INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 20, 443–458 (1995)

ANALYSIS OF FINITE ELEMENT SCHEMES FOR CONVECTION-TYPE PROBLEMS

GIANNI COMINI, MARCO MANZAN AND CARLO NONINO

Istituto di Fisica Tecnica e di Tecnologie Industriali, Università di Udine, Viale Ungheria 41, 33100 Udine, Italy

SUMMARY

Various finite element schemes of the Bubnov–Galerkin and Taylor–Galerkin types are analysed to obtain the expressions of truncation errors. This way, dispersion errors in the transient, and diffusion errors both in the transient and in the steady state, are identified. Then, with reference to the transient advection–diffusion equation, stability limits are determined by means of a general von Neumann procedure. Finally, the operational equivalence between Taylor–Galerkin methods, utilized for pseudo-transient calculations, and Petrov–Galerkin methods, derived for the steady state forms of the advection–diffusion equation, is illustrated. Theoretical conclusions are supported by the results of numerical experiments

KEY WORDS: finite elements; advection-diffusion; diffusion errors; dispersion errors; stability

INTRODUCTION

The finite element method, based on Galerkin formulations, has become a well-established procedure for the solution of convection-type problems, where both advection and diffusion mechanisms must be accounted for in the mathematical model. On the other hand, for advection-dominated problems, standard Bubnov–Galerkin procedures can lead to space oscillations, or 'wiggles', in the numerical solution unless the mesh is refined to reduce the local Peclet number.[1,2] The use of 'upwinding', or Petrov–Galerkin types of discretization, as an alternative treatment for node-to-node oscillations, has been investigated by many authors.[1–3] However, Petrov–Galerkin methods have been derived for the steady-state forms of the transport equations and their extension to the transient solutions requires special care.[1,2,4–7] Instead, for the transient advection–diffusion equations, alternative discretization procedures, connected with special time integration schemes, have gained wide acceptance in recent years.[1,2,8,9]

Among the new methods, the balancing tensor diffusivity, the characteristic-based Galerkin and the Taylor–Galerkin methods are the most popular. Despite their widely different mathematical and physical foundations, all these methods lead to discretized equations that are either very similar, when the source terms are variable, or identical, when the source terms are constant. The resulting discretized equations always contain upwinding terms that are dependent on the time increment used and are characterized by effects that do not disappear when the steady state has been reached. Thus, even if we are only interested in steady-state solutions, we can still use one of these algorithms to find pseudo-transient solutions without 'wiggles'. On the other hand, with sufficiently refined meshes, standard Bubnov–Galerkin discretizations and implicit time integration schemes can still be considered a viable choice for the solution of convection-type problems. In fact, the standard fully implicit scheme allows the use of very large time steps and, even if it is characterized by a numerical diffusion in the transient, it does not present any numerical diffusion

CCC 0271–2091/95/060443–16

Received 2 February 1994
Revised 2 November 1994

when the steady state has been reached.[9] The standard Crank–Nicolson scheme is not plagued by numerical diffusion, and is unconditionally stable, even though for relatively large time steps is characterized by serious dispersion errors.

At this point, the global picture can be rather confusing: we appreciate unconditional stability but we want zero numerical diffusion with no dispersion and, occasionally, we like to suppress the wiggles without having to refine the mesh. We can have, at a price, any one of these nice features but we cannot have all of them built into a single algorithm. Obviously, at the end, we will have to compromise, but not before having very carefully evaluated all the possible approaches. In this paper several popular finite element procedures of the Bubnov–Galerkin and Taylor-Galerkin types are investigated, with the aim of finding numerical dispersion errors, diffusion errors and stability limits. Numerical dispersion errors are evaluated in the transient, while numerical diffusion errors are evaluated both for transient and steady-state situations. Actually, it has been established long ago that numerical diffusion errors can be different for transient and pseudo-transient calculations, but to the authors' knowledge this distinction has been made only in the context of finite difference methods.[10] In this paper, the stability limits of the investigated algorithms are also reported. The limits are obtained by means of a very general procedure which is normally employed in the context of the finite difference method.[11] Finally, well-known theoretical analyses[1,2] are invoked to demonstrate the operational equivalence between Taylor–Galerkin methods, utilized for pseudo-transient calculations, and Petrov–Galerkin methods, derived for the steady-state forms of the advection–diffusion equations.

In the next sections, the theoretical results are discussed in detail and supported by numerical experiments, with the aim of establishing practical guidelines to identify the most suitable algorithms for any problem of the convection type.

DIFFUSION AND DISPERSION ERRORS

The performances of finite element schemes with respect to diffusion and dispersion errors are usually evaluated with reference to the pure advection equation to ensure that all diffusion and dispersion effects have a non-physical origin. The effect of diffusion is to smooth out gradients and space variations of the independent variable. Therefore, we can say that an algorithm is affected by artificial, or numerical, diffusion when the computed amplitudes of the elementary wave components are always smaller than the corresponding exact, or analytical, values and, consequently, perturbations are excessively damped. On the other hand, numerical schemes can also show a non-physical, 'negative' diffusion and, in such a case, perturbations are amplified, sometimes to an extent that leads to numerical instability. In pure advection problems, all perturbations propagate in the same direction and with the same velocity of the flow. Consequently, we can say that an algorithm is affected by dispersion errors if the various wave components of the numerical solution have different velocities and spread apart as the solution proceeds in time, creating a pattern of trailing, or leading, waves.

For a constant property incompressible fluid, the pure advection equation can be written in the form

$$\frac{\partial T}{\partial t} = T_t = -u\frac{\partial T}{\partial x} = -uT_x \tag{1}$$

where T is the temperature, t is the time and u is the velocity. If u is time independent, from equation (1) we obtain immediately

$$T_{tt} = \frac{\partial T_t}{\partial t} = -\frac{\partial}{\partial t}\left(u\frac{\partial T}{\partial x}\right) = -u\frac{\partial T_t}{\partial x} = u^2\frac{\partial^2 T}{\partial x^2} = u^2 T_{xx} \tag{2}$$

The time discretization of equation (1) can be achieved by computing the value of the unknown T, at the time level $(n + 1)\Delta t$, by means of Taylor series expansions of initial point $n\Delta t$, written in the form

$$T^{n+1} = T^n + \Delta t T_t^{n+\alpha} \tag{3}$$

or in the form

$$T^{n+1} = T^n + \Delta t T_t^n + \tfrac{1}{2}\Delta t^2 T_{tt}^{n+\gamma} \tag{4}$$

where α and γ are weighting parameters assuming a suitable value in the interval between 0 and 1. Obviously, for α and γ arbitrarily chosen, the above equations are not exact anymore, and equation (3) yields an accuracy of the first order for the time expansion, while equation (4) leads to an accuracy of the second order.

From equations (3) and (1), we obtain the algorithms of the Bubnov–Galerkin type (BG) for the time discretization of the pure advection equation

$$\frac{T^{n+1} - T^n}{\Delta t} \cong T_t^{n+\alpha} = -uT_x^{n+\alpha} \cong -u[\alpha T_x^{n+1} + (1 - \alpha)T_x^n] \tag{5}$$

In equation (5), the advection term can be evaluated anywhere in the time interval between $n\Delta t$ and $(n + 1)\Delta t$. Accordingly, for $\alpha = 0$ we obtain the BG explicit algorithm (BG-E), for $\alpha = \frac{1}{2}$ we obtain the BG Crank–Nicolson algorithm (BG-C) and for $\alpha = 1$ we have the BG fully implicit algorithm (BG-I). Similarly, equations (4), (1) and (2) yield the algorithms of the Taylor–Galerkin type (TG) for the time discretization of the pure advection equation

$$\frac{T^{n+1} - T^n}{\Delta t} \cong T_t^n + \frac{\Delta t}{2} T_{tt}^{n+\gamma} \cong -uT_x^n + \frac{u^2\Delta t}{2}[\gamma T_{xx}^{n+1} + (1 - \gamma)T_{xx}^n] \tag{6}$$

In equation (6) the advection term is always evaluated at the time level n, while the diffusion-like term can be evaluated anywhere in the time interval between $n\Delta t$ and $(n + 1)\Delta t$. Accordingly, for $\gamma = 0$ we obtain the TG fully explicit algorithm (TG-E), for $\gamma = \frac{1}{2}$ we obtain the TG Crank–Nicolson algorithm (TG-C) and for $\gamma = 1$ we have the TG implicit algorithm (TG-I).

Equations (5) and (6) can be discretized, with respect to the space co-ordinates, using a standard Galerkin process. For the typical node i, using elements of equal size Δx and linear shape functions, we obtain the assembled finite element equations[12]

$$\begin{aligned}
&\frac{1}{6\Delta t}[(T_{i-1}^{n+1} + 4T_i^{n+1} + T_{i+1}^{n+1}) - (T_{i-1}^n + 4T_i^n + T_{i+1}^n)] \\
&= -\frac{u}{2\Delta x}[\alpha(T_{i+1}^{n+1} - T_{i-1}^{n+1}) + (1 - \alpha)(T_{i+1}^n - T_{i-1}^n)] \\
&\quad + \beta\frac{u^2\Delta t}{2\Delta x^2}[\gamma(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + (1 - \gamma)(T_{i-1}^n - 2T_i^n + T_{i+1}^n)]
\end{aligned} \tag{7}$$

and

$$\begin{aligned}
\frac{1}{\Delta t}(T_i^{n+1} - T_i^n) = &-\frac{u}{2\Delta x}[\alpha(T_{i+1}^{n+1} - T_{i-1}^{n+1}) + (1 - \alpha)(T_{i+1}^n - T_{i-1}^n)] \\
&+ \beta\frac{u^2\Delta t}{2\Delta x^2}[\gamma(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + (1 - \gamma)(T_{i-1}^n - 2T_i^n + T_{i+1}^n)]
\end{aligned} \tag{8}$$

for consistent and lumped capacity matrices, respectively. Clearly, the values of parameters α, β, and γ determine the particular BG or TG algorithms, as shown in Table I where the subscripts in the acronyms identifying the algorithms refer to the use of consistent (C) or lumped (L) capacity matrices.

In determining the accuracy of the various schemes, it is worth noting that equations (7) and (8) can be considered numerical representations of the following modified transport equation:[10,13]

$$T_t + uT_x = a_{\mathrm{et}}T_{xx} + \varepsilon_t^* = \varepsilon_t \tag{9}$$

where a_{et} is the transient equivalent numerical diffusion, ε_t^* is the residual transient truncation error and ε_t is the global transient truncation error. Obviously, in the steady state we have

$$T_{i\pm 1}^{n+1} = T_{i\pm 1}^{n} = T_{i\pm 1} \tag{10}$$

and thus equations (7) and (8) reduce to the same form

$$\frac{u}{2\Delta x}(T_{i+1} - T_{i-1}) = \beta\frac{u^2\Delta t}{2\Delta x^2}(T_{i-1} - 2T_i + T_{i+1}) \tag{11}$$

Table I. Values of the parameters α, β and γ to be used in equations (7), (8) and (11) to obtain the finite element schemes considered for the solution of the advection equation (1). Symbols a_{et} and a_{es} denote the transient and the steady-state equivalent diffusion coefficients, while symbols ε_t and ε_s indicate the transient and the steady-state truncation errors.

Scheme	α	β	γ	a_{et}	ε_t	a_{es}	ε_s
BG$_{\mathrm{C}}$-E	0	0	–	$-\frac{u^2\Delta t}{2}$	$O(\Delta t, \Delta x^4)$	0	$O(\Delta x^2)$
BG$_{\mathrm{L}}$-E	0	0	–	$-\frac{u^2\Delta t}{2}$	$O(\Delta t, \Delta x^2)$	0	$O(\Delta x^2)$
BG$_{\mathrm{C}}$-C	$\frac{1}{2}$	0	–	0	$O(\Delta t^2, \Delta x^4)$	0	$O(\Delta x^2)$
BG$_{\mathrm{L}}$-C	$\frac{1}{2}$	0	–	0	$O(\Delta t^2, \Delta x^2)$	0	$O(\Delta x^2)$
BG$_{\mathrm{C}}$-I	1	0	–	$\frac{u^2\Delta t}{2}$	$O(\Delta t, \Delta x^4)$	0	$O(\Delta x^2)$
BG$_{\mathrm{L}}$-I	1	0	–	$\frac{u^2\Delta t}{2}$	$O(\Delta t, \Delta x^2)$	0	$O(\Delta x^2)$
TG$_{\mathrm{C}}$-E	0	1	0	0	$O(\Delta t^2, \Delta x^4)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$
TG$_{\mathrm{L}}$-E	0	1	0	0	$O(\Delta t^2, \Delta x^2)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$
TG$_{\mathrm{C}}$-C	0	1	$\frac{1}{2}$	0	$O(\Delta t^2, \Delta x^4)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$
TG$_{\mathrm{L}}$-C	0	1	$\frac{1}{2}$	0	$O(\Delta t^2, \Delta x^2)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$
TG$_{\mathrm{C}}$-I	0	1	1	0	$O(\Delta t^2, \Delta x^4)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$
TG$_{\mathrm{L}}$-I	0	1	1	0	$O(\Delta t^2, \Delta x^4)$	$\frac{u^2\Delta t}{2}$	$O(\Delta x^2)$

which can be considered the numerical representation of the transport equation

$$uT_x = a_{es}T_{xx} + \varepsilon_s^* = \varepsilon_s \tag{12}$$

where a_{es}, ε_s^* and ε_s are the steady-state equivalent numerical diffusion, the residual truncation error and the global truncation error, respectively.

For all the schemes referred to in Table I, we can repeatedly use the general Taylor series expansion of T around the time instant $t = n\Delta t$ and the grid point i, having co-ordinate x

$$\begin{aligned} T(x \pm \Delta x, t + \Delta t) = T(x, t) &\pm \Delta x T_x(x, t) + \Delta t T_t(x, t) \\ &+ \frac{\Delta x^2}{2} T_{xx}(x, t) \pm \Delta x \Delta t T_{xt}(x, t) + \frac{\Delta t^2}{2} T_{tt}(x, t) \pm \frac{\Delta x^3}{6} T_{xxx}(x, t) \\ &+ \frac{\Delta x^2 \Delta t}{2} T_{xxt}(x, t) \pm \frac{\Delta x \Delta t^2}{2} T_{xtt}(x, t) + \frac{\Delta t^3}{6} T_{ttt}(x, t) + \cdots \end{aligned} \tag{13}$$

This way, with reference to equations (7) and (8) we obtain the general expression of the transient truncation error ε_t, referred to the grid point i and to the time level n

$$\begin{aligned} \varepsilon_t = &- \Delta t\left(\frac{1}{2} T_{tt} + \alpha u T_{xt} - \beta \frac{u^2}{2} T_{xx}\right) - \Delta t^2\left(\frac{1}{6} T_{ttt} + \alpha \frac{u}{2} T_{xtt} - \beta\gamma \frac{u^2}{2} T_{xxt}\right) \\ &- \Delta x^2\left(\frac{\delta}{6} T_{xxt} + \frac{u}{6} T_{xxx}\right) + O(\Delta t^3, \Delta x^4) \end{aligned} \tag{14}$$

where α, β, γ are defined in Table I, while we have $\delta = 1$ or $\delta = 0$ for consistent and lumped capacity matrices, respectively.

Equation (14) can be conveniently rearranged by recursive applications of the advection equations (1) which yield, in addition to equation (2), such expressions as

$$T_{xxt} = - uT_{xxx}, \quad T_{xtt} = u^2 T_{xxx}, \quad T_{ttt} = - u^3 T_{xxx} \tag{15}$$

Setting $\alpha = 0$ in equation (14) and taking into account equations (2) and (15) we can express the transient truncation error for the BG-E and the TG algorithms in the form

$$\begin{aligned} \varepsilon_t = &- \Delta t\left(\frac{u^2}{2} T_{xx} - \beta \frac{u^2}{2} T_{xx}\right) + \Delta t^2\left(\frac{u^3}{6} T_{xxx} - \beta\gamma \frac{u^3}{2} T_{xxx}\right) \\ &- \Delta x^2\left(\frac{u}{6} T_{xxx} - \delta \frac{u}{6} T_{xxx}\right) + O(\Delta t^3, \Delta x^4) \end{aligned} \tag{16}$$

The transient truncation error for the BG-C schemes can be obtained from equation (14) by setting $\alpha = \frac{1}{2}$ and taking into account equations (2) and (15)

$$\varepsilon_t = - \Delta t^2 \frac{u^3}{12} T_{xxx} - \Delta x^2\left(\frac{u}{6} T_{xxx} - \delta \frac{u}{6} T_{xxx}\right) + O(\Delta t^3, \Delta x^4) \tag{17}$$

Finally, by setting $\alpha = 1$ in equation (14) and taking into account equations (2) and (15) we can obtain the transient truncation error for the BG-I algorithms in the form

$$\varepsilon_t = \Delta t \frac{u^2}{2} T_{xx} - \Delta t^2 \frac{u^3}{3} T_{xxx} - \Delta x^2\left(\frac{u}{6} T_{xxx} - \delta \frac{u}{6} T_{xxx}\right) + O(\Delta t^3, \Delta x^4) \tag{18}$$

In equations (16)–(18) we have $\delta = 1$ or $\delta = 0$ for consistent or lumped capacity matrices, respectively. It is apparent that in these three equations the coefficient of the Δx^2 term becomes equal to zero for $\delta = 1$, while it is different from zero for $\delta = 0$. Therefore, we have an $O(\Delta x^4)$ truncation error in the space discretization if we use consistent capacity matrices and only an $O(\Delta x^2)$ truncation error if we use lumped capacity matrices. It must be pointed out, however, that this holds true only if the mesh is regular and equally spaced.

In the steady state, all time derivatives in equation (14) become equal to zero, and we immediately obtain the steady-state truncation error in the form

$$\varepsilon_s = \beta \frac{u^2 \Delta t}{2} T_{xx} - \frac{u \Delta x^2}{6} T_{xxx} + O(\Delta x^4) = \beta a_{es} T_{xx} + O(\Delta x^2) \tag{19}$$

Equation (19) has a general validity and can be utilized for all the algorithms referred to in Table I.

Theoretical analysis[5,14] and numerical experiments indicate that diffusion errors are influenced by the terms of the truncation error containing even order derivatives, while dispersion errors essentially depend on the terms of ε_t containing odd derivatives of order three or higher. Therefore, a simple inspection of equations (16)–(19) can already lead to interesting conclusions on the behaviour of the different algorithms. In particular, from comparisons with equations (9) and (12) we can say that the numerical diffusion coefficients a_{et} and a_{es}, i.e., the coefficients of the terms containing second-order derivatives, are always equal to $\pm \Delta t u^2/2$ when different from zero.

The transient and the steady-state equivalent diffusion coefficients, and the order of magnitude of the truncation errors resulting from equations (16)–(19) are reported in Table I. From this table we can see that the BG-C schemes alone are characterized by truncation errors that do not include a numerical diffusion term, either in the transient or in the steady-state analyses. Instead, the other BG algorithms exhibit numerical diffusion in the transient but not in the steady state, while TG algorithms do not exhibit numerical diffusion in the transient, but present numerical diffusion in the steady state. In particular, BG-E algorithms show a negative numerical diffusion in the transient analysis and, consequently, these algorithms are unstable if the partial differential equation does not include a physical diffusion term. Besides, it is worth noting that consistent capacity matrices always yield better space accuracies than lumped capacity matrices. However, the absence of numerical diffusion is not the only criterion to judge the performance of an algorithm. In fact, also dispersion errors play an important role,[2,10–12] but we prefer to defer our considerations on this issue until, in a later section, we are able to support them with the results of some numerical experiments.

STABILITY ANALYSIS

In analysing the numerical stability, we consider the transient advection–diffusion equation. For a constant-property incompressible fluid this equation can be written as

$$\frac{\partial T}{\partial t} = T_t = -u \frac{\partial T}{\partial x} + a \frac{\partial^2 T}{\partial x^2} = -u T_x + a T_{xx} \tag{20}$$

where a is the thermal diffusivity. The time discretization of equation (20) can be achieved by following the same steps outlined in the previous section. In fact, the only addition, with respect to equation (1), is the physical diffusion term that, on the other hand, can be discretized in the same way as the numerical diffusion term. Thus, for the typical node i, we obtain the assembled finite element equations

$$\frac{1}{6\Delta t}[(T_{i-1}^{n+1}+4T_i^{n+1}+T_{i+1}^{n+1})-(T_{i-1}^n+4T_i^n+T_{i+1}^n)]$$
$$=-\frac{u}{2\Delta x}[\alpha(T_{i+1}^{n+1}-T_{i-1}^{n+1})+(1-\alpha)(T_{i+1}^n-T_{i-1}^n)]$$
$$+\beta\frac{u^2\Delta t}{2\Delta x^2}[\gamma(T_{i-1}^{n+1}-2T_i^{n+1}+T_{i+1}^{n+1})+(1-\gamma)(T_{i-1}^n-2T_i^n+T_{i+1}^n)]$$
$$+\frac{a}{\Delta x^2}[\vartheta(T_{i-1}^{n+1}-2T_i^{n+1}+T_{i+1}^{n+1})+(1-\vartheta)(T_{i-1}^n-2T_i^n+T_{i+1}^n)] \quad (21)$$

and

$$\frac{1}{\Delta t}(T_i^{n+1}-T_i^n)=-\frac{u}{2\Delta x}[\alpha(T_{i+1}^{n+1}-T_{i-1}^{n+1})+(1-\alpha)(T_{i+1}^n-T_{i-1}^n)]$$
$$+\beta\frac{u^2\Delta t}{2\Delta x^2}[\gamma(T_{i-1}^{n+1}-2T_i^{n+1}+T_{i+1}^{n+1})+(1-\gamma)(T_{i-1}^n-2T_i^n+T_{i+1}^n)]$$
$$+\frac{a}{\Delta x^2}[\vartheta(T_{i-1}^{n+1}-2T_i^{n+1}+T_{i+1}^{n+1})+(1-\vartheta)(T_{i-1}^n-2T_i^n+T_{i+1}^n)] \quad (22)$$

for consistent and lumped capacity matrices, respectively. Clearly, the values of parameters α, β, γ and ϑ determine the particular BG or TG algorithms, as shown in Table II, where the first three letters and the subscript have the same meaning illustrated in the previous section. The last letter, instead, refers to the time discretization algorithm utilized for the diffusion term: E stands for explicit ($\vartheta=0$), C for Crank–Nicolson ($\vartheta=\frac{1}{2}$), and I for fully implicit ($\vartheta=1$). Again, we can point out that in one-dimensional problems, with elements of equal size, linear shape functions, and lumped capacity matrices, finite element algorithms lead to the same discretized equations that are yielded by finite differences.

Following the same steps illustrated in the previous section, we can also find the truncation errors of the various algorithms used for the solution of advection–diffusion problems. However, the only difference with respect to the results reported in Table I is that the addition of the physical diffusion terms changes to $O(\Delta x^2)$ the truncation error in the space discretization of all algorithms with consistent capacity matrices, while it does not alter the space accuracy of algorithms with lumped capacity matrices.

Equations (21) and (22) can be conveniently rewritten as

$$b_1T_{i-1}^{n+1}+(1-b_1-b_2)T_i^{n+1}+b_2T_{i+1}^{n+1}=d_1T_{i-1}^n+(1-d_1-d_2)T_i^n+d_2T_{i+1}^n \quad (23)$$

where

$$\begin{aligned} b_1&=\tfrac{1}{6}-\tfrac{1}{2}\alpha Co-\beta\gamma Fo_e-\vartheta Fo\\ b_2&=\tfrac{1}{6}+\tfrac{1}{2}\alpha Co-\beta\gamma Fo_e-\vartheta Fo\\ d_1&=\tfrac{1}{6}+\tfrac{1}{2}(1-\alpha)Co+\beta(1-\gamma)Fo_e+(1-\vartheta)Fo\\ d_2&=\tfrac{1}{6}-\tfrac{1}{2}(1-\alpha)Co+\beta(1-\gamma)Fo_e+(1-\vartheta)Fo \end{aligned} \quad (24)$$

Table II. Values of the parameters α, β, γ and ϑ to be used in Equations (21) and (22) to obtain the finite element schemes considered for the solution of the energy equation (20). The various schemes are characterized by different stability limits

Scheme	α	β	γ	ϑ	Stability limits
BG$_C$-EE	0	0	—	0	$Co \leqslant \frac{1}{6} Pe$; $Co \leqslant \frac{2}{Pe}$
BG$_L$-EE	0	0	—	0	$Co \leqslant \frac{1}{2} Pe$; $Co \leqslant \frac{2}{Pe}$
BG$_C$-CC	$\frac{1}{2}$	0	—	$\frac{1}{2}$	Unconditionally stable
BG$_L$-CC	$\frac{1}{2}$	0	—	$\frac{1}{2}$	Unconditionally stable
BG$_C$-II	1	0	—	1	Unconditionally stable
BG$_L$-II	1	0	—	1	Unconditionally stable
TG$_C$-EE	0	1	0	0	$Co \leqslant -\frac{1}{Pe} + \sqrt{\frac{1}{3} + \left(\frac{1}{Pe}\right)^2}$
TG$_L$-EE	0	1	0	0	$Co \leqslant -\frac{1}{Pe} + \sqrt{1 + \left(\frac{1}{Pe}\right)^2}$
TG$_C$-EC	0	1	0	$\frac{1}{2}$	$Co \leqslant \frac{\sqrt{3}}{3}$
TG$_L$-EC	0	1	0	$\frac{1}{2}$	$Co \leqslant 1$
TG$_C$-EI	0	1	0	1	$Co \leqslant \frac{1}{Pe} + \sqrt{\frac{1}{3} + \left(\frac{1}{Pe}\right)^2}$
TG$_L$-EI	0	1	0	1	$Co \leqslant \frac{1}{Pe} + \sqrt{1 + \left(\frac{1}{Pe}\right)^2}$
TG$_C$-CC	0	1	$\frac{1}{2}$	$\frac{1}{2}$	Unconditionally stable
TG$_L$-CC	0	1	$\frac{1}{2}$	$\frac{1}{2}$	Unconditionally stable
TG$_C$-II	0	1	1	1	Unconditionally stable
TG$_L$-II	0	1	1	1	Unconditionally stable

and the coefficients of the general scheme (21) with consistent capacity matrices, while

$$\begin{aligned} b_1 &= -\tfrac{1}{2}\alpha Co - \beta\gamma Fo_e - \vartheta Fo \\ b_2 &= \tfrac{1}{2}\alpha Co - \beta\gamma Fo_e - \vartheta Fo \\ d_1 &= \tfrac{1}{2}(1-\alpha)Co + \beta(1-\gamma)Fo_e + (1-\vartheta)Fo \\ d_2 &= -\tfrac{1}{2}(1-\alpha)Co + \beta(1-\gamma)Fo_e + (1-\vartheta)Fo \end{aligned} \tag{25}$$

are the coefficients of the general scheme (22) with lumped capacity matrices,

$$Co = u\frac{\Delta t}{\Delta x} \tag{26}$$

is the cell Courant number related to the advection term

$$Fo = a\frac{\Delta t}{\Delta x^2} \tag{27}$$

is the cell Fourier number related to the physical diffusion term, and

$$Fo_e = \left(\frac{u^2\Delta t}{2}\right)\frac{\Delta t}{\Delta x^2} = \frac{Co^2}{2} \tag{28}$$

is the equivalent cell Fourier number related to the diffusion-like term.

Following the von Neumann method for stability analysis, we assume that the numerical solution can be expressed by means of a Fourier series, whose typical term is

$$T^n_{i\pm 1} = F^n e^{i\sigma(x_i \pm \Delta x)} \tag{29}$$

where $i = \sqrt{-1}$ is the imaginary unit and σ is the wave number.[11] Substituting definition (29) into equation (23) and using the identities

$$e^{\pm i\sigma\Delta x} = \cos(\sigma\Delta x) \pm i\sin(\sigma\Delta x) \tag{30}$$

we obtain the amplification factor

$$G = \frac{F^{n+1}}{F^n} = \frac{1-(d_1+d_2)[1-\cos(\sigma\Delta x)] - i(d_1-d_2)\sin(\sigma\Delta x)}{1-(b_1+b_2)[1-\cos(\sigma\Delta x)] - i(b_1-b_2)\sin(\sigma\Delta x)} \tag{31}$$

The modulus of the amplification factor G must be lower than or equal to one if the solutions are to remain bounded. Thus, multiplying by the conjugate of G we obtain the condition which must be satisfied to have stability

$$|G|^2 = |G\bar{G}| = \frac{D_1\sin^4(\sigma\Delta x/2) + D_2\sin^2(\sigma\Delta x/2) + 1}{B_1\sin^4(\sigma\Delta x/2) + B_2\sin^2(\sigma\Delta x/2) + 1} \leqslant 1 \tag{32}$$

where

$$\begin{aligned} D_1 &= 16d_1d_2, \quad D_2 = 4[(d_1-d_2)^2 - (d_1+d_2)] \\ B_1 &= 16b_1b_2, \quad B_2 = 4[(b_1-b_2)^2 - (b_1+b_2)] \end{aligned} \tag{33}$$

It has been shown in Reference 11 that the necessary and sufficient conditions to satisfy equation (32) are

$$D_1 - B_1 + D_2 - B_2 \leqslant 0 \tag{34}$$

and

$$D_2 - B_2 \leqslant 0 \tag{35}$$

Condition (34) leads to the diffusion limit for stability, which is independent of the velocity u and is of the form

$$Fo = \frac{Co}{Pe} \leqslant K \tag{36}$$

In equation (36), K is a suitable numerical constant and Pe is the cell Peclet number,

$$Pe = \frac{Co}{Fo} = \left(\frac{u\Delta t}{\Delta x}\right)\left(\frac{\Delta x^2}{a\Delta t}\right) = \frac{u\Delta x}{a} \tag{37}$$

Condition (35), instead, leads to the advection limit for stability, which is of the form

$$Co \leqslant f\left(\frac{1}{Pe}\right) \tag{38}$$

and depends on the velocity. For the schemes referred to the Table II, limits (36) and (38) can be expressed by means of simple algebraic relations. If both limits are always satisfied, the algorithm is said to be unconditionally stable.

Among the Bubnov–Galerkin schemes, the Crank–Nicolson and the fully implicit algorithms are unconditionally stable, while the BG-EE schemes are characterized both by a diffusion and an advection limit. Taylor–Galerkin algorithms instead are either unconditionally stable or are characterized only by an advection limit.

STEADY-STATE PROBLEMS

It has already been pointed out that steady-state solutions can be found by means of a pseudo-transient simulation. Thus, it is worth analysing the behaviour of the finite element schemes for the advection–diffusion equation (20) when the steady state has been reached. In such a case, equation (20) becomes

$$u\frac{\partial T}{\partial x} - a\frac{\partial^2 T}{\partial x^2} = uT_x - aT_{xx} \tag{39}$$

while both equations (21) and (22) reduce to

$$\frac{u}{2\Delta x}(T_{i+1} - T_{i-1}) - \frac{1}{\Delta x^2}\left(a + \beta\frac{u^2\Delta t}{2}\right)(T_{i-1} - 2T_i + T_{i+1}) = 0 \tag{40}$$

or to the alternative form

$$-\left(\frac{1+\beta Co}{2} + \frac{1}{Pe}\right)T_{i-1} + \left(\beta Co + \frac{2}{Pe}\right)T_i - \left(\frac{1-\beta Co}{2} - \frac{1}{Pe}\right)T_{i+1} = 0 \tag{41}$$

It can be shown that, with a given mesh, wiggles can be avoided and temperature gradients can be correctly represented if the cell Peclet number does not become so large to change the sign of the coefficient of T_{i+1}. This requirement yields

$$\frac{1-\beta Co}{2} - \frac{1}{Pe} \leqslant 0 \tag{42}$$

Therefore, for Bubnov–Galerkin schemes ($\beta = 0$), we obtain

$$Pe \leqslant 2 \tag{43}$$

while, for Taylor–Galerkin schemes ($\beta = 1$), we obtain

$$Pe \leqslant \frac{2}{1 - Co} \tag{44}$$

i.e. the usual conditions to avoid, in the steady-state, the typical '$2\Delta x$' wave patterns.

Obviously, in the steady state, we can also use Petrov–Galerkin formulations[2] with various forms of weighting functions that lead to different 'upwind' finite element schemes. In one dimension, with elements of equal size, upwinding can be introduced using a weighting parameter λ in the assembled equation for the typical node i

$$\frac{u}{2\Delta x}[\lambda(T_i - T_{i-1}) + (1-\lambda)(T_{i+1} - T_{i-1})] + \frac{a}{\Delta x^2}(T_{i-1} - 2T_i + T_{i+1}) = 0 \tag{45}$$

We have full upwinding for $\lambda = 1$, and a central difference representation of the advection term for $\lambda = 0$. Equation (45) can also be written in the form

$$-\left(\frac{1+\lambda}{2}+\frac{1}{Pe}\right)T_{i-1}+\left(\lambda+\frac{2}{Pe}\right)T_i-\left(\frac{1-\lambda}{2}-\frac{1}{Pe}\right)T_{i+1}=0 \tag{46}$$

which is identical to equation (41) if β is equal to 1 and the cell Courant number is equal to the weighting parameter λ. This proves the equivalence of steady-state, Petrov–Galerkin solutions and pseudo-transient Taylor–Galerkin solutions. Since in one-dimensional problems, with elements of equal size, Petrov–Galerkin schemes yield solutions that are exact at nodes when an optimal weighting parameter

$$\lambda_{\mathrm{opt}} = \coth\left(\frac{Pe}{2}\right)-\frac{2}{Pe} \tag{47}$$

is used,[2] Taylor–Galerkin schemes can also be optimized by assuming $Co = \lambda_{\mathrm{opt}}$. Actually, if we use the full explicit lumped Taylor–Galerkin scheme (TG$_L$-EE), we do not even have to utilize equation (47) because it has been shown that nearly optimal, pseudo-transient solutions can be obtained by choosing a cell Courant number as close as possible to the limiting, or critical, value for stability.[2]

NUMERICAL EXPERIMENTS

The results of two numerical tests are reported here with the purpose of showing the relevant features of some of the algorithms discussed in this paper. In the numerical tests, for the sake of simplicity, we have used dimensionless variables x', u', t' and T', defined in terms of suitable reference values.

One-dimensional propagation of a cosine wave ($a = 0$)

A cosine temperature wave, propagating in a non-conductive fluid with constant, uniform velocity $u' = 1$, is considered. At $t' = 0$ the maximum ($T' = 1$) is located at $x' = 0$. The boundary conditions are $T' = 0$ at $x' \to \pm\infty$. Since the physical diffusion is equal to zero, the wave maintains its shape during the propagation. The numerical solutions are obtained using a uniform mesh of linear elements of size $\Delta x' = 0{\cdot}1$. Different time steps are chosen to obtain different cell Courant numbers. Selected temperature distributions at $t' = 8$ are shown in Figure 1 for six algorithms, namely the BG$_C$-C, BG$_C$-I, TG$_C$-E, TG$_L$-E, TG$_C$-C, TG$_C$-I schemes. The BG-E algorithm, because of its negative numerical diffusion, is always unstable in the case of pure advection, while the lumped capacity algorithms different from the TG$_L$-E are of little practical interest, and thus are not considered here.

As it appears in Figure 1(a), the BG$_C$-C algorithm is not affected by numerical diffusion, but is characterized by numerical dispersion errors which increase with the Courant number. From equation (17) we can see, in fact, that the terms containing third-order derivatives do not cancel out, thus yielding a significant dispersion error. The results reported in Figure 1(b) concern the BG$_C$-I algorithm which, according to equation (18), is affected by numerical diffusion and numerical dispersion. However, the numerical diffusion dominates, and the trailing waves are almost completely damped.

In Figure 1(c) and 1(d) we have reported the results obtained with the TG$_C$-E and TG$_L$-E algorithms, respectively. As we can see from equation (16), for $\beta = 1$ and $\gamma = 0$ the transient truncation error of the algorithm with consistent capacity matrices ($\delta = 1$) contains a third-order derivative whose coefficient is different from zero. Instead, with $\beta = 1, \gamma = 0$ and lumped capacity

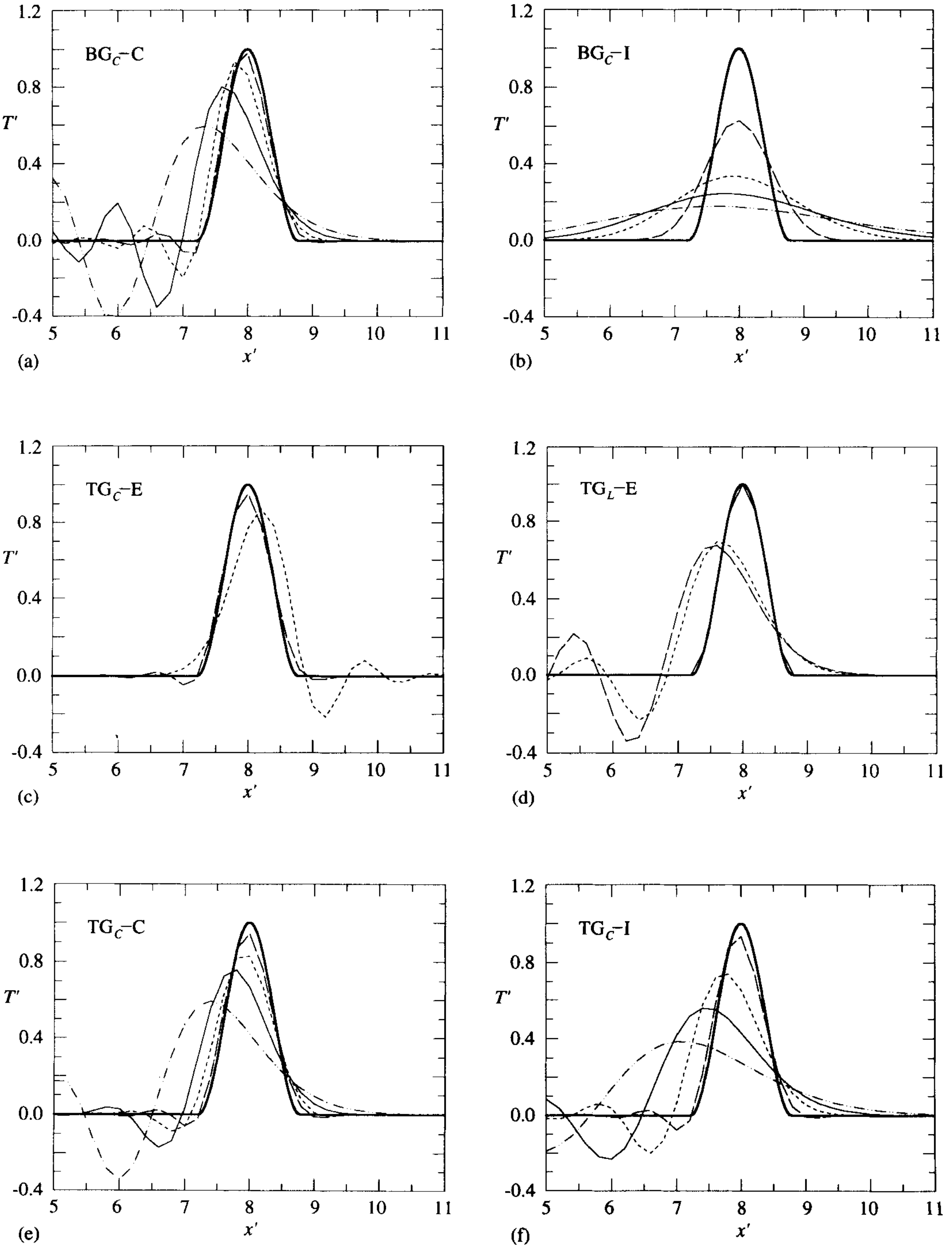

Figure 1. Propagation of a cosine wave ($a = 0$). Temperature distribution at $t = 8$, obtained with different algorithms and different cell Courant numbers: (——) exact; (— — —) $C_0 = 0.1$; (- - - -) $C_0 = 0{\cdot}5$; (——) $Co = 1$; (— - - —) $Co = 2$

matrices ($\delta = 0$), the truncation error (16) becomes

$$\varepsilon_t = \Delta t^2 \frac{u^3}{6} T_{xxx} - \Delta x^2 \frac{u}{6} T_{xxx} + O(\Delta t^3, \Delta x^4) \tag{48}$$

where, for $u\Delta t/\Delta x = Co = 1$, the terms containing third-order derivatives cancel out, thus minimizing the dispersion error. This is confirmed by Figure 1(d), where we can see how the solution improves when Co approaches the value of 1, which also represents the stability limit. For $Co = 1$ the numerical and analytical solutions are almost indistinguishable.

Finally, in Figures 1(e) and 1(f) we have reported the results obtained with the TG_C-C and TG_C-I algorithms, respectively. These schemes are unconditionally stable but, as we can see, they are affected by significant dispersion errors. From Figures 1(c), 1(e) and 1(f) it is apparent that the dispersion error produces strong leading waves with the TG_C-E algorithm ($\gamma = 0$), weak trailing waves with the TG_C-C algorithm ($\gamma = \frac{1}{2}$) and strong trailing waves with the TG_C-I algorithm ($\gamma = 1$). This means that the position of the waves moves forward for increasing γ, while their strength seems to reach a minimum at some intermediate value of γ between 0 and 1. As pointed out in Reference 15, dispersion errors can be minimized by using consistent capacity matrices ($\delta = 1$) and assuming $\gamma = \frac{1}{3}$. In this case, equation (16) becomes

$$\begin{aligned}\varepsilon_t &= \Delta t^2 \left(\frac{u^3}{6} T_{xxx} - \frac{1}{3}\frac{u^2}{2} T_{xxt}\right) - \Delta x^2 \left(\frac{u}{6} T_{xxx} + \frac{u}{6} T_{xxx}\right) + O(\Delta t^3, \Delta x^4) \\ &= O(\Delta t^3, \Delta x^4)\end{aligned} \tag{49}$$

where there are no terms containing third-order derivatives. This algorithm, however, is only conditionally stable,[15] with a limit that for pure advection is $Co = 1$.

Boundary-layer-type problem ($a > 0$)

The stationary solution to equation (20) in the domain $0 \leqslant x' \leqslant 1$, with a constant, uniform advection velocity $u' = 1$ and boundary conditions $T' = 0$ at $x' = 0$ and $T' = 1$ at $x' = 1$, is sought as a transient evolution from the initial temperature distributions $T' = x'$. The analytical solution to the problem is[2]

$$T' = \frac{1 - e^{Pe^* x'}}{1 - e^{Pe^*}} \tag{50}$$

where Pe^* is the global Peclet number evaluated in terms of the same reference length used in the definition of the dimensionless variable x'. The numerical tests have been performed using a time step $\Delta t' = 0{\cdot}1$ and a mesh consisting of 10 linear elements of size $\Delta' x = 0{\cdot}1$, yielding a cell Courant number $Co = 1$. The global Peclet number is $Pe^* = 25$, corresponding to a cell Peclet number $Pe = 2{\cdot}5$.

The results shown in Figure 2(a) are obtained using the BG_C-II algorithm, and represent the temperature profiles at different times during the pseudo-transient simulation. Since the cell Peclet number is greater than 2 the solutions present an oscillatory behaviour, characterized by the usual $2\Delta x$ pattern. It is worth noting that the numerical diffusion, which is typical of the BG_C-II algorithm in the transient, does not prevent spatial oscillations from appearing even in the early stages of the numerical simulation. The oscillatory solution obtained for $t' \rightarrow \infty$ is different from the analytical one, but is coincident with the finite element solution that would be found by directly solving the stationary problem. In fact, in the steady state, the BG_C-II algorithm does not produce any numerical diffusion. Repeating the simulation with a cell Peclet number lower

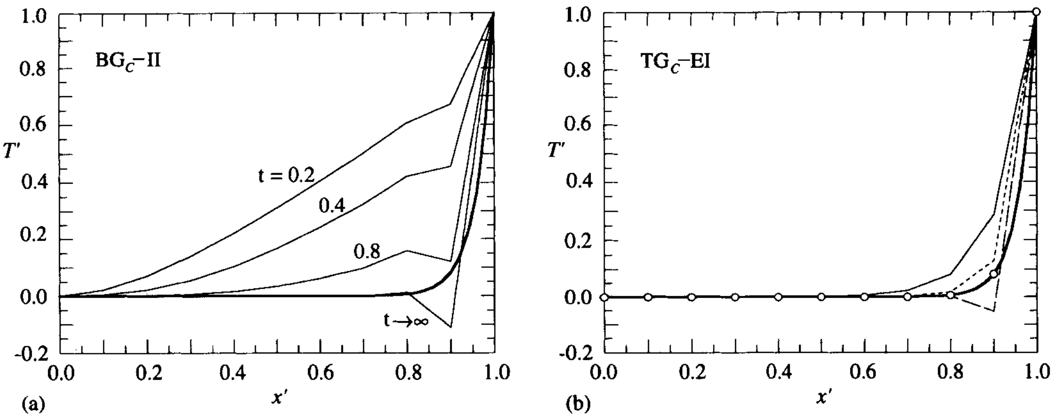

Figure 2. Boundary layer type problem ($a > 0$). Temperature distributions obtained: (a) in a pseudo-transient simulation, using the BG_C-II algorithm and $Co = 1$; (b) at steady state, using the TG_C-EI algorithm and different cell Courant numbers. (——) exact; (– – –) $Co = 0{\cdot}1$; (- - - -) $Co = 0{\cdot}5$; (——) $Co = 1$; (○) $Co = 0{\cdot}38$

than 2, we would find a steady-state solution which is nearly coincident with the analytical solution.[9]

In Figure 2(b) we report the steady-state solution of the same problem obtained with the TG_C-EI algorithm for different cell Courant numbers. We can see that this algorithm produces steady-state temperature distributions corresponding to values of effective diffusivity

$$a_e = a + a_{es} = a + \frac{u^2 \Delta t}{2} \tag{51}$$

higher than the physical diffusivity a in equation (39). The corresponding effective cell Peclet number is[8]

$$Pe_e = \frac{u\Delta x}{a_e} = \frac{1}{\dfrac{1}{Pe} + \dfrac{\Delta t}{2}u^2} = \frac{Pe}{1 + \dfrac{CoPe}{2}} \tag{52}$$

lower than the physical cell Peclet number. Therefore, the solution can be maintained oscillation-free if sufficiently high values of the cell Courant number are used. In Figure 2(b) it is also shown that exact nodal values can be obtained for $Co = Co_{\mathrm{opt}}$, which, in this case, is equal to 0·38, according to equation (47).

Similar results would have been obtained also by using other Taylor–Galerkin algorithms, since all of them, as shown in Table II, are affected by the same steady-state numerical diffusion.

CONCLUSIONS

The results outlined in the previous sections can lead to criteria for the choice of the most suitable algorithm for any given task. With reference to the Bubnov–Galerkin and Taylor–Galerkin schemes, we can offer the following guidelines.

Pseudo-transient simulations

In the solution of steady-state problems by means of pseudo-transient simulations, stability and absence of stationary numerical diffusion are characteristics of great interest. Therefore, fully implicit Bubnov–Galerkin (BG-II) schemes can be confidently employed since they allow the use of very large time steps to reach convergence and, in the steady state, are not affected by numerical diffusion. The only reason not to use BG-II schemes for pseudo-transient simulations can be the appearance of wiggles, connected with large cell Peclet numbers, i.e. with meshes that are too coarse with respect to the local velocity. If the required mesh refinement is too expensive, we can consider the alternative of using one of the Taylor–Galerkin schemes, characterized by a stationary numerical diffusion that plays the role of an upwinding term.

Transient simulations

In truly transient simulations, we can use Bubnov–Galerkin schemes of the Crank–Nicolson type (BG-CC) which are unconditionally stable and are not affected by numerical diffusion, even if they present significant numerical dispersion errors for relatively large cell Courant numbers. As an alternative, Taylor–Galerkin schemes might also be considered since they are not affected by numerical diffusion in transient calculations.

Taylor–Galerkin schemes

Many schemes of this type have been proposed and, therefore, their respective ranges of applicability are still the object of research. Here we can only say that fully explicit Taylor–Galerkin algorithms (TG-EE) can be very convenient with lumped capacity matrices since these algorithms only require the inversion of a diagonal matrix to march ahead in time. However, to have better stability characteristics, the TG-EI and the TG-II schemes can also be considered. The TG-EI schemes perform better than the TG-EE schemes with respect to stability, but are not unconditionally stable like the TG-II schemes. On the other hand, if a direct solver is used, with the TG-EI schemes it is not necessary to perform the system matrix factorization at every time step. In fact, since only the constant physical diffusion term is dealt with implicitly, a re-solution facility can be used. Instead with TG-II schemes, also the variable diffusion-like term is dealt with implicitly and, consequently, a new matrix factorization must be performed at every time step. Finally, we must remember that all Taylor–Galerkin algorithms are affected by dispersion errors which increase for increasing cell Courant numbers, thus deteriorating the overall accuracy of the solutions.

ACKNOWLEDGEMENTS

The support to this research by MURST and CNR is gratefully acknowledged.

REFERENCES

1. O. C. Zienkiewicz, R. Loehner, K. Morgan and S. Nakazawa, 'Finite elements in fluid mechanics—a decade of progress', in R. H. Gallagher, *et al.* (eds.), *Finite Elements in Fluids*, Vol. 5, Wiley, London, 1984, pp. 1–26.
2. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Vol. 2, 4th edn., McGraw-Hill, London, 1991.
3. T. J. R. Hughes and A. Brooks, 'A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: application to streamline-upwind procedure', in R. H. Gallagher, *et al.* (eds.), *Finite Element in Fluids*, Vol. 4, Wiley, London, 1982, pp. 47–65.
4. T. E. Tezduyar and D. K. Ganjoo, 'Petrov–Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: applications to transient convection–diffusion problems', *Comput. Methods Appl. Mech. Eng.*, **59**, 49–71 (1986).
5. C.-C. Yu and J. C. Heinrich, 'Petrov–Galerkin methods for the time-dependent convective transport equation', *Int. j. numer. methods fluids*, **23**, 883–901 (1986).
6. F. Shakib and T. J. R. Hughes, 'A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space–time Galerkin/least-square algorithms', *Comput. Methods Appl. Mech. Eng.*, **87**, 35–58 (1991).

7. R. Codina, 'Stability analysis of the forward Euler scheme for the convection-diffusion equation using the SUPG formulation in space', *Int. j. numer. methods fluids*, **36,** 1445–1464 (1993).
8. P. M. Gresho, S. T. Chan, R. L. Lee and C. D. Upson, 'A modified finite element method for solving the time-dependent, incompressible Navier–Stokes equations. Part 1: theory', *Int. j. numer. methods fluids*, **4,** 557–598 (1984).
9. G. Comini, S. Del Giudice and C. Nonino, 'Taylor–Galerkin algorithms for convection-type problems', in L. C. Wrobel, *et al.*, (eds.), *Advanced Computational Methods in Heat Transfer II*, Vol. 1, Computational Mechanics Publications, Southampton, 1992, pp. 535–554.
10. P. J. Roache, 'On artificial visocity', *J. Comput. Phys.*, **10,** 169–184 (1972).
11. C. Hirsch, *Numerical Computation of Internal and External Flows*, Vol. 1, Wiley, Chichester, UK, 1988.
12. D. W. Pepper and A. J. Baker, 'Finite differences versus finite elements', in W. J. Minkowycz *et al.* (eds.), *Handbook of Numerical Heat Transfer*, Chapter 13, Wiley, New York, 1988, pp. 519–577.
13. Y. Chen and R. A. Falconer, 'Advection–diffusion modelling using the modified QUICK scheme', *Int. j. numer. methods fluids*, **15,** 1171–1196 (1992).
14. H. Ninomiya and K. Onishi, *Flow Analysis Using a PC*, Computational Mechanics Publications, Southampton, 1991.
15. L. Bottura and O. C. Zienkiewicz, 'Experiments on iterative solution for the semi-implicit characteristic-Galerkin algorithm', *Commun. appl. numer. methods*, **6,** 387–393 (1990).