Chapter 4

Numerical tools

The elaboration of a continuum model has led us to develop a complex equation set (1.9.106). To solve it while taking into account all the problem specificities, the development of special and appropriate numerical tools is needed.

In this chapter we first investigate the so-called "bubble function" method, which is an original technique to stabilize the Finite Element Method (FEM) for advection-dominated problems. It will be seen that numerical results are of intermediate quality in 2D even if a computational cost study leads us to deeper investigate this method. Moreover, extension of the method to 3D problems turns out to be quite complicated. Therefore it can be concluded that the well-known SUPG method is a good selection to give some preliminary results. A particular issue of this chapter concerns the moving front problem, which has to be taken into account in order to solve the complete flow problem in a porous medium. Main alternatives are roughly exposed and an algorithm is suggested even it was not implemented.

The first section addresses the numerical difficulties to solve the complete problem (1.9.106). Different strategies are successively detailed and related results are discussed.

4.1 Numerical problem identification

Observing the system (1.9.106), it can be remarked that the primary variables are the Darcy velocity $\langle \underline{v}_f \rangle$, the modified pressure $\langle P_f \rangle^f$, the temperature $\langle T \rangle$ and the species concentration $\langle c_f \rangle^f$. First of all it appears that equations of this system are strongly coupled. This coupling results from the modelling of different physical phenomena :

- the transport phenomenon and the hydrodynamic dispersion effect. These effects depend on the velocity field, which is determined by solving the Darcy and mass conservation equations ; the velocity field is present in all equations and plays a key role in the energy and species conservation equations by influencing them in two ways (transport and dispersion) ;

- the viscosity dependence on temperature and species concentration is sometimes non-negligible. Therefore, the Darcy as well as the Forchheimer equations are influenced by the variations of these variables ;

- the chemical transformation of the fluid, taken into account by the kinetic function f_c, again includes dependence upon temperature and conversion degree.

The two last effects play a considerable role in the modelling of RTM and SRIM processes in view of the curing reaction experienced by the resin.

In addition to this coupling, individual equations are nonlinear. Indeed, the kinetic function brings nonlinearities in the temperature and species conservation equations. Moreover both the viscosity and the hydrodynamic dispersion effects generate a nonlinear equation behaviour.

Taking into account the continuous character of the problem, appropriate time and space discretizations must be chosen. Further, some mathematical specificities of the energy and species conservation equations will lead us to adapt the numerical discretization tools.

Finally, taking into account the moving front effect is required to model flows in unsaturated porous media (such as in RTM and SRIM processes).

4.2 Decoupled iterative strategy

In order to address the problem nonlinearity and coupled nature, a **decoupled iterative strategy** is developed. Decoupling means that each equation is individually solved for a related variable, considering that other variables are known. The global iterative system consists of all the individual solvers and manages them in an appropriate way. Fig. 4.1 shows an example of global solver. For example, in this case, the temperature and species solvers recieve the pressure and velocity fields calculated from the preceding step. The nonlinearities of the temperature and species conservation equations are treated by iterative methods. Individual solvers are details in the next sections.

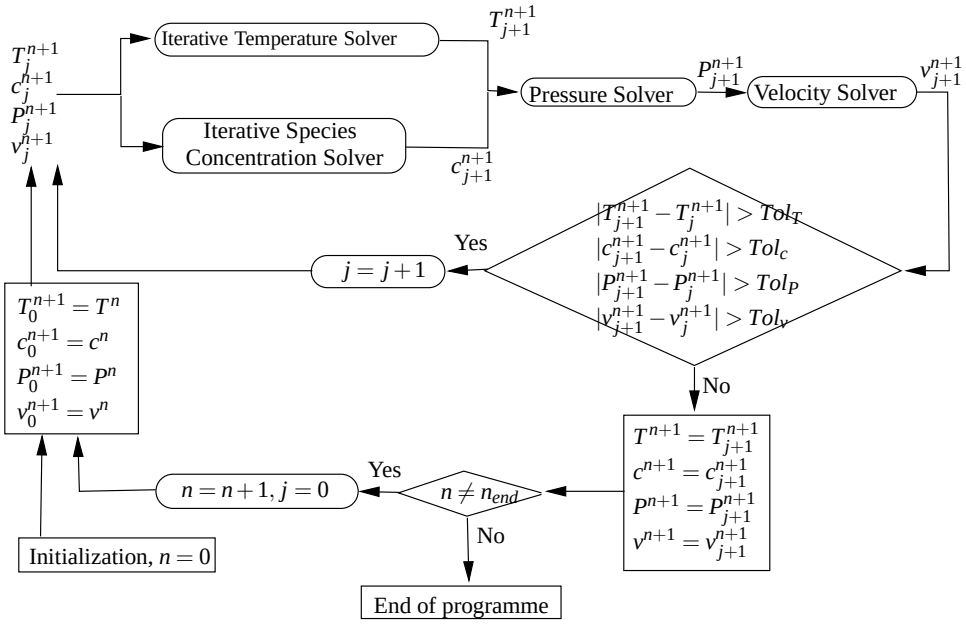


Figure 4.1: Global iterative algorithm. The entries follow the arrows. The superscripts refer to the time step while the subscripts refer to the iteration of the global scheme. The field variables with only a superscript represent the converged solution at the time associated with the superscript. Tol_* indicate the error tolerances relative to the searched fields.

4.3 Space and time discretization

The equations of our model (1.9.106) are constructed under the continuity assumption (Chapter 1). Solving this system implies using discretization tools, except in some particular cases where an analytical solution exists. Since both the unknowns and the equations are space and time dependent, different space and time discretization tools are generally used. The Taylor-Galerkin method for example firstly applies a Taylor development in order to make the time derivative discrete while the space discretization is then obtained by means of the Finite Element Method (FEM).

In this work, another scheme, the classical **semi-discrete** scheme, is investigated. The first step provides a system of ordinary differential equations (ODE's) and the second step consists in the integration of this system with respect to time. In our approach, space is discretized by FEM in the first step. This discretization gives then an ODE system whose unknowns are the nodal values which are themselves functions of time. Letting $\phi_i(x)$ represent the shape function associated with node i and $U_i(t)$ denote its nodal value, the approximate solution u_h is written as :

$$u_h = \sum_i \phi_i(x)\, U_i(t). \tag{4.3.1}$$

In the second step, time integration can be performed by classical methods such as the Explicit Euler (EE), Implicit Euler (IE) or Crank-Nicolson (CN) methods.

In order to assess the solution convergence, it must be kept in mind that space discretization produces a given ODE system. Space discretization is characterized by its characteristic size, noted h. The first step consists in ensuring the convergence of the ODE solution by decreasing the time step Δt associated with the time discretization (in order to reach the solution of the semi-discrete scheme for a given h). Once this is done, h can be decreased and convergence tests with time must again be performed (in order to reach the solution of the semi-discrete scheme for the new h). The cycle is then repeated until convergence.

In the next subsections, the FEM technique is briefly recalled as well as some classical time integration schemes. Related notations are also introduced.

4.3.1 The Finite Element Method (FEM) and related notations

Consider the boundary-value problem with Dirichlet conditions[1] formulated as follows :

$$\text{Find } u \in U \text{ such that : } \begin{cases} \mathscr{L}u = f & \text{on } \Omega \\ u = 0 & \text{on } \Gamma = \partial\Omega \end{cases} \tag{4.3.2}$$

where $\mathscr{L}$ is an operator associated with the original problem, U denotes the solution space, f is a "regular" source function, Ω is the problem open domain and Γ its boundary. This formulation is known as the **strong formulation** since the operator may contain high order derivatives. The variational approach allows us to reduce this order by means of integrations by part. This method consists in a projection of the equation on a well chosen functional space and leads to the problem **weak formulation** :

$$\text{Find } u \in U \text{ s.t. } \begin{cases} a(u,v) = (f,v), \quad \forall v \in V & \text{in } \Omega \\ u = 0 & \text{on } \Gamma = \partial\Omega \end{cases} \tag{4.3.3}$$

where V is the projection space, $(\cdot,\cdot)$ denotes the scalar product on U and where $a(u,v) = (\mathscr{L}u,v) = (u,\mathscr{L}^*v)$, $\mathscr{L}^*$ being the adjoint operator of $\mathscr{L}$.

In fact, the derivation order reduction results in an extension of U since less regular functions u can be introducted in Eq. (4.3.3) than in Eq. (4.3.2).

In order to apply the FEM technique, the domain Ω has firstly to be discretized in closed subdomains $e_k, k = 1,2,...,E < \infty$, such that their interiors are disjoint sets and their union forms the closure of the approximate domain $\Omega_h : \bigcup_{k=1}^{E} e_k = \bar{\Omega}_h$ (Fig. 4.2). These subdomains are defined by a set of so called **nodes** $p_i, i \in P$, which is essentially composed of two types of nodes, viz. the interior nodes and the boundary nodes (denoted by P° and $\underline{P}$ respectively, with $P = P^\circ \cup \underline{P}$). The FEM technique is deeply related with the problem weak formulation and consists in discretizing the system by selecting subsets of the functional spaces associated with the unknown functions and the test-functions : $U_h \subseteq U$ and $V_h \subseteq V$. Restricting these functions on e_k, they are finally determined by the related nodal values on the associated elements (interpolation). Accordingly, following A. Magnus (34), the *FEM is a "trialogue" between element geometry, function spaces and interpolation.*

The solution is then approximated by a function $u_h \approx u$ which is decomposed in a sum of functions defined on the neighbouring elements :

$$u_h = \sum_{i \in P} \phi_i(\underline{x})\, U_i, \tag{4.3.4}$$

[1] Only Dirichlet boundary conditions are here considered by sake of simplicity.

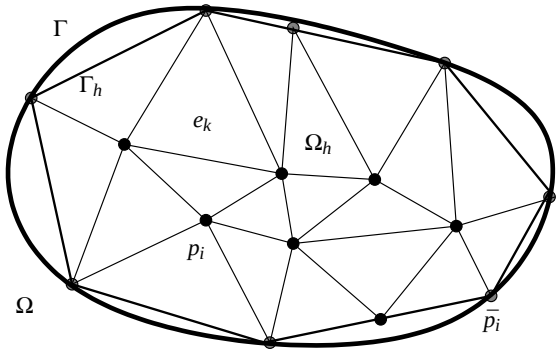


Figure 4.2: Approximation of the domain Ω by Ω_h and example of discretization in elements e_k. Here $\overline{p_i}$ are the nodes belonging to the boundary Γ_h, (with i in $\overline{P}$), while p_i are the nodes belonging to the domain interior with in this case $i \in P^\circ$.

where U_i denotes the value of function u_h at node p_i and is called the **nodal value** of u_h.

This decomposition (4.3.4) is further introduced into the operator defined by (4.3.3) and is projected on a sufficient number of test-functions by discretizing the projection space V in the same way as U. For example, the **Galerkin** method consists in choosing the discrete projection space roughly equal to the discrete unknown space. The weak formulation is then written as :

$$\text{Find } U_i,\ i \in P^\circ \text{ s.t. } \begin{cases} a(\sum_{i\in P} U_i\phi_i, \phi_j) = (f, \phi_j), & \forall \phi_j,\ j \in P^\circ \\ U_i = 0 & i \in \overline{P}. \end{cases} \tag{4.3.5}$$

If the form $a(\cdot,\cdot)$ is bilinear, this formulation provides a linear system in which the unknowns are the nodal values U_i. The FEM method will be deeper investigated in the next sections.

Of course other space discretization techniques exist, such as provided by the Finite Difference Method (FDM) or the Finite Volume Method (FVM). On the one hand, a complex geometry is not easily treated by FDM contrarily to FEM because of the difficulties related to structured mesh management. On the other hand, FEM tools have already been developed in the Applied Mechanics Division of UCL (MEMA) for some years and collected in the C++ library `NumLab`. Consequently, even if FVM also exhibits the domain adaptability quality, FEM has been selected to perform space discretization.

4.3.2 Some time integration schemes

Since space discretization provides a set of equations where the unknowns depend on time, let us consider the simple generic problem :

$$\frac{du}{dt} = f(u,t), \quad \text{with} \quad u(0) = u_0 \, . \tag{4.3.6}$$

Classical integration schemes, with their numerical stability and precision properties are here recalled :

Explicit Euler (EE) scheme :

$$\frac{u_{n+1} - u_n}{\Delta t} = f_n \tag{4.3.7}$$

This scheme is conditionally stable and the global solution accuracy is $\mathscr{O}(\Delta t)$.

Implicit Euler (IE) scheme :

$$\frac{u_{n+1} - u_n}{\Delta t} = f_{n+1} \tag{4.3.8}$$

This scheme is unconditionally stable and the global solution accuracy is $\mathscr{O}(\Delta t)$.

Crank-Nicolson (CN) scheme :

$$\frac{u_{n+1} - u_n}{\Delta t} = \frac{f_{n+1} + f_n}{2} \tag{4.3.9}$$

This scheme is unconditionally stable and the global solution accuracy is $\mathscr{O}\left(\Delta t^2\right)$.

In the above equations, $f_n = f(u_n, t_n)$ and u_n is the approximation of $u(t_n)$. The Crank-Nicolson method is implicit. Nevertheless, if $f(u,t)$ is linear in u, this scheme turns out to become closer to explicit and is then very simple to implement. In the next subsection, the stability concept is deeper investigated.

4.3.3 Numerical stability and coercivity

Before handling the advection-diffusion problem, it is important to address the stability concept[2] and its link with coercivity. First of all, let us remark that physical and

[2] Numerical stability must here be understood in the sense of absence of oscillations in the stationary state. We have used this vocabulary in order to remain consistent with the most important literature dealing with this topic. Otherwise, the stability concept should be related to the time evolution of the solution.

numerical stability must be considered as different concepts. Indeed, as an example, a stable numerical method can be applied to an unstable physical problem. The notion of physical stability means that small modifications of the problem inputs produce only slight variations of the solution :

$$\|\delta f\| < \varepsilon \Rightarrow \|\delta u\| < \mathscr{O}(\varepsilon), \tag{4.3.10}$$

refering to the strong problem formulation (4.3.2).

Numerical stability is related to the model governing equations and the selected space-time discretization. In fact, for each problem and associated numerical method, there exists a link between the eigenvalues of the problem operator $\mathscr{L}$ and the discretization refinement as characterized by the parameters Δt and h (typically the CFL condition for a pure convection problem).

In this work, focusing on a weak formulation approach, numerical stability depends on several factors, such as the selected approximation space and projection operator discretization.

Let us now assume that $a(\cdot,\cdot)$ is a bilinear symmetric and coercive operator. Coercivity means that :

$$\exists c > 0 \text{ such that } \forall u \in V, \quad a(u,u) \geq c\|u\|^2. \tag{4.3.11}$$

It is possible to show that **the more the operator is coercive, the more it is stable**. In Eq. (4.3.3), taking $U = V$ and letting u and $u + \delta u$ be the solutions with right-hand sides f and $f + \delta f$ respectively, from the bilinearity property, it follows that $a(\delta u, v) = (\delta f, v) \ \ \forall v \in V$. Taking $v = \delta u$, by coercivity and scalar product inequality, it can be deduced that :

$$c\|\delta u\|^2 \leq a(\delta u, \delta u) = (\delta f, \delta u) \leq \|\delta f\| \ \|\delta u\|. \tag{4.3.12}$$

The stability condition 4.3.10 is then easily recovered. Hence a stabilization technique consists in increasing the operator coercivity. This strategy is developed in the next section and application to the advection-dominated diffusion problem is further carried out.

4.4 Stabilized methods

In order to implement temperature and species concentration solvers, it is necessary to highlight the mathematical and numerical properties of the related equations, Eqs.

(1.6.94) and (1.7.103). The energy equation (1.6.94) is written as

$$\begin{aligned}&(\varepsilon_f \rho_f C_f + \varepsilon_s \rho_f C_s)\tfrac{\partial \langle T\rangle}{\partial t} + \rho_f C_f \langle \underline{v}_f\rangle \cdot \underline{\nabla}\langle T\rangle = \\ &\underline{\nabla}\cdot\left[(\underline{k_e} + \underline{K}_D)\,\underline{\nabla}\langle T\rangle\right] + \rho_f H_R f_c(\langle c_f\rangle^f, \langle T\rangle) + \eta\ \langle \underline{v}_f\rangle \cdot \underline{K}^{-1}\cdot\langle \underline{v}_f\rangle\ .\end{aligned} \tag{4.4.13}$$

Whereas this equation is mathematically parabolic (due to the presence of a first-order time derivative and of a second-order space partial derivatives), it exhibits a hyperbolic as well as a parabolic numerical character through its convection and conduction terms respectively. In the case of material forming processes such as RTM or SRIM, transport effects are very important and could even dominate the conduction terms. Numerically, this means that the problem pertains to the hyperbolic-parabolic type while being hyperbolicity dominated. It is well known that the Galerkin FEM is unstable to solve this class of problems. Hence, strategies enhancing the FEM stability have been and remain nowadays investigated in the scientific community.

In order to address this issue, the existing stabilized methods are shortly reviewed. So-called "Bubble Function" strategies are presented with a particular attention paid to the Residual Free Bubble (RFB) method. The link between the stabilized, multiscale and bubble function approaches is then highlighted.

4.4.1 Stabilized method and classical tools

As shown previously, the principle of a stabilized method lies in enhancing the coercivity constant of a classical method (for example the Galerkin method). Nevertheless, in order to keep the method **consistency**, the added term must depend on the residual of the strong formulation, $\mathscr{L}u - f$. Hence, problem (4.3.3) is rewritten as (35) :

Find $u_h \in U_h$ s.t.

$$\begin{cases} a(u_h, v_h) + (\tau(\mathscr{L}u_h - f), \mathbb{L}v_h) = (f, v_h), \quad \forall v_h \in V_h & \text{in } \Omega \\ u_h = 0 & \text{on } \Gamma = \partial\Omega, \end{cases} \tag{4.4.14}$$

where $\mathbb{L}$ is usually a differential operator and τ an algebraic operator weighting the residual effect. The determination of the value of this last parameter as well as a justification of the method have thoroughly been investigated and the conclusions were finalized by Hughes (35) whose paper also suggests some next developments.

Three classical stabilized method are listed herebelow :

- $\mathbb{L} = \mathscr{L}$: Galerkin Least Square (GLS) method (36),

- $\mathbb{L} = \mathscr{L}_{adv}$: Streamline Upwinding Petrov-Galerkin (SUPG) method (37), where $\mathscr{L}_{adv}(\cdot) = \underline{w} \cdot \nabla(\cdot)$.

- $\mathbb{L} = -\mathscr{L}^*$: an original method developed by L.P. Franca in (38).

This latter author demonstrates how these methods improve the coercivity of the operator $a(\cdot,\cdot)$ for the advection-diffusion case (38).

The next subsection addresses the bubble function approach, whose master idea is basically different. After some developments it is demonstrated that this method can be joined to the stabilized methods.

4.4.2 Bubble function strategy

The general bubble function approach will firstly be presented, before developing the specific case of Residual Free Bubbles (RFB).

General approach

Solving PDE's by FEM technique will involve the solution of linear systems. When the discretization gives a large number of unknowns, the solutions becomes computationally expensive. Hence, refining the mesh in order to reach a more accurate solution has the drawback to increase the system dimension and the solution cost.

The bubble function approach consists in extending the solution approximation set without increasing significantly the system size. For this purpose, local functions are added to the original approximation set. Each of these functions is related to a given element and vanishes everywhere else, except on its own element, and hence is called a **bubble** function. The technique modifies the global system by incorporating local effects by this way. Finally this modification is introduced step-by-step in the local matrices in such a way that the final system size is not affected. This strategy is called **static condensation** and is illustrated herebelow.

Let us first consider the space discretization $\mathscr{T}_h$ and the approximation set U and add to this set the not yet defined bubble function set $\mathscr{B}$, which is disjoint from U in order

to provide a new approximation set H :

$$H = U \oplus \mathscr{B} \quad \text{and} \quad \forall u_H \in H, \text{ there exists a single decomposition } u_H = u_U + u_{\mathscr{B}}. \tag{4.4.15}$$

The set $\mathscr{B}$ collects specific bubble function sets associated with each element :

$$\mathscr{B} = \oplus B_T \text{ where } B_T \subset \mathscr{H}_0^1(T),\ T \in \mathscr{T}_h, \tag{4.4.16}$$

where $\mathscr{H}_0^1(T)$ is the Sobolev space of order 1 consisting of functions which vanish on the boundary of element T. Applying the Galerkin method to the weak formulation (4.3.3) with the approximation and projection sets equal to H gives :

$$\begin{cases} a(u_H, v_H) = (f, v_H),\ \forall v_H \in H, & \text{where} \quad u_H = u_V + u_{\mathscr{B}} \text{ and } v_H = v_V + v_{\mathscr{B}} \\ u_H = 0 & \text{on } \Gamma, \end{cases} \tag{4.4.17}$$

In view of the bubble function properties, their influence is purely local on their related element. Hence, **static condensation** can be performed :

$$a(u_{\mathscr{B}}, v_{\mathscr{B}}) = (f, v_{\mathscr{B}}) - a(u_V, v_{\mathscr{B}}) = (f, v_{\mathscr{B}}) - (\mathscr{L} u_V, v_{\mathscr{B}}) = (z, v_{\mathscr{B}}), \tag{4.4.18}$$

where $z = f - \mathscr{L} u_V$. Scalar product properties allow us to further write this equation as :

$$a(u_{\mathscr{B}}, v_{\mathscr{B}}) = (z, v_{\mathscr{B}}) = (P_{\mathscr{B}} z, v_{\mathscr{B}}), \tag{4.4.19}$$

where $P_{\mathscr{B}}$ is a projection operator on the space $\mathscr{B}$. Problem (4.4.19) can actually be seen as a variational formulation on the space $\mathscr{B}$. Let us define $S_{\mathscr{B}}$ as the operator generating the solution to this problem for a given z :

$$u_{\mathscr{B}} = S_{\mathscr{B}}(P_{\mathscr{B}} z). \tag{4.4.20}$$

Let us now consider any $v_V \in V$ and apply the residual projection onto v_V, taking into account the value obtained for $u_{\mathscr{B}}$. The following equations are successively found :

$$\begin{array}{rcl} a(u_V, v_V) + a(u_{\mathscr{B}}, v_V) & = & (f, v_V) \\ a(u_V, v_V) + (u_{\mathscr{B}}, \mathscr{L}^* v_V) & = & (f, v_V) \\ a(u_V, v_V) + (S_{\mathscr{B}} P_{\mathscr{B}}(-z), -\mathscr{L}^* v_V) & = & (f, v_V) \\ a(u_V, v_V) + (S_{\mathscr{B}} P_{\mathscr{B}}(\mathscr{L} u_V - f), -\mathscr{L}^* v_V) & = & (f, v_V)\ \ \forall v_V \in V\ . \end{array} \tag{4.4.21}$$

Analyzing the formulations (4.3.3) and (4.4.21), it appears that extending the approximation set and performing static condensation is equivalent to adding a perturbation term to the l.h.s. of the Galerkin method. This term writes as follows :

$$(S_{\mathscr{B}} P_{\mathscr{B}}(\mathscr{L} u_V - f), -\mathscr{L}^* v_V). \tag{4.4.22}$$

If the stabilizing perturbations were known, the question would be to know if it is possible to produce them with a given bubble function set, or further to generate a

bubble function set generating these stabilizing perturbations. The response to these questions is partially given in (39). The authors indeed show that, for λ belonging to the interval $[0,\lambda_0]$ where λ_0 is the smallest eigenvalue of the operator $S_{\mathscr{B}}$, there exists a bubble function set such that $S_{\mathscr{B}}P_{\mathscr{B}} = \lambda I$, where I is the identity operator. The construction of this set is described in the paper.

It also appears that Eqs. (4.4.14) and (4.4.21) become quite similar if $S_{\mathscr{B}}P_{\mathscr{B}} = \tau I$. The link between the stabilized methods (4.4.14) and the bubble function approach (4.4.14) is then clearly established, provided $\tau \leq \lambda_0$.

This remark closes the general bubble function approach. Let us now turn our attention to Residual Free Bubbles, which are in fact a specific class of shape functions.

Residual Free Bubbles

This method will be useful to determine a good bubble function set. Its principle is very attractive since it is based on the cancellation of the residual of the problem strong formulation inside each element. In other words, bubble functions are constructed such that the problem is exactly solved inside each element :

$$\mathscr{L}u_{\mathscr{B}} = -(\mathscr{L}u_V - f), \text{ in int}T, \ \forall T \in \mathscr{T}_h. \tag{4.4.23}$$

As bubble functions vanish on the element boundaries, they do not correct the approximation there. The final solution still remains an approximation of the exact solution, except if the approximation set U contains the exact interpolation of the solution.

However, even if the above principle is basic and elegant, the solution is generally difficult to find except again in some specific cases (40). In order to avoid these complications, the bubble function space can be approximated. This approach has been investigated by L.P. Franca et al. in (41), where the strong formulation of the residual cancellation (4.4.23) is replaced by a weak formulation which is solved by application of FEM technique. In other words, this is a kind of subgrid scale approach as shown in the following lines.

Let us consider a linear operator $\mathscr{L}$, together with the shape functions of element T, noted ψ_i^T, and the number of shape functions per element, noted n_ψ. The bubble functions are decomposed as : $u_{\mathscr{B}} = \sum_{1\leq i\leq n_\psi} \alpha_i\phi_i + \phi_f$, where the functions ϕ_i and ϕ_f are determined by the following equations :

$$\begin{cases} \mathscr{L}\phi_i = -\mathscr{L}\psi_i^T & \text{in } T \\ \phi_i = 0 & \text{on } \partial T \end{cases} \tag{4.4.24}$$

and

$$\begin{cases} \mathscr{L}\phi_f = f & \text{in } K \\ \phi_f = 0 & \text{on } \partial K \end{cases} \tag{4.4.25}$$

Accordingly, if $u_V = \sum_{1\leq i\leq n_\psi} c_i \psi_i^T$ and $u_{\mathscr{B}} = \sum_{1\leq i\leq n_\psi} \alpha_i \phi_i + \phi_f$, from the selection $\alpha_i = c_i$, it can be deduced that the residual inside each element is cancelled since $\mathscr{L}u_{\mathscr{B}} = -(\mathscr{L}u_V - f)$. In order to solve problems (4.4.24) and (4.4.25), their weak formulation can be solved by FEM technique. However, the Galerkin method might still remain unstable for these local problems and stabilized methods should then be used (SUPG, GLS, ...). Moreover, in this case, the bubble function obtained will only be approximations of the ideal bubble functions.

Therefore, applying this approach to a linear operator will give an approximation of the RFB method. However, on the one hand it might seem not completely satisfactory to use known stabilized methods to develop new techniques, while on the other hand the restriction to linear operators is in most cases too strong.

In conclusion, it appears that the bubble function method is very attractive in its principles. However, the construction of a good bubble function set is usually a difficult task. A procedure is then to approximate this set by FEM technique. Studying bubble functions besides provides some understanding of the stabilization origin. The final formulation after static condensation can be compared with known stabilized methods and hence can lead to the elaboration of new stable discretization tools for given operators. Finally, noting that a multiscale approach has been developed by Hughes in the bubble function context (35), it can be pointed out that the ideas of this strategy are very important in the theory of the stabilized methods. This is the object of the next subsection.

4.4.3 Multiscale approach

Space discretization implies that scales smaller than the mesh size are not taken into account in the discrete system. Discretization then acts like a filter after which only large scales are addressed while subgrid scales are erased. Small scales, also called microscopic scales, pertain to the unresolvable solution part, while the macroscopic scales are captured by discretization. However microscopic scales may have an important influence on the real solution. In that case, they must been taken into account and their presence must be included in the formulation. Hence, the central idea of the multiscale approach lies in introducing microscopic effects in the large scale scheme. However, a strong assumption of this theory is that subgrid scales identically vanish on element boundaries. This assumption has the effect of localizing the calculations of subgrid scales in the sense that these problems are uncoupled from element to element.

Our goal is here to develop an adequate formulation of these principles, starting from the strong formulation (4.3.2) and its weak version (4.3.3).

Unknown and test functions are decomposed in two parts, corresponding to large ($\bar{u}$) and small scales (u') respectively :

$$\left\{ \begin{array}{l} u = \bar{u} + u' \\ v = \bar{v} + v' \end{array} \right. \tag{4.4.26}$$

The weak problem formulation can then be rewritten as :

$$a(\bar{u}+u', \bar{v}+v') = (f, \bar{v}+v') \; . \tag{4.4.27}$$

In case of a **linear** operator, this problem is the same as :

$$\begin{array}{lrcl} (1) & a(\bar{u},\bar{v}) + a(u',\bar{v}) & = & (f,\bar{v}) \\ (2) & a(\bar{u},v') + a(u',v') & = & (f,v') \; . \end{array} \tag{4.4.28}$$

The strong formulation of the second equation is then :

$$\left\{ \begin{array}{lcll} \mathscr{L}u' & = & -(\mathscr{L}\bar{u} - f) & \text{in } T \\ u' & = & 0 & \text{on } \partial T, \; \forall T \in \mathscr{T}_h \end{array} \right. \tag{4.4.29}$$

These equations induce localization of the subgrid scales in order to satisfy the assumption of their element to element independence. Hence, the problems can be solved independently on each element, for example by use of the Green's function g associated to problem (4.4.29) :

$$\left\{ \begin{array}{lcll} \mathscr{L}\mathrm{g} & = & \delta & \text{in } T \\ \mathrm{g} & = & 0 & \text{sur } \partial T, \; \forall T \in \mathscr{T}_h. \end{array} \right. \tag{4.4.30}$$

The solution of Eq. (4.4.29) is then :

$$u'(y) = -\int_T \mathrm{g}(x,y)(\mathscr{L}\bar{u} - f)(x), \tag{4.4.31}$$

or, defining the operator M_T by comparison with Eq. (4.4.31),

$$u' = M_T(\mathscr{L}\bar{u} - f). \tag{4.4.32}$$

Finally, the large scale part of Eq. (4.4.28) transforms into :

$$\begin{array}{rcl} a(\bar{u},\bar{v}) + a(u',\bar{v}) & = & a(\bar{u},\bar{v}) + (u', \mathscr{L}^*\bar{v}) \\ & = & a(\bar{u},\bar{v}) + (-M_T(\mathscr{L}\bar{u} - f), -\mathscr{L}^*\bar{v}) \quad = \quad (f,\bar{v}). \end{array} \tag{4.4.33}$$

This formulation can be considered as Galerkin formulation modified by a term including microscopic effects.

4.5 Link between stabilized, bubble function and multiscale approaches

This section draws a link between the previous stabilized strategies. Let us now summarize their respective effects and analyze the case where these effects are similar. A crucial point consists in determining the parameter of the SUPG method, thereby showing that this parameter has a non-arbitrary value in order to maximize the related stabilisation effect.

The relationship between stabilized methods, bubble function techniques and the multiscale approach becomes obvious by comparing the respective strategies summarized in Eq. (4.4.14), Eq. (4.4.21) and Eq. (4.4.33). These equations become similar if their related perturbation terms are equal :

$$(\tau_T(\mathscr{L}u_h - f), \mathbb{L}v_h) = (S_{\mathscr{B}}P_{\mathscr{B}}(\mathscr{L}u_V - f), -\mathscr{L}^* v_V) = (-M_T(\mathscr{L}\bar{u} - f), -\mathscr{L}^*\bar{v}) \tag{4.5.34}$$

where $u_h = u_V = \bar{u}$ and $v_h = v_V = \bar{v}$.

It is then possible to evaluate the parameter τ_T with a well-established theoretical background. Indeed, comparing the third formulation of the stabilized methods Eq. (4.4.14) with the multiscale approach provides the relation :

$$\tau_T(\mathscr{L}u_h - f) = -M_T(\mathscr{L}u_h - f). \tag{4.5.35}$$

With the definition of M_T given by Eq. (4.4.31) and Eq. (4.4.32), it follows that :

$$\tau_T \delta(y - x) = \mathrm{g}(x,y), \tag{4.5.36}$$

and hence τ_T can be evaluated from the equation

$$\int_T \int_T \tau_T \delta(y - x) dT_x dT_y = \int_T \int_T \mathrm{g}(x,y) dT_x dT_y \tag{4.5.37}$$

as :

$$\tau_T = \frac{1}{|T|} \int_T \int_T \mathrm{g}(x,y) dT_x dT_y. \tag{4.5.38}$$

Therefore, it appears that τ has a non-arbitrary value.

In conclusion, the link between stabilized methods and the multiscale approach now provides a justification to the selection of the SUPG parameter (which was up to now qualified as *magic*). These results could be considered as concluding the investigation of stabilized methods. The aim of the next section is to illustrate these conclusions in the 2D case. Typically, the bubble function strategy will be shown to be potentially

similar to the SUPG strategy and it will be explained how this method can help in determining new stabilized methods.

This last section has mainly been inspired from the outstanding work of T.J.R. Hughes in (35). The above theories will now be applied to the advection-diffusion operator.

4.6 Application : the advection-diffusion operator

The 2D case is mainly treated in this part by successively developing the SUPG and bubble function strategies. The relation with the multiscale approach is illustrated and conclusions are drawn.

Consider a 2D euclidean space Ω, its boundary Γ and a regular triangulation $\mathscr{T}_h$. Focusing on the advection-diffusion operator, which is written as

$$\mathscr{L}(\cdot) = -\underline{\nabla}\cdot(\kappa\underline{\nabla}(\cdot)) + \underline{w}\cdot\underline{\nabla}(\cdot)\ , \tag{4.6.39}$$

the related Dirichlet problem writes as :

$$\left\{\begin{array}{ll} \mathscr{L}u = f & \text{in } \Omega \\ u = 0 & \text{on } \Gamma = \partial\Omega\ . \end{array}\right. \tag{4.6.40}$$

This is the problem strong formulation, analogous to Eq. (4.3.2). The diffusion coefficient $\kappa = Cst > 0$ is assumed to be constant while velocity and source fields $\underline{w}$ and f are constant on each element $T \in \mathscr{T}_h$.

The variational approach provides the weak formulation of this problem, which consists in finding $u \in \mathscr{H}_0^1(\Omega)$ such that :

$$a(u,v) = (f,v),\ \forall v \in \mathscr{H}_0^1(\Omega) \tag{4.6.41}$$

where

$$\begin{array}{lcl} a(u,v) & = & \int_\Omega \kappa\underline{\nabla}u\cdot\underline{\nabla}v + (\underline{w}\cdot\underline{\nabla}u)v \\ (f,v) & = & \int_\Omega fv\ . \end{array} \tag{4.6.42}$$

Firstly, it is interesting to demonstrate that this problem has a unique solution. As a Hilbert space is here considered, the Lax-Milgram theorem can be applied. Hence, the assertion is verified if the operator $a(\cdot,\cdot)$ is bilinear and coercive on $\mathscr{H}_0^1(\Omega)$. The first property is easily verified. In order to establish coercivity, let us recall that the form

$a(u,v) = \int_\Omega \underline{\nabla} u \cdot \underline{\nabla} v$ is coercive. Then, assuming that the velocity field $\underline{w}$ is such that $\underline{\nabla} \cdot a \leq 0$[3] and by application of Green's theorem, we find :

$$\int_\Omega \underline{\nabla} \cdot (\underline{w} u u) = \underbrace{\int_{\partial\Omega} u^2 \underline{w} \cdot \underline{n}}_{=0} = \underbrace{\int_\Omega \underline{\nabla} \cdot \underline{w}\, u^2}_{\leq 0} + \int_\Omega 2(\underline{w} \cdot \underline{\nabla} u) u, \ \forall u \in \mathscr{H}_0^1(\Omega) \ . \quad (4.6.43)$$

This equation then provides the inequality :

$$\int_\Omega (\underline{w} \cdot \underline{\nabla} u) u \geq 0, \ \forall u \in \mathscr{H}_0^1(\Omega), \quad (4.6.44)$$

and hence the coercivity of the total operator (4.6.42) is established. Therefore, problem (4.6.41) has a unique solution.

4.6.1 SUPG method

In some sense the SUPG approach consists in taking into account the information provided by the flow direction. This is achieved by modifying the test-functions of the Galerkin method in a non-conformal way such that the flow direction is introduced in the new formulation. The SUPG method holds its name from this non-conformal way modification of the test-functions is performed ("Streamline Upwinding") and from the modification technique which conserves consistency and is hence called "Petrov-Galerkin" (PG). An example of functions fulfilling these specificities is given by :

$$v_h + \tau_T^{supg} \underline{w} \cdot \underline{\nabla} v_h, \quad (4.6.45)$$

where τ_T^{supg} is a not yet defined positive factor have the physical dimension of a time constant. The original method, applied with the approximation set V_h which is a space

[3] In our case, we have even $\underline{\nabla} \cdot \underline{w} = 0$ on each element as the velocity field is element-constant.

consisting of continuous and piecewise polynomial functions, is written as (42) :

$$\left\{\begin{array}{l} \text{Find } u_h \in V_h \text{ s.t.} \\ a(u_h,v_h)_{supg} = \int_\Omega \kappa \underline{\nabla} u_h \cdot \underline{\nabla} v_h + (\underline{w}\cdot\underline{\nabla} u_h) v_h + \\ \sum_{T\in\mathscr{T}_h} \tau_T^{supg} \int_T ((\underline{w}\cdot\underline{\nabla} u_h - \underline{\nabla}\cdot(\kappa\underline{\nabla} u_h) - f)\ \underline{w}\cdot\underline{\nabla} v_h \\ = \int_\Omega f v_h,\ \forall v_h \in V_h \\ \text{with} \left\{\begin{array}{l} \tau_T^{supg}(Pe_T) = \frac{h_T}{2\|\underline{w}\|}\xi(Pe_T), \\ Pe_T = \frac{m_k \|\underline{w}\|_p h_T}{2\kappa}, \\ \xi(Pe_T) = \left\{\begin{array}{ll} Pe_T, & 0 \le Pe_T < 1, \\ 1, & Pe_T \ge 1, \end{array}\right. \\ m_k = \min\left\{1/3, 2\tilde{C}_k\right\}, \\ \tilde{C}_k \sum_{T\in\mathscr{T}_h} h_T^2 \|\Delta v\|_{0,T}^2 \le \|\underline{\nabla} v\|_0^2,\ v \in V_h\ . \end{array}\right. \end{array}\right. \tag{4.6.46}$$

In the above, Pe_T is a numerical Peclet number, related to the mesh size and the approximation set. It represents the ratio between convection and diffusion terms and takes into account the approximation set degree. Indeed, the more refined the mesh ($h_T \searrow$) and the larger the approximation set degree ($m_k \searrow$), the higher the numerical accuracy will be. Indeed, when the Peclet number decreases, the test-function modification will be reduced. In this case, mesh and approximation set properties are quite sufficient to capture the transport phenomenon.

The consistency of SUPG is obvious when considering that the added term depends on the strong formulation residual. Where the Galerkin method is known to be unstable for the advection dominated diffusion problem, Brezzi et al. have proved in (42) that the modifying term of Eq. (4.6.46) increases the formulation coercivity and therefore renders more stable the operator $a_{supg}(\cdot,\cdot)$.

Let us now examine this method with the piecewise linear approximation space $V_h^1 = \left\{v_h \in \mathscr{C}^0(\overline{\Omega}) \mid v_{hT} \in P_1(T)\right\}$. The problem is then written as :

$$\left\{\begin{array}{l} \text{Find } u_1 \in V_h^1 \text{ s.t.} \\ a(u_1,v_1)_{supg} = \int_\Omega \kappa \underline{\nabla} u_1 \cdot \underline{\nabla} v_1 + (\underline{w}\cdot\underline{\nabla} u_1) v_1 + \sum_{T\in\mathscr{T}_h} \tau_T^{supg} \int_T (\underline{w}\cdot\underline{\nabla} u_1 - f)\ \underline{w}\cdot\underline{\nabla} v_1 \\ = \int_\Omega f v_1,\ \ \forall v_1 \in V_1 \end{array}\right. \tag{4.6.47}$$

The value of τ_T^{supg} will be considered for two extreme cases of the Peclet number Pe_T.

Convection dominating diffusion case ($Pe_T \to \infty$)

The following value is obtained :

$$\tau_T^{supg} = \frac{h_T}{2\|\underline{w}\|} \tag{4.6.48}$$

The factor associated with the velocity simplifies in method (4.6.47) and the modification is only mesh dependent (via h_T).

Diffusion dominating convection case ($Pe_T \to 0$)

The following value is obtained :

$$\tau_T^{supg} = \frac{m_k h_T^2}{4\kappa} \tag{4.6.49}$$

This factor is small and hence the modification with respect to the Galerkin method is small too.

4.6.2 Bubble function method

The bubble function method examined in (43; 44; 45) is here developed. It mainly relies on RFB assumptions. The bubble space is not determined by analytical functions but by the equations that RFB bubble functions must verify.

Firstly, a piecewise linear function space is again selected. This set is associated with the not yet determined bubble function space $\mathscr{B}$:

$$V_h = V_h^1 \oplus \mathscr{B} \tag{4.6.50}$$

where

$$\begin{array}{lcl} V_h^1 & = & \text{continuous and piecewise linear function space ,} \\ \mathscr{B} & = & \left\{ b_T \in \mathscr{H}_0^1(T),\ T \in \mathscr{T}_h \right\}. \end{array} \tag{4.6.51}$$

A function $u_h \in V_h^1$ is decomposed in a unique way in $u_h = u_1 + u_B$, where $u_{1|T} \in P_1(T)$ and $u_B = \sum_{T \in \mathscr{T}_h} c_T b_T$ (with c_T being real weighting coefficients).

In order to apply the RFB method, appropriate bubble functions are determined in such a way that the strong formulation residual vanishes on the interior of each element :

$$\begin{array}{c} \mathscr{L}u_h = f \text{ in int } T,\ T \in \mathscr{T}_h \\ \Leftrightarrow \\ \mathscr{L}u_B = -(\mathscr{L}u_1 - f) \text{ in int } T,\ T \in \mathscr{T}_h \end{array} \tag{4.6.52}$$

This is equivalent to imposing on each $T \in \mathscr{T}_h$ the relation

$$\mathscr{L}(c_T b_T) = -\mathscr{L}u_1 + f \tag{4.6.53}$$

which writes as :

$$-c_T \kappa \Delta b_T + c_T \underline{w} \cdot \underline{\nabla} b_T = -\underline{w} \cdot \underline{\nabla} u_1 + f = Cst_{|T} \tag{4.6.54}$$

This formulation still requires to determine two unknown functional parameters on each element (c_T and b_T). The two steps of the static condensation procedure are now applied, viz. the projection on the bubble function space and the determination of the modification with respect to original Galerkin method.

1. The projection onto the bubble function space provides the expression for c_T. Indeed, this operation successively gives :

$$\int_\Omega \kappa \underline{\nabla} u_h \cdot \underline{\nabla} v_B + (\underline{w} \cdot \underline{\nabla} u_h) v_B = \int_\Omega f v_B,\ \forall v_B \in \mathscr{B}$$

$$\int_T \kappa \underline{\nabla}(u_1 + c_T b_T) \cdot \underline{\nabla} b_T + (\underline{w} \cdot \underline{\nabla}(u_1 + c_T b_T))\, b_T = \int_T f b_T,\ T \in \mathscr{T}_h$$

   As a piecewise linear interpolation space is considered ($u_1 \in V_h^1$) and as bubble functions vanish on the boundaries of T, Green's theorem gives

$$\int_T \kappa c_T \underline{\nabla} b_T \cdot \underline{\nabla} b_T + (\underline{w} \cdot \underline{\nabla} u_1)\, b_T = \int_T f b_T,\ T \in \mathscr{T}_h.$$

   Since f et $\underline{w}$ are element-constant, the value of c_T can finally be inferred as :

$$c_T = \frac{\int_T b_T}{\kappa \int_T \|\underline{\nabla} b_T\|^2} (f - \underline{w} \cdot \underline{\nabla} u_1)_{|T}. \tag{4.6.55}$$

2. The step relating to bubble function projection onto V_h^1 successively gives (43) :

$$\begin{aligned}
&\int_\Omega \kappa \underline{\nabla} u_B \cdot \underline{\nabla} v_1 + (\underline{w} \cdot \underline{\nabla} u_B) v_1 \\
&= 0 + \sum_{T \in \mathscr{T}_h} \int_T (\underline{w} \cdot \underline{\nabla}(c_T b_T)) v_1 \\
&= 0 - \sum_{T \in \mathscr{T}_h} c_T \int_T (\underline{w} \cdot \underline{\nabla} v_1) b_T \\
&= 0 - \sum_{T \in \mathscr{T}_h} c_T (\underline{w} \cdot \underline{\nabla} v_1)_{|T} \int_T b_T \\
&= 0 + \sum_{T \in \mathscr{T}_h} \frac{(\int_T b_T)^2}{\kappa \int_T \|\underline{\nabla} b_T\|^2} (\underline{w} \cdot \underline{\nabla} u_1 - f)_{|T} (\underline{w} \cdot \underline{\nabla} v_1)_{|T} \\
&= 0 + \sum_{T \in \mathscr{T}_h} \frac{1}{|T|} \frac{(\int_T b_T)^2}{\kappa \int_T \|\underline{\nabla} b_T\|^2} \int_T (\underline{w} \cdot \underline{\nabla} u_1 - f) (\underline{w} \cdot \underline{\nabla} v_1) \\
&= 0 + \sum_{T \in \mathscr{T}_h} \tau_T^b \int_T (\underline{w} \cdot \underline{\nabla} u_1 - f) (\underline{w} \cdot \underline{\nabla} v_1),
\end{aligned}$$

where τ_T^b is defined as :

$$\tau_T^b = \frac{1}{|T|} \frac{(\int_T b_T)^2}{\kappa \int_T \|\underline{\nabla} b_T\|^2}. \tag{4.6.56}$$

The bubble function formulation, equivalent to (4.4.21), is finally rewritten as :

$$\left\{ \begin{array}{l}
\text{Find } u_1 \in V_h^1 \text{ s.t.} \\
a(u_1, v_1)_b = \int_\Omega \kappa \underline{\nabla} u_1 \cdot \underline{\nabla} v_1 + (\underline{w} \cdot \underline{\nabla} u_1) v_1 + \sum_{T \in \mathscr{T}_h} \tau_T^b \int_T (\underline{w} \cdot \underline{\nabla} u_1 - f) (\underline{w} \cdot \underline{\nabla} v_1) \\
= \int_\Omega f v_1, \forall v_1 \in V_h^1 \quad \text{with } \tau_T^b = \frac{1}{|T|} \frac{(\int_T b_T)^2}{\kappa \int_T \|\underline{\nabla} b_T\|^2} \ .
\end{array} \right. \tag{4.6.57}$$

Comparing this method (4.6.57) with the SUPG method (4.6.47), it can be seen that both methods are similar when applied with a piecewise linear discretization space V_h^1, provided the respective residual weighting factor τ_T^b and τ_T^{SUPG} are close together.

The "magic" factor τ_T of SUPG was initially determined by numerical experimentation on well-known benchmarks, such as the rotating cone test. On the other hand, it appears that τ_T^b depends on the bubble function space and is hence not arbitrary. Bubble functions are indeed determined by Eq. (4.6.54). Introducing c_T in Eq. (4.6.54) ,

we successively find :

$$\begin{aligned} & c_T(-\kappa\Delta b_T + \underline{w}\cdot\underline{\nabla} b_T) = -\underline{w}\cdot\underline{\nabla} u_1 + f \\ & \frac{\int_T b_T}{\kappa\int_T \|\underline{\nabla} b_T\|^2}(f - \underline{w}\cdot\underline{\nabla} u_1)(-\kappa\Delta b_T + \underline{w}\cdot\underline{\nabla} b_T) = -\underline{w}\cdot\underline{\nabla} u_1 + f \\ & -\kappa\Delta b_T + \underline{w}\cdot\underline{\nabla} b_T = \frac{\kappa\int_T \|\underline{\nabla} b_T\|^2}{\int_T b_T} \ . \end{aligned} \tag{4.6.58}$$

This problem is equivalent to determine b_T such that :

$$\begin{cases} -\kappa\Delta b_T + \underline{w}\cdot\underline{\nabla} b_T = 1 & \text{in } T \\ b_T = 0 & \text{on } \partial T \end{cases} \tag{4.6.59}$$

In order to demonstrate this equivalence, the first equation of (4.6.59) is successively multiplied by b_T and integrated. Integration by parts of this equation provides the equality :

$$\kappa\int_T \|\underline{\nabla} b_T\|^2 = \int_T b_T \ . \tag{4.6.60}$$

Finally, Eq. (4.6.60) provides a simplified definition of τ_T^b :

$$\tau_T^b = \frac{1}{|T|}\int_T b_T . \tag{4.6.61}$$

In conclusion, a numerical method (4.6.57) has been developed on the basis of so-called bubble functions, whose space $\mathscr{B}$ and weighting factor are determined by Eq. (4.6.59) and Eq. (4.6.61) respectively.

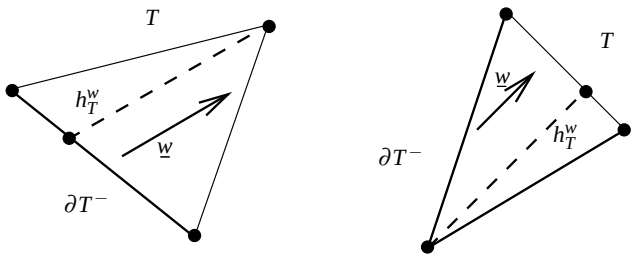
As previously, let us now examine two extreme cases in order to compare the respective weighting factors of the bubble function and SUPG methods.

Convection dominating diffusion case ($Pe_T \to \infty$)

When $\kappa \ll h\|\underline{w}\|$, the limit of Eq. (4.6.59) is written as :

$$\begin{cases} \underline{w}\cdot\underline{\nabla}\tilde{b}_T = 1 & \text{in } T \\ \tilde{b}_T = 0 & \text{on } \partial T^- \end{cases} \tag{4.6.62}$$

where $\tilde{b}_T$ is the solution of this equation and ∂T^- is the inlet boundary of T ($\underline{w}\cdot\underline{n} \leq 0$). It can be demonstrated (45) that, if b_T is the solution of Eq. (4.6.59), $\lim_{\kappa\to 0}\int_T b_T =$



(a) (b)

Figure 4.3: Different relative orientations of the elements with respect to the velocity field (cases (a) and (b)). Determination of the largest segment parallel to the velocity and entirely contained in the element.

$\int_T \tilde{b}_T$. Considering then the first equation of (4.6.62), the gradient of $\tilde{b}_T$ is constant on each element. The projection of the gradient on the velocity is then :

$$\frac{\underline{w}}{\|\underline{w}\|} \cdot \underline{\nabla} \tilde{b}_T = \frac{1}{\|\underline{w}\|}. \tag{4.6.63}$$

Bubble functions are thus approximated by "pyramids" whose height is zero on the inlet side(s) and maximum on the intersection of the largest segment parallel to the velocity and entirely contained in the element with outlet boundary (Fig. 4.3). If this segment length is $h_T^{\underline{w}}$, the pyramid height is then $h_T^{\underline{w}} \frac{1}{\|\underline{w}\|}$. Bubble function integral approximation can then be deduced from these observations :

$$\int_T b_T \approx \int_T \tilde{b}_T = \frac{h_T^{\underline{w}}}{3\|\underline{w}\|} |T|, \tag{4.6.64}$$

and τ_T^b is finally deduced as :

$$\tau_T^b = \frac{h_T^{\underline{w}}}{3\|\underline{w}\|}. \tag{4.6.65}$$

This expression is of the same order of magnitude as for SUPG method (where it is equal to $\frac{h_T}{2\|\underline{w}\|}$) even if the bubble function method provides less coercivity since $1/2 > 1/3$ and $diam T = h_T \geq h_T^{\underline{w}}$.

Diffusion dominating convection case ($Pe_T \to 0$)

In this case, the bubble function method verifies the following limit equation :

$$\begin{cases} -\kappa\Delta\tilde{b}_T = 1 & \text{in } T \\ \tilde{b}_T = 0 & \text{on } \partial T^- \end{cases} \tag{4.6.66}$$

In this case, it can be demonstrated that Eq. (4.6.66) provides (45) :

$$\tau_T^b = \frac{1}{|T|}\int_T b_T \approx C\frac{h_T^2}{\kappa}. \tag{4.6.67}$$

This result is exactly the same as the one provided by the SUPG method. The stability of the bubble function and SUPG methods is then the same for diffusion dominated problems.

4.6.3 Link with the multiscale approach

The operator is now treated with the multiscale strategy (4.4.33) (46) :

$$a(\bar{u},\bar{v}) + (-M_T(\mathscr{L}\bar{u} - f), -\mathscr{L}^*\bar{v}) = (f,\bar{v}) \tag{4.6.68}$$

where

$$M_T(p(x)) = -\int_T \mathrm{g}(x,y)p(x) \tag{4.6.69}$$

with g solution of

$$\begin{cases} \mathscr{L}\mathrm{g} &= \delta & \text{in } T \\ \mathrm{g} &= 0 & \text{on } \partial T,\ \forall T \in \mathscr{T}_h\ . \end{cases} \tag{4.6.70}$$

where δ is the Dirac's measure. Let us keep V_h^1 as the approximation space in order to compare the methods. Then $\bar{u} \in V_h^1$ and $\bar{v} \in V_h^1$. The adjoint of the advection-diffusion operator on $\mathscr{H}_0^1(\Omega)$ is $\mathscr{L}^*(\cdot) = -\nabla\cdot(\kappa\nabla(\cdot)) - w\cdot\nabla(\cdot)$. As f et w are element-constant, the multiscale method finally writes as :

$$\begin{cases} \text{Find } u_1 \in V_h^1, \text{ s.t.} \\ \\ a(u_1,v_1) + \sum_{T\in\mathscr{T}_h} \tau_T^{ms}\int_T (w\cdot\nabla u_1 - f)\ w\cdot\nabla v_1 = (f,v_1),\ \ \forall v_1 \in V_h^1 \\ \\ \text{where } \ \tau_T^{ms} = \frac{1}{|T|}\int_T\int_T \mathrm{g}(x,y) \end{cases} \tag{4.6.71}$$

The following remarks are then deduced by comparison with the previous methods :

- The relation (4.5.38) between the weighting factor τ_T and the Green's function is retrieved.

- Defining the bubble functions as

$$b_T(x) = \int_T \mathrm{g}(x,y)\ 1_y\ dT_y, \tag{4.6.72}$$

and from definition of Green's function, b_T is deduced to be solution of the equation

$$\begin{cases} -\kappa\Delta b_T + w\cdot\nabla b_T = 1 & \text{in } T \\ b_T = 0 & \text{on } \partial T. \end{cases} \tag{4.6.73}$$

This confirms the result provided by Eq. (4.6.59).

- Moreover, from the bubble function definition (4.6.72) and by (4.6.71), it appears that $\tau_T^{ms} = \frac{1}{|T|}\int_T b_T$. This gives the same relation between bubble functions and stabilization term as previously (4.6.61).

4.6.4 Conclusions

This subsection closes the theoretical study and literature summary about the bubble function method. Main conclusions are listed herebelow :

- The bubble function stabilization effect has been demonstrated by means of a comparison with stabilized methods and more precisely with the multiscale approach theory. Accordingly, the effect of the "magic" factor τ has been explained by application of a rigorous method (35) and its optimal value has been determined.

- The choice of the bubble function space remains a hard task. Although a building methodology has been described in (39), this technique is more theoretical than applicable.

- RFB functions are particular bubble functions. Their specificity lies in the fact that they satisfy the problem strong formulation inside each element. It is proved that applying the Galerkin method with a classical approximation space together with the RFB function set is as stable as other stabilized methods. For a linear problem, this space can be obtained by approximation with a subgrid model and a two-scale FEM technique. This method is developed in next section.

- The bubble function space has been approximately determined in extremal cases. The shape and properties of the bubble functions have been illustrated accordingly. Hence it has appeared that the bubble function space strongly depends on the problem.
- Until now, the only advantage of using bubble functions is to provide some intuitive understanding of stabilized methods. The bubble function method can also be used to determine a posteriori errors (44).
- Finally, this approach opens a way for the design and investigation of new stabilized schemes.

The next section is devoted to the development of a new stabilized method, which is based on the principles explained in this present section. Numerical results will illustrate the difference between classical approaches and this new strategy.

4.7 Bubble function-SUPG method

We here develop an original bubble function method devoted to approximate the RFB technique. Indeed, on the one hand, as material forming processes involve many unknown fields, it is important that the computational time required to stabilize the numerical scheme remains moderate, requirement which is fulfilled by the bubble function strategy by means of the static condensation technique. On the other hand, the bubble function method principle is smart and simple and it is besides expected to get rid of the "magic" parameter τ in some way.

The method we have implemented will firstly be presented and analyzed by comparing the related results with SUPG and Galerkin results. The adjunction of a streamline effect will then be motivated and a method combining the advantages of bubble function and SUPG approaches will be designed. Results will again be analyzed. A computational cost study will be performed and the extension to the 3D case will finally be considered.

4.7.1 An original RFB approximation

We have chosen to work with piecewise linear bubble functions (Fig. 4.4), to which a so-called generation is attached. The higher the generation is, the higher the number

of bubble function related to each element will be (and the more each element will be refined) (Fig. 4.5). It is then expected to reach a required precision inside each element and hence a good approximation of the RFB space. Since piecewise linear functions low integration rules are sufficient, a piecewise linear space is again chosen for the basic approximation set, and the set U of Eq. (4.4.15) is selected as V_h^1.

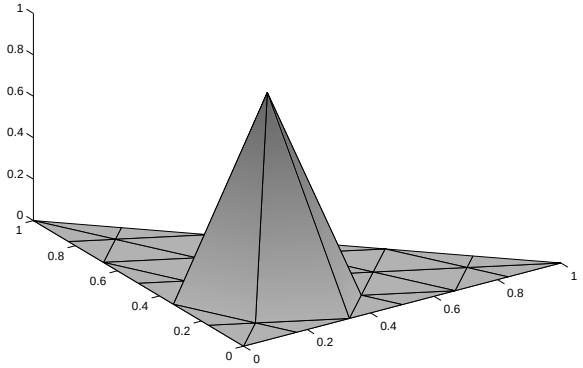


Figure 4.4: Example of piecewise linear bubble function, third generation. The bubble function vanishes on the element boundary

In order to define bubble function generation, an element meshing rule has to be established. In the present case, we consider an equilateral triangle, which is easily mapped to the actual element. The generation i ($i \geq 1$) is then defined by cutting the element side by $2i-1$ perpendicular and equally spaced lines (Fig. 4.5).

With this strategy, three kinds of node types appear :

- The element boundary nodes, which are located between the nodes of the basic mesh : the associated shape functions identically vanish in order to fulfill the bubble requirements. These nodes are also called "false" nodes.
- The bubble function nodes : each nodal value exactly determines the related local bubble function (Fig. 4.4).
- The basic mesh nodes : these nodes are defined by the global mesh and are associated with the basic problem unknowns.

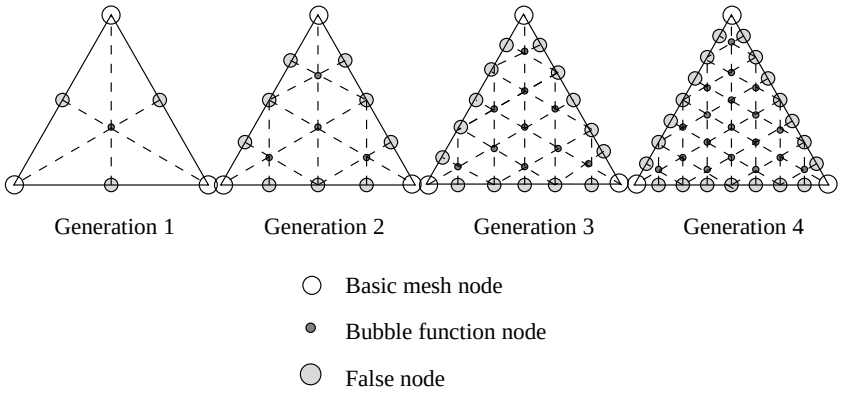


Figure 4.5: Different generations of bubble functions

In an FEM implementation, each element is treated by integration of the problem weak formulation. Considering that there are n_{T_G} shape functions and n_{T_B} bubble functions per element, the projection related to the problem weak formulation is performed on $n_{T_G} + n_{T_B} = n_{T_T}$ functions for each element. The local stiffness matrix size is then $n_{T_T} \times n_{T_T}$. The static condensation technique will decrease this size up to the "Galerkin" method size (the same size as obtained with Galerkin method), which is exactly the element shape function number n_{T_G}. The "Galerkin" matrix A is then transformed into another matrix A' by means of static condensation operations.

$$\begin{matrix}\phi_f \rightarrow \\ \phi_B \rightarrow\end{matrix}\left\{\begin{array}{|c|c|}\hline A & \cdots \\ & \cdots \\ & \cdots \\ \hline \cdots & \ddots \\ \hline\end{array}\right\}\begin{matrix}n_{T_G} \\ n_{T_B}\end{matrix} \quad \begin{matrix}\longrightarrow \\ \text{(static condensation)}\end{matrix} \quad \left[\begin{array}{c} A' \end{array}\right]\Bigg\} n_{T_G} \tag{4.7.74}$$

Numerical results obtained by means of this method will be compared with Galerkin and SUPG results. The SUPG method was implemented taking into account some remarks of T. de Mulder (47). He indeed points out that SUPG is quite sensitive to the element size h_T. Evaluating correctly the element size then appears to be crucial.

Before analyzing these results, let us conclude that a simple idea should be to combine the advantages of the bubble function and SUPG methods in order to design a more stable strategy, viz. the **Bubble function-SUPG** strategy. In this case, bubble functions would undergo a streamline upwinding effect by an increased weighting of the incoming information in order to treat more accurately the transport phenomenon inside each subelement.

The next subsection is devoted to provide a numerical comparison between these methods on a well known benchmark, viz. the rotating cone problem.

4.7.2 Numerical results

The rotating cone experiment consists in considering the evolution of a coloured cone in a rotating flow, including a more or less pronounced diffusion effect. This problem is modelled by the non-stationary advection-diffusion equation. With the definition of $\mathscr{L}$ provided by Eq. (4.6.39), this equation is written as :

$$\frac{\partial u}{\partial t} + \mathscr{L}(u) = 0. \tag{4.7.75}$$

We have fixed the flow angular velocity at 1 rad/s. The initial cone profile takes the shape of a square cosine function of period $1/3$ centered at point $(0.25, 0.5)$ in standard units (Fig. 4.6). The calculation domain is the $[0,1] \times [0,1]$ square. Two different cases have been considered. The first case deals with a slight transport domination in front of the diffusion effect. The Peclet number, defined as $Pe = \|\underline{w}\| L/\kappa$ where L is the characteristic length selected as the cone diameter, is then equal to 16.66 ($\kappa = 10^{-2}$ m^2/s, $\|\underline{w}\| = 0.5$ m/s and $L = 0.33$ m). The second case deals with a convection dominated problem, since $Pe = 1666$ ($\kappa = 10^{-4}$ m^2/s, $\|\underline{w}\| = 0.5$ m/s and $L = 0.33$ m).

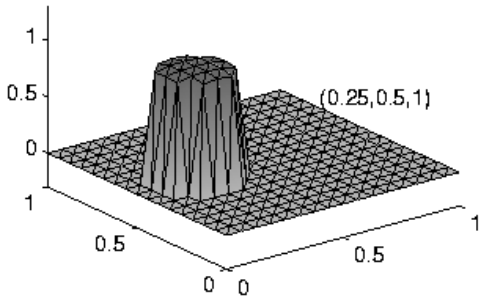


Figure 4.6: Initial cone, calculation domain and mesh, together with initial position and height of the cone top.

The domain discretization is performed with triangles forming a structured mesh com-

posed of 33×33 nodes. The element Peclet numbers are then $Pe_h = 1.56$ and $Pe_h = 156$ respectively for the two investigated cases[4]. Time integration is performed by means of Crank-Nicolson scheme and the integration time step is $2\pi/200$ s.

Figures 4.7 and 4.8 show the results obtained after one cone rotation with Galerkin, SUPG, bubble function and Bubble-SUPG methods, for $Pe_h = 1.56$ and $Pe_h = 156$ respectively whereas Fig. 4.9 illustrates a comparison between results obtained with different bubble function generations for $Pe_h = 156$.

In Fig. 4.7 it appears that the Galerkin approach is quite stable for this problem, even if the top of the cone shows a slight solution overshoot. When the advection influence is increased, the Galerkin method turns out to be unstable and oscillations as well as a large overshoot appear (Fig. 4.8). However, the SUPG technique stabilizes the numerical operator and oscillations as well as the overshoot decrease (Fig. 4.8). One can still observe a remaining oscillation near the cone basis. When used alone, the bubble function method results exhibit a large overshoot. However, there are less oscillations when a higher generation is used (Fig. 4.9). Even if these results are not satisfying, the oscillations decrease has led to us to combine the SUPG and bubble function strategies. The combined method seems convincing since the overshoot is decreased and the spurious oscillations are almost totally suppressed (Figs. 4.8 and 4.9).

In conclusion, the Bubble function-SUPG methods produce a better cone height conservation and bring less spurious oscillations. Bubble functions alone lead to poor results even if oscillations are decreased. Since the use of a structured mesh could play some role in these observations, a similar study should be carried out on unstructured meshes. Another solution should be to increase the bubble generation number, but in this case the related computational cost could be too high as it is illustrated in the next section. Finally, an interesting study could be to analyze the respective effects of the global and local mesh refinements in order to determine the optimal 'local-global' discretization ratio. Such strategies are indeed known to lead to 'super-convergence' effects (48; 49; 50).

4.7.3 Computational cost

In order to estimate the computational cost of a problem solution, some specific parameters of the discrete problem are listed in Table 4.1. The 2D problem is treated

[4] As observed by T. de Mulder (47), the element Peclet number definition is not univocal and depends on the element size evaluation. In our study, we have simply taken as element size the length of the side adjacent to the element square angle, viz. $h = 1/32$ m.

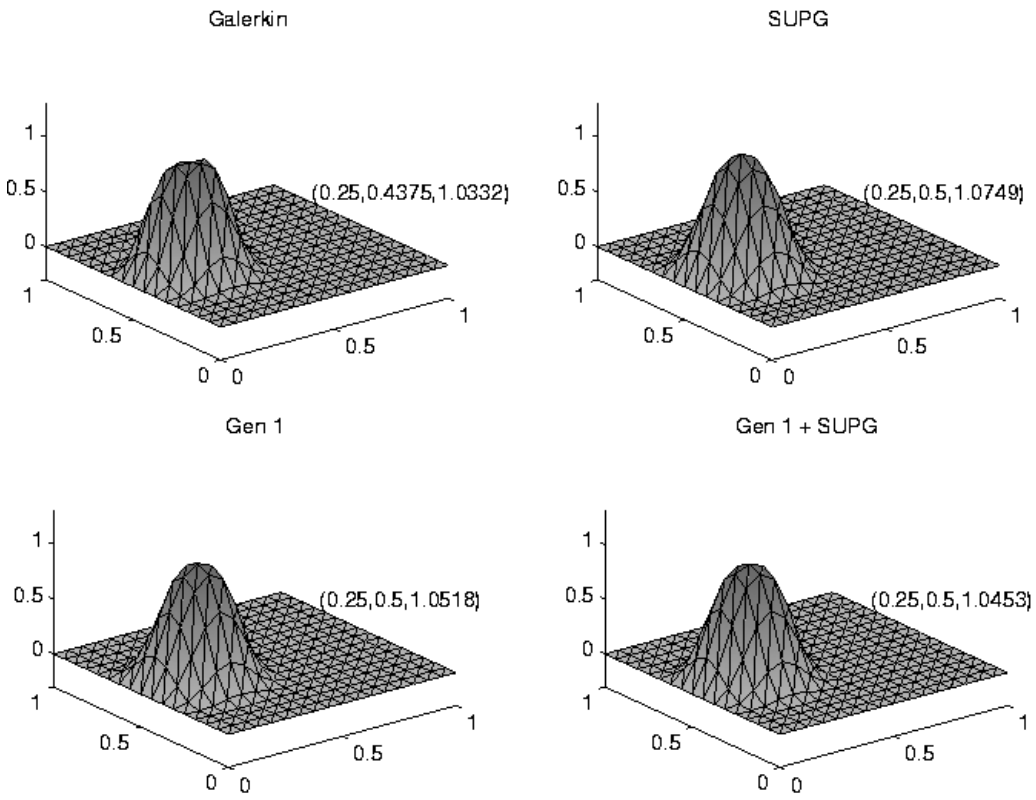


Figure 4.7: Rotating cone test for $Pe_h = 1.56$ on a 33×33 node mesh. Comparison between the results obtained by means of Galerkin, SUPG and first generation of bubble function (Gen1) methods, without or with SUPG effect inside the subelements. Position and height of the cone top after one rotation.

first.

Classical frontal or band solvers imply to perform N^2 pivoting operations with matrices of size $M \times M$. The computational cost of these methods is then $\mathcal{O}(N^2M^2)$. Two ways of improving the solution accuracy are examined, viz. (i) a global mesh refinement and (ii) an element refinement performed by means of the bubble function method.

In a refinement of factor a the unknown number is multiplied by a^2 and the band or

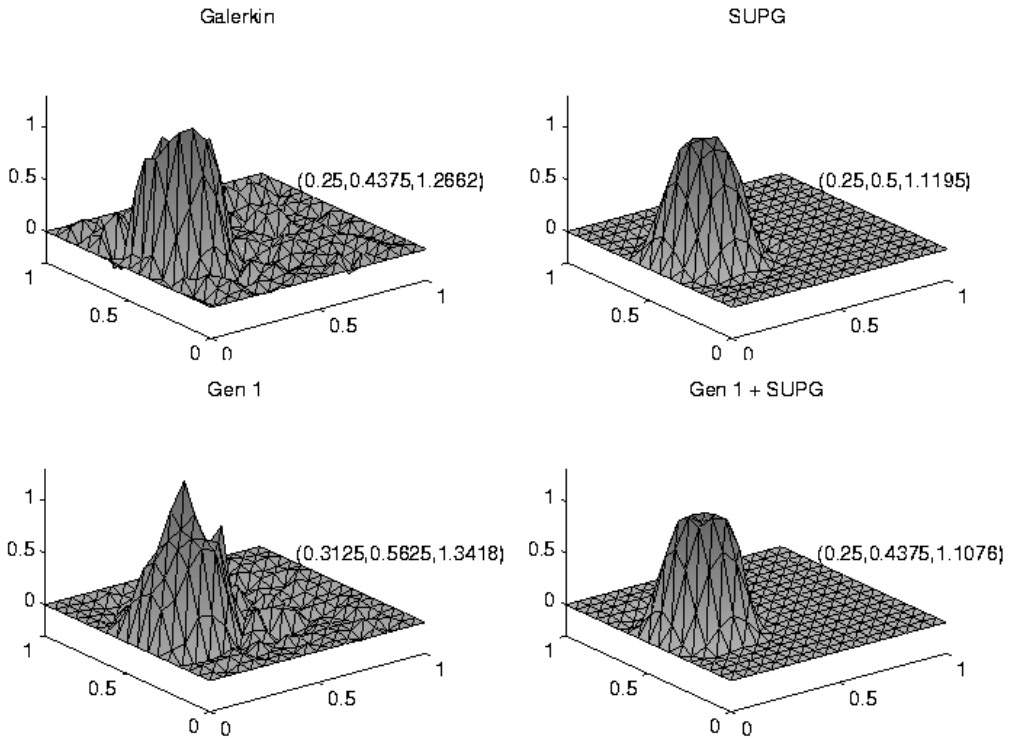


Figure 4.8: Rotating cone test for $Pe_h = 156$ on a 33×33 node mesh. Comparison between the results obtained by means of Galerkin, SUPG and first generation of bubble function (Gen1) methods, without or with SUPG effect inside the subelements. Position and height of the cone top after one rotation.

front length by a. The total computational cost becomes $\mathcal{O}(a^4N^2M^2)$ and the original cost is multiplied by a^4.

With the bubble function method, static condensation is responsible of the operation number increase. With classical gaussian elimination, this cost increase is roughly $\mathcal{O}\left(cb^2\right)$ per element when static condensation is performed by full matrix operations. Assuming that the element number is of the same order as the node number ($\mathcal{O}(T) = \mathcal{O}(N)$), the related cost increase is then $\mathcal{O}(N^2cb^2)$. These observations thus provide the computational cost summary recalled in Table 4.2.

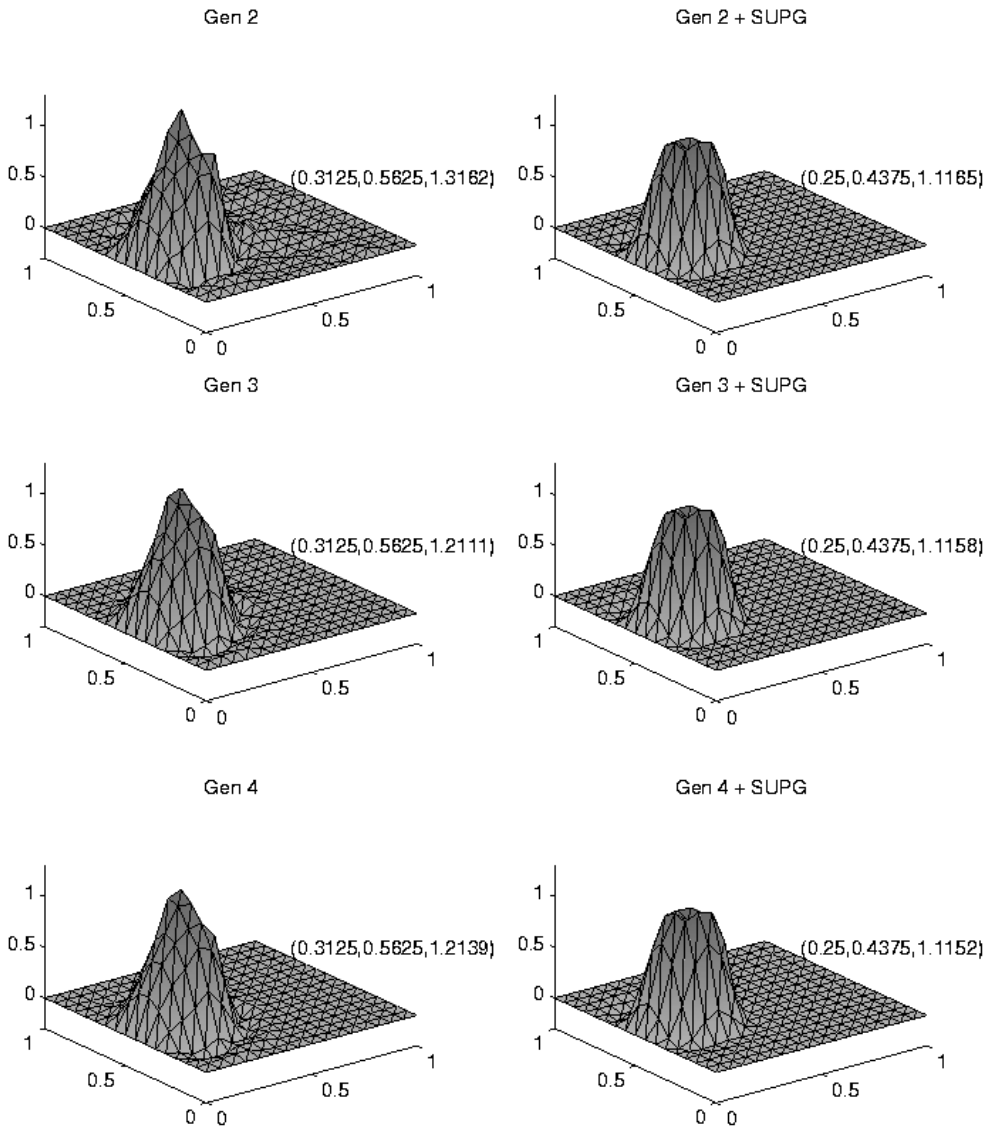


Figure 4.9: Rotating cone test for $Pe_h = 156$ on a 33×33 node mesh. Comparison between results obtained by means of second, third and fourth generation of bubble function (Gen2,Gen3,Gen4) methods, without or with SUPG effect inside the subelements. Decreasing of the oscillations near the cone basis when choosing a higher generations. Position and height of the cone top after one rotation.

Symb.	Description
N	Characteristic number of nodes on a side of the domain
M	Front or band length ($\sim \mathcal{O}(N)$ in 2D problems, $\sim \mathcal{O}(N^2)$ in 3D problems)
T	Number of elements
a	Mesh refinement factor
b	"True" number of bubble functions inside each element
c	Total node number inside each element (mesh nodes + true bubble function nodes)

Table 4.1: Parameters of the numerical solution

Method	Computational cost
Band or frontal	$\mathcal{O}(N^2M^2)$
Same with refinement factor a	$\mathcal{O}(a^4N^2M^2)$
Bubble functions with node number b	$\mathcal{O}(N^2M^2)+\mathcal{O}(N^2cb^2)$

Table 4.2: Computational cost of different refinement strategies for 2D problems.

This analysis shows that with refinement such that the effective element size is equal, bubble methods need less computational cost since in most cases $a^4M^2 > cb^2$. Nevertheless, on the inter-element boundaries, the solution will be less accurate since the interpolation is still performed on the original mesh. Indeed, as the bubble functions vanish along the inter-element boundaries, they have no influence on the inter-element interpolation. For similar rough meshes, the bubble function method is expected to have slightly more accurate and much more stable results with a cost increase. This method should besides be relevant in the 3D case but this extension brings a lot of difficulties as it is explained in next section.

4.7.4 3D Extension

The attractive aspect of our bubble function approach lies in the ability to automatically generate bubble functions in order to reach a given accuracy. Unfortunately, in the 3D case, subdividing a tetrahedron in small tetrahedra is not so easy. The first generation is however obtained by a similar way as in 2D, by dividing each edge by its bissector plane and producing 48 tetrahedra and 15 potential new nodes (with only a single bubble function). At this point, we did not find any automatic generation strategy.

The geometrical complexity and the automatic subdivision technique difficulties pertain to the 3D treatment. Hence, although the improvements obtained by using 2D

bubble functions were quite promising, the 3D extension has not been investigated. Therefore, the SUPG technique was used in 3D numerical simulations (see Chapter 5).

4.8 Moving front issue

The last numerical problem to address in order to simulate SRIM or RTM processes is to propose an algorithm for solving the moving front problem. In the modelling of material forming processes, a fluid continuously fills the mold in which it is injected. The void space is then filled by the injected fluid and a moving front separates both phases. In general, an accurate description of the moving boundary is important especially when surface phenomena have a large influence on the flow or when it is crucial that no void space remains present after mold filling. The phase domains then evolve with time and the related equations as well. Two main strategies exist to address this issue numerically :

- The mold mesh is fixed and a filling variable is added. This variable evolves from 0 when the neighbourhood of the considered point is totally empty of liquid to 1 when it is saturated.
- The mesh is adapted at each time step in order to fit the saturated domain. Its boundary then encloses the saturated zone, and the non-static part of this border is called the moving front.

The advantage of the first strategy lies in keeping the same mesh during the entire simulation. Nevertheless, in order to precisely describe the front, a high mesh refinement is needed. Application of the second approach is better suited when accurate knowledge of the front is desired. However, computing a new mesh at each time step is obviously a main disadvantage of this technique.

We here present an algorithm based on the second strategy. Indeed, front tracking is very important in RTM and SRIM processes mainly in order to avoid gas jailing. This algorithm involves different substeps at each time step :

1. Integration of some unknown fields on the old mesh (assiciated with the previous time step).
2. Determination of the moving front position.

3. Treatment of fluid-fluid or fluid-wall meetings.
4. Field extrapolation on the new domain taking the boundary conditions into account.
5. New mesh determination.
6. Final field interpolation in the appropriate regions.
7. Correction of the extrapolated fields.
8. Implicit calculation of the pressure field.

Two different solvers are needed, first for the flow and second for the geometry. The first solver involves solving the flow equations and besides to use interpolation and extrapolation techniques in order to take into account the mesh evolution. The geometrical solver deals with front motion, front-front and front-wall meetings, and remeshing. Solving accurately the advection-dominated diffusion equation is very important for both the flow and the geometrical solvers.

Unfortunately, the geometrical solver was not yet fully implemented in our research group. Hence, the extrapolation and interpolation techniques could not be investigated and the complete solution of the problem has not been achieved.