

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4576536号
(P4576536)

(45) 発行日 平成22年11月10日(2010.11.10)

(24) 登録日 平成22年9月3日(2010.9.3)

(51) Int.Cl. F 1
G 0 6 F 11/36 (2006.01) G 0 6 F 9/06 6 2 O R

請求項の数 8 (全 33 頁)

<p>(21) 出願番号 特願2006-175756 (P2006-175756) (22) 出願日 平成18年6月26日 (2006.6.26) (65) 公開番号 特開2008-4043 (P2008-4043A) (43) 公開日 平成20年1月10日 (2008.1.10) 審査請求日 平成19年3月5日 (2007.3.5)</p>	<p>(73) 特許権者 504171134 国立大学法人 筑波大学 茨城県つくば市天王台一丁目1番1 (74) 代理人 100074631 弁理士 高田 幸彦 (72) 発明者 伊藤 祥司 茨城県つくば市天王台一丁目1番1 国立大学法人 筑波 大学内 審査官 林 毅</p>
--	---

最終頁に続く

(54) 【発明の名称】 収束求解アルゴリズムに対する性能比較表示装置及び性能表示比較システム

(57) 【特許請求の範囲】

【請求項1】

収束求解する収束求解アルゴリズムの性能を表示するものであって、
該収束求解アルゴリズムが解くべき問題、解法及び解法に併用される収束性向上のための前処理、係数行列に施すスケールリング、係数行列や行や列の順番を入れ替えるオーダリング及び変換公式のいずれかの併用される技法のメニュー項目を配列して格納する記憶手段、
収束へのCPU時間及び収束までの演算反復回数である収束所要回数のいずれか又は全部の表示項目及び解くべき問題、解法及び解法に併用される技法の各配列されたメニュー項目を表示する表示手段と、前記表示項目からCPU時間及び収束所要回数のいずれか又は全部の項目、及び各配列されたメニュー項目から選択された、解くべき問題、解法及び解法に併用される技法のパラメータ項目を表示するパラメータ選択手段を備えた設定画面表示手段、
選択されたパラメータ項目を受け取り、解くべき問題に対する解法と解法に併用された技法の組み合わせを形成し、各組み合わせでのCPU時間及び収束所要回数の取得のいずれか又は全部の演算処理を行い、演算処理された値同士の比較による相対的な判定値の演算処理を行って、データベースに演算結果を格納させる演算処理手段、及び
各組み合わせの相対的な判定値の演算処理結果を受け取り、各組み合わせの該演算処理結果を画面対比表示するものであって、
Y軸に解くべき問題のパラメータ項目が、そしてX軸に解法及び解法に併用された技法

10

20

のそれぞれのパラメータ項目が設定された X - Y 座標軸上で、双方のパラメータ項目で形成された X - Y 座標位置に、解くべき問題に対する CPU 時間又は収束所要回数についての性能比較結果を前記演算処理して求められた相対的な判定値で画面表示する結果画面表示手段、

とからなることを特徴とする収束求解アルゴリズム性能表示装置。

【請求項 2】

請求項 1 において、前記演算処理手段は、前記相対的な判定値の演算処理の前に、取得された CPU 時間又は収束所要回数が設定した真の残差ノルムの基準値の範囲内にあるか、範囲外にあるか、について判定を行うことを特徴とする収束求解アルゴリズム性能比較表示装置。

10

【請求項 3】

請求項 1 または 2 において、前記結果画面表示手段が、画面に、前記 X - Y 座標位置に表示される前記相対判定値の結果が判定値に対応して該位置毎に色分けして表示されることを特徴とする収束求解アルゴリズム性能比較表示装置。

【請求項 4】

請求項 1 から 3 のいずれかにおいて、前記結果画面表示手段が、画面に CPU 時間及び収束所要回数の組み合わせで、前記演算処理して求められた相対的な判定値で表示することを特徴とする収束求解アルゴリズム性能比較表示装置。

【請求項 5】

請求項 1 において、前記結果画面表示手段が、画面に解法ごと及び併用する技法ごとにグループ分けして CPU 時間又は収束所要回数を表示することを特徴とする収束求解アルゴリズム性能比較表示装置。

20

【請求項 6】

請求項 1 に記載したと収束求解アルゴリズム性能表示装置、及び前記収束求解アルゴリズム性能表示装置に通信手段を介して接続されたクライアント端末から構成され、

前記収束求解アルゴリズム性能表示装置の設定画面表示手段の画面に表示された選択手段について前記通信手段を介して該クライアント端末が選択操作可能であり、かつ該クライアント端末の結果画面表示手段が、前記収束求解アルゴリズム性能表示装置の結果画面表示手段に表示される CPU 時間又は収束所要回数についての性能比較がされた性能比較結果を表示すること

30

を特徴とする収束求解アルゴリズム性能表示システム。

【請求項 7】

請求項 1 に記載した収束求解アルゴリズム性能表示装置による収束求解アルゴリズム性能表示方法において、

演算処理手段が、選択されたパラメータ項目を受け取り、解くべき問題に対する解法と解法に併用された技法の組合せを形成し、各組み合わせの CPU 時間及び収束所要回数の取得のいずれか又は全部の演算処理を行い、演算処理された値同士の比較による相対的な判定値の演算処理を行って、データベースに演算結果を格納させ、

40

結果画面表示手段が、各組み合わせでの相対的な判定値の演算処理結果を受け取り、各組み合わせの該演算処理表示するものであって、

Y 軸に解くべき問題のパラメータ項目が、そして X 軸に解法及び解法に併用された技法のそれぞれのパラメータ項目が設定された X - Y 座標軸上で、双方のパラメータ項目で形成された X - Y 座標位置に、解くべき問題に対する CPU 時間又は収束所要回数についての性能比較結果を前記演算処理して求められた相対的な判定値で画面表示すること

を特徴とする収束求解アルゴリズム性能比較表示方法。

【請求項 8】

請求項 7 において、前記演算処理手段が、前記相対的な判定値の演算処理、の前に、取得された CPU 時間又は収束所要回数について設定した真の残差ノルムの基準値の範囲内にあるか、範囲外にあるか、について判定を行うことを特徴とする収束求解アルゴリズム

50

性能比較表示方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、収束求解アルゴリズム性能比較表示装置およびこの収束求解アルゴリズム性能比較表示装置、性能表示比較システム及び収束求解アルゴリズム性能比較表示方法に関する。

【背景技術】

【0002】

自然現象や工学現象の解析では収束求解アルゴリズムを使用した数値シミュレーションが行われることが盛んである。

【0003】

収束求解アルゴリズムに関する特許文献として、例えば、次に示すものが知られている。

【0004】

特許文献1には、問題定義入力装置により取り込まれた定義情報、プログラム様式情報から構文解析装置が構文解析情報を作成し、アルゴリズム格納装置は、収束求解アルゴリズムの擬似コードの順序列によって表現されるアルゴリズム情報を格納し、計算式生成装置は、構文解析情報から収束求解アルゴリズムに必要な計算式情報を生成することが記載されている。

【0005】

特許文献2には、子モデルパラメータファイルから子モデルパラメータを入力して、入力した子モデルパラメータに基づいてK個のモデル評価値を求めて記憶部の評価値ファイルに記憶する評価値計算部を備えた遺伝的アルゴリズムマシンの適応評価器が記載されている。

【0006】

特許文献3には、入力データに含まれるパラメータを用いて、入力データの特性を評価する多角的アルゴリズム運用システムが記載されている。

【0007】

【特許文献1】特開2004-86760号公報

【特許文献2】特開2006-12114号公報

【特許文献3】特開2002-150260号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

前述したように、自然現象や工学現象の解析では、数値アルゴリズムを使用した数値シミュレーションが行われることがしばしばあり、それらを記述する方程式は線形方程式の場合、大規模な連立1次方程式

$$A x = b$$

の求解に帰着される。

【0009】

また、固有値問題の場合には次の方程式

$$A x = B x$$

の求解に帰着される。

【0010】

また、代数方程式を使用する場合、

$$f(x) = 0$$

の求解に帰着される、他の問題についても上記同様にそれぞれの方程式の求解に帰着される。

【0011】

10

20

30

40

50

このような求解において、数値シミュレーションに要する計算時間の大半がこの計算に費やされるため、速く、正確に解くことが重要である。

問題を解法するために多種多様の収束求解アルゴリズムの採用が可能であり、一体どの収束求解アルゴリズムを採用すれば速く、正確に解くことになるのか不明である。

係数行列 A が対称正定値であれば、迷わず C G 法や C h o l e s k y 法が選択されるが、係数行列が非対称のような場合にはどの収束求解アルゴリズムを選択すれば速く、正確に解くことになるのかが判らない。

【 0 0 1 2 】

収束求解アルゴリズムの研究者（アルゴリズム開発者）は、数学的な観点から新しい収束求解アルゴリズムを提案し、収束特性、高精度を得ようとし、数値シミュレーションの研究者（アルゴリズムユーザ）は、実際に数値シミュレーションを行い、解くべき問題に応じてオリジナルのプログラムや収束求解アルゴリズム作成し、いかにして問題を解くか、を研究し、速い方が良く、許容精度内であれば十分とする。

【 0 0 1 3 】

このような、アルゴリズムユーザのニーズに応じて問題を解決するに当って、許容精度内で、速く求解する収束求解アルゴリズムの採用を推奨する方法が求められる。

【 0 0 1 4 】

本発明者は先に、かかる点に鑑みて、アルゴリズムユーザのニーズに応じて多種多様に存在する収束求解アルゴリズムを使用して問題を求解するに当って、許容精度内で、速く求解することのできる収束求解アルゴリズム性能比較表示装置あるいはこの収束求解アルゴリズム性能比較表示装置を用いた収束求解アルゴリズム性能比較表示方法を提供する特許出願を行った。

【 0 0 1 5 】

本発明は、更に上述の収束求解アルゴリズムの性能比較表示に当って対象とする収束求解アルゴリズムについて使用の目的に対する性能比較を行ってその結果を表示することのできる収束求解アルゴリズム性能比較表示装置あるいは／およびこの収束求解アルゴリズム性能比較表示装置を用いた収束求解アルゴリズム性能比較表示方法を提供することを目的とする。

[課題を解決するための手段]

【 課題を解決するための手段 】

【 0 0 1 6 】

本発明は、収束求解する収束求解アルゴリズムの性能を表示するものであって、該収束求解アルゴリズムが解くべき問題、解法及び解法に併用される収束性向上のための前処理、係数行列に施すスケールリング、係数行列や行や列の順番を入れ替えるオーダリング及び変換公式のいずれかの併用される技法のメニュー項目を配列して格納する記憶手段、

収束への C P U 時間及び収束までの演算反復回数である収束所要回数のいずれか又は全部の表示項目及び解くべき問題、解法及び解法に併用される技法の各配列されたメニュー項目を表示する表示手段と、前記表示項目から C P U 時間及び収束所要回数のいずれか又は全部の項目、及び各配列されたメニュー項目から選択された、解くべき問題、解法及び解法に併用される技法のパラメータ項目を表示するパラメータ選択手段を備えた設定画面表示手段、

選択されたパラメータ項目を受け取り、解くべき問題に対する解法と解法に併用された技法の組み合わせを形成し、各組み合わせでの C P U 時間及び収束所要回数の取得のいずれか又は全部の演算処理を行い、演算処理された値同士の比較による相対的な判定値の演算処理を行って、データベースに演算結果を格納させる演算処理手段、及び

各組み合わせの相対的な判定値の演算処理結果を受け取り、各組み合わせの該演算処理結果を画面对比表示するものであって、

Y 軸に解くべき問題のパラメータ項目が、そして X 軸に解法及び解法に併用された技法のそれぞれのパラメータ項目が設定された X - Y 座標軸上で、双方のパラメータ項目で形

10

20

30

40

50

成された X - Y 座標位置に、解くべき問題に対する CPU 時間又は収束所要回数についての性能比較結果を前記演算処理して求められた相対的な判定値で画面表示する結果画面表示手段、

とからなることを特徴とする収束求解アルゴリズム性能表示装置を提供する。

【 0 0 1 7 】

また、本発明は前記演算処理手段は、前記相対的な判定値の演算処理の前に、取得された CPU 時間又は収束所要回数が設定した真の残差ノルムの基準値の範囲内にあるか、範囲外にあるか、について判定を行うことを特徴とする収束求解アルゴリズム性能比較表示装置を提供する。

【 0 0 1 8 】

また、本発明は前記結果画面表示手段が、画面に、前記 X - Y 座標位置に表示される前記相対判定値の結果が判定値に対応して該位置毎に色分けして表示されることを特徴とする収束求解アルゴリズム性能比較表示装置を提供する。

【 0 0 1 9 】

また、本発明は前記結果画面表示手段が、画面に CPU 時間及び収束所要回数の組み合わせで、前記演算処理して求められた相対的な判定値で表示することを特徴とする収束求解アルゴリズム性能比較表示装置を提供する。

【 0 0 2 0 】

また、本発明は前記結果画面表示手段が、画面に解法ごと及び併用する技法ごとにグループ分けして CPU 時間又は収束所要回数を表示することを特徴とする収束求解アルゴリズム性能比較表示装置を提供する。

【 0 0 2 1 】

また、本発明は前記収束求解アルゴリズム性能表示装置、及び前記収束求解アルゴリズム性能表示装置に通信手段を介して接続されたクライアント端末から構成され、

前記収束求解アルゴリズム性能表示装置の設定画面表示手段の画面に表示された選択手段について前記通信手段を介して該クライアント端末が選択操作可能であり、かつ該クライアント端末の結果画面表示手段が、前記収束求解アルゴリズム性能表示装置の結果画面表示手段に表示される CPU 時間又は収束所要回数についての性能比較がされた性能比較結果を表示すること

を特徴とする収束求解アルゴリズム性能表示システムを提供する。

【 0 0 2 2 】

また、本発明は前記収束求解アルゴリズム性能表示装置による収束求解アルゴリズム性能表示方法において、

演算処理手段が、選択されたパラメータ項目を受け取り、解くべき問題に対する解法と解法に併用された技法の組合せを形成し、各組み合わせの CPU 時間及び収束所要回数の取得のいずれか又は全部の演算処理を行い、演算処理された値同士の比較による相対的な判定値の演算処理を行って、データベースに演算結果を格納させ、

結果画面表示手段が、各組み合わせでの相対的な判定値の演算処理結果を受け取り、各組み合わせの該演算処理表示するものであって、

Y 軸に解くべき問題のパラメータ項目が、そして X 軸に解法及び解法に併用された技法のそれぞれのパラメータ項目が設定された X - Y 座標軸上で、双方のパラメータ項目で形成された X - Y 座標位置に、解くべき問題に対する CPU 時間又は収束所要回数についての性能比較結果を前記演算処理して求められた相対的な判定値で画面表示すること

を特徴とする収束求解アルゴリズム性能比較表示方法を提供する。

【 0 0 2 3 】

また、本発明は前記演算処理手段が、前記相対的な判定値の演算処理、の前に、取得された CPU 時間又は収束所要回数について設定した真の残差ノルムの基準値の範囲内にあるか、範囲外にあるか、について判定を行うことを特徴とする収束求解アルゴリズム性能比較表示方法を提供する。

10

20

30

40

50

【発明の効果】

【0024】

本発明は、上述した記憶手段、演算処理手段、データベース、設定画面表示手段、データ取得処理手段および結果画面表示手段によって構成されることにより、アルゴリズムユーザのニーズに応じて多種多様に存在する収束求解アルゴリズムを使用して問題を求解するに当たって、許容精度内で、対象の収束求解アルゴリズムについて性能比較し、相対判定を行うことのできる収束求解アルゴリズム性能比較表示装置、あるいはこの収束求解アルゴリズム性能比較表示装置を用いた収束求解アルゴリズム性能比較表示方法を提供することができる。

【発明を実施するための最良の形態】

10

【0025】

以下、本発明の実施例を図面に基づいて説明する。

【実施例】

【0026】

図1は、本発明の実施例である収束求解アルゴリズム性能比較表示装置の構成を示すブロック図である。

【0027】

図1において、収束求解アルゴリズム性能比較表示装置100は、記憶手段11、演算処理手段(1)12、データベース13、演算処理手段(2)14、表示結果手段15からなり、記憶手段11および演算処理手段(1)12は提供側計算機システム1を構成し、演算処理手段(2)14および結果表示手段15は提供側計算機システム2(数字30で示す)を構成し、データベース13はこれらの提供側計算機システム1および2に接続され、データの授受を行う。収束求解アルゴリズム性能比較表示装置100はクライアント端末200に接続され、データの授受がなされる。

20

【0028】

記憶手段11は複数のパラメータ21を備える。これらのパラメータは収束求解アルゴリズム20と関連づけられる。収束求解アルゴリズム20は、多数の項目、例えば固有問題($Ax = Bx$)の求解用の線形方程式($Ax = b$)の求解用の、あるいは代数方程式($f(x) = 0$)の求解用の収束求解アルゴリズム項目からなる、上記以外の数値計アルゴリズム項目を含む。各項目は、各パラメータと関連づけて記憶される。パラメータ21は、解くべき問題22、解法23および解法と併用する技法24によって構成され表現される。これらの三要素によって構成され、表現されることは公知の事項である。例えば、線形方程式の場合、解くべき問題22の項目としては、例えば行列、右辺項、初値、他で構成され、表現され、解法23の項目としては、例えば直接法、定常反復法、定常反復法、他で構成され、表現され、解法と併用する技法24の項目としては、例え前処理、スケールリング、オーダリング、他で構成され、表現される。固有値問題の場合、解くべき問題22の項目としては、例えば行列A、行列B、初期値、他から構成され、表現され、解法23の項目としては、べき乗法系統、QR法、反復法系統、他で構成され、表現され、解法と併用する技法24の項目としては、変換公式、シフト、他で構成され、表現される、代数方程式の場合も同様である。

30

40

【0029】

以下、主に線形方程式($Ax = b$)に例をとって説明することにする。

「数値計算によって解くべき問題」とは、例えば、線形方程式

$$Ax = b, \quad (A: \text{係数行列}, x: \text{解ベクトル}, b: \text{右辺項ベクトル})$$

などを指す。これを解くための解法と、その求解効率を向上させるための技法(前処理やスケールリングなど)の例として、「前処理付き共役勾配法」の収束求解アルゴリズムを図4に示す。(共役勾配法はCG法とも呼ばれ、本実施例システムの各説明や計算機システム内でも「CG」と表示されている。)

ここで、 x_k が解ベクトルであり、 r_k が残差ベクトルである。これら以外の p_k 、 k 、 k は、アルゴリズムを構成する補助的なベクトルおよびスカラーである。

50

【 0 0 3 0 】

上記の「begin」から「end」までの間の5つの式を、残差ベクトルのノルム（上記の $\|r_k\|$ の値）が許容値に収束するまで繰り返し実行する。枠で囲った部分が「前処理」と呼ばれる演算で、この行列Kの作り方（前処理方法）次第で、収束状況が変わる。解法がBiCGStabなど他のものになると、反復計算させる式の構成が変わる。K = I（Iは単位行列）に相当する場合は、一般に「前処理なし」の収束求解アルゴリズムである。

【 0 0 3 1 】

収束判定では、 $\|r_k\|$ の値について評価するが、典型的な判定方法は $\|r_k\| \|b\|$ による。

ここで、 $\|b\|$ が許容値であり（本システムでは10のマイナス12乗である）、bは線形方程式の右辺項ベクトルである。収束したと判定されたときまでに要した反復回数が「収束所要回数」である。

【 0 0 3 2 】

CPU時間は、上記の収束求解アルゴリズムが開始される前の「前処理行列K生成に関する計算時間」と「反復求解で収束求解アルゴリズム中の残差が収束するまでの時間」との合計時間である。

【 0 0 3 3 】

収束グラフは、残差ベクトルのノルム $\|r_k\|$ を各反復ごとにデータファイル（図6, 7の拡張子「.rsd」のファイル）に格納しその \log_{10} （常用対数）を取ったものをプロットして作成する。このグラフから、元の線形方程式に対して反復解法に基づく収束求解アルゴリズムの効果などの様子が確認できる。ただし、ここでの残差ベクトルとは、あくまでも収束求解アルゴリズム中のデータであり、計算過程における丸め誤差の混入などにより、一見、収束求解アルゴリズム中の残差では収束したように見えても、実際には解が得られていない場合もある、そのようなときには、真の残差ノルムで評価する。

【 0 0 3 4 】

真の残差ノルムとは、収束求解アルゴリズムにより得られた数値解

$$(\hat{x})$$

を用いて、

$$\|b - A\hat{x}\|$$

を評価したものである。図20の「Data Table」内の「Res. norm (b - Ax)」は、この情報をあらわしている。

【 0 0 3 5 】

1) 収束求解アルゴリズム

数値計算も更に細分化され、以下のような学問上の分野がある。

「線形方程式、固有値問題、特異値分解(SVD)、代数方程式、数値積分、関数近似、その他」

本実施例システムでは、これらの数値計算分野を対象とするが、説明書類では、具体例として線形方程式に関する収束求解アルゴリズムの例を取り上げた。他の分野についても同様の考え方が応用される。

【 0 0 3 6 】

2) 解くべき問題

線形方程式($Ax = b$)は、係数行列と呼ばれる行列A (Matrix)と右辺項と呼ばれるベクトルbを用いて構成され、最終的に数値解のベクトルxを求める。

本実施例システムでは、行列はテスト用行列を集めたWebサイトのデータを使用し、最終的には100種類程度の行列を用意する予定である。これらのデータはテキスト形式のファイルである。

10

20

30

40

50

Matrix Market <http://math.nist.gov/MatrixMarket/>
Sparse Matrix Collection

<http://www.cise.ufl.edu/research/sparse/matrices/>

【0037】

具体的に行列名の一部を紹介すると、以下のような名称のものが存在する。

1138__bus、494__bus、662__bus、685__bus、add20、
add32、bcssstk14、bcssstk15、bcssstk16、bcssstm2
6、gr__30__30、memplus、nos1、nos2、nos3、nos4、n
os5、nos6、nos7、s1rmq4m1、s1rmt3m1、s2rmq4m1
、s2rmt3m1

10

(全て、Matrix Marketで提供されている行列)

さらに、本実施例システムの現状では、右辺項は実行プログラムの中で作成している。具体的には、解ベクトルxのデータをあらかじめ適当な値で用意し、これを係数行列Aにかけることで、右辺項ベクトルbが作成される(この演算は、次項で説明するLisでサポートされている機能である)。

【0038】

3) 解法

線形方程式を解くための代表的な解法には、以下のようなものがある。

直接解法：LU分解に基づくガウス消去法、コレスキー分解に基づくガウス消去法、他
定常反復解法：Jacobi法、Gauss-Seidel法、SOR法、他

20

非定常反復解法：CG法、BiCG法、CGS法、BiCGStab法、BiCGStab(l)法、GPBiCG法、Orthomin法、GMRES法、TFQMR法、他
本実施例システムの現状では、定常反復解法と非定常反復解法で解法プログラムが構成されており、直接解法も取り入れることができる。解法プログラムは、以下のサイトで用意されているフリーライブラリのLis (Library of Iterative Solvers for Linear Systems)を用い、提供側計算機システム1では、実行形式プログラムとして用意されている。

Lis <http://ssi.is.s.u-tokyo.ac.jp/lis/>

【0039】

Lisの仕様として、解法(solver)には下記のとおりsolverIDが付けられている。〔 〕内の2桁の数値がsolverIDである。

30

CG[01]、BiCG[02]、CGS[03]、BiCGStab[04]、BiCGStab(l=2)[05]、GPBiCG[06]、TFQMR[07]、Orthomin[08]、GMRES[09]、Jacobi[10]、Gauss-Seidel[11]、SOR[12]

現在、BiCGStab(l)法のパラメータlの値は、l=2としている。

【0040】

4) 解法と併用する技法

線形方程式用の解法と併用する技法(技法)には、以下のようなものがある。

前処理：反復解法の数値解への収束性に影響を及ぼす技法であり、収束性向上を意図して用いる。

40

スケールリング：収束求解アルゴリズムが安定して解を求められるよう、係数行列に対して施す技法。

オーダリング：主に求解効率の向上を意図して、係数行列の行や列の順番を入れ替える技法。

変換公式：求解効率の向上や方程式を解き易い形式に変形するために、主に係数行列を同値な式に変換すること。

その他

本実施例システムの現状では、前処理を実装している。これ以外の技法も今後取り入れる予定である。前処理のプログラムは、解法と同様Lisを用いて、提供側計算機システ

50

ム 1 では、実行形式プログラムとして用意されている。

【 0 0 4 1 】

L i s の仕様として、前処理 (p r e c o n d i t i o n e r) には下記のとおり p r e c o n d I D が付けられている。 [] 内の 2 桁の数値が p r e c o n d I D である。 n o n e [0 0] (前処理なし)、P J a c o b i [0 1]、I L U [0 2]、S S O R [0 3]、H y b r i d [0 4]、I + S [0 5]、S A I N V [0 6]、S A A M G [0 7]

【 0 0 4 2 】

このように、記憶手段 1 1 が収束求解アルゴリズム毎に、解くべき問題、解法および解法と併用する技法の各項目について設定したパラメータ 2 1、各パラメータについて設定され、収束グラフ、CPU 時間および収束所要回数の動作仕様を決定する動作仕様ファクターおよび各項目から選択された動作仕様ファクターの組み合わせに関連づけられた収束グラフ、CPU 時間および収束所要回数の演算を行うコンピュータプログラムを格納して構成される。

10

【 0 0 4 3 】

演算処理手段 (1) 1 2 は、入力データファイル 2 5 および実行形式プログラムファイル 2 6 を備えて構成される。実行形式プログラムファイル 2 6 は解法および解法と併用する技法を格納したファイルであり、入力データファイル 2 5 は解くべき問題のデータを格納したファイルであり、これらのファイル自体は公知の事項である。

【 0 0 4 4 】

図 2 は、収束求解アルゴリズム性能比較表示装置 1 0 0 とクライアント端末 2 0 0 の一部詳細を示すブロック図である。図 2 において、パラメータ 5 3 におけるパラメータ項目選択はパラメータ項目選択 (方法 1) 5 3 A、パラメータ項目選択 (方法 2) 5 3 B およびパラメータ項目選択 (方法 3) 5 3 C を想定した。それ以上にパラメータ項目が設定されるようになってよい。

20

【 0 0 4 5 】

結果画面表示手段 4 には、パラメータ項目選択 (方法 1) 5 3 A、パラメータ項目選択 (方法 2) 5 3 B およびパラメータ項目選択 (方法 3) 5 3 C に対応して結果画面表示 (方法 1) 5 4 A、結果画面表示 (方法 2) 5 4 B および結果画面表示 (方法 3) 5 4 C を行うことができる。

30

【 0 0 4 6 】

パラメータ項目選択 (方法 1) 5 3 A では、例えば

- 1) 表示項目の設定
- 2) 解くべき問題
- 3) 解法
- 4) 解法と併用する技法
- 5) 真の残差ノルムに対する許容値

が選択、設定される。

【 0 0 4 7 】

これに対応して結果画面表示手段 5 4 の内容を示す結果画面表示 (方法 1) 5 4 A では、例えば

40

- 1) 収束グラフ
- 2) CPU 時間
- 3) 収束所要回数
- 4) 真の残差ノルム値

が表示される。

【 0 0 4 8 】

パラメータ項目選択 (方法 2) 5 3 B では、例えば

- 1) 解くべき問題
- 2) 真の残差ノルムに対する許容値

50

が選択、設定される。

【 0 0 4 9 】

これに対応して結果画面表示（方法 2）5 4 B では、例えば

- 1) 収束求解アルゴリズム
- 2) 比較項目（CPU 時間、収束所要回数）
- 3) 真の残差ノルム
- 4) 3) の相対判定結果

が表示される。

【 0 0 5 0 】

パラメータ項目選択（方法 3）5 3 C では、例えば

- 1) グループ分けの項目（解法、併用する技法）
- 2) 比較する項目
- 3) 真のノルムに対する許容値
- 4) 解くべき問題の性質

が選択、設定される。

【 0 0 5 1 】

これに対応して結果画面表示（方法 3）では、例えば

- 1) 収束求解アルゴリズム
- 2) 解くべき問題
- 3) 比較項目の相対判定結果
- 4) 設定したパラメータ値

が表示される。

【 0 0 5 2 】

データベース 1 3 は、データベース（1）1 3 A とデータベース（2）1 3 B とからなり、データベース（1）1 3 A はパラメータとして、

- 1) 収束グラフ
- 2) CPU 時間
- 3) 収束所要回数
- 4) 真の残差ノルム値

をそれぞれ関連づけて格納している。

【 0 0 5 3 】

データベース（2）1 3 B はパラメータとして、

- 1) 問題の性質（1）
- 2) 問題の性質（2）
- 3) 問題の性質（3）

をそれぞれ関連づけて格納している。

【 0 0 5 4 】

提供側計算機システム 2 は、演算処理手段（2 A）1 4 A、演算処理手段（2 B）1 4 B および結果画面表示手段 5 4 A からなる。

【 0 0 5 5 】

演算処理手段（2 B）1 4 A は、比較項目に対する判定、その結果に基づく並び替えおよびその他の計算を行う。

データベース 1 の作成は、収束求解アルゴリズムの性能テスト問題を格納している Web サイト（例えば、MatrixMarket [<http://math.nist.gov/MatrixMarket/>], University of Florida Sparse Matrix Collection, [<http://www.cise.ufl.edu/research/sparse/matrices/>], Sparse matrix test problems <http://users.tkk.fi/~kouhia/sparse.html>] など、本例の装置では、Matrix Market のデータを用いている。) からテスト問題のデータファイルと一緒にテスト問題に関する性質の情報（行列サイズ、代表的特徴や、固有値などのスペクトル特性、他）を転送している。

【 0 0 5 6 】

10

20

30

40

50

前者については、提供側計算機システム1の「入力データファイル」として使用している。後者の情報については、UNIX（登録商標）の基本ツールであるawkを利用してテキスト形式のデータファイルに整形したものであり、これをデータベース(2)13Bと呼んでいる。

【0057】

データベース(2)13Bについて：

データベース(2)13Bの「解くべき問題」は、線形方程式の場合では、「行列」を指している。ここで、問題の性質1や性質2とは、行列の「大きさ」、「行列の代表的特徴（対称、非対称や正定値性など）」や「条件数」などを指している。

データベース(2)13Bでは、解くべき問題の名称（行列名）をキーにして、その行列の性質を関連づけている。

10

【0058】

演算処理手段(2B)14Bについて：

図8に示す入力画面にて設定された「収束における真の残差ノルムの許容値」はパラメータ「criterion」に格納され、演算処理手段(2B)14Bで受け取られる。

真の残差ノルムを比較判定する対象となるアルゴリズムは、図2の方法1の場合には、選択されたアルゴリズムが対象であり、方法2と3の場合には、全アルゴリズム(12種類の解法と8種類の前処理全ての組合せ)が対象である。

【0059】

1)方法1の場合：

20

クライアント端末（表示装置）から渡されたパラメータを、フリーソフトウェアphpの機能を用いて分析する。

図8の入力画面にて設定された全パラメータとそれらに格納された値は、パラメータ受渡しハンドラ - 配列に格納され（phpの動作仕様であり、公知）提供側計算機システム2では、あらためてハンドラを一配列（ここでは「POST」という配列名で説明する）から各パラメータと変数値を受け取る。

表示形式は、パラメータ「SEL」に値が格納される（SEL=POST[SEL]）。

行列名は、パラメータ「matrix」に行列名が格納される（matrix=POST[matrix]）

解法名は、パラメータ「solverID」（配列形式）に解法のIDが格納される（同様）。

技法名は、パラメータ「precondID」（配列形式）に技法のIDが格納される（同様）

30

。許容値は、パラメータ「criterion」に許容値が格納される（同様）。

【0060】

提供側計算機システム2内部での制御の順番は、

- (1)パラメータ「matrix」の値の受け取り
 - (2)パラメータ「solverID」の値の受け取りと解法の個数（配列の要素数）の分析
 - (3)パラメータ「precondID」の値の受け取りと解法の個数（配列の要素数）の分析
 - (4)パラメータ「criterion」の値の受け取り
 - (5)パラメータ「SEL」の値の受け取りとその値に応じた制御
- である。

40

【0061】

このとき、真の残差ノルムがcriterion以下ならば収束したと判定し、criterionよりも大きい値ならば見かけ上の収束と判定して、以下のように対応付ける。

状況	パラメータ「stat」に設定する値	パラメータ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col（黄色）
反復回数の上限	mat.itr.	war_col（赤色）
ブレークダウン	brk.dwn	war_col（赤色）

データテーブル表示のhtmlを作成するとき、上記の状況に応じて、対応する行のバックグラウンド色（htmlの機能で着色）を「col_stat」の色で表示する。表示出力例は、図2

50

2 のとおりである。

【 0 0 6 2 】

2) 方法 2 の場合 :

パラメータ入力の方法は、後述する図 9 あるいは図 1 0 (行列名を指定する) のとおりである。選択された行列 1 つに対する、全アルゴリズム (1 2 種類の解法と 8 種類の前処理全ての組合せ) のデータについて処理が行われる。解法 ID、前処理 ID とアルゴリズム名の対応は、「loop_solpre」で行われる。

【 0 0 6 3 】

「.log」拡張子のファイル中のデータで「真の残差ノルム値」と「criterion」の値とを比較判定する。このとき、真の残差ノルムがcriterion以下ならば収束したと判定し、criterionよりも大きい値ならば見かけ上の収束と判定して、以下のように対応付ける。

状況	パラメータ「stat」に設定する値	パラメータ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col (黄色)
反復回数の上限	mat.itr.	war_col (赤色)
ブレークダウン	brk.dwn	war_col (赤色)

表示するデータ一式 (逐次番号、アルゴリズム、収束所要回数、CPU時間、真の残差のノルム値、求解状況の情報、col_stat) を配列 (INFO_dat) に格納する。

【 0 0 6 4 】

配列 INFO_dat に対して比較項目 (sort_key の値) に応じ、CPU時間が収束所要回数のどちらかで昇順にソート (php の関数の使用) する。

ソート後、col_stat の値が NULL のレコード (収束したもの) を対象にして最初のレコードの比較項目の値をパラメータ「TheFastest」に代入し、以後のレコードに対して、比較項目の相対評価の指標 (現状では「TheFastest ÷ 比較項目の値」) を算出し、少数第一位までをパラメータ「Rat」に代入する。

データテーブル表示のhtmlを作成するとき、上記の求解状況に応じて、対応する行のバックグラウンド色 (html の機能で着色) を「col_stat」の色で表示する。表示出力例は、図 2 3 のとおりである。

収束したものについては、gnuplot を用いて比較項目のデータの棒グラフを描く。

【 0 0 6 5 】

3) 方法 3 の場合 :

パラメータの入力方法は、図 1 0 (行列名を指定しない場合) のとおりとしている。

データベース (1) 1 3 A から行列名一覧を獲得する。各々の行列に対して全アルゴリズム (1 2 種類の解法と 8 種類の前処理全ての組合せ) のデータについて、以下の処理が行われる。

「.log」拡張子のファイルの中から、表示するデータ一式 (レコード番号、アルゴリズム名、収束所要回数、CPU時間、真の残差のノルム、求解状況の情報、col_stat) を配列 (INFO_dat) に格納する。

配列 INFO_dat をコピーして、INFO_dat0 を作成する。

「INFO_dat」を、比較項目 (sort_key の値) に応じ、CPU時間が収束所要回数のどちらかで昇順にソート (php の関数を使用) する。そして、昇順のレコードの先頭から、INFO_dat [i] [alg] (求解状況、i はレコードの番号) = conv. の条件を満たす最初のレコードの比較項目の値を [TheFastest] に代入する。

【 0 0 6 6 】

図 3 に演算処理手段 (1) 1 2 の詳細をフローチャートで示す。図 3 において、実行形式プログラム 2 6 は、入力データファイル 2 5 の行列 A , B , C ... (後述する 1 1 3 8 __ bus など) を使用して、解法 ID 0 1 , 0 2 ... 1 2 と解法と併用する技法の ID 0 0 , 0 1 , ... , 0 7 を逐次選択し実行する。

各々の行列に対して解法群 (0 1 ~ 1 2) と技法 (0 0 ~ 0 7) の全組み合わせ、すな

10

20

30

40

50

わち前述のパラメータについて設定され、収束グラフ、CPU時間および収束所要回数の動作仕様を決定する動作仕様ファクター（因子）の組み合わせを適用して求解し、データを生成する。

【0067】

4) 入力データファイル

「解くべき問題」の実際のファイルである。

ここで用意されたファイルが、6)項の実行形式プログラムに対する入力データとなる。

【0068】

5) 実行形式プログラムファイル

コンピュータ言語で表現された解法および解法と併用する技法を、実行形式のファイルとして用意する。

Lisの場合、C言語とFortran90のプログラムから構成され、これらに線形方程式求解用のmainプログラムを作成し、コンパイルして実行形式プログラム（ロードモジュール）のファイルを作成する。ここで、コンパイラは商用のものやフリーのものをを用いることができる。

【0069】

このように、演算処理手段である演算処理手段(1)12はコンピュータプログラムを使用して、各項目から選択された前記動作仕様ファクターの組み合わせに関連づけられた収束グラフ、CPU時間および収束所要回数について演算処理を行う。

【0070】

図3にて、提供側計算機システム1(12)で生成された諸計算結果のファイルがデータベース(1)13Aに格納される。また、収束求解アルゴリズムの性能テスト問題を格納しているサイト41から収束求解アルゴリズムの性能を提供側計算機システム1(12)に送信すると、提供側計算機システム1(12)はファイル転送および情報の整理42を行って整理された収束求解アルゴリズムの性能テスト問題をデータベース(2)13Bに格納する。また、クライアント端末(表示装置)200から検索実行のリクエストがあったときには、提供側計算機システム2(30)を経由してデータベース13のファイル検索が行われ、その中から必要なデータが提供側計算機システム2に渡される。

【0071】

本実施例システムの現状では、このデータベース13は、提供側計算機システム2の中に組み込まれているが、今後はデータベースのみを単独のシステムとしてもよい。

【0072】

図5は、以上説明した本発明の実施例についてのフローチャートを説明する図である。図5において、収束求解アルゴリズムについて、画面上に、比較項目を含むパラメータの設定を行う(S1)。画面上に、真の残差ノルム基準値の設定を行う(S2)。真の残差ノルム値が真の残差ノルム基準値の範囲内にあるかについて判定を行い(S3)、範囲内にある数値アルゴリズムの順に画面上で、数値アルゴリズムのリストの並び替えを行う(S4)。各数値アルゴリズムについて、あるいは上位リストにある収束求解アルゴリズム(これらを対象の収束求解アルゴリズムという)について比較項目についての性能データの出力を行う(S5)。各性能間の性能比較、すなわち対象の収束求解アルゴリズムについての性能の相对比较による相対判定を行うと共に対象の収束求解アルゴリズムについての相对比较値(相対判定値)の算出を行う(S6)。最終的に、相对比较、相对比较値に対応して、対象の収束求解アルゴリズムを画面上で、色分け表示を行う(S7)。図2における方法1では、上記のS1~S3までの処理フローが該当する。方法2では、上記のS1~S6までの処理フローが該当する。方法3では、上記のS1~S7までの処理フロー全てが該当する。

【0073】

図5に示すステップについて以下、詳細に説明する。

データベース13に載せる際の提供状況やデータファイルの内部形式の例について図6

10

20

30

40

50

および図7を参照して説明する。

【0074】

図6にデータベース13Aに載せる際の情報提供状況を示す。上述のように、解くべき問題の設定31、解法の設定32、解法と併用する技法の設定33がなされ、演算処理手段(1)12での演算処理34がなされる。図6に記載された事項をそのまま次に示す。

【0075】

解くべき問題の設定31(内部では行列名で表現。行列名の例:1138__bus)

その他の行列名の一部を具体的に挙げると、以下のような名称のものが存在する。

1138__bus, 494__bus, 662__bus, 685__bus, add20, add32, bcsttk14, bcsttk15, bcsttk16, bcsttm26, gr__30__30, memplus, nos1, nos2, nos3, nos4, nos5, nos6, nos7, s1rmq4m1, s1rmt3m1, s2rmq4m1, s2rmt3m1

(全て、Matrix Marketで提供されている行列)

【0076】

解法の設定32

(内部では解法のIDで表現。解法IDの例:BiCGStabならば04)

本実施例システムでは、解法は12種類としている。

解法と併用する技法の設定33

(内部では技法のIDで表現。技法IDの例:PJacobiならば01)

本実施例システムでは、この技法(前処理)は8種類ある。

演算処理手段(1)での扱い

下記のようなテキストファイルとして格納する。

【0077】

演算処理手段(1)での扱い34

各パラメータへの値の設定

matrix = 1138__bus, solverID = 04, preconditionID = 0

1

検索対象であるファイル名の構成

matrix - solverID preconditionID 0000 . xxx

(具体例:1138__bus - 04010000 . xxx)

ここで、xxxはファイル種別を表す拡張子である。

sol: 解ベクトルを格納したファイル

rsd: 反復ごとの残差ノルム情報を格納したファイル(収束グラフのデータ)

log: CPU時間や収束所用回数などのデータを格納したファイル

0000は予備として用意する。

【0078】

本実施例では解法12種類×技法8種類で、96とおりの組合せができ、各々に対し3種類の拡張子を持ったファイルを構成している。解法・技法、拡張子の数によってファイル数を増減することができる。

図7に各ファイルの内部形成の例(全てテキストデータ)を示す。図7に示すように、動作仕様ファクターに関連して収束所要回数、CPU時間、求解状況、真の残差のノルム値が格納される。

このように、データベース13Aには、演算処理手段の演算結果である、各項目から選択された前記動作仕様ファクターの組み合わせに関連づけて収束グラフ、CPU時間および収束所要回数、求解状況、真の残差のノルム値についてのデータが格納される。

【0079】

次に提供側計算機システム2(30)について説明する。

演算処理手段(2A)14Aは、後述するパラメータの項目設定画面を使用して制御を行い、各パラメータ項目の選択と動作仕様ファクターの設定および検索対象であるファイ

10

20

30

40

50

ル名の構成を行う。すなわち、設定パラメータからファイル名の作成を行う。

後述するクライアント端末 200 (表示装置) から検索実行のリクエストを受けて、データベースのファイル検索を行い、その中から必要なデータを受け取り、クライアント端末 200 (表示装置) に表示できるようにデータを加工する。以下、詳述する。

【0080】

図 8、図 9 はそれぞれ線形方程式アルゴリズム性能評価の情報検索システムを示し、図 8 は図 2 に示すパラメータ項目選択 (方法 1) に基づく入力画面を示し、図 9 は図 2 に示すパラメータ項目選択 (方法 2) に基づく入力画面を示す。線形方程式の場合のパラメータ入力、動作仕様ファクター入力およびデータ検索、計算結果の精度判定、表示依頼の画面 (画面表示装置の設定画面表示手段の画面) を表わしている。画面表示と各種制御は、フリーソフトウェアの php を使用し、画面表示では html の表示仕様にしたがっている。

10

この図 8 では、丸印の項目を選択、あるいは設定してる。

【0081】

2 重丸の項目が収束における真の残差ノルムの許容値を設定する欄である。ここの値の設定次第で、図 22 の結果出力の内容は異なってくる。

この 2 重丸の項目は、「真の残差のノルム」に対する許容値を指す。アルゴリズム中で用いられている残差のノルムが収束していても、そのときに得られた解を元々の線形方程式に代入して得られた真の残差ノルムを評価したとき、収束していないと判定される場合も少なくない。

ここで指定された値は、パラメータ「criterion」に代入される。

20

【0082】

図 9 では、丸印のところを選択している点が図 8 と異なる。この項目をチェックすると、全アルゴリズムの組み合わせ (現在は、12 解法 × 8 前処理で 96 通り) に対して、比較したい項目 (「CPU time」か「収束所要回数」) の昇順に結果一覧を表示する。

この図では、「CPU time」の昇順 (早い順) に並べるよう設定している。もう一方は、「収束所要回数」の昇順 (少ない回数で収束した順) に並べるものである。

画面表示の制御では、以下の 2 種類の選択肢がある。

「全ての結果 (収束グラフ データテーブル 出力テーブル)

または

「[CPU time. 収束所要回数] (選択メニュー) の昇順にソートする」

30

どちらも選択しなかった場合のデフォルトは「全ての結果」である。

【0083】

1) 「全ての結果」を選択した場合 (図 8) :

図 19 ~ 21 が表示される (詳しく説明すると、「全ての結果」の中でも、図 19 の「収束グラフ」、図 20 の「データテーブル」、図 21 の「出力データ」の項目を個別に選択することもできる。) 図 8 の「収束における真の残差ノルムの許容値」を設定することにより、データテーブルの表示では、図 21 のとおり、真の残差ベクトルのノルムの値と比較 (計算結果の精度判定) して「収束した」と「見かけ上の収束」とを判別する。

【0084】

2) 「[CPU time. 収束所要回数] の昇順にソートする」を選択した場合 (図 9) :

40

2-1) CPU time の昇順にソートした場合 :

図 23 が表示される。

2-2) 収束所要回数の昇順にソートした場合 :

図 23 と同様に表示される。

【0085】

図 10 は、図 2 の「パラメータ項目選択 (方法 3)」に基づく入力画面である。

図 9 では、行列ごとの情報を表示したが、ここでは、複数の行列と全ての求解アルゴリズムの組み合わせを、比較項目の相対評価の指標 (Rat.) に基づき色可視化情報を用いて一覧表にして表示する。

【0086】

50

丸で囲んだ箇所が、選択メニューによる入力項目である。各入力項目の選択肢の内容は、第1優先ソート項目(第1優先グループ分け項目)の選択、結果の表示で、「解法のグループ」の単位で表記するのか、「前処理のグループ」の単位で表示するのかが選択する。比較項目(「CPU時間」か「収束所要回数」か)を選択するあるいは行列名の選択を行うことである。

【0087】

図8において、「赤色」とは画面上で赤色で表示される項目であり、画面上の「全ての結果」、「All solvers」「none」を示している。Allは全ての組み合わせを選択する場合で、優先度が高いものとしている。そして、解法1に対して1つの欄の前処理を選択できるものとし、この入力欄は複数用意し、解法の選択数に対応するものとしている。

10

【0088】

この設定で実行した結果表示の1例を図11に示す。

図11および図12に記載されている事項を以下に示す。各パラメータとそれらに格納された値は、phpの機能(公知)を用いて提供側計算機システム2を制御するプログラムへ、パラメータ受渡しハンドラーの配列として渡される。

【0089】

((1))表示項目

パラメータ「SEL」に以下のIDを代入する。[]内の数値は、SELに代入されるIDを表す。

全ての結果 [99]

収束グラフ [1]

データテーブル [2]

出力データ [3]

20

【0090】

((2))行列の選択(解くべき問題の内の行列)

パラメータ「matrix」に行列名を代入する。

図のように行列名がメニューに並んでおり、選択された行列名がパラメータmatrixに代入される。

右辺項や初期値他については、各々デフォルト値が用意されている。本実施例システムでは、これらについてはデフォルト値のみ有効としている。

30

【0091】

((3))解法(図12)

図8のとおり、解法名がメニューに並んでおり、選択された解法は2桁のIDとしてパラメータ「solverID」に代入される。ここで、解法を複数選択することが可能であり、パラメータsolverIDは配列形式になっている。具体的な解法名とIDの対応は以下のとおりであり、[]内の数値がIDである。

CG[01], BiCG[02], CGS[03], BiCGStab[04], BiCGStab(l=2)[05], GPBiCG[06], TFQMR[07], Orthomin[08], GMRES[09], Jacobi[10], Gauss-Seidel[11], SOR[12]

40

本実施例システムでは、BiCGStab(l)法のパラメータlの値は、l=2としている。

個々に選択する以外に、動作仕様ファクターとして「All solvers」(全解法)を選択すると、解法のIDとして[99]が設定され、solverID[0]=99となる。

例えば図8のように個々に選択したときは、

1つ目の解法として「CG」を選択 solverID[1]=01

2つ目の解法として「BiCGStab」を選択 solverID[2]=04

【0092】

((4))前処理(解法と併用する技法)(図12)

図8のとおり、前処理名がチェックボックスに並んでおり、選択された前処理のIDが

50

パラメータ「precondID」に代入される。ここで前処理を複数選択することが可能でありパラメータprecondIDは配列形式になっている。具体的な前処理名とIDの対応は以下のとおりであり、[]内の数値がIDである。

none [0 0] (前処理なし), P J a c o b i [0 1], I L U [0 2], S S O R [0 3], H y b r i d [0 4], I + S [0 5], S A I N V [0 6], S A A M G [0 7]

個々に選択する以外に、「All」(全前処理)を選択すると、前処理のIDとして[9 9]が設定される。

【 0 0 9 3 】

前述の「((3)) 解法」にて個々に選択した解法に対応した前処理を選んだとき、例えば1つ目の解法として「CG」を選択し、それと併用する前処理として

「P J a c o b i」と「S S O R」を選択

precondID [1] [0] = 0 1 , precondID [1] [1] = 0 3

2つ目の解法として「BiCGStab」を選択し、それと併用する前処理として

「none (前処理なし)」と「ILU」を選択

precondID [2] [0] = 0 0 , precondID [2] [1] = 0 2

使用する制御言語系のphpの仕様により、配列のインデックスは0から始まるが、実質的には1番目を指している。

解法と併用するその他の技法(スケーリング, オーダリング他)については、本実施例システムの現状では実装されていないが、今後、実装してもよい。

【 0 0 9 4 】

((5)) データ検索と表示を依頼するボタン

次に、各パラメータ項目の選択と動作仕様ファクターの設定について述べる。

クライアント端末200(表示装置)から渡されたパラメータを、フリーソフトウェアphpの機能を用いて分析する。

図8の入力画面にて設定された全パラメータとそれらに格納された値(動作仕様ファクター)は、パラメータ受渡しハンドラー配列に格納され(phiの動作仕様であり、公知)、提供側計算機システム2では、あらためてハンドラー配列(ここでは「POST」という配列名で説明する)から各パラメータと変数値を受け取る。

【 0 0 9 5 】

((1)) で選択された表示形式は、パラメータ「SEL」に値が格納される(SEL = POST [SEL])。

((2)) で選択された行列名は、パラメータ「matrix」に行列名が格納される(matrix = POST [matrix])。

((3)) で選択された解法名は、パラメータ「solverID」(配列形式)に解法のIDが格納される(同様)。

((4)) で選択された技法名は、パラメータ「precondID」(配列形式)に技法のIDが格納される(同様)。

【 0 0 9 6 】

提供側計算機システム2(30)内部での制御の順番は、

(1) パラメータ「matrix」の値の受け取り

(2) パラメータ「solverID」の値の受け取りと解法の個数(配列の要素数)の分析

(3) パラメータ「precondID」の値の受け取りと解法の個数(配列の要素数)の分析

(4) パラメータ「SEL」の値の受け取りとその値に応じた制御であり、それらの処理内容は図13 - 図17で示す。図13 - 図17に記載されている事項を以下に示す。

【 0 0 9 7 】

図13について：

10

20

30

40

50

(1) パラメータ「matrix」の値の受け取り

matrix = POST[matrix]

例：図8の場合は、この受け取りの後、matrix = 1 1 3 8 __ bus と代入されている。

(2) パラメータ「solverID」の値の受け取りと解法の個数（配列の要素数）の分析

phpの機能を用いて、solverID[SS]のサイズ（解法の個数）をカウントする（phpの機能を利用する。公知の技術）、文字列SSは、選択した解法の順番（解法のIDでは無く、図8で複数選択された解法の順番）を示すパラメータである。

【0098】

解法のIDが「99」以外するとき：

(このとき、solverID[0] = 0と代入されている)

例えば図8の画面のとおりに入力だと、

1つ目の解法として「CG」を選択した

solverID[1] = POST[solverID[1]]

このとき、solverID[1] = 01と代入されている。

【0099】

2つ目の解法として「BiCGStab」を選択した

solverID[2] = POST[solverID[2]]

このとき、solverID[2] = 04と代入されている。

【0100】

解法のIDに「99」が存在するとき：

(全解法を選択したとき、このとき、solverID[0] = 99と代入されている)

j = 1, 2, ..., 12

POST[solverID[j]] = 2桁化されたjの値

(例：01, 02, ..., 12)

このとき、solverID[1] = 01, solverID[2] = 02, ..., solverID[12] = 12と代入されている。

【0101】

図14について：

(3) パラメータ「precondID」の値の受け取りと前処理の個数（配列の要素数）の分析

phpの機能を用いて、precondID[SS][PP]のサイズ（各解法に対して選択した前処理の個数）をカウントする。SSは(2)で選択した解法の順番であり、PPは各解法に対して選択した前処理の順番を示すパラメータである。

【0102】

前項(2)にて選択した解法に対応した前処理を選んだとき、例えば(2)において、解法のIDに「99」が存在するとき(solverID[0] = 99のとき)(図8における選択のとき)：

全解法と併用する前処理として「ILU」(ID = 02)が選択されている場合には、

precondID[1][0] = POST[precondID[0][0]],

precondID[2][0] = POST[precondID[0][0]],

.....

precondID[12][0] = POST[precondID[0][0]],

このとき、precondID[1][0] = 02, precondID[2][0] = 02, ..., precondID[12][0] = 02

と、全てにILUのIDが代入されている。

【0103】

使用する制御言語系のphpの仕様により、前処理の配列のインデックスは0から始まるが、実質的には1番目を指している。

10

20

30

40

50

【 0 1 0 4 】

図 1 5 について：

(2) において、解法の ID に「 9 9 」が存在しないとき ($solverID[0] = 0$ のとき) ；

前処理の ID に [9 9] が存在しないとき：

1 つ目の解法として「 CG 」を選択し、それと併用する前処理として「 P J a c o b i 」と「 S S O R 」を選択

$precondID[1][0] = POST[precondID[1][0]]$,
 $precondID[1][1] = POST[precondID[1][1]]$,
 このとき、 $precondID[1][0] = 01$ (P J a c o b i) , $precondID[1][1] = 03$ (S S O R) と代入されている。 10

【 0 1 0 5 】

2 つ目の解法として「 B i C G S t a b 」を選択し、それと併用する前処理として「 n o n e (前処理なし) 」と「 I L U 」を選択

$precondID[2][0] = POST[precondID[2][0]]$,
 $precondID[2][1] = POST[precondID[2][1]]$,
 このとき、 $precondID[2][0] = 00$ (n o n e) , $precondID[2][1] = 02$ (I L U) と代入されている。

使用する制御言語系の php の仕様により、前処理の配列のインデックスは 0 から始まるが、実質的には 1 番目を指している。 20

【 0 1 0 6 】

前処理の ID に [9 9] が存在するとき (全前処理「 All 」が選択されたとき) ；

あらためて、全前処理の ID を格納する。以下の例では、1 つ目に選択した解法に対して、全前処理を選んだときの前処理 ID の格納について説明している。

$$j = 0, 1, \dots, 7$$

$$POST[precondID[1][j]] = 2 \text{ 桁化された } j \text{ の値}$$

(例 : 0 0 , 0 1 , \dots , 0 7)

このとき、

$precondID[1][0] = 00$, $precondID[1][1] = 01$, ... ,
 $precondID[1][7] = 07$ 30

と全ての前処理の ID が代入されている。

(4) パラメータ「 SEL 」の値の受け取りとその値に応じた制御

$SEL = POST[SEL]$ として、パラメータ SEL の値を受け取る。

【 0 1 0 7 】

以下の 1) ~ 3) のときは、方法 1 である。

- 1) SEL の値が 1 または 9 9 のとき、グラフ描画の処理を実行する。
- 2) SEL の値が 2 または 9 9 のとき、データテーブルを作成し表示する処置を実行する。
- 3) SEL の値が 3 または 9 9 のとき、出力データ一覧を表示する処理を実行する。
- 4) SEL の値が 9 0 のとき、検索結果表示 (方法 2) を実行する。

【 0 1 0 8 】

以下の 1) ~ 5) は方法 2 についての説明であるが、方法 1 においても、収束状況に応じたパラメータ「 col _ s t a t 」への値代入と画面表示では、同じ方法を用いている。 40

【 0 1 0 9 】

1) 全解法の ID (0 1 ~ 1 2) と全前処理の ID (0 0 ~ 0 7) から、対応する求解アルゴリズム名を構成して、データを抽出するファイル名も図 6 のとおり構成していく (使うファイルは図 7 の「 .log 」拡張子のファイル) が、各々のファイル名を構成する度に以下の情報を設定していく。また、この処理では、逐次番号 (解法 ID と前処理 ID が増えるごとの番号) を付けていく。

図 7 中の「 .log 」拡張子のファイル内の「 CPU 時間」, 「収束所要回数」, 「真の残差 50

のノルム値」とパラメータ「criterion」とを比較評価（方法1と同じ）した結果と「求解状況」の情報に応じて以下の表のとおり設定する。

状況	パラメータ「stat」に設定する値	パラメータ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col (黄色)
反復回数の上限	mat.itr.	war_col (赤色)
ブレークダウン	brk.dwn.	war_col (赤色)

これらと、アルゴリズム名 (precondNMsolverNM) と逐次番号 (num_sel) も配列に格納する。

```
INFO_dat [ num_sel ] [ num ] = num_sel
INFO_dat [ num_sel ] [ alg ] = preconditionSolverNM
INFO_dat [ num_sel ] [ itr ] = 抽出した収束所要回数
INFO_dat [ num_sel ] [ cpu ] = 抽出したCPU時間
INFO_dat [ num_sel ] [ rsd ] = 抽出した真の残差ノルム
INFO_dat [ num_sel ] [ sta ] = 抽出した求解状況
INFO_dat [ num_sel ] [ col ] = col_stat
```

2) 1) で得られた配列「INFO_dat」に対して、比較項目 (sort_keyの値) に応じ、CPU時間か収束所要回数のどちらかで昇順ソート (phpの関数を使用) する。

3) ソート後、col_statの値がNULLのレコード (収束したもの) を対象にして、最初のレコードの比較項目の値をパラメータ「TheFastest」に代入し、以後のレコードに対して、比較項目の相対評価の指標 (現状では「TheFastest ÷ 比較項目の値」) を算出し、少数第1位までをパラメータ「Rat」に代入する。

4) Ratの値ごとに、図24のデータテーブルをhtmlで表示する。このとき、対応する行のバックグラウンド色を「col_stat」の色で表示する。

5) 収束したものについては、gnuplotを用いて比較項目のデータの棒グラフを描く。

(5) パラメータ「SEL」の値の受け取りとその値に応じた制御
SEL = POST {SEL} として、パラメータSELの値を受け取る。

【0110】

1) SELの値が1または99のとき、グラフ描画の処理を実行する。

- 1) 下記の「loop_solpre」を実行する。
- 2) グラフにするデータは、図7中の「.rsd」拡張子のファイルをプロットする。
- 3) グラフ描画では、フリーソフトウェアのgnuplotを用いる。
- 4) gnuplotの制御は、本プログラムでgnuplotへの入力データ (gnuplot制御プログラム) を生成し実行する。

2) SELの値が2または99のとき、データテーブルを作成し表示する処理を実行する。

- 1) 下記の「loop_solpre」を実行する。
- 2) CPU時間、収束所要回数などのデータを表の形式にする。これらのデータは、図7中の「.log」拡張子のファイルに格納されている。このファイルから該当データをphpの機能を用いて抽出する。

3) 図7中の「.log」拡張子のファイル内の「真の残差のノルム値」とパラメータ「criterion」とを比較評価した結果 (真の残差のノルムがcriterion以下ならば、収束した判定。criterionよりも大きい値ならば、見かけ上の収束と判定。) と「求解状況」の情報に応じて、以下の表のとおり設定する。

状況	パラメータ「stat」に設定する値	パラメータ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col (黄色)
反復回数の上限	mat.itr.	war_col (赤色)
ブレークダウン	brk.dwn.	war_col (赤色)

4) データテーブル表示のhtmlを作成するとき、上記の状況に応じて、対応する行のバックグラウンド色を「col_stat」の色で表示する。

【 0 1 1 1 】

3) SELの値が3または9 9 のとき、出力データ一覧を表示する処理を実行する。

- 1) 下記の「loop_solpre」を実行する。
- 2) 図7中の「.log」拡張子のファイルで該当するものを、全て画面に表示する。
- 3) SELの値が9 0 のとき、検索結果表示(方法2)を実行する。

【 0 1 1 2 】

図18に、検索対象であるファイル名を構成する処理を示す。

前述のパラメータを受け取り、図18の「loop_solpre」プログラムを用いて、データベースにアクセスするためのファイル名を構成し、ファイル検索する。

「loop_solpre」プログラムで得られた情報から、(A)を用いてデータベースのファイルを検索する。

各結果表示の際には、(B)を用いてアルゴリズム名の方を表示する。

ここで、上記パラメータ値からファイル名を構成する方法は、図6と同様である。

これらの制御では、phpというフリーソフトウェアを用いる。

このようにして図8に示すように、各パラメータの表示手段と各パラメータの選択手段、収束グラフ、CPU時間および収束所要回数のいずれかもしくはこれらの組み合わせの表示手段とこれらの内のいずれかを選択する選択手段、各パラメータを組み合わせた形式での動作仕様ファクターの組み合わせを複数表示する表示手段といずれかの組み合わせの1つまたは複数を選択する選択手段、を1つの画面に同時に表示する設定画面表示手段が構成される。

【 0 1 1 3 】

また、画面上で、パラメータのいずれか、収束グラフ、CPU時間および収束所要回数のいずれかもしくはこれらの組み合わせ、および各パラメータを組み合わせ形式での動作仕様ファクターの組み合わせの1つまたは複数選択すると、これらの組み合わせについてデータベースに格納されたデータを検索し、取得する処理を行うデータ取得処理手段が構成される。

【 0 1 1 4 】

そして、設定画面表示手段には、解法についてのパラメータの前記動作仕様ファクターは複数並列して表示され、該複数並列した前記動作仕様ファクター毎に、解法と併用する技法についてのパラメータの動作仕様ファクターが表示され、この解法と併用する技法についてのパラメータの動作仕様ファクターは複数の動作仕様ファクターが複数表示される。

【 0 1 1 5 】

次に、結果表示手段15について説明する。

図19、図20は検索結果表示例を示す。図20は、図19から続く画面である(画面スクロールしたところ)。

図19は、図8のとおりの設定にて検索実行したときの結果表示であり、選択した解法と、それらと併用する前処理技法を組合せた収束求解アルゴリズムによる収束グラフが画面表示されている。グラフの縦軸は収束求解アルゴリズムの中で算出されている残差ベクトルの2ノルムを計算(公知の評価方法)し、その値のlog10を取ったものである。横軸は、反復解法の反復回数を表している。

図20は、CPU時間や収束所要回数等の情報を一覧にした表が画面表示されている。

【 0 1 1 6 】

「Data Table」の表の項目は次のとおりである。

No .	選択した求解アルゴリズムに付した番号
Pre c - Sol v e r	収束求解アルゴリズム名。選択した解法と、それと併用する前処理技法を組合せた名称。
I t e r .	収束所要回数。収束求解アルゴリズムで得られる近似解が収束するまにかかった反復回数。求解アルゴリズムの残差ベクトルのノルムに対する収束判定は、10のマイナス12乗(コンピュータ上の表現では1 . 0

10

20

30

40

50

e - 12) で判定した。

C P U t i m e 収束までに要した C P U 時間 [単位 : 秒] を表す。

R e s . n o r m 真の残差の値 . 求解アルゴリズム中の残差ベクトルを元に収束と判定されても、求められた近似解を代入した真の残差 ($b - A x$) は、収束には程遠い値である場合もある。

S t a t u s 真の残差の値から収束した (c o n v .) が、見かけ上の収束 (n o c o n v .) が、全く収束していない (「 m a x . i t r . (反復回数の上限に到達) 」 「 b r k . d w n . (ブレークダウン : 何らかの数値的不安定により反復計算の続行が不可能となった) 」) を表示する。

10

D a t a T a b l e の番号を横軸にして、各々の C P U 時間をビジュアルに比較評価し易くするために棒グラフを表示している。

この図の一番下にある「 C o n v e r g e d : 1 , 2 , 4 , 6 」の表示は、収束求解アルゴリズムの番号を表示している。

【 0 1 1 7 】

図 2 1 は、図 2 0 からさらに続く画面 (画面スクロールしたところ) で、図 6 , 7 中の拡張子「 . l o g 」のファイルの内容 (O u t p u t d a t a) をそのまま表示したものである。

【 0 1 1 8 】

結果画面表示手段 1 5 には、パラメータ S E L の値に応じて、必要な情報を画面表示できるように編集する。本実施例システムでは、これらの制御は p h p を用い、画面表示用には h t m l の仕様に従った出力を行っている。

20

【 0 1 1 9 】

データベース 1 3 の中から、反復解法の収束の様子を表す残差のデータ (図 7 では「 1 1 3 8 _ b u s - 0 4 0 1 0 0 0 0 . r s d 」である) をグラフ表示するにあたっては、フリーソフトウェアの g n u p l o t を用いている。g n u p l o t を実行するのに必要な入力ファイルも p h p の機能を用いて一時ファイルを作成し、それを用いて残差データ (テキスト形式) から画像データ (バイナリ形式) を生成し、画面表示する。

【 0 1 2 0 】

C P U 時間や収束所用回数などのデータを表示するにあたっては、テキスト形式のログファイル (図 7 では、「 1 1 3 8 _ b u s - 0 4 0 1 0 0 0 0 . l o g 」である) の中から該当箇所を抽出し、表形式として画面表示する。

30

【 0 1 2 1 】

このように結果画面表示手段 1 5 は、データ取得処理手段で取得したデータに基づいて収束グラフ、C P U 時間および収束所要回数のいずれかもしくはこれらの組み合わせ表示することを行う。そして、結果画面表示手段 1 5 の画面には、前記収束グラフが前記動作仕様ファクターの複数の組み合わせに基づいて複数同時に表示される。

【 0 1 2 2 】

図 2 2 は、図 5 において、収束求解アルゴリズムについて、画面上に比較項目を含むパラメータの設定を行い (S 1)、真の残差ノルムの基準値の設定を行い (S 2)、真の残差ノルム値が真の残差ノルム基準値の範囲内にあるかについて判定 (S 4) を行っている状況を示す。

40

【 0 1 2 3 】

図 2 2 に示すように、真の残差ノルムが $1.0 e - 08$ (図 8 で設定した値) 以下であれば、「conv. (収束した) 」と判定して、白色表示、数値アルゴリズム中の残差ノルムでは収束したが、真の残差ノルムが許容値以下の場合には「no conv. (見かけ上の収束) 」と判定して、黄色表示、それら以外の場合には、赤色表示する。この例では、「max.itr. (最大反復回数) 」に到達したことを示している。

【 0 1 2 4 】

図 2 3 は、図 5 において、ステップ S 3 によって判定した結果に基づいて数値アルゴリ

50

ズムのリストの並び替えを行い (S 4)、対象の収束求解アルゴリズムについて、比較項目についての性能データの出力を行い (S 5)、対象の収束求解アルゴリズムについての性能を相対比較による相対判定を行うと共に対象の収束求解アルゴリズムについての相対比較値 (相対判定値) の算出を行った (S 6) 例を示す。そして、この例は、最終的に、相対比較、相対比較値に対応して、対象の収束求解アルゴリズムについて画面上で、上記同様に、色分け表示 (白、黄、赤) を行う。

【 0 1 2 5 】

テーブルの項目は図 2 2 と示す項目とほぼ同じであるが、相対判定の指標として「Rat.」 (Ratio比率) の項目が追加されている。現状ではRat.は、「Rat. = (最初の比較項目の値) ÷ (対象の収束求解アルゴリズムの比較項目の値)」と定義している。図 9 の例の場合 (CPU timeの昇順にソートする) には、Rat. = (最速のCPU時間) ÷ (対象の収束求解アルゴリズムのCPU時間)」である。算出した値については、上記同様に色分け表示することができる。

10

【 0 1 2 6 】

図 2 4 は、図 1 0 にて第 1 優先の項目として「解法」、および、行列のタイプとして「対称」を選択して実行したときの出力結果を表わしている。図 2 4 において、設定の内容は次の通りである。

【 0 1 2 7 】

第 1 優先のソート項目：「解法」
 昇順にソートする表示結果：「CPU時間」
 真の残差ノルムに対する許容値：「1.0 e - 0 8」
 行列のタイプ：「対称行列」

20

解法と前処理について、IDと名称の組合せについて表示している。

一覧表の横軸であり、上段に解法のID、下段に前処理のIDを表示している。

【 0 1 2 8 】

図 1 0 で「解法」を第 1 優先にグループ分けしたとすると、この表示では、

解法(Solver) : 01 01 01 01...01 02 02 02 02...02 03.....12
 前処理(Precond.) : 00 01 02 03...07 00 01 02 03...07 01.....07

という順序となり、結局、各解法ごとにグループ分けされている。これと同じ表示は、表の下部にも再度表示している。

30

【 0 1 2 9 】

一覧表の縦軸は、指定されたタイプの行列名一覧を表示している。ここで、行列のタイプは、図 1、2 のデータベース 1 3 B から情報を採取する。

【 0 1 3 0 】

各々の行列についての代表的特性から、「Size V.」 (行方向の行列サイズ) と「Cond. Num.」 (行列の条件数) を表示している。これらはともに、図 1、2 のデータベース 1 3 B から情報を採取する。

【 0 1 3 1 】

各々の行列に対する全アルゴリズムの性能を、比較項目 (この図では、CPU時間) の相対判定値として横方向 (各行列ごと) に並べている。ここでの相対判定値とは、前述の「Rat.」を 1 0 倍した値 (表示の都合で、小数点のような冗長な文字を省くために 1 0 倍した) のことである。図 2 3 では縦方向へ比較項目の昇順に並んでいるが、ここでは、横方向へアルゴリズムのID順 (特に図 1 0 で設定されたとおり) に並んでいる。

40

【 0 1 3 2 】

図 2 3 における「conv.」に該当する項目については、相対判定値を記載しており、それ以外の項目については、すべて「*」 (アスタリスク) のみを表示している。「conv.」以外の項目について、ピリオドで表示するのは「no conv.」のみにして、それ以外の「max.itr.」や「brk.dwn.」などについては、「.」 (ピリオド) を表示させる。

【 0 1 3 3 】

また、相対判定値 (Rat) の数値に応じて色分けもしている。色分けの説明は前述した

50

とおりである。

相対判定値に応じた色分け情報は、スコア (score) が 0 が白色、10 が黒色としている。その間の 1 ~ 9 に対しては、薄い黄色から濃い赤色へとグラデーションしている。

【0134】

次に図 1 について、クライアント端末 200 について説明する。クライアント端末 200 は、表示装置 51 を備える。表示装置 51 は、問題の表示 52、パラメータ項目選択 53、検索結果表示 54、および、より妥当な収束求解アルゴリズムの選定表示 55 を行う。

【0135】

図 3 に示すように、クライアント端末 200、すなわち表示装置 51 は各パラメータの動作仕様ファクターの設定 61 を行い、検索や比較の実行指示 62 によって検索、比較要求を提供側計算機システム 2 (30) にて行い、および返却された結果を検索や比較の結果として表示 63 (検索結果表示) を行う。

10

【0136】

1) 問題の表示

自然現象や工学現象の解明では、数値シミュレーションを用いた解析が盛んである。これらのシミュレーションでは、多くの場合、線形方程式や固有値問題などを始めとする多くの種類の数値計算の問題を解くことに帰着される。ところが、線形方程式を例にとると、収束求解アルゴリズムも様々なものが存在し、対象とする問題の性質によっては、その性能が十分に発揮されないようなものもある。線形方程式以外の数値計算においても同じような状況である。従って、実際の数値計算シミュレーションにあたっては、どの収束求解アルゴリズムを適用したら良いか指針が欲しいところである。クライアントは、クライアント端末 200 に問題の表示を行うことになる。

20

【0137】

2) パラメータ項目選択：

図 8 を参照して、解きたい問題の性質に似たタイプの問題を選定することを行う。すなわち、パラメータの項目および各パラメータ項目についての動作仕様ファクターの選定を行う。これによって、検索要求がなされる。すなわち、ここでは前述した方法 1、方法 2 のパラメータの項目選択を行う。また、図 10 では複数の問題 (行列) に対する全アルゴリズムの結果一覧を得るための設定を行う。すなわち、方法 3 のパラメータの項目選択を行う。ここで、「行列の選択」欄を選択した場合 (空白にしなかった場合) には、方法 2 が行われる。

30

【0138】

3) 検索結果表示：

図 19、図 20、図 21 に示すようにして検索結果が通信手段を介してフィードバックされ、表示装置 51 の画面上に表示される。すなわち、ここでは前述した方法 (1)、方法 (2)、方法 (3) のパラメータの項目選択に対応した検索結果を表示する。

【0139】

4) 収束求解アルゴリズムの選定：

ここでは、収束求解アルゴリズムに対する性能を比較し、相対判定を行う。

40

検索結果を参考にして、自分の問題を解くのに適していると思われる収束求解アルゴリズムを選定し、再度検索要求を行い、検索結果を表示することを行う。

【0140】

図 3 において、提供側計算機システム 2 (30) では入力されたパラメータ項目、パラメータ項目毎の動作仕様ファクターの選択によって入力パラメータ解析 71 が行われ、該当ファイルを検索 72 してデータベース 13 に該当ファイルを探してデータとしての提供を受け、対象データを表示用に編集 73 を行う。編集された結果は、検索結果としてクライアント端末にフィードバックされ、検索や比較の結果表示 63 がなされることになる。

【0141】

このように、収束求解アルゴリズム性能比較表示装置 100 は、設定画面表示手段の画

50

面に表示されたいずれかの選択手段についても通信手段を介してのクライアント端末からの指令信号によって選択操作可能である。

【0142】

以上、説明したように、収束求解アルゴリズム性能比較表示装置は、次の要素で構成し得ることになる。

【0143】

収束求解アルゴリズム毎に、解くべき問題、解法および解法と併用する技法の各項目について設定したパラメータ、各パラメータについて設定され、比較項目の動作仕様を決定する動作仕様ファクターおよび各項目から選択された前記動作仕様ファクターの組み合わせに関連づけられた比較項目の演算を行うコンピュータプログラムを格納すること。

10

前記コンピュータプログラムを使用して、各項目から選択された前記動作仕様ファクターの組み合わせに関連づけられた比較項目について演算処理すること。

前記演算処理手段の演算結果である、各項目から選択された前記動作仕様ファクターの組み合わせに関連づけて比較項目についてのデータを格納すること。

各パラメータの表示手段と各パラメータの選択手段、比較項目の表示手段とこれらの内のいずれかを選択する選択手段、各パラメータを組み合わせた形式での前記動作仕様ファクターの組み合わせを複数表示する表示手段といずれかの組み合わせの1つまたは複数を選択する選択手段、を1つの画面に同時に表示すること。

前記画面上で、パラメータのいずれか、比較項目の内のいずれかもしくはこれらの組み合わせ、および各パラメータを組み合わせ形式での前記動作仕様ファクターの組み合わせの1つまたは複数選択すると、これらの組み合わせについて前記データベースに格納されたデータを検索し、取得する処理を行うこと。

20

該データ取得処理手段で取得したデータに基づいて比較項目のいずれかもしくはこれらの組み合わせについて相対比較を行い、複数の数値アルゴリズムについて性能比較と相対判定すること。

性能比較結果を表示すること。

前記性能比較処理手段は、前記相対比較の前に、設定した真の残差ノルムの基準値の範囲内にあるか、範囲外にあるか、について判定を行うこと。

前記結果画面表示手段の画面には、前記相対評価の結果が色分けして表示すること。

【図面の簡単な説明】

30

【0144】

【図1】本発明の実施例の構成を示すブロック図。

【図2】図1の一部詳細を示す図。

【図3】図1のフローチャート図。

【図4】前処理付き共役勾配法の数値アルゴリズムの例を示す図。

【図5】本発明の実施例のフローを示す図。

【図6】データベースに載せる際の情報提供状況を示す図。

【図7】動作仕様ファクターに関連して収束所要回数、CPU時間が格納される状況を示す図。

【図8】図1のパラメータ項目選択（方法1）に基づく入力画面図。

40

【図9】図1のパラメータ項目選択（方法2）に基づく入力画面図。

【図10】図1のパラメータ項目選択（主に方法3．行列名を指定した場合には方法2としても機能する）に基づく入力画面図。

【図11】図8～9の画面にてパラメータに代入する例を示す図。

【図12】図8～9の画面にてパラメータに代入する例を示す図。

【図13】パラメータの値の受け取り、パラメータの値の受け取りと解法の個数（配列の要素数）の分析を説明する図。

【図14】パラメータの値の受け取りと前処理の個数（配列の要素数）の分析を説明する図。

【図15】解法のIDに「99」が存在しないときの説明図。

50

【図16】パラメータ「SEL」の値の受け取りとその値に応じた制御を説明する図。(その1)

【図17】パラメータ「SEL」の値の受け取りとその値に応じた制御を説明する図。(その2)

【図18】検索対象であるファイル名を構成する処理の図。

【図19】結果表示例図。

【図20】図19から続く画面を示す図。

【図21】図19から続く画面を示す図。

【図22】方法1により判定結果を示す図。

【図23】方法2により相対判定結果を示す図。

【図24】方法3により出力結果を示す図。

【符号の説明】

【0145】

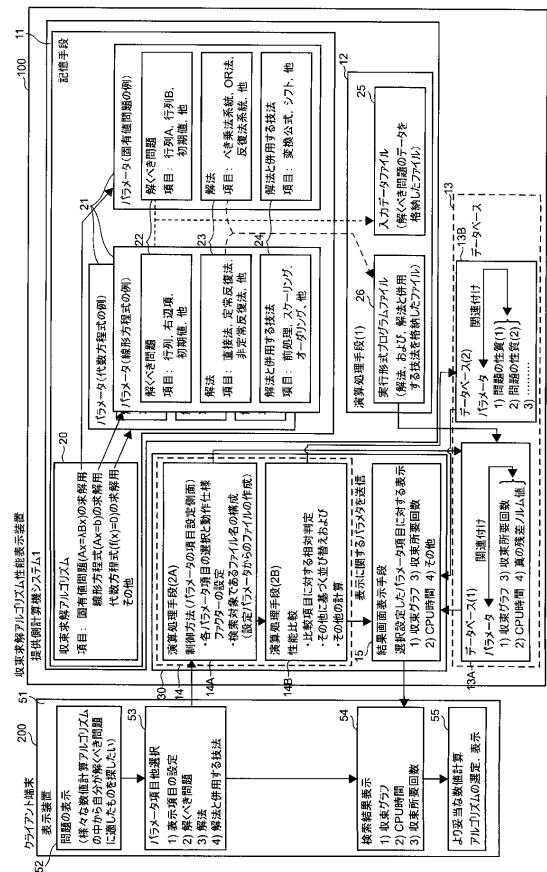
11...記憶手段、12...演算処理手段(1)、13...データベース、13A...データベース(1)、13B...データベース(2)、14...演算処理手段(2)、14A...演算処理手段(2A)、14B...演算処理手段(2B)、15...結果画面表示手段、20...収束求解アルゴリズム、21...パラメータ、22...解くべき問題、23...解法、24...解法と併用する技法、25...入力データファイル、26...実行形式プログラムファイル、51...表示装置、52...問題の表示、53...パラメータ項目他選択、54...検索結果表示、55...より妥当な収束求解アルゴリズムの選定・表示、61...各パラメータの動作仕様ファクター(例えば値)の設定、62...検索実行指示、63...検索結果表示、71...入力パラメータ解析、72...該当ファイル検索、73...対象データを表示用に編集、100...収束求解アルゴリズム性能比較表示装置、200...クライアント端末。

10

20

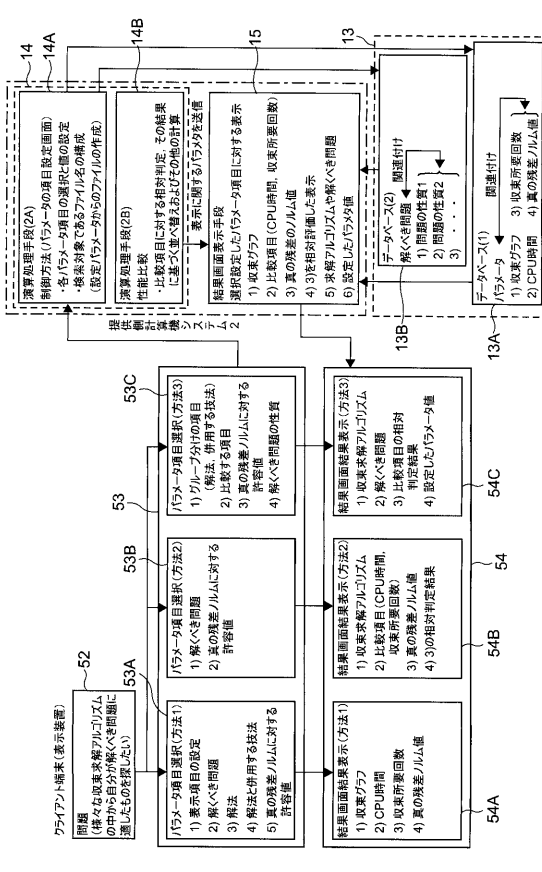
【図1】

図 1

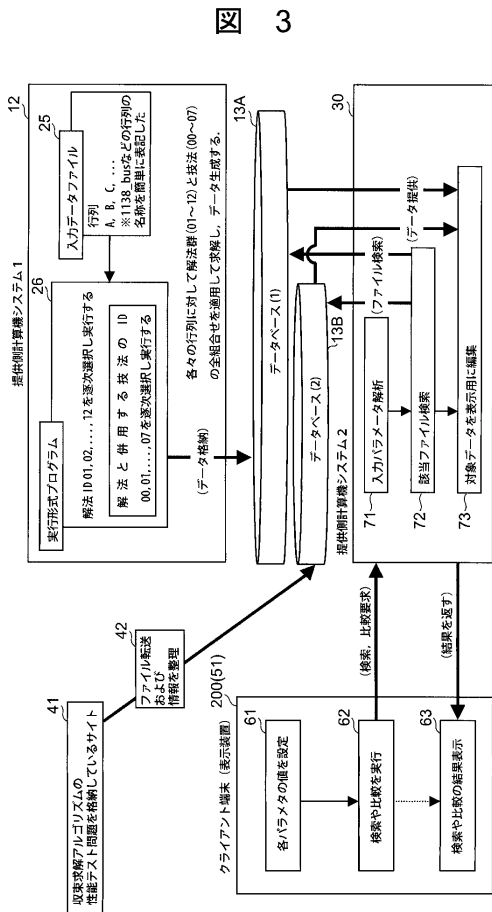


【図2】

図 2



【図3】



【図4】

図4は、最適化アルゴリズムの数学的表現を示している。

・ x_0 : 適当な初期値を与える。
 ・ $r_0 = b - Ax_0, p_0 = K^{-1}r_0$
 for $k = 0, 1, \dots$, until $\|r_k\| \leq \epsilon \|b\|$ do
 begin

$$\alpha_k = \frac{(K^{-1}r_k, r_k)}{(p_k, Ap_k)}$$

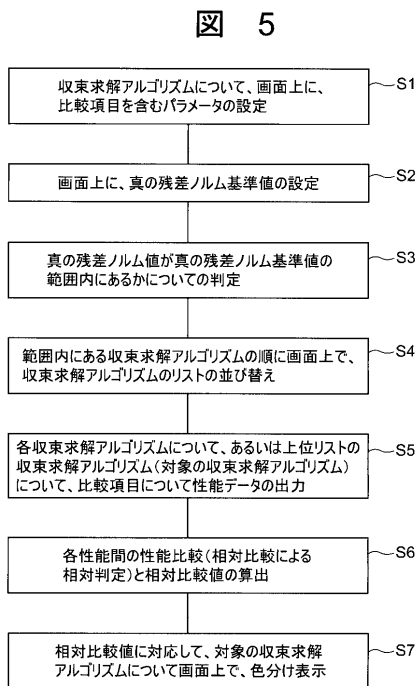
$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k Ap_k$$

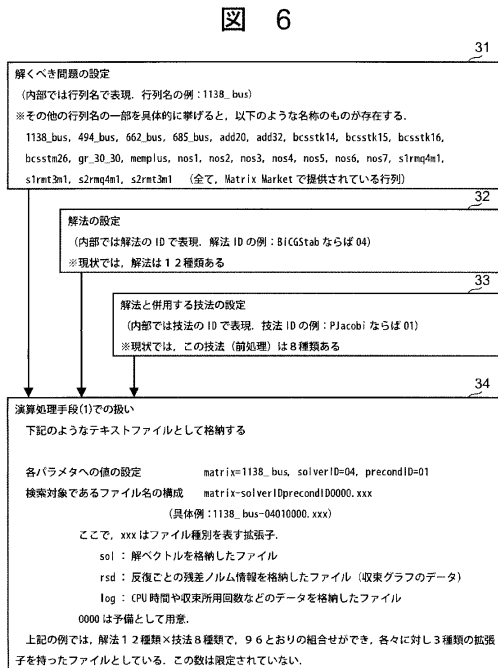
$$\beta_k = \frac{(K^{-1}r_{k+1}, r_{k+1})}{(K^{-1}r_k, r_k)}$$

$$p_{k+1} = K^{-1}r_{k+1} + \beta_k p_k$$
 end

【図5】



【図6】



【 図 7 】

図 7

1138_bus-04010000.sol

```

1 9.99997105112556905462e-01
2 9.99420216753297285806e-01
3 9.99291805648155118114e-01
4 9.99892583580029303825e-01
5 9.99900544205886698812e-01
6 9.99878961769620988598e-01
7 9.99890408421171050790e-01
... 以下省略 ...

```

1138_bus-04010000.rsd

```

4.331952e-03
1.578029e-03
1.044702e-03
9.643088e-04
8.726076e-04
7.950370e-04
7.054724e-04
6.781388e-04
... 以下省略 ...

```

1138_bus-04010000.log

```

1138 x 1138 matrix 2596 entries
SOLVER# BI08STAB
PHECON# Jacobi
lis_solve is LIS MAXITER (code=4)
# iter = 139 time = 4.217069e-01
# (n=4, 279613e-04 i=4, 212790e-01
# Last Residual = 1.864383e-02

```

Diagram showing arrows from log output to labels: 収束所要回数 (convergence iterations), 求解状況 (solution status), CPU時間 (CPU time), 真の残差のノルム値 (true residual norm value).

【 図 8 】

図 8

線形方程式アルゴリズム性能評価の情報検索システム

ステップ1)以下のパラメタを設定して下さい。(赤色の項目はデフォルト値です)
ステップ2)“実行”ボタンをクリックして下さい。

検索実行 | 入力値をクリア

表示させる項目を選択: (“全ての結果”の項目は優先度が高いです。)
 全ての結果 (収束グラフ データテーブル 出力データ)
 または
 CPU time | の昇順にソートする

行列の選択: [1138_bus] |

収束における真の残差ノルムの許容値: [1.0e-08] |

全アルゴリズムについて、アルゴリズム中の残差ノルムは1.0e-12で収束判定している。

右辺項と初期値について
 →右辺項: 解ベクトルの全要素を1.0として作成した(Lisのデフォルト)
 →初期値: 全要素を0.0として作成した(Lisのデフォルト)

解法と前処理の選択:

解法(空白は無視される)	前処理 (“All”は優先度が高いです)
<input checked="" type="checkbox"/> All solvers	<input type="radio"/> none <input type="radio"/> PJacobi <input type="radio"/> ILU
あるいは、下記から	<input type="radio"/> SSOR <input type="radio"/> Hybrid <input type="radio"/> i+S
いくつかの解法を選択する)	<input type="radio"/> SAINV <input type="radio"/> SAAMG
<input type="checkbox"/> All	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> i+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU

【 図 9 】

図 9

線形方程式アルゴリズム性能評価の情報検索システム

ステップ1)以下のパラメタを設定して下さい。(赤色の項目はデフォルト値です)
ステップ2)“実行”ボタンをクリックして下さい。

検索実行 | 入力値をクリア

表示させる項目を選択: (“全ての結果”の項目は優先度が高いです。)
 全ての結果 (収束グラフ データテーブル 出力データ)
 または
 CPU time | の昇順にソートする
 CPU time
 収束所要回数 | の昇順にソートする

収束における真の残差ノルムの許容値: [1.0e-08] |

全アルゴリズムについて、アルゴリズム中の残差ノルムは1.0e-12で収束判定している。

右辺項と初期値について
 →右辺項: 解ベクトルの全要素を1.0として作成した(Lisのデフォルト)
 →初期値: 全要素を0.0として作成した(Lisのデフォルト)

解法と前処理の選択:

解法(空白は無視される)	前処理 (“All”は優先度が高いです)
<input checked="" type="checkbox"/> All solvers	<input type="radio"/> none <input type="radio"/> PJacobi <input type="radio"/> ILU
あるいは、下記から	<input type="radio"/> SSOR <input type="radio"/> Hybrid <input type="radio"/> i+S
いくつかの解法を選択する)	<input type="radio"/> SAINV <input type="radio"/> SAAMG
<input type="checkbox"/> All	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> i+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU

【 図 10 】

図 10

全アルゴリズムと全行列に対する性能比較テーブル表示システム

ステップ1)以下のパラメタを設定する。
ステップ2)“実行”ボタンをクリックして下さい。

比較実行 | 入力値をクリア

第1優先のソート項目: [CPU時間(CPU time)] | の昇順にソートする

行列の選択: [] |

単一の行列に対する性能を比較する時は空白の場合は全ての行列について比較)

収束時の真の残差ノルムに対する基準値を設定: [1.0e-08] |

全アルゴリズムについて、アルゴリズム中の残差ノルムは1.0e-12で収束判定している。

行列のタイプ: [あらゆるタイプ(Any)] |

右辺項と初期値について
 →右辺項: 解ベクトルの全要素を1.0として作成した(Lisのデフォルト)
 →初期値: 全要素を0.0として作成した(Lisのデフォルト)

【 図 1 1 】

図 11

(1) 表示項目
パラメタ「SEL」に以下の ID を代入する。[]内の数値は、SEL に代入される ID を表示。
全ての結果 [99]
収束グラフ [1]
データテーブル [2]
出力データ [3]

(2) 行列の選択 (解くべき問題の内の行列)
パラメタ「matrix」に行列名を代入する。
下図のように行列名がメニューに並んでおり、選択された行列名がパラメタ matrix に代入される。

行列の選択 [1138_bus]	
1138_bus	
494_bus	
682_bus	
685_bus	
初期値:全	
add92	
解法 (空白は無し)	前処理 ("ALL"は優先度が高いです。)
<input type="checkbox"/> All solvers	<input checked="" type="radio"/> none <input type="radio"/> PJacobi <input type="radio"/> ILU
(あるいは、下記からいくつかの解法を選び)	<input type="radio"/> SSOR <input type="radio"/> Hybrid <input type="radio"/> I+S
<input type="checkbox"/> CG	<input type="checkbox"/> SAINV <input type="radio"/> SAAMG
<input type="checkbox"/> BiCGStab	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
	<input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG

右辺項や初期値他については、各々デフォルト値が用意されている。本実施例システムでは、これらについてはデフォルト値のみが有効である。

【 図 1 2 】

図 12

(3) 解法
図11のとおり、解法名がメニューに並んでおり、選択された解法は2桁の ID としてパラメタ「solverID」に代入される。ここで、解法を複数選択することが可能であり、パラメタ solverID は配列形式になっている。具体的な解法名と ID の対応は以下のとおりであり、[]内の数値が ID である。
CG [01], BiCG [02], CGS [03], BiCGStab [04], BiCGStab(l=2) [05], GPBiCG [06], FTQMR [07], Orthomin [08], GMRES [09], Jacobi [10], Gauss-Seidel [11], SOR [12]
※ 現在、BiCGStab(l)法のパラメタ l の値は、l=2 としている。
個々に選択する以外に、「All Solvers」(全解法)を選択すると、解法の ID として [99] が設定され、solverID[0]=99 となる。
例えば図11のように個々に選択したときは、
1 目目の解法として「CG」を選択 → solverID[1]=01
2 目目の解法として「BiCGStab」を選択 → solverID[2]=04

(4) 前処理 (解法と併用する技法)
図11のとおり、前処理名がチェックボックスに並んでおり、選択された前処理の ID がパラメタ「precondID」に代入される。ここで前処理を複数選択することが可能でありパラメタ preconditionID は配列形式になっている。具体的な前処理名と ID の対応は以下のとおりであり、[]内の数値が ID である。
none [00] (前処理なし), PJacobi [01], ILU [02], SSOR [03], Hybrid [04], I+S [05], SAINV [06], SAAMG [07]
個々に選択する以外に、「All」(全前処理)を選択すると、前処理の ID として [99] が設定される。
前述の「(3) 解法」にて個々に選択した解法に対応した前処理を選んだとき、例えば
1 目目の解法として「CG」を選択し、それと併用する前処理として「PJacobi」と「SSOR」を選択 → preconditionID[1][0]=01, preconditionID[1][1]=03
2 目目の解法として「BiCGStab」を選択し、それと併用する前処理として「none (前処理なし)」と「ILU」を選択 → preconditionID[2][0]=00, preconditionID[2][1]=02
※使用する制御言語系の php の仕様により、配列のインデックスは 0 から始まるが、実質的には 1 番目を指している。
※解法と併用するその他の技法 (スケールリング、オーダリング他) については、本実施例システムでは実装されていないが、実装してもよい。

(5) データ検索と表示を依頼するボタン

【 図 1 3 】

図 13

(1) パラメタ「matrix」の値の受け取り
matrix = POST[matrix]
例: 図8の場合は、この受け取りの後、matrix = 1138_bus と代入されている。

(2) パラメタ「solverID」の値の受け取りと解法の個数 (配列の要素数) の分析
php の機能を用いて、solverID[SS] のサイズ (解法の個数) をカウントする (php の機能を利用する。公知の技術)。文字列 SS は、選択した解法の順番 (解法の ID では無く、図6で複数選択された解法の順番) を示すパラメタ。
解法の ID が「99」以外るとき:
(このとき、solverID[0]=0 と代入されている)
例えば図8の画面のとおり入力だと、
1 目目の解法として「CG」を選択した → solverID[1] = POST[solverID[1]]
このとき、solverID[1] = 01 と代入されている。
2 目目の解法として「BiCGStab」を選択した → solverID[2] = POST[solverID[2]]
このとき、solverID[2] = 04 と代入されている。
解法の ID に「99」が存在するとき:
(全解法を選択したとき、このとき、solverID[0]=99 と代入されている)

解法と前処理の選択	
解法 (空白は無視される)	前処理 ("ALL"は優先度が高いです。)
<input checked="" type="checkbox"/> All solvers	<input type="radio"/> none <input type="radio"/> PJacobi <input type="radio"/> ILU
(あるいは、下記からいくつかの解法を選択する)	<input type="radio"/> SSOR <input checked="" type="radio"/> Hybrid <input type="radio"/> I+S
	<input type="radio"/> SAINV <input type="radio"/> SAAMG

j = 1, 2, ..., 12
POST[solverID[j]] = 2 桁化された j の値
→ (例: 01, 02, ..., 12)
このとき、solverID[1]=01, solverID[2]=02, ..., solverID[12]=12 と代入されている。

【 図 1 4 】

図 14

(3) パラメタ「precondID」の値の受け取りと前処理の個数 (配列の要素数) の分析
php の機能を用いて、precondID[SS] [PP] のサイズ (各解法に対して選択した前処理の個数) をカウントする。SS は (2) で選択した解法の順番であり、PP は各解法に対して選択した前処理の順番を示すパラメタ。
前項 (2) にて選択した解法に対応した前処理を選んだとき、例えば
(2)において、解法の ID に「99」が存在するとき (solverID[0]=99 のとき):

全解法と併用する前処理として「Hybrid」(ID=04) が選択されている場合には、

```
precondID[1][0] = POST[precondID[0][0]],
precondID[2][0] = POST[precondID[0][0]],
...
precondID[12][0] = POST[precondID[0][0]].
```

このとき、precondID[1][0]=04, precondID[2][0]=04, ..., precondID[12][0]=04 と、全てに Hybrid の ID が代入されている。
※使用する制御言語系の php の仕様により、前処理の配列のインデックスは 0 から始まるが、実質的には 1 番目を指している。

この説明は図15に続く。

【 図 1 5 】

図 15

図14からの続き。

(2)において、解法のIDに「99」が存在しないとき (solverID[0]=0のとき) :

前処理のIDに「99」が存在しないとき (図8の選択のとき) :

1つ目の解法として「G」を選択し、それと併用する前処理として「PJacobi」と「SSOR」を選択

```
precondID[1][0] = POST[precondID[1][0]],
precondID[1][1] = POST[precondID[1][1]],
```

このとき、precondID[1][0]=01 (PJacobi)、precondID[1][1]=03 (SSOR)と代入されている。

2つ目の解法として「BiCGstab」を選択し、それと併用する前処理として「none (前処理なし)」と「ILU」を選択

```
precondID[2][0] = POST[precondID[2][0]],
precondID[2][1] = POST[precondID[2][1]],
```

このとき、precondID[2][0]=00 (none)、precondID[2][1]=02 (ILU)と代入されている。

※使用する制御言語系のphpの仕様により、前処理の配列のインデックスは0から始まるが、実質的には1番目を指している。

前処理のIDに「99」が存在するとき (全前処理「All」が選択されたとき) :

あらためて、全前処理のIDを格納する。以下の例では、1つ目に選択した解法に対して、全前処理を選んだときの前処理IDの格納について説明している。

```
j = 0, 1, ..., 7
POST[precondID[1][j]] = 2階化されたjの値
→ (例: 00, 01, ..., 07)
```

このとき、precondID[1][0]=00, precondID[1][1]=01, ..., precondID[1][7]=07と全ての前処理のIDが代入されている。

【 図 1 6 】

図 16

(4) パラメタの値の受け取りとその値に応じた制御
SEL = POST[SEL]として、パラメタSELの値を受け取る
以下の1)~3)のときは、方法1である。

- SELの値が1または99のとき、グラフ描画の処理を実行する。
- SELの値が2または99のとき、データテーブルを作成し表示する処理を実行する。
- SELの値が3または99のとき、出力データ一覧を表示する処理を実行する。
- SELの値が90のとき、検索結果表示(方法2)を実行する。

1) 全解法のID(01~12)と全前処理のID(00~07)から、対応する求解アルゴリズム名を構成して、データ抽出するファイル名も図6のとおり構成していく(使うファイルは図7の「.log」拡張子のファイル)が、各々のファイル名を構成する度に以下の情報を設定していく。また、この処理では、逐次番号(解法IDと前処理IDが増えることの番号)を付けていく。

図7中の「.log」拡張子のファイル内の「CPU時間」、「収束所要回数」、「真の残差のノルム値」とパラメタ「criteron」とを比較評価(方法1と同じ)した結果と「求解状況」の情報に応じて、以下の表とおり設定する。

状況	パラメタ「stat」に設定する値	パラメタ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col (黄色)
反復回数の上限	mat. itr.	war_col (赤色)
ブレークダウン	brk. dwn	war_col (赤色)

これらと、アルゴリズム名(precondNMsolverNM)と逐次番号(num_sel)も配列に格納する。

```
INFO_dat[num_sel][num] = num_sel
INFO_dat[num_sel][alg] = precondNMsolverNM
INFO_dat[num_sel][itr] = 抽出した収束所要回数
INFO_dat[num_sel][cpu] = 抽出したCPU時間
INFO_dat[num_sel][rsd] = 抽出した真の残差ノルム
INFO_dat[num_sel][sta] = 抽出した求解状況
INFO_dat[num_sel][col] = col_stat
```

- 1)で得られた配列「INFO_dat」に対して、比較項目(sort_keyの値)に応じ、CPU時間が収束所要回数のどちらかで昇順にソート(phpの関数を使用)する。
- ソート後、col_statの値がNULLのレコード(収束したもの)を対象にして、最初のレコードの比較項目の値をパラメタ「TheFastest」に代入し、以後のレコードに対して、「TheFastest」比較項目の値を算出し、少数第1位までをパラメタ「Rat」に代入する。
- col_statの色ごとに、図23のデータテーブルをhtmlで表示する。このとき、対応する行のバックグラウンド色を「col_stat」の色で表示する。
- 収束したものについては、gnup lotを用いて比較項目のデータの棒グラフを描く。

【 図 1 7 】

図 17

(5) パラメタ[SEL]の値の受け取りとその値に応じた制御
SEL = POST[SEL]として、パラメタSELの値を受け取る

- SELの値が1または99のとき、グラフ描画の処理を実行する。
 - 下記の「loop_solpre」を実行する。
 - グラフにするデータは、図7中の「.rsd」拡張子のファイルをプロットする。
 - グラフ描画では、フリーソフトウェアのgnup lotを用いる。
 - gnup lotの制御は、本プログラムでgnup lotへの入力データ (gnup lot制御プログラム)を生成し実行する。
- SELの値が2または99のとき、データテーブルを作成し表示する処理を実行する。
 - 下記の「loop_solpre」を実行する。
 - CPU時間、収束所要回数などのデータを表の形式にする。これらのデータは、図7中の「.log」拡張子のファイルに格納されている。このファイルから該当データをphpの機能を用いて抽出する。
 - 図7中の「.log」拡張子のファイル内の「真の残差のノルム値」とパラメタ「criteron」とを比較評価した結果(真の残差のノルムがcriteron以下ならば、収束したと判定、criteronよりも大きい値ならば、見かけ上の収束と判定)と「求解状況」の情報に応じて、以下の表とおり設定する。

状況	パラメタ「stat」に設定する値	パラメタ「col_stat」
収束	conv.	無し
見かけ上の収束	no conv.	msg_col (黄色)
反復回数の上限	mat. itr.	war_col (赤色)
ブレークダウン	brk. dwn	war_col (赤色)

 - データテーブル表示のhtmlを作成するとき、上記の状況に応じて、対応する行のバックグラウンド色を「col_stat」の色で表示する。
- SELの値が3または99のとき、出力データ一覧を表示する処理を実行する。
 - 下記の「loop_solpre」を実行する。
 - 図7中の「.log」拡張子のファイルで該当するものを、全て画面に表示する。
- SELの値が90のとき、検索結果表示(方法2)を実行する。

【 図 1 8 】

図 18

loop_solpre プログラム

選択された全解法のIDを逐次選択する。

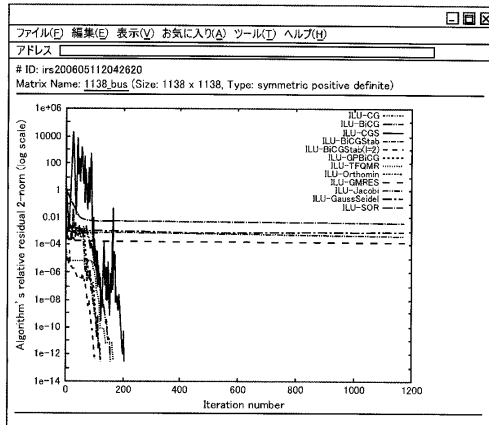
当該の解法と併用する前処理のIDを逐次選択する。

(A) 図6のとおり、ファイル名を構築する。

(B) 解法IDと前処理IDを、実際のアルゴリズム名に変換する。

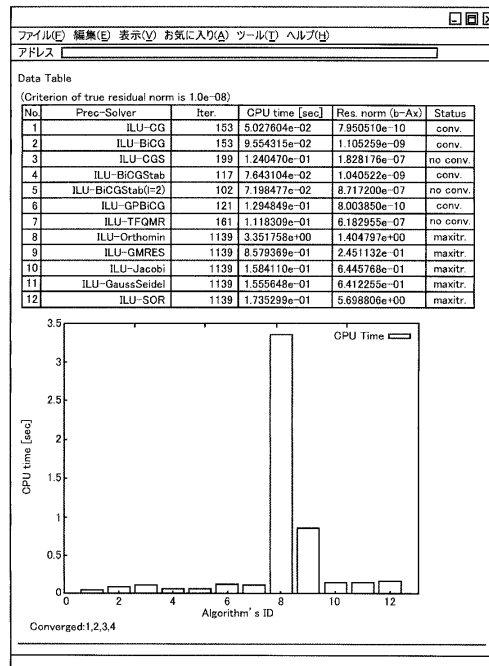
【 図 19 】

図 19



【 図 20 】

図 20



【 図 21 】

図 21

Output data

= Solver1 ID : 01 =
 Preconditioning1 ID : 02
 1138 x 1138 matrix 2596 entries
 SOLVER# CG
 PRECON# ILU
 Iis_solve is normal end
 # iter = 153 time = 5.027604e-02
 # (p=2.233839e-02 i=2.793765e-02)
 # Last Residual = 7.950510e-10

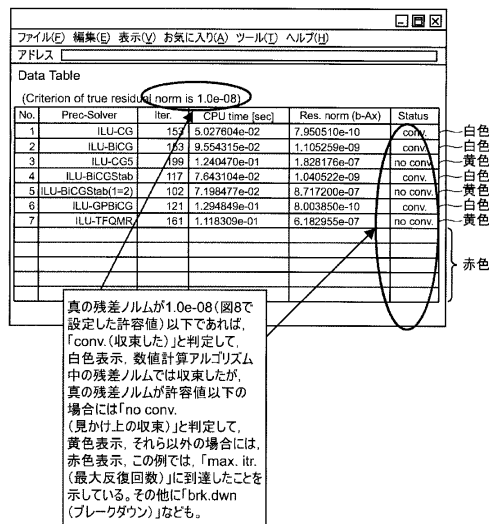
= Solver2 ID : 02 =
 Preconditioning2 ID : 02
 1138 x 1138 matrix 2596 entries
 SOLVER# BICG
 PRECON# ILU
 Iis_solve is normal end
 # iter = 153 time = 9.554315e-02
 # (p=4.294324e-02 i=5.259891e-02)
 # Last Residual = 1.105259e-09

= Solver3 ID : 03 =
 Preconditioning3 ID : 02
 1138 x 1138 matrix 2596 entries
 SOLVER# CGS
 PRECON# ILU
 Iis_solve is normal end
 # iter = 199 time = 1.240470e-01
 # (p=5.547853e-02 i=6.856751e-02)
 # Last Residual = 1.828176e-07

= Solver4 ID : 04 =
 Preconditioning4 ID : 02
 1138 x 1138 matrix 2596 entries
 SOLVER# BICGSTAB
 PRECON# ILU
 Iis_solve is normal end
 # iter = 117 time = 7.643104e-02
 # (p=1.481056e-03 i=7.494998e-02)
 # Last Residual = 1.040522e-09

【 図 22 】

図 22



真の残差ノルムが1.0e-08(図8で設定した許容値)以下であれば、「conv.(収束した)」と判定して、白色表示。数値計算アルゴリズム中の残差ノルムでは収束したが、真の残差ノルムが許容値以下の場合には「no conv.」(見かけ上の収束)と判定して、黄色表示、それ以外の場合には、赤色表示、この例では、「max. itr.(最大反復回数)」に到達したことを示している。その他に「brk.dwn(ブレークダウン)」なども。

白色
 白色
 黄色
 白色
 黄色
 白色
 黄色
 赤色

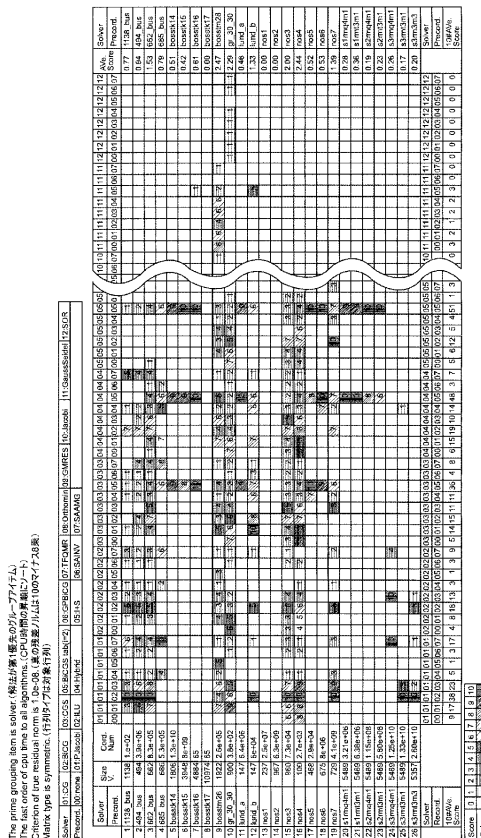
【 図 2 3 】

図 23

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)						
アドレス						
#ID:irs200605112101662						
Matrix Name : 1138 bus (Size: 1138 x 1138, Type:symmetric positive definite)						
The fast order of cpu time						
(Criterion of true residual norm is 1.0e-08)						
No.	Prec-Solver	Iter	CPU time [sec]	Rat.	Res. norm (b-Ax)	Status
1	ILU-CG	153	5.027604e-02	0.7	7.950510e-10	conv.
2	ILU-BICGStab	117	7.843104e-02	0.6	1.040522e-09	conv.
3	SAAMG-CG	29	8.977103e-02	0.6	4.941702e-10	conv.
4	ILU-BICG	153	9.554315e-02	0.5	1.105259e-09	conv.
5	SAAMG-BICGStab	16	9.731293e-02	0.5	8.540608e-10	conv.
6	SAAMG-GMRES	26	9.824681e-02	0.5	1.161343e-09	conv.
7	SAAMG-Orthpmin	29	1.215320e-01	0.4	8.167779e-10	conv.
8	ILU-GPBCG	121	1.294849e-01	0.4	8.030350e-10	conv.
9	SAAMG-BICG	29	1.614621e-01	0.3	5.941702e-10	conv.
10	SSOR-CG	513	1.631810e-01	0.3	3.27079e-10	conv.
11	SAAMG-CFBCG	16	1.732411e-01	0.3	3.18263e-10	conv.
12	PJacobi-CG	1032	1.866560e-01	0.3	1.42666e-09	conv.
13	SSOR-BICG	513	3.336132e-01	0.2	9.41184e-10	conv.
14	PJacobi-CGS	1016	3.526411e-01	0.1	1.06583e-09	conv.
15	PJacobi-BICG	1032	3.554239e-01	0.1	1.42666e-09	conv.
16	SAINV-CG	1032	4.350929e-01	0.1	1.42666e-09	conv.
17	I+S-CGS	983	4.513860e-01	0.1	1.01964e-09	conv.
18	I+S-B(CGStab)=2	888	4.594188e-01	0.1	7.20257e-13	conv.
19	Hybrid-BICGStab	57	4.618230e-01	0.1	1.06602e-09	conv.
20	Hybrid-CGS	51	4.952440e-01	0.1	1.03369e-09	conv.
21	Hybrid-CG	122	4.969899e-01	0.1	1.33457e-09	conv.
22	PJacobi-GPBCG	1040	5.686980e-01	0.1	1.38574e-09	conv.
23	SAINV-BICG	1032	6.925852e-01	0.1	1.42666e-09	conv.
24	SAINV-CGS	1016	7.095069e-01	0.1	1.16583e-09	conv.
25	Hybrid-GMRES	157	7.210549e-01	0.1	1.18628e-09	conv.
26	I+S-GPBCG	1016	7.972131e-01	0.1	1.39791e-12	conv.
27	Hybrid-BICG	122	9.894919e-01	0.1	1.33457e-09	conv.
28	SAINV-GPBCG	1040	1.125030e+00	0.0	1.38574e-09	conv.

【 図 2 4 】

図 24



フロントページの続き

(56)参考文献 特開2005-151539(JP,A)

伊藤・長谷川著,線形方程式求解アルゴリズムに対する性能情報検索システムの開発,先進的計算基盤システムシンポジウム SAC SIS 2006 論文集,日本,社団法人情報処理学会,2006年 5月22日,第2006巻、第5号,第236-237頁

(58)調査した分野(Int.Cl.,DB名)

G06F 11/36