

Implementation of a XFEM toolbox in Diffpack

Md Naim Hossain, Daniel Alves Paladim, Frank Vogel, Stéphane P. A. Bordas

inuTech GmbH, Fürther Str. 212, 90429 Nuremberg, Germany, mdnaim.hossain@inutech.de,
frank.vogel@inutech.de

Cardiff School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University,
Queen's Buildings, The parade, Cardiff CF24 3AA, Wales, UK, AlvesPaladimD@cardiff.ac.uk,
bordasS@cardiff.ac.uk

Key Words: *Extended Finite Element Method, Diffpack, Enrichment Function, Levelset Method.*

ABSTRACT

The Diffpack Development Framework is an object-oriented software environment for the numerical solution of partial differential equations (PDEs). By its design, Diffpack intends to close the gap between black-box simulation packages and technical computing environments using interpreted computer languages. The framework provides a high degree of modeling flexibility, while still offering the computational efficiency needed for most demanding simulation problems in science and engineering. Technically speaking, Diffpack is a collection of C++ libraries with classes, functions and utility programs. The numerical functionality is embedded in an environment of software engineering tools supporting the management of Diffpack development projects. Diffpack supports a variety of numerical methods with distinct focus on the finite element method (FEM) but has no inherent restrictions on the types of PDEs and therefore applications to be solved.

The key point of partition of unity enriched methods such as XFEM and GFEM is to help capture discontinuities and singularities or large gradients in solutions, which are not well resolved by h or p refinement [1]. The general idea is that the mesh need not conform to the moving boundaries so that minimal or no remeshing is required during the analysis. Our main motivation is to provide a generic implementation of enrichment within a flexible C++ environment, namely the Diffpack platform. The work was inspired by some of our earlier work [6,9] and that of other colleagues [5,7,8].

We demonstrate how object-oriented programming is particularly useful for the treatment of data structures and operations associated with XFEM: mesh-geometry interaction, non-standard integration rules, application of boundary conditions, treatment of level set data [2,6]. We detail the implementation of such features and verify and validate their implementation based on [5]. We show results based on unshifted, shifted [1] and study the behaviour of the stable generalized finite element method (SGFEM) to avoid blending effects and help control the conditioning of the system matrix [4]. For integration of elements cut by interface we use an in-house Delaunay Triangulation algorithm proposed by [3,5] and presented in detail in a companion paper.

REFERENCES

- [1]. Fries, T.P.- and Belytschko T, The extended /generalized finite element method : An overview of the method and its applications. *Int. J. Numer. Meth. Engng.*, Vol. **84**:253–304, 2010.
- [2]. Hans Petter Langtangen, *Computational Partial Differential Equations, Numerical Methods and Diffpack Programming*, Second edition, Springer, 2002.
- [3]. D. F. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, Vol. **24**, No. 2, 1981.
- [4]. I. Babuška and U. Banerjee, *Stable Generalized Finite Element Method (SGFEM)*, 2011.
- [5]. Sukumar, N., N. Moës, B. Moran, and T. Belytschko, Extended finite element method for three-dimensional crack modelling. *Int. J. Numer. Meth. Engng.*, Vol. **48**, 1549–1570, 2000.
- [6]. Bordas S, Nguyen PV, Dunant C, Guidoum A, Nguyen-Dang H, An extended finite element library, *International Journal for Numerical Methods in Engineering*, Vol. **71** (6), 703-732, 2007
- [7]. R. Chamrová and B. Patzák: Object-oriented programming and the extended finite-element method. *Engineering and Computational Mechanics*, Vol. **163** (EM4):271-278, 2010.
- [8]. André Massing, Mats G. Larson, Anders Logg, Efficient implementation of finite element methods on non-matching and overlapping meshes in 3D, 2012.
- [9]. Multiscale fracture open-source : PERMIX Project <https://sites.google.com/site/permixproject>.