29

The visual diagnosis on numerical calculation of PDE problems and experiments

Yukio Umetani
Hitachi Central Research Laboratory
Kokubunji, Tokyo 185, Japan. Telephone: +81-423-23-1111. Fax: +81-423-27-7700. Email: umetani@crl.hitachi.co.jp

Abstract

A systematic method to diagnose the causes of symptoms found in the PDE solving process by visual means is proposed and tested. This method diagnoses the causes of the symptoms by reducing them to the attribute of dominant eigenvectors or to the column vectors of the matrix of the discretized equation. This method uses the shape-preserving nature of eigenvectors in linear transformations. A contour map representation of vector values is utilized also to aid this reduction process by human cognitive capability. This method was incorporated into the high level PDE solver PDEQSOL. The results of its application to several PDE problems show the feasibility of this method. Incorporating this kind of method will enhance the ease-of-use of interactive PDE systems, and also will give a good testbed to judge their quality in various application environments.

Keywords

Partial differential equations, diagnosis, eigenvector, matrix, discretization, linear transformation, numerical simulation, numerical software

1 INTRODUCTION

The advent of high performance workstations in recent years enables the problem solver of PDEs (Partial Differential Equations) to have more interactive and human-friendly interfaces compared to earlier. Current users of PDE solvers often encounter some numerical troubles such as unbelievable or dubious results,local hazards, oscillations, or divergence of solutions. To get reasonable solutions they are forced to do some kind of trouble shooting or verification. The reasons for this trouble are diverse, ranging from the instability

R. F. Boisvert (ed.), *Quality of Numerical Software*

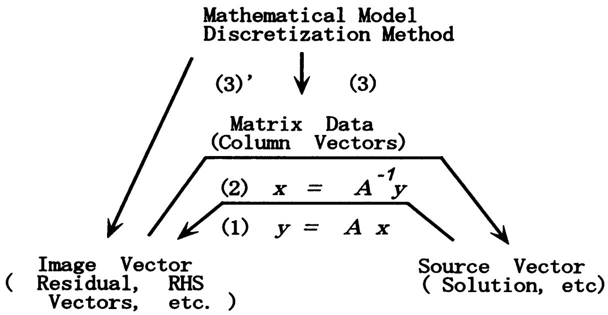

Figure 1 Relation of information components of the PDE solving process.

of physical phenomena, inadequate mathematical models, unsuitable discretization or numerical algorithm, misuse of the numerical software, to a trivial input error.

Many of these troubles may be avoided by enhancing the robustness and/or the guidance facility of solvers, or by reducing the amount of input. But the solver cannot completely check the dynamic behavior of the problem. The situation is delicate when the numerical results are on the border of correctness. Existing solvers do not help the user with interpretations in this situation. If one can provide the user with the information to clearly show the relation between results and causes, then it help the user in these delicate situations.

For this purpose, we have developed a systematic method to identify the causes of a symptom through a series of visual diagnostics. This approach also gives good information about the quality of the numerical software in individual application environments.

The organization of this paper is as follows. We give the method for diagnosis in Section 2. The diagnosis system incorporating this method is described in Section 3. Then a case study with two examples is described in Section 4, and finally concluding remarks are given in Section 5.

2 METHOD FOR VISUAL DIAGNOSIS

The core process of the PDE solver is in the numerical solution of a discretized linear equation. Figure 1 shows the relations of the information components in a PDE solving process. The information at the origin of an arrow determines the information at the destination. The matrix data (A), the source vector (x), and the image vector (y) are the three principal pieces of information in a PDE solving process. They are related each other by the arrows (1) ($y = Ax$), and (2) ($x = A^{-1}y$). The Matrix data is determined by the mathematical model, and discretization method. This is presented by the arrow (3). Also the RHS vector (the right hand side of the discretized linear equations) is determined by the mathematical model and discretization method (arrow (3)'). The difficulty of the diagnosis depends on the degree in which the spatial local correspondences from the destination to the origin of each arrow is preserved. Arrow (3) and (3)' have no problem. Arrow (1) also preserves the correspondences because y is composed of a linear combination of column vectors of A, and A is a sparse matrix whose non-zero elements are near the diagonal. Any symptoms in y are attributed to the column vectors related to the area where the symptoms lie, and these column vectors are again attributed to the mathematical model or discretization method through arrow (3).

The difficulty lies in the diagnosis of arrow (2). The same method as for arrow (1) cannot be used, because A^{-1} is not sparse. The eigenvector expansion of x is used in this case. The eigenvector preserves its shape, i.e., in most cases (especially when A is symmetric or Hermite) the ratio among vector elements does not change through the multiplication by A. If a symptom in x is related to some eigenvector of A, then the same shape may appear in y, or it is attributed to the column vectors of A constituting that eigenvector. Then the local correspondences on arrow (3) or (3)' can be used. When A is not symmetric but real valued, the real and the imaginary parts of the eigenvectors are used instead of eigenvectors themselves. If A is weakly nonsymmetric, these vectors behave almost the same as eigenvectors.

2.1 Diagnosis for arrow (1)

Let the jth column vector of A be a_j, and the jth element of x be x_j. Then we have

$$\begin{aligned} y &= Ax \\ &= a_1x_1 + a_2x_2 + .\ \ . + a_jx_j + .\ \ . + a_nx_n. \end{aligned} \tag{1}$$

To obtain the contribution from each column vector for given y, one needs to solve the following simultaneous equations.

$$(a_1, a_2, .\ \ ., a_j, .\ \ ., a_n)\, x = y \tag{2}$$

or

$$((a_1, a_i), (a_2, a_i), .\ \ ., (a_j, a_i), .\ \ .(a_n, a_i))\, x = (y, a_i), \quad i = 1, n \tag{3}$$

Here, (a_j, a_i) stands for the inner product of the vectors a_j and a_i. In general, equation (3) is easier than (2) to solve.

When the symptom is local, one can use the following local minimization technique (least squares method). This is more economical than (2) or (3). Let L be the set of nodes which belong to the specified local region, and let A^l, a_j^l be the matrix or column vector consisting of rows or elements of A or a_j belonging to L. Also let y^l be the projection of y to L. Then consider minimizing the norm of the residual on L, namely R^l.

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

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

Then we have

$$\partial R^l/\partial x = -2A^{lt}y^l + 2A^{lt}A^lx \tag{5}$$

and, assuming $\partial R^1/\partial x = 0$ at the minimum point,

$$A^{lt}A^l x = A^{lt}y^l. \tag{6}$$

From $A^l = (a^l_1, a^l_2, .. a^l_j, .. a^l_n)$, (6) becomes as follows.

$$((a^l_1, a^l_i), (a^l_2, a^l_i), .., ((a^l_j, a^l_i), .., (a^l_n, a^l_i))x = (y^l, a^l_i), \quad i = 1, n \tag{7}$$

To solve this equations, one needs to eliminate the rows where $a^l_i = 0$ as well as the corresponding x elements and column vectors. Therefore, the computation is much more economical than for solving (2) or (3).

2.2 Diagnosis for arrow (2)

Let v_j be the jth eigenvector of A, and λ_j be the corresponding eigenvalue, namely,

$$Av_j = \lambda_j v_j, \quad j = 1, n \tag{8}$$

holds. When A has n independent eigenvectors, an arbitrary vector x can be expanded by a linear combination of eigenvectors as follows.

$$\begin{aligned} x &= c_1 v_1 + .. + c_j v_j + .. + c_n v_n && (9) \\ &= A^{-1}(c_1 A v_1 + .. + c_j A v_j + .. + c_n A v_n) \\ &= A^{-1}(c_1 \lambda_1 v_1 + .. + c_j \lambda_j v_j + .. + c_n \lambda_n v_n) && (10) \end{aligned}$$

For a given x, one can evaluate the contributing values from the eigenvector using formulas similar to (2), (3), (7), by simply replacing a_j, x_j, and y by v_j, c_j, and x, respectively. Once a dominant eigenvector of x is obtained, one can also find it in the image vector y by the formula (10), or it can be attributed to the properties of matrix column vectors. The latter relation is easily obtained from formula (8). Let v_j be such an eigenvector, and let $v_{j,i}$ be the ith component of v_j. Then we have

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

The contributions of column vectors are easily calculated by this formula.

3 DIAGNOSIS SYSTEM

This method was incorporated into the high level PDE solving system PDEQSOL (Umetani, 1991). PDEQSOL's basic operation is the solution of a system of linearized PDEs. This high level feature gives the system the capability to conduct the diagnosis for each basic operation. It also gives backward/forward progress capability according to the results

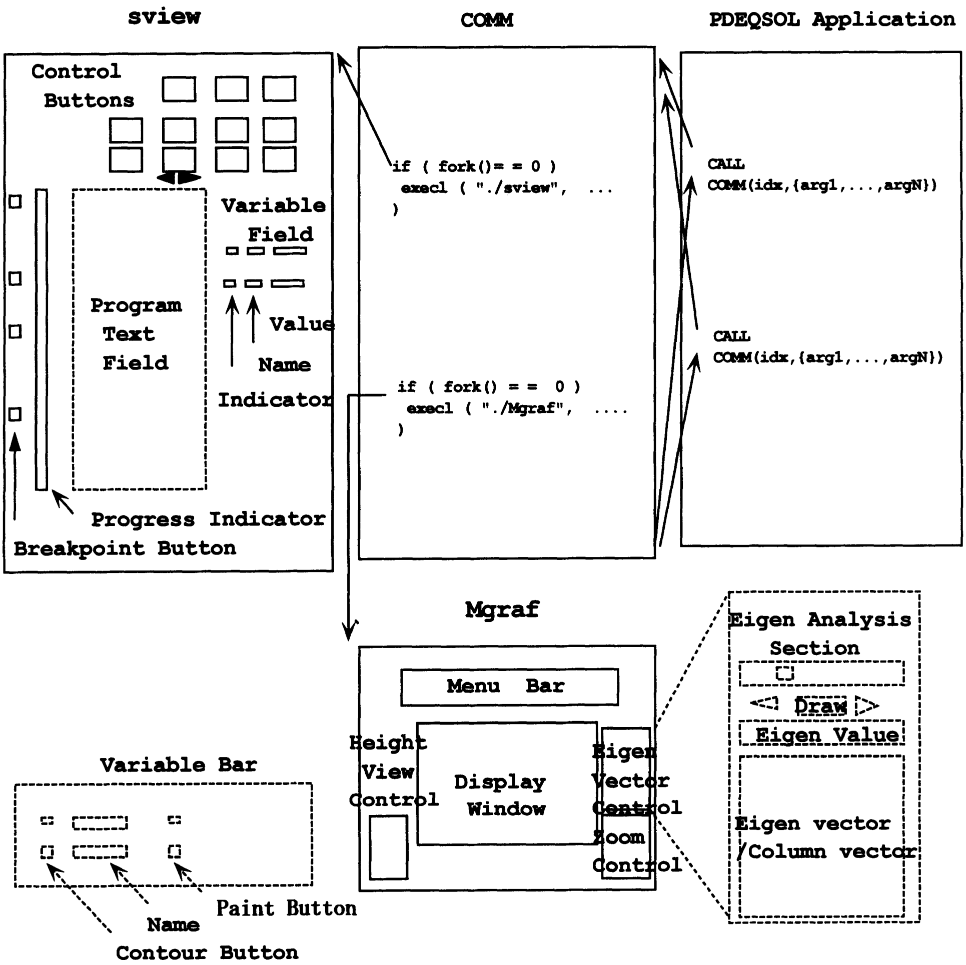

Figure 2 Overview of diagnosis system of PDEQSOL.

of the diagnosis. The two-dimensional post- processor Mgraf developed at the Hitachi Dublin Laboratory is used for visualization.

3.1 Overview

Figure 2 shows the overview of the diagnosis system. This system has two windows, the control window Sview and the visualization window Mgraf. Both windows are created and opened at the initial call of the communication subroutine COMM from PDEQSOL. The control window has four fields, namely the progress control buttons on top, the program text field in the center, the breakpoint buttons and progress indicator on left side, and the display/value-setting field for scalar variables on right side. The visualization window consists of six parts, namely the menu bar on top, the display window in the center, the height view control section in the left corner, the eigenvector control section on right side, the zoom control section in the right corner, and the separated variable bar. The eigenvector control section includes a slide bar and a list field to display eigenvector numbers or column vector numbers and to allow selection among them.

The control process in charge of the control window, the visualization process managing the visualization window, and the invisible PDEQSOL application are three independent processes mutually communicating through UNIX™ socket interfaces and signal functions. The first two processes are created by the application process at the initial call of COMM subroutine.

3.2 Functions

Figure 3 shows an example of the control window Sview. The SCHEME part of PDEQSOL source text is displayed in the program text field. The break button is associated with each COMM call in the text. One can specify execution breakpoints by pushing these buttons. PDEQSOL applications give control to the control process at each breakpoint from the COMM subroutine, and allow interactive control through operations on the control window. Also, PDEQSOL application sends array values specified as arguments of COMM to the visualization process, and the visualization process displays them. The values of scalar arguments are sent to the control process and are displayed on the control window. At each breakpoint, a new copy of current application process is created if the resources allow that. It uses the fork() system command, and is put into the wait state, prepared for the recovery action from the control window.

As shown in Figure 3, the progress control button set consists of ten buttons. Three buttons on the first line are related with progress control. Among them, *adv* returns control to the PDEQSOL application and proceeds to the next breakpoint. *Recov* returns control to the waiting application process in the last breakpoint and kills the current application process. And application resumes execution from last breakpoint. *Quit* terminates whole execution. Four buttons on the second line are concerned with the display of internal states of the linear equation solving process. These buttons are important for diagnosis. *Eigmod* and *eigen* are related to the calculation and display of the matrix eigenvector set. The *coef* button allows the display of column vectors of the matrix on the visualization window by mapping them to nodepoints. The *RHS* button displays the right-hand-side vector of the linear equation. The buttons on the third line are concerned with the display of the convergence of linear equation solver when iterative solver is used.

Figure 4 shows an example of the visualization window. The visualization window displays contour maps and color maps of the variables specified on the variable bar. One can compare the contour maps of several variables on the display window. The eigenvector selected by the slide bar or in the list field of the eigenvector section is also displayed on the display window. By selecting items from the *Assoc* submenu, one can evaluates the contributing value of individual eigenvectors or matrix column vectors to a certain target vector by using the least squares method. These menus are most important for diagnosis. As a result of the *Assoc* operation, the vector numbers with high contributing values are listed in the eigenanalysis section. The area for association can be restricted to a certain subdomain using zoom control section.

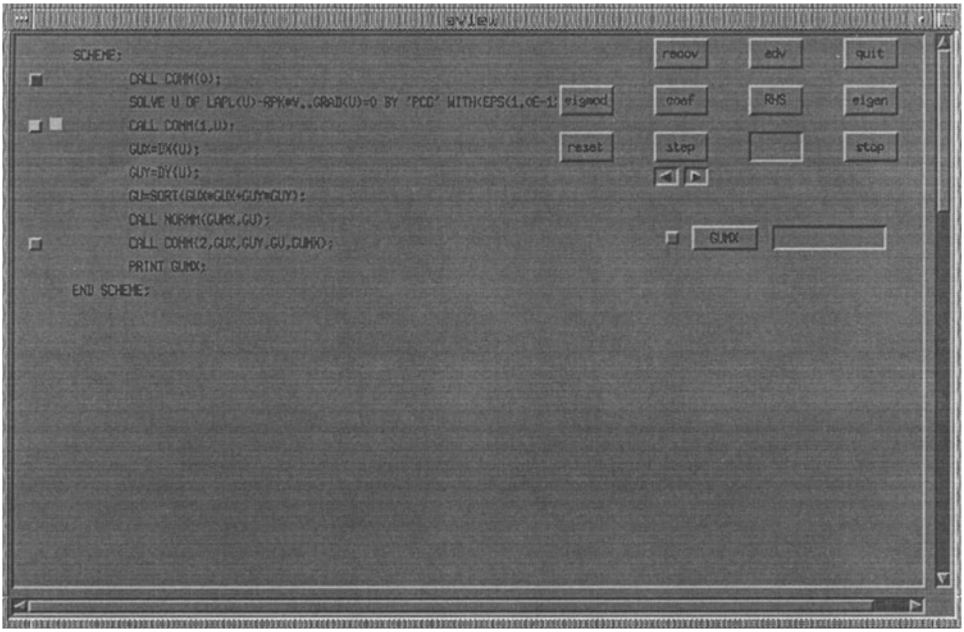

Figure 3 An example of the control window Sview.

The diagnosis system is coded in the C language with the X/Motif graphics libraries. It is operational on standard UNIX™ workstations.

4 EXPERIMENTS

We tested the feasibility of the proposed method with two cases.

4.1 Case 1

The first case is a two-dimensional heat distribution problem in a locally intensive wind velocity field. The temperature distribution contours have a peculiar double local extrema (a "double-eye" symptom) as shown in Figure 4. We try to find out the causes of this symptom. The equation to solve is

$$\Delta U - \rho \boldsymbol{V}..\mathsf{grad} U = 0 \tag{12}$$

where

$$\rho = 1.0,$$
$$\boldsymbol{V} = (v_x, v_y), v_x = 0.0, v_y = C1exp(-C2 * (x - x_0)^2 - C2(y - y_0)^2),$$

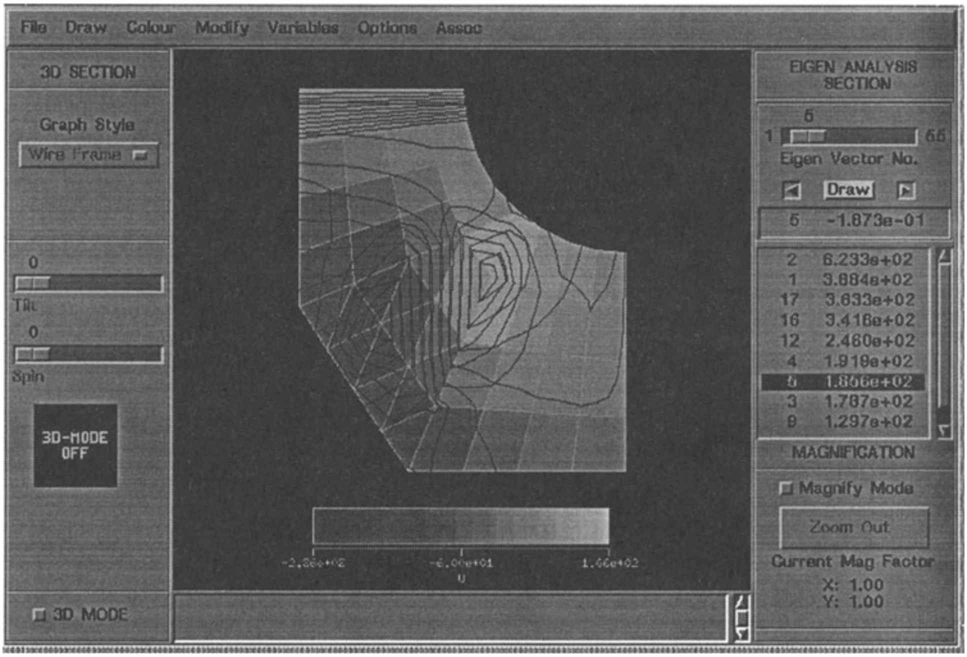

Figure 4 An example of the visualization window showing the double-eye symptom and the association with the eigenvector.

$C1 = 1000.0, C2 = 100.0$

The y component of the velocity field has a sharp peak around the point (x0, y0) located at the center of the domain. Because the wind blows in the vertical direction, the computed temperature distribution seems strange. The temperature on the upper boundary is fixed at 100C and on the right side at 0C. Neumann conditions are given on the other boundaries. The Galerkin Finite Element Method with linear basis function is used for discretization. Preconditioned Biconjugate Gradient Method is used to solve discretized simultaneous linear equations.

Because this symptom appears in the solution vector, the diagnosis path (2) in Figure 1 is selected. Therefore, the contribution of eigenvectors is evaluated using formula (9). Figure 4 also shows the result of the *Assoc eigen* operation. The top nine eigenvectors are listed in eigenanalysis section in the order of contribution. Selecting and comparing each of them with the symptom in the display window, we find the contour map of 5th eigenvector fits well to the double-eye symptom of solution as shown in Figure 4. Because this eigenvector does not resemble the RHS vector appearing near the upper boundary, the cause of this symptom is determined to be a property of matrix itself. So, the contribution of matrix column vectors is evaluated using formula (11). Figure 5 is the result of *Assoc coef* operation on the 5th eigenvector. The numbers of major matrix columns contributing to the shape of the 5th eigenvector are listed in the list field of eigenanalysis section. Also

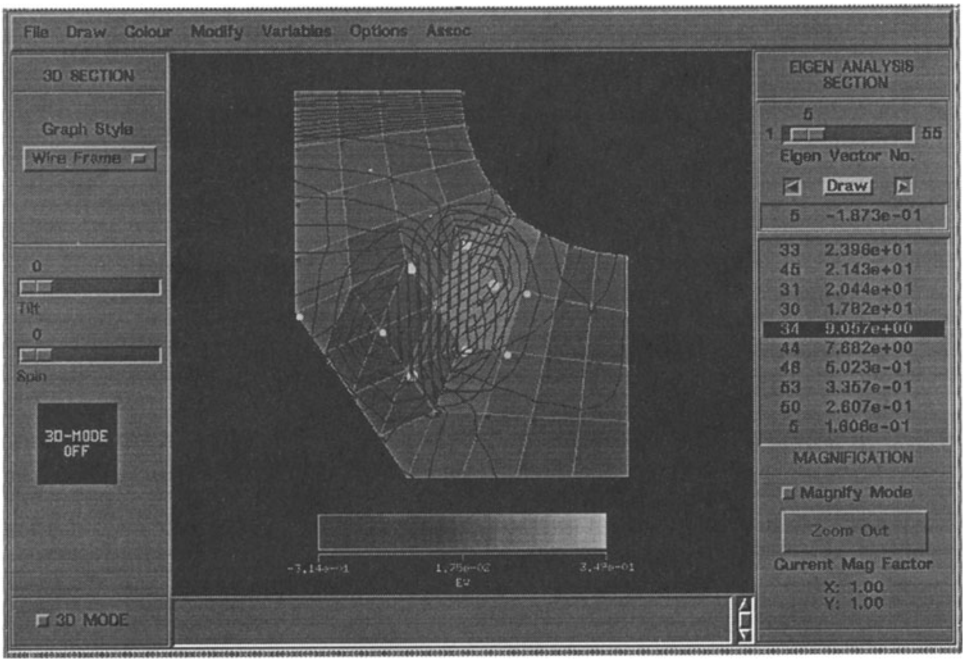

Figure 5 Association of the double-eye symptom with the matrix column vectors.

the nodes corresponding to these column numbers are marked by big balls. By pushing one of these balls, one can draw the contour map of selected column vector as shown in white line. Its shape clearly shows the lack of the diagonal dominance due to the mesh being too coarse for a locally intensive velocity field. In this way, the double-eye symptom in the solution space is linked to the property of the matrix through a specific eigenvector which captures the shape of the symptom.

4.2 Case 2

The second case is a transient material diffusion problem within a slender two dimensional region. The source of the material is placed on the ground near the inlet (left side). The wind is blowing from the left to the right at a fixed velocity which depends only on the height. The density of the material is set to zero at the inlet and ceiling. Neumann conditions are given at the ground and outlet. The initial distribution is zero in all regions. Let DD be the material density. The governing diffusion equation is as follows.

$$\rho(\partial DD/\partial t + \boldsymbol{V}..\mathrm{grad}DD) = k\Delta DD + Q \tag{13}$$

where

$\rho = 1.2, k = 1.5,$

$$V = (v_x, v_y), v_x = VM(1 - y^2), v_y = 0.0, VM = 10.0,$$
$$Q = Q_0 ep/(ep^2 + (x-1)^2 + y^2), Q_0 = 10.0, ep = 0.01$$

A local fine mesh near the material source is used. The discretization used is the Galerkin Finite Element Method with piecewise linear basis functions. The following semi-implicit scheme is used for the time marching to keep the discretized matrix symmetric.

$$DD = D0 + dlt(k\Delta DD + Q)/\rho - dlt\boldsymbol{V}..\mathsf{grad}D0 \tag{14}$$

$D0$ stands for the distribution at one time step behind, and $dlt = 0.1$ is used. The Preconditioned Conjugate Gradient method is used to solve the discretized linear equations. The interesting observation is that mass separation from the material source occurs after certain time steps. Let us diagnose this phenomenon.

The result of *Assoc eigen* operation of the solution vector DD at the time step where the mass separation begins indicates an eigenvector which correlates well with the dint of mass separation. Then comparing it with the RHS vector using relation (10), one knows that it is related well. Because the reason for the mass separation is now attributed to the RHS vector, the investigation on arrow (3)' in Figure 1 follows. This investigation reveals the fact that the reason is in the semi-implicit treatment of the velocity term. The computation using a full-implicit scheme

$$DD = D0 + dlt * (k\Delta DD + Q)/\rho - dlt\boldsymbol{V}..\mathsf{grad}DD \tag{15}$$

shows no mass separation.

In this case, the shape preserving nature of eigenvector in the linear transformation effectively served as a bridge between the solution space and RHS vector space. This, in turn, led to an explanation for the symptom observed.

5 CONCLUDING REMARKS

As shown in the previous examples, the eigenvectors of the coefficient matrix of the discretized linear equation prove to capture well the features of symptoms that appear in the solution space. They also proved useful to relate them to properties of the matrix or RHS vector space. By alternating between associations using the least squares method described in Section 2, and using intuitive pattern matching, one can make an effective search to diagnose symptoms. The interactive design of this system, including recovery features, also aids an effective search process.

The current bottleneck of this method is the high computing cost of the full eigenvector calculation. This situation will be improved in future by the use of high performance workstations. The next challenge is to apply this method to three dimensional applications. The use of the parallel machine will be promising for this purpose. This kind of system will enhance the ease-of-use of numerical software, and also will serve as a good testbed to judge software and solution quality under various application environments.

6 ACKNOWLEDGMENT

The proposed method for diagnosis was initially conceived by the author during his stay at Purdue University Computer Science Department (Umetani, 1992). The author expresses his gratitude to Prof. John R. Rice for various support during his stay.

REFERENCES

Umetani, Y., Konno, C. et al. (1991) Visual PDEQSOL: A Visual and Interactive Environment for Numerical Simulation, in *Proceedings of the IFIP TC2/WG 2.5 Working Conference on Programming Environments for High-Level Scientific Problem Solving*, (ed. P. W. Gaffney), North-Holland, 259-267.

Umetani, Y. (1992) The Visual Diagnosis of the Numerical Calculations for PDE Problems. Purdue University Computer Science Department Technical Report CSD-TR-92-082.

DISCUSSION

Speaker : Y. Umetani

J. Rice : Can you compare the amount of computation required to do the visual diagnosis with the amount of computation required to solve the partial differential equation?

Y. Umetani : At a statement where diagnosis is requested, the computation of whole eigenvector sets using the QR decomposition method occupies most of the computation time. It amounts to 50 to 100 times more than the time to solve the partial differential equation in the case of a medium sized 2-dimensional problem. It will be reduced in the future by a change of algorithm and selective calculation of low-mode eigenvectors. The total amount of computation required for diagnosis varies depending on the number of statements where the diagnosis is required.