ELSEVIER

Comput. Methods Appl. Mech. Engrg. 156 (1998) 185–210

Computer methods in applied mechanics and engineering

Comparison of some finite element methods for solving the diffusion–convection–reaction equation

Ramon Codina

Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports, Universitat Politècnica de Catalunya, Gran Capità s/n, Edifici C1, 08034 Barcelona, Spain

Received 12 April 1997

Abstract

In this paper we describe several finite element methods for solving the diffusion–convection–reaction equation. None of them is new, although the presentation is non-standard in an effort to emphasize the similarities and differences between them. In particular, it is shown that the classical SUPG method is very similar to an explicit version of the Characteristic-Galerkin method, whereas the Taylor–Galerkin method has a stabilization effect similar to a sub-grid scale model, which is in turn related to the introduction of bubble functions. © 1998 Elsevier Science S.A.

1. Introduction

The objective of this paper is to compare several finite element methods for solving the linear diffusion–convection–reaction equation from the point of view of the formulation of the methods, describing the motivations that lead to them. Some of these methods have the transient problem as starting point, whereas the others are developed by considering first the stationary equations. Although none of them takes into account whether there is a reaction term in the equations or not, this will lead to an important difference between the methods, as we shall see. This 'reaction' term will be simply a term proportional to the unknown, thus having in fact the physical meaning of absorption for scalar equations.

The methods that will be described and the acronyms that will be used to refer to them are the following:

- SUPG: Streamline-upwind/Galerkin method [1].
- ST-GLS: Space–time Galerkin/least-squares method [2].
- SGS: Subgrid scale method [3–5].
- CG: Characteristic Galerkin method [6–9].
- TG: Taylor–Galerkin method [10].

Essentially, all these methods consist in the addition of a stabilizing term to the original Galerkin formulation of the problem. Except for minor modifications, this term can be written as the L^2 product within each element domain of the residual of the equation to be solved by an operator applied to the test function, the former being multiplied by a numerical parameter. The different methods differ in which is the operator acting on the test function and in the design of the algorithmic parameter. It has to be pointed out that the description of the different methods presented here is not necessarily the same as in the original references where the methods are first described. Our objective is trying to present them in such a way that the similarities and differences between them can be easily identified.

We have organized the paper as follows. The statement of the problem is presented in the next section. Although most of the methods are first derived for the scalar case, it is also important to consider the vector equations, since they introduce some additional features of the schemes. In particular, we shall comment on the

PII S0045-7825(97)00206-5

conservation properties of the methods to be described, a concept particularly relevant when these methods are applied to systems of conservation laws.

Sections 3–7 are devoted to the description of the different methods, trying to point out their particularities, but also stressing their similarities. In Section 8 we make some remarks about the satisfaction of the discrete maximum principle in very simple cases. This will highlight one of the major differences in the behavior of the methods in the presence of a reaction term. Also, this simple analysis will dictate how the algorithmic parameters of the methods must behave. At the end of this section we present some very simple numerical tests that will serve to see the difference just mentioned in the numerical answers. This is the only point in which we shall comment on the numerical merits of the different schemes.

Finally, in Section 9 we summarize all the methods described and collect some of the observations made along the paper.

2. Statement of the problem

2.1. Differential form

The partial differential equation that we want to solve numerically is, in the scalar transient case,

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{a}u - k\nabla u) + su = f \quad \text{in } \Omega\,, \quad t \in (0, T)\,, \tag{1}$$

where u is the (scalar) unknown, $\boldsymbol{a}$ is the convection velocity, $k > 0$ is the diffusion coefficient, $s \geqslant 0$ is the absorption coefficient and f is the source term. The n_{sd}-dimensional domain where the problem is to be solved has been denoted by Ω and the time interval by $[0, T]$. For simplicity, Eq. (1) will be supplied with the homogeneous Dirichlet boundary condition

$$u = 0 \quad \text{on } \partial\Omega\,, \quad t \in (0, T)\,, \tag{2}$$

and an initial condition of the form

$$u = u^0 \quad \text{in } \Omega\,, \quad t = 0\,. \tag{3}$$

Although most of the methods to be described in this paper start from the scalar equation (1), we will be also interested in its vector counterpart, that we write as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U}) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U} = \boldsymbol{F} \quad \text{in } \Omega\,, \quad t \in (0, T)\,, \tag{4}$$

where now $\boldsymbol{U}$ and $\boldsymbol{F}$ are vectors of n_{unk} unknowns and $\boldsymbol{A}_i$, $\boldsymbol{K}_{ij}$ and $\boldsymbol{S}$ are $n_{\mathrm{unk}} \times n_{\mathrm{unk}}$ matrices ($i, j = 1, \ldots, n_{\mathrm{sd}}$). The usual summation convention is implied in Eq. (4).

To simplify the discussion, we assume that the diffusion matrices $\boldsymbol{K}_{ij}$, verify $\boldsymbol{K}_{ij} = \boldsymbol{K}_{ji}^{\mathrm{t}}$ and the bilinear form $\boldsymbol{X}_i\boldsymbol{K}_{ij}\boldsymbol{Y}_j$, with $\boldsymbol{X}_i$ and $\boldsymbol{Y}_j$ vectors of n_{unk} components, is positive definite. Matrices $\boldsymbol{A}_i$ and $\boldsymbol{S}$ are not necessarily symmetric, but it is assumed that there is a matrix $\boldsymbol{T}$ associated to a linear change of variables such that $\hat{\boldsymbol{A}}_i = \boldsymbol{T}\boldsymbol{A}_i\boldsymbol{T}^{-1}$ are symmetric and if $\hat{\boldsymbol{S}} = \boldsymbol{T}\boldsymbol{S}\boldsymbol{T}^{-1}$ then $\partial\hat{\boldsymbol{A}}_i/\partial x_i + \hat{\boldsymbol{S}} + \hat{\boldsymbol{S}}^{\mathrm{t}}$ is positive semi-definite. As in the scalar case, we consider only homogeneous Dirichlet conditions. Under all these conditions, the problem is well posed.

It will be useful in what follows to introduce the following notation:

$$\mathscr{L}_{\mathrm{conv,c}}(\boldsymbol{U}) := \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U})\,, \tag{5}$$

$$\mathscr{L}_{\mathrm{ds}}(\boldsymbol{U}) := -\frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U}\,, \tag{6}$$

$$\mathscr{L}(\boldsymbol{U}) := \mathscr{L}_{\mathrm{conv,c}}(\boldsymbol{U}) + \mathscr{L}_{\mathrm{ds}}(\boldsymbol{U})\,. \tag{7}$$

Eq. (4) can now be written as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \mathscr{L}(\boldsymbol{U}) = \boldsymbol{F}\,. \tag{8}$$

Instead of writing Eq. (4) in the *conservative* or *divergence* form, we will be also interested in what we call the *non-conservative* version, in which the convective operator $\mathscr{L}_{\text{conv,c}}(\boldsymbol{U})$ is replaced by

$$\mathscr{L}_{\text{conv,nc}}(\boldsymbol{U}) := \boldsymbol{A}_i \frac{\partial \boldsymbol{U}}{\partial x_i} . \tag{9}$$

For the linear equation that we consider, both operators in Eqs. (5) and (9) give the same equation if, using the latter, matrix $\boldsymbol{S}$ is redefined as

$$\boldsymbol{S} \leftarrow \boldsymbol{S} + \frac{\partial \boldsymbol{A}_i}{\partial x_i} . \tag{10}$$

However, in the applications the expressions (5) or (9) for the convective term come from different forms of the differential equations to be solved (in other words, matrices $\boldsymbol{A}_i$ in (5) and (9) are different). These may be equivalent at the differential level (i.e. for smooth solutions), but in the case of nonlinear problems they may have different weak solutions, a point of special relevance when the numerical solution of these equations is considered. The importance of using the conservative form of the equations is well known in the literature (see e.g. [11]). We shall comment on the conservation properties of the different schemes to be considered. Unless otherwise stated, we shall always consider that the equation is written using the conservative operator (5). Nevertheless, the operator in Eq. (9) will be useful for presenting the different methods.

2.2. Weak form

In order to write the weak form of Eq. (4) with the boundary condition $\boldsymbol{U} = \boldsymbol{0}$ in $\partial\Omega$, let us introduce the space

$$\mathscr{W} := (H_0^1(\Omega))^{n_{\text{unk}}} , \tag{11}$$

that is, the space of vector functions the components of which are square integrable, have square integrable first derivatives and vanish on $\partial\Omega$. We assume that $\boldsymbol{F} \in (L^2(\Omega))^{n_{\text{unk}}}$ ($\boldsymbol{F}$ in the dual of space of $\mathscr{W}$ would be enough) and that all the coefficient matrices $\boldsymbol{A}_i$, $\boldsymbol{K}_{ij}$ and $\boldsymbol{S}$ are time independent and have bounded coefficients (i.e. have coefficients in $L^\infty(\Omega)$), the former having also bounded spatial derivatives.

Let $L^2(0, T; \mathscr{W})$ be the space of vector functions which for each fixed t belong to $\mathscr{W}$ and the $\mathscr{W}$-norm of which is square integrable in time. The weak form of the problem consists in finding a vector function $\boldsymbol{U} \in L^2(0, T; \mathscr{W})$ such that

$$\left(\boldsymbol{V}, \frac{\partial \boldsymbol{U}}{\partial t}\right) + a(\boldsymbol{U}, \boldsymbol{V}) - l(\boldsymbol{V}) = 0 \quad \forall \boldsymbol{V} \in \mathscr{W} , \tag{12}$$

where, using the matrix notation to denote the scalar product of two vectors,

$$\left(\boldsymbol{V}, \frac{\partial \boldsymbol{U}}{\partial t}\right) := \int_\Omega \boldsymbol{V}^{\text{t}} \frac{\partial \boldsymbol{U}}{\partial t} \, \mathrm{d}\Omega , \tag{13}$$

$$a(\boldsymbol{U}, \boldsymbol{V}) := \int_\Omega \boldsymbol{V}^{\text{t}} \frac{\partial}{\partial x_i} (\boldsymbol{A}_i \boldsymbol{U}) \, \mathrm{d}\Omega + \int_\Omega \frac{\partial \boldsymbol{V}^{\text{t}}}{\partial x_i} \boldsymbol{K}_{ij} \frac{\partial \boldsymbol{U}}{\partial x_j} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{V}^{\text{t}} \boldsymbol{S} \boldsymbol{U} \, \mathrm{d}\Omega , \tag{14}$$

$$l(\boldsymbol{V}) := \int_\Omega \boldsymbol{V}^{\text{t}} \boldsymbol{F} \, \mathrm{d}\Omega . \tag{15}$$

Although this is the standard (continuous) weak form of the problem, the derivation of the stabilization methods does not usually start from it. For example, the GLS method is traditionally based on a space–time weak form, whereas the TG and the CG methods start from a particular time discretization of the differential equations before obtaining their weak form, proceeding then to their spatial approximation. We shall treat this point in more detail in the following sections.

2.3. General expression of the stabilization methods

Let us consider a finite element partition $\{\Omega^e\}_{e=1}^{n_{\text{el}}}$ of the domain Ω and let $\mathcal{W}_h$ be the associated finite element space to approximate $\mathcal{W}$ ($\mathcal{W}_h \subset \mathcal{W}$). The standard Galerkin method applied to Eq. (14) consists in finding $U_h \in L^2(0,T;\ \mathcal{W}_h)$ such that

$$\left(V_h, \frac{\partial U_h}{\partial t}\right) + a(U_h, V_h) - l(V_h) = 0 \quad \forall V_h \in \mathcal{W}_h\,. \tag{16}$$

This equation can now be discretized in time using finite differences or finite elements.

All the stabilized methods that will be described in what follows consist in adding to the left-hand side of Eq. (16) (or to an appropriate time-discrete version of it) a term of the form

$$\boxed{r(U_h, V_h) = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \mathcal{P}^e(V_h)^{\text{t}} \boldsymbol{\tau}^e \mathcal{R}^e(U_h)\, \mathrm{d}\Omega} \tag{17}$$

where $\boldsymbol{\tau}$ is a $n_{\text{unk}} \times n_{\text{unk}}$ matrix of algorithmic parameters with dimensions of time, $\mathcal{P}(V_h)$ is a certain operator applied to the test function and $\mathcal{R}(U_h)$ is a residual of the differential equation to be solved. All these terms will be specified later on for each particular method. The superscript e in Eq. (17) has been used to indicate that the terms are evaluated elementwise (when space–time finite elements are used, these elements have to be considered also in the space–time domain). This superscript will be omitted in the description of the different methods. We shall write also

$$\int_{\Omega'} := \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e}\,. \tag{18}$$

2.4. Conservation properties

Let us consider the case in which $\boldsymbol{F} = \boldsymbol{0}$ (and thus $l(V) = 0$) and $\boldsymbol{S} = \boldsymbol{0}$. Suppose also that no boundary conditions are prescribed on $\partial\Omega$. If we define the advective and diffusive fluxes of the unknown $\boldsymbol{U}$ as

$$\boldsymbol{F}_i^{\text{adv}} := \boldsymbol{A}_i \boldsymbol{U}\,, \qquad \boldsymbol{F}_i^{\text{diff}} := -\boldsymbol{K}_{ij} \frac{\partial \boldsymbol{U}}{\partial x_j}\,, \tag{19}$$

the bilinear form that should appear in Eq. (12) is

$$a(\boldsymbol{U}, \boldsymbol{V}) = \int_\Omega \boldsymbol{V}^{\text{t}} \frac{\partial}{\partial x_i} \boldsymbol{F}_i^{\text{adv}}\, \mathrm{d}\Omega - \int_\Omega \frac{\partial \boldsymbol{V}^{\text{t}}}{\partial x_i} \boldsymbol{F}_i^{\text{diff}}\, \mathrm{d}\Omega + \int_{\partial\Omega} \boldsymbol{V}^{\text{t}} n_i \boldsymbol{F}_i^{\text{diff}}\, \mathrm{d}\Gamma\,, \tag{20}$$

where n_i is the ith component of the external unit normal $\boldsymbol{n}$. If we take $\boldsymbol{V}$ constant in Ω and use the divergence theorem for the first integral in Eq. (20), this reduces to

$$a(\boldsymbol{U}, \boldsymbol{V}) = \boldsymbol{V}^{\text{t}} \int_{\partial\Omega} n_i (\boldsymbol{F}_i^{\text{adv}} + \boldsymbol{F}_i^{\text{diff}})\, \mathrm{d}\Gamma\,, \tag{21}$$

and the integral statement of Eq. (4) reads now

$$\boldsymbol{V}^{\text{t}} \left[\int_\Omega \frac{\partial \boldsymbol{U}}{\partial t} + \int_{\partial\Omega} n_i (\boldsymbol{F}_i^{\text{adv}} + \boldsymbol{F}_i^{\text{diff}})\, \mathrm{d}\Gamma \right] = 0\,, \tag{22}$$

for all constant vectors $\boldsymbol{V}$. This implies that the bracketed term must be zero, which is a global conservation statement for the unknown $\boldsymbol{U}$, its fluxes being $\boldsymbol{F}_i^{\text{adv}}$ and $\boldsymbol{F}_i^{\text{diff}}$.

Obviously, the discrete problem will inherit this property provided that the stabilizing term defined in Eq. (17) verifies

$$r(\boldsymbol{U}_h, \boldsymbol{V}_h) = 0\,, \tag{23}$$

for all constant vectors V_h. This will in turn happen if

$$\mathscr{P}^e(V_h) = \mathbf{0} \quad \forall\, V_h \text{ constant in } \Omega^e\,, \quad e = 1, \ldots, n_{\mathrm{el}}\,. \tag{24}$$

If condition (24) holds (for $\boldsymbol{S} = \mathbf{0}$), the discrete formulation will be globally conservative for the situation considered. In fact, boundary conditions for $\boldsymbol{U}$ can be also prescribed, since they can be replaced by consistent fluxes computed by imposing that the discrete weak form of the problem holds also for any V_h not necessarily zero on $\partial\Omega$ [2].

3. The SUPG method

The original SUPG formulation was first designed for the steady version of Eq. (1) as a method to avoid the numerical oscillations found using the classical Galerkin approach when k is very small. It was already known from the experience in using centered finite difference methods that this misbehavior can be avoided by introducing numerical diffusion [12]. A further improvement was to introduce this diffusion only along the streamlines [13,14], leading to numerical schemes less overdiffusive, particularly in the crosswind direction. The final step was to put the introduction of the streamline diffusion in the context of weighted residual methods [15], leading to the SUPG method that up to now has been extensively used in convection dominated problems.

The extension to the transient problem that we consider here is based on a previous discretization in time of the continuous equation (1). Suppose for example that this discretization is done according to the generalized trapezoidal rule. Let Δt be the time step size of a (for simplicity) uniform partition of the time interval $[0, T]$. If we use a superscript to refer to the time step counter and define

$$v^{n+\theta} := \theta v^{n+1} + (1-\theta)v^n\,, \qquad \Delta v^n := v^{n+1} - v^n \tag{25}$$

for any function v and for $0 \leqslant \theta \leqslant 1$, the generalized trapezoidal rule applied to Eq. (1) consists in, given u^n, find u^{n+1} satisfying the boundary conditions and

$$\frac{\Delta u^n}{\Delta t} + \mathscr{L}(u^{n+\theta}) = f\,, \tag{26}$$

where $\mathscr{L}$ is now the scalar counterpart of the operator defined in Eq. (7). Particular cases of interest are $\theta = 1$ (backward Euler), $\theta = 1/2$ (Crank–Nicolson) and $\theta = 0$ (forward Euler). From u^k, $k = 0, 1, 2, \ldots$, one can construct an approximation in time to u of second order if $\theta = 1/2$ and of first order in the rest of the cases.

The standard Galerkin method applied to Eq. (26) consists in finding a finite element approximation u_h^{n+1} to u^{n+1} such that

$$\left(v_h, \frac{\Delta u_h^n}{\Delta t}\right) + a(u_h^{n+\theta}, v_h) - l(v_h) = 0 \quad \forall\, v_h \in \mathscr{W}_h\,, \tag{27}$$

which is the time discrete version of Eq. (16) when the generalized trapezoidal rule is employed for the temporal approximation.

The stabilizing term introduced by the SUPG method has the form given by Eq. (17), where now $\mathscr{P}(v_h)$ is the non-conservation form of the convective operator applied to the test function and $\mathscr{R}(u_h)$ is the residual of Eq. (26), that is,

$$\mathscr{P}_{\mathrm{SUPG}}(v_h) = \mathscr{L}_{\mathrm{conv,nc}}(v_h) = \boldsymbol{a} \cdot \nabla v_h\,, \tag{28a}$$

$$\begin{aligned}\mathscr{R}_{\mathrm{SUPG}}(u_h) &= \frac{\Delta u_h^n}{\Delta t} + \mathscr{L}(u_h^{n+\theta}) - f \\ &= \frac{\Delta u_h^n}{\Delta t} + \nabla \cdot (\boldsymbol{a} u_h^{n+\theta} - k \nabla u_h^{n+\theta}) + s u_h^{n+\theta} - f\,.\end{aligned} \tag{28b}$$

The SUPG method is consistent, in the sense that the stabilizing term given by Eq. (17) with $\mathscr{R}(u_h)$ defined in Eq. (28b) is zero if u_h is the solution of the continuous (in space) Eq. (26).

It remains to define the algorithmic parameter τ, which is often called ‘intrinsic time’. The way in which it was originally computed goes back to the original idea of the SUPG method, that is, to add numerical diffusion. For that, let us consider the simple one-dimensional model equation:

$$a\frac{\mathrm{d}u}{\mathrm{d}x} - k\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = 0\,, \quad 0 < x < 1\,, \tag{29}$$

with $u(0)$ and $u(1)$ given. If the partition of $[0, 1]$ is uniform, h being the element size, and linear elements are employed, it can be shown that the numerical solution is nodally exact if

$$\tau = \frac{\alpha h}{2a}\,, \tag{30}$$

where

$$\alpha(\mathrm{Pe}) = \coth(\mathrm{Pe}) - \frac{1}{\mathrm{Pe}}\,, \quad \mathrm{Pe} := \frac{ah}{2k}\,. \tag{31}$$

In Eq. (31), Pe is the so-called (cell) Péclet number and α is the upwind function (a different expression for α is obtained when quadratic elements are used [16]).

In the general case, the strategy usually adopted is to compute τ in Eq. (17) using the straightforward extension from the 1D case, perhaps with slight 'ad hoc' modifications to improve the accuracy in time [1]. More recently, other ways of computing τ have been proposed on the basis of the convergence analysis of the method, although this has been done mainly for the method to be described in the next section.

The extension of the SUPG method to the vector equation (4) is obvious, except for the definition of τ, that now is a matrix of algorithmic parameters. The expressions of $\mathscr{P}$ and $\mathscr{R}$ are

$$\boxed{\begin{aligned} \mathscr{P}_{\mathrm{SUPG}}(V_h) &= \mathscr{L}_{\mathrm{conv,nc}}(V_h) = A_i\frac{\partial V_h}{\partial x_i}\,, \\ \mathscr{R}_{\mathrm{SUPG}}(U_h) &= \frac{\Delta U_h^n}{\Delta t} + \mathscr{L}(U_h^{n+\theta}) - F \\ &= \frac{\Delta U_h^n}{\Delta t} + \frac{\partial}{\partial x_i}(A_i U_h^{n+\theta}) - \frac{\partial}{\partial x_i}\left(K_{ij}\frac{\partial U_h^{n+\theta}}{\partial x_j}\right) + SU_h^{n+\theta} - F \end{aligned}} \tag{32}$$

The definition of τ is not so obvious. The ways to compute it that have been proposed are all based on the idea of reproducing what happens in the scalar case when the equations in Eq. (4) can be uncoupled. This is possible if all the matrices A_i and K_{ij} can be diagonalized in the same basis, which is not true in most applications of the method. This problem was first discussed in detail in [17].

REMARKS

(1) In the extension to the vector case given by Eq. (32) it is difficult to see 'physically' how the SUPG method acts, that is, the numerical diffusion that it introduces. There is another way of extending the SUPG methods to systems that has been widely used in practice. The idea is to consider the system of equations uncoupled at the moment of applying the SUPG ideas. Suppose that

$$\mathrm{diag}(A_i) = \mathrm{diag}(a_{i,1}, \ldots, a_{i,n_{\mathrm{unk}}})\,, \tag{33}$$

where $a_{i,k}$ is the ith component of the advection speed a_k of the kth equation. If $V_{h,k}$ is the kth component of V_h, we could also define

$$\mathscr{P}_{\mathrm{SUPG}}(V_h) = \mathrm{diag}\,\mathscr{L}_{\mathrm{conv,nc}}(V_h) := \mathrm{diag}(A_i)\frac{\partial V_h}{\partial x_i} = \left(a_{i,1}\frac{\partial V_{h,1}}{\partial x_i}, \ldots, a_{i,n_{\mathrm{unk}}}\frac{\partial V_{h,n_{\mathrm{unk}}}}{\partial x_i}\right)^{\mathrm{t}} \tag{34}$$

instead of $\mathscr{P}_{\text{SUPG}}(V_h)$ in Eq. (32). Moreover, the 'natural' way to compute $\boldsymbol{\tau}$ in this case is simpler, since it can be taken as

$$\boldsymbol{\tau} = \text{diag}(\tau_1, \ldots, \tau_{n_{\text{unk}}}) , \tag{35}$$

where τ_k is computed for the kth scalar equation using $\boldsymbol{a}_k$ as advection speed and a characteristic diffusion of this equation. The drawback of this approach is that the definition of $\tilde{\mathscr{P}}_{\text{SUPG}}(V_h)$ in Eq. (34) depends on the variables U used, that is, it is not a definition intrinsic to the equation to be solved. We shall come back to this point later on.

(2) It has to be noted that the term $r(U_h, V_h)$ in Eq. (17) with $\mathscr{R}(U_h)$ defined in Eq. (32) will contribute to the mass matrix of the final algebraic system, that is, the matrix that multiplies the discrete temporal derivative. For the scalar case (Eq. (1)), it can be easily checked that this contribution is skew-symmetric if τ is the same for all the elements (which, in general, is not the case), $\boldsymbol{a}$ is divergence free and $u_h = 0$ on $\partial\Omega$. This is due to the fact that, in this case

$$\sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau^e(\boldsymbol{a} \cdot \nabla u_h) u_h \,\mathrm{d}\Omega = \tau \int_\Omega \nabla \cdot \left(\boldsymbol{a} \frac{u_h^2}{2} \right) \mathrm{d}\Omega = \tau \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{a} \frac{u_h^2}{2} \,\mathrm{d}\Omega = 0 . \tag{36}$$

This type of contribution to the mass matrix does not appear if the SUPG method is used in a space–time finite element discretization of the problem as explained in the following section. Although this is the approach advocated in [2] as the 'standard' SUPG method, it is not what is commonly used in practice. This point is further discussed below.

(3) Clearly, the SUPG method defined by Eq. (32) is conservative, since condition (24) is verified.

4. The space–time Galerkin/least-squares method

In the previous section we have described the SUPG method as a finite element formulation to discretize in space a partial differential equation, assuming that the temporal discretization has been already carried out. In particular, we have considered that this discretization has been done using the generalized trapezoidal rule.

Although the approach described above is very common in practical computations, Johnson et al. proposed to use the SUPG method together with a finite element discretization also in space [18], based on the use of the discontinuous Galerkin method in time introduced by Lesaint and Raviart for the space discretization of transport equations [19]. The idea behind this was to be able to treat the temporal derivative like the first spatial derivatives.

On the other hand, Hughes et al. [20] found that if in the classical Stokes problem for an incompressible fluid the pressure gradient is viewed as a 'convective term' and a SUPG-like strategy is employed for it, it is possible to avoid the need for using different finite element interpolations for the velocity and the pressure satisfying the so-called Babuška–Brezzi stability condition (see e.g. [21]), which is needed if the standard Galerkin approach is employed. The method proposed first was based on perturbing the original velocity test function of the Galerkin method with a term proportional to the gradient of the pressure test function. The next step was to consider all the Stokes operator applied to the test functions as perturbing term [22,23]. Going back to the convection–diffusion equation, this idea led to the so-called Galerkin/least-squares (GLS) method [2], which is naturally used together with the space–time approach described earlier [24]. In what follows, we consider this space–time method as the natural extension of the GLS method for the steady-state problem, and we refer to them as the space–time Galerkin/least-squares (ST–GLS) formulation.

Before writing down the equations for the ST-GLS method, let us apply the discontinuous Galerkin (DG) method to Eq. (1). To simplify the notation, we consider first the scalar equation, although the following ideas can be directly applied to systems of equations.

Let $t^n = n\,\Delta t$ and $I^n = [t^n, t^{n+1}]$. The idea of the DG method is to discretize an integral form of the problem to be solved in the space–time slab $Q^n = \Omega \times I^n$, enforcing weakly the continuity of the unknown function at time t^n. Both this unknown function and the test functions are allowed to be discontinuous between different space–time slabs. Moreover, these can be discretized using completely independent finite element partitions. The simplest way to construct the element domains is to discretize Ω and to take the elements of Q^n of the form

$\Omega^e \times I^n$, where Ω^e is an element of the partition of Ω. However, there is no need at all to consider elements prismatic in time.

Let us denote by v^n_+ the upper limit as $t \to t^n$ of a function v of time and by v^n_- the lower limit. The weak form of Eq. (1) in the space–time slab Q^n enforcing weakly the continuity condition $u^n_+ = u^n_-$ for the finite element approximation u_h leads to

$$\int_{Q^n} \left[v_h \frac{\partial u_h}{\partial t} + v_h \nabla \cdot (\boldsymbol{a} u_h) + k \nabla v_h \cdot \nabla u_h + s v_h u_h \right] \mathrm{d}\Omega \, \mathrm{d}t + \int_\Omega v^n_{h,+} (u^n_{h,+} - u^n_{h,-}) \, \mathrm{d}\Omega = \int_{Q^n} v_h f \, \mathrm{d}\Omega \, \mathrm{d}t \,. \tag{37}$$

This equation must hold for all the test functions v_h defined in the time slab Q^n. If we use the definition of the bilinear form a given in Eq. (14) (now for the scalar case), Eq. (37) can also be written as

$$\int_{I^n} \left[\left(v_h, \frac{\partial u_h}{\partial t} \right) + a(u_h, v_h) \right] \mathrm{d}t + (v^n_{h,+}, u^n_{h,+} - u^n_{h,-}) = \int_{I^n} (v_h, f) \, \mathrm{d}t \,. \tag{38}$$

The ST-GLS method can now be formulated easily. The idea is to add to the basic discontinuous Galerkin method given by Eq. (38) a least-squares form of the residual of Eq. (1). This leads to the addition of a stabilizing term of the form indicated by Eq. (8), where now the element domains Ω^e must be understood as space time elements and $\mathscr{P}$ and $\mathscr{R}$ are

$$\mathscr{P}_{\text{ST-GLS}}(v_h) = \frac{\partial v_h}{\partial t} + \mathscr{L}(v_h) \,, \tag{39a}$$

$$\mathscr{R}_{\text{ST-GLS}}(u_h) = \frac{\partial u_h}{\partial t} + \mathscr{L}(u_h) - f \,. \tag{39b}$$

As for the SUPG method, it remains to define how to compute the algorithmic parameter τ. Let us consider first the steady-state problem. The convergence analysis in this case dictates that τ must behave as [2]

$$\tau = \begin{cases} C \dfrac{h^2}{k} & \text{when Pe is small} \\ C' \dfrac{h}{|\boldsymbol{a}|} & \text{when Pe is high} \end{cases} \tag{40}$$

where C and C' are positive constants independent of the mesh size and the Péclet number Pe. It is also determined from the convergence analysis when the expression for τ must change from one case to the other. Condition (40) is fulfilled if τ is taken as

$$\tau = \frac{\alpha h}{2|\boldsymbol{a}|} \,, \quad \text{with} \quad \alpha = \min\{C_1 \,\text{Pe}, C_2\} \,. \tag{41}$$

The problem now is how to compute the constants C_1 and C_2. From the analysis of the convergence of the method it is found that they are related to the constants appearing in the interpolation error of the finite element approximation used and also in inverse estimates [2]. Therefore, a value for these constants is needed in order to obtain an expression for τ. However, this can be achieved only in some simple situations. See [25] for further discussion about this point.

There is also the possibility of using for the GLS method the same parameter τ as for the SUPG method. In fact, the function α appearing in Eq. (41) may be viewed as an asymptotic approximation of the upwind function given in Eq. (30), in this case with $C_1 = 1/3$ and $C_2 = 1$. Also, for quadratic elements it is found that one can take $C_1 = 1/9$ and $C_2 = 1/2$ if the pointwise error for the one-dimensional model problem (29) is to be minimized [16]. Finally, for transient problems (that is, for the ST-GLS method) several expressions for τ have been proposed based on the minimization of the error for some model problems (see e.g. [26]).

The choice of the parameter τ or, equivalently, the upwind function α, has been a controversial issue in the use of both the SUPG and the GLS methods. Another expression for it can be found in the context of the method described in the next section.

As for the SUPG method, the extension to the vector case is straightforward, except for the definition of $\boldsymbol{\tau}$. The vector counterpart of Eq. (39) is

$$\boxed{\begin{aligned}
\mathscr{P}_{\text{ST-GLS}}(\boldsymbol{V}_h) &= \frac{\partial \boldsymbol{V}_h}{\partial t} + \mathscr{L}(\boldsymbol{V}_h)\\
&= \frac{\partial \boldsymbol{V}_h}{\partial t} + \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{V}_h) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{V}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{V}_h\\
\mathscr{R}_{\text{ST-GLS}}(\boldsymbol{U}_h) &= \frac{\partial \boldsymbol{U}_h}{\partial t} + \mathscr{L}(\boldsymbol{U}_h) - \boldsymbol{F}\\
&= \frac{\partial \boldsymbol{U}_h}{\partial t} + \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U}_h) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U}_h - \boldsymbol{F}
\end{aligned}} \tag{42}$$

REMARKS

(1) Observe that the ST-GLS method does *not* satisfy condition (24) unless $\partial \boldsymbol{A}_i/\partial x_i = \boldsymbol{0}$ and therefore the global conservation property (22) will not hold in general.

(2) For comparison purposes, it is interesting to obtain the equations for the ST-GLS method in the particular case of constant-in-time interpolation. In this case, let us write

$$u_h^{n+1} \equiv u_{h,+}^{n} = u_{h,-}^{n+1}, \tag{43}$$

and similarly for the test functions, for which we omit the superscript since they will be the same for all time intervals. The discontinuous Galerkin method given by Eq. (38) now reduces to

$$\Delta t\, a(u_h, v_h) + (v_{h,+}^n, u_{h,+}^n - u_{h,-}^n) = \int_{I^n} (v_h, f)\, \mathrm{d}t\,. \tag{44}$$

The left-hand side of this equation may be written as

$$\int_{Q^n} v_h f\, \mathrm{d}\Omega\, \mathrm{d}t = \Delta t \int_{\Omega} v_h\left(\frac{1}{\Delta t}\int_{I^n} f\, \mathrm{d}t\right) \mathrm{d}\Omega\,. \tag{45}$$

If we consider f time dependent and *continuous in time*, in Eq. (27) we have that

$$l(v_h) = \int_{\Omega} v_h f^{n+\theta}\, \mathrm{d}\Omega\,, \tag{46}$$

from where it is seen that Eq. (44) is the same as Eq. (27) for $\theta = 1$ but taking an average of f in the time interval I^n instead of the value at time t^{n+1}.

(3) Again in the case of constant-in-time interpolation, the terms $\mathscr{P}$ and $\mathscr{R}$ given in Eq. (39) now reduce to

$$\mathscr{P}_{\text{ST-GLS}}(v_h) = \mathscr{L}(v_h)\,, \tag{47a}$$

$$\mathscr{R}_{\text{ST-GLS}}(u_h) = \mathscr{L}(u_h) - f\,. \tag{47b}$$

Since the time derivative of u_h does not appear in $\mathscr{R}$, the ST-GLS method will not modify the mass matrix resulting from the DG method given by Eq. (44). This is an important difference between this method and the SUPG method as presented in the previous section. Also, in the general case, that is, for interpolations in time not necessarily constant, the contribution to the mass matrix provided by the ST-GLS method will be symmetric, in contrast to what happens using the SUPG method after a finite difference time discretization.

(4) In [2], and motivated by the analysis in [18], the SUPG method is used together with the discontinuous Galerkin method in time. The terms $\mathscr{P}$ and $\mathscr{R}$ are then defined as

$$\mathscr{P}_{\text{SUPG}}(\boldsymbol{V}_h) = \frac{\partial \boldsymbol{V}_h}{\partial t} + \mathscr{L}_{\text{conv,nc}}(\boldsymbol{V}_h)\,, \tag{48a}$$

$$\mathscr{R}_{\text{SUPG}}(\boldsymbol{U}_h) = \frac{\partial \boldsymbol{U}_h}{\partial t} + \mathscr{L}(\boldsymbol{U}_h) - \boldsymbol{F}\,. \tag{48b}$$

5. The subgrid scale method

5.1. Basic concepts

The method that is presented in this section can be derived using different arguments. Two of them will be briefly drawn here, but also the classical Taylor–Galerkin method described in Section 7 may be viewed as a subgrid scale method. However, the TG method is derived from the transient problem, whereas for our purposes now it suffices to consider only the steady-state case. Also, for the sake of simplicity in the notation we concentrate in the scalar equation. The variational formulation of the continuous problem consists then in finding u such that

$$a(u, v) = l(v) \tag{49}$$

for all test functions v.

The subgrid scale (SGS) methods were first introduced by Hughes in [3,4], which we follow closely here. As we shall see, the SGS methods are in fact a family of stabilization techniques the first of which is perhaps that due to Douglas and Wang [27] for the Stokes problem. There they presented a finite element formulation allowing equal velocity–pressure interpolation similar to the GLS method, the only difference being the sign of the viscous operator applied to the test function. Instead of taking $-\nu\nabla^2 \boldsymbol{v}_h$ (∇^2: Laplacian, ν: kinematic viscosity, $\boldsymbol{v}_h$: velocity test function) they took $+\nu\nabla^2\boldsymbol{v}_h$. This method was later on generalized by Franca et al. [28] and applied to the convection–diffusion equation. The method they ended up with is similar to the GLS method but instead of taking (for the steady-state problem)

$$\mathcal{P}_{\mathrm{GLS}}(v_h) = \mathcal{L}(v_h) = \nabla\cdot(\boldsymbol{a}v_h) - \nabla\cdot(k\nabla v_h) + sv_h \tag{50}$$

they took

$$\mathcal{P}_{\mathrm{SGS}}(v_h) = -\mathcal{L}^*(v_h) = \boldsymbol{a}\cdot\nabla v_h + \nabla\cdot(k\nabla v_h) - sv_h\,, \tag{51}$$

where $\mathcal{L}^*$ is the adjoint operator of $\mathcal{L}$. Observe that for the boundary condition $u = 0$ on $\partial\Omega$ that we have considered, the diffusive operator $\nabla\cdot(k\nabla(\cdot))$ is self-adjoint. The convergence analysis of the method with $\mathcal{P}$ given in Eq. (51) and $\nabla\cdot\boldsymbol{a} = 0$ can be found in [28].

The SGS methods presented in [3,4] are a generalization of the above stabilization procedure, which, as we shall see, can be recovered as a particular case. Also, particular cases are the stabilization methods based on the introduction of bubble functions to the finite element space. They first attracted interest since it was recognized that the GLS method for the Stokes problem using linear elements is equivalent (up to the choice of the algorithmic parameters) to the use of the Galerkin method with linear elements enriched with bubble functions [29,30], which are known to be stable [31]. This connection was later on exploited by several authors (see e.g. [32]), who proposed different stabilization procedures based on the use of different bubble functions. However, these techniques are related with the use of the term $\mathcal{P}$ given in Eq. (51), and not with the GLS method, as it was pointed out in [5].

Let us describe now the idea of the SGS methods presented by Hughes in [3,4]. Suppose that the unknown u is split as $u = \bar{u} + u'$, where $\bar{u}$ is the part of u which can be represented by the finite element mesh, whereas u' accounts for the unresolvable scales of u, that is, for the variations of u that cannot be reproduced because of the mesh size. For example, $\bar{u}$ may be defined as the component of u in the finite element space and u' its component in the orthogonal complement (with respect to a certain inner product) in $\mathcal{W}$.

The strong assumption of what follows is that we assume that u' vanishes on the boundaries of the elements, that is, $u' = 0$ on $\partial\Omega^e$ for $e = 1, 2, \ldots, n_{\mathrm{el}}$. In this case, u' is the solution of the problem

$$\begin{aligned} &\mathcal{L}(u') = f - \mathcal{L}(\bar{u}) && \text{in } \Omega^e\,, \\ &u' = 0 && \text{on } \partial\Omega^e\,, \end{aligned} \tag{52}$$

which can be solved for u' in terms of the resolvable scale $\bar{u}$ and the Green's function g for the operator $\mathcal{L}$. This leads to

$$u'(y) = -\int_{\Omega'} g(\boldsymbol{x}, y)(\mathcal{L}(\bar{u}) - f)(\boldsymbol{x})\,\mathrm{d}\Omega_x =: M(\mathcal{L}(\bar{u}) - f)(y)\,, \tag{53}$$

where M is an integral operator and the integral is defined in Eq. (18).

Let us split also the test function v as $v = \bar{v} + v'$. The problem for the resolvable scale $\bar{u}$ is

$$a(\bar{u}, \bar{v}) + a(u', \bar{v}) = l(\bar{v}) . \tag{54}$$

Since u' is assumed to vanish on the boundaries of the elements, we have that

$$a(u', \bar{v}) = (\mathscr{L}^*(\bar{v}), u') , \tag{55}$$

where the integral in the L^2 product is again that defined in Eq. (18). Inserting the expression for u' in Eq. (53) into Eq. (55) and using this in Eq. (54) we find that

$$a(\bar{u}, \bar{v}) + (\mathscr{L}^*(\bar{v}), M(\mathscr{L}(\bar{u}) - f)) = l(\bar{v}) . \tag{56}$$

Observe that up to this point we have not considered any numerical approximation, that is, Eq. (56) is exact up to the assumption that $u' = 0$ on $\partial\Omega^e$.

Suppose now that $\bar{u}$ is approximated by u_h using finite elements and let v_h be the test function corresponding to $\bar{v}$. Since the Green's function for problem (52) is in general unknown, it must be also approximated. This amounts to approximate the integral operator M by $\tilde{M}_h$. The resulting approach can be cast in the general expression (17) provided that rather than considering τ and $\mathscr{R}(u_h)$ independently we take its product in the definition of the formulation. This is then defined by

$$\mathscr{P}_{\mathrm{SGS}}(v_h) = -\mathscr{L}^*(v_h) , \tag{57a}$$

$$\tau\mathscr{R}_{\mathrm{SGS}}(u_h) = -\tilde{M}_h(\mathscr{L}(u_h) - f) . \tag{57b}$$

This is the general expression of the subgrid scale models introduced in [3,4]. Particular forms of approximating the operator M, that is, of approximating the Green's function g, will lead to different subgrid scale models. Two of these forms are considered in the following. Let us also remark that the extension of what follows to the transient problem is straightforward. Having in mind the idea of treating the temporal derivative as the first spatial derivatives (that is, as the convective term) and using a discontinuous interpolation in time, the extension to transient problems is obtained simply by adding $\partial/\partial t$ to the convective operator in all what follows.

5.2. Algebraic approximation to M

Suppose that the Green's function $g(\boldsymbol{x}, \boldsymbol{y})$ is approximated by

$$g(\boldsymbol{x}, \boldsymbol{y}) \approx \tilde{g}(\boldsymbol{x}, \boldsymbol{y}) := \tau(\boldsymbol{y})\delta(\boldsymbol{y} - \boldsymbol{x}) , \tag{58}$$

where $\tau(\boldsymbol{y})$ is a function to be determined and δ is the Dirac delta distribution. Using $\tilde{g}$ in Eq. (53) leads to

$$u'(\boldsymbol{y}) \approx -\tau(\boldsymbol{y})(\mathscr{L}(\bar{u}) - f)(\boldsymbol{y}) , \tag{59}$$

or, equivalently,

$$\tilde{M}_h = -\tau(\boldsymbol{y}) . \tag{60}$$

In this case, the equation to be solved is

$$a(u_h, v_h) + (-\mathscr{L}^*(v_h), \tau(\mathscr{L}(u_h) - f)) = l(v_h) , \tag{61}$$

so that we recover the method proposed by Franca et al. described at the beginning of this section.

The problem now is how to compute the function τ. If we take it as piecewise constant and impose that the double integral of the LHS and the RHS of Eq. (58) coincide, it is found that

$$\tau = \frac{1}{\mathrm{meas}(\Omega^e)} \int_{\Omega^e} \int_{\Omega^e} g(\boldsymbol{x}, \boldsymbol{y}) \, \mathrm{d}\Omega_x \, \mathrm{d}\Omega_y . \tag{62}$$

It can be shown that for the model problem (29) this equation yields also the value of τ given by eqs. (30) and (31) [4].

The extension to systems and transient problems using the discontinuous Galerkin method for the time discretization leads to

$$\boxed{\begin{aligned}\mathscr{P}_{\mathrm{SGS}}(\boldsymbol{V}_h) &= \frac{\partial \boldsymbol{V}_h}{\partial t} - \mathscr{L}^*(\boldsymbol{V}_h)\\ &= \frac{\partial \boldsymbol{V}_h}{\partial t} + \boldsymbol{A}_i^{\mathrm{t}} \frac{\partial \boldsymbol{V}_h}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{V}_h}{\partial x_j}\right) - \boldsymbol{S}^{\mathrm{t}}\boldsymbol{V}_h\\ \mathscr{R}_{\mathrm{SGS}}(\boldsymbol{U}_h) &= \frac{\partial \boldsymbol{U}_h}{\partial t} + \mathscr{L}(\boldsymbol{U}_h) - \boldsymbol{F}\\ &= \frac{\partial \boldsymbol{U}_h}{\partial t} + \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U}_h) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U}_h - \boldsymbol{F}\end{aligned}} \tag{63}$$

Matrix $\boldsymbol{\tau}$ can be defined again as in Eq. (62), now with g being a matrix.

This is the version of the SGS method that we consider in our comparisons, although it is only a particular SGS model. Another possibility is described below.

REMARK. If in the original Eq. (4) the conservative form of the convective term given by Eq. (5) is replaced by the non-conservative one in Eq. (9), the perturbation $\mathscr{P}$ for the GLS and the SGS methods become

$$\mathscr{P}_{\mathrm{GLS}}(\boldsymbol{V}_h) = \frac{\partial \boldsymbol{V}_h}{\partial t} + \boldsymbol{A}_i \frac{\partial \boldsymbol{V}_h}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{V}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{V}_h \tag{64a}$$

$$\mathscr{P}_{\mathrm{SGS}}(\boldsymbol{V}_h) = \frac{\partial \boldsymbol{V}_h}{\partial t} + \frac{\partial}{\partial x_i}(\boldsymbol{A}_i^{\mathrm{t}}\boldsymbol{V}_h) + \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{V}_h}{\partial x_j}\right) - \boldsymbol{S}^{\mathrm{t}}\boldsymbol{V}_h\,. \tag{64b}$$

From this we see that the GLS method satisfies the conservation property (24) when the non-conservation form of the equation is used, whereas the SGS is 'consistent', in the sense that condition (24) is fulfilled when the equations are written in conservation form and it is not when the non-conservation form is used.

5.3. Approximation to M through bubble functions

In this case, instead of constructing $\tilde{M}_h$ through an approximation to the Green's function g, we consider directly the finite element approximation to the unresolvable scales u'. The continuous problem of which this function is solution is

$$a(\bar{u}, v') + a(u', v') = l(v')\,, \tag{65}$$

which is nothing but the variational formulation of problem (52).

The function u' can now be approximated by using bubble functions as

$$u'(\boldsymbol{x}) \approx u_h'(\boldsymbol{x}) = \sum_{j=1}^{n_{\mathrm{bub}}} \psi_j(\boldsymbol{x}) u_{h,j}'\,, \tag{66}$$

where n_{bub} is the number of bubble functions ψ and $u_{h,j}'$ are the nodal values of u_h'. Observe that this function satisfies the original assumption of being zero on the element boundaries.

The discrete counterpart of problem (65) is the following system of algebraic equations:

$$\begin{aligned}\sum_{j=1}^{n_{\mathrm{bub}}} a(\psi_j, \psi_k) u_{h,j}' &= l(\psi_k) - a(\bar{u}, \psi_k)\\ &= \int_{\Omega'} \psi_k(\boldsymbol{x})(f - \mathscr{L}(\bar{u}))(\boldsymbol{x})\,\mathrm{d}\Omega\,,\end{aligned} \tag{67}$$

for $k = 1, \ldots, n_{\mathrm{bub}}$. In the last equality we have made use of the fact that the bubble functions vanish on the element boundaries. If a_{jk}^{-1} is the jk component of the inverse of the matrix the jk component of which is $a(\psi_j, \psi_k)$, from (67) we have that

$$u_h'(y) = -\sum_{j,k=1}^{n_{\text{bub}}} \int_{\Omega'} \psi_j(y) a_{jk}^{-1} \psi_k(x)(\mathscr{L}(\bar{u}) - f)(x)\, \mathrm{d}\Omega_x\,, \tag{68}$$

from where it follows that the use of bubble functions has an inherent approximation to the Green's function given by

$$g(x, y) \approx \sum_{j,k=1}^{n_{\text{bub}}} \psi_j(y) a_{jk}^{-1} \psi_k(x)\,. \tag{69}$$

Once this approximation has been established, we proceed as in the previous case: the expression for $u_h'(y)$ in terms of the resolvable scales u_h is inserted in the discrete version Eq. (54), yielding an equation for u_h alone.

6. The Characteristic Galerkin method

6.1. Basic concepts

One of the first methods that were designed for solving the transient convection diffusion equation was the method of characteristics, which in combination with the finite element method for the spatial discretization was called Characteristic Galerkin (CG) method [6,7]. The basic idea of this method is described next.

While the leading ideas of the GLS and the SGS methods can be applied to either scalar or vector equations, the CG method must be developed for scalar problems. This is the case that we consider first.

Let us denote by $X(x_{\text{ref}}, t_{\text{ref}}; t)$ the trajectory of the particle that at time $t = t_{\text{ref}}$ is located at the spatial point x_{ref}, so that $X(x_{\text{ref}}, t_{\text{ref}}; t_{\text{ref}}) = x_{\text{ref}}$. This trajectory, or characteristic, will be the solution of the problem

$$\frac{\mathrm{d}}{\mathrm{d}t} X(t) = a(X(t), t)\,, \tag{70a}$$

$$X(t_{\text{ref}}) = x_{\text{ref}}\,. \tag{70b}$$

For what follows it is interesting to consider the case in which the advective velocity a depends explicitly on time, as indicated in Eq. (70a). In the short-hand notation $X(t)$ it is understood that X depends also on t_{ref} and x_{ref} through the initial condition (70b). We have that

$$\frac{\mathrm{d}}{\mathrm{d}t} u(X(t), t)\Big|_{t=t_{\text{ref}}} = \left(\frac{\partial u}{\partial t} + a \cdot \nabla u\right)\Bigg|_{x=x_{\text{ref}},\, t=t_{\text{ref}}}\,. \tag{71}$$

If we write the convective term in Eq. (1) as $a \cdot \nabla u + (\nabla \cdot a)u$ and use the redefinition (10) (now for the scalar case), Eq. (1) may be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t} u(X(t), t) + \mathscr{L}_{\text{ds}}(u)(X(t), t) = f(X(t))\,, \tag{72}$$

where we have stressed the fact that all the terms are evaluated at $x = X(t)$. The idea now is to discretize the derivative $\mathrm{d}/\mathrm{d}t$ using a finite difference scheme, that is, to discretize the total derivative in Eq. (1) along the characteristics.

Suppose now that we have the solution at time t^n and we want to compute it at time t^{n+1} using the generalized trapezoidal rule, as in Section 3 for the SUPG method. Let t_{ref} be a reference time in $[t^n, t^{n+1}]$. The time discretization of Eq. (72) that we consider is

$$\begin{aligned}&\frac{1}{\Delta t}[u(X(t^{n+1}), t^{n+1}) - u(X(t^n), t^n)] + \theta \mathscr{L}_{\text{ds}}(u)(X(t^{n+1}), t^{n+1})\\ &\quad + (1-\theta)\mathscr{L}_{\text{ds}}(u)(X(t^n), t^n) = \theta f(X(t^{n+1})) + (1-\theta) f(X(t^n))\,,\end{aligned} \tag{73}$$

where $\theta \in [0, 1]$. Once arrived at this equation there are two possibilities, yielding two different versions of the CG method.

6.2. Interpolation of the unknown along the characteristics

This was the method proposed in [6,7]. Suppose that $\theta = 1$ and that $t_{\text{ref}} = t^{n+1}$, and write simply $\boldsymbol{x}$ for $\boldsymbol{x}_{\text{ref}}$. Eq. (73) in this case reduces to

$$\frac{1}{\Delta t}[u(\boldsymbol{x}, t^{n+1}) - u(\boldsymbol{X}(t^n), t^n)] + \mathscr{L}_{\text{ds}}(u)(\boldsymbol{x}, t^{n+1}) = f(\boldsymbol{x}) . \tag{74}$$

We may think of $\boldsymbol{x}_{\text{ref}}$ as the configuration at time t^{n+1}. Therefore, the problem is how to evaluate the term $u(\boldsymbol{X}(t^n), t^n)$. For this, it is necessary first to integrate the equation of the characteristics in order to express $\boldsymbol{X}(t^n)$ in terms of the current configuration. This may be done by using either Eqs. (75) or (77) below, depending on the order of accuracy desired. In general, the result will not coincide with any node of the finite element mesh, that is, $\boldsymbol{X}(t^n)$ will lie within an element. This element must be identified and after this the unknown $u(\boldsymbol{X}(t^n), t^n)$ must be interpolated.

6.3. Local expansion of the unknown along the characteristics

We derive now an explicit expression for $u(\boldsymbol{X}(t^{n+1}), t^{n+1})$ and $u(\boldsymbol{X}(t^n), t^n)$ using a Taylor expansion in the neighborhood of $\boldsymbol{x}_{\text{ref}}$. This will allow us to obtain a semi-discrete system of equations where all the terms will be evaluated at the same point of the same spatial domain, thus avoiding the need of finding $\boldsymbol{X}(t^n)$ as described above. This idea can be found in [8,9].

In order to be able to obtain a second order approximation along the characteristics, we must take $\theta = 1/2$ in Eq. (73). This is the value of θ adopted in what follows.

The expansion can be done in particular for $t_{\text{ref}} = t^n + \Delta t/2 = t^{n+1/2}$ and $t_{\text{ref}} = t^n + \Delta t = t^{n+1}$. The first option yields the classical Crank–Nicolson discretization of Eq. (1), whereas the second introduces some additional terms that enhance the stability of the numerical scheme. From the geometrical standpoint, if $t_{\text{ref}} = t^{n+1/2}$ Eq. (73) (with $\theta = 1/2$) may be viewed as centered discretization along the characteristics. On the other hand, for $t_{\text{ref}} = t^{n+1}$ we move backwards. This is relative to the particle we follow—in both cases, though, the discretization is formally of second order.

Let us consider the case $t_{\text{ref}} = t^{n+1}$ and so $\boldsymbol{X}(t^{n+1}) = \boldsymbol{x}_{\text{ref}}$ (see Eqs. (70)). To emphasize that $\boldsymbol{x}_{\text{ref}}$ is arbitrary, we shall write $\boldsymbol{x}$ instead of $\boldsymbol{x}_{\text{ref}}$.

The solution of problem (70a) may be approximated up to second order as follows:

$$\begin{aligned}\boldsymbol{X}(t^n) &= \boldsymbol{X}(t^{n+1}) - \Delta t\, \boldsymbol{a}(\boldsymbol{X}(t^{n+1}), t^n) + \mathrm{O}(\Delta t^2)\\ &= \boldsymbol{x} - \Delta t\, \boldsymbol{a}^n + \mathrm{O}(\Delta t^2) ,\end{aligned} \tag{75}$$

and therefore,

$$\begin{aligned}\boldsymbol{a}(\boldsymbol{X}(t^n), t^n) &= \boldsymbol{a}(\boldsymbol{x} - \Delta t\, \boldsymbol{a}^n + \mathrm{O}(\Delta t^2), t^n)\\ &= \boldsymbol{a}^n - \Delta t\, \boldsymbol{a}^n \cdot \nabla \boldsymbol{a}^n + \mathrm{O}(\Delta t^2) .\end{aligned} \tag{76}$$

Eq. (76) allows to obtain the following third-order approximation to the trajectory $\boldsymbol{X}$:

$$\begin{aligned}\boldsymbol{X}(t^n) &= \boldsymbol{X}(t^{n+1}) - \frac{\Delta t}{2}[\boldsymbol{a}(\boldsymbol{X}(t^{n+1}), t^{n+1}) + \boldsymbol{a}(\boldsymbol{X}(t^n), t^n)] + \mathrm{O}(\Delta t^3)\\ &= \boldsymbol{x} - \Delta t\, \boldsymbol{a}^{n+1/2} + \frac{\Delta t^2}{2}\boldsymbol{a}^n \cdot \nabla \boldsymbol{a}^n + \mathrm{O}(\Delta t^3) ,\end{aligned} \tag{77}$$

where $\boldsymbol{a}^{n+1/2} = [\boldsymbol{a}^{n+1} + \boldsymbol{a}^n]/2$. Using the approximation (77) we obtain, for any smooth function v,

$$\begin{aligned}v(\boldsymbol{X}(t^n), t^n) &= v\left(\boldsymbol{x} - \Delta t\, \boldsymbol{a}^{n+1/2} + \frac{\Delta t^2}{2}\boldsymbol{a}^n \cdot \nabla \boldsymbol{a}^n + \mathrm{O}(\Delta t^3), t^n\right)\\ &= v^n - \Delta t\, \boldsymbol{a}^{n+1/2} \cdot \nabla v^n + \frac{\Delta t^2}{2}(\boldsymbol{a}^n \cdot \nabla \boldsymbol{a}^n) \cdot \nabla v^n\\ &\quad + \frac{\Delta t^2}{2}\boldsymbol{a}^{n+1/2} \otimes \boldsymbol{a}^{n+1/2} : \nabla(\nabla v^n) + \mathrm{O}(\Delta t^3) .\end{aligned} \tag{78}$$

Using the fact that $\boldsymbol{a}^{n+1/2} = \boldsymbol{a}^n + \mathrm{O}(\Delta t)$ we may write Eq. (78) as

$$v(X(t^n), t^n) = v^n - \Delta t\, \boldsymbol{a}^{n+1/2} \cdot \nabla v^n + \frac{\Delta t^2}{2} \boldsymbol{a}^n \cdot \nabla(\boldsymbol{a}^n \cdot \nabla v^n) + \mathrm{O}(\Delta t^3)\,. \tag{79}$$

This holds for any function v. A simplified version of this approximation is

$$v(X(t^n), t^n) = v^n - \Delta t\, \boldsymbol{a}^n \cdot \nabla v^n + \mathrm{O}(\Delta t^2)\,. \tag{80}$$

Using Eq. (79) in the discretization of the temporal derivative in Eq. (73) (with $\theta = 1/2$) and Eq. (80) to approximate the rest of the terms evaluated at $\boldsymbol{x} = \boldsymbol{X}(t^n)$ and $t = t^n$ we finally obtain

$$\frac{1}{\Delta t}[u^{n+1} - u^n] + \boldsymbol{a}^{n+1/2} \cdot \nabla u^n + \mathscr{L}_{\mathrm{ds}}(u^{n+1/2}) - f - \frac{\Delta t}{2} \boldsymbol{a}^n \cdot \nabla[\mathscr{L}(u^n) - f] = 0\,. \tag{81}$$

Once this semidiscrete problem has been obtained, we may further approximate values at the time level $n + 1/2$ by values at n, thus obtaining a fully explicit scheme. This involves only an approximation of the temporal argument of the functions.

If last term in Eq. (81) is multiplied by a test function v and the result is integrated by parts it is found that

$$-\frac{\Delta t}{2} \int_\Omega v \boldsymbol{a}^n \cdot \nabla[\mathscr{L}(u^n) - f^n]\, \mathrm{d}\Omega = \frac{\Delta t}{2} \int_\Omega \nabla \cdot (\boldsymbol{a}^n v)[\mathscr{L}(u^n) - f^n]\, \mathrm{d}\Omega\,, \tag{82}$$

where we have made use of the fact that $v = 0$ on $\partial\Omega$. For other boundary conditions constant in time it can be also assumed that $\mathscr{L}(u^n) - f^n = 0$ on $\partial\Omega$.

If now the weak form of Eq. (81) is discretized, it is seen that the contribution due to the use of the Characteristic Galerkin method with respect to the standard Galerkin approach has the general form (17), with

$$\mathscr{P}_{\mathrm{CG}}(v_h) = -\mathscr{L}^*_{\mathrm{conv,nc}}(v_h) = \nabla \cdot (\boldsymbol{a} v_h)\,, \tag{83a}$$

$$\mathscr{R}_{\mathrm{CG}}(u_h) = \mathscr{L}(u_h) - f\,, \tag{83b}$$

where all the terms are evaluated at time step n. Observe that the integral of the RHS of eq. (82) has to be understood as the sum of the integrals over the element interiors for the discrete problem, that is, in the sense of Eq. (18).

According to the previous derivation, the numerical parameter τ in this case is $\Delta t/2$, the same for all the elements. However, it is shown in [33] that if instead of taking $t_{\mathrm{ref}} = t^{n+1}$ we take $t_{\mathrm{ref}} = \gamma t^{n+1} + (1 - \gamma)t^n$, then $\tau = \gamma\, \Delta t/2$. The parameter γ is free: it represents the position on the characteristic at which the total time derivative is discretized. This justifies the use of variables τ's.

From the previous derivation of the CG method it is readily seen that there are other schemes with the same accuracy. Our motivation has been to express as many terms as possible evaluated at time step n, although some terms could be equally evaluated at time step $n + 1$, leading to implicit versions of the CG method.

Once the final equations discretized in time have been obtained, it is possible to change the time step at which some terms are evaluated. This will modify the accuracy of the scheme (and perhaps also its stability), but only in the time variable, not along the characteristics.

REMARKS

(1) It is interesting to note that if the fully explicit version of the CG method is considered and linear finite elements are used, the critical time step above which the scheme becomes unstable turns out to give a value of $\Delta t/2$ very close to the intrinsic time of the SUPG method given in Eqs. (30) and (31) (see [34]). Also, if $\boldsymbol{a}$ is divergence free it is seen from Eqs. (28a) and (83a) that $\mathscr{P}_{\mathrm{CG}}(v_h) = \mathscr{P}_{\mathrm{SUPG}}(v_h)$.

(2) Observe that the contributions introduced by the CG to the Galerkin terms of the discrete system do not affect the mass matrix.

(3) The CG does *not* satisfy the global conservation property (34), since the expression of $\mathscr{P}$ for this method is that given by Eq. (83a), no matter whether the convective term is written in conservative or non-conservative form.

(4) The particular version of the CG method presented here, although the most popular, is not the only one possible. Different versions can be derived by taking as point of departure different discretizations in time.

6.4. Extension to systems

The extension of the CG method to systems of equations is not direct. Since it is based on the fact that the partial time derivative plus the convective one are written as a total time derivative, we must identify an advection speed $\boldsymbol{a}_k$ for the kth scalar equation, $k = 1, \dots, n_{\text{unk}}$. If these are different for each equation, we have to consider different characteristics for each equation of the system, that is, different curves $\boldsymbol{X}_k(t)$ solution of problems like (70) with $\boldsymbol{a}_k$ as advection speed.

Suppose for a moment that instead of using expression (5) for the convective term it is replaced by (9) and the redefinition (10) of matrix $\boldsymbol{S}$. If we introduce the notation

$$\begin{aligned}
&\mathscr{L}^{\circ}_{\text{conv,nc}}(\boldsymbol{U}) := \mathscr{L}_{\text{conv,nc}}(\boldsymbol{U}) - \text{diag}\,\mathscr{L}_{\text{conv,nc}}(\boldsymbol{U}) \\
&\frac{\mathrm{d}}{\mathrm{d}t_k} := \frac{\partial}{\partial t} + \boldsymbol{a}_k \cdot \nabla\,, \\
&\frac{\mathrm{D}}{\mathrm{D}t} := \text{diag}\left(\frac{\mathrm{d}}{\mathrm{d}t_1}, \dots, \frac{\mathrm{d}}{\mathrm{d}t_{n_{\text{unk}}}}\right),
\end{aligned} \tag{84}$$

with diag $\mathscr{L}_{\text{conv,nc}}$ defined in Eq. (34), then Eq. (4) may be written as

$$\frac{\mathrm{D}\boldsymbol{U}}{\mathrm{D}t} + \mathscr{L}^{\circ}_{\text{conv,nc}}(\boldsymbol{U}) + \mathscr{L}_{\text{ds}}(\boldsymbol{U}) = \boldsymbol{F}\,, \tag{85}$$

where the kth equation of this system is evaluated along the curve $\boldsymbol{X}_k(t)$.

The same procedure as for the scalar case can be repeated now. After using a finite difference approximation to $\mathrm{D}\boldsymbol{U}/\mathrm{D}t$ we can either interpolate the unknowns along the different characteristics or use a local expansion to avoid this interpolation.

If the second approach is used, the method we end up with is defined by

$$\boxed{\begin{aligned}
&\mathscr{P}_{\text{CG}}(\boldsymbol{V}_h) = -(\text{diag}\,\mathscr{L}_{\text{conv,nc}})^*(\boldsymbol{V}_h) = \frac{\partial}{\partial x_i}(\text{diag}(\boldsymbol{A}_i)\boldsymbol{V}_h) \\
&\mathscr{R}_{\text{CG}}(\boldsymbol{U}_h) = \mathscr{L}(\boldsymbol{U}_h) - \boldsymbol{F} \\
&\qquad = \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U}_h) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U}_h - \boldsymbol{F}
\end{aligned}} \tag{86}$$

This method is similar to the version of the SUPG given by Eq. (34) and has the same drawback: it needs a particular set of variables $\boldsymbol{U}$ for its definition, for if we make a linear change of variables $\hat{\boldsymbol{U}} = \boldsymbol{T}\boldsymbol{U}$, with $\boldsymbol{T}$ a matrix of (for example) constant coefficients, the matrices $\boldsymbol{A}_i$ in Eq. (4) must be replaced by $\hat{\boldsymbol{A}}_i = \boldsymbol{T}\boldsymbol{A}_i\boldsymbol{T}^{-1}$, and $\text{diag}(\hat{\boldsymbol{A}}_i) \neq \boldsymbol{T}\,\text{diag}(\boldsymbol{A}_i)\boldsymbol{T}^{-1}$ (except, of course, if $\boldsymbol{A}_i = a_i\boldsymbol{I}$, where $\boldsymbol{I}$ is the $n_{\text{unk}} \times n_{\text{unk}}$ identity matrix).

Concerning matrix $\boldsymbol{\tau}$, in principle it is simply $\tau\boldsymbol{I}$, with $\tau = \Delta t/2$. However, as it was mentioned for the scalar case, the use of variable τ's is justified. Moreover, one can also think of using different τ's for the different equations, thus leading to an expression of $\boldsymbol{\tau}$ similar to that given by Eq. (35). This approach is very common in practice and often justified by the use of local time stepping techniques when the transient evolution is not important [34].

7. The Taylor–Galerkin method

The Taylor–Galerkin method was first introduced by Donea in [10] as the finite element counterpart of the Lax–Wendroff scheme for finite difference methods. Here, we derive a general version of the explicit form of this formulation for Eq. (8). There are also implicit versions of this method, although they have to be motivated using other reasoning.

Let us consider the following Taylor expansion of the unknown $\boldsymbol{U}$ at time step n:

$$U^{n+1} = U^n + \frac{\partial U^n}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 U^{n+\theta}}{\partial t^2}\Delta t^2 + \mathrm{O}(\Delta t^3), \tag{87}$$

where $0 \leqslant \theta \leqslant 1$. For the moment, let us take $\theta = 0$ (see Remark 3 below). If U satisfies Eq. (8) then

$$U^{n+1} = U^n + [F^n - \mathcal{L}(U^n)]\,\Delta t + \frac{1}{2}\left[\frac{\partial F^n}{\partial t} - \frac{\partial}{\partial t}(\mathcal{L}(U))^n\right]\Delta t^2 + \mathrm{O}(\Delta t^3). \tag{88}$$

As before, we assume that F is time independent. Otherwise, the term $\partial F/\partial t$ should be kept in what follows. If the solution U of Eq. (8) is sufficiently smooth and the coefficient matrices A_i, K_{ij} and S are time independent, we have that

$$\frac{\partial}{\partial t}(\mathcal{L}(U)) = \mathcal{L}\left(\frac{\partial U}{\partial t}\right) = \mathcal{L}(F - \mathcal{L}(U)). \tag{89}$$

Using this in Eq. (88) and neglecting the term $\mathrm{O}(\Delta t^3)$ we find the following time discretization of Eq. (8):

$$\frac{U^{n+1} - U^n}{\Delta t} = F - \mathcal{L}(U^n) - \frac{\Delta t}{2}\mathcal{L}(F - \mathcal{L}(U))^n. \tag{90}$$

If this equation is now multiplied by a test function V and integrated over Ω the last term in the RHS of Eq. (90) leads to

$$\frac{\Delta t}{2}\int_\Omega V^{\mathrm{t}}\mathcal{L}(F - \mathcal{L}(U))^n\,\mathrm{d}\Omega = \frac{\Delta t}{2}\int_\Omega \mathcal{L}^*(V)^{\mathrm{t}}(F - \mathcal{L}(U))^n\,\mathrm{d}\Omega. \tag{91}$$

From this we see that when the weak form of Eq. (90) is discretized the contribution due to the use of the TG method with respect to the standard Galerkin approach has again the general form (17), now with

$$\boxed{\begin{aligned}
\mathcal{P}_{\mathrm{TG}}(V_h) &= -\mathcal{L}^*(V_h)\\
&= A_i^{\mathrm{t}}\frac{\partial V_h}{\partial x_i} + \frac{\partial}{\partial x_i}\left(K_{ij}\frac{\partial V_h}{\partial x_j}\right) - S^{\mathrm{t}}V_h\\
\mathcal{R}_{\mathrm{TG}}(U_h) &= \mathcal{L}(U_h) - F\\
&= \frac{\partial}{\partial x_i}(A_iU_h) - \frac{\partial}{\partial x_i}\left(K_{ij}\frac{\partial U_h}{\partial x_j}\right) + SU_h - F
\end{aligned}} \tag{92}$$

where in principle all the terms are evaluated at time step n. As for the CG method, the parameter τ in this case is $\Delta t/2$, the same for all the elements, and the integral of the RHS of Eq. (91) has to be understood as the sum of the integrals over the element interiors for the discrete problem.

Although this is the essential of the TG method, it is not exactly what was proposed in [10], where the second derivatives of the test function appearing in Eq. (92) were neglected and the presence of an absorption term in the original equation was not considered. Our derivation is therefore more general and highlights the relationship between the TG method and the methods presented in the previous sections.

REMARKS

(1) Observe that to derive the expressions in Eq. (92) we have used that the coefficient matrices and the force vector are not time dependent. This assumption was not essential for any of the previous methods.

(2) The present development of the TG method (and also of the Lax–Wendroff method when finite differences are used) is only valid for deriving explicit schemes. To obtain a fully implicit one we should start from a Taylor expansion to express U^n in terms of values at time step $n+1$. This would lead to

$$U^n = U^{n+1} - \frac{\partial U^{n+1}}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 U^{n+\theta}}{\partial t^2}\Delta t^2 + \mathrm{O}(\Delta t^3). \tag{93}$$

If now the same procedure as for the explicit case is followed, the 'stabilizing' term that appears has the wrong sign. Therefore, the stability of the original Galerkin scheme will be in fact worsened.

(3) Although fully implicit schemes cannot be obtained with the previous reasoning, they can be (and they have been) used in practice. The idea is simply to evaluate all the terms appearing in the RHS of Eq. (90) at the time step $n+1$. However, it must be mentioned that what is usually known as 'implicit Taylor–Galerkin' scheme is obtained by taking in Eq. (87) $\theta = 1$. This leads to an explicit scheme where the stabilizing term is treated implicitly.

(4) As in the case of the CG method, other versions of the TG could be obtained by starting from different explicit time discretizations.

8. On the discrete maximum principle for scalar equations

8.1. Background

In this section we shall consider the steady version of the scalar equation (1). To simplify further the discussion we take all the coefficients of the equation constant. Thus, the problem we consider is to find u such that

$$-k\nabla^2 u + \boldsymbol{a}\cdot\nabla u + su = f \quad \text{in } \Omega\,, \tag{94a}$$

$$u = 0 \quad \text{on } \partial\Omega\,. \tag{94b}$$

For the continuous problem (94) it is well known that the maximum principle holds, that is, the solution attains its maximum at the boundary when f is non-positive. The boundary condition (94b) can be generalized to $u = u_{\mathrm{d}}$, with the given function u_{d} non-negative. The question is whether this property is inherited by the discrete problem or not.

For problem (94) the SUPG and the CG methods in one hand and the SGS and the TG methods in the other coincide, except in the definition of the algorithmic parameter τ. This is why we shall only compare the SUPG, the GLS and the SGS methods here. The study of the discrete maximum principle (DMP) will show an important difference in the behavior of these three methods.

Let n_{tp} be the total number of nodes of the finite element mesh and n_{fp} the number of interior nodes. The finite element discretization of the problem will lead to an algebraic system of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\,, \tag{95}$$

where $\boldsymbol{x}$ stands for the vector containing the nodal unknowns x_i, $i = 1, \dots, n_{\mathrm{tp}}$. The values x_i, $i = n_{\mathrm{fp}} + 1, \dots, n_{\mathrm{tp}}$ are known from the Dirichlet boundary conditions. Matrix $\boldsymbol{A}$, whose components will be denoted a_{ij}, will have dimensions $n_{\mathrm{fp}} \times n_{\mathrm{tp}}$ and the vector $\boldsymbol{b}$ coming from the source term will have components b_i, $i = 1, \dots, n_{\mathrm{fp}}$.

As shown in [35] for linear elements, the satisfaction of the DMP, viz.

$$\max_{i=1,\dots,n_{\mathrm{tp}}} \{x_i\} = x_m\,, \quad \text{with} \quad n_{\mathrm{fp}} + 1 \leqslant m \leqslant n_{\mathrm{tp}}\,, \tag{96}$$

leads to uniform convergence of the finite element solution. Therefore, no spurious oscillations will appear, not even in the vicinity of sharp layers. On the other hand, the DMP follows (see e.g. [36]) if $b_i \leqslant 0$, $i = 1, \dots, n_{\mathrm{fp}}$, and matrix A in (95) is of non-negative type, that is,

$$a_{ij} \leqslant 0 \quad \text{for } i \neq j,\ i = 1, \dots, n_{\mathrm{fp}},\ j = 1, \dots, n_{\mathrm{tp}}\,, \tag{97a}$$

$$\sum_{j=1}^{n_{\mathrm{tp}}} a_{ij} \geqslant 0\,, \quad i = 1, \dots, n_{\mathrm{fp}}\,. \tag{97b}$$

Since the assembly operator is nothing but the adequate sum of the element contributions, it suffices to check conditions (97a) and (97b) for the element matrices, hereafter denoted by $A^{(e)}$. Let us split them into their diffusive, convective and reactive contributions as

$$\boldsymbol{A}^{(e)} = \boldsymbol{A}_{\mathrm{d}}^{(e)} + \boldsymbol{A}_{\mathrm{c}}^{(e)} + \mathbf{A}_{\mathrm{r}}^{(e)} . \tag{98}$$

Using the Galerkin method, these matrices have components

$$\begin{aligned} a_{\mathrm{d}ij}^{(e)} &= k \int_{\Omega^e} \nabla N_i \cdot \nabla N_j \, \mathrm{d}\Omega , \\ a_{\mathrm{c}ij}^{(e)} &= \int_{\Omega^e} N_i \boldsymbol{a} \cdot \nabla N_j \, \mathrm{d}\Omega , \\ a_{\mathrm{r}ij}^{(e)} &= s \int_{\Omega^e} N_i N_j \, \mathrm{d}\Omega , \end{aligned} \tag{99}$$

and the components of the element contributions to the vector $\boldsymbol{b}$ are

$$b_i^{(e)} = \int_{\Omega^e} N_i f \, \mathrm{d}\Omega , \tag{100}$$

where N_i is the interpolation function associated to the ith node of the finite element mesh. In what follows, we shall restrict ourselves to the case of linear finite elements using the standard shape functions. In this case, condition (97b) is trivial to check:

$$\sum_{j=1}^{n_{\mathrm{tp}}} a_{ij} = s \int_{\Omega} N_i \, \mathrm{d}\Omega \geqslant 0 . \tag{101}$$

Therefore, only condition (97a) and that $b_i^{(e)} \leqslant 0$ when $f \leqslant 0$ have to be verified.

8.2. Conditions on τ

In general diffusion–convection problems, with $s = 0$, it is impossible in general to satisfy the discrete maximum principle, as it was shown in [36]. Observe that in this case the SUPG, the GLS and the SGS methods are all the same when linear elements are employed. The difference arises when $s > 0$.

Let us consider first the one-dimensional problem:

$$-k \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + a \frac{\mathrm{d}u}{\mathrm{d}x} + su = f , \quad 0 < x < 1 , \tag{102a}$$

$$u(0) = u(1) = 0 . \tag{102b}$$

Without loss of generality, we shall take $a \geqslant 0$. Using a uniform partition of the interval [0, 1] of diameter h, the matrices in Eq. (98) are now given by

$$\boldsymbol{A}_{\mathrm{d}}^{(e)} = \frac{k}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} , \quad \boldsymbol{A}_{\mathrm{c}}^{(e)} = \frac{a}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} , \quad \boldsymbol{A}_{\mathrm{r}}^{(e)} = \frac{sh}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} . \tag{103}$$

Condition (97a) implies in this case that

$$-\frac{k}{h} \pm \frac{a}{2} + \frac{sh}{6} \leqslant 0 . \tag{104}$$

Observe that if a finite difference method is used, matrix $\boldsymbol{A}_{\mathrm{r}}^{(e)}$ is diagonal, and so the reaction term would not appear in Eq. (104). Thus, for this case the presence of reaction does not influence the satisfaction of the DMP.

It is easy to see that the application of the SUPG, the GLS or the SGS methods to problem (102) is equivalent to the use of the Galerkin method with modified values of the parameters k, a and s. These values are

$$\bar{k} = k + \tau a^2 , \tag{105a}$$

$$\bar{a} = a - (\xi + 1)\tau a s , \tag{105b}$$

$$\bar{s} = s - \xi \tau s^2 , \tag{105c}$$

where $\xi = 0$ for the SUPG method, $\xi = -1$ for the GLS method and $\xi = 1$ for the SGS method.

Using these effective values for the coefficients of Eq. (102a), condition (104) reads

$$-\frac{k}{h} \pm \frac{a}{2} + \frac{sh}{6} - \tau\left[\frac{a^2}{h} \mp \frac{\xi + 1}{2} as + \xi \frac{s^2 h}{6}\right] \leqslant 0 . \tag{106}$$

This condition is impossible to fulfill for all values of k, a and s, although it provides information about the behavior of the different methods.

First, let us remark that when $s = 0$ condition (106) reduces to

$$\tau \geqslant \frac{h}{2a}\left(1 - \frac{1}{\mathrm{Pe}}\right), \tag{107}$$

which is the condition that prevents node to node oscillations.

In the limit case $a = 0$ the situation is different. The SUPG method ($\xi = 0$) does not introduce any modification to the Galerkin method, and therefore it is impossible in general to satisfy condition (106) for $a = 0$. For the GLS method ($\xi = -1$) it is easy to see that (106) implies $\tau < 0$, which is incompatible with the case $a > 0$ that leads to condition (107).

Only the SGS method ($\xi = 1$) behaves well in the case $a = 0$. If we define the dimensionless number

$$\mathrm{Ab} := \frac{sh^2}{2k}, \tag{108}$$

which is a measure of the relative importance of the absorption and diffusion terms, condition (106) yields

$$\tau \geqslant \frac{1}{s}\left(1 - \frac{3}{\mathrm{Ab}}\right). \tag{109}$$

Although the SGS method allows to satisfy condition (106) when $a = 0$, in the general case this is impossible. To see this, observe that the bracketed term in this inequality can be zero for values that lead to a positive left-hand side. Nevertheless, the limit cases analyzed above provide useful design criteria for τ. First, it is easy to see that for linear elements and in the case $s = 0$ the parameter τ verifies condition (107) if it computed as indicated in Eq. (41) with $C_1 = 1/3$ and $C_2 = 1$, but also if we take

$$\tau = \frac{h}{2a}\frac{\mathrm{Pe}}{\mathrm{Pe} + 1} = \frac{1}{\frac{4k}{h^2} + \frac{2a}{h}}, \quad \text{for } s = 0 . \tag{110}$$

Similarly, in the limit case $a = 0$ condition (109) is verified if we take

$$\tau = \frac{1}{s}\frac{\mathrm{Ab}}{\mathrm{Ab} + 2} = \frac{1}{\frac{4k}{h^2} + s}, \quad \text{for } a = 0 . \tag{111}$$

In order to ensure that the DMP holds not only matrix $\boldsymbol{A}$ in Eq. (95) must be of non-negative type, but also the components of $\boldsymbol{b}$ must be non-positive. In the 1D case using linear elements considered above and assuming f to be piecewise constant, the condition $b_i \leqslant 0$ leads to

$$\int_{\Omega^e} \left[N_i + \tau\left(a\frac{\mathrm{d}N_i}{\mathrm{d}x} - sN_i\right)\right] \mathrm{d}x \geqslant 0 , \tag{112}$$

which yields

$$\tau \leqslant \frac{1}{s + \frac{2a}{h}} . \tag{113}$$

From Eqs. (110), (111) and (113) we see that a general expression for τ that satisfies all the requirements established so far is

$$\tau = \frac{1}{\dfrac{4k}{h^2} + \dfrac{2a}{h} + s} . \tag{114}$$

This is the expression that we shall use in the following numerical tests. It has been derived in a very simple case, but provides the way in which τ must behave when the coefficients of Eq. (94a) change. Also, this expression is easy to generalize to systems of equations, although we shall not pursue further this point in this paper.

In general multidimensional situations, none of the schemes considered here yields a non-negative matrix in the algebraic system (95), not even in the case $s = 0$, as it was shown in [36]. However, when $\boldsymbol{a} = \boldsymbol{0}$ a simple analysis reveals that the SGS method, with τ given by Eq. (114), has the same properties as the standard Galerkin method with $s = 0$. In particular, condition (97a) can be satisfied if the finite element mesh is regular enough.

8.3. Numerical tests

In this section we present a very simple test case to see the different behavior of the SUPG, the GLS and the SGS methods when $s > 0$. The data for problem (94) we have taken are the following. The domain Ω where the problem is to be solved is the unit square, $\bar{\Omega} = [0, 1] \times [0, 1]$, discretized using a uniform mesh of 20×20 bilinear elements (yielding 441 nodal points). The source term has been taken as $f = 1$, constant, and the diffusion coefficient has been set to $k = 10^{-4}$. The velocity vector has been taken as $\boldsymbol{a} = |a|(\cos(\pi/3), \sin(\pi/3))$, so that it is not aligned with the finite element mesh.

Three different cases have been considered, corresponding to dominant advection, dominant reaction and combination of advection and reaction effects. These cases are:

(a) $|a| = 1 , \quad s = 0.0001$

(b) $|a| = 0.0001 , \quad s = 1$

(c) $|a| = 0.5 , \quad s = 1$

Results for the first case are shown in Fig. 1. Only those corresponding to the SUPG method have been shown, since for small values of s the GLS and the SGS methods give also the same results. The solution shows some oscillations near the boundary layer created due to the smallness of k.

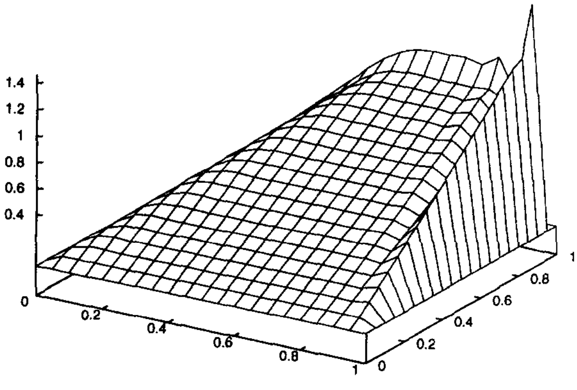
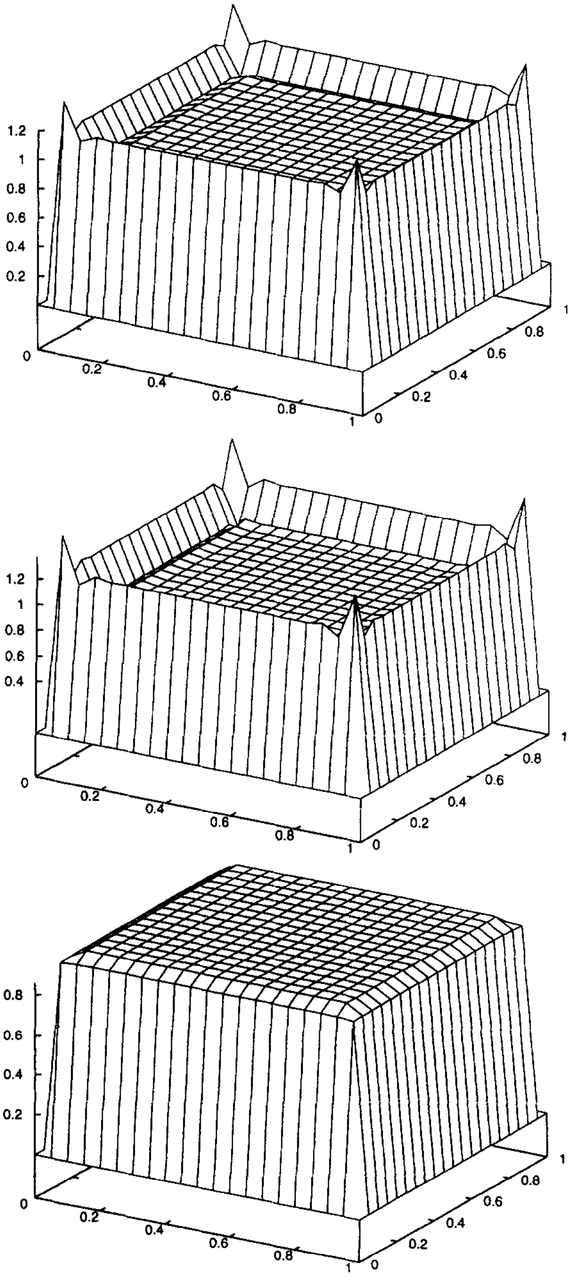


Fig. 1. Case $|a| = 1$, $s = 0.0001$, SUPG method.

Results for case (b) are shown in Fig. 2. The difference between the three methods is there obvious. Only the SGS method does not present any oscillation. The overshoot for the GLS method is stronger than that obtained with the SUPG formulation.

In (c) the effects of convection and reaction are both present, and thus there are oscillations for the three methods due to the presence of convection. Results are shown in Fig. 3. It can be noticed that, even though the convective and reactive terms have a similar influence in the solution for the values of the parameters taken, Ab is much smaller than Pe, that is to say, the oscillations are dominated by those due to convection.

In Fig. 4 we have plotted the results obtained using the SUPG method on a much finer mesh of 52 × 52 bilinear elements refined near the boundaries. There is only an overshoot at (1, 1) in the cases with convection.



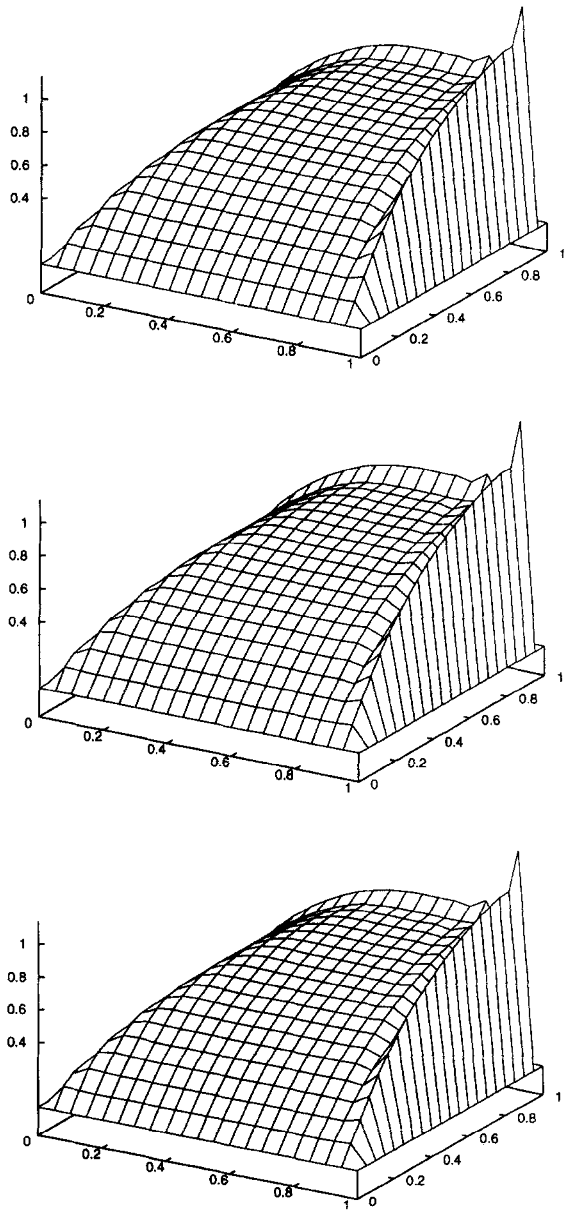


Fig. 2. Case $|\boldsymbol{a}| = 0.0001$, $s = 1$. From the top to the bottom: SUPG, GLS and SGS methods.

Fig. 3. Case $|\boldsymbol{a}| = 0.5$, $s = 1$. From the top to the bottom: SUPG, GLS and SGS methods.

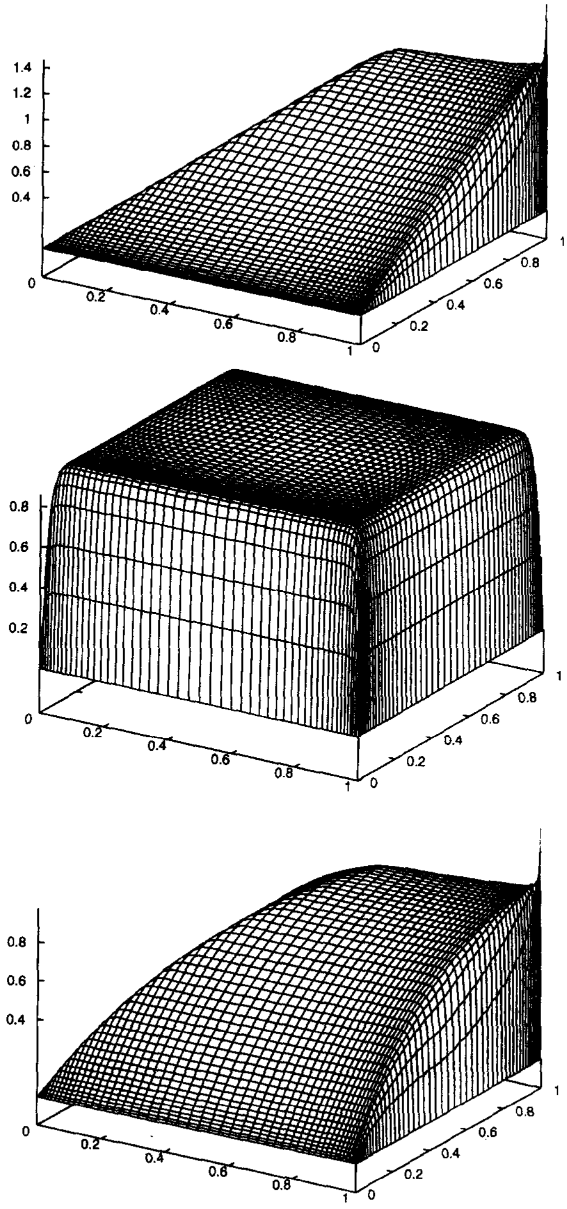


Fig. 4. Fine mesh using the SUPG method. From the top to the bottom: $|\boldsymbol{a}| = 1$, $s = 0.0001$; $|\boldsymbol{a}| = 0.0001$, $s = 1$; $|\boldsymbol{a}| = 0.5$, $s = 1$.

9. Comparison of the methods and conclusions

All the methods that have been considered heretofore can be defined by the expression of $\mathcal{P}$, $\mathcal{R}$ and $\boldsymbol{\tau}$ in Eq. (17).

Concerning the expression of the element residual $\mathcal{R}(\boldsymbol{U}_h)$, it is given by

$$\mathcal{R}(\boldsymbol{U}_h) = \mathcal{L}(\boldsymbol{U}_h) - \boldsymbol{F} = \frac{\partial}{\partial x_i}(\boldsymbol{A}_i\boldsymbol{U}_h) - \frac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij}\frac{\partial \boldsymbol{U}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{U}_h - \boldsymbol{F}\,, \tag{115}$$

with two remarks to be made. First, in the case of the SUPG method, if it is applied to the problem already discretized in time, the incremental quotient $\Delta \boldsymbol{U}_h / \Delta t$ must be included in $\mathcal{R}$. This leads to a non-symmetric modification of the mass matrix associated to the original Galerkin method. However, there is the possibility of using the SUPG method together with a space–time finite element discretization. In this case, the term to be added to $\mathscr{R}$ in Eq. (115) is $\partial \boldsymbol{U}_h / \partial t$. The same happens when space–time finite elements are used for the GLS and the SGS methods.

For the CG and the TG methods, matrix $\boldsymbol{\tau}$ is in principle $\Delta t / 2\boldsymbol{I}$, although the use of different time steps for the different equations and at different points can be justified heuristically. For the SUPG, GLS and SGS methods its design is based on simple model problems and using 'ad hoc' extensions, sometimes helped by convergence analyses. The design idea that we have followed here is based on the discrete maximum principle, although that based on the approximation of the Green's function described in Section 5 seems promising.

The real difference between the stabilizing properties of the different methods is the operator $\mathscr{P}$ applied to the test functions. It is responsible for the conservation properties of the scheme. In particular, condition (24) is in general satisfied by the SUPG, the SGS and the TG methods, but not by the GLS and the CG methods. The expressions of $\mathscr{P}$ for all these techniques are collected in the table below, where the calls refer to the following remarks:

(1) If the SUPG method is used together with a space–time interpolation, then the term $\partial \boldsymbol{V}_h / \partial t$ must be added to $\mathscr{P}(\boldsymbol{V}_h)$. Another possible version of the SUPG method is to take $\mathscr{P}$ as indicated in Eq. (34). Observe that this approach is closely related to the CG method and has the same drawback: its definition is based on a particular set of variables $\boldsymbol{U}$.

(2) This expression corresponds to the constant-in-time approximation in the case of transient problems, which is first order in time. In general, the term $\partial \boldsymbol{V}_h / \partial t$ should be added.

(3) The version of the SGS included here corresponds to the case of an algebraic approximation (60) to the operator M defined in Eq. (53).

(4) This corresponds to the case in which a local expansion of the unknown along the characteristics is performed. In the previous derivation, the term $\mathscr{R}(\boldsymbol{U}_h)$ was evaluated at the time step n where the unknown was already computed, but there are also other possibilities, all based on the use of the Crank–Nicolson scheme for the discretization of the total derivative. In the scalar case, observe that $\nabla \cdot (\boldsymbol{a} v_h) = \boldsymbol{a} \cdot \nabla v_h + v_h \nabla \cdot \boldsymbol{a}$, and thus $\mathscr{P}(v_h)$ is the same as for the SUPG method if $\nabla \cdot \boldsymbol{a} = 0$. In general, $\boldsymbol{a} \cdot \nabla v_h = \mathscr{L}_{\text{conv}}(v_h)$ and $\nabla \cdot (\boldsymbol{a} v_h) = -\mathscr{L}^*_{\text{conv}}(v_h)$.

(5) As in the previous case, the term $\mathscr{R}(\boldsymbol{U}_h)$ is in principle evaluated at the time step n where the unknown is already computed. This scheme is derived from a finite difference time discretization and resembles very closely the algebraic subgrid scale model, the difference being in the expression for $\boldsymbol{\tau}$. Recall that its derivation is based on the fact that the coefficient matrices are not time dependent.

Expressions of $\mathscr{P}$

Method		$\mathscr{P}(\boldsymbol{V}_h)$
SUPG[1]	$\mathscr{L}_{\text{conv,nc}}(\boldsymbol{V}_h)$	$= \boldsymbol{A}_i \dfrac{\partial \boldsymbol{V}_h}{\partial x_i}$
ST-GLS[2]	$\mathscr{L}(\boldsymbol{V}_h)$	$= \dfrac{\partial}{\partial x_i}(\boldsymbol{A}_i \boldsymbol{V}_h) - \dfrac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij} \dfrac{\partial \boldsymbol{V}_h}{\partial x_j}\right) + \boldsymbol{S}\boldsymbol{V}_h$
SGS[3]	$-\mathscr{L}^*(\boldsymbol{V}_h)$	$= \boldsymbol{A}_i^{\mathrm{t}} \dfrac{\partial \boldsymbol{V}_h}{\partial x_i} + \dfrac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij} \dfrac{\partial \boldsymbol{V}_h}{\partial x_j}\right) - \boldsymbol{S}^{\mathrm{t}}\boldsymbol{V}_h$
CG[4]	$-(\text{diag}\ \mathscr{L}_{\text{conv,nc}})^*(\boldsymbol{V}_h)$	$= \dfrac{\partial}{\partial x_i}(\text{diag}(\boldsymbol{A}_i)\boldsymbol{V}_h)$
TG[5]	$-\mathscr{L}^*(\boldsymbol{V}_h)$	$= \boldsymbol{A}_i^{\mathrm{t}} \dfrac{\partial \boldsymbol{V}_h}{\partial x_i} + \dfrac{\partial}{\partial x_i}\left(\boldsymbol{K}_{ij} \dfrac{\partial \boldsymbol{V}_h}{\partial x_j}\right) - \boldsymbol{S}^{\mathrm{t}}\boldsymbol{V}_h$

References

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

[2] 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.

[3] T.J.R. Hughes, Multiscale phenomena: Green's function, subgrid scale models, bubbles, and the origins of stabilized formulations, in M. Morandi Cecchi, K. Morgan, J. Periaux, B.A. Schrefler and O.C. Zienkiewicz, eds., Proc. 9th Int. Conf. on Finite Elements in Fluids. New Trends and Applications (Venezia, Italy, 1995).

[4] T.J.R. Hughes, Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Comput. Methods Appl. Mech. Engrg. 127 (1995) 387–401.

[5] L.P. Franca and C. Farhat, Bubble functions prompt unusual stabilized finite element methods, Comput. Methods Appl. Mech. Engrg. 123 (1994) 299–308.

[6] J. Douglas and T.F. Russell, Numerical methods for convection dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 19 (1982) 871–885.

[7] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier–Stokes equations, Numer. Math. 38 (1982) 309–332.

[8] R. Löhner, K. Morgan and O.C. Zienkiewicz, The solution of non-linear hyperbolic equation systems by the finite element method, Int. J. Numer. Methods Fluids 4 (1984) 1043–1063.

[9] O.C. Zienkiewicz and R. Codina, A general algorithm for compressible and incompressible flow. Part I: The split, characteristic based scheme, Int. J. Numer. Methods Fluids 20 (1995) 869–885.

[10] J. Donea, A Taylor–Galerkin method for convection transport problems, Int. J. Numer. Methods Engrg. 20 (1984) 101–119.

[11] R.J. LeVeque, Numerical Methods for Conservation Laws (Birkhäuser, 1990).

[12] J. von Neumann and R.D. Richtmyer, A method for the numerical calculation of hydrodynamical shocks, J. Appl. Phys. 21 (1950) 232.

[13] D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz and J.C. Heinrich, A note on upwinding and anisotropic balancing dissipation in finite element approximations to convective diffusion problems, Int. J. Numer. Methods Engrg. 15 (1980) 1705–1711.

[14] T.J.R. Hughes and A. Brooks, A multi-dimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes, ed., FEM for Convection Dominated Flows (ASME, New York, 1979).

[15] T.J.R. Hughes and A.N. Brooks, A theoretical framework for Petrov–Galerkin methods, with discontinuous weighting functions: applications to the streamline upwind procedure, in: R.H. Gallagher, D.M. Norrie, J.T. Oden and O.C. Zienkiewicz, eds., Finite Element in Fluids, vol. IV (Wiley, London, 1982) 46–65.

[16] R. Codina, E. Oñate and M. Cervera, The intrinsic time for the SUPG formulation using quadratic elements, Comput. Methods Appl. Mech. Engrg. 94 (1992) 239–262.

[17] 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.

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

[19] P. Lesaint and P.A. Raviart, On a finite element method for solving the neutron transport equation, in: C. de Boor, ed., Mathematical aspects of the Finite Element Method (Academic Press, 1974).

[20] T.J.R. Hughes, L.P. Franca and M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: A stable Petrov–Galerkin formulation for the Stokes problem accommodating equal-order interpolations, Comput. Methods Appl. Mech. Engrg. 59 (1986) 85–99.

[21] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods (Springer-Verlag, 1991).

[22] T.J.R. Hughes and L.P. Franca, A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces, Comput. Methods Appl. Mech. Engrg. 65 (1987) 85–96.

[23] L.P. Franca and R. Stenberg, Error analysis of some Galerkin least-squares methods for the elasticity equations, SIAM J. Numer. Anal. 28 (1991) 1680–1697.

[24] F. Shakib and T.J.R. Hughes, A new finite element formulation for computational fluid dynamics: IX. Fourier Analysis of space–time Galerkin/least-squares algorithms, Comput. Methods Appl. Mech. Engrg. 87 (1991) 35–58.

[25] I. Harari and T.J.R. Hughes, What are C and h?: Inequalities for the analysis and design of finite element methods, Comput. Methods Appl. Mech. Engrg. 97 (1992) 157–192.

[26] F. Shakib, T.J.R. Hughes and Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 89 (1991) 141–219.

[27] J. Douglas and J. Wang, An absolutely stabilized finite element method for the Stokes problem, Math. Comput. 52 (1989) 495–508.

[28] L. Franca, S.L. Frey and T.J.R. Hughes, Stabilized finite element methods: I. Application to the advective–diffusive model, Comput. Methods Appl. Mech. Engrg. 95 (1992) 253–276.

[29] R.E. Bank and B.D. Welfert, A comparison between the mini-element and the Petrov–Galerkin formulations for the generalized Stokes problem, Comput. Methods Appl. Mech. Engrg. 83 (1990) 61–68.

[30] F. Brezzi, M.O. Bristeau, L. Franca, M. Mallet and G. Rogé, A relationship between stabilized finite element methods and the Galerkin method with bubble functions, Comput. Methods Appl. Mech. Engrg. 96 (1992) 117–129.

[31] D.N. Arnold, F. Brezzi and M. Fortin, A stable finite element for the Stokes equations, Calcolo 21 (1984) 337–344.

[32] C. Baiocchi, F. Brezzi and L.P. Franca, Virtual bubbles and Galerkin-least-squares type methods (Ga.L.S), Comput. Methods Appl. Mech. Engrg. 105 (1993) 125–141.

[33] R. Codina, Numerical solution of the incompressible Navier–Stokes equations with Coriolis forces based on the discretization of the total time derivative, CIMNE Report, Num. 102, 1996.

[34] O.C. Zienkiewicz and R.L. Taylor, The Finite Element Method, 4th edition, Vol. 2 (McGraw-Hill, 1989).

[35] P.G. Ciarlet and P.A. Raviart, Maximum principle and uniform convergence for the finite element method, Comput. Methods Appl. Mech. Engrg. 2 (1973) 17–31.

[36] R. Codina, A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection–diffusion equation, Comput. Methods Appl. Mech. Engrg. 110 (1993) 325–342.