JOURNAL OF COMPUTATIONAL PHYSICS **102**, 88–97 (1992)

Spectral Methods in Time for a Class of Parabolic Partial Differential Equations

GLENN IERLEY, BRIAN SPENCER,* AND RODNEY WORTHING†

Department of Mathematical Sciences, Michigan Technological University, Houghton, Michigan 49931

Received November 29, 1990; revised August 15, 1991

In this paper, we introduce a fully spectral solution for the partial differential equation $u_t + uu_x + vu_{xx} + \mu u_{xxx} + \lambda u_{xxxx} = 0$. For periodic boundary conditions in space, the use of a Fourier expansion in x admits of a particularly efficient algorithm with respect to expansion of the time dependence in a Chebyshev series. Boundary conditions other than periodic may still be treated with reasonable, though lesser, efficiency. For all cases, very high accuracy is attainable at moderate computational cost relative to the expense of variable order finite difference methods in time. © 1992 Academic Press, Inc.

I. INTRODUCTION

Spectral (global) methods for resolving spatial structure in the numerical solution of partial differential equations are now at a mature stage of development. The journal literature is fairly extensive even if the number of scholarly treatises is minuscule in comparison to the literature on the more widely used finite difference and finite element methods. The application of spectral methods to resolution in the time domain is less well explored, but it is natural to expect that this method is well suited to applications in which extreme accuracy is desired. The articles appearing in [1] make it clear that the efficiency and robustness of a spectral approach in time remain to be established for a broad class of problems. This paper is intended to contribute to that end by focusing on the details of two particular applications clarifying, one hopes, the general principle with an illustrative example.

A familiar result, cited to justify spectral expansions in the spatial domain, is the well-known asymptotic exponential convergence rate for spectral basis functions which are the solution of a singular Sturm–Liouville equation. For example, the function $f(x) = e^{-\beta x}$ has as its Chebyshev expansion,

$$f(x) = \sum_{k=0}^{\infty} f_k T_k(x), \qquad \text{where} \quad f_k = \frac{2}{\pi c_k} I_k(\beta) \tag{1}$$

(where $c_k = 1$ except for $c_0 = 2$), but a glance at a set of tables shows that exponential decay does not immediately obtain for the coefficient spectrum. For problems in which it is not feasible to increase the resolution owing to an excessive increase in computation, coordinate transformations defining a one-to-one mapping given by some $z(x)$ are one obvious notion for improving convergence. In the example above there is obviously a map such that $f(z(x)) = x$ so that the f_k's vanish identically for $k > 1$. For some equations, a simple analytic mapping is useful and then the differential equation may be recast in the new coordinates in exact terms. A more general approach is to update the mapping as part of an iterative solution method—a spectral analogue of adaptive mesh refinement for finite difference and finite element formulations. This trades off the problem of resolving f for one of resolving the map, z. An optimal coordinate transform would balance the resolution required of each. There are many interesting theoretical issues of approximation theory including the asymptotic rate of convergence of an optimal map which seem to be little explored.

An alternative strategy which shows great promise is spectral element domain decomposition, as in [2]. Various strategies for subdividing the domain are easily envisioned, and the possibility of an adaptive version is also apparent, but optimal algorithms of any generality remain to be developed. The general expectation, and our motivation here, is that the order of increase in accuracy is algebraic in the number of elements and exponential (at least for C^∞ functions) in the order of the basis set on each element. These considerations are of natural interest in connection with the present work, owing to the fact that a spectral method in time will nearly always be a spectral element method; i.e., constrained by the spectral radius of some iterative matrix operator, it will generally be necessary to subdivide the time domain into some number of elements.[1]

We exploit the potential for accuracy of spectral methods in treating two problems. A first application of the spectral

* Permanent address: Department of Engineering Science and Applied Mathematics, Northwestern University, Evanston, IL 60201.

† Permanent address: Department of Mathematics, MIT, Cambridge, MA 02139.

[1] In experiments so far, we have only enforced C^0 continuity of adjacent elements, the natural choice for a first-order evolution operator.

0021-9991/92 $5.00

approach is to the family of partial differential equations of the form:

$$u_t + (u-c)\,u_x + \nu u_{xx} + \mu u_{xxx} + \lambda u_{xxxx} = 0 \qquad (2)$$

for the case of periodic boundary conditions. This equation, first advanced by [3], possesses in various limits the well-known and much studied cases of integrable Korteweg de Vries ($\nu = \lambda = 0$), chaotic Kuramoto–Sivashinsky ($\mu = 0$), and classical Burgers ($\mu = \lambda = 0$) equations. The full equation is the subject of more recent study exploiting only the Galilean and translational invariances of the generic case, for example, [4–7]. In [7], the imposition of periodic boundary conditions permits one to make contact in the limit of a large period with a variety of homoclinic orbits of the reduced ordinary differential equation obtained from the similarity variable, $x - ct$, characterized (in contrast with the special case of integrability and consequent scale invariance) by a discrete spectrum of allowed values of c relative to a fixed reference frame. Additionally, and more important in general application, for lengthy integration the choice of periodic solutions ensures that the endpoints act as neither source nor sink. This issue is a potential complication of alternative conditions such as absorbing boundaries.

Even if, owing to an implicit treatment, they impose no stability constraint, the fourth-order spatial derivatives in (2) can uncomfortably restrict the size of the time step required to achieve a given accuracy. The virtue of the spectral method is that local accuracy in x and t on the order of machine accuracy is attainable at moderate computational expense. Naturally the notion of global error does not play so pivotal a role in the extended integration of exponentially sensitive, i.e., chaotic, systems, since word length alone provides a typically brief upper bound for times beyond which detailed numerical comparison as between machines, let alone other algorithms on a given machine, is rendered meaningless. For particular cases of (2) which exhibit special properties, such as integrability, one may speak of various *statistical* measures of evolution and then particular means such as the symplectic integration scheme devised by [8] are useful in producing results more faithful to symmetries of the original continuous problem than naïve integration schemes. For the generic equation (2), of course, one does not have the luxury of a large number of confining constraints serving to define a preferred class of time stepping algorithms. For the Kuramoto–Sivashinsky problem, [9] notes a variety of problems stemming from insufficient accuracy in integration, among them, states near the transition to chaos incorrectly rendered *stable* as an artifact of numerical error. They suggest local (absolute) error control on the order of 10^{-10} is minimally necessary to avoid computations which are even qualitatively misleading.

Although not as efficient numerically for reasons to be discussed, we show also the application of the method to the special case of Burgers equation given by

$$u_t + uu_x = \nu u_{xx}. \qquad (3)$$

in a domain with homogeneous, rather than periodic, boundary conditions. The results are compared to those from a survey paper by [10] for a specific test problem possessing an explicit solution generated from the standard Cole–Hopf transformation. Comparison with the exact results in that case will allow us to assess the accuracy and efficiency with less heuristic uncertainty. It emerges that computation of the expression for the exact solution is a more exacting process than generating the same accuracy by spectral means.

II. THE METHOD

To solve (2), we introduce the representation

$$u(x,t) = \sum_{j=0}^{M-1} \sum_{k=1}^{N} u_{j,k} T_j(\tau)\, e^{ikx}, \qquad (4)$$

where $\tau = 2(t - t_i)/(t_{i+1} - t_i) - 1$ and T_j is the usual Chebyshev polynomial. The function u will satisfy an initial condition $u(x, -1) = U(x, t_i)$. The t_i's are at discrete intervals chosen to be roughly as large as permitted by M, consistent with convergence to some prespecified tolerance.

Substituting (4) into (2) and using orthogonal projection produces MN coupled quadratically nonlinear algebraic equations which are of the form:

$$\mathcal{L}(u)_{j,k} \equiv D_{j,l} u_{l,k} + ik/2(u * u)_{j,k} - (ick + \nu k^2 + i\mu k^3 - \lambda k^4)\, u_{j,k} = 0. \qquad (5)$$

The matrix D is standard (see [11]). It is of the form

$$\frac{2}{t_{i+1} - t_i} \begin{pmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & \cdots \\ 0 & 0 & 4 & 0 & 8 & 0 & 12 & \cdots \\ 0 & 0 & 0 & 6 & 0 & 10 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 8 & 0 & 12 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & \cdots \\ \vdots & \vdots & & & & & & \end{pmatrix}. \qquad (6)$$

As is common in the application of Chebyshev polynomials, we use the tau method and so we discard the N equations corresponding to the projection with T_{M-1}, replacing these with the N equations

$$\sum_{j=0}^{M-1} (-1)^j u_{j,k} = U_k, \qquad k = 1, \ldots, N, \qquad (7)$$

where the U_k are the Fourier expansion coefficients of $U(x, t_i)$.

Most of the computation time for this problem is in computing the two-dimensional convolution of uu_x. To avoid

aliasing error in space, we use the $\frac{3}{2}$ rule to transform onto $3(N+1)$ real space points in x. Although for sufficiently large M it would be advantageous to do the same in time, in fact, it proves more efficient to use the $\mathcal{O}(M^2)$ algorithm based on the elementary identity $T_jT_l=(T_{j+l}+T_{|j-l|})/2$ up to fairly large M. The crossover point will depend upon the precise efficiency of the software available to the user. In our computations, it was about $M=32$. The IMSL FFT and cosine transform routines, in particular, seem to execute rather slowly, especially on the Sun 4/260. For example, a fully unrolled 48-point Winograd version of the FFT executed in 9.6 ms compared to 50.3 ms for the IMSL routines F2TR(B, F). The direct time convolution is best coded for a vector machine by forming a matrix operator for the first member of the convolution, $\tilde{u}_k$ (the real space half transform at $x=x_k$, $k=1, ..., 2N+2$), and then contracting with the M component vector for the second member.

Our solution of (5) is based upon Newton's method; thus we proceed in iterative fashion to solve

$$D_{j,l}\delta u_{l,k}^{(n+1)} + ik(u^{(n)} * \delta u^{(n+1)})_{j,k} - (ick + \nu k^2 + i\mu k^3 - \lambda k^4)\,\delta u_{j,k}^{(n+1)} = -\mathscr{L}(u^{(n)})_{j,k}. \tag{8}$$

and then we set $u_{j,k}^{(n+1)} = u_{j,k}^{(n)} + \delta u_{j,k}^{(n+1)}$. Note that δu satisfies a homogeneous boundary condition at $\tau=-1$. We start the algorithm with $u_{j,k}^{(0)} = \delta_{0,j}U_k$.

Now the solution of (8) is yet too cumbersome to permit rapid solution, since the convolution term couples all MN equations. We thus introduce a subiteration scheme in the form

$$D_{j,l}\delta u_{l,k}^{(n+1,m+1)} - (ick + \nu k^2 + i\mu k^3 - \lambda k^4)\,\delta u_{j,k}^{(n+1,m+1)} = -\mathscr{L}(u^{(n)})_{j,k} - ik(u^{(n)} * \delta u^{(n+1,m)})_{j,k} \tag{9}$$

which supplants the use of (8). Under normal circumstances, the nested iteration on m in (9) converges in three to five iterations to the equivalent solution of (8) for $\delta u_{j,k}^{(n+1)}$, but without requiring direct inversion of the coupled implicit operator. Convergence in m is followed by a step in n. There is a potential trade-off to consider in optimizing the total operations count. Fewer iterations of the inner loop in m slow the convergence of the Newton's method iteration in n while more inner iterations, up to a point, accelerate the outer loop. Over a fair range of choices, however, the *total* number of iterations required is approximately constant so that execution time for the algorithm is relatively insensitive to the choice of inner iteration loop count. However, note that each step in n requires two forward transforms and one reverse, while stepping in m needs only *one* forward and one reverse since $u^{(n)}$ remains fixed; thus there is slight advantage to erring on the high side in iteration of the inner loop since convolution accounts for nearly all of the computation time.

Convergence in Newton's method is monitored by the quantity

$$\Delta \equiv \frac{1}{MN}\sqrt{\sum_{j,k}\delta u_{j,k}^2}. \tag{10}$$

If Δ exceeds one, the time step is halved as this normally denotes divergence of Newton's method. Provided $\varepsilon<\Delta<1$, where ε is a specified tolerance, we continue with Newton's method at the current time step unless the total number of steps has become excessive. As soon as $\Delta<\varepsilon$, we compute the spectral indicator

$$\Delta_S = \max_k |u_{M,k}| / \max_{j<M,k} |u_{j,k}|. \tag{11}$$

Now if $\Delta>\Delta_S$, we halve the time step, and use for our initial guess a map from the solution already found on the interval $[-1, 0]$ onto the interval $[-1, 1]$. This projection merely requires N matrix multiplications of order M and the appropriate matrix operator is precomputed at the start of the program. The resulting initial guess usually converges in one or two more iterations of Newton's method. (The same projection method is employed once at the end of the run if, as normally will occur, the final time step does not coincide with the specified interval for integration.)

Provided $\Delta \leqslant \Delta_S$, we accept the answer for u on the interval $[t_i, t_{i+1}]$, and evaluate $u(x, 1)$ to provide the initial condition for the next interval. When $\Delta \leqslant \varepsilon_S\Delta_S$, then our time steps are deemed too small, and the next interval is provisionally doubled in size. The choice of ε_S is M dependent. For large M, it must be fairly small, since the effect on the spectral decay of doubling the time interval can be substantial. Typical values are $\varepsilon_S=10^{-3}$ for $M=17$ and 2×10^{-2} for $M=9$ for the runs in Section III. In those runs with an average time step, $(t_{i+1}-t_i)$, of Δt, variation in the spatio-temporal evolution of the solution permits intermittent periods with steps as large as $16\Delta t$ and requires occasional isolated steps as small as $\Delta t/16$. Obviously for larger M, one could change the step size by a factor smaller than 2 for greater potential efficiency, but we have not explored that possibility.

The solution of (5) is now usefully decoupled into N separate problems of order M.[2] In a periodic domain, this

[2] In what follows, we now wish to distinguish the matrix dimension, which we have so far termed M, from the maximum order polynomial which differs by one. As in FORTRAN IV, one sweated constantly to distinguish indexing of arrays from 1 to M from the order of polynomial represented, 0 to $(M-1)$, so too here we would be faced with unreasonable typographic inelegance to make the same distinction. Still, to introduce yet another variable for quantities so trivially related seems unnecessarily prolix. Thus we adopt the expedient of employing the same symbol to mean two different things. Simple logic will discriminate usage by context. Although the strategy may seem perverse, as Joe Keller has pointed out, we use an enormous variety of ways to rewrite zero and nobody seems to object to that practice.

decoupling obtains by virtue of the fact that the linear operators are all constant coefficient and are thus diagonal for a basis set of complex Fourier modes. The N decoupled problems, $\hat{\mathbf{A}}(k)\, u_k = b_k$, $k = 1, ..., N$, each assume the following form for the matrix operator,

$$\hat{\mathbf{A}}(k) = \begin{pmatrix} \lambda_k & 1 & 0 & 3 & 0 & 5 & \cdots \\ 0 & \lambda_k & 4 & 0 & 8 & 0 & \cdots \\ 0 & 0 & \lambda_k & 6 & 0 & 10 & \cdots \\ \vdots & \vdots & & & & & \\ 0 & 0 & \cdots & 0 & 0 & \lambda_k & 2M \\ 1 & -1 & \cdots & 1 & -1 & 1 & -1 \end{pmatrix}. \tag{12}$$

(Note that we absorb the earlier prefactor $2/(t_{i+1} - t_i)$ into the definition of λ_k. Also, the last row of $\hat{\mathbf{A}}(k)$ is replaced with coefficients of $T_j(-1) = (-1)^j$ to enforce the initial condition on u.) This matrix is easily reduced to quasitridiagonal form by elementary row operations and would seem straightforward to solve by simple Gaussian elimination of the last row. Unfortunately even for $M = 9$, the condition number of the resultant matrix can be astronomical for small λ_k; for example, $\lambda_k = 0.01$ leads to a condition number of 10^{23}.

Contingent upon a fixed eigenspectrum for the implicit spatial operator, an efficient ($\mathcal{O}(M)$) and highly accurate solution for all $|\lambda_k|$ is found as follows: As noted above, elementary manipulation of the matrix suffices to bring it and the right-hand side to the respective forms:

$$\begin{pmatrix} 1 & -1 & \cdots & 1 & -1 & 1 & -1 \\ \lambda_k & 1 & -\lambda_k/2 & 0 & 0 & 0 & \cdots \\ 0 & \lambda_k & 4 & -\lambda_k & 0 & 0 & \cdots \\ 0 & 0 & \lambda_k & 6 & -\lambda_k & 0 & \cdots \\ \vdots & \vdots & & & & & \\ 0 & 0 & \cdots & \lambda_k & 2M-4 & -\lambda_k & 0 \\ 0 & 0 & \cdots & 0 & \lambda_k & 2M-2 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \lambda_k & 2M \end{pmatrix} \times \begin{pmatrix} b_M \\ b_0 - b_2/2 \\ b_1 - b_3 \\ b_2 - b_4 \\ \vdots \\ b_{M-3} - b_{M-1} \\ b_{M-2} \\ b_{M-1} \end{pmatrix}. \tag{13}$$

We term the matrix above $\mathbf{A}$ and the vector b. We next introduce an auxiliary tridiagonal matrix, defined as

$$\mathbf{T} = \begin{pmatrix} 1 & -\lambda_k & \cdots & 0 & 0 & 0 & 0 \\ \lambda_k & 1 & -\lambda_k & 0 & 0 & 0 & \cdots \\ 0 & \lambda_k & 4 & -\lambda_k & 0 & 0 & \cdots \\ 0 & 0 & \lambda_k & 6 & -\lambda_k & 0 & \cdots \\ \vdots & \vdots & & & & & \\ 0 & 0 & \cdots & \lambda_k & 2M-4 & -\lambda_k & 0 \\ 0 & 0 & \cdots & 0 & \lambda_k & 2M-4 & -\lambda_k \\ 0 & 0 & \cdots & 0 & 0 & \lambda_k & 2M \end{pmatrix}. \tag{14}$$

Next define $\mathbf{C}$ as $\mathbf{C} = \mathbf{T}^{-1}\mathbf{A} - \mathbf{I}$ and also $x_0 = \mathbf{T}^{-1} b$. Now let $\mathbf{S} = -\mathbf{C}(\mathbf{I} + \mathbf{C})^{-1}$ and denote the fourth and last columns of the matrix $\mathbf{S}$ by S_4 and S_M, respectively. Finally, we need a vector v^{T} defined by

$$[0\ (1-\lambda_k)\ -\tfrac{1}{2}\ 1\ -1\ 1 \cdots (-1)^{M-1}\ 0]. \tag{15}$$

In terms of these quantities, the exact solution of $\mathbf{A}x = b$ may be written as

$$x = x_0 + (v^{\mathrm{T}} \cdot x_0)\, S_4 + x_0(M)\, S_M - (x_0(3)/2)\, e_1, \tag{16}$$

where e_1 denotes an M-component unit vector with first component 1 and $x_0(k)$ the kth component of the vector x_0. The key to this algorithm is that the matrix $\mathbf{C}$, as well as its iterates, are all rank two. This fact is what permits us to find an explicit solution in the form above.

To employ this result, for each (complex) eigenvalue, λ_k, we precompute and store S_4 and S_M as well as two vectors of constants arising from Gaussian elimination without pivoting in the inversion of $\mathbf{T}$. The number of eigenvalues is the product of the spatial resolution multiplied by the allowed number of halvings and doublings of the nominal time step. In practice this memory requirement is moderate and the overhead for the computation of S_4 and S_M is slight, but an incremental computational saving is possible, since these computations are performed only as the indicated step size is invoked for the first time. Assuming the values of S_4 and S_M are stored in advance, the present algorithm (16) executes in virtually the same time as direct elimination with back substitution and has, moreover, the virtue of producing accuracy indistinguishable from that of the IMSL routine DLSARG which uses Gaussian elimination with pivoting and subsequent iterative refinement. Of the total CPU time per step required in solution of (5) on a Cray Y-MP, less than 8% is required for inversion of $\mathbf{A}$—nearly all the remaining 92% is spent on the convolution uu_x.

When λ_k is large, the accurate direct solution of (13) noted above is effected by permuting the rows by one to bring the first into the last position. The first $M-1$ entries in this row are then eliminated and the solution found after back substitution. This approach fails for moderate λ_k with the useful cutoff depending upon the dimension of $\mathbf{A}$.

For example, for $M=9$, the direct solution yields relative errors less than 10^{-10} only for $|\lambda_k|>1$, while at $M=17$, a reasonable cutoff is $|\lambda_k|>10$. At small values of λ_k, a simple and useful iterative strategy is based on the splitting $\mathbf{A}=\mathbf{M}+\mathbf{N}$, where $\mathbf{N}$ contains all the terms proportional to λ_k. With extensive iteration this will converge for $|\lambda_k|$ as large as 0.5, but in practical terms is best restricted to values of 0.05 or smaller. (An asymptotic estimate of the spectral radius of the iteration matrix for small $|\lambda_k|$ correctly anticipates an interesting variation of radius with phase angle in the complex plane, but space does not permit discussion here.) For the class of problems considered in this paper, neither the small nor the large λ_k algorithm confers any advantage over (16) but, as we note in the conclusion, for two-dimensional problems the memory requirements of (16) may become excessive at high spatial resolution. In that instance, a hybrid algorithm using direct solution for large λ_k, (16) at moderate values, and splitting with iteration at small λ_k would seem to optimize performance for fixed memory.

III. APPLICATION TO THE KURAMOTO–SIVASHINSKY PROBLEM

A test problem for the Kuramoto–Sivashinsky problem is provided by reference to the work of [9]. Our equation is the differentiated form of theirs, thus as an initial condition to reproduce the modal amplitudes of their Figs. 26 and 27, we set

$$u(x,0)=-18.3\sin 6x+0.1\cos x \tag{17}$$

and choose $\nu=1$, $\mu=0$, and $\lambda=4/\alpha$ (where α is taken from their example and has a value of 206.25). The domain is $[-\pi,\pi]$.

At first glance it might appear rather surprising that the most demanding portion the numerical evolution of this problem is the vicinity of $t=0$, a fact not at all apparent from examination of Fig. 1. A way of understanding this is that the nonlinear time dependent problem accumulates a statistical distribution parameterized by time of the spatial spectra. Almost none of the members of this ensemble have all their energy content in the first few harmonics.

For the purpose of comparison to establish timing figures, a finite difference code based upon repeated Romberg extrapolation to a maximum order of 14 was implemented as likely being the most efficient finite difference based solution method for very high accuracy solutions. (The standard IMSL package DIVPBS was hopelessly inefficient for this purpose.) This finite difference method tends to produce a solution with local error everywhere approximately equal to the specified tolerance while the spectral method, as earlier noted, achieves its tolerance only rarely ($\Delta=\Delta_S$) and is normally much better;

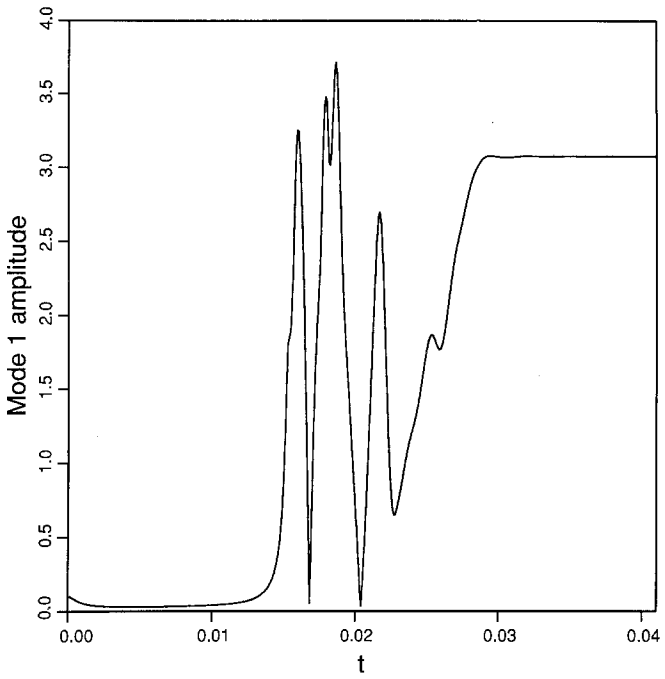

FIG. 1. Mode one amplitude in the evolution of (2) from the initial condition given in (17). This is to be compared with [9, Fig. 26].

thus a local tolerance for Romberg integration which was a factor of approximately 20 smaller than that given for the spectral integrator was required to produce a solution at the end of the test integration period ($t=0.24\overline{24}$) of comparable spatial error. Using a tolerance of 10^{-10} for the spectral integration resulted in a solution at the final time with error evidently bounded by 3.7×10^{-7} in a time of 24 s on the UCSD Cray Y-MP. Similar accuracy from the Romberg scheme required about 200 s. This disparity was quite comparable to the factor of 10 found in timings of both algorithms on a Sun 4/260.

Error curves from a number of numerical solutions for varying tolerance, spatial resolution, and machine are illustrated in Fig. 2.[3] Our measure of error is the common logarithm of the modulus of the difference between a given approximate solution for the mode one amplitude (e.g., Fig. 1) and a higher precision solution regarded as "exact" by comparison. For a problem such as this where we have no a priori knowledge of the correct solution, it is normally the case that our only indication of the accuracy of the results is the internal consistency of successive computations. Figure 2 shows such a pattern. For our "exact" solution we use the result of integration from a Sun 4/260 with 47 modes in x and a local tolerance of 10^{-13}. The curve labeled "1" is, in fact, a superposition of two curves as may be seen at large t where they diverge. This pair of runs shows

[3] In this section M is held fixed at 17. In the next section we explore in more systematic fashion the convergence in M.

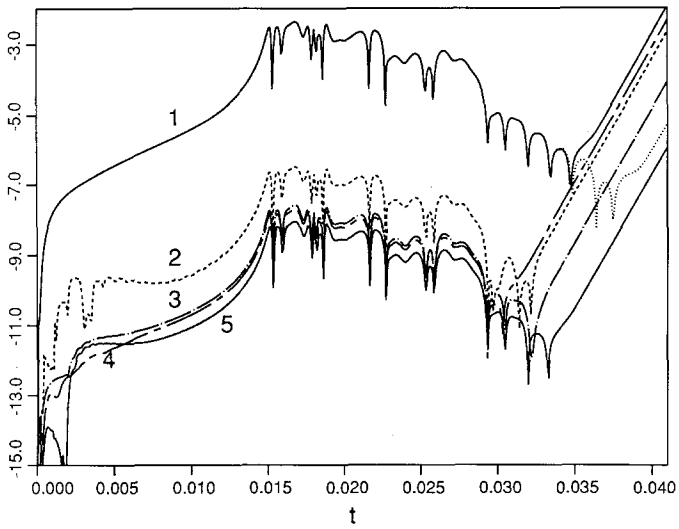

FIG. 2. Variations in accuracy due to varying tolerance, spatial resolution, and machine. (Vertical axis in units of $\log_{10}$ of the absolute value of the error in the amplitude of mode one.)

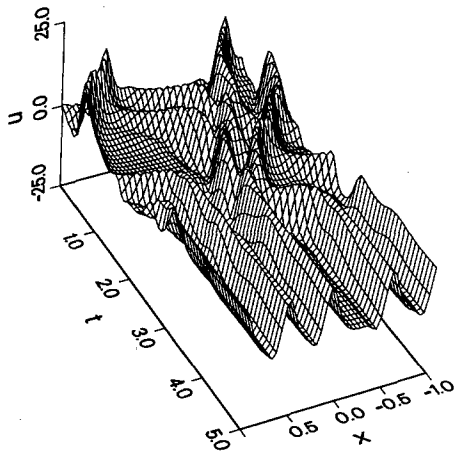

FIG. 3. Typical scattering experiment, two particles to four, in the solution of (2) for the parameters (ν, μ, λ) equal to (2, 1, 1). The domain is $[-15, 15]$. Note the far more regular solution away from the chaotic realm of Kuramoto and Sivashinsky. Far larger time steps are possible making high accuracy solutions very cheap to compute.

the evolution of a 31-mode model run on the Cray, and for comparison to see the effect on roundoff error, a Sun 4/260. The 4-bit longer mantissa of the latter gives a perceptible improvement in roundoff error. The explicitly allowed local error of 10^{-10} leads to perceptible phase error by about $t = 0.033$ (while the amplitude error, up until that time, is approximately constant), but the subsequent deviation of the pair of curves only slightly later at $t = 0.035$ suggests that even (Cray) roundoff perturbations on the order of 10^{-14} have begun to make their influence felt, so that the autonomous *finite order* system (5) has, for practical purposes, become numerically unpredictable. Curve "2," (a Cray run, as are curves 3 and 4) shows the effect of increasing the spatial resolution to $N = 47$ to be principally an amplitude adjustment. In curve "3," the local tolerance is decreased to 10^{-12}, producing approximately an order of magnitude improvement in the solution, but a further decrease to 10^{-13} shows from curve "4" that there is essentially no gain in accuracy. Further accuracy is precluded by roundoff error. This is also suggested by the consistent offset from curve "3" of curve "5", run on a Sun 4/260 at a tolerance of 10^{-12}. We conclude from these results that the drop in the amplitude of mode one from [9] which commences in their figure at about $t = 0.035$ is actually deferred until some time after 0.041. Trying to fix this time with precision greater than ~ 0.005 appears not to be possible with double precision arithmetic (Cray single precision).

Before passing to the next section, it is worth noting that evolution of (2) for generic values of the parameters leads to far more regular behavior. A sample run using the parameters (ν, μ, λ) equal to (2, 1, 1) is presented in Fig. 3. The sharply localized solitary waves evolve in an interesting fashion, which is the subject of current study. Fortunately the numerical exploration is computationally much cheaper to conduct as large time steps are possible. Away from the Kuramoto–Sivashinsky limit, it is important that such computations be done with an appropriate choice of the phase speed, c. Failing to do the computation in a moving reference frame otherwise unduly restricts the time step. The value of the phase speed can be extracted quite accurately from the evolution of the complex phase in any of the first few Fourier modes in the expansion of u and used to continually update the reference frame as the calculation proceeds.

IV. APPLICATION TO BURGERS EQUATION

In [10], Burgers equation is solved for $|x| \leqslant 1$, $t > 0$, with boundary conditions $u(\pm 1, t) = 0$ and an initial condition of $u(x, 0) = -\sin \pi x$.[4] The viscosity is fixed at $\nu = 10^{-2}/\pi$. Evolution is followed in most of the numerical experiments to a time of $3/\pi$. The purpose of that paper is to compare spectral methods in space with finite difference methods for a problem, as the above, in which regions of sharp variation occur. The x-t solution surface (obtained here by a Chebyshev series in both variables) for $0 \leqslant x \leqslant 1$ and $0 \leqslant t \leqslant 1$ is shown in Fig. 4. (The solution is antisymmetric on $[-1, 1]$.)

[4] Somewhere between our initial study of Burgers equation and the detailed computations for this work, the minus sign of the initial condition given in [10] was lost. Since the sign change is inconsequential and Fig. 4 seems marginally more esthetic plotted as a hill than a valley, the reader may simply invert either this article or [10] in comparing the two. Our thanks to a sharp-eyed referee for spotting the discrepancy.

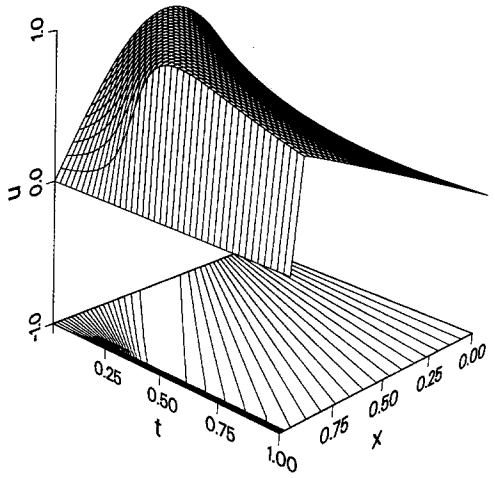

FIG. 4. Solution to Burgers equation with $\nu = 10^{-2}/\pi$ for $0 \leqslant t \leqslant 1$ and $\sin \pi x$ initial condition.

The results for the L_2 error defined by

$$\frac{1}{2T}\int_{-1}^{1} dx \int_{0}^{T} dt\,(u(x,t)-\bar{u}(x,t))^2 \tag{18}$$

(where $\bar{u}$ is the exact solution) are shown in Figs. 5 and 6 for a sequence of experiments as the above, except for $\nu = 0.1$ and $T = 1$. In the first of these, the spectral order of each element in time is four. (Spatial resolution was fixed at sufficiently high order so as to make a negligible contribution to the error.) The solution was computed for a number of intervals ranging from 1 to 128 and exhibits fourth-order accuracy. The second computation used a single time step,

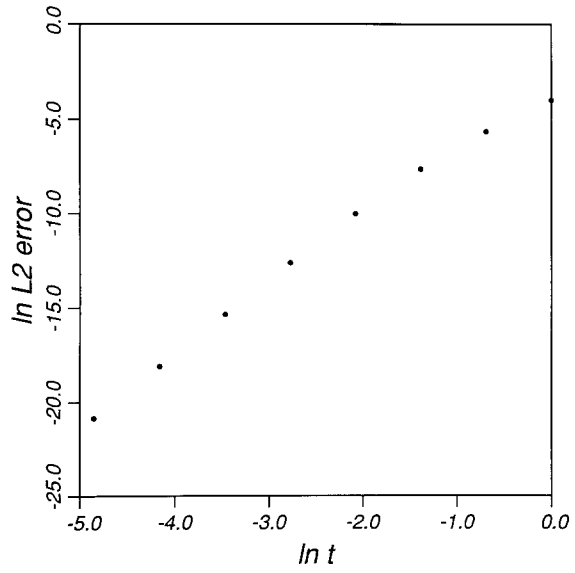

FIG. 5. The L_2 error as defined in the text shown as a function of decreasing time step size. Note the algebraic dependence shown clearly by the log–log plot.

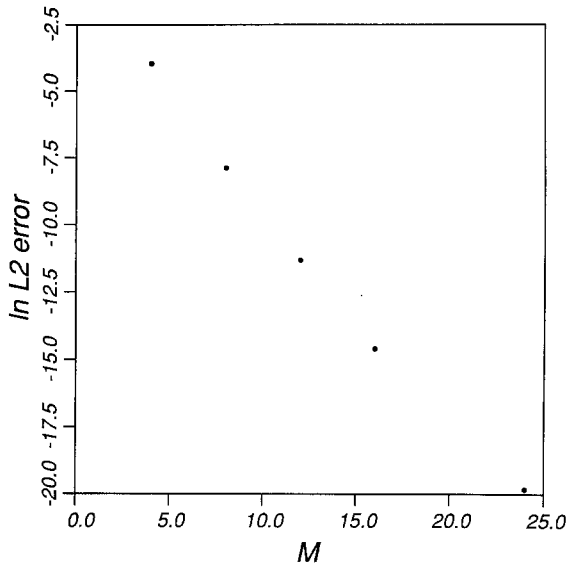

FIG. 6. The L_2 error as defined in the text shown as a function of increasing order of the basis set. The semi-log plot suggests an exponential dependence of the error. (Departure of the last point reflects slight influence of spatial error.)

with order varied from 4 to 24. Figure 6 displays the resulting expected exponential convergence. Two representative x-t error surfaces are shown in Figs. 7 and 8. The occurrence of interior grid scale high frequency ripple in such plots usually indicates which variable is underresolved. Figure 7 shows underresolution of t, while in Fig. 8, error is limited by spatial resolution. The exact solution given in [10] for a $\sin \pi x$ initial condition appears in two forms: a series, (2.4), useful for $\nu > 0.05$ and an integral representation, (2.5), useful for $\nu \leqslant 0.05$. Reference [10] uses Hermite integration to generate results accurate to seven digits. For the present work requiring higher precision, 48-point Hermite integra-

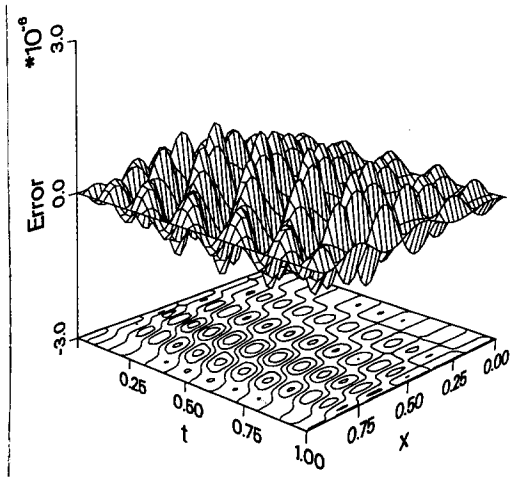

FIG. 7. The error in the computed spectral solution corresponding to the fourth data point in Fig. 6, with $M = 16$. Note the higher frequency in the t direction showing the error to be dominated by inadequate resolution in t.

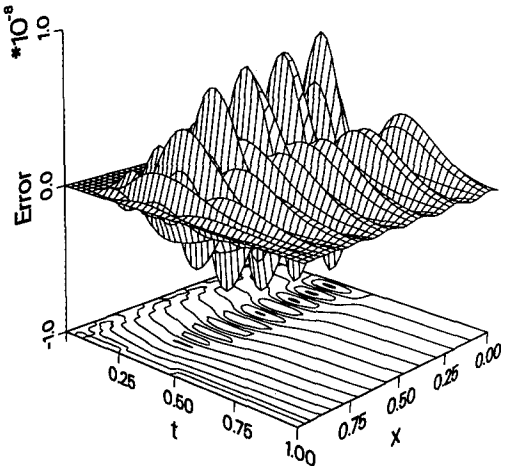

FIG. 8. The error in the computed spectral solution corresponding to Fig. 6, for $M = 32$ (not plotted). Now the error is dominated by inadequate spatial resolution.

tion was satisfactory for runs at small viscosity but accuracy commensurate with the limiting precision that is possible with the use of spectral methods does not appear to be achievable with higher order Hermite integration of (2.5), owing to accumulation of roundoff error.

To adapt the discussion from Section II, here we have one added element that is required for the solution, explicit matrix diagonalization in x. This stems from the use of Chebyshev polynomials in x for which the derivative matrix operator, as we have seen in II, is upper triangular. As first employed for Chebyshev polynomials in the solution of the Poisson equation by [12], the linear spatial operator is decomposed as

$$e\Lambda e^{-1} \tag{19}$$

where e is the eigenvector matrix, Λ is a diagonal matrix with entries $\lambda_1, \lambda_2, ..., \lambda_N$, and e^{-1} is the inverse of the eigenvector matrix. (For the Fourier case, $e = I$.) Here we incur the added overhead associated with casting left- and right-hand sides of (9) into the transform space by application of the matrix e^{-1}. This adds $\mathcal{O}(MN^2)$ operations to each iteration of (9). In contrast, periodic boundary conditions coupled with a Fourier representation permit marked efficiency of solution. For a vector machine, this operational distinction is not a crucial, since the matrix multiplications are typically fully vectorizable.

V. DISCUSSION AND CONCLUSIONS

A more ambitious approach for the problem of spectral methods in time is that of approximate factorization discussed in [13]. Those results show clear promise, but more experience of that and the present method is desirable to clarify the relative performance of each. Our experience is that for moderate viscosities, the present explicit treatment of advective terms can be quite effective. The limits of its utility are twofold. Application of Newton's method (8) in MN-dimensional space has the obvious restriction that the initial guess must fall within the basin of attraction of the desired solution. We leave the characterization of that to others and limit ourselves to simpler observation on the subiteration scheme (9), for which the origin of a time stepping restriction resides in the spectral radius of the iteration matrix for

$$\left(\frac{2}{\Delta t}\partial_\tau - c\partial_x + \nu\partial_x^2 + \mu\partial_x^3 + \lambda\partial_x^4\right)\delta u^{(n+1,m+1)} = \partial_x\left(u^{(n)}\delta u^{(n+1,m)}\right), \tag{20}$$

where Δt is adjusted to ensure convergence and accuracy with a fixed choice of M. A model problem to exhibit the potential difficulties of an explicit treatment of the advective term is the simpler hyperbolic problem[5]

$$\frac{2}{\Delta t}\partial_\tau u^{(n+1)} = \partial_x u^{(n)}. \tag{21}$$

(There is no loss of generality in normalizing the wave speed to one.) This scheme will converge when

$$\left|\frac{\Delta t\lambda_{\max}^{(x)}}{2\lambda_{\min}^{(\tau)}}\right| < 1, \tag{22}$$

where $\lambda_{\max}^{(x)}$ is the largest eigenvalue of the ∂_x operator, and $\lambda_{\min}^{(t)}$, the smallest modulus eigenvalue of the ∂_t operator after reduction to $(M-1)\times(M-1)$ to incorporate the implicit imposition of an initial condition on u (see, e.g., [12]). That eigenspectrum (which also characterizes the ∂_x operator when Chebyshev polynomials are used in x) is displayed in Fig. 9 for $M = 25$. Computation of $\lambda_{\min}^{(t)}$ is difficult for large M, becoming unreliable for $M > 35$ with double precision. Empirically, $\lambda_{\min}^{(t)} \sim c_M M$, where c_M is a constant somewhat greater than 2. The scale of the largest eigenvalue of the ∂_x operator depends upon the basis set. For the Fourier representation, though, it is simply N and thus (22) becomes

$$\frac{N\,\Delta t}{2c_M M} < 1. \tag{23}$$

[5] Note that [14] presents an extensive treatment of the nonconstant coefficient hyperbolic problem including detailed error estimates for comparison with finite difference methods. The infinite order accuracy of spectral methods is there shown to be the most efficient strategy to achieve a given resolution.

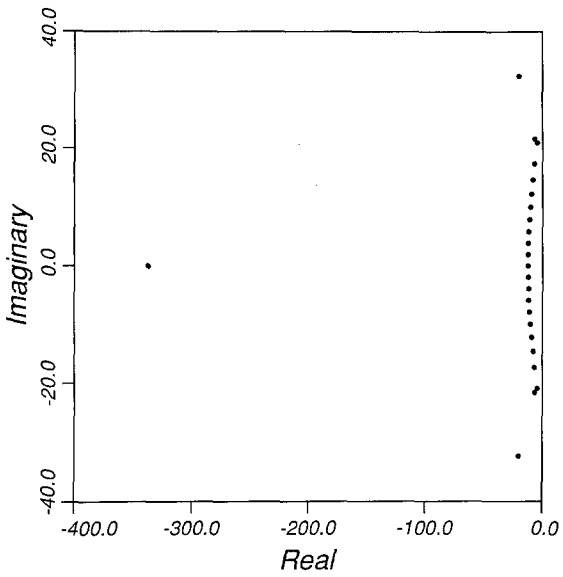

FIG. 9. Spectrum of the first derivative Chebyshev operator of order 25 which satisfies an arbitrary boundary condition on the function at one end. Note the disparity in vertical and horizontal scales.

Now $2\Delta t/M$ is the coarsest mesh size for the interior Chebyshev collocation point spacing for the element interval of length Δt, while N is inversely proportional to the Fourier (equal) grid spacing in x; thus the time stepping restriction for subiteration assumes a familiar form: $\Delta t < \alpha\, \Delta x$, with a coefficient of proportionality, α, near unity (for a problem with unit wave speed). The neglected terms on the left-hand side of (20) simply shift the spectrum of ∂_τ in the complex plane. For the diagonal Fourier basis in x, the shift is elementary. Broadly speaking, the added terms render the estimate in (23) too pessimistic.

When Chebyshev polynomials are used in x for the solution of (21), the largest spatial eigenvalue scales as N^2, reflecting the high boundary layer resolution. The spectrum in Fig. 9 is instructive in showing that this extremal eigenvalue is isolated. The remainder of the spectrum scales as N. The eigenfunction which accompanies this outlier is sharply boundary layered. At first glance, the N^2 dependence severely constrains the time stepping, a difficulty widely noted in the application of Chebyshev methods in space. Of course, the operator which concerns us in practice is not the linear one chosen here for simplicity, but the nonconstant coefficient operator. Sample computations show that often the pessimistic N^2 spectral range of the ∂_x operator is sharply reduced for the full operator. This comes about when the convolution operator $u^{(n)}$ yields small projection on the Nth row of the inverse eigenvector matrix e^{-1} in (19). This is not a rare occurrence.

The ratio above in (22) could be improved by applying a shift to the operators on both sides. As a practical matter, however, overly large shifts lead to an iteration matrix all of whose eigenvalues are clustered near one, so the convergence rate tends to zero. Of course more sophisticated dynamic acceleration methods such as nonstationary Chebyshev or related methods may warrant consideration, but the general experience of runs on (2) including the Kuramoto–Sivashinsky limit is that time stepping is constrained, not by divergence of either inner or outer iteration, but by the accuracy requirement. For *low* accuracy solutions, this would not continue to be the case.

We have done some preliminary experiments, applying spectral methods in time to the unforced two-dimensional vorticity equation,

$$\omega_t + J(\psi, \omega) = \nu \nabla^2 \omega, \qquad \text{where} \quad \omega = \nabla^2 \psi \tag{24}$$

with doubly periodic boundary conditions at low resolution. The obvious parallel of (9), namely,

$$\begin{aligned} D_{j,m}\delta\psi_{m,k,l}^{(n+1,p+1)} + \nu(k^2+l^2)\,\delta\psi_{j,k,l}^{(n+1,p+1)} \\ = -(k^2+l^2)^{-1}\,[\mathscr{L}(\psi^{(n)}) + J(\psi^{(n)}, \delta\omega^{(n+1,p)}) \\ + J(\delta\psi^{(n+1,p)}, \omega^{(n)}]_{j,k,l} \end{aligned} \tag{25}$$

(where k and l are the x and y wavenumbers, respectively) gives iterative results generally in accord with the one-dimensional findings when used in our initial experiments on the evolution of a localized circularly symmetric vortex of Gaussian profile. We shall report on this at greater length in a subsequent paper, but one point we note in passing is that just as the evolution shown in Fig. 1 masks the great difficulty in accurately resolving the initial transient; so too for the two-dimensional problem with a circularly symmetric initial condition embedded in a lattice of discrete symmetry do we find a sensitive initial phase of adjustment.

A potential drawback of the application of (16) is the memory requirement for S_4 and S_M, which here runs to $16MN_xN_yt_d$ bytes, where t_d is the allowed number of doublings and halvings of the nominal time step. The hybrid approach sketched at the end of Section II offers one resolution to the excessive memory needed at high spatial resolution.

ACKNOWLEDGMENTS

One of us (GI) wishes to thank the Office of Naval Research for its continuing support of these efforts under Contract N00014-90-J-1201 with Scripps Institution of Oceanography. We are particularly grateful to Clark R. Givens for his generous and expert counsel which has so greatly enlightened and inspired all our efforts in linear algebra, both large and small.

REFERENCES

1. R. G. Voight, D. Gottlieb, and M. Y. Hussaini (Eds.), *Spectral Methods for Partial Differential Equations* (SIAM, Philadelphia, 1984).

2. A. Patera, *J. Comput. Phys.* **54**, 468 (1984).
3. D. J. Benney, *J. Math. Phys.* **45**, 150 (1966).
4. T. Kawahara, *Phys. Rev. Lett.* **51**, 381 (1983).
5. S. Toh and T. Kawahara, *J. Phys. Soc. Jpn* **54**, 1257 (1985).
6. T. Kawahara and M. Takaoka, *Physica D* **39**, 43 (1989).
7. C. Elphick, G. Ierley, O. Regev, and E. A. Spiegel, *Phys. Rev. A* **44**, 1110 (1991).
8. P. J. Channell and C. Scovel, *Nonlinearity* **3**, 231 (1990).
9. J. M. Hyman, B. Nicolaenko, and S. Zaleski, *Physica D* **23**, 265 (1986).
10. C. Basdevant, M. Deville, P. Haldenwang, J. M. Lacroix, J. Ouzzani, R. Peyret, P. Orlandi, and A. T. Patera, *Comput. Fluids* **14**, 23 (1986).
11. D. Gottlieb and S. Orszag, *Numerical Analysis of Spectral Methods* (SIAM, Philadelphia, 1977).
12. D. Haidvogel and T. Zang, *J. Comput. Phys.* **30**, 167 (1979).
13. Y. Morchoisne, *Spectral Methods for Partial Differential Equations* edited by R. G. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM, Philadelphia, 1984), p. 240.
14. H. Tal-Ezer, *SIAM J. Numer. Anal.* **23**, 11 (1986).