A simple algorithm for the solution of two-dimensional diffusion equations

1986 J. Phys. A: Math. Gen. 19 L579

(http://iopscience.iop.org/0305-4470/19/10/004)

J. Phys. A: Math. Gen. 19 (1986) L579-L584. Printed in Great Britain

LETTER TO THE EDITOR

A simple algorithm for the solution of two-dimensional diffusion equations

A Birnboim[†], R Rankin, R Marchand and C R James

Department of Electrical Engineering, The University of Alberta, Edmonton, Alberta, Canada T6G 2G7

Received 29 November 1985

Abstract. A simple point iterative method is presented which is based on a splitting scheme natural to the finite difference form of two-dimensional diffusion problems. A proof is given of the unconditional stability and convergence of the algorithm and a comparison is made with other more commonly used iterative methods.

The finite difference solution of many boundary value problems reduces to the problem of inverting a large sparse matrix equation defined by $x = A^{-1}y$. Often the matrix A has a number of diagonals in a band immediately adjacent to the main diagonal and has two distinct groups of diagonals situated either side of, and far from, the central band. In this letter we restrict ourselves to a study of equations used to describe diffusion in two-dimensional hydrodynamic codes, although in principle the method to be described applies equally well to any system of partial differential equations which, in finite difference form, has a matrix structure similar to that mentioned above.

In the following diffusion problem we shall consider a plasma described by a two-fluid model in which the charge species (electrons and ions) are free to move in two spatial coordinates r and z. The physical processes to be modelled include energy exchange from the hotter to the colder fluid and diffusion of the species temperatures in each of the r and z directions. For simplicity, a five-point spatial difference operator is employed in which the temperature in a given cell is coupled to its four nearest neighbours. However, it should be noted that the algorithm can readily be extended to an eight-neighbour approximation. If the equations are differenced on a spatial grid having N_r and N_z nodal points in the r and z directions, respectively, the temperatures T_e and T_i (stored in a vector x) are advanced in time according to

$$Ax = y \cdot \qquad (1)$$

In this expression, A represents an $n \times n$ matrix (with $n = 2N_rN_z$) in which most of the entries are zero except for those occupying the diagonals described above. First, note that the matrix in equation (1) can be shown to be real, symmetric, irreducible and positive definite. Moreover, all of the off-diagonal elements of A are non-positive, so that A is an irreducible Stieltjes matrix (a positive definite, symmetric M matrix). Such matrices often arise in the finite difference form of partial differential equations.

[†] Permanent address: ADA, PO Box 2250, Haifa 31021, Israel.

To solve for the temperatures at each grid point, a matrix equation is constructed such that the elements in vector x alternate between T_e and T_i. This is accomplished by scanning the mesh row by row (or column by column), covering all N_z cells for a fixed r position (or vice versa). With this form of ordering, the matrix has a band of five diagonals centred on the main diagonal and has two distant diagonals with entries $2N_z$ rows above and below the diagonal element in each column of A. Each row i of A therefore has the following structure:

$$(0, 0\ldots, -R_{i-2N_z}, 0, \ldots, -Z_{i-2}, -W_{i-1}, D_i, -W_i, -Z_i, 0\ldots, -R_i\ldots 0, 0) \qquad (2)$$

where W_i is the exchange term between ions and electrons in a given cell. In each row, one of W_i, W_{i-1} is zero depending on the parity of i. The conductivity term coupling a cell to its neighbour in the $+z(+r)$ directions is $Z_i(R_i)$ while $Z_{i-2}(R_{i-2N_z})$ represents coupling to the $-z(-r)$ direction. The diagonal term D_i is defined by

$$D_i = B_i + W_i + Z_i + R_i + W_{i-1} + Z_{i-2} + R_{i-2N_z} \qquad (3)$$

where B_i arises from the time derivarive $C_v \partial T/\partial t$ in the diffusion equation for T_e or T_i and is proportional to the specific heat of the relevant species. The signs in equations (2) and (3) are chosen so that all of the elements in (3) are non-negative.

The solution of equation (1) is complicated due to the presence of the far diagonal terms in equation (2). Many methods for finding the solution to equation (1) have been proposed in the published literature [1, 2 and references therein]. The most straightforward is the splitting method, which reduces the two-dimensional problem to several stages of a one-dimensional problem, i.e. diffusion in $z(r)$ is first treated followed by diffusion in $r(z)$. However, this scheme, while straightforward to implement, only provides an approximate solution. Another way of solving equation (1) is by iteration. The most common methods in this group include point Jacobi and successive relaxation. Gauss-Seidel represents a special case of the latter with a relaxation parameter $\omega = 1$. In the above methods A is first split into diagonal, upper and lower triangular matrices according to

$$A = D - L - U. \qquad (4)$$

Although the iteration method converges in the above schemes, the convergence rate for the problem discussed in this letter can often be very slow. In the point Gauss-Seidel method (which has a more rapid rate of convergence than point Jacobi), for example, the rate of convergence is determined by the spectral radius of $(D-L)^{-1}U$. Since U is the transpose of L when A is symmetric, the spectral radius in cases of strong diffusion, or strong exchange of energy between electrons and ions, may be near to unity. Gauss-Seidel iteration, although very efficient as far as storage requirements is concerned, is poor in effect of the number of iterations required.

One of the most successful algorithms for solving equation (1) appears to be the ICCG method due to Meijerink and Van Der Vorst [3]. In this method matrix A is first decomposed into an approximate factorisation $A \simeq LL^{\mathrm{T}}$ where L is lower triangular and $L^{\mathrm{T}} =$ transpose (L). An incomplete Choleski decomposition is used to calculate the elements of L by disregarding all entries except those which correspond to non-zero elements appearing in A. In this way an approximate Choleski decomposition is performed which prevents fill-in of the matrix L. However, for problems which are local in time, as in the diffusion equation, it should still happen that $(LL^{\mathrm{T}})^{-1}A \simeq I$. A conjugate gradient method applied to the system $[L^{-1}AL^{\mathrm{T}-1}]L^{\mathrm{T}}x = L^{-1}y$ is thus found to converge in a small number of iterations, while the storage required to implement the method is moderate. However, the iterations cannot begin until A has been factored into the product $A = LL^{\mathrm{T}}$, and each conjugate gradient iteration is composed of several successive steps. A full description and comparison with other methods is given in the paper by Kershaw [2].

We now introduce an iterative method which is conceptually and numerically very simple. As the splitting to be performed exploits the two-dimensional origins of matrix A, its convergence is expected to be faster than other more general point iterative methods. We first note that without the far diagonals which arise from diffusion in r, the problem reduces to that of simultaneously solving the ion and electron temperature equations in one dimension. This can be done by using a double sweep Gaussian elimination process [4]. Thus, for two-dimensional problems we write matrix A as

$$A = M - N \tag{5}$$

in which M is a matrix containing the five centrally grouped diagonals and where N contains the negative of the remaining two far diagonals. Notice that (5) implies $N \geqslant 0$ (all elements are non-negative in N) and that M is real, symmetric and has only non-positive off-diagonal elements. From equations (2) and (3), and the Gerschgorin circle theorem, it follows that M is positive definite so that it is also a Stieltjes matrix. The structure of M is such that it can be reduced to N_r irreducible matrices dimensioned $2N_z \times 2N_z$, each one of which determines a two-temperature one-dimensional conductivity problem. Equation (1) can then be solved iteratively with the solution at the mth iteration given by

$$x_i^{(m)} = M^{-1}(y + Nx^{(m-1)}) = M^{-1}y' \tag{6}$$

where $y' = y + Nx^{(m-1)}$. The inversion of the five-diagonal matrix M can be done exactly using a double sweep technique. For $i = 1, 2, \ldots, n$, we define coefficients E, F, G, as follows:

$$\begin{aligned}
K_i &= Z_{i-2}G_{i-2} + W_{i-1} \\
H_i &= D_i - Z_{i-2}E_{i-2} - K_iG_{i-1} \\
E_i &= Z_i/H_i \\
G_i &= (W_i + K_iE_{i-1})/H_i \\
F_i &= (y_i' + Z_{i-2}F_{i-2} + K_iF_{i-1})/H_i.
\end{aligned} \tag{7}$$

The E, F and G vectors are used recursively to sweep back from $i = n$ to $i = 1$ to obtain the solution through

$$x_i = F_i + E_ix_{i+2} + G_ix_{i+1}\,\cdot \tag{8}$$

For an isolated system the boundary conditions are such that all variables outside the interval $i = 1, \ldots, n$ are taken to be zero. For two-dimensional diffusion the elimination can be performed on each submatrix of M separately, demonstrating the relation between the point iterative scheme defined by (5) and (6) and a block Jacobi method for this problem. Thus, if the iterations in (6) converge, an exact solution to the two-dimensional problem is obtained using a working area of only three one-dimensional vectors (E, F and G), each of length $2N_z$. However, the algorithm defined by equations (5) and (6) still applies when M cannot be divided into submatrices. In both cases, it may be preferable to calculate the vectors E, G, H and K once in the first iteration and store them in vectors of length n, as only F must be evaluated in subsequent iterations. This increases storage requirements, but reduces the total work per iteration by approximately a factor of two. In this case, the matrix coefficients D_i and W_i need not be stored, as they are not used to calculate F.

We now prove that the iterative process defined by equation (6) is always convergent. One way of doing this is to prove that equation (5) represents a regular splitting and then invoke the convergence theorem for regular splitting methods [1]. It is instructive, however, to prove the convergence directly. We begin by defining the error vector in iteration number m as

$$\varepsilon^{(m)}=x^{(m)}-x \tag{9}$$

where x represents the exact solution vector of (1). Thus

$$\varepsilon^{(m)}=M^{-1}N\varepsilon^{(m-1)}=(M^{-1}N)^m\varepsilon^{(0)} \tag{10}$$

so that the algorithm converges provided the spectral radius $\rho(M^{-1}N)$ is less than unity. From equation (5) it can be deduced that

$$M^{-1}N=(A+N)^{-1}N=(I+Q)^{-1}Q \tag{11}$$

where

$$Q=A^{-1}N. \tag{12}$$

Thus, if τ is an eigenvalue of Q associated with eigenvector x, i.e. if

$$Qx=\tau x \tag{13}$$

then

$$M^{-1}Nx=(I+Q)^{-1}Qx=\frac{\tau}{1+\tau}x. \tag{14}$$

Therefore $\nu=\tau/(1+\tau)$ is an eigenvalue of $M^{-1}N$ associated with the same eigenvector x. Conversely, if

$$(I+Q)^{-1}Qx=\nu z \tag{15}$$

then

$$Qz=\nu(I+Q)z \tag{16}$$

which implies that for any non-trivial vector z, ν is not equal to unity and

$$Qz=\frac{\nu}{1-\nu}z=\tau z. \tag{17}$$

Thus, a one-to-one correspondence between the eigenvalues of Q and those of $M^{-1}N$ exists. Let us now exploit the properties of M and A to prove that $\rho(MN^{-1})<1$. Since A is an irreducible Stieltjes matrix, $A^{-1}>0$ (see for example [1], corollary 3, p 85), and likewise since M is composed of irreducible blocks, $M^{-1}\geqslant 0$. Because $N\geqslant 0$ it follows that $M^{-1}N\geqslant 0$ and $Q=A^{-1}N\geqslant 0$. From these properties it follows from the Perron–Frobenius theorem that the non-negative matrix $M^{-1}N$ has a simple positive eigenvalue $\nu_{\max}$ equal to $\rho(M^{-1}N)$ with its associated eigenvector x also positive. This eigenvector is also an eigenvector of $Q\geqslant 0$ which implies by (13) that $\tau\geqslant 0$. Therefore

$$\nu_{\max}=\rho(M^{-1}N)=\frac{\tau}{1+\tau}<1 \tag{18}$$

which proves the desired convergence of the method.

We have described a simple, fast, storage saving algorithm which can be used to solve two-dimensional diffusion problems. For a matrix A (satisfying $Ax = y$) with four diagonals adjacent to the main diagonal, and two far diagonals, an exact inversion of the matrix containing the main five-diagonal band is first performed and an iterative procedure is applied to correct the solution by taking account of the distant diagonals. The convergence of the above method has also been proven. The method is expected to have a fast rate of convergence when the elements in the matrix N are small compared to the elements in the main matrix M, where $A = M - N$. Since A can be constructed in such a way that N accounts for diffusion in either one of the two spatial directions, r or z, in problems where diffusion is stronger in the r direction, an interchange between the roles of r and z will improve efficiency. Comparing convergence rates to other methods, we note that, for two regular splittings of the form $A = M_1 - N_1 = M_2 - N_2$, with $A^{-1} > 0$ and $N_2 - N_1 \geqslant 0$, the first splitting is known to converge more rapidly. Thus the present method is faster than point Jacobi in which M only contains the main diagonal. Comparison with Gauss-Seidel is less straightforward as different diagonals are found in N in each case. From equation (2) the matrix N in the present method contains two diagonals R_{i-2N_z} and R_i, respectively, while in Gauss-Seidel splitting it would contain three diagonals W_i, Z_i, R_i. Obviously, from a physical point of view, when the exchange rates of energy from T_e to T_i are large the present method converges faster than Gauss-Seidel.

As stated at the beginning, a reliable and widely used method for solving (1) is the ICCG method. We believe that in general the present algorithm will not be as efficient as the ICCG method, in the sense that it usually requires more iterations in order to converge. It should be noted, however, that the work per iteration is much less compared to the ICCG method. Thus it is possible that for non-pathological problems the present algorithm will be faster in terms of CPU time. As an illustration, we compare the performances of both methods in the simulation of an experiment currently being performed at the University of British Columbia [5]. A laminar helium gas jet, with an approximately rectangular cross section, is irradiated at right angles by a CO_2 laser. The width of the jet is 1.2 mm and the electron density is half of the critical density when fully ionised. A 2.84 ns triangular laser pulse, which reaches its maximal 10^{14} W cm^{-2} power after 1.2 ns, is focused to a 100 μm Gaussian spot (measured in vacuum) at the centre of the jet. The simulation was performed with a two-dimensional Eulerian code using a spatial grid with 60 cells in the z direction and 25 cells in the r direction. This mesh required that a 3000 × 3000 matrix be inverted at each time step to obtain the solution of the diffusion and equilibration equations. Details of the code were the same for both of the algorithms and are only briefly described here. The finite difference approximation of the temperature diffusion and equilibration equation is given in [6]. Flux corrected transport routines are used for the hydrodynamic motion of the plasma, and laser deposition and propagation is described by a paraxial ray solution to Maxwell's equations [7]. The ionisation of the plasma is derived from a collisional-radiative atomic physics model. Profiles of the temperature, density, etc, will be presented elsewhere. Here, it is sufficient to point out that the solutions obtained using both methods were found to be almost identical. The laser energy, which was deposited in a narrow region around the z axis, diffused rapidly in both the r and z directions. For example, after 2 ns, high temperatures (of about 60% of the maximal temperature at this time) were found as far as 7 mm from the z axis. This illustrates the fact that the conductivity is not highly directed in this example. In figure 1 we plot the CPU time spent within the diffusion subroutine against the physical time over

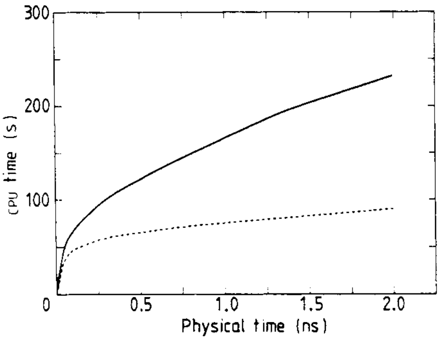

Figure 1. CPU time taken for the diffusion stage plotted as a function of the physical time modelled in the simulation. The full curve is for the ICCG method and the broken curve is for the point iterative method.

which the experiment evolved. The full curve shows results obtained with the conjugate gradient algorithm, using incomplete Cholesky decomposition to precondition the matrix A. The broken curve corresponds to the present point iterative method. The latter algorithm is faster throughout the entire time covered in the simulation.

In summary, we have presented a simple algorithm for solving simultaneously the temperature diffusion and equilibration equations in a two-dimensional two-component plasma. The method is straightforward to program, is relatively inexpensive to run and is shown to be unconditionally convergent. The algorithm is contained in (1), (5) and (6) and can be implemented in approximately sixty FORTRAN lines. It is mainly in order to preserve this basic simplicity that we do not deal here with combined methods, such as a conjugate gradient algorithm applied to matrix A, which could be preconditioned by matrix M. Such a method would converge rapidly when $M^{-1}A$ becomes close to unity. Finally, we note that, although the convergence proof has been presented for two-fluid, two-dimensional five-cell diffusion, it is clearly not restricted to that particular case. The method can readily be extended to treat multiple fluids in two or more dimensions. Convergence of the resulting algorithm would be proven in much the same way.

The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for its continuing financial support.

References

[1] Varga R S 1962 *Matrix Iterative Analysis* (Englewood Cliffs, NJ: Prentice-Hall)
[2] Kershaw D S 1978 *J. Comput. Phys.* **26** 43
[3] Meijerink J A and Van Der Vorst H A 1977 *Math. Comput.* **31** 148
[4] Birnboim A 1985 *J. Comput. Phys.* to be published
[5] Meyer J 1985 Private communication
[6] Rankin R, Birnboim A, Marchand R and James C R 1986 *Comput. Phys. Commun.* to be published
[7] McMullin J N, Capjack C E and James C R 1981 *Comput. Phys. Commun.* **23** 31