Implementation of a XFEM toolbox in Diffpack

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

1 inuTech GmbH, Nuremberg, Germany
2 Cardiff School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University, Wales, UK

International Conference on Extended Finite Element Methods – XFEM 2013, September 11 – 13, 2013, Lyon, France
Outline

• Objective
• Diffpack
• XFEM
• Diffpack FEM module
• Object-oriented approach of XFEM using Diffpack
• Numerical example of XFEM toolbox in Diffpack
• Conclusion
• Acknowledgement
Objective
Innovative Numerical Technologies

Objective

• Our main objective is to provide a generic implementation of enrichment within a flexible C++ environment, namely the Diffpack platform.

• Flexibility of using existing Diffpack FEM classes with some modification.

• How object-oriented programming is useful for the treatment of data structures and operations associated with XFEM.

• Implementation of different features of XFEM with verifying and validating based on some numerical example.
Diffpack
**Diffpack**

- Diffpack is a object-oriented software environment with main emphasis on numerical solutions of partial differential equations.

- Diffpack is a collection of C++ libraries with classes, functions and other utility programs.

- Diffpack supports a variety of numerical methods with distinct focus of FEM but has no inherent restrictions of the type of PDEs.
**Diffpack® is a Development Environment**

- **PDEs**

\[
K(S) = \lambda_o(S) + \lambda_w(S),
\]

\[
f(S) = \lambda_w(S) / K(S),
\]

\[
h(S) = -\lambda_o(S)f(S)P_c(S),
\]

\[
\lambda_w = k_w(...),
\]

\[
\lambda_o = k_o(...).
\]

\[
- \nabla \cdot [K(S)\nabla P] = q,
\]

\[
S_i + \nabla \cdot [vf(S)] = \nabla \cdot (h(S)\nabla S),
\]

\[
v = -K(S)\nabla P
\]

Object-Oriented (C++) Tools for the numerical Modeling and Solution of Differential Equations
Diffpack® Summary

• is a problem-solving environment for simulation problems
• are numerical libraries for PDE solution (> 600 C++ Classes)
• simplifies the solver development process significantly
• nicely complements standard FEM-programs

Learn more about it from http://www.diffpack.com

Innovative Numerical Technologies
Extended Finite Element Method (XFEM)
Level sets and Enrichment

- Level set method is a numerical technique for tracking moving boundaries and interfaces.

\[ \varphi(x, t) = \pm \min_{x_I \in \Gamma(t)} \| x - x_I \| \]

- Enrichment is enriching the finite element approximation by additional functions that model the flaws and other geometric entities.

\[ \psi(x) = H(\phi(x)) = \begin{cases} 0 : & \phi(x) \leq 0 \\ 1 : & \phi(x) > 0 \end{cases} \]

- Strong discontinuity (Heaviside function)

- Weak discontinuity (kink)
Shifting:
- Some elements do not hold kronecker delta property.
- Needs shifting using $[\psi(x) - \psi(x_i)]$ to maintain Kronecker delta property in FE approximation.

Blending:
- Blending elements create due to localization of enrichment function and it blends in between enriched part and non-enriched part.
- Problem occurs due to non-partition of unity properties.
- Damage the convergence rate.

Remedies:
- Using Ramp function vanishing the enrichment in Blending Elements.

\[ u^h(x) = \sum_{i \in I} N_i(x)u_i + \sum_{i \in I^*} N_i^*(x) \cdot R(x) \cdot [\psi(x) - \psi(x_i)]a_i \]
In 3 ways we can evaluate the enrichment functions:

- **Unshifted**: \( \Phi^h(x_i) = \Phi_i \), returns the value at a given point
  
  Problem: Destroy the cronecker delta property of the field approximation.
  
  Solution: Using enrichment function by,

- **Shifted mode**: \( \Phi(x) - \Phi(x_i) \)
  
  Conserves the cronecker delta property of FE function.

- **Using Stable Generalized Finite Element Method** (according Babuska and Banerje, 2012):
  
  \[
  \Phi(x) - \sum N_i(x) \Phi(x_i)
  \]

  Conserves the cronecker delta property and avoid blending problems by recovering the damage of the convergency providing well conditioned stiffness matrix.
Diffpack FEM Module
Object Oriented Design – Fundamental principles

• Encapsulation - Clustering together data and functionality (methods).

• Inheritance - Reuse of existing code by derived classes.

• Abstraction/Polymorphism - Transparent use of derived classes.

• Communication using message – General interface and safe data handling.
• Diffpack has more than 600 C++ Classes solving with FDM, FEM, FVM and working with several other methods.
• We will discuss how to solve a poisson problem using FEM classes.
Way of creating a simulator and solve problem by Diffpack FEM module

- **STEP 1:** Write down the weak formulation of the PDE
- **STEP 2:** Derive a simulator class from FEM
- **STEP 3:** Equip your simulator with problem-specific class handles
  - GridFE, FieldFE
  - DegFreeFE
  - LinEqAdmFE
- **STEP 4:** Implement member functions ('overwriting' virtual FEM-functions)
  - Scan
  - solveProblem
  - fillEssBC
  - Integrands
  - resultReport
- **STEP 5:** Initialize Diffpack, the simulator, and call the main member functions.
Object-oriented approach of XFEM using Diffpack
Derived from existing FEM classes

<table>
<thead>
<tr>
<th>FEM classes</th>
<th>XFEM classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridFE</td>
<td>GridFE</td>
</tr>
<tr>
<td>FieldFE</td>
<td>FieldXFE</td>
</tr>
<tr>
<td>FieldsFE</td>
<td>FieldsXFE</td>
</tr>
<tr>
<td>DegFreeFE</td>
<td>DegFreeXFE</td>
</tr>
<tr>
<td>LinEqAdmFE</td>
<td>LinEqAdmX</td>
</tr>
<tr>
<td>ElmitgRules</td>
<td>XElmitgRules</td>
</tr>
<tr>
<td>ElmMatVec</td>
<td>XElmMatVec</td>
</tr>
<tr>
<td>ErrorNorms</td>
<td>XErrorNorms</td>
</tr>
<tr>
<td>FEM</td>
<td>XFEM</td>
</tr>
<tr>
<td>FiniteElement</td>
<td>XFiniteElement</td>
</tr>
</tbody>
</table>

- Finite element grid with the information nodal coordinates, element topology and boundary indicating nodes.
- Collect the finite element vector and scalar field.
- Relates degree of freedom in a linear system to the nodal values of discrete FEM and XFEM.
- Provides a user-friendly interface the various modules involved when solving a linear system of equations.
- Performs numerical integration over FEM and XFEM.
- Represents the element matrix and vector that are frequently used when programming finite element methods.
- Estimation of the difference between two fields (exact and approximated)
- Offers the standard finite element algorithms for the assembly loop, including numerical integration over elements.
- All information on a finite element and Extended finite element for a programmer.
New classes for XFEM toolbox

- To generate level set in the mesh
- Provides a common interface for the enrichment of the mesh
- Stores the enrichment functions and the nodes enriched by those
- Implements the signed distance level-set description for curves or surfaces
- Calculate Delaunay triangulation for elements cut by interface
- Base class for the XFEM enrichment functions for Heaviside, Kink and Ramp function for blending
- Base class generates integration rules for different functions

Innovative Numerical Technologies
Solving a XFEM program in Diffpack step by step

• **STEP 1:** Meshing the geometry by diffpack preprocessor and load the geometry by GridFE

• **STEP 2:** Create a levelSet object using derived classes from GeoInterface (Polyline, Circle) and the virtual method calculateLevelSet.

• **STEP 3:** Selection of the nodes to be enriched and the enrichment functions
  - User must implement virtual method enrich of the class Enricher.
  - Specify how each function will evaluate (shifted, unshifted or SXFEM).
  - Output of the enrichment is stored in an object of the class Enrichment.
  - An object of the class DegFreeXFE is created linking nodes and dofs with unknown vector.

• **STEP 4:** Application of Dirichlet boundary conditions. This is done by DegFreeXFE.

• **STEP 5:** Calculation of the element matrix and vector by the method integrands using XElmMatVec class.

• **STEP 6:** Assembly of the system is done by the object of class LineqAdmXFE.

• **STEP 7:** Solve the system.

• **STEP 8:** Load the solution on a FieldXFE object through DegFreeXFE and Enrichment.
Design of XFEM module
Numerical example of XFEM toolbox in Diffpack
Bi-Material Bar

Bar Length of L with two different material $E_1$, $E_2$ and $x_b = L/2$

Exact displacement:

$$\forall x \in [0, b] : \quad u(x) = \frac{Fx}{AE_1}$$

$$\forall x \in ]b, L] : \quad u(x) = \frac{Fb}{AE_1} + \frac{F(x - b)}{AE_2}$$

<table>
<thead>
<tr>
<th>Norm</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1-norm</td>
<td>2.19803e-14</td>
</tr>
<tr>
<td>L2-norm</td>
<td>5.6033e-15</td>
</tr>
<tr>
<td>L∞-norm</td>
<td>1.738e-15</td>
</tr>
</tbody>
</table>
Crack using heaviside enrichment function

\[ E = 10000 \]
\[ v = 0.3 \]

Order of convergency = 1

Order of convergency = 0.5
Crack using Heaviside enrichment function

Ux
Bilinear element
160X160

Uy
Bilinear element
160X160

innuTech
Cardiff University
Innovative Numerical Technologies
Circular hole

E = 1e4, \( v = 0.3 \)

\( t = 1 \), plane strain

Order of convergence = 2

Order of convergence = 1
Conclusion
Conclusion

• A flexible, evolutive, versatile and easy to use object oriented XFEM toolbox in Diffpack platform is presented.

• Object oriented code allows easy code expansion for XFEM module with minor modification of Diffpack existing FEM module.

• We demonstrate useful treatment of mesh-geometry interaction, non-standard integration rules, application of boundary conditions, treatment of level set data by OOP.

• Different kind of enrichment functions operation and integration cut by interface using Delaunay triangulation focused in detail in the companion presentation by Cardiff University.
Acknowledgement
Acknowledgement

- Special thanks to Daniel Alves Paladim from Cardiff University who did the major part of this XFEM toolbox in Diffpack.

- A very special thanks to Prof. Stephane Bordas from Cardiff University and Frank Vogel from inuTech GmbH for their valuable and insightful guidance.

- Financial support of the Framework Programme 7 Initial Training Network Funding under grant number 289361 “Integrating Numerical Simulation and Geometric Design Technology”.
Thanks for your attention