

Time accurate consistently stabilized mesh-free methods for convection dominated problems

Antonio Huerta* and Sonia Fernández-Méndez

Departament de Matemàtica Aplicada III, E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain.

SUMMARY

The behavior of high-order time stepping methods combined with mesh-free methods is studied for the transient convection-diffusion equation. Particle methods, such as the Element-Free Galerkin (EFG) method, allow to easily increase the order of consistency and, thus, to formulate high-order schemes in space and time. Moreover, second derivatives of the EFG shape functions can be constructed with a low extra cost and are well defined, even for linear interpolation. Thus, consistent stabilization schemes can be considered without loss in the converge rates.

KEY WORDS: Meshless, Mesh-free, finite elements, convection-diffusion, transient, least-squares, streamline-upwind Petrov-Galerkin

1. INTRODUCTION

A great deal of effort has been devoted in the recent years to the development of finite element methods for the numerical approximation of transient convection-diffusion problems. It is well known that the standard Galerkin is not ideally suited to deal with the spatial discretization of convection dominated problems. Thus, other spatial discretization techniques are necessary to introduce the suitable amount of numerical dissipation in the presence of internal and boundary layers. In truly transient problems, another equally important aspect is to ensure an adequate coupling between the spatial approximation and the time integration scheme. Now, it is also well known that the combination of a standard Galerkin spatial discretization with classical second order accurate time stepping schemes, such as Lax-Wendroff and Crank-Nicolson methods, fails to produce satisfactory numerical results when convection dominates the transport process (no internal or boundary layers being present) [1, 2]. In conclusion, transient convection-diffusion problems need stabilization because of the internal and boundary

*Correspondence to: Professor Antonio Huerta, Departament de Matemàtica Aplicada III, E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, SPAIN. E-mail: antonio.huerta@upc.es

Contract/grant sponsor: Ministerio Ciencia y Tecnología; contract/grant number: REN2001-0925-C03-01

layers, and high-order time stepping schemes for accuracy in convection dominated problems. Note that in real engineering problems the relative weight of the convective and diffusive term may vary in space and time. This effect is usually accentuated by a non uniform mesh due to geometrical considerations. In order to overcome some of these difficulties, in [3] a unified approach for high-order implicit time stepping is presented. Then, in [4, 5] these methods are combined with classical stabilization techniques in the framework of the finite element method.

However, finite element methods are usually non consistently stabilized. The second derivatives needed in the stabilization term are usually neglected. This implies a loss of convergence, which is more dramatic in high-order time accurate methods as it will be shown here. Several solutions are apparent and some of them have been explored. For instance, the stabilization parameter (*i.e.* the intrinsic time τ) could be modified for each integration scheme in order to ensure high-order time convergence rates, even if second derivatives are still not computed. Another alternative in order to preclude the computation of second derivatives is to take into account flux jumps across element boundaries [6]. Finally, second derivatives can be approximated [7]. This implies a global reconstruction for linear elements (a global least-squares problem). For high order elements, a local (element by element) reconstruction or the usual computation of the second derivatives can be envisioned. As claimed in [7] both alternatives have a similar computational cost and ensure the theoretical convergence rates.

In mesh-free methods, see [8], second derivatives are computed at a reduced extra cost, that is an extra cost considerably lower than in finite elements. Moreover the mesh-free interpolation space is a subset of $\mathcal{H}^2(\Omega)$ and thus, the standard stabilization techniques can be formulated naturally in $\mathcal{H}^2(\Omega)$. Second order space derivatives of the interpolation are well defined in the whole domain even with linear consistency. There are not neglected terms in the weak form and the convergence rates of the high-order time integration schemes are preserved. Moreover, the moving least-squares (MLS) interpolation allows increasing the order of the spatial discretization in a simple manner. With a mesh-free interpolation, space and time high-order consistent stabilized formulations are easily defined and implemented.

This paper extends the time accurate schemes with classical stabilization techniques presented in [4] to mesh-free methods. And compares the performance of such consistently stabilized methods with finite element results. It is organized as follows. In section 3 the Galerkin formulation, the stabilized least-squares formulation and the Streamline-Upwind Petrov-Galerkin formulation are considered in combination with two different time stepping multi-stage implicit methods, with second and fourth order respectively. In fact, all formulations are first recalled in the framework of finite elements in section 3.2. Then, section 3.3 is devoted to the particularization of these formulations for a mesh-free interpolation. In section 4, a 1D numerical example allows to analyze the converge in time for the proposed formulations. The numerical experiments agree with the conclusions derived in the previous sections. Finally, numerical examples with mesh-free interpolation are shown in section 5 in order to illustrate the importance of time accurate computations and the need of stabilized formulations in the resolution of transient convection-diffusion problems.

2. PRELIMINARIES ON THE ELEMENT FREE GALERKIN METHOD

This section will not be devoted to develop or discuss mesh-free methods in detail. There are well known references with excellent presentations of mesh-free methods, see for instance

[10, 11, 12, 13, 14]. Here some basic notions will be recalled in order to introduce the notation and the approach employed in following sections.

Mesh-free methods, or particle methods, are based in a functional interpolation of the form:

$$u(x) \simeq u^\rho(x) = \sum_{j \in I^\rho} N_j^\rho(x) u(x_j), \quad (1)$$

given a number of particles $\{x_j\}_{j \in I^\rho}$ in the domain Ω , $\Omega \subset \mathbb{R}^2$. The interpolation functions, $N_j^\rho(x)$, must be determined in a proper manner. In reproducing kernel particle methods [12, 13] (RKPM) the interpolation functions are obtained in the framework of the MLS interpolation. The element free Galerkin method [15, 16] (EFG) can be viewed as a particular case of the previous formulation [13].

Let us recall, in the context of the EFG method, how the interpolation (shape) functions are obtained. They are defined as:

$$N_j^\rho(x) = \mathbf{P}^T(x_j) \boldsymbol{\alpha}(x) \phi\left(\frac{x - x_j}{\rho}\right), \quad (2)$$

where the vector $\boldsymbol{\alpha}(x)$ in \mathbb{R}^{l+1} is unknown and $\mathbf{P}(x) = \{p_0(x), p_2(x), \dots, p_l(x)\}^T$ includes a complete basis of polynomials of degree less or equal m . In one dimension, it is usual that $p_i(x)$ coincides with the monomials x^i , and, in this particular case, $l = m$. The function $\phi(x)$ is a weighting function (positive, even and with compact support) which characterizes the mesh-free method. For instance, if $\phi(x)$ is continuous together with its first k derivatives, the interpolation is also continuous together with its first k derivatives. In (2) the weighting function has been translated, i.e. centered in x_j , and its support scaled by the dilation parameter ρ .

The unknown vector $\boldsymbol{\alpha}(x)$ is determined imposing the so-called reproducibility or consistency condition. In fact, it is equivalent to a Moving Least Squares development. This reproducibility condition imposes that the interpolation proposed in (1) is exact for all the polynomials in \mathbf{P} , i.e.

$$\mathbf{P}(x) = \sum_{j \in I^\rho} \mathbf{P}(x_j) N_j^\rho(x). \quad (3)$$

After substitution of (2) in (3) the usual linear system of equations, that determines $\boldsymbol{\alpha}(x)$, is obtained:

$$\mathbf{M}(x) \boldsymbol{\alpha}(x) = \mathbf{P}(x), \quad (4)$$

with

$$\mathbf{M}(x) = \sum_{j \in I^\rho} \mathbf{P}(x_j) \mathbf{P}^T(x_j) \phi\left(\frac{x - x_j}{\rho}\right). \quad (5)$$

The matrix $\mathbf{M}(x)$ must be regular. In [13] and [17] there is a discussion on the necessary conditions for the particle distribution. Finally, the approximation that will be used in the weak form of the boundary value problem is (1) where the shape functions $N_j^\rho(x)$ are defined by (2), (4) and (5), and u_j^ρ are the coefficients to be determined.

For computational purposes, it is usual to centre in x_j and scale with ρ also the polynomials involved in previous expressions. Thus, another equivalent expression for the shape functions is usually employed, see [13, 17] for details.

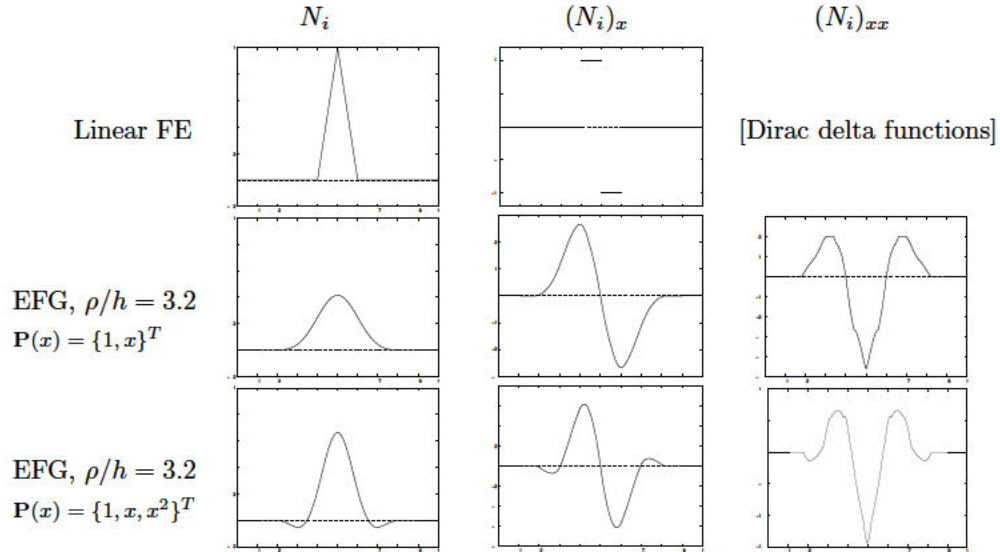


Figure 1. Shape function and derivatives for the finite element and the EFG interpolation

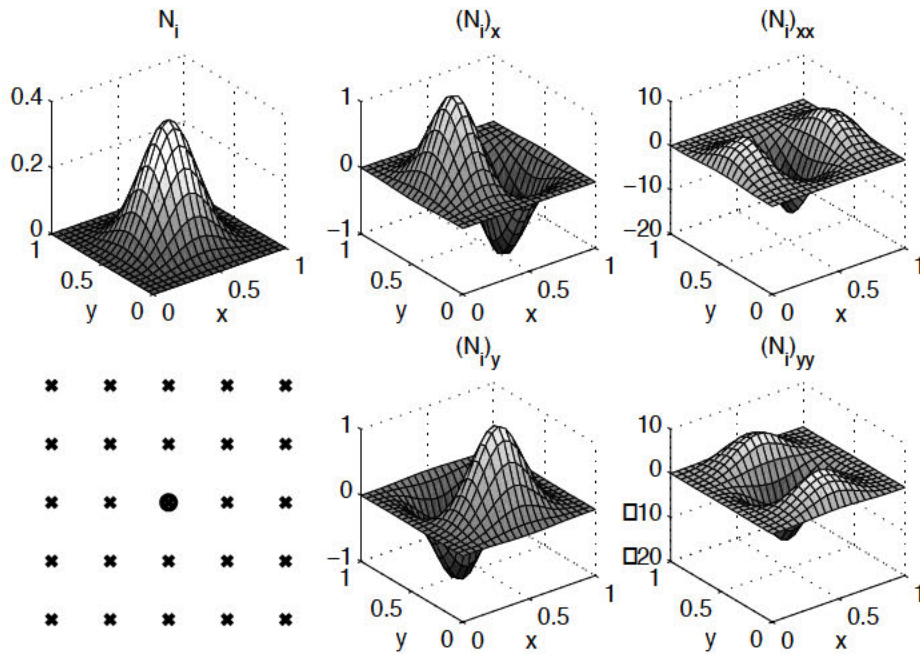


Figure 2. 2D particle distribution, EFG shape function and its derivatives

Remark 1. The weighting function introduced in (2) may be defined in various manners. Here, for simplicity a rectangular support is chosen. Thus, $\phi(\mathbf{x}) = \phi_{1D}(|x_1|) \phi_{1D}(|x_2|)$ where ϕ_{1D} is the cubic spline in one dimension,

$$\phi_{1D}(s) = \begin{cases} \frac{2}{3} + 4(s-1)s^2 & 0 \leq s \leq 0.5 \\ \frac{4}{3}(1-s)^3 & 0.5 \leq s \leq 1 \\ 0 & 1 \leq s. \end{cases}$$

However similar results can be obtained with circular supports, $\phi(\mathbf{x}) = \phi_{1D}(|\mathbf{x}|)$. Since $\phi \in \mathcal{C}^2$, the EFG shape functions are \mathcal{C}^2 (see [13]) and thus, as can be seen in Figures 1 and 2, first and second derivatives of the shape functions are well defined the whole the domain, even with linear consistency.

Remark 2. The interpolation is characterized by the order of consistency required, *i.e.* the basis of polynomials employed \mathbf{P} , and by the ratio between the dilation parameter ρ and the particle distance h , ρ/h (see [17]). Note that the order of the interpolation can be easily increased and thus high-order schemes, in both space and time, can be implemented in a simple way.

Remark 3 (Computation of derivatives) Belytschko and coworkers proposed in [8] an efficient computation of the derivatives of the shape functions. It is based on implicit derivation of (4). For example, in 1D the first and second derivatives of the shape function (2) can be expressed as

$$\begin{aligned} \frac{dN_j^\rho}{dx} &= \mathbf{P}^T(x_j) \left[\boldsymbol{\alpha}_x(x) \phi\left(\frac{x-x_j}{\rho}\right) + \boldsymbol{\alpha}(x) \frac{1}{\rho} \phi_x\left(\frac{x-x_j}{\rho}\right) \right], \\ \frac{d^2N_j^\rho}{dx^2} &= \mathbf{P}^T(x_j) \left[\boldsymbol{\alpha}_{xx}(x) \phi\left(\frac{x-x_j}{\rho}\right) + 2\boldsymbol{\alpha}_x(x) \frac{1}{\rho} \phi_x\left(\frac{x-x_j}{\rho}\right) + \boldsymbol{\alpha}(x) \frac{1}{\rho^2} \phi_{xx}\left(\frac{x-x_j}{\rho}\right) \right], \end{aligned}$$

where \square_x denotes de derivative $d\square/dx$. Vectors $\boldsymbol{\alpha}_x$ and $\boldsymbol{\alpha}_{xx}$ can be obtained solving two small linear system of equations

$$\mathbf{M} \boldsymbol{\alpha}_x = \mathbf{P}_x - \mathbf{M}_x \boldsymbol{\alpha}, \quad \mathbf{M} \boldsymbol{\alpha}_{xx} = \mathbf{P}_{xx} - \mathbf{M}_{xx} \boldsymbol{\alpha} - 2\mathbf{M}_x \boldsymbol{\alpha}_x,$$

which are easily deduced by implicit derivation of (4). Note that matrix $\mathbf{M} \in \mathbb{R}^{(l+1) \times (l+1)}$ and its factorization has been computed previously in order to obtain vector $\boldsymbol{\alpha}$, so computation of first and second derivatives only involves the resolution of two small systems of equations through two backward and forward substitutions. Second derivatives of the shape functions can be easily computed in EFG with negligible extra computational cost.

3. TIME AND SPACE DISCRETIZATION FOR THE TRANSIENT CONVECTION-DIFFUSION EQUATION

We are concerned with an accurate solution of the following transient convection-diffusion boundary value problem with constant coefficients: “Given the velocity field $\mathbf{a}(\mathbf{x})$, the constant diffusion coefficient ν and the source term $s(\mathbf{x}, t)$, find $u(\mathbf{x}, t)$ such that it verifies

$$u_t + \mathbf{a} \cdot \nabla u - \nu \nabla^2 u = s \quad \text{in } \Omega \times (0, T), \quad (6)$$

and the initial,

$$u(x, 0) = u_0(x) \quad \text{in } \Omega,$$

and boundary conditions,

$$\begin{aligned} u &= u_D && \text{in } \Gamma, \\ \nu \nabla u \cdot \mathbf{n} &= f && \text{in } \partial\Omega \setminus \Gamma, \end{aligned}$$

where \mathbf{n} is the unitary outward normal vector". Symbolically the partial differential equation (6) may be rewritten as

$$u_t + \mathcal{L}(u) = s, \quad (7)$$

where the spatial differential operator is defined as

$$\mathcal{L} := \mathbf{a} \cdot \nabla - \nu \nabla^2. \quad (8)$$

The numerical solution of such a problem clearly involves a double discretization process, that is, time discretization and space discretization. In this paper, the methodology proposed in [4] is considered: (1) multi-stage time stepping schemes of high accuracy are used and (2) standard stabilization techniques are implemented. The objective is to adapt the standard stabilization techniques, well known in stationary problems, to transient problems in combination with multi-stage time stepping schemes. Consequently, in the following, time discretization precedes spatial discretization.

3.1. Time discretization

For illustration proposes, only two implicit multi-stage methods are presented. A more complete presentation can be found in [3]. These schemes can be written in incremental form as

$$\frac{\Delta \mathbf{u}}{\Delta t} - \mathbf{W} \Delta \mathbf{u}_t = \mathbf{w} u_t^n, \quad (9)$$

where the unknown $\Delta \mathbf{u} \in \mathbb{R}^{\mathbf{n}_{\text{stg}}}$ is a vector whose dimension is the number of stages, \mathbf{n}_{stg} . The vector $\Delta \mathbf{u}_t$ is the partial derivative of $\Delta \mathbf{u}$ with respect to time. The time derivatives in (9) are replaced by spatial derivatives using the original differential equation (7):

$$\frac{\Delta \mathbf{u}}{\Delta t} + \mathbf{W} \mathcal{L}(\Delta \mathbf{u}) = \mathbf{w} [s^n - \mathcal{L}(u^n)] + \mathbf{W} \Delta s \quad (10)$$

The precise definition of $\Delta \mathbf{u}$, Δs , \mathbf{w} and \mathbf{W} depends on each particular method:

Second order Padé approximation: R_{11} (Crank-Nicolson)

$$\boxed{\begin{aligned} \Delta \mathbf{u} &= u^{n+1} - u^n, & \Delta s &= s^{n+1} - s^n, \\ \mathbf{W} &= \frac{1}{2} & \mathbf{w} &= 1. \end{aligned}} \quad (11)$$

Note that in this case $\mathbf{n}_{\text{stg}} = 1$ and the vectors and matrix in (9) become scalars. In fact, this scheme corresponds to the well known Crank-Nicolson method.

Fourth order Padé approximation: R_{22}

$$\boxed{\begin{aligned} \Delta \mathbf{u} &= \begin{Bmatrix} u^{n+\frac{1}{2}} - u^n \\ u^{n+1} - u^{n+\frac{1}{2}} \end{Bmatrix}, & \Delta \mathbf{s} &= \begin{Bmatrix} s^{n+\frac{1}{2}} - s^n \\ s^{n+1} - s^{n+\frac{1}{2}} \end{Bmatrix}, \\ \mathbf{W} &= \frac{1}{24} \begin{bmatrix} 7 & -1 \\ 13 & 5 \end{bmatrix}, & \mathbf{w} &= \frac{1}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}. \end{aligned}} \quad (12)$$

For this methods the truncation error induced by (9) is precisely $\mathcal{O}(\Delta t^{2n_{\text{stg}}})$. However, it is important to note that the methodology proposed here is independent of these two particular schemes and can be implemented with any multi-stage algorithm.

Note that equations (9) and (10), with the corresponding initial and boundary conditions, define a problem in *strong form* which must be solved at each time step n . That is, at each time step, the solution at time t^n , u^n , is assumed to be known and the partial differential equation (10), which only involves spatial derivatives, must be solved in order to obtain $\Delta \mathbf{u}$. Then, the solution at time t^{n+1} , u^{n+1} , is computed from u^n and $\Delta \mathbf{u}$.

Once the time discretization has been performed, both the mesh-free EFG interpolation or the finite element interpolation can be used in the discretization of the resulting partial differential equation, (9) or (10), at each time step. Next, the Galerkin formulation and the standard stabilization techniques are recalled in the context of the finite element interpolation, then these formulations are extended for the mesh-free interpolation.

Remark 4. The multi-stage time integration schemes used here, and proposed in [3], have strong similarities with implicit Runge-Kutta methods based on the Lobatto IIIA quadrature. While the present schemes are based on non-overlapping step increments, the corresponding Runge-Kutta methods employ total increments. For instance, the method corresponding to R_{22} would use

$$\begin{Bmatrix} u^{n+1/2} - u^n \\ u^{n+1} - u^n \end{Bmatrix}$$

as the unknown. This minor detail does not affect linear problems with a Galerkin formulation. However, see [5], it has major consequences as regards numerical stability when implemented in connection with stabilized spatial formulations.

3.2. Finite Element spatial discretization. Galerkin and stabilized formulations

Application of the Galerkin formulation to the time stepping scheme (9) results in the following weak form

$$\left(\mathbf{v}, \frac{\Delta \mathbf{u}}{\Delta t} \right)_{\Omega} - \left(\mathbf{v}, \mathbf{W} \Delta \mathbf{u}_t \right)_{\Omega} = \left(\mathbf{v}, \mathbf{w} u_t^n \right)_{\Omega} \quad \forall \mathbf{v} \in [\mathcal{V}_0^h]^{n_{\text{stg}}}, \quad (13)$$

with \mathcal{V}_0^h subset of the usual functional space $\mathcal{H}_0^1 = \{v | v \in \mathcal{H}^1, v = 0 \text{ on } \partial\Omega \setminus \Gamma\}$ and with the standard definition of the vector scalar product $(\mathbf{v}, \mathbf{u})_{\Omega} = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} d\Omega$.

Note that the right-hand side of (13) is known, and that the second term of the left-hand side, *i.e.* $\Delta \mathbf{u}_t$, includes the Laplace operator, which is integrated by parts (see [3, 4] for extended details).

In order to stabilize the convective term in a consistent manner (ensuring that the solution of the differential equation is also solution of the weak form), Hughes and coworkers have proposed several techniques [18, 9]. An extra term weighted over the element interiors is added

to the Galerkin weak form. This added term is a function of the residual of the differential equation to ensure consistency. These methods are designed for the steady convection-diffusion equation and subsequently extended to transient problems with second order time schemes and to space-time formulations. The combination of high order time stepping schemes such as (9) with standard stabilization techniques is proposed in [4, 5]. The considered stabilization techniques are *streamline-upwind Petrov-Galerkin* (SUPG), *Galerkin least-squares* (GLS), *sub-grid scale* (SGS) and a *least-squares* (LS) type stabilization. Here, only the LS and the SUPG formulation are recalled. However, generalization to other formulations is straightforward.

In order to have a consistent stabilization a residual must be defined. The residual in this case is chosen after time discretization. Thus, from (9) the residual is defined as

$$\mathcal{R}(\Delta \mathbf{u}) := \frac{\Delta \mathbf{u}}{\Delta t} - \mathbf{W} \Delta \mathbf{u}_t - \mathbf{w} u_t^n \quad (14)$$

and the consistently stabilized weak form is

$$\left(\mathbf{v}, \frac{\Delta \mathbf{u}}{\Delta t} \right)_\Omega - \left(\mathbf{v}, \mathbf{W} \Delta \mathbf{u}_t \right)_\Omega + \sum_e (\tau \mathcal{P}(\mathbf{v}), \mathcal{R}(\Delta \mathbf{u}))_{\Omega^e} = \left(\mathbf{v}, \mathbf{w} u_t^n \right)_\Omega. \quad (15)$$

Note that the stabilization term is added to the Galerkin weak form. In the stabilization term a free parameter τ appears, the intrinsic time scale, see [9, 6]). The operator \mathcal{P} characterizes the stabilization technique, see for instance [19] for a general presentation. The truncation errors are not explicitly indicated: a term $\mathcal{O}(\Delta t^{2n_{\text{stg}}})$ can be added to (14) and (15).

The SUPG stabilization is defined by taking

$$\mathcal{P}(\mathbf{v}) := \mathbf{W}(a \cdot \nabla) \mathbf{v}. \quad (16)$$

Note that the matrix \mathbf{W} , which affects the convection term, induces a *non scalar* stabilization (each equation of the multistage formulation is affected by different coefficients). The weak form for the SUPG method is obtained after substitution of (16) in equation (15).

Remark 5. The stabilization term involves the residual, which includes the second-order term $\nabla^2 u$. When linear finite elements are used this term vanishes or is largely under-represented, with the corresponding degradation in the consistency of the stabilized formulation. The lack of consistency leads to errors of order $\mathcal{O}(\tau)$, apart from the errors inherent to the time integration scheme.

Remark 6. In order to keep the convergence rates in time, several possibilities can be useful. The stabilization parameter τ can be defined to be asymptotically of order $\mathcal{O}(\Delta t^{2n_{\text{stg}}})$. That is, a specific intrinsic time τ should be designed for each one of the time integration schemes. Other possibility is to include flux jump terms across the element boundaries in the stabilized formulation to take into account the neglected terms (see [6] for details). In fact, reference [7] shows that when linear finite elements are used the lack of consistency due to the neglected terms leads also to reduced convergence in space. For linear finite elements, it proposes a global reconstruction of second derivatives. This method recovers the ability to approximate the residual in the stabilization term yielding a better consistency, through an iterative process. However, the increase in the computational cost is not negligible: a system of equations with global mass matrix must be solved at each iteration.

Remark 7. The use of high-order finite elements, such as quadratic elements, allows including second derivatives of the approximation in the residual in the stabilization term, and thus, consistent stabilized formulations can be defined. However, the computational cost and the implementation difficulties are highly increased due to the computation of second derivatives of the element mapping [7].

To implement a *least-squares* (LS) formulation the spatial strong form is directly employed to construct the integral equation. Here, since time discretization is already performed, equations (9) and (10) are used. Consequently, one gets

$$\left(\frac{\mathbf{v}}{\Delta t} + \mathbf{W}\mathcal{L}(\mathbf{v}), \mathcal{R}(\Delta\mathbf{u})\right)_\Omega = 0. \quad (17)$$

The correct implementation of a standard least-squares formulation requires to work in \mathcal{H}^2 , *i.e.* \mathbf{v} and \mathbf{u} should be in subspaces of $[\mathcal{H}^2(\Omega)]^{\text{nstg}}$. A mixed formulation reduces the regularity of the spaces but increases the number of nodal unknowns, see [20]. Here, an equivalent form to (17) following the same rationale as for standard stabilized methods, see equation (15), is devised. It is equivalent in the sense that its unique solution is also the unique solution of (9) or (17). The first argument in (17) is split by linearity and the term containing $\mathcal{L}(\mathbf{v})$ is only evaluated in the element interiors, namely

$$(\mathbf{v}, \mathcal{R}(\mathbf{u}))_\Omega + \sum_e (\Delta t \mathbf{W}\mathcal{L}(\mathbf{v}), \mathcal{R}(\Delta\mathbf{u}))_{\Omega_e} = 0. \quad (18)$$

Note that in (18) the interpolation and tests functions can be taken in a subspace of $[\mathcal{H}^{1+}]^{\text{nstg}}$, where $\mathcal{H}^1 \subsetneq \mathcal{H}^{1+} := \{w \in \mathcal{H}^1(\Omega) / w|_{\Omega_e} \in \mathcal{H}^2(\Omega_e) \text{ for all element } \Omega_e\} \subsetneq \mathcal{H}^2$. The final formulation is still symmetric (note that this is not the case for other stabilization techniques such as SUPG, GLS or SGS because \mathbf{W} is, in general, not symmetric), and now \mathcal{C}^0 finite elements can be used. This approach can also be cast in the form of a standard stabilization technique with the following definition of the operator \mathcal{P} and the intrinsic time τ

$$\mathcal{P}(\mathbf{v}) := \mathbf{W}\mathcal{L}(\mathbf{v}), \text{ and } \tau := \Delta t. \quad (19)$$

Remark 8. As in the standard stabilization techniques, some terms with second-order derivatives are neglected or under-represented when finite elements are considered to discretize the least-squares formulation. This leads to errors of order $\mathcal{O}(\Delta t)$ added to the errors of the time integration scheme.

3.3. EFG spatial discretization. Galerkin and stabilized formulations

The EFG interpolation (1), see section 2, can also be considered in the spatial discretization of the Galerkin weak form (13). However, in the presence of internal or boundary layers the typical instabilities of the Galerkin formulation will soon appear. In fact, all the considered stabilized formulations (15) can be discretized with the mesh-free interpolation. Moreover, with a mesh-free method, the interpolation space is a subset of $\mathcal{H}^2(\Omega)$, and thus, this other equivalent expression for the stabilized formulations (15) can be considered

$$\left(\mathbf{v}, \frac{\Delta\mathbf{u}}{\Delta t}\right)_\Omega - (\mathbf{v}, \mathbf{W}\Delta\mathbf{u}_t)_\Omega + (\tau\mathcal{P}(\mathbf{v}), \mathcal{R}(\Delta\mathbf{u}))_\Omega = (\mathbf{v}, \mathbf{w}u_t^n)_\Omega. \quad (20)$$

Note that, since the interpolation is performed in $\mathcal{H}^2(\Omega)$, the integration of the stabilization term can be done in the whole domain Ω . Moreover, the natural expression of the least-squares formulation, equation (17), in $\mathcal{H}^2(\Omega)$ can be employed.

Remark 9. Note that, with the EFG interpolation space, second-order derivatives are well represented, and thus, consistent stabilized formulations can be easily implemented. In fact, there are two possible advantages of the mesh-free interpolation in the resolution of the transient convection-diffusion equation: (1) the order of consistency can be easily increased, and thus, high-order (in space and time) formulations can be implemented in a simple way, and (2) the smoothness of the interpolant allows the definition of consistent stabilized formulations. There are no neglected terms and thus the convergence rates of the time integration schemes are preserved.

4. CONVERGENCE OF GALERKIN AND STABILIZED FORMULATIONS

In this section, a numerical example is considered in order to analyze the convergence in time of the presented formulations and corroborate the conclusions derived in section 3. The 1D convection-diffusion equation with constant coefficients is solved,

$$u_t + au_x = \nu u_{xx}, \quad x \text{ in } (0, 2), \quad t \text{ in } (0, 1),$$

with homogeneous Dirichlet boundary conditions. The initial condition, at $t = 0$, is chosen such that the analytical solution is known

$$u(x, t) = \frac{\sigma_0}{\sigma} e^{-\frac{(x-(x_0+at))^2}{2\sigma^2}}, \quad \sigma^2 = \sigma_0^2 + 2\nu t$$

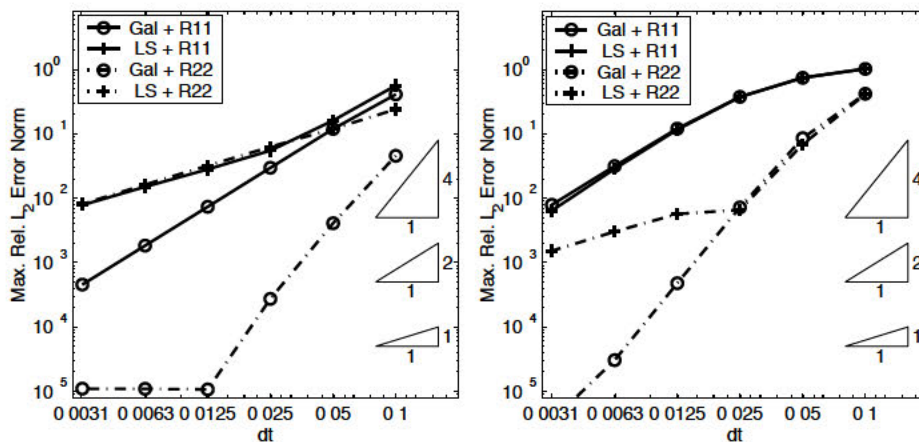


Figure 3. Finite elements convergence results with $h = 0.001$, $a = 1$, $\nu = 10^{-2}$ (left), and $\nu = 10^{-4}$ (right).

Both finite elements and particle results are studied. Figure 3 shows the evolution of the error against the time step for linear finite elements, with element size $h = 0.001$, and for two different values of the diffusion parameter ν . In all figures the error is evaluated in the $\mathcal{L}^2(\Omega)$ norm. In each plot, results are shown for the Galerkin, Gal, and the *incomplete* least-squares, ILS, formulation with two different Padé time stepping schemes: R_{11} and R_{22} with order two and four respectively. In all figures ILS denotes the *incomplete* least-squares formulation, that

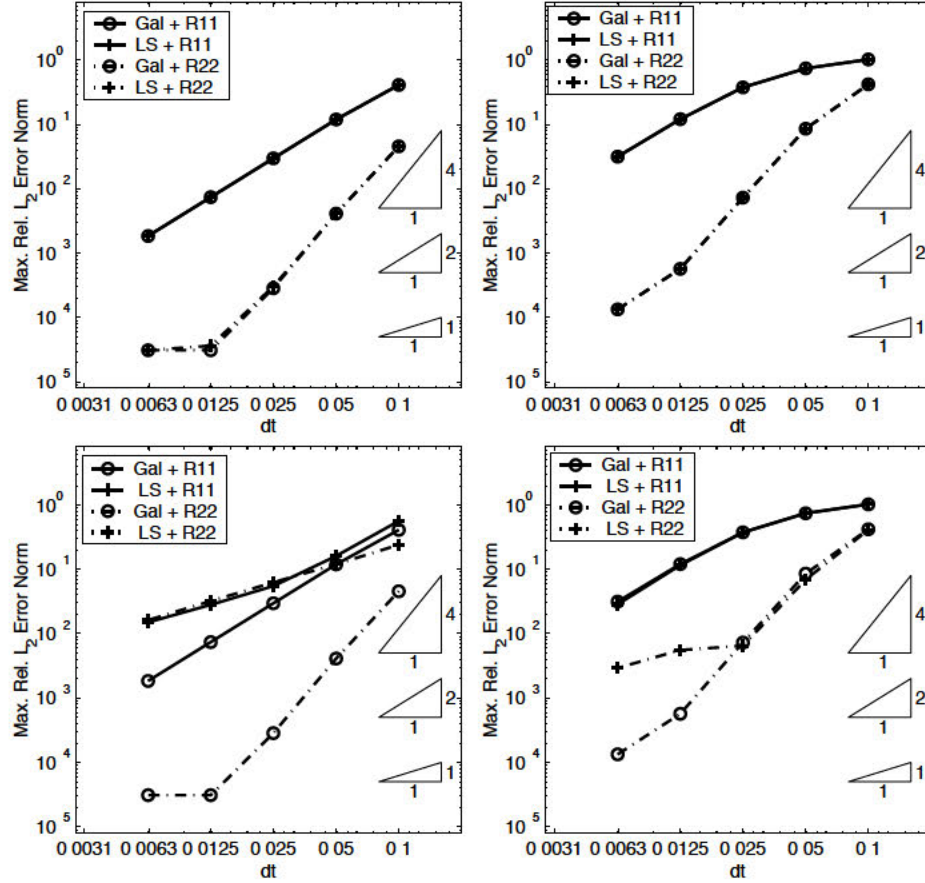


Figure 4. EFG convergence results with $h = 0.01$ (distance between particles), $\rho/h = 3.2$, $\mathbf{P} = \{1, x\}^T$, $a = 1$, $\nu = 10^{-2}$ (left), and $\nu = 10^{-4}$ (right).

is, the particularization of the least-squares formulation (18) for linear finite elements, and thus, with some neglected terms. The convergence rates are as expected when a Galerkin formulation is used. However, when the least-squares formulation (18) is used, the lack of consistency due to the neglected terms leads to errors of order $\mathcal{O}(\Delta t)$, with the corresponding degradation in the convergence rates. When the truncation error of the time integration scheme is small enough, the error is only $\mathcal{O}(\Delta t)$, instead of $\mathcal{O}(\Delta t^2)$ or $\mathcal{O}(\Delta t^4)$.

Figure 4 shows the convergence results for the EFG interpolation with linear consistency, distance between particles $h = 0.01$ and $\rho/h = 3.2$. Results are depicted for the Galerkin formulation, the consistent least-squares formulation (17) and the non-consistent ILS formulation. Once again, the ILS formulation corresponds to the weak form with the same $\mathcal{O}(\Delta t)$ neglected terms as with linear finite elements in Figure 3. The convergence rates of the time stepping schemes are preserved when a consistent weak form is used. In fact, the convergence curves coincide with the Galerkin convergence curves. As expected, convergence degrades, similarly to finite elements, if the non consistent formulation ILS is used.

In order to preserve the high-order convergence rates of the time stepping schemes, a consistent formulation must be used. With a mesh-free method, such as EFG, second order derivatives of the interpolation are well defined in the whole domain, and thus, consistent stabilized formulations, without neglected terms, can be easily defined and implemented.

Remark 10. Note that the spatial discretization limits the precision. That can be observed in the Galerkin case with the R_{22} scheme and $\nu = 10^{-2}$. Moreover, note that with the mesh-free interpolation it is enough to discretize the domain with 201 particles in order to obtain an accuracy in space similar to the accuracy with 2000 finite elements, more or less 10^{-5} . That is, a number of degrees of freedom ten times smaller in EFG than in finite elements. In fact, it is well known that with linear consistency the finite element shape functions can be recovered from the EFG shape functions when $\rho/h \simeq 1$ and, in most of the problems, the EFG interpolation with an appropriate ratio, for example $\rho/h = 3.2$, performs better than the finite element interpolation.

Similar results are obtained for the SUPG stabilized formulation, see figures 5 and 6. When finite elements are used, the intrinsic time τ is computed using the formula proposed in [9], see [6] for an interesting review, that is

$$\tau = \tau_{\text{FE}} := \frac{\Delta t}{2} \left(1 + \left(\frac{a\Delta t}{h} \right)^2 + 36 \left(\frac{\nu\Delta t}{h^2} \right)^2 \right)^{-\frac{1}{2}}. \quad (21)$$

Note that h can be interpreted as a measure of the radius of the support of the FE shape functions, equivalent to the dilation parameter ρ in EFG. So, when the EFG interpolation is considered the intrinsic time is computed with the same expression (21) just replacing the element size h by the dilation parameter ρ , that is,

$$\tau = \tau_{\text{EFG}} := \frac{\Delta t}{2} \left(1 + \left(\frac{a\Delta t}{\rho} \right)^2 + 36 \left(\frac{\nu\Delta t}{\rho^2} \right)^2 \right)^{-\frac{1}{2}}. \quad (22)$$

Once again, when particles are used, the consistent SUPG formulation in $\mathcal{H}^2(\Omega)$ (20) preserves the convergence rates of the time integration scheme (see Figure 6). On the other hand, when linear finite elements are used, results are degraded due to the neglected $\mathcal{O}(\tau)$ terms, see Figure 5. For, $\nu = 10^{-2}$ the intrinsic time τ is small enough so that the effect of this $\mathcal{O}(\tau)$ error is negligible in comparison with the truncation errors of the R_{11} time stepping scheme and almost negligible with the R_{22} scheme. However, for the convection dominated problem with $\nu = 10^{-4}$, the error $\mathcal{O}(\tau)$ reduces drastically the convergence rates: the error is of order $\mathcal{O}(\tau)$ when the truncation errors of the time stepping schemes are small enough.

5. NUMERICAL EXAMPLES

There are two important topics in the resolution of the transient convection-diffusion equation: (1) accurate transport of the unknown quantity u is necessary, and thus, high-order time stepping schemes are needed, and (2) in the presence of boundary or internal layers, it is necessary to stabilize the solution in order to avoid oscillations. Sections 3.3 and 4 already discussed how the smoothness of the EFG interpolation allows to easily define stabilized

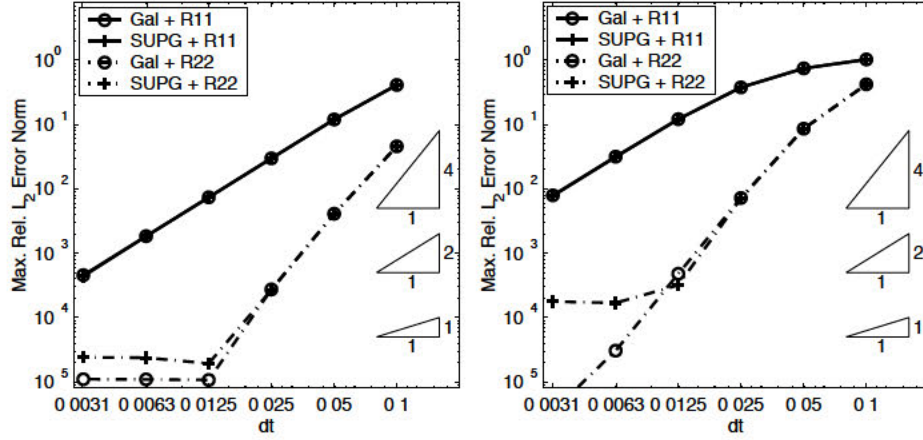


Figure 5. Finite elements convergence results with $h = 0.001$, $a = 1$, $\nu = 10^{-2}$ (left), and $\nu = 10^{-4}$ (right).

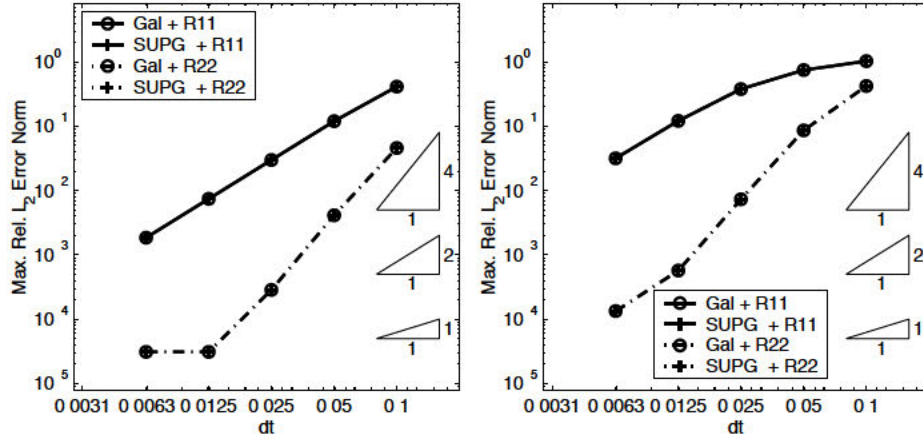


Figure 6. EFG convergence results with $h = 0.01$, $\rho/h = 3.2$, $\mathbf{P} = \{1, x\}^T$, $a = 1$, $\nu = 10^{-2}$ (left), and $\nu = 10^{-4}$ (right).

formulations in a consistent manner and preserve the convergence rates of the high-order time stepping schemes. In this section, some numerical examples are shown in order to see the influence of the different stabilization techniques and the time stepping schemes in the numerical solution with EFG. The first example is a 1D example whose solution tends to a stationary convection dominated solution. Therefore, the same problems of the stationary equation, that is boundary layers, are present in the transient solution and thus stabilized formulations are needed. The second example is a 2D example where both problems of the transient convection-diffusion equation are present: (1) accurate transport of the unknown is needed and (2) boundary layers appear in the solution due to the Dirichlet boundary conditions. Therefore, high-order time stepping schemes and stabilized formulations are needed

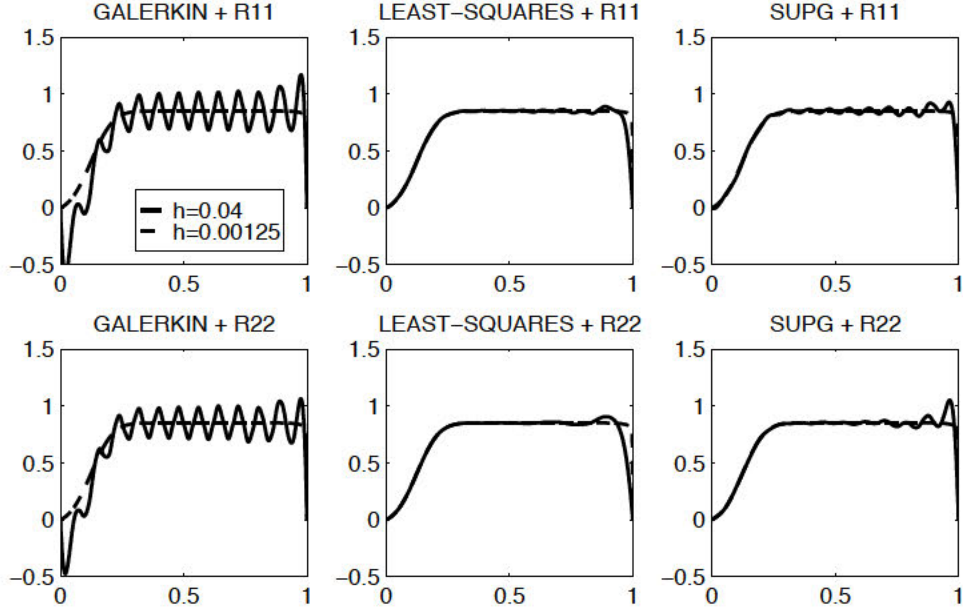


Figure 7. R_{11} and R_{22} solution at $t = 1$ for $a = 1$, $\nu = 10^{-3}$, $\rho/h = 3.2$ with $c = 1$ and $c = 3$ respectively

in order to obtain an accurate solution.

5.1. 1D example

The transient convection-diffusion equation, $u_t + au_x = \nu u_{xx} + s$, is solved in $[0, 1]$ with velocity $a = 1$, diffusion $\nu = 10^{-3}$, initial condition $u(x, 0) = 0$, homogeneous Dirichlet boundary conditions and source term $s(x) = 5e^{-(10(x-\frac{1}{8}))^2}$. The time step Δt is chosen such that the Courant number $c := a\Delta t/h$ is $c = 1$ for the R_{11} scheme and $c = 3$ for the R_{22} scheme, where h is the distance between particles.

Figure 7 shows the solution at $t = 1$ obtained with the R_{11} and the R_{22} time stepping schemes. The Galerkin solution suffers from the typical oscillations in the presence of boundary layers and stabilization techniques are necessary to improve the solution. As can be seen from the results, the diffusion added with the least-squares stabilization is larger than with the SUPG technique: it is easy to see that $\tau_{\text{SUPG}} \leq \tau_{\text{LS}} = \Delta t$ when formula (22) is used to compute τ_{SUPG} . In this example, the best result is obtained with the high-order time integration scheme R_{22} and the least-squares stabilization.

5.2. 2D example

The 2D problem described in Figure 8 is solved with small diffusion, $\nu = 10^{-5}$. The source term and the velocity field are defined as follows:

$$\mathbf{a} = (-y, x) \quad s = e^{-t^{10}} \begin{cases} \cos(\pi/2\sqrt{x^2 + y^2}) & \text{if } \sqrt{x^2 + y^2} \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

TIME ACCURATE STABILIZED MESH-FREE METHODS

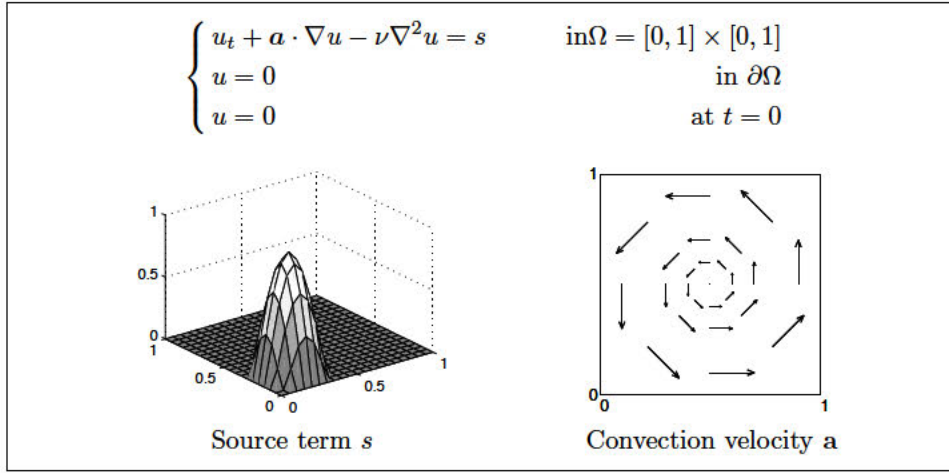


Figure 8. Problem statement for the 2D example

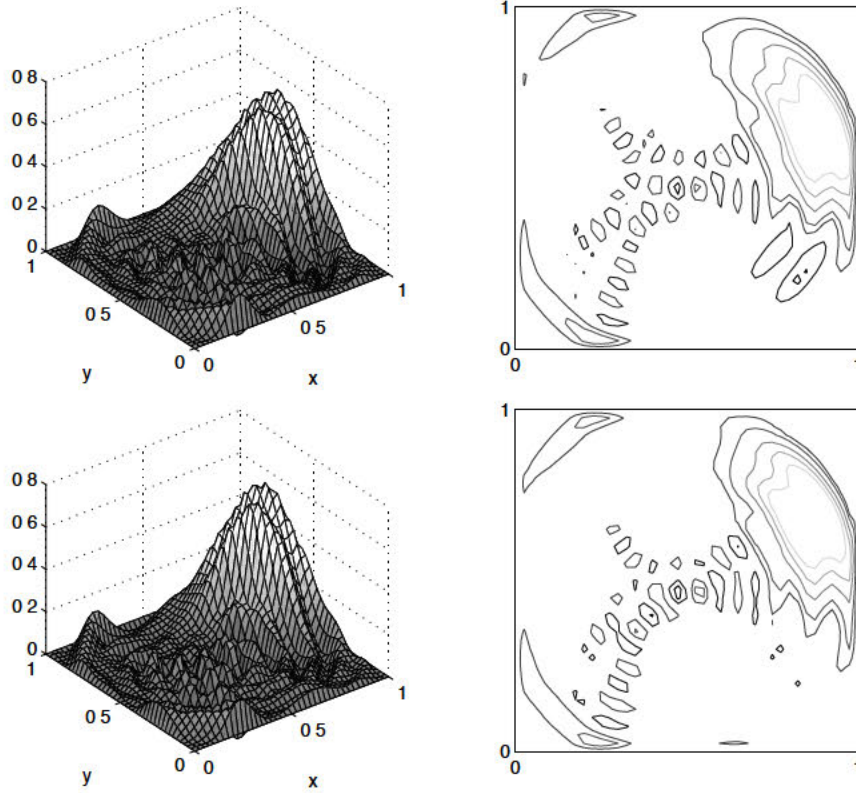


Figure 9. Galerkin solution and contour plot at $t = 15.9$ for $\nu = 10^{-5}$, $h = 0.05$, $\rho/h = 3.2$ with $c = 1$ for R_{11} and $c = 3$ for R_{22}

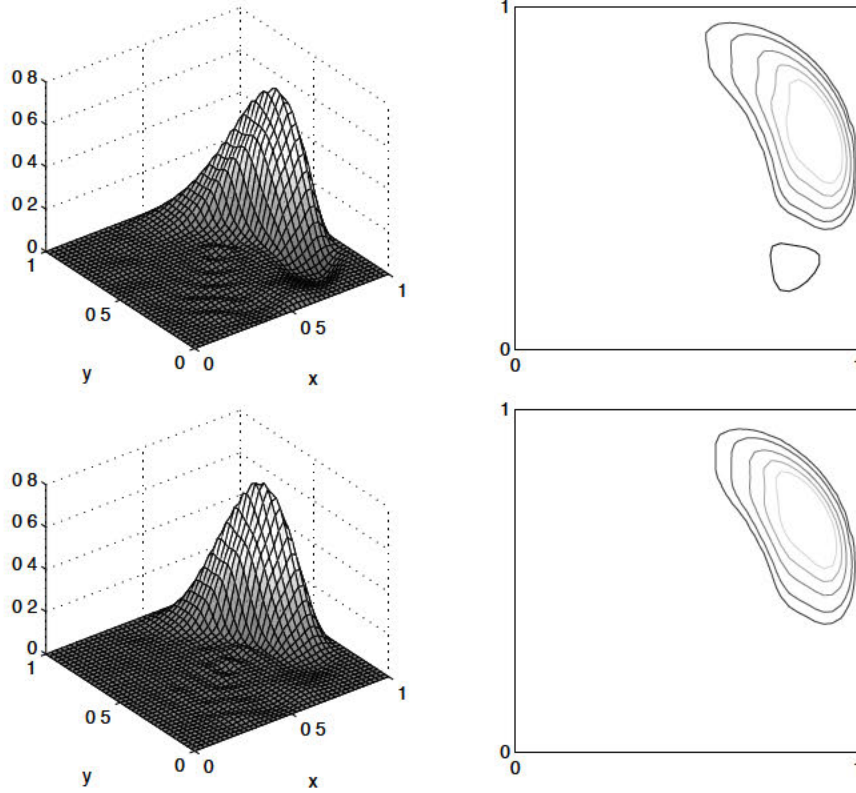
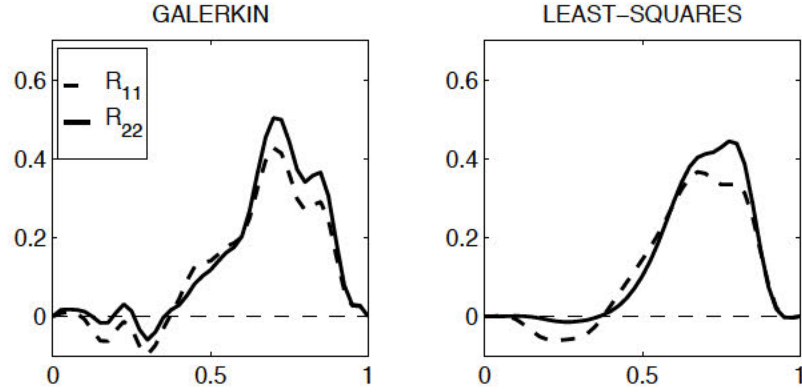
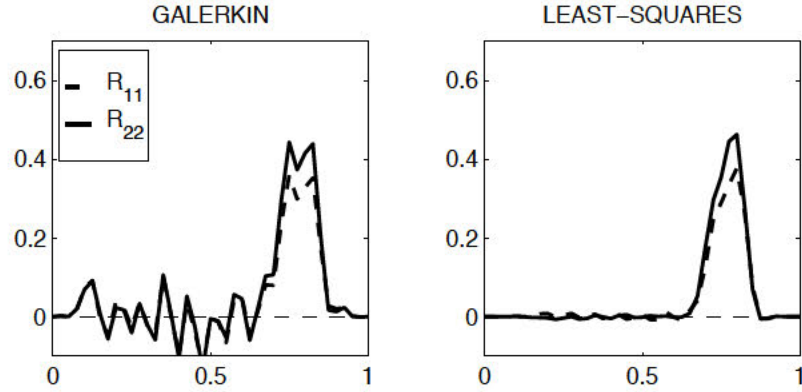


Figure 10. Least-squares solution and contour plot at $t = 15.9$ for $\nu = 10^{-5}$, $h = 0.05$, $\rho/h = 3.2$ with $c = 1$ for R_{11} and $c = 3$ for R_{22}

Figures 9 and 10 shows the numerical solution obtained at $t = 15.9$ for the Galerkin and the least-squares formulations, with a 21×21 distribution of particles, $\rho/h = 3.2$ and linear consistency. The R_{11} and R_{22} time integration schemes have been used with Courant 1 and 3 respectively. Different sections of the obtained numerical solutions at $t = 15.9$ are depicted in Figures 11 and 12.

Boundary layers are present in the solution due to the convective character of the equation and the homogeneous Dirichlet boundary conditions. Thus, the typical instabilities of the Galerkin formulation soon appear. The numerical solution is clearly improved for the stabilized least-squares formulation: oscillations are alleviated and almost suppressed in the whole domain. However, important phase and amplitude errors can be observed in the numerical solution obtained for the R_{11} time stepping scheme. In Figure 11, it is easy to see that the R_{11} solution presents negative values, with no physical sense. Obviously, the high-order R_{22} scheme provides a much better solution.


 Figure 11. Section along $x = 0.8$

 Figure 12. Section along $y = x$

6. CONCLUDING REMARKS

In this paper, the formulation proposed in [4] for transient convection-diffusion problems is used in the context of mesh-free methods. By performing the time discretization before the spatial one, standard stabilization techniques can be adapted to high-order time stepping schemes. However, with linear finite elements the lack of consistency due to neglected terms (terms with second order spatial derivatives) may degrade the convergence rates of the time integration scheme. In order to keep the high-order convergence rates the intrinsic time τ should be adapted to each time integration scheme or second derivatives should be globally reconstructed, with the corresponding increase in the computational cost.

This paper proposes to perform the spatial discretization using mesh-free interpolation functions. With the EFG method, second derivatives of the approximation can be computed with negligible extra computational cost. The mesh-free interpolation space is a subset of $\mathcal{H}^2(\Omega)$ and thus, the standard stabilization techniques can be particularized for the EFG

interpolation in $\mathcal{H}^2(\Omega)$. There are not neglected terms and the convergence rates of the high-order time integration schemes are preserved. Moreover, in EFG the order of the space discretization can be increased in a simple way. With a mesh-free interpolation, space and time high-order consistent stabilized formulations are easily defined and implemented.

REFERENCES

1. Donéa J, Quartapelle L, Selmin V. An analysis of time discretization in the finite element solution of hyperbolic problems. *J. Comput. Phys.* 1987; **70**:463–499.
2. Morton KW. *Numerical solution of convection-diffusion problems*. Applied Mathematics and Mathematical Computation, Editors Knops RJ, Morton KW, volume 12, Chapman & Hall 1996.
3. Donéa J, Roig B, Huerta A. High-order accurate time-stepping schemes for convection-diffusion problems. *Comp. Meths. Appl. Mech. Engr.* 2000; **182**:249–275.
4. A. Huerta, J. Donea. Time-accurate solution of stabilized convection-diffusion-reaction equations: I. Time and space discretization. *Comm. Numer. Methods Engrg.*, in press.
5. A. Huerta, B. Roig, J. Donea. Time-accurate solution of stabilized convection-diffusion-reaction equations: I. Accuracy analysis and examples. *Comm. Numer. Methods Engrg.*, in press.
6. T. Tezduyar, Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Comp. Meths. Appl. Mech. Engr.* 2000; **190**:411–430.
7. K. E. Jansen, S. S. Collins, C. Whiting, F. Shakib. A better consistency for low-order stabilized finite element methods. *Comp. Meths. Appl. Mech. Engr.* 1999; **174**:153–170.
8. T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, W. K. Liu. Smoothing and Accelerated Computations in the Element Free Galerkin Method. *Journal of Computational and Applied Mathematics* 1996; **74**:111–126.
9. Shakib F, Hughes TJR. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/Least-Squares algorithms. *Comp. Meths. Appl. Mech. Engr.* 1991; **87**:35–58.
10. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl. Meshless Methods: an Overview and Recent Developments. *Comp. Meths. Appl. Mech. Engr.* 1996; **139**:3–47.
11. W.K. Liu, T. Belytschko and J.T. Oden, eds. Meshless Methods. *Comp. Meths. Appl. Mech. Engr.* 1996; **139**.
12. W.K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belytschko, C. Pan, R.A. Uras, C.T. Chang. Overview and Applications of the Reproducing Kernel Particle Methods. *Archives of Computational Methods in Engineering, State of the Art Reviews* 1996; **3**:3–80.
13. W. K. Liu, S. Li, T. Belytschko. Moving least square reproducing kernel methods. (I) Methodology and convergence. *Comp. Meths. Appl. Mech. Engr.* 1997; **143**:113–154.
14. B. Nayroles, G. Touzot, P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
15. T. Belytschko, Y. Y. Lu, L. Gu. Element-free Galerkin methods. *Int. J. Numer. Meth. Engr.* 1994; **37**:229–256.
16. D. Organ, M. Fleming, T. Terry, T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics* 1996; **18**:225–235.
17. A. Huerta, S. Fernández-Méndez. Enrichment and coupling of the finite element and meshless methods. *Int. J. Numer. Meth. Engr.* 2000; **48**:1615–1636.
18. T. Hughes, *The Finite Element Method*, Prentice-Hall: New Jersey, 1987.
19. R. Codina. On stabilized finite elements methods for linear systems of convection-diffusion-reaction equations. *Comp. Meths. Appl. Mech. Engr.* 2000; **188**:61–82.
20. G. Carey, A. Pehlivanov, Y. Shen, A. Bose, K. Wang. Least-squares finite elements for fluid flow and transport. *Int. J. Numer. Meth. Fluids* 1998; **27**:97–107.