

VARIATIONAL SOLVER FOR ELLIPTIC PROBLEMS IN ATMOSPHERIC FLOWS

PIOTR K. SMOLARKIEWICZ*, LEN G. MARGOLIN**

We discuss a conjugate gradient type method — the conjugate residual — suitable for solving linear elliptic equations that result from discretization of complex atmospheric dynamical problems. Rotation and irregular boundaries typically lead to nonself-adjoint elliptic operators whose matrix representation on the grid is definite but not symmetric. On the other hand, most established methods for solving large sparse matrix equations depend on the symmetry and definiteness of the matrix. Furthermore, the explicit construction of the matrix can be both difficult and computationally expensive. An attractive feature of conjugate gradient methods in general is that they do not require any knowledge of the matrix; and in particular, convergence of conjugate residual algorithms do not rely on symmetry for definite operators. We begin by reviewing some basic concepts of variational algorithms from the perspective of a physical analogy to the damped wave equation, which is a simple alternative to the traditional abstract framework of the Krylov subspace methods. We derive two conjugate residual schemes from variational principles, and prove that either definiteness or symmetry ensures their convergence. We discuss issues related to computational efficiency and illustrate our theoretical considerations with a test problem of the potential flow of a Boussinesq fluid flow past a steep, three-dimensional obstacle.

1. Introduction

The differential equations modeling of many important meteorological phenomena often support high-speed propagating gravity or acoustic waves, and can become prohibitively expensive to solve using explicit numerical integration methods. A common alternative is to formulate such problems implicitly. This in turn leads to a linear elliptic problem, often with complex boundary conditions. The model elliptic equation, representative of those that arise in atmospheric applications, may be written as

$$\mathcal{L}(\phi) - R = 0 \tag{1a}$$

where the operator \mathcal{L} takes, in continuous notation, the form

$$\mathcal{L}(\phi) = \nabla \cdot (\mathbf{P}\mathbf{F}(\phi)) - A\phi \tag{1b}$$

* National Center for Atmospheric Research, Boulder, Colorado 80307. The National Center for Atmospheric Research is sponsored by the National Science Foundation.

** Los Alamos National Laboratory, Los Alamos, New Mexico 87545. Los Alamos National Laboratory is operated by the University of California for the U.S. Department of Energy.

Each of the $M \leq 3$ components of the pressure force vector \mathbf{PF} may depend both on the pressure ϕ and its spatial partial derivatives

$$PF^I(\phi) = \sum_{J=1}^M C^{I,J} \frac{\partial \phi}{\partial x^J} + D^I \phi \quad (1c)$$

where A , R , $C^{I,J}$ and D^I are known functions of the coordinates. The boundary conditions of interest include periodic, Dirichlet (i.e., ϕ is assumed to be known at the boundary), or Neumann type (i.e., the normal component of \mathbf{PF} is assumed known at the boundary).

There are several difficulties associated with inverting (1) for ϕ on discrete meshes. First, modern prognostic atmospheric models employ large grids ($\sim 10^6$ grid points) so that solving (1) at each of numerous time steps represents a serious computational task. Second, in atmospheric applications the discretized operator \mathcal{L} frequently does not possess certain regularity properties (discussed later in this paper) that makes some otherwise attractive methods for (1) inadequate. Third, there are only a few classes of methods that are capable of handling (1) at a reasonable computational cost within the context of a prognostic dynamical code. These methods, however, are fairly recent and not yet well-established in atmospheric modeling.

The computational complexity of the problem in (1), would make generally available software packages particularly attractive. However, even the most powerful and advanced multigrid software package (MUDPACK) that is available nowadays to the atmospheric community (Adams, 1989; 1991; 1993; Adams *et al.*, 1992) is inadequate for the general case of (1). This is not due to any inadequacy of multigrid techniques — MUDPACK could be customized for (1); John Adams, personal communication — but rather reflects the difficulties associated with designing general, efficient, yet user-friendly software. Furthermore, powerful software packages are usually written in terms of matrix inversion and, in a general three-dimensional case, may require storing up to 27 matrices of coefficients (Kapitza and Eppel, 1985). Also, such software packages often require some form of initialization, which becomes computationally intensive if $C^{I,J}$ and D^I are time-dependent — as is usually the case in implicit elastic codes, as well as anelastic codes in time-dependent geometry.

The difficulties associated with the numerical solution of (1) are recognized in the atmospheric literature as a serious drawback of those fundamental formulations of the equations of motion that lead to complex elliptic problems (Durrant, 1989; Skamarock and Klemp, 1992). This situation, where numerical difficulties limit one's abilities to investigate potentially attractive formulations of physical problems, is annoying. The current study seeks to overcome such limitations by documenting a simple, yet powerful scheme capable of solving (1) for a variety of complex atmospheric applications. The strategy adopted here differs from that in the mathematical literature where (1) is usually discussed in terms of solving a large sparse matrix equation. Here we keep in mind that (1) results from a discretization of the atmospheric equations of motion, and so constitutes an element of a much larger problem. Under these conditions, several considerations enter into the choice of a solver or the underlying technique on which a solver should be based. First, the solver should strictly preserve the symme-

tries of the solution¹. Second, the overall accuracy and efficiency of the solver need to be evaluated in the context of the entire dynamical model, and not merely for elementary test problems. Insofar as elementary tests (e.g., the constant-coefficient Poisson equation with Dirichlet boundary conditions) are concerned, the scheme discussed below is not expected to be competitive with generally available software such as MUDPACK or FISHPACK (Swarztrauber and Sweet, 1975). However, this same scheme becomes highly competitive when applied to complex dynamical codes, where neither MUDPACK nor FISHPACK can be implemented in a straightforward manner, *viz.*, their implementation may require substantial effort (e.g., block iterations or predictor-corrector procedures to accommodate the cross-derivatives appearing in (1)) which is both cumbersome and unnecessary, providing (1) can be solved in its generic form.

The scheme discussed in this report is a variant of conjugate gradient (CG) methods known as conjugate residuals (CR). Although CG methods have a long tradition in the mathematical literature (see Ashby *et al.*, 1990; Stoer, 1983; for comprehensive reviews), they have become important in practical applications only recently. In the meteorological literature, CG methods were primarily considered in the context of objective analysis and data assimilation (see Ramamurthy and Navon, 1992; for an overview). Insofar as computational fluid dynamics is concerned, we are only aware of two recent works (Kapitza and Eppel, 1992; Smith *et al.*, 1992) that discuss the successful implementation of the CG approach to modeling geophysical flows. It is important to realize that there are many different versions of CG schemes which, from the viewpoint of basic methodology, often differ only by details. In order to expose the flexibility of the approach and available degrees of freedom, in Section 2 we review some basic concepts of variational elliptic solvers. We begin our discussion with the three-term recurrence formula, following the development of Birkhoff and Lynch (1984, Chapter 5). Although this is a rather unconventional approach in the context of CG methods, we believe it is particularly illustrative in the context of atmospheric modeling. In Section 3, we focus attention on an algorithm suitable for atmospheric applications and discuss two alternate formulations of the CR scheme. Sections 2 and 3 are devoted to the mathematical foundations of variational methods. The practical issue of computational efficiency is addressed in Section 4, where we illustrate our theoretical considerations with examples of three-dimensional potential flows of Boussinesq fluid past a steep, isolated hill.

2. Variational Solvers

Iterative solvers for (1) are often derived from the physical analogy to the damped wave equation

$$\frac{\partial^2 \phi}{\partial \tau^2} + \frac{1}{T} \frac{\partial \phi}{\partial \tau} = \mathcal{L}(\phi) - R \quad (2)$$

whose steady-state solution in pseudo-time τ represents the solution to the governing problem (1) (cf. Birkhoff and Lynch, 1984, Chapter 4.15). Discretizing (2) in pseudo-

¹ Our experience with MUDPACK indicates that schemes based on Gauss-Seidel relaxation technique may generate small asymmetries in solutions, which is undesirable when studying such phenomena as the growth of physical instabilities in a direction normal to the mean flow.

time with an increment $\Delta\tau$ and damping scale $T \equiv \eta^{-1}\Delta\tau$ (both assumed uniform in space) and using, respectively, centered- and one-sided differencing for the first and second term on the lhs of (2), leads to a three-term recurrence formula

$$\phi^{n+1} = \gamma\phi^n + (1 - \gamma)\phi^{n-1} + \beta(\mathcal{L}(\phi^n) - R) \quad (3)$$

where $\gamma = (2 + \eta)/(1 + \eta)$, $\beta = (\Delta\tau)^2/(1 + \eta)$, and the superscript n labels successive iterations. Spatial discretization of (3) leads to the equation

$$\phi_{\mathbf{i}}^{n+1} = \gamma\phi_{\mathbf{i}}^n + (1 - \gamma)\phi_{\mathbf{i}}^{n-1} + \beta(\mathcal{L}_{\mathbf{i}}(\phi^n) - R_{\mathbf{i}}) \quad (3')$$

where the subscript $\mathbf{i} = (i^1, \dots, i^M)$ denotes position on a computational grid ($i^J = 1, \dots, N^J$), and $\mathcal{L}_{\mathbf{i}}$ refers to discrete forms of (1b) and (1c). For simplicity of notation, we shall further refer to (3) with understanding that it implies (3') on the grid.

The recurrence formula in (3) constitutes the common basis for many iterative schemes. Depending on the choice of the parameters γ and β , (3) becomes, for instance, Richardson (*viz.* Jacobi), Frankel, Chebyshev, or a CG technique (Birkhoff and Lynch, 1984). The underlying goal of all possible schemes embodied by (3) is computational efficiency — i.e., convergence to a steady state in as few iterations as possible, in order to minimize the effort associated with costly evaluations of $\mathcal{L}(\phi)$ on the mesh. In terms of the physical analogy to (2), efficiency implies advancing the solution in pseudo-time τ as far as possible with the largest $\Delta\tau$ admitted by the computational stability of the scheme. In this context, the Richardson method ($\gamma = 1$) reduces, in essence, to numerical evaluation of the diffusion equation with the maximal constant β admitted by the linear stability of the scheme, whereas the Frankel or Chebyshev algorithms integrate (2) with, respectively, constant or variable β and γ determined by the spectral properties of the \mathcal{L} operator. Although the latter technique may be quite efficient for simple elliptic problems (cf. Potter 1973, Section IV.6.e), it is of little use in the general case of (1), when the spectral structure of \mathcal{L} is difficult to determine.

As in the Chebyshev method, CG schemes integrate (2) with $\Delta\tau$ and T variable in pseudo-time; however, the resulting parameters γ and β in (3) are determined from variational principles. Defining a local residual error at the n -th iteration as

$$r^n \equiv \mathcal{L}(\phi^n) - R \quad (4)$$

equation (3) may be rewritten as

$$\phi^{n+1} = \gamma\phi^n + (1 - \gamma)\phi^{n-1} + \beta r^n \quad (5)$$

The requirement that ϕ^{n+1} satisfies periodic, Dirichlet, or Neumann boundary conditions (given their fulfillment at preceding iterations) implies, respectively: periodic, $r^n = 0$, or the normal component of $\mathbf{PF}(r^n) = 0$ boundary conditions on r^n . Acting with the *linear* operator \mathcal{L} on both sides of (5), assuming spatial *homogeneity* of γ and β , and adding $-R$ to both sides of the resulting equation, leads to the recurrence formula

$$r^{n+1} = \gamma r^n + (1 - \gamma)r^{n-1} + \beta\mathcal{L}(r^n) \quad (6)$$

which predicts residual errors $(\mathcal{L}(\phi^{n+1}) - R)$ one iteration ahead of (5). Consequently, (6) may be employed to determine an optimal selection of γ and β in (5) which assures a minimal total error after $n + 1$ iterations. In essence, this is the basic idea of CG methods. Apparently, many different variational schemes can be designed depending on the means of error minimization (norm selection), the temporal discretization in (2) (e.g., ϕ^n in $\mathcal{L}(\phi)$ on the rhs of (3) may be replaced, in the spirit of Adams-Bashforth approximations, by a linear combination of ϕ^n and ϕ^{n-1} thereby introducing an additional degree of freedom), or the order of the wave equation (2) (i.e., the lhs of (2) may be further expanded by including higher-order derivatives with different pseudo-time scales).

The classic CG algorithm for symmetric negative definite² operators \mathcal{L} is due to Hestenes and Stiefel (1952). It relies on powerful theorems (see Birkhoff and Lynch, 1984, Chapter 5.3) that assure the minimization of the error $\langle -(\phi^{n+1} - \bar{\phi})\mathcal{L}(\phi^{n+1} - \bar{\phi}) \rangle$ and convergence of (5) to the exact (in the absence of roundoff errors) solution $\bar{\phi} = \mathcal{L}^{-1}(R)$ in a finite number of iterations³ for any initial guess ϕ^0 if and only if the residuals of any two $k \neq l$ iterates are orthogonal in the sense

$$\sum_i r_i^k r_i^l \equiv \langle r^k r^l \rangle = 0 \tag{7}$$

(see Birkhoff and Lynch, 1984, Chapter 5.3)

An important corollary is the \mathcal{L} -orthogonality of the differences of subsequent iterates:

$$\forall_{k \neq l} \langle (\phi^{k+1} - \phi^k)\mathcal{L}(\phi^{l+1} - \phi^l) \rangle = 0 \tag{8}$$

The relationship (7) leads immediately to optimized coefficients γ and β in (5): redefining $\beta \equiv \beta' \gamma$ and requiring $\langle r^{n+1} r^n \rangle = \langle r^{n+1} r^{n-1} \rangle = 0$ results in

$$\beta' = -\frac{\langle r^n r^n \rangle}{\langle r^n \mathcal{L}(r^n) \rangle} \tag{9a}$$

and

$$\gamma = \frac{\langle r^{n-1} r^{n-1} \rangle}{\langle r^{n-1} r^{n-1} \rangle - \beta' \langle r^{n-1} \mathcal{L}(r^n) \rangle} \tag{9b}$$

which together with (5) and (6) completes the algorithm.

The traditional form of CG schemes as they usually appear in the literature can be recovered from (5) after some manipulations. We rewrite (5) as

$$\phi^{n+1} = \phi^n + \beta^n \left(\frac{(\gamma^n - 1)\beta^{n-1}}{\beta^n} \cdot \frac{(\phi^n - \phi^{n-1})}{\beta^{n-1}} + r^n \right) \equiv \phi^n + \beta^n (\alpha^n p^{n-1} + r^n)$$

² In order to associate \mathcal{L} with the diffusion-type operator, we refer to negative definiteness rather than, as is more traditional in the mathematical literature, positive definiteness, and adjust signs accordingly (e.g., note the definition of the residual in (4)).

³ Theoretically, this makes the CG algorithm a direct rather than an iterative method; see Birkhoff and Lynch for a discussion.

where $\alpha^n \equiv (\gamma^n - 1)\beta^{n-1}/\beta^n$, $p^n \equiv (\phi^{n+1} - \phi^n)/\beta^n$, and the superscripts appearing on γ , β , and α refer to values of the coefficients at different iterations. This leads to the algorithm in the form

$$\phi^{n+1} = \phi^n + \beta^n p^n \quad (10a)$$

$$r^{n+1} = r^n + \beta^n \mathcal{L}(p^n) \quad (10b)$$

$$p^{n+1} = \alpha^{n+1} p^n + r^{n+1} \quad (10c)$$

Employing the orthogonality relationships and the symmetry of \mathcal{L} , the coefficients β^n and α^{n+1} can be recovered either from (9a) and (9b) or, more directly, from (10b) and (10c) by requiring $\langle r^{n+1} r^n \rangle = \langle p^{n+1} \mathcal{L}(p^n) \rangle = 0$. After some manipulations (see appendix A for details), the latter relations lead to

$$\beta^n = - \frac{\langle r^n r^n \rangle}{\langle p^n \mathcal{L}(p^n) \rangle} \quad (11a)$$

$$\alpha^{n+1} = \frac{\langle r^{n+1} r^{n+1} \rangle}{\langle r^n r^n \rangle} \quad (11b)$$

which complete the classic CG scheme (cf. Hestenes and Stiefel, 1952, algorithm 3.1). Several different but mathematically equivalent expressions can be derived by exploiting (7), (8) and the symmetry of the operator \mathcal{L} (cf. Hestenes and Stiefel, 1952, Theorem 5.5).

The CG schemes discussed above strongly depend on the symmetry of the elliptic operator \mathcal{L} . In essence, they derive from the minimization of the “energy” norm $\langle \phi^n R \rangle - \langle \frac{1}{2} \phi^n \mathcal{L}(\phi^n) \rangle$, whose minimum (for negative definite \mathcal{L}) is equal to the exact solution $\bar{\phi}$ only if the relationship $\langle \xi \mathcal{L}(\zeta) \rangle = \langle \zeta \mathcal{L}(\xi) \rangle$ holds for all fields ξ and ζ (cf. Birkhoff and Lynch, 1984, Chapter 4.9). If this relation, heavily exploited in the proofs of key theorems and in the derivation of (11), does not hold, i.e., if \mathcal{L} is not symmetric, then the schemes in eqs. (5), (6), (9) and eqs. (10)–(11) are not equivalent, and their convergence is not ensured. In typical atmospheric applications, the discrete counterparts of (1b) and (1c) are usually negative definite (i.e., the governing equation (1) is elliptic) but are rarely symmetric. Frequently, the lack of the symmetry is due to difficulties in imposing free-slip, rigid-lid conditions at irregular lower boundaries (we shall return to this point later in this paper).

The CR schemes overcome such difficulties, in essence, by a direct minimization of the $\langle r^{n+1} r^{n+1} \rangle$ norm in (5), which leads to algorithms as in (5) and (6) or (10a), (10b) and (10c) but with different coefficients β and γ or β and α , respectively. In the next section, we shall discuss two alternate forms of a CR solver convenient for atmospheric applications. As in the classic CG scheme discussed above, we will start with the three-term recurrence formulae (5) and (6), and then follow with the traditional form (10).

3. Conjugate Residual Schemes

Based on semi-intuitive considerations about the accumulation of roundoff errors, we anticipate that certain groupings of the terms in the scheme, as well as certain forms of the coefficients γ and β in (5) and (6) will be more attractive than other, mathematically equivalent forms. From this point, therefore, all key formulae will take exactly the same form as they appear in the numerical codes discussed in Section 4 of this study.

The recurrence formulae (5) and (6) are rewritten as

$$\phi^{n+1} = \gamma(\phi^n - \phi^{n-1}) + \phi^{n-1} + \beta r^n \quad (12a)$$

$$r^{n+1} = \gamma(r^n - r^{n-1}) + r^{n-1} + \beta \mathcal{L}(r^n) \quad (12b)$$

The coefficients γ and β are determined from the minimization requirements

$$\frac{\partial}{\partial \gamma} \langle r^{n+1} r^{n+1} \rangle = \frac{\partial}{\partial \beta} \langle r^{n+1} r^{n+1} \rangle = 0 \quad (13)$$

which combined with (12b) lead to the orthogonality relationships

$$\langle r^{n+1}(r^n - r^{n-1}) \rangle = \langle r^{n+1} \mathcal{L}(r^n) \rangle = 0 \quad (14)$$

The relations in (14) imply an inhomogeneous system of two linear equations for γ and β :

$$AA \cdot \gamma + AB \cdot \beta = -AC \quad (15a)$$

$$AB \cdot \gamma + BB \cdot \beta = -BC \quad (15b)$$

with the corresponding solutions

$$\gamma = \frac{(BC \cdot AB - AC \cdot BB)}{(AA \cdot BB - AB \cdot AB)} \quad (16a)$$

$$\beta = \frac{(AB \cdot AC - AA \cdot BC)}{(AA \cdot BB - AB \cdot AB)} \quad (16b)$$

and the definitions of the coefficients

$$AA \equiv \langle (r^n - r^{n-1})(r^n - r^{n-1}) \rangle \quad (17a)$$

$$AB \equiv \langle (r^n - r^{n-1}) \mathcal{L}(r^n) \rangle \quad (17b)$$

$$AC \equiv \langle (r^n - r^{n-1}) r^{n-1} \rangle \quad (17c)$$

$$BB \equiv \langle \mathcal{L}(r^n) \mathcal{L}(r^n) \rangle \quad (17d)$$

$$BC \equiv \langle \mathcal{L}(r^n) r^{n-1} \rangle \quad (17e)$$

For any initial guess ϕ^0 at $n = 0$, γ is set to unity, which implies $\beta = -\frac{\langle r^0 \mathcal{L}(r^0) \rangle}{\langle \mathcal{L}(r^0) \mathcal{L}(r^0) \rangle}$ from the minimization of (12b). Here $r^0 = \mathcal{L}(\phi^0) - R$ is computed from the definition in (4); also r^{-1} and ϕ^{-1} are set to zero, for convenience of coding. By design, γ and β in (16) minimize the norm of a total residual error without invoking the symmetry of \mathcal{L} . In appendix B, we provide an elementary proof of the monotone convergence of (12) that relies on the negative-definiteness of, but does not require the symmetry of the operator \mathcal{L} . Conversely, the convergence of CR schemes discussed in the literature (cf. Ashby *et al.*, 1990, Section 5.2) usually invokes symmetry of the operator but does not require its definiteness. Apparently, either symmetry or definiteness is sufficient for the convergence of a CR scheme. This flexibility is rarely noted in the literature where symmetry is frequently assumed *a priori*.

Using (12b) and the orthogonality relationships (14), one can derive a number of auxiliary relations and different forms for γ and β that hold in absence of roundoff error. Employing those relations, one can eliminate two of the five sums in (17) (see appendix B). In our experience, however, it is not always worthwhile to employ analytic relationships to simplify (17a)–(17e). On vector machines, the elimination of two sums in large dynamical codes results in a negligible speed-up of computations, whereas it may slow down the convergence due to the accumulation of roundoff errors.

As in the classic CG scheme discussed in the preceding section, the alternate, traditional form of the CR algorithm can be derived directly from (10) by requiring

$$\frac{\partial}{\partial \beta^n} \langle r^{n+1} r^{n+1} \rangle = \frac{\partial}{\partial \alpha^{n+1}} \langle r^{n+2} r^{n+2} \rangle = 0 \quad (18)$$

The minimization requirements (18) combined with (10b) lead to the orthogonality relationships

$$\langle r^{n+1} \mathcal{L}(p^n) \rangle = \langle r^{n+2} \mathcal{L}(p^n) \rangle = 0 \quad (19)$$

which imply

$$\beta^n = -\frac{\langle r^n \mathcal{L}(p^n) \rangle}{\langle \mathcal{L}(p^n) \mathcal{L}(p^n) \rangle} \quad (20a)$$

$$\alpha^{n+1} = -\frac{\langle \mathcal{L}(r^{n+1}) \mathcal{L}(p^n) \rangle}{\langle \mathcal{L}(p^n) \mathcal{L}(p^n) \rangle} \quad (20b)$$

straightforwardly from (10b) and (10c). The derivation of (20b) makes use of an important relation

$$\mathcal{L}(p^{n+1}) = \mathcal{L}(r^{n+1}) + \alpha^{n+1} \mathcal{L}(p^n) \quad (21)$$

which results from acting with \mathcal{L} on both sides of (10c). Note that (20b) requires both $\mathcal{L}(p^n)$ and $\mathcal{L}(r^{n+1})$ in every iteration. This double evaluation of \mathcal{L} can be avoided by using (21), which expresses $\mathcal{L}(p^k)$ through its value at preceding iteration and

the current value of $\mathcal{L}(r^k)$. Similar to the algorithm in (17), (16) and (12) (hereafter CR1), the current scheme ((20a), (10a), (10b), (20b), (10c) and (21); hereafter CR2) starts with the minimization of $r^1 = r^0 + \beta \mathcal{L}r^0$ which is realized, for any initial guess ϕ^0 , by setting $r^0 = p^0$ and evaluating r^0 from its definition in (4). By design, the coefficients β and α in (20) minimize the norm of a total residual error and assure the monotone convergence of iterations without invoking the symmetry of \mathcal{L} . As with the CR1 scheme in appendix B, the proof of convergence reduces to showing that $\langle r^{n+1}, r^{n+1} \rangle = \langle r^n, r^n \rangle + \beta^n \langle r^n, \mathcal{L}(r^n) \rangle$ and that $\beta^n \geq 0$, with the equality holding if and only if $\langle r^n, r^n \rangle = 0$.

The discussion so far documents that many forms of either similar or equivalent variational schemes are possible. The CR2 form appears in Eisenstat *et al.* (1983), whereas Stoer (1983) discusses some other forms that can be obtained from CR2 after employing (19), (21) and assuming the symmetry of \mathcal{L} . The CR1 formulation is definitely less popular, as we have not encountered it in the literature. Insofar as applications are concerned, CR1 and CR2 are essentially equivalent and, in our experience, usually lead to identical results. The CR2 scheme is somewhat more efficient than CR1 as it requires evaluating two fewer sums per iteration. On the other hand, CR1 has the advantage of being a straightforward derivative of the damped wave equation (2) and, therefore, provides a natural basis for further developments. In our research of CG methods we have also considered higher-order wave equation augmentations of (1), which lead to more-than-three term recurrence formulae (5) and (6) and form the basis of the truncated *orthomin* methods (see Eisenstat *et al.*, 1983; and Stoer, 1983; for discussions). Although some of those schemes do result in a slightly faster asymptotic convergence rate than that in CR1 and CR2, overall, we have not found them advantageous for applications. Their somewhat better performance in nontrivial convergence tests (potential flows of density-stratified fluids past two- and three-dimensional mountains) has been usually offset by the increase in computational effort that is required. In the next section we shall elaborate on the performance of the advocated algorithms, and their utility for atmospheric applications.

4. Computational Efficiency of CG Schemes

4.1. Introductory Remarks

In assessing the computational efficiency of elliptic solvers, it is customary to consider both the storage required and the number of arithmetic operations involved as well as to evaluate their asymptotic convergence rates. However, such quantities may not be the best measure of efficiency in the context of a dynamical code used for a diversity of applications. The CR schemes described above require storage for four auxiliary fields; however the application models addressed do not require any additional memory allocation that would not already be available in the code regardless of the solver employed. Furthermore, depending on the design of a particular model, the coefficients of the operator \mathcal{L} may be either preprocessed and stored (requiring up to 27 additional matrices but increasing the efficiency of the scheme), or $\mathcal{L}(\phi)$ may be computed directly from (1b) and (1c) at every iteration. Thus the issue of

computational efficiency is, to a large extent, a function of the code architecture as well as the complexity of the applications addressed.

Insofar as estimates of the convergence rates are concerned, it is important to realize that those are informative only in a relative sense. The convergence of iterative methods depends, in general, on the eigenvalue spectrum of \mathcal{L} . For any given algorithm, convergence may vary substantially for different problems, depending on the temporal and spatial variability of the coefficients in (1), boundary conditions, spatial discretization, and the staggering and size of the mesh. The convergence of variational schemes may be further accelerated by some form of operator preconditioning (see Kapitza, 1988; Smith *et al.*, 1992; Stoer, 1983; for discussions and examples of applications). This adds another dimension to the problem. In essence, preconditioning procedures replace (1) (e.g., by means of operator splitting and/or composition) with a modified governing equation $\mathcal{L}'(\phi) - R' = 0$ that can be more easily inverted on the grid. This has little to do with the CG formalism itself, as it reduces the elliptic problem at hand to a simpler problem to be solved with a CG scheme. In general, the more work done by the preconditioner, the less work needs to be done by the CG algorithm, and vice versa. Powerful preconditioners can substantially complicate the entire solution procedure, but in many applications may lead to significant improvements in the overall efficiency of computations. However, in large prognostic problems (especially where the coefficients of \mathcal{L} evolve in time), the additional operations and storage required by such procedures often offset the increased efficiency due to accelerated convergence. In the applications discussed below, we shall employ a simple preconditioner dictated by the problem's physics.

The relative performance of different CG schemes has been evaluated frequently in the literature (see Kapitza and Eppel, 1987; Saad and Schultz, 1986; Smith *et al.*, 1992; for examples) using a diversity of test problems. Here we compare the classic CG algorithm, the CR2 scheme, and the elementary Jacobi (Richardson) method using an idealized test problem that is representative of atmospheric applications. The Jacobi scheme is included as a reference. Like CG schemes, the Jacobi scheme does not require an explicit knowledge of the matrix resulting from the discretization of \mathcal{L} . Since its convergence rate is known to be half that of the Gauss-Seidel (Liebman) relaxation scheme (cf. Roache, 1972, Section III.B.2), including the Jacobi scheme in our comparison also provides information about the relative performance of the CG schemes compared to basic relaxation techniques. The physical problem we chose is potential flow of a Boussinesq fluid past an isolated three-dimensional mountain, whose height and width are comparable to each other as well as to the depth and horizontal extent of the domain (Fig. 1). This test problem ensures significant variability of the coefficients in (1). Before elaborating on the results of the convergence tests, we will discuss the model design and the formulation of the relevant elliptic problem in some detail. Since the following discussion introduces the background for an anelastic model that will be reported elsewhere, we shall refer to a density-stratified anelastic fluid rather than to a Boussinesq fluid, and will recover the Boussinesq approximation later.

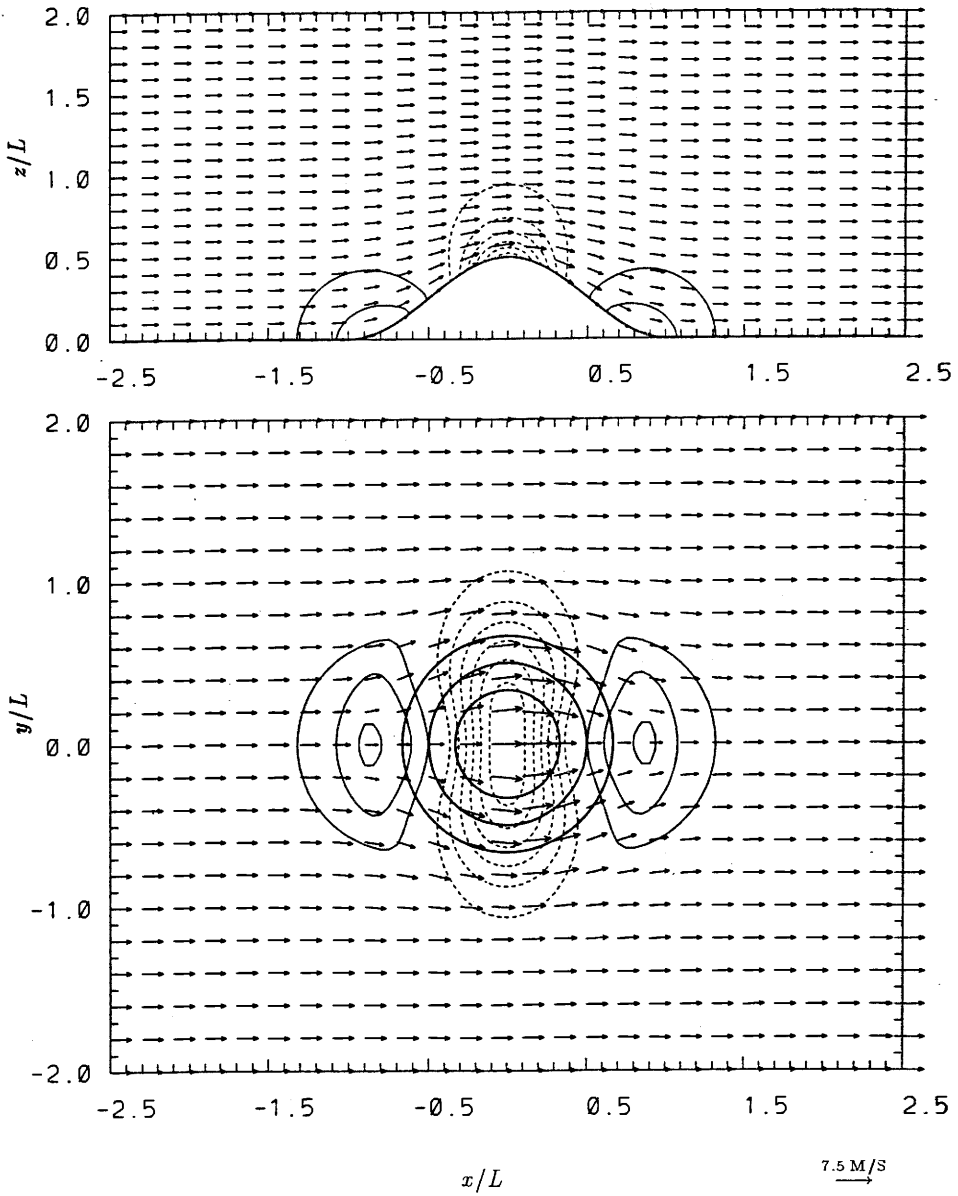


Fig. 1. Potential flow solution after 500 iterations with CR2 scheme on the A grid. The upper and lower panels show the solution at the center plane and surface, respectively. The velocity vectors are superimposed on isolines of Bernoulli pressure perturbation (normalized by the density) displayed with contour interval $0.5 \text{ m}^2 \text{ s}^{-2}$. Solid lines and dashed contours refer to positive and negative field values, respectively; zero contour lines are not shown.

4.2. Model Description

The governing equations of the potential flow of a density-stratified anelastic fluid may be written in the form

$$\mathbf{v} = \mathbf{v}_e - \nabla\phi \quad (22a)$$

$$\nabla \cdot \bar{\rho}\mathbf{v} = 0 \quad (22b)$$

where \mathbf{v}_e , ϕ , and $\bar{\rho}$ denote the ambient velocity, velocity potential, and reference density $\bar{\rho} = \bar{\rho}(z)$ respectively. Equations (22) are cast in a nonorthogonal terrain-following system of coordinates $[x, y, z] = [x_c, y_c, H(z_c - h)/(H - h)]$ with the subscript c referring to Cartesian coordinates, H to a model depth, and $h = h(x_c, y_c)$ to the mountain profile. Since the terrain-following transformation is fairly standard in atmospheric applications, we shall freely introduce the metric coefficients $G^{IJ} = \sum_K (\partial x^I / \partial x_c^K) \cdot (\partial x^J / \partial x_c^K)$ and the Jacobian of the transformation $G = \text{Det}(\partial \mathbf{x}_c / \partial \mathbf{x}) = (\text{Det}(G^{IJ}))^{-1/2}$ and refer the reader interested in further details to Clark (1977), Gal-Chen and Somerville (1975) and the recent work of Kapitza and Eppel (1992). The transformed eqs. (22) take the form

$$u = u_e - \frac{\partial\phi}{\partial x} - G^{13} \frac{\partial\phi}{\partial z} \quad (23a)$$

$$v = v_e - \frac{\partial\phi}{\partial y} - G^{23} \frac{\partial\phi}{\partial z} \quad (23b)$$

$$w = w_e - G^{-1} \frac{\partial\phi}{\partial z} \quad (23c)$$

$$\frac{\partial\rho^*u}{\partial x} + \frac{\partial\rho^*v}{\partial y} + \frac{\partial\rho^*\omega}{\partial z} = 0 \quad (23d)$$

where

$$\rho^* \equiv G\bar{\rho} \quad (24)$$

and the contravariant "vertical" velocity ω in (23d) is related to the covariant velocity components through

$$\omega = G^{-1}w + G^{13}u + G^{23}v \quad (25)$$

The discrete boundary value problem for the velocity potential is formulated in three steps. First, the spatial discretizations of (23a), (23b), and (23c) are substituted into (25); second, the discrete forms of (23a), (23b), and the result of the first substitution are substituted into (23d). These manipulations lead to

$$\nabla \cdot (\rho^* \mathbf{P}\mathbf{F}^*(\phi)) = 0 \quad (26)$$

where the components of the $\mathbf{P}\mathbf{F}^*$ force are

$$PF^*X = u_e - \frac{\partial\phi}{\partial x} - G^{13} \frac{\partial\phi}{\partial z} \quad (27a)$$

$$PFY^* = v_e - \frac{\partial\phi}{\partial y} - G^{23} \frac{\partial\phi}{\partial z} \tag{27b}$$

$$PFZ^* = \omega_e - G^{13} \frac{\partial\phi}{\partial x} - G^{23} \frac{\partial\phi}{\partial y} - G^{33} \frac{\partial\phi}{\partial z} \tag{27c}$$

with

$$\omega_e = G^{-1}w_e + G^{13}u_e + G^{23}v_e$$

and

$$G^{33} = (G^{13})^2 + (G^{23})^2 + 1/G^2$$

In the third step, (26) is premultiplied by $-\frac{1}{\rho^*}$ resulting finally in

$$-\frac{1}{\rho^*} \nabla \cdot (\rho^* \mathbf{PF}^*(\phi)) \equiv -\frac{1}{\rho^*} \nabla \cdot (\rho^* (\mathbf{v}^* - \mathbf{PF}(\phi))) = 0 \tag{28}$$

where $\mathbf{v}^* \equiv (u_e, v_e, \omega_e)$. Equation (28) is a modified form of (1) with $A \equiv D \equiv 0$ where: a) the rhs of (1a) (here, $R = -\nabla \cdot (\rho^* \mathbf{v}^*)$) has been included under the divergence operator in (1b); b) ρ^* has been factored out from C^{IJ} coefficients in (1c) as well as R in (1a); and c) the entire (1a) has been preconditioned with the inverse of ρ^* . The first two modifications have been introduced essentially for the convenience of coding, whereas the third ensures uniform convergence throughout the entire domain of integration regardless of the variability of $\bar{\rho}$ and G . In a future study, we will discuss gravity wave dynamics in deep atmospheres $\sim \mathcal{O}(10^5)$ m, where the density of air decreases to $\sim \mathcal{O}(10^{-4})$ kg m⁻³. There, a simple density preconditioner in (27) assures the same degree of incompressibility (the same accuracy) throughout the entire depth of the model.

The elliptic problem in (28) is supplemented by Neumann boundary conditions on the lower and upper surfaces of the model, and by either periodic or Neumann conditions at the lateral boundaries. The Neumann conditions appear in consequence of specifying the normal velocity components at the boundaries in (22a). By design, they reduce to setting appropriate components of \mathbf{PF}^* in (28) to the specified values. Mathematically, (28) admits any specification of the normal velocities that satisfy the integrability condition $\oint \rho^* \mathbf{PF}^* \cdot \mathbf{n} \, d\sigma = 0$. Physically, for a potential flow past an isolated obstacle, we assume $\omega = 0$ ($PFZ^* = 0$) at the lower and upper boundaries, and $u = u_e$ ($PFX^* = u_e$) and $v = v_e$ ($PFY^* = v_e$) at the lateral Neumann boundaries.

The discrete form of (28) follows from the definitions of the partial derivative operators on the grid. Here, we consider two grid configurations: the unstaggered A-grid, where all variables are defined at the same positions; and the staggered B-grid, where a pressure variable (here the velocity potential) is staggered one-half grid interval in all directions with respect to all other variables defined at the same grid positions. The partial derivative operators are approximated with standard centered-differences appropriate to each grid configuration, e.g.:

$$\left. \frac{\partial\psi}{\partial x} \right|_{i,j,k} = \frac{1}{2\Delta X} (\psi_{i+1,j,k} - \psi_{i-1,j,k}) \tag{29a}$$

and

$$\left. \frac{\partial \psi}{\partial x} \right|_{i,j,k} = \frac{1}{4\Delta X} (\psi_{i+1/2,j+1/2,k+1/2} - \psi_{i-1/2,j+1/2,k+1/2} + \psi_{i+1/2,j-1/2,k+1/2} - \psi_{i-1/2,j-1/2,k+1/2} + \psi_{i+1/2,j+1/2,k-1/2} - \psi_{i-1/2,j+1/2,k-1/2} + \psi_{i+1/2,j-1/2,k-1/2} - \psi_{i-1/2,j-1/2,k-1/2}) \quad (29b)$$

for the A- and B-grids respectively. At the Neumann boundaries, the discrete partial differential operators in (29) are undefined wherever they reach outside the computational domain. A standard procedure at the boundaries is to replace the centered approximations with one-sided difference formulae (cf. Chorin, 1968; Glowinski, 1992) where required. In essence, this is equivalent to retaining the centered-differences in (29) while assuming extrapolated values of a pressure variable (consistent with Neumann conditions) at fictitious points outside the domain. This seemingly minor aspect of model design has important consequences. Because of local anisotropies of the difference formulae, the numerical operator \mathcal{L} may not be symmetric, and spurious vorticity may be generated at the free-slip boundaries. In the general three-dimensional case of (1), this is a nontrivial issue that so far does not seem to have a rigorous yet practical solution (cf. Bernard and Kapitza, 1992)⁴. Here we adopt the following approach. At the Neumann boundaries, we evaluate all but the normal partial derivatives in (27) explicitly from pressure field values available on the grid. On the A-grid, this employs standard centered-differences as in (29a), whereas on the B-grid, the difference formulae in (29b) assume extrapolated pressure values at fictitious points outside the computational domain. The normal derivatives are then computed from the boundary conditions and substituted into (27) for other-than-normal components of \mathbf{PF} . On the A-grid, the divergence operator uses one-sided differencing in the direction normal to the boundary, whereas on the B-grid, it employs centered-differences as in (29b). For either grid, the symmetry of the discrete elliptic operator in (28) is not assured.

The convergence tests discussed in the next section assume a constant $\bar{p} = \rho_o$ characteristic of the Boussinesq approximation ($\rho_o = 1 \text{ kg m}^{-3}$), $[u_e, v_e, w_e] = [5, 0, 0] \text{ m s}^{-1}$, and the axially-symmetric cosine hill $h = 0.5h_o(1 + \cos(\pi r/L))$ if $r = (x^2 + y^2)^{1/2} \leq L$ ($h = 0$ otherwise) with $h_o = 1.5 \cdot 10^3 \text{ m}$, and $L = 3 \cdot 10^3 \text{ m}$. The uniform mesh consists of $NX \times NY \times NZ = 101 \times 81 \times 41$ grid points with grid increment $\Delta X = \Delta Y = \Delta Z = 0.15 \cdot 10^3 \text{ m}$. The boundary conditions are rigid-lid in x and z , and periodic in y .

4.3. Results of Convergence Tests

Here we return to the main topic of Section 4, and elaborate on the relative performance of selected CG schemes within the potential flow model discussed above. The results of a series of tests are summarized in Fig. 2. Although the discussion in Section 3 referred to the norm $\langle r^n r^n \rangle^{1/2}$ of the residual errors, here we display

⁴ In three spatial dimensions, a possible formal solution may require incorporation of six two-dimensional, and twelve one-dimensional additional elliptic equations.

$e = \max \left| \frac{1}{\rho^*} \nabla \cdot (\rho^* \mathbf{v}) \right| \Delta t = \max |r^n| \Delta t$ which is more informative for the physical problem. A time step $\Delta t = 10\text{s}$ has been introduced for the sake of normalization with reference to the Courant number $\mathcal{C} = \|\mathbf{v}\Delta t/\Delta X\| \approx 1$ that would result if the flow were allowed to evolve in time due to an imposed stratification of a typical buoyancy frequency $N = 10^{-2}\text{s}^{-1}$. The evolution of errors is displayed over a range of iteration numbers that is about two orders of magnitude larger than is required in typical applications with prognostic dynamical models. Nevertheless, the asymptotic convergence rates are informative, since the effects of the iterations at subsequent time steps of a prognostic model are cumulative for the slowly evolving scales of motion. Although the discussion below refers to a particular test problem, the conclusions are much more general and reflect our experience with many different applications.

Figure 1 shows the solution after 500 iterations with the CR2 scheme on the A-grid; the isolines of the Bernoulli pressure perturbation, normalized by density,

$$p' = -\frac{1}{2\rho_o} \left(\nabla\phi \cdot \nabla\phi - \langle \nabla\phi \cdot \nabla\phi \rangle \right)$$

are superimposed with the flow vectors. The maximal residual error is essentially zero ($e < 10^{-10}$; Fig. 2a). The solution exhibits an excellent upwind/downwind symmetry, which indicates zero drag on the mountain and, consequently, the lack of spurious vorticity generation at the boundaries. This documents *a posteriori* the adequacy of the finite-difference approximations at the Neumann boundaries, which were discussed in the preceding section. Figure 3 shows the equivalent solution on the B-grid. As with the A-grid, this solution exhibits excellent upwind/downwind symmetry. The maximal residual error $e \approx 10^{-4}$ (Fig. 2b) is six orders of magnitude larger than that on the A-grid, and there are some apparent differences between the two solutions. An obvious question to ask is whether these differences are due to the slower convergence of the iterative solver on the B-grid than on the A-grid (Fig. 2), or rather are due to the different distribution of truncation errors associated with the finite-difference approximations to the partial derivatives in (29). The solution on the B-grid after 5000 iterations (not shown) is hardly distinguishable from that in Fig. 3 even though the residual error is four orders of magnitude smaller (Fig. 2b). This suggests that truncation errors are responsible for the differences in the solutions on the two grids. To substantiate this assertion, we show in Fig. 4 the solution on the A-grid after 100 iterations where the residual error is about the same as that on the B-grid after 500 iterations. Comparing Figs. 1 and 4 clearly shows that the differences between the two solutions are hardly noticeable, the six orders of magnitude difference in the maximal residual error notwithstanding (cf. Fig. 2a). Finally, Fig. 5 shows the solution on the B-grid after 100 iterations where the residual error is about twice of that for the solution in Fig. 4. Comparing this solution with that in Fig. 3 shows small but noticeable differences that are, however, much smaller than those between Figs. 1 and 3—apparently, the latter differences are due to the different discretization formulae in (29).

The discussion above exposes another important aspect of the computational efficiency of CG schemes. Since the expense associated with an iterative solution of

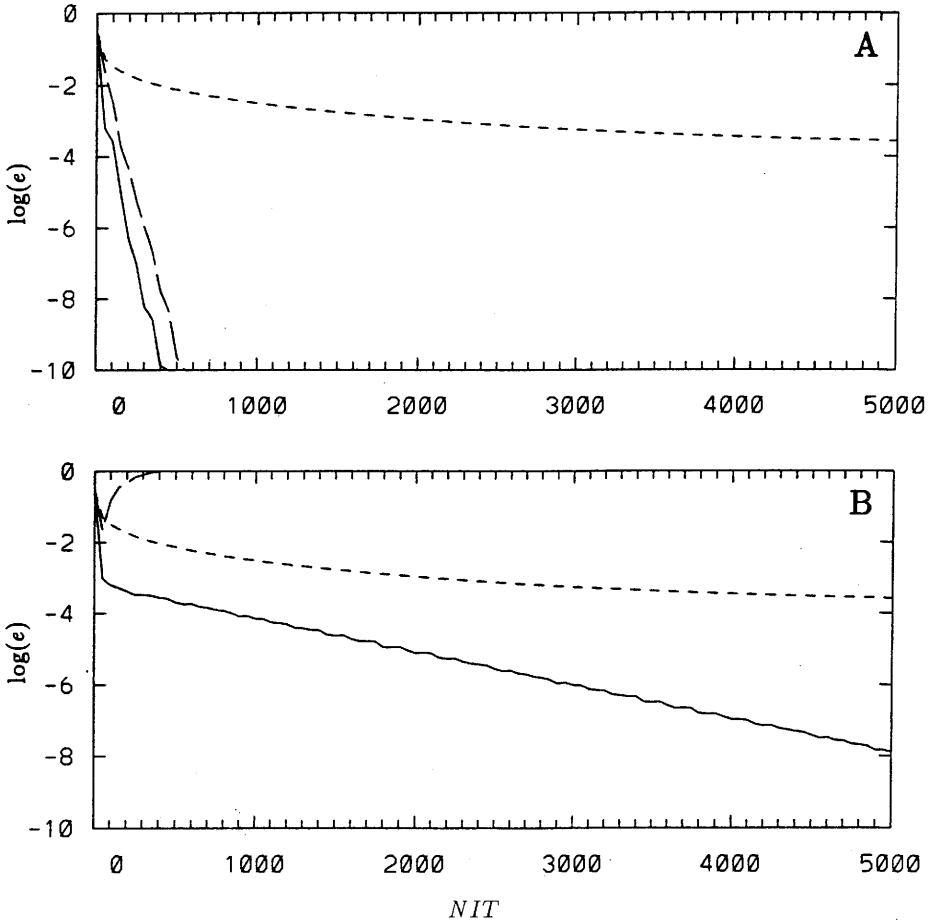


Fig. 2. Convergence rates of different iterative schemes for the potential flow test problem. The logarithm of the error measure $e = \max \left| \frac{1}{\rho^*} \nabla \cdot (\rho^* \mathbf{v}) \right| \Delta t$ is displayed in function of a number of iterations. Panels A and B refer to the solutions on the A- and B-grid, respectively. Solid lines are for CR2 algorithm, long-dashed lines for the classic CG scheme, and short-dashed lines for the Jacobi method.

an elliptic problem is proportional to the number of iterations, the choice of a stopping criteria has a direct impact on the overall efficiency of the scheme. Unfortunately, there is no strict rule when to stop the iteration process (cf. Roache, 1972, Section III-D; for discussion). In principle, one could continue to iterate until one achieves convergence to the machine precision; however, this is highly impractical, as illustrated by the examples above. The maximal residual error $e \approx 10^{-4}$ is already small and further diminishment would have negligible impact on the overall accuracy of the solutions. Thus, relating the stopping criteria to a physically meaningful accuracy requirement (here, $e \sim 10^{-4}C$) is preferable to a specified reduction of the initial residual error.

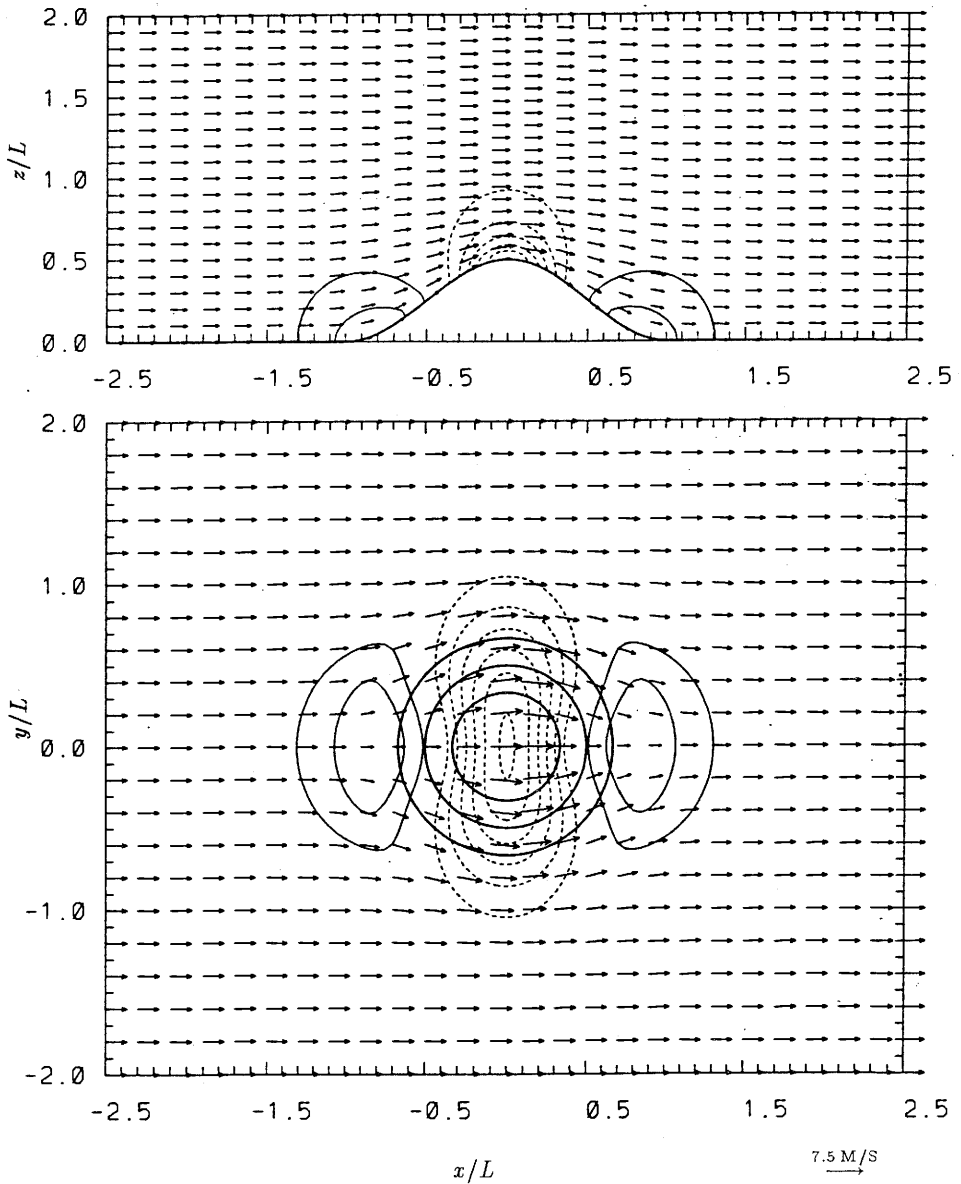


Fig. 3. As in Fig. 1 but for the solution on the B grid.

One conclusion of the discussion so far is that it is not the asymptotic convergence rate but the overall convergence properties that are of primary interest for applications. We now focus on the details of the convergence rates of different schemes during the first 100 iterations, where CR schemes produced overall accurate results. Fig. 6 shows a magnified portion of Fig. 2. It is evident in the figure that even after 5 iterations, CR schemes reduce the initial residual error to a value obtained only after

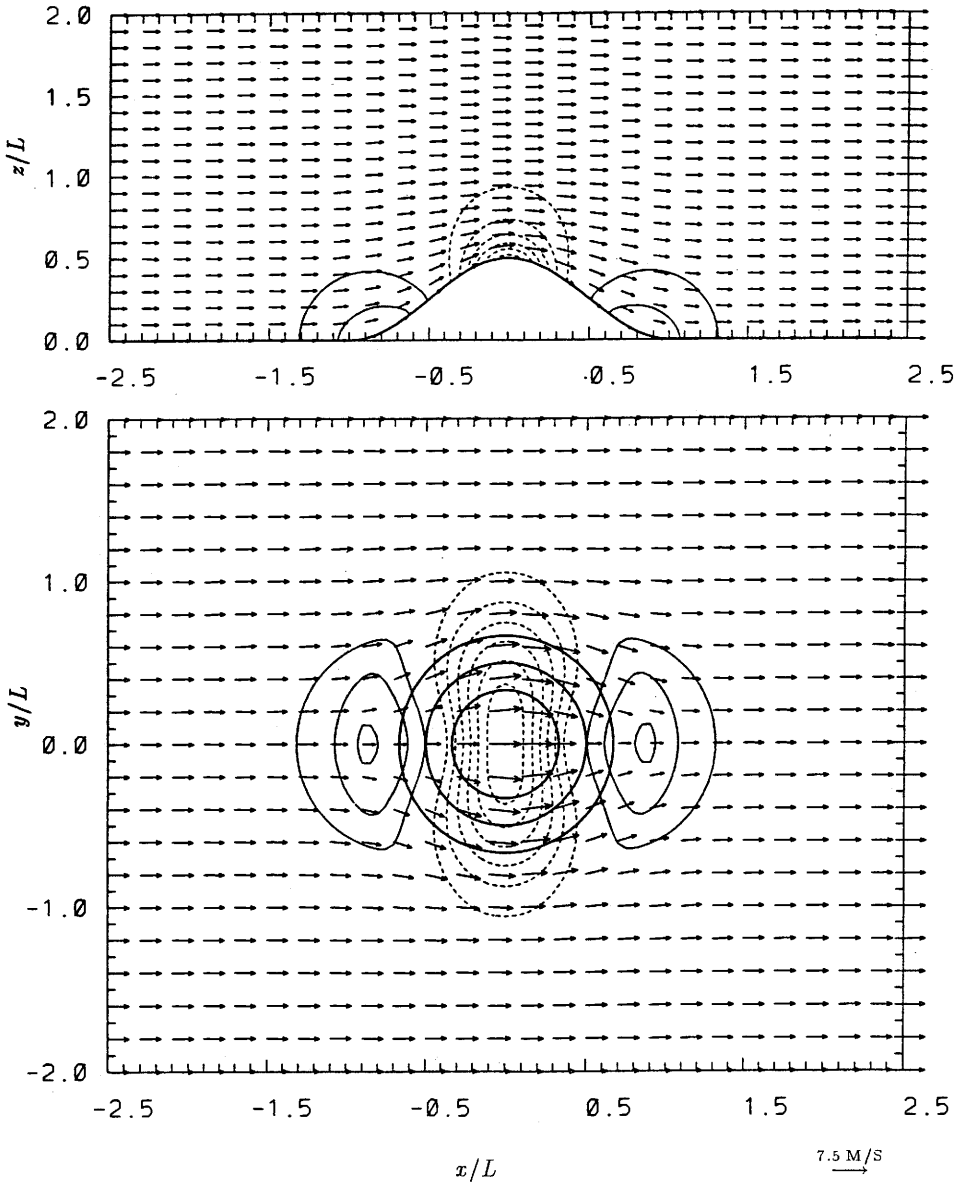


Fig. 4. As in Fig. 1 but after 100 iterations.

100 iterations with the Jacobi method. After 50 iterations they achieve an accuracy that requires ~ 2000 Jacobi iterations (cf. Fig. 2). Since the major computational expense in CR schemes is still associated with evaluating \mathcal{L} on the grid and not with the evaluations of the sums in (17) or (20), these comparisons also provide a rough estimate of the comparative overall efficiency of the two methods. Figure 6 demonstrates that the classic CG algorithm is unreliable for the class of applications

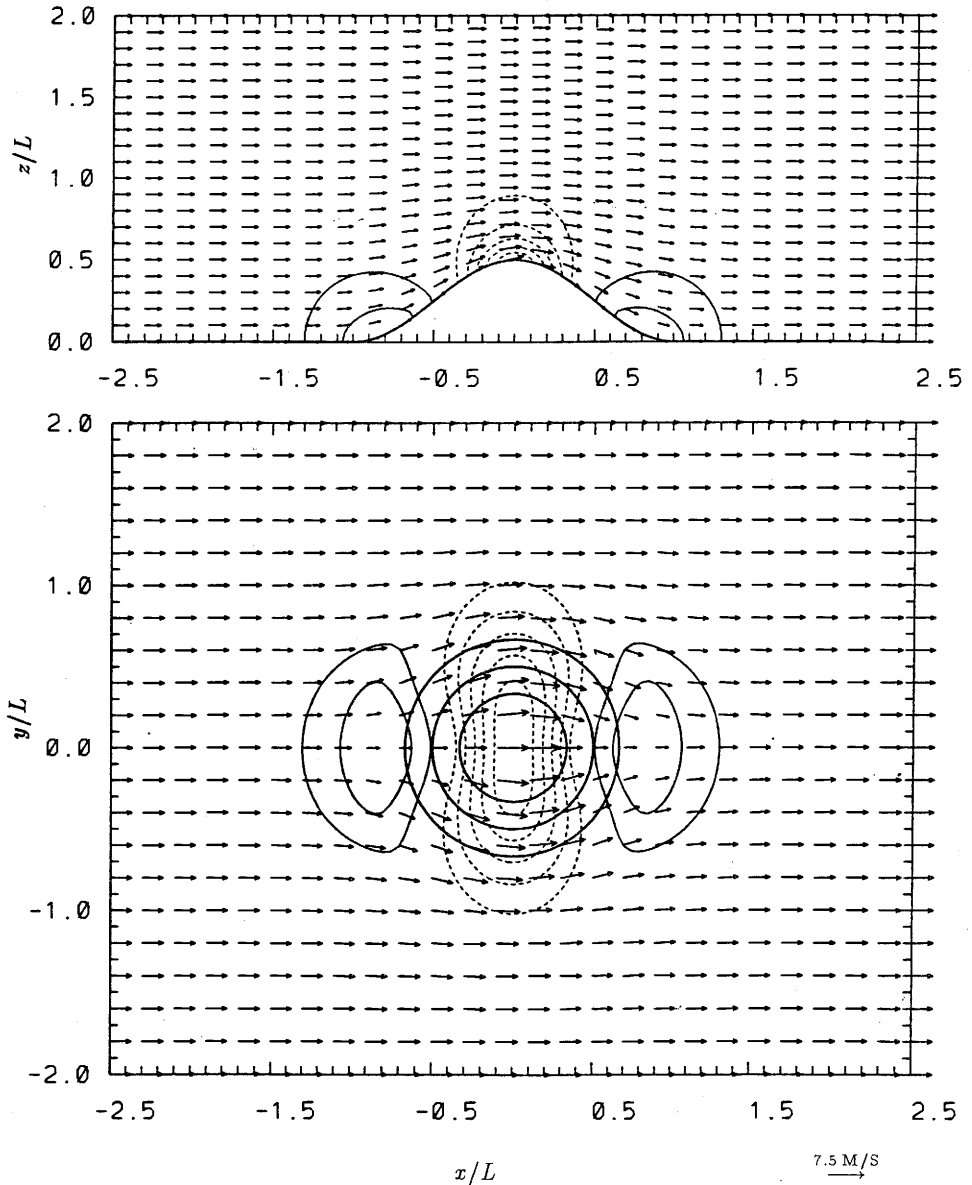


Fig. 5. As in Fig. 1 but for the solution on the B grid after 100 iterations (cf. Fig. 3).

addressed. As the symmetry of \mathcal{L} is not ensured⁵, there is no guarantee that the latter scheme would converge. This is evident on the B-grid. On the A-grid, the CG scheme does converge for this particular test problem but even so, its overall performance is worse than that of the CR schemes. In contrast to CR schemes, the

⁵ We have explicitly verified the negative-definiteness of \mathcal{L} , which is also illustrated by the convergence of the Jacobi scheme.

lack of symmetry of the discrete operator destroys the mathematical equivalence of alternate versions of the classic CG algorithm. In our test problem, the scheme based on the three-term recurrence formula (Eqs. (5), (6) and (9)) does converge on the B-grid but, in turn, does not converge on the A-grid (not shown). Again, even when it converges, its overall performance is worse than that of the CR schemes⁶.

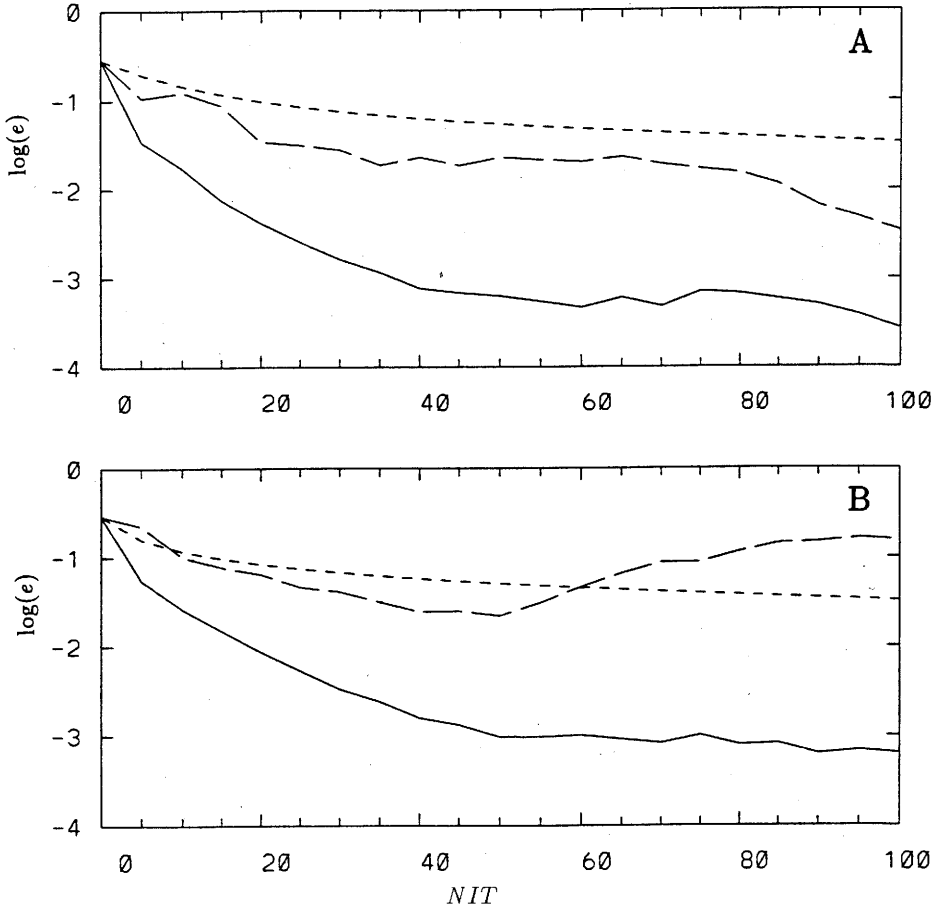


Fig. 6. Magnified fragments of Fig. 2a and 2b.

Insofar as the absolute measures of computational efficiency are concerned, the actual cost of the CR2 scheme per iteration and per grid point is $3.4 \cdot 10^{-7}$ s on a single processor CRAY YMP for the A-grid and twice that on the B-grid. The latter is due to the B-grid requiring about twice as many arithmetic operations for evaluating the \mathcal{L} operator as does the A-grid. These numbers are representative of a complete nonhydrostatic anelastic code, whose start-up procedure has been employed here for test purposes, with more elaborate pressure forces than those in (27) and with

⁶ However, when \mathcal{L} is both symmetric and definite, CG converges faster to the exact solution than CR (cf. Chandra, 1978, Section 4.5).

the operator \mathcal{L} computed from (1b) and (1c) at every iteration. Thus the numbers quoted are conservative as they take no advantage from preprocessing nor from storing the coefficients of the sparse matrix (with up to 27 diagonals) that represents the elliptic operator on the grid. Given well-posed initial conditions that satisfy the mass continuity equation (22b) and the imposed boundary conditions, typical applications with the anelastic model require (in our experience) a few tens of iterations per time-step in order to achieve adequate overall accuracy. This already makes CR schemes effective enough to be useful in a variety of complex atmospheric applications. A further gain in computational efficiency is derived from the generality of the solver that integrates (1) directly in its generic form. The latter admits attractive formulations of entire dynamical models that are both simple and efficient.

5. Concluding Remarks

A majority of numerical models used for the simulation of atmospheric dynamics solve initial value problems. In principle, if one were provided with unlimited computational resources such models could be designed without any reference to solving elliptic equations characteristic of boundary value problems. However, the solutions to these initial value problems contain high-speed waves, which are usually of only minor relevance to the physics being modeled, but whose explicit treatment imposes severe restrictions on the computational timestep. As a practical matter, the latter often makes a straightforward explicit integration of the equations prohibitively expensive. A common alternative is to use time-implicit methods and/or filtered formulations of the governing equations that are free of the high-speed waves. These, however, lead ultimately to elliptic equations that also require efficient integration schemes.

The complexity of the elliptic problems that arise in atmospheric applications can be a serious obstacle to meteorological modeling. The availability of a general, simple, yet efficient solution technique would represent an important advance that could significantly influence current trends in model design. Unfortunately these desirable properties are found to conflict with each other: efficient and general approaches such as multigrid require substantial programming effort, whereas simple general methods such as Jacobi iteration are extremely inefficient.

This paper is the result of our search for a general method that offers a reasonable compromise between efficiency and simplicity. While exploring various alternatives, we found that a particular class of conjugate gradient methods, the conjugate residual (CR) schemes, appears attractive for atmospheric applications. Although theoretically less efficient than multigrid methods, the CR algorithms lead to accurate and efficient computations when incorporated into a variety of dynamical models.

The strength of the CR schemes lies in their robustness, combined with extreme simplicity, flexibility and reasonable efficiency. CR schemes do not require any explicit knowledge of the matrix resulting from the discretization of the elliptic operator. Their coding is straightforward, and does not require any experience with elliptic problems. Moreover, their computational efficiency can be further improved by preconditioning, or even by incorporation of multigrid ideas.

The conjugate gradient methods have a long tradition in the mathematical literature. There they are usually discussed in the abstract terms of Krylov subspace

methods. In this paper, we have presented the CG approach from a somewhat unconventional perspective — we have derived the CR schemes from the physical analogy to the damped wave equation. Although such an interpretation may not capture all of the subtle properties of the advocated algorithms, it has certain advantages. First, it is simple. Second, it offers additional insights into the entire class of methods. Third, it exposes new degrees of freedom and the potential for further development. In particular, it has led us to an alternate proof that ensures convergence of CR schemes for negative-definite but not necessarily symmetric operators \mathcal{L} in (1). The latter is important for many atmospheric applications.

Acknowledgements

The authors gratefully acknowledge discussions with J. Adams, R. Bernard, J. Dukowicz, H. Kapitza, I. Orlanski, and R. Smith. Special thanks go to J. Dukowicz for his in depth personal review of the manuscript and constructive comments. This work has been supported in part by the Department of Energy “Computer Hardware, Advanced Mathematics, Model Physics” (CHAMMP) research program.

Appendix A

Variational solvers: Further details

The derivation of (11a) from (10) and the orthogonality relationships $\langle r^{n+1}r^n \rangle = \langle p^{n+1}\mathcal{L}(p^n) \rangle = 0$, starts with the formation of the inner product of (10b) with r^n :

$$\langle r^{n+1}r^n \rangle = 0 = \langle r^n r^n \rangle + \beta^n \langle r^n \mathcal{L}(p^n) \rangle \quad (\text{A1})$$

Employing the symmetry of \mathcal{L} , (A1) may be rewritten as

$$\langle r^n r^n \rangle + \beta^n \langle p^n \mathcal{L}(r^n) \rangle = 0 \quad (\text{A2})$$

Acting with \mathcal{L} on both sides of (10c) taken at the n -th iteration results in

$$\mathcal{L}(p^n) = \alpha^n \mathcal{L}(p^{n-1}) + \mathcal{L}(r^n) \quad (\text{A3})$$

Substituting from (A3) for $\mathcal{L}(r^n)$ in (A2) gives

$$\langle r^n r^n \rangle + \beta^n \langle p^n \mathcal{L}(p^n) \rangle - \beta^n \alpha^n \langle p^n \mathcal{L}(p^{n-1}) \rangle = 0 \quad (\text{A4})$$

Since $\langle p^n \mathcal{L}(p^{n-1}) \rangle = 0$ from the preceding iteration, (A4) implies β^n in (11a). Note that from the negative definiteness of \mathcal{L} , we must have $\beta^n \geq 0$ with $\beta^n = 0$ implying $\langle r^n r^n \rangle = 0$, *viz.*, the exact solution at the n -th iteration.

The derivation of (11b) starts with the formation of the inner product of (10c) with $\mathcal{L}(p^n)$:

$$\langle p^{n+1}\mathcal{L}(p^n) \rangle = 0 = \alpha^{n+1} \langle p^n \mathcal{L}(p^n) \rangle + \langle r^{n+1}\mathcal{L}(p^n) \rangle \quad (\text{A5})$$

Substituting for $\mathcal{L}(p^n)$ from (10b) in the second term on the rhs of (A5) results in

$$0 = \alpha^{n+1} \langle p^n \mathcal{L}(p^n) \rangle + \frac{1}{\beta^n} \langle r^{n+1}r^{n+1} \rangle - \frac{1}{\beta^n} \langle r^{n+1}r^n \rangle \quad (\text{A6})$$

The last term on rhs of (A6) vanishes by the orthogonality of the residuals, and also $\langle p^n \mathcal{L}(p^n) \rangle = -\langle r^n r^n \rangle / \beta^n$ from (11a). Thus (A6) becomes

$$0 = -\alpha^{n+1} \langle r^n r^n \rangle + \langle r^{n+1} r^{n+1} \rangle \tag{A7}$$

which implies (11b).

Appendix B

Conjugate residual schemes: Further details

An outline of the convergence proof of the algorithm in (12), (16) and (17) starts with forming the inner product of (12b) with r^{n+1}

$$\langle r^{n+1} r^{n+1} \rangle = \gamma \langle r^{n+1} (r^n - r^{n-1}) \rangle + \langle r^{n+1} r^{n-1} \rangle + \beta \langle r^{n+1} \mathcal{L}(r^n) \rangle \tag{B1}$$

Since the first and the third term on the rhs of (B1) vanish from the orthogonality relationships in (14), (B1) reduces to

$$\langle r^{n+1} r^{n+1} \rangle = \langle r^{n+1} r^{n-1} \rangle \tag{B2}$$

From the first of the relations in (14),

$$\langle r^{n+1} r^{n-1} \rangle = \langle r^{n+1} r^n \rangle \tag{B3}$$

After substituting for the rhs of (B2) we derive

$$\langle r^{n+1} r^{n+1} \rangle = \langle r^{n+1} r^n \rangle \tag{B4}$$

which implies another useful relationship

$$\langle r^{n+1} (r^{n+1} - r^n) \rangle = 0 \tag{B5}$$

Substituting (12b) for r^{n+1} on the rhs of (B4) leads to

$$\langle r^{n+1} r^{n+1} \rangle = \gamma \langle r^n (r^n - r^{n-1}) \rangle + \langle r^n r^{n-1} \rangle + \beta \langle r^n \mathcal{L}(r^n) \rangle \tag{B6}$$

Invoking (B5) and (B4) at the preceding n -th iteration reduces (B6) to

$$\langle r^{n+1} r^{n+1} \rangle = \langle r^n r^n \rangle + \beta \langle r^n \mathcal{L}(r^n) \rangle \tag{B7}$$

The monotone convergence of the algorithm in (12), (16) and (17) requires $\langle r^{n+1} r^{n+1} \rangle \leq \langle r^n r^n \rangle$ with the equality holding if and only if $\langle r^n r^n \rangle = 0$. Since \mathcal{L} has been assumed negative definite, the latter requires $\beta \geq 0$ in (B7) with the equality holding only if $\langle r^n r^n \rangle = 0$. In order to show this, consider the following arguments.

Exploiting the orthogonality relationships discussed above, it is easy to show that the coefficients in (17a) and (17c) satisfy

$$AC = \langle r^n r^n \rangle - \langle r^{n-1} r^{n-1} \rangle = -AA \tag{B8}$$

whereupon the numerator in (16b) satisfies

$$AB \cdot AC - AA \cdot BC = AC(AB + BC) = \left(\langle r^n r^n \rangle - \langle r^{n-1} r^{n-1} \rangle \right) \langle r^n \mathcal{L}(r^n) \rangle \tag{B9}$$

From (B8) and (17a), $\langle r^n r^n \rangle - \langle r^{n-1} r^{n-1} \rangle \leq 0$, which together with the negative definiteness of \mathcal{L} results in a nonnegative numerator in (16b). Now the denominator in (16) is always nonnegative by the Cauchy-Schwarz inequality; this proves that β in (16b) is nonnegative for all n . The special case of $\beta = 0$ implies either $\langle r^n \mathcal{L}(r^n) \rangle = 0$ (which by the negative definiteness of \mathcal{L} implies the exact solution) or $\langle r^n r^n \rangle = \langle r^{n-1} r^{n-1} \rangle$. By induction and (B7), the latter implies $\beta = 0$ for all preceding iterations up to $\beta^0 = -\frac{\langle r^0 \mathcal{L}(r^0) \rangle}{\langle \mathcal{L}(r^0) \mathcal{L}(r^0) \rangle}$ (see discussion following (17e)). This proves that $\langle r^0 r^0 \rangle = 0$ and thus $\langle r^n r^n \rangle = 0$ for all n .

Note that the relationship (B8) may be exploited to eliminate one of the five sums in (17). Another sum may be eliminated, for instance, by relating AA to the coefficients' values at a preceding iteration: Rewriting (12b) as $r^{n+1} - r^n = (\gamma^n - 1)(r^n - r^{n-1}) + \beta^n \mathcal{L}(r^n)$, forming the inner product $\langle (r^{n+1} - r^n)^2 \rangle$, and employing the definitions in (17) leads to $AA^{n+1} = (\gamma^n - 1)^2 AA^n + 2(\gamma^n - 1)\beta^n AB^n + (\beta^n)^2 BB^n$, where the superscripts on the coefficients refer to their values at appropriate iterations.

References

- Adams J.C. (1989): *MUDPACK: Multigrid portable FORTRAN software for efficient solution of linear elliptic partial differential equations*. — Appl. Math. Comput., v.34, pp.113–146.
- Adams J.C. (1991): *Recent enhancements in MUDPACK, a multigrid software package for elliptic partial differential equations*. — Appl. Math. Comput., v.43, pp.79–93.
- Adams J.C. (1993): *MUDPACK-2: Multigrid software for approximating elliptic partial differential equations on uniform grids with any resolution*. — Appl. Math. Comput., v.53, pp.235–249.
- Adams J.C., Garcia R., Gross B., Hack J., Haidvogel D. and Pizzo V. (1992): *Applications of multigrid software in atmospheric sciences*. — Mon. Wea. Rev., v.120, pp.1447–1458.
- Ashby S.F., Manteuffel T.A. and Saylor P.E. (1990): *A taxonomy for conjugate gradient methods*. — SIAM J. Numer. Anal., v.27, pp.1542–1568.
- Bernard R.S. and Kapitza H. (1992): *How to discretize the pressure gradient for curvilinear MAC grids*. — J. Comput. Phys., v.99, pp.288–298.
- Birkhoff G. and Lynch R.E. (1984): *Numerical solutions of elliptic problems*. — SIAM, Philadelphia, pp.3.
- Chandra R. (1978): *Conjugate gradient methods for partial differential equations*. — Ph.D. Thesis, Dept. Computer Science, Yale Univ., New Haven, CT, pp.258.
- Chorin A.J. (1968): *Numerical solution of the Navier-Stokes equations*. — Math. Comp., v.22, pp.742–762.
- Clark T.L. (1977): *A small-scale dynamic model using a terrain-following coordinate transformation*. — J. Comput. Phys., v.24, pp.186–214.
- Durran D.R. (1989): *Improving the anelastic approximation*. — J. Atmos. Sci., v.46, pp.1453–1461.

- Eisenstat S.C., Elman H.C. and Schultz M.H. (1983): *Variational iterative methods for nonsymmetric systems of linear equations*. — SIAM J. Numer. Anal., v.20, pp.345–357.
- Gal-Chen T. and Somerville R.C.J. (1975): *On the use of a coordinate transformation for the solutions of the Navier-Stokes equations*. — J. Comput. Phys., v.17, pp.209–228.
- Glowinski R. (1992): *Ensuring well-posedness by analogy; Stokes problem and boundary control for the wave equation*. — J. Comput. Phys., v.103, pp.189–221.
- Hestenes M.R. and Stiefel E. (1952): *Methods of conjugate gradients for solving linear systems*. — J. Res. NBS, v.49, pp.409–436.
- Kapitza H. (1988): *Truncated incomplete factorization for conjugate-gradient methods in two and three dimensions*. — Appl. Math. Comp., v.28, pp.73–87.
- Kapitza H. and Eppel D. (1985): *A 3-D Poisson solver based on a conjugate gradient algorithm*. — GKSS-Forschungszentrum Geesthacht GmbH. Report GKSS 85/E/23, pp.34.
- Kapitza H. and Eppel D. (1987): *A 3-D Poisson solver based on a conjugate gradients compared to standard iterative methods and its performance on vector computers*. — J. Comput. Phys., v.68, pp.474–484.
- Kapitza H. and Eppel D. (1992): *The non-hydrostatic mesoscale model GESIMA. Part 1: Dynamical equations and tests*. — Beitr. Phys. Atmosph., v.65, pp.129–146.
- Potter D. (1973): *Computational Physics*. — John Wiley & Sons, London, pp.279.
- Ramamurthy M.K. and Navon I.M. (1992): *The conjugate-gradient variational analysis and initialization method: An application to MONEX SOP 2 data*. — Mon. Wea. Rev., v.120, pp.2360–2377.
- Roache P.J. (1972): *Computational Fluid Dynamics*. — Hermosa Publishers, Albuquerque, pp.446.
- Saad Y. and Schultz M.H. (1986): *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. — SIAM J. Sci. Stat. Comput., v.7, pp.856–869.
- Skamarock W.C. and Klemp J.B. (1992): *The stability of time-split numerical methods for the hydrostatic and nonhydrostatic elastic equations*. — Mon. Wea. Rev., v.120, pp.2109–2127.
- Smith R.D., Dukowicz J.K. and Malone R.C. (1992): *Parallel ocean general circulation modeling*. — Physica D, v.60, pp.38–61.
- Stoer J. (1983): *Solution of large linear systems of equations by conjugate gradient type methods*. — Springer-Verlag: Mathematical Programming: The State of the Art, Bachem A., Grottschel M. and Korte B. (Eds.), pp.655.
- Swarztrauber P. and Sweet R. (1975): *Efficient FORTRAN subprograms for the solution of elliptic equations*. — NCAR TN/IA-109, July, pp.138.

Received: February 23, 1994