A PETROV-GALERKIN TYPE METHOD FOR SOLVING $Ax = b$, WHERE A IS SYMMETRIC COMPLEX.

H.A. van der Vorst
Technical University Delft,
PO Box 356, 2600 AJ Delft, Netherlands

J.B.M. Melissen
Nederlandse Philips Bedrijven, CFT Automation/CPS,
P.O. Box 218, 5600 MD Eindhoven, Netherlands

Abstract

Discretisation of for instance steady state eddy current equations may lead to a linear system $Ax = b$ in which the complex matrix A is not Hermitian, but may be chosen symmetric. In the positive definite Hermitian case, an iterative algorithm for solving this system can be defined. The residual vectors can be made mutually orthogonal by means of a two-term recursion relation which leads to the well-known Conjugate Gradients method.
For the non-Hermitian symmetric case orthogonality cannot be achieved in this simple way. In this paper another two-term recursion for the residuals is considered and although the generated residuals are not orthogonal, this algorithm leads to an iterative method which is very similar to the CG-method.

Introduction

One of the reasons for treating linear steady state AC eddy current problems separately from the more general transient case stems from the fact that in this special situation an elegant time-separation can be performed using complex arithmetic. One of the disadvantages of this approach however is that the linear system generated by the space discretization is also complex.
For Hermitian ($\bar{A}^T = A$) positive definite systems a straight-forward extension of the Conjugate Gradients algorithm can be given, but for most problems this class is too restrictive. In this paper we will concentrate on another class of complex systems which are symmetric ($A^T = A$). This situation may occur in three dimensional AC linear eddy current problems where conducting and non-conducting regions coexist. Examples are the Carmen formulation [2] which uses a modified vector potential (with an axial gauge in the time direction) in eddy current regions coupled to a magnetic scalar potential in air and the RS formulation [1]. In both cases the system can be made symmetric [2] by a careful use of interface conditions, but not complex symmetric. Unfortunately the conjugate gradients method cannot be used straightforwardly. Experience shows however that if the complex inner product in the CG algorithm is replaced by its real counterpart the method, a useful algorithm is obtained. We will show that this method is actually a projection type method and an analysis of the applicability will be given.

The Conjugate Orthogonal Conjugate Gradient method

The well-known projection type methods, such as the Conjugate Gradients method [3],[5], the Conjugate Gradients-Squared method [6] and GMRES [4] are based on forming an (orthogonal) basis for the so-called Krylov subspace. The j-dimensional Krylov subspace $K^j(A; r_0)$ for a linear system $Ax = b$ with non-singular A is defined as the span of the vectors $v_0, Av_0, \ldots, A^{j-1}v_0$ for a given initial residual $v_0 = r_0 = b - Ax_0$.
For a Hermitian matrix ($A = A^H = \bar{A}^T$) such an orthogonal basis can be constructed using a three term recursion relation for the residual vectors, thus forming the basis for the Conjugate Gradients method. However this attractive property is lost when A is only symmetric ($A = A^T$), but not Hermitian. We will show that it is still possible in this case to construct a useful basis for $K^j(A; r_0)$ by means of a three term recurrence relation.
The key in our approach is the replacement of the Hermitian orthogonality between the residual vectors by a conjugate orthogonality relation:

$$(\bar{v}_j, v_k) = 0, \quad \text{if} \quad j \neq k. \tag{1}$$

Here $(\cdot,\cdot)$ denotes the standard Hermitian inner product for complex vectors:

$$(x, y) = \sum_j \bar{x}_j y_j$$

Note that $(\bar{x}, Ay) = (\bar{A}^T\bar{x}, y) = (\overline{Ax}, y) = (\bar{y}, Ax)$. Now let the sequence $\{v_j\}$ be generated as follows:

1. v_0 is the starting vector: $v_0 = r_0 = Ax_0 - b$; $v_{-1} = 0$; $\beta_0 = 0$
2. $v_{j+1} = Av_j + \alpha_j v_j + \beta_j v_{j-1}$ for $j \geq 1$, where α_j, β_j follow from $(\bar{v}_{j+1}, v_j) = 0$ and $(\bar{v}_{j+1}, v_{j-1}) = 0$ respectively.

Then, obviously, the vectors $\{v_0, \ldots, v_j\}$ form a basis for $K^{j+1}(A; r_0)$. For $0 \leq k \leq j-2$ it follows that

$$(\bar{v}_{j+1}, v_k) = (\bar{v}_k, Av_j + \alpha_j v_j + \beta_j v_{j-1}) = (\bar{v}_k, Av_j) = (\bar{v}_j, Av_k) =$$

$$= (\bar{v}_j, v_{k+1} - \alpha_k v_k - \beta_k v_{k-1}) = 0.$$

As the other cases are obvious by construction, we have now shown that the vectors v_j are conjugate orthogonal.
The relations for the v_j can be written in matrix notation as

$$AV_j = V_j T_j + v_{j+1} e_j^T, \quad \text{with}$$

$$T_j = \begin{pmatrix} -\alpha_0 & -\beta_1 & & & \\ 1 & -\alpha_1 & -\beta_2 & & \\ & 1 & -\alpha_2 & \ddots & \\ & & \ddots & \ddots & -\beta_j \\ & & & 1 & -\alpha_j \end{pmatrix}$$

V_j is the matrix which has $v_0, \ldots, v_j$ as its columns and e_j is the j-th complex basis vector in $\mathbb{C}^{j+1}$.
The basis $v_0, \ldots, v_j$ can be used to construct a solution x_{j+1} of the linear system in $K^{j+1}(A; r_0)$ which is chosen such that $r_{j+1} = Ax_{j+1} - b$ is conjugate orthogonal to $K^{j+1}(A; r_0)$. Then automatically $r_{j+1} = Ax_{j+1} - b \in K^{j+2}(A; r_0)$, because, since $Ax_0 - b \in K^1(A; r_0) \subset K^{j+1}(A; r_0)$, it follows that $Ax_{j+1} - b = A(x_{j+1} - x_0) + r_0 \in K^{j+2}(A; r_0)$ and by requiring that $Ax_{j+1} - b$ is conjugate orthogonal to $K^{j+1}(A; r_0)$ we have that $r_{j+1} = Ax_{j+1} - b = \gamma_{j+1} v_{j+1}$ for some scalar γ_{j+1}. (It is evident that $v_{j+1} = 0$ implies that we have reached the solution.) It can be seen that as long as $V_j^T V_j T_j$ is nonsingular there is a unique y_{j+1} in $\mathbb{C}^{j+1}$ such that

0018-9464/90/0300-0706$01.00 © 1990 IEEE

$$\gamma_{j+1}v_{j+1} + b = Ax_{j+1} = AV_j y_{j+1} = V_j T_j y_{j+1} + v_{j+1}e_j^T y_{j+1} \quad (2)$$

Combined with the requirement that v_{j+1} is conjugate orthogonal to $v_0, \ldots, v_j$ this leads to

$$V_j^T V_j T_j y_{j+1} = V_j^T b \quad (3)$$

Note that $V_j^T V_j$ is a diagonal matrix with $(\bar{v}_{k-1}, v_{k-1})$ as its k-th diagonal element. Since $(\bar{v}_j, v_j)$ may be zero for $v_j \neq 0$, one cannot be assured that $V_j^T V_j$ is non-singular. We suggest to restart the process when a vector v_j is encountered for which $(\bar{v}_j, v_j)/(v_j, v_j)$ is too small, or to switch to some other (but more expensive) iteration process, such as, e.g., GMRES [4].
Assuming then that $V_j^T V_j$ is non-singular, y_{j+1} may be solved from

$$T_j y_{j+1} = (V_j^T V_j)^{-1} V_j^T b \quad (4)$$

Note that, when starting with $x_0 = 0$, the latter relation reduces to

$$T_j y_{j+1} = e_1,$$

where e_1 is the first unit vector in $\mathbf{C}^{j+1}$. Either one of these systems can be solved by standard Gauss elimination (with partial pivoting!) or the QR method (Givens rotations). Pivoting may be necessary since T_j is not necessarily similar to a positive definite matrix as in the case when A is Hermitian. Once y_{j+1} has been determined, x_{j+1} can be computed as $x_{j+1} = V_j y_{j+1}$. When T_j is nonsingular then y_{j+1} is unique.
Equation (3) for y_{j+1} is equivalent with the relation

$$V_j^T(Ax_{j+1} - b) = 0,$$

which expresses the fact that $Ax_{j+1} - b$ is orthogonal to $\bar{v}_0, \bar{v}_1, \ldots, \bar{v}_j$. This is a special case of Petrov-Galerkin conditions for x_{j+1}. In the standard conjugate gradients procedure, for Hermitian A, these conditions are replaced by the Ritz-Galerkin conditions $V_j^H(Ax_{j+1} - b) = 0$ (for different vectors v_j of course). If v_k=0 for some k the method finds the exact solution after k steps (provided that $V_k^T V_k$ is nonsingular).
If we assume that no partial pivoting for T is required, it is easily seen that the new iteration process is equivalent with the conjugate gradients procedure in which the (complex Hermitian) inner product has been replaced by the bilinear form

$$[x, y] = \sum x_j y_j = (\bar{x}, y)$$

For this reason we have named this process the Conjugate Orthogonal Conjugate Gradients method (COCG). For the solution of $Ax = b$ with a preconditioner K, it reads, in its simplest form (with the above assumptions), like:

x_0 given; $v_0 = b - Ax_0$;
$p_{-1} = 0$; $\beta_{-1} = 0$;
$w_0 = K^{-1}v_0$
$\varrho_0 = (v_0, w_0)$
For $j = 0, 1, 2, \ldots$
  $p_j = w_j + \beta_{j-1}p_{j-1}$
  $u_j = Ap_j$
  $\mu_j = (\bar{u}_j, p_j)$; if $\mu_j = 0$ then **quit** (failure)
  $\alpha_j = \varrho_j/\mu_j$
  $x_{j+1} = x_j + \alpha_j p_j$
  $v_{j+1} = v_j - \alpha_j u_j$
  if x_{j+1} is accurate enough then **quit** (convergence)
  $w_{j+1} = K^{-1}v_{j+1}$
  $\varrho_{j+1} = (\bar{v}_{j+1}, w_{j+1})$; if $|\varrho_{j+1}|$ small then **quit** (failure)
  $\beta_j = \varrho_{j+1}/\varrho_j$
next j

(u_j and w_j can be stored in the same vector.)

The above process has already been used successfully, e.g., in the eddy current code Carmen [2]. This, in fact, has motivated the formulation and the analysis of our method, which was not fully understood until now.

A number of problems remains to be solved. We have assumed that T can be decomposed without any pivoting. However it is doubtful whether this is the case for all problems of interest in which A is only symmetric, but not Hermitian. In case of failure ($\mu_j \approx 0$), we suggest to form T explicitly and to check the decomposition process for T carefully (switching to partial pivoting or QR when necessary). Admittedly, this has the disadvantage that all the vectors v_j have to be stored in order to construct the solution. The storage requirements might be limited by restarting the iteration after a fixed number of steps.
Of course, the above formulated conjugate orthogonal conjugate gradients procedure can be combined with usual precondition techniques as long as the preconditioner K can be written in the form $K = LL^t$, with L a lower triangular matrix, such that the preconditioned matrix $L^{-1}AL^{-T}$ is still symmetric.
We have observed that in relevant situations the COCG process has a convergence behaviour which is similar to the biconjugate gradients method. Since COCG requires only about half the amount of work per iteration as BiCG, this implies that COCG takes about half the cpu time required for BiCG.
Other competitors of the new method could be GMRES [4] or CGS [6]. In GMRES an orthogonal basis for the Krylov subspace is generated implicitly, so that GMRES may be expected to show better convergence results. However, since the orthogonalization process in GMRES becomes increasingly expensive, for growing iteration count, and since the amount of computer storage for GMRES also grows linearly, it is often advantageous to use COCG (and to switch to GMRES in case of failure). In our experience CGS gave no advantage over COCG.

Numerical examples

We will illustrate our method by comparing it with BiCG and CGS for some eddy current examples. The next table shows the number of iterations for the three methods, both in single and double precision complex arithmetic. An incomplete Choleski decomposition was used as a preconditioner.

Problem	Number of equations	Complex*8 COCG	Complex*8 BiCG	Complex*16 COCG	Complex*16 CGS
Sphere	184	10	10	10	5
Videohead	1414	>1414	>1414	224	372
Coil on plate	3724	109	129	64	63
MRI	6768	44	115	44	30
Crankshaft	10415	178	192	123	89

Table 1: Number of iterations for different problems, methods

The problems used for testing were: "Sphere": a simple conducting sphere in a homogeneous magnetic AC field. "Videohead" is a section of a rectangular videohead including the gap, showing large aspect ratio. "Coil on plate" is a copper plate with a slightly tilted coil just above it. "MRI" is a thin aluminum shield around a complicated system of coils. "Crankshaft" is a rather complicated conducting crankshaft in a magnetic field.

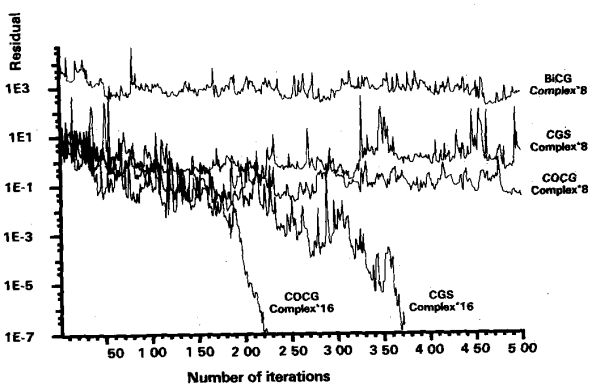

Figure 1: Relative residuals for example "Video"

Figure 1 shows the behaviour of the residuals during the iteration processes for the videohead example. Problems of this type are notorious for their difficulty, caused by large aspect ratios and strong couplings on the metal-air interface. As can be seen from figure 1, the use of double precision complex arithmetic is to be advised strongly for this kind of problems. The reason for this is a severe loss of orthogonality during the process (which has been seen to occur after 3-4 iterations already) in the single precision case. For most examples the COCG method takes about the same number of iterations as the BiCG method, which is twice as expensive. It is only in very favourable instances that the number of CGS iterations is about half the number of COCG iterations (a CGS iteration is also twice as expensive as a COCG iteration), but most of the times it is not competitive.

References

[1] C.S. Biddlecombe, E.A. Heighway, J. Simkin, C.W. Trowbridge: *"Methods for eddy current computation in three dimensions"* IEEE Trans. Mag. **MAG-18**, (1982), p.492-497

[2] C.R.I. Emson, J. Simkin: *"An optimal method for 3-D eddy currents"* IEEE Trans. Mag. **MAG-19**, (1983), p.2450-2452

[3] M.R. Hestenes, E. Stiefel: *"Methods of conjugate gradients for solving linear systems"* Nat. Bur. Standards. J. Res. **49**, (1952) p.409-436

[4] Y. Saad, M.H. Schultz: *"GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems"* SIAM J. Sci. Statist. Comput. **7**, (1986) p.856-869

[5] T.K. Sarkar, X. Yang, E. Arvas: *"A limited survey of various conjugate conjugate gradient methods for solving complex matrix equations arising in electromagnetic wave interactions"* Wave Motion **10**, (1988) p.527-546

[6] P. Sonneveld: *"CGS: a fast Lanczos-type solver for nonsymmetric linear systems"* SIAM J. Sci. Statist. Comput. **10**, (1989) p.36-52