COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 87 (1991) 35–58
NORTH-HOLLAND

A NEW FINITE ELEMENT FORMULATION FOR COMPUTATIONAL FLUID DYNAMICS: IX. FOURIER ANALYSIS OF SPACE–TIME GALERKIN/LEAST-SQUARES ALGORITHMS*

Farzin SHAKIB
Ames Research Center, Moffett Field, CA 94035-4000, USA

Thomas J.R. HUGHES
Division of Applied Mechanics, Durand Building, Stanford University, Stanford, CA 94305, USA

Received 14 January 1990
Revised manuscript received 19 March 1990 and 17 May 1990

A Fourier stability and accuracy analysis of the space–time Galerkin/least-squares method as applied to a time-dependent advective-diffusive model problem is presented. Two time discretizations are studied: a constant-in-time approximation and a linear-in-time approximation. Corresponding space–time predictor multi-corrector algorithms are also derived and studied. The behavior of the space–time algorithms is compared to algorithms based on semidiscrete formulations.

1. Introduction

The space–time Galerkin/least-squares method for the solution of advective-diffusive systems has been presented in [1–3]. This method is based on the time-discontinuous Galerkin formulation, to which a least-squares operator is added. The time-discontinuous Galerkin method generates a complete space–time finite element discretization, eliminating the need for an additional ordinary differential equation solver to discretize time. In addition, it yields a time marching algorithm with a traditional step-by-step format. The mathematical properties of the time-discontinuous Galerkin method for unsteady problems are completely analogous to those of the Galerkin method for the steady case, as observed by Johnson et al. [4]. This is important since the construction of a good method based on the Galerkin method for steady problems then may be readily extended to unsteady problems.

The Galerkin method and the time-discontinuous Galerkin method lack sufficient stability. A manifestation of this lack of stability is spurious oscillations generated by unresolved internal and boundary layers. The added operator of the Galerkin/least-squares method improves stability while maintaining accuracy. The methodology is similar to SUPG (see [5, 6]), however, it is conceptually more straightforward and simplifies mathematical analysis.

An attribute of the space–time Galerkin/least-squares method is its rich mathematical

* This research was sponsored by the NASA Langley Research Center under Grant NASA-NAG-1-361 and the IBM Almaden Research Center under Grant No. 604912.

structure. Johnson et al. [4] obtained finite element error estimates for linear scalar advective-diffusive equations and first-order symmetric hyperbolic systems. Hughes et al. [1] (and Hughes et al. [7] using SUPG) established finite element error estimates for linear symmetric advective-diffusive systems. Johnson et al. [8–11] and Szepessy [12] presented convergence proofs for nonlinear hyperbolic conservation laws. All these works employ functional analysis techniques and the results are applicable to unstructured space–time meshes. Nävert [13] presented a Fourier analysis for a time-dependent scalar advective equation on a uniform piecewise linear space–time triangular mesh. Fourier analysis is applicable only to uniformly structured space–time meshes.

In this paper we present Fourier stability and accuracy analyses of the space–time Galerkin/least-squares method applied to a time-dependent advective-diffusive model problem. The intention here is to gain insight into the behavior of the method for a particular discretization, and to develop efficient predictor multi-corrector approximations which retain the good stability and accuracy properties of the underlying space–time methods. For application of these algorithms to the compressible Euler and Navier–Stokes equations, see [3, 14].

The paper is outlined as follows. In Section 2, the time-dependent model problem is presented. The space–time Galerkin/least-squares method is described in Section 3. A constant-in-time approximation and associated predictor multi-corrector algorithm are developed and analyzed in Section 4. In Section 5, a linear-in-time approximation and associated predictor multi-corrector algorithm are developed and analyzed. In Section 6, the space–time algorithms are compared with algorithms emanating from semidiscrete formulations. The results are summarized in Section 7, and conclusions are drawn in Section 8.

2. Time-dependent advective-diffusive model problem

The linear scalar time-dependent advective-diffusive model problem is

$$u_{,t} + a u_{,x} = \kappa u_{,xx} \quad \text{for } x \in]0, L[,\ t > 0\,, \tag{1}$$

$$u(x, t) = u_0(x) \quad \text{at } t = 0\,, \tag{2}$$

where u is the scalar unknown, a is the advective speed, and κ is the coefficient of diffusivity; a and κ are assumed constant. Without loss of generality, we assume that a is positive. We further assume the following periodic boundary conditions:

$$u(L, t) = u(0, t)\,, \tag{3}$$

$$u_{,x}(L, t) = u_{,x}(0, t)\,. \tag{4}$$

These enable us to express the exact solution in a Fourier series, a typical term having the form

$$u = \mathrm{e}^{\nu t + \mathrm{i}Kx}\,, \tag{5}$$

where K is the spatial wave number; ν determines the temporal evolution of the solution and may be written as

$$\nu = -\xi + \mathrm{i}\omega\,, \tag{6}$$

where ξ is the damping coefficient, ω is the frequency and $\mathrm{i} = \sqrt{-1}$. Substituting (5) and (6) in the model equation (1) yields

$$\xi = \kappa K^2\,, \tag{7}$$

$$\omega = -aK\,. \tag{8}$$

We note that the solution is stable for $\kappa \geq 0$. We will also require the numerical solution to be stable under this condition.

3. Space–time Galerkin/least-squares method

3.1. Finite element space

Consider the partition $0 = t_0 < t_1 < \cdots < t_N = T$ of the time interval $I =]0, T[$. Denote by $I_n =]t_n, t_{n+1}[$ the nth time interval. Clearly $I = \cup_{n=0}^{N-1} I_n \cup \{t_1, t_2, \ldots, t_{N-1}\}$. A space–time slab is then defined as

$$Q_n = \Omega \times I_n\,, \tag{9}$$

where Ω $(=]0, L[)$ denotes the spatial domain; see Fig. 1.

For the nth space–time slab, let the spatial domain be subdivided into n_{el} elements, Ω^e, $e = 1, \ldots, n_{\mathrm{el}}$. We define space–time element domains as

$$Q_n^e = \Omega^e \times I_n\,, \quad e = 1, \ldots, n_{\mathrm{el}}\,. \tag{10}$$

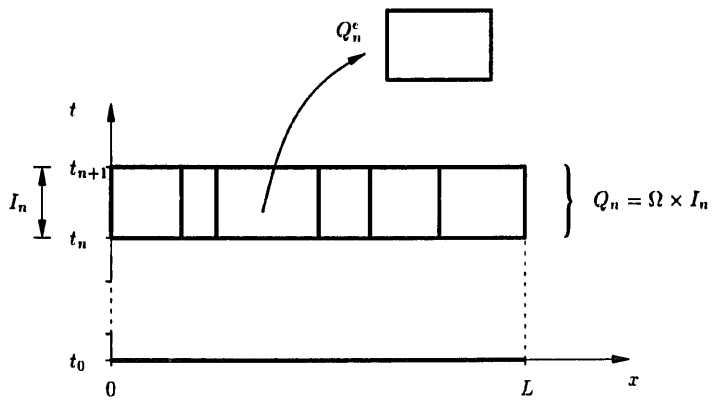

Fig. 1. A space–time slab.

Within each space–time element the trial solution and weighting function are approximated by kth-order polynomials. These functions are assumed $\mathscr{C}^0$ continuous throughout each space–time slab, but are allowed to be discontinuous across the interfaces of the slabs, namely at times $t_1, t_2, \ldots, t_{N-1}$. The finite element spaces are defined as follows:
trial functions:

$$\mathscr{S}^h = \bigcup_{n=0}^{N-1} \mathscr{S}_n^h , \tag{11}$$

$$\mathscr{S}_n^h = \{u^h \mid u^h \in \mathscr{C}^0(Q_n),\ u^h|_{Q_n^e} \in \mathscr{P}_k(Q_n^e),\ u^h(L,\cdot) = u^h(0,\cdot)\} ; \tag{12}$$

weighting functions:

$$\mathscr{V}^h = \bigcup_{n=0}^{N-1} \mathscr{V}_n^h , \tag{13}$$

$$\mathscr{V}_n^h = \{w^h \mid w^h \in \mathscr{C}^0(Q_n),\ w^h|_{Q_n^e} \in \mathscr{P}_k(Q_n^e),\ w^h(L,\cdot) = w^h(0,\cdot)\} . \tag{14}$$

The superscript h represents the space–time mesh parameter (e.g., a suitable definition of h is $h = \max_{e,n}\{\Delta x^e, a\Delta t_n, (\kappa\,\Delta t_n)^{1/2}\}$, where $\Delta x^e = \text{meas}(\Omega^e)$ and $\Delta t_n = t_{n+1} - t_n$) and $\mathscr{P}_k$ denotes the space of complete kth order polynomials. Note that for the case under consideration $\mathscr{S}_n^h = \mathscr{V}_n^h$. Considering that the finite element functions are discontinuous at the space–time slab interfaces, it is useful to employ the notation

$$w^h(t_n^{\pm}) = \lim_{\varepsilon \to 0^{\pm}} w^h(t_n + \varepsilon) . \tag{15}$$

3.2. Weighted residual formulation

The space–time Galerkin/least-squares weighted residual formulation of (1) is: For $n = 0, 1, \ldots, N-1$, find $u^h \in \mathscr{S}_n^h$ such that for all $w^h \in \mathscr{V}_n^h$

$$\int_{I_n}\int_{\Omega} (w^h u_{,t}^h + w^h a u_{,x}^h + w_{,x}^h \kappa u_{,x}^h)\,\mathrm{d}x\,\mathrm{d}t + \int_{\Omega} w^h(t_n^+)(u^h(t_n^+) - u^h(t_n^-))\,\mathrm{d}x$$
$$+ \sum_{e=1}^{n_{\text{el}}} \int_{I_n}\int_{\Omega^e} (w_{,t}^h + a w_{,x}^h - \kappa w_{,xx}^h)\tau(u_{,t}^h + a u_{,x}^h - \kappa u_{,xx}^h)\,\mathrm{d}x\,\mathrm{d}t = 0 , \tag{16}$$

where

$$u^h(t_0^-) = u_0 . \tag{17}$$

The first two integrals in (16) constitute the time-discontinuous Galerkin formulation. The first integral is the usual Galerkin integral and the second integral is a jump condition. The jump condition imposes a weakly enforced continuity condition across the slab interfaces and is the mechanism by which information is propagated from one slab to another.

The last integral in (16) is the least-squares operator, where τ is the least-squares metric. On a uniform space–time mesh (i.e., one in which $\Delta x^e = \Delta x$, $e = 1, \ldots, n_{\text{el}}$, and $\Delta t_n = \Delta t$,

$n = 0, 1, \ldots, N-1$), τ is defined by (see [3])

$$\tau = \left(\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2a}{\Delta x} \right)^2 + 9 \left(\frac{4\kappa}{\Delta x^2} \right)^2 \right)^{-1/2} . \tag{18}$$

When the transient problem (1) is solved to obtain a steady solution, we use in place of τ,

$$\tau_{\text{space}} = \left(\left(\frac{2a}{\Delta x} \right)^2 + 9 \left(\frac{4\kappa}{\Delta x^2} \right)^2 \right)^{-1/2} . \tag{19}$$

Note that τ_{space} is the limit of τ as $\Delta t \to \infty$.

The variational equation is 'consistent', in the sense that the exact solution of (1) satisfies (16). Moreover, (16) is L_2-stable. This can be seen by replacing w^h by u^h in (16), and summing the results over the N space–time slabs, viz.,

$$\frac{1}{2} \| u^h(t_N^-) \|_\Omega^2 + \kappa \| u_{,x}^h \|_Q^2 + \frac{1}{2} \sum_{n=0}^{N-1} \| u(t_n^+) - u(t_n^-) \|_\Omega^2 + \sum_{n=0}^{N-1} \sum_{e=1}^{n_{\text{el}}} \| \sqrt{\tau} (u_{,t}^h + a u_{,x}^h - \kappa u_{,xx}^h) \|_{Q_n^e}^2 = \frac{1}{2} \| u^h(t_0^-) \|_\Omega^2 . \tag{20}$$

Note that stability is enhanced by the presence of the diffusion term, the jump condition, and the least-squares term.

Let $\alpha = (a \Delta x)/(2\kappa)$ denote the element Peclet number. We refer to the case $\alpha > 1$ as the 'advection dominated case', and $\alpha < 1$ as the 'diffusion dominated case'. Denote the error in the finite element solution by $e = u^h - u$. It can be shown that after N time steps the error satisfies

$$\frac{1}{2} \| e(t_N^-) \|_\Omega^2 + \kappa \| e_{,x} \|_Q^2 + \frac{1}{2} \sum_{n=0}^{N-1} \| e(t_n^+) - e(t_n^-) \|_\Omega^2 + \sum_{n=0}^{N-1} \sum_{e=1}^{n_{\text{el}}} \| \sqrt{\tau} (e_{,t} + a e_{,x} - \kappa e_{,xx}) \|_{Q_n^e}^2 \leq c h^\beta \| u \|_{k+1,Q}^2 , \tag{21}$$

where c is independent of h, and

$$\beta = \begin{cases} 2k+1 , & \text{advection dominated case} , \\ 2k , & \text{diffusion dominated case} . \end{cases} \tag{22}$$

4. Constant-in-time approximation

4.1. Discretization

In this approximation, the u^h and w^h interpolations are linear in space and constant in time. We assume a uniform space–time mesh. In the nth space–time slab, the finite element functions are

$$u^h(x,t) = \sum_{j=1}^{n_{\text{np}}} N_j(x) u^h_{j;(n+1)} \quad \text{for } x \in \Omega,\ t \in I_n\,, \tag{23}$$

$$w^h(x,t) = \sum_{j=1}^{n_{\text{np}}} N_j(x) w^h_{j;(n+1)} \quad \text{for } x \in \Omega,\ t \in I_n\,, \tag{24}$$

where $u^h_{j;(n+1)}$ is the nodal value of u^h for spatial node j at time t_{n+1}, as shown in Fig. 2. $w^h_{j;(n+1)}$ is the corresponding nodal value of the weighting function; n_{np} is the number of spatial nodal points ($n_{\text{np}} = n_{\text{el}} + 1$), and $N_j(x)$ is the spatial shape function for node j, assumed to be piecewise linear.

Substituting (23) and (24) into the variational equation (16) yields

$$0 = \sum_{i=1}^{n_{\text{np}}} w^h_{i;(n+1)} \Bigg\{ \Delta t \int_0^L \left(N_i a \sum_{j=1}^{n_{\text{np}}} N_{j,x} u^h_{j;(n+1)} + N_{i,x} \kappa \sum_{j=1}^{n_{\text{np}}} N_{j,x} u^h_{j;(n+1)} \right) \mathrm{d}x$$
$$+ \int_0^L N_i \left(\sum_{j=1}^{n_{\text{np}}} N_j u^h_{j;(n+1)} - \sum_{j=1}^{n_{\text{np}}} N_j u^h_{j;(n)} \right) \mathrm{d}x + \Delta t \int_0^L \left(a N_{i,x} \tau a \sum_{j=1}^{n_{\text{np}}} N_{j,x} u^h_{j;(n+1)} \right) \mathrm{d}x \Bigg\}\,. \tag{25}$$

This equation can be written in the form

$$\sum_{i=1}^{n_{\text{np}}} w^h_{i;(n+1)} G_i(\boldsymbol{u}^h_{(n+1)}, \boldsymbol{u}^h_{(n)}) = 0\,, \tag{26}$$

where

$$\boldsymbol{u}^h_{(n+1)} \overset{\text{def}}{=} \{u^h_{1;(n+1)}, \ldots, u^h_{n_{\text{np}};(n+1)}\}^{\text{t}}\,, \tag{27}$$

$$\boldsymbol{u}^h_{(n)} \overset{\text{def}}{=} \{u^h_{1;(n)}, \ldots, u^h_{n_{\text{np}};(n)}\}^{\text{t}} \tag{28}$$

and the G_i's are linear with respect to $\{\boldsymbol{u}^h_{(n+1)}, \boldsymbol{u}^h_{(n)}\}$. Equation (26) holds for all $w^h_{i;(n+1)} \in \mathscr{R}$, $i = 1, \ldots, n_{\text{np}} - 1$, with $w^h_{n_{\text{np}};(n+1)} = w^h_{1;(n+1)}$. This leads to

$$G_i(\boldsymbol{u}^h_{(n+1)}, \boldsymbol{u}^h_{(n)}) = 0 \quad \text{for } i = 1, \ldots, n_{\text{np}} - 1\,. \tag{29}$$

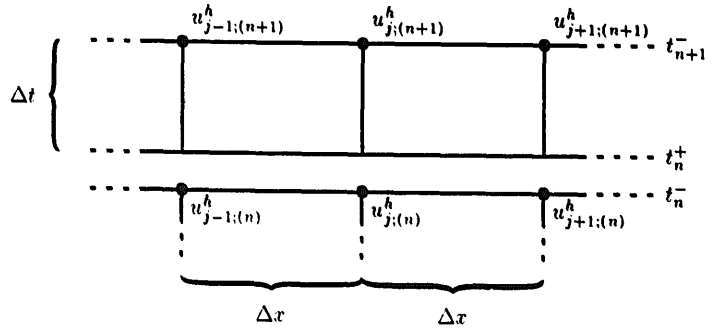

Fig. 2. Nodal configuration of a uniform space–time mesh for the constant-in-time approximation.

These equations comprise a system of linear algebraic equations

$$G(u^h_{(n+1)}, u^h_{(n)}) \overset{\text{def}}{=} \{G_1, G_2, \ldots, G_{n_{\text{np}}-1}\}^{\text{t}} = \mathbf{0}\,, \tag{30}$$

having $n_{\text{np}} - 1$ equations and $n_{\text{np}} - 1$ unknowns, namely the components of $u^h_{(n+1)}$.

From the finite difference perspective, G_i gives rise to a three-point spatial difference stencil. The stencil centered around the internal node j is

$$\begin{aligned}
&\left(\frac{\Delta x}{6} - \kappa\,\frac{\Delta t}{\Delta x} - a\,\frac{\Delta t}{2} - a^2\tau\,\frac{\Delta t}{\Delta x}\right)u^h_{j-1;(n+1)} + \left(4\,\frac{\Delta x}{6} + 2\kappa\,\frac{\Delta t}{\Delta x} + 2a^2\tau\,\frac{\Delta t}{\Delta x}\right)u^h_{j;(n+1)} \\
&\quad + \left(\frac{\Delta x}{6} - \kappa\,\frac{\Delta t}{\Delta x} + a\,\frac{\Delta t}{2} - a^2\tau\,\frac{\Delta t}{\Delta x}\right)u^h_{j+1;(n+1)} \\
&\quad = \frac{\Delta x}{6}\,u^h_{j-1;(n)} + 4\,\frac{\Delta x}{6}\,u^h_{j;(n)} + \frac{\Delta x}{6}\,u^h_{j+1;(n)}\,.
\end{aligned} \tag{31}$$

This is the algorithmic equation for a typical internal node. The algorithmic equations for the end points (i.e., $j = 1$ and $j = n_{\text{np}}$) are identical, due to the periodic boundary conditions.

REMARK 1. In a semidiscrete Galerkin formulation the time-discretized form of $u^h_{,t}$ provides a mechanism for propagating information from one time step to the next. Since the interpolation functions are assumed constant in time, this term vanishes in the space–time formulation (25). The propagation of information is, however, provided by the jump term. The difference stencil resulting from the jump term has the same form as the time-derivative term in a typical semidiscrete Galerkin formulation.

REMARK 2. Since the interpolation functions are assumed piecewise linear in space, the second derivatives $w^h_{,xx}$ and $u^h_{,xx}$ are zero on element interiors. In this case, the Galerkin/least-squares formulation results in the same equations as the SUPG formulation (see [1]).

4.2. *Stability and accuracy analysis*

To analyze the stability and accuracy of (31), we assume that the discrete solution takes on the form

$$u^h_{j;(n)} = \mathrm{e}^{\nu^h n\,\Delta t + \mathrm{i}Kj\,\Delta x}\,, \tag{32}$$

where ν^h is the discrete counterpart of ν. Similar to (6), ν^h is written in the form

$$\nu^h = -\xi^h + \mathrm{i}\omega^h\,, \tag{33}$$

where ξ^h is the algorithmic damping coefficient and ω^h is the algorithmic frequency.

For convenience, we introduce the so-called amplification factor,

$$\zeta^h \overset{\text{def}}{=} \mathrm{e}^{\nu^h\,\Delta t}\,, \tag{34}$$

and the dimensionless wave number,

$$\bar{K} \stackrel{\text{def}}{=} \Delta x\, K\,. \tag{35}$$

$\bar{K}$ is a measure of the spatial mesh refinement. The numerical method can resolve Fourier components up to $\bar{K} = \pi$, which corresponds to two elements per wavelength. For accuracy, we are interested in Fourier components resolved by at least ten elements per wavelength, corresponding to $\bar{K} \simeq 0.6$. For stability we are interested in the entire resolvable range, namely $\bar{K} \in [0, \pi]$.

Substituting the above relations into (31) leads to

$$\zeta^h((\tfrac{2}{3} + \tfrac{1}{3}\cos(\bar{K})) + (2 - 2\cos(\bar{K}))(\bar{\kappa} + \bar{a}^2\bar{\tau})) + \mathrm{i}\zeta^h \bar{a}\sin(\bar{K}) = \tfrac{2}{3} + \tfrac{1}{3}\cos(\bar{K})\,, \tag{36}$$

where $\bar{a} = a\,\Delta t/\Delta x$, $\bar{\kappa} = \kappa\,\Delta t/\Delta x^2 \geqslant 0$, and $\bar{\tau} = \tau/\Delta t \geqslant 0$ are dimensionless quantities. Equation (36) relates the amplification factor to the dimensionless wave number. The stability condition is $|\zeta^h| \leqslant 1$ for all $\bar{K} \in [0, \pi]$. Manipulating (36) leads to an explicit expression for the amplification factor:

$$|\zeta^h|^2 = \frac{(2 + \cos(\bar{K}))^2}{9\sin^2(\bar{K})\bar{a}^2 + ((2 + \cos(\bar{K})) + 6(1 - \cos(\bar{K}))(\bar{\kappa} + \bar{a}^2\bar{\tau}))^2}\,. \tag{37}$$

Clearly, $|\zeta^h| \leqslant 1$, which establishes the unconditional stability of the algorithm. This confirms the previously obtained stability result, namely (20).

To analyze the accuracy of (31), we substitute (33) and (34) into (36) and then expand ξ^h and ω^h in powers of Δt, Δx and τ. This results in

$$\xi^h = \kappa K^2 + (a^2K^2/2 - \kappa^2K^4/2)\,\Delta t + (a^2K^2)\tau + (\kappa K^4/12)\,\Delta x^2 + \mathrm{O}(\Delta t^2, \tau^2, \Delta x^4)\,, \tag{38}$$

$$\omega^h = -aK + (a\kappa K^3)\,\Delta t + (a^3K^3)\,\Delta t\,\tau + (aK^5/180)\,\Delta x^4 + \mathrm{O}(\Delta t^2, \Delta t\,\tau^2, \Delta x^6)\,. \tag{39}$$

Comparing these with ξ and ω for the exact solution (i.e., (7) and (8)) shows that the method is first-order accurate with respect to Δt. For $a \neq 0$, ω^h is fourth-order accurate with respect to Δx, and for $a = 0$ there is no frequency error. The least-squares term has a weak effect on the frequency. This can be seen from the dependency of ω^h on $\Delta t\,\tau$ (as opposed to τ as in the expression for ξ^h). In the advection-dominated case, where $\tau = \mathrm{O}(\Delta t, \Delta x)$, ξ^h is first-order with respect to Δx. In the diffusion-dominated case, where $\tau = \mathrm{O}(\Delta t, \Delta x^2)$, ξ^h improves to second-order with respect to Δx. Note that for the Galerkin method for pure advection (i.e., $\tau = \kappa = 0$), ξ^h is fourth-order with respect to Δx. However, it is well-known that this method often exhibits high-frequency spatial oscillations.

For the pure advection problem (i.e., $\kappa = 0$), the algorithmic damping ratio, ξ^h/ω^h, and the algorithmic frequency ratio, ω^h/ω, are computed for $\tau = 0$ and $\tau = \tau_{\text{space}}$, Courant numbers $C_{\Delta t} \stackrel{\text{def}}{=} \Delta t\, a/\Delta x = 0.2, 0.4, 0.6, 0.8$ and 1.0, and $\bar{K} \in [0, \pi]$. The algorithmic damping ratio is a measure of numerical dissipation error, and the algorithmic frequency ratio is a measure of numerical dispersion, or phase, error. These results are plotted in Fig. 3. Note that the

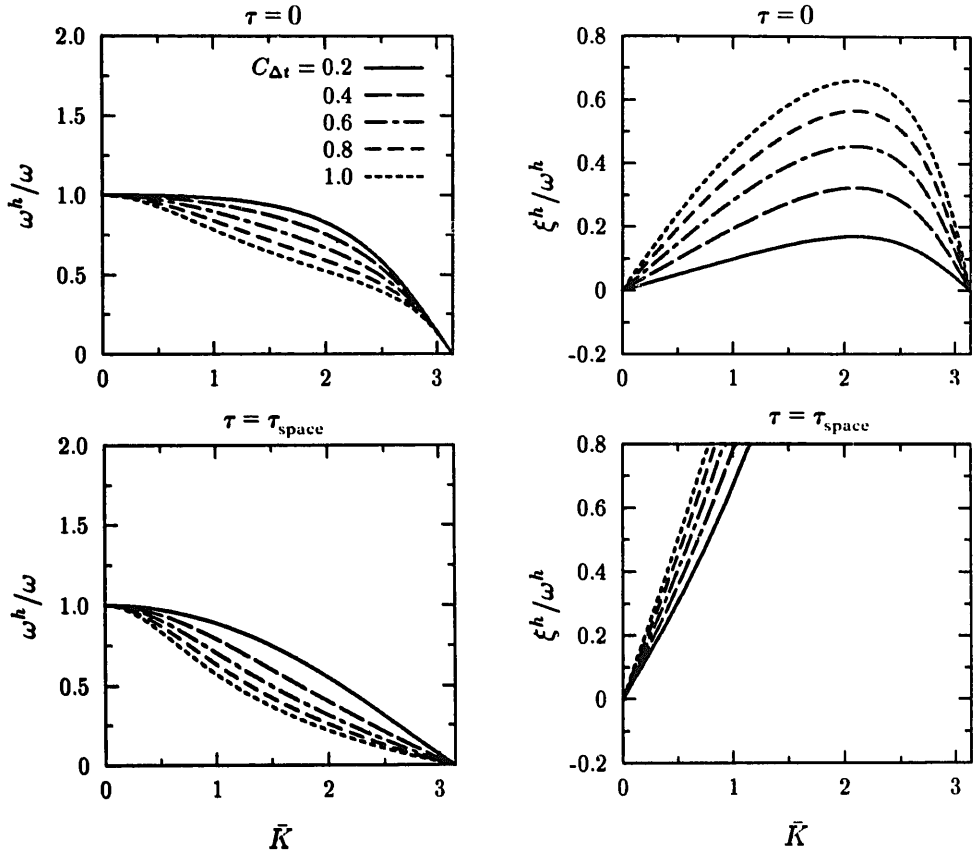

Fig. 3. Pure advection problem. Algorithmic frequency and damping ratios for the constant-in-time approximation.

Galerkin method ($\tau = 0$) has zero damping at $\bar{K} = \pi$. This is indicative of potential node-to-node oscillatory behavior. In contrast, the Galerkin/least-squares method (i.e., $\tau = \tau_{\text{space}}$) has strong damping, which renders this method very stable and excellent for obtaining steady solutions.

For an advection-diffusion problem, the algorithmic damping ratio, ξ^h/ξ (N.B.: this definition is different from the one for the pure advection problem), and the algorithmic frequency ratio, ω^h/ω, are computed and plotted in Fig. 4. To see the effect of mesh refinement, the parameters a, κ and K are kept constant and Δx is varied linearly from 0 to π/K. The values of the fixed parameters are chosen such that the element Peclet number is equal to $\bar{K}$. These results show that the Galerkin method is underdamped for large values of $\bar{K}$, as opposed to the Galerkin/least-squares method which exhibits strong damping. On the other hand, the least-squares term has little effect on the frequency ratio throughout the entire range of $\bar{K}$.

The temporal accuracy for the constant-in-time approximation is first-order as expected, while the spatial accuracy of the Galerkin/least-squares method appears low, due to the presence of the τ term in (38). The modified equation approach of Warming and Hyett [15] also exhibits a similar dependence upon τ in the local truncation error, namely, for (31) the computed error is

$$\varepsilon = (-\tfrac{1}{2}u_{,tt})\,\Delta t + (-a^2 u_{,xx})\tau + (\tfrac{1}{12}\kappa u_{,xxxx})\,\Delta x^2 + O(\Delta t^2, \tau^2, \Delta x^4)\,. \tag{40}$$

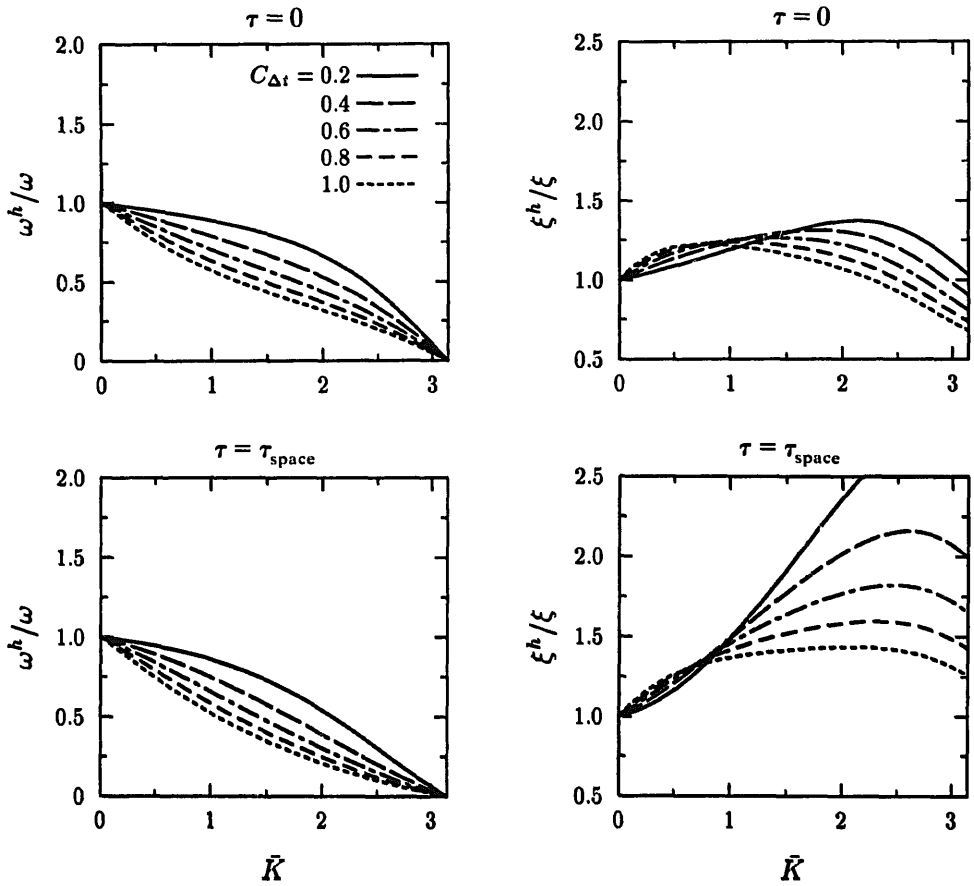

Fig. 4. Advection-diffusion problem ($\alpha = \bar{K}$). Algorithmic frequency and damping ratios for the constant-in-time approximation.

In the case of the steady model problem the computed error is

$$\varepsilon = (-a^2 u_{,xx})\tau + (\tfrac{1}{12}\kappa u_{,xxxx})\,\Delta x^2 + O(\tau^2, \Delta x^4)\,. \tag{41}$$

However, this result is known to be pessimistic in that it is shown in [3] that for the steady case the method is higher-order accurate, even nodally exact for appropriately defined τ.

4.3. *Predictor multi-corrector algorithms*

In this section we develop a predictor multi-corrector implementation of the linear algebraic problem (30). The idea is to replace (30) with an iterative procedure whose solution after a limited number of iterations exhibits similar accuracy properties to the solution of (30) (i.e., exhibits the same order of accuracy with the same leading term coefficients). Let $\boldsymbol{u}^{(i)}$ denote the ith iterative approximation of $\boldsymbol{u}^h_{(n+1)}$. Due to the linearity of $\boldsymbol{G}$ with respect to its first argument, we can write

$$\mathbf{0} = \boldsymbol{G}(\boldsymbol{u}^h_{(n+1)}, \boldsymbol{u}^h_{(n)}) = \boldsymbol{G}(\boldsymbol{u}^{(i)}, \boldsymbol{u}^h_{(n)}) + \frac{\partial \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{u}^h_{(n)})}{\partial \boldsymbol{u}}\,\Delta \boldsymbol{u}^{(i)}\,, \tag{42}$$

where

$$\Delta \boldsymbol{u}^{(i)} = \boldsymbol{u}^{(i+1)} - \boldsymbol{u}^{(i)} . \tag{43}$$

Let

$$\boldsymbol{M} = \frac{\partial \boldsymbol{G}(\boldsymbol{u}, \boldsymbol{u}^h_{(n)})}{\partial \boldsymbol{u}} , \tag{44}$$

$$\boldsymbol{R}^{(i)} = \boldsymbol{G}(\boldsymbol{u}^{(i)}, \boldsymbol{u}^h_{(n)}) . \tag{45}$$

These are referred to as the consistent 'mass' matrix and residual vector, respectively. With these notations (42) can be written as

$$\boldsymbol{M}\, \Delta \boldsymbol{u}^{(i)} = -\boldsymbol{R}^{(i)} . \tag{46}$$

The predictor multi-corrector algorithm is

ALGORITHM 4.1. (Predictor multi-corrector algorithm based on the constant-in-time approximation)

(Predictor phase)

$\boldsymbol{u}^{(0)} = \boldsymbol{u}^h_{(n)}$

(Multi-corrector phase)

For $i = 0, 1, \ldots, i_{\max} - 1$

$\boldsymbol{M}^* \Delta \boldsymbol{u}^{(i)} = -\boldsymbol{R}^{(i)}$

$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \Delta \boldsymbol{u}^{(i)}$

$\boldsymbol{u}^h_{(n+1)} = \boldsymbol{u}^{(i_{\max})}$

where $i_{\max}$ is the number of corrector passes and $\boldsymbol{M}^*$ is an 'approximation' of $\boldsymbol{M}$.

This algorithm is residual driven. In general, $\boldsymbol{M}^*$ is designed based on considerations of stability, convergence rate, and computational convenience. The parameter $i_{\max}$ is selected to achieve sufficient accuracy. Here we consider two definitions for $\boldsymbol{M}^*$:

Implicit:

In this case, we set $\boldsymbol{M}^* = \boldsymbol{M}$. The solution of the original algebraic system is obtained after the first corrector pass (i.e., $\boldsymbol{R}^{(1)} = \boldsymbol{0}$).

Explicit:

In this case, we set $\boldsymbol{M}^* = \Delta x \mathbf{I}$, corresponding to using nodal quadrature on the jump term. Obviously, the inverse of $\boldsymbol{M}^*$ is trivially obtained. Contrary to the implicit case, the solution of the algebraic problem $\boldsymbol{G}$ is not obtained after a finite number of passes. This algorithm is also only conditionally stable. The square of the amplification factor for the one corrector pass (i.e., $i_{\max} = 1$) is

$$|\zeta^h|^2 = \bar{a}^2 \sin^2(\bar{K}) + (1 - 4\sin^2(\bar{K}/2)(\bar{\kappa} + \bar{a}^2\bar{\tau}))^2 , \tag{47}$$

which leads to the following stability conditions:

$$\tfrac{1}{2}\bar{a} \leqslant \bar{\kappa} + \bar{a}^2\bar{\tau} \leqslant \tfrac{1}{2} . \tag{48}$$

The first inequality defines a lower limit for $\bar{\tau}$. For the pure advection problem, this requires $\tau \geqslant \Delta x/2a$, where equality is achieved precisely by the definition of τ_{space}. The second inequality suggests a definition of an algorithmic Courant number:

$$C_\tau \overset{\text{def}}{=} 2\bar{\kappa} + 2\bar{a}^2\bar{\tau} = 2\kappa \frac{\Delta t}{\Delta x^2} + 2a^2\tau \frac{\Delta t}{\Delta x^2} . \tag{49}$$

Hence, for stability,

$$C_\tau \leqslant 1 . \tag{50}$$

The expansions of ξ^h and ω^h for one corrector pass are

$$\xi^h = \kappa K^2 + (-\tfrac{1}{2}a^2K^2 + \tfrac{1}{2}\kappa^2K^4)\,\Delta t + (a^2K^2)\tau + (\tfrac{1}{12}\kappa K^4)\,\Delta x^2 + O(\Delta t^2, \tau^2, \Delta x^4) , \tag{51}$$

$$\omega^h = -aK + (-a\kappa K^3)\,\Delta t + (-a^3K^3)\,\Delta t\,\tau + (\tfrac{1}{6}aK^3)\,\Delta x^2 + O(\Delta t^2, \Delta t\,\tau^2, \Delta x^4) . \tag{52}$$

These show that for the order of accuracy of the underlying method (31) is *not* attained by one corrector pass; cf. (38) and (39). The expansion for two corrector passes (i.e., $i_{\max} = 2$) are

$$\xi^h = \kappa K^2 + (a^2K^2/2 - \kappa^2K^4/2)\,\Delta t + (a^2K^2)\tau + (\kappa K^4/12)\,\Delta x^2 + O(\Delta t^2, \tau^2, \Delta x^4) , \tag{53}$$

$$\omega^h = -aK + (a\kappa K^3)\,\Delta t + (a^3K^3)\,\Delta t\,\tau + (aK^5/180)\,\Delta x^4 + O(\Delta t^2, \Delta t\,\tau^2, \Delta x^6) , \tag{54}$$

which coincide with those of the underlying method; cf. (38) and (39). The present method is generally only used to obtain steady solutions. In these circumstances there is little motivation to perform more than one corrector pass.

5. Linear-in-time approximation

5.1. Discretization

In this approximation, the u^h and w^h interpolations are bilinear on each space–time element. As in the constant-in-time approximation, the spatial domain is uniformly meshed with spacing Δx. Recall, n_{np} is the number of spatial nodal points. The total number of nodal points for each space–time slab is then $2n_{\text{np}}$: n_{np} nodal points at t_{n+1}^-, with nodal values denoted by $u^h_{j;(n+1)}$, and n_{np} nodal points at t_n^+, with nodal values denoted by $\tilde{u}^h_{j;(n)}$; see Fig. 5.

We refer to $u^h_{j;(n+1)}$ as the primary variables and to $\tilde{u}^h_{j;(n)}$ as the secondary variables. For the nth space–time slab we have

$$u^h(x,t) = \sum_{j=1}^{n_{\text{np}}} N_j(x)(\pi_n(t)u^h_{j;(n+1)} + \tilde{\pi}_n(t)\tilde{u}^h_{j;(n)}) \quad \text{for } x \in \Omega,\ t \in I_n , \tag{55}$$

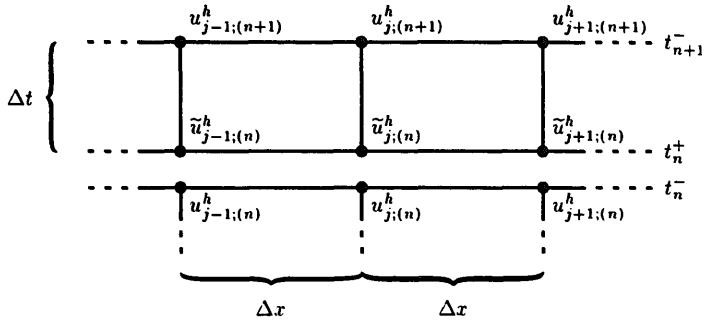

Fig. 5. Nodal configuration of a uniform mesh for the linear-in-time approximation.

$$w^h(x,t)=\sum_{j=1}^{n_{\mathrm{np}}} N_j(x)(\pi_n(t)w^h_{j;(n+1)}+\tilde{\pi}_n(t)\tilde{w}^h_{j;(n)}) \quad \text{for } x\in\Omega,\ t\in I_n\,, \tag{56}$$

where, as before, $N_j(x)$ is the piecewise linear shape function of spatial node j and $\pi_n(t)$ and $\tilde{\pi}_n(t)$ are the linear temporal shape functions defined by

$$\pi_n(t)=(t-t_n)/\Delta t\,, \tag{57}$$

$$\tilde{\pi}_n(t)=(t_{n+1}-t)/\Delta t\,. \tag{58}$$

Substitution into the variational equation (16) leads to the following three-point in space, two-point in time, difference stencil:

$$\begin{aligned}
&\frac{\Delta x}{12}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\left(\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+4\begin{Bmatrix}u^h_{j;(n+1)}\\ \tilde{u}^h_{j;(n)}\end{Bmatrix}+\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&+\kappa\frac{\Delta t}{6\Delta x}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\left(-\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+2\begin{Bmatrix}u^h_{j;(n+1)}\\ \tilde{u}^h_{j;(n)}\end{Bmatrix}-\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&+a\frac{\Delta t}{12}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\left(-\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&+a^2\tau\frac{\Delta t}{6\Delta x}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\left(-\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+2\begin{Bmatrix}u^h_{j;(n+1)}\\ \tilde{u}^h_{j;(n)}\end{Bmatrix}-\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&+\frac{a\tau}{2}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\left(-\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&+\tau\frac{\Delta x}{6\Delta t}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\left(\begin{Bmatrix}u^h_{j-1;(n+1)}\\ \tilde{u}^h_{j-1;(n)}\end{Bmatrix}+4\begin{Bmatrix}u^h_{j;(n+1)}\\ \tilde{u}^h_{j;(n)}\end{Bmatrix}+\begin{Bmatrix}u^h_{j+1;(n+1)}\\ \tilde{u}^h_{j+1;(n)}\end{Bmatrix}\right)\\
&=\frac{\Delta x}{6}\begin{Bmatrix}0\\ u^h_{j-1;(n)}\end{Bmatrix}+4\frac{\Delta x}{6}\begin{Bmatrix}0\\ u^h_{j;(n)}\end{Bmatrix}+\frac{\Delta x}{6}\begin{Bmatrix}0\\ u^h_{j+1;(n)}\end{Bmatrix}.
\end{aligned} \tag{59}$$

The rows of the 2×2 matrices in this system reveal the temporal coupling between the primary and secondary variables, and the columns reveal the coupling between the first and second algorithmic equations. The first three terms on the left-hand side and right-hand side of (59) derive from the discontinuous Galerkin part of the formulation, while the remaining three terms derive from the least-squares contribution. The temporal coupling in the mass term (i.e., the first 2×2 matrix in (59)) is not symmetric. This matrix is the sum of the matrices contributed by the $w^h u^h_{,t}$-term and the jump condition term,

$$\frac{\Delta x}{12}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad \frac{\Delta x}{6}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

respectively. Each is singular, but their sum is positive-definite.

5.2. *Stability and accuracy analysis*

Equation (59) centered in turn at $j-1$, j and $j+1$ yields six scalar equations with five primary and five secondary unknowns. Eliminating the five secondary unknowns results in a five-point difference stencil centered at j. A symbolic manipulation program, SMP, was used to carry out these operations. The resulting difference stencil is very long and complicated, consequently we will not present it here.

In the limiting cases of pure advection and pure diffusion, the Fourier analysis of the five-point stencil shows that the algorithm is unconditionally stable. This is consistent with the finite element stability result, namely (20), which established stability throughout the entire range of advective-diffusive parameters. The Fourier analysis is not performed in closed form for the combined advective-diffusive case because of the considerable complexity of the equations.

The expansions of ξ^h and ω^h for the pure advection case are

$$\xi^h = \left(\frac{a^2K^4}{72}\right)\Delta t^3 + \left(\frac{a^4K^4}{12}\right)\Delta t^2 \tau + O(\Delta t^4, \Delta t^2 \tau^2, \Delta x^6), \tag{60}$$

$$\omega^h = -aK + (a^5K^5)\,\Delta t^4 + \left(\frac{a^5k^5}{54}\right)\Delta t^3 \tau + \left(\frac{aK^5}{180}\right)\Delta x^4 + O(\Delta t^5, \Delta t^4 \tau, \Delta x^6), \tag{61}$$

and in the pure diffusion case, with $\tau = 0$, they are

$$\xi^h = \kappa K^2 + \left(\frac{\kappa^4 K^8}{72}\right)\Delta t^3 + \left(\frac{\kappa K^4}{12}\right)\Delta x^2 + O(\Delta t^4, \Delta x^4), \tag{62}$$

$$\omega^h = 0. \tag{63}$$

These results show that the method is third-order accurate with respect to Δt.

The two cases studied in Section 4.2 are reconsidered here for the linear-in-time approximation and the results are plotted in Figs. 6 and 7. Since for this method we are interested in time accuracy, the space–time version of τ, (18), was used in the analysis. The main features to be observed in the figures are the high degree of accuracy achieved in the low wave number region and, for the Galerkin/least-squares method, the high degree of dissipation in the high wave number region.

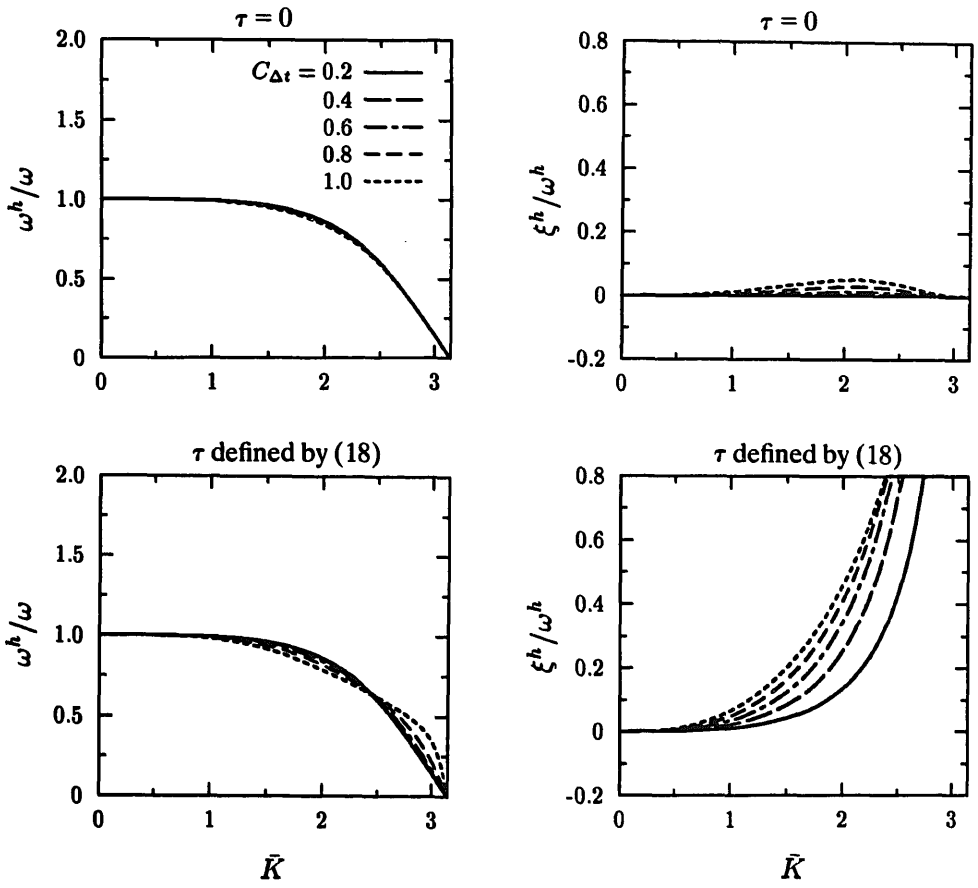

Fig. 6. Pure advection problem. Algorithmic frequency and damping ratios for the linear-in-time approximation.

5.3. *Predictor multi-corrector algorithms*

If we proceed to formulate the predictor multi-corrector algorithm for the linear-in-time case in the same way as for the constant-in-time case we obtain a nonsymmetric linear system having $2(n_{\text{np}}-1)$ equations in $2(n_{\text{np}}-1)$ unknowns, with a bandwidth double that of the constant-in-time case. Moreover, the explicit algorithm based on this process is unconditionally unstable.

To circumvent these shortcomings, we multiply (59) by the matrix

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{64}$$

The transformed equations are arranged such that in the first equation, denoted by $\boldsymbol{R}=\boldsymbol{0}$, the primary variable $\boldsymbol{u}^h_{(n+1)}$ is viewed as the unknown, whereas the secondary variables $\tilde{\boldsymbol{u}}^h_{(n)}$ and the solution for the previous time step $\boldsymbol{u}^h_{(n)}$ are viewed as known; in the second equation, denoted by $\tilde{\boldsymbol{R}}=\boldsymbol{0}$, $\tilde{\boldsymbol{u}}^h_{(n)}$ is viewed as unknown, whereas $\boldsymbol{u}^h_{(n+1)}$ and $\boldsymbol{u}^h_{(n)}$ are viewed as known. (The definition of $\tilde{\boldsymbol{u}}^h_{(n)}$ may be inferred from (27).) Let $\boldsymbol{u}^{(i)}$ and $\tilde{\boldsymbol{u}}^{(i)}$ denote the ith iterative approximation of $\boldsymbol{u}^h_{(n+1)}$ and $\tilde{\boldsymbol{u}}^h_{(n)}$, respectively, and let

$$\boldsymbol{R}^{(i)} = \boldsymbol{R}(\boldsymbol{u}^{(i)}; \tilde{\boldsymbol{u}}^{(i)}, \boldsymbol{u}^h_{(n)}), \tag{65}$$

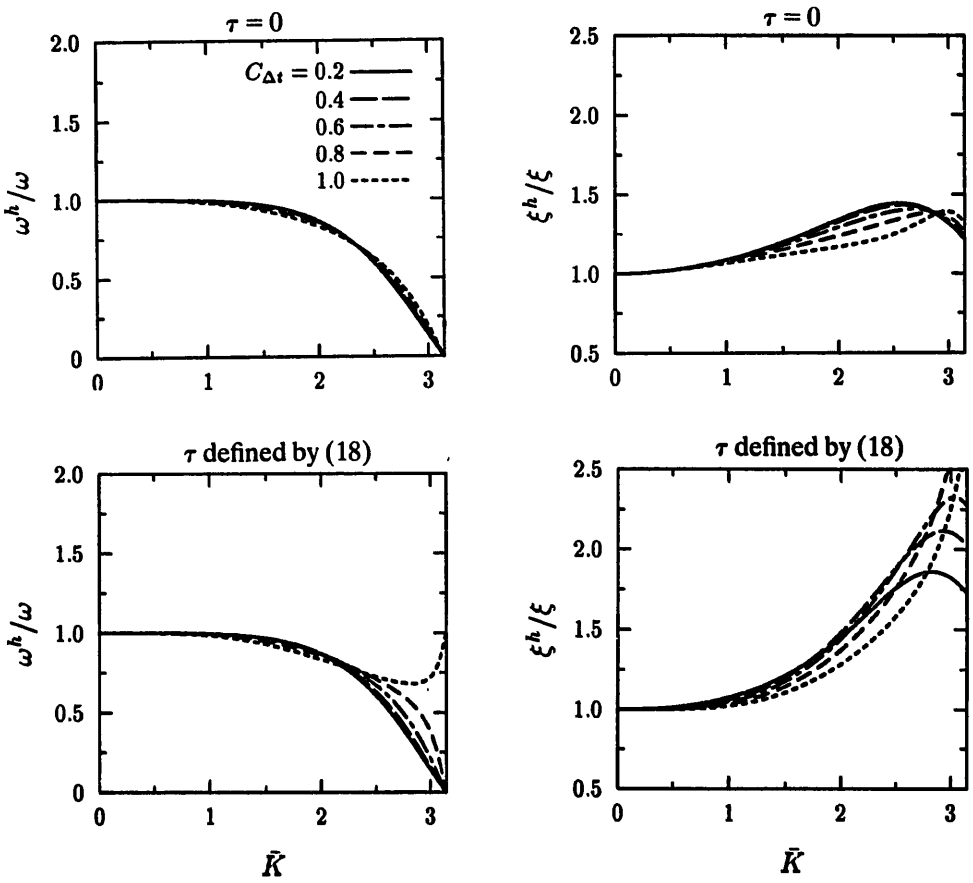

Fig. 7. Advection-diffusion problem ($\alpha = \bar{K}$). Algorithmic frequency and damping ratios for the linear-in-time approximation.

$$\tilde{R}^{(i)} = \tilde{R}(\tilde{u}^{(i)}; u^{(i+1)}, u^h_{(n)}), \tag{66}$$

$$M = \partial R(u; \tilde{u}, u_{(n)}) / \partial u, \tag{67}$$

$$\tilde{M} = \partial \tilde{R}(\tilde{u}; u, u_{(n)}) / \partial \tilde{u}. \tag{68}$$

These are, in order, the primary and secondary residual vectors and corresponding consistent 'mass' matrices. With these definitions, the predictor multi-corrector algorithm is given by

ALGORITHM 4.2. (Predictor multi-corrector algorithm based on the linear-in-time approximation)

(Predictor phase)
$u^{(0)} = u^h_{(n)}$
$\tilde{u}^{(0)} = u^h_{(n)}$
(Multi-corrector phase)
$M^* \Delta u^{(0)} = -R^{(0)}$
$u^{(1)} = u^{(0)} + \Delta u^{(0)}$

For $i = 1, \ldots, i_{\max} - 1$

$$\tilde{\boldsymbol{M}}^* \Delta\tilde{\boldsymbol{u}}^{(i-1)} = -\tilde{\boldsymbol{R}}^{(i-1)}$$
$$\tilde{\boldsymbol{u}}^{(i)} = \tilde{\boldsymbol{u}}^{(i-1)} + \Delta\boldsymbol{u}^{(i-1)}$$
$$\boldsymbol{M}^* \Delta\boldsymbol{u}^{(i)} = -\boldsymbol{R}^{(i)}$$
$$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \Delta\boldsymbol{u}^{(i)}$$
$$\boldsymbol{u}^h_{(n+1)} = \boldsymbol{u}^{(i_{\max})}$$

where $\boldsymbol{M}^*$ and $\tilde{\boldsymbol{M}}^*$ are approximations of $\boldsymbol{M}$ and $\tilde{\boldsymbol{M}}$, respectively.

This algorithm is also residual driven. We consider two sets of definitions for $\boldsymbol{M}^*$ and $\tilde{\boldsymbol{M}}^*$:

Implicit:

In this case, we set $\tilde{\boldsymbol{M}}^* = \boldsymbol{M}^* = (\boldsymbol{M} - (\text{the } a\tau/2 \text{ term}))$. That is, the fifth term in (59) is omitted. The reason for this is that the $a\tau/2$ term is skew symmetric and has an adverse effect on stability. Setting $\tilde{\boldsymbol{M}}^* = \boldsymbol{M}^*$ eliminates storage and factorization costs of $\tilde{\boldsymbol{M}}^*$ (if direct solution methods are used).

The amplification factor for $i_{\max} = 1$ is

$$|\zeta^h|^2 = \frac{9\sin^2(\bar{K})\bar{a}^2 + 4((2+\cos(\bar{K})) - 3(1-\cos(\bar{K}))(\bar{\kappa} + \bar{a}^2\bar{\tau}))^2}{9\sin^2(\bar{K})\bar{a}^2 + 4((2+\cos(\bar{K})) + 3(1+\cos(\bar{K}))(\bar{\kappa} + \bar{a}^2\bar{\tau}))^2}\,. \tag{69}$$

It is clear from this expression that $|\zeta^h| \leq 1$. Consequently, this algorithm is unconditionally stable. The expansions of ξ^h and ω^h are

$$\xi^h = \kappa K^2 + (-a^2\kappa K^4/4 + \kappa^3 K^6/12)\,\Delta t^2 + (a^2K^2)\tau + (\kappa K^4/12)\,\Delta x^2 + O(\Delta t^3, \tau^2, \Delta x^4)\,, \tag{70}$$

$$\omega^h = -aK + (a^3K^3/12 - a\kappa^2K^5/4)\,\Delta t^2 + (-a^3\kappa K^3/2)\,\Delta t^2\tau + (aK^5/180)\,\Delta x^4 + O(\Delta t^3, \Delta t^2\tau^2, \Delta x^6)\,. \tag{71}$$

These results indicate that the one-pass algorithm is second-order accurate with respect to Δt. Numerical testing has shown that unconditionally stability is maintained for subsequent corrector passes. Third-order accuracy with respect to Δt is obtained for $i_{\max} = 2$. For $i_{\max} = 3$, third-order accuracy with respect to Δt with optimal coefficients (i.e., those of the underlying basic linear-in-time approximation) is obtained. Further corrector passes do not improve upon the order of accuracy of the solution; they only drive the solution closer to that of the underlying linear-in-time approximation.

An alternative set of left-hand-side matrices is: $\boldsymbol{M}^* = (\boldsymbol{M} - (\text{the } a\tau/2 \text{ term}))$, as before, and $\tilde{\boldsymbol{M}}^* = (\tilde{\boldsymbol{M}} - (\text{the } a\tau/2 \text{ term}))$. These definitions also result in unconditional stability. Third-order accuracy with optimal coefficients is achieved in the second corrector pass. However, double the matrix storage of the previous algorithm is required for this method and it is then not considered further herein.

Explicit:

In this case, we set $\tilde{\boldsymbol{M}}^* = \boldsymbol{M}^* = \Delta x\,\mathbf{I}$. The first corrector pass of this algorithm produces identical result to the one-pass explicit constant-in-time algorithm; see Section 4.3. Recall from (48) that there is a lower limit for $\bar{\tau} = \tau/\Delta t$. For advection-dominated problems, this

limit is higher than the space–time value of $\bar{\tau}$, defined by (18), and recommended for calculating transient response. Thus, in order to obtain stable response for the explicit one-pass algorithm, we must adhere to the conditions imposed by (48). Furthermore, in the case of advection-dominated problems, the second corrector pass results in unconditional instability, irrespective of the amplitude of τ. However, we have observed that this instability accumulates slowly and linearly. If more corrector passes are performed the method regains conditional stability ($C_\tau \leqslant 1$), and its solution ultimately converges to the solution of the linear-in-time approximation. See [3, Section 4.4.3] for a numerical illustration of this phenomenon.

6. Comparison with semidiscrete formulations

Heretofore, most finite element techniques have been based on a semidiscrete formulation consisting of a finite element discretization in space, followed by a finite difference discretization in time. As a basis for comparison, we present one such formulation which is the counterpart of the fully discrete space–time formulation described previously. For more detailed analyses of these methods we refer to [16–18].

As before, here the spatial domain, $\Omega =]0, L[$, is divided into n_{el} equal-length elements, Ω^e, $e = 1, \ldots, n_{\text{el}}$. The finite element spaces are now defined only with respect to the spatial domain:

$$\mathcal{S}^h = \{u^h \mid u^h \in \mathcal{C}^0(\Omega),\ u^h|_{\Omega^e} \in \mathcal{P}_k(\Omega^e),\ u^h(L, \cdot) = u^h(0, \cdot)\}\,, \tag{72}$$

$$\mathcal{V}^h = \{w^h \mid w^h \in \mathcal{C}^0(\Omega),\ w^h|_{\Omega^e} \in \mathcal{P}_k(\Omega^e),\ w^h(L) = w^h(0)\}\,. \tag{73}$$

Here, the mesh parameter is simply $h = \Delta x$.

The semidiscrete weighted residual formulation corresponding to the Galerkin/least-squares formulation can be stated as: Find $u^h(\cdot, t) \in \mathcal{S}^h$ such that for all $w^h \in \mathcal{V}^h$

$$\int_\Omega (w^h u^h_{,t} + w^h a u^h_{,x} + w^h_{,x} \kappa u^h_{,x})\,\mathrm{d}x + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} (a w^h_{,x} - \kappa w^h_{,xx})\tau(u^h_{,t} + a u^h_{,x} - \kappa u^h_{,xx})\,\mathrm{d}x = 0\,. \tag{74}$$

Note the lack of symmetry of the second integral, in contrast to the corresponding term in the space–time formulation. While the Galerkin method (i.e., $\tau = 0$) is L_2-stable, it is well-known that spurious oscillations may appear in high frequency components.

Proceeding as in Section 4.1, u^h, $u^h_{,t}$ and w^h are interpolated linearly in space as

$$u^h(x, t) = \sum_{j=1}^{n_{\text{np}}} N_j(x) u^h_j(t) \quad \text{for } x \in \Omega,\ t \geqslant 0\,, \tag{75}$$

$$u^h_{,t}(x, t) = \sum_{j=1}^{n_{\text{np}}} N_j(x) \dot{u}^h_j(t) \quad \text{for } x \in \Omega,\ t \geqslant 0\,, \tag{76}$$

$$w^h(x) = \sum_{j=1}^{n_{\text{np}}} N_j(x) w_j^h \quad \text{for } x \in \Omega \,, \tag{77}$$

where now the nodal values u_j^h and $\dot{u}_j^h$ are functions of time. Substitution into the weak form, (74), leads to $n_{\text{np}} - 1$ linear ordinary differential equations

$$G_i(\boldsymbol{u}^h(t), \dot{\boldsymbol{u}}^h(t)) = 0 \quad \text{for } i = 1, \ldots, n_{\text{np}} - 1 \,, \tag{78}$$

where

$$\boldsymbol{u}^h \overset{\text{def}}{=} \{u_1^h, u_2^h, \ldots, u_{n_{\text{np}}}^h\} \,, \tag{79}$$

$$\dot{\boldsymbol{u}}^h \overset{\text{def}}{=} \{\dot{u}_1^h, \dot{u}_2^h, \ldots, \dot{u}_{n_{\text{np}}}^h\} \,. \tag{80}$$

(Recall that $u_{n_{\text{np}}}^h = u_1^h$ and $\dot{u}_{n_{\text{np}}}^h = \dot{u}_1^h$.)

Equation (78) necessitates the introduction of a time-stepping algorithm, such as the *generalized trapezoidal rule*:

$$\boldsymbol{u}_{(n+1)}^h = \boldsymbol{u}_{(n)}^h + (1-\gamma)\,\Delta t\, \dot{\boldsymbol{u}}_{(n)}^h + \gamma\, \Delta t\, \dot{\boldsymbol{u}}_{(n+1)}^h \,, \tag{81}$$

where $\boldsymbol{u}_{(n)}^h$ and $\dot{\boldsymbol{u}}_{(n)}^h$ are the approximations of $\boldsymbol{u}^h(t_n)$ and $\dot{\boldsymbol{u}}^h(t_n)$, respectively, and $\gamma \in [0, 1]$ is a parameter. The combination of (81) and

$$G_i(\boldsymbol{u}_{(n)}^h, \dot{\boldsymbol{u}}_{(n)}^h) = 0 \,, \quad i = 1, \ldots, n_{\text{np}} - 1 \,, \tag{82}$$

$$G_i(\boldsymbol{u}_{(n+1)}^h, \dot{\boldsymbol{u}}_{(n+1)}^h) = 0 \,, \quad i = 1, \ldots, n_{\text{np}} - 1 \,, \tag{83}$$

leads to the following three-point spatial difference stencil centered at node j:

$$\begin{aligned}
&\left(\frac{\Delta x}{6} - \kappa\gamma\,\frac{\Delta t}{\Delta x} - a\gamma\,\frac{\Delta t}{2} + \frac{a\tau}{2} - a^2\tau\gamma\,\frac{\Delta t}{\Delta x}\right)(u_{j-1;(n+1)}^h - u_{j-1;(n)}^h) \\
&\quad + \left(4\,\frac{\Delta x}{6} + 2\kappa\gamma\,\frac{\Delta t}{\Delta x} + 2a^2\tau\gamma\,\frac{\Delta t}{\Delta x}\right)(u_{j;(n+1)}^h - u_{j;(n)}^h) \\
&\quad + \left(\frac{\Delta x}{6} - \kappa\gamma\,\frac{\Delta t}{\Delta x} + a\gamma\,\frac{\Delta t}{2} - \frac{a\tau}{2} - a^2\tau\gamma\,\frac{\Delta t}{\Delta x}\right)(u_{j+1;(n+1)}^h - u_{j+1;(n)}^h) \\
&= \left(\kappa\,\frac{\Delta t}{\Delta x} + a\,\frac{\Delta t}{2} + a^2\tau\,\frac{\Delta t}{\Delta x}\right)u_{j-1;(n)}^h + \left(-2\kappa\,\frac{\Delta t}{\Delta x} - 2a^2\tau\,\frac{\Delta t}{\Delta x}\right)u_{j;(n)}^h \\
&\quad + \left(\kappa\,\frac{\Delta t}{\Delta x} - a\,\frac{\Delta t}{2} + a^2\tau\,\frac{\Delta t}{\Delta x}\right)u_{j+1;(n)}^h \,.
\end{aligned} \tag{84}$$

This equation relates the unknown values at time level $n+1$ to the known values at time level n. By changing the value of γ, different properties are obtained. We are primarily interested in the following three cases:

(1) *Backward differences* ($\gamma = 1$):

Up to the terms involving $a\tau/2$, this case is identical to the constant-in-time formulation (31). The additional terms contribute a skew-symmetric matrix to the left-hand side, and the sign of these terms is opposite to that of the $a\,\Delta t/2$ terms. (Note that one can symmetrize the left-hand-side matrix by setting $\tau = \Delta t$.) The amplification factor in this case is

$$|\zeta^h|^2 = \frac{9\sin^2(\bar{K})\bar{a}^2\bar{\tau}^2 + (2+\cos(\bar{K}))^2}{9\sin^2(\bar{K})\bar{a}^2(1-\bar{\tau})^2 + ((2+\cos(\bar{K})) + 6(1-\cos(\bar{K}))(\bar{\kappa}+\bar{a}^2\bar{\tau}))^2}\,. \tag{85}$$

As $|\zeta^h| \leq 1$, unconditional stability is obtained. The expansions of ξ^h and ω^h are

$$\xi^h = \kappa K^2 + \left(\frac{a^2K^2}{2} - \frac{\kappa^2K^4}{2}\right)\Delta t + (-a^2\kappa K^4)\tau^2 + \left(\frac{\kappa K^4}{12}\right)\Delta x^2 + O(\Delta t^2, \tau^3, \Delta x^4)\,, \tag{86}$$

$$\omega^h = -aK + (a\kappa K^3)\,\Delta t + (-a\kappa K^3)\tau + (aK^5/180)\,\Delta x^4 + O(\Delta t^2, \tau^2, \Delta x^6)\,. \tag{87}$$

These results show that the algorithm is first-order time-accurate, with identical coefficients of the Δt terms as for the constant-in-time approximation; see (38) and (39). However, the main effect of τ has been shifted from ξ^h to ω^h.

(2) *Trapezoidal* ($\gamma = \frac{1}{2}$):

This algorithm may be considered the counterpart of the one-pass implicit third-order predictor multi-corrector algorithm discussed in Section 5.3. The amplification factor is

$$|\zeta^h|^2 = \frac{9\sin^2(\bar{K})\bar{a}^2(1+2\bar{\tau})^2 + 4((2+\cos(\bar{K})) - 3(1-\cos(\bar{K}))(\bar{\kappa}+\bar{a}^2\bar{\tau}))^2}{9\sin^2(\bar{K})\bar{a}^2(1-2\bar{\tau})^2 + 4((2+\cos(\bar{K})) + 3(1+\cos(\bar{K}))(\bar{\kappa}+\bar{a}^2\bar{\tau}))^2}\,. \tag{88}$$

Since $\zeta^h \leq 1$, unconditional stability is attained. The expansion of ξ^h and ω^h are

$$\xi^h = \kappa K^2 + \left(-\frac{a^2\kappa K^4}{4} + \frac{\kappa^3K^6}{12}\right)\Delta t^2 + (-a^2\kappa K^4)\tau^2 + \left(\frac{\kappa K^4}{12}\right)\Delta x^2 + O(\Delta t^3, \tau^3, \Delta x^4)\,, \tag{89}$$

$$\omega^h = -aK + \left(\frac{a^3K^3}{12} - \frac{a\kappa^2K^5}{4}\right)\Delta t^2 + (-a\kappa K^3)\tau + \left(\frac{aK^5}{180}\right)\Delta x^4 + O(\Delta t^3, \tau^2, \Delta x^6)\,. \tag{90}$$

As anticipated, the time accuracy is second-order.

(3) *Forward differences* ($\gamma = 0$):

In order to obtain an explicit algorithm, the left-hand-side is lumped by adding its row elements. The contributions of the $a\tau/2$ terms are eliminated in the process. This case then reduces to the one-pass explicit first-order predictor multi-corrector algorithm described in Section 4.3.

Table 1
Accuracy and stability of the constant- and linear-in-time approximations

Approximation	Dissipation	Dispersion	Stability
Constant-in-time	Δt, τ, Δx^2	Δt, $\Delta t\,\tau$, Δx^4	unconditional
Linear-in-time (advective limit)	Δt^3, $\Delta t^2\tau$, Δx^6	Δt^4, $\Delta t^3\tau$, Δx^4	unconditional
Linear-in-time (diffusive limit)	Δt^3, Δx^2	none	unconditional

Table 2
Accuracy and stability of predictor multi corrector space–time and semidiscrete algorithms

Approximation	Algorithm	Dissipation	Dispersion	Stability
Constant-in-time	One-pass explicit	Δt, τ, Δx^2	Δt, $\Delta t\,\tau$, Δx^2	$a\frac{\Delta t}{\Delta x} \leq 2(\kappa + a^2\tau)\frac{\Delta t}{\Delta x^2} \leq 1$
	Two-pass explicit[a]	Δt, τ, Δx^2	Δt, $\Delta t\,\tau$, Δx^4	$a\frac{\Delta t}{\Delta x} \leq 2(\kappa + a^2\tau)\frac{\Delta t}{\Delta x^2} \leq 1$
Linear-in-time	One-pass implicit	Δt^2, τ, Δx^2	Δt^2, $\Delta t^2\tau$, Δx^4	unconditional
	Two-pass implicit[a,b]	Δt^3, $\Delta t^2\tau$, Δx^2	Δt^4, $\Delta t^3\tau$, Δx^4	unconditional
	One-pass explicit	Δt, τ, Δx^2	Δt, $\Delta t\,\tau$, Δx^2	$a\frac{\Delta t}{\Delta x} \leq 2(\kappa + a^2\tau)\frac{\Delta t}{\Delta x^2} \leq 1$
	Two-pass explicit	Δt^2, $\Delta t\,\tau$, Δx^2	Δt^2, $\Delta t^2\tau$, Δx^4	unstable when advection dominates
	Three-pass explicit[a,b]	Δt^3, $\Delta t^2\tau$, Δx^2	Δt^4, $\Delta t^3\tau$, Δx^4	conditional
Semidiscrete	Backward differences	Δt, τ^2, Δx^2	Δt, τ, Δx^4	unconditional
	Trapezoidal	Δt^2, τ^2, Δx^2	Δt^2, τ, Δx^4	unconditional
	Forward differences	Δt, τ, Δx^2	Δt, $\Delta t\,\tau$, Δx^2	$a\frac{\Delta g}{\Delta x} \leq 2(\kappa + a^2\tau)\frac{\Delta t}{\Delta x^2} \leq 1$

[a] Accuracy and stability results quoted also hold for subsequent passes.
[b] Results quoted for this algorithm are partially based on numerical tests rather than analytical derivations.

7. Summary

The stability and accuracy results presented in the previous sections are summarized in Tables 1 and 2.

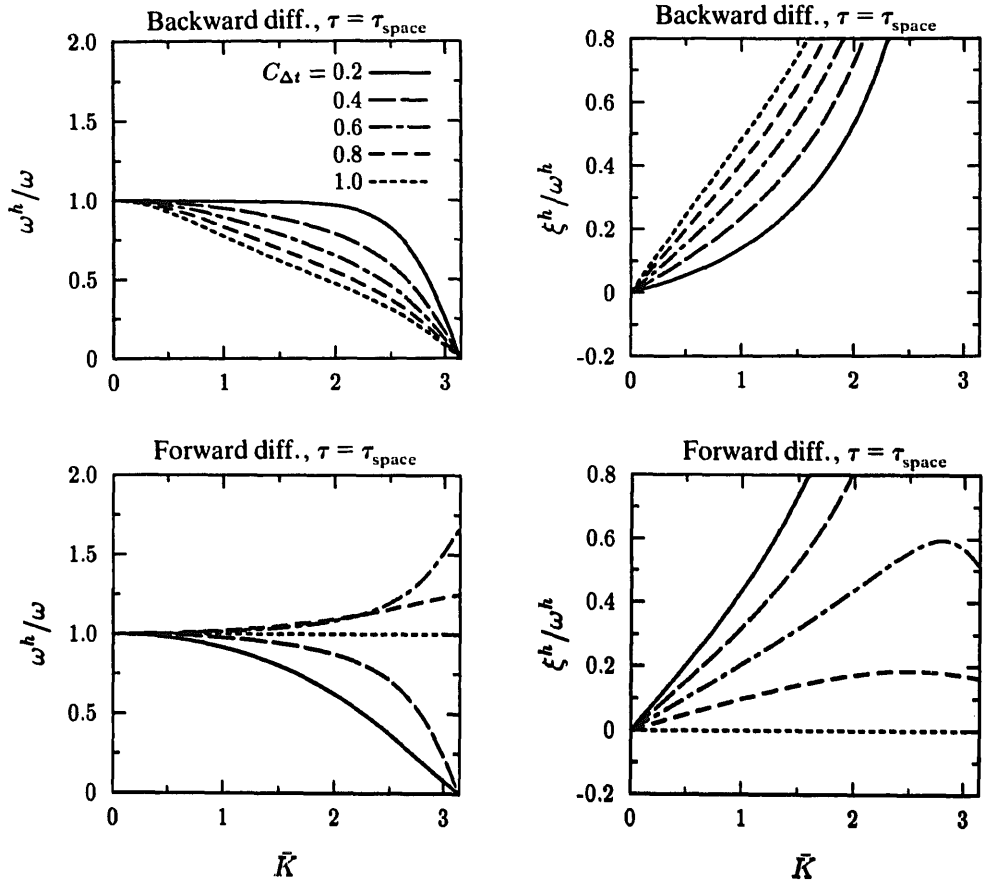

Fig. 8. Pure advection problem. Algorithmic frequency and damping ratios for semidiscrete algorithms primarily used to obtain steady solutions. The constant-in-time one-pass explicit algorithm is the same as the semidiscrete forward differences algorithm. It is interesting to compare the semidiscrete backward differences results with the constant-in-time results (see Fig. 3).

The accuracy measures for the pure advection example of Section 4.2 were computed for the above algorithms, and the results are plotted in Figs. 8 and 9. In the case of the algorithms used to obtain steady solutions (presented in Fig. 8) we employed τ_{space}. The full space–time τ, (18), was employed to obtain the results shown in Fig. 9.

8. Conclusions

We have presented a Fourier stability and accuracy analysis of the constant-in-time and linear-in-time approximations of the space–time Galerkin/least-squares method applied to a time-dependent linear scalar advective-diffusive model problem. Both approximations are unconditionally stable.

The constant-in-time approximation is only first-order time accurate, but possesses very strong damping properties and is computationally efficient. It is concluded that this algorithm is well-suited for solving steady problems.

The linear-in-time approximation leads to an algebraic system having twice as many

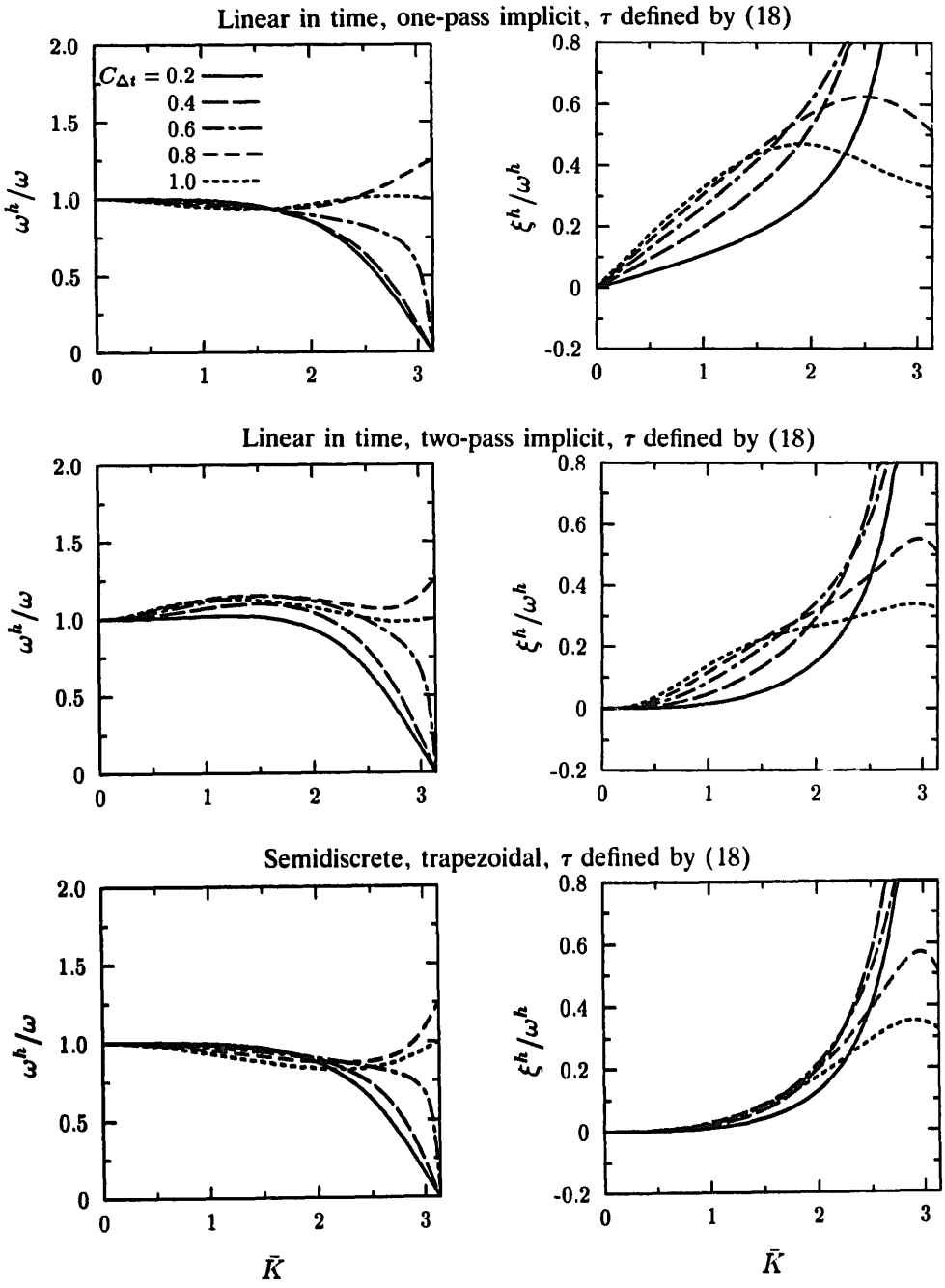

Fig. 9. Pure advection problem. Algorithmic frequency and damping ratios for predictor multi-corrector space–time and semidiscrete trapezoidal algorithms used primarily to obtain unsteady solutions.

equations and unknowns as the constant-in-time approximation. We presented a technique to reduce this system to two weakly-coupled subsystems, and a predictor multi-corrector algorithm which employs the same left-hand-side matrix for both subsystems. The two-pass implicit version of this algorithm is third-order time accurate, which renders it effective for obtaining unsteady solutions. The least-squares operator increases the numerical dissipation at high wave numbers without significantly affecting dispersive properties.

Acknowledgment

The authors would like to express their appreciation to Greg Hulbert for helpful comments.

References

[1] T.J.R. Hughes, L.P. Franca and G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173–189.

[2] T.J.R. Hughes and F. Shakib, Computational aerodynamics and the finite element method, AIAA 26th Aerospace Sciences Meeting, Paper No. 88-0031, Reno, Nevada, January 1988.

[3] F. Shakib, Finite element analysis of the compressible Euler and Navier–Stokes equations, Ph.D. Thesis, Division of Applied Mechanics, Stanford University, 1989.

[4] C. Johnson, U. Nävert and J. Pitkäranta, Finite element methods for linear hyperbolic problems, Comput. Methods Appl. Mech. Engrg. 45 (1984) 285–312.

[5] T.J.R. Hughes, Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier–Stokes equations, Internat. J. Numer. Methods Fluids 7 (1987) 1261–1275.

[6] T.J.R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Engrg. 58 (1986) 305–328.

[7] T.J.R. Hughes, L.P. Franca and M. Mallet, A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multi-dimensional advective-diffusive systems, Comput. Methods Appl. Mech. Engrg. 63 (1987) 97–112.

[8] C. Johnson and A. Szepessy, Convergence of a finite element method for a nonlinear hyperbolic conservation law, Technical Report 1985-25, Mathematics Department, Chalmers University of Technology, Göteborg, Sweden, 1985.

[9] C. Johnson and A. Szepessy, A shock-capturing streamline diffusion finite element method for a nonlinear hyperbolic conservation law, Technical Report 1986-09, Mathematics Department, Chalmers University of Technology, Göteborg, Sweden, 1986.

[10] C. Johnson and A. Szepessy, On the convergence of streamline diffusion finite element methods for hyperbolic conservation flaws, in: T.E. Tezduyar and T.J.R. Hughes, eds., Numerical Methods for Compressible Flows–Finite Difference, Element and Volume Techniques, AMD Vol. 78 (ASME, New York, 1986) 75–91.

[11] C. Johnson, A. Szepessy and P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws, Technical Report 1987-21, Mathematics Department, Chalmers University of Technology, Göteborg, 1987.

[12] A. Szepessy, Convergence of a shock-capturing streamline diffusion finite element method for a scalar conservation law in two space dimensions, Technical Report 1988-07, Mathematics Department, Chalmers University of Technology, Göteborg, Sweden, 1988.

[13] U. Nävert, A finite element method for convection-diffusion problems, Ph.D. Thesis, Department of Computer Science, Chalmers University of Technology, Göteborg, Sweden, 1982.

[14] F. Shakib, T.J.R. Hughes and Z. Johan, A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis, Comput. Methods Appl. Mech. Engrg. 75 (1989) 415–456.

[15] R.F. Warming and B.J. Hyett, The modified equation approach to the stability and accuracy analysis of finite-difference methods, J. Comput. Physics 14 (1974) 159–179.

[16] A.N. Brooks and T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[17] T.J.R. Hughes and T.E. Tezduyar, Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations, Comput. Methods Appl. Mech. Engrg. 45 (1984) 217–284.

[18] T.J.R. Hughes and T.E. Tezduyar, Analysis of some fully-discrete algorithms for the one-dimensional heat equation, Internat. J. Numer. Methods Engrg. 21 (1985) 163–168.