Refinement indicators

Documentation

Version 1.1

Hend Ben Ameur François Clément Pierre Weis

2014-02-28

1 Outline

Ref-indic is an adaptive parameterization platform using refinement indicators.

2 What is Ref-indic?

Ref-indic implements the adaptive parameterization algorithm using refinement indicators to solve inverse problems set as minimization problems of the form

$$\min_p \frac{1}{2}\|d - \mathcal{F}(p)\|^2,$$

where $\mathcal{F}$ models the cause-to-effect relation from parameter p to measures d (it is the function to invert). Typically, computing measures $\mathcal{F}(p)$ for some parameter p involves the resolution of a set of PDEs depending on the distributed parameter p (e.g., a function of the space variable).

Parameter p is searched for under the form $p = \mathcal{P}m$ where $\mathcal{P}$ is a piecewise constant parameterization operator splitting the domain of function p into a given number of—constant—pieces; m is called *coarse parameter* (one value per piece), and in contrast, p is now called *fine parameter* (one value per point). Piecewise constant parameterization operators are fully and uniquely defined by the choice of a partition of the space domain (with no empty part), hence the identification of partitions and parameterization operators.

To avoid over-parameterization problems, pieces are progressively split one at a time in an optimal manner to build an optimal sequence of parameterization operators $(\mathcal{P}_n)_{n=1,2,\ldots}$ where each $\mathcal{P}_n$ is associated with a partition into n pieces. Hence the name "adaptive parameterization algorithm".

The approach is efficient for inverse problems in which modeling operator $\mathcal{F}$ is almost linear with respect to low frequency contents of the parameter p, and may become more and more nonlinear as the frequency contents increase. It is not suited at all to the case where low frequencies of the unknown parameter are related to a highly nonlinear behavior of the model, as for seismic inversion.

3 Algorithm

Input data are a dimension management (`vector`, or `best_component_only`), and an initial vector partition $\mathcal{P}_1$. An user-supplied external program provides the following functions:

$$\texttt{optim}: \ \mathcal{P} \longmapsto (m^\star, J^\star) = (\arg\min_m J(\mathcal{P}m), J(\mathcal{P}m^\star)), \quad \texttt{grad}: \ p' \longmapsto \nabla_p J(p').$$

For `vector` dimension management, all components are always subject to the same scalar partition. For `best_component_only` dimension management, different components may be subject to different scalar partitions.

A *cutting* $c_{\mathbf{P}}$ splits some part $\mathbf{P}$ of some partition $\mathcal{P}$ into nonempty disjoint subparts $\mathbf{P}^+$ and $\mathbf{P}^-$.

Initialization

0. Compute optimal parameter associated with initial vector partition $\mathcal{P}_1$,

$$(m_1, J_1) = \texttt{optim}(\mathcal{P}_1).$$

Iterations For $n \geq 1$, until stopping criterion is satisfied, do:

1. Compute fine gradient, $g_n = \texttt{grad}(\mathcal{P}_n m_n)$.

2a. Compute scalar first order indicator for all scalar cuttings in all parts for all components,

$$\forall k, \ \forall \mathbf{P} \in \mathcal{P}_n^k, \ \forall c_{\mathbf{P}} = (\mathbf{P}^+, \mathbf{P}^-) \in C_{\mathbf{P}}, \quad I^k_{c_{\mathbf{P}}} = \left| \sum_{i \in \mathbf{P}^+} (g_n^k)_i - \sum_{i \in \mathbf{P}^-} (g_n^k)_i \right|,$$

2b. For `vector` dimension management, compute vector first order indicator for all vector cuttings in all parts,

$$\forall \mathbf{P} \in \mathcal{P}_n, \ \forall c_{\mathbf{P}} \in C_{\mathbf{P}}, \quad I_{c_{\mathbf{P}}} = \|(I^k_{c_{\mathbf{P}}})_k\|^2.$$

3. Select best candidate cuttings (highest first order indicators),
`vector`: $C^\star \subset \bigcup_{\mathbf{P} \in \mathcal{P}_n} C_{\mathbf{P}}$, `best_component_only`: $C^\star \subset \bigcup_k \bigcup_{\mathbf{P} \in \mathcal{P}_n^k} C_{\mathbf{P}}$.

4. Compute exact indicator for all selected cuttings,

$$\forall c^\star \in C^\star, \quad (m_{c^\star}, J_{c^\star}) = \texttt{optim}(\mathcal{P}_{c^\star}).$$

5. Pick best selected cutting (lowest exact indicator): $c^{\text{opt}} = \arg\min_{c^\star} J_{c^\star}$

$$(\mathcal{P}_{n+1}, m_{n+1}, J_{n+1}) = (\mathcal{P}_{c^{\text{opt}}}, m_{c^{\text{opt}}}, J_{c^{\text{opt}}}).$$