JOURNAL OF COMPUTATIONAL PHYSICS 51, 423–443 (1983)

The Multigrid Method for Accelerated Solution of the Discretized Schrödinger Equation

F. F. GRINSTEIN, H. RABITZ, AND A. ASKAR*

Department of Chemistry, Princeton University
Princeton, New Jersey 08544

Received September 16, 1982

The use of the new multigrid method for the numerical solution of the discretized Schrödinger equation is discussed. A simple approach is chosen to circumvent the difficulties associated with the fact that the Schrödinger differential operator is not positive definite. The scheme is applied to a two-dimensional model problem which includes the essential features present in actual scattering problems. The results show that significant savings can be achieved both in computational work and core memory requirements. The multigrid method should greatly increase the utility of meshing procedures for solving the partial differential equations of quantum mechanics.

1. INTRODUCTION

The theoretical treatment of molecular collisions necessitates solving Schrödinger's equation, either in a differential or integral form. In either case the most popular present procedure for its solution is the close coupling method [1]. In this approach, the wave function is expanded in a basis set over all the collision coordinates except a relative translational variable. As well as being computationally expensive, there are additional difficulties with this approach for reactive, and especially for dissociative collisions. With these factors in mind, direct treatment of the Schrödinger partial differential equation (PDE) exhibits promising aspects [3, 17]. The multigrid (MG) method [4–7, 15] considered in this paper, is proving to be very successful for solving similar PDE problems in various engineering disciplines. We will demonstrate below that the procedure can be efficiently applied also to quantum mechanical problems.

A wide class of methods for the solution of partial differential equations is based on discretization of the field variables, such as provided by the finite difference or the finite element methods [2]. A major practical difficulty is related to the large size of the resulting algebraic equations. These systems are banded and sparse (i.e., only a small number of nonzero elements exist in the band). Hence, significant savings can be achieved by accounting for this property even within a traditional direct solver such as Gauss elimination. Indeed, the computational effort involved using a full

* Permanent address: Mathematics Department, Bogazici University, Istanbul, Turkey.

0021-9991/83 $3.00

$N \times N$ matrix is $O(4N^3/3)$ additions, while in a scheme, making explicit use of the bandedness of the matrix, the operations count is $O(4B^2N)$, where B is the half-width of the band.[1] If d is the dimensionality of the space, and n the average number of discrete values for each coordinate, typically one has $N = O(n^d)$ and $B = O(N^{1-1/d})$. Hence the computational effort for the direct solution of the algebraic equations resulting from the discretization of the PDE is $O(n^{3d-2})$. It is seen that even with a modest number such as $n = 10$, the computational effort in 1-, 2-, 3-, and 4-dimensional spaces, are in the ratios $1 : 10^3 : 10^6 : 10^9$. Clearly, the increase of the computational work with dimensionality by direct methods is quite dramatic. This makes problems with more than three coordinates prohibitive, even with the most sophisticated computers and these conventional numerical methods.

In classical field theories we have $d \leqslant 3$, and in this case direct solutions are feasible, although leading to large size problems. It is clear, however, that for quantum mechanical problems in higher dimensional spaces direct solutions through discretization of the Schrödinger equation are presently impossible. This consideration is at the source of the close-coupling approaches [1] in scattering.

The situation described above pertains only to the direct solution of the algebraic equations. Indeed, even in two and three dimensions, for large problems involving several thousand unknowns, iterative methods are usually preferred [13]. In order to guarantee convergence of the standard methods, a basic requirement is for the matrix of the linear algebraic system to be positive definite. For the Schrödinger operator $H - E$, this condition is not fulfilled, because it admits both oscillatory and nonoscillatory-type solutions (i.e., negative and positive eigenvalues, respectively). This consideration is true both for the finite difference and finite element discretizations, and precludes the straightforward use of the ordinary iterative schemes (e.g., Gauss–Seidel and its variations) for the solution of Schrödinger's equation. Alternatively, the resulting sequence of successive iterates could be transformed into a convergent sequence [9, 11], or more elaborated iteration schemes could be considered in this case (see Section II). The price to be paid, however, is a slow rate of convergence and rather high computational work, although presumably still lower than that involved with direct methods. The remarkable aspect of the multigrid technique is that the comparable operations count for the solution of the algebraic equations resulting from the discretization of PDEs is $O(N)$ (typically, $40N$ for Poisson problems [4]) rather than $O(N^{3-2/d})$ by conventional methods. Thus, even for problems involving only a few hundred unknowns, the MG method already appears as more efficient than any other conventional approach. The technique offers computational efficiency and memory savings with possible dramatic consequences for the solution of the quantum mechanical and other chemical problems. Indeed, comparing with our previous example, the computational efforts in one, two, three, and four dimensions are now in the ratios $1 : 10 : 10^2 : 10^3$, considering again the number of discrete points involved as $N = 10^d$.

In Section II a stability analysis is presented for the Gauss–Seidel iteration scheme,

[1] Additions are taken as units for operations count, see Section V.

applied to the finite difference discretization of the two-dimensional Schrödinger equation. The purpose is to illustrate the convergence or divergence of the iteration from the point of view of local Fourier analysis, and to provide the basis for understanding the MG scheme as applied to Schrödinger's equation. Sections III and IV present a general discussion of the MG method, and the particular form used in our calculations. Results for an illustrative case corresponding to a two-dimensional model problem are reported in Section V. This case was specifically chosen in order to best demonstrate the main features and capabilities of the MG method as applied to Schrödinger's equation. With the numerical tools outlined below and their exhibited success, the way is now open for applications to more sophisticated problems. The final conclusions are given in Section VI.

II. Stability Analysis

The basic limitations of traditional iterative schemes, discussed above, can be visualized in a useful way by considering a local Fourier analysis of the error of the resulting iterates. To be specific, consider the case of the two-dimensional Schrödinger equation

$$\left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + K^2\right] u(x, y) = 0, \tag{2.1}$$

where $K^2 = K^2(x, y) = k^2 - V(x, y)$ with k^2 being the the energy and $V(x, y)$ being the potential. K^2 is assumed to be constant in the local stability analysis that follows. Using a five-point formula for the Laplacian operator, we can express (2.1) in the following way:

$$u_{j+1,m} + u_{j,m+1} + u_{j-1,m} + u_{j,m-1} + (K^2h^2 - 4)\, u_{j,m} = 0, \tag{2.2}$$

where h is the constant spacing of the grid G^h, $u_{j,m} = u(x_j, y_m)$, and $u_{j\pm1,m\pm1} = u(x_j \pm h, y_m \pm h)$. Rewriting Eq. (2.2) with $D = K^2h^2 - 4$ leads to

$$u_{j,m} = -[u_{j+1,m} + u_{j,m+1} + u_{j-1,m} + u_{j,m-1}]/D. \tag{2.3}$$

From (2.3) we may also obtain the basic Gauss–Seidel iterative equation ($n = 1, 2, \ldots$)

$$u_{j,m}^{(n)} = -[u_{j+1,m}^{(n-1)} + u_{j,m+1}^{(n-1)} + u_{j-1,m}^{(n)} + u_{j,m-1}^{(n)}]/D. \tag{2.4}$$

The standard stability analysis for the iteration is based on considering the error $v_{jm}^{(n)} = u_{jm}^{(n)} - u_{jm}$ in the nth iteration, and following the growth or decay of its Fourier components from step n to step $n + 1$. It is readily seen that $v_{jm}^{(n)}$ satisfies the same equation as (2.4), as a consequence of linearity

$$v_{j,m}^{(n)} = -[v_{j+1,m}^{(n-1)} + v_{j,m+1}^{(n-1)} + v_{j-1,m}^{(n)} + v_{j,m-1}^{(n)}]/D. \tag{2.5}$$

In order to perform a local Fourier analysis of the error in the scheme, we set [4]

$$v_{j,m}^{(n)} = A_{\mathbf{q}}^{(n)} \exp[i\mathbf{q} \cdot \mathbf{r}_{j,m}]. \tag{2.6}$$

Here, $\mathbf{q}$ and $A_{\mathbf{q}}^{(n)}$ are, respectively, the wave-vector and the amplitude of the particular Fourier component in the error, and $\mathbf{r}_{j,m}$ the position vector at G^h. Significant simplification in the analysis is achieved by choosing an equally spaced Cartesian mesh, where we have

$$\mathbf{r}_{j,m} = (hj, hm).$$

Hence, the error in the wavefunction is written as

$$v_{j,m}^{(n)} = A_{\boldsymbol{\theta}}^{(n)} \exp[i\theta_1 j + i\theta_2 m],$$

where

$$\theta_1 = q_1 h, \qquad \theta_2 = q_2 h, \qquad \boldsymbol{\theta} = (\theta_1, \theta_2).$$

The ratio of the amplitudes $|A_{\boldsymbol{\theta}}^{(n+1)}/A_{\boldsymbol{\theta}}^{(n)}|$ being less than or greater than one, indicates, respectively, stability or instability of the iteration. Having this in mind, we substitute (2.6) into (2.5) to obtain the relevant quotient

$$\mu(\theta_1, \theta_2) = |A_{\boldsymbol{\theta}}^{(n+1)}/A_{\boldsymbol{\theta}}^{(n)}|$$

$$\mu(\theta_1, \theta_2) = |[\exp(i\theta_1) + \exp(i\theta_2)]/[D + \exp(-i\theta_1) + \exp(-i\theta_2)]|. \tag{2.7}$$

The next step in the analysis, is to obtain upper bounds for $\mu(\theta_1, \theta_2)$ in two regimes: high frequencies, where $\pi/2 \leqslant |\theta_i| \leqslant \pi$, and low frequencies, where $0 \leqslant |\theta_i| < \pi/2$. Before doing so, the range of values considered for h (and hence for D) have to be restricted by considerations of physical resolution or numerical accuracy for the representation of $u(x, y)$ through discretization. The parameter K^2 is the wave-vector squared, and on the nondimensional units used here we should have $|K^2|\, h^2 < 1$, for a reasonable numerical resolution of the wave.

The high frequency error components, those visible on G^h but not on G^{2h}, have a smoothing factor η given by

$$\eta = \max_{\pi/2 \leqslant |\theta_i| \leqslant \pi} \mu(\theta_1, \theta_2)$$

$$= \mu\left(\frac{\pi}{2}, \frac{\pi}{2}\right) = \left[1 + \left(\frac{D}{2}\right)^2\right]^{-1/2} < 1.$$

Hence, they are always filtered after a few relaxation steps, independently of the sign and value of K^2 (recall that $D = K^2 h^2 - 4$).

For the low frequency components, however, we shall find that the errors will be attenuated or amplified, depending on the sign of K^2. Indeed, Eq. (2.7) now leads to

$$\eta' = \max_{0 \leqslant |\theta_j| < \pi/2} \mu(\theta_1, \theta_2) = \mu(0, 0) = \left| 1 + \frac{D}{2} \right|^{-1}, \tag{2.9}$$

from which we obtain,

$$\eta' > 1, \qquad -4 < D < -3 \qquad (K^2 > 0,\ K^2h^2 < 1), \tag{2.10a}$$

$$\eta' < 1, \qquad D < -4 \qquad (K^2 < 0). \tag{2.10b}$$

Thus, inequality (2.10a), which pertains to the range of oscillating solutions, indicates that the low frequency components of the error are amplified, and consequently, the iterative process diverges. Similarly, the second inequality (2.10b) predicts that for the case of nonoscillatory solutions (tunneling) those components are damped through the relaxations, although at a much slower rate than the high frequency ones. Hence, the sheme converges in this case as well.

In summary, the preceding analysis clearly indicates that the low frequencies are responsible for both the eventual divergence or the slow convergence of the iterations, and that relaxation sweeps are very efficient for smoothing errors, i.e., for eliminating their high frequency components. More generally, these conclusions are actually valid for other Gauss–Seidel-type iteration schemes, the relevant feature being the local nature of the relaxation process. The possibly troublesome low frequency errors will be dealt with by a special procedure in the MG scheme below.

III. The Multigrid Method

The MG method will be discussed here in connection with the numerical solution of PDEs, although it may also be applied to integral equations. In the conventional methods, the boundary value problem is first discretized in some way (e.g., finite elements or finite differences on a grid) and the resulting system of algebraic equations is then submitted to a numerical solver. In the MG method, however, discretization and solution processes are intermixed with and greatly benefit from each other [6]. Attention is concentrated on a cooperative solution process involving a sequence of uniform grids $\{G^1 \subset G^2 \subset \cdots \subset G^M\}$ with regularly varying mesh spacings, usually decresing in a gemetrical progression. Moreover, relaxation sweeps are performed over each grid, and coarse-to-fine grid and fine-to-coarse grid interpolation schemes (discussed below) allow for transfers from one grid to the other. It is critical to distinguish this approach from conventional relaxation in which we solve successively on finer grids, followed by comparison of the results for convergence behavior.

The MG method has been presented in several reviews. Nevertheless, we present the main ideas in this paper for completeness, since most readers in the area of quantum mechanics are likely not familiar with the method.

A. *Two Level Analysis of the MG Method*

Having in mind the application of the method to Schrödinger's equation, suppose the PDE of interest is

$$Lu = 0, \tag{3.1}$$

where L is a linear differential operator, and given boundary conditions (Dirichlet, Neumann, or mixed type) are specified. We attempt the solution of the discretized version of Eq. (3.1) at grid level $k+1$ ($k \geqslant 1$)

$$L^{k+1}u^{k+1} = 0, \tag{3.2}$$

where L^{k+1} involves the grid G^{k+1} having a characteristic mesh spacing h_{k+1} (see Fig. 1). At this point, and in order to avoid notational confusion, we list the various variables we shall be dealing with:

u:exact solution to $Lu = 0$,

u^{k+1}:exact solution to $L^{k+1}u^{k+1} = 0$,

U^{k+1}:approximation to u^{k+1},

V^{k+1}:error of U^{k+1}, i.e., $V^{k+1} = u^{k+1} - U^{k+1}$.

Moreover, we shall also be dealing with the quantities $\hat{u}^k$, $\hat{U}^k$, and $\hat{V}^k$, defined in the same way for the associated inhomogeneous equation $L^k\hat{u}^k = \hat{f}^k$, required at the correction stage discussed below, where the inhomgeneity $\hat{f}^k$ will be seen to be a

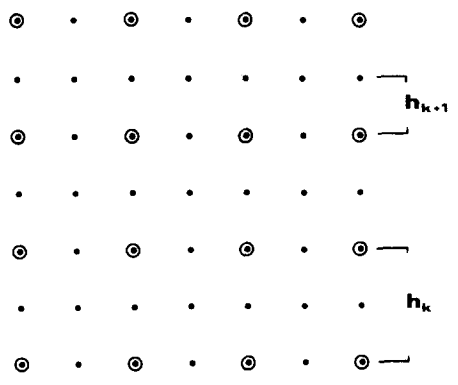

FIG. 1. Diagram of two successive grids of the sequence. Points (●) belong to G^{k+1} while points ⊙ are the common points of G^k and G^{k+1}.

function of U^{k+1}. The interpolation operators allow for the transfer of values from one grid to the immediately coarser or finer grid. The fine-to-coarse interpolation operator from grid G^{k+1} to G^k is denoted by I_{k+1}^k, and involves a weighted average over the points of the finer grid. Usually, and this is the case here, a limiting case of this weighted averaging is used, the so-called injection, by which the transfer involves retaining the same values at the common points of both grids (i.e., $g^k(x) = I_{k+1}^k g^{k+1}(x)$, at common prints x, and for a given function $g(x)$ defined on G^k and G^{k+1}). Conversely, the coarse-to-fine interpolation operation from G^k to G^{k+1} is performed through I_k^{k+1}, and involves generating values at the additional points of the finer grid by a polynomial interpolation.

We now proceed to deriving the basic equation and ideas to be used in the *MG* scheme. After a few relaxation sweeps on Eq. (3.2), we obtain an approximation U^{k+1} to u^{k+1} and, since by definition V^{k+1} is the corresponding error, the following equality holds:

$$L^{k+1}(U^{k+1} + V^{k+1}) = 0, \tag{3.3}$$

where only U^{k+1} is a known quantity at this stage. The problem is now that of reducing V^{k+1} further. According to the discussion in Section II, at this point V^{k+1} is dominated by low frequency components (i.e., those with wavelengths large as compared to the mesh spacing) and reducing V^{k+1} by more relaxation sweeps is a slowly convergent process (or even impossible). Rather than proceeding with iterations, it is much more efficient to make a transfer to the grid G^k on which the basic spacing is the double of that of G^{k+1} (i.e., $h_k = 2h_{k+1}$) where we solve, also by relaxation, for the corrections V^k. In order to obtain the equations for V^k, we apply the interpolation operator I_{k+1}^k on both sides of (3.3) to transfer values to the kth grid. We get

$$I_{k+1}^k(L^{k+1}V^{k+1}) = -I_{k+1}^k(L^{k+1}U^{k+1}), \tag{3.4}$$

where the right-hand side is known, and constitutes an inhomogeneous term for a suitable equation for V^k. On the other hand, $I_{k+1}^k L^{k+1}V^{k+1}$ involves the transfer of $L^{k+1}V^{k+1}$ to G^k, where it can be approximated quite well by L^kV^k, owing to the assumed smoothness of V^k. Hence, we can rewrite Eq. (3.4) as

$$L^kV^k = -I_{k+1}^k L^{k+1}U^{k+1}. \tag{3.5}$$

Equivalently, by adding $L^kI_{k+1}^k U^{k+1}$ on both sides of Eq. (3.4), and changing variables from V^k to

$$\hat{u}^k = I_{k+1}^k U^{k+1} + V^k, \tag{3.6}$$

where $I_{k+1}^k U^{k+1}$ is known, the equation to be solved is

$$L^k\hat{u}^k = \hat{f}^k = L^kI_{k+1}^k U^{k+1} - I_{k+1}^k L^{k+1}U^{k+1}. \tag{3.7}$$

After obtaining a solution $\hat{U}^k$ approximating $\hat{u}^k$ within a prescribed tolerance, we return to the finer grid G^{k+1}. The last approximation obtained for u^{k+1}, namely,

$$U_{\text{old}}^{k+1} = U^{k+1},$$

is then updated, by addition of the calculated correction $V^{k+1} = I_k^{k+1} V^k = I_k^{k+1}(\hat{U}^k - I_{k+1}^k U^{k+1})$. In this way, we get the new approximation U_{new}^{k+1} at grid G^{k+1}

$$U_{\text{new}}^{k+1} = U_{\text{old}}^{k+1} + I_k^{k+1}(\hat{U}^k - I_{k+1}^k U_{\text{old}}^{k+1}), \tag{3.8}$$

where we note that in general $I_k^{k+1} I_{k+1}^k \neq 1$. The iterations on Eq. (3.2) are now reinitiated taking U_{new}^{k+1} as initial guess.

This basic two-level analysis does not necessarily imply that the flow of the cycle initiates at G^{k+1}, or that G^{k+1} necessarily is the finest grid in the sequence. Furthermore, the solution of Eq. (3.7) itself is also approached in this way. The improvement of an approximate solution $\hat{U}^k$ to Eq. (3.7), obtained after a few iterations, is pursued by further reducing $\hat{V}^k$ through the use of increasingly coarser grids, where the solution at G^1, the coarsest, is obtained iteratively or directly depending on the nature of the problem. Since the operations count on the coarse grids is much smaller than that on the finest grid, this overall process leads to a very fast solver. The details of the implementation of these procedures are the subject of Section IV. We shall refer, specifically, to the MG algorithm known as the *full approximation scheme* [6], which involves the use of Eqs. (3.7) and (3.8), or similar ones with suitable modifications, at the correction stages. By recalling Eq. (3.6) it is seen that $\hat{u}^k$ can be seen as the representation on the coarser grid G^k of the sum (on grid G^{k+1}) of the basic approximation U^{k+1} and its correction V^{k+1}. Hence, $\hat{u}^k$ represents on G^k the full current approximation, and this suggests the name for the scheme, as compared to the *correction scheme* [6] (also called Cycle C in [4]) which involves operating with the corrections themselves through Eq. (3.5).

B. *Indefinite Problems*

In order to apply the MG method in the case of Schrödinger's equation, the stability matter discussed in Section II has to be considered. As indicated, the Gauss–Seidel iteration scheme (as well as many other standard relaxation procedures which are variations of it) is divergent for a nonpositive definite operator such as that associated to Schrödinger's equation. Hence, the MG method, which comprises such a relaxation method as one of its steps, is not directly applicable in this case. There are, however, several ways to circumvent this difficulty. First, any nonsingular indefinite operator can be changed into a positive definite one by "squaring" it. To be specific, suppose the problem to be solved is reduced to the algebraic equations

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}.$$

Here, $\mathbf{A}$ is an indefinite matrix representing the indefinite operator discretization. Multiplying the equation by $\mathbf{A}^{\mathrm{T}}$ (or by $\mathbf{A}^{\dagger}$ if the matrix is complex) yields a new

algebraic system for which the matrix $\mathbf{A}^T\mathbf{A}$ involved is guaranteed to be positive definite by construction.[2] In this way, the multigrid method can be applied in its standard form. The price to be paid, however, is a matrix having twice the original bandwith (or equivalently a differential operator of twice the order) and thus, slow rates of convergence can be expected. A second possibility is to introduce as variables the components of the gradient of the wave function ψ along with the function ψ itself [12]. This makes the operator definite and has the additional advantage of increasing the order of accuracy of the derivatives, as these are calculated directly. The price this time, is an increase in the number of unknowns. Nevertheless, since the computational effort grows linearly with the number of unknowns, the above increase is still outweighted by the savings brought by the MG method. Finally, the use of other convergent but rather elaborate iterative schemes [18] has recently been proposed [8].

A simpler approach is used in this paper: the difficulty with divergence is avoided by performing a direct solution on the coarsest grid. The coarsest grid has to be tight enough in order to include a range of low frequency Fourier components in the error, associated with the negative eigenvalues, responsible for the divergence of the scheme. This process renders the entire MG cycle to be convergent. The approach was formally discussed by Brandt [4] and Nicolaides [16] who also reported numerical tests in the case of Helmholtz's equation [5; 16] (i.e., $K^2 < 0$ and constant). In the case of Schrödinger's equation, however, there does not seem to exist criteria which will allow us to determine beforehand, how fine the coarsest grid has to be. Clearly, the eigenvalue spectrum of the associated matrix depends in a nontrivial way on the energy, the potential, and the boundaries and dimensions of the domain involved. This problem is analogous in many ways to deciding on a basis set size with close coupling or eigenfunction expansions. In practice, the divergence of the MG cycle because of an insufficiently fine mesh spacing on the first level, can be rapidly determined, just by studying the numerical interplay between the coarsest grids.

IV. The Full MG Appoximation Scheme for the Schrödinger Equation

This scheme appears to be very suitable for the case of the Schrödinger equation, particularly, because it can be easily adapted to deal with indefinite problems. In what follows, we shall describe a practical implementation of the algorithm outlined in the previous section, where the analysis of its operation was restricted to the case of only two grids. In the general case, we deal with a sequence of grids $\{G^1, G^2, ..., G^M\}$, G^1 being the coarsest and G^M the finest. If h_k is the mesh spacing in G^K, then $h_{k+1} = h_k/2$, and on each grid line every point of G^k is every other point of G^{k+1} (See Fig. 1). Moreover, h_M depends on the discretization accuracy required for

[2] An alternative, suggested by A. Jameson (private communication), is to consider the system

$$x = \mathbf{A}^T\mathbf{y} \text{ and } \mathbf{A}\mathbf{x} = \mathbf{b}, \text{ which yields } \mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{b}.$$

the solution of the differential equation. A classical flow starts on the coarsest grid G^1 and involves the use of standard operation blocs outlined below. During the execution, each one of the blocks is used several times and accessed according to certain convergence criteria. In our description, we shall distinguish between the current finest grid level m $(1 \leqslant m \leqslant M)$, i.e., the finest for which an approximate solution has this far been obtained, and the current operational grid level k $(1 \leqslant k \leqslant m)$. At level m we shall be dealing with iterations for the homogeneous

$$L^m u^m = 0, \tag{4.1}$$

while at level $k < m$ (i.e., at the correction stages), the relaxation sweeps are done for an inhomogeneous equation

$$L^k \hat{u}^k = \hat{f}^k \tag{4.2}$$

which coincides with Eq. (3.7) for $k = m - 1$, and where $\hat{f}^k$ is given in general by Eq. (4.18).

The basic operation blocks are described below in a more or less static fashion. We shall then use the particular flow of the algorithm for the model problem in Section V, to illustrate how and when these six blocks are typically accessed. It is important to keep in mind when going through their description, that they are not necessarily accessed in sequential order.

1. *Solution on the coarsest grid.* We access this block at two stages:

(i) at the initialization of the program, when we solve for u^1 in

$$L^1 u^1 = 0, \tag{4.3}$$

(ii) at a correction stage, when at a operational level $k = 1 < m$, where we solve for $\hat{u}^1$ in

$$L^1 \hat{u}^1 = \hat{f}^1. \tag{4.4}$$

As discussed above, because of the nondefinitiveness of the differential operator, a direct solution of the corresponding algebraic equation has to be performed in both cases. This is accomplished, by using a Gauss elimination solver which takes advantage of the bandedness and sparseness of the matrices involved, with regards to both storage and computational procedures.

After the solution process has been accomplished, a transfer of operations is made to either block 2 or 6 depending on whether this block was accessed at stage (i) or (ii), respectively.

2. *Interpolation to a new finer grid.* The access to this block is undertaken whenever the solution U^m at the current finest level m has been obtained within a prescribed tolerance ε_m (if $m = 1$, after the direct solution of Eq. (4.3)), and if $m < M$. If $m = M$ the algorithm ends. By using a coarse-to-fine interpolation, an initial guess

$$U^{m+1} = I_m^{m+1} U^m \tag{4.5}$$

is given for u^{m+1}, the solution of $L^{m+1}u^{m+1} = 0$. The order of this polynomial interpolation should be such as to ensure that the error of the quantities interpolated to the finer grid G^{m+1} will be as amooth as possible. If p is the order of our discretization process (i.e., $Lu - L^{m+1}u = O[(h_{m+1})^p]$) and n is the order of the differential operator, then the numerical resolution we expect for the solution at grid G^{m+1} is $O[(h_{m+1})^{n+p}]$. The self-consistency of the approximation scheme then requires the coarse-to-fine interpolation to have at least that same order at this mesh refinement stage. For the Schrödinger equation, and if $p = 2$ as in Eq. (2.2), this requires at least cubic interpolation. This operation can be performed in a very efficient way by using the difference equation itself [14]. To be specific, and having in mind our model problem in Section V, consider the two-dimensional Schrödinger diference equation on grid G^{m+1} obtained as a straightforward extension of Eq. (2.2)

$$u_{j+1,i} + u_{j,i+1} + u_{j-1,i} + u_{j,i-1} + D_{j,i}u_{j,i} = 0, \tag{4.6}$$

where $D_{j,i} = [k^2 - V(x_j, y_i)]\, h_{m+1}^2 - 4$.

The interpolation will be made up of three steps in this case. In the first, a simple injection is performed at the common points of G^m and G^{m+1}. By recalling Fig. 1 (with $k = m$), it is seen that the additional points of G^{m+1} for which values of U^{m+1} have to be generated fall into two categories, depending on whether their nearest common points (of G^m and G^{m+1}) are at a distance $\sqrt{2}\, h_{m+1}$ or h_{m+1}. By "rotating" Eq. (4.6) by 45° and replacing h_{m+1} by $\sqrt{2}\, h_{m+1}$ the values at the points of the first group are determined by the resulting equation, using the values at the common points. Finally, the values at the points of the remaining group are generated using

[redacted]

set equal to that new value.

3. *Relaxation sweep and estimate for the residual norm.* This block is accessed at the operational level $k \leqslant m$ with the purpose of improving the full curent approximation U^k (if $k = m$), or its representation $\hat{U}^k$ at grid G^k (if $k < m$), by means of a relaxation sweep. Correspondingly, the iteration is done either for the homogeneous equation $L^k u^k = 0$ or for the inhomogeneous one $L^k \hat{u}^k = \hat{f}^k$. Since these equations differ only on their right-hand side (the boundary conditions are also the same), this operation can be performed by a routine defined for all grids with an inhomogeneity which is set equal to zero when $k = m$. A measure of the improvement obtained through the iteration is the residual norm e_k, defined as follows:

$$\begin{aligned} e_k = \|R^k\| &= \|L^k U^k\|, & k = m, \\ &= \|L^k \hat{U}^k - \hat{f}^k\|, & k < m, \end{aligned} \tag{4.7}$$

where U^k and $\hat{U}^k$ refer to the approximate solutions after the iteration, and the norm is a discrete version of a suitable continuous norm. In practice, a convenient choice is

the Euclidean norm which, in two dimensions and with equal mesh spacing in both directions, is defined by

$$\|r^k(x)\| = \left[\sum_{x_i \in G^k}{}' |r^k(x_i)|^2 \right]^{1/2} \cdot h_k. \tag{4.8}$$

Moreover, an estimate for e_k can be obtained at a reduced computational cost by evaluating the so-called dynamical residual norm, which uses quantities that are calculated anyhow during the iteration. In what follows we outline the steps involved in a typical sweep for the two-dimensional Schrödinger equation with Dirichlet boundary conditions. The equation is taken to be inhomogeneous in order to include the two relevant cases (i.e., $k = m$ and $k < m$). Starting with the corresponding difference equation (cf. Eq. (4.6))

$$u_{j+1,i} + u_{j,i+1} + u_{j-1,i} + u_{j,i-1} + D_{j,i} u_{j,i} = f_{j,i}, \tag{4.9}$$

where $f \equiv \hat{f}$ and $\hat{u} \equiv u$ if $k < m$, and $f \equiv 0$ if $k = m$, we obtain the basic equation for the Gauss–Seidel iterative scheme considered here

$$u_{j,i} = |f_{j,i} - \tilde{u}_{j+1,i} - \tilde{u}_{j,j+1} - u_{j-1,i} - u_{j,i-1}|/D_{j,i}. \tag{4.10}$$

The terms $\tilde{u}_{j+1,i}$ and $\tilde{u}_{j,i+1}$ are components of the starting approximation for u (U^k or $\hat{U}^k$, before the iteration), while $u_{j,i}$, $u_{j-1,i}$, and $u_{j,i-1}$ refer to those of the improved one. Operationally, the block involves the successive calculation of the quantities

$$A_{j,i} = f_{j,i} - \tilde{u}_{j+1,i} - \tilde{u}_{j,i+1} - u_{j-1,i} - u_{j,i-1}, \tag{4.11}$$

$$B_{j,i} = A_{j,i} - D_{j,i} \tilde{u}_{j,i}, \tag{4.12}$$

$$u_{j,i} = A_{j,i}/D_{j,i}, \tag{4.13}$$

where we note that the required "starting" values $u_{j,1}$ and $u_{1,i}$ are fixed by the boundary conditions. The quantity $B_{j,i}$ is called the dynamical residual, and gives an estimate for the actual residual

$$R_{j,i} = f_{j,i} - u_{j+1,i} - u_{j,i+1} - u_{j-1,i} - u_{j,i-1} - D_{j,i} u_{j,i}. \tag{4.14}$$

It is interesting to note in passing, that by comparison of Eqs. (4.12) and (4.13) it also follows that

$$B_{j,i} = D_{j,i}(u_{j,i} - \tilde{u}_{j,i}). \tag{4.15}$$

As compared to the evaluation of $R_{j,i}$ through (4.14) requiring a multiplication and five additions per grid point (also the cost of a relaxation sweep per grid point) the calculation of $B_{j,i}$ is done at a reduced cost involving only a multiplication and one addition.

After the iteration has been completed and the residual norm e_k estimated, we transfer to block 4 for a convergence test.

4. *Convergence test.* At each level, with the exception of the first where the algebraic equations are solved exactly, to convergence tests are made: (a) the dynamic residual e_k calculated in block 3 is compared with a prescribed tolerance ε_k, (b) the convergence rate is checked by calculating the ratio of the dynamic residuals e_k and $\tilde{e}_k$ for the present and preceding iterates at the same grid G^k, followed by comparison of $e_k/\tilde{e}_k$ with a parameter η which is also prescribed. The choice of the parameters is discussed below. Particularly important is the choice of ε_m, the prescribed tolerance at the current finest grid level ($k = m$), which is taken to be comparable to the truncation error norm at level m. The choice of ε_k for $k < m$, which is relevant when correcting the current solution at level m, has a different definition given in the discussion of block 5. Depending on the results of these two tests, three possibilities exist:

(i) If $e_k \leqslant \varepsilon_k$ convergence is achieved, and we transfer to either block 2 or 6, respectively, depending on whether $k = m$ or $k < m$. The distinction comes when noting that in the first case the calculations at the current finest grid are ended and we proceed to the next finer grid seeking a further improvement of the solution by mesh refinement. For $k < m$, although the calculations at G^k have converged, we have yet to complete the cycle achieving convergence at the current finest level within the tolerance ε_m.

(ii) If $e_k > \varepsilon_k$, while $e_k/\tilde{e}_k \leqslant \eta$, the rate of convergence of the iterations is considered acceptable at the present grid and the transfer is again made to block 3 for a new sweep.

(iii) If $e_k > \varepsilon_k$ and $e_k/\tilde{e}_k > \eta$, the convergence rate is considered unsatisfactory at the present grid, and we transfer the operations to the next coarser grid, as described in block 5. Having that transfer in mind, the value of the operational level index k is changed from k to $k - 1$.

5. *Coarse grid correction.* Coming from block 4, and in order to proceed with the iterations at the new operational level k, we need three pieces of information: a starting approximation $\hat{U}^k$, the inhomogeneity $\hat{f}^k$ and the definition of a tolerance ε_k. The initial approximation $\hat{U}^k$ is obtained by means of a fine-to-coarse interpolation from G^{k+1}

$$\hat{U}^k = I_{k+1}^k \hat{U}^{k+1}, \tag{4.16}$$

where $\hat{U}^{k+1} = U^{k+1} = U^m$ if $k = m - 1$.

In defining $\hat{f}^k$ we follow the same line of reasoning as that leading to Eq. (3.7), except that instead of having the homogeneous equation (3.3) as the starting point, we now have, in general, the inhomogeneous one

$$L^{k+1}(\hat{U}^{k+1} + \hat{V}^{k+1}) = \hat{f}^{k+1}, \tag{4.17}$$

which of course coincides with (3.3) if $k = m - 1$, in which case $\hat{f}^{k+1} \equiv 0$, $\hat{V}^{k+1} = V^{k+1}$, and again, $\hat{U}^{k+1} = U^{k+1}$. By doing so, we get

$$\hat{f}^k = L^k \hat{U}^k + I_{k+1}^k (\hat{f}^{k+1} - L^{k+1} \hat{U}^{k+1}). \tag{4.18}$$

Finally, the tolerance ε_k for the iterations at this level is taken to be a fraction of the last value for the dynamical residual norm e_{k+1} at the preceding operational level. That is,

$$\varepsilon_k = \delta e_{k+1}, \tag{4.19}$$

where $0 < \delta < 1$ is also a prescribed parameter of the MG scheme.

Having defined the problem for grid G^k, the operations are transferred to block 3, where a relaxation sweep is to be performed.

6. *Transfer of corrections to a finer grid.* This block is accessed whenever conergence at an operational level $k < m$ has been attained. We use the same arguments leading to Eq. (3.8) with suitable modifications (see discussion in block 5). If $\hat{U}_{\text{old}}^{k+1}$ is the solution at level $k+1$ to be improved, and $\hat{U}^k$ is the solution obtained at level k within the prescribed tolerance, we get the improved solution $\hat{U}_{\text{new}}^{k+1}$ through the equation

$$\hat{U}_{\text{new}}^{k+1} = \hat{U}_{\text{old}}^{k+1} + I_k^{k+1} (\hat{U}^k - I_{k+1}^k \hat{U}_{\text{old}}^{k+1}). \tag{4.20}$$

Since we are dealing here with the coarse-to-fine interpolation or corrections, the requirements on the order of the polynomial interpolation involved are not as high as those in the mesh refinement stage (see block 2). In this case, usually linear-type interpolation is sufficient.

In preparation for the transfer to block 3, where an iteration will be performed taking $\hat{U}_{\text{new}}^{k+1}$ as the starting approximation, the operational level index k is changed from k to $k+1$.

As discussed above, the flow of the program through the different grids is controlled by the parameters η and δ, while a knowledge is assumed of the truncation error norm at each level. That error clearly is not known, unless the exact solution to the continuous problem is given. An important byproduct of the full approximation scheme is that it allows for estimates of those errors, evaluated within the cycle of the program. Returning to the two-level analysis of Section IIIA, recall that after obtaining an approximate solution U^m at level m we seek an improvement at level $m-1$ by solving Eq. (3.7), namely,

$$L^{m-1} \hat{u}^{m-1} = \tau_m^{m-1} = L^{m-1} I_m^{m-1} U^m - I_m^{m-1} L^m U^m \tag{4.21}$$

The right-hand side of (3.21) can be regarded as the local truncation error at level

$m-1$ relative to level m [6]. This follows by comparison with the equation that defines the usual local truncation error at level $m-1$,

$$\tau^{m-1} = L^{m-1}u - Lu, \tag{4.22}$$

where u is the solution to the original continuous problem. For sufficiently small h_{m-1} we expect

$$\tau^{m-1} \approx a(h_{m-1})^p, \tag{4.23}$$

where p is the order of approximation of the discretization process ($p=2$ in our case), and a is independent of h_{m-1}. Thus, we also expect to have

$$\tau^m \approx (h_m/h_{m-1})^p \tau^{m-1}. \tag{4.24}$$

Moreover, by comparison of Eqs. (4.21) and (4.22) it follows that

$$\tau_m^{m-1} \approx \tau^{m-1} - \tau^m, \tag{4.25}$$

and by recalling (4.23) we get

$$\tau^{m-1} \approx [1-(h_m/h_{m-1})^p]^{-1} \tau_m^{m-1} \tag{4.26}$$

which gives an estimate for the local truncation error at level $m-1$. By now using again (4.24), we finally get the desired estimate for level m

$$\tau^m \approx (h_m/h_{m-1})^p/[1-(h_m/h_{m-1})^p]\, \tau_m^{m-1}. \tag{4.27}$$

The above analysis then suggests taking ε_m to be given by

$$\varepsilon_m = \|\tau^m\| \approx \alpha \|\tau_m^{m-1}\| = \alpha \|\hat{f}^{m-1}\|, \tag{4.28}$$

where $\hat{f}^{m-1}$ is calculated by (4.18) and α is a suitable constant to be prescribed. A natural choice for α (perhaps not necessarily optimal) clearly is

$$\alpha = (h_m/h_{m-1})^p/[1-(h_m/h_{m-1})^p] \tag{4.29}$$

which for our case ($h_m/h_{m-1} = \frac{1}{2}$ and $p=2$) yields $\alpha = \frac{1}{3}$.

In executing the algorithm we must then choose three parameters: η, δ, and α. Criteria for determining optimal values for η and δ are given in [4]. Usually, η is taken to be equal to the smoothing factor (e.g., that given by Eq. (2.8)). In this way, the relaxation sweeps at each level are expected to do that part of the work for which they are efficient (i.e., filtering high frequency errors). In practice, however, the flow

more dependent on that of α which actually fixes the accuracy that can be obtained in solving the discretized problem at each level.

Finally, we note that MG cycles having fixed flows can be devised, with the entire flow prescribed in advance depending on input parameters but not on internal checks.

They would, of course, be preferred in order to avoid the evaluation of residual norms, and hence reduce the computational work [7]. It should be noted, however, that the choice of fixed flows may require extensive experience with the class of problems under consideration.

V. A Model Problem

The following example is presented here in order to make concrete the use of the MG method, the magnitude of savings that can be achieved, and the order of accuracies that can be expected. Some details of a practical nature with regard to the application of the algorithm, will also be indicated.

The problem consists in solving the two-dimensional stationary Schrödinger equation

$$[-\nabla^2 + V(x, y)]\,\psi(x, y) = k^2\psi(x, y), \tag{5.1}$$

where the potential is chosen as

$$V(x, y) = A_1 \exp\{[-(x - x_{01})^2 - (y - y_{01})^2]/\sigma_0^2\} - A_2 \exp\{[-(x - x_{02})^2 - (y - y_{02})^2]/\sigma_0^2\}$$

on the rectangular domain $\{0 \leqslant x \leqslant X,\ 0 \leqslant y \leqslant Y\}$ with the boundary conditions

$$\psi(0, y) = \psi(X, y) = 0, \qquad \psi(x, 0) = \psi(x, Y) = \sin x.$$

The parameters used in the actual calculations were $A_1 = 3$, $A_2 = 1$, $k = 1$, $x_{01} = 6.6$, $x_{02} = 15.4$, $y_{01} = y_{02} = 1.575\pi$, $\sigma_0^2 = 4.93$, $X = 7\pi$, and $Y = 3.15\pi$. The potential $V(x, y)$ is made up of a Gaussian hill of height A_1 and a Gaussian well of depth A_2 centered, respectively, at (x_{01}, y_{01}) and (x_{02}, y_{02}). The problem contains all the essential features that might cause numerical difficulties, and is quite typical of those that appear in scattering problems, with regions of waves and tunneling simultaneously present. The PDE (5.1) is discretized by a five-point finite difference scheme as in Eq. (2.2), the Gauss–Seidel relaxation scheme defined by Eq. (4.10) is used for all iterations and the MG algorithm is the full approximation scheme discussed in Section IV. The switching parameters for the flow of the algorithm were chosen to be $\eta = 0.6$ (an upperbound to the smoothing factor as given by Eq. (2.9)), $\delta = 0.4$ and $\alpha = \frac{1}{3}$. In order to explicitly see what has happened in the actual execution of the program for our model problem, we present its concrete flow in Fig. 2. The grid-levels 1–4, with the number of intervals in the x and y directions as indicated in parenthesis, are those involved in the calculations. The numbers inside the squares indicate the number of iterations that were performed at each stage. The arrows indicate the succession of events. More specifically, the flow starts at level 1 ($m = 1$) where a direct solution of the algebraic equations yields U^1. The interpolation I_1^2 transfers U^1 to the new current finest grid level 2 (m is set equal to 2).

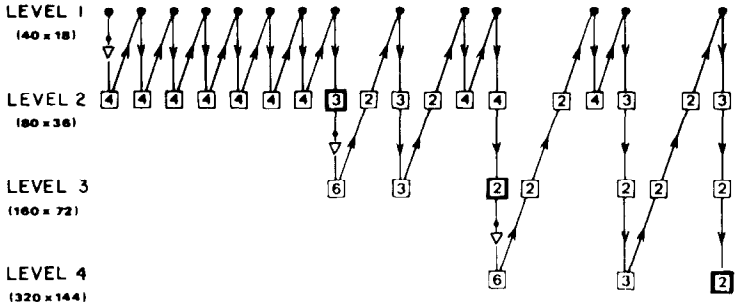

FIG. 2. Schematic diagram of the flow for the model problem. The number of relaxation sweeps at each level are indicated inside the squares. A thicker square indicates that the stopping criteria $\|L^m U^m\| \leqslant \varepsilon_m$ at that level was satisfied. Stages at which the operation blocks are used: Block 1: ●; Block 2: ↓ (cubic interpolation); Blocks 3 and 4: □, ▣; Block 5: ↑ (injection of residuals); Block 6: ↓ (bilinear interpolation of corrections). The number of intervals at the corresponding grids are indicated in the parenthesis (e.g., (40×18)) at each level.

The approximation $U^2 = I_1^2 U^1$, thus obtained, serves as a first guess for a series of four iterations that are performed at this level. From this point, the flow returns back to level 1 (now a correction stage, with $k = 1 < m = 2$), since the rate of convergence is no longer satisfactory after four iterations. The process of transfering between levels 1 and 2 is repeated several times until we satisfy (at the thicker square with a number 3 inside) the stopping criteria in level 2 at the third iteration. At this point, the flow goes on to the third level (m is set equal to 3). Again, when the convergence rate becomes unsatisfactory at this level after six iterations, a correction is pursued at the next coarser grid. This takes the flow to level 2 ($k = 2 < m = 3$). After several transfers between levels 1 and 2 and eventually 3, convergence is attained at level 3 (thicker both with a number 2 inside) after two iterations. Since convergence has been achieved at level 3, the flow goes onto level 4 (m is set equal to 4). The first time at this level the convergence rate becomes unsatisfactory after six iterations, and the flow goes to levels 3, 2, and 1, respectively. Eventually, at the end of the line, convergence at level 4 is also achieved after two iterations. Here the program ends.

Table I presents a comparison of the computational effort by the MG and direct methods. The computational efforts are measured by the number of additions, counted as one unit, and multiplications and divisions, counted as three and nine, respectively. These appear as somewhat standard units for measuring computational effort [10]. For the MG calculations, the operation count is done directly during the computations, including relaxation sweeps, interpolations, and direct solutions on the coarse grid.[3] For the direct solver, the operations count is estimated by the formula

$$(9 + 4B)\, BN \simeq (9 + 4N^{1/2})\, N^{3/2},$$

[3] Estimates for the computational work associated with traditional MG cycles involve only relaxation sweeps, which dominate most of the work (see, e.g., [4]). In particular, the work in the first level is typically negligible. In our calculations, however, this is not the case, mainly because direct solutions are always required on the coarsest grid.

TABLE I

Results for the Model Problem

Level	Grid	N^a	h	Memory Requirements[b]		Accuracy in[c] Norm Relative to Level 4	Work[d]		Estimated Factor of Savings in Work
				MG	Direct		MG	Direct	
1	40×18	663	.55	$.1M$	$.1M$	6.7×10^{-1}	6.	6.	1
2	80×36	2765	.275	$.2M$	$.8M$	2.4×10^{-2}	4.9×10	$1. \times 10^{2}$	2
3	160×72	11289	.1375	$.5M$	$6.6M$	6.6×10^{-3}	7.1×10	1.6×10^{3}	23
4	320×144	45617	.06875	$1.6M$	$53.0M$	—	1.1×10^{2}	2.6×10^{4}	236

[a] $N = (n-1) \times (m-1)$: number of unknowns for the $n \times m$ grid. Boundary points are excluded.

[b] Memory in bytes. Double precision assumed; $M = 10^6$ bytes.

[c] Accuracy defined for level $m \leqslant 3$, as $(\sum_{G_c} (U_{jk}^{(m)} - U_{jk}^{(4)})^2 / \sum_{G_c} (U_{jk}^{(4)})^2)^{1/2}$, where G_c is the coarsest grid.

[d] A unit of work is equal to $\simeq 3.2 \times 10^5$ additions, equivalent to a relaxation sweep at the finest grid (1 multiplication and 4 additions per grid point).

where, as before, N is the number of unknowns, and B is the half-bandwith of the matrix of the system of algebraic equations. This expression is obtained in a straightforward way by performing the actual operations count for a Gauss elimination solver involving a banded matrix.

As mentioned before, Fig. 2 is a sketch of the actual cycle corresponding to the MG figures of Table I. A full thicker square indicates that the stopping criteria $\|L^m U^m\| \leqslant \varepsilon_m$ was satisfied at that level. The accuracies in column 7 of Table I refer to these last stages. It must be appreciated that the direct solution of the equations at the fourth level, involving ~45,000 unknowns, and even at the previous one with ~11,000 unknowns, is not feasible in core (on an IBM 3081) with a banded matrix solver, because of memory requirements.[4] The last column of the table gives the estimated factors of savings in work at each level. These factors are dependent on the model problem and also somewhat dependent on the particular flow of the MG cycle (note that only estimates can be given for the truncation error at each level). Nevertheless, the figures clearly indicate that the MG method can produce dramatic computational savings. Referring to the storage requirements for the MG method in Table I, it should be noted that they can be reduced by more efficient programming (see discussion on storage requirements for the MG algorithm in [7]. The storage figures in Table I correspond to a code which followed the sample program in [4] with regards to the basic arrays involved, except for a relative increase of somewhat less than 50% in storage space (required to define the potential at each grid).

A higher degree of optimization can still be expected, through the reduction of both computational work and storage requirements, by using adaptive discretization [6, 7]. This would allow for efficient adaptation of the algorithm to the particular features of the potential and boundary conditions involved. These requirements are crucial, in particular, on the first level where direct solutions are to be performed.

VI. Discussion

The computational work and core memory savings for realistic accuracies, as discussed and illustrated for the example of Table I, are quite encouraging. The multigrid method constitutes an extremely efficient procedure for solving Schrödinger's equation in its original partial differential equation form.

The problem of deciding on an optimal and robust scheme in order to deal with the nondefiniteness of the problem demands further study. The particular scheme used in this paper may not be efficient when relatively large domains and/or energies are involved, due to the requirements on the "tightness" of the coarsest grid to ensure the stability of the algorithm. The results of our numerical experiments indicated, in particular, that besides the usual requirements for numerical resolution of the discretization process, larger domains required finer meshes. This latter requirement

[4] The storage requirements for the noniterative block technique [17] are much less. However, the method involves more arithmetic computations per solution as compared to the banded matrix solver.

is not confined to the MG approach, but rather is inherent in physical problems exhibiting oscillating solutions for which point-wise approximations are sought [19].

The memory and computational time savings should open up larger classes of problems for accurate numerical solution. In this work the method is applied to the simple finite difference discretization of Schrödinger's equation. It can be coupled in a straightforward manner, with the more flexible finite element discretization [3]. Furthermore, for the time-dependent Schrödinger equation with implicit time-integration schemes, the method would be desirable solution procedure at the discrete time steps.

Finally, we note that outside the realm of linear equations, the multigrid technique has also been used in conjunction with nonlinear problems. Hence, it would provide a viable procedure for the solution of nonlinear partial differential equations such as those of interest in chemical kinetics.

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research and the National Science Foundation. The authors acknowledge very useful and stimulating discussions on the multigrid method with Professors Antony Jameson and Achi Brandt. We also thank Professor Ahmet S. Cakmak for a critical reading of the manuscript, and Dr. Aaron Temkin for his useful comments.

Note added in proof. The authors wish to thank Drs. E. C. Sullivan and A. Temkin for running the SEPDE code [17] for the model problem of Section V. The comparison with their results provided a valuable check of the accuracy of the numerical solutions obtained with the MG method.

REFERENCES

1. B. ADLER, S. FERNBACH, AND M. ROTENBERG, Eds., *in* "Atomic and Molecular Scattering," Vol. 10, Academic Press, New York, 1971.
2. W. F. AMES, "Numerical Methods for Partial Differential Equations," Academic Press, New York, 1977.
3. A. ASKAR, A. S. CAKMAK, AND H. RABITZ, *Chem. Phys.* **29** (1978), 61; **33** (1978), 267; *J. Chem. Phys.* **72** (1980), 5287.
4. A. BRANDT, *Math. Comput.* **31** (1977), 333.
5. A. BRANDT, *in* "Mathematical Software III" (J. R. Rice, Ed.), p. 277. Academic Press, New York, 1977).
6. A. BRANDT, *in* "Numerical Analysis of Singular Perturbation Problems" (P. W. Hemker and J. J. H. Miller, Eds.), p. 53, Academic Press, New York, 1979.
7. A. BRANDT, *in* "Special Topics of Applied Mathematics" (J. Frehse, D. Pallaschke, and B. Trottenberg, Eds.), p. 91, North-Holland, Amsterdam, 1980.
8. A. BRANDT AND S. TA'ASAN, "Multigrid Methods for Highly Oscillatory Problems," Research Report, Weizmann Institute, Rehovot, 1982.
9. C. BREZINSKI, "Accélération de la Convergence en Analyse Numérique," Springer-Verlag, Berlin, 1977.
10. C. BREZINSKI, *J. Comput. Appl. Math.* **2** (1976), 113.

11. H. K. CHENG AND H. M. HAFEZ, *in* "Pade Approximant Methods and Its Application to Mechanics" (H. Cabannes, Ed.), p. 101, Springer-Verlag, New York, 1976.
12. G. J. FIX, M. D. GUNZBERGER, AND R. A. NICOLAIDES, *Comput. Math. Appl.* **5** (1979), 87.
13. L. A. HAGEMAN AND D. M. YOUNG, "Applied Iterative Methods," Academic Press, New York, 1981.
14. S. M. HYMAN, *J. Comput. Phys.* **23** (1977), 124.
15. A. JAMESON, *in* "Proceedings of the AIAA Computational Fluid Dynamics Conference," p. 122, Williamsburg, 1979.
16. R. A. NICOLAIDES, *Math. Comput.* **32** (1978), 1082; **33** (1979), 933.
17. E. C. SULLIVAN AND A. TEMKIN, *Comput. Phys. Comm.* **25** (1982), 97.
18. K. TANABE, *Numer. Math.* **17** (1971), 203.
19. A. BRANDT, private communication.