

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4576535号
(P4576535)

(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 I
G06F 17/12 (2006.01) G O 6 F 17/12
G06F 11/36 (2006.01) G O 6 F 9/06 6 2 O R

請求項の数 5 (全 24 頁)

<p>(21) 出願番号 特願2006-102171 (P2006-102171) (22) 出願日 平成18年4月3日(2006.4.3) (65) 公開番号 特開2007-279836 (P2007-279836A) (43) 公開日 平成19年10月25日(2007.10.25) 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 多胡 滋</p>
---	--

最終頁に続く

(54) 【発明の名称】 収束求解アルゴリズム性能表示装置および収束求解アルゴリズム性能表示方法

(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、
収束状況を示す収束グラフ、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、前記表示項目から収束グラフ、収束への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パラメータ項目の組み合わせを形成し、各組み合わせでの収束グラフデータの取得、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、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
各組み合わせの演算処理結果を受け取り、各組み合わせの演算処理結果を画面对比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。

10

20

【請求項 2】

請求項 1 において、前記結果画面表示手段は、画面に演算処理された収束グラフデータを用いて複数の収束グラフを同時に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装置。

【請求項 3】

請求項 1 において、前記結果画面表示手段は、画面に複数の CPU 時間もしくは収束所要回数を同時に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装置。

【請求項 4】

請求項 1 から 3 のいずれかにおいて、前記設定画面表示手段は、通信手段を介してのクライアント端末からの指令信号を参照して表示項目についてパラメータ項目として選択させ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。

10

【請求項 5】

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、

前記設定画面表示手段が、収束グラフ、収束への CPU 時間及び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し、前記表示項目から収束グラフ、収束への 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し、

前記演算処理手段が、パラメータ項目を受け取り、解法と解法に併用された技法のパラメータ項目の組み合わせを形成し、各組み合わせでの収束グラフデータの取得、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、データベースに演算結果を格納させ、

20

前記結果画面表示手段が、各組み合わせの演算処理結果を受け取り、各組み合わせの演算処理結果を画面对比表示すること

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表示装置およびこの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関する。

30

【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しばしばある。

【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。

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。

40

【0004】

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。

【0005】

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

50

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。

【0006】

【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】

【0007】

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

$$Ax = b$$

の求解に帰着される。

【0008】

また、固有値問題の場合には次の方程式

$$Ax = Bx$$

の求解に帰着される。

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。

【0009】

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。

【0010】

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。

【0011】

係数行列Aが対称正定値であれば、迷わずCG法やCholesky法が選択されるが、係数行列が非対象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。

【0012】

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、速い方がよい、許容精度内であれば十分とする。

【0013】

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。

【0014】

本発明は、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表示装置あるいはこの収束求解アルゴリズム性能表示装置を用いた収束求解アルゴリズム性能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】

【0015】

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リング、係数行列や行や列の順番を入れ替えるオーダリ

10

20

30

40

50

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、

収束状況を示す収束グラフ、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、前記表示項目から収束グラフ、収束への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パラメータ項目の組み合わせを形成し、各組み合わせでの収束グラフデータの取得、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、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

各組み合わせの演算処理結果を受け取り、各組み合わせの演算処理結果を画面对比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。

【0016】

本発明は、また、前記結果画面表示手段は、画面に演算処理された収束グラフデータを用いて複数の収束グラフを同時に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装置を提供する。

【0017】

本発明は、また、前記結果画面表示手段は、画面に複数のCPU時間もしくは収束所要回数を同時に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装置を提供する。

【0018】

本発明は、また、前記設定画面表示手段は、通信手段を介してのクライアント端末からの指令信号を参照して表示項目についてパラメータ項目として選択させ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。

【0019】

本発明は、また、前記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、

前記設定画面表示手段が、収束グラフ、収束へのCPU時間及び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し、前記表示項目から収束グラフ、収束への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し、

前記演算処理手段が、パラメータ項目を受け取り、解法と解法に併用された技法のパラメータ項目の組み合わせを形成し、各組み合わせでの収束グラフデータの取得、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、データベースに演算結果を格納させ、

前記結果画面表示手段が、各組み合わせの演算処理結果を受け取り、各組み合わせの演算処理結果を画面对比表示すること

を特徴とする収束求解アルゴリズム性能表示方法を提供する。

【0020】

また、本発明は、記憶手段によって、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、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形式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するコンピュータプログラムを格納し、

演算処理手段によって、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収束グラフ、CPU時間および

10

20

30

40

50

収束所要回数について演算処理し、

データベースに、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収束グラフ、CPU時間および収束所要回数についてのデータを格納し、

設定画面表示手段によって、各パラメータの表示手段と各パラメータの選択手段、前記収束グラフ、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し、

データ取得処理手段によって、前記画面上で、パラメータのいずれか、前記収束グラフ、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つまたは複数を選択すると、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、取得する処理を行い、および

結果画面表示手段によって、該データ取得処理手段で取得したデータに基づいて前記収束グラフ、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表示し、

前記設定画面表示手段の画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すること

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方法を提供する。

【発明の効果】

【0021】

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、速く求解することのできる収束求解アルゴリズム性能表示装置、あるいはこの収束求解アルゴリズム性能表示装置を用いた収束求解アルゴリズム性能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】

【0022】

以下、本発明の実施例を図面に基づいて説明する。

【実施例】

【0023】

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表示装置の構成を示すブロック図である。

図1において、収束求解アルゴリズム性能表示装置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。

【0024】

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、例えば固有値問題($Ax = Bx$)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、上記以外の収束求解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 2 2 の項目としては、例えば行列、右辺項、初期値、他で構成され、表現され、解法 2 3 の項目としては、例えば直接法、定常反復法、非定常反復法、他で構成され、表現され、解法と併用する技法 2 4 の項目としては、例えば前処理、スケーリング、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 2 2 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 2 3 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 2 4 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。

【 0 0 2 5 】

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。

10

【 0 0 2 6 】

「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 3 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 、 q_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。

【 0 0 2 7 】

20

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 2 8 】

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。

【 0 0 2 9 】

30

ここで、 ϵ 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 0 】

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 1 】

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 4, 5 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、真の残差で評価する。

40

【 0 0 3 2 】

真の残差とは、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、

50

$$\|b - Ax\|$$

を評価したものである。図14の「Data Table」内の「Res. norm (b - Ax)」は、この情報をあらわしている。

【0033】

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。

【0034】

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。

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

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

【0035】

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、bcstkt14、bcstkt15、bcstkt16、bcstktm26、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)

【0036】

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)。

【0037】

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

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

【0038】

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。

【 0 0 3 9 】

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

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

CG [0 1]、BiCG [0 2]、CGS [0 3]、BiCGStab [0 4]、BiCGStab (l = 2) [0 5]、GPBiCG [0 6]、Orthomin [0 7]、GMRES [0 8]、TFQMR [0 9]、Jacobi [1 0]、Gauss-Seidel [1 1]、SOR [1 2]

現在、BiCGStab (l) 法のパラメータ l の値は、l = 2 としている。

【 0 0 4 0 】

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。

スケーリング：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 】

Lis の仕様として、前処理 (preconditioner) には下記のとおり precondition ID が付けられている。 [] 内の 2 桁の数値が precondition ID である。

none [0 0] (前処理なし)、PJacobi [0 1]、ILU [0 2]、SSOR [0 3]、Hybrid [0 4]、I + S [0 5]、SAINV [0 6]、SAMG [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。

【 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 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 2 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 add 3 2 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。

【 0 0 4 5 】

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

10

20

30

40

50

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。

【0046】

5) 入力データファイル

2) 項「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。

【0047】

6) 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用いることができる。

【0048】

このように、演算処理手段である演算処理手段(1)12に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。

【0049】

提供側計算機システム1で生成された諸計算結果のファイルが格納される。また、クライアント端末（表示装置）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を経由してデータベース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。

【0050】

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。

【0051】

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は図4および図5を参照して説明する。

【0052】

図4にデータベース13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(2)12での演算処理34がなされる。図4に記載された事項をそのまま次に示す。

解くべき問題の設定

(内部では行列名で表現。行列名の例：add32)

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。

138__bus, 494__bus, 662__bus, 685__bus, add20, add32, bcsstk14, bcsstk15, bcsstk16, bcsstm26, 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)

【0053】

解法の設定

(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

(内部では技法のIDで表現。技法IDの例：PJacob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。

10

20

30

40

50

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。

【0054】

各パラメータへの値の設定

`matrix = add32, solverID = 04, preconditionID = 01`

検索対象であるファイル名の構成

`matrix - solverIDpreconditionID0000.xxx`

(具体例: `add32 - 040100000.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。

【0055】

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。

【0056】

図5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5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時間が格納される。

このように、データベース13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についてのデータが格納される。

【0057】

次に提供側計算機システム2について説明する。

演算処理手段(2)14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の作成を行う。

【0058】

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。

【0059】

図6は、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、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、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の表示仕様に従っている。図中の((1))~((5))は実行手順でもあり、それらの内容は以下のとおりである。なお、(())は図中では丸で示してある。

【0060】

((1))本システムにより処理された結果の表示項目を選択する。

((2))数値計算したい問題をプルダウン形式のメニューから選択する(図6では、線形方程式の係数行列で動作仕様ファクターとして「add32」という行列を選択した)。また、この例では、右辺項と初期値は、使用した数値計算ライブラリlisのデフォルト値を使用した。

((3))解法をプルダウン形式のメニューから選択する(図6では、「CG」と「BiCGStab」という解法を選択しているところ)。ただし、一番上の欄だけは全解法を選択し、同一の技法(次の((4))参照)を併用する。

((4))解法と併用する技法をチェックボックスの中から選択する(図6では、CG法に対して「PJacobi」と「SSOR」、BiCGStab法に対して

10

20

30

40

50

「none (前処理無し)」と「ILU」という、各々2種類の前処理技法を選択した)。

ただし、一番上の欄だけは、上述のとおり全解法に対して1種類の前処理のみを選択する。

((5))データの検索と結果表示を依頼するためのボタン。

【0061】

図6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

【0062】

この設定で実行した結果表示の例を図7に示す。

上述した((1))~((2))の詳細は次のとおりであり、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。

【0063】

図7および図8に記載されている事項を以下に示す。

((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。

20

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

【0064】

((3))解法(図8)

図6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], Orthomin[07], GMRES[08], TFQMR[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6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

【0065】

50

((4)) 前処理 (解法と併用する技法) (図 8)

図 6 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 6 6 】

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「I L U」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 6 7 】

((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 2 0 0 (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。

【 0 0 6 8 】

図 6 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 (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。

((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 6 9 】

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「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 9 - 図 1 2 で示す。図 9 - 図 1 2 に記載されている事項を以下に示す。

10

20

30

40

50

【 0 0 7 0 】

図 9 について：

(1) パラメータ「matrix」の値の受け取り

matrix = POST[matrix]

例：図 6 の場合は、この受け取りの後、matrix = add32 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 6 で複数選択された解法の順番）を示すパラメータである。

10

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 6 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。

【 0 0 7 1 】

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。

20

【 0 0 7 2 】

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)

$j = 1, 2, \dots, 12$

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。

【 0 0 7 3 】

図 10 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。

【 0 0 7 4 】

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。

【 0 0 7 5 】

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

50

るが、実質的には1番目を指している。

【0076】

図11について：

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

前処理のIDに「99」が存在しないとき (図6の選択のとき)：

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「PJacobi」と「SSOR」を選択

$\text{precondID}[1][0] = \text{POST}[\text{precondID}[1][0]]$,

$\text{precondID}[1][1] = \text{POST}[\text{precondID}[1][1]]$,

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

10

【0077】

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

$\text{precondID}[2][0] = \text{POST}[\text{precondID}[2][0]]$,

$\text{precondID}[2][1] = \text{POST}[\text{precondID}[2][1]]$,

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

20

【0078】

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。

【0079】

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。

【0080】

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

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

(例：00, 01, ..., 07)

30

このとき、

$\text{precondID}[1][0] = 00$, $\text{precondID}[1][1] = 01$, ... ,

$\text{precondID}[1][7] = 07$

と全ての前処理のIDが代入されている。

【0081】

図12について：

(4) パラメータ「SEL」の値の受け取りとその値に応じた制御

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

【0082】

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。

40

1) 図13の「loop_solpre」を実行する。

2) グラフにするデータは、図4, 5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。

【0083】

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) 図13の「loop_solpre」を実行する。

50

- 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4, 5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。

【0084】

- 3) 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。
- 1) 図13の「loop_solpre」を実行する。
 - 2) ファイル図4, 5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。

【0085】

図13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13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。

【0086】

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4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6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。

【0087】

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た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。

【0088】

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。

【0089】

次に、結果表示手段15について説明する。

図14は検索結果表示例を示す。

図14は、図6のとおり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。収束グラフおよび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。参考までに、図6の((1))で「収束グラフ」のみを選択した場合には、図14の収束グラフより上部のみが表示され、「Data Table」以下は表示されない。

【0090】

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。

【0091】

「Data Table」の表の項目は次のとおりである。

No . 選択した収束求解アルゴリズムに付した番号
 Prec - Solver 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。

【 0 0 9 2 】

Iter . 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
 までにかかった反復回数。収束求解アルゴリズムの残差ベクトルに
 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1.0e-12）で判定した。

【 0 0 9 3 】

Cpu time 収束までに要したCPU時間[単位：秒]を表す。 10
 Res . norm 真の残差の値。収束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($b - Ax$)は、収束には程遠い値である場合もあ
 る。

【 0 0 9 4 】

Status 真の残差の値から収束した(conv.)か、見かけ上の収束(n
 o conv.)か、全く収束していない(「max.itr.(
 反復回数の上限に到達)」「brk.dwn.(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)」
)かを表示する。 20

【 0 0 9 5 】

図15は、図14から続く画面である（画面スクロールしたところ、冒頭の表は重複し
 て表示している）。

図15において、データテーブルとCPU時間を示す棒グラフが示される。図6の（（
 1））で「データテーブル」のみを選択した場合には、図15の表と棒グラフのみが表示
 される。

Data Tableの番号を横軸にして、各々のCPU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「Converged: 1, 2, 3, 4」の表示は、収束した数
 値計算アルゴリズムの番号を表示している。 30

【 0 0 9 6 】

図16は、図15からさらに続く画面（画面スクロールしたところ）、これは、図4、
 5中の拡張子「.log」のファイルの内容(Output data)をそのまま表示
 したものである。

【 0 0 9 7 】

図6の（（1））で「出力データ」のみを選択した場合には、図16に示す「Outp
 ut data」のみが表示される。

結果画面表示手段15には、パラメータSEL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phpを用い、画面表示用
 にはhtmlの仕様に従った出力を行っている。 40

【 0 0 9 8 】

データベース13の中から、反復解法の収束の様子を表す残差のデータ（図5では「a
 dd32-04010000.rsd」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gnuplotを用いている。gnuplotを実行するのに必要な入力
 ファイルもphp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 0 9 9 】

CPU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5では、「add32-04010000.log」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。 50

【 0 1 0 0 】

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、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 0 1 】

次に図 1 について、クライアント端末 2 0 0 について説明する。クライアント端末 2 0 0 は、表示装置 5 1 を備える。表示装置 5 1 は、問題の表示 5 2、パラメータ項目選択 5 3、検索結果表示 5 4、および、より妥当な収束求解アルゴリズムの選定表示 5 5 を行う。

10

【 0 1 0 2 】

図 2 に示すように、クライアント端末 2 0 0、すなわち表示装置 5 1 は各パラメータの動作仕様ファクターの設定 6 1 を行い、検索実行指示 6 2 によって検索要求を提供側計算機システム 2 に行い、および返却された結果を検索結果として表示 6 3 (検索結果表示) を行う。

【 0 1 0 3 】

1) 問題の表示

自然現象や工学現象の解明では、収束求解シミュレーションを用いた解析が盛んである。そ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収束求解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。線形方程式以外の収束求解においても同じような状況である。従って、実際の収束求解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 2 0 0 に問題の表示を行うことになる。

20

2) パラメータ項目他選択：

図 6 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。

3) 検索結果表示：

図 1 4、図 1 5、図 1 6 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 5 1 の画面上に表示される。

30

4) 収束求解アルゴリズムの再選定：

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。

【 0 1 0 4 】

図 2 において、提供側計算機システム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 7 1 が行われ、該当ファイルを検索 7 2 してデータベース 1 3 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 7 3 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、検索結果表示 6 3 がなされることになる。

40

【 0 1 0 5 】

このように、収束求解アルゴリズム性能表示装置 1 0 0 は、設定画面表示手段の画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。

以上のように構成される収束求解アルゴリズム性能表示装置によって以下の収束求解アルゴリズム性能表示方法が構成される。

【 0 1 0 6 】

記憶手段 1 1 によって、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、収

50

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形式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するコンピュータプログラムを格納すること。

【0107】

演算処理手段(1)12によって、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すること。

【0108】

データベース13に、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収束グラフ、CPU時間および収束所要回数についてのデータを格納すること。

10

【0109】

演算処理手段(2)14の設定画面表示手段によって、各パラメータの表示手段と各パラメータの選択手段、前記収束グラフ、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。

【0110】

演算処理手段(2)14のデータ取得処理手段によって、前記画面上で、パラメータのいずれか、前記収束グラフ、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

【0111】

結果画面表示手段15によって、該データ取得処理手段で取得したデータに基づいて前記収束グラフ、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を表示すること。

【0112】

そして、以上の構成によって、前記設定画面表示手段の画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すること。

30

【図面の簡単な説明】

【0113】

【図1】本発明の実施例の構成を示すブロック図。

【図2】図1の一部詳細を示すフローチャート図。

【図3】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4】データベースに載せる際の情報提供状況を示す図。

【図5】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。

40

【図6】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・表示依頼の画面を示す図。

【図7】図6の画面にて((1))と((2))の値をパラメータに代入する例。

【図8】図6の画面にて((3))~((4))の値をパラメータに代入する例および((5))の説明。

【図9】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10】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11】(2)において、解法のIDに「99」が存在しないときの説明図。

50

- 【図12】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。
- 【図13】検索対象であるファイル名を構成する処理の図。
- 【図14】結果表示例図。
- 【図15】図14から続く画面を示す図。
- 【図16】図15から続く画面を示す図。
- 【符号の説明】
- 【0114】

11...記憶手段、12...演算処理手段(1)、13...データベース、14...演算処理手段(2)、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表示装置、200...クライアント端末。

10

【図1】

【図2】

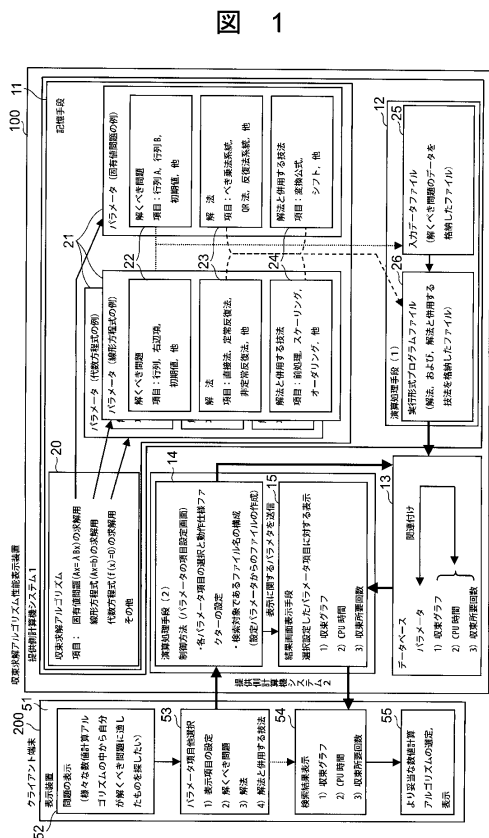
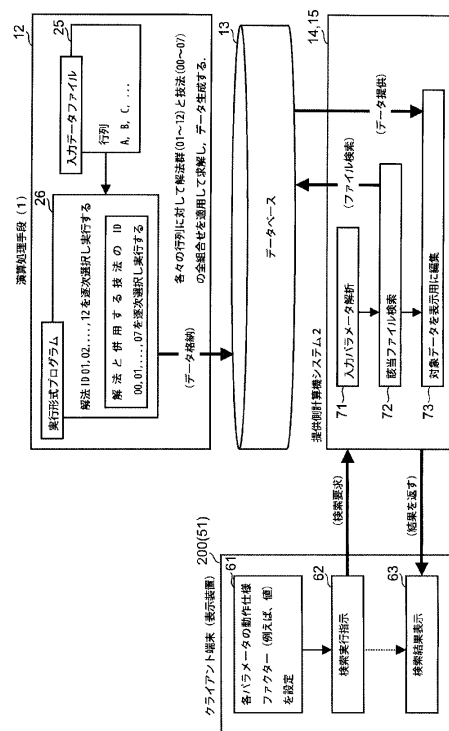


図 1

図 2



100 11 記憶手段
12 演算処理手段(1)
13 データベース
14 演算処理手段(2)
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表示装置
200 クライアント端末

【図3】

図 3

x_0 : 適当な初期値を与える.
 $f_0 = b - Ax_0, D_0 = K^T f_0$
 for $k = 0, 1, \dots$, until $\| r_k \| \leq \epsilon \| b \|$ do
 begin

$$\alpha_k = \frac{(K^T r_k)_k}{(P_k A P_k)_k}$$

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

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

$$\beta_k = \frac{(K^T r_{k+1})_{k+1}}{(K^T r_k)_k}$$

$$D_{k+1} = [K^T r_{k+1}] + \beta_k D_k$$
 end

【図4】

図 4

31 解くべき問題の設定
 (内部では行列名で表現。行列名の例: add32)
 ※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ある。
 1138_bus, 494_bus, 662_bus, 685_bus, add20, add32, bcscstk14, bcscstk15, bcscstk16, bcscstk26, 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)

32 解法の設定
 (内部では解法の ID で表現。解法 ID の例: B1CGstab ならば 04)
 ※現状では、解法は 12 種類ある

33 解法と併用する技法の設定
 (内部では技法の ID で表現。技法 ID の例: PJacobi ならば 01)
 ※現状では、この技法 (前処理) は 8 種類ある

34 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

各パラメタへの値の設定 matrix=add32, solverID=04, preconditionID=01
 検索対象であるファイル名の構成 matrix=solverIDpreconditionID0000.xxx
 (具体例: add32-04010000.xxx)

ここで、xxx はファイル種別を表す拡張子。
 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。

上記の例では、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としている。この数は限定されていない。

【図5】

図 5

add32-04010000_sol

```

1 1.000000000000011701751e+00
2 1.00000000000000910383e+00
3 1.00000000000001909584e+00
4 1.0000000000000198540e+00
5 1.0000000000000355271e+00
6 9.9999999999997557500e-01
7 9.999999999998183112027e-01
... 以下省略 ...

```

add32-04010000_rsd

```

1.251841e-01
7.953604e-02
3.740018e-02
2.781561e-02
1.992116e-02
8.839245e-03
7.883846e-03
5.811023e-03
... 以下省略 ...

```

add32-04010000_log

```

4960 x 4960 matrix 23884 entries
SOLVER# B1CGSTAB
PRECON# Jeonbi
lis_solve is normal end
# iter = 06 time = 2.031096e-01
# (r=2.004862e-03 r=2.011020e-01)
# Last Residual = 9.641893e-13

```

収束所要回数

CPU 時間

【図6】

図 6

ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
 アドレス

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) 以下のパラメー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) ①
 全ての結果 収束グラフ データテーブル 出力カード

行列の選択: [add32] ②

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 type="checkbox"/> All solver	<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択する)	<input type="radio"/> SSOR <input type="radio"/> Hybrid <input type="radio"/> I+S
	<input type="radio"/> SAINV <input type="radio"/> SAAMG
CG	<input type="checkbox"/> All <input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input checked="" type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
BCG	<input type="checkbox"/> All <input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
CGS	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
BCGStab	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
BiCGStab(L)	<input type="checkbox"/> All <input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
Gpis	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
Orthomin	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
GMRES	<input type="checkbox"/> All <input type="checkbox"/> none <input type="checkbox"/> PJacobi <input type="checkbox"/> ILU
TFQMR	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
JACOBI	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
GaussSeidel	
SOR	

入力欄は複数
用意されている
(一番上の欄
だけ、選択の
要領が異なる)

【 図 7 】

図 7

(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]

(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	add32	
右辺項と初期値	1138 bus	差を1.0として作成した(Lisのデフォルト)作成した(Lisのデフォルト)
-右辺項:解	494 bus	
-初期値:全	682 bus	
	685 bus	
	add20	
	add32	

解法 (空白は無視される)	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="checkbox"/> PJacobi <input type="checkbox"/> ILU
(あるいは、下記からいくつかの解法を選択する)	<input type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
<input type="checkbox"/> CG	<input type="checkbox"/> none <input checked="" type="checkbox"/> PJacobi <input type="checkbox"/> ILU
	<input checked="" type="checkbox"/> SSOR <input type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> SAAMG
<input checked="" type="checkbox"/> BiCGStab	<input checked="" type="checkbox"/> none <input type="checkbox"/> PJacobi <input checked="" 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 8 】

図 8

(3) 解法
 図6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], Orthomin [07], GMRES [08], TFQMR [09], Jacobi [10], Gauss-Seidel [11], SOR [12]
 ※ 現在、BiCGStab(1)法のパラメタ1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6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

(4) 前処理 (解法と併用する技法)
 図6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 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 [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「PJacobi」と「SSOR」を選択 → precondID[1][0]=01, precondID[1][1]=03
 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]=00, precondID[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 9 】

図 9

(1) パラメタ「matrix」の値の受け取り
 matrix = POST[matrix]
 例: 図6の場合は、この受け取りの後、matrix = add32 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」以外の場合:
 (このとき、solverID[0]=0と代入されている)
 例: 図6の画面のとおり入力だと、
 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。
 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。
 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="checkbox"/> PJacobi <input type="checkbox"/> ILU
(あるいは、下記からいくつかの解法を選択する)	<input type="checkbox"/> SSOR <input checked="" type="checkbox"/> Hybrid <input type="checkbox"/> I+S
	<input type="checkbox"/> SAINV <input type="checkbox"/> 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 10 】

図 10

(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11に続く。

【 図 1 1 】

図 11

図10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6の選択のとき) :

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「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は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つ目を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倍化された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 2 】

図 12

(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。

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) 図13の「loop_solpre」を実行する。
2) グラフにするデータは、図4, 5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) 図13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4, 5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) 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。

```
1) 図13の「loop_solpre」を実行する。
2) ファイル図4, 5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。
```

【 図 1 3 】

図 13

loop_solpre プログラム

選択された全解法のIDを逐次選択する。

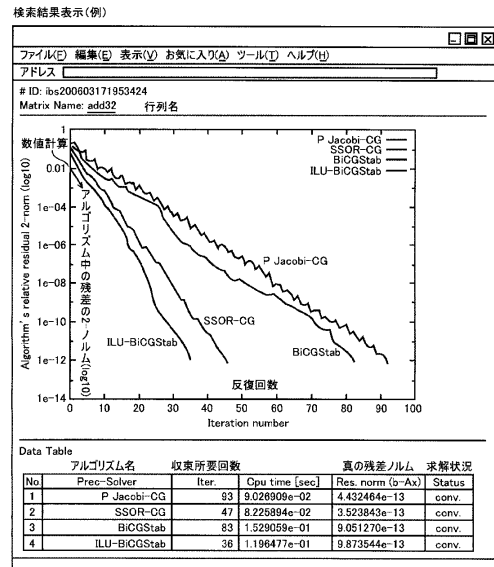
当該の解法と併用する前処理のIDを逐次選択する。

(A) 図4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。

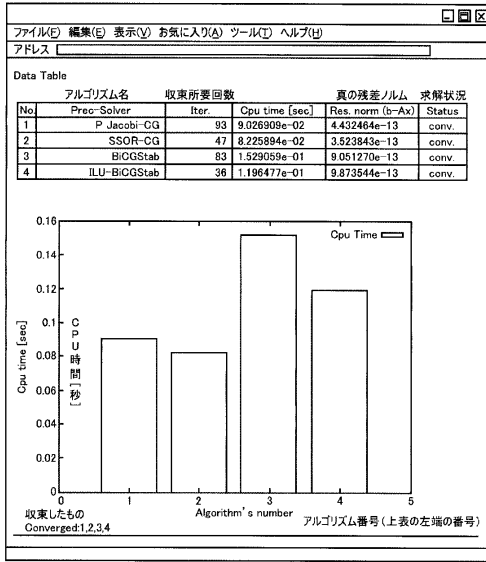
【 図 1 4 】

図 14



【 図 1 5 】

図 15



【 図 1 6 】

図 16

```

Output data
= Solver1 ID : 01 =
Preconditioning1 ID : 01
4960 x 4960 matrix 23884 entries
SOLVER# CG
PRECON# Jacobi
lis_solve is normal end
# iter = 93 time = 9.026303e-02
# (p=-6.455183e-03 i=-8.381301e-02)
# Last Residual = 4.432464e-13

Preconditioning1 ID : 03
4960 x 4960 matrix 23884 entries
SOLVER# CG
PRECON# SSOR
lis_solve is normal end
# iter = 47 time = 8.225834e-02
# (p=-4.020357e-02 i=-4.205537e-02)
# Last Residual = 3.523843e-13

= Solver2 ID : 04 =
Preconditioning2 ID : 00
4960 x 4960 matrix 23884 entries
SOLVER# BICGSTAB
PRECON# None
lis_solve is normal end
# iter = 93 time = 1.529053e-01
# (p=-4.410744e-03 i=-1.528618e-01)
# Last Residual = 3.051270e-13

Preconditioning2 ID : 02
4960 x 4960 matrix 23884 entries
SOLVER# BICGSTAB
PRECON# ILU
lis_solve is normal end
# iter = 36 time = 1.196477e-01
# (p=-6.889820e-03 i=-1.127573e-01)
# Last Residual = 9.873544e-13

= Solver3 ID : 00 =
= Solver4 ID : 00 =
= Solver5 ID : 00 =

```

フロントページの続き

- (56)参考文献 特開2000-298665(JP,A)
特開平11-242664(JP,A)
特開平09-034944(JP,A)
特開平05-135091(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 17/12
G06F 11/36