



An adaptive enrichment algorithm for advection-dominated problems.

Rémi Abgrall, Arnaud Krust

**RESEARCH
REPORT**

N° 7866

January 2012

Project-Teams Bacchus



An adaptive enrichment algorithm for advection-dominated problems.

Rémi Abgrall^{*†}, Arnaud Krust ^{* †}

Project-Teams Bacchus

Research Report n° 7866 — January 2012 — 21 pages

Abstract: We are interested in developing a numerical framework well suited for advection-diffusion problems when the advection part is dominant. In that case, given Dirichlet type boundary condition, it is well known that a boundary layer develops. In order to resolve correctly this layer, standard methods consist in increasing the mesh resolution and possibly increasing the formal accuracy of the numerical method. In this paper, we follow another path: we do not seek to increase the formal accuracy of the scheme but, by a careful choice of finite element, to lower the mesh resolution in the layer. Indeed the finite element representation we choose is locally the sum of a standard one plus an enrichment. This paper proposes such a method and with several numerical examples, we show the potential of this approach. In particular we show that the method is not very sensitive to the choice of the enrichment.

Key-words: enrichment, adaptivity, stabilized finite elements, XFEM, boundary-layer

^{*} Bacchus Team, INRIA Bordeaux Sud-Ouest, 351 cours de la libération, 33400 Talence, France

[†] Institut de mathématiques de Bordeaux, Univ. Bordeaux, 351 cours de la libération, 33400 Talence, France

**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Un algorithme d'enrichissement adaptatif pour les problèmes où l'advection domine.

Résumé : Nous nous intéressons à développer un cadre numérique adapté aux problèmes d'advection-diffusion lorsque l'advection est dominante. Dans ce cas, étant données des conditions au bord de type Dirichlet, il est connu qu'une couche limite se développe. Pour résoudre correctement cette couche, les méthodes standards consistent à augmenter la résolution du maillage et éventuellement l'ordre formel de la méthode numérique. Dans ce papier, nous suivons une autre voie: nous ne cherchons pas à augmenter l'ordre formel du schéma, mais à baisser la résolution du maillage dans la couche limite par un choix judicieux des éléments finis utilisés. En effet, nous choisissons une représentation par éléments finis qui soit la somme d'une approximation classique plus une fonction d'enrichissement. Ce papier propose une telle méthode et montre le potentiel de cette approche avec de nombreux exemples numériques. En particulier, nous montrons que cette méthode n'est pas trop sensible au choix de l'enrichissement.

Mots-clés : enrichissement, adaptation, éléments finis stabilisés, XFEM, couche-limite

Contents

1	Introduction	3
2	Numerical scheme: an Enriched Stabilized Finite Element Method	4
2.1	Space discretization and notations	4
2.2	Formulation of the problems	5
2.3	Enrichment in the XFEM spirit	5
2.4	Interpolation error on the enriched space	6
2.5	Stabilization	7
2.6	Boundary conditions	8
2.7	Non-linearity treatment	9
2.8	How to choose enrichment functions	10
3	An automated enrichment adaptation	11
3.1	Some explanation about mesh adaptation	12
3.2	Reconstruction of derivatives	13
3.3	Algorithm	13
3.4	Exponential enrichment	13
3.5	Polynomial enrichment	14
4	Towards h/e-adaptivity	18
5	Concluding remarks	20

1 Introduction

We are interested in advection-dominated problems with viscosity. It is well known that this kind of problems require specific treatment, especially when a boundary-layer develops. A large family of schemes well suited for these problems are the stabilized finite elements methods. They deal with the oscillations that destroy the accuracy of non-stabilized schemes, mostly by adding artificial viscosity. This is not enough to deal accurately with boundary-layers. Indeed, when the boundary-layer is under-resolved, any scheme based on a classical polynomial approximation cannot be of optimal order. An usual cure to this problem is to refine the mesh near the boundary. Automated strategies to do so are now very efficient and they achieve to refine the mesh so as to capture the boundary-layer or other irregularities of the solution (see [7]).

Another approach to deal with this kind of problems is to modify the basis functions of the approximation space so as to capture small scale effects. Indeed, under an inf-sup condition, it is well known that the error of a scheme is bounded by the interpolation error on the approximation space. This approach has already been extensively used for fracture mechanics, and more recently for advection-dominated problems [1],[14],[11]. These methods enrich the space with exact solutions of the homogeneous equation or at least solutions that are derived analytically from the problem. These methods show outstanding stability and accuracy properties, but they require a very good *a priori* knowledge of solutions of the homogeneous problem. The adaptive variational multiscale method (AVMM) [12] is a try to avoid this drawback by a local, numerical resolution of the small scales but is designed for elliptic problems.

We want to combine elements from all these methods in order to design an automated and robust numerical scheme for advection-dominated problems. We thus start from a general XFEM framework, which shows good potential in the case of an analytically derived enrichment. But as our goal is to deal with Navier-Stokes equations, we cannot assume such a knowledge to design the enrichment space. For the scheme to be robust with respect to the choice of enrichment functions, we stabilize the scheme to obtain a stabilized XFEM-like method.

A further step is the development of an algorithm to chose the enrichment functions. This algorithm, like the *h-adaptivity*, requires a first resolution on a first grid and uses derivatives reconstruction, but instead of only telling where to refine, this information is used to build enrichment functions. We have tested two families of enrichment functions, one inspired from solutions of an advection-diffusion equation, and another one with reconstructed derivatives used as approximations of the first terms of the Taylor expansion of the solution.

The outline of this paper is as follows. Section 2 presents the general stabilized XFEM-like framework where notations are introduced and the scheme is detailed. An interpolation error bound is given. It also explains how to deal with Dirichlet boundary conditions. The next part, section 3, describes our algorithm of *enrichment adaptivity* and gives some numerical results. In section 4 the reader can find a comparison between *h-adaptivity* and enrichment. Preliminary results for a mixed algorithm are shown. The paper ends with concluding remarks, section 5.

2 Numerical scheme: an Enriched Stabilized Finite Element Method

In this section, we describe the general framework of a robust enriched scheme using stabilized finite elements method, inspired by the concepts of the XFEM scheme. We show that a reasonable improvement of the approximation can be obtained with rather little knowledge of the solution. For a good review on enrichment methods, the reader should refer to [8].

2.1 Space discretization and notations

Let Ω be a subset of \mathbb{R}^2 and $\partial\Omega$ its frontier. We restrict ourselves to the two dimensional case for the sake of simplicity of exposure only. Let $\Gamma \subset \partial\Omega$ be the part of the frontier where Dirichlet boundary conditions are imposed. Let \mathcal{T}_h be a conformal mesh on Ω , where h is a characteristic size of the mesh. In this paper, we only consider \mathcal{T}_h to be a triangulation though it has no impact on the scheme formulation. Let n_v be the number of vertexes, they are denoted by $X_i \in \mathbb{R}^2$ for i in $\{1, \dots, n_v\}$. For a given vertex X_i , we define Ω_i as the union of triangles sharing that vertex. We denote by \mathcal{N}_i the n_v standard “hat” functions of a fixed degree p , \mathcal{N}_i being associated with node X_i . These functions \mathcal{N}_i generate the space V_p of continuous piecewise polynomial functions of degree p . Any function v_p in V_p can be written as a unique linear combination of the \mathcal{N}_i functions: $v_p = \sum_{i=1}^{n_s} v_p^i \mathcal{N}_i$ with $v_p^i = v_p(X_i)$.

2.2 Formulation of the problems

We are interested in solving advection-dominated problems. In all this paper, we consider two examples. The first one is a 2D homogeneous advection-diffusion problem

$$\begin{cases} \lambda \cdot \nabla u - \varepsilon \Delta u = 0 & \text{in } \Omega = [0, 1]^2, \\ u(X) = \exp\left(\frac{\lambda \cdot X}{\varepsilon}\right) & \text{on } \Gamma = \partial\Omega. \end{cases} \quad (\text{Pb}_{\text{Adv-diff}})$$

Its exact solution is $u(X) = \exp\left(\frac{\lambda \cdot X}{\varepsilon}\right)$. This toy problem is linear and develops a boundary-layer. In the numerical experiments, we have taken $\lambda = (1, 0)$ and $\varepsilon = 10^{-2}$ except in section 4, figure 10, where $\varepsilon = 10^{-3}$.

The second example is the 1D unsteady Burgers equation

$$\begin{cases} \partial_t u + \partial_x \frac{u^2}{2} - \varepsilon \partial_x^2 u = 0 & \text{in } \Omega = [0, 1]_x \times [0, 1]_t, \\ u|_{\Gamma} = u_D & \text{on } \Gamma. \end{cases} \quad (\text{Pb}_{\text{Burgers}})$$

Here, Γ is $\{0, 1\}_x \times [0, 1]_t \cup [0, 1]_x \times \{0\}_t$. We know that the functions $u(x, t) = b - a \tanh\left(\frac{a(x-bt)}{2\varepsilon}\right)$ are solutions of the Burgers equation; we thus take u_D as the restriction on Γ of such a function for given parameters a , b , and ε . We have taken $a = 1$, $b = \frac{1}{2}$ and $\varepsilon = 10^{-2}$. Time is seen as a space variable so that our 2D spatial code can solve this problem. This problem is non linear and presents an internal layer.

For both problems, we follow the standard finite elements procedure to obtain a variational problem: find $u_h \in V_h$ satisfying equation 1.

$$\begin{cases} a(u_h, v_h) = f(v_h) & \forall v_h \in V_h, \\ u_h|_{\Gamma} = u_D & \text{on } \Gamma. \end{cases} \quad (1)$$

Although both problems can be seen as 2D problems, the extension to 3D problems is straightforward.

2.3 Enrichment in the XFEM spirit

In this section, we define the approximation space V_h we use. The idea of XFEM is to use the set of functions \mathcal{N}_i as a partition of unity to guaranty the conformity of the finite elements space, whatever the type of enrichment provided it is regular enough. Considering an enrichment function ψ , we can define the approximation space V_h as the sum of a polynomial space V_p and of an enrichment space V_e which is spanned by the functions $\psi \mathcal{N}_i$, i.e.

$$V_h = V_p + V_e = V_p + \psi V_p, \quad (2)$$

and any function v_h in V_h writes

$$v_h = \sum_{i=1}^{n_s} (v_p^i + v_e^i \psi) \mathcal{N}_i.$$

It is a very convenient way to write things, but we can have much more flexibility. Indeed, we let the possibility for the function ψ to be different for every

vertex X_i . More precisely, let ψ_i for i in $\{1, \dots, n_s\}$ be n_s different enrichment functions. We define V_e as the space spanned by the functions $\psi_i \mathcal{N}_i$ and then v_h in V_h can be expressed as

$$v_h = \sum_{i=1}^{n_s} (v_p^i + v_e^i \psi_i) \mathcal{N}_i.$$

The choices $\psi_i = \psi - \psi(X_i)$ or $\psi_i = \psi/\psi(X_i)$ span the same approximation space as $\psi_i = \psi$. They are already widely used for their numerical properties, but we do not restrict our choice to these examples. As long as the functions ψ_i are continuous on Ω_i — the union of triangles sharing vertex X_i — and because V_p is a space of continuous functions, every function in V_h is continuous on Ω . The method is thus conformal, that is to say, $V_h \subset H^1(\Omega)$.

A few remarks must be stated now.

1. We are not limited to a single enrichment function, neither are we to keep the polynomial space. A more general formulation of V_h is

$$V_h = \sum_j^{n_e} \psi^j V_p,$$

where n_e is the number of enrichment functions. In the case described previously (see (2)), $n_e = 2$, with $\psi^1 = 1$ (for the polynomial space) and $\psi^2 = \psi$.

2. For writing convenience, the same space V_p is used here for every function ψ^j though it is not mandatory. We could think of different polynomial degrees or spaces generated by different meshes, or restricted to different regions.
3. A big issue of the XFEM is that it is generally not easy to find a basis of V_h . Indeed, it is hard to guarantee, and not always true, that the functions \mathcal{N}_i and $\psi_i \mathcal{N}_i$ are linearly independent. The cure we use for this problem, the simplest one, is to enrich only in the regions where it is needed so that the functions ψ_i are very different from 1. This has the beneficial side effect to spare degrees-of-freedom.

2.4 Interpolation error on the enriched space

In this conforming finite elements framework, it is well known that the total error $\|u_h - u_{ex}\|$ is bounded by the interpolation error

$$\|u_h - u_{ex}\| \leq C_1 \inf_{v_h \in V_h} \|u_{ex} - v_h\|.$$

This inequality tells us that if we can improve the interpolation error of the approximation space, the total error decreases. More precisely, we can obtain estimates on the L^q norms of the error and its gradient, i.e. the norms of the error in W_q^s where s is 0 or 1. Let n be the dimension of Ω , e.g. $n = 2$ or $n = 3$, and assume that $p + 1 - \frac{n}{q} > 0$ holds for a certain q such that $1 \leq q \leq \infty$. In the case of a polynomial approximation space V_p and if $u_{ex} \in W_q^{p+1}(\Omega)$, a standard interpolation error estimate is

$$\inf_{v_p \in V_p} \|u_{ex} - v_p\|_{W_q^s(\Omega)} \leq C_2 h^{p+1-s} |u_{ex}|_{W_q^{p+1}(\Omega)}. \quad (3)$$

For a complete proof, see [3]. As the enriched space V_h contains V_p we have at least

$$\inf_{v_h \in V_h} \|u_{ex} - v_h\|_{W_q^s(\Omega)} \leq C_2 h^{p+1-s} |u_{ex}|_{W_q^{p+1}(\Omega)}. \quad (4)$$

In fact we can improve this inequality.

We decompose V_h into a polynomial space and an enrichment space: $V_h = V_p + V_e$. Under the hypotheses stated above, the space V_p is W_q^1 conformal. Assume the enrichment functions ψ_i are regular enough — this assumption is not too strong as we are interested in boundary-layers in which no singular behavior is generally seen —, the enriched space is thus also included in W_q^1 . For all $v \in W_q^s(\Omega)$ and $v_h \in V_h$, we have thus $\|v - v_h\|_{W_q^s(\Omega)}^q = \sum_{T \in \mathcal{T}_h} \|v - v_h\|_{W_q^s(T)}^q$.

Let \mathcal{I}_h^p be the Lagrange interpolation operator on V_p . For all functions v_e in V_e and for all v in $W_q^{p+1}(\Omega)$, the function $-v_e + \mathcal{I}_h^p(v - v_e)$ belongs to V_h . Following the proof of the inequality (3), see [3] theorem 4.4.20, we obtain

$$\begin{aligned} \inf_{v_h \in V_h} \|u_{ex} - v_h\|_{W_q^s(\Omega)} &\leq \|u_{ex} - v_e + \mathcal{I}_h^p(u_{ex} - v_e)\|_{W_q^s(\Omega)} \\ &= \left(\sum_{T \in \mathcal{T}_h} \|u_{ex} - v_e + \mathcal{I}_h^p(u_{ex} - v_e)\|_{W_q^s(T)}^q \right)^{\frac{1}{q}} \\ &\leq C_2 h^{p+1-s} \left(\sum_{T \in \mathcal{T}_h} |u_{ex} - v_e|_{W_q^{p+1}(T)}^q \right)^{\frac{1}{q}}. \end{aligned}$$

It is important to note that $\sum_{T \in \mathcal{T}_h} |u_{ex} - v_e|_{W_q^{p+1}(T)}^q$ is not equal to $|u_{ex} - v_e|_{W_q^{p+1}(\Omega)}^q$ because v_e does not belong to $W_q^{p+1}(\Omega)$ in general.

The constant C_2 is the same as in (3), and if we take $v_e = 0$, we get back (4). So, the effect of enrichment is mainly to lower the constant.

We remark that if V_h is chosen such that, for a certain $v_e \in V_e$, the term $\inf_{v_e \in V_e} \left(\sum_{T \in \mathcal{T}_h} |u_{ex} - v_e|_{W_q^{p+1}(T)}^q \right)^{\frac{1}{q}}$ is $O(h^n)$, we then have a global interpolation error of order $O(h^{n+p+1-s})$. In other words, a well chosen set of enrichment functions can improve the order of the scheme. For a detailed proof in the case of polynomial enrichment, the reader can refer to [2].

2.5 Stabilization

In the context of advection-dominated problems, it is well known that some stabilization procedure is needed in order to avoid spurious oscillations to develop and destroy the accuracy of the scheme. It has been shown in [1] that a well chosen set of enrichment functions is enough to avoid these oscillations. Unfortunately, very little knowledge of the solution can be assumed *a priori* to choose the set of enrichment functions, like, for example, in the case of the Navier-Stokes equations. This explains our choice to enrich with a very limited number of enrichment functions (1 or 2 in the numerical experiments show in this paper). The aim is only to improve the approximation space. The stabilization properties of enrichment can therefore not be expected and it is not the aim of the present method to provide any improvement of the stability of the scheme. Indeed, if no additional stabilization procedure is used, a non-perfect choice of the enrichment functions would lead to oscillations in the solution. Since our

aim is to obtain a robust scheme able to handle fluid problems in spite of a poor *a priori* knowledge of the solution, a standard stabilization procedure is kept.

The variational formulation given by (1) is modified as follows: find $u_h \in V_h$ such that

$$\begin{cases} a(u_h, v_h) + s(u_h, v_h) = f(v_h) & \forall v_h \in V_h, \\ u_h|_{\Gamma} = u_D & \text{on } \Gamma, \end{cases} \quad (5)$$

with $s(u_h, v_h) = h \int_{\Omega} \mathcal{L}(v_h) \tau \mathcal{L}(u_h)$. The functional \mathcal{L} depends on the problem. For example, in (Pb_{Adv-diff}) we have $\mathcal{L}(u) = \lambda u - \varepsilon \Delta u$ and in (Pb_{Burgers}), $\mathcal{L}(u) = \partial_t u + \partial_x \frac{u^2}{2} - \varepsilon \partial_x^2 u$.

2.6 Boundary conditions

As we are interested in boundary layer problems, we want to enrich a region of Ω intersecting Γ , the part of the frontier where Dirichlet boundary conditions are imposed. We thus have to deal with enrichment functions at the boundary. The equation

$$u_h|_{\Gamma} = u_D \quad \text{on } \Gamma$$

hides the fact that it gives only a linear relation between the coefficients of the polynomial part and those of the enrichment part.

At the discrete level, the system to solve, $a(u_h, v_h) + s(u_h, v_h) = f(v_h)$, has size $\dim(V_h)$ equations for $\dim(V_h)$ unknowns. Let S_{Γ} be the set of vertexes X_i on Γ . If we add the constraints $u_h = u_D$ on S_{Γ} , we now have $\dim(V_h) + \text{card}(S_{\Gamma})$ equations for only $\dim(V_h)$ unknowns. For the system to be invertible, we somehow have to either lower the number of equations by taking a smaller space of test functions, or introduce additional unknowns. The most general way to reformulate this problem to make it invertible is with Lagrange multipliers. For a good understanding of this method, the reader is invited to refer to [4].

Let W_h be a space of Lagrange multipliers. The variational problem (5) becomes: find $(u_h, p_h) \in V_h \times W_h$ such that

$$\begin{cases} a(u_h, v_h) + s(u_h, v_h) - \int_{\Gamma} p_h v_h = f(v_h) & \forall v_h \in V_h, \\ \int_{\Gamma} q_h (u_h - u_D) = 0 & \forall q_h \in W_h. \end{cases} \quad (6)$$

The way the boundary condition is imposed highly depends on the definition of W_h . Let V_0 be the space of functions that are zero on Γ in the following meaning:

$$V_0 = \left\{ v \in V_h \mid \int_{\Gamma} v q_h = 0, \forall q_h \in W_h \right\}.$$

Note that it is generally untrue that $V_0 \subset H_0^1$. Hence imposing weakly Dirichlet boundary condition, as in eq. (6), is not equivalent to imposing them strongly. It is not always easy to find a basis of V_0 , but in case it can be found, eq. (6) can be restated as follows. Assume a function \tilde{u} in V_h satisfying $\int_{\Gamma} q_h (\tilde{u} - u_D) = 0$ for all $q_h \in W_h$ can be constructed. Find $u_0 \in V_0$ satisfying equation (7).

$$a(u_0 + \tilde{u}, v_h) + s(u_0 + \tilde{u}, v_h) = f(v_h) \quad \forall v_h \in V_0. \quad (7)$$

The solution u_h in V_h of problem (6) is then given by $u_h = u_0 + \tilde{u}$. Once a function \tilde{u} is chosen, the problem (7) is invertible. It has $\dim(V_0)$ unknowns and $\dim(V_0)$ equations.

Let S_Γ be the set of vertexes X_i on Γ . If we choose W_h generated by the Dirac δ functions associated with each vertex in S_Γ , the formulation (7) is exactly what we generally do for a standard, non-enriched FEM.

$$W_h = \{\delta_{X_i} | X_i \in S_\Gamma\}.$$

It has the advantage that we know how to construct a basis of V_0 and it is the choice we have made for our numerical experiments. Indeed, we can choose, for example \tilde{u} in V_p defined by

$$\tilde{u}(X_i) = \begin{cases} u_D(X_i) & \text{if } X_i \in S_\Gamma \\ 0 & \text{else.} \end{cases}$$

Recall the functions \mathcal{N}_i are the basis functions of V_p . Suppose X_i is in S_Γ . If $\psi_i(X_i)$ is equal to 0, the function $\phi_i = \psi_i \mathcal{N}_i$ already belongs to V_0 . Assume now $\psi_i(X_i)$ is not equal to 0, the function $\phi_i = (1 - \psi_i/\psi_i(X_i))\mathcal{N}_i$ belongs to V_0 . A basis of V_0 is given by

$$V_0 = (\mathcal{N}_i, \mathcal{N}_i \psi_i | X_i \notin S_\Gamma) + (\phi_i | X_i \in S_\Gamma).$$

Remark 2.1 *For the sake of simplicity, we have only written here the case when the whole domain Ω is enriched, but our code can deal with more general cases, as numerical experiments show.*

Remark 2.2 *In the case of enrichment with more than one function per vertex, the number of constraints to impose, i.e. the dimension of W_h , is still an open question. We have kept the same definition of W_h with two enrichment functions per node as with only one enrichment function, and numerical experiments show that it works in practice. For even more enrichment functions, it seems that constraints must be added, though we did not investigate on this issue.*

2.7 Non-linearity treatment

Let n_0 be the dimension of V_0 and $\{\phi_1, \dots, \phi_{n_0}\}$ a basis of V_0 . Let $A(u_h)$ be in \mathbb{R}^{n_0} such that the i -th component $A(u_h)_i$ of $A(u_h)$ is defined by

$$A(u_h)_i = a(u_h, \phi_i) + s(u_h, \phi_i).$$

Similarly, we define F in \mathbb{R}^{n_0} by $F_i = f(\phi_i)$. The problem (7) is reformulated as: find u_h in V_h (assuming $u_h = \tilde{u} + u_0$, where u_0 in V_0 is the true unknown, see previous paragraph) such that

$$A(u_h) = F. \quad (8)$$

When $a(\cdot, \cdot)$ is non linear in u , $A(\cdot)$ in equation (8) is also non linear. Here we use a Newton algorithm to solve (8), i.e.

$$A(u_h^n) + \partial_u A(u_h^n)(u_h^{n+1} - u_h^n) = F$$

h	dofs	L^∞ error	L^1 error	L^2 error
$7.420 \cdot 10^{-2}$	284	$1.340 \cdot 10^{-16}$	$5.787 \cdot 10^{-18}$	$1.671 \cdot 10^{-17}$
$3.664 \cdot 10^{-2}$	1026	$2.621 \cdot 10^{-16}$	$2.751 \cdot 10^{-18}$	$1.576 \cdot 10^{-17}$
$1.850 \cdot 10^{-2}$	3984	$4.166 \cdot 10^{-16}$	$2.612 \cdot 10^{-18}$	$1.615 \cdot 10^{-17}$
$1.546 \cdot 10^{-2}$	6060	$6.570 \cdot 10^{-16}$	$3.543 \cdot 10^{-18}$	$2.210 \cdot 10^{-17}$
$7.818 \cdot 10^{-3}$	24114	$4.538 \cdot 10^{-15}$	$1.317 \cdot 10^{-17}$	$8.856 \cdot 10^{-17}$

Figure 1: Enrichment with exact solution (Pb_{Adv-diff})

The Jacobian $\partial_u A(u_h^n)$ is approximated by a finite difference

$$\partial_u A(u_h^n) \approx \frac{A(u_h^n + u_\delta) - A(u_h^n)}{\delta},$$

where δ is a parameter which has to be small enough for the method to converge efficiently. The function satisfying the boundary conditions \tilde{u} can be chosen as the initial guess u_h^0 . At each step, we thus have to solve a sparse linear system. Our code uses the sparse direct solver PaStiX [10] to do so.

2.8 How to choose enrichment functions

A theorem proved in [13] says that the global interpolation error is bounded by the sum of local interpolation errors. This tells us that to improve the scheme, we just have to find enrichment functions that are “locally” close to the exact solution.

The most natural way to build an enrichment function is to take something that looks like an exact solution of a problem similar to the one we are trying to solve. It is the idea used in [1] or [14] and even in the DEM [6].

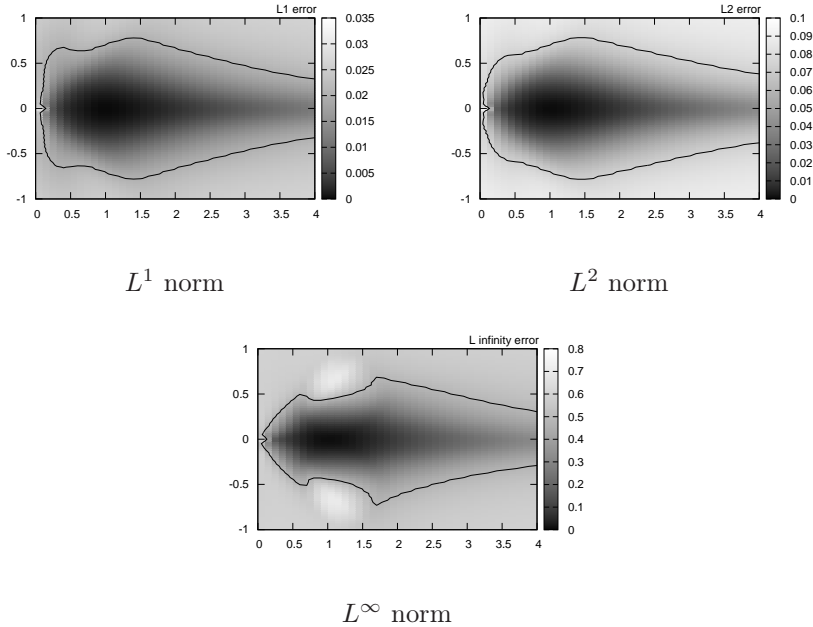
In the case $\varepsilon \ll \|\lambda\|$, let us consider the homogeneous advection-diffusion equation $\lambda \cdot \nabla u - \varepsilon \Delta u = 0$ without taking care of the boundary condition. The function $u(X) = \exp(\lambda \cdot X / \varepsilon)$ is an exact solution with a steep profile that mimics the expected behavior in a boundary layer. Inspired by this example, we enrich at vertex X_i by the function

$$\psi_i(X) = \exp(\beta \cdot (X - X_i)), \quad (9)$$

for a given vector $\beta \in \mathbb{R}^2$. This choice is very natural since there is a boundary-layer-like profile. We are interested in the impact of how we choose β on the error.

It is clear that the choice of vector β is crucial for good approximation properties. For example, in the case of (Pb_{Adv-diff}), if $\beta = \lambda/\varepsilon$, the exact solution u is spanned by V_h . Table 1 shows that, in this case, machine precision is achieved as expected. In all the test cases shown in this section, the number of degrees of freedom (dofs) is exactly twice as large as the number of vertexes since we have enriched the whole computational domain.

We are interested in how these errors vary with respect to β . To answer this question, we have run several experiments on a single mesh, with β in $[0, 4] \times [-1, 1]$ by steps of 0.05, that is to say roughly 3200 experiments. The resulting errors are given by Figure 2. On these pictures, the coordinates are those of the chosen β s and the grey levels describe the error. These simulations

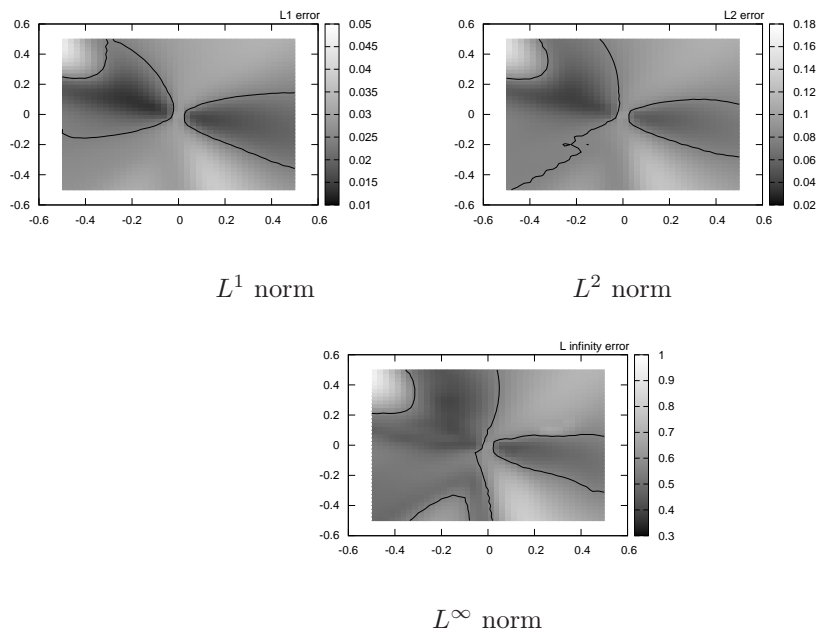
Figure 2: Error according to the choice of β ($\text{Pb}_{\text{Adv-diff}}$)

are all run on a single mesh with 513 vertexes, that is to say 1026 dofs. For comparison, we have also run a simulation with a standard SUPG P^1 scheme on a mesh with 1075 vertexes. The black line on Figure 2 is the isoline of errors given by this SUPG scheme. The “flame” shape of the errors indicate that, as expected, the enrichment gives better results when β has the direction of the strongest gradient. But it also shows that even in cases we don’t know what the optimal value for β is, a rough guess can bring good improvement to the scheme. We have run a similar set of experiment on ($\text{Pb}_{\text{Burgers}}$). We now compute the error of the scheme with β in $[-0.5, 0.5]^2$. The geometry of the error is here more complicated to interpret in terms of good choice for β , but what is clear, once again, is that it has to be roughly in the direction of the gradient of the solution (here $(1, -1/2)$).

Remark 2.3 *We have shown that the chosen stabilized XFEM-like framework is well suited to solve advection-dominated problems. We think that virtually any enrichment function showing a gradient growing in the direction of the highest gradient of the solution can bring improvement to the approximation space (other functions have been tested, such as $\psi(X) = \lambda \cdot X$, $\psi(X) = \lambda \cdot X^2$, etc).*

3 An automated enrichment adaptation

As was shown in section 2.8, the scheme is robust enough to handle enrichment functions very far from an analytical form of the solution, as long as they present a phenomenon that is not well represented by the classical polynomial space.

Figure 3: Error according to the choice of β (Pb_{Burgers})

The major drawback is that enrichment still relies on some *a priori* knowledge of the solution. Inspired from the *h/p-adaptivity*, some attempts have been made in the past few years to develop automated algorithms to adapt the enrichment functions. The term *enrichment adaptivity* or *e-adaptivity* has been proposed in [5] for such algorithms in the context of fracture mechanics. We also have to mention the adaptive variational multiscale method [12] which was designed for elliptic problems and relies on the numerical resolution of local Dirichlet problems on a fine grid.

In this section, we explain our new algorithm that aims at answering two questions: where and with what to enrich?

3.1 Some explanation about mesh adaptation

Since several years, the *h-adaptivity* community has developed efficient methods to answer two questions, very similar to the ones we are trying to answer: where and to what extent should we refine? We will here describe briefly an algorithm — among others — of mesh adaptation. For further details, the reader should refer, for example, to [7]. This algorithm starts with an initial mesh, \mathcal{T}^0 , associated with a polynomial space V_p^0 . Given a mesh \mathcal{T}^n and its associated polynomial space V_p^n , a solution u_p^n in V_p^n is computed. Then an approximation of its Hessian matrix Hu_p^n is reconstructed (see section 3.2). Built on this Hessian, an anisotropic metric field \mathcal{M}_n is defined on the whole domain. The new mesh \mathcal{T}^{n+1} is computed so that every element edge are of length 1 this metric \mathcal{M}_n .

3.2 Reconstruction of derivatives

Standard techniques are used to reconstruct derivatives of u_p belonging to V_p . Recall that Ω_i is the union of triangles sharing vertex X_i . We can define an average of $\nabla u_p(X_i)$ by

$$\nabla u_p(X_i) \approx \frac{1}{|\Omega_i|} \int_{\Omega_i} \nabla u_p = \frac{1}{|\Omega_i|} \sum_{T \subset \Omega_i} \int_T \nabla u_p.$$

Then, we approximate ∇u_p in V_p as

$$\nabla u_p = \sum_i \nabla u_p(X_i) \mathcal{N}_i.$$

Since we have an approximation of ∇u_p in V_p , we follow the same procedure to reconstruct derivatives of ∇u_p , i.e. second order derivatives of u_p . In the same way, we can build higher order derivatives of u_p .

3.3 Algorithm

The algorithm we propose takes two steps. As for the *h-adaptivity* algorithm described in §3.1, we need a first approximation in some initial approximation space V_p . This gives us a solution u_p and we can approximate ∇u_p and the Hessian Hu_p . The same error estimator can be used to know where to enrich, though in our code we use something simpler: we enrich where the norm of ∇u_p is higher than a specified threshold. We now have to tune the enrichment functions according to the information contained in the approximated derivatives of u_p . In sections 3.4 and 3.5, we propose two ways to do so, respectively by an exponential and a polynomial enrichment. The second step of the algorithm is the resolution on the enriched space.

Remark 3.1 *If, for a specific problem, some information on the solution can be known a priori, such as an analytical form of the boundary layer, of course it can be used in combination or instead of the initial resolution on V_p in order to properly enrich.*

3.4 Exponential enrichment

One of the enrichment functions we want to test is inspired from the example given in section 2.8. We still enrich with an exponential, but now the vector β in (9) depends on the vertex X_i considered and is noted β_i . As pointed out in the discussion of section 2.8, the important thing is that β_i be in the direction of the gradient of the solution. The simplest thing that comes into mind is to directly take β_i as an approximation of the gradient of u_p at points X_i , that is $\beta_i = \nabla u_p(X_i)$. The enrichment functions then write

$$\psi_i(X) = \exp(\beta_i \cdot (X - X_i)).$$

Numerical experiments. Figures 4 and 5 show, respectively, the error convergence of (Pb_{Adv-diff}) and (Pb_{Burgers}) on uniformly refined triangular unstructured meshes. Both show improvement of the error compared to a classical

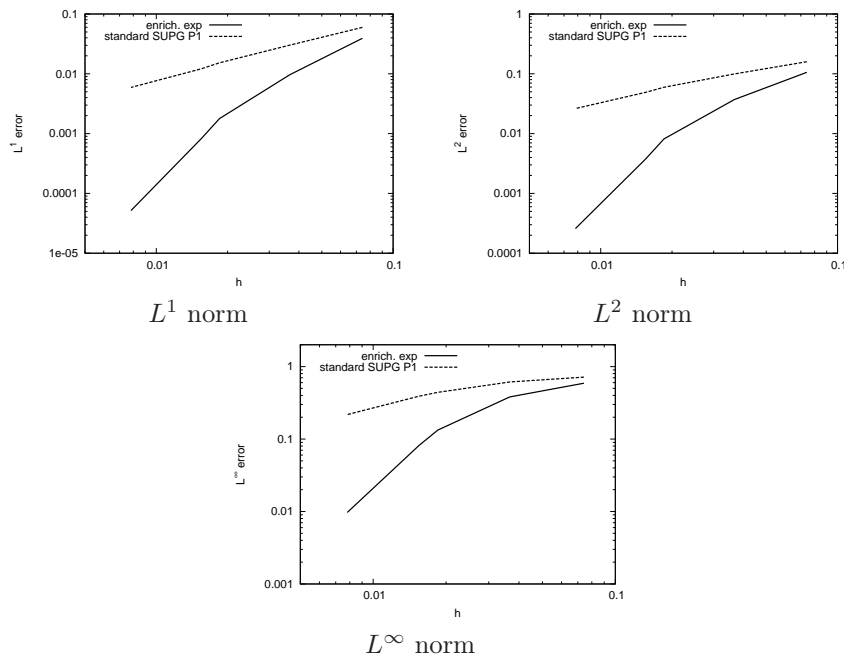


Figure 4: Error convergence with automated choice of exponential enrichment ($\text{Pb}_{\text{Adv-diff}}$)

stabilized P^1 scheme. Though the impact of enrichment for ($\text{Pb}_{\text{Burgers}}$) is not as good as for ($\text{Pb}_{\text{Adv-diff}}$), these experiments show that exponential enrichment can also improve the resolution of internal layers. Table 6 show the evolution of the number of degrees of freedom. With our choice of threshold, the increase remains around 10%.

3.5 Polynomial enrichment

In the traditional p -adaptivity, the formal order of accuracy of the scheme is increased. This requires a lot of degrees of freedom. For example in 2D, let n_s be the number of vertexes in the mesh. A P^1 approximation space has dimension n_s while a P^2 space has dimension about $4n_s$. This costs a lot, but maybe all of these functions are not needed. The enrichment method may be less expensive: if we have a single enrichment function, and assuming we enrich the whole domain, $\dim(V_h) = 2n_s$. If we have two enrichment functions, $\dim(V_h) = 3n_s$. We could enrich a P^1 space with up to 3 enrichment functions at roughly the same cost than taking a P^2 approximation space.

It is difficult to find an analytical solution to enrich with, and even in cases it can be done, it is very problem-dependent. Our idea is to use a first approximation on V_p , then to reconstruct the derivatives of the solution as described in §3.2. From these derivatives, we can build polynomial functions corresponding to approximations of terms in the Taylor expansion of the solution.

For example, we know, see e.g. [7], that the main part of the interpolation error comes first from a bad approximation of the Hessian matrix. We

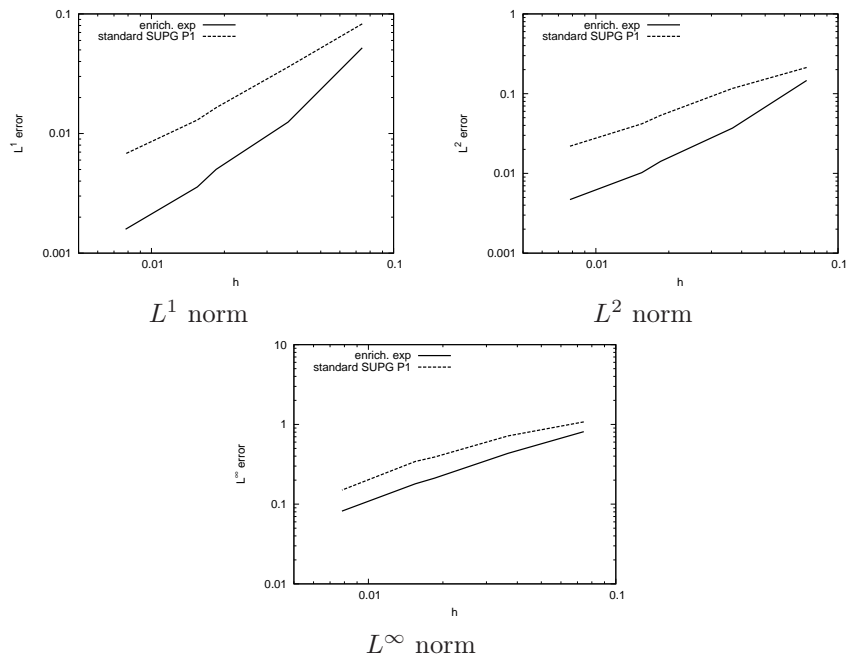
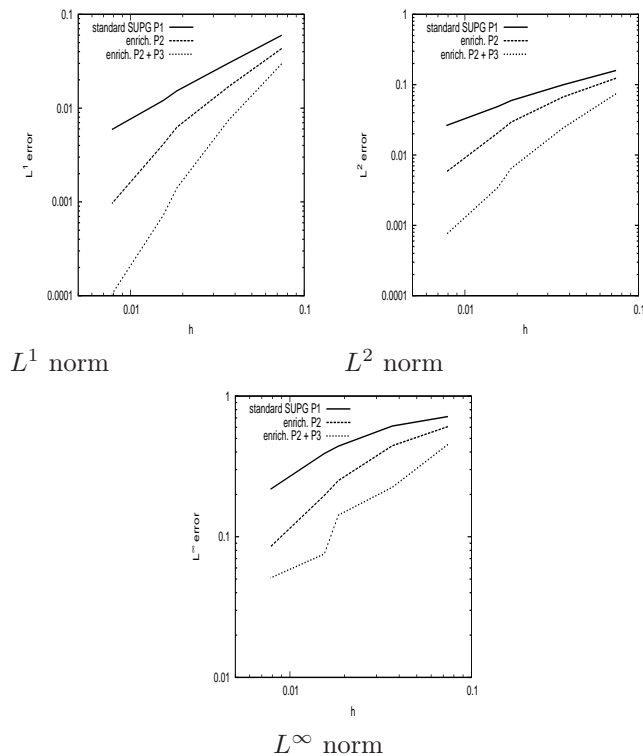


Figure 5: Error convergence with automated choice of exponential enrichment ($\text{Pb}_{\text{Burgers}}$)

h	dofs	vertexes
$7.420 \cdot 10^{-2}$	174	142
$3.664 \cdot 10^{-2}$	592	513
$1.850 \cdot 10^{-2}$	2202	1992
$1.546 \cdot 10^{-2}$	3324	3030
$7.818 \cdot 10^{-3}$	12968	12057

Figure 6: Evolution of the number of degrees of freedom ($\text{Pb}_{\text{Adv-diff}}$)

Figure 7: Error convergence with adapted polynomials (Pb_{Adv-diff})

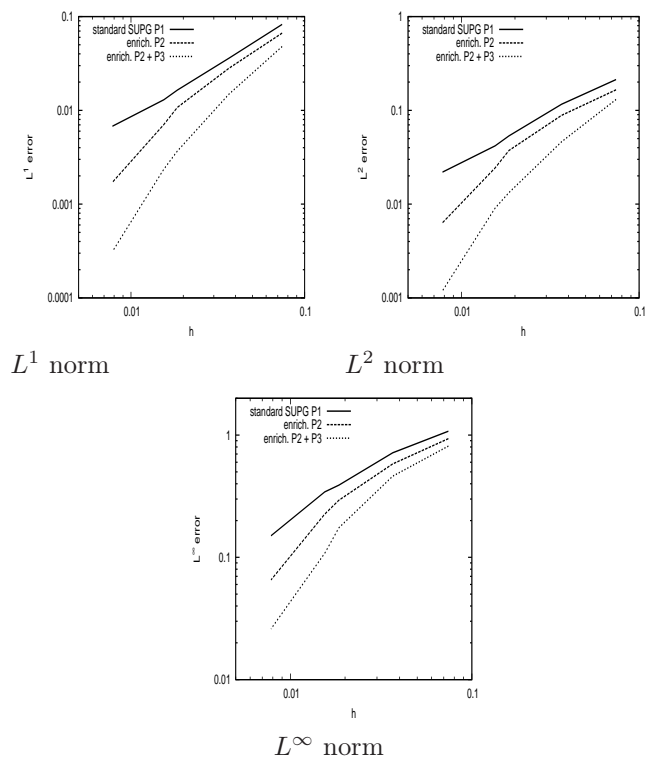
can thus use, as an enrichment function, the term of the Taylor development corresponding to the hessian,

$$\psi_i(X) = \frac{1}{2}(X - X_i)^T H u_p(X - X_i).$$

In all the experiments, this enrichment is referred to as “P2”. In the same fashion, we can enrich with higher terms of the Taylor development. The term with third order derivatives is referred to as “P3”.

Numerical experiments. Figures 7 and 8 show respectively the error convergence for (Pb_{Adv-diff}) and for (Pb_{Burgers}) on uniformly refined unstructured triangular meshes. Two enrichments are tested. First, the P2 described previously. Second, each vertex is enriched with two functions, the P2 and the P3 — referred to as P2 + P3 on the figures. On both problems, the enrichment improves the solution resulting in a lower error. The good results of the enrichment P2 + P3 are to be noted, as it is only a slightly more expensive than the P2 enrichment.

Remark 3.2 • We have also tried, in the case of a single enrichment function, to enrich with a single P³ function taking into account first, second and third order derivatives, but it didn’t work as well as expected for reasons that are still obscure.

Figure 8: Error convergence with adapted polynomials (Pb_{Burgers})

- *It is not clear to what extent the information taken from high order derivatives stays relevant. We have tried with second and third order derivatives and the method achieved good results, but what happens for derivatives of order 4 and higher ? We expect the reconstruction to be less and less accurate at each order. Thus, for very high order enrichment, an iterative procedure must be considered. The first few derivatives are computed on the initial solution u_p , then a new solution on the enriched space can be computed, higher order derivatives reconstructed and the space are again enriched with higher order terms. If needed, these operations must be repeated.*

4 Towards h/e -adaptivity

The main weakness of our algorithm is that it relies on a first computation on V_p . The h -adaptation algorithm also does, but it uses **less** information from this initial resolution. Indeed, the construction of the metric field takes somehow the absolute value of the Hessian matrix, whereas we use the full matrix to enrich. For example, think of a problem where there is an under-resolved singularity. The solution on the initial mesh will be very bad near the singularity, maybe with oscillations, etc. An error indicator can then prescribe to refine this region, but the information will be totally irrelevant to build enrichment functions. In other words, the solution has to be accurate enough to provide relevant information for our algorithm. Also, as enrichment requires additional degrees-of-freedom, it is important to know if they would not be better used in mesh refinement.

With the help of the mesher *bamg* [9], we have run a mesh adaptation algorithm to solve (Pb_{Adv-diff}). At each iteration of the loop, the error threshold for refinement is lowered, so that we observe a convergent behavior. This is the curve “ $P1$ adapted refinement” in Figures 9 and 10. For comparison, we have also plotted the error obtained with a homogeneous refinement on the whole domain (curve “ $P1$ homogeneous refinement” on the same Figure). As expected, the adapted refinement gives better results. We have also plotted the $P2$ enriched space on uniformly refined meshes. It is interesting to see that on coarse grids, it is a better strategy to refine the mesh than to enrich the space. But once the grid is fine enough for the derivative reconstruction to be accurate, degrees-of-freedom should better be used to enrich the space. At each iteration of the adaptation procedure, we can also enrich on the adapted mesh, taking advantage of both point-of-views. This gives the last curve of Figure 9 “enrich. $P2$ adapted refinement”. A good, yet simple, strategy would be to refine the mesh up to a certain threshold, then enrich only on the last mesh.

We have also run the same experiment, but with $\varepsilon = 10^{-3}$ in (Pb_{Adv-diff}). Figure 10 shows even more clearly that on coarse grids, it is better to refine the grid than to enrich, but very good improvement can be expected as soon as the mesh is fine enough.

In order to develop a truly h/e -adaptivity algorithm, the next step is to have an error estimator that takes the enrichment part into account. This is expected to lead to a faster convergence in the adaptation process.

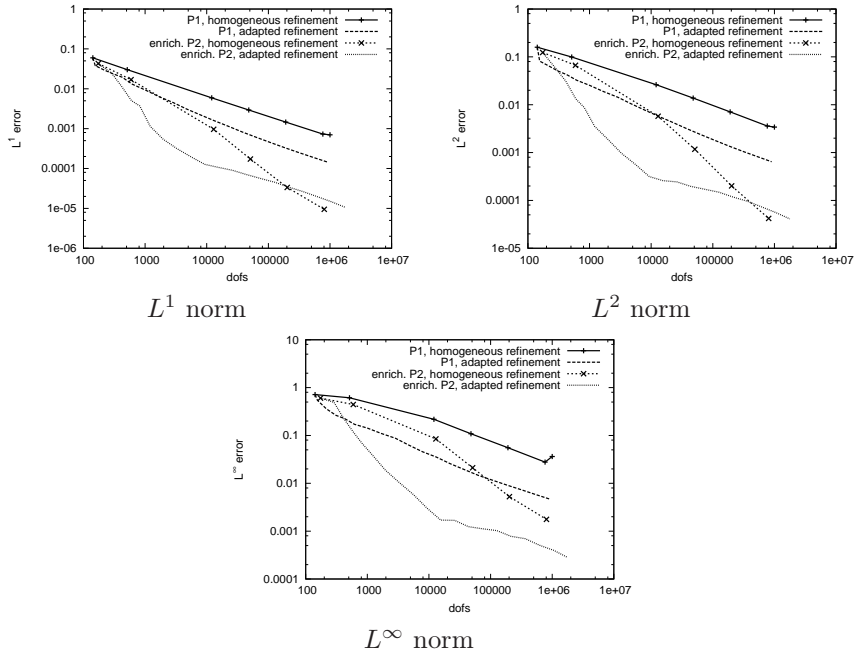


Figure 9: Enrichment vs. Mesh refinement ($Pb_{Adv-diff}$)

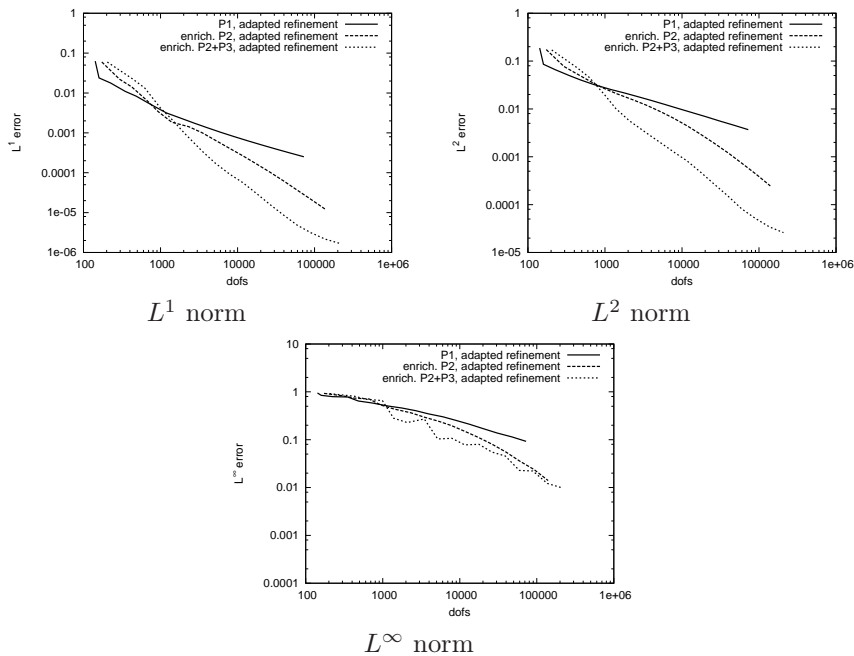


Figure 10: Enrichment on adaptively refined meshes ($Pb_{Adv-diff}$)

5 Concluding remarks

In this paper, we have shown that the stabilized Enriched Finite Element Method is an appropriate framework for enrichment thanks to its robustness. We have also proposed a new automated enrichment algorithm that revealed very efficient on test cases presenting a boundary layer, even on non-linear equations. Although our goal is to deal with advection-dominated problems, we think that the proposed algorithm can be used in a more general multiscale context. Some improvement remains to be made in order to fully incorporate enrichment in the *h-adaptation* procedure. The application of our algorithm to Navier-Stokes equations is a natural follow up of the present work, and is being investigated.

Acknowledgments

The authors have been supported by the FP7 Advanced Grant #226316 "AD-DECCO".

References

- [1] S. Abbas, A. Alizada, and T.-P. Fries. The XFEM for high-gradient solutions in convection-dominated problems. *International Journal for Numerical Methods in Engineering*, 82(8):1044–1072, May 21 2010.
- [2] C. Anitescu and U. Banerjee. Approximation properties of the generalized finite element method. *Adv. Comput. Math.*, 34:369–390, May 2011.
- [3] S. C. Brenner and L. R. Scott. *The mathematical theory of Finite Element Methods, second edition*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.
- [4] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
- [5] M. Dufloot and S. Bordas. A posteriori error estimation for extended finite elements by an extended global recovery. *International Journal for Numerical Methods in Engineering*, 76(8):1123–1138, 2008.
- [6] C. Farhat, I. Kalashnikova, and R. Tezaur. A higher-order discontinuous enrichment method for the solution of high Peclet advection-diffusion problems on unstructured meshes. *International Journal for Numerical Methods in Engineering*, 81(5):604–636, JAN 29 2010.
- [7] P. Frey and F. Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):5068 – 5082, 2005. Unstructured Mesh Generation.
- [8] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.

-
- [9] F. Hecht. Bidimensional anisotrope mesh generator (bamg).
http://pauillac.inria.fr/cdrom_a_graver/www/bamg/eng.htm.
- [10] P. Hénon, P. Ramet, and J. Roman. PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems. *Parallel Computing*, 28(2):301–321, Jan. 2002.
- [11] I. Kalashnikova, R. Tezaur, and C. Farhat. A discontinuous enrichment method for variable-coefficient advection-diffusion at high pécelet number. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2010.
- [12] M. G. Larson and A. Malqvist. Adaptive variational multiscale methods based on a posteriori error estimation: Energy norm estimates for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 196(21-24):2313 – 2324, 2007.
- [13] J. Melenk and I. Babuska. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289 – 314, 1996.
- [14] T. Sheu, S. Tsai, and M. Wang. A monotone finite element method with test space of Legendre polynomials. *Computer Methods in Applied Mechanics and Engineering*, 143(3-4):349–372, APR 30 1997.



**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

351, Cours de la Libération
Bâtiment A 29
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399