Conf-9409241--6

LA-UR -94-4035

NONLINEAR DIFFERENCE APPROXIMATIONS FOR EVOLUTIONARY PDES

Donald A. Jones

Len G. Margolin

Andrew C. Poje

Institute for Geophysics and Planetary Physics
Los Alamos National Laboratory
Los Alamos, NM 87545

Photograph: by Chris J. Lindberg

NONLINEAR DIFFERENCE APPROXIMATIONS FOR EVOLUTIONARY PDES

DONALD A. JONES
LEN G. MARGOLIN
and
ANDREW C. POJE

Institute for Geophysics and Planetary Physics
Los Alamos National Laboratory
Los Alamos, NM 87545

ABSTRACT

We describe a procedure to improve both the accuracy and computational efficiency of finite difference schemes used to simulate nonlinear PDEs. The underlying idea is that of enslaving, which is the estimation of the small unresolved scales in terms of the larger resolved scales. We discuss details of the procedure and illustrate them in the context of the forced Burgers' equation in one dimension. We present computational examples that demonstrate the predicted increases in accuracy and efficiency.

1. Introduction

In this paper we describe a procedure for improving both the accuracy and the computational efficiency of finite difference schemes that are used to solve nonlinear dissipative PDEs. The increase in accuracy, in theory, is the result of estimating the effects of the small scales that are unresolved on the mesh in terms of the larger resolved scales. The modified scheme is actually constructed to approximate the solution that the original scheme would generate on a finer mesh. Although the new scheme is more expensive to compute, we have found that this is more than compensated by the increase in accuracy,

Our procedure depends on the nonlinearity of the equations and is similar to that used in deriving nonlinear Galerkin schemes[1,2,3]. In particular, both procedures construct an enslaving relation -- a diagnostic relation between the unresolved small scales and the larger resolved scales. Two differences are that we do not assume the solution has an inertial manifold, and that we are using finite difference methods rather than expansion in orthogonal polynomials. The theory of the approximate inertial manifold, upon nonlinear Galerkin schemes are based, provides the closure assumptions that lead to the enslaving relation. Our closure is similar, but is justified by the assumption that the time derivative is not

part of the principal balance of terms; thus our method is designed for flows that slowly evolve or that exhibit small scale fluctuations about a larger scale steady state.

The construction of enslaving relations in the context of finite difference schemes is not at all straightforward. In a previous paper[4] (hereafter MJ) we addressed some of these difficulties, including the identification of the small scale modes and the derivation of the enslaving relations. We applied our theory to the one-dimensional Burgers' equation, and showed a significant increase in accuracy when compared to the unimproved scheme. However the approach outlined in that paper has several deficiencies. In particular, the construction procedure for the enslaving does not easily generalize to multiple dimensions, especially with irregular boundaries. In addition, we restricted our attention to the solution in steady state and so considered the spatial truncation error but not the error due to time differencing. Finally, even in one dimension, we found difficulties using Dirichlet boundary conditions.

In a more recent paper[5] we have generalized our procedure to remedy these problems. The main thrust of this paper was to establish a more mathematical foundation for our theory, as well as to provide rigorous error estimates. Although this paper included several computational examples, the discussion of practical details necessary to derive the enslaved schemes was very brief. Our principal purpose in this paper is to provide a fuller description of those details.

2. General Description

We consider the numerical solution of a PDE representing the evolution of a nonlinear dissipative dynamical system. We suppose we are given a discretized finite-difference equation that approximates the PDE to some order of accuracy. Our goal is to modify this discretization so as to produce solutions of higher accuracy. Of course, one can always improve the accuracy of a well-posed numerical scheme by increasing the resolution. Thus, an important aspect of our modification is that the new equations should be computationally more efficient -- i.e., the modified scheme should require less CPU than the original scheme to produce a solution of a specified level of accuracy.

Our strategy is to derive a modified scheme on a coarse mesh that reproduces the accuracy of the original scheme on a twice-finer mesh. It is important to note that modifications based on this strategy, even if completely successful, cannot improve the order of accuracy. That is, the error of the original scheme on the finer mesh has the same dependence on the derivatives of the solution, but with smaller coefficients. The implementation of this strategy follows the following steps.

1) We implement the original scheme on the twice-finer mesh. There are twice as many degrees of freedom on this fine mesh as there are on the coarse mesh.
2) We define a transformation of variables that maps the degrees of freedom of the fine mesh onto the coarse mesh. Half the variables on the coarse mesh represent the average solution in a cell -- the large scales -- and the other half represent smaller scales that will be unresolved on the coarse mesh.
3) We form the evolution equations for the new variables.
4) We apply a closure assumption to the new evolution equation for the small scales. This closure involves ignoring the time derivative, and yields a prognostic relation between the small scales and the large scales.
5) We solve the prognostic equation to express the small scales in terms of the larger scales. This is the enslaving relation.
6) We implement this relation in the new equation for the large scales. The result is our improved discretization.

We illustrate these steps with the forced Burgers' equation, which is a simple example of an equation that has both nonlinearity and dissipation. We emphasize that our arguments are more generally applicable. We employ the following spatial discretization:

$$\frac{\partial u_k}{\partial t} + \frac{u_{k+1}^2 - u_{k-1}^2}{2\,\delta x} - \lambda \frac{u_{k+1} - 2\,u_k + u_{k-1}}{\delta x^2} = f_k \qquad (1)$$

Equation (1) does not specify the method of time integration. We will return to this issue in section 8. Here λ is the coefficient of viscosity and f is a force that is known, but may vary in space and time. The cell size is δx, which characterizes the coarse mesh -- the mesh on which we will compute. We assume that there are N real cells on this mesh, so that the subscripts k=1,N refer to spatial index. The particular differencing of the nonlinear (advective) term is neither unique nor optimal, but is second-order accurate in space and is chosen just for illustration.

3. Coordinate Transformations

As our first step, we consider the scheme (1) applied to a twice-finer mesh, whose cells are $\delta x/2$ in length. On this fine mesh, there are approximately 2N cells (depending on the precise relation between the two meshes) and so 2N degrees of freedom. We want to identify N degrees of freedom with the solution

on the coarse mesh, and the other N degrees of freedom with the smaller, unresolved scales. To facilitate this, we write (1) as a coupled system of two variables (α,β) where, say, α is the solution in the odd-numbered cells and β is the solution in the even-numbered cells.

$$\begin{aligned}\frac{\partial \alpha_k}{\partial t} &= -\frac{\beta^2_{k+1/2} - \beta^2_{k-1/2}}{2\,\delta x} + 4\lambda\frac{\beta_{k+1/2} + \beta_{k-1/2} - 2\,\alpha_k}{\delta x^2} + f_k \\ \frac{\partial \beta_{k+1/2}}{\partial t} &= -\frac{\alpha^2_{k+1} - \alpha^2_k}{2\,\delta x} + 4\lambda\frac{\alpha_{k+1} + \alpha_k - 2\,\beta_{k+1/2}}{\delta x^2} + f_{k+1/2}\end{aligned} \tag{2}$$

The next step is to define a transformation of variables $a=a(\alpha,\beta)$ and $b=b(\alpha,\beta)$ such that a represents the larger scales and b represents the smaller scales. In MJ we derived this transformation geometrically by introducing basis sets on the coarse and the fine meshes. This was a useful but unnecessary step, and here we will proceed more directly.

The important aspect of the transformation derived in MJ is that $a\sim O(a)$ whereas $b\sim O(\partial a/\partial x\ \delta x)$. The particular transformation used in MJ is

$$\begin{aligned}a_k &\equiv \frac{\alpha_k + \beta_{k+1/2}}{2} \\ b_k &\equiv \frac{\beta_{k+1/2} - \alpha_k}{2}\end{aligned} \tag{3}$$

One important virtue of the transformation (3) is that it has an exact inverse, which facilitates the derivation of the new equations. However (3) also has a disadvantage for Dirichlet boundary conditions. In effect, (3) describes each pair of fine cells as the left and right halves of a coarse cell. Since we conceive the variables as being located at the cell-centers, none of the variables on the fine mesh are coincident with variables on the coarse mesh. This implies that boundary conditions cannot be specified at exactly the same points on the coarse and fine meshes, and that some averaging is necessary to apply Dirichlet conditions, with consequent loss of accuracy.

This geometric interpretation suggests that an alternate transformation to (3) might be more suitable. Instead of defining a symmetrically in terms of (α,β), we identify a primarily with either α or β. In addition, we want to identify b with the smallest stencil estimating the difference Δa. A general form for a linear transformation with these properties is

$$a_k \equiv A\,\alpha_k + \frac{(1-A)}{2}\left(\beta_{k+1/2} + \beta_{k-1/2}\right)$$
$$b_k \equiv \frac{\beta_{k+1/2} - \beta_{k-1/2}}{4} \tag{4}$$

where A is a centering parameter. The factor 4 in the bottom of the b-equation is chosen for convenience. Equation (4) corresponds to the relationship where the centers of the a-cells of the coarse mesh, and the α-cells of the fine mesh coincide.

The transformation (4) has a disadvantage that makes it more difficult to use -- equation (4) has no exact local inverse. However, since (3) is only second-order accurate in space, we expect that an approximate inverse would work if it is sufficiently accurate. It is not difficult to show that a second-order accurate inverse of (4) can be written as

$$\alpha_k \equiv \left(\frac{5-A}{4}\right) a_k + \frac{(A-1)}{8}\left(a_{k+1} + a_{k-1}\right)$$
$$\beta_{k+1/2} \equiv \frac{a_{k+1} + a_k}{2} - \left(\frac{2-A}{2}\right)\left(b_{k+1} - b_k\right) \tag{5}$$

Equation (5) is not unique, but does have the narrowest stencil -- i.e., is the most local inverse to (4) -- that is second-order accurate. We emphasize that (4) and (5) have been derived independently of any particular PDE.

4. The Transformed Equations

The next step is to form the equations that govern the evolution of the variables a and b. These are combinations of α and β equations, with the coefficients given by (4):

$$\frac{\partial a_k}{\partial t} \equiv A\frac{\partial \alpha_k}{\partial t} + \frac{(1-A)}{2}\left(\frac{\partial \beta_{k+1/2}}{\partial t} + \frac{\partial \beta_{k-1/2}}{\partial t}\right)$$
$$\frac{\partial b_k}{\partial t} \equiv \frac{1}{4}\left(\frac{\partial \beta_{k+1/2}}{\partial t} - \frac{\partial \beta_{k-1/2}}{\partial t}\right) \tag{6}$$

We substitute for the derivatives on the right-hand-side of the equations using (2), which leads to equations for a and b, but still written in terms of α and β. We can then use (5) to eliminate α and β in terms of a and b. This procedure is

straightforward, but leads to a very complex set of equations. In the next sections we will discuss several aspects of these equations, which will lead to considerable simplifications both in their appearance and their application.

5. Compatibility Relations

We begin by writing the new equations in symbolic form

$$\begin{aligned} \frac{\partial a_k}{\partial t} &= - N_a(a_k) + \frac{4\lambda}{\delta x^2} D_a(a_k) + F_a(f_k) \\ \frac{\partial b_k}{\partial t} &= - N_b(b_k) + \frac{4\lambda}{\delta x^2} D_b(b_k) + F_b(f_k) \end{aligned} \tag{7}$$

Here N_a and N_b are the nonlinear terms, which we do not write out yet. The force functions are:

$$\begin{aligned} F_a(f_k) &\equiv A f_k + \frac{(1-A)}{2} (f_{k+1/2} + f_{k-1/2}) \\ F_b(f_k) &\equiv \frac{1}{4} (f_{k+1/2} - f_{k-1/2}) \end{aligned} \tag{8}$$

The dissipative terms are:

$$\begin{aligned} D_a(a_k) \equiv \frac{A(3-A)}{4} (a_{k+1} - 2a_k + a_{k-1}) + \frac{(2-A)(1-2A)}{2} (b_{k+1} - b_{k-1}) \\ - \frac{(A-1)^2}{16} (a_{k+2} - a_{k+1} - a_{k-1} + a_{k-2})) \end{aligned} \tag{9}$$

and

$$\begin{aligned} D_b(b_k) \equiv (A-1) (a_{k+2} - 2a_{k+1} + 2a_{k-1} - a_{k-2}) \\ + (2-A) (b_{k+1} - 2b_k + b_{k-1}) \end{aligned} \tag{10}$$

Now to derive the enslaving, we set the time derivative in the b-equation (7b) to zero:

$$N_b(b_k) = \frac{4\lambda}{\delta x^2} D_b(b_k) + F_b(f) \qquad (11)$$

and attempt to solve (11) algebraically for b_k. Independent of N_b, the form of D_b immediately presents a problem -- the variable b appears as a second difference in (10), which is not easily invertible. In MJ, we used truncation analysis to find an ODE that we solved for the enslaving. This solution required two constants of integration, whose estimation degraded the accuracy of our enslaving. Furthermore, the ODE technique is much more difficult to apply in multiple dimensions, especially in a domain with irregular boundaries. Here we introduce an alternative that is both simpler and more accurate.

We derive a set of subsidiary relations by composing (4) with (5). This yields

$$b_k \approx \frac{a_{k+1} - a_{k-1}}{8} - \frac{2 - A}{8}(b_{k+1} - 2b_k + b_{k-1}) \qquad (12a)$$

and

$$b_{k+1} - b_{k-1} \approx \frac{a_{k+1} - 2a_k + a_{k-1}}{2} \qquad (12b)$$

These are approximate identities, which we have termed compatibility relations. Substituting the second compatibility relation (12b) in the diffusional part of the b-equation, (10), yields an expression diagonal in b -- that is, containing only b_k. This considerably simplifies the construction of the enslaving relation. Also, substituting the first compatibility relation (12a) into our expression for the diffusional part of the a-equation (9) leads immediately to an expression that is independent of b_k. This has some advantages in semi-implicit methods where the diffusion term is treated implicitly.

6. Compression Relations

Another feature that is evident in both diffusional terms (9) and (10) is that they involve terms with indices $k \pm 2$, implying that the stencil of the new scheme is wider than that of the original scheme. A wider stencil does not introduce any real difficulties. On the other hand, it is not an essential element of the theory and can be compressed. Consider for example $(a_{k+2} - 2a_k + a_{k-2})$. Ordinary truncation analysis shows

$$a_{k+2} - 2a_k + a_{k-2} = 4\frac{\partial^2 a}{\partial x^2}\delta x^2 + O(\delta x^4) \qquad (13)$$

Thus we may substitute

$$a_{k+2} - 2a_k + a_{k-2} \cong 4(a_{k+1} - 2a_k + a_{k-1}) \tag{14}$$

without degrading the accuracy of the scheme which is only of second order. Approximations such as (14) are termed compression relations.

There is a subtlety in using these compression relations. The point is most easily illustrated by considering Burgers' equation (1). Recalling our assumption that the time derivative is not part of the principal balance of terms, we can write

$$\frac{u_{k+1}^2 - u_{k-1}^2}{2\,\delta x} - \lambda \frac{u_{k+1} - 2u_k + u_{k-1}}{\delta x^2} \cong f_k \tag{15}$$

Let U be a typical velocity, and $L >> \delta x$ be a typical length scale of the problem. A simple scale analysis then indicates

$$\lambda \sim UL >> U\,\delta x \tag{16}$$

The point is that in a nondimensional sense, λ is a much larger constant than unity. In the same sense, third-order derivatives in the dissipative term are similar in magnitude to second-order derivatives in the nonlinear term and so cannot be discarded.

Equation (15) provides a useful approximation for more general compression relations. For example, we can write

$$\begin{aligned} a_{k+2} - 2a_{k+1} + a_k &\cong \frac{\delta x}{2\lambda}[a_{k+2}^2 - a_k^2] - \frac{\delta x^2}{\lambda} f_{k+1} \\ a_k - 2a_{k-1} + a_{k-2} &\cong \frac{\delta x}{2\lambda}[a_k^2 - a_{k-2}^2] - \frac{\delta x^2}{\lambda} f_{k-1} \end{aligned} \tag{17}$$

so that we can compress a term appearing in (10) for D_b:

$$\begin{aligned} a_{k+2} - 2a_{k+1} + 2a_{k-1} - a_{k-2} \cong{}& \frac{2\,\delta x}{\lambda}[a_{k+1}^2 - 2a_k^2 + a_{k-1}^2] \\ & - \frac{\delta x^2}{\lambda}[f_{k+1} - f_{k-1}] \end{aligned} \tag{18}$$

Note that (15) is derived from the original discretization (1), while the a-variable will be evolved by the modified discretization that we are deriving. Thus in writing (17), we make an additional small approximation beyond ignoring the time derivatives.

7. The Centering Parameter

When the various approximations described in the previous sections are applied to the general form (7) the resulting evolution equations take the form:

$$\frac{\partial a_k}{\partial t} = -\frac{1}{8\,\delta x}\left[a_{k+1}^2 - a_{k-1}^2 + 16\,a_k b_k\right] + \frac{1}{4\,\delta x}\left[2\,b_k(a_{k+1}+a_{k-1}) - (a_{k+1}b_{k+1} - a_{k-1}b_{k-1})\right] + \frac{\lambda}{\delta x^2}\left[a_{k+1} - 2a_k + a_{k-1}\right] + Af_k + \frac{1-A}{2}(f_{k+1/2} + f_{k-1/2}) \tag{19}$$

and

$$b_k = \frac{a_{k+1} - a_{k-1}}{8} - \frac{\delta x}{64\,\lambda}\left[a_{k+1}^2 - 2a_k^2 + a_{k-1}^2\right] + \frac{\delta x^2}{32\,\lambda}(f_{k+1/2} - f_{k-1/2}) \tag{20}$$

Note that a remarkable simplification has occurred -- the centering parameter A has dropped out of the equations except for the forcing term in the a-equation.

The enslaving relation (20) is already diagonal and requires no iteration. In terms of magnitude, the first term on the RHS of (20) is the largest. We have constructed b to estimate the first derivative of a, and so a first-order estimate turns out to be independent of the evolution equation -- in fact, really arises from the first compatibility relation (12a). The second term on the RHS represents a nonlinear diffusion, whereas the third term adds subgrid scale information about the spatial tendency of the forcing.

In the a-equation, the diffusion term remains unchanged. This is in contrast to our results in MJ where the diffusive term contains significant contributions from the b-variables. As we have noted in the previous section, the viscous coefficient λ is large in a dimensionless sense, and so we would prefer to make errors in the nonlinear term rather than the diffusion term. In particular, b is an auxiliary variable, and has no physical specification in the case of Dirichlet boundary conditions, introducing the possibility of error (see MJ for a discussion).

Finally we consider the choice of the centering parameter A. We expect the choice A = 0.5 to be the most accurate, and would be preferred when the force has significant variation on the scale of a cell. In some cases, the evaluation of the force can be expensive, and then it might be preferable to choose A=1.0 We note that the force is a known function of space and time. The incorporation of fine-scale information about the forcing in the enslaving can be an important source of the improvement in the accuracy of the solution.

8. Time Integration

The enslaving relation (20) is independent of the details of the time integration since it is derived by ignoring the time derivative. However the order of accuracy of the time integration can affect the formulation of our improved algorithm. We note that the maximum timestep permitted by (2), which is solved on the twice-finer mesh, is necessarily smaller than could be used on the coarse mesh. After the change of variables leading to (7), the system is still equivalent to (2) and so is also governed by the smaller timestep of the fine mesh. It is the closure, setting the time derivative of b to zero, that allows us to increase the timestep. Use of this larger timestep is necessary if the new algorithm is to be more efficient than the old.

It is important to remember that the algorithm (2) has both spatial errors and temporal errors. Our strategy has been to reduce the spatial errors of the coarse mesh to those of the fine mesh. However the larger timesteps permitted by our algorithm will certainly lead to larger temporal errors than would be present in the fine-mesh solution integrated with the smaller timestep. These errors may become noticeable as the magnitude of the time derivative increases, and indicate the need for a more accurate time integration scheme.

To illustrate this point, we present the following computational example. We consider the forced Burgers' equation on the interval $x \in [0,1]$, and with Dirichlet boundary conditions u(0)=u(1)=0. We choose an exact solution

$$u(x,t) = \sin(\pi x)\{1 + \varepsilon[x\cos(\omega t) + (1-x)\sin(\omega t)]\} \qquad (21)$$

where ε and ω are positive constants that measure the importance of the time variation. As these parameters are increased, the time derivative of the solution grows and the validity of our enslaving relation decreases. By substituting (21) into Burgers' equation, we can derive the force that generates this solution. For our example, we discretize this force, and use it to drive the enslaved system (19) and (20) as well as the original system (1), each using two different time-integration schemes. Finally, we measure the accuracy of each numerical

solution by comparing it with the exact solution in the L^∞ norm -- i.e., the maximum deviation of the computed nodal values from the true nodal values of the exact solution.

In figure 1, we show the error ratio of the original scheme [equation (2) to the enslaved second-order scheme [equations (19) and (20)], using forward Euler time integration. Each of the four panels contains three calculations with a distinct value of ω and three value of ε. In figure 2, we repeat these twelve calculations, but using a second-order accurate MacCormick integration scheme. The solid line on each plot of figures 1 and 2, at the error ratio $\approx$ 1.6 represents the breakeven level, below which the enslaved scheme becomes computationally less efficient than the original scheme. The total amount of time spent above (below) this line indicates whether the enslaved scheme is more (less) efficient than the original scheme. In figure 1, there are significant regions where the error ratio drops below the breakeven line. These regions become much smaller in figure 2 where the higher-order MacCormick scheme is used. Note that in both figures, the error ratio is greater than unity almost everywhere. These points are discussed in more detail in JMP. All calculations in figures 1 and 2 use the same mesh and the same timestep.

In figures 3 and 4 we illustrate the dependence of the accuracy on the order of the integration scheme. Both figures are graphs of the average error (integrated over a problem cycle) as a function of the timestep used. The numerical data in figure 3 was generated with a forward Euler scheme and shows the error of both the original and the enslave schemes increases significantly with timestep. The data in figure 4 was generated with a MacCormick scheme, and shows no sensitivity to the timestep. In both cases, the extreme values of $\omega=10\pi$ and $\varepsilon=10$. Note also the original and the enslaved schemes allow the same maximum timestep for both time integration schemes.

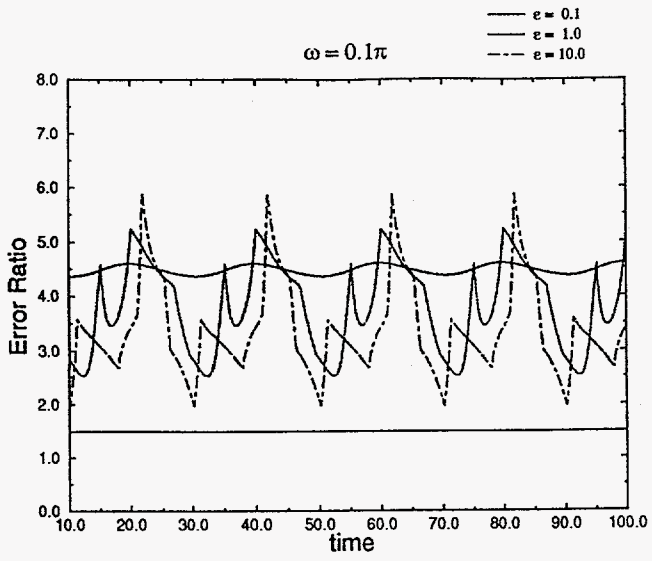
9. Acknowledgments

This work was supported in part by the U.S. Department of Energy's CHAMMP program. The authors also acknowledge the support of the Institute for Geophysics and Planetary Physics (IGPP) and the Center for Nonlinear Studies (CNLS) at Los Alamos National Laboratory.

10. References

1. D.A. Jones, L.G. Margolin, and E.S. Titi, On the Effectiveness of the Approximate Inertial Manifold, *Theoretical and Computational Fluid Dynamics* (to appear, 1995)

2. M.S. Jolly, I.G. Kevredidis, and E.S. Titi, *Physica*, **D44** (1990), 38-60.
3. E.S. Titi, *J. Math. Anal. Appl.*, **149** (1990), 540-557.
4. L.G. Margolin and D.A. Jones, *Physica* **D60** (1992), 175-184.
5. D.A. Jones, L.G. Margolin, and A.C. Poje, Enslaved Finite Difference Schemes for Nonlinear Dissipative PDEs, submitted.


ω = 0.1π
ε = 0.1
ε = 1.0
ε = 10.0
Error Ratio
time

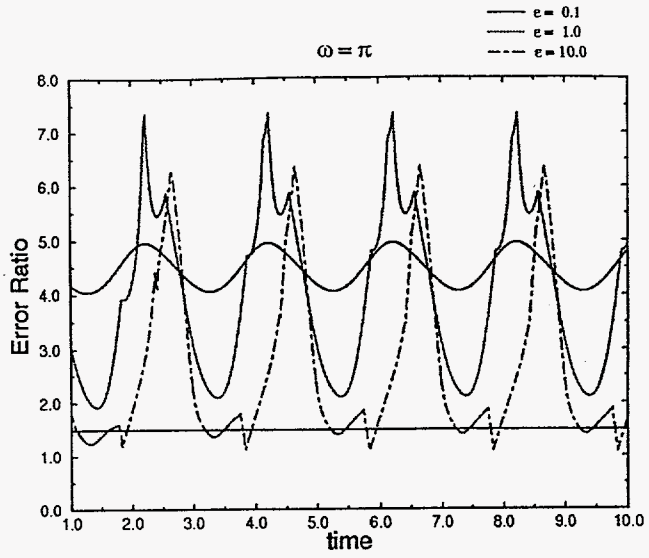
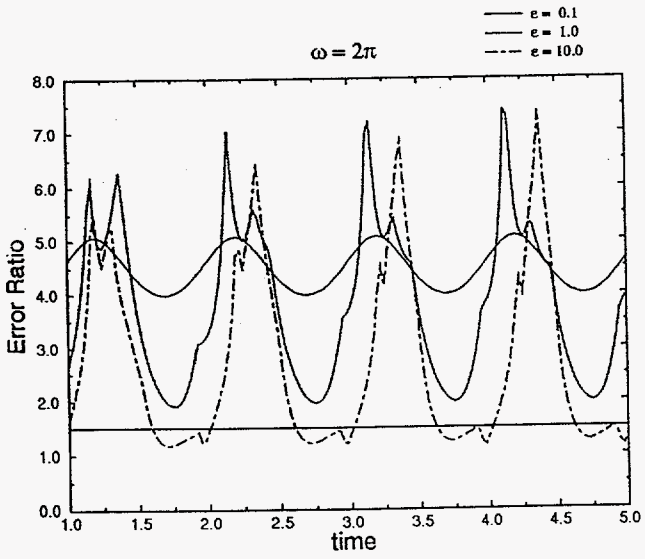

ω = π
ε = 0.1
ε = 1.0
ε = 10.0
Error Ratio
time

Fig. 1


ω = 2π
ε = 0.1
ε = 1.0
ε = 10.0
Error Ratio
time

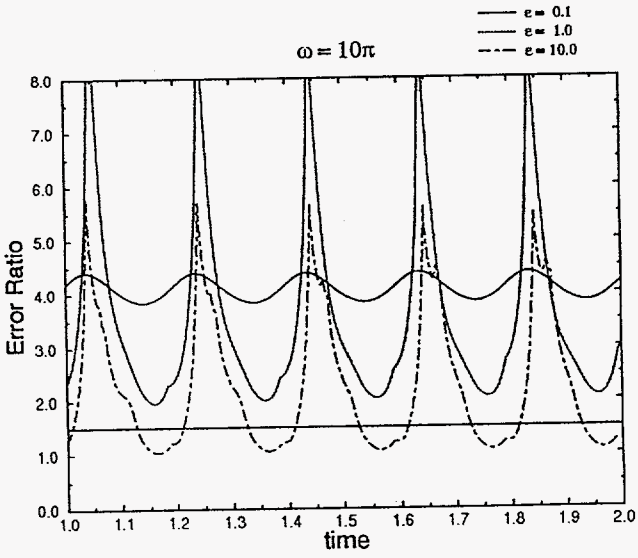

ω = 10π
ε = 0.1
ε = 1.0
ε = 10.0
Error Ratio
time

Fig2

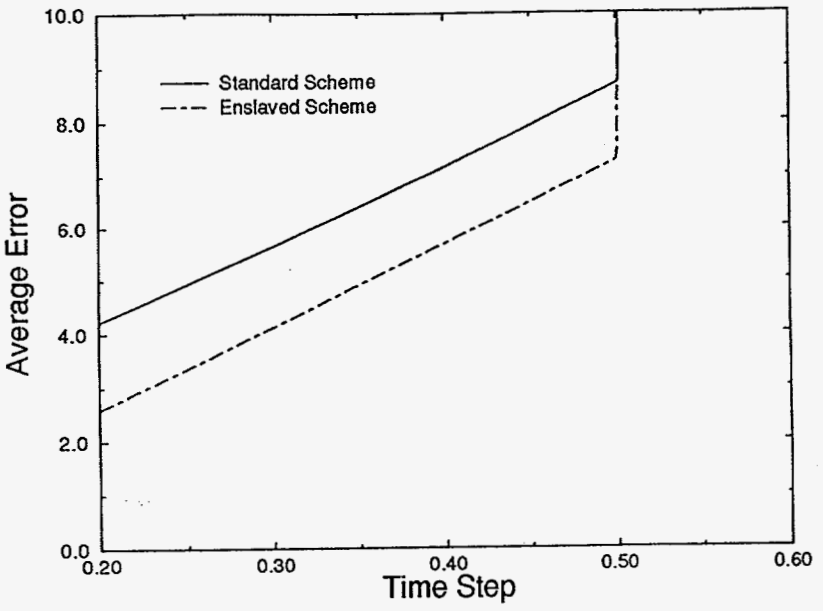

Standard Scheme
Enslaved Scheme
Average Error
Time Step
10.0
8.0
6.0
4.0
2.0
0.0
0.20
0.30
0.40
0.50
0.60

Fig 3

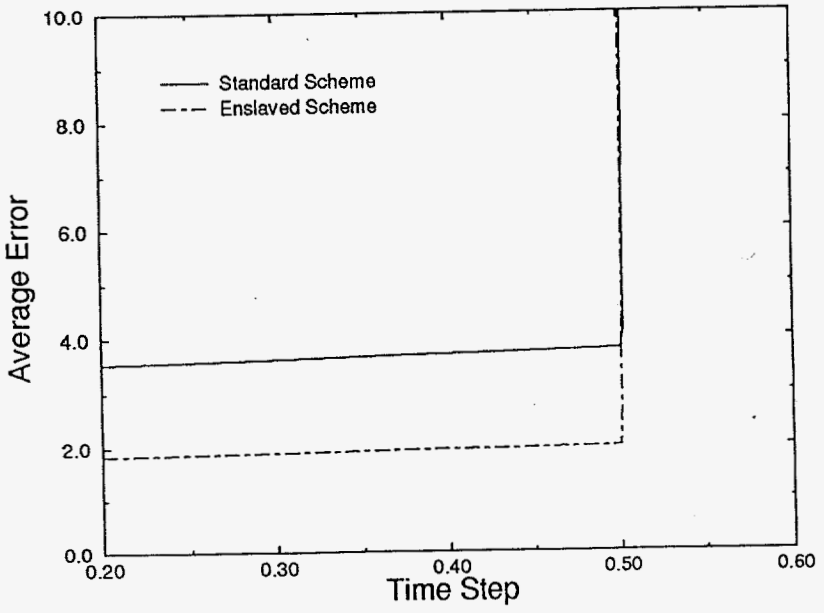

Standard Scheme
Enslaved Scheme
Average Error
Time Step
10.0
8.0
6.0
4.0
2.0
0.0
0.20
0.30
0.40
0.50
0.60

Fig 4