Computers and Structures 147 (2015) 14–25

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

An automatic multilevel refinement technique based on nested local meshes for nonlinear mechanics

L. Barbié [a,b], I. Ramière [a,*], F. Lebon [b]

[a] CEA, DEN, DEC, SESC, F-13108 Saint-Paul Lez Durance, France
[b] LMA, CNRS, UPR 7051, Aix-Marseille Univ, Centrale Marseille, 31, Chemin Joseph Aiguier, F-13402 Marseille Cedex 20, France

ARTICLE INFO

Article history:
Accepted 1 October 2014
Available online 6 November 2014

Keywords:
Local discontinuity
Adaptive mesh refinement
Uniform non-data-fitted meshes
A posteriori error estimation
Nonlinear solid mechanics
Pellet–cladding interaction

ABSTRACT

In this paper an adaptive multilevel mesh refinement method, coupled with the Zienkiewicz and Zhu a posteriori error estimator, is applied to solid mechanics with the objective of conduct reliable nonlinear studies in acceptable computational times and memory space. Our automatic approach is first verified on linear behaviour, on 2D and 3D simulations. Then a nonlinear material behaviour is studied. Advantages and limitations of the local defect correction method in solid mechanics problems in terms of refinement ratio, error level, CPU time and memory space are discussed. This kind of resolution is also compared to a global h-adaptive resolution.

1. Introduction

Adaptive refinement methods (e.g. [1–5]) are devoted to solve problems with various characteristics length-scale in acceptable computational times and memory space. The aim of this paper is to deal with problems having local discontinuities. Among the refinement methods available, those inducing small elements (h-refinement method [1,6] for example) are more efficient to simulate local discontinuities than those increasing the order of the polynomial basis (p-refinement method [2,7] for example). Moreover the pollution error [8,9] (i.e. the residual error due to the non-refined part of the problem) depends on the mesh size and may be controlled by refining the element size but not by modifying the order of the polynomial basis. That is the reason why we decided to use adaptive mesh refinement (AMR) techniques. One of the constraints of the study was to use an existing industrial solver, that means to change only pre-processing and post-processing operations. Thus, we chose to use local multi-grid methods [4] consisting in generating local sub-grids with finer and finer discretisation step. Furthermore, it induces simple meshes (uniform, structured and regular). As we study elliptic problems discretised by the finite element (FE) method, it is interesting to use structured regular meshes for the reason that it produces well-defined problems. The local defect correction (LDC) [10,4] method was retained because it is not specific to flux conservative problems.

The refinement process is piloted thanks to the well-known Zienkiewicz and Zhu (ZZ) a posteriori error estimator [11] founded on stress smoothing.

This paper is based upon Barbié et al. [12] but the current version includes results obtained from additional research on the efficiency of the coupled LDC-ZZ method for linear behaviour (see also [13]), a further study on three-dimensional test case where different initial meshes are compared and an extended work concerning the nonlinear test case, particularly with the use of refinement ratio greater than 2.

2. Adaptive mesh refinement approach

2.1. Local defect correction method

The local defect correction (LDC) method was introduced by Hackbusch [10]. Its convergence was also proved by Hackbusch [14]. It is based on the multi-grid process [15]. A global coarse grid is used on the whole domain, and only local fine sub-grids are set on areas where more precision is required. An example of nested grids is shown on Fig. 1. The local fine grid lies on a zone of interest defined on the coarse grid. Such type of local sub-grid can be defined recursively until reaching the desired accuracy.

[*] Corresponding author.

http://dx.doi.org/10.1016/j.compstruc.2014.10.008
0045-7949/© 2014 Civil-Comp Ltd and Elsevier Ltd. All rights reserved.

Prolongation and restriction operators are defined to link several levels of computation. Coarse and fine problems are then sequentially computed until the solution converges on the coarser grid. Such an iterative process is traditionally represented by a $\wedge$-cycle, as on Fig. 2.

2.1.1. Problem definition

Let us consider the problem $(\mathcal{P})$ defined on an open domain Ω of boundary Γ :

$$(\mathcal{P}) : \begin{cases} \mathcal{L}(u) = f \text{ in } \Omega \\ \text{B.C. on } \Gamma \end{cases} \tag{1}$$

with:

$$\left\{ \begin{array}{lll} \mathcal{L} & : & \text{usually nonlinear operator} \\ u & : & \text{solution} \\ f & : & \text{right-hand member} \\ \text{B.C.} & : & \text{boundary conditions} \end{array} \right.$$

A set of nested domains $\Omega_l, 0 \leqslant l \leqslant l^*$, with $\Omega_0 = \Omega$ and l^* the number of levels, is then defined. Each domain is discretised by a grid G_l of boundary Γ_l. The space step h_l of the mesh G_l is defined by $h_l = h_{l-1}/r = h_0/r^l$ with r the refinement ratio. The local discrete problem on each grid G_l at iteration k writes:

$$(\mathcal{P}_l^k) : \begin{cases} \mathcal{L}_l(u_l^k) = f_l^k \text{ in } G_l \\ \text{appropriate B.C. on } \Gamma_l \end{cases} \tag{2}$$

where $\mathcal{L}_l$ is the discrete operator associated to $\mathcal{L}_{|\Omega_l}$ on G_l and $f_l^0 = f_{|G_l}$.

The boundary conditions will be specified during the prolongation step while the right-hand member f_l^k will be defined during the restriction step.

2.1.2. Prolongation step: boundary conditions

As recommended by [13], at the prolongation step the problem $(\mathcal{P}_l^k)$ is solved with $f_l^k = f_l^{k-1}$.

On the coarsest grid G_0, the boundary conditions of the whole problem are applied.

The boundary conditions on the other grids $G_l, 1 \leqslant l \leqslant l^*$ are represented on Fig. 3:

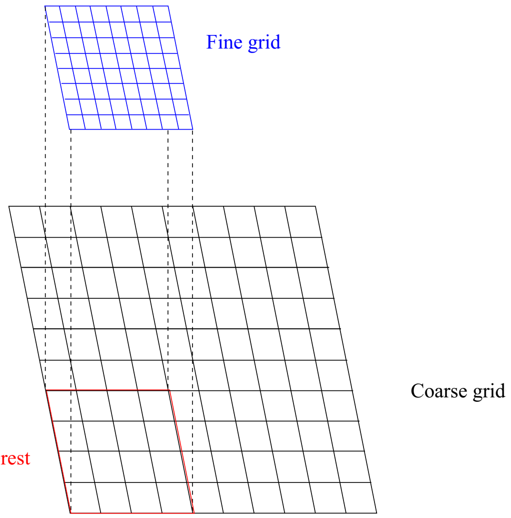

Fig. 1. Example of nested meshes used in LDC method.

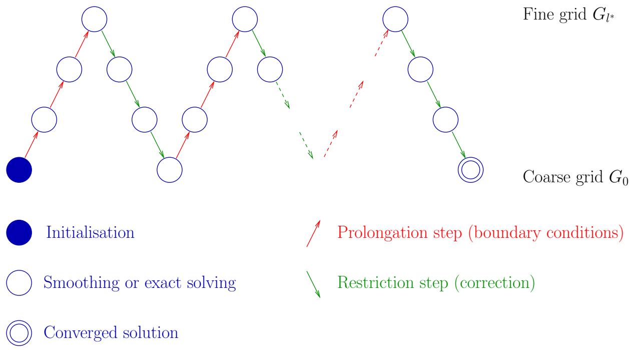
Fine grid G_{l^*}
Coarse grid G_0
Initialisation
Prolongation step (boundary conditions)
Smoothing or exact solving
Restriction step (correction)
Converged solution

Fig. 2. Representation of LDC process: $\wedge$-cycles.

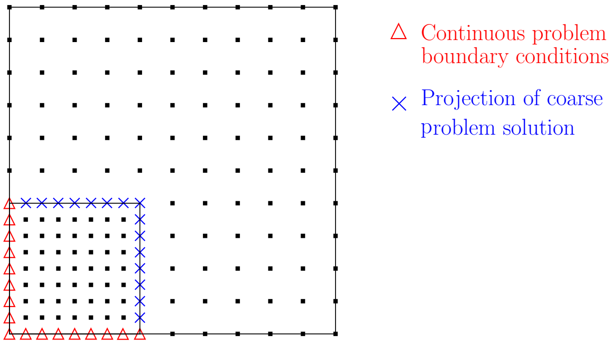
Continuous problem boundary conditions
Projection of coarse problem solution

Fig. 3. Prolongation step: boundary conditions on G_l ($l \neq 0$).

- If $\Gamma_l \cap \Gamma \neq \varnothing$, the boundary conditions of the original problem $(\mathcal{P})$ are used.
- On the other part of the boundary, Dirichlet boundary conditions are applied. A projection operator P_{l-1}^l applied on the next coarser solution u_{l-1}^k enables us to obtain the Dirichlet values.

2.1.3. Restriction step: defect correction

At the restriction step, the boundary conditions defined on the prolongation step are kept to solve the problem $(\mathcal{P}_l^k)$. For each grid level $l, 0 \leqslant l \leqslant l^* - 1$, the restriction step consists in correcting the right-hand side of the problem $(\mathcal{P}_l^k)$ via a defect calculated from the next finer solution u_{l+1}^k.

Two sets of nodes of G_l have to be defined, see Fig. 4. A_l contains the nodes of the grid G_l strictly included on the domain discretised by G_{l+1}. $\mathring{A}_l$ is made up of the interior nodes of A_l (in the sense of the discretisation scheme).

First, the solution of the problem $(\mathcal{P}_{l+1}^k)$ is restricted to the nodes of A_l :

$$\tilde{u}_l^k(x) = (R_{l+1}^l u_{l+1}^k)(x) \quad \forall x \in A_l \tag{3}$$

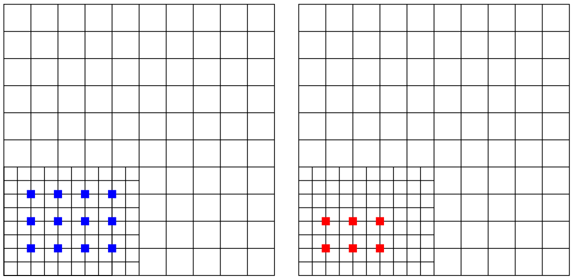

Fig. 4. Restriction zone A_l on the left and correction zone $\mathring{A}_l$ on the right (e.g. for operator Δ).