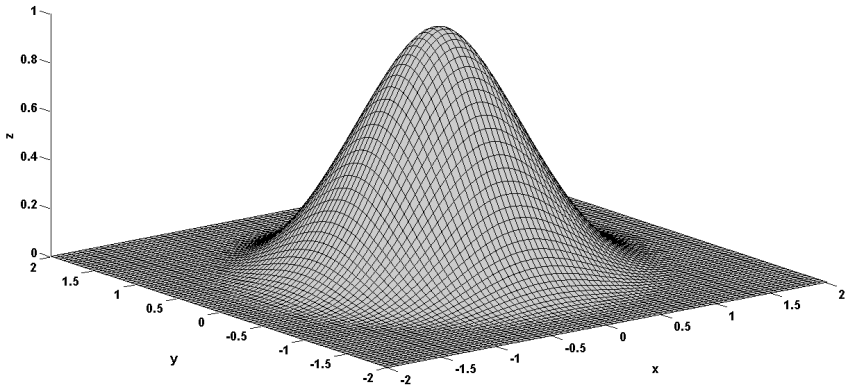
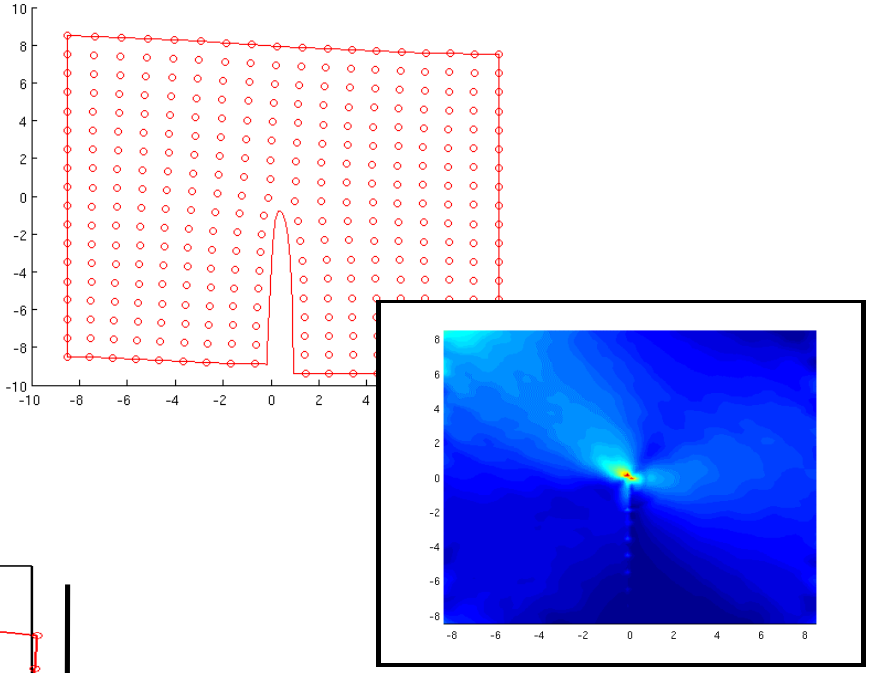
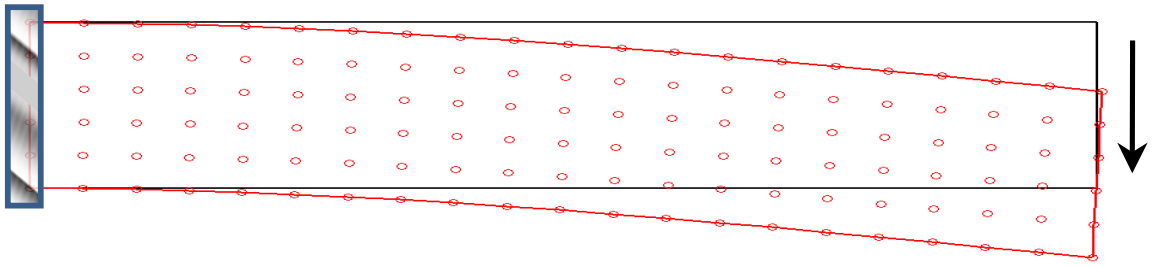
Project Presentation
« *Element-Free Galerkin Method* »

Alternative numerical methods in continuum mechanics by L. Noels







Written by:

Majérus Julien

Verheylewegen Guillaume

Outline

- Introduction
- Meshless
- Weak form equations
- Integration Scheme
- Examples
- Meshless in fracture mechanics
- Examples
- Summary
- Bibliography

Introduction

In the last decades, new computational methods were developed in order to solve effectively problems with discontinuities, moving boundaries, large deformations. One of the approaches was:

Element-Free Method
Meshless Method
Mesh-Free Method

No mesh support

Introduction - History

1977 - Smooth particle hydrodynamics (SPH) (Lucy)
1992 - Diffuse element method (DEM) (Nayroles et al.)
1994 - Element free Galerkin method (EFG) (Belytschko et al.; Organ et al.)
1995 - Reproducing kernel particle method (RKPM) (Liu et al.)
1996 - Hp-clouds method (Duarte and Oden, 1996)
1997 - Partition of unity method (PUM) (Babuska and Melenk)
1998 - Local boundary integral equation method (LBIE) (Zhu, Zhang, and Atluri)
1998 - Meshless local Petrov±Galerkin method (MLPG) (Atluri and Zhu).
2004 - Meshless Natural Element method (NEM) (Yvonnet).

…

Introduction - Utility of the meshless method ?

The famous discretized method, Finite Element Method (FEM), which has been widely used in many engineering problem simulations, exhibits some limitations:

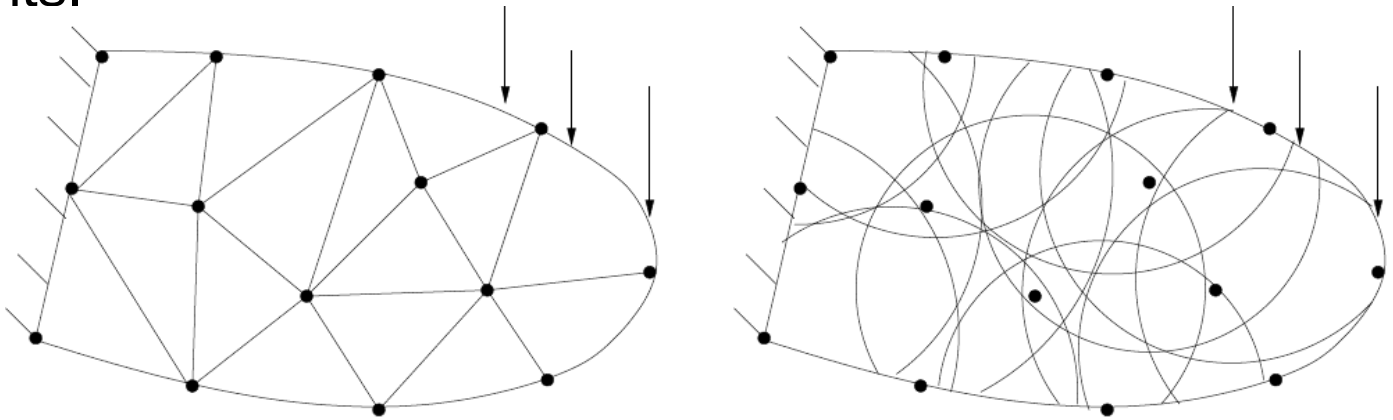
- Interpolation fails when elements become too distorted
- FE re-meshing is time-consuming, but necessary in some cases :
  - New surface creation : fracture mechanics
  - Large element distortion : finite strains
- Shear bands orientation are mesh dependent

Meshless method - Formulation

Meshless and FE methods are both based on a discretized displacement field:

$$u(x,y) = \sum_{i=1}^{N} N_i(x,y) u_i$$

Meshless consists in subdividing the structure in nodes. At each node, we associate a weight function and a shape function defined on a small domain. FEM uses shape functions which are defined on what we call elements.

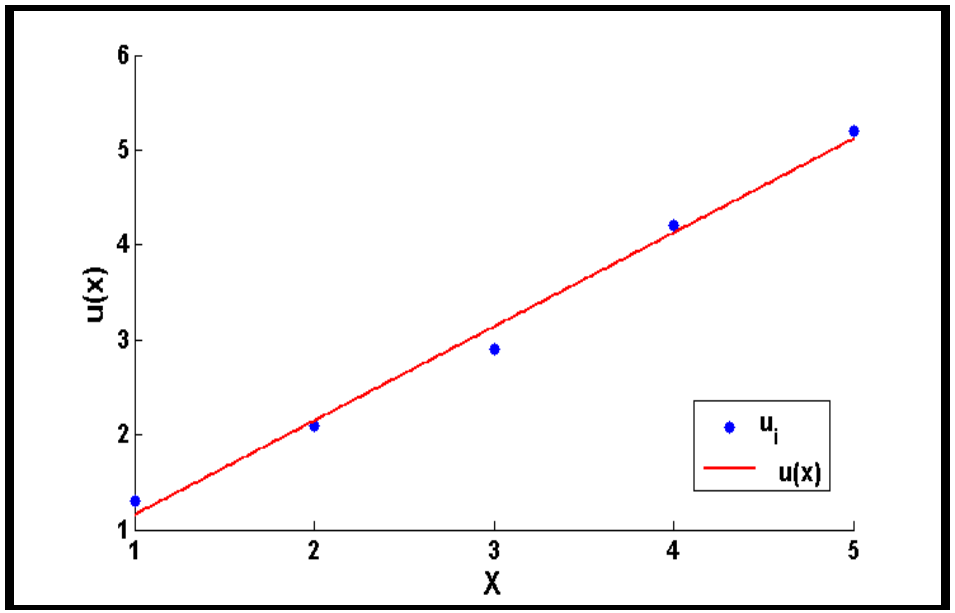


FE method - Meshless method

(Duflot PhD 2005)

Approximation definition (MLS)

Moving Least Squares (MLS)



MLS approximation

- Field approximation:

$$u(x) = \boldsymbol{p}^T(x)\boldsymbol{a}(\boldsymbol{x})$$

Where $\boldsymbol{p}^T(\boldsymbol{x})$ is the approximation base and $\mathbf{a}(\mathbf{x})$ the coefficients.

- The MLS consists in minimizing the weighted norm:

$$S - \sum_{i-1}^{N} w_i[\boldsymbol{p}^T(x_i)\boldsymbol{a}(\boldsymbol{x}) \quad u_i]^2$$

$$\frac{\partial S}{\partial \mathbf{a}} = 0$$

- This last equation can be written as: $\boldsymbol{A(x)a(x)=b(x)}$ where

$$\boldsymbol{A}(\boldsymbol{x}) - \sum_{i-1}^{N} w_i(\boldsymbol{x})\boldsymbol{p}(x_i)\boldsymbol{p}^T(x_i)$$

$$\boldsymbol{b}(\boldsymbol{x}) - \sum_{i-1}^{N} w_i(\boldsymbol{x})\boldsymbol{p}(x_i)u_i$$

Approximation definition (MLS) (2)

Based on the MLS approximation. Field ***u(x)*** which is defined on all domain, can be written as:

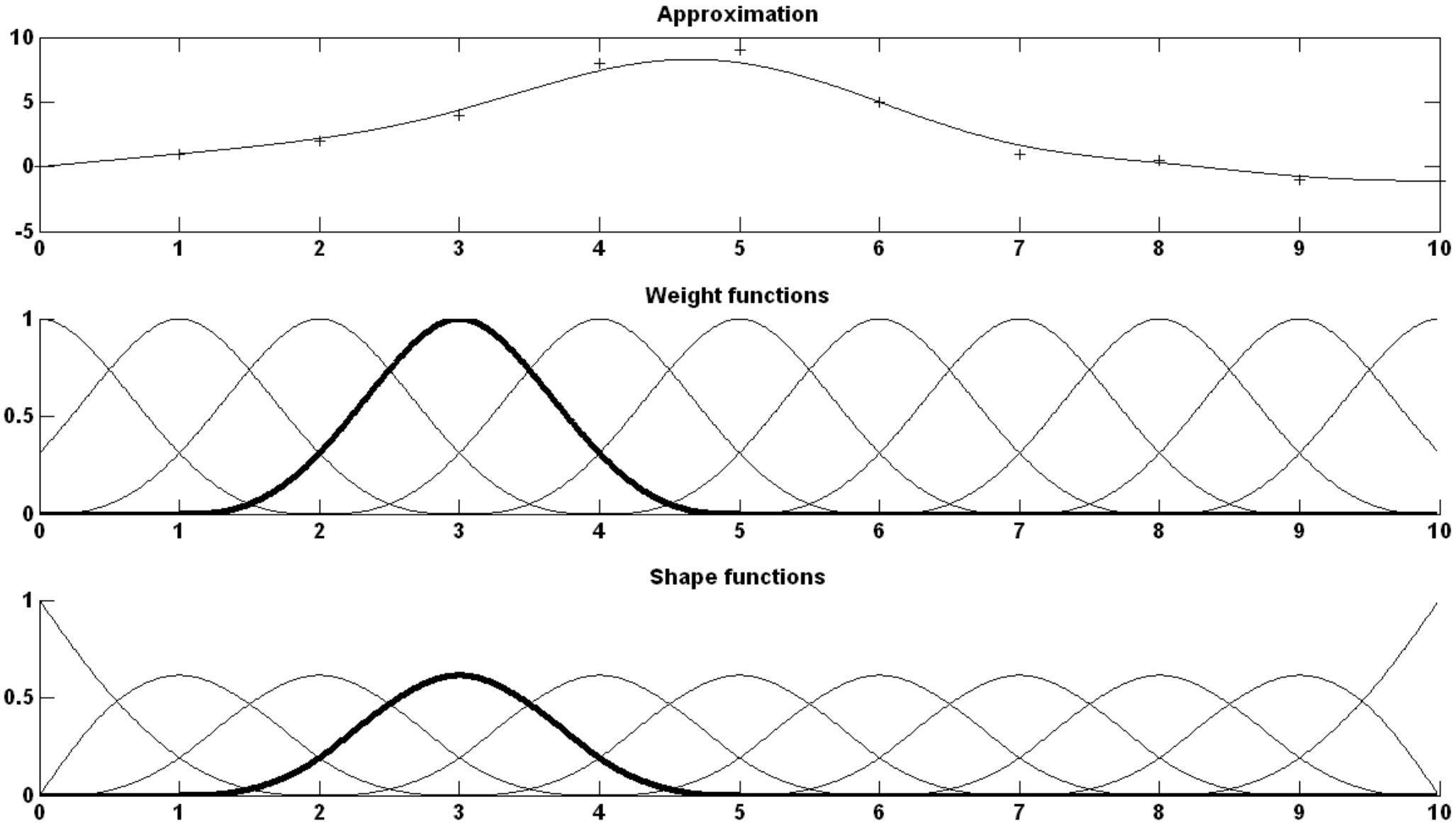
$$u(x) - \sum_{i-1}^{N} \phi_i(x) u_i$$

Where $\Phi_i(\boldsymbol{x})$ is denoted as the shape function associated to the node i

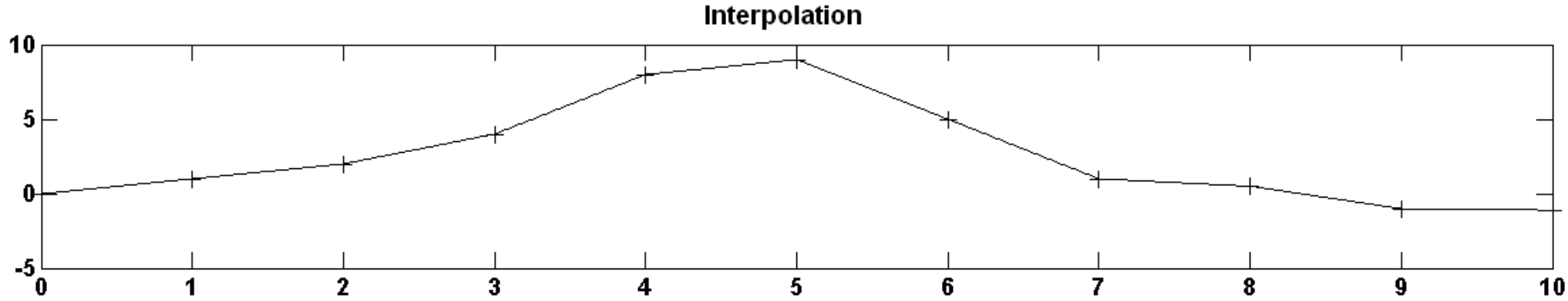
$$\phi_i(x) = c^T(x) p(x_i) w_i(x)$$

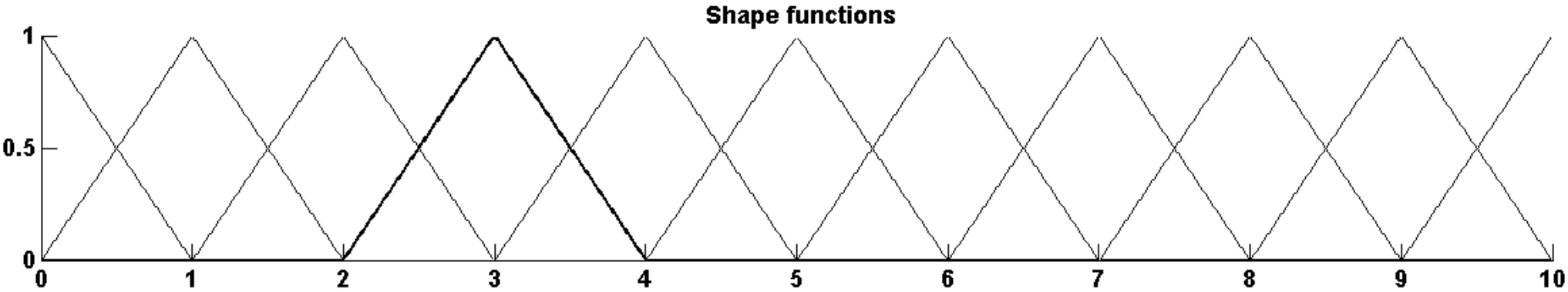
$$c(x) = A^{-1}(x) p(x)$$

Approximation definition (MLS) (3)



Interpolation definition (FEM)





Shape functions definition

The shape functions $\Phi_i(\boldsymbol{x})$ are independent of any mesh by definition. Shape function depends on:

- The approximation base
- A weight function $\boldsymbol{p}(\boldsymbol{x})$
- The domain of influence $w_i(\boldsymbol{x})$

Note that these parameters can be chosen arbitrarily and even differ between different nodes in the domain.

Shape functions properties $\Phi_i(x)$

- Zero-value outside the domain of the weight function.

- Non-Zero value on the whole domain of influence, if ***A(x)*** is not singular for any point. (There is a minimum size for the domain of the weight function)

- $\phi_i(\boldsymbol{x}_j) \neq \delta_{i,j}$ and thus $u(\boldsymbol{x}_i) \neq u_i$ .The shape functions don't interpolate the displacement at nodes. This is a major difference with FEM where one gets interpolation at nodes. Note that it exists different kinds of approximations, or in some cases interpolation as the C-NEM method which uses "natural element" based on Voronoi diagram)

- Partition of unity: $\sum_{i=1}^{n} \phi_i(\boldsymbol{x}) = 1 \quad \forall \boldsymbol{x} \in \Omega$ Rigid body modes can be computed.

- Consistency : If the exact solution belongs to the base of the shape functions, then the solution is exactly fitted by the MLS approximation.

- Continuity: $\phi_i(\boldsymbol{x}_j) \in C^{min(l,q)}$ where $p(x) \in C^l$ and $w_i(x) \in C^q$

Base functions definition $p(x)$

	1D	2D	3D
Constant	$[1]$	$[1]$	$[1]$
Linear	$[1,x]$	$[1,x,y]$	$[1,x,y,z]$
Quadratique	$[1,x,x^2]$	$[1,x,y,x^2,y^2,xy]$	$[1,x,y,z,x^2,y^2,z^2,xy,xz,yz]$

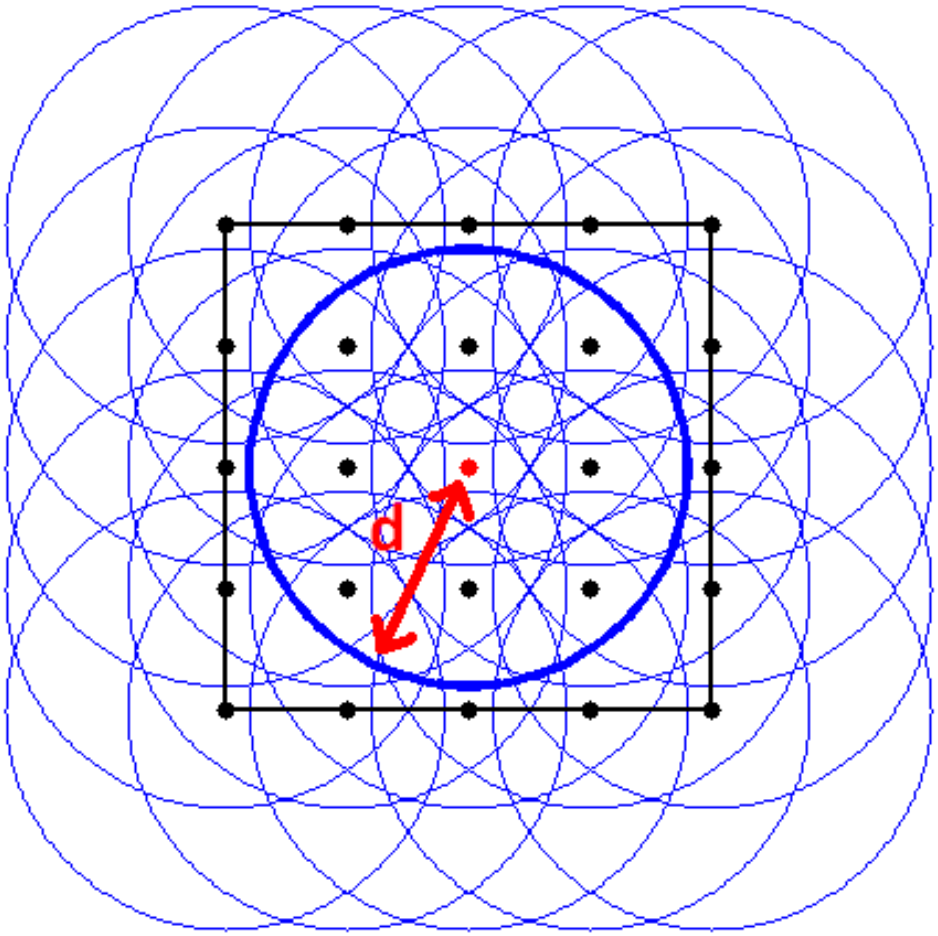
Since base function are $C^{\infty}(\Omega)$, shape functions have the same degree of continuity as the weight function.

The choice of the base function will define the size of the matrix ***A***. Since this one has to be inverted at each node, it is time-consuming. In addition to this, the node domain has to be bigger with a higher order of the base if we want to avoid ***A*** to be singular.

$$\phi_i(\boldsymbol{x}) = \boldsymbol{c}^T(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_i)w_i(\boldsymbol{x})$$
$$\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x})$$

Weight function definition $w_i(x)$

The weight function is a function which decrease with the distance from node. Outside the node domain it must be equal to zero.



- Circular domain (2D)

$$w_i(x) \quad f\left(\frac{x - x_i}{d_i}\right)$$

- Rectangular domain (2D)

$$w_i(x) = f\left(\frac{x - x_i}{d_i^x}\right) f\left(\frac{y - y_i}{d_i^y}\right)$$

Choice of Weight function (2)

If we define $s = \left| \frac{x_i - x}{d_i} \right|$, in the literature we find functions:

- Troncated Gaussian

$$f(s) = \begin{cases} e^{-\left(\frac{s}{\alpha}\right)^2} & \text{if } |s| \leq 1 \\ 0 & \text{if } |s| > 1 \end{cases}$$

Not $C^0(\Omega)$!

- Modified Gaussian

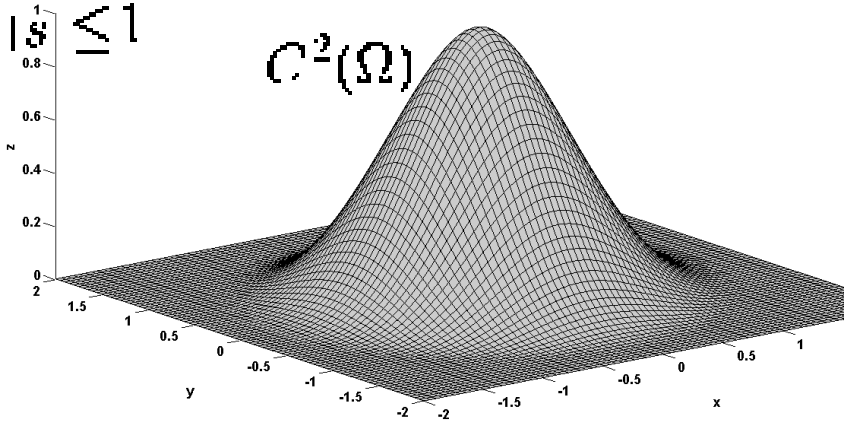
$$f_2(s) = \begin{cases} \frac{e^{-\left(\frac{s}{\alpha}\right)^2} - e^{-\left(\frac{1}{\alpha}\right)^2}}{1 - e^{-\left(\frac{1}{\alpha}\right)^2}} & \text{si } |s| \leq 1 \\ 0 & \text{si } |s| > 1 \end{cases}$$

$C^0(\Omega)$

- Spline of order 4

$$f(s) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4 & \text{if } |s| \leq 1 \\ 0 & \text{if } |s| > 1 \end{cases}$$

$C^2(\Omega)$



Weak form of elasticity equations (FEM)

If we consider an arbitrary test function $\delta \boldsymbol{u}$, we can write the weak form of the static equilibrium equations:

$$-\int_{\Omega} \delta \boldsymbol{u} . (\nabla^T \boldsymbol{\sigma} + \bar{\boldsymbol{b}}) d\Omega + \int_{\Gamma_t} \delta \boldsymbol{u} . (\boldsymbol{\sigma} \boldsymbol{n} - \bar{\boldsymbol{t}}) d\Gamma = 0$$

By part integration and Gauss theorem:

$$\int_{\Omega} \nabla_s \delta \boldsymbol{u}^T : \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \boldsymbol{u} . \bar{\boldsymbol{b}} d\Omega - \int_{\Gamma_t} \delta \boldsymbol{u} . \bar{\boldsymbol{t}} d\Gamma = 0$$

The Galerkin procedure assume that same function are used :

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{N} \phi_i(\boldsymbol{x}) q_i$$

$$\delta \boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{N} \phi_i(\boldsymbol{x}) d_i$$

Weak form of elasticity equations (FEM) (2)

With the elastic material behavior ***D***:

$$\sum_{i=1}^{N}\sum_{i=1}^{N}\int_{\Omega}(\nabla\phi_i\boldsymbol{d}_i)^T\boldsymbol{D}(\nabla\phi_j\boldsymbol{q}_j)d\Omega \quad \sum_{i=1}^{N}\int_{\Omega}\phi_i\boldsymbol{d}_i^T\bar{\boldsymbol{b}}d\Omega \quad \sum_{i=1}^{N}\int_{\Gamma_t}\phi_i\boldsymbol{d}_i^T\bar{\boldsymbol{t}}d\Gamma - 0$$

As $\delta\boldsymbol{u}$ is arbitrary, the last equation can be written on the form:

$$\boldsymbol{K}_{ij}\boldsymbol{q}_j = \boldsymbol{g}_i$$

$$\boldsymbol{K}_{ij} = \int_{\Omega}\boldsymbol{B}_i^T\boldsymbol{D}\boldsymbol{B}_j d\Omega$$

$$\boldsymbol{g}_i = \int_{\Omega}\phi_i\bar{\boldsymbol{b}}d\Omega + \int_{\Gamma_t}\phi_i\bar{\boldsymbol{t}}d\Gamma$$

Weak form of elasticity equation (Meshless)

FEM equation are still valid for meshless.

However, due to the not interpolating shape functions, the Essential Boundary Conditions (EBC) can not be enforced the same way. One can **not impose** a nodal displacement as in FEM.

As a consequence, modification have to be taken into account to be able to satisfy EBC. Three main methods (a,b,c) have been developed.

(a) Weak form with Lagrangian multiplicator

In this weak form, Lagrangian multiplicators are used to impose the displacement boundary conditions on Γ_u

$$\int_\Omega \nabla_s \delta u^T : \sigma d\Omega - \int_\Omega \delta u . \bar{b} d\Omega - \int_{\Gamma_t} \delta u . \bar{t} d\Gamma - \int_{\Gamma_u} \delta \lambda . (u - \bar{u}) d\Gamma - \int_{\Gamma_u} \delta u . \lambda d\Gamma = 0$$

With the Galerkin procedure and the arbitrary value of δ , we can write in matrix form

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} q \\ \lambda \end{Bmatrix} = \begin{Bmatrix} g \\ h \end{Bmatrix}$$

$$K_{ij} = \int_\Omega B_i^T D B_j d\Omega \qquad G_{ik} = -\int_{\Gamma_u} \phi_i \psi_k S d\Gamma$$

$$g_i = \int_\Omega \phi_i \bar{b} d\Omega + \int_{\Gamma_t} \phi_i \bar{t} d\Gamma \qquad h_k = -\int_{\Gamma_u} \psi_k S \bar{u} d\Gamma$$

ψ_k is an interpolating shape function with the curvilinear variable s along Γ_u

$\boldsymbol{S}$ is a diagonal matrix whose element ii is equal to 1 if u_i is imposed and 0 otherwise. (useful if, e.g., u_x is imposed but u_y is free)

(b) Weak form without Langrangian multiplicator

This method is similar to the previous one. Lagrange multiplier have a physical meaning: they represent reaction forces at the boundary. These reaction forces ***t*** can be directly identified in the weak form:

$$\int_{\Omega} \nabla_s \delta \boldsymbol{u}^T : \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \boldsymbol{u} . \bar{\boldsymbol{b}} d\Omega - \int_{\Gamma_t} \delta \boldsymbol{u} . \bar{\boldsymbol{t}} d\Gamma - \int_{\Gamma_u} \delta \boldsymbol{t} . (\boldsymbol{u} - \bar{\boldsymbol{u}}) d\Gamma - \int_{\Gamma_u} \delta \boldsymbol{u} . \boldsymbol{t} d\Gamma = 0$$

Where $\boldsymbol{t} = (\boldsymbol{D} : \nabla_s \boldsymbol{u}) . \boldsymbol{n}$
$\delta \boldsymbol{t} = (\boldsymbol{D} : \nabla_s \delta \boldsymbol{u}) . \boldsymbol{n}$

Thus the system is simply: ***K q*** = ***g***

$$\boldsymbol{K}_{ij} = \int_{\Omega} \boldsymbol{B}_i^T \boldsymbol{D} \boldsymbol{B}_i d\Omega - \int_{\Gamma_u} (\phi_i \boldsymbol{S} \boldsymbol{N} \boldsymbol{D} \boldsymbol{B}_j + \boldsymbol{B}_i^T \boldsymbol{D}^T \boldsymbol{N}^T \boldsymbol{S} \phi_j) d\Gamma$$

$$\boldsymbol{g}_i = \int_{\Omega} \phi_i \bar{\boldsymbol{b}} d\Omega + \int_{\Gamma_t} \phi_i \bar{\boldsymbol{t}} d\Gamma - \int_{\Gamma_u} \boldsymbol{B}_i^T \boldsymbol{D}^T \boldsymbol{N}^T \boldsymbol{S} \bar{\boldsymbol{u}} d\Gamma$$

The system of equations is smaller than in the previous method

(c) Weak form with penalty factor

Here we use a penalty factor in the equation:

$$\int_{\Omega} \nabla_s \delta \boldsymbol{u}^T : \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \boldsymbol{u} . \bar{\boldsymbol{b}} d\Omega - \int_{\Gamma_t} \delta \boldsymbol{u} . \bar{\boldsymbol{t}} d\Gamma - \frac{\beta}{2} \delta \left(\int_{\Gamma_u} \| \boldsymbol{u} - \bar{\boldsymbol{u}} \|^2 d\Gamma \right) = 0$$

$$\boldsymbol{K} \boldsymbol{q} = \boldsymbol{g}$$

$$\boldsymbol{K}_{ij} = \int_{\Omega} \boldsymbol{B}_i^T \boldsymbol{D} \boldsymbol{B}_i d\Omega - \beta \int_{\Gamma_u} \phi_i \phi_j d\Gamma$$

$$\boldsymbol{g}_i = \int_{\Omega} \phi_i \bar{\boldsymbol{b}} d\Omega + \int_{\Gamma_t} \phi_i \bar{\boldsymbol{t}} d\Gamma - \beta \int_{\Gamma_u} \phi_i \bar{\boldsymbol{u}} d\Gamma$$

The difficulty of this method is too chose the right β. Too high the system is badly conditioned, and too low the imposed displacements are not accurate.

Integration Scheme

Integrals over the domain Ω and Γ have to be evaluated. In contrast with the FEM, we have not any "element" which are a natural solution to integrate over the domain. Three main solutions can be used.

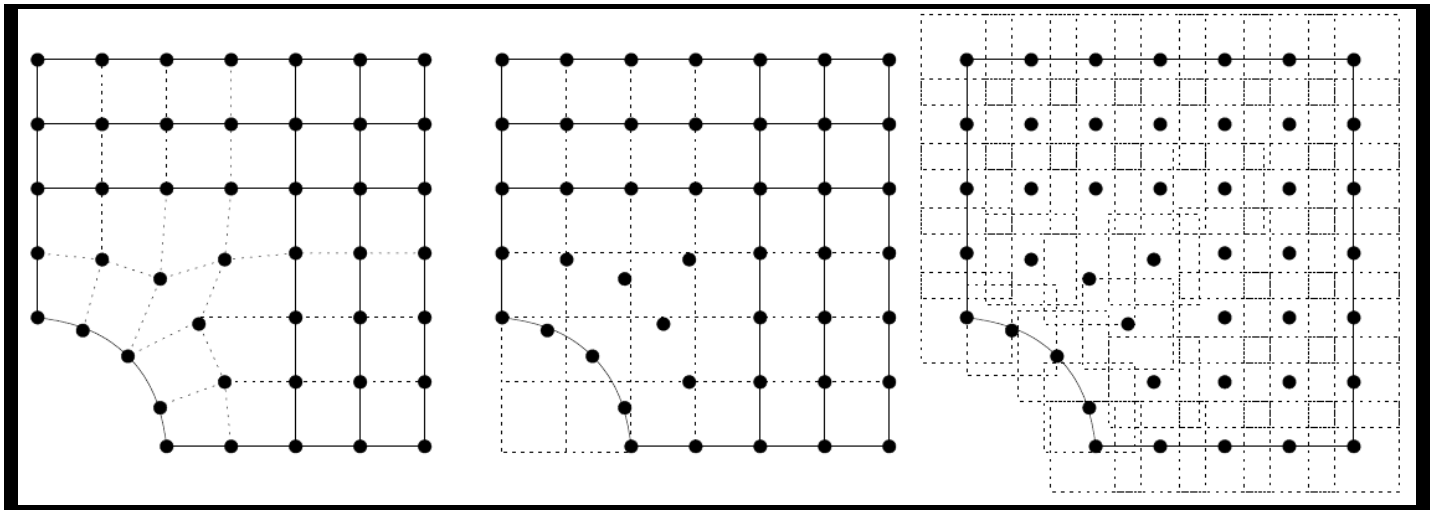


Fig: Integration scheme, (a) "meshnode" , (b) "background mesh", (c) "node domain"
(Duflot PhD 2005)

(a) Use nodes distribution to form "elements" where a Gauss integration will be performed on each element.

(b) Use an independent structured mesh over the domain where Gauss integration will be performed.

(c) Integrate over nodes "integration domain"

Process

STIFFNESS: $K_{ij} = \underbrace{\int_{\Omega} B_i^T D B_i d\Omega}_{1} - \underbrace{\int_{\Gamma_u} (\phi_i S N D B_j + B_i^T D^T N^T S \phi_j) d\Gamma}_{2}$

LOOP1 (At each integration point x_g over Ω)
  LOOP2 (at each node i)
    $w_i(x_g)$
    $A_g - \sum_{i-1}^{N} w_i(x_g) p(x_i) p^T(x_i)$
  END LOOP2
  LOOP3 (at each node i)
    $\phi_i(x_g) = (A_g^{-1} p(x_g))^T p(x_i) w_i(x_g)$
    $\phi_i^I(x_g)$
  END LOOP3
  ADD to K
END LOOP1

1

LOOP4 (At each integration point x_g over Γ_u)
  ...
  ADD to K
END LOOP4

2

EXTERNAL FORCES $g_i = \int_{\Omega} \phi_i \bar{b} d\Omega + \int_{\Gamma_t} \phi_i \bar{t} d\Gamma - \int_{\Gamma_u} B_i^T D^T N^T S \bar{u} d\Gamma$

SOLVE $Kq = g$

GET DISPLACEMENT FIELD: $u(x) - \sum_{i-1}^{N} \phi_i(x) q_i$

Algorithm optimisation

Université de Liège

The time consumed in the algorithm come from:

- The number of integration points in each cell

- The inversion of the matrix A. For a linear choice of base functions the size is only 3x3, but it has to be inverted at each integration point.

- The loop over each node (LOOP2,LOOP3). Since the weight function is non zero only in a small domain around the related node, at each integration point only few node around this point will get a weight function different from zero. By consequence it's insane to loop over all nodes. One way to improve this is to take into account the relation between nodes by a quadtree formulation which will give an tree fast to scan.

Miscellaneous

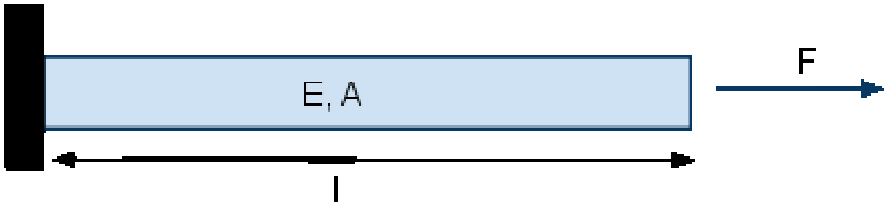
Truly meshless method:

- Non-element interpolation technique
- Non-element approach for integrating the weak form

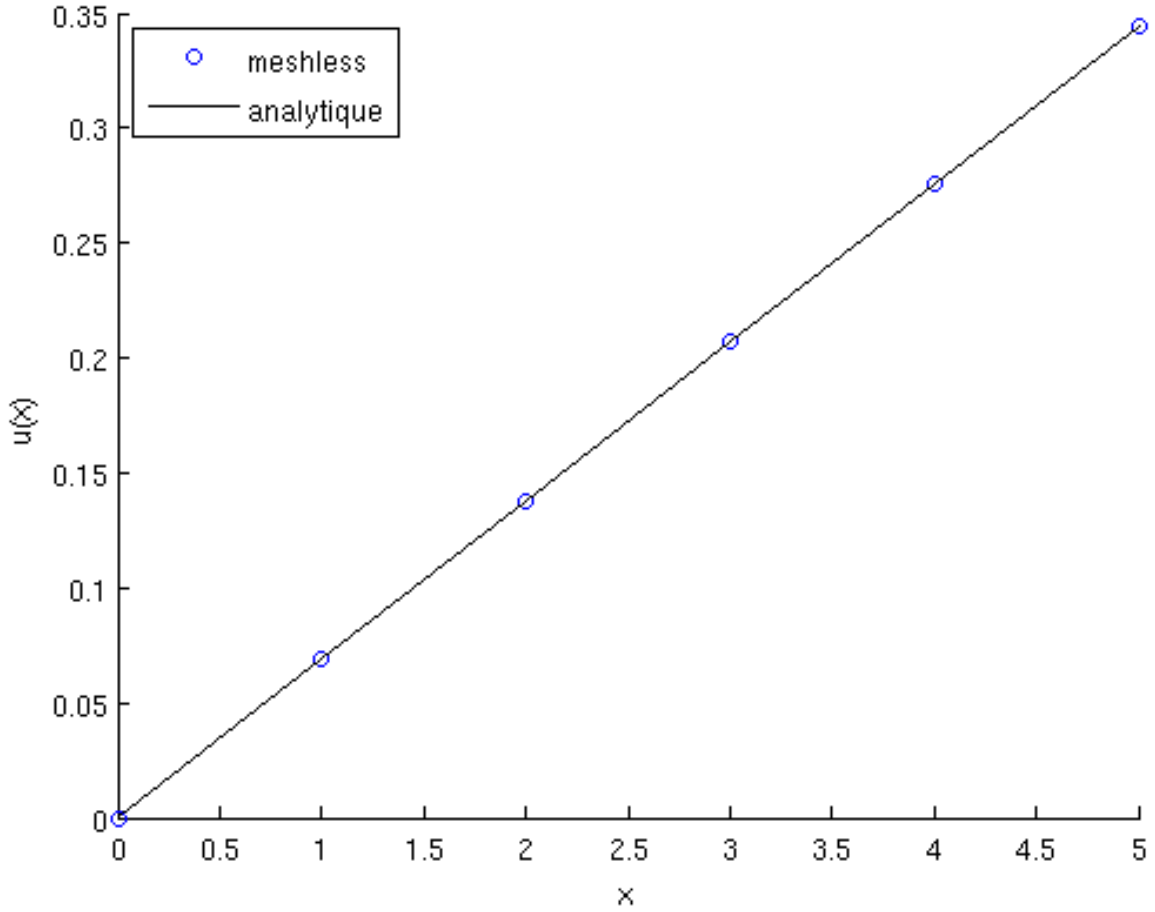
Example

- a truly meshless method = Meshless local Petrov-Galerkin method (MLPG), no need of mesh or "integration mesh »
- a meshless method = Element free Galerkin method (EFG), need of "integration mesh".

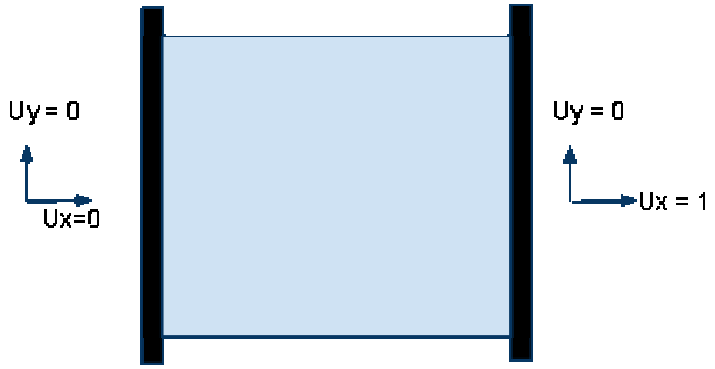
1D example : the bar

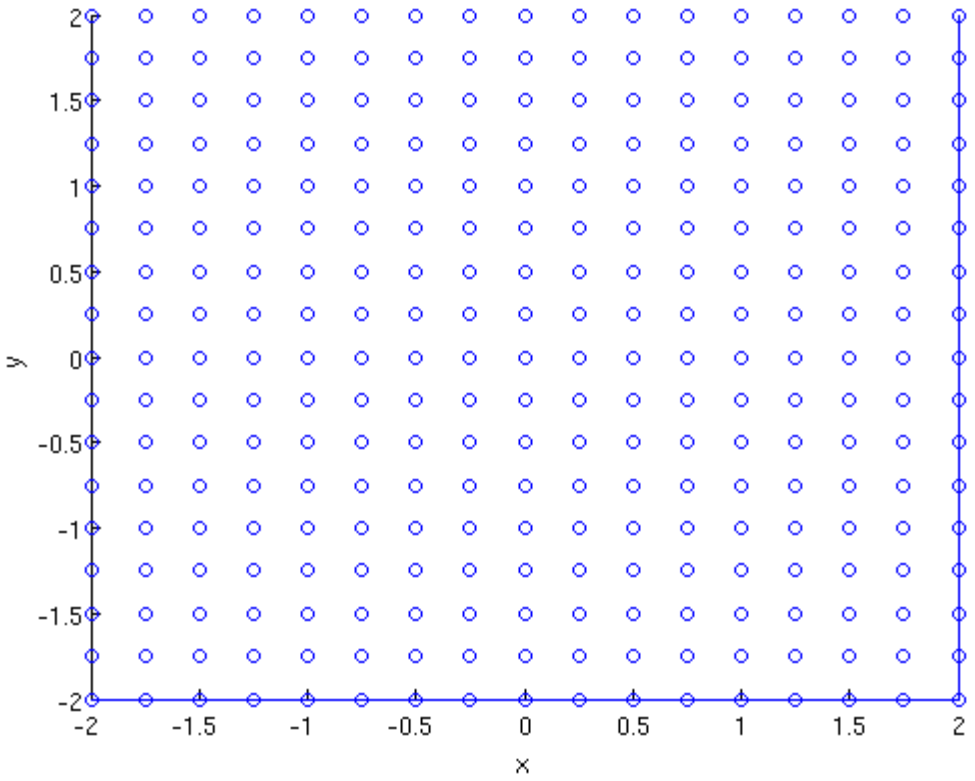


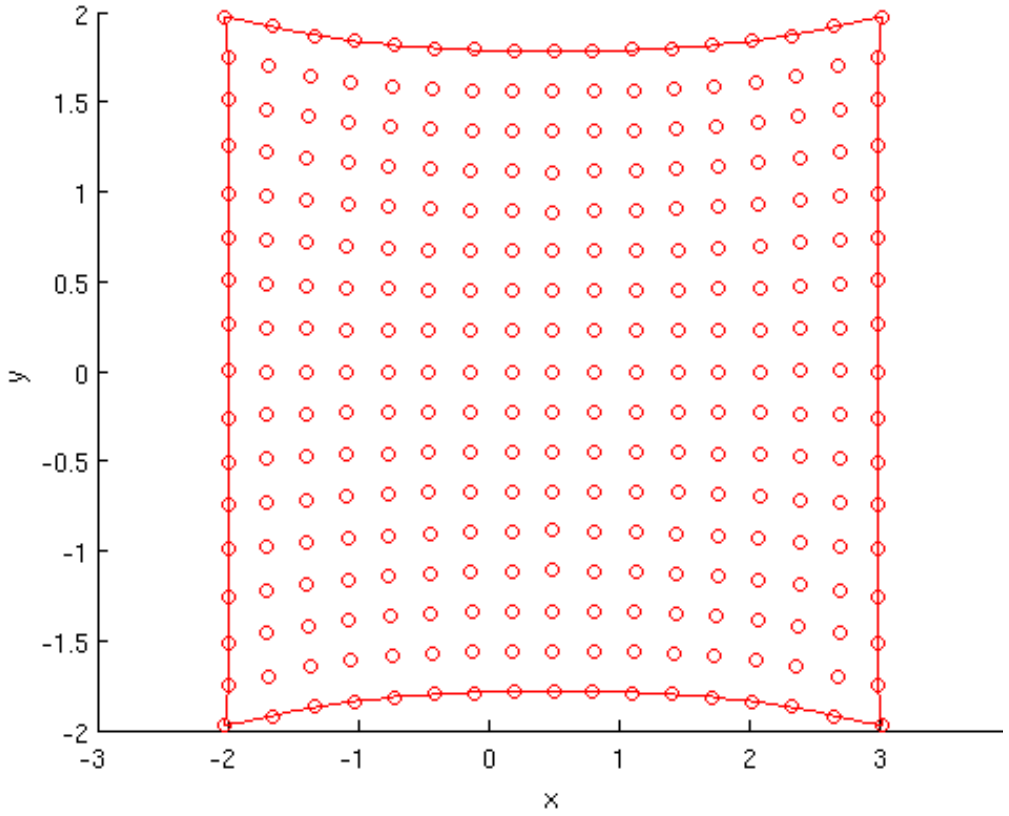
Analytical solution : $$u(x) = \frac{lF}{EA}x$$



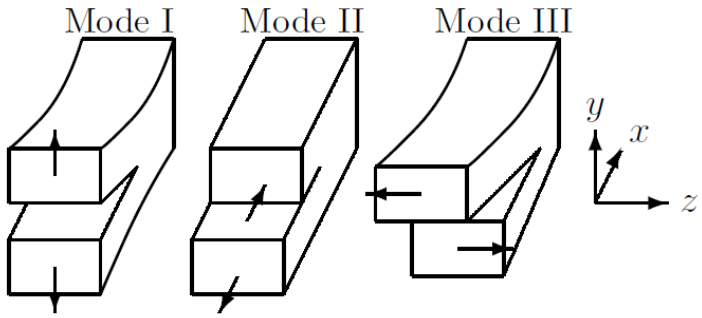
2D example : bi-clamped plane strain body







Fracture mechanics - reminder



A crack involves a singular displacement field close to the crack tip. Under the hypothesis of Linear Elastic Fracture Mechanics (LEFM), it can be shown that the displacement is asymptotically almost proportional to $\sqrt{r}$ where r is the distance from the crack tip. For example, for a crack opening in mode I only, the displacement is :

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{K_I}{2\mu}\sqrt{\frac{r}{2\pi}} \begin{bmatrix} \cos\frac{\theta}{2}(\kappa - 1 + 2\sin^2\frac{\theta}{2}) \\ \sin\frac{\theta}{2}(\kappa + 1 + 2\cos^2\frac{\theta}{2}) \end{bmatrix}$$

$$\kappa = \frac{3-\nu}{1+\nu} \quad \text{(plane stress)}$$

$$\kappa = 3 - 4\nu \quad \text{(plane strain)}$$

Note that if one can accurately compute the stress intensity factor KI, the displacement field is (asymptotically) completely defined. Similar relationships can be shown for the two other modes.

How to insert a crack in the model

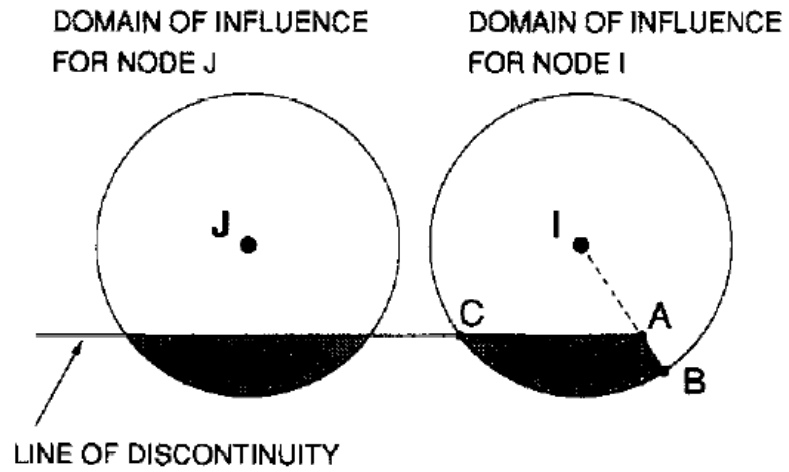
A crack introduces a discontinuity in the displacement field. The model should be able to reproduce this discontinuity. Two possibilities:

- The crack belongs to the boundary of the domain. (ex: in FEM the mesh has to be conform to the crack. In propagation, a new mesh is necessary at each step)

- The shape functions are discontinuous. As a consequence $u(x) \quad \sum_i \phi_i(x) u_i$ will be discontinuous too. (e.g.: X-FEM & meshless)

In fact, the discontinuity in the shape function is introduced through a discontinuity in the weight function. Remember that $\phi_i(x) = c^T(x) p(x_i) w_i(x)$

How to insert a crack in the model: a) visibility criterion



Belytschko et al. /Cornput. Methods Appl. Mech. Engrg. 139 (1996) 3-47

The crack acts as an opaque surface. If a point can't "see" a node, then the weight (related to the node) of this point is set to zero, i.e.:

$$w_i(x) = \begin{cases} w_i(x) & \text{if point } x \text{ can "see" the node i} \\ 0 & \text{otherwise} \end{cases}$$

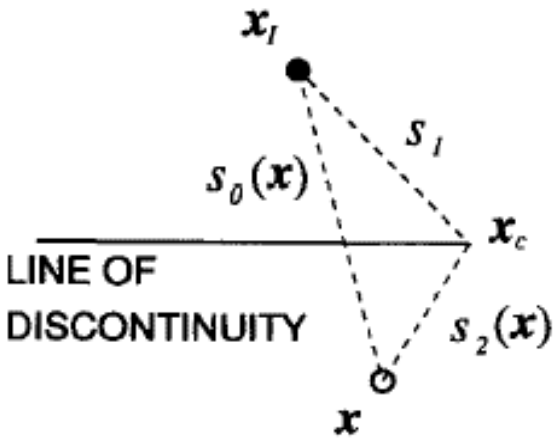
- Easy to compute
- But introduces a spurious discontinuity (along AB)

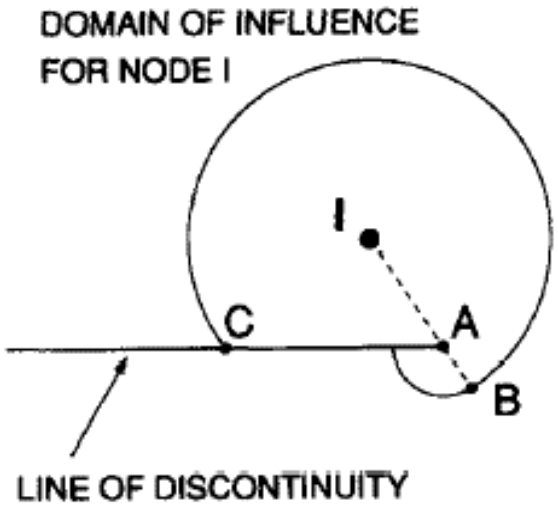
How to insert a crack in the model: b) Diffraction criterion

Idea : avoid the spurious discontinuity introduced by the visibility criterion

Solution : Use the "diffracted" distance d' in order to compute the weight w(d'). If the line between the point x and the node xi crosses the crack, the diffracted distance is :

$$d' = \max(S_0=d\ ,\ S_1+S_2)$$





Belytschko et al. /Cornput. Methods Appl. Mech. Engrg. 139 (1996) 3-47

How to deal with the singularity

The displacement field is singular close to the crack tip. The node density should be increased in this region. Several methods exist, but the main idea is to introduce the asymptotic solution in the model. Either we introduce new degrees of freedom (as X-FEM does), or we can simply enrich de base :

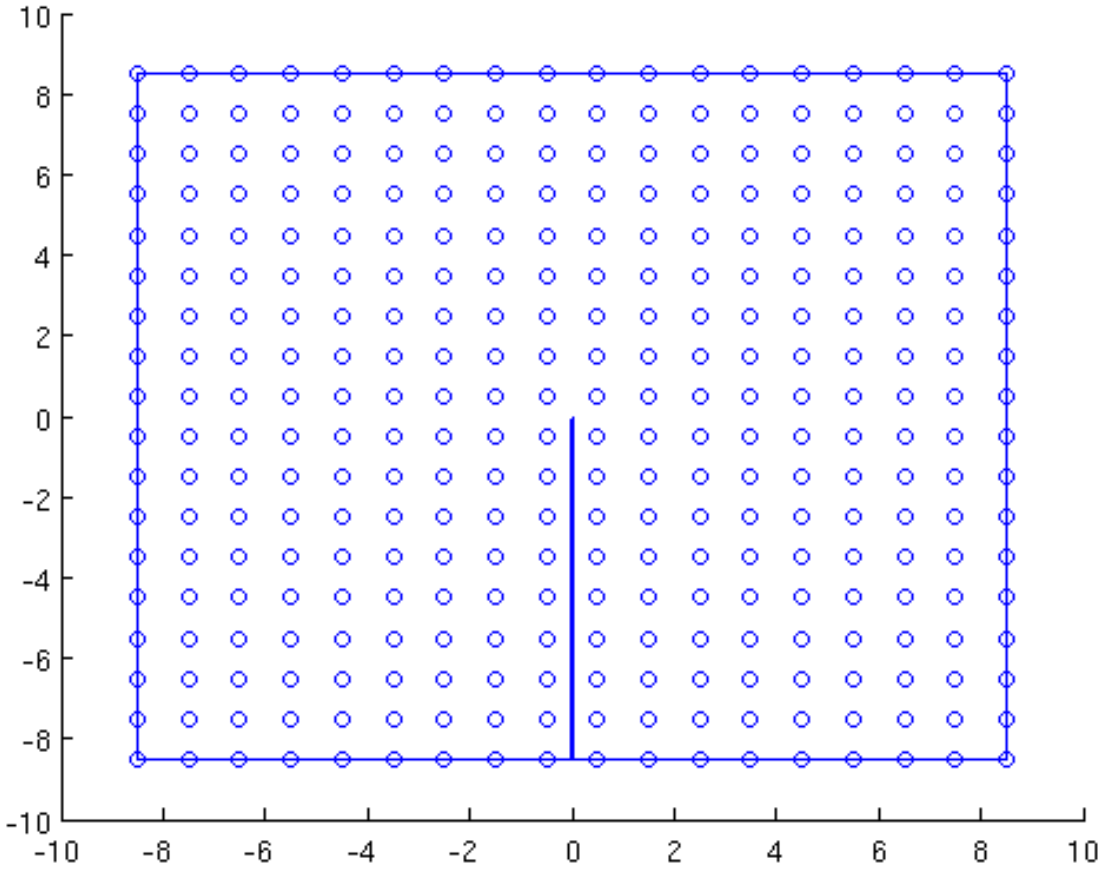
$$\boldsymbol{P}^T = \left[1, x, y, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right]$$

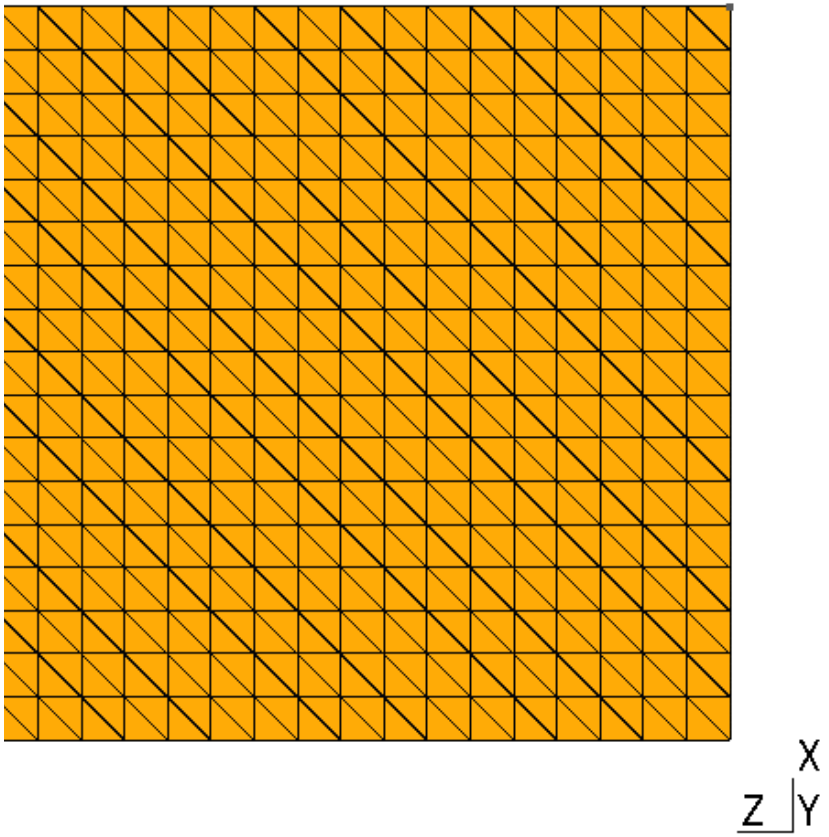
The matrix **A** that has to be inverted to compute the shape functions is now a 7 by 7 matrix. Moreover, as shape functions are now able to represent a singular field, a higher number of integration point is necessary.

Numerical example: 2D mixed-mode loading - Discretisation

Université de Liège

Because the formulation is quite similar to the X-FEM one, results will be compared to a problem solved with XFEM (the latter has been computed with the FEM/X-FEM software Morfeo).

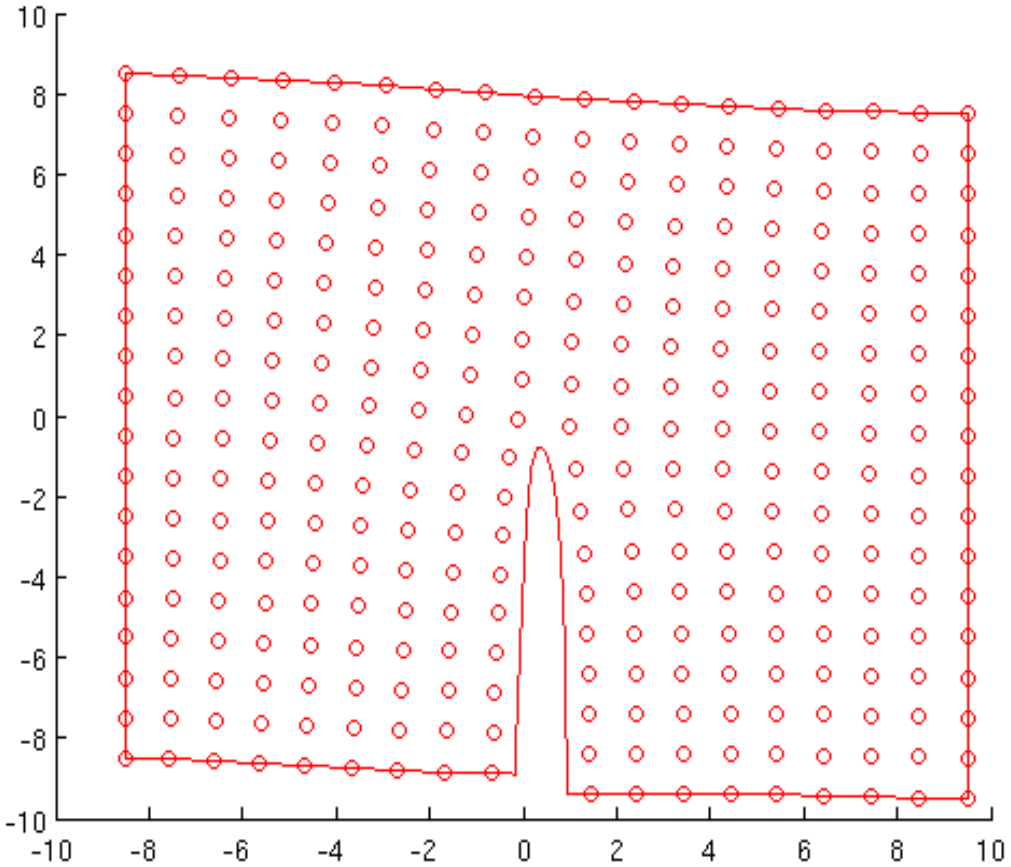


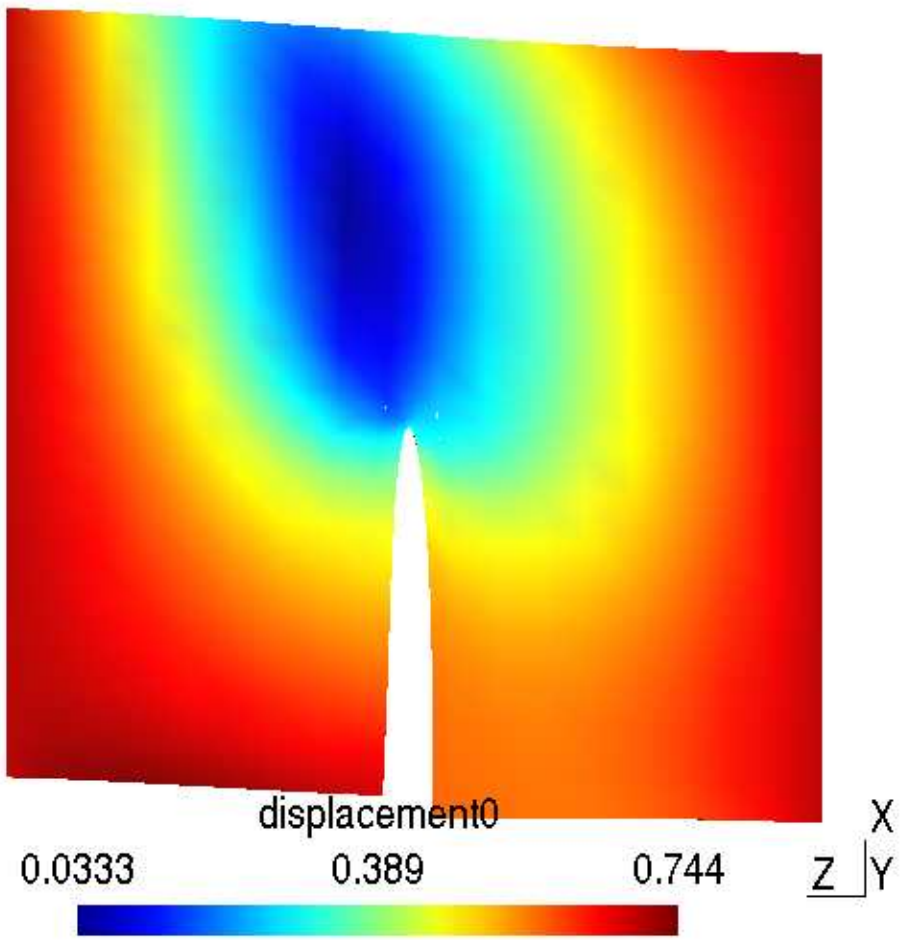


Meshless (Matlab) – *XFEM (Morfeo)*

Numerical example: 2D mixed-mode loading - Displacements

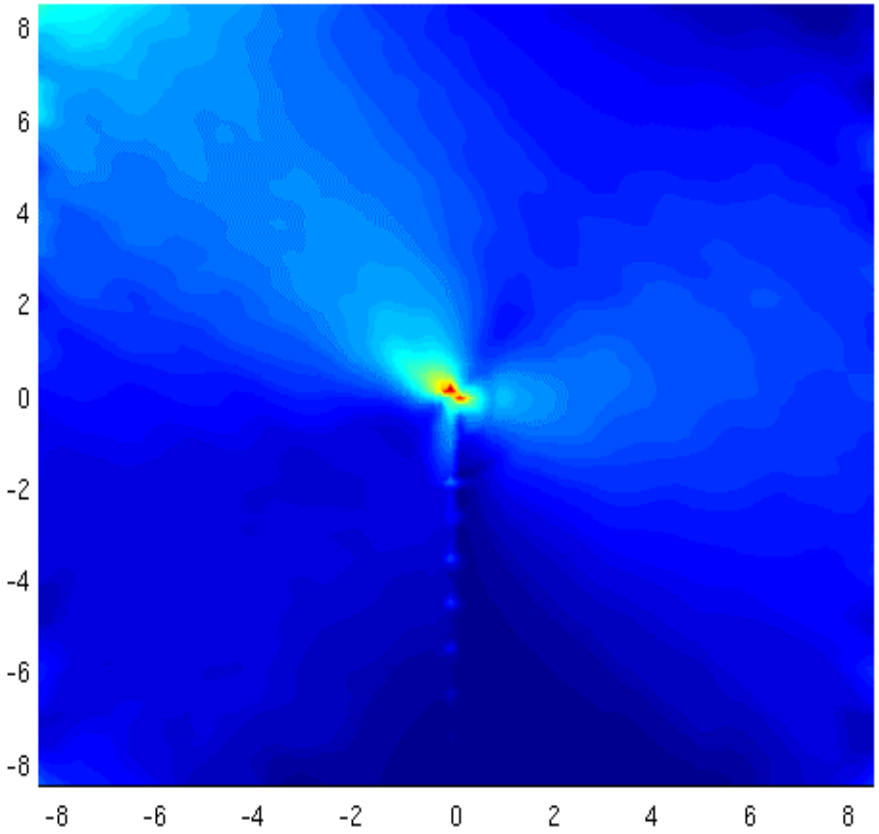
Imposed displacement (traction + shear)

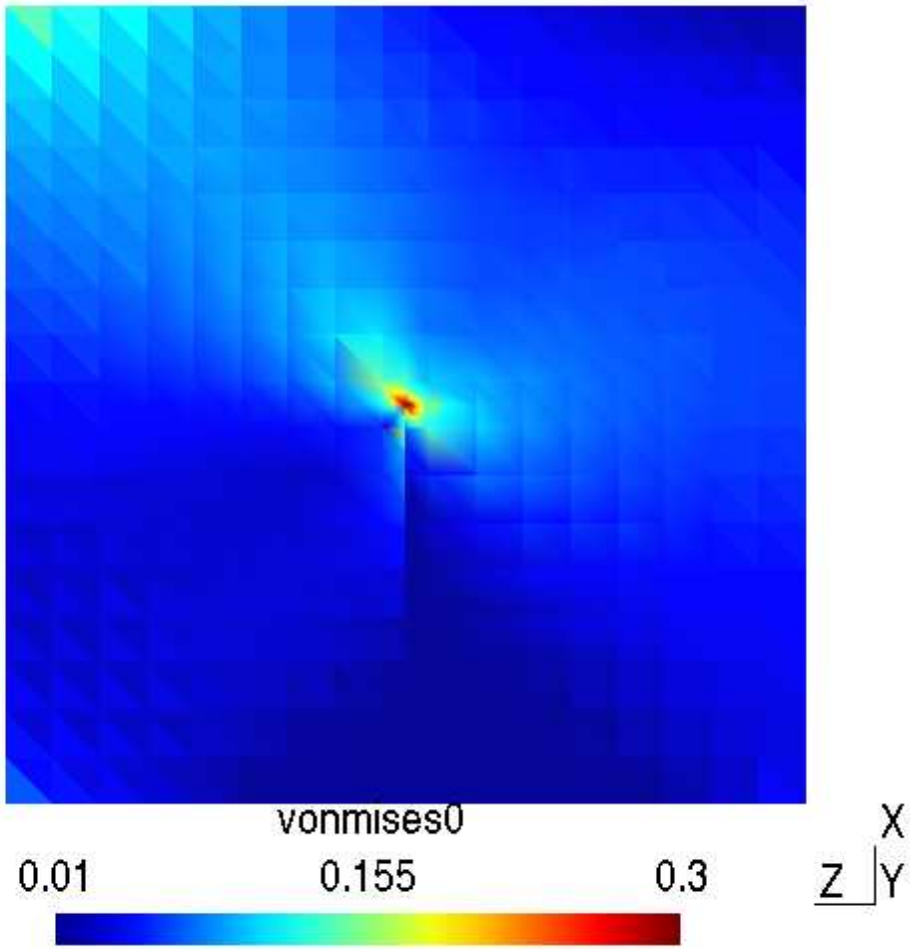




Meshless (Matlab) – *XFEM (Morfeo)*

Numerical example: 2D mixed-mode loading - Von Mises stresses





Meshless (Matlab) – *XFEM (Morfeo)*

Stress Intensity Factors Computation

Stress intensity factors are extremely important because they completely characterise the displacement and stress field near the crack tip. Moreover, they are necessary to predict the crack propagation direction and the crack propagation speed in fatigue. The most accurate method to compute them is the J integral. For a straight crack in the direction x:

$$J = \int_{\Gamma} \left(\frac{\sigma_{ij}\varepsilon_{ij}}{2}\delta_{1j} - \sigma_{ij}\frac{\partial u_i}{\partial x_1} \right) n_j d\Gamma$$

In practice, J is evaluated over a domain, not a contour (divergence theorem). The relationship between J and the SIF's is :

$$J = \frac{K_I^2}{E^*(E,\nu)} + \frac{K_{II}^2}{E^*(E,\nu)} + \frac{K_{III}^2}{2\mu}$$

Mode I only ($K_{II} = K_{III} = 0$): K_I directly determined with J
Mixed mode: More equations are necessary. It is possible to show that by superposing (linear elasticity framework!) auxiliary virtual solutions to the numerical solution (use of interaction integrals), one can extract the stress intensity factors. The demonstration of this is beyond the scope of this presentation. However, we have applied this method :

Stress Intensity Factors Computation

Meshless : K_I = **0.17262** K_{II} = **0.096375**

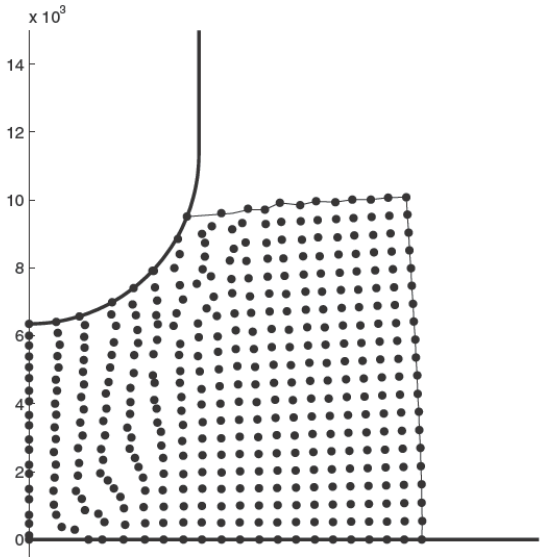
XFEM : K_I = **0.171621** K_{II} = **0.0969399**

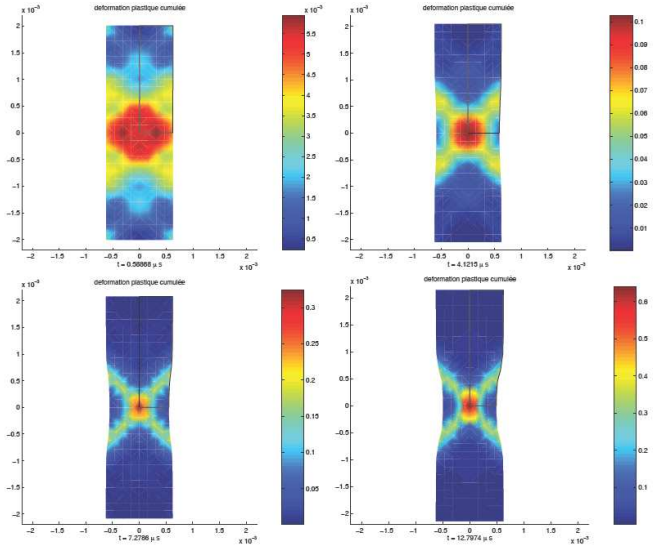
Our model is not so bad!

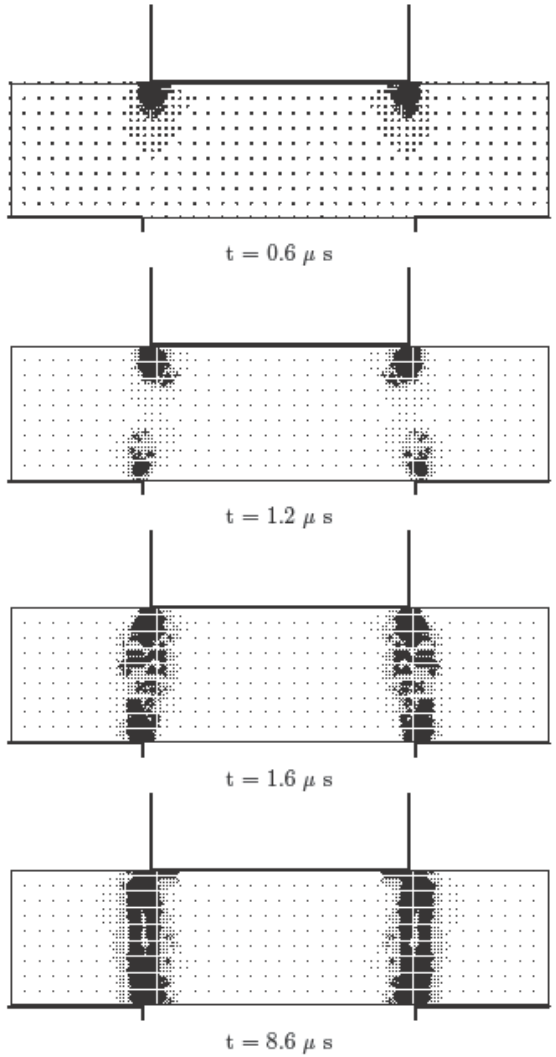
Summary

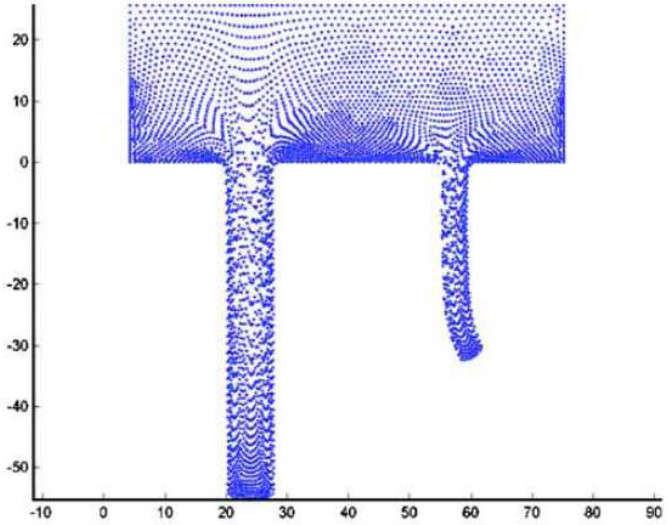
- Only a set of node is needed
- But a mesh is generally needed in order to compute a Gauss quadrature
- One can easily modify the base functions or the weight functions where we want in the domain
- Arbitrary discontinuities can be modelled without modifying the set of node
- Stress and strain computed with the meshless methods are smooth (in FEM they are discontinuous between elements)

Other applications









Bibliography

- M. Duflot, Application des méthodes sans maillage en mécanique de la rupture, Thèse de doctorat, Ulg, 2005
- T. Belytschko, Y.Y. Lu, L.Gu, Element-free Galerkin methods, International Journal for Numerical Methods in Engineering vol 37, 229-256, 1994
- T. Belytschko, L. Gu, Y.Y. Lu, Fracture and crack growth by element free Galerkin methods, Modelling and Simulation in Materials Science and Engineering vol 2, 519-534, 1994
- T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Computer methods in applied mechanics and engineering, vol 139, 3-47, 1996
- J. Yvonnet, Nouvelles approches sans maillage basées sur la méthode des éléments naturels pour la simulation numérique des procédés de mise en forme, Thèse de doctorat, Ecole Nationale Supérieur d'Arts et Métiers Centre Paris, 2004.