tilde{\mathbf{W}}$  の場合には、(17) の係数  $1/2$  がとれて、 $c_{kl}^{(j-1)}$  は小行列の成分の総和そのものとなり、対角成分  $d_{kk}^{(j-1)}$  は対角部分の小行列から作られる 2 重差分そのものである。したがって、 $\tilde{\mathbf{W}}\mathbf{A}\tilde{\mathbf{W}}^T$  の対角成分は以下のように木の上で計算することができる。

4 分木上においては、(17) に現れる  $2 \times 2$  小行列の成分  $c_{kl}^{(j)}$  は、一方のセル  $l$  に含まれる全要素から他方のセル  $k$  に含まれる全選点への影響の総和である (これを相互作用係数と呼ぶことにする)。(17)<sub>1</sub> は、隣接するセルを統合して大きなセル同士の相互作用係数を求める演算であり、繰り返せば、ついには、全要素から全選点への影響の総和に至る。一方、(17)<sub>2</sub> では、隣接するセル同士の相互作用係数だけが使われており、離れた位置にあるセル同士の相互作用係数は不要である。そこで、各セルについて、それに隣接するセルからの相互作用係数  $\mathbf{C} = \{c_0, c_1, \dots, c_M\}$  だけを計算し、それを使って親セルの相互作用係数を計算していけば、つねに (17)<sub>2</sub> の計算が可能となる。ただし、この計算の過程において、子セルにとっては遠方にあるセルとの相互作用係数を求める必要があるため、これを多重極表現を使って計算することにより、(17)<sub>1</sub> の計算を大幅に省略しようというのがここでの基本方針である。

まず、遠方演算の方法を示そう。一般に、基本解の多重極表現と局所展開項との間には相似の関係がある。2

次元ポテンシャル問題では、基本解の多重極モーメントと局所展開とは

$$M_0 = 1, M_n = z^n/n \quad \text{基本解の多重極係数 (20)}$$

$$u(\mathbf{x}) = \sum_n L_n z^n \quad \text{局所展開} \quad (21)$$

である<sup>6)</sup>。変換  $\tilde{\mathbf{W}}$  の場合には、セル間の相互作用係数は、すべて 1 の重みをもつある要素グループからある選点グループへの影響の総和であるから、要素グループを  $Y$ 、選点グループを  $X$  とすると、

$$\sum_{i \in X} u(\mathbf{x}_i) = \sum_n L_n(Y) \left[ \sum_{i \in X} z_i^n \right] \quad (22)$$

を計算すればよい。ここに、 $L_n(Y)$  は重み 1 の要素グループ  $Y$  により生成される場の局所展開係数である。(20) によれば、カッコ内は選点においた基本解による選点グループ  $X$  を含むセルの多重極モーメント  $M_n(X)$  から求められるから、 $X$  と  $Y$  が互いに遠方であれば、上式は

$$\sum_{i \in X} u(\mathbf{x}_i) = L_0(Y)M_0(X) + \sum_{n \geq 1} n L_n(Y)M_n(X) \quad (23)$$

として計算できる。Galerkin 法を使う場合には基本解の代わりに要素の影響関数を使えば同じことができる。

親セル  $X$  の隣接セル  $Y$  からの相互作用係数は、Fig.6 を参照して、子セル  $i$  の近傍セルの集合 ( $i$  を含む) を  $\Omega_i$  とすると、 $i$  の相互作用係数と (23) から、

$$c_Y(X) = \sum_{i \in X} \left\{ \sum_{j \in Y \cap \Omega_i} c_j(i) + \sum_{k \in Y - \Omega_i} \left[ L_0(k)M_0(i) + \sum_{n \geq 1} n L_n(k)M_n(i) \right] \right\} \quad (24)$$

によって計算することができる。

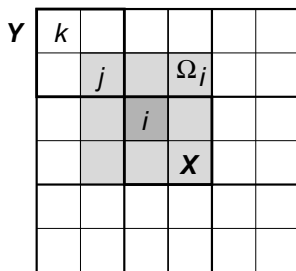


Fig.6 Calculation of interactive coefficients on quad tree

以上により、wavelet 変換行列  $\tilde{\mathbf{W}} \mathbf{A} \tilde{\mathbf{W}}^T$  の対角成分を計算するアルゴリズムは次のようになる。

- (1) 重みを 1 として、要素による多重極係数、選点においた基本解による多重極係数をすべてのセルについて求める (高速多重極法の第 1 段階と同じ)。

- (2) 葉のセルからはじめて上向計算を行なう。

- (a) 分解アルゴリズムによる対角成分の計算

- 子セルの相互作用係数を使って子セル間の相互作用行列  $\{c_{kl}^{(j)}\}$  を作り、その行列の分解計算を行なう。
- 葉セルの場合には、それに含まれる全要素についての部分行列について、2 分木上の分解計算を行なう。

この計算によりセルに含まれる部分行列の成分の総和  $c_0$  が求まる。

- (b) 隣接するセルについて相互作用係数  $C$  を求める。

- 子セルの相互作用係数を利用して、(24) により相互作用係数を計算する。
- 葉セルの場合には、直接計算によって相互作用係数を求める。

- (c) (a)–(b) を繰り返して根まで至る。

- (3)  $\mathbf{WAW}^T$  の対角成分の計算

$\mathbf{WAW}^T$  の対角成分を計算するために  $\tilde{\mathbf{W}}$  と  $\mathbf{W}$  との関係を求める。 $\mathbf{W}$  は  $\tilde{\mathbf{W}}$  に正規化のためのスケールリングを施したものであるが、葉の深さがまちまちの 2 分木上の変換なので、要素 (あるいは選点) に対応する葉の深さに依存したスケールリングも必要である。したがって、 $\mathbf{W}$  と  $\tilde{\mathbf{W}}$  との関係は

$$\mathbf{W} = \mathbf{S}_1 \tilde{\mathbf{W}} \mathbf{S}_2 \quad (25)$$

のように表すことができる。ここに、 $\mathbf{S}_1, \mathbf{S}_2$  は左右からのスケールリングのための対角行列であって、 $\mathbf{S}_1$  の成分は節の深さに依存する正規化のための係数であり、該当する wavelet 基底の 2 分木上のレベルを  $n$ 、最大のレベルを  $n_{\max}$  とするとき、 $1/\sqrt{2^{n_{\max}-n}}$  で与えられる。また、 $\mathbf{S}_2$  の成分は葉の 2 分木上のレベルの最大のレベルからの高さによる換算係数であって、対応する要素のレベルを  $n$  とするとき、 $\sqrt{2^{n_{\max}-n}}$  である。これらはいずれも  $\mathbf{W}^T$  の計算と同様の手続きによって求めることができる。

(25) によって

$$\mathbf{WAW}^T = \mathbf{S}_1 \tilde{\mathbf{W}} \mathbf{S}_2 \mathbf{A} \mathbf{S}_2 \tilde{\mathbf{W}}^T \mathbf{S}_1 \quad (26)$$

であるから、 $\mathbf{S}_2$  の成分によって与えられる重みを持った行列  $\mathbf{S}_2 \mathbf{A} \mathbf{S}_2$  について、さきの方法で  $\tilde{\mathbf{W}} \mathbf{S}_2 \mathbf{A} \mathbf{S}_2 \tilde{\mathbf{W}}^T$  の対角成分を計算し、結果を  $\mathbf{S}_1$  によってスケールリングすれば  $\mathbf{WAW}^T$  の対角成分を計算することができる。

#### 4. 数値解析例

数値例として、単純な形状における Dirichlet 問題を解析した場合の計算例を示す。文献<sup>1)</sup>によれば、ポテンシャル問題の場合には、反復回数は境界形状にはあまり依存しないことがわかっているため、ここでは、正方形領域および L 型領域 (正方形からその 1/4 を取り除い

た形状)を代表例として取り上げた。ここで使用した反復法は GPBi-CG<sup>7)</sup>である。

Fig.7 に、反復回数に対する相対誤差  $\epsilon = \|\mathbf{r}\|/\|\mathbf{b}\|$  ( $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ ) の減少の様子を示す。要素数は  $N = 16K$  ( $K = 1024$ ) である。前処理を行わない場合と比べて、wavelet 変換による前処理を行なった場合には誤差は極めて早く減少する。

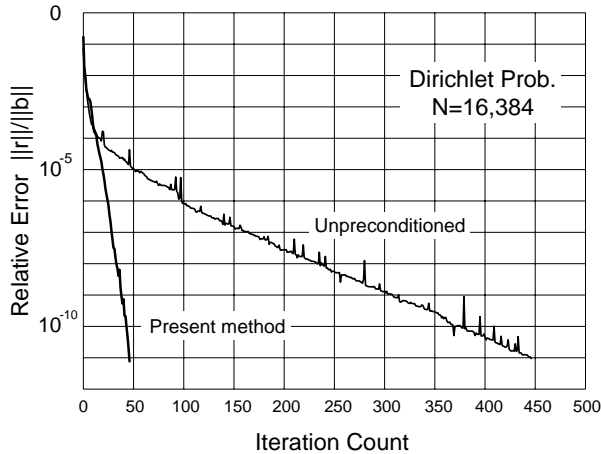


Fig.7 Relative error of iteration: square region

Fig.8 は、正方形領域における要素数と収束までの反復回数との関係を示したものである。要素数は  $N = 128 \sim 512K$  とし、収束判定の基準は  $\epsilon < 10^{-11}$  とした。いずれの場合もグラフはほぼ直線的であり、前処理のない場合の反復回数は  $\sqrt{N}$  に比例し、wavelet 変換による前処理を行なった場合には  $N^{1/4}$  に比例する程度の反復回数となっている。Fig.9 は L 型領域の場合の結果であるが、Fig.8 とほぼ同様の結果が得られている。

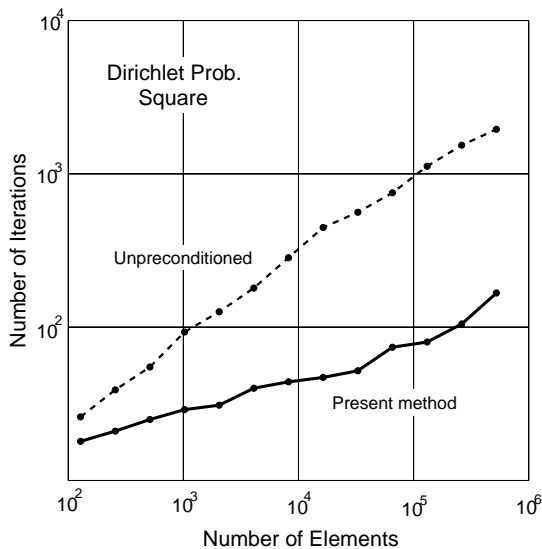


Fig.8 Number of iterations for the number of elements: relative error  $\epsilon < 10^{-11}$

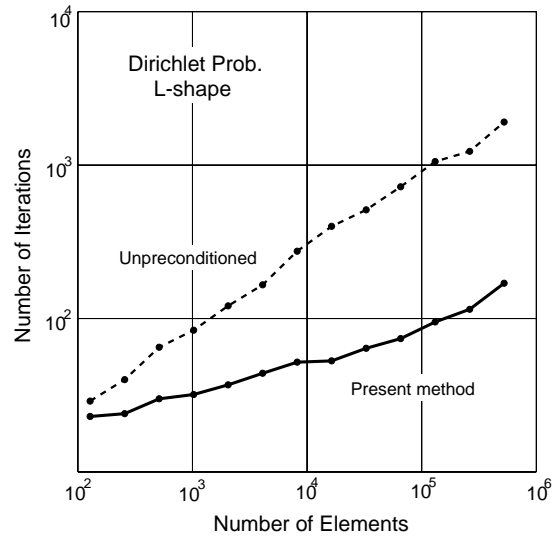


Fig.9 Number of iterations for the number of elements: relative error  $\epsilon < 10^{-11}$

## 5. おわりに

2次元ポテンシャル問題において、wavelet 変換を利用した境界要素反復解析のための前処理法を高速多重極境界要素法に適用するための計算手法を提案した。計算を4分木の上で進め、多重極展開を利用することにより、要素数  $N$  の Dirichlet 問題に対して  $N^{1/4}$  に比例する程度の反復回数で収束する前処理法が得られた。

本前処理法は他の種類の問題にも容易に適用できるものであり、現在、適用範囲の拡張および計算法の改善について研究中である。

## 参考文献

- 1) 福井卓雄: Wavelet 変換を用いた境界要素反復解法における前処理, BEM・テクノロジー・コンファレンス論文集, 9 (1999), pp. 85-90.
- 2) 福井卓雄: 大規模波動問題における境界要素反復解析の前処理法について, 計算工学講演会論文集, 6 (2001), pp. 607-610.
- 3) Barrett, R., M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, (1994), SIAM.
- 4) Van der Vorst, H.A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 13, (1992), pp. 631-644.
- 5) Chui, C.K. (桜井明, 新井勉 訳): ウェーブレット応用, 東京電機大学出版局, (1997)
- 6) 福井卓雄, 服部純一, 土居野優: 高速多重極法の境界要素解析への応用, 構造工学論文集, 43A, (1997), pp. 373-382.
- 7) Zhang, S.-L.: GPBi-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.*, 18 (1997), pp. 537-551.