

## 数値シミュレーションの可視化診断システム

梅谷 征雄  
日立製作所 中央研究所

偏微分方程式を用いて定式化される数値シミュレーションの過程に出現する様々な症状を、視覚的な手段を用いて系統的に診断する方法を提案する。この方法は症状の原因を、パターンマッチの手法を一部用いて、離散化方程式の係数行列から得られる固有ベクトルないし行列列ベクトルの性質に帰着させるものである。その過程で固有ベクトルの線形変換による不変性を利用する。またベクトル値を画面上に等高線で表現し、人間の認知能力を生かした重ね合わせによる比較を行なう。この方法による可視化診断機能を偏微分ソルバ DEQSOL に組み込み、風速場内の温度分布問題や拡散問題など数例の診断に適用してその有効性を確認した。

## The Visual Diagnosis on Numerical Calculation of PDE Problems and Experiments

Yukio Umetani  
Hitachi Central Research Laboratory  
Kokubunji, Tokyo 185, Japan  
E-mail: umetani@crl.hitachi.co.jp

A systematic method to diagnose the causes of symptom found in PDE solving process by visual means is proposed and tested. This method diagnoses the causes of the symptom by reducing them to the attribute of dominant eigen vectors or to the column vectors of matrix of discretized equation. This method uses the shape-preserving nature of eigenvectors in linear transformation. Also contour map representation of vector values is utilized to aid this reduction process by human cognitive capability. This method was incorporated into the high level PDE solver DEQSOL. The results of application to several PDE problems show the feasibility of this method.

# 1 はじめに

強力な画面表示機能を持つワークステーションの登場により、シミュレーションのプリ/ポスト部のマンマシン部のインターフェースは改善されたが、計算本体部分への取り組みは立ち後れているように見受けられる。拡散や流体、電磁場など偏微分系のシミュレーションにおいても、その過程で数値解析者は解の振動や発散から疑わしき結果に至るまで様々な数値異常を体験するのが常であり、このような局面に画面機能を役立てることが本研究の目的である。数値異常の原因は、現象や数学モデルの不安定性、離散化やアルゴリズムの不適切、ソルバの誤使用から単純な入力ミスまで様々あり、それに応じて多様な観点からのトラブルシューティングが必要になる。ソルバの頑固さを高めたりガードを堅くしてこのような数値異常を起き難くするのが伝統的なアプローチであるが、特に動的な変化の大きい問題に対しては限界があり、また過剰な頑固さやガードは性能と適用範囲を犠牲にするので経済性の観点からトレードオフが必要になる。本手法はボトムアップのアプローチを取り、原因は何んであれ異常が起きることを前提とし、そこから画面機能を使って効率良く原因に遡及する方法を追求した。この方法はまた、数理的な事象を感覚的に把握したり、逆に感覚で把握できない数理事象を明らかにするのも役立つことを期待している。なお計算過程の可視化に関して藤野氏の研究 ([2]) があり参考にさせて頂いた。

## 2 可視化診断方法

偏微分方程式系の数値シミュレーションの核は、離散化の結果得られる大規模な線形方程式の求解にある。図 1 はその局面でのデータの関係を示す。離散化行列  $A$ 、解空間ベクトル  $x$ 、像空間ベクトル  $y$  が主なデータである。それらは図の矢印に示すような関係、(1)  $y = Ax$ 、(2)  $x = A^{-1}y$  で結ばれている。離散化行列は (3) に示すように、数学モデルと離散化方式から決められる。解空間ベクトルの一つである RHS (右辺) ベクトルも (3)' に示すように数学モデルと離散化行列から決められる。

診断すなわち原因遡及の容易さは、各矢印の先から元に向かって物理空間上でローカルな対応が保たれる程度に掛かっている。その点で (3)、(3)' は微分が近隣のメッシュ点の値のみを使うことから最も容易である。矢印 (1) もローカルな対応を保つ。y ベクトルは A 行列を構成する列ベクトルの 1 次結合からなり、微分演算

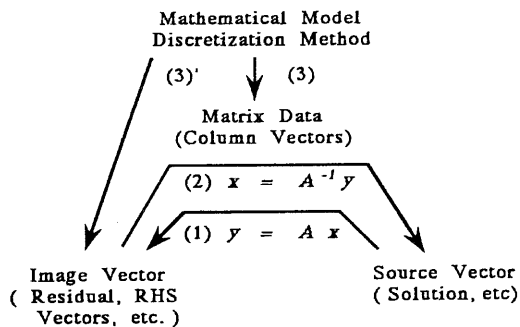


Figure 1: 偏微分方程式求解過程における情報の関係

の離散化の性質から  $A$  は対角要素の周辺にのみ非ゼロ要素を持つ疎行列である。従ってそれらの列ベクトルも疎なベクトルであり、y ベクトル空間の如何なる症例も、その原因は症例の近辺に対角値をもつ列ベクトルの性質に帰着される。そして列ベクトルの性質は空間的ローカリティを保ちながら矢印 (3) により数学モデルと離散化方式に帰着できる。

矢印 (2) 上の原因遡及はもっと困難である。 $A^{-1}$  はもはや疎行列でないので矢印 (1) の議論は通用しない。これに対しては固有ベクトルの性質を使うことができる。固有ベクトルは線形変換により方向を変えないのでベクトルを等高線で表示した時に変換の前後で似た紋様を描くことが期待できる。特に固有値が 1 に近い時は両者はほとんど一致するはずである。そこで、 $x$  空間の症例に近い固有ベクトルを見つけられればそれは  $y$  空間でも見つかるか、あるいはその固有ベクトルを形成する A 行列の列ベクトルの性質に帰着させることが可能である。その後は矢印 (3)' ないし (3) を使っていずれの場合も数学モデルと離散化方式に帰着できる。A が非対称行列の時は一般に固有ベクトルに複素値が現れ、等高線による表示が難しくなるが、この場合は固有ベクトルの実部と虚部で代行する。A が実行列ならば固有値、固有ベクトルに共役関係があることから、これらで任意のベクトルを展開できる。A の非対称性が弱い時にはこれらはほとんど固有ベクトルと同じ性質を持つ。固有ベクトルを用いる方法は実は矢印 (1) 上の遡及にも使うことができる。

以上のスケッチに基づく遡及方式は次の通りである。

## 2.1 矢印 (1) 上の遡及

行列  $A$  の  $j$  番目の列ベクトルを  $a_j$ 、ベクトル  $x$  の第  $j$  要素を  $x_j$  とすると

$$y = Ax = a_1x_1 + a_2x_2 + \dots + a_jx_j + \dots + a_nx_n \quad (1)$$

が成り立つ。与えられた  $y$  に対して各列ベクトルの寄与を求めるには次の連立一次方程式を解けば良い。

$$(a_1, a_2, \dots, a_j, \dots, a_n) x = y \quad (2)$$

または

$$((a_1, a_i), (a_2, a_i), \dots, (a_j, a_i), \dots, (a_n, a_i)) x = (y, a_i), \quad i = 1, n \quad (3)$$

ここで、 $(a_j, a_i)$  は  $a_j$  と  $a_i$  の内積を意味する。一般に式 (3) の方が式 (2) よりも解きやすい。

$y$  空間内の症例がローカルなときは、次のような最小自乗法によるフィッティングを行なうことができる。この方が (2) または (3) を解くよりも経済的である。

$L$  を指定されたローカルな領域に属する離散点 (節点) の集合とし、 $A^l, a_j^l$  を  $L$  に属する行ないし要素のみからなる  $A, a_j$  の部分行列ないし射影ベクトルとする。また  $y^l$  を  $y$  の  $L$  への射影ベクトルとする。このとき、次のように  $L$  上の残差ノルムの最小化を試みる。

$$\begin{aligned} R^l &= (y^l - A^l x, y^l - A^l x) \\ &= (y^l, y^l) - 2(y^l, A^l x) + (A^l x, A^l x) \\ &= (y^l, y^l) - 2(A^{lt} y^l, x) + (A^l x, A^l x) \end{aligned} \quad (4)$$

( $A^{lt}$  is the transpose of  $A^l$ )

従って

$$\partial R^l / \partial x = -2A^{lt} y^l + 2A^{lt} A^l x \quad (5)$$

最小点で  $\partial R^l / \partial x = 0$  より

$$A^{lt} A^l x = A^{lt} y^l \quad (6)$$

が得られる。 $A^{lt} = (a_1^l, a_2^l, \dots, a_j^l, \dots, a_n^l)$  より (6) は次のように書ける。

$$((a_1^l, a_i^l), (a_2^l, a_i^l), \dots, ((a_j^l, a_i^l), \dots, (a_n^l, a_i^l))) x = (y^l, a_i^l), \quad i = 1, n \quad (7)$$

これを解くために、 $a_i^l = 0$  となる第  $i$  行と第  $i$  列ならびに  $x$  の第  $i$  要素を除去する。従って計算負荷は  $L$  が小さい時極めて軽くなる。

## 2.2 矢印 (2) 上の遡及

$v_j$  を  $A$  の  $j$  番目の固有ベクトル、 $\lambda_j$  をそれに対応する固有値とする。すなわち

$$A v_j = \lambda_j v_j, \quad j = 1, n \quad (8)$$

$A$  が  $n$  個の一次独立な固有ベクトルを持つ時、任意の解空間ベクトル  $x$  は固有ベクトルの一次結合で表現で

きる。

$$\begin{aligned} x &= c_1 v_1 + \dots + c_j v_j + \dots + c_n v_n \\ &= A^{-1} (c_1 A v_1 + \dots + c_j A v_j + \dots + c_n A v_n) \\ &= A^{-1} (c_1 \lambda_1 v_1 + \dots + c_j \lambda_j v_j + \dots + c_n \lambda_n v_n) \end{aligned} \quad (9)$$

$x$  が与えられた時、それに対する各固有ベクトルの寄与値、すなわち展開係数を式 (2), (3), (7) にて  $a_j, x_j, y$  を  $v_j, c_j, x$  で置き換えた式を用いて求めることができる。 $x$  に対する寄与の大きな固有ベクトルが求めれば、それは式 (10) より像空間ベクトル  $y$  にも認められる可能性があり、あるいは行列列ベクトルの性質に帰着される。後者の場合、固有ベクトルに対する列ベクトルの寄与は、式 (8) の性質から次のように簡単に求めることができる。 $v_j$  を固有ベクトル、 $v_{j,i}$  を固有ベクトルの第  $i$  要素とする時

$$\begin{aligned} v_j &= \lambda_j^{-1} A v_j \\ &= \lambda_j^{-1} (a_1 v_{j,1} + \dots + a_i v_{j,i} + \dots + a_n v_{j,n}) \end{aligned} \quad (11)$$

## 3 可視化診断システム

この方法による診断機能を高水準偏微分ソルバ DEQSOL ([1]) に組み込んだ。DEQSOL の単位処理は一つの線形化された偏微分方程式の求解にあり、単位処理に上記の診断方式を適用した。また単位処理間をつなぐために対話実行機能を付与した。可視化のために、日立ダブリン研究所で開発された 2 次元ポストプロセッサ Mgraf を接続、使用した。

### 3.1 概要

図 2 に診断システムの構造を示す。このシステムはコントロールウインドウ (sview) と可視化ウインドウ (Mgraf) の 2 つの画面を持つ。両画面とも、DEQSOL アプリケーションが最初に連絡サブルーチン COMM を呼ぶ時に作られ、開かれる。

コントロールウインドウは、上部の進行制御ボタン、中央のプログラムテキストフィールド、左サイドのブレークボタン群と進行表示ランプ、右サイドにある変数値表示/設定フィールドの 4 つのフィールドからなる。可視化ウインドウは、上部のメニューバー、中央の表示エリア、左下の立体図コントロール、右サイドの固有ベクトルコントロール、右下のズームコントロール、ならびに独立した変数バーの 6 つの部分からなる。固有ベクトルコントロールは表示すべき固有ベクトルの選択を行なうスライドバーと、固有ベクトル/行列列ベクトル番号の

表示と選択を行なうリストフィールドを含む。

コントロールウインドウを管理するコントロールプロセス、可視化ウインドウを管理する可視化プロセスならびに DEQSOL アプリケーションの3つのプロセスは、UNIX<sup>TM</sup>のソケットとシグナル関数を介して通信する。前2つのプロセスはアプリケーションプロセスが COMM 連絡サブルーチンを最初に呼ぶ時に生成する。

### 3.2 機能

図3はコントロールウインドウの例を示す。DEQSOL ソーステキストのスキーム部分がプログラムテキストフィールドに表示される。テキスト内の COMM コールに対応して、ブレークボタンが取られる。ボタンのいずれかを押下することによりブレークポイントが設定され、ブレークポイントにおいては DEQSOL アプリケーションが COMM サブルーチンからコントロールプロセスに制御を渡し、コントロールウインドウからの対話制御を可能にする。またアプリケーションは、COMM サブルーチンの引数に指定された配列の値を可視化プロセスに送り、可視化プロセスはそれらを表示する。変数引数の値はコントロールプロセスに送られ、コントロールウインドウに表示される。各々のブレークポイントでは復帰再実行に備えて fork() システムコマンドを用いてアプリケーションプロセスのコピーが取られ、待機状態に置かれる。

図3に示す通り進行制御ボタンは10個のボタンからなる。最上段の3つのボタンは実行制御に関するもので、'adv'は次のブレークポイントへの進行を、'recov'は直近のブレークポイントへの復帰を、'quit'は終了を指示する。2段目の4つのボタンは、線形計算時の内部状態の表示を指示し、可視化診断に用いられる。'Eigmod'と'eigen'は線形計算と同時に係数行列の固有ベクトルの計算を行ない可視化ウインドウに表示することを指示する。'coef'は行列ベクトルの表示を指示する。'RHS'ボタンは方程式右辺の定数ベクトルの表示を指示する。3段目の3つのボタンはPCGのような反復ソルバが使われた時の解への収束過程の表示に関するものである。

図4に可視化ウインドウの例を示す。可視化ウインドウは変数バーで指定された配列の等高線図とカラーマップを表示エリアに表示する。固有ベクトルコントロールのスライドバーからリストフィールドで選択された固有ベクトルを表示することもできる。'Assoc'サブメニューの選択により、指定されたベクトルに対する固有ベクトル

ないし行列ベクトルの寄与を最小自乗フィッティングにより求めることができる。これは可視化診断の主役の一つである。'Assoc'指定の結果、寄与値の高いベクトルの番号がリストフィールドに表示される。寄与度を求める空間領域をズームコントロールで制限することができる。

可視化診断システムはC言語とX/Motifグラフィックライブラリを用いて作られており、標準のUNIX<sup>TM</sup>ワークステーションの上で動作する。

## 4 診断例

提案した方法の有効性を2つのケースにより評価した。

### 4.1 ケース1

第1のケースは部分的に強い風速場内の静的温度分布を求める問題である。求めた温度分布は図4に示すように特徴的な2つ目玉を持つのでこの原因を診断する。解くべき方程式は次のとおりである。

$$\Delta U - \rho \mathbf{V} \cdot \text{grad} U = 0 \quad (12)$$

ここで

$$\rho = 1.0,$$

$$\mathbf{V} = (v_x, v_y), v_x = 0.0, v_y = C1 \exp(-C2 * (x - x_0)^2 - C2(y - y_0)^2),$$

$$C1 = 1000.0, C2 = 100.0$$

すなわち、風速場のy要素は点(x0, y0)の周りに急なピークを持ち、この点は領域の中心に位置している。風が縦方向に吹くにも関わらず2つ目玉は横に並んでいるのでこの解は不審である。上端の温度は100Cに右端の温度は0Cに固定し、その他の境界にはノイマン条件を与え断熱とする。離散化には一次の基底関数によるガレルキン式有限要素法を用い、線形方程式の解法には前処理付き Biconjugate Gradient 法を用いた。

症状が解ベクトル空間にでているので、このケースは図1の矢印(2)の遡及になる。そこでまず、解に対する固有ベクトルの寄与を式(9)により求めた。図5に可視化ウインドウ上での'Assoc' eigen 操作の結果を示す。固有ベクトルコントロール部のリストフィールドに寄与の高い9つのベクトル番号が表示されている。それらを上から順番に選択し、表示フィールドにて解と重ね合わせて比較してゆくと、図に示すように5番固有ベクトルが2つ目玉の特徴を良く捉えていることが判る。このベクトルは白線で示す像ベクトル空間の右辺ベクトルとは

合致しないので、症状の原因は行列の性質にあると判断される。そこで式 (11) を用いて 5 番固有ベクトルに対する行列ベクトルの寄与を求めた。図 6 に可視化ウィンドウ上で 'Assoc' coef 操作を行なった結果を示す。寄与度の高い列ベクトルの番号がリストフィールドにリストアップされると共に表示フィールド内でそれらに対応する節点が太丸でマークされる。太丸をクリックすると図の白線で示すように、選ばれた列ベクトルの値が等高線で表示される。その形から、この列ベクトルが対角優位性を欠き、もとの方程式の性質を反映した離散化を行っていないことが直ちに見てとれる。これは、風速場に対してメッシュの分割が粗過ぎることに要因がある。

## 4.2 ケース 2

2 番目のケースは細長い領域内の時間項のある物質拡散問題である。拡散源は入口 (左側) 近くの床に置かれている。風が左から右に高さのみに依存する分布に従って定速に吹いているものとする。物質の密度は入口と上端でゼロ、下端と出口でノイマン条件を設定する。初期の物質密度を全域でゼロとする。物質密度分布を  $DD$  とおくと、支配方程式は次のようになる。

$$\rho(\partial DD/\partial t + \mathbf{V} \cdot \text{grad} DD) = k\Delta DD + Q \quad (13)$$

ここで

$$\rho = 1.2, k = 1.5,$$

$$\mathbf{V} = (v_x, v_y), v_x = VM(1 - y^2), v_y = 0.0, VM = 10.0,$$

$Q = Q_0 ep / (ep^2 + (x-1)^2 + y^2)$ ,  $Q_0 = 10.0$ ,  $ep = 0.01$  これに対し次のような semi-implicit スキームを用いて時間進行を行なうと、 $DD$  の分布に物質源からの剥離がみとめられ、これを診断の対象とした。

$$DD = D0 + dtt(k\Delta DD + Q) / \rho - dtt \mathbf{V} \cdot \text{grad} D0 \quad (14)$$

ここで、 $D0a$  は 1 時間ステップ前の分布を意味し、時間刻み  $dtt = 0.1$  を用いた。線形方程式の解法には、前処理付き Conjugate Gradient 法を用いた。

これもケース 1 と同じく解ベクトル空間の症例の診断であるので、剥離の始まる時点での  $DD$  に対して 'Assoc' eigen 操作により寄与度の大きな固有ベクトル群を求め、表示フィールド内で  $DD$  と較べてゆき、剥離の始まりを示す  $DD$  のへこみに合致するものを得た。この固有ベクトルが実は式 (10) の関係により像空間の RHS ベクトルと相関していることが判り、RHS ベクトルの相関部分があるような形となる理由を図 1 矢印 (3)' のルートから追求することにより原因が式 (14) における風速項

の陽的な扱いにあることをつきとめた。事実、風速項も陰的に扱う full-implicit scheme を使用すると剥離は起こらないことが確認できる。

このケースは、線形変換における固有ベクトルの形状不変性が解ベクトル空間と像ベクトル空間を橋渡ししている例である。

## 5 おわりに

4 章の診断例に見られるとおり、係数行列の固有ベクトルは解空間の特徴的な症状を良く捕捉し、またそれらを像空間や行列ベクトルの性質に関連づけられることが判った。最小自乗法による論理的な関連づけとパターンマッチによる直観的関連づけを交互に用いることにより効率的な選及を行なうことができる。過去の状態への復帰を含む診断システムの対話的な機能によりさらに長い連鎖の選及が行なえる。

本方式の現時点のボトルネックは固有ベクトル計算のコストにあるが、将来は高性能ワークステーションの導入によりもっと楽になることが期待できる。また、実用面の課題は本格的な 3 次元問題への挑戦であり、これに対しては並列計算機の活用が有望と思われる。

## 謝辞

2 章に述べた可視化診断方法は、発表者が 1992 年に米国 Purdue 大学計算機科学科に滞在中に発想を得たものである ([3])。滞在中の種々のサポートにたいし、Prof. John R. Rice 学科長に感謝申し上げる。

## References

- [1] 佐川暢俊、金野千里、梅谷征雄：数値シミュレーション言語 DEQSOL、情報処理学会論文誌第 30 巻第 1 号、1989
- [2] Seiji. Fujino, et al : Visualization of Convergence Behavior of Bi-CG STAB Method, Proc. of International Symposium on Supercomputing, Fukuoka, Nov. 6-8, 1991
- [3] Yukio Umetani : "The Visual Diagnosis of the Numerical Calculations for PDE Problems", Purdue University Computer Science Department Technical Report CSD-TR-92-082, 1992

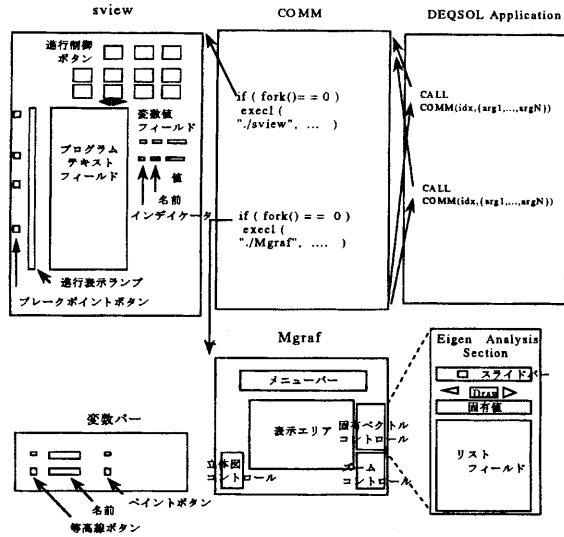


Figure 2: 可視化診断システムの構成

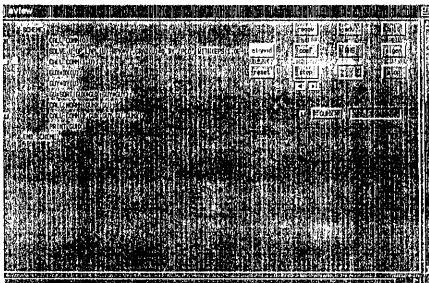


Figure 3: コントロールウインドウの一例

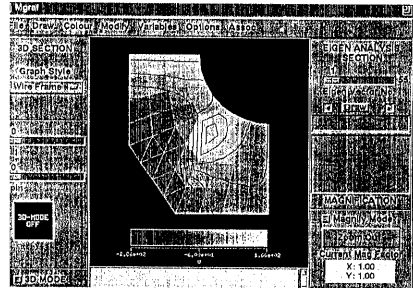


Figure 4: 可視化ウインドウの一例 (2つ目玉の症例)

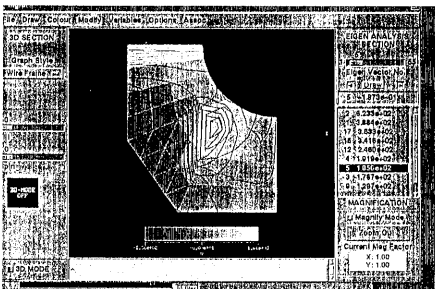


Figure 5: 固有ベクトルとの関連づけ

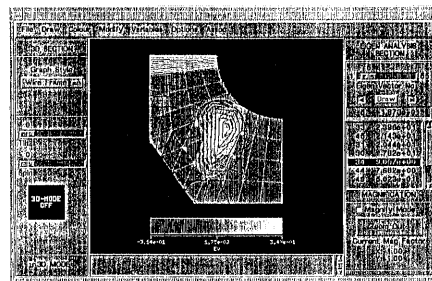


Figure 6: 行列列ベクトルとの関連づけ