

The multi-dimensional refinement indicators algorithm for optimal parameterization

Hend Ben Ameur, François Clément, Pierre Weis, Guy Chavent

▶ **To cite this version:**

Hend Ben Ameur, François Clément, Pierre Weis, Guy Chavent. The multi-dimensional refinement indicators algorithm for optimal parameterization. Journal of Inverse and Ill-posed Problems, 2008, 16 (2), pp.107-126. 10.1515/JIIP.2008.008 . inria-00079668v6

HAL Id: inria-00079668

https://inria.hal.science/inria-00079668v6

Submitted on 16 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

inria-00079668, version 6 - 16 Jan 2008



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Multidimensional Refinement Indicators Algorithm for Optimal Parameterization

Hend Ben Ameur — François Clément — Pierre Weis — Guy Chavent

N° 5940

Juin 2006

Thèmes NUM et SYM



ISSN 0249-6399 ISRN INRIA/RR--5940--FR+ENG

The Multidimensional Refinement Indicators Algorithm for Optimal Parameterization

Hend Ben Ameur[*] , François Clément[†] , Pierre Weis[‡] , Guy Chavent[§]

Thèmes NUM et SYM — Systèmes numériques et Systèmes symboliques
Projets Estime et Cristal

Rapport de recherche n° 5940 — Juin 2006 — 16 pages

Abstract:

The estimation of distributed parameters in a partial differential equation (PDE) from measures of the solution of the PDE may lead to underdetermination problems. The choice of a parameterization is a frequently used way of adding a priori information by reducing the number of unknowns according to the physics of the problem. The refinement indicators algorithm provides a fruitful adaptive parameterization technique that parsimoniously opens the degrees of freedom in an iterative way. We present a *new general form of the refinement indicators algorithm* that is applicable to the *estimation of distributed multidimensional parameters* in any PDE. In the linear case, we state the relationship between the refinement indicator and the decrease of the usual least-squares data misfit objective function. We give numerical results in the simple case of the identity model, and this application reveals the *refinement indicators algorithm as an image segmentation technique*.

Key-words: inverse problem, parameterization, optimal refinement, image segmentation, fonctional programming

[*] LAMSIN, ENIT, BP 37, Le Belvédère, 1002 Tunis, Tunisie. hbenameur@yahoo.ca.
[†] Projet Estime. Francois.Clement@inria.fr.
[‡] Projet Cristal. Pierre.Weis@inria.fr.
[§] Projet Estime. Guy.Chavent@inria.fr.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

L'algorithme des indicateurs de raffinement multidimensionel pour une paramétrisation optimale

Résumé :

L'estimation de paramètres distribués dans des équations aux dérivées partielles (EDP) à partir de mesures de la solution de l'EDP peut mener à des problèmes de sous-détermination. Le choix d'une paramétrisation est un moyen usuel pour ajouter de l'information a priori en réduisant le nombre d'inconnues en relation avec la physique du problème. L'algorithme des indicateurs de raffinement fourni une technique de paramétrisation adaptative fructueuse qui ouvre parcimonieusement les degrés de liberté de façon itérative. Nous présentons une *nouvelle forme générale de l'algorithme des indicateurs de raffinement* qui s'applique à l'estimation des paramètres multi-dimensionnels dans toute EDP. Dans le cas linéaire, nous établissons le lien entre l'indicateur de raffinement et la décroissance de la fonction objectif des moindres carrés quantifiant l'erreur aux données. Nous donnons des résultats numériques pour le cas simple du modèle identité, et cette application permet de voir l'*algorithme des indicateurs de raffinement comme une technique de segmentation d'image*.

Mots-clés : problème inverse, paramétrisation, raffinement optimal, segmentation d'image, programmation fonctionnelle

1 Introduction

The inverse problems of parameter estimation consist in identifying unknown parameters which are space dependent coefficients in a system of partial differential equations modeling a physical problem. The parameter estimation problem can be set as a minimization problem of an objective function defined as the least-squares misfit between measurements and the corresponding quantities computed with a chosen parameterization of the parameters (solution of the partial differential equations system under consideration).

Experimental measurements are expensive, thus one of the difficulties when solving a parameter estimation inverse problem is that the data is usually insufficient to estimate the value of the parameter in each cell of the computational mesh, the parameter estimation problem is underdetermined. Therefore, a parameterization of the sought parameter is chosen to reduce the number of unknowns. The multiscale parameterization approach is popular for many applications, e.g. see [15, 6, 17, 1]. It consists in solving the problem through successive approximations by refining uniformly the parameter at the next finer scale all over the domain and stopping the process when the refinement does not induce significant decrease of the objective function any more. This method is very attractive as it may provide a regularizing effect on the resolution of the inverse problem and it may also circumvent local minima problems as shown in [10, 11]. But its main drawback is the over-parameterization problem due to the uniform refinement of the parameter, see [7, 8].

To avoid this pitfall, the refinement indicators algorithm provides an adaptative parameterization technique that parsimoniously opens the degrees of freedom in an iterative way driven at first order by the model to locate the discontinuities of the sought parameter. The detailed definition of the technique was described in [2] for the application to the estimation of hydraulic transmissivities, and a variant of the idea was first briefly presented in [9] for the application to the history matching of an oil reservoir.

In the present work, we extend the definition of refinement indicators to the more general case of multidimensional distributed parameters, we make the link between the (first order) refinement indicator and the exact decrease of the objective function in the linear case, and we propose a generic version of the algorithm that may be applied to any inverse problem of parameter estimation in a system of partial differential equations. We show numerical results for the simple case of the identity model. The application of the refinement indicators algorithm to the identity model applied to images is interpreted as an amazing image segmentation technique.

The paper is organized as follows. Section 2 is devoted to theoretical developments about adaptive parameterization through the refinement indicators notion: we quantify the effect of refinement on the decrease of the objective function, and we generalize the definition of refinement indicators for multidimensional parameters. We present in Section 3 the general form of the refinement indicators algorithm, we discuss the issue of finding the best cutting for multidimensional parameters and briefly point out implementation subtleties of the algorithm. In section 4, we focus on the identity model; we then apply the multidimensional refinement indicators algorithm to the particular case of RGB color images.

2 Adaptive inversion and refinement indicators

The inverse problem of parameter estimation is first set as the minimization of a least-squares objective function. Then, the notion of adaptive parameterization induced by zonations of the domain is presented in the discrete case for piecewise constant parameters. The decrease of the optimal value of the objective function resulting from the splitting of one zone into two subzones is then quantified for the linearized problem in the continuous case, and refinement indicators are finally defined in the general case from a first order approximation.

2.1 Least-squares inversion

When defining a system of partial differential equations (PDE) modeling a physical behavior in a domain $\Omega \subset \mathbb{R}^N$ (e.g. with $N = 1$, 2 or 3), one usually introduces parameters such as material properties. We consider the case where these parameters are *distributed* and possibly *vector valued*, i.e. belong to a space P of functions defined over Ω with values in $\mathbb{R}^{n_p}$ ($n_p \geq 1$ is the *dimension* of the vector parameter $p(x)$ at any $x \in \Omega$).

Some of these parameters may not be accessible by direct measurements, so they have to be estimated indirectly. If $p_{\text{true}} \in P$ denotes such an unknown parameter that we are looking for in a set of admissible parameters P^{ad}, and if $d \simeq \mathcal{F}(p_{\text{true}}) \in \mathbb{R}^{n_d}$ (the *data*) denotes the corresponding vector of available measurements of the solution of the PDE, one can attempt an indirect determination of p_{true} from d by solving the *least-squares inverse problem*:

$$\text{minimize } J(p) \text{ for } p \in P^{\text{ad}} \tag{1}$$

where $J(p)$ is the least-squares misfit between d and the corresponding quantities $\mathcal{F}(p)$ computed from the current parameter p,

$$\mathcal{J}(p) = \frac{1}{2}\|d - \mathcal{F}(p)\|^2. \tag{2}$$

The *forward operator* $\mathcal{F}$ is the composition of the *model operator* $\mathcal{M}$ (which computes the solution of the PDE for a given parameter p), with the *observation operator* $\mathcal{O}$ (which computes the output of the observation device applied to the solution of the PDE). For example, observations can be made of the values of the solution of the PDE at a set of measurement points of the domain Ω.

Minimizing such an objective function avoids the difficulty of inverting $\mathcal{M}$, which is impossible in most cases.

2.2 Adaptive parameterization

In parameter estimation problems, the data is usually insufficient to estimate the value of the (possibly vector valued) unknown parameter at each point $x \in \Omega$ (i.e. in each grid cell after discretization). It is usually impossible to increase the number of measurements, and one solution is then to reduce the number of unknowns by searching for the parameter in a subspace of P of finite (and small) dimension.

For that, we proceed in two steps: first we construct a finite dimensional subspace P_n of the infinite dimensional space P, then we look for an approximation p_n of the unknown and infinite dimensional true parameter in $P_n^{\text{ad}} = P_n \cap P^{\text{ad}}$. The dimension of P_n should remain small enough to avoid over-parameterization, i.e. the Jacobian of $\mathcal{F}$ as a function from P_n to $\mathbb{R}^{n_d}$ should have a full rank, and the approximation p_n should explain the data up to the noise level. Notice that here both P_n and p_n are unknown.

For a given subspace P_n, a natural choice for p_n is a minimizer of the objective function J in P_n^{ad}.

It is classical to build the subspace P_n in an iterative way by successive refinements as in the multiscale approach, see [15, 6, 17, 1]. We use the *adaptive parameterization* technique developed in [2]. These approaches are also known to have a regularizing effect on the resolution of the inverse problem, e.g. see [10, 15, 16, 14, 5].

The adaptive parameterization approach consists in adding one (vectorial) degree of freedom at a time. The new degree of freedom is chosen according to *refinement indicators* which evaluate the benefit for the objective function of the introduction of a given family of degrees of freedom. Hence, each subspace P_n is of dimension $n \times n_p$. The process is stopped when p_n explains the data up to the noise level.

It is convenient to consider P_n as the range of an—unknown—*parameterization map*

$$\mathcal{P}_n : m_n \in M_n^{\text{ad}} \longmapsto p_n \in P_n^{\text{ad}} \tag{3}$$

where m_n is the *coarse* parameter (of small finite dimension), by opposition to the *fine* parameter p_n (of large, and possibly infinite dimension). M_n^{ad} is the space of admissible coarse parameters. Typically, m_n is made of the coefficients of $p_n \in P_n$ on a basis of P_n, in which case $\mathcal{P}_n$ is a linear operator. But the parameterization map can also be nonlinear, as for example in the case where m_n is made of the coefficients of a closed form formula defining the fine parameter p_n. For any parameterization map $\mathcal{P}_n$, we define the same objective function on M_n^{ad} by

$$\mathcal{J}_n(m_n) = \mathcal{J}(\mathcal{P}_n(m_n)), \tag{4}$$

and the least-squares problem becomes:

$$\text{minimize } J_n(m_n) \text{ for } m_n \in M_n^{\text{ad}}. \tag{5}$$

Let $Z_n = (Z_{n,j})_{1\le j\le n}$ be a partition of the closure of the domain Ω in n parts, i.e. for all $j \in \{1,\dots,n\}$ $Z_{n,j} \subset \overline{\Omega}$, $\bigcup_{1\le j\le n} Z_{n,j} = \overline{\Omega}$, and $Z_{n,j} \cap Z_{n,k} = \emptyset$ as soon as $j \neq k$. We consider that the subspace P_n is made of piecewise constant (vector valued) functions on this partition. We call *zonation* the partition Z_n, and *zone* each part $Z_{n,j}$ of the zonation. The parameterization map associated with the zonation Z_n is then

$$\mathcal{P}_n : m_n = (m_{n,j})_{1\le j\le n} \quad \longmapsto \quad p_n = (x \in \Omega \mapsto m_{n,j} \in \mathbb{R}^{n_p} \text{ when } x \in Z_{n,j}). \tag{6}$$

It associates the coarse parameter m_n with the function p_n which takes on each zone $Z_{n,j}$ the constant value $m_{n,j}$.

In the iterative process, when going from P_n to P_{n+1}, we introduce a new (vectorial) degree of freedom by dividing one zone of Z_n into two subzones, thus producing a new zonation Z_{n+1} having one zone more than Z_n.

2.3 Quantifying the effect of refinement for the linearized problem

Considering a current zonation, the best refinement would be the one corresponding to the largest decrease of the objective function. This decrease can only be computed by actually solving the least-squares problem (5) for each possible refinement to be tested, hence only a very small number of them could be evaluated before reaching a decision. So, we search for a closed form formula for the decrease of the objective function for the *linearized* problem. We shall still denote by $\mathcal{F}$ the linearized forward map.

For simplicity, we consider the case where the optimization problem (5) is unconstrained, so that $P_n^{\text{ad}} = P_n$ and $M_n^{\text{ad}} = \mathbb{R}^{n\times n_p}$ for all n.

Let us compute the decrease of the optimal objective function (4) resulting from refining the current zonation by splitting one of the zones into two subzones by means of a continuum of intermediate minimization problems constraining the discontinuity jump at the interface between the subzones.

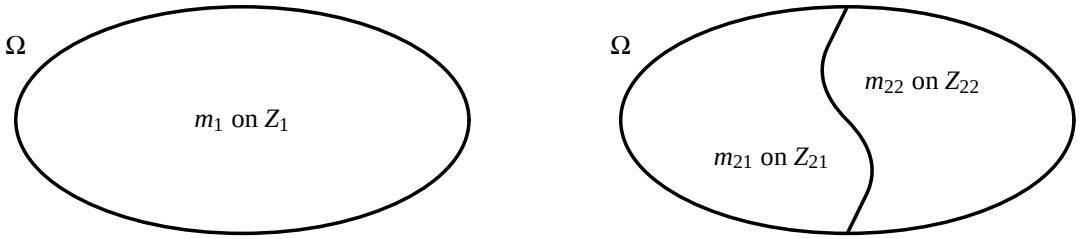


Figure 1: Refinement of a single zone zonation represented in a two-dimensional space for simplicity. Left: one-zone parameterization $\mathcal{P}_1$ with $\overline{\Omega} = Z_{1,1}$. Right: two-zone parameterization $\mathcal{P}_2$ with $\overline{\Omega} = Z_{2,1} \cup Z_{2,2}$.

Let us consider the case where only one zone covers the whole domain, corresponding to the one-zone parameterization $\mathcal{P}_1$, see Figure 2.3 (left). The refinement builds the two-zone parameterization $\mathcal{P}_2$ of Figure 2.3 (right). We denote by $m_1^{\text{opt}} = (m_{1,1}^{\text{opt}})$ and $\jmath_1^{\text{opt}}$, $m_2^{\text{opt}} = (m_{2,1}^{\text{opt}}, m_{2,2}^{\text{opt}})^T$ and $\jmath_2^{\text{opt}}$ the optimal coarse parameter and objective function respectively when considering the parameterizations $\mathcal{P}_1$ and $\mathcal{P}_2$.

If the discontinuity jump $c^{\text{opt}} = m_{2,1}^{\text{opt}} - m_{2,2}^{\text{opt}}$ was known, then minimizing $\jmath_2$ (i.e. minimizing $\jmath$ considering the parameterization $\mathcal{P}_2$) under the constraint $m_{2,1} - m_{2,2} = c^{\text{opt}}$ would give us the same optimal coarse parameter m_2^{opt} and the same optimal objective function value $\jmath_2^{\text{opt}}$, the ones obtained without any constraints. And minimizing $\jmath_2$ and under the constraint $m_{2,1} - m_{2,2} = 0$ would keep the optimal values m_1^{opt} and $\jmath_1^{\text{opt}}$, obtained with the parameterization $\mathcal{P}_1$. Thus, when the discontinuity jump c goes continuously from 0 to c^{opt}, then the minimum of $\jmath_2$ under the constraint $m_{2,1} - m_{2,2} = c$ goes continuously (and even in a continuously differentiable way when $\mathcal{F}$ is continuously differentiable) from the minimum obtained for a single zone to the minimum obtained with the two zones.

The components $m_{2,1}$ and $m_{2,2}$ of the coarse parameter are column vectors of dimension n_p (i.e. $n_p = 1$ in the scalar case). We can denote the constraint on the discontinuity in matrix form by

$$Am_2 = c \tag{7}$$

where $m_2 = (m_{2,1}, m_{2,2})^T$ and the $n_p \times 2n_p$ rectangular matrix A is of the form

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 \end{pmatrix}. \tag{8}$$

We denote the (linear) direct operator in terms of the coarse parameter m_n by $\mathcal{F}_n = \mathcal{F} \circ \mathcal{P}_n$. In the following computation of the decrease of the optimal value of the objective function, we will only consider the two-zone parameterization $\mathcal{P}_2$. Hence, when no ambiguity can arise, we will simply write $\mathcal{F}$ for $\mathcal{F}_2$, $\jmath$ for $\jmath_2$ and m for m_2. The gradient of the quadratic objective function $\jmath\,(= \jmath_2)$ is given by

$$\nabla \jmath\,(m) = \mathcal{F}^T(\mathcal{F}\,m - d). \tag{9}$$

The Lagrangian function associated with the minimization of $\jmath$ under the constraint (7) writes

$$\mathcal{L}^c(m, \lambda) = \jmath\,(m) + \langle \lambda, Am - c \rangle \tag{10}$$

where λ is the Lagrange multiplier associated with the constraint. It is a vector of dimension n_p. Then, the Lagrange condition ensures that at the optimum, $m^{c,\mathrm{opt}} = \arg\min_{Am=c} \jmath(m)$ and $\lambda^{c,\mathrm{opt}}$ are obtained by solving

$$\left\{ \begin{array}{l} \dfrac{\partial \mathcal{L}^c}{\partial m}(m,\lambda) = \nabla \jmath(m) + A^T\lambda = 0, \\ \dfrac{\partial \mathcal{L}^c}{\partial \lambda}(m,\lambda) = Am - c = 0. \end{array} \right. \tag{11}$$

Using the expression (9) of the gradient, the Lagrange condition for $c = 0$ becomes

$$\left\{ \begin{array}{l} \mathcal{F}^T(\mathcal{F} m^{0,\mathrm{opt}} - d) + A^T\lambda^{0,\mathrm{opt}} = 0, \\ \\ Am^{0,\mathrm{opt}} = 0. \end{array} \right. \tag{12}$$

The last equation writes $m^{0,\mathrm{opt}}_{2,1} = m^{0,\mathrm{opt}}_{2,2} (= m^{\mathrm{opt}}_{1,1})$, and we have

$$\jmath^{0,\mathrm{opt}} = \jmath_{\mathcal{P}_2}(m^{0,\mathrm{opt}}) = \jmath^{\mathrm{opt}}_1. \tag{13}$$

The optimal coarse parameter $m^{\mathrm{opt}} (= m^{\mathrm{opt}}_2) = \arg\min \jmath(m)$ (i.e., without constraint) satisfies the optimality condition

$$\nabla \jmath(m^{\mathrm{opt}}) = \mathcal{F}^T(\mathcal{F} m^{\mathrm{opt}} - d) = 0. \tag{14}$$

Let $e = m^{\mathrm{opt}} - m^{0,\mathrm{opt}}$. Developing the squared norm yields

$$\jmath^{\mathrm{opt}} = \jmath^{0,\mathrm{opt}} + \frac{1}{2}\|\mathcal{F} e\|^2 + \left\langle \mathcal{F} m^{0,\mathrm{opt}} - d, \mathcal{F} e \right\rangle.$$

But, from the optimality condition (14), we have

$$\left\langle \mathcal{F} m^{0,\mathrm{opt}} - d, \mathcal{F} e \right\rangle = -\|\mathcal{F} e\|^2$$

and taking the difference between (14) and the first equation of (12) leads to

$$\mathcal{F}^T\mathcal{F} e = A^T\lambda^{0,\mathrm{opt}} = \begin{pmatrix} \lambda^{0,\mathrm{opt}} \\ -\lambda^{0,\mathrm{opt}} \end{pmatrix}.$$

Therefore, when $\mathcal{F}^T\mathcal{F}$ is invertible[1], we finally obtain

$$\jmath^{\mathrm{opt}}_1 - \jmath^{\mathrm{opt}}_2 = \frac{1}{2}\|A^T\lambda^{0,\mathrm{opt}}\|^2_{(\mathcal{F}^T\mathcal{F})^{-1}} = \frac{1}{2}\left\langle A^T\lambda^{0,\mathrm{opt}}, (\mathcal{F}^T\mathcal{F})^{-1} A^T\lambda^{0,\mathrm{opt}} \right\rangle \tag{15}$$

where the norm and the scalar product are now defined in the space of coarse parameters for the refined zonation (and not in the space of data as before).

Remark 1 *In the general case, an n-zone parameterization is refined into an $(n+1)$-zone parameterization by splitting only one zone into two subzones. Then, the matrix defining the new discontinuity jump of the parameter is a block matrix containing only one nonzero block corresponding to the discontinuity of the parameter between the two subzones and this block is equal to the matrix A of (8). Hence, the computation of the decrease of the optimal value of the objective function is exactly the same.*

Because the number of zones is supposed to stay small, after discretization, the direct operator $\mathcal{F}_n$ is usually injective. But applying $(\mathcal{F}_n^T\mathcal{F}_n)^{-1}$ is at least as expensive as solving the linearized minimization problem. Thus, the exact computation of the decrease of the optimal value of the objective function for the linearized problem is generally not compatible with the idea of a very fast computation of an indicator on the quality of the new zonation. Nevertheless, an important exception is the case of the identity direct operator; it will be developed in section 4. The next subsection is devoted to a first order computation in the general nonlinear case.

[1] E.g., when $\mathcal{F}$ is injective in the finite-dimensional case.

2.4 Refinement indicators for multidimensional parameters

Following the notations of the previous subsection, let $\jmath^{c,\mathrm{opt}}$ denote the optimal value of the objective function obtained with the parameterization $\mathcal{P}_2$ under the constraint (7). A first order development of $\jmath^{c,\mathrm{opt}}$ with respect to the discontinuity jump c at $c=0$ can be written

$$\jmath^{c,\mathrm{opt}} = \jmath^{0,\mathrm{opt}} + \frac{\partial \jmath^{c,\mathrm{opt}}}{\partial c}_{|c=0} c + \dots$$

The norm of the quantity $\frac{\partial \jmath^{c,\mathrm{opt}}}{\partial c}_{|c=0}$ tells us, at the first order, how large would be the difference between $\jmath^{c,\mathrm{opt}}$ and $\jmath^{0,\mathrm{opt}} = \jmath_1^{\mathrm{opt}}$ (from (13)). So, it gives us the first order effect on the optimal value of the objective function produced by refining and allowing a discontinuity of the parameter between the two subzones of the parameterization $\mathcal{P}_2$. Similarly to the particular scalar case developed in [2], this norm is called the *refinement indicator* associated with the splitting of the zone $Z_{1,1}$ into the two subzones $Z_{2,1}$ and $Z_{2,2}$, and it is denoted by $\mathcal{I}$.

Deriving the expression (10) of the Lagrangian with respect to c gives

$$\frac{\partial \mathcal{L}^c}{\partial c}(m,\lambda) = \frac{\partial \jmath}{\partial c}(m) - \lambda.$$

Hence, from the Lagrange condition (11), we have at the optimum for $c=0$

$$\frac{\partial \jmath}{\partial c}(m^{0,\mathrm{opt}}) - \lambda^{0,\mathrm{opt}} = 0.$$

Then, since

$$\frac{\partial \jmath^{c,\mathrm{opt}}}{\partial c}_{|c=0} = \frac{\partial \jmath}{\partial c}(m^{c,\mathrm{opt}})_{|c=0} = \frac{\partial \jmath}{\partial c}(m^{0,\mathrm{opt}}),$$

this shows that the refinement indicator is nothing but the norm of the Lagrange multiplier,

$$\mathcal{I} = \|\lambda^{0,\mathrm{opt}}\|. \tag{16}$$

Remark 2 *In this section, we have considered that the vector parameter was a piecewise constant function (i.e. constant in each zone). The idea of refinement indicators is based on defining a zonation by identifying the interfaces between the zones that are related to the discontinuities of the parameter. To estimate more regular parameters, one can use similar refinement indicators in the more general case where the vector parameter is a piecewise higher order function, which is continuous in each zone and presents discontinuities at the interfaces between the zones.*

3 Refinement indicators algorithm

To solve numerically the parameter estimation problem, we consider a discretization of the domain Ω by a fine mesh $\mathcal{T}_h = (K_i)_{i\in I}$, i.e. with $\overline{\Omega} = \bigcup_{i\in I} K_i$. Let $n_I = \mathrm{card} I$ be the number of cells. The fine parameter is then the vector $p = (p_i)_{i\in I}$ which approximates the parameter on each cell K_i of $\mathcal{T}_h$. We consider zonations $Z_n = (Z_{n,j})_{1\le j\le n}$ following the mesh such that the associated parameterization map (3) is

$$\begin{array}{rccc} \mathcal{P}_n : & (\mathbb{R}^{n_p})^n & \longrightarrow & (\mathbb{R}^{n_p})^{n_I} \\ & m_n = (m_{n,j})_{1\le j\le n} & \longmapsto & p_n = (p_{n,i})_{i\in I} \text{ with } p_{n,i} = m_{n,j} \text{ when } K_i \subset Z_{n,j}. \end{array} \tag{17}$$

In other words, when dealing with a discrete problem, a zonation is related to a partition of the set I of indices:

$$Z_{n,j} = \bigcup_{i\in I_{n,j}} K_i \qquad \text{with} \quad I = \bigsqcup_{1\le j\le n} I_{n,j} \text{ (disjoint union).}$$

The finite set of measurement points is indexed by $I_m \subset I$. Keeping the same notations, the discrete objective function to minimize writes

$$\jmath(p) = \frac{1}{2}\sum_{i\in I_m} (d_i - u_i)^2 \tag{18}$$

where d_i is the measurement in the cell K_i and u_i denotes the discrete approximation of $\mathcal{F}(p)$ in the same cell K_i.

3.1 The adaptive parameterization technique

In practice, the optimal coarse parameter m_n^{opt} associated with the current parameterization $\mathcal{P}_n$ is computed by applying a gradient algorithm to the minimization of the corresponding objective function $\mathcal{J}_n$. Hence, not only $m_n^{\text{opt}} = (m_{n,j}^{\text{opt}})_{1\le j\le n}$ is available, but also the coarse gradient $\nabla\mathcal{J}_n(m_n^{\text{opt}})$, which vanishes at the reached minimum. The key is to compute the gradient of $\mathcal{J}_n$ for the current parameterization by the adjoint approach considering the fine discretization $\mathcal{T}_h$ of Ω. This approach provides, at no additional cost, the *fine gradient*

$$\nabla_p \mathcal{J}(p_n^{\text{opt}}) = \left(\frac{\partial \mathcal{J}}{\partial p_i}(p_n^{\text{opt}})\right)_{i\in I} \tag{19}$$

where $p_n^{\text{opt}} = \mathcal{P}_n m_n^{\text{opt}}$, since the *coarse gradient* of $\mathcal{J}_n$ is given by $\nabla\mathcal{J}_n(m_n) = \mathcal{P}_n^T \nabla\mathcal{J}(p_n)$, and hence, it is simply obtained by summing the components of the fine gradient inside each zone of the current parameterization $\mathcal{P}_n$.

A refinement of the current parameterization $\mathcal{P}_n$ is obtained by *cutting* one zone $Z_{n,j}$ into two subzones Z_{n,j_1}^{cut} and Z_{n,j_2}^{cut} with $1 \le j_1, j_2 \le n+1$. The resulting $(n+1)$-zone zonation Z_n^{cut} is a candidate for Z_{n+1}. This operation is fully characterized by the subsets of indices corresponding to the new subzones which satisfy the disjoint union $I_{n,j} = I_{n,j_1}^{\text{cut}} \sqcup I_{n,j_2}^{\text{cut}}$. The associated parameterization is denoted by $\mathcal{P}_n^{\text{cut}}$ and the corresponding objective function by $\mathcal{J}_n^{\text{cut}}$. Opening the new (vectorial) degree of freedom while keeping the same value in the two subzones does not change the optimum of the objective function, and the components of the gradient $\nabla\mathcal{J}_n^{\text{cut}}((m_n^{\text{cut}})^{0,\text{opt}})$ are all equal to zero except for components j_1 and j_2 that are opposite vectors (of dimension n_p). From the first equation in (11) and the form of matrix A, we deduce that the Lagrange multiplier $(\lambda_n^{\text{cut}})^{0,\text{opt}}$ is equal to the component j_2 of the previous gradient. Hence, the refinement indicator $\mathcal{I}_n^{\text{cut}}$ associated with the current refinement can be computed easily, at almost no additional cost, from the partial derivatives of the objective function with respect to the fine parameter in the subzones by

$$\mathcal{I}_n^{\text{cut}} = \left\| \sum_{i/K_i\subset Z_{n,j_1}^{\text{cut}}} \frac{\partial \mathcal{J}}{\partial p_i}(p_n^{\text{opt}}) \right\| = \left\| \sum_{i/K_i\subset Z_{n,j_2}^{\text{cut}}} \frac{\partial \mathcal{J}}{\partial p_i}(p_n^{\text{opt}}) \right\|. \tag{20}$$

Thus, we can define a large number of tentative cuttings producing possible refinements in each zone, and compute their corresponding refinement indicators.

Since the refinement indicators give us only a first order information on the decrease of the optimal objective function, one can select a set of cuttings associated with some of the highest values of the indicators, and not only with the largest one. Then, the objective function is minimized for all the parameterizations obtained by implementing each of these selected cuttings. And finally, the cutting defining the next zonation is the one leading to the largest decrease of the objective function. In the general case, this minimization phase is very expensive. It is thus important to keep the number of selected cuttings very small.

The initial value of the coarse parameter for these minimizations is obtained from the previous optimal value by duplicating the value in the split zone, i.e.

$$(m_{n,j_1}^{\text{cut}})^{\text{init}} = (m_{n,j_2}^{\text{cut}})^{\text{init}} = m_{n,j}^{\text{opt}} \qquad \text{and} \qquad (m_{n,j'}^{\text{cut}})^{\text{init}} = m_{n,j'}^{\text{opt}} \text{ for } j' \ne j_1, j_2. \tag{21}$$

The parameterization given by (17) is injective, and we can define its least-squares pseudoinverse by $\tilde{\mathcal{P}}_n = (\mathcal{P}_n^T \mathcal{P}_n)^{-1} \mathcal{P}_n^T$. It computes the mean value on each zone. When considering the scalar product weighted by the measure of each cell of the fine mesh, the previous pseudoinverse corresponds to the *projection* map given by

$$\begin{array}{rccl} \tilde{\mathcal{P}}_n : & (\mathbb{R}^{n_p})^{n_I} & \longrightarrow & (\mathbb{R}^{n_p})^n \\ & p_n = (p_{n,i})_{i\in I} & \longmapsto & m_n = (m_{n,j})_{1\le j\le n} \text{ with } m_{n,j} = \dfrac{\sum_{i\in I_{n,j}} \text{meas}(K_i)\, p_{n,i}}{\text{meas}(Z_{n,j})}. \end{array} \tag{22}$$

Then, (21) rewrites in the more compact form $(m_n^{\text{cut}})^{\text{init}} = \tilde{\mathcal{P}}_n^{\text{cut}} p_n^{\text{opt}}$.

At the end, the next $(n+1)$-zone parameterization map $\mathcal{P}_{n+1}$ has only one (vectorial) degree of freedom more, but it produces a significant benefit effect on the objective function.

Remark 3 *The adaptive parameterization technique is independent of the fine discretization $\mathcal{T}_h$ of the domain Ω. In particular, it is valid for regular or irregular meshes, structured or unstructured meshes.*

3.2 A generic algorithm for adaptive parameterization

The following algorithm can be applied to the estimation of distributed multidimensional parameters in any partial differential equation.

Initialization

-1. Choose an initial n_0-zone parameterization $\mathcal{P}_{n_0}$ and an initial coarse parameter $m_{n_0}^{\text{init}}$.

0. Compute $m_{n_0}^{\text{opt}} = \arg\min \mathcal{J}_{n_0}(m_{n_0})$ from $m_{n_0}^{\text{init}}$ and $p_{n_0}^{\text{opt}} = \mathcal{P}_{n_0} m_{n_0}^{\text{opt}}$.

Iterations For $n \geq n_0$, do until convergence:

1. Compute the fine gradient $g_n^{\text{opt}} = \nabla_p \mathcal{J}(p_n^{\text{opt}})$.
2. Choose a set of cuttings C_n defining parameterization candidates $(\mathcal{P}_n^k)_{k \in C_n}$.
3. For all $k \in C_n$, compute the corresponding indicator I_n^k from g_n^{opt} using (20).
4. Compute the largest indicator $I_n^{\max}$.
5. Select a subset $C_n(I_n^{\max}) \subset C_n$ of cuttings associated with the highest indicators.
6. For all $k \in C_n(I_n^{\max})$,
   compute $(m_n^k)^{\text{opt}} = \arg\min \mathcal{J}_n^k(m_n^k)$ from $(m_n^k)^{\text{init}} = \tilde{\mathcal{P}}_n^k p_n^{\text{opt}}$.
7. Keep the best cutting $k_0 = \arg\min_{k \in C_n(I_n^{\max})} \mathcal{J}_n^k((m_n^k)^{\text{opt}})$
   and set $\mathcal{P}_{n+1} = \mathcal{P}_n^{k_0}$, $m_{n+1}^{\text{opt}} = (m_n^{k_0})^{\text{opt}}$ and $p_{n+1}^{\text{opt}} = \mathcal{P}_{n+1} m_{n+1}^{\text{opt}}$.

This generic algorithm may be adapted by the user to match his/her needs. The convergence criterion can be specified, for instance: a small value of the objective function (meaning that the data are correctly fitted), a small value of the largest refinement indicator (meaning that there is no more important discontinuities left to discover), the maximum number of zones is reached... In addition, the user is free to choose the refinement strategy defining the full set of cuttings and the subset of those with the highest indicators.

Furthermore, as in [2], it is possible to add at the end of each iteration a coarsening step that allows zones corresponding to coarse parameters with close values to be merged.

3.3 Best cutting for multidimensional parameters

Let us suppose that we want to find among all possible cuttings the one corresponding to the best indicator. For example, this is useful in section 4 with the identity direct operator for which this best cutting is related to the best decrease of the objective function. But this *absolute* best cutting may also be useful in the general case to quantify the adequacy of the chosen set of cuttings to the inverse problem trough the ratio of the best indicator within the chosen set of cuttings to the absolute best indicator. A high ratio shows that there exists cuttings outside the chosen set that provide higher indicators.

In the scalar case, it is obvious from equation (20) that the best cutting for a given zone corresponds to follow the sign of the fine gradient (19). In the multidimensional case, this is no more so simple as finding the best cutting amounts to solve a discrete optimization problem. We give here several possible heuristics.

The simpler idea consists in defining the sign of a vector and then to use the same technique as in the scalar case. There are several possible definitions for the sign of a vector. Of course, none of them are fully satisfactory, but they can provide pertinent zonations, as it can be seen in section 4 where the sign of a three-dimensional vector is obtained by *majority vote* of the signs of all components. It could also have been defined as the sign of the sum of all components of the vector.

Another simple idea is to consider a zonation for each component of the parameter. Then, the algorithm for an n_p-dimensional parameter on a grid of size n_I would be equivalent to the algorithm for a scalar parameter on a grid of size $n_p \times n_I$. Again, with the example of section 4 where the data are colored images, $n_p = 3$ and n_I is the number of pixels. In this case, the algorithm for an RGB (Red/Green/Blue) image of size n_I would be equivalent to the algorithm for 3 grayscale images of the same size n_I. This kind of decomposition reminds the dawn of color photography where three black-and-white pictures were taken with three colored filters, and more recently the 3-LCD video projectors. The main drawback of this technique is that the compound zonation corresponding to the superimposition of the zonations for all the components may contain a lot of small zones. Indeed, the intersection of n_p zonations with n zones each can produce up to n^{n_p} zones. In other words, we would add up to $\sum_{k=0}^{n_p-1} C_{n_p}^k n^k$ degrees of freedom per component at each iteration! Which is completely in contradiction with the adaptive parameterization goal of providing the coarsest parameter, i.e. adding at most only one degree of freedom per component per iteration.

Keeping the attractive idea of computing a *local* indicator for each—scalar—component of the parameter separately[2], we can select at each iteration among the best cuttings for each component the one providing the highest *global* indicator. Then, this cutting is applied to all components, thus guaranteeing the addition of a single degree of freedom per component. Of course, this heuristics does not always provide the highest possible global indicator, but it seems reasonable to us and we suggest to use it.

3.4 Implementation issues

The first issue is the genericity of the implementation of the algorithm. Numerical results shown in [2] were obtained with a Fortran 77 program where the refinement algorithm was hard-coded within the model for the very specific problem of the estimation of hydraulic transmissivities in a parabolic flow equation set on a planar regular mesh. Once the technique has been validated on this example, it is very advisable to propose to the community a generic implementation that will allow to apply the refinement indicator adaptive parameterization algorithm to any problem implemented in a wide variety of programming languages.

The second issue is related to the definition of the set of cuttings, and more particularly to the implementation of it. In the general case, it is not possible to compute the true parameterization, and then, the adaptive parameterization technique is based on the definition of a set of cuttings, which would preferably be adapted to the problem. In [2], the two-dimensional mesh was supposed regular and rectangular, and the refinements were of four elementary kinds: horizontal cutting, vertical cutting, oblique cutting and checkerboard cutting. At each iteration, the indicators related to all possible cuttings within the selected family for all current zones were computed. Testing larger sets of cuttings, e.g. by enriching the elementary cutting offer, is very interesting because it will allow the user to use a priori information adapted to the problem. Furthermore, the ability to specify that some elementary cuttings should only apply to a particular part of the domain will allow to interpret more closely the a priori information.

The last issue concerns the performance of the implemented code. At each iteration, steps 3 and 6 of the algorithm correspond respectively to the computation of all the indicators associated with the chosen cuttings and to the actual minimization for the selected cuttings. Both steps involve fully independent computations that may run in parallel.

All these aspects show the necessity for high-level programming capabilities. In [3], we choose Objective Caml (OCAML) which is a fast modern high-level functional programming language based on sound theoretical foundation and formal semantics[3]. It is particularly well suited for the implementation of complex algorithms. The generic driver is written in OCAML and data exchange with the worker follow a simple and safe communication protocol that is implemented in several common programming languages including OCAML, C, C++ and FORTRAN. Here the generic tasks sent to the worker are `"cost"`, `"grad"` and `"optim"` respectively to implement the functions $p \longmapsto \mathcal{F}(p)$, $p \longmapsto \nabla \mathcal{J}(p)$ and $(m, \mathcal{P}) \longmapsto (m^{\text{opt}}, \mathcal{J}^{\text{opt}}) = (\arg\min \mathcal{J}_{\mathcal{P}}(m), \mathcal{J}_{\mathcal{P}}(m^{\text{opt}}))$. The design of a mini-language allow a flexible and convenient way to define the set of cuttings. Furthermore, *automatic* parallelization capabilities through the skeleton programming system provided by OCAMLP3L[4] has recently been successfully used in the field of Scientific Computation for domain decomposition applications, see [12, 13].

4 Application to the identity model

To illustrate the use of our OCAML implementation, detailed in [3], and the capabilities of the refinement indicators algorithm in the multidimensional case, presented in the previous section, we consider the simplest model ever: the identity model. Thus, in this section, we assume that $\mathcal{F} = \text{Id}$. Of course, in this very simple case, the inverse problem does not present any difficulty as the minimum of the objective function (18) is exactly zero, and it is obtained for $p^{\text{opt}} = d$. Note that the fine gradient is then simply given by the difference $\nabla \mathcal{J}(p) = p - d$. Nevertheless, assuming that the data is actually an image of size $n_I = n_x \times n_y$, then the mesh is regular and rectangular, the cells are called *pixels*, and the application of the refinement indicators algorithm amounts in this case to determine groups of pixels with the same color—or gray level. This is the basic idea of image segmentation for edge detection, shape recognition, compression/simplification, and so on...

The segmentation of a grayscale image is simply a scalar example, but even in this case, the determination of the best cutting providing the highest decrease of the objective function is not an easy task, see [4]. The case of color images is more complicated as the color information is typically multidimensional. In computer applications, it is usually represented using a three-dimensional color space—hence $n_p = 3$: either RGB, for the Red/Green/Blue color cube, HSV, for the Hue/Saturation/Value color cone, or HSL, for the Hue/Saturation/Lightness color double cone. Of course, it is

[2] For which the *best* cutting associated with the highest local indicator simply follows the sign of the—scalar—fine gradient for each component.
[3] See `http://www.ocaml.org/`.
[4] See `http://www.ocamlp3l.org/`.

possible to define a scalar discrete representation of colors, e.g. by packing the 24-bit RGB encoding into a single integer ranging from 0 to $2^{24}-1$. But, such a nonlinear mapping produces weird effects when averaging on a zone: the mean color between red pixels and blue pixels may become an odd green, instead of the natural purple value. We consider here the RGB representation of color images.

In the case of the identity model, the operator ${\mathcal{F}_n}^T \mathcal{F}_n$ of subsection 2.3 reduces to the diagonal matrix ${\mathcal{P}_n}^T \mathcal{P}_n$ which gives the numbers of pixels per subzone. Then, equation (15) rewrites

$$\jmath_1^{\text{opt}} - \jmath_2^{\text{opt}} = \frac{1}{2} \frac{n_j}{n_{j_1} n_{j_2}} I_{n,j}^2 \tag{23}$$

where n_j is the number of cells in the selected zone $Z_{n,j}$, and n_{j_1} and n_{j_2} are respectively the number of cells in the subzones Z_{n,j_1}^{cut} and Z_{n,j_2}^{cut}. Although the decrease of the objective function provided by a cutting is proportional to the square of the refinement indicator associated with this cutting, unfortunately the proportionality coefficient depends on the cutting itself. Nevertheless, for high values of a, the function $x \longmapsto \frac{a}{x(a-x)}$ is almost flat in the neighborhood of $a/2$, and the highest indicator almost yields the best decrease of the objective function, at least during the first iterations of the algorithm. Furthermore, there is no more interaction between the pixels during the minimization process. Hence, there is no need to compute the refinement indicators for all zones at each iteration, but rather, it is advisable to compute only the indicators related to the two most recent subzones, and to keep in memory the values for the unchanged zones. An optimized version of the refinement indicators algorithm dedicated to the diagonal models is described in [4].

For this first multidimensional application of the algorithm, we have simply defined the *pseudo*-sign of the fine vector gradient on each pixel by the sum of the signs for each component. Let sgn be the sign function on the field of real numbers (i.e., $\text{sgn}(x) = -1$ when $x < 0$, $\text{sgn}(x) = +1$ when $x > 0$ and $\text{sgn}(0) = 0$). Then, we define the sign of any triple by $\text{sgn}(x,y,z) = \text{sgn}(\text{sgn}(x) + \text{sgn}(y) + \text{sgn}(z))$. As developed in subsection 3.3, this is a very basic heuristics to deal with the multidimensional case, but it is sufficient for our present illustration purposes.

In subsection 3.2, the choice for the refinement strategy defining the set of cuttings and the subset of those with the highest indicators was let to the user. Here, we consider two strategies. The first one is a mild flavor of the *best* strategy consisting in choosing at each iteration the cutting related to the highest indicator which guarantees in the case of the identity model the best descent of the objective function. In fact, we follow the pseudo-sign of the gradient as defined above. This strategy actually provides an image segmentation technique. The second one is called the *dichotomy* strategy. It is an example of strategy based on elementary cuttings deployed in all zones. The only cuttings allowed are those splitting a zone vertically or horizontally in its middle. This illustrates what could happen in the general case where the best strategy is not necessarily optimal and where the adaptive parameterization technique is based on the choice for a set of cuttings. Notice that this could provide interesting creative filters for image processing.

The next Figures follow the same scheme. They are all made of four images. The data image d is displayed on the lower left corner. The segmented image p_n^{opt} is displayed—in true colors—on the lower right corner, and the corresponding n-zone zonation $\mathcal{P}_n$ is displayed in false colors on the upper right corner. The pseudo-sign of the fine gradient $\nabla \jmath\,(p_n^{\text{opt}})$ is displayed on the upper left corner where positive values are depicted in white, negative values in black and small absolute values in gray. Some monitoring counters are also displayed. This is actually the interactive display of the OCAML driver at runtime, see [3].

The first series of experiments exploit the best strategy, hence we may expect the fastest decrease of the objective function. The results obtained after iterations number 1, 2, 5 and 10 are respectively represented in Figure 2, 3, 4 and 5.

At the beginning, in Figure 2, the initial zonation contains only one zone, the initial parameter is $m_1^{\text{init}} = (0,0,0)$—i.e., a black image—and the initial value of the objective function is $\jmath_1^{\text{init}} = 8.0\,10^9$. After the initialization phase, the objective function is $\jmath_1^{\text{opt}} = 1.2\,10^9$, meaning that 62% of the data are explained[5]. Then, the pseudo-sign of the first fine gradient g_1^{opt} already depicts the head of the kangaroo. This cutting is associated with the—high—refinement indicator $I_1^{\text{best}} = 1.0\,10^7$.

After the fifth iteration, in Figure 4, 90% of the data are explained and the main features of the original picture are visible. The last highest indicator was already only 7.4% of the first one. With 10 zones, in Figure 5, 94% of the data are explained. For the eye, the segmented image is very close to the original one. The last highest indicator was a flat 1.7% of the first one.

The second series of experiments use the dichotomy strategy. The result obtained after 2 and 100 iterations are respectively represented in Figures 6 and 7.

The first iteration is exactly the same as with the best strategy in Figure 2. Then, the first cutting does not reveal at all the head of the kangaroo. The image with two zones, in Figure 6, only explain one percent more of the data than the one

[5] At iteration n, the percentage of data explained is defined as $100 \times \left(1 - \sqrt{\frac{\jmath_n^{\text{opt}}}{\jmath_1^{\text{init}}}}\right)$.

with one zone. The corresponding indicator is only 28% of the first indicator with the best strategy. Indeed, the chosen family of cuttings is not well suited for the problem. Even at iteration number 100, in Figure 7, only 87% of the data are explained and the last highest indicator was about 1% of the first one.

Of course, this strategy is inefficient for the segmentation of images, but exploring a subset of possible cuttings at each iteration acts as a preconditioner, and it may also help in avoiding to fall into local minima in the general case. Notice that after 100 iterations of the dichotomy strategy, the gradient keeps very sharp details that has already disappeared after only 10‘iterations of the best strategy. We could have done a better job by allowing any vertical and horizontal cuttings, but there would have been much more indicators to evaluate. The dichotomy run took 54 seconds, compared to the 3 seconds for the best run (only add `tind` and `topt`, as the total time `ttot` also takes into account the display).

5 Conclusions

We give a general formulation of the refinement indicator algorithm for the estimation of *vector valued* distributed parameters in *any* partial differential equation.
The main findings are:
(i) in the multidimensional case, the refinement indicator associated with a new degree of freedom is the norm of the corresponding Lagrange multiplier.
(ii) In the linear case, the refinement indicator also measures the decrease of the optimum least-squares data misfit objective function when the corresponding degree of freedom is open.
(iii) When applied to the inversion of the identity, the vector valued version of the refinement indicators algorithm provides a surprisingly powerful tool for the segmentation of color images (where colors are three-dimensional vectors in the RGB model).

Acknowledgments

The authors thank the GDR MOMAS for providing partial financial support.

References

[1] V. Akcelik, G. Biros, and O. Ghattas. Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation. In *Proc. of the 15th Internat. Conf. for High Performance Computing and Communications*, 2002.

[2] H. Ben Ameur, G. Chavent, and J. Jaffré. Refinement and coarsening indicators for adaptive parametrization: application to the estimation of the hydraulic transmissivities. *Inverse Problems*, 18:775–794, 2002.

[3] H. Ben Ameur, F. Clément, and P. Weis. An ocaml implementation of the refinement indicators algorithm. *Submitted*, 2006.

[4] H. Ben Ameur, F. Clément, and P. Weis. An optimal control algorithm for image segmentation. *Submitted*, 2006.

[5] H. Ben Ameur and B. Kaltenbacher. Regularization of parameter estimation by adaptive discretization using refinement and coarsening indicators. *Inverse and ill posed problems*, 10(6):561–583, 2002.

[6] C. Bunks, F. M. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *Geophysics*, 60:1457–1473, 1995.

[7] C. Chardaire-Rivière, G. Chavent, J. Jaffré, and J. Liu. Multiscale representation for simultaneous estimation of relative permeabilities and capillary pressure, paper SPE 20501. In *65th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, New Orleans, Louisiana, September 23–26, 1990*, pages 303–312. Society of Petroleum Engineers, Richardson, Texas, 1990.

[8] E. Chardigny, P. Siegel, R. Mosé, and P. Ackerer. Parameter identification for complex groundwater systems. In A. Aldama et al, editor, *Computational Methods in Water Resources XI, Cancun, Mexico, July 22–26, 1996*, pages 305–312. Computational Mechanics Publications, 1996.

[9] G. Chavent and R. Bissell. Indicator for the refinement of parametrization. In M. Tanaka and G. S. Dulikravich, editors, *Inverse Problems in Engineering Mechanics (Proceedings of the International Symposium on Inverse Problemsin Engineering Mechanics 1998 (ISIP'98), Nagano, Japan)*, pages 309–314. Elsevier, 1998.

[10] G. Chavent and J. Liu. Multiscale parametrization for the estimation of a diffusion coefficient in elliptic and parabolic problems. In A. El Jai and M. Amouroux, editors, *5th IFAC Symposium on Control of Distributed Parameter Systems, Perpignan, June 26–29, 1989*, pages 315–324. Université de Perpignan, France, 1989.

[11] F. Clément, G. Chavent, and S. Gómez. Migration-based traveltime waveform inversion of 2-D simple structures: A synthetic example. *Geophysics*, 66:845–860, 2001.

[12] F. Clément, V. Martin, A. Vodicka, R. Di Cosmo, and P. Weis. Domain decomposition and functional programming with OCamlP3l. In M. Papadrakakis, E. Onate, and B. Schrefler, editors, *Proc. of the Internat. Conf. on Computational Methods for Coupled Problems in Science and Engineering*. CIMNE, 2005.

[13] F. Clément, V. Martin, A. Vodicka, R. Di Cosmo, and P. Weis. Domain decomposition and skeleton programming with OCamlP3l. *Parallel Computing*, 2006.

[14] H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, Dortdrecht, 1996.

[15] J. Liu. A multiresolution method for distributed parameter estimation. *SIAM J. Sci. Comput.*, 14:389–405, 1993.

[16] J. Liu. A sensitivity analysis for least-squares ill-posed problems using the Haar basis. *SIAM J. Numer. Anal.*, 31:1486–1496, 1994.

[17] T. L. Pratt, J. F. Dolan, J. K. Odum, W. J. Stephenson, R. A Williams, and M. E. Templeton. Multiscale seismic imaging of active fault zones for hazard assesment: A case study of the Santa Monica fault zone, Los Angeles, California. *Geophysics*, 63:479–489, 1998.

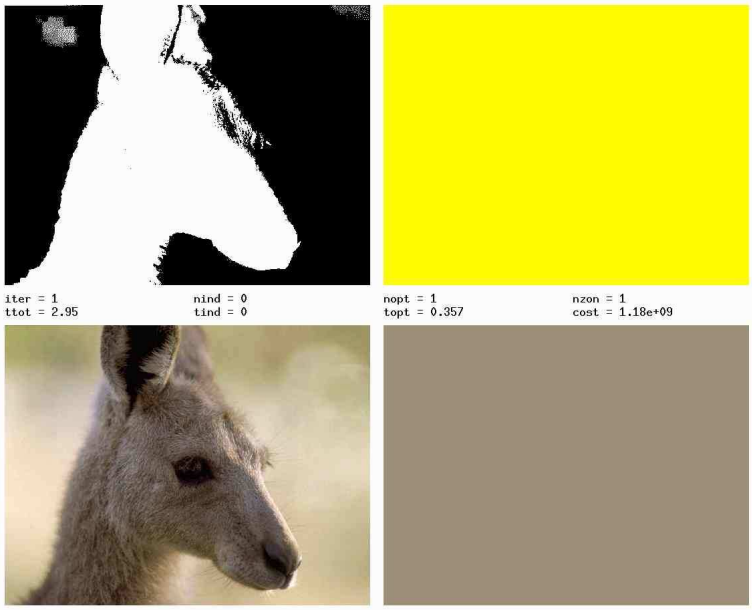


Figure 2: Best strategy, 1 zone. Bottom: data image d (left), segmented image p_1^{opt} (right). Top: pseudo-sign of the gradient g_1^{opt} (left), 1-zone zonation $\mathcal{P}_1$ (right).

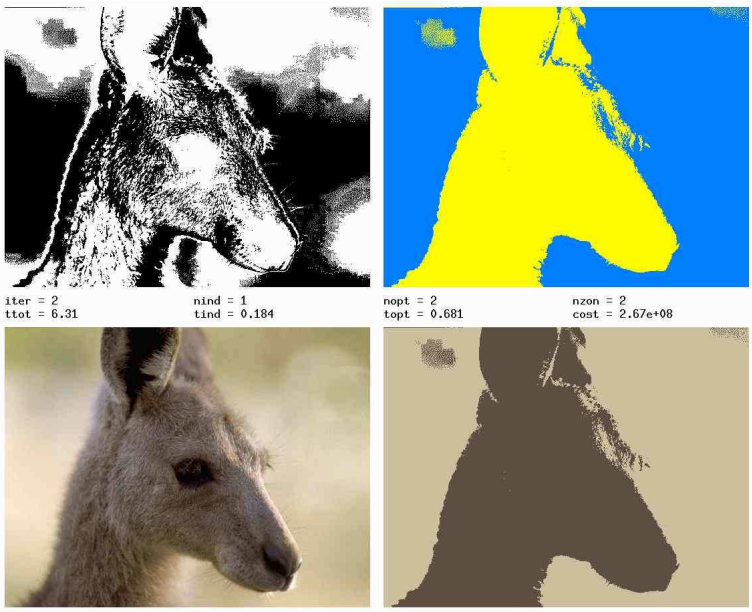


Figure 3: Best strategy, 2 zones. Bottom: data image d (left), segmented image p_2^{opt} (right). Top: pseudo-sign of the gradient g_2^{opt} (left), 2-zone zonation $\mathcal{P}_2$ (right). The refinement indicator for the first cutting was $I_1^{\text{best}} = 1.0\,10^7$.