

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4778558号
(P4778558)

(45) 発行日 平成23年9月21日 (2011.9.21)

(24) 登録日 平成23年7月8日 (2011.7.8)

(51) Int. Cl.

F I

G 0 6 F 17/12 (2006.01)

G 0 6 F 17/12

G 0 6 F 17/16 (2006.01)

G 0 6 F 17/16

P

請求項の数 3 (全 24 頁)

(21) 出願番号 特願2008-531918 (P2008-531918)
 (86) (22) 出願日 平成18年8月30日 (2006.8.30)
 (86) 国際出願番号 PCT/JP2006/317065
 (87) 国際公開番号 W02008/026261
 (87) 国際公開日 平成20年3月6日 (2008.3.6)
 審査請求日 平成20年9月9日 (2008.9.9)

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100074099
 弁理士 大菅 義之
 (74) 代理人 100133570
 弁理士 ▲徳▼永 民雄
 (72) 発明者 清水 香苞
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内
 審査官 漆原 孝治

最終頁に続く

(54) 【発明の名称】 有限要素法と境界要素法による結合方程式の高速演算処理方法

(57) 【特許請求の範囲】

【請求項1】

HDDヘッドとHDD媒体との間の動作状態を数値解析するための解析対象を格子状にモデル化し、境界要素法によって生成された連立方程式と、有限要素法によって生成された連立方程式の係数を、同一の行列内に配置し、高速にマルチCPU計算機で処理するための方法であって、

該マルチCPU計算機は、

有限要素法によって生成された連立方程式の係数を、該行列の列番号が小さい側と大きい側に配置して、該マルチCPU計算機の複数のCPUにデータを割り振り、

境界要素法によって生成された連立方程式の係数を、有限要素法によって生成された連立方程式の係数が配置されていない、該行列の列番号が中央付近に配置して、該マルチCPU計算機の複数のCPUにデータを割り振り、

有限要素法によって生成された連立方程式の係数について、カットヒル・マキー並べ替えを行い、

境界要素法によって生成された連立方程式の係数については、境界要素法で扱う面の幾何形状より、未知数間の係数が大きいもの同士が行列内で近い位置に来るような並べ替えを行い、

有限要素法によって生成された連立方程式の並べ替え後の係数について、不完全コレスキー分解を行い、

境界要素法によって生成された連立方程式の並び替え後の係数については、対角成分の

10

20

上下のユーザが指定する所定の幅内の係数について、不完全コレスキー分解を行なう、ことを特徴とする方法。

【請求項 2】

HDDヘッドとHDD媒体との間の動作状態を数値解析するための解析対象を格子状にモデル化し、境界要素法によって生成された連立方程式と、有限要素法によって生成された連立方程式の係数を、同一の行列内に配置し、高速にマルチCPU計算機に処理させるためのプログラムであって、

該マルチCPU計算機に、

有限要素法によって生成された連立方程式の係数を、該行列の列番号が小さい側と大きい側に配置させ、該マルチCPU計算機の複数のCPUにデータを割り振るステップと、

境界要素法によって生成された連立方程式の係数を、有限要素法によって生成された連立方程式の係数が配置されていない、該行列の列番号が中央付近に配置させ、該マルチCPU計算機の複数のCPUにデータを割り振るステップと、

有限要素法によって生成された連立方程式の係数について、カットヒル・マキー並べ替えを行わせるステップと、

境界要素法によって生成された連立方程式の係数については、境界要素法で扱う面の幾何形状より、未知数間の係数が大きいもの同士が行列内で近い位置に来るような並べ替えを行わせるステップと、

有限要素法によって生成された連立方程式の並べ替え後の係数について、不完全コレスキー分解を行わせるステップと、

境界要素法によって生成された連立方程式の並び替え後の係数については、対角成分の上下のユーザが指定する所定の幅内の係数について、不完全コレスキー分解を行わせるステップと、

を実行させるためのプログラム。

【請求項 3】

HDDヘッドとHDD媒体との間の動作状態を数値解析するための解析対象を格子状にモデル化し、境界要素法によって生成された連立方程式と、有限要素法によって生成された連立方程式の係数を、同一の行列内に配置し、高速に処理するマルチCPU計算機であって、

有限要素法によって生成された連立方程式の係数を、該行列の列番号が小さい側と大きい側に配置して、該マルチCPU計算機の複数のCPUにデータを割り振る手段と、

境界要素法によって生成された連立方程式の係数を、有限要素法によって生成された連立方程式の係数が配置されていない、該行列の列番号が中央付近に配置して、該マルチCPU計算機の複数のCPUにデータを割り振る手段と、

有限要素法によって生成された連立方程式の係数について、カットヒル・マキー並べ替えを行う手段と、

境界要素法によって生成された連立方程式の係数については、境界要素法で扱う面の幾何形状より、未知数間の係数が大きいもの同士が行列内で近い位置に来るような並べ替えを行う手段と、

有限要素法によって生成された連立方程式の並べ替え後の係数について、不完全コレスキー分解を行う手段と、

境界要素法によって生成された連立方程式の並び替え後の係数については、対角成分の上下のユーザが指定する所定の幅内の係数について、不完全コレスキー分解を行なう手段と、

を備えることを特徴とするマルチCPU計算機。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、有限要素法と境界要素法による結合方程式を高速に解いて、電子デバイス等の性能評価などを行うための高速演算処理方法に関する。

【背景技術】

【0002】

有限要素法で得られた方程式と境界要素法で得られた方程式を結合した結合方程式を生成し、これを数値解析する場合（静電場、静磁場、動磁場、電磁場のシミュレーション）、大規模連立方程式を並列計算機環境で高速に解く必要がある。

【0003】

計算速度を向上するためには、計算機間の通信量を少なくするためのデータ並べ替えが必用である。また、境界要素法による連立方程式の行列は大きくなるため、前処理を行う領域を制限する必要があるが、領域の制限のしかたによって計算時間が大きく変わる。

【0004】

有限要素法と境界要素法を結合する数値解析分野では、従来、大規模連立方程式を直接的に解く手法（直説法）が使用されている。直接法は反復法に比べて計算負荷が大きく、膨大な計算時間を要する。

【0005】

有限要素法と境界要素法を結合する数値解析（静電場、静磁場、動磁場、電磁場）は、モデルの移動を考慮した解析が可能となるため、有効な解析手法である。通常、境界要素法で作成される連立方程式は0成分が無い密行列となり、解法には多くの時間を要することが分かっている。このような特徴を持つ連立方程式は反復法で解くことができるが、不完全コレスキー分解による前処理を行うことができないため収束効率が悪い。そのため、直接法が使用されている。また、有限要素法と結合しても、行列の密な領域が存在するため、同様に直接法による解法を用いるのが従来の解法である。

【0006】

図1は、行列の種類を説明する図である。

図1（a）は、疎行列である。疎行列とは、対角成分の近傍成分以外の係数が0となっている行列である。図1（b）は、密行列である。密行列とは、0となる係数がない行列のことである。図1（c）は、疎密行列である。疎密行列とは、行列のある部分は疎行列としての性質を持ち、他の部分は密行列としての性質を持つ、疎行列と密行列が混ざったような行列のことである。

【0007】

特許文献1においては、並列計算機の計算効率を上げるため、連立一次方程式の前処理行列を作成し、数値解の収束性を向上させる技術が開示されている。

【特許文献1】特開平5-81310号公報

【発明の開示】

【0008】

本発明の課題は、有限要素法と境界要素法とを結合する数値解析において演算すべき行列の演算を高速に行うことができる高速演算処理方法を提供することである。

本発明の高速演算処理方法は、有限要素法と境界要素法とを用いて、解析対象を格子状にモデル化し、連立方程式を生成した際の連立方程式を構成する係数行列の高速演算処理方法であって、境界要素法でモデル化された境界の格子で規定される各辺について、同一平面状になく、かつ、互いの間の距離が小さい方から順に2つの辺を取得し、該互いの距離が小さい方から順に取得された2つの辺に対して得られる、連立方程式の係数行列の行列要素を、該係数行列の対角要素近傍に並べ替えることを特徴とする。

【0009】

本発明に基づく並べ替えを行うことで、1万自由度以上の未知数の一般的なモデルを用いた数値解析において、100倍以上高速に計算することが可能である。直接法では計算に必要なメモリ容量が多く、2GBで5万自由度が限界であるが、本発明を用いると100万自由度まで計算可能である。

【0010】

従来の手法の手法では、直接法（計算時間増大、メモリ容量の増大化）を用いており、従来の不完全コレスキー分解の前処理付BiCG法反復法（並列化効率の低下）を使ったとし

10

20

30

40

50

ても、並べ替え処理無しでは並列化効率低下し、バンド幅制限をしなければ、計算時間増加、メモリ使用量が増加を招いてしまう。これに対し、本発明（並べ替え処理+バンド幅制限）では、並列計算の高速化を実現可能である。

【図面の簡単な説明】

【0011】

【図1】行列の種類を説明する図である。

【図2】本発明の主要部の流れを示すフローチャートである。

【図3】本発明の実施形態の概略を説明する図である。

【図4】数値解析すべきモデルとそれによってできる行列について説明する図（その1）である。

10

【図5】数値解析すべきモデルとそれによってできる行列について説明する図（その2）である。

【図6】マルチCPU計算機での計算の仕方を説明する図（その1）である。

【図7】マルチCPU計算機での計算の仕方を説明する図（その2）である。

【図8】FEMで得られた連立方程式とBEMで得られた連立方程式の連立方法を説明するフローチャートである。

【図9】Cuthill-McKeeの並べ替え手法を説明する図（その1）である。

【図10】Cuthill-McKeeの並べ替え手法を説明する図（その2）である。

【図11】Cuthill-McKeeの並べ替え手法を説明する図（その3）である。

【図12】Cuthill-McKeeの並べ替え手法を説明する図（その4）である。

20

【図13】BEM領域に対する並べ替え処理を説明する図（その1）である。

【図14】BEM領域に対する並べ替え処理を説明する図（その2）である。

【図15】BEM領域に対する並べ替え処理を説明する図（その3）である。

【図16】BEM領域に対する並べ替え処理を説明する図（その4）である。

【図17】BEM領域に対する並べ替え処理を説明する図（その5）である。

【図18】前処理を説明するための図である。

【図19】マルチCPU型の並列計算機において本発明の実施形態の計算を行う場合の各CPUへのデータの振り分け方法を説明するフローチャート（その1）である。

【図20】マルチCPU型の並列計算機において本発明の実施形態の計算を行う場合の各CPUへのデータの振り分け方法を説明するフローチャート（その2）である。

30

【発明を実施するための最良の形態】

【0012】

本発明の実施形態では、境界要素法で扱う面の幾何形状より、未知数間の係数が大きいもの同士が行列内で近い位置にくるような並べ替えを行うことで、反復による収束を速くするための前処理の効率化を図る。

【0013】

図2は、本発明の主要部の流れを示すフローチャートである。

ここでは、有限要素法と境界要素法とを用いて、解析対象を格子状にモデル化し、連立方程式を生成した際の連立方程式を構成する係数行列の高速演算処理方法における並べ替え処理方法を示す。

40

【0014】

まず、Step1で、境界要素法でモデル化された境界の格子で規定される各辺について、同一平面状になく、かつ、互いの間の距離が小さい方から順に2つの辺を取得する。

Step2では、前記解析対象のモデルが、1つの境界によって2つの領域A及び領域Bに分割される場合、前記係数行列の中央部分に、該1つの境界に境界要素法を用いて得られた連立方程式の係数を配置し、その前後に、該2つの領域A及び領域Bに有限要素法を適用して得られた連立方程式の係数を配置する。

【0015】

Step3では、前記係数行列において、境界要素法による係数の前部分に配置される有限要素法による前記領域Aについての係数は、領域Aを記述する未知数であって、かつ

50

、前記1つの境界を記述する未知数の内、最もインデックス値が小さい未知数に関する連立方程式の係数を開始位置として、カットヒル・マキー (Cuthill-McKee) 並べ替えを行う。また、前記係数行列において、境界要素法による係数の後ろ部分に配置される有限要素法による前記領域Bについての係数は、領域Bを記述する未知数であって、かつ、前記1つの境界を記述する未知数の内、最もインデックス値が大きい未知数に関する連立方程式の係数を開始位置として、カットヒル・マキー並べ替えを行う。

【0016】

Step 4では、境界要素法の係数について、該互いの距離が小さい方から順に取得された2つの辺に対して得られる、連立方程式の係数行列の行列要素を、該係数行列の対角要素近傍に並べ替える。

10

【0017】

図3は、本発明の実施形態の概略を説明する図である。

境界要素法で解く領域は密領域であるため、反復解法を用いて前処理に不完全コレスキー分解を行うと、結果的に完全なコレスキー分解となってしまう、直接法による解法と同じになってしまう。そこで、密領域の全ての係数を使用せずに、前処理による反復法の収束性を高める必要がある。

【0018】

有限要素法と境界要素法の混合解法では、図3左図のように行列の対角要素付近にノンゼロの値が集中する領域1と正方形に近い形の領域にノンゼロの値が集中する領域2が存在する。有限要素法解析では一般に行列全体に対して不完全コレスキー分解を行うが、このような行列に対して不完全コレスキー分解を行うと、領域2において計算時間の増加を招いてしまい非効率である。そこで領域2に対して図3右図のように対角要素の近傍の値だけを対象に不完全コレスキー分解を行うことで、前処理時間の短縮化を行う。

20

【0019】

すなわち、本発明の実施形態においては、有限要素法と境界要素法に基づく連立方程式の合成より成り立つ行列において、全ての未知数中の境界要素法で使用する未知数の並べ替えを初めに行い、続いて残りの未知数の並べ替えを行う。

【0020】

図4及び図5は、数値解析すべきモデルとそれによってできる行列について説明する図である。

30

図4(a)のモデルは、HDDヘッドとHDD媒体との間の動作状態を数値解析するためのモデルである。このHDDヘッド11と媒体10を含む空間をそれぞれモデルBとモデルAとして、格子に区切り、各格子点について物理量を考える。モデルAとモデルBの相互作用は、格子点間の相互作用として記述し、相互作用そのものは物理理論(たとえば、電磁気学)の法則にしたがって記述する。媒体10を含むモデルA及び、HDDヘッド11を含むモデルBはそれぞれ、媒体10、HDDヘッド11のみではなく、その周りの空気領域も含めてモデル化したものである。このモデルAとBの内部は、有限要素法を用いて計算されるが、モデルAとモデルBの両方の最外周領域は、2つのモデルの最外周を1つの外周を表すモデルとしてモデル化し、外周を格子に区切って、物理量を計算する。図4(a)の場合、媒体10とHDDヘッド11が主に相互作用する境界要素面14が、モデルAとモデルBの相対する外周面として記述される。このような外周境界の計算は、境界要素法を用いて計算される。

40

【0021】

以上のようにモデル化された解析対象について、有限要素法と境界要素法を用いて連立方程式を立て、その係数を行列として整理すると、図4(b)のような行列となる。すなわち、モデルAの内部を記述する係数は、図4(b)の行列の領域1のように対角要素の周辺にノンゼロの係数が集中する。同様に、モデルBの内部を記述する係数は、領域3のように対角要素の周辺にノンゼロの係数が集中する。一方、境界をモデル化し、境界要素法で連立方程式をつくり、係数を行列表示すると、領域2のように、その部分だけ、密行列のようになっている。

50

【0022】

境界要素法で使用する未知数の並べ替えは、代数的にモデル端部に位置する未知数（図4（a）のa点のような点における、求めるべき物理量をあらわす x などの未知数。モデルが格子化されているので、各格子点での物理量を x_i のようにインデックス i をつけて記述する）を開始 $No=1$ とし、それ以外の未知数との行列成分（連立方程式は、 $a_{ij}x_j + a_{i,j+1}x_{j+1} + \dots = b_i$ のような式からなっており、 x_k を $No=1$ とした場合、 a_{ik} ）が最も大きい未知数を $No=2$ として登録する。最後に登録された未知数 $No=n$ に対して、まだ登録されていない未知数との行列成分が最も大きい未知数を最後の No に $n+1$ 番目で登録する。以上の手順でナンバーリングされた番号を境界要素法領域における連立方程式の未知数の順番と定義する。

10

【0023】

図4（a）のように、境界要素法を使用する場合、モデルの領域が複数の領域に分割されている。媒体移動を伴うHDDヘッドのライト・リードシミュレーションのように、ヘッドと媒体というように2つの大きな領域に分割されている場合に、以下の並べ替えを行うことで効率的な領域分割を実現する。

【0024】

連立方程式より作成される行列の領域1はモデル左側の媒体+空気領域の境界要素面以外の未知数から構成し、領域2は境界要素面の未知数から構成し、領域3はモデル右側のリードヘッド+空気領域の境界要素面以外の未知数から構成する。

【0025】

20

領域2の並べ替え後に、媒体側未知数でかつ領域2の未知数のインデックス番号が最も小さい番号を開始番号として領域1の並べ替えを行う。同様に、HDDヘッド側未知数でかつ領域2の未知数のインデックス番号が最も大きい番号を開始番号として領域3の並べ替えを行う。領域1～3の並べ替え後の順番に従って、最後に領域1→領域2→領域3の通し番号でインデックス番号付けを行って並べ替えを終了する。

【0026】

境界要素法の未知数間の係数（未知数 x_i と未知数 x_j 間の係数は a_{ij} ）は、3次元解析の場合は未知数が定義される面の単位法線ベクトルと、問題とする未知数が定義される位置から、相互作用を考慮すべき未知数が定義される位置への位置ベクトルとの内積に比例するため、同一平面に存在する面内の未知数間の係数は0となる。そこで、連立方程式作成過程における計算では、境界要素面を構成する全ての未知数の組み合わせに対して成分計算は行わず、予め同一平面上に存在するという情報を面要素グループに設定し、異なる面要素グループとの組み合わせに含まれる面に属する未知数間の係数だけを成分計算することで計算時間の短縮化を行う。

30

【0027】

なお、図4（b）に示されるように、対角要素近傍のノンゼロ領域の幅をバンド幅と呼ぶ。領域2のバンド幅は広いため、前処理において、そのままのバンド幅を使用すると計算時間が増加してしまう。前処理に使用するバンド幅は、必ずしも各領域のバンド幅と同じである必要はない。そこで、図5（a）の領域2に対してはユーザーが任意にバンド幅を指定ユーザ処理を行う領域を図5（b）の領域2のように制限することを可能とする。

40

【0028】

図6及び図7は、マルチCPU計算機での計算の仕方を説明する図である。

複数のCPUを使用する並列計算では、ユーザが使用する計算機ノード数を入力し、行列を図6のようにブロック分割する。また、各計算機内に複数のCPUが存在する場合、ユーザが並列スレッド数を入力すると、計算機ノード数×スレッド数で行列をブロック分割する。

【0029】

要素数を単純に分割するだけでは、各CPUに加わる計算負荷が均等にならない。計算負荷は、モデル形状やメッシュ（格子）分割方法、FEM（Finite Element Method：有限要素法）領域であるかBEM（Boundary Element Method：境界要素法）領域であるかに依存する

50

。そこで、各未知数に対する一回の反復計算あたりの計算時間をテスト計算により予め算出し、計算付加が均等になるように各CPUに未知数を割り当てる。

【0030】

図6では、CPU1～5に行列の係数を上から順に割り当てている。図6のCPU3に割り当てられた係数は、ユーザが制限したバンド幅がCPU3に割り当てられた係数のインデックス幅より大きい場合の前処理サイズである。一方、図7(a)のCPU3に割り当てられた係数は、ユーザが制限したバンド幅がCPU3に割り当てられた係数のインデックス幅より小さい場合の前処理サイズである。

【0031】

ユーザが設定したBEM領域のバンド幅制限サイズとCPUに割り当てられた未知数サイズの大小関係を自動的に判断し、図7(b)のようなBEM領域の前処理ブロックを、図7(c)のように、バンド幅制限サイズに自動的に制限して計算を行う。

10

【0032】

有限要素法で生成される行列のノンゼロ成分は、同じ材料物性グループ内では、大きい値と小さい値との差は小さい。しかしながら、境界要素法によって生成される行列成分は、未知数と未知数の間の係数の値は、未知数が配置されている位置間の距離の2乗に反比例する(3次元解析)ため、係数の大きさは幅広く分布している。そこで未知数間の係数が大きい値が行列対角部分に集中するような並べ替えを行い、対角近傍成分だけを用いて反復解法における不完全コレスキー分解を行う。どの範囲までの対角近傍成分幅(制限バンド幅)を使用するかは、ユーザが予め数値入力して指定する。また、最大係数に対してどの値までを前処理に含めるかという閾値をユーザが指定して、制限バンド幅を計算することを可能にする。

20

【0033】

HDDのヘッド-媒体解析やモータ解析のように、境界要素領域によって、解析すべきモデル全体が2つの領域に分割されるケースが多い。行列としては、中央に境界要素法で得られた係数がきて、前後に有限要素法で得られた係数を配置することで、FEM-BEM全体行列の前処理を効率的に行うことができ、かつ並列計算における計算機間の通信料を少なくすることができる。

【0034】

有限要素法等を使用する数値解析では、大規模な連立方程式を作成し、解く必要がある。連立方程式は、 A を係数行列、 x を未知数の列ベクトル、 b を非斉次項の列ベクトルとして、 $Ax=b$ の形で表現でき、 A の性質によって解法が異なる。 A を行列として表現すると、対称行列の場合と非対称行列の場合の大きく分けて2つの場合がある。また、行列 A の成分が0の領域が多い疎行列、0以外の領域が多い密行列、そして、疎と密がそれぞれ存在する疎密行列の3種類が存在する。一般に有限要素法解析では疎行列、境界要素法解析では密行列、有限要素法と境界要素法の混合解析では疎密行列を解く必要がある。

30

【0035】

連立方程式を行列で表現すると以下のようになる。

【数 1】

$$\begin{pmatrix} A_{11} & : & A_{1n} \\ : & : & : \\ A_{n1} & : & A_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ : \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ : \\ b_n \end{pmatrix}$$

10

$$\text{対称行列} : \begin{pmatrix} A_{11} & B_{12} \\ B_{21} & A_{22} \end{pmatrix} \quad (B_{21} = B_{12})$$

$$\text{非対称行列} : \begin{pmatrix} A_{11} & B_{12} \\ B_{21} & A_{22} \end{pmatrix} \quad (B_{21} \neq B_{12})$$

20

図 8 は、FEM で得られた連立方程式と BEM で得られた連立方程式の連立方法を説明するフローチャートである。

【 0 0 3 6 】

以下のような手順により、連立方程式を得ることができる。

ステップ S 1 0 : 連立方程式係数の初期化 $A_{ij} = 0$ 、 $B_i = 0$

ステップ S 1 1 : FEM に基づく連立方程式の作成

解析領域全ての格子点に要素（インデックス）番号を付加する。

各要素の辺 i- 辺 j 番目の係数を計算して A_{ij} に追加する。

磁性体要素の場合 B_i に値を追加する。

電流要素の場合 B_i に値を追加する。

30

ステップ S 1 2 : BEM に基づく連立方程式の作成

BEM 境界全ての辺 i に対して、それ以外の辺 j の係数を計算して A_{ij} に追加する。

ステップ S 1 3 : 連立方程式作成完了

FEM 連立方程式係数の作成過程における係数計算は以下のようにして行う。

透磁場率： μ 、電流ベクトル：

$$\vec{J}$$

、磁石の磁化ベクトル：

40

$$\vec{M}$$

各要素に対して

【数 2】

$$A_{ij} = \frac{1}{\mu} \int_V (\nabla \times N^T) \cdot (\nabla \times N) dV$$

を計算しながら全ての要素に対しする大規模な A_{ij} を計算する。ここで、 N は、モデルの形状を現す補間関数であり、補間関数をモデルの格子に沿って離散化して、列ベクトルとしたものである。 dV は体積要素である。

10

【0037】

電流要素に対しては

【数 3】

$$B_i = \int_V N^T J dV$$

を計算しながら大規模な B_i を作成する。

【0038】

20

磁性体要素に対しては

【数 4】

$$B_i = \int_V N^T (\nabla \times N) M dV$$

を計算しながら大規模な B_i を作成する。

並べ替え処理は、以下の手順で行う。

(1) BEMでモデル化される領域で分離されるFEMでモデル化される領域AとBを作成する。

30

(2) BEMでモデル化された境界を構成する面領域Cの辺をID=−1とする。

(3) 面領域Cの重心を計算する。

(4) 領域Aに属し、かつ、重心から最も離れた位置の辺の番号Nを取得する。

(5) 領域Bに属し、かつ、N番目の辺から最も離れた位置の辺の番号Mを取得する。

(6) 全ての、辺番号が与えられている辺のIDを0に初期化する。

(7) 領域Aの辺番号Nを開始番号とし、この辺のIDを1とする。

(8) ID=1の辺に隣接する辺番号が0の場合、この辺をID=2とする。

(9) ID=nの辺に属する辺番号が0の場合、辺ID=n+1とする。

(10) 領域Aの辺ID=0が無くなるまで繰り返し、最後のID=Xを記憶する。

(11) 領域Bの辺番号Mを開始番号とし、この辺のIDを1とする。

40

(12) ID=1の辺に隣接する辺番号が0の場合、この辺をID=2とする。

(13) ID=nの辺に属する辺番号が0の場合、辺ID=n+1とする。

(14) 領域Bの辺ID=0が無くなるまで繰り返す。

(15) ID=Xを初期値として、逆の並べ替えをする（領域A）。

(16) ID=−1の辺領域を順番に並べ替える（面領域C）

(17) 領域Bを面領域Cの後に順番に並べる（領域B）

並べ替えは、BEM領域とそれ以外のFEM領域に対して処理を行う。FEM領域に対しては、Cuthill-McKeeの並べ替え手法を用いる。

【0039】

図9～図12は、Cuthill-McKeeの並べ替え手法を説明する図である。

50

FEM+BEMの並べ替えでは、BEMで必要とする未知数以外のデータに対しては、Cuthill-Mc kee並べ替えによって並べ替えを行う。図9に示されるように、データ構造は、節点データは接続する辺番号および辺で結ばれる隣接節点番号をトポロジーデータとして持つ。磁場計算では、辺に未知数を配置するため、並べ替えに使用する並べ替え番号を辺に配置する。

<節点データ構造>

```
struct node_tpg{
    int node_num[4];    //点⇒隣接点
    int edge_num[4];    //点⇒辺
};
```

10

辺データは、辺両端の節点番号をトポロジーデータとして持つ。

<辺データ構造>

```
struct edge_tpg
{
    int node_num[2];    //辺⇒2点
    int n_order;        //並べ替え番号
};
```

Cuthill-McKee並べ替えは、開始となる辺番号を指定し、辺が持つ節点が接続する辺番号の順で並べ替えを行う。近接する辺データを同心円状に広げていく過程で、参照するリストと蓄えるためのリストが必要となるため、リストデータを2つ準備し、それぞれのリストをバックグラウンドリスト、フォアグラウンドリストとして切り替え可能とする。

20

【0040】

初期化では、全ての辺番号の並べ替え番号を-1とし、-1以外番号が設定されているデータは、既に並べ替えされたデータとして判定する。初めに開始となる一つの辺番号を入力し、その辺番号を開始位置として、辺→節点→辺→節点・・・と順番に辺を呼び出し、呼び出される順に辺がナンバーリングされる。

【0041】

具体的には、既に追加されているフォアグラウンドのリストの辺に対して、両端の節点番号N1,N2を取得し、それぞれの持つ辺番号(未だ並べ替えされていない)をバックグラウンドリストに追加する。バックグラウンドリストに追加する際に、n_order=Nedgeを代入し、Nedge=Nedge+1とする。

30

【0042】

図10～図12のフローチャートに従って説明する。

まず、ステップS20において、すべての辺番号の並べ替え番号を-1とする。2つのリストをバックグラウンドリストとする。ステップS21において、エッジ番号を一つのバックグラウンドリストに追加し、別のリストをフォアグラウンドリストとする。また、作業変数Nallを1に初期化する。ステップS22において、フォアグラウンドリストとバックグラウンドリストをバックグラウンドリストとし、リストデータを空にする。バックグラウンドリストをフォアグラウンドリストとする。ステップS23において、フォアグラウンドイリスとからエッジ番号を取得し、Nedgeに設定する(Nedge0とする)。

40

【0043】

ステップS24において、取得された辺の両端ノードN1、N2を取得する。ステップS25において、ノードN1の辺を取得し、I=0とする。ステップS26において、NedgeがIより大きいかな否かを判断する。ステップS26の判断がNoの場合には、図12に進む。ステップS26の判断がYesの場合には、ステップS27において、ノードN1の辺(Nedge1とする)を取得する。ステップS28において、辺(Nedge1)の並べ替え番号が0より小さいかな否か、すなわち、並べ替え済みかな否かを判断する。ステップS28の判断がNoの場合には、Iを1増加してステップS26に戻る。ステップS28の判断がYesの場合には、ステップS29において、この辺番号をバックグ

50

ラウンドリストに追加し、この辺 (N e d g e 1) の並べ替え番号を N a l l にし、N a l l を 1 増加し、I を 1 増加して、ステップ S 2 6 に戻る。

【0044】

ステップ S 3 0 において、ノード N 1 の辺を取得し、 $I = 0$ とする。ステップ S 3 1 において、N e d g e が I より大きいかな否かを判断する。ステップ S 3 1 の判断が N o の場合には、ステップ S 3 5 に進む。ステップ S 3 1 の判断が Y e s の場合には、ステップ S 3 2 において、ノード N 1 の辺 (N e d g e 1 とする) を取得する。ステップ S 3 3 において、辺 (N e d g e 1) の並べ替え番号が 0 より小さいかな否か、すなわち、並べ替え済みかな否かを判断する。ステップ S 3 3 の判断が N o の場合には、I を 1 増加してステップ S 3 1 に戻る。ステップ S 3 3 の判断が Y e s の場合には、ステップ S 3 4 において、この辺番号をバックグラウンドリストに追加し、この辺 (N e d g e 1) の並べ替え番号を N a l l にし、N a l l を 1 増加し、I を 1 増加して、ステップ S 3 1 に戻る。

10

【0045】

ステップ S 3 5 では、バックグラウンドリストが空かな否かを判断する。ステップ S 3 5 の判断が N o の場合には、図 1 0 の A に戻る。ステップ S 3 5 の判断が Y e s の場合には、処理を終了する。

【0046】

図 1 3 ～ 図 1 7 は、BEM 領域に対する並べ替え処理を説明する図である。

この処理は、境界要素法の処理を行うプロセスが連立方程式を生成する前に、境界要素法を適用する境界部分の格子の配列番号を並べ替え、この並べ替えられた配列番号に基づいてプロセスが連立方程式を生成すれば、係数値の大きな連立方程式の係数が行列の対角要素付近に集まるようになるというものである。

20

【0047】

BEM で作成される方程式は、辺同士が互いに近い場合に係数が大きくなる。連立方程式の対角部分に大きな数値を配置するためには、幾何学的に近い位置にある辺番号が近くなるように工夫を行う必要がある。そこで、BEM 境界面に対して背景にセル状の格子をつくり、各セルに対応する辺を登録可能なリストを作成し、このリストのセルの中に辺の中心位置が存在する辺をリストの当該セルに代入することで並べ替えを行う。

【0048】

図 1 3 では、BEM 境界は、モデル A とモデル B の Y - Z 面に存在している。その BEM 境界に背景セルを配置した様子が図 1 4 である。

30

図 1 4 においては、グリッド座標の原点は (x_0, y_0) とし、Z 軸方向に dz 刻み、Y 軸方向に dy 刻みでグリッド座標値が増加する。位置 (i, j) のグリッド座標値は、 $(z_0 + i * dz, y_0 + j * dy)$ となる。また、セル (I, J) は、Z 座標が $z_0 + I * dz \sim z_0 + (I + 1) * dz$ 、Y 座標が $y_0 + J * dy \sim y_0 + (J + 1) * dy$ の範囲の領域である。

ゆえに、追加する辺の座標値 (Z, Y) が分かれば、セルの (I, J) が以下の式で計算できる。

$$I = (Z - z_0) / dz \text{ の整数部}$$

$$J = (Y - y_0) / dy \text{ の整数部}$$

図 1 5 は、BEM 境界を構成する辺の並べ替えの全体フローチャートである。

【0049】

ステップ S 4 0 で、上記のように配列リスト (背景のセル状リスト) へ辺番号を代入する。ステップ S 4 1 で、配列リストを用いた並べ替えを行う。

40

図 1 6 は、配列リストへの辺番号代入の処理を詳細に示したフローチャートである。

【0050】

ステップ S 4 5 において、コンピュータのキャッシュに $I_{max} \times J_{max}$ のセルからなる配列リストのデータエリアを確保し、その値を空にし、変数 N を格納する領域をレジスタに確保し、このレジスタ値を 0 にする。ステップ S 4 6 において、BEM 境界を構成する辺の数を、境界要素法を実行するプロセスから取得し、他のレジスタに N e d g e _ a l l として設定する。ステップ S 4 7 において、境界要素法を実行するプロセスが生成した、格子化された BEM 境界を構成する辺番号を取得する。通常は、この辺番号はばらばら

50

な配列となっている。この辺番号をNedgeとする。ステップS48において、Nedgeの辺番号を有する辺の座標（z、y）を、境界要素法を実行するプロセスが生成したモデルから取得し、配列リストを有する背景セルの座標（I、J）を図14で説明した計算式を演算器で計算することによって取得する。ステップS49において、背景セル座標（I、J）の配列リストに辺番号Nedgeを追加する。ステップS50において、NがNedge_allより小さいか否かを判断する。ステップS50の判断がYesの場合には、Nを1増加し、ステップS47に戻る。ステップS50の判断がNoの場合には、処理を終了する。

【0051】

図17は、配列リストを用いた並べ替えの処理を詳細に示したフローチャートである。

ステップS55において、3つのレジスタに変数N、I、Jを設定し、N=0、I=0、J=0と初期化する。ステップS56において、キャッシュの配列（I、J）の配列リストが空か否かを判断する。ステップS56の判断がYesの場合には、ステップS58に進む。ステップS56の判断がNoの場合には、ステップS57において、キャッシュの配列リストから辺番号Nedgeを取得し、NedgeをNに置き換える。そして、Nを1増加してステップS56に戻る。Nedgeは、一般に境界要素法を行うプロセスがランダムに境界の格子点に与えた番号である。これを順次1ずつ増加するNに置き換えることにより、格子点に与えられる番号が順序良く並べられるようになる。

【0052】

ステップS58において、レジスタのIを1増加し、ステップS59において、IがImaxより小さいか否かを判断する。ステップS59の判断がYesの場合、ステップS56に戻る。ステップS59の判断がNoの場合、ステップS60において、レジスタのJを1増加し、ステップS61において、JがJmaxより小さいか否かを判断する。ステップS61の判断がYesの場合には、ステップS56に戻る。ステップS61の判断がNoの場合には、処理を終了する。

【0053】

図18は、前処理を説明するための図である。

非対称行列を解くための反復手法にBiCG法が公知の技術として存在する。反復回数を減らすために前処理を行う手法が一般に用いられている。Ax=bという方程式に対して、正方行列AをLDUという行列の積で分解すると以下ようになる。

【0054】

【数5】

$$A = LDU =$$

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{1n} \\ A_{21} & A_{22} & A_{23} & A_{2n} \\ A_{31} & A_{32} & A_{33} & A_{3n} \\ \vdots & & & \\ A_{n1} & & & A_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & 0 \\ L_{21} & 1 & & \\ L_{31} & & 1 & \\ \vdots & & & 1 \\ L_{n1} & L_{n2} & L_{n3} & 1 \end{pmatrix} \begin{pmatrix} D_{11} & & & 0 \\ & D_{22} & & \\ & & D_{33} & \\ 0 & & & D_{nn} \end{pmatrix} \begin{pmatrix} 1 & U_{12} & U_{13} & U_{1n} \\ & 1 & & U_{2n} \\ & & 1 & \vdots \\ & & & 1 & 1 \end{pmatrix}$$

ここで、L,D,Uは以下の性質を持つ。

【0055】

10

20

30

40

【数 6】

$$A_{ij} = \sum_{k=1}^j L_{ik} D_{kk} U_{kj} \quad (1 \leq i, 1 \leq j)$$

$$U_{ii} = 1, \quad L_{ii} = 1$$

$$U_{ij} = 0 \quad (i > j) \quad , \quad L_{ij} = 0 \quad (i < j)$$

10

ここで対角成分は、

【数 7】

$$A_{jj} = \sum_{k=1}^j L_{jk} D_{kk} U_{kj} = \sum_{k=1}^{j-1} L_{jk} D_{kk} U_{kj} + L_{jj} D_{jj} U_{jj} = \sum_{k=1}^{j-1} L_{jk} D_{kk} U_{kj} + L_{jj} D_{jj} U_{jj}$$

$$\Rightarrow D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk} D_{kk} U_{kj}$$

20

非対角成分 ($i < j$) は、

【数 8】

$$A_{ij} = \sum_{k=1}^i L_{ik} D_{kk} U_{kj} = \sum_{k=1}^{i-1} L_{ik} D_{kk} U_{kj} + L_{ii} D_{ii} U_{ij} = \sum_{k=1}^{i-1} L_{ik} D_{kk} U_{kj} + D_{ii} U_{ij} \quad (i < j)$$

$$\Rightarrow U_{ij} = (A_{ij} - \sum_{k=1}^{i-1} L_{ik} D_{kk} U_{kj}) / D_{ii}$$

30

また、非対角成分 ($i > j$) は、

【数 9】

$$A_{ij} = \sum_{k=1}^j L_{ik} D_{kk} U_{kj} = \sum_{k=1}^{j-1} L_{ik} D_{kk} U_{kj} + L_{ij} D_{jj} U_{jj} = \sum_{k=1}^{j-1} L_{ik} D_{kk} U_{kj} + L_{ij} D_{jj} \quad (i > j)$$

$$\Rightarrow L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik} D_{kk} U_{kj}) / D_{jj}$$

このような演算により完全にL,D,U行列を計算してしまうと、直接解法のように演算量が多くなってしまふ。そこで、不完全コレスキー分解では以下のような処理を行う。

40

$$A_{ij}=0 \Rightarrow L_{ij}=0, U_{ij}=0$$

行列Aが疎行列であれば、L、Uの成分が0の領域がおくなり、L、Uを求める過程の演算量が少なくなる。しかしながら、BEMの連立方程式の行列Aは完全に密 ($\neq 0$) になるため、前処理が完全なコレスキー分解となってしまうために、演算量が多くなる。そこで、予め設定した前処理用のバンド幅2S領域を用いて以下のような不完全なLDU分解を前処理として採用する。

【0 0 5 6】

$A_{ij}=0$ または $j-i>S$ または $i-j>S$ のとき、 $U_{ij}=0, L_{ij}=0$

すなわち、図 1 8 で、演算をバンド幅 2 S に限定することを意味する。

50

【0057】

【数10】

$$D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk} D_{kk} U_{kj}$$

$$U_{ij} = (A_{ij} - \sum_{k=1}^{i-1} L_{ik} D_{kk} U_{kj}) / D_{ii} \quad (i < j, \text{ 但し } j-i < S)$$

10

$$L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik} D_{kk} U_{kj}) / D_{jj} \quad (i > j, \text{ 但し } i-j < S)$$

なお、以上の前処理では、バンド幅を規定して、対角要素を挟んだバンド幅内の係数についてのみ前処理を行っているが、別の方法として、係数の値に閾値を設け、係数の値が閾値より大きい係数についてのみ前処理を行うことも可能である。

【0058】

20

図19及び図20は、マルチCPU型の並列計算機において本発明の実施形態の計算を行う場合の各CPUへのデータの振り分け方法を説明するフローチャートである。

計算負荷は連立方程式 $Ax=b$ の行列 A のゼロ以外成分の大きさで決まる。各CPUには、 A_{ij} の i 成分の範囲を割り当てる。

【0059】

図19、及び、図20のフローチャートでは、 A_{ij} の $\neq 0$ 成分を N でカウントしながら、 $AN[N]=I$ に縦の番号を代入する。

ステップS70において、 $I=0$ 、 $J=0$ 、 $N=0$ と変数を初期化する。ステップS71において、CPUの数(N_CPU)と、行列のサイズ(I_{max} 、 J_{max})を計算機に入力する。ステップS72において、行列のある要素 A_{ij} が0か否かを判断する。ステップS72の判断がYesの場合には、ステップS74に進む。ステップS72の判断がNoの場合には、変数 $AN[N]$ に I を代入し、 N を1増加して、ステップS74に進む。ステップS74においては、 I が I_{max} より小さいか否かを判断する。ステップS74の判断がYesの場合には、ステップS72に進む。ステップS74の判断がNoの場合には、ステップS75において、 J が J_{max} より小さいか否かを判断する。ステップS75の判断がYesの場合には、ステップS72に戻る。ステップS75の判断がNoの場合には、ステップS76に進む。ステップS76では、 $D=N/N_CPU$ を計算する。これは、各CPUの演算量を簡易的に求めたものである。

30

【0060】

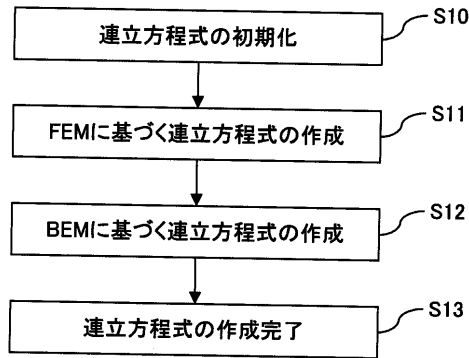
ステップS77において、 $I=0$ 、 $J=0$ 、 $N=0$ 、 $CPU[0]=0$ と、変数を初期化する。ステップS78において、 I が D より小さいか否かを判断する。ステップS78の判断がYesの場合には、ステップS80に進む。ステップS78の判断がNoの場合には、ステップS79において、 $I=0$ 、 $J=J+1$ 、 $CPU[J]=AN[N]$ を計算する。ステップS80において、 I と N を1ずつ増加し、ステップS81において、 N が M より小さいか否かを判断する。ステップS81の判断がYesの場合には、ステップS78に戻る。ステップS81の判断がNoの場合には、処理を終了する。

40

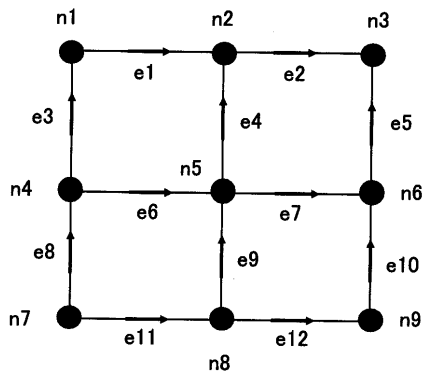
【0061】

ここで、 J 番目のCPUには $CPU[J] \sim CPU[J+1]$ の成分が割り当てられる。

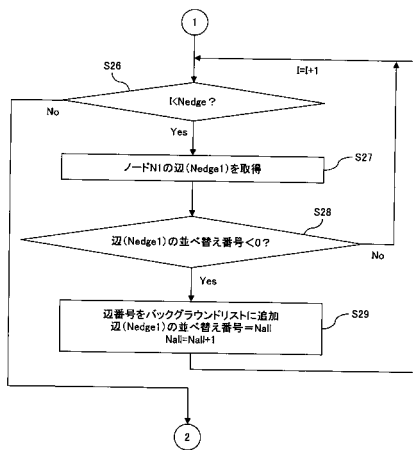
【図 8】



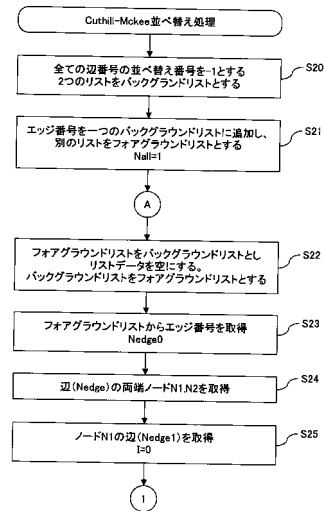
【図 9】



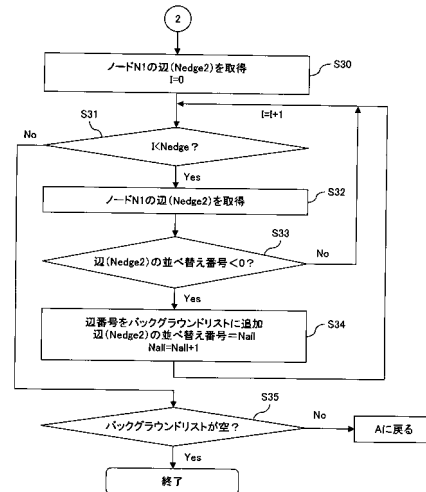
【図 11】



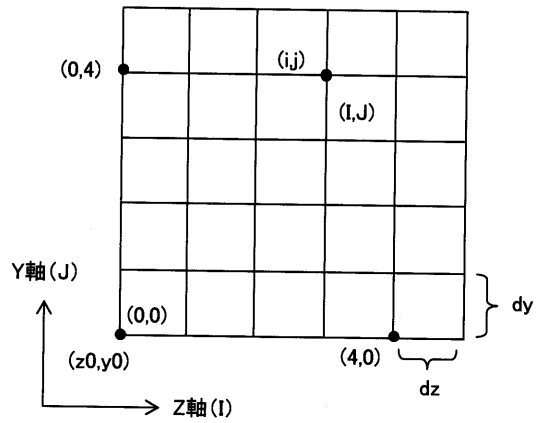
【図 10】



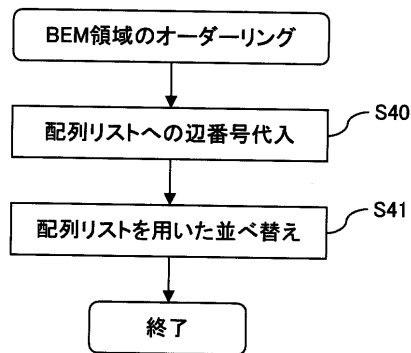
【図 12】



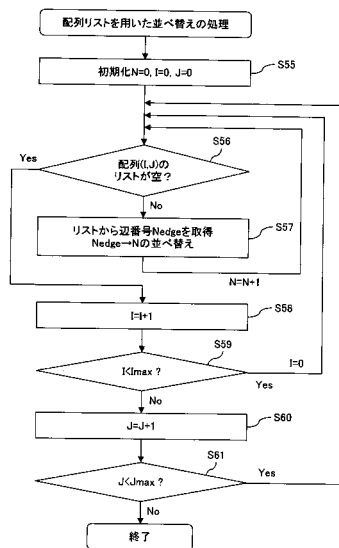
【図 14】



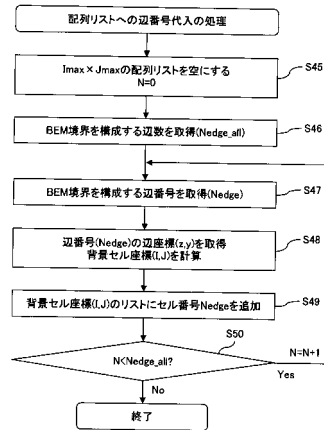
【図 15】



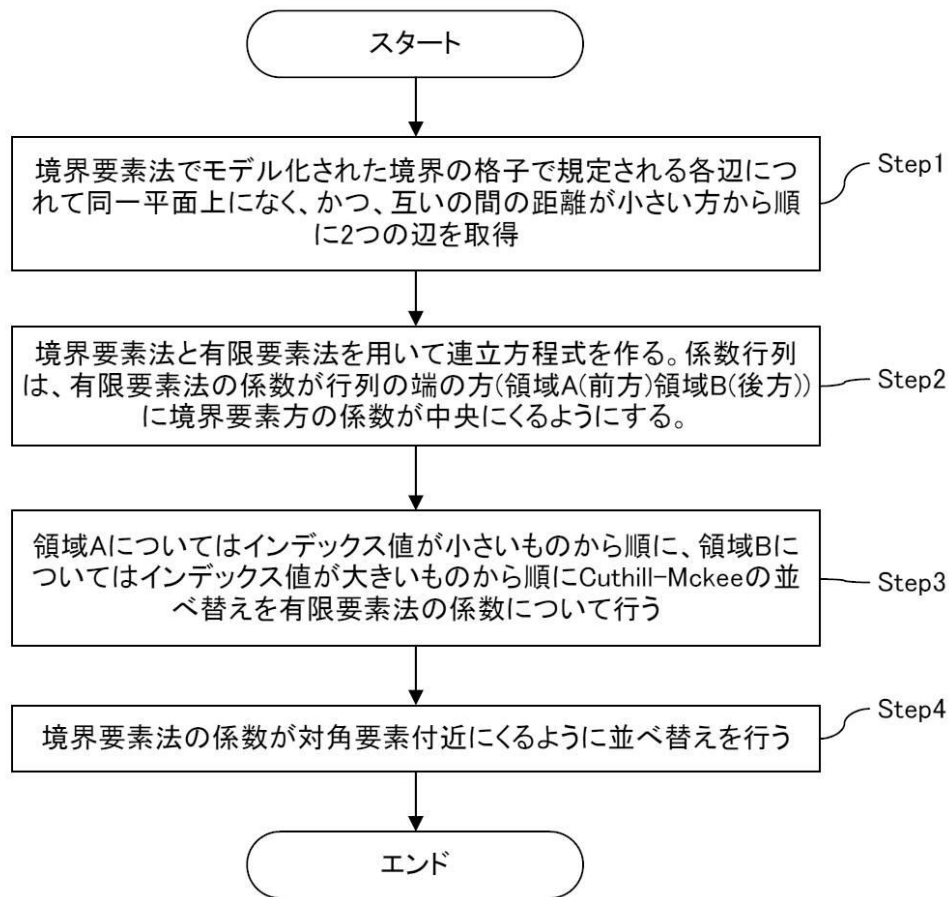
【図 17】



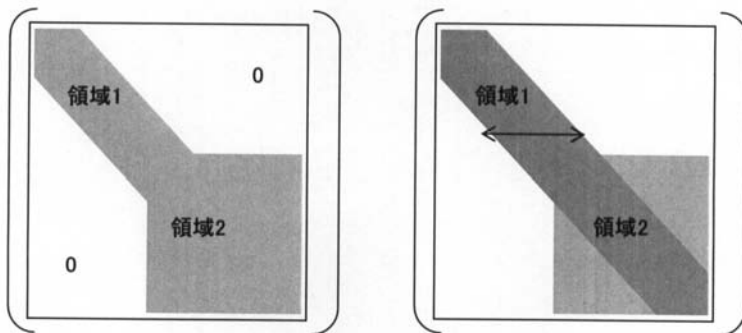
【図 16】



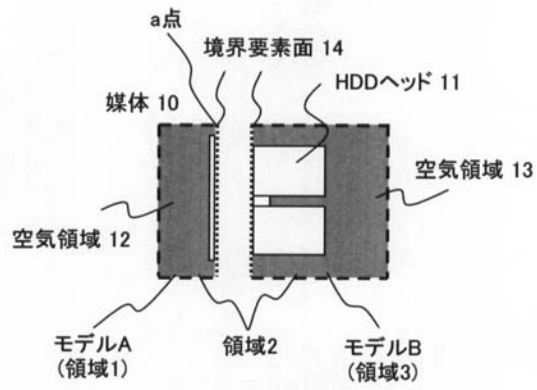
【図 2】



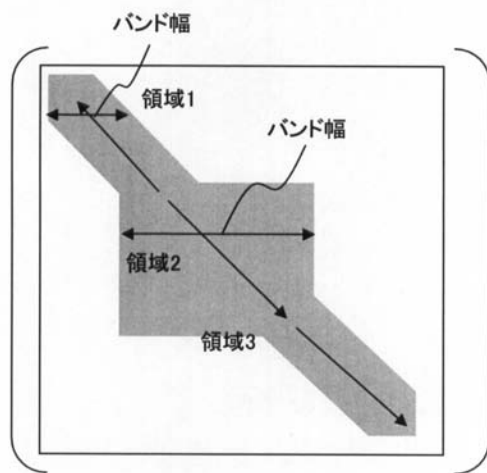
【図 3】



【図 4】

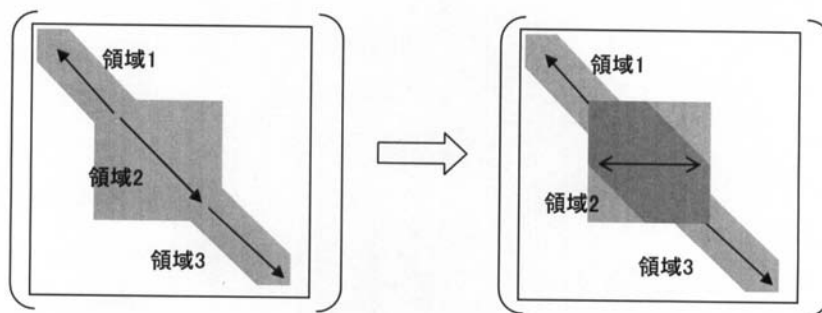


(a)



(b)

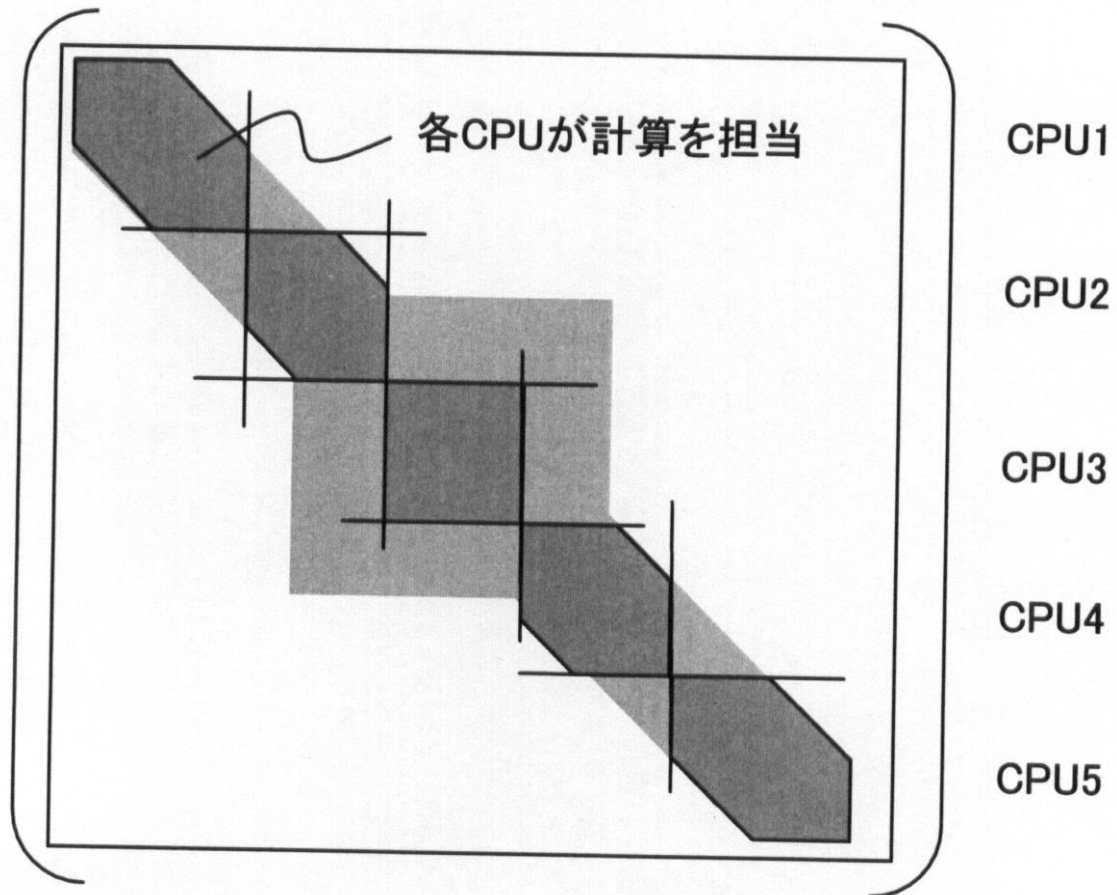
【図 5】



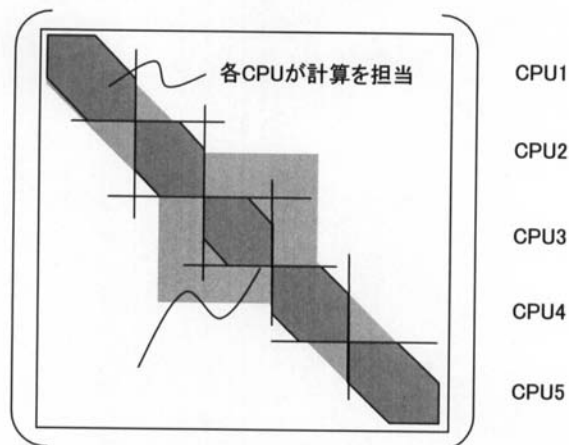
(a)

(b)

【図 6】



【図 7】



(a)

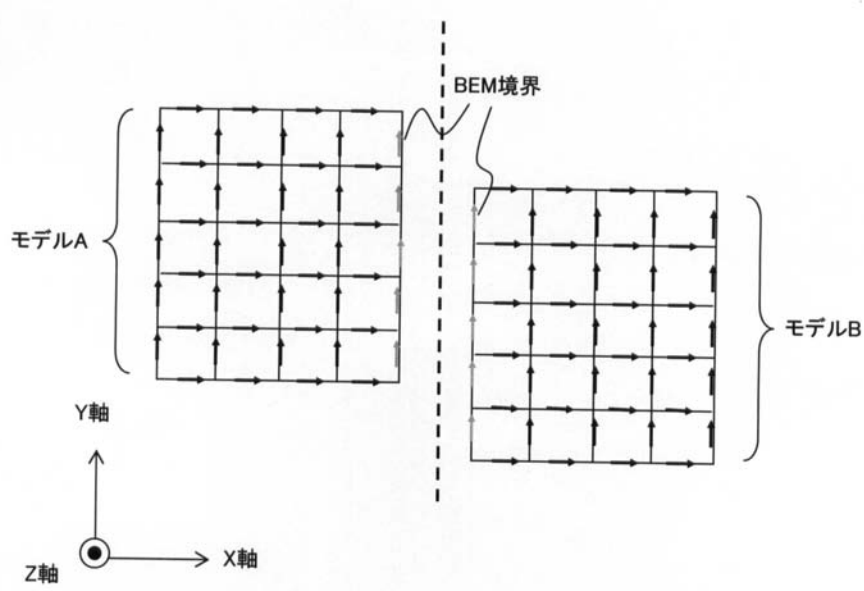
BEM領域の前処理ブロック

(b)

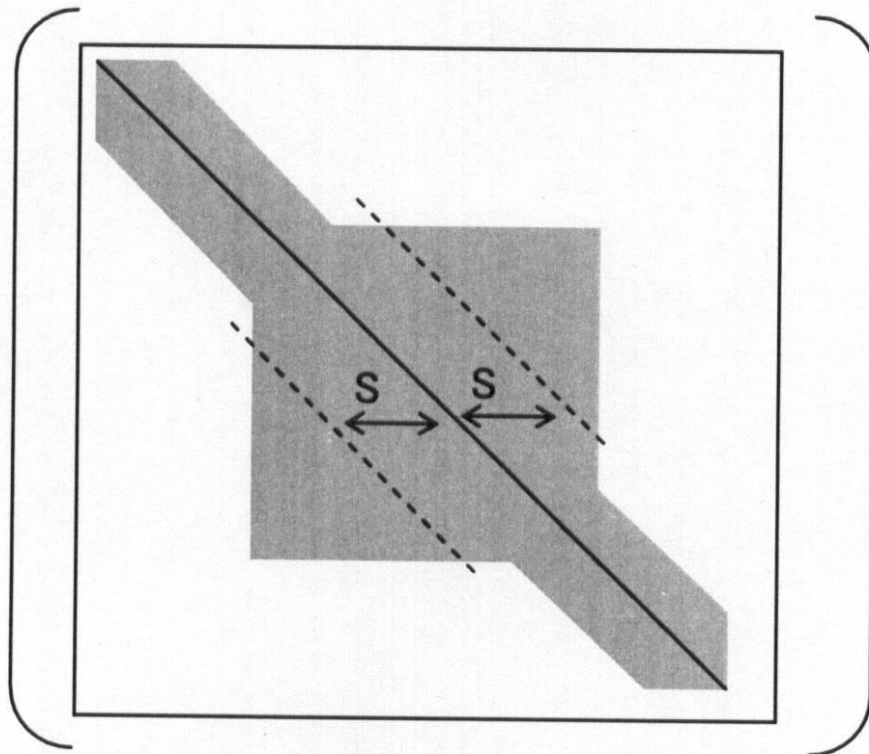
バンド幅制限サイズ

(c)

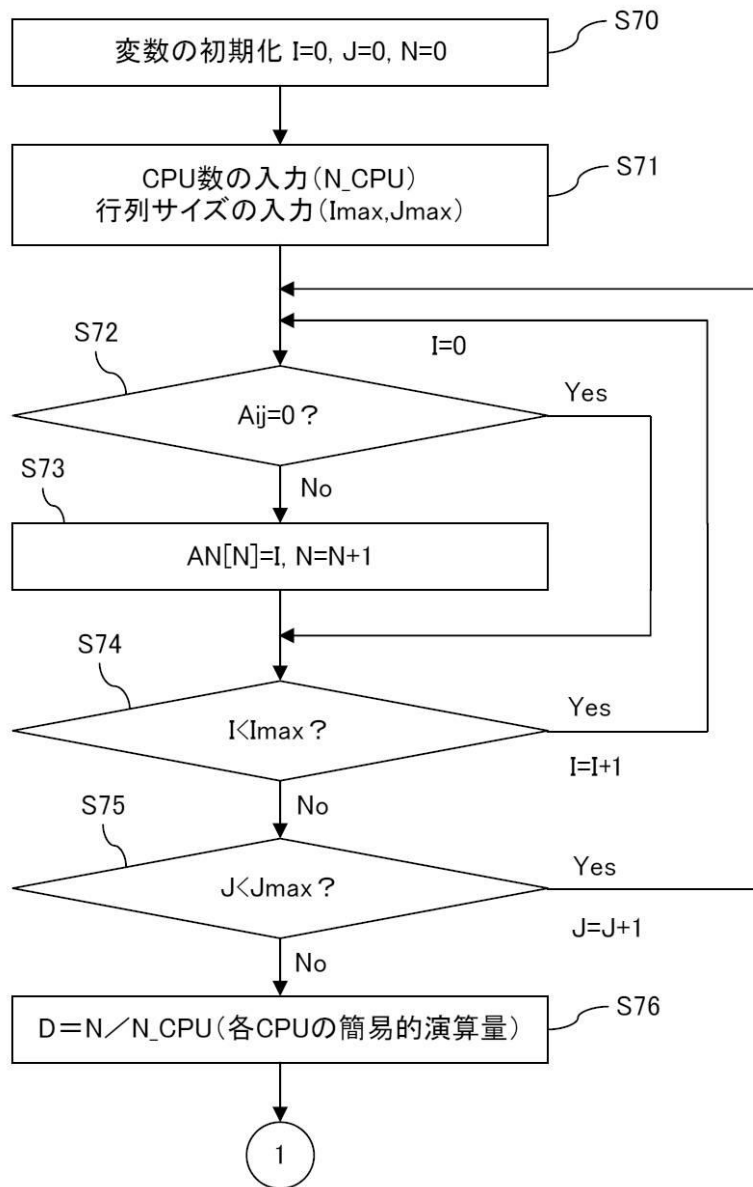
【図 1 3】



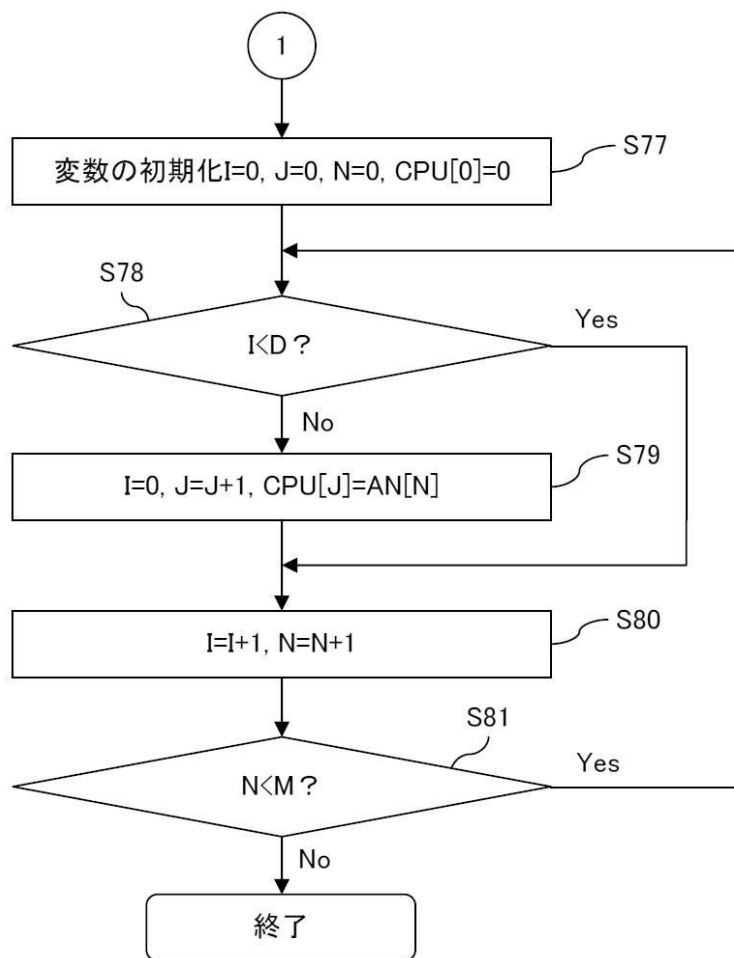
【図 1 8】



【図 19】



【図 20】



フロントページの続き

(56)参考文献 特開昭61-175774 (JP, A)

前川 仁孝 Yoshitaka Maekawa, 並列・分散処理 有限要素・境界要素併用法の並列処理手法, 電気学会研究会資料, 日本, 社団法人電気学会, 1996年 9月11日, 第96-23/30巻, 第41-50頁

寒川 光 Hikaru Samukawa, 分散メモリ型並列計算機のための多重スカイライン法 Multiple Skyline Method for Distributed Memory Parallel Computers, 情報処理学会研究報告 Vol. 96 No. 97 IPSJ SIG Notes, 日本, 社団法人情報処理学会 Information Processing Society of Japan, 1996年10月 4日, 第96巻 第97号, 第25-30頁

(58)調査した分野(Int.Cl., DB名)

G06F 17/12

G06F 17/16