Document downloaded from:

http://hdl.handle.net/10251/88299

This paper must be cited as:

Nadal, E.; Leygue, A.; Chinesta, F.; Beringhier, M.; Ródenas García, JJ.; Fuenmayor Fernández, FJ. (2015). A separated representation of an error indicator for the mesh refinement process under the proper generalized decomposition framework. Computational Mechanics. 55(2):251-266. doi:10.1007/s00466-014-1097-y.

The final publication is available at

http://dx.doi.org/10.1007/s00466-014-1097-y

Copyright Springer Verlag (Germany)

Additional Information

A separated representation of an error indicator for the mesh refinement process under the Proper Generalized Decomposition framework

E. Nadal[1], A. Leygue[1], F. Chinesta[1] M. Beringhier[2]
J.J. Ródenas[3], F.J. Fuenmayor[3]

[1] *GEM, UMR CNRS-Centrale Nantes*
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France
{Enrique.Nadal;Adrien.Leygue;Francisco.Chinesta}@ec-nantes.fr

[2] *Institut P', ENSMA, UPR CNRS*
Département Physique et Mécanique des Matériaux
1 Avenue Clément Ader, BP 40109, 86961 Futuroscope cedex, France
marianne.beringhier@ensma.fr

[3] *Centro de Investigación en Ingeniería Mecánica*
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain
{jjrodena;ffuenmay}@mcm.upv.es

Abstract

Today industries do not only require fast simulation techniques but also verification techniques for the simulations. The Proper Generalized Decomposition (PGD) has been situated as a suitable tool for fast simulation for many physical phenomena. However, so far, verification tools for the PGD are under development. The PGD approximation error mainly comes from two different sources. The first one is related with the truncation of the PGD approximation and the second one is related with the discretization error of the underlying numerical technique. In this work we propose a fast error indicator technique based on recovery techniques, for the discretization error of the numerical technique used by the PGD technique, for refinement purposes.

Keywords: Verification; Error estimation; Proper Generalized Decomposition; Adaptive Mesh Refinement

1 Introduction

High technology industries require techniques for simulation involving a considerable amount of parameters, such as material properties, geometrical parameters, etc. Well-established mesh-based simulation techniques, are not able to deal into this new framework since they would require a complete analysis for each set of parameters. A powerful alternative to those traditional methods is the Proper Generalized Decomposition (PGD) [1,2], based on the use of a separated representation strategy. This technique is able to evaluate a solution for all the parametric space circumventing the so called *curse of dimensionality*, typical of mesh-based techniques. The PGD technique has been successfully applied to a variety of different problems [3–5] showing a considerable improvement in computational speed and making possible to perform in real time and in light devices, such as tables or smartphones, complex calculations which, so far, were only possible in expensive computers.

The PGD technique provides a numerical approximation to the exact solution. Thus, the technique also requires verification procedures in order to guarantee the quality of the solution. Under the PGD framework, we are facing two main sources of error. The first one is due to the truncation of the separated representation of the solution. Error indicators evaluating this source of error have been already successfully presented [6–8]. The second source of error is due to the discretization used to evaluate each of the functions of the separated representation of the solution. Each one of these functions is obtained via a numerical approach, i.e. the Finite Element Method, for which several error indicators are available in literature [9–12].

In this contribution we propose a technique which is able to provide a fast local error indicator into each discretization space, separately, that can be used to locally adapt the mesh. The mesh adaptation is a process that will be carried out while obtaining the PGD solution. In some situations the PGD solver is used off-line in order to create virtual charts. However, in other situations it is used on-line as a solver of multi-parametric problems or problems with distorted geometries as in the case of plates or shells [13,14] or flows into laminates [15] in which space separation (in-plane out-of-plane) is required. In these situations the space domain is represented as a tensor product (2D + 1D) obtaining 3D solutions at the cost of 2D solutions. This increases the efficiency with respect to standard techniques and avoids the use of highly distorted 3D meshes. Therefore, it is interesting to evaluate fast refinement indicators to reduce the computational cost to obtain the solution for a certain level of accuracy. When the mesh is properly adapted, robust and more expensive verification techniques [8,7] could eventually be used. In the proposed method, the quantity to evaluate the error has been taken as the difference, in L_2-norm, between the exact flux field and the flux obtained with the numerical approach. However, the exact solution is not available in general. Then, using the ideas presented by Zienkiewicz and Zhu [16], the exact flux field will be substituted by an improved solution. This improved solution will be obtained with a smoothing process applied to the flux provided by the numerical solution.

In the next section we will define the steady-state advection-diffusion equation which will serve as a model to illustrate the error indication technique. Section 3 is devoted to the development of the proposed error indication technique and the smoothing process for the flux field. Section 4 shows the results obtained which successfully validate the proposed technique, and finally in section 5 we present some final remarks.

2 Problem statement. Separated representation

2.1 The model

In order to describe the proposed strategy of mesh adaptation in parametric separated representations we consider the steady-state advection-diffusion equation:

$$\mathbf{v} \cdot \nabla u - k \; \Delta u = f \tag{1}$$

with $u(\mathbf{x})$, $\mathbf{x} \in \Omega$, $\Omega = \Omega_x \times \Omega_y = (0, L) \times (0, H)$, $\mathbf{v} = (v_x, v_y)$ constant in Ω, and the Dirichlet boundary conditions $\bar{u}$ over the boundary $\partial\Omega$.

The weighted residual integral form related to Eq. (1) reads

$$\int_{\Omega_x \times \Omega_y} u^* \left(v_x \frac{\partial u}{\partial x} + v_y \frac{\partial u}{\partial y} - k \frac{\partial^2 u}{\partial x^2} - k \frac{\partial^2 u}{\partial y^2} - f \right) \, \mathrm{dx} \, \mathrm{dy} = 0 \quad \forall u^* \in V = V_x \otimes V_y \tag{2}$$

where $V_x = \left\{ u_x^* | u_x^* \in H^1(\Omega_x), u_x^*|_{\Gamma_x^D} = 0 \right\}$, $V_y = \left\{ u_y^* | u_y^* \in H^1(\Omega_y), u_y^*|_{\Gamma_y^D} = 0 \right\}$ and Γ_x^D and Γ_y^D are the corresponding Dirichlet boundaries.

As shown in (2), we have used a space domain represented as a tensor product. This is a general case in which we have separate all spacial dimensions. The PGD permits to solve this kind of problems on a separated space representation framework increasing the efficiency of the method with respect to standard space representations due to the fact that the PGD avoids the use of highly distorted 3D meshes.

2.2 Separated representation involving the space coordinates

The PGD solution of N terms is sought in the separated form

$$u_N(x,y) = \sum_{i=1}^{N} X_i(x)Y_i(y) \tag{3}$$

Note that we assume that with N the PGD solution has converged.

At enrichment step n of the PGD algorithm, we have already computed the approximation

$$u^{n-1}(x,y) = \sum_{i=1}^{n-1} X_i(x)Y_i(y) \tag{4}$$

and we wish to obtain the next one, i.e.

$$u^n(x,y) = u^{n-1}(x,y) + X_n(x)Y_n(y) = \sum_{i=1}^{n-1} X_i(x)Y_i(y) + X_n(x)Y_n(y) \tag{5}$$

An alternating direction iterative scheme is then used to solve the non-linear problem for $X_n(x)$ and $Y_n(y)$. At iteration p, we must compute $X_n^p(x)$ from $Y_n^{p-1}(y)$, and then $Y_n^p(x)$ from $X_n^p(x)$. Let us detail the first step. At this stage, the approximation reads

$$u^n(x,y) = \sum_{i=1}^{n-1} X_i(x)Y_i(y) + X_n^p(x)Y_n^{p-1}(y) \tag{6}$$

where all functions except $X_n^p(x)$ are known. Selecting $u^*(x,y) = X_n^*(x)Y_n^{p-1}(y)$ for the test function and introducing (6) into (2), we obtain

$$\begin{gathered}\int_{\Omega_x \times \Omega_y} X_n^* Y_n^{p-1} \\ \left(v_x \frac{dX_n^p}{dx} Y_n^{p-1} + v_y X_n^p \frac{dY_n^{p-1}}{dy} - k \frac{d^2X_n^p}{dx^2} Y_n^{p-1} - k X_n^p \frac{d^2Y_n^{p-1}}{dy^2} \right) \text{ dx dy} = \\ - \int_{\Omega_x \times \Omega_y} X_n^* Y_n^{p-1} \\ \sum_{i=1}^{n-1} \left(v_x \frac{dX_i}{dx} Y_i + v_y X_i \frac{dY_i}{dy} - k \frac{d^2X_i}{dx^2} Y_i - k X_i \frac{d^2Y_i}{dy^2} \right) \text{ dx dy} + \\ \int_{\Omega_x \times \Omega_y} X_n^* Y_n^{p-1} f \text{ dx dy}\end{gathered} \tag{7}$$

Integration over Ω_y then yields the weighted residual form of a one-dimensional problem for the unknown function $X_n^p(x)$, which involves known coefficients

α^x, β^x, γ^x, δ_i^x, χ_i^x, υ_i^x and ξ^x whose definition can be found in [17].

$$\int_{\Omega_x} X_n^* \left(\alpha^x \frac{d^2 X_n^p}{dx^2} + \beta^x \frac{dX_n^p}{dx} + \gamma^x X_n^p \right) \,\mathrm{dx} =$$

$$- \int_{\Omega_x} X_n^* \sum_{i=1}^{n-1} \left(\delta_i^x \frac{d^2 X_i}{dx^2} + \chi_i^x \frac{dX_i}{dx} + \upsilon_i^x X_i \right) \,\mathrm{dx} + \int_{\Omega_\mathrm{x}} \mathrm{X_n^* \xi^x \,dx} \qquad (8)$$

The corresponding strong form is a one-dimensional convection-diffusion-reaction equation with a source term, for which quasi-optimal stabilization methods exist [18] in case they where needed. The interested reader can refer to [19] for a deeper analysis of this topic and numerical tests proving the performance of this approach. Note that in this case we have chosen to separate both spatial coordinates, but the developments that follows are also valid in the case we do not separate the space coordinates or in the case we perform a in-plane out-of-plane separation.

2.3 Parametric solution

Following our former works (see [20,5,3,21] and the references therein) one could consider model parameters (as well as boundary conditions, initial conditions or geometrical parameters) as extra-coordinates, in order to calculate a parametric solution. Thus, for example, in order to compute the solution of problem (1) for any value of the diffusion coefficient $k \in \Omega_k$, it suffices considering k as another coordinate (like x or y) and looking for $u(x, y, k)$ as described in [17]. Thus, instead of (3) we will obtain the parametric solution:

$$u_N(x, y, k) = \sum_{i=1}^{N} X_i(x) Y_i(y) K_i(k) \qquad (9)$$

2.4 Discretization

When using a discretization technique on the associated weak form, functions $X_i(x)$ and $Y_i(y)$, requires a continuous approximation. The simplest choice consists of using piecewise linear functions on a mesh of Ω_x and Ω_y respectively.

On the other hand, in the parametric case, the simplest choice for approximating functions $K_i(k)$ consists of using piecewise constant functions on a mesh of the parametric domain Ω_k.

The number of elements of the different meshes will be noted by $\mathcal{M}_x$, $\mathcal{M}_y$

and $\mathcal{M}_k$, the number of the nodes $\mathcal{N}_x$, $\mathcal{N}_y$ and $\mathcal{N}_k$, and a particular element in each one by Ω_x^r, Ω_y^s and Ω_k^t respectively. Thus the discrete approach to the separated representation solution given in (9) is denoted in the following manner:

$$u_N^h(x,y,k) = \sum_{i=1}^{N} X_i^h(x) \cdot Y_i^h(y) \cdot K_i^h(k) \tag{10}$$

From now on N is fixed. For the sake of notation simplicity the indexes N and h will be omitted from now on.

3 Error measures and smoothing process

3.1 Definition of the error measures

As mentioned before, the objective of this contribution is to propose a novel method to determine the zones of each domain Ω_x and Ω_y which require a finer discretization. In order to introduce the error measures, in this section we will assume that the analytical solution is known. Traditionally, the mesh adaptivity have been driven by the local error estimator. Different techniques have been proposed in literature [9,10,16]. These techniques were developed for the Finite Element Method where the local error indicator is evaluated by integrating the residual equation, or some equivalent quantity, into the domain of each element. We will base our work on the recovery-type error estimation techniques which represent the techniques preferred by the practitioners. For the refinement process within the PGD framework, we are interested in decoupling the error corresponding to each discretization space. Because of this reason, in this work we propose the use of the solution gradient difference as error indicator according to the following expression:

$$(E^x(k))^2 = \int_{\Omega_x} \int_{\Omega_y} (q_x - q_x^h)^2 \,\mathrm{dx}\,\mathrm{dy} \tag{11}$$

for the flux in x direction and in a similar way the one involving the y-coordinate

$$(E^y(k))^2 = \int_{\Omega_x} \int_{\Omega_y} (q_y - q_y^h)^2 \,\mathrm{dx}\,\mathrm{dy}. \tag{12}$$

where q_x and q_y represent the fluxes provided by the analytical solution obtained with the N terms of the PGD representation of the exact solution for any value of the diffusion coefficient k. q_x^h and q_y^h represent the PGD solution obtained with N terms of the discrete approach. Note that with the previous definitions we have decoupled the error contributions in x and y directions, allowing for an indicator to guide the refinement process in each one.

It is interesting to notice that, for instance, the quantity E^x does not indicate the total error of the elements in x direction, but only the part of the error due to the discretization in Ω_x that is affecting to the elements in Ω_x. That is, E^x does not take into account the part of the error due to the other discretizations (y or k) in an element in the Ω_x. Traditional residual-based approaches provide high quality error estimations both, local and global. However, the standard implementations provide a result that couples the discretization error in all domains that is not the one we need to guide the refinement process. Despite, the proposed quantity (11), does not provide the total error in an element, it provides the necessary information to guide the refinement process: the error for each element $r \in \Omega_x$ due to the discretization in Ω_x. The same study can be performed for elements $s \in \Omega_y$.

Consider that under the PGD framework we have a separated representation of the solution, then we will also obtain a separated representation of the fluxes for the analytical solution:

$$q_x(x,y,k) = \frac{\partial u(x,y,k)}{\partial x} = \sum_{i=1}^{N} \frac{dX_i(x)}{dx} Y_i(y) K_i(k) \tag{13}$$

$$q_y(x,y,k) = \frac{\partial u(x,y,k)}{\partial y} = \sum_{i=1}^{N} X_i(x) \frac{dY_i(y)}{dy} K_i(k) \tag{14}$$

and for the discrete one that is discontinuous across the elements of both meshes, Ω_x and Ω_y.

$$q_x^h(x,y,k) = \frac{\partial u(x,y,k)}{\partial x} = \sum_{i=1}^{N} \frac{dX_i^h(x)}{dx} \cdot Y_i^h(y) \cdot K_i^h(k) \tag{15}$$

$$q_y^h(x,y,k) = \frac{\partial u(x,y,k)}{\partial y} = \sum_{i=1}^{N} X_i^h(x) \cdot \frac{dY_i^h(y)}{dy} \cdot K_i^h(k) \tag{16}$$

Due to the separate representation, both integrals (11) and (12) can be carried out very efficiently because $(q_x - q_x^h)^2$ and $(q_y - q_y^h)^2$ have a separated forms that allows writing the double integral as a sum of products of two one-dimensional integrals. Both functions $(q_x - q_x^h)^2$ and $(q_y - q_y^h)^2$, after performing a post-compression using the separated representation constructor (see chapter 3 in [17]) can be written as

$$(q_x - q_x^h)^2 \;=\; \sum_{i=1}^{Q^x} \alpha_i^x F_i^x(x) \cdot G_i^x(y) \cdot H_i^x(k) \tag{17}$$

$$(q_y - q_y^h)^2 \;=\; \sum_{i=1}^{Q^y} \alpha_i^y F_i^y(x) \cdot G_i^y(y) \cdot H_i^y(k) \tag{18}$$

where functions $F_i^{\cdot}$, $G_i^{\cdot}$ and $H_i^{\cdot}$ are normalized and Q^x and Q^y (not necessary equal) are the number of the corresponding modes for each quantity. Note that, by construction, if we do not use the post-compression, expressions (17) and (18) will have $3N^2$ terms, which in practical situations implies an excessive cost for the error analysis. Since the difference between the analytical flux and the numerical approximation is dominated by the term with the derivatives, the post-compression technique is very effective since it is able to reduce number of terms to a value close to N. Thus, the error analysis is simplified. The computational cost associated to the post-compression procedure is small, in comparison to solve the main problem, since no differential operator is involved.

Now, we define the following normalized quantities at the element level. Note that $e^{\cdot}_{\cdot,\cdot}$ defines the error indicator in the elements of the corresponding coordinate, while $\kappa^{\cdot}_{\cdot}$ are only weighting terms associate to the other coordinates:

$$\begin{cases} e_{i,r}^{q_x} = \int_{\Omega_x^r} F_i^x(x)\ \mathrm{dx} \\ \kappa_{i,s}^{q_x} = \int_{\Omega_y^s} G_i^x(y)\ \mathrm{dy} \\ \kappa_i^{q_x} = H_i^x(k) \end{cases} \tag{19}$$

and

$$\begin{cases} \kappa_{i,r}^{q_y} = \int_{\Omega_x^r} F_i^y(x)\ \mathrm{dx} \\ e_{i,s}^{q_y} = \int_{\Omega_y^s} G_i^y(y)\ \mathrm{dy} \\ \kappa_i^{q_y} = H_i^y(k) \end{cases} \tag{20}$$

Using the integration scheme allowed by the separated representation, we obtain the following quantities

$$(E_r^x(k))^2 = \sum_{i=1}^{Q^x} \left(\alpha_i^x e_{i,r}^{q_x} \left(\sum_{s=1}^{\mathcal{M}_y} \kappa_{i,s}^{q_x} \right) \kappa_i^{q_x} \right) \tag{21}$$

$$(E_s^y(k))^2 = \sum_{i=1}^{Q^y} \left(\alpha_i^y e_{i,s}^{q_y} \left(\sum_{r=1}^{\mathcal{M}_x} \kappa_{i,r}^{q_y} \right) \kappa_i^{q_y} \right) \tag{22}$$

which represent the indicator of the error at each element for each value of k. However, since the PGD solution is valid for any $k \in \Omega_k$, we are not interested in obtaining a different mesh for each value of the parameter k but a single mesh valid for any $k \in \Omega_k$. Therefore, the refinement indicator at each element will the evaluated from expressions (21) for elements in Ω_x and from (22) for elements in Ω_y, such as at each element r or s the error measure is the maximum value obtained for the range of values of k according to the following expressions:

$$(\mathcal{E}_r^{q_x})^2 = \max_{k \in \Omega_k} \sum_{i=1}^{Q^x} \left(\alpha_i^x e_{i,r}^{q_x} \left(\sum_{s=1}^{\mathcal{M}_y} \kappa_{i,s}^{q_x} \right) \kappa_i^{q_x} \right) \tag{23}$$

$$(\mathcal{E}_s^{q_y})^2 = \max_{k \in \Omega_k} \sum_{i=1}^{Q^y} \left(\alpha_i^y e_{i,s}^{q_y} \left(\sum_{r=1}^{\mathcal{M}_x} \kappa_{i,r}^{q_y} \right) \kappa_i^{q_y} \right) \tag{24}$$

This value can be used for driving the mesh adaptivity process as it accounts for the information of all modes and also the worst situation related with the value of the parameter k. Note that the distribution of the error in each discretization space is defined by $e_{i,r}^{q_x}$ and $e_{i,s}^{q_y}$, respectively. The other terms only weight the error indicator depending on the value of the corresponding parameter.

Additionally, following with the previous idea, we consider important to remark that the measures $\mathcal{E}_r^{q_x}$ or $\mathcal{E}_s^{q_y}$ do not indicate the total error at the element r or s respectively, but the error at element r or s due to the discretization in their domain, Ω_x or Ω_y, respectively. This provides the necessary information to guide the refinement process. Values $\mathcal{E}_r^{q_x}$ or $\mathcal{E}_s^{q_y}$ are not sensible to the discretization in the other dimensions.

3.2 Error indicator

In the previous section we have introduced an error indicator useful for mesh adaptivity, however it is based on the use of the PGD exact analytical solution. In the Finite Element Method (FEM) framework, Zienkiewicz and Zhu introduced the so called ZZ error estimator [16]. The idea behind this error estimation technique is to substitute the unknown analytical solution (usually the flux) by an improved one. Generally, the flux obtained with the Finite Element (FE) solution is discontinuous through the element edges. In literature we can find several recovery techniques with different properties. The most relevant for this work will be reported in section 3.3.

In the proposed approach for 1D problems the improved solution is computed by smoothing the derivatives of functions $q_x^h(x, y, k)$ and $q_y^h(x, y, k)$, enforcing interelement continuity, following the global least square fitting process described in section 3.3.

Such smoothed functions will be represented by q_x^* and q_y^* which also have a separated representation

$$q_x^*(x, y, k) = \sum_{i=1}^{N} \tilde{X}_i(x) \cdot Y_i^h(y) \cdot K_i^h(k) \tag{25}$$

$$q_y^*(x, y, k) = \sum_{i=1}^{N} X_i^h(x) \cdot \tilde{Y}_i(y) \cdot K_i^h(k) \tag{26}$$

where $\tilde{X}_i(x)$ and $\tilde{Y}_i(y)$ denote the smoothed derivatives $\left(\frac{dX_i^h(x)}{dx}\right)^*$ and $\left(\frac{dY_i^h(y)}{dy}\right)^*$ respectively. Note that in the previous expressions we only improve the functions affected by the derivatives. The main reason of this choice is that, since we are interested in decoupling the error in the corresponding direction, it is not needed to improve the field in the other directions to get an accurate refinement indicator. Moreover, this choice permit a higher performance of the post-compressor since it is able to reduce the number of terms to a value close to N.

Now, we define a global error indicator $\hat{E}^x$ from the computed and smoothed derivatives with respect to the x-coordinate, and similarly to the y coordinate, according to

$$(E^x(k))^2 \approx (\hat{E}^x(k))^2 = \int_{\Omega_x} \int_{\Omega_y} (q_x^* - q_x^h)^2 \text{ dx dy} \tag{27}$$

$$(E^y(k))^2 \approx (\hat{E}^x(k))^2 = \int_{\Omega_x} \int_{\Omega_y} (q_y^* - q_y^h)^2 \text{ dx dy} \tag{28}$$

From this equation and following a procedure similar to the procedure shown before we can obtain the following error indicators and weighting terms:

$$\begin{cases} \hat{e}_{i,r}^{q_x} = \int_{\Omega_x^r} \hat{F}_i^x(x) \text{ dx} \\ \hat{\kappa}_{i,s}^{q_x} = \int_{\Omega_y^s} \hat{G}_i^x(y) \text{ dy} \\ \hat{\kappa}_i^{q_x} = \hat{H}_i^x(k) \end{cases} \tag{29}$$

$$\begin{cases} \hat{\kappa}_{i,r}^{q_y} &= \int_{\Omega_x^r} \hat{F}_i^y(x) \, \mathrm{dx} \\ \hat{e}_{i,s}^{q_y} &= \int_{\Omega_y^s} \hat{G}_i^y(y) \, \mathrm{dy} \\ \hat{\kappa}_i^{q_y} &= \hat{H}_i^y(k) \end{cases} \tag{30}$$

and finally the indicator at each element for guiding the refinement process, taking into account the most adverse situation due to the parameter k, as previously indicated:

$$(\hat{\mathcal{E}}_r^{q_x})^2 = \max_{k \in \Omega_k} \sum_{i=1}^{Q^x} \left(\hat{\alpha}_i^x \hat{e}_{i,r}^{q_x} \left(\sum_{s=1}^{\mathcal{M}_y} \hat{\kappa}_{i,s}^{q_x} \right) \hat{\kappa}_i^{q_x} \right) \tag{31}$$

$$(\hat{\mathcal{E}}_s^{q_y})^2 = \max_{k \in \Omega_k} \sum_{i=1}^{Q^y} \left(\hat{\alpha}_i^y \hat{e}_{i,s}^{q_y} \left(\sum_{r=1}^{\mathcal{M}_x} \hat{\kappa}_{i,r}^{q_y} \right) \hat{\kappa}_i^{q_y} \right) \tag{32}$$

where ˆ indicates that the quantities are evaluated with the smoothed field instead of using the analytical one.

3.3 Smoothing procedure

In the previous subsection we have assumed that an improved solution, q_x^* and q_y^*, for the fluxes is already available. Now we will detail the smoothing strategy used for each one of the modes. In literature we can find a great amount of different techniques to provide the improved fluxes. Most of them uses local (small) problems in order to decrease the computational cost. The use of smoothing techniques for error estimation appeared first with the contribution of Zienkiewicz and Zhu [16], where they use the nodal averaging technique consisting in a nodal representation of the flux field. The smoothed flux solution is interpolated using the Finite Element shape functions. This technique is fast and accurate enough for a linear interpolation of the solution. The same authors introduced the SPR technique [22,23], which consists in a local (patch-wise) least squares fitting of a polynomial surface to the superconvergent solution of the elements conforming the patch. The nodal smoothed value is obtained by particularizing the polynomial surface in the node. Generally, it is well known that the main drawback of recovery type techniques is the lack of accuracy along the boundaries of the domain. Further improvements to the SPR technique have been introduced [24–27] in order to prevent the lack of accuracy of the smoothed field along the boundaries of the domain and also to increase the accuracy into the domain.

More recently, Ródenas and coworkers introduced the so-called SPR-C technique [11], where the “C” stands for the constraints used to impose the local satisfaction of the equilibrium equations using the Lagrange multipliers technique. The SPR-C technique was also applied in the XFEM context by Ródenas *et al.* [28,29] and finally adapted to geometry-mesh independent FE formulations [30]. Other techniques providing also accurate improved solutions are the Moving Least Square (MLS) based techniques [31,32]. These MLS-based techniques introduce internal and boundary equilibrium in their formulation which provides accurate results even along the boundaries. In this case, instead of creating patches, these techniques localize the influence area by using a weighting function, making them suitable for meshless methods.

In the PGD framework we can use any of the techniques described above. Probably the most interesting ones are those that include internal and boundary equilibrium and are able to deal with singularities, such as the SPR-C technique. However, its use in the PGD framework involves special adaptation since it would be required to equilibrate each mode. Therefore, as a first approach we propose to use the so called global smoothing technique as a recovery process. Although the global smoothing technique is usually avoided because the recovery process would involve a high computational cost associated to the resolution of a system of equations of the size of the problem. In the case of the PGD it can be used thanks to the fact that we can separate the dimensions of the problem (the 3D problem at hand has been converted into 3 1D problems), without considerably increasing the computational cost of the process. In this case, we perform a least squares fitting technique minimizing the following functional into all the whole Ω_x domain for the flux in x direction:

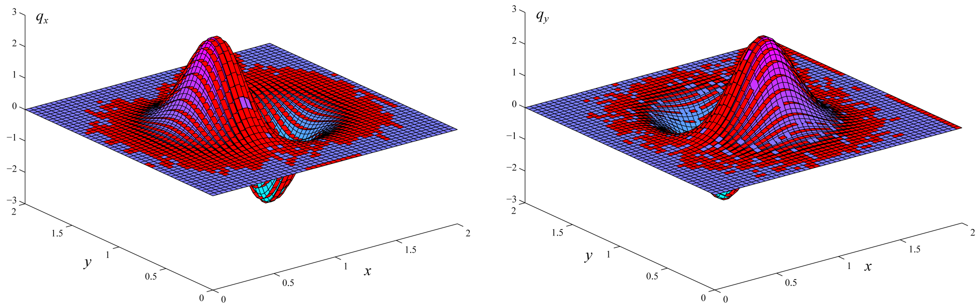
$$\int_{\Omega_x} \left(\frac{dX_i^h(x)}{dx} - \tilde{X}_i(x) \right)^2 dx \tag{33}$$

and for the the flux in y direction:

$$\int_{\Omega_y} \left(\frac{dY_i^h(y)}{dy} - \tilde{Y}_i(y) \right)^2 dy \tag{34}$$

where the smoothed fluxes are represented by their nodal values according to the following expressions for the x and y directions:

$$\tilde{X}_i(x) = \sum_{j=1}^{\mathcal{N}_x} N_j(x) \mathcal{X}_j \tag{35}$$



(a) Difference between the fluxes in x direction.

(b) Difference between the fluxes in y direction.

Fig. 1. Problem 1. Difference between the fluxes directly obtained form de computed solution (in red) and the one obtained with the smoothing process (in blue).

$$\tilde{Y}_i(y) = \sum_{j=1}^{\mathcal{N}_y} N_j(y)\mathcal{Y}_j \tag{36}$$

being $\mathcal{X}_j$ and $\mathcal{Y}_j$ the smoothed nodal values. Figure 1 compares q_x^h with q_x^* and q_y^h with q_y^*. Figure 1 shows the flux solution for Problem 1 defined in section 4.1. The fluxes obtained with the raw discrete (red) solution are discontinuous along the corresponding derivative direction while the smoothed ones (blue) are continuous.

3.4 h-adaptive refinement strategy

Once the local error indicators for the x ($\hat{\mathcal{E}}_r^{q_x}$) and y ($\hat{\mathcal{E}}_s^{q_y}$) directions have been evaluated, they will be used to guide the h-adaptive refinement process. In the bibliography we can find a variety of processes to do this. Some of them simply take the elements with higher error values and reduce their size. Other more sophisticated, construct a field that contains the element size distribution for the new mesh. This element size distribution is obtained to decrease the error in the new mesh to a certain value, taking into account the *a priori* known convergence rate of the solution. To obtain an optimal mesh, two criteria are usually considered: a)to uniformly distribute the local error in the new mesh or, b) to minimize the number of elements in the new mesh. It can be proven that both criteria are equivalent. The experience shows that these strategies tends to finally provide similar meshes when the asymptotic range is achieved. In this contribution, where only 1D meshes are used, we will use the method introduced in [33,34] that tends to uniformly distribute the error in the mesh. Thus the ratio $r^e_{\cdot}$ of the size of the new element to the size of the actual element can be written as follows [33]:

$$r_{\cdot}^{e} = \left(\frac{\hat{E}_{\text{new}}^{\cdot}}{\hat{\mathcal{E}}_{\cdot}^{\cdot}}\right)^{\frac{1}{c}} \left(\hat{\mathcal{E}}_{\cdot}^{\cdot}\right)^{\frac{1}{c(c+1)}} \left(\sum_{\cdot}^{\mathcal{M}_{\cdot}} \left(\hat{\mathcal{E}}_{\cdot}^{\cdot}\right)^{\frac{2}{(c+1)}}\right)^{\frac{-1}{2c}} \tag{37}$$

where $\hat{E}_{\text{new}}^{\cdot}$ represents the global desired error for the new mesh and c is the asymptotic convergence rate of the FE solution. In this case, in which we are using linear shape functions, $c = 1$ since it is the asymptotic convergence rate of the derivatives of the problem at hand. Note that . indicates the corresponding index for the x and y directions. This global new error can be obtained as follows:

$$\hat{E}_{\text{new}}^{\cdot} = \gamma \hat{E}_{\text{old}}^{\cdot} = \gamma \sqrt{\sum_{\cdot}^{\mathcal{M}_{\cdot}} \left(\hat{\mathcal{E}}_{\cdot}^{\cdot}\right)^{2}} \tag{38}$$

where $0 < (1 - \gamma) < 1$ is the error reduction factor for the new mesh. Note that in this particular implementation the element will only be split $n_{\cdot}^{e} \in \mathbb{N}$ times. This splitting procedure is interesting under the PGD framework since it allows to easy projection techniques between different meshes. $n_{\cdot}^{e}$ will be evaluated in the following manner:

$$n_{\cdot}^{e} = \text{floor}\left[\frac{\log r_{\cdot}^{e}}{\log 2}\right] + 1 \tag{39}$$

After some refinement steps, the asymptotic range is achieved. Thus the refinement procedure will tend to generate a h-uniform refinement steps from a h-adapted pattern, thus minimizing the degrees of freedom (NDoF) for a given accuracy level.

4 Numerical results

In the previous section we have introduced a novel error indicator procedure of expressions (23) and (24) which provide the error indicator for the refinement process in each element considering all modes obtained with the PGD algorithm. In order to verify the method we have carried out several analyses with two different problems, the first one with known analytical solution and the second one with an unknown and more complex solution.

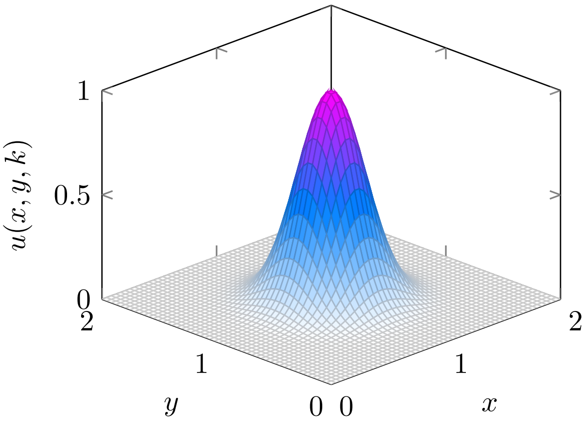


Fig. 2. Problem 1. Solution independent of the values of k.

4.1 Problem 1. Problem with analytical solution

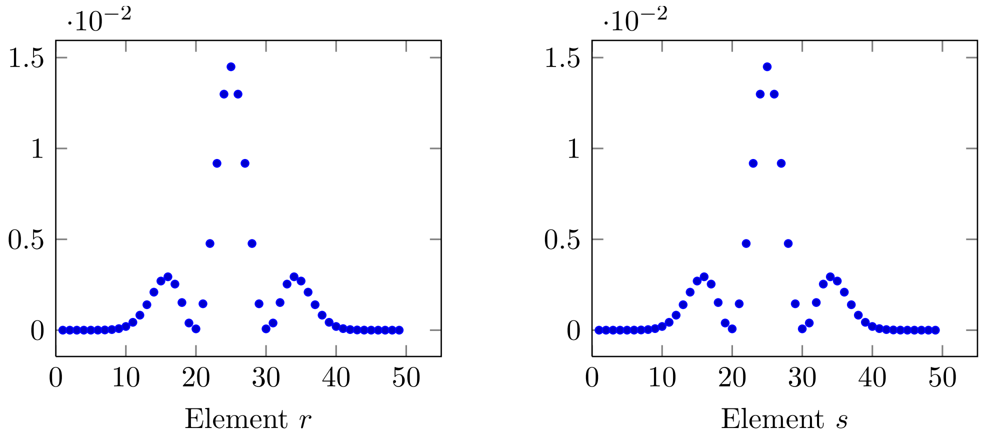
In this case we have solved the problem defined in (1) within the domain $\Omega = \Omega_x \times \Omega_y \times \Omega_k = (0,2) \times (0,2) \times (5 \cdot 10^{-3}, 5 \cdot 10^{-2})$. We have discretized the domain in $\mathcal{N}_x = 50$, $\mathcal{N}_y = 50$ and $\mathcal{N}_k = 30$ nodes. The velocity field is taken as $\mathbf{v} = (1,1)$ in Ω, with homogeneous Dirichlet boundary conditions. The analytical solution of this problem is represented in the following expression:

$$u(x,y,k) = x(x-2)e^{-10(x-1)^2} y(y-2)e^{-10(y-1)^2} \tag{40}$$

which can be exactly represented by a single mode in the PGD decomposition. Figure 2 shows a representation of the analytical solution. The source term is evaluated with the following expression since, in this case the analytical solution is known:

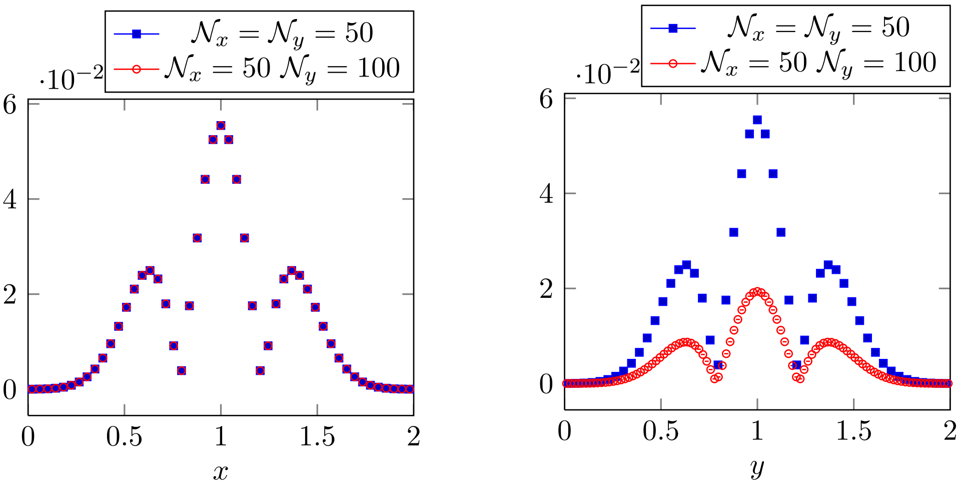
$$f(x,y,k) = \mathbf{v} \cdot \nabla u - k\ \Delta u \tag{41}$$

First of all, we will use the exact flux solution in order to check the behavior of the proposed error estimation technique. Figure 3 shows the local error distribution (at each element along the x and y coordinates). Figure 3a represents the flux in the x direction while Figure 3b represents flux in the y direction. The other terms (see (23) and (24)) will only weight the error value depending on the value corresponding parameter. We observe that in both figures, 3a and 3b, there is an increase of the error near center of the domain due to the high variation of the gradient of the solution in that area, as expected. This indicates that a finer discretization over this area will be required. It is also important to remark that due to the symmetry of the solution, the error distributions in x and y directions are identical.



(a) $e^{qx}_{1,r}$ (b) $e^{qy}_{1,s}$

Fig. 3. Problem 1. Local error indicators $e^{\cdot}_{\cdot,\cdot}$ considering the exact analytical solution.

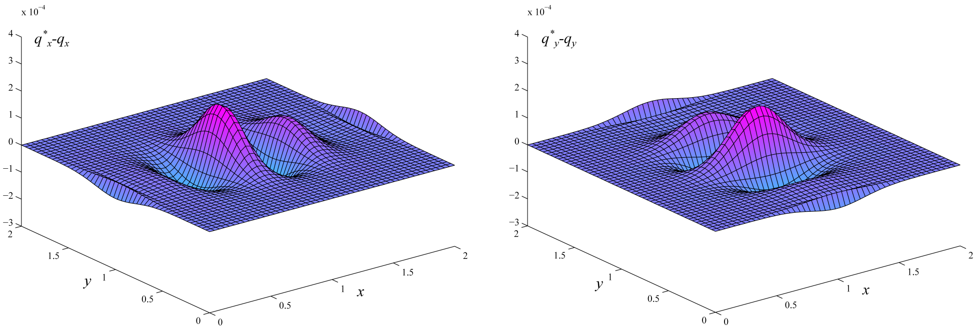


(a) $\mathcal{E}^{qx}_{1,r}$ (b) $\mathcal{E}^{qy}_{1,s}$

Fig. 4. Problem 1. Exact error indicator considering all modes and for each element r or s of the corresponding discretization.

Figure 4 shows the exact error indicator in each element for both discretization spaces (x and y) weighted according to expressions (23) and (24), respectively. In this figure we have solved the problem with two different discretizations. The first one (blue dots) the one that will be considered later, $\mathcal{N}_x = \mathcal{N}_y = 50$, and the new one (red dots) only modifying the discretization in the y coordinate, $\mathcal{N}_x = 50 \ \mathcal{N}_y = 100$. As expected, we observe, in both cases, an increase in the error level for the elements near the center of the discretization space, which coincides with high gradients of the solution. Moreover this figure shows that the influence in the discretization of one dimension to the other dimensions has a negligible effect since the curves in figure 4a overlap each other due to the error definitions used in this work.

Once the error indicators defined have been successfully checked, we will com-



(a) $q_x^* - q_x$

(b) $q_y^* - q_y$

Fig. 5. Problem 1. Difference between the smoothed fluxes q^* and the exact ones q.

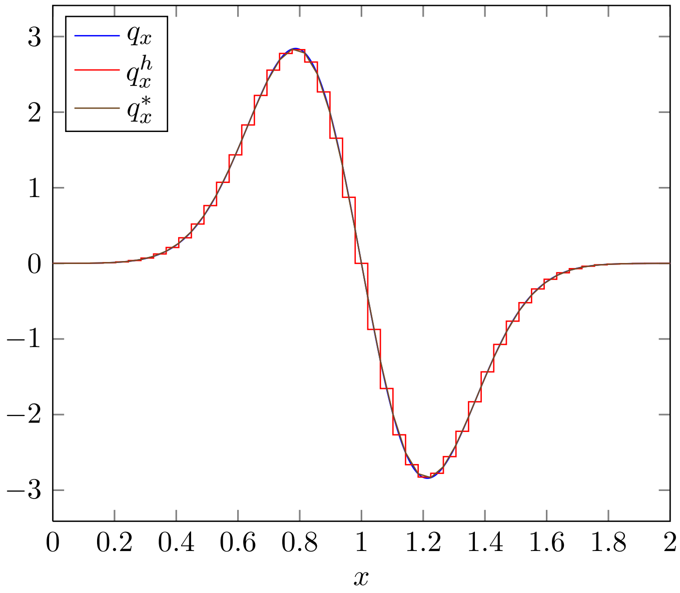
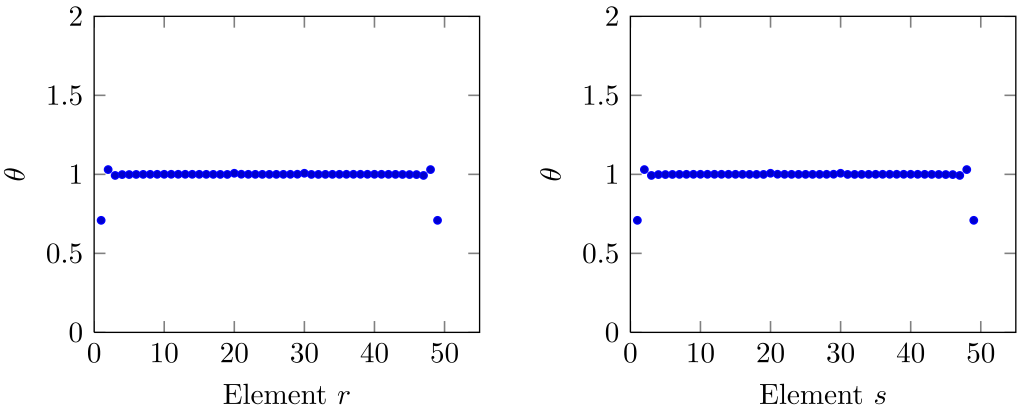


Fig. 6. Problem 1. Comparison of fluxes along x direction at the coordinate $y = 1.02$.

pare their results with those obtained with the smoothing process. First, in figure 5 we compare the smoothed flux and the exact one. We observe a negligible difference between the exact solution and the smoothed approximation into the domain (10^{-4}). Additionally in figure 6 we have presented the flux in x direction (for the coordinate $y = 1.02$) obtained with the exact solution q_x, with the PGD solution q_x^h and with the smoothed solution q_x^*. We observe the improvement in the flux representation provided by the smoothed field (brown line) in comparison with the discontinuous solution q_x^h. This will provide good error estimations when the PGD field is compared with the smoothed field.

In order to compare the results obtained with the exact flux and those obtained with the smoothed flux we define the effectivity index which is the relation between these quantities:



(a) $\hat{e}^{q_x}_{1,r}$

(b) $\hat{e}^{q_y}_{1,s}$

Fig. 7. Problem 1. Effectivity θ of the local error indicators $\hat{e}^{\cdot}_{\cdot,\cdot}$ considering the proposed smoothing technique.

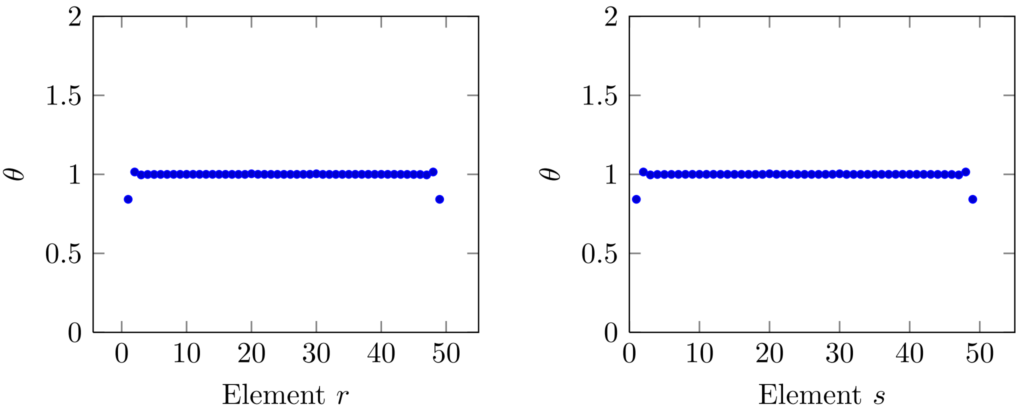
$$\theta = \frac{\hat{e}}{e} \quad \text{or} \quad \theta = \frac{\hat{\mathcal{E}}}{\mathcal{E}} \tag{42}$$

depending on the quantity considered.

Figures 7a and 7b show, for the x and y direction, the local effectivities obtained when the proposed smoothing technique is used. We observe that the error indicator provides good approximation into the domain with local effectivities close to 1, making it suitable for guiding h-adaptive refinement process. The information of the error distribution is in figures 7a and 7b. As commented before the quality of the error indicator is slightly poor near the boundaries since the error in that zones is small. This problem, common in smoothing techniques, has been already reported by other authors [24,11,30], providing a variety of improved smoothing techniques which efficiently solves this lack of accuracy along the boundaries in the standard FE framework. Figure 8 shows the local effectivity index when considering the influence of all modes (1 in this case) and the corresponding weighting according expressions (31) and (32). As a consequence of the effectivities presented in figure 7 we obtain a poor accuracy near the boundaries of the domain. However, in the rest of the domain the use of the recovered field for the error indicator provides satisfactory results.

4.1.1 Refinement strategy

Figure 9 presents the h-adapted meshes obtained with the proposed technique when the same initial mesh is used in each spacial domain. We observe that the refinement is mainly concentrated in the middle of each discretization domain, diminishing the discretization error in the zones where it is required. It is also



(a) $\hat{\mathcal{E}}_{1,r}^{q_x}$

(b) $\hat{\mathcal{E}}_{1,s}^{q_y}$

Fig. 8. Problem 1. Local effectivity of the error indicator considering all modes for each element r or s of the corresponding discretization.

important to notice that the refinement process produces identical meshes in both dimensions (x and y). However when different initial mesh is used in each spacial domain, the refinement process produces different meshes as can be noticed in figure 10 since the refinement processes depends on the initial mesh. Despite of this fact, the meshes produced concentrate the elements in the regions where the local error indicator takes higher values which coincide with the zones where the solution is more difficult to describe.

Figure 11 presents the convergence of the error indicator presented in this work. We observe that when a uniform refinement process is used the error indicator decreases smoothly and the convergence rate is practically 1, as shown in figures 11c and 11d. When the h-adaptive refinement process is activated (brown curve, same initial discretization, and black curve, different initial discretization), first we observe that the error rapidly decreases until the corresponding h-adapted patter is obtained, then it provides an uniform refinement form this h-adapted pattern, reaching a convergence rate close to 1. This behaviour is standard in h-adaptive procedures.

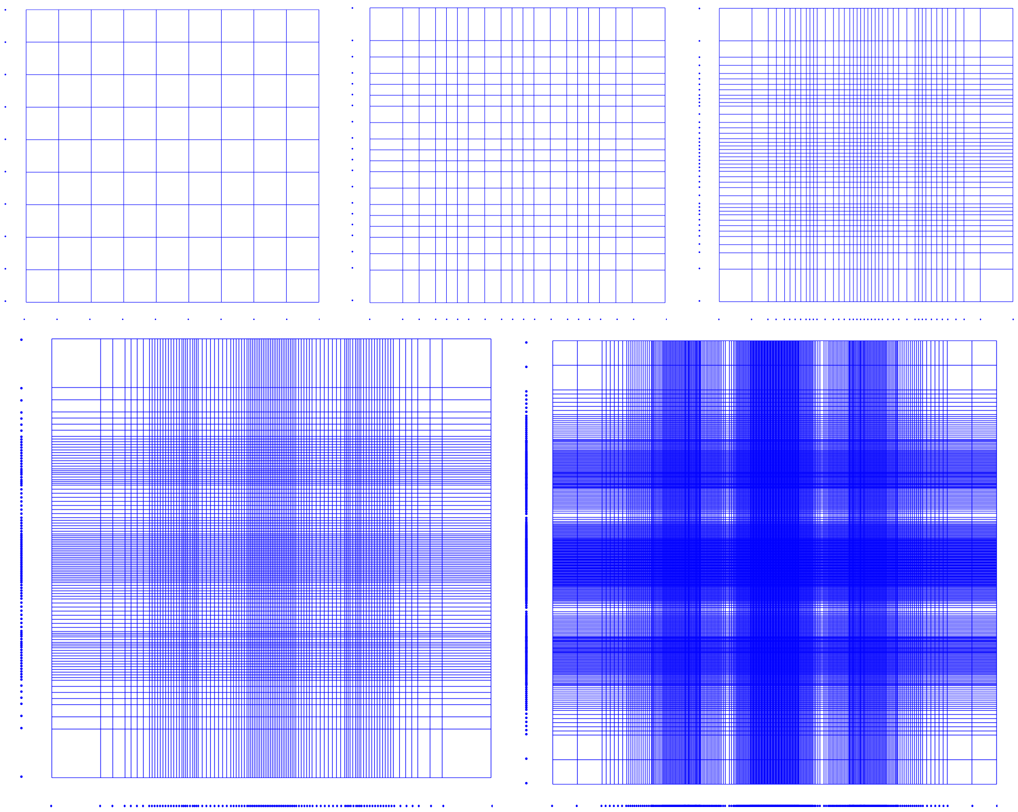


Fig. 9. Problem 1. h-adapted meshes obtained with the proposed technique with the smoothed solution. Ω_x and Ω_y are considered to be refined simultaneously. Dots near the 2D meshes indicate the discretization used for each domain. These meshes corresponds to the plot $\hat{E}^x$ adap. in figure 11.

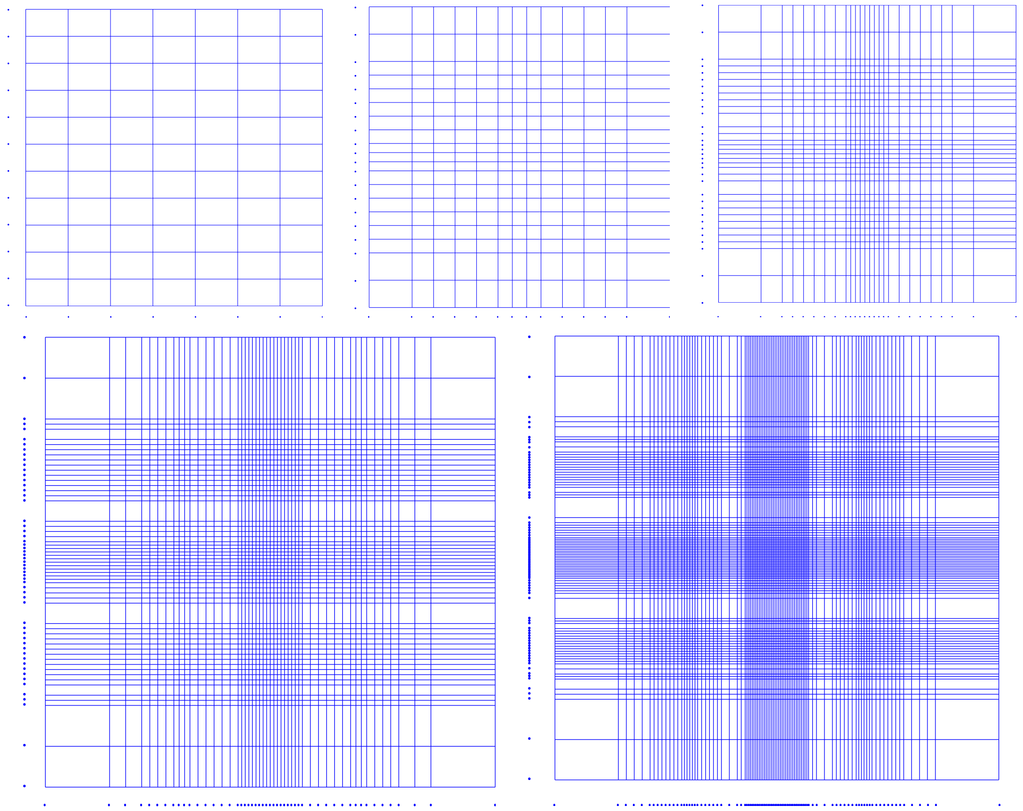
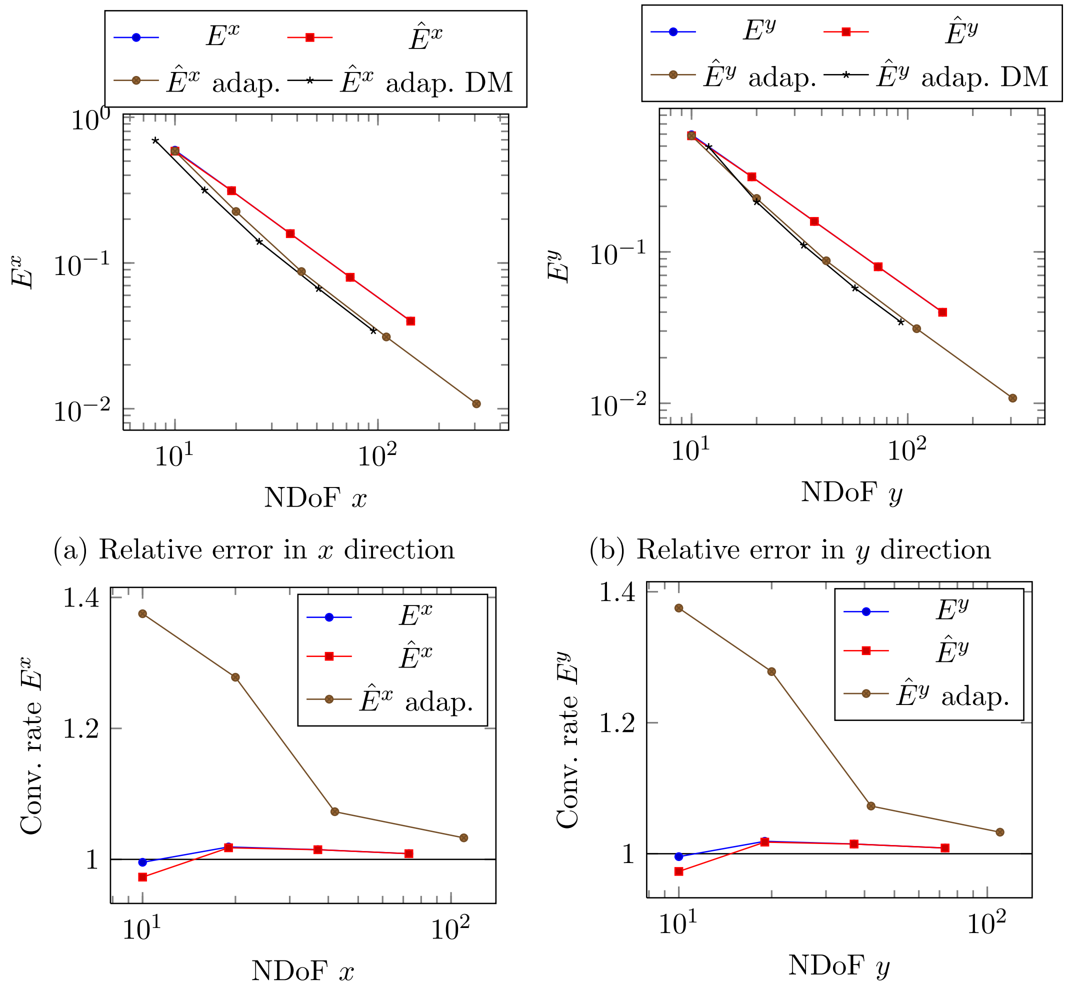


Fig. 10. Problem 1. h-adapted meshes obtained with the proposed technique with the smoothed solution. Ω_x and Ω_y are considered to be refined simultaneously but with a different initial mesh. Dots near the 2D mesh indicate the discretization used for each domain. These meshes corresponds to the plot $\hat{E}^x$ adap. DM in figure 11.



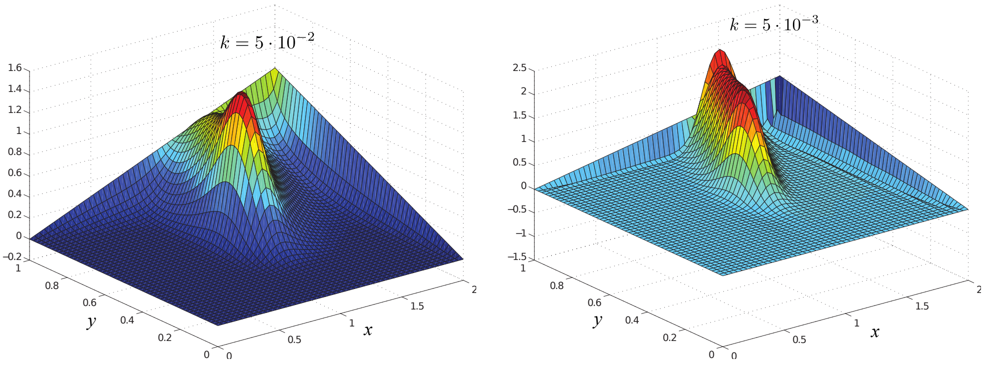
(a) Relative error in x direction

(b) Relative error in y direction

(c) Convergence rate of E^x

(d) Convergence rate of E^y

Fig. 11. Problem 1. Convergence analysis for the refinement process. The back line (DM) corresponds to the adaptive process when different initial mesh in each domain (x and y) is considered.



(a) Solution for $k = 5 \cdot 10^{-2}$.

(b) Solution for $k = 5 \cdot 10^{-3}$.

Fig. 12. Problem 2. Solutions for different values of k. Observe the effect of the source term in the interior and the boundary layers.

4.2 Problem 2. Problem without analytical solution

Once the error indicator has been tested in a problem with known analytical solution, we run the test for the problem defined in (1) within the domain $\Omega = \Omega_x \times \Omega_y \times \Omega_k = (0,2) \times (0,1) \times (5 \cdot 10^{-3}, 5 \cdot 10^{-2})$, and $\mathbf{v} = (1,1)$ in Ω. We have discretized the domain in $\mathcal{N}_x = 51$, $\mathcal{N}_y = 53$ and $\mathcal{N}_k = 30$ nodes. The following Dirichlet boundary conditions are applied:

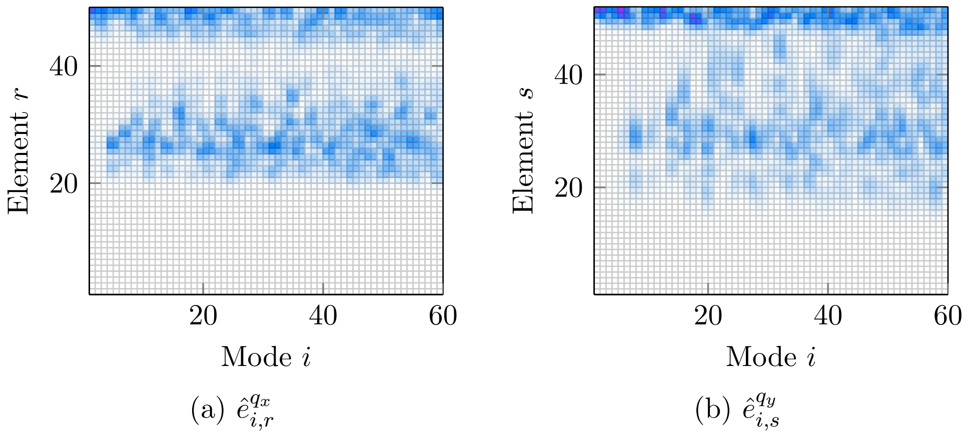
$$\begin{cases} u(0,y) = u(x,0) = 0 \\ u(x,1) = \frac{x}{2} \\ u(2,y) = y \end{cases} \tag{43}$$

The source therm considered is:

$$f(x,y) = 10e^{-100(x-1)^2} e^{-100(y-0.5)^2}. \tag{44}$$

The numerical solution of this equation (figure 12) involves a boundary layer in the vicinity of right and upper boundaries, whose characteristic length decreases as the Peclet's ($\propto \frac{1}{k}$) number increases. Thus, mesh refinement should be specially needed for high values of the Peclet's number due to the increase in the variation of the gradient of the solution. In first place, we show (in figure 12) the solution for two different values of the diffusion coefficient k. We observe how the smoothness of the solution decreases as k decreases.

Figure 13 shows the local contributions to the error indicator for each one of the modes (Q^x and Q^y) defined in (17) and (18). As in the previous example the error information is retained in the term $\hat{e}^{q_x}_{1,r}$ for the flux in x direction



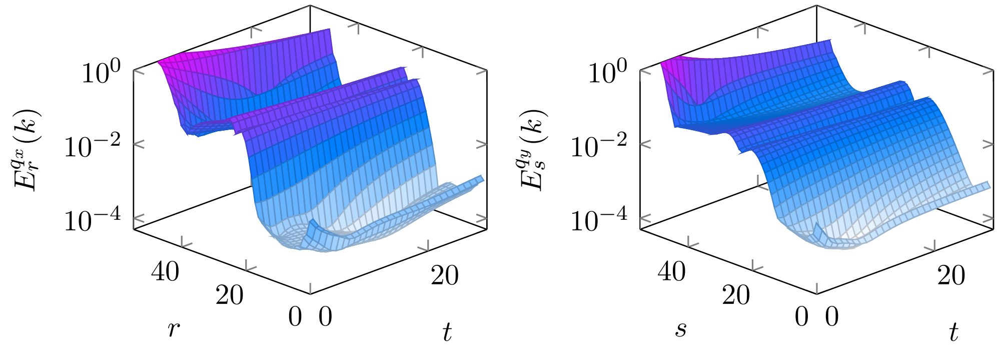
(a) $\hat{e}^{q_x}_{i,r}$ (b) $\hat{e}^{q_y}_{i,s}$

Fig. 13. Problem 2. Local error indicator $\hat{e}^{\cdot}_{\cdot,\cdot}$ considering the smoothed solution.

and in $\hat{e}^{q_y}_{1,s}$ for the y direction. We observe that in both cases (figures 13a and 13b) the local error measurement indicates that the mesh should be refined near the boundary layer and also in the zone where the source term localizes for all modes, as expected.

Figure 14 shows the dependence between the error in each discretzation space (x and y) with the value of the parameter k. We can observe that, there is a certain dependency between the error at each element in x and y with the value of the diffusion coefficient. We observe that for higher values of the diffusion coefficient k (higher values of t) the error decreases for this particular problem. For this reason we have defined the error indicator for the refinement process as the maximum value of the error at each element with respect to k. That is, taking the higher value in the graphs in figure 14 for each element of the space discretization. Figure 15 shows the higher error at each element. This quantity will be used to guide the refinement of the mesh in the x and y domains. Since the worst situation according to the value k is considered, we can guarantee the appropriate error reduction independently of the value of the parameter k. It is also important to remark that if Ω_k is modified, the error indicator will be also affected in the same sense.

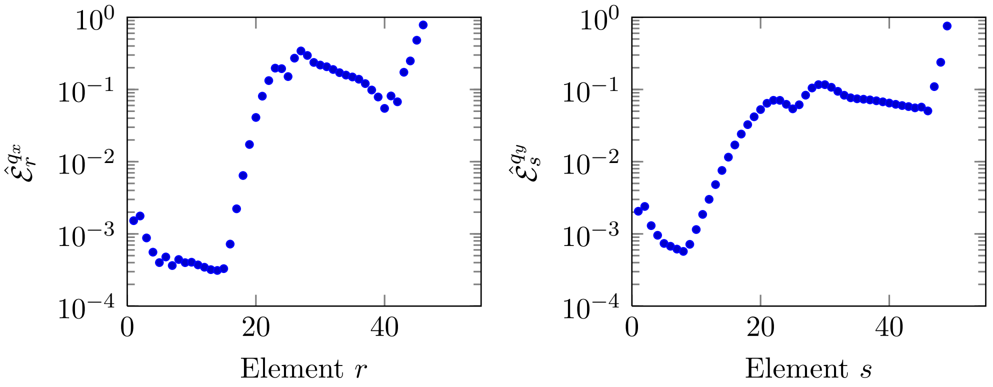
As indicated before, the smoothness of the solution in this case depends on the value of the conductivity k (see figure 12). Thus the error indicator will also depend on the subset of Ω_k we are interested in. Continuing with the previous argument, figure 16 shows the difference in the local error indicator for the refinement process when two subsets of the Ω_k are considered. It is observed that, in general, for higher conductivity values the error decreases since the smoothness of the solution increases, as expected. This indicates that depending on the region of the parametric space we are really interested in, different refined meshes will be obtained.



(a) Relative error convergence, x.

(b) Relative error convergence, y.

Fig. 14. Problem 2. Error indicator considering all modes and for each element r or s of the corresponding discretization for each value of k.



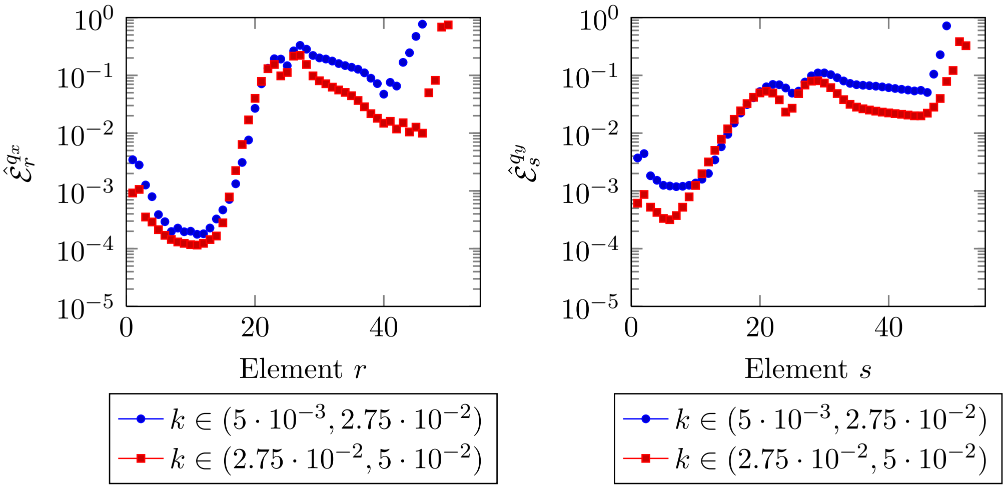
(a) $\hat{\mathcal{E}}_r^{q_x}$

(b) $\hat{\mathcal{E}}_s^{q_y}$

Fig. 15. Problem 2. Error indicator considering all modes and for each element r or s of the corresponding discretization.

4.2.1 Refinement process

Figure 17 shows the sequence of h-adapted meshes obtained with the proposed technique. In this case we have used the same initial discretization in each domain, but since the problem is not symmetric, different refined meshes are obtained in each domain. Despite of that, we clearly observe that the mesh is refined around the zones where the error indicator detects higher error levels, as expected. Figure 18 shows the convergence analysis for this problem. In this case, because of the solution complexity, the asymptotic range is only achieved in the lasts meshes. We also observe the advantage in the use of the h-adaptive process since the same error level is obtained with a considerably decrease in the number of degrees of freedom, thus improving the efficiency of the method. Figures 18c and 18d show the convergence rate. In both situations, uniform and adaptive refinements, the method tends to produce the right convergence rate when the asymptotic rage is reached.



(a) $\hat{\mathcal{E}}_r^{q_x}$

(b) $\hat{\mathcal{E}}_s^{q_y}$

Fig. 16. Problem 2. Error indicator considering all modes and for each element r or s of the corresponding discretization for two different subsets of Ω_k.

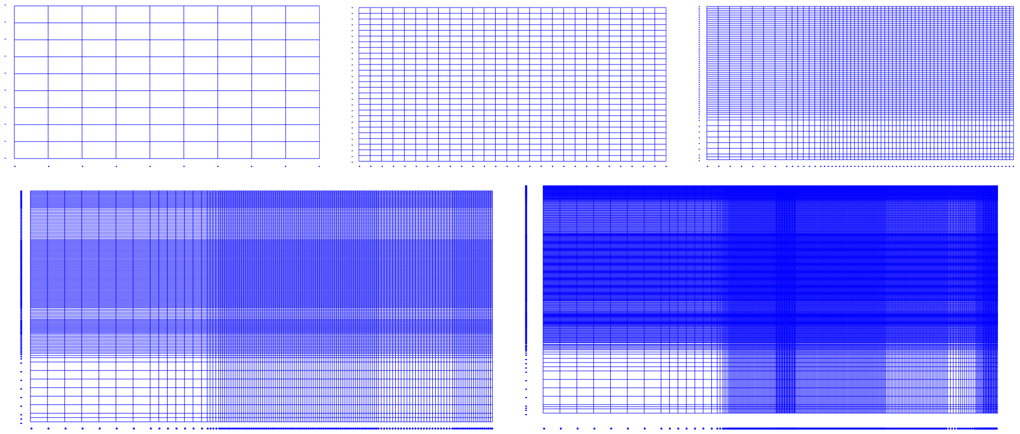
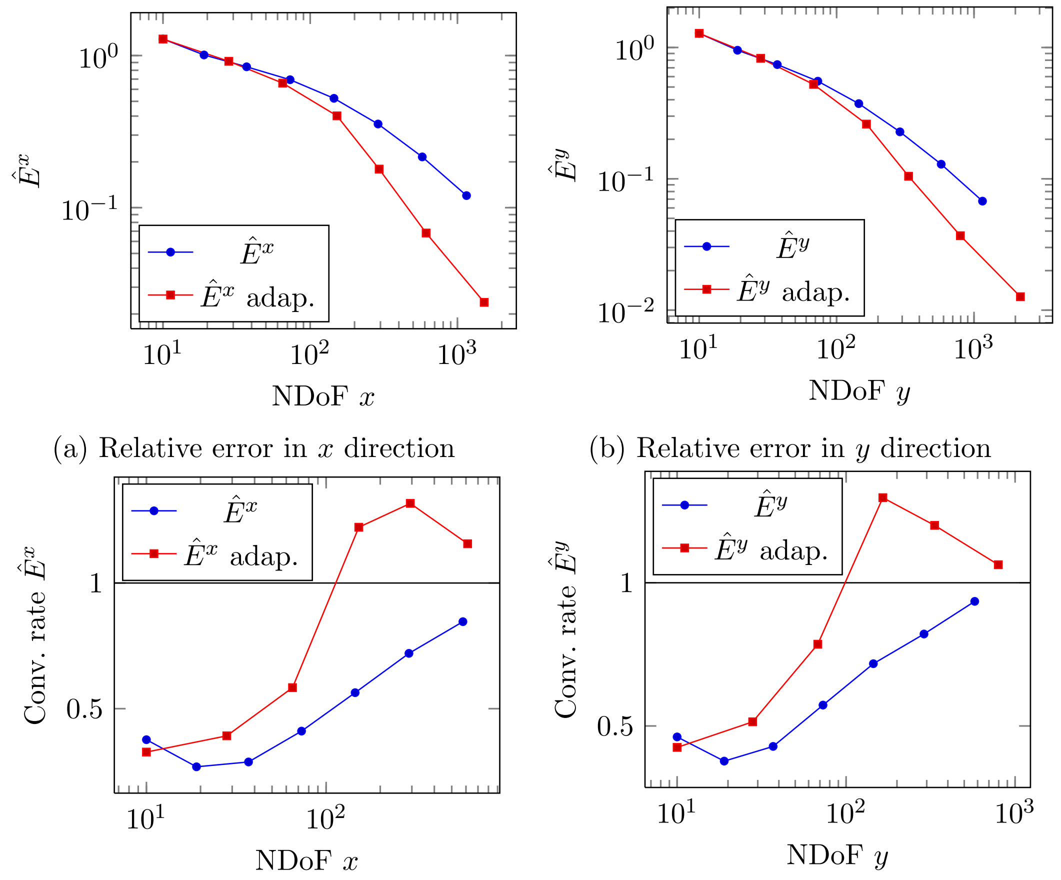


Fig. 17. Problem 1. h-adapted meshes obtained with the proposed technique with the smoothed solution. Ω_x and Ω_y are considered to be refined simultaneously. Dots near the 2D meshes indicates the discretization used in each mesh for the x and y domains.



(a) Relative error in x direction

(b) Relative error in y direction

(c) Convergence rate of $\hat{E}^x$

(d) Convergence rate of $\hat{E}^y$

Fig. 18. Problem 2. Convergence analysis for the refinement process. Values obtained for the higher error level at each element, with respect to k, are considered.

5 Conclusions

In this contribution we have presented a novel error indicator procedure under the PGD framework. This error indicator detects the part of the error due to the discretization used by the numerical method to solve each one of the separated functions used for the separate representation of the solution. The presented error indicator is able to decouple the error source among the different domains (Ω_x and Ω_y) providing the required information to perform a different mesh adaptivity process into each domain. Furthermore, the error estimation technique (computed offline) is computationally efficient since it only requires a smoothing process of the solution obtained form the numerical technique for each one of the functions in each one of the modes. Additionally, since the error indicator also admits a separated representation, we can evaluate the error due to the discretization process for each particular set of parameters, online, with a small computational cost. This allows to particularize the error in the subspace we are interested in as in the case shown in the last example. In this first contribution we present a smoothing process that is having some lack of accuracy along the boundaries of the domain. As argued before, in the FE framework this is common in smoothing-based techniques and a variety of solutions have been already proposed. Further research will be required to incorporate these advanced smoothing techniques into the PGD framework to avoid the lack of accuracy along the boundaries.

6 Acknowledgements

Authors 5 and 6 thank the financial support of the research project DPI2013-46317-R of the Ministerio de Economía y Competitividad (Spain). The funding from Universitat Politècnica de València and Generalitat Valenciana (PROMETEO/2012/023) are also acknowledged. These authors also thank the support of the Framework Programme 7 Initial Training Network Funding under grant number 289361 "Integrating Numerical Simulation and Geometric Design Technology."

References

[1] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids," *Journal of Non-Newtonian Fluid Mechanics*, vol. 139, pp. 153–176, Dec. 2006.

[2] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids," *Journal of Non-Newtonian Fluid Mechanics*, vol. 144, pp. 98–121, July 2007.

[3] F. Chinesta, P. Ladeveze, and E. Cueto, "A Short Review on Model Order Reduction Based on Proper Generalized Decomposition," *Archives of Computational Methods in Engineering*, vol. 18, pp. 395–404, Oct. 2011.

[4] E. Giner, B. Bognet, J. J. Ródenas, A. Leygue, F. J. Fuenmayor, and F. Chinesta, "The Proper Generalized Decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics," *International Journal of Solids and Structures*, vol. 50, pp. 1710–1720, May 2013.

[5] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings, "An overview of the proper generalized decomposition with applications in computational rheology," *Journal of Non-Newtonian Fluid Mechanics*, vol. 166, no. 11, pp. 578–592, 2011.

[6] A. Ammar, F. Chinesta, P. Diez, and A. Huerta, "An error estimator for separated representations of highly multidimensional models," *Computer Methods In Applied Mechanics And Engineering*, vol. 199, no. 25-28, pp. 1872–1880, 2010.

[7] J. P. Moitinho de Almeida, "A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics," *International Journal for Numerical Methods in Engineering*, vol. 94, pp. 961–984, 2013.

[8] P. Ladevèze and L. Chamoin, "On the verification of model reduction methods based on the proper generalized decomposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 2032–2047, June 2011.

[9] P. Ladevèze and D. Leguillon, "Error estimate procedure in the finite element method and applications," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 485–509, 1983.

[10] I. Babuška and W. C. Rheinboldt, "A-posteriori error estimates for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[11] J. J. Ródenas, M. Tur, F. J. Fuenmayor, and A. Vercher, "Improvement of the superconvergent patch recovery technique by the use of constraint equations: the SPR-C technique," *International Journal for Numerical Methods in Engineering*, vol. 70, no. 6, pp. 705–727, 2007.

[12] P. Díez, N. Parés, and A. Huerta, "Recovering lower bounds of the error by postprocessing implicit residual a posteriori error estimates," *International Journal for Numerical Methods in Engineering*, vol. 56, no. 10, pp. 1465–1488, 2003.

[13] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, and A. Poitou, "Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity," *Computer Methods in Applied Mechanics and Engineering*, vol. 201-204, pp. 1–12, Jan. 2012.

[14] B. Bognet, A. Leygue, and F. Chinesta, "Separated representations of 3D elastic solutions in shell geometries," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, no. 1, pp. 1–4, 2014.

[15] C. Ghnatios, F. Chinesta, and C. Binetruy, "3D Modeling of squeeze flows occurring in composite laminates," *International Journal of Material Forming*, pp. 1–11, Oct. 2013.

[16] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 337–357, 1987.

[17] F. Chinesta, R. Keunings, and A. Leygue, *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer.* Springer Publishing Company, Incorporated, Oct. 2013.

[18] J. Donea and A. Huerta, *Finite element methods for flow problems.* J. Wiley and Sons, 2002.

[19] D. Gonzalez, E. Cueto, F. Chinesta, P. Diez, and A. Huerta, "SUPG-Based Stabilization of Proper Generalized Decompositions for High-Dimensional Advection-Diffusion Equations," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 13, pp. 1216–1232, 2013.

[20] F. Chinesta, A. Ammar, and E. Cueto, "Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models," *Archives of Computational Methods in Engineering*, vol. 17, no. 4, pp. 327–350, 2010.

[21] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta, "PGD-Based Computational Vademecum for Efficient Design, Optimization and Control," *Archives of Computational Methods in Engineering*, vol. 20, pp. 31–59, Jan. 2013.

[22] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique," *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1331–1364, 1992.

[23] O. C. Zienkiewicz and J. Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity," *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pp. 1365–1382, 1992.

[24] T. Kvamsdal and K. M. Okstad, "Error estimation based on Superconvergent Patch Recovery using statically admissible stress fields," *International Journal for Numerical Methods in Engineering*, vol. 42, no. 3, pp. 443–472, 1998.

[25] N. E. Wiberg and F. Abdulwahab, "Patch recovery based on superconvergent derivatives and equilibrium," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 16, pp. 2703–2724, 1993.

[26] N. E. Wiberg, F. Abdulwahab, and S. Ziukas, "Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 20, pp. 3417–3440, 1994.

[27] T. Blacker and T. Belytschko, "Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 3, pp. 517–536, 1994.

[28] J. J. Ródenas, O. A. González-Estrada, J. E. Tarancón, and F. J. Fuenmayor, "A recovery-type error estimator for the extended finite element method based on singular+smooth stress field splitting," *International Journal for Numerical Methods in Engineering*, vol. 76, no. 4, pp. 545–571, 2008.

[29] J. J. Ródenas, O. A. González-Estrada, P. Díez, and F. J. Fuenmayor, "Accurate recovery-based upper error bounds for the extended finite element framework," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 37-40, pp. 2607–2621, 2010.

[30] E. Nadal, *Cartesian grid FEM (cgFEM): High performance h-adaptive FE analysis with efficient error control. Application to structural shape optimization.* PhD thesis, Universitat Politècnica de València, 2014.

[31] B. L. Karihaloo and Q. Z. Xiao, "Modelling of stationary and growing cracks in FE framework without remeshing: a state-of-the-art review," *Computers & Structures*, vol. 81, no. 3, pp. 119–129, 2003.

[32] O. A. González-Estrada, J. J. Ródenas, F. Chinesta, and F. J. Fuenmayor, "Enhanced error estimator based on a nearly equilibrated moving least squares recovery technique for FEM and XFEM," *Computational Mechanics*, vol. 52, pp. 321–344, 2013.

[33] F. J. Fuenmayor and J. L. Oliver, "Criteria to achieve nearly optimal meshes in the h-adaptive finite element mehod," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 23, pp. 4039–4061, 1996.

[34] F. Fuenmayor, J. Restrepo, J. Tarancón, and L. Baeza, "Error estimation and h-adaptive refinement in the analysis of natural frequencies," *Finite Elements in Analysis and Design*, vol. 38, pp. 137–153, Dec. 2001.