JOURNAL OF COMPUTATIONAL PHYSICS 119, 62–74 (1995)

Space-Time Spectral Element Methods for One-Dimensional Nonlinear Advection–Diffusion Problems

PINHAS BAR-YOSEPH, EDUARD MOSES, UZI ZRAHIA, AND ALEXANDER L. YARIN

Computational Mechanics Laboratory (CML), Faculty of Mechanical Engineering, Technion-IIT, Haifa 32000, Israel

Received February 2, 1994; revised October 13, 1994

The following space-time Galerkin spectral element methods are developed and applied to solve the Burgers equation with small viscosity: (a) coupled methods, consisting of an explicit method for hyperbolic dominated equations and an implicit method for parabolic dominated equations; (b) two splitting methods which solve the hyperbolic substep explicitly and the parabolic one implicitly (one uses spectral elements in the explicit part and the other uses the Adams–Bashforth multistep method). A subcycling technique, in which several convective steps are taken for each implicit viscous step was also investigated for the two splitting methods. A stability analysis of the four methods is performed and subsequent results are debated. A convergence study and a comparison of computer execution time for the four methods is made and the results are discussed. Comparative study leads to the conclusion that the space-time spectral element splitting method with subcycling is superior to the other methods presented in terms of robustness and computer execution time. The number of subcycles should be kept low (2–3) in order to avoid significant loss of accuracy. The coupled explicit method is also applied to the solution of the one-dimensional coupled continuity, momentum, and energy equations for non-isothermal flow of an ideal gas with temperature dependent properties in a cylindrical duct of variable radius.

1. INTRODUCTION AND PROBLEM STATEMENT

1.1. Scalar Equation

Consider the Burgers equation which in its conservation form can be written as

$$\frac{\partial U}{\partial t}+\frac{1}{2}\frac{\partial}{\partial x}(U^2)=\nu\frac{\partial^2 U}{\partial x^2};\quad x\in(-1,1),t>0, \tag{1}$$

with the initial condition

$$U(x,0)=g_1(x);\quad x\in[-1,1], \tag{2}$$

and the boundary conditions

$$\begin{aligned} U(-1,t)&=g_2(t);\quad t\geq 0,\\ U(1,t)&=g_3(t);\quad t\geq 0, \end{aligned} \tag{3}$$

where for most fluid mechanics applications ν is a small positive parameter.

Two test cases have been singled out from the literature and are analysed in this work: the "*sine*" test case and the "*ramp*" one [1, 2] for which the initial and boundary conditions are:

$$\begin{aligned} &\text{Example 1. The sine:}\\ &\quad g_1(x)=-\sin(\pi x);\quad g_2(t)=0;\quad g_3(t)=0.\\ &\text{Example 2. The ramp:}\\ &\quad g_1(x)=-x;\qquad\ g_2(t)=0;\quad g_3(t)=0. \end{aligned} \tag{4}$$

For small values of the viscosity ν, Burgers equation admits a thin nonsingular inner layer that must be resolved if accurate numerical solutions are to be obtained [1–4]. Burgers equation is a very suitable model for testing various computational algorithms for convection dominated flows where severe gradients are anticipated. A great deal of effort has been made, during the past few years, in attempting to compute efficiently the solution of this equation for very small values of viscosity [1–13]. Among the computational methods, finite difference [1, 8], spectral [1], spectral element [1, 4], space-time finite element [2, 3, 5, 6, 9–11, 13], and boundary element [7] methods have been most used. A comparison of spectral methods with finite difference ones is very instructive [1]; it is shown that the real superiority of spectral methods is achieved when a natural decomposition of the solution domain is employed (see also [12, 14]). One of the main conclusions of this comprehensive study is that "with spectral discretization, higher order time schemes should be used." The main objective of the present study is to address this issue by presenting new space-time spectral element methods which offer high order spatial and temporal discretizations. High resolution temporal schemes are reviewed in [15–17] and the references therein.

1.2. System of Equations

As a realistic application of the procedures developed in this work, we will also present the solution of the coupled continuity, momentum, and energy equations for the flow of an ideal gas

0021-9991/95 $6.00

(air) with temperature dependent properties through a cylindrical duct of variable radius. The equations are

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho U) + 2\frac{\rho U}{\mathcal{R}}\frac{d\mathcal{R}}{dx} = 0 \\
&\frac{\partial}{\partial t}(\rho U) + \frac{\partial}{\partial x}(\rho U^2 + p) + 2\frac{\rho U^2}{\mathcal{R}}\frac{d\mathcal{R}}{dx} \\
&\quad - \frac{4}{3}\frac{\partial}{\partial x}\left(\nu\frac{\partial U}{\partial x}\right) + 8\frac{\nu U}{\mathcal{R}^2} = 0 \\
&\frac{\partial}{\partial t}\left[\rho\left(e + \frac{U^2}{2}\right)\right] + \frac{\partial}{\partial x}\left\{\left[\rho\left(e + \frac{U^2}{2}\right) + p\right]U\right\} \\
&\quad + 2\frac{U}{\mathcal{R}}\left[\rho\left(e + \frac{U^2}{2}\right) + p\right]\frac{d\mathcal{R}}{dx} - \frac{\partial}{\partial x}\left(\kappa\frac{\partial T}{\partial x}\right) \\
&\quad + 2\frac{h}{\mathcal{R}}(T - T_w) = 0,
\end{aligned} \tag{5}$$

where ρ is the gas density; U denotes the gas velocity; $\mathcal{R}(x)$ is the variable radius of the duct (x being the duct axial coordinate); p is the pressure; e denotes the internal energy; T is the temperature; index w denote the *wall* of the duct; ν is the kinematic viscosity; κ denotes the conductivity, and h is the heat transfer coefficient calculated from

$$\mathrm{Nu} = 0.0483\,\mathrm{Re}^{0.783} \tag{6}$$

with Nu as the Nusselt number based on $\mathcal{R}$ and Re as the Reynolds number based on $\mathcal{R}$. The duct is connected to a constant pressure reservoir from which gas flows to the atmosphere. The connection is done via an opening and closing valve. The boundary conditions and geometric setup will be presented in Section 5.

1.3. Computational Method

The solution method chosen in this work is the spectral element method which was first introduced by Patera [18]. This is a high order weighted residuals method which exploits the rapid convergence rates of spectral methods while retaining the geometric flexibility of the low order isoparametric finite element methods. According to the semidiscrete finite element approach, the spatial domain is broken up into macro spectral elements. Within each element the dependent variables are expanded in terms of pth-order tensor product Lagrangian interpolants through the Legendre–Gauss–Lobatto collocation points. The Legendre–Gauss–Lobatto points are clustered near elemental boundaries and are chosen because of their interpolation and quadrature properties. The group finite element approach is applied for approximating the convective term [19]. The semidiscrete equations are generated using the weighted residuals method. Time marching is usually done by multistep

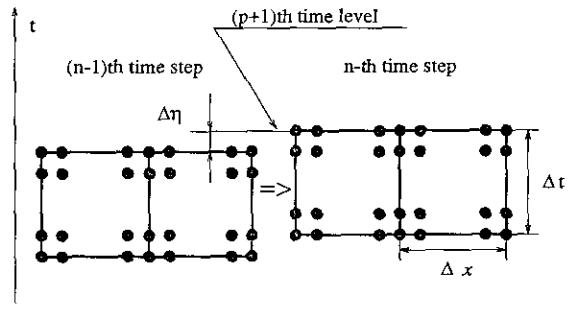

FIG. 1. Graphical depiction of the timewise translation of a mesh consisting of two elements of degree 3 (the horizontal displacement is not physical but only for visualization purposes).

techniques [15, 20]. This yields unbalanced schemes (i.e., exponential convergence in space for problems with smooth computational versus algebraic convergence in time). Hence, it would be more natural to use spectral approximations both in space and time.

Applying the Galerkin method in both time and space yields the weak formulation of the problem. The computational domain at each time step contains a set of spectral elements in the spatial direction and one element in the temporal direction. A time level within each element is defined as all the nodes having the same time coordinate. The nonlinear algebraic equations obtained from this formulation present a coupling of all the time levels belonging to the solution domain at each time step. A direct solution of this global set of equations appears at first sight to be less efficient than a semi-discrete approach unless extremely fast solvers of large sparse nonsymmetric linear systems are employed [21]. An alternative strategy which shows great promise is the following [16]:

(i) The nonlinear set of equations including all time levels in the solution domain are solved directly for the first time step only. Afterwards, a time increment is chosen such that there is an overlapping time region between the first domain of the solution and the second one (Fig. 1). This overlapping includes $(p - 1)$ time levels. The approximate solution within the first domain is used to predict the approximate solution within the second domain for the overlapping time levels. This is done by applying an interpolation operator from the first domain onto the second one. The solution of the remaining upper time level which belongs to the second domain is then calculated through the set of equations from the Galerkin spectral element formulation.

(ii) The coefficient matrix of the approximate solution is partitioned into submatrices so that each submatrix located off-diagonal is a diagonal matrix, while those which are located on the diagonal are full matrices (Fig. 2 and [3, 16, 17]). This particular structure is used for constructing explicit/implicit algorithms to solve the unknowns located at the upper non-overlapping time level.

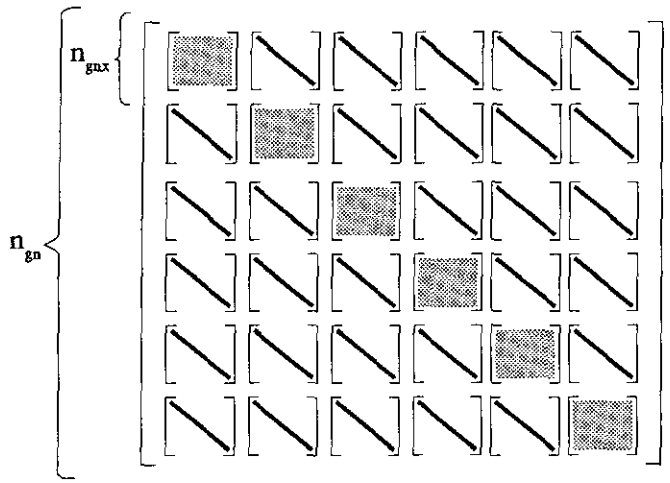

FIG. 2. Graphical outline of the special sparse structure of the global coefficient matrix for $p = 5$.

The present work appears to be the first attempt to solve non-linear one-dimensional conservation laws by space–time spectral elements.

An outline of the remainder of the paper is as follows: in Section 2, the general space–time spectral element formulation is established; in Section 3, the procedure highlighted above is applied to yield the following space–time spectral element methods to solve the Burgers equation for the whole range of viscosity:

(i) Coupled methods:

ia. Hyperbolic dominated equations are treated explicitly.

ib. Parabolic dominated equations are treated implicitly.

(ii) Splitting methods:

iia. The first fractional step uses an explicit scheme to march the nonlinear advection term. In the second fractional step the viscous correction is carried out via an implicit scheme.

iib. The convective term is treated explicitly using pth-order Adams–Bashforth multistep schemes. In the second step, the viscous term is treated implicitly using the above-mentioned space-time spectral element approach.

A modified Galerkin finite elements method for solving the time-dependent Navier–Stokes equations has been developed by Gresho *et al.* [22, 23]. The explicit (forward Euler) method was implemented with a subcycling strategy that permits less frequent updates of the pressure field, relative to the stability-limited processes of advection and diffusion, with little loss of accuracy. In the present study, a different subcycling technique, in which several convective steps are taken for each implicit viscous step is suggested and numerically investigated.

Stability analysis and numerical results are presented and discussed in Sections 4, 5, and 6. Finally, conclusions are drawn in Section 7.

2. SPECTRAL ELEMENT FORMULATION

The solution methods proposed in this work will be explicitly presented as applied to the viscous Burgers equation. Extension to systems of equations is straightforward.

2.1. Spatial Discretization

A weak formulation in the spatial direction is yielded by use of the method of weighted residuals. The solution is approximated as

$$U(\xi, t) \approx {}_e\bar{\mathbf{u}}^{\mathrm{T}}(t) \cdot {}_e\mathbf{h}(\xi)$$
$$U^2(\xi, t) \approx {}_e\bar{\mathbf{u}}^{2^{\mathrm{T}}}(t) \cdot {}_e\mathbf{h}(\xi), \tag{7}$$

where $\mathbf{h}$ denotes the vector of trial functions; U is the exact solution; u is the approximate solution; the overbar denotes the semidiscrete model; a dot stands for the usual scalar product; $\bar{\mathbf{u}}^2$ denotes the diadic product of $\bar{\mathbf{u}}$ by itself; $\xi \in [-1, 1]$ is the local (element level) spatial coordinate; subscript e stands for "*element*"; superscript T denotes *transposition*. As the Burgers equation (1) is cast in a conservative form, group formulation [19] was used (Eq. (7b)) to deal with its nonlinearity; i.e., the flux terms were approximated in the same way as the solution itself.

Using this approximation in conjunction with the Bubnov–Galerkin method yields

$$\mathbf{M}\frac{d\bar{\mathbf{u}}}{dt} = -\mathbf{A}\bar{\mathbf{u}}^2 - \nu\mathbf{B}\bar{\mathbf{u}}, \tag{8}$$

where $\mathbf{M}$, $\mathbf{A}$, and $\mathbf{B}$ are given by

$$\mathbf{M} = \mathop{\mathscr{A}}_{e=1}^{n_{\mathrm{el}}} {}_e\mathbf{M}$$
$$\mathbf{A} = \mathop{\mathscr{A}}_{e=1}^{n_{\mathrm{el}}} {}_e\mathbf{A} \tag{9}$$
$$\mathbf{B} = \mathop{\mathscr{A}}_{e=1}^{n_{\mathrm{el}}} {}_e\mathbf{B}$$

here $\mathscr{A}$ is the finite element assembly operator; n_{el} is the total number of space elements; ${}_e\mathbf{M}$, ${}_e\mathbf{A}$, and ${}_e\mathbf{B}$ are the corresponding matrices of the eth element,

$${}_eM_{ij} = \int_{-1}^{1} {}_eh_i \, {}_eh_j \, d\xi$$

$$
{}_eA_{ij} = \frac{2}{{}_e\Delta x}\int_{-1}^{1} {}_eh_i \frac{d_e h_j}{d\xi}\, d\xi \qquad (10)
$$

$$
{}_eB_{ij} = \frac{4}{{}_e\Delta x^2}\int_{-1}^{1} \frac{d_e h_i}{d\xi}\frac{d_e h_j}{d\xi}\, d\xi,
$$

and ${}_e\Delta x$ is the x-wise length of the eth element.

The interpolation vector of the eth element, ${}_e\mathbf{h}$, is defined via the Legendre–Lagrange polynomials which are discretely orthogonal with respect to the Gauss–Lobatto quadrature. Applying this quadrature to the ${}_e\mathbf{M}$ matrix leads naturally to a diagonal (lumped) matrix whose diagonal terms are

$$
{}_eM_{ii} = w_i = \int_{-1}^{1} {}_eh_i^2(\xi)\, d\xi, \qquad (11)
$$

where w_i are the Gauss–Lobatto weights.

2.2. Spatial and Temporal Discretization

The residuals of Eq. (8) are weighted by the same $\mathbf{h}$ family of trial functions which were used above (only depending on t now) to yield the space-time weak formulation.

Each element of $\bar{\mathbf{u}}(t)$ and $\bar{\mathbf{u}}^2(t)$ is approximated as

$$
\begin{aligned}
\bar{u}_i(\eta) &= \mathbf{u}_i^{\mathrm{T}} \cdot \mathbf{h}(\eta) \\
\bar{u}_i^2(\eta) &= \mathbf{u}_i^{2^{\mathrm{T}}} \cdot \mathbf{h}(\eta),
\end{aligned} \qquad (12)
$$

where $\eta \in [-1, 1]$ is the local (element level) temporal coordinate; the absence of the overbar on $\mathbf{u}$ denotes both spatial and temporal discretisation; $\mathbf{u}_i$ denotes a subvector of $\mathbf{u}$ containing the discrete values of the solution corresponding to the global spatial nodal position i.

Using Eq. (12) in conjunction with the space-time Bubnov–Galerkin method yields

$$
\frac{1}{2}\frac{\Delta t}{\Delta x}\mathbf{C}\mathbf{u}^2 + 2\frac{\nu \Delta t}{\Delta x^2}\mathbf{D}\mathbf{u} + \mathbf{E}\mathbf{u} = \mathbf{0}, \qquad (13)
$$

where a uniform spatial grid (Δx) and uniform time step (Δt) were considered for the sake of emphasizing the two dimensionless numbers characterizing parabolic and hyperbolic equations, respectively,

$$
\begin{aligned}
\alpha_h &= \frac{\Delta t}{\Delta x} \quad \text{—Courant number} \\
\alpha_p &= \frac{\nu \Delta t}{\Delta x^2} \quad \text{—diffusion (Fourier) number.}
\end{aligned} \qquad (14)
$$

$\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{u}$ are to be described below (the right-hand side vector is equal to $\mathbf{0}$ only because the system is shown before the imposition of initial and boundary conditions).

We will hereafter use trial and basis functions of the same degree in both spatial and temporal directions, as it was shown to be the optimal combination for both purely parabolic and purely linear hyperbolic equations [16, 17]. Then there are $(p + 1)$ nodal points per element in both the temporal and spatial directions. $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{E}$ are $n_{\mathrm{gn}} \times n_{\mathrm{gn}}$ matrices, composed of $(p + 1) \times (p + 1)$ submatrices, each of them consisting of $n_{\mathrm{gnx}} \times n_{\mathrm{gnx}}$ elements; n_{gn} denotes the total number of nodes in a space-time strip extending over one element temporally and all the mesh spatially; $n_{\mathrm{gn}} = (p + 1) \cdot n_{\mathrm{gnx}}$; n_{gnx} denotes the global number of nodes in the spatial direction; $n_{\mathrm{gnx}} = n_{\mathrm{el}} \cdot p + 1$ and n_{el} denotes the number of spatial elements. $\mathbf{u}$ consists of $(p + 1)$ subvectors $\mathbf{u}^m$ one above the other ($m \in [1, p + 1]$); each subvector $\mathbf{u}^m$ consists of n_{gnx} elements and contains the approximate solution at temporal level m. $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{E}$ are given by

$$
\begin{aligned}
&\mathbf{C}^{mn} = \mathbf{D}^{mn} = \mathbf{0} && \forall\, m \neq n \\
&\mathbf{C}^{nn} = w^n\mathbf{A} && \forall\, m = n \\
&\mathbf{D}^{nn} = w^n\mathbf{B} && \\
&\mathbf{E}^{mn} = {}_eA_{mn}\,\mathrm{Diag}(\mathbf{w}) && \forall\, m, n,
\end{aligned} \qquad (15)
$$

where a pair of superscripts denotes a submatrix and Diag($\mathbf{w}$) stands for a diagonal matrix having the elements of $\mathbf{w}$ (Eq. (11)) on its diagonal; w^n are the Gauss–Lobatto weights. Figure 2 depicts the structure of the coefficient matrix after assemblying. This special sparse structure remains the same for other evolution equations not having mixed derivatives [3, 16, 17].

3. COMPUTATIONAL ALGORITHMS

By examining system (13) it is observed that, once the initial and boundary conditions are imposed, it can in principle be solved to yield the solution over one space-time strip. Thereafter, the last time level can be used as the initial condition for the next strip and the solution is marched in time. This kind of solution which would be fully implicit and expected to have the best features of stability and accuracy can be efficiently programmed on a parallel supercomputer. Implementation of fast iterative methods for solving this algebraic set of equations having large, sparse, nonsymmetric matrices will not be attempted in this work.

An alternative way of solving the system of equations is the following [16, 17]; assuming that we start with a space-time strip in which the solution is fully known, we can advance it time-wise with the time interval separating the last time level from the before-the-last one by using the interpolation operator. In this way the only time level at which the new solution is unknown is the last (i.e., the $(p + 1)$th) time level (Fig. 1). Examining system (13) and Fig. 2 we observe that the solution at the last time level of the strip can be yielded by one of the following procedures:

(1) Coupled methods:

ia. The first p subvectors of $\mathbf{u}$ are calculated via interpolation from the p overlapping time levels of the element and the first p submatrix equations in Eq. (13) are discarded while corresponding terms in the last one are transferred to the right-hand side. The remaining submatrix equation is:

$$\tfrac{1}{2}\alpha_h \mathbf{C}^{p+1,p+1}(\mathbf{u}^{p+1})^2 + 2\alpha_p \mathbf{D}^{p+1,p+1}\mathbf{u}^{p+1} + {}_eA_{p+1,p+1}\mathbf{u}^{p+1} = \mathbf{F}(\mathbf{u}^1, ..., \mathbf{u}^p), \tag{16}$$

where $\mathbf{F}$ contains all the interpolated time levels previous to the last one. As can be seen, the n_{gnx} unknowns which we are left with in Eq. (16) are coupled as both $\mathbf{C}^{p+1,p+1}$ and $\mathbf{D}^{p+1,p+1}$ are full matrices. This version is therefore an implicit one and is denoted as ISTSE (implicit space time spectral elements). The nonlinear system (16) is solved by the Gauss–Siedel iterative method. As the off-diagonal terms are transferred to the right-hand side, advantage is taken of the following property:

$$\begin{aligned} {}_eA_{i,i} &= 0 \qquad \forall i \neq 1, i \neq p+1 \\ {}_eA_{1,1} &= 0.5 \\ {}_eA_{p+1,p+1} &= -0.5. \end{aligned} \tag{17}$$

For this reason the diagonal nonlinear terms vanish in all but the first and last of the equations (see Eq. (15)). This leaves us with a linear set of decoupled equations in each iteration of the Gauss–Seidel solver, allowing an especially efficient solution.

ib. The first $(p-1)$ and the $(p+1)$th submatrix equations in Eq. (13) are discarded to account for interpolation results. The remaining matrix equation is

$${}_eA_{p,p+1}\mathbf{u}^{p+1} = \mathbf{F}(\mathbf{u}^1, ..., \mathbf{u}^p). \tag{18}$$

It is observed that in Eq. (18) the nodal values of the solution are fully uncoupled. This version is an explicit one and is denoted as ESTSE (explicit space time spectral elements).

(ii) Splitting methods:

iia. • In the first fractional step (the hyperbolic one) system (13) is solved via ESTSE for $\nu = 0$, providing a solution, $\hat{\mathbf{u}}$, of the purely nonlinear hyperbolic part of the problem.

$${}_eA_{p,p+1}\hat{\mathbf{u}}^{p+1} = \mathbf{F}(\mathbf{u}^1, ..., \mathbf{u}^p; \alpha_h). \tag{19}$$

• The second fractional step (the parabolic one) consists of solving system (13) with the nonlinear terms calculated via the solution of the first substep:

$$2\alpha_p \mathbf{D}^{p+1,p+1}\mathbf{u}^{p+1} + {}_eA_{p+1,p+1}\mathbf{u}^{p+1} = \mathbf{F}(\mathbf{u}^1, ..., \mathbf{u}^p, \hat{\mathbf{u}}^{p+1}). \tag{20}$$

The resulting system is linear and readilly solved to provide the hyperbolic–parabolic definitive solution.

It is observed that, as the ESTSE solution of the nonlinear hyperbolic part is expected to require smaller time steps for stability than the ISTSE solution of the parabolic one [16, 17], this approach offers the possibility to employ a subcycling technique, i.e., several hyperbolic explicit substeps for one parabolic implicit correction. Thus, the conditional stability of the ESTSE algorithm for linear purely hyperbolic equations [16] is combined with the unconditional stability of the ISTSE one for the purely parabolic equations [17] to offer a robust and CPU-time efficient solution. This approach will be denoted as EISTSE (Explicit Implicit Space Time Spectral Elements) without subcycling and as CEISTSE with subcycling (C stands for subCycling).

iib. • The nonlinear terms are treated via a multistep pth-order Adams–Bashforth scheme, thus yielding an explicit solution to the first substep of the previous approach.

• The second substep is identical to that in EISTSE.

This procedure is similar to EISTSE and will provide the opportunity to estimate the performance of the latter by comparison with a well-known scheme. This approach will be denoted as EIABSTSE (explicit implicit Adams–Bashforth space time spectral elements).

4. STABILITY

The stability of the ISTSE and the ESTSE approaches was examined via the matrix method. For the linear diffusion–convection equation this approach would be exact. Obviously, the algebraic systems being nonlinear, the eigenvalues depend on the solution itself and, thus, on time. Therefore, by using the method of local linearization, the maximum modulus of eigenvalues was calculated along the whole process of solution and checked to be within the unit circle. It must be emphasized that the above criterion is a necessary but not a sufficient one in our case, as it covers only local linearized stability [8]. Another typically nonlinear instability source is the aliasing error, if it grows above a critical threshold during a long time integration [8]. We have performed very long time integrations ($\mathbb{O}(10^4)$ time steps) and did not detect any aliasing-related instability.

In order to calculate the amplification matrix, each system was rearranged into the form

$$\mathbf{u}^t = \mathbf{K}\mathbf{u}^{t-\Delta t}, \tag{21}$$

where the superscripts indicate the time level of the corresponding solutions and $\mathbf{K}$ is the amplification matrix which can be decomposed as

$$\mathbf{K} = \mathbf{P}\mathbf{H}, \tag{22}$$

where $\mathbf{H}$ is the interpolation matrix and $\mathbf{P}$ is specific to each method.

4.1. Interpolation Matrix

The elemental interpolation matrix is a $(p + 1) \times (p + 1)$ matrix, denoted ${}_e\mathbf{H}$ and defined as

$${}_eH_{i,j} = {}_eh_i(\eta_j + \Delta\eta), \tag{23}$$

where η_j are the locations of the elemental time nodes (the Legendre–Gauss–Lobatto quadrature points) and $\Delta\eta$ denotes the time interval between the before-the-last and the last elemental time levels, over which interpolation is required (Fig. 1).

The global interpolation matrix $\mathbf{H}$ has the same dimensions and inner structure (i.e., partition into submatrices) that was highlighted in the previous section and is given by

$$\mathbf{H}^{m,n} = {}_eH_{m,n}\bar{\mathbf{I}}, \tag{24}$$

where a pair of superscripts denotes a submatrix, $\bar{\mathbf{I}}$ is given by

$$\bar{\mathbf{I}} = \mathop{\mathcal{A}}_{e=1}^{n_{el}} \mathbf{I}, \tag{25}$$

and $\mathbf{I}$ is the $(p + 1)$ identity matrix.

4.2. P Operator

ESTSE. After rearranging system (18), the $\mathbf{P}$ matrix in Eq. (22) is given by

$$\begin{array}{lll}
\mathbf{P}^{m,n} = \bar{\mathbf{I}} & \text{if } m = n & \\
\mathbf{P}^{m,n} = \mathbf{0} & \forall m \neq n & \text{for } m \leq p \\
\mathbf{P}^{m,n} = \mathbf{T} & \text{if } n = p & \\
\mathbf{P}^{m,n} = {}_eA_{p,n}\bar{\mathbf{I}} & \forall n \neq p, n \neq p+1 & \text{for } m = p+1 \\
\mathbf{P}^{m,n} = \mathbf{0} & \text{if } n = p+1, &
\end{array} \tag{26}$$

where

$$\mathbf{T} = \left(\frac{1}{2}\alpha_h\mathbf{C}^{p,p}\mathbf{u}^p + 2\alpha_p\mathbf{D}^{p,p}\right)\frac{1}{{}_eA_{p,p+1}}. \tag{27}$$

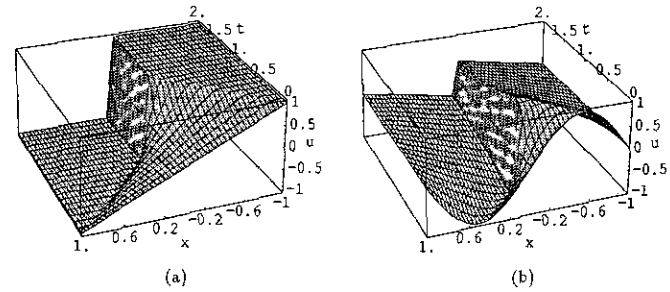

FIG. 3. Perspective view of the solution of the two test cases: (a) Ramp; (b) Sine ($\nu = 1.0 \times 10^{-3}$, $n_{el} = 50$, $p = 8$, $\Delta t = 1.0 \times 10^{-2}$).

ISTSE. By rearrangement of system (16), its $\mathbf{P}$ matrix can be decomposed as

$$\mathbf{P} = \mathbf{Q}^{-1}\mathbf{R}, \tag{28}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are given by

$$\begin{array}{lll}
\mathbf{Q}^{m,n} = \bar{\mathbf{I}} & \text{if } m = n & \\
\mathbf{Q}^{m,n} = \mathbf{0} & \forall m \neq n & \text{for } m \leq p \\
\mathbf{Q}^{m,n} = \mathbf{S} & \text{if } m = n & \\
\mathbf{Q}^{m,n} = \mathbf{0} & \forall m \neq n & \text{for } m = p+1 \\
\mathbf{R}^{m,n} = \bar{\mathbf{I}} & \text{if } m = n & \\
\mathbf{R}^{m,n} = \mathbf{0} & \forall m \neq n & \text{for } m \leq p \\
\mathbf{R}^{m,n} = {}_eA_{m,n}\bar{\mathbf{I}} & \forall n & \text{for } m = p+1
\end{array} \tag{29}$$

and

$$\mathbf{S} = \tfrac{1}{2}\alpha_h\mathbf{C}^{p+1,p+1}\mathbf{u}^{p+1} + 2\alpha_p\mathbf{D}^{p+1,p+1} + {}_eA_{m,n}\bar{\mathbf{I}}. \tag{30}$$

5. RESULTS AND DISCUSSION

The solutions of the two test cases considered in this study (Eq. (4)) are presented in Fig. 3. The time step was kept constant and equal to $\Delta t = 0.01$ and 200 time steps were performed.

5.1. Stability

5.1.1. *Locally Linearized Analysis*

ESTSE. For exemplification purposes, ESTSE was put in a finite difference stencil form for $p = 2$:

$$i\text{-even}, \quad \nu\frac{u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}}{\Delta x^2} + \frac{1}{2}\frac{u_{i+1}^{n-1^2} - u_{i-1}^{n-1^2}}{2\Delta x} + \frac{u_i^n - u_i^{n-2}}{2\Delta t} = 0;$$

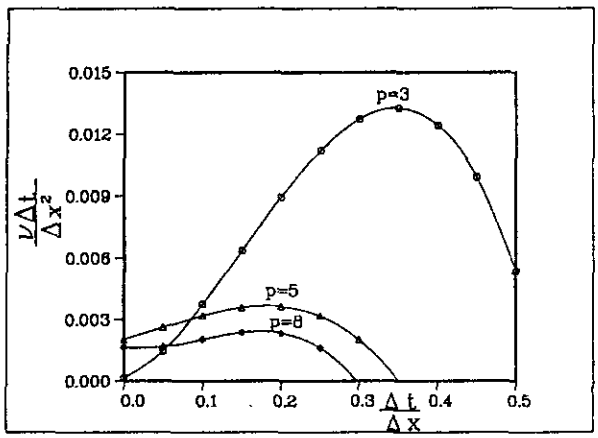

FIG. 4. ESTSE. Stability diagram for various polynomial degrees.

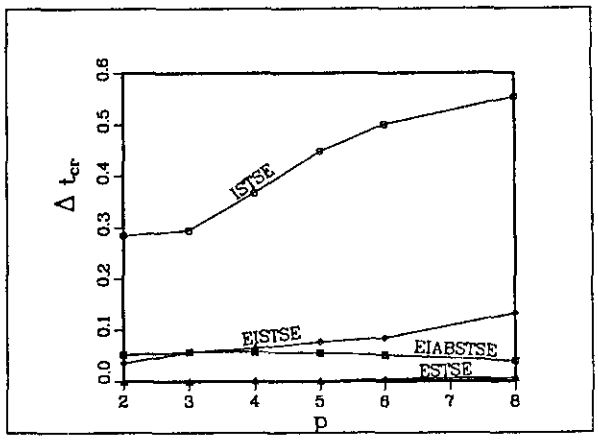

FIG. 6. A posteriori stability analysis—comparison between the four methods ($\nu = 1.0 \times 10^{-2}$, $n_{dof} = 120$, 2000 time steps).

$$i\text{-odd}, \quad \nu\frac{u_{i+2}^{n-1} - 8u_{i+1}^{n-1} + 14u_i^{n-1} - 8u_{i-1}^{n-1} + u_{i-2}^{n-1}}{2\Delta x^2} \tag{31}$$
$$+\frac{1}{2}\frac{u_{i+2}^{n-1^2} - 4u_{i+1}^{n-1^2} + 4u_{i-1}^{n-1^2} - u_{i-2}^{n-1^2}}{4\Delta x}$$
$$+\frac{u_i^n - u_i^{n-2}}{2\Delta t},$$

where superscripts denote the time level and subscripts denote spatial position. As can be easilly seen, the first equation of (31) describes the central difference scheme of second order in both space and time, which is well known to be unconditionally unstable [8]. This was also verified by calculating the time variation of the eigenvalues. It was found that, no matter what time steps or viscosities where used, the maximum modulus of the eigenvalues, even from the first step, was 1.41 or higher.

For higher order polynomials ($p \geq 3$) it was found that there exists in the plane α_p–α_h a region of stability which is enclosed between the respective critical curve and the horizontal axis (Fig. 4).

ISTSE. Figure 5 shows that this approach is stable for higher α_p values than the previous one. It can be observed that $\alpha_h \approx 0.25$ allows a maximum value of α_p for all degrees. At higher values of α_h the maximum admissible α_p begins to decrease, apparently tending to zero as α_h approaches 1. The stability region dwindles as the order of the method increases.

5.1.2. *A Posteriori Analysis*

A posteriori stability analysis was performed by means of the minimum time step that causes instability to set in, for two different values of viscosity ($\nu = 10^{-2}$ and $\nu = 10^{-3}$) and a constant number of degrees of freedom per time step. As can be observed from comparing Fig. 6 to Fig. 7 (n_{dof} is the number of degrees of freedom), at $\nu = 10^{-2}$ the ISTSE scheme is by far the most stable (Fig. 6) featuring the greater critical time

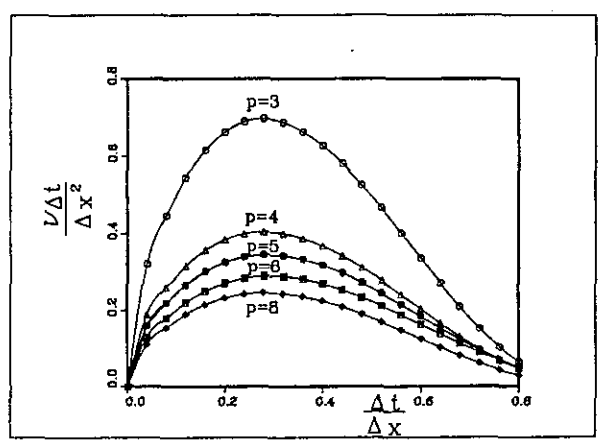

FIG. 5. ISTSE. Stability diagram for various polynomial degrees.

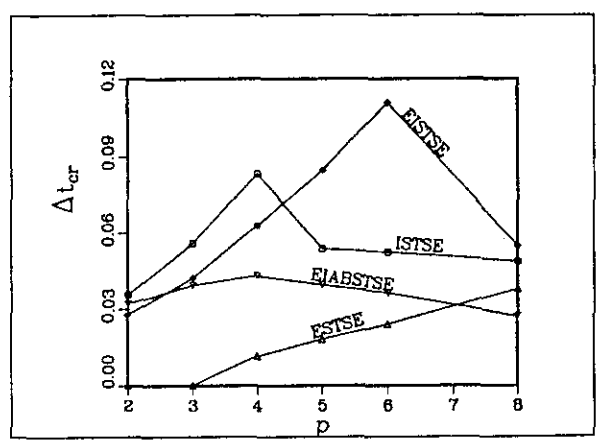

FIG. 7. A posteriori stability analysis—comparison between the four methods ($\nu = 1.0 \times 10^{-3}$, $n_{dof} = 120$, 2000 time steps).

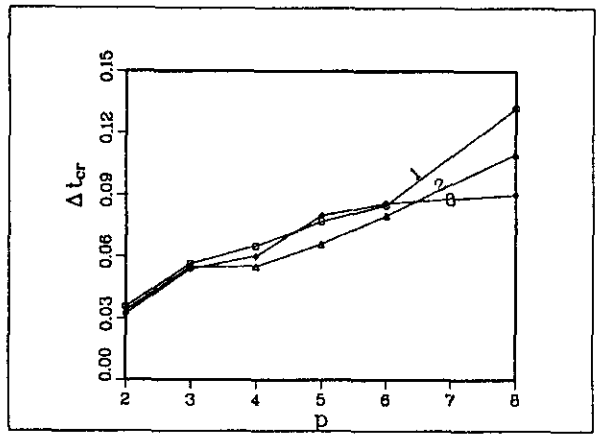

FIG. 8. A posteriori stability analysis—subcycling schemes ($\nu = 1.0 \times 10^{-2}$, $n_{dof} = 120$, 2000 time steps).

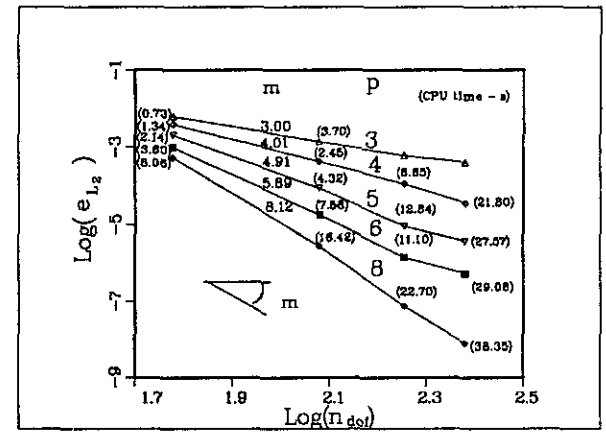

FIG. 10. ESTSE. Convergence rates for various polynomial degrees.

step size. At $\nu = 10^{-3}$, though, scales are somewhat reversed (Fig. 7), as ISTSE loses stability while ESTSE gains some, so that they are comparable. It can be observed that the two splitting methods (EISTSE and EIABSTSE) have a critical time step $\Delta t_{cr} \in [0.05, 0.1]$ at both viscosities. Thus then, at lower viscosities it is conceivable that EISTSE becomes actually more stable than ISTSE, especially for higher polynomial degrees.

This observation shows that there may be a possibility to use a splitting method with subcycling, which would have the same stability features as the implicit one but would consume less CPU time. This is illustrated in Figs. 8 and 9 which show the combined influence of the polynomial degree and the number of subcycling steps on the critical time step size under the same conditions cited above. It is observed that a major change in the values of the critical time step size can occur when passing from no subcycling (curve 1) to subcycling (curves 2 and 3). Afterwards, it is observed that the number of subcycling steps does not have a great impact on the critical time step size.

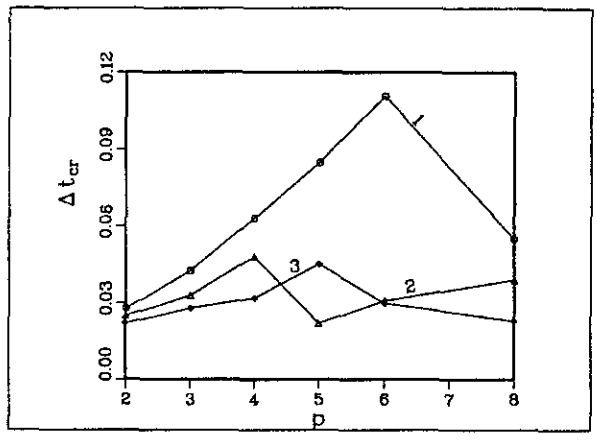

FIG. 9. A posteriori stability analysis—subcycling schemes ($\nu = 1.0 \times 10^{-3}$, $n_{dof} = 120$, 2000 time steps).

5.2. Convergence

Convergence analysis was achieved in order to compare the performance of the four methods. Comparison is further enhanced by highlighting the convergence rate for a well-known, ABCN, fractional step method using the third-order Adams–Bashforth method in the hyperbolic substep and the Crank–Nicholson method in the parabolic one [1].

The error was measured both by the discrete L_2 and spatial H_1 norms. Results are calculated for the *ramp* test case with $\nu = 1.0 \times 10^{-3}$. The norms were calculated in two different ways, excluding the inner layer and including it. The L_2 norm of the error, excluding the inner layer, is given by

$$e_{L_2} = \int_0^1 \left(\sqrt{(1/t) \int_{-1}^{t-1} (u - U)^2 \, dx} \right) dt, \tag{32}$$

while the H_1 norm under the same conditions is

$$e_{H_1} = \int_0^1 \left(\sqrt{(1/t) \int_{-1}^{t-1} [(u - U)^2 + (\partial u/\partial x - \partial U/\partial x)^2] \, dx} \right) dt, \tag{33}$$

where $x = t - 1$ is the location from which the analytic solution begins to descend from $U = 1$ in the x–t plane, for $t \in [0, 1]$ (see Fig. 3). The integrations are performed by the $(p + 1)$ points Gauss–Lobatto quadrature rule.

The global (including the inner layer) L_2 norm of the error is given by

$$\bar{e}_{L_2} = \int_0^1 \left(\sqrt{(1/2) \int_{-1}^{1} (u - U)^2 \, dx} \right) dt. \tag{34}$$

Figures 10–14 show the rates of convergence of the four approaches outlined in this work, as well as of the ABCN

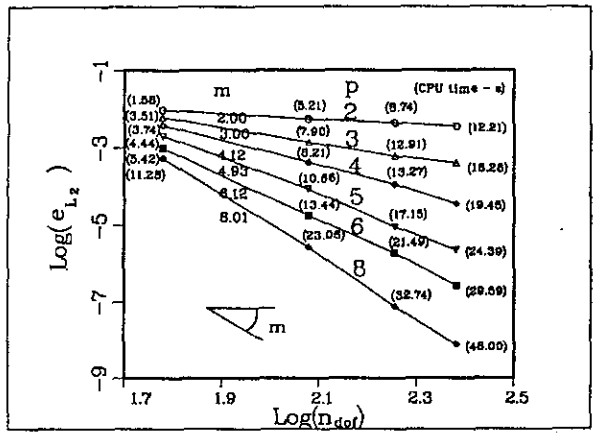

FIG. 11. ISTSE. Convergence rates for various polynomial degrees.

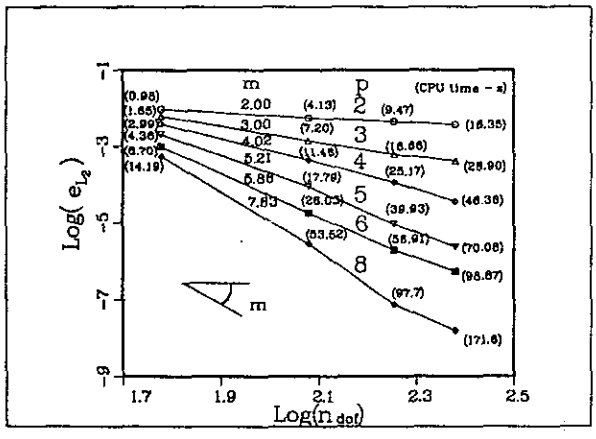

FIG. 12. EISTSE. Convergence rates for various polynomial degrees.

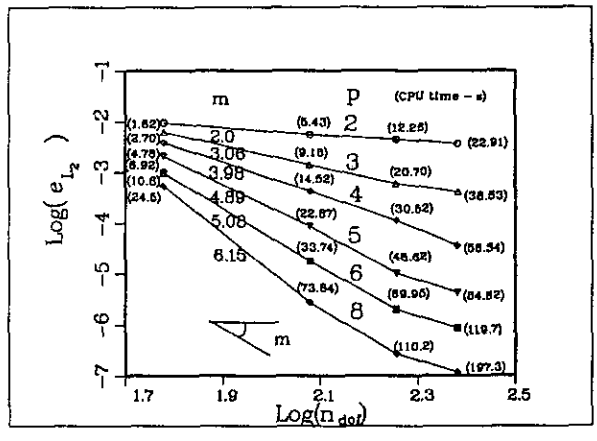

FIG. 13. EIABSTSE. Convergence rates for various polynomial degrees.

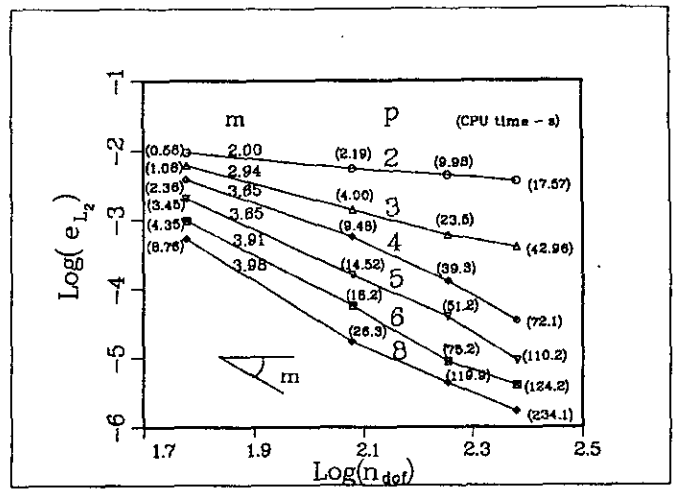

FIG. 14. ABCN. Convergence rates for various polynomial degrees.

mentioned above, relative to the L_2 norm which excludes the inner layer. The rate of convergence, m, was calculated by linear regression.

Figures 10, 11, and 12 show that the ESTSE, ISTSE, and EISTSE allow a spectral rate of convergence, m, more or less equal to the degree of the trial and basis functions.

Figure 13 shows that the EIABSTSE scheme is capable of spectral convergence rates up to $p = 4$ after which its rates are off the expected values (e.g., an average of $m = 6.15$ for $p = 8$). As this reduction occurs only for greater numbers of degrees of freedom, it is assumed to be caused by roundoff error interference in the Adams–Bashforth integration substep (as a similar phenomenon is not in evidence in the EISTSE case, the implicit parabolic part seems to be not responsible in this instance).

Figure 14 shows that the rates of convergence of ABCN are significantly lower than those of the other methods, due to the third-order Adams–Bashforth integration scheme employed.

Figure 15 shows the rate of convergence of the EISTE

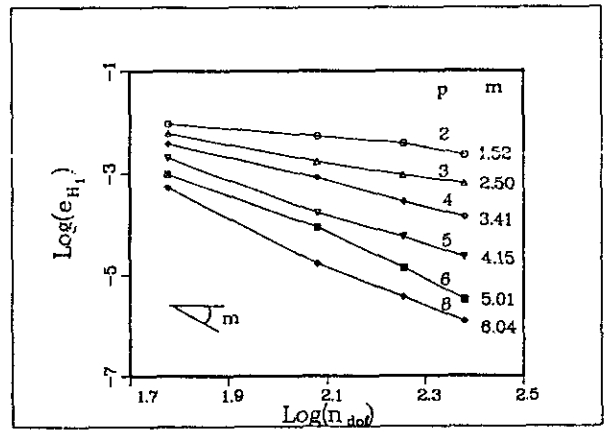

FIG. 15. H_1-based convergence rates for various polynomial degrees.

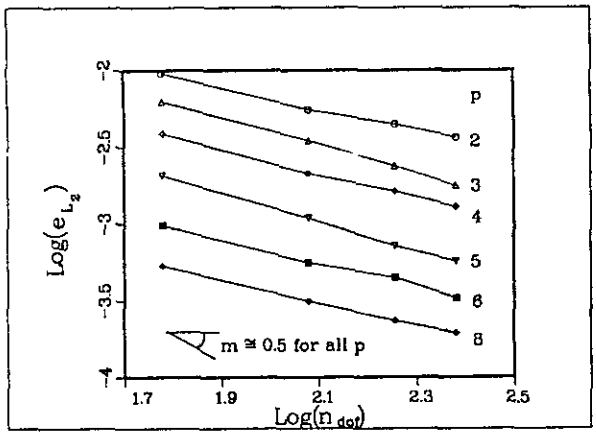

FIG. 16. Convergence rates for various polynomial degrees, including inner layer.

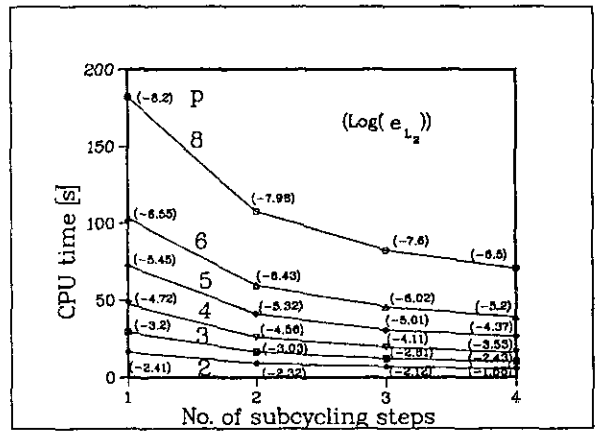

FIG. 18. CPU time requirements versus no. of subcycling steps for constant number of degrees of freedom ($\nu = 1.0 \times 10^{-3}$, $n_{dof} = 240$).

scheme, based on the H_1 norm and excluding the inner layer. It can be observed that, as expected, the rates of convergence are less than the previous ones.

Figure 16 shows the L_2-based rate of convergence of the EISTSE scheme including the inner layer. As expected, it is observed that, for a uniform mesh, no gain in the rate of convergence resulted by increasing the polynomial degree. The rate of convergence remains equal to approximately 0.5 for all the polynomial degrees considered [3, 24].

5.3. CPU Time Requirements

All calculations were performed on a DEC 5000/200 workstation. CPU time consumption, in seconds, is presented in Figs. 10–14 (the numbers in brackets). It is observed that, in terms of CPU time consumption, ESTSE is more efficient than ISTSE by about 20% in the most accurate case (i.e., $p = 8$ and $n_{dof} = 240$) while in some other cases its CPU time consumption is smaller by a factor of three (e.g., $p = 4$ and $n_{dof} = 90$). This wide variation is due to the stability, requiring in some cases the reduction of the time step for the ESTSE approach to converge, while the same requirement was not imposed on ISTSE. It is also observed that in two cases it was deemed useless to mark CPU times for ESTSE (i.e., $p = 3$, $n_{dof} = 180$, 240), as the time step required for convergence was so small that CPU time ranged in the thousands of seconds.

Comparing the CPU times required by ESTSE and EISTSE (Figs. 10 and 12), it is seen that they differ by about an order of magnitude in some cases and by a factor of three in other cases. This is caused by similar difficulties caused by the stability time step restriction. Having to reduce the time step for the EISTSE approach means having to solve more algebraic linear systems in its parabolic implicit substep and, hence, a more drastic increase in CPU time. As will be shown, this problem finds its cure in subcycling.

Figure 13 shows CPU times comparable to those of EISTSE which points out that it suffers from the same difficulty posed by the need to reduce the time step in order to enforce stability. This tendency can be greatly improved by subcycling, as will be shown further on.

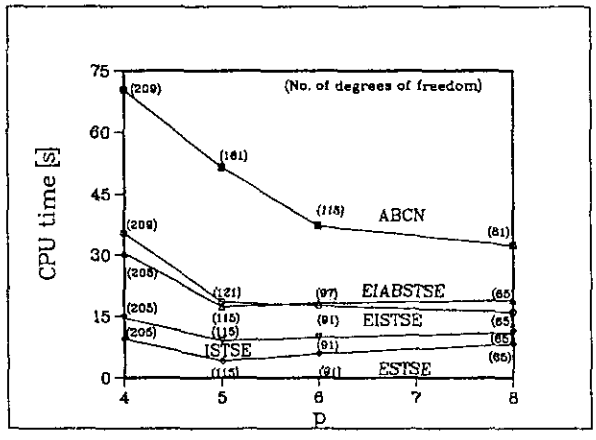

FIG. 17. CPU time requirements versus polynomial degree for constant error ($e_{L_2} = 1.0 \times 10^{-4}$, $\nu = 1.0 \times 10^{-3}$).

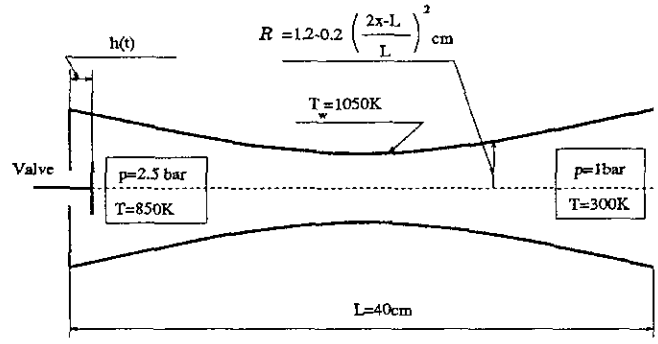

FIG. 19. Geometric setup and boundary conditions for air flow in a duct.

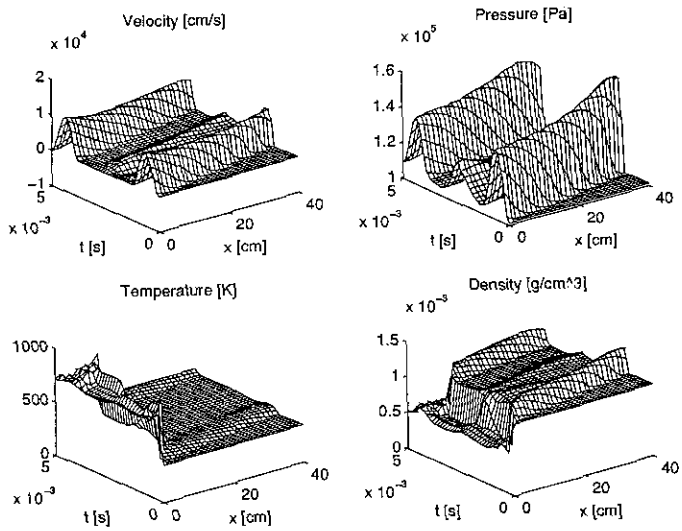

FIG. 20. Velocity, pressure, temperature, and density distributions as functions of time.

While consuming very small CPU time for the lower number of degrees of freedom, ABCN (Fig. 14) seems to need significantly more CPU time in the region of higher numbers of degrees of freedom as its stability-imposed time step drastically decreases.

The above comparison is, nevertheless, not entirely fair, as the accuracy obtained is not constant. A comparison of CPU time requirements of the various schemes under discussion for a constant $e_{L_2} = 1.0 \times 10^{-4}$ is shown in Fig. 17. ISTSE is observed to require about 50% of the CPU time consumed by EISTSE for the same error requirement while the CPU time consumption of ESTSE are reduced to about 30%. EIABSTSE is observed not to differ significantly from EISTSE in this respect, as expected, while ABCN consumes the highest CPU times.

The influence of subcycling on accuracy and CPU time requirements for a constant $n_{\text{dof}} = 240$ and for the CEISTSE scheme is depicted in Fig. 18. It is observed that two or even three subcycling steps cause a very small loss of the accuracy while bringing a reduction in CPU time of up to 50%. Thus, CEISTSE can certainly compete with ISTSE from a CPU time requirement point of view. It is also more stable in the low viscosity regime, where ISTSE has very band stability features. CEISTSE can compete with ESTSE also, especially in the regime of higher viscosities, where the latter has stability problems, as mentioned before. Thus, CEISTSE provides a robust and computationally efficient method, although it involves a certain loss of accuracy.

6. SYSTEM OF EQUATIONS

In order to show the applicability of the algorithms developed in the present work, we will now present the solution of Eqs. (5). The geometric configuration and boundary conditions are presented in Fig. 19. The function $h(t)$ is given by

$$h(t) = \begin{cases} \dfrac{\mathcal{R}}{10} \sin\left(\pi \dfrac{4t}{t_{\text{final}}}\right) & \text{if } 0 \le t \le \dfrac{t_{\text{final}}}{4} \text{ or } \dfrac{3t_{\text{final}}}{4} \le t \le t_{\text{final}} \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

The solution has been performed via the ESTSE scheme and results are presented in Fig. 20 for a uniform mesh of 200 elements of degree 10 (reference solution). The integration has been carried out for a time interval of 5 ms, using a time step of 1 μs. The wiggles in the temperature and density solutions

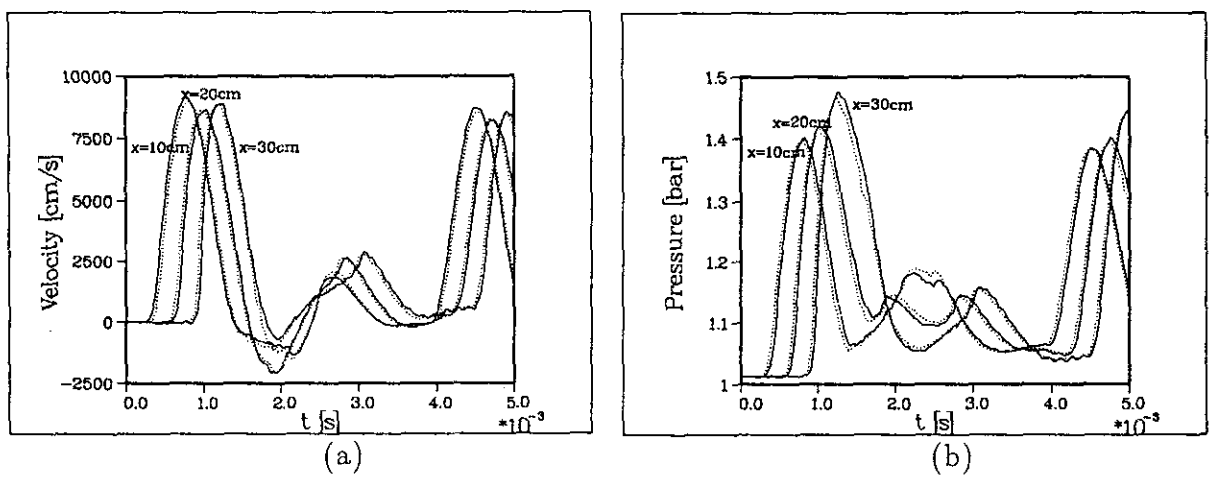

FIG. 21. Comparison of results in present work with results yielded by the commercial code *Fluent*. Solid line: present calculations; dotted line: *Fluent* results: (a) velocity; (b) pressure.

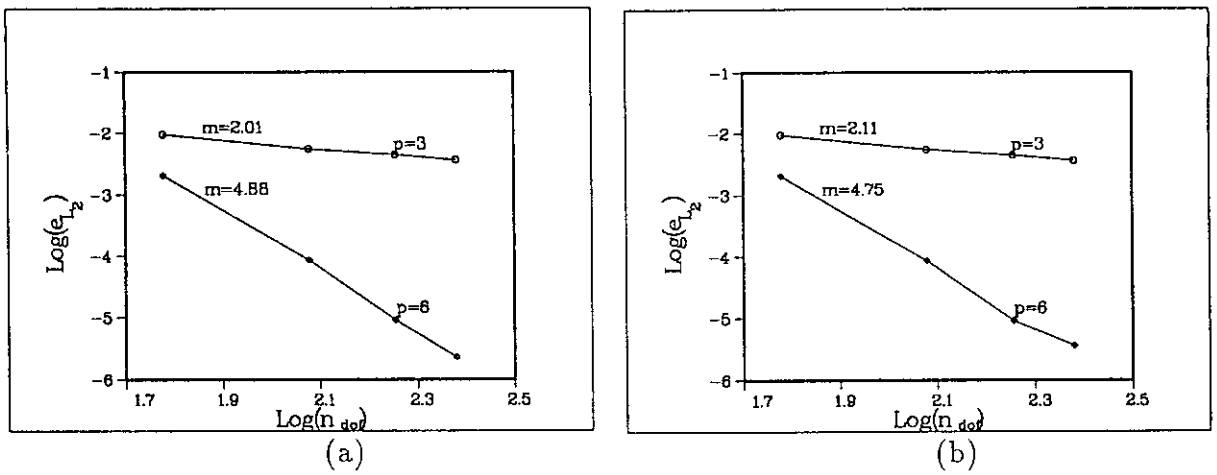

FIG. 22. Convergence rates for the polynomial degrees $p = 3$ and $p = 6$: (a) velocity; (b) pressure.

(Fig. 20) can be reduced in magnitude by either refining the mesh (via a geometrical progression) in the vicinity of the valved end of the duct and using low order elements, or by increasing the order of the elements at this location and exploiting the natural clustering of the Gauss–Lobatto points near the elemental boundaries.

For the sake of comparison, the same problem was solved by using the commercial code *Fluent* [25]. A comparison of the velocities and pressures (as a functions of time) calculated in the present work with those yielded by the *Fluent* code at three selected points is presented in Fig. 21. It is mentioned that the *Fluent* results have been averaged over the section of the duct (since the code allows the minimum two-dimensional problems). It is observed that the results of our calculations coincide fairly well with those of the commercial code.

Convergence rates were subsequently calculated using the L_2 norm with respect to the above-mentioned solution as a reference. Convergence rates for velocity and temperature are presented in Fig. 22 for polynomial degrees $p = 3$ and $p = 6$. It is observed from the above-mentioned figure that convergence rates are satisfactorilly good, although somewhat lower than in the case of the Burgers equation and similar polynomial degrees.

7. CONCLUDING REMARKS

The present work outlines four spectral element-based methods applied to solve the Burgers equation, two of a coupled nature and two of a splitting one. The performance of the four methods is analysed from the point of view of the convergence rates based on the L_2 and H_1 error norms. An a posteriori convergence study is performed both by inclusion of the inner layer and without it. Stability of the four methods is analysed by local linearization and the results are compared to an a posteriori stability analysis. A subcycling technique enabling fast, cost-effective time integration with little loss of accuracy is investigated in the framework of a splitting method. Results show that this approach is superior to the others in terms of robustness and CPU time consumption. A system of nonlinear conservation laws has been solved via the *explicit space-time spectral elements*. This solution models a realistic application involving the non-isothermal flow of a compressible fluid in a variable radius pipe. This approach has been extended to some combustion related problems [26, 27]. Based on our expanding numerical experience, [2, 3, 14, 16, 17, 28, 29], we anticipate that discontinuous spectral elements in the time domain will enhance stability and accuracy of the numerical solution. This topic is currently under investigation.

REFERENCES

1. C. Basdevant, M. Deville, P. Haldenwang, J. M. Lacroix, J. Ouazzani, R. Peyret, P. Orlandi, and A. T. Patera, *Comput. & Fluids* **14,** 23 (1986).
2. P. Bar-Yoseph, *Comput. Mech.* **5,** 145 (1989).
3. P. Bar-Yoseph and D. Elata, *Int. J. Numer. Methods Engrg.* **29,** 1229 (1990).
4. C. Mavriplis, *Comput. Methods Appl. Mech. Engrg.* **116,** 77 (1994).
5. R. C. Mittal and P. Singhal, *Comput. Numer. Methods Engrg.* **9,** 397 (1993).
6. E. Varoğlu and W. D. Liam Finn, *Int. J. Numer. Methods Engrg.* **16,** 171 (1980).
7. K. Kakuda and N. Tosaka, *Int. J. Numer. Methods Engrg.* **29,** 245 (1989).
8. H. Hirsch, *Numerical Computation of Internal and External Flows* (Wiley, Chichester, 1989).
9. J. Caldwell and P. Smith, *Appl. Math. Modelling* **6,** 381 (1982).
10. H. Nguyen and J. Reynen, *Comput. Methods Appl. Mech. Engrg.* **42,** 331 (1984).
11. D. Winterscheidt and K. S. Surna, *Int. J. Numer. Methods Engrg.* **36,** 3629 (1993).
12. R. Arina and C. Canuto, *J. Comput. Phys.* **105,** 290, (1993).
13. D. J. Evans and A. R. Abdullah, *Int. J. Numer. Methods Engrg.* **23,** 1145 (1986).

14. P. Bar-Yoseph, M. Israeli, and S. Weichandler, "The Asymptotic Spectral Element Method," in *2nd Int. Conf. on Spectral and High Order Methods, Montpellier, June 22–26 1992*, edited by C. Bernardi and Y. Maday.

15. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).

16. U. Zrahia and P. Bar-Yoseph, *Comput. Methods Appl. Mech. Engrg.* **116,** 135 (1994).

17. P. Bar-Yoseph, "Spectral Element Methods for Time Dependent Problems," in *3rd Int. Conf. on Boundary and Finite Element, Constanta, Romania, May 25–27,* edited by M. Ieremia.

18. A. T. Patera, *J. Comput. Phys.* **54,** 468 (1984).

19. C. A. J. Fletcher, *Comput. Methods Appl. Mech. Engrg.* **37,** 225 (1983).

20. Y. Maday and A. T. Patera, "Spectral Element Methods for the Incompressible Navier–Stokes Equations," in Noor, A. K., editor, *State of the Art Surveys in Computational Mechanics,* edited by A. K. Noor (ASME, New York, 1989), p. 71.

21. M. Kadioglu and S. Mudtrich, *J. Comput. Phys.* **102,** 348 (1992).

22. P. M. Gresho, S. T. Chan, R. L. Lee, and C. D. Upson, *Int. J. Comput. Methods in Fluids* **4,** 557 (1984).

23. P. M. Gresho, S. T. Chan, R. L. Lee, and C. D. Upson, *Int. J. Comput. Methods in Fluids* **4,** 619 (1984).

24. U. Navert, Ph.D. thesis, Goteborg University, Goteborg, Sweden, 1982 (unpublished).

25. *Fluent—User's Guide*, Technical report, Fluent Inc., June 1993.

26. E. Moses, Ph.D. thesis, Technion—Israel Institute of Technology, Haifa, Israel, 1994.

27. E. Moses, A. L. Yarin, and P. Bar-Yoseph, *Combust. & A Flame*, to appear.

28. D. Aharoni and P. Bar-Yoseph, *Comput. Mech.* **9,** 359 (1992).

29. P. Bar-Yoseph, D. Elata, and M. Israeli, *Int. J. Numer. Methods Engrg.* **36,** 679 (1993).