An extra dof-free and well-conditioned XFEM

Longfei Wen[1,2], *Rong Tian[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Kexueyuan Nanlu 6, Haidian, Beijing 100190, China
[2]University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing 100049, China

*Corresponding author: rongtian@NCIC.AC.CN

Abstract

An improved XFEM (in short, *i*XFEM) is introduced. The core of *i*XFEM is an extra-dof free and *interpolating* PU enrichment based on a (moving) least-squares. Compared with the current XFEM, *i*XFEM does not introduce extra dof in PU enrichment and is well-conditioned in geometrical refinement. Selected numerical examples are provided to demonstrate its numerical performance.

Keywords: selectively interpolating moving least squares, partition of unity, XFEM

Introduction

The extended finite element method (XFEM [Belytschko and Black (1999); Moes *et al*. (1999); Belytschko *et al*. (2001)]) is a numerical technique based on the generalized finite element method (GFEM [Duarte *et al*. (2000); Duarte *et al*. (2001); Simone et al. (2006); Duarte and Kim (2008); O'Hara *et al*. (2009); Strouboulis *et al*. (2000a; 2000b; 2001)]) and the partition of unity method (PUM [Babuška and Melenk (1996; 1997)]). Only local parts of the domain are enriched and the mesh generation does not need to consider the internal structure. It adds special enrichment functions to approximate discontinuous fields and has been used for general interface phenomena e.g. in the framework of multi-material problems [Sukumar et al. (2001)], solidification [Chessa *et al*. (2002)], shear bands [Areias and Belytschko (2006)], dislocations [Belytschko and Gracie (2007)], and multi-field problems [Zilian and Legay (2008)].

Recently, a new GFEM without extra dof has been proposed as an improvement on the existing GFEM [Tian (2013)]. Based on the extra-dof free PU approximation, we have developed an improved XFEM with two distinguished features: (1) extra-dof-free; (b) well-conditioned in the so-called geometrical refinement, which means the size of enriched sub-domain is fixed during mesh refinement. This short paper is intended for a brief introduction to the improved version of XFEM and a comparison with the two existing counterparts: the originally standard XFEM and the corrected XFEM [Fries (2008)]. By standard XFEM we mean a XFEM without blending element treatment. The corrected XFEM is an improved version over the standard XFEM by eliminating the blending element issue using a neat ramp formulation.

The extra-dof free and interpolating XFEM

Extra-dof free PU enrichment

Let Ω^e be a sub-domain where all nodes are enriched. Let P_i^r denote nodal patch i composed of nodes from Ω^e, where r is the patch size. The patch size is either the size of nodal support combining $m \geq 1$ layer(s) of elements surrounding node i for a

structured or an non-uniform mesh or simply the radius of an influence circle at node *i* for a unstructured uniform mesh. Node *i* is called "patch star" of P_i^r, the topological centre of a patch. $\{x_k | x_k \in P_i^r\}$ is the node set of P_i^r, where index *i* is solely kept for the patch star and *k* $(k \neq i)$ for non-patch star nodes on the patch.

On the patch, we construct an interpolant of $u_i(\mathbf{x}),\ \mathbf{x} \in P_i^r$, using the nodal values at $\{x_k | x_k \in P_i^r\}$ as

$$u_i(x) = \sum_{k=1}^{n_i} L_k^{P_i^r}(x) u_k\ , \quad x, x_k \in \mathrm{P}_i^r \tag{0}$$

where $L_k^{P_i^r}(x)$ is the shape function at *k*-th node of the node set, which also forms a partition of unity, $\sum_{k=1}^{n_i} L_k^{P_i^r}(x) \equiv 1$, n_i is the number of nodes on P_i^r, and u_k is the conventional nodal unknown. Noted is that this approximation is local to patch P_i^r. The approximation is called patch-wise local approximation.

Using the patch-wise local approximation $u_i(\mathbf{x})$ as a local approximation at node *i* and substitute it for the nodal unknown u_i in the following standard FEM

$$u^h(x) = \sum_{i=1}^{N} N_i(x) u_i \tag{0}$$

we obtain a new PU approximation

$$u^h(x) = \sum_{i=1}^{N} \left(N_i \left(\sum_{k=1}^{n_i} L_k^{P_i^r} u_k \right) \right) \tag{0}$$

where *N* and u_k are the same as those in the standard FEM (2). The difference is that in the new approximation (3) each u_i now is associated with n_i enrichment functions:

$$\underbrace{N_i L_i^{P_i^r}}_{\text{patch star node}}\ ,\ \underbrace{N_1 L_i^{P_1^r},\ N_2 L_i^{P_2^r},\ \cdots,\ N_{k,k\neq i} L_i^{P_k^r}, \cdots,\ N_{n_i} L_i^{P_{n_i}^r}}_{\text{non patch star nodes}} \tag{0}$$

where $L_i^{P_k^r}$ (k = 1, 2, …i, …,n_i) are the local functions constructed on patch *k* with regard to node *i*, and index *k* can be understood as either the *k*-th node of the patch *i* or the *k*-th patch containing node *i*. Expanding and regrouping all the terms associated with u_i we obtain

$$u^h(x) = \sum_{i=1}^{N} \left(N_i L_i^{P_i^r} + \sum_{k=1, k\neq i}^{n_i} N_k L_i^{P_k^r} \right) u_i = \sum_{i=1}^{N} \tilde{N}_i u_i \tag{0}$$

where n_i, the number of nodes on patch *i* in (1), now really means the number of patches containing node *i*, but the two numbers are the same, and $\tilde{N}_i$ denotes the new shape function, which is a summation of n_i functions.

The new PU approximation offers two unique features. One is that there is no extra dof; all nodal dofs are the conventional of the standard FE mesh. The other is that the new PU approximation interpolates as long as the patch-wise local approximation interpolates at its patch star, *no matter it interpolates or not at the rest of nodes on the patch*. Proof is immediate: if and only if $L_i^{P_i^r}(\mathbf{x}_i) = 1$, then $u^h(\mathbf{x}_i) = u_i$ because $N_{k,k\neq i}(\mathbf{x}_i) = 0$. $L_i^{P_i^r}(\mathbf{x}_i) = 1$ is called a *one-point interpolating* condition.

In order to construct the one-point interpolating patch-wise local approximation, a selectively interpolating (moving) least squares approximation is employed.

Selectively interpolating (moving) least squares approximation

An approximation of the field function $u_i(x)$ on the patch *i* is constructed by,

$$u_i(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}),\ \mathbf{x} \in \mathrm{P}_i^r \tag{0}$$

where

$$\begin{aligned}\mathbf{p}(\mathbf{x}) &= \left[p_1(\mathbf{x}), p_2(\mathbf{x}), ..., p_m(\mathbf{x})\right]^{\mathrm{T}} \\ &= \left[1, \frac{x-x_i}{ch}, \frac{y-y_i}{ch}, f(r,\theta) - f(r_i,\theta_i)\right]^{\mathrm{T}}\end{aligned} \tag{0}$$

$$f(r,\theta) = \sqrt{r}\cos\frac{\theta}{2},\ \sqrt{r}\sin\frac{\theta}{2},\ \sqrt{r}\sin\frac{\theta}{2}\sin\theta,\ \sqrt{r}\cos\frac{\theta}{2}\sin\theta$$

is the normalized or shifted crack tip basis vector in two dimensions, and **a** is the vector of unknown coefficients. In order to construct a moving least-squares approximation passing the patch star, the discrete L_2 error norm is defined by the following *constrained* form,

$$J = \frac{1}{2}\sum_{k=1}^{n_i} w_k(\mathbf{x})\left(\mathbf{p}^{\mathrm{T}}(\mathbf{x}_k)\mathbf{a} - u_k\right)^2 + \lambda\left(\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{a} - u_i\right) \tag{0}$$

For implemental convenience, the patch star *i* is also included in the summation term, λ is the Lagrange multiplier that is used to enforce the satisfaction of

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{a} = u_i \tag{0}$$

Minimizing the L_2 norm with regard to **a** and $\lambda\ (\partial J/\partial \mathbf{a} = 0, \partial J/\partial \lambda = 0)$ leads to a Selectively interpolating moving least squares approximation:

$$u_i(\mathbf{x}) = \sum_{k=1}^{n_i} \phi_k^{\mathrm{P}_i^r}(\mathbf{x}) u_k \tag{0}$$

$$\phi_k^{\mathrm{P}_i^r}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\left(\mathbf{A}^{-1}\mathbf{p}_k - \frac{1}{\mathbf{A}_{11}^{-1}}\mathbf{A}_{(1)}^{-1}\mathbf{A}_{(1)}^{-\mathrm{T}}\mathbf{p}_k + \frac{1}{\mathbf{A}_{11}^{-1}}\mathbf{A}_{(1)}^{-1}\delta_{ik}\right) \tag{0}$$

$$A = \sum_{k=1}^{n_i} w_k p_k p_k^T \tag{0}$$

where $A_{(1)}^{-1}$ is the first column of $\mathbf{A}^{-1}$, A_{11}^{-1} is the first element of $A_{(1)}^{-1}$, δ is the Kronecker delta. It can be verified that the SIMLS shape function $\phi_k^{\mathrm{P}_i^r}(\mathbf{x})$ interpolates at patch star *i*

$$\phi_i^{\mathrm{P}_i^r}(\mathbf{x}_i) = 1,\ \phi_{j,j\neq i}^{\mathrm{P}_i^r}(\mathbf{x}_i) = 0 \tag{0}$$

while approximating at the non-patch star nodes.

By letting $w_k \equiv 1$, the local approximation is reduced to a least-squares type which is termed the Selectively Interpolating Least-Squares (SILS) local approximation. *The SILS local approximation simplifies the calculation of derivatives and therefore is much more computationally cheap*.

For both the SIMLS and the SILS, it can be verified that

$$\phi_i^{\mathrm{P}_i^r}(\mathbf{x}_i)=1,\ \phi_{j\neq i}^{\mathrm{P}_i^r}(\mathbf{x}_i)=0 \text{ and } u_i(\mathbf{x}_i)\equiv u_i \tag{0}$$

which means that the local approximation interpolates at the patch star. It can be easily verified that *using this kind of local approximation, iXFEM interpolates at all nodes although the local operation is in nature an approximation.*

Due to the limit of space, the rest details of the method are omitted here and will be presented in the conference.

Selected numerical examples

A crack tip benchmark problem is shown in Fig. 1. A square area with side-length of 2*a* and a crack of *a*, where *a* = 0.5. The uniform mesh is used. The enriched sub-domain is an area containing the crack tip while the rest of the domain is the standard FEM. The enriched sub-domain is the same in size during mesh refinement, which is respectively the half (Fig. 1(a)) and the one third (Fig. 1(b)) of the domain size. The patch size takes $r=2h$ on Ω_{GFE}. 10×10 Gaussian quadrature is only used on the 2×2 elements directly neighbored to the crack-tip in Fig. 1(a) and subdivided 2×2 cells on the element containing the crack tip in Fig. 1(b). On the rest elements, 2×2 Gaussian quadrature is employed. Young's modulus E = 1000 and the Poisson ratio v = 0.3 are assumed for plain strain. Along the outer boundary of the area, the displacements are prescribed to the following exact solution:

$$u=2(1+v)\frac{K_{\mathrm{I}}}{E}\sqrt{\frac{r}{2\pi}}\cos\frac{\theta}{2}\left[1-2v+\sin^2\frac{\theta}{2}\right]$$

$$v=2(1+v)\frac{K_{\mathrm{I}}}{E}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2}\left[2-2v-\cos^2\frac{\theta}{2}\right]$$

where K_{I} is taken as 1.253.

The two patterns of crack layout, mesh aligned and mesh independent, are considered for convergence tests (Fig. 1). The SIMLS and the SILS are tested and compared.

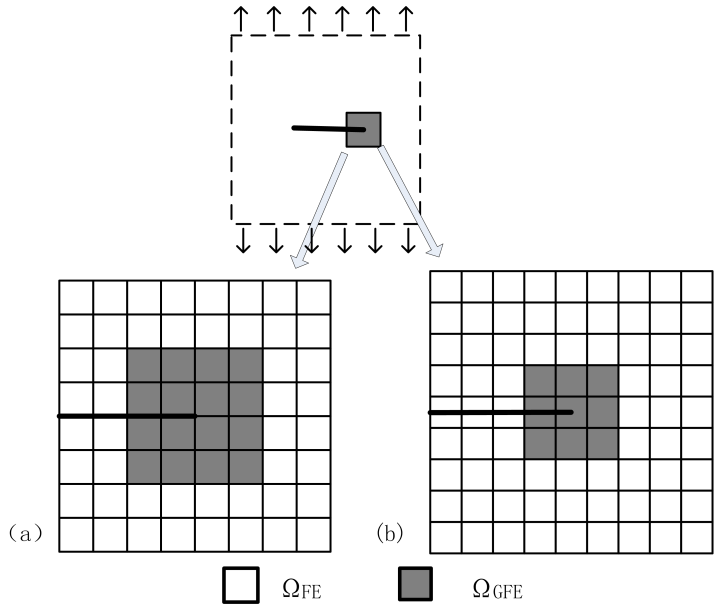

Figure 1. Crack tip benchmark problems. (a) mesh aligned crack; (b) mesh independent crack.

Assessment on convergence in geometrical refinement

Errors are measured by the L_2 and energy norms defined respectively below

$$\|u\| = \left[\int_\Omega \left(\mathbf{u}^h - \mathbf{u}^{\text{exact}}\right)^{\mathrm{T}} \left(\mathbf{u}^h - \mathbf{u}^{\text{exact}}\right) \mathrm{d}\Omega\right]^{\frac{1}{2}}, \|e\| = \left[\int_\Omega \left(\boldsymbol{\varepsilon}^h - \boldsymbol{\varepsilon}^{\text{exact}}\right)^{\mathrm{T}} \left(\boldsymbol{\sigma}^h - \boldsymbol{\sigma}^{\text{exact}}\right) \mathrm{d}\Omega\right]^{\frac{1}{2}}$$

Convergences are tested in the case of the so-called geometrical refinement. The two kernels, the SIMLS and the SILS, are tested. The results are displayed in Fig. 2. It is shown that *i*XFEM delivers optimal convergence for the crack tip benchmark problem.

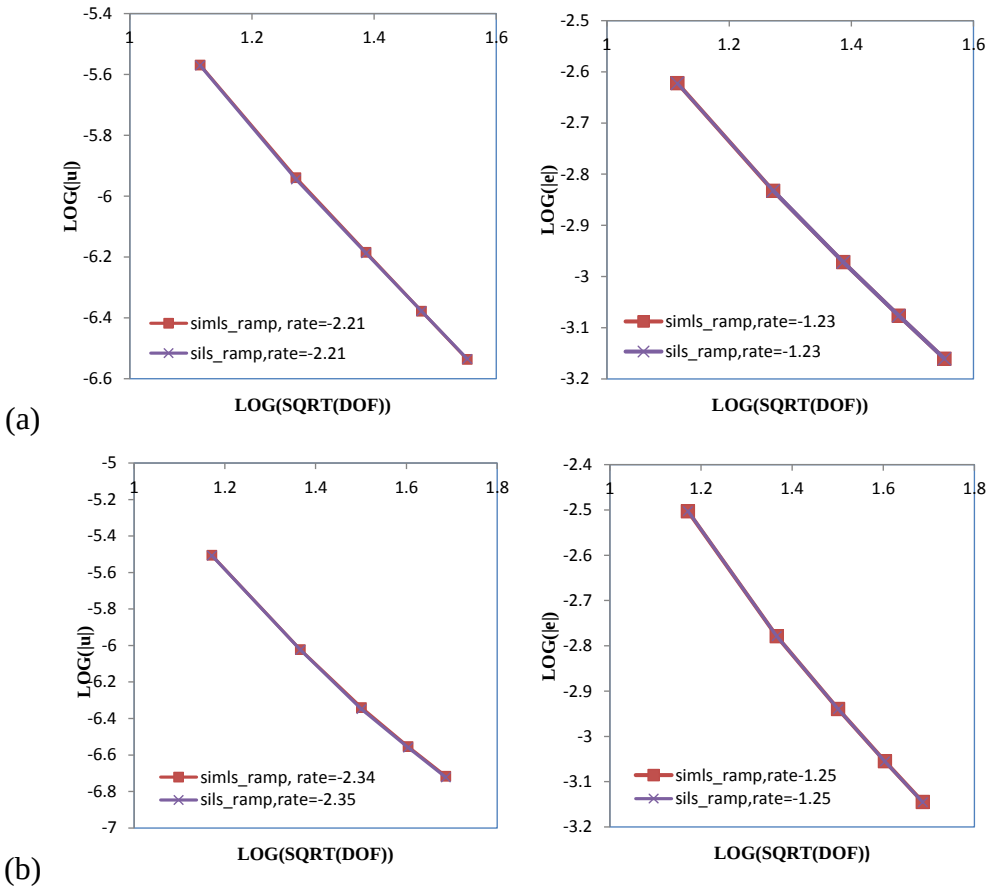

Figure 2. Convergences in geometrical refinement.
(a) mesh aligned crack; (b) mesh independent crack.

Assessment on accuracy of stress intensity factors

Comparisons are based on the crack configuration of Fig. 1(b). The SILS approximation is adopted as it is the same accurate as the SIMLS (refer to Fig. 2) but is more computationally efficient. The patch size takes r=2h. *i*XFEM and XFEM use the same size of enriched domain. *i*XFEM is compared with the standard XFEM and the corrected XFEM proposed by Fries [Fries (2008)]. 7×7 Gaussian quadrature is used on the enriched elements and 2×2 on the rest. The embedded direct solver of Matlab® is used for linear systems. This circumvents bad conditioning in XFEM, but the solver complains the singularity of the linear system in the corrected XFEM. The normalized K_{I} values for each method are shown in Table 1.

Table 1. Normalized K_I values

element size (h)	*i*XFEM		XFEM		corrected XFEM	
	dof	$\boldsymbol{K_I}$	**dof**	$\boldsymbol{K_I}$	**dof**	$\boldsymbol{K_I}$
1/9	220	1.0075	344	1.0068	344	1.0045
1/15	540	1.0022	828	1.0007	828	1.0010
1/21	1004	1.0010	1520	1.0000	1520	1.0002
1/27	1612	1.0006	2420	0.9993	2420	0.9999
1/33	2364	1.0004	3528	0.9990	3528	0.9999
1/39	3260	0.9999	4844	0.9991	4844	0.9998
1/45	4300	0.9996	6368	0.9990	6368	0.9998
1/51	5484	0.9997	8100	0.9990	8100	0.9998

Assessment on conditioning properties

The conditioning is studied for the above same example by computing the maximum and the minimum eigenvalues of the global stiffness matrix before essential boundary treatment (the rigid body modes are included). The results are listed in Table 2. The eigenvalues of the standard FEM (crack and enrichment) are provided for reference. An eigenvalue less than 10^{-15} is taken to be zero. The variation of the condition number versus mesh size h is plotted in Fig. 3 for the standard XFEM, FEM and *i*XFEM.

Table 2. Conditioning properties
(the eigenvalues of the standard FEM are provided for reference)

	***i*XFEM**			**XFEM**		
element size (h)	**max eigenvalue**	**min eigenvalue**	**# of zero eigenvalues**	**max eigenvalue**	**min eigenvalue**	**# of zero eigenvalues**
1/9	6.61E+03	3.40	3	6.19E+03	1.14E-03	3
1/15	5.77E+03	1.47	3	6.44E+03	3.63E-05	3
1/21	5.60E+03	0.813	3	6.52E+03	2.93E-06	3
1/27	5.63E+03	0.517	3	6.57E+03	4.72E-07	3
1/33	5.66E+03	0.358	3	6.60E+03	1.15E-07	3
1/39	5.67E+03	0.263	3	6.62E+03	3.64E-08	3
1/45	5.68E+03	0.201	3	6.64E+03	1.34E-08	3
1/51	5.69E+03	0.159	3	6.65E+03	5.70E-09	3
	corrected XFEM			**FEM**		
element size (h)	**max eigenvalue**	**min eigenvalue**	**# of zero eigenvalues**	**max eigenvalue**	**min eigenvalue**	**# of zero eigenvalues**
1/9	6.10E+03	2.41E-06	**7**	5.18E+03	51.7	3
1/15	6.36E+03	4.17E-08	**7**	5.30E+03	22.7	3
1/21	6.44E+03	3.77E-09	**7**	5.34E+03	12.1	3
1/27	6.49E+03	6.83E-10	**7**	5.36E+03	7.50	3
1/33	6.53E+03	1.81E-10	**7**	5.37E+03	5.10	3
1/39	6.56E+03	6.11E-11	**7**	5.37E+03	3.69	3
1/45	6.58E+03	2.44E-11	**7**	5.37E+03	2.79	3
1/51	6.60E+03	1.10E-11	**7**	5.38E+03	2.19	3

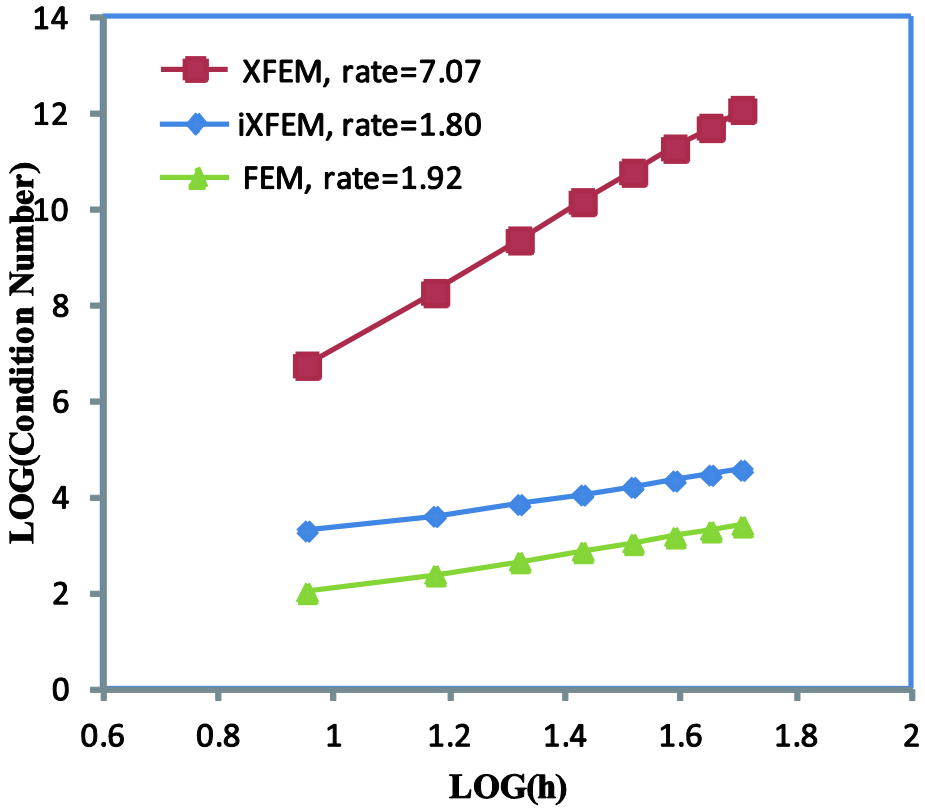

Figure 3. Variation of condition number versus mesh size.

The following conclusions can be drawn: (a) *i*XFEM offers the similar stability as the standard FEM (Fig.3) and it is much better conditioned than the current XFEM. The condition number of the global stiffness matrix in *i*XFEM grows with $h^{-1.8}$, where h is the mesh size, whereas the condition number in XFEM grows with h^{-7}; For reference, the condition number in the standard FEM grows with $h^{-1.9}$ in the same tests. (b) the corrected XFEM contains four spurious zero eigenvalues in stiffness matrices, which signals singularity of the stiffness matrix. The minimum nonzero eigenvalue is also generally smaller than that in XFEM.

Assessment on expenses on equation solving

The computational expense of the three methods is compared in terms of the number of dofs to be solved and the number of convergence iterations in a linear solve. The reason we choose an iterative solver for comparison is that subspace iterative methods are de-facto solvers in large scale problems and a direct solver is hardly scalable for problems at scale.

The conjugate gradient method is used to solve the linear system of the methods. The convergence tolerance is set to 10^{-10} for *i*XFEM and 10^{-8} for the XFEMs. The standard conjugate gradient method *without* preconditioning is used for *i*XFEM and the SSOR preconditioned conjugate gradient method, which is also an embedded solver of Matlab®, is used for XFEM and the corrected XFEM to circumvent bad conditioning. The data are listed in Table 3.

Although a better pre-conditioner or an alternative robust solver definitely can be found for the XFEMs, we still can safely conclude that the standard XFEM and the corrected XFEM obviously suffer from very slow convergence due to bad conditioning; *i*XFEM, in contrast, not only computes a smaller size of linear system but also converges remarkably easily. The latter observation is as expected and should be understandable.

Table 3. The number of dofs to be solved and convergence iterations

h	*i*XFEM (w/o preconditioning) (error tolerance: 10^{-10})		XFEM (preconditioned) (error tolerance: 10^{-8})		corrected XFEM (preconditioned) (error tolerance: 10^{-8})	
	dofs	***iterations***	***dofs***	***iterations***	***dofs***	***iterations***
1/9	220	70	344	276	344	447
1/15	540	93	828	1068	828	2337
1/21	1004	111	1520	2389	1520	5796
1/27	1612	130	2420	4395	2420	10000*
1/33	2364	149	3528	6801	3528	10000*

* the maximum number of iterations is reached but convergence is not yet observed

Crack growth simulation in a double cantilever beam

In this section, *i*XFEM is compared to the standard and the corrected XFEMs for crack growth simulation. The dimensions of the double cantilever beam (see Fig. 4) are 6cm × 2cm and an initial pre-crack with length of a = 2cm is considered. Plane stress conditions are assumed with Young's modulus E = 1000 and the Poisson ratio v = 0.3. The crack is given a small perturbation at the tip of length Δa=0.1cm in with initial angle dθ=5.71° as shown in Fig. 3. A structured mesh (60 × 180) is used and the crack advances 0.1cm at each step. The stress intensity factors are computed using the interaction integral method. The evolution of the crack paths are shown in Fig. 5. The crack path obtained using *i*XFEM for a fine mesh of 150 × 450 is provided as the reference solution.

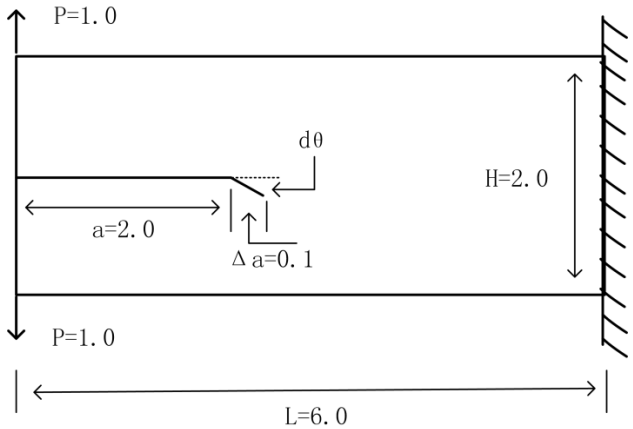

Figure 4. Geometry for the double cantilever beam

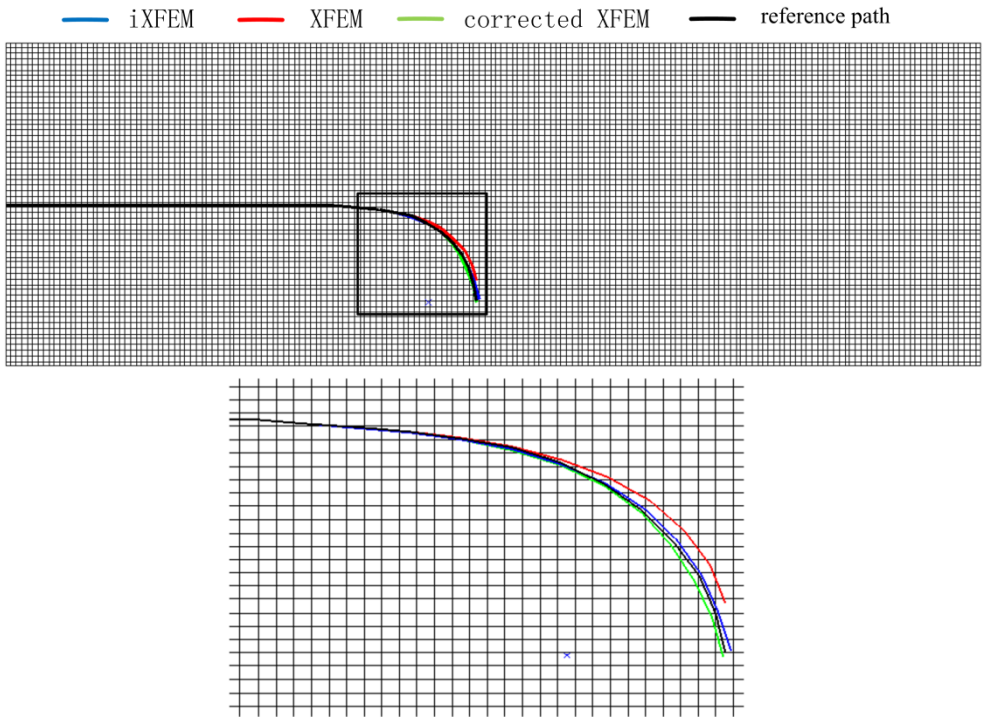

Figure 5. Double cantilever beam: comparison of crack path after 11 steps using the standard XFEM and *i*XFEM for dθ=5.71 °.

Due to the limit of space, the rest of numerical examples are to be presented in the conference.

Conclusions

An improved XFEM, in short *i*XFEM, has been briefly introduced. *i*XFEM has been compared with the standard XFEM and the corrected XFEM in terms of convergences, accuracy, conditioning properties, and the expenses on equation solving. The following conclusions are drawn:

(1) The *i*XFEM offers the similarly excellent accuracy as the standard XFEM and the corrected XFEM provided that the blending element issue is dealt with properly and the enriched domain is the same in size.

(2) *i*XFEM shows remarkable improvement on conditioning. The condition number of the global stiffness matrix in *i*XFEM grows with $h^{-1.8}$, where h is the mesh size, while the existing XFEM grows with h^{-7}. In the corrected XFEM, four spurious zero eigenvalues are detected, which means the singularity of global stiffness matrix. For reference, the condition number in the standard FEM grows with $h^{-1.9}$ in the same tests. *i*XFEM shows the similar conditioning as the standard FEM. Due to the good conditioning, *i*XFEM is robust to deliver optimal convergences in geometrical refinement.

(3) *i*XFEM is computationally efficient. The shape function of *i*XFEM involves a matrix inversion operation. This increases computational expenses to a certain extent. On the other hand, for the same size of enriched domain, *i*XFEM uses less dofs and convergence iteration is also much faster compared with the standard and the corrected XFEMs.

Acknowledgements

The MATLAB XFEM code of Pais [Pais (2011)] is used for the standard XFEM computation. The corrected XFEM was implemented based on this code. The work was supported by National Foundation of Sciences of China (Grant #: 91130026).

References

Belytschko T., and Black, T. (1999) Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* **45**, 601–620.

Moes, N., Dolbow, J. and Belytschko T. (1999) A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* **46**,131–150.

Belytschko, T. Moes, N., Usui, S., and Parimi C.(2001) Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* **50**, 993–1013.

Duarte, C.A., Babuska, I., and Oden, J.T. (2000) Generalized finite element methods for three-dimensional structural mechanics problems. *Computers & Structures* **77**, 215–232.

Duarte, C.A., Hamzeh, O.N., Liszka, T.J., and Tworzydlo W.W. (2001) A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering* **190**, 2227–2262.

Simone, A., Duarte, C.A., and Van der Giessen E. (2006) A generalized finite element method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering* **67**, 1122–1145.

Duarte C.A., and Kim D.J. (2008) Analysis and applications of a generalized finite element method with global-local enrichment functions. *Computer Methods in Applied Mechanics and Engineering* **197**(6-8), 487–504.

O'Hara, P., Duarte, C.A., and Eason T. (2009) Generalized finite element analysis for three dimensional problems exhibiting sharp thermal gradients. *Computer Methods in Applied Mechanics and Engineering* **198**, 1857–1871.

Strouboulis, T., Babuška, I, and Copps K. (2000) The design and analysis of the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering* **181**(1–3), 43–69.

Strouboulis, T., Copps, K., and Babuška, I. (2000) The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering* **47**, 1401–1417.

Strouboulis, T., Copps, K., and Babuška, I. (2001) The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering* **190**(32–33), 4081– 4193.

Strouboulis, T., Zhang, L., and Babuška, I. (2003) Generalized finite element method using mesh-based handbooks: application to problems in domains with many voids. *Computer Methods in Applied Mechanics and Engineering* **192**, 3109–3161.

Babuška, I., Melenk, J.M. (1997) Partition of unity method. *International Journal for Numerical Methods in Engineering* **40**, 727 –758.

Melenk, J.M., and Babuška, I. (1996) The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* **139**, 289 –314.

Sukumar, N., Chopp, D.L., Moes, N., and Belytschko, T. (2001) Modeling holes and inclusions by level sets in the extended finite-element method. *Comp. Methods Appl. Mech. Engrg.* **190**, 6183 – 6200.

Chessa, J., Smolinski, P., and Belytschko, T. (2002) The extended finite element method (XFEM) for solidification problems. *International Journal for Numerical Methods in Engineering* **53**, 1959–1977.

Areias, P.M.A. and Belytschko, T. (2006) Two-scale shear band evolution by local partition of unity. *International Journal for Numerical Methods in Engineering* **66**, 878–910.

Belytschko, T. and Gracie, R. (2007) On XFEM applications to dislocations and interfaces. *International Journal of Plasticity* **23**, 1721 –1738.

Zilian, A. and Legay A. (2008) The enriched space-time finite element method (EST) for simultaneous solution of fluid-structure interaction. *International Journal for Numerical Methods in Engineering* **75**, 305–334.

Tian, Rong (2013) Extra-dof-free and linearly independent enrichments in GFEM. *Computer Methods in Applied Mechanics and Engineering* **266**: 1-22.

Fries T.P.. (2008) A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering* **75**(5), 503-532.

Pais, M., MATLAB Extended Finite Element (MXFEM) Code v1.3, www.matthewpais.com, 2011.