《論　文》

論 15-9

境界要素解析の並列計算のための アルゴリズムの検討：解の収束性†

岩瀬英仁*・神谷紀生**・北　栄輔***

ABSTRACT Algorithms for the parallel computing of the boundary element analysis are considered in this paper. Domain decompositions for the boundary element analysis on workstations connected by LAN are compared in terms of various iteration methods, such as Uzawa and Schwarz schemes. Convergency of the schemes is discussed for the selection of parameter and material constants. Identification of the boundary conditions and the their renewal on the fictitious internal boundaries affect the convergence of the schemes. Workstation cluster computing system is constructed and applied to example problems. Difficulty in conventional boundary element subdomain method is reduced by the present scheme.

1. 緒　　言

大規模数値解析におけるコンピュータメモリの容量不足とCPUの演算速度の問題点を解決する計算手法として，領域分割法が以前から研究されてきた．この方法は解析領域を仮想的な境界（内部境界）によって複数個の部分領域に分割し，それぞれの領域を別個のCPUにより並列計算させ，隣接する領域間でデータ交換しながら，境界上の解が収束するまで反復計算を行う．この方法により各コンピュータの扱うデータ量が少なくなり，また個々の連立方程式の次元が小さくなる．その結果計算時間が短縮され，前述の問題点を解決する実用的な方法となりうる．ただし，この方法では，反復計算の回数と解の収束性が問題となり，少ない反復回数で高精度の解が得られる計算手法を開発することが望まれる．

境界値問題に対して領域分割法の考え方を最初に示したのは*Schwarz*であり，部分領域を仮想境界付近でオーバーラップさせて計算する．その後，有限要素解析において実用を目的とする研究が進められ，吉岡[1]らは境界上の解の計算方法にUzawa法[2)3)]，共役勾配法を用いて，3次元弾性問題に良好な結果を得ている．堀江ら[4]はクラスターコンピューティング上で，並列ガウス消去法，並列共役勾配法および領域分割法による性能比較を行い，領域分割法が最も有効であることを示した．境界要素解析における研究では*Hribersek*ら[5]がナビエ・ストークス問題の境界領域定式による数値解析に*Schwarz*の方法を用いている．*Avrashi*ら[6]は2次元弾性問題を，BSM (Boundary strip method) を用いた領域分割法で解いている．境界要素法に関して，この他には研究はみあたらず，基本的検討をする必要がある．

本研究では，クラスターコンピューティングによる境界要素解析について，領域分割法による並列計算法を検討し，その可能性を調べる．仮想境界上の解の収束計算方法として，Uzawa法とSchwarz法2種類の合計3つの方法を考えるが，これらはすでに有限要素解析において利用されてきたものである．ここでは，それらを境界要素解析に適用し，それぞれについて評価し，性能を比較する．

2. 領域分割境界要素法

境界要素法における領域分割法あるいは部分分割法はよく知られている．この方法は，解析対象領域が異なる支配方程式あるいは物理パラメータによって表わされる場合，および単一の支配方程式による場であっ

Algorithms for Boundary Element Parallel Computing: Convergency of Solution. By *Hidehito Iwase* (Graduate School of Nagoya Univ.), *Norio Kamiya* (School of Informatics and Sciences, Nagoya Univ.) and *Eisuke* Kita (Graduate School of Engineering, Nagoya Univ.).

*名古屋大学大学院人間情報学研究科
**名古屋大学情報文化学部
***名古屋大学大学院工学研究科

†1996年3月11日受付　1996年6月5日再受付

ても複雑な形状である場合に用いられる．前者では，各部分領域の支配方程式が異なるので，それぞれについて別個に考える必要があることは当然である．一方，後者については，積分方程式に用いられる基本解の特性によるためと，解析の精度および効率の面で必要になることがある．基本解はソース点からの距離に関連する影響関数であるから，一般に遠距離においてソース点の効果は小さくなるし，また，ソース点と影響を考える点の間に別の領域が入り込めば，その効果は直接的でなくなる．積分方程式を離散化して得られる代数方程式の係数行列はいわゆる密行列であるが，領域の部分分割を行うことにより，もとよりも小さい部分行列で全体の係数行列を構成することができる．

以上のように，境界要素法における領域分割法の長所を挙げることができるが，上記の最後の点に関して，次に述べるような短所がある．以下では次の支配方程式と境界条件で与えられる問題を考える（**Fig 1(a)**）．

$$\mu\nabla^2=0 \quad (\text{in } \Omega) \tag{1}$$

$$u=\bar{u}(\text{on } \Gamma_1), \quad q\left(\equiv\mu\frac{\partial u}{\partial n}\right)=\bar{q}(\text{on } \Gamma_2) \tag{2}$$

ここで，Ω は解析対象領域を，Γ_1, Γ_2 はポテンシャル u の指定境界，フラックス q の指定境界を示す．また，n は境界での単位外向法線ベクトル，上付きバーは指定境界条件値を示す．なお，式(1)の係数 μ は Ω における物性パラメータであり，たとえば熱伝導問題であれば，μ は熱伝導率に相当する．Laplace 方程式の基本解 u^*, q^* を用いて，式(1)を積分方程式に変換すれば，

$$cu=\int_\Gamma (u^*q-q^*u)\,d\Gamma \tag{3}$$

となる．ここで，c は各節点の位置する境界形状によって決まる定数である．上式を N 個の境界要素 Γ_j によって離散化すれば，境界節点 i について次式を得る．

$$c_iu_i=\sum_{j=1}^{N}\int_{\Gamma_j} (u^*q-q^*u)\,d\Gamma \tag{4}$$

境界要素上の節点を選点にとり，各選点ごとに成り立つ式をまとめれば次式を得る．

$$\boldsymbol{Gq}=\boldsymbol{Hu} \tag{5}$$

ここで，$\boldsymbol{u}, \boldsymbol{q}$ はすべての境界節点における u, q からなるベクトル，$\boldsymbol{H}, \boldsymbol{G}$ はそれらの係数行列であり，基本解と内挿関数の積を境界要素上で積分して計算される．単一領域の問題では，境界条件によって式(5)を配列しなおして連立方程式が得られるので，これを解いて未知境界値が決定される．

全体領域を仮想境界（異種材料の複合体では材料ごとによって分けられる境界を指すが，以下では仮想境界と表記して説明する）により m 個の部分領域（$\Omega_1, \Omega_2, \cdots, \Omega_m$）分割した場合を考える（**Fig 1(b)**）．このとき各部分領域について，積分方程式を離散化すれば，次の式が得られる．

$$\boldsymbol{G}^i\boldsymbol{q}^i=\boldsymbol{H}^i\boldsymbol{u}^i \quad (i=1, 2, \cdots, m) \tag{6}$$

ここで，$\boldsymbol{u}^i, \boldsymbol{q}^i$ は仮想境界も含め，Ω_i の境界上の節点における u, q のベクトル，$\boldsymbol{H}^i, \boldsymbol{G}^i$ はそれらの係数行列である．なお，右肩の添字は領域の番号を表している．領域 Ω_k と Ω_l の間の仮想境界 Γ_{kl} 上ではポテンシャル値，フラックス値に関して，次の適合条件，平衡条件が成り立つ．

$$\boldsymbol{u}^k=\boldsymbol{u}^l, \quad \boldsymbol{q}^k=-\boldsymbol{q}^l \tag{7}$$

ただし，

$$q^k\equiv\left(\mu\frac{\partial u}{\partial n}\right)^k \tag{8}$$

部分領域に分けるとき，式(6)を境界条件(7)，(2)とともに解くことになる．より具体的に示すと次のようになる．部分領域 Ω_i は外部領域（実境界）と仮想境界によって囲まれるので，外部境界における u, q のベクトルを改めて $\boldsymbol{u}^i, \boldsymbol{q}^i$，仮想境界 Γ_{kl} におけるそれらを $\boldsymbol{u}_{kl}, \boldsymbol{q}_{kl}$ と表わせば，式(6)は

$$\boldsymbol{G}^i\boldsymbol{q}^i+\sum \boldsymbol{G}^i_{kl}\boldsymbol{q}_{kl}=\boldsymbol{H}^i\boldsymbol{u}^i+\sum \boldsymbol{H}^i_{kl}\boldsymbol{u}_{kl} \tag{9}$$

となる．ここで $\boldsymbol{G}^i, \boldsymbol{H}^i$ は $\boldsymbol{q}^i, \boldsymbol{u}^i$ に掛かる係数行列であり，式(6)のものとは異なる．また，$\boldsymbol{G}^i_{kl}, \boldsymbol{H}^i_{kl}$ は $\boldsymbol{q}_{kl}, \boldsymbol{u}_{kl}$ の係数行列である．なお，式(9)のシグマは仮想境界に関する和（Ω_i に関連する仮想境界の個数だけ

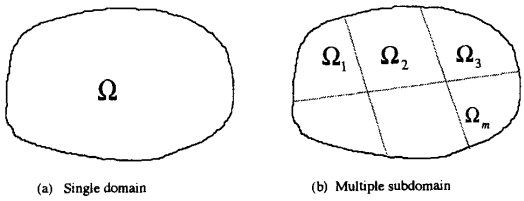


Fig. 1 Domain under consideration.

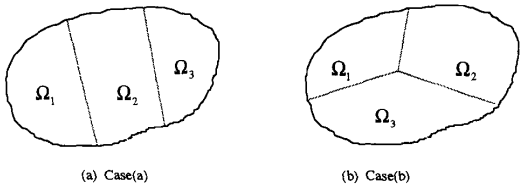


Fig. 2 Three-subdomains decomposition.

和をとる）を表すものとする．

Fig 2(a), (b)に示すように3つの部分領域に分けたとき，式(9)を仮想境界の境界条件により結合するとそれぞれつぎの式が得られる．

$$\begin{bmatrix} \boldsymbol{G}^1 & \boldsymbol{G}^1_{12} & -\boldsymbol{H}^1_{12} & 0 & 0 & 0 & 0 \\ 0 & -\boldsymbol{G}^2_{12} & -\boldsymbol{H}^2_{12} & \boldsymbol{G}^2 & \boldsymbol{G}^2_{23} & -\boldsymbol{H}^2_{23} & 0 \\ 0 & 0 & 0 & 0 & -\boldsymbol{G}^3_{23} & -\boldsymbol{H}^3_{23} & \boldsymbol{G}^3 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^1 \\ \boldsymbol{q}_{12} \\ \boldsymbol{u}_{12} \\ \boldsymbol{q}^2 \\ \boldsymbol{q}_{23} \\ \boldsymbol{u}_{23} \\ \boldsymbol{q}^3 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{H}^1 & 0 & 0 \\ 0 & \boldsymbol{H}^2 & 0 \\ 0 & 0 & \boldsymbol{H}^3 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^2 \\ \boldsymbol{u}^3 \end{bmatrix} \qquad (10)_1$$

$$\begin{bmatrix} \boldsymbol{G}^1 & \boldsymbol{G}^1_{12} & -\boldsymbol{H}^1_{12} & 0 & 0 & 0 \\ 0 & -\boldsymbol{G}^2_{12} & -\boldsymbol{H}^2_{12} & \boldsymbol{G}^2 & \boldsymbol{G}^2_{23} & -\boldsymbol{H}^2_{23} \\ 0 & 0 & 0 & 0 & -\boldsymbol{G}^3_{23} & -\boldsymbol{H}^3_{23} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{q}^1 \\ \boldsymbol{q}_{12} \\ \boldsymbol{u}_{12} \\ \boldsymbol{q}^2 \\ \boldsymbol{q}_{23} \\ \boldsymbol{u}_{23} \\ \boldsymbol{q}^3 \\ \boldsymbol{q}_{13} \\ \boldsymbol{u}_{13} \end{bmatrix}$$

$$\begin{matrix} 0 & \boldsymbol{G}^1_{13} & -\boldsymbol{H}^1_{13} \\ 0 & 0 & 0 \\ \boldsymbol{G}^3 & \boldsymbol{G}^3_{13} & -\boldsymbol{H}^3_{13} \end{matrix}$$

$$= \begin{bmatrix} \boldsymbol{H}^1 & 0 & 0 \\ 0 & \boldsymbol{H}^2 & 0 \\ 0 & 0 & \boldsymbol{H}^3 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^2 \\ \boldsymbol{u}^3 \end{bmatrix} \qquad (10)_2$$

式$(10)_1$と$(10)_2$を比べればわかるように，部分領域数は同じであるにもかかわらず全体領域を分割する方法が変われば，未知数ベクトル，係数行列のならびが異なる．すなわち，これは分割の仕方に対応して，式を構成するためのアルゴリズムを変えなければならないことになる．さらに，実境界の境界条件に対応して，未知量と既知量の配列が変わることにも注意が必要である．

このような状況は，ソフトウェアのブラックボックス化にとって大きい障害となることは明らかである．このための対処法として，仮想境界上の点は，領域Ω_kに属する点と領域Ω_lに属する別々の点で独立であるとして，式(6)をそれぞれの領域についてならべ，そのあとに式(7)をつけ加えて全体の方程式を構成する方法がある．しかしながら，このときは未知数の数がきわめて多くなるとともに，行列成分の係数が特殊な形になることは明らかである．

3. 並列化を前提とした領域分割法

以上述べたように，仮想境界上の条件（式(7)）を用いて全体領域の連立方程式を解く方法には不都合がある．そこで，この問題を解決するために，有限要素法ですでに用いられている領域分割による解法を境界要素法に適用する．この場合，各部分領域ごとの境界要素解析を独立に行うので，先に述べた問題点を解決できる期待をもつ計算方法である．

各部分領域の積分方程式を離散化して得られる式(6)を各部分領域ごとに独立して計算するには，仮想境界の境界条件を与えなければならない．仮想境界の境界条件は元来ポテンシャル，フラックスとも未知量なので，あらかじめポテンシャルかフラックスのいずれかを境界条件として指定し，その仮定のもとで計算を実行する．ただし，ある部分領域について，実境界をも含めて，その全境界がフラックス指定になることは避けなければならない．計算結果を基に，仮定した境界条件を適合条件，平衡条件を考慮して更新し，最終的に収束値が得られるまで反復計算を行う．

仮想境界上の境界条件の更新方法は，Uzawa法（以下方法(1)とよぶ），*Schwarz*のノイマン—ノイマン法[7]（方法(2)），*Schwarz*のディリクレ—ノイマン法[7]（方法(3)）を採用する．*Schwarz*の2つの方法はよく知られたものであり，ここでは部分領域の重なりを用いない，非オーバラップ法による（境界要素法では原則としてオーバラップ部分は考えられない）．**Fig 3**でΩ_k, Ω_lを自領域，隣接領域として説明する．

方法(1)では，隣接領域，自領域とも仮想境界上での境界条件はポテンシャル値（ディリクレデータ）を与える．つまり，

$$u^k=\bar{u}^k, \quad u^l=\bar{u}^l \qquad (11)$$

そして，フラックス値を求める．更新方法は，ポテンシャル値を前回計算分の隣接領域のフラックス値q^l_t

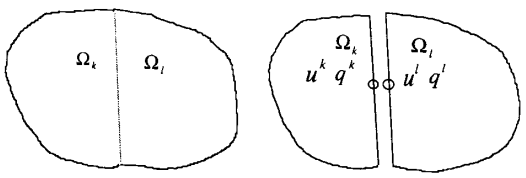


Fig. 3 Domain under consideration Ω_k and its adjacent domain Ω_l.

と自領域のフラックス値 q_s^k の差で修正するようにしたものであり，仮想境界上のフラックス値が収束するとともに，ポテンシャル値も収束する．なお，このように求めたポテンシャル値はその値を隣接領域境界でのポテンシャル値とする．

$$u_{s+1}^k = u_s^k - \alpha(q_s^l + q_s^k) \quad (12)$$

$$u_{s+1}^l = u_{s+1}^k \quad (13)$$

この更新式における下添字 s はイタレーションの回数を表す．

方法(2)では，隣接領域，自領域とも仮想境界条件はフラックス値（ノイマンデータ）とする．つまり，

$$q^k = \bar{q}^k, \quad q^l = \bar{q}^l \quad (14)$$

としてポテンシャル値を求める．更新方法はフラックス値を，前回計算分の隣接領域のポテンシャル値 q_s^l と自領域のポテンシャル値 q_s^k の差で修正するようにしたものであり，仮想境界上のポテンシャル値が収束するとともに，フラックス値も収束する．なお，隣接領域境界のフラックス値は適合条件により，更新される．

$$q_{s+1}^k = q_s^k + \beta(u_s^l - u_s^k) \quad (15)$$

$$q_{s+1}^l = -q_{s+1}^k \quad (16)$$

方法(3)では，自領域の仮想境界条件はポテンシャル値，隣接領域の仮想境界条件はフラックス値とする．つまり，

$$u^k = \bar{u}^k, \quad q^l = \bar{q}^l \quad (17)$$

そして，それぞれの未知量が決定される．更新方法は，ポテンシャル値を前回計算分の隣接領域のポテンシャル値 u_s^l と自領域のポテンシャル値 u_s^k の差で修正するようにしたものであり，ポテンシャル値が収束すればフラックス値も収束する．

$$u_{s+1}^k = u_s^k + \gamma(u_s^l - u_s^k) \quad (18)$$

$$q_{s+1}^l = -q_s^k \quad (19)$$

なお，*Schwarz* の方法(2)，(3)では部分領域の仮想境界上でフラックス値指定(ノイマンデータ)とするが，これによってすべての境界上でノイマン条件となることがあるので，これは避ける必要がある．なお，式(12)，(15)，(18)に現れる α, β, γ は定数である．

4. クラスターコンピューティングによる並列計算

領域分割法により，各部分領域ごとにその領域に関する境界要素解析が独立に実行されるので，部分領域ごとの並列計算が可能となる．本研究では，領域分割法を並列計算により実行するのに，複数のワークステーションによるクラスターコンピューティング環境を用い，計算効率の改善を図る．以下にクラスターコンピューティングのシステムと並列アルゴリズムについて説明する．

クラスターコンピューティングは，ネットワークで接続された複数台のワークステーションを仮想的に1台のコンピュータとみなし，各ワークステーションにタスクを分散させて，並列計算させることで処理能力を上げる方法である．ここで用いたワークステーションは Fujitsu S-4/EC であり，**Fig 4** に示すように $m+1$ 台のワークステーションから構成されており，それぞれはイーサネットケーブルで接続されている．また，同時にこれらは他のワークステーションとも結合されているので，各ワークステーションの演算能力は同じであるが，常時不特定多数ユーザが利用しているので演算能力は時々刻々変化する可能性がある．$m+1$ 台の内の1台がホストマシンで，他 m 台がサブマシンである．ホストが計算プロセスの制御を担当し，各サブマシンは担当する部分領域の境界要素解析を実行する．データの送受信は各ワークステーションにインストールされている PVM を利用して実現する．

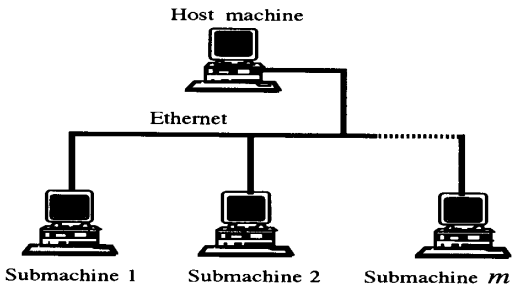


Fig. 4 Cluster computing system for parallel BEM analysis.

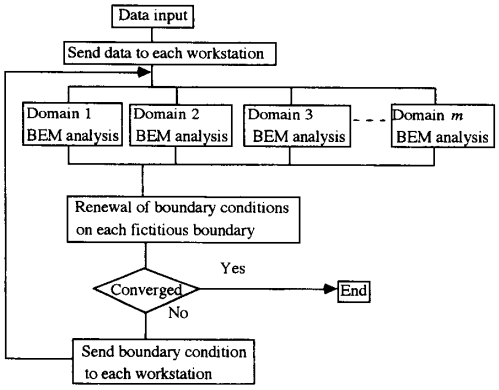


Fig. 5 Algorithm of parallel BEM by domain decomposition cluster computing.

本研究での領域分割法による並列境界要素解析のアルゴリズムを **Fig 5** に示す．ホストのワークステーションが，各部分領域の解析を担当するワークステーションに境界形状，境界条件データを送信し，そのデータを用いて各ワークステーションは境界要素解析を実行する．求めたデータをホストに送信し，ホストは仮想境界上で求められた未知量が収束したかどうかチェックする．収束していなければ各仮想境界上の境界条件としての既知量を更新し，各ワークステーションに送信し，再度計算を実行する．仮想境界上の未知量が収束するまで反復計算が実行される．なお，各サブマシンでの境界要素解析は通常の解析法によって，しかも全体領域に対するものよりも，はるかに小規模で行われる．

5. 仮想境界上の境界条件の更新方法の比較検討

前述の3つの方法の定数 α, β, γ はユーザがあらかじめ指定しなければならない．これらの定数の設定によって解の収束の様子がかなり違い，収束，発散，振動の場合がある．これらを適切にきめるための確立した方法はないと思われるので，ユーザは試行錯誤により，解が収束するように適切な設定値を決めなければならない．これらの定数の取り得る範囲は，仮想境界上の未知量の初期値の取り方，材料定数，仮想境界上の境界要素数などによって影響されると考えられるので，本研究では3つの方法におけるそれぞれの定数の範囲と解の収束の関連を調べた（なお，反復計算の初期値の取り方は実際面では重要な点であるが，本研究では，これを以下の例題を含めすべて0にとり，この影響について今回は取り上げないこととした）．

用いた例題は，**Fig 6** に示すように材料の異なる2領域が接続されていて，境界条件が図に示すように与えられている場合のラプラス方程式の2次元問題である．境界要素分割は仮想境界（ここでは正確には内部境界であり，仮想境界でない）以外は正方形の1辺は6要素に分割した．境界要素は要素内の関数値を一定とする一定要素を用いた．仮想境界上の要素は1, 2, 3, 6, 12のように分割数を変化させて収束の様子を調べた．なお，Ω_1 の材料定数は μ^1，Ω_2 の材料定数は μ^2 とし，それらの比を次のように定義する．

$$\lambda=\frac{\mu^1}{\mu^2} \tag{20}$$

5.1 解が収束する定数の範囲

仮想境界上の境界条件とそれらの更新方法について先に述べた3つの方法それぞれにおいて，仮想境界上の境界要素数 t を 1, 2, 3, 6, 12材料比 λ を 0.5, 1, 2 と変化させた場合で，解が発散しない範囲を調べた．その結果を **Table 1** に示す．方法(1)では，t と λ を変化させた場合，解の収束する定数 α の範囲は変化した．方法(2), (3)では，λ を変えるごとに解の収束する定数 β, γ の範囲が変化したが，t の変化に対しては影響を受けないことがわかった．

5.2 解の収束性

3つの方法それぞれにおいて，$\lambda=0.5$, $t=3$, 6, 12の場合で，各定数を $\alpha=1.0$, 0.5, 0.25, $\beta=0.2$, $\gamma=0.3$ とした時の解の収束（ここでは誤差が 10^{-3} 以下になった場合をいう）の様子を **Fig 7～9** に示す．図の縦軸は，内部境界上の中央の要素におけるポテンシャル値の計算値と厳密解との差を表す誤差であり，横軸は反復回数である．

3つの方法を比較すればつぎのことがわかる．方法(1)は収束するまでのイタレーション回数が最も多い．方法(1), (2)では，仮想境界上の境界要素数が増加すると，イタレーション回数は増加する傾向にある．方法(2)では，解は振動しながら収束していく傾向がみ

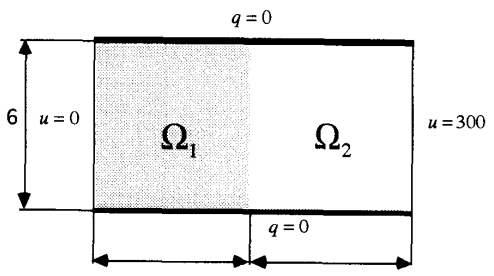


Fig. 6 Example of domain composed of two subdomains.

Table 1 Range of parameters for changes of number of elements and material ratio.

Number of elements t	Material ratio λ	Method (1) α	Method (2) β	Method (3) γ
1	0.5	0～8.0	0～0.2	0～1.8
	1	0～4.5	0～0.15	0～0.9
	2	0～3.5	0～0.1	0～0.4
2	0.5	0～2.5	0～0.2	0～1.8
	1	0～1.8	0～0.15	0～0.9
	2	0～1.3	0～0.1	0～0.4
3	0.5	0～1.2	0～0.2	0～1.8
	1	0～0.8	0～0.15	0～0.9
	2	0～0.5	0～0.1	0～0.4
6	0.5	0～0.5	0～0.2	0～1.8
	1	0～0.4	0～0.15	0～0.9
	2	0～0.2	0～0.1	0～0.4
12	0.5	0～0.25	0～0.2	0～1.8
	1	0～0.18	0～0.15	0～0.9
	2	0～0.13	0～0.1	0～0.4

られる．方法(3)は最もはやく解が収束し，解はほとんど振動しない．なお Fig 7～9 において，誤差がいったん急激に減少したあとで，一定の値に近づく現象がみられるが，これは誤差を評価した以外の点での解の収束が反復計算の時点で十分でないために，それらの影響が出たものと思われる．

以上のことから，いずれの方法でもは解は収束するが，この例題では，方法(3)がもっとも適切であると思われる．ただし，収束した解の精度は t に依存する．なお，すでに述べたように，実用上の問題点として，方法(2)，(3)では状況によってはノイマン条件だけの部分領域が設定されることになるので，そのような恐れがあるときは，ある仮想境界では方法(1)によりディリクレ境界条件の指定も含まれるようにする．

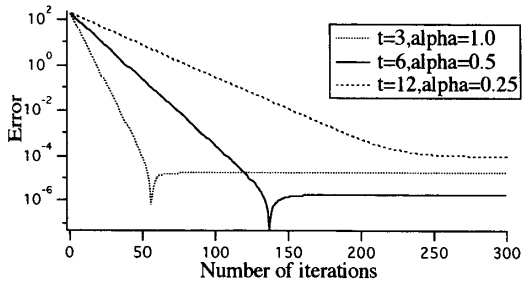


Fig. 7 Convergence of solution in Method (1).

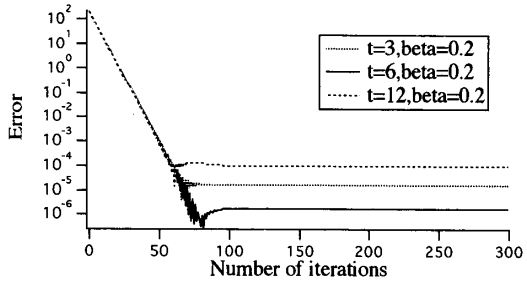


Fig. 8 Convergence of solution in Method (2).

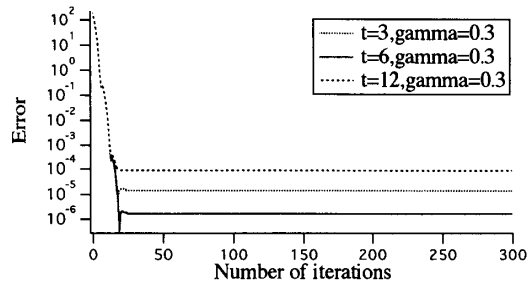


Fig. 9 Convergence of solution in Method (3).

そうすれば，ノイマン境界条件だけの部分領域を避けることができる．

そこでつぎに，方法(1)と(2)，(1)と(3)を混合したとき，反復計算によって収束解が得られるかどうかをつぎの例で検討する．**Fig 10**に示すような一様材質の長方形領域を 3 つの部分領域に分割する．実境界の境界条件は上下辺でフラックス指定であるので，仮想境界 Γ_{12}, Γ_{23}上でフラックス指定とすると，部分領域 Ω_2 のすべての境界がノイマン条件となってしまう．そこで，(a)仮想境界 Γ_{12}を方法(2)(β=0.08) で，仮想境界 Γ_{23}を方法(1)(α=0.3) で境界値の更新を行う場合，(b)仮想境界 Γ_{12}を方法(3)(γ=0.4) で，仮想境界 Γ_{23}を方法 (1)(α=0.4) で境界値の更新をを行う場合を考える．各部分領域の要素数は 1 辺につき 6 要素の合計24要素（一定要素）を用いる．反復計算の結果は，仮想境界上の中央の要素におけるポテンシャル値を

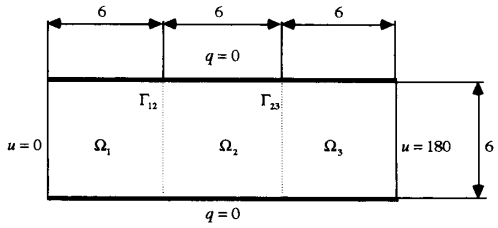
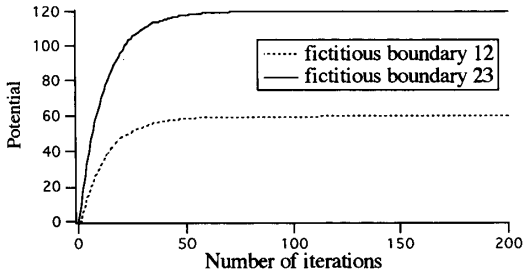
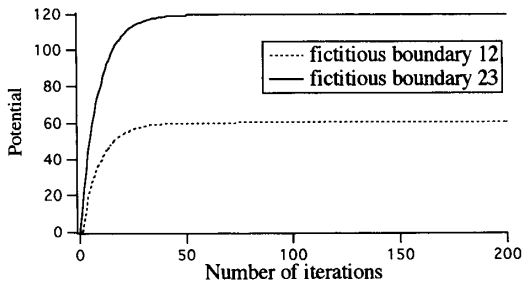


Fig. 10 Three-decomposed domain.



(a) Combination of Methods (2) and (1).



(b) Combination of Methods (3) and (1).

Fig. 11 Convergence of solution in combined methods.

Fig 11に示す．図の表示にfictitious boundary 12はΓ_{12}を，fictitious boundary 23はΓ_{23}を表す．この結果のように良い収束性がみられ，部分領域ごとに仮想境界での更新方法を変えても妥当な結果が得られることがわかった．

6. 結　　言

領域分割法による境界要素解析が，部分領域単位で並列計算可能となるアルゴリズムであることを示し，反復計算の収束性を部分領域の仮想境界での更新方法とともに検討した．これにより従来用いられていた領域分割法における，方程式の構成およびプログラミングの困難さが緩和され，汎用ソフトウェアの開発に役立つものと思われる．さらに，この領域分割法に対応するワークステーション・クラスター並列コンピューティングシステムを構築した．

なお，ここで用いた仮想境界における未知量の更新には，方法ごとに別々の定数が用いられた．これらの値はさし当たり，試行錯誤により定めたが，より一般的な決定法を確立する必要がある．さらに構築したクラスターコンピューティング並列計算システムの効率については別の機会に検討することにする．

参 考 文 献

1) 吉岡　顕，矢川元基，吉村　忍，曽根田直樹，大規模・超高速計算力学のためのネットワーク・コンピューティング手法の開発，日本機械学会論文集（A），**57**，1991-1999 (1991).

2) Glowinski, R., Dinh, Q. V., and Periaux, J., Domain Decomposition Methods for Nonlinear Problems in Fluid Dynamics, *Comput. Meth. Appl. Mech. Engng.*, **40**, 27-109, 1983.

3) 矢川元基，曽根田直樹，パラレルコンピューティング，培風館 (1991).

4) 堀江知義，倉前宏行，ネットワーク上のワークステーションによる並列有限要素解析および性能評価，日本機械学会論文集（A），**61**, 861-868 (1994).

5) Hribersek, M., and Skerget, L., Domain Decompostion Methods for Vorticity Transport Equation in Boundary Domain Integral Method, *Proc. Boundary Element Technology* X, eds. Aliabadi, M. H., Brebbia, C. A., Dular, P., and Nicolet, A., pp. 161-168, Comp. Mech. Pub., (1995).

6) Avrashi, J., Michael, O., and Rosenhouse, G., Domain Decomposition Technique Using the Boundary Spectral Strip Method, *Proc. Boundary Element Technology* X, eds. Aliabadi, M. H., Brebbia, C. A., Dular, P., and Nicolet, A., pp. 203-210, Comp. Mech. Pub., (1995).

7) Lai, C. H., Diakoptics, Domain Decomposition and Parallel Computing, *Comput. Jour.*, **37**, 840-846, (1994).