Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 367802, 8 pages http://dx.doi.org/10.1155/2014/367802



## Research Article

# A New Iterative Method for Linear Systems from XFEM

## Jianping Zhao, 1,2 Yanren Hou, 2 Haibiao Zheng, 2 and Bo Tang 3

- <sup>1</sup> College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, China
- <sup>2</sup> School of Mathematics and Statistics, Center for Computational Geosciences, Xi'an Jiaotong University, Xi'an 710049, China
- <sup>3</sup> School of Mathematics and Computer, Hubei University of Arts and Science, Xiangyang, Hubei 441053, China

Correspondence should be addressed to Haibiao Zheng; hbzheng@xjtu.edu.cn

Received 12 September 2013; Accepted 16 December 2013; Published 14 January 2014

Academic Editor: Trung Nguyen Thoi

Copyright © 2014 Jianping Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The extend finite element method (XFEM) is popular in structural mechanics when dealing with the problem of the cracked domains. XFEM ends up with a linear system. However, XFEM usually leads to nonsymmetric and ill-conditioned stiff matrix. In this paper, we take the linear elastostatics governing equations as the model problem. We propose a new iterative method to solve the linear equations. Here we separate two variables U and Enr, so that we change the problems into solving the smaller scale equations iteratively. The new program can be easily applied. Finally, numerical examples show that the proposed method is more efficient than common methods; we compare the  $L_2$ -error and the CPU time in whole process. Furthermore, the new XFEM can be applied and optimized in many other problems.

## 1. Introduction

Many physical phenomena can be modeled by partial differential equations with singularities and interfaces. Much attention has been focused on the development of interface problem in fluid dynamics and material science. The standard finite difference and finite element methods may not be successful in giving satisfactory numerical results for such problems. Hence, many new methods have been developed. Some of them are developed with the modifications in the standard methods, so that they can deal with the discontinuities and the singularities. Peskin developed immersed boundary method (IBM) [1] basically to model blood flow in the heart. Immersed interface method (IIM) [2-6] was basically developed because of the need for a second-order accurate finite difference method to solve linear elliptic, parabolic partial differential equations in which an integral equation may not be available. IIM will give a nonsymmetric system of equations. The extended finite element method (XFEM) [7, 8] enables the accurate approximation of solutions that involve jumps, kinks, singularities, and other locally nonsmooth features within elements. This is achieved by enriching the polynomial approximation space of the classical finite

element method. The generalized finite element method (GEFM)/XFEM [9-11] has shown its potential in a variety of applications that involve nonsmooth solutions near interfaces. Among them are the simulation of cracks, shear bands, dislocations, solidification, and multifield problems. A large number of examples can be found in the real world where field quantities change rapidly over length scales that are small with respect to the observed domain. For the modeling of such phenomena, the resulting solutions typically involve discontinuities, singularities, high gradients, or other nonsmooth properties. For the approximation of nonsmooth solutions, one of the approaches is to enrich a polynomial approximation space such that the nonsmooth solutions can be modeled independently of the mesh. Methods that extend polynomial approximation spaces are called "enriched methods" in this work. The enrichment can be achieved by adding special shape functions (which are customized so as to capture jumps, singularities, etc.) to the polynomial approximation space. The total stiffness matrix consists of four parts: the FEM part, the enrich nodes part, and another two parts are the connecting parts between the real nodes and enrich nodes. Compared with FEM, the matrix size increases O(N), while the FEM stiff matrix size is  $O(N^2)(N \sim O(h^{-1}))$ .

But engineering calculations in XFEM always encountered the problem of solving ill-conditioned matrix. It made the XFEM widely.

In this paper, according to the characteristics of the stiffness matrix, we use an iterative method to avoid dealing with complicated stiff matrix. First we can use the result of linear equation from standard FEM. The result can be used to be the initial value for iteration. Then we decompose the unknown values to two parts; each part is governed by two linear equations, whose dimension is smaller than the whole stiff matrix. The new programm can be easily applied. Finally, numerical examples show that the proposed method is more efficient than common methods. Furthermore, the new method works well for some problems while the common method (such as LU) fails. The new iterative method can be applied to many other problems.

The outline of this paper is as follows. Section 2 introduces the XFEM. Subsequently, the main part of this paper is Section 3 that we describe a new efficient and feasible method. We also analyze the time complexity of the algorithm. In Section 4, we briefly mention details on numerical examples. A final conclusion is drawn in Section 5.

#### 2. The Extended Finite Element Method

The basic mathematical foundation of the partition of unity finite element method (PUFEM) was discussed in [12, 13]. They illustrated that PUFEM can be used to employ the structure of the differential equation under consideration to construct effective and robust methods. The global solution of PUFEM has been the theoretical basis of the local partition of unity finite element method, to be called later the extended finite element method.

The first effort for developing the extended finite element method can be traced back to 1999, [7] presented a minimal enmeshing finite element method for crack growth. They added discontinuous enrichment functions to the finite element approximation to account for the presence of the crack. The method allowed the crack to be arbitrarily aligned within the mesh, though it required enmeshing for severely curved cracks. The method was improved and called extend Finite Element Method (XFEM) in [8, 9, 11, 14–16].

In mathematics, relevant theoretical parts of doing are almost same in [17–19]. Of course, XFEM is still evolving currently and keeping the new extension about (such as fluid-structure interaction): mathematics stiff matrix remained prone to the sick.

Consider a domain  $\Omega$  in  $\mathbb{R}^n$  which is discredited by a set of elements  $\Omega_e$  such that

$$\bigcup_{e \in s} \Omega_e = \Omega, \tag{1}$$

and a set of nodes

$$\mathcal{T} = \bigcup_{I \in \mathcal{N}} x_I. \tag{2}$$

The total ordering of the element basis is given by  $\varepsilon$  and the total ordering of the nodes is given by  $\mathcal{N}$ . The standard finite element basis is defined as follows:

$$\mathscr{B}^{\text{std}} = \bigcup_{I \in \mathcal{N}} N_I(x), \qquad (3)$$

where  $N_I(x)$  is the finite element basis or shape function of node I;  $N_I(x)$  is assumed to be of compact support and piecewise continuously differentiable. Typically,  $\mathcal{B}^{\text{std}}$  spans the space of piecewise continuous polynomials of a specific order. The XFEM aims to alleviate the burden associated with mesh generation for problems with voids and interfaces.. It does not require the finite element mesh to conform to internal boundaries. The essence of the XFEM lies in subdividing a model into two distinct parts: mesh generation for the domain (excluding internal boundaries) and enrichment of the finite-element approximation by additional functions. A partitioning of a typical domain into its enriched and unenriched subdomains is shown in Figure 2(a). We defined the subdomain where the enrichment is applied as  $\Omega^{Enr}$ This region's feature is that the enrichment is dominant. The elements covering  $\Omega^{Enr}$  are the enriched elements, so

$$\bigcup_{e \in \varepsilon^{\rm Enr}} \Omega_e = \Omega^{\rm Enr},\tag{4}$$

where  $\varepsilon^{\rm Enr}$  is the ordering of the subset of elements which are to be enriched. Any node I for which the support of  $N_I(x)$  overlaps or is contained in  $\Omega^{\rm Enr}$  is enriched node, and the set is defined by the ordering  $\mathscr{N}^{\rm Enr}$ ,  $\mathscr{N}^{\rm Enr}\subset\mathscr{N}$ . For an enrichment function  $\psi(x)$ , the enriched basis for the local partition of unity method is

$$\mathcal{B}^{\mathrm{Enr}} = \mathcal{B}^{\mathrm{std}} \oplus \bigcup_{I \in \mathcal{N}^{\mathrm{Enr}}} N_{I}(x) \psi(x). \tag{5}$$

The approximation enriched with a local partition of unity the field u(x) is given by the sum of a standard finite element approximation and a linear combination of the enriched basis functions

$$u^{h}(x) = \sum_{I \in \mathcal{N}} N_{I}^{\text{std}}(x) u_{I} + \sum_{I \in \mathcal{N}^{\text{Enr}}} N_{J}^{\text{Enr}}(x) \psi(x) q_{J}, \quad (6)$$

where  $N_I^{\rm std}(x)$  and  $N_J^{\rm Enr}(x)$  are the shape functions for the standard part and the partition of unity, respectively; different order interpolates may be used for the standard and enriched shape functions.

For example, we consider the linear elastostatics governing equations

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0$$
 in  $\Omega$ ,  
 $\boldsymbol{\sigma} = \mathbf{C} : \varepsilon$ , (7)  
 $\varepsilon = \nabla_{s} \mathbf{u}$ ,

where  $\sigma$  is cauchy stress and  $\mathbf{b}$  is the body force per unit volume. The first function is called equilibrium equation. The second function is the equilibrium equation that describes

the relation of strain and displacement. The third equation is for Hooke's law. The parameter C implies two parameters  $\mu = G = E/(2(1+\nu))$  and shear elasticity,  $\lambda = E\nu/(1+\nu)(1-2\nu)$  (Lamé constants are  $\lambda$ ,  $\mu$  and Poisson's ratio is  $\nu$ ). Consider

$$\mathbf{u} = \overline{\mathbf{u}} \quad \text{on } \Gamma_{u},$$

$$\sigma \cdot \mathbf{n} = \mathbf{t} \quad \text{on } \Gamma_{t},$$

$$\sigma \cdot \mathbf{n}_{h}^{i} = 0 \quad \text{on } \Gamma_{h}^{i} \quad (i = 1, 2, ..., m),$$

$$\left[\sigma \cdot \mathbf{n}_{I}^{i}\right] = 0 \quad \text{on } \Gamma_{I}^{i} \quad (i = 1, 2, ..., n).$$
(8)

The boundary  $\Gamma$  is composed of the sets  $\Gamma_u$ ,  $\Gamma_t$ ,  $\Gamma_h^i$ , and  $\Gamma_I^i$  such that

$$\Gamma = \Gamma_{i} \cup \Gamma_{i} \cup \Gamma_{i}^{i} \cup \Gamma_{i}^{i}, \tag{9}$$

where n is the unit outward normal to  $\Omega$  and  $\overline{u}$  and  $\mathbf{t}$  are prescribed displacements and traction, respectively. The weak formulation of (12) is well known and reads as follows: find  $u \in V$  such that

$$\int_{\Omega} \sigma(\mathbf{u}) : \varepsilon(\mathbf{v}) d\Omega = \int_{\Omega} \mathbf{b} \cdot \mathbf{v} d\Omega + \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma \quad \forall \nu \in V_0.$$
(10)

Here  $V = H^1(\Omega)$  and  $V_0 = \{ v \in V : v = 0 \text{ on } \Gamma_u \}.$ 

For the finite element discretization, let  $\tau_h$  be a regular rectangular element of domain  $\Omega$  and define the mesh parameter  $h = \max_{T \in \tau_h} \{ \operatorname{diam}(T) \}$ .

Consider the Bubnov-Galerkin implementation for the XFEM in two-dimensional linear elasticity. In the XFEM, finite-dimensional subspaces  $V^h \subset V$  and  $V^h_0 \subset V_0$  are used as the approximating trial and test spaces. The finite dimensional space  $V^h$  is given by

$$V^{h} = \left\{ v \in V : v|_{T} \in Q_{k}\left(T\right), \forall T \in \tau_{h} \right\}. \tag{11}$$

The weak form of the discrete problem can be stated as follows: find  $u^h \in V^h$  such that

$$\int_{\Omega^{h}} \sigma\left(\mathbf{u}^{h}\right) : \varepsilon\left(\mathbf{v}^{h}\right) d\Omega$$

$$= \int_{\Omega^{h}} \mathbf{b} \cdot \mathbf{v}^{h} d\Omega + \int_{\Gamma_{t}^{h}} \bar{\mathbf{t}} \cdot \mathbf{v}^{h} d\Gamma \quad \forall v^{h} \in V_{0}^{h} \subset V_{0}, \tag{12}$$

where  $V^h \subset V$  and  $V^h_0 \subset V_0$ . In a Bubnov-Galkerkin procedure, the trial function  $u^h$  and the test function  $v^h$  are represented as linear combinations of the same shape functions. The trial and test functions are

$$\mathbf{u}^{h}(\mathbf{x}) = \sum_{I} \phi_{I}(\mathbf{x}) \, \mathbf{u}_{I} + \sum_{J} \phi_{J} \psi(\varphi(\mathbf{x})) \, \mathbf{a}_{J},$$

$$\mathbf{v}^{h}(\mathbf{x}) = \sum_{I} \phi_{I}(\mathbf{x}) \, \mathbf{v}_{I} + \sum_{I} \phi_{J} \psi(\varphi(\mathbf{x})) \, \mathbf{b}_{J},$$
(13)

where *I* denotes the set of all nodes in the mesh and  $J = \{j \in I : O_{\varepsilon} \mathbf{x}_{i} \cup \Gamma \neq \emptyset\} (\varepsilon \leq h)$  denotes the set of nodes near the

interface.  $\phi_I(\mathbf{x})$  are the finite-element shape functions,  $\varphi(\mathbf{x})$  is the level set function, and  $\psi(\varphi(\mathbf{x}))$  is the enrichment function for material interface.

There are several kinds of enrichment functions, such as abs-enrichment function, step-enrichment function, and Ramp-enrichment function. The choice of function is based on the behavior of the solution near the interface.

The Ramp function is defined as

$$\psi_{\text{Ramp}} = \begin{cases} 1 & \varphi > 0 \\ 1 - 2\varphi & \varphi \le 0. \end{cases}$$
 (14)

This enrichment function yields only continuous solutions. The advantage is that it automatically satisfies the continuity condition [u] = 0 and does not require the use of Lagrange multipliers.

The step-enrichment function is defined as

$$\psi_{\text{Step}} = \begin{cases} 1 & \varphi > 0, \\ -1 & \varphi \le 0. \end{cases} \tag{15}$$

This enrichment function can yield a continuous or discontinuous solution across the interface but requires Lagrange multipliers to apply the Dirichlet jump condition.

On substituting the trial and test functions form (17), and using the arbitrariness of nodal variations, the following discrete system of linear equations is obtained:

$$Kd = F, (16)$$

where

$$\mathbf{K}_{IJ} = \int_{\Omega^h} \mathbf{B}_I^T \mathbf{C} \mathbf{B}_J \, d\Omega, \tag{17}$$

$$\mathbf{F}_{I} = \int_{\Gamma_{t}^{h}} \widehat{\boldsymbol{\phi}}_{I} \bar{\mathbf{t}} \, d\Gamma + \int_{\Omega^{h}} \widehat{\boldsymbol{\phi}}_{I} \mathbf{b} \, d\Omega. \tag{18}$$

 $\widehat{\phi}_I \equiv \phi_I$  for a finite-element displacement degree of freedom, and  $\widehat{\phi}_I \equiv \phi_I \psi$  for an enriched degree of freedom. In the above equations, **C** is the constitutive matrix for an isotropic linear elastic material:

$$C = \begin{pmatrix} \lambda + 2G & 1 & 0 \\ 1 & \lambda + 2G & 0 \\ 0 & 0 & G \end{pmatrix}, \tag{19}$$

and the matrix  $B_I$  is given by

$$\mathbf{B}_{I} = \begin{pmatrix} \widehat{\phi}_{I,x} & 0\\ 0 & \widehat{\phi}_{I,y}\\ \widehat{\phi}_{I,y} & \widehat{\phi}_{I,x} \end{pmatrix}. \tag{20}$$

We can get the integral terms, and the matrices computed from the integral terms are block matrices of the form

$$K = \begin{pmatrix} K^{UU} & K^{UA} \\ K^{AU} & K^{AA} \end{pmatrix}, \qquad F = \begin{pmatrix} F^{U} \\ F^{A} \end{pmatrix},$$

$$d = \begin{pmatrix} U \\ \text{Enr} \end{pmatrix}.$$
(21)

The next step is solving the linear equations system.

## 3. The Program for Solving the Equation

We need to solve the following equations:

$$\begin{pmatrix} M_{n\times n}^{11} & M_{n\times m}^{12} \\ M_{m\times n}^{21} & M_{m\times m}^{22} \end{pmatrix} \begin{pmatrix} U_{n\times 1} \\ \operatorname{Enr}_{m\times 1} \end{pmatrix} = \begin{pmatrix} R_{n\times 1}^{1} \\ R_{m\times 1}^{2} \end{pmatrix}. \tag{22}$$

Here  $n \sim O(N^2)$  and  $m \sim O(N)$ , the block matrix  $M^{11}$  is same in FEM and  $(M^{21})' = M^{12}$  are sparse matrix.

We must point out that the matrix  $M^{12}$  has up to  $m^2$  nonzero elements. We can use the modified program, in (22) can be converted to two sub-problems: to solve  $(M^{22})^{-1}$  and solve U by n-order linear equations. Now, we list the detailed steps.

Algorithm 1. (i) We use the ordinary method to solve

$$M^{11}U^{(0)} = R^1, (23)$$

the results for the FEM equation and set the result as  $U^{(0)}$ ,  $E^{(0)} = \mathbf{0}$ . The matrix  $M^{11}$  is block tridiagonal symmetric, sparse matrix. There are a lot of methods to solve this equation efficiently.

(ii) Solve the small scale equation

$$M^{22} \operatorname{Enr}^{(i+1)} = R^2 - M^{21} U^{(i)}, \quad i = 0, 1, \dots$$
 (24)

The matrix  $M^{22}$  is a sparse tridiagonal, where the dimension is much smaller than  $M^{11}$ . This equation can be solved quicker than (i).

(iii) Then we use (25) to solve  $U^{(i+1)}$ ;

$$M^{11}U^{(i+1)} = R^1 - M^{12} \operatorname{Enr}^{(i+1)}, \quad i = 0, 1, \dots,$$
 (25)

the computation time complexity is the same with the first one.

(iv) If the norm  $\|U^{(i+1)} - U^{(i)}\| \le w$  or  $i > I_{\max}$  (giving the maximum iterative times), output the result  $U = U^{(i+1)}$ , Enr = Enr<sup>(i+1)</sup>. Else, return to step (ii).

In this part, we can use the common FEM solver such as LU or other useful solvers. We do not change the stiffness matrix  $M^{11}$  and  $M^{22}$ , these two matrixes are symmetric, positive, definite and banded. We can compute easily for (23), (24), (25). The matrix conditions is not changed;  $\Re(M^{11}) = O(h^{-2})$  [17]. First of all, the store can choose a one-dimensional. We can choose a one-dimensional storage when we use triangular decomposition for the matrix M.

*Remark 2.* If the matrix  $M^{11}$  and  $A^{22}$  are not tridiagonal, we often use LU or GMRES method [20] to solve (23), (24), and (25)

The iteration has the following advantages.

- (1) Other than the half bandwidth of zero elements do not have storage, nor involved in the calculation, the amount of storage crunch, the impact of rounding error is relatively small, even if it is ill-conditioned matrix, we often obtain a higher accuracy.
- (2) The operator only uses FEM linear equation solver; it is a simple calculation.

- (3) The original matrix deposit triangular matrix and diagonal matrix deposit area can overlap, thus avoiding to take up a lot of the middle unit of work, saving memory.
- (4) This solution can be extended to block direct solution.
- (5) With the iterative method, it is easy to build and test the general program and you can more accurately estimate the computation time; the user is relatively easy to grasp the algorithm process.

Also if we using in [21] to solve the equation,  $((2N)^2 + 4N)$  equations use LU solver which need  $O((2N)^2 + 4N)^3$  Multiplication, here if the number of iterations is I, we use up to  $O((2N)^6 + (4N)^3)$  multiplications.

#### 4. Numerical Test

In this section, we report two experiments to verify the efficacy and accuracy of the new iterative method in XFEM. We use Matlab and modify software [22] to demonstrate Algorithm 1. For the convenience of presentation, we introduce the following notations:

SFEM means the standard finite element method; XFEM means the extended finite element method; *h* means the mesh size;

GMRES means the generalized minimal residual iterative method;

 $L_2$ -error means  $||u^h - u||_0$ ;

 $E_h$  means the  $L_2$ -error when the mesh size equals h; iterations mean the iterative times in the algorithm; orders mean the numerical error order by XFEM:

orders = 
$$\frac{\log(E_{h_1}/E_{h_2})}{\log(h_1/h_2)}$$
. (26)

Here  $h_1 \le h_2$ . First, we list Example 1 for one-dimensional problems.

Example 1. The exact solution is

$$u = \begin{cases} -\frac{(x-0.5)^2}{2k_2} - \frac{(x-0.5)(k_1 - k_2)}{4k_2(k_2 + k_1)} + \frac{1}{4(k_2 + k_1)}, \\ -\frac{(x-0.5)^2}{2k_1} - \frac{(x-0.5)(k_1 - k_2)}{4k_1(k_2 + k_1)} + \frac{1}{4(k_2 + k_1)}, \\ & \text{if } x \in \Omega_1, \end{cases}$$

$$(27)$$

Here  $\Omega_1=[0,0.5]$  and  $\Omega_2=(0.5,1]$  the coefficient k=1, if  $x\in\Omega_1, k=10$ , if  $x\in\Omega_2$ . And F is point force from the righ; F=10. We use Figure 1 to describe the  $L_2$  error change when we use different h. We can easily see that, when the step is smaller than 1/80, the whole computing time has economized one order of magnitude in Example 1.

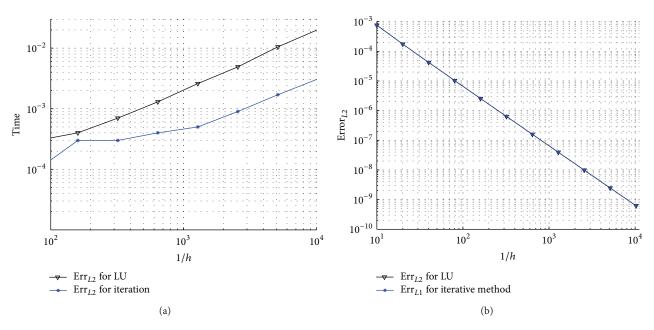


FIGURE 1: (a) The comparison of time by iteration and LU with different mesh. (b) The comparison of  $L_2$ -error by iteration and LU with different mesh.

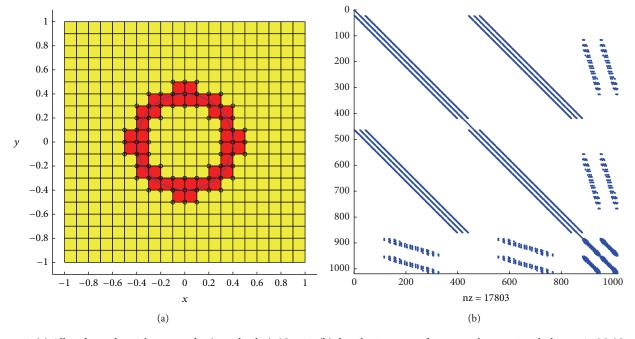


FIGURE 2: (a) All nodes and enrichment nodes (in red color); N = 20; (b) distribution map of nonzero elements in whole matrix M; N = 20.

*Example 2.* For the two-dimensional solid test, we use the same enrichment function and shape function to compute (4). We define  $\Omega = [0,1] \times [0,1]$  with the material interface

$$\Gamma: x^2 + y^2 = a^2 = 0.1608.$$
 (28)

Let Young's modulus and Poisson's ratio in

$$\Omega_1 = \left\{ (x, y) \in \Omega \mid x^2 + y^2 \le 0.404^2 \right\}$$
(29)

be  $E_1 = 100$ ,  $v_1 = 0$  and let those in

$$\Omega_2 = \left\{ (x, y) \in \Omega \mid x^2 + y^2 \le \frac{1}{4} \right\}$$
(30)

be  $E_2 = 10$ ,  $v_2 = 0$ , but area force in domain  $f_x = f_y = 0$ .

h	$\mathrm{DOF}_{\mathrm{FEM}}$	$\mathrm{DOF}_{\mathrm{XFEM}}$	$conds_FEM$	$\operatorname{conds}_{\operatorname{XFEM}}$	Sparsity of matrix (%)
$\frac{1}{20}$	882	1018	$1.7627 \times 10^4$	$7.90617 \times 10^4$	1.72
$\frac{1}{40}$	3362	3626	$3.75035 \times 10^4$	$3.20917 \times 10^5$	0.4774
$\frac{1}{60}$	12800	13120	$1.02381 \times 10^5$	$1.30023 \times 10^6$	0.222
$\frac{1}{80}$	12800	13120	$1.02381 \times 10^5$	$1.30023 \times 10^6$	0.13
$\frac{1}{100}$	25600	26240	$4.07606 \times 10^5$	$5.23974 \times 10^6$	0.0829

TABLE 1: DOF, sparsity of matrix, and condition number of linear equation by FEM and XFEM for Example 2.

TABLE 2: The CPU time for different solvers in Example 2.

h	$\mathrm{FEM}_{\mathrm{LU}}$	${\rm XFEM}_{\rm iteration}$	XFEM <sub>LU</sub> (s)	XFEM <sub>GMRES</sub> (s)
$\frac{1}{20}$	1.2166	1.3129	2.6019	2.1449
$\frac{1}{40}$	5.7133	6.5198	7.3678	8.9494
$\frac{1}{60}$	17.8348	20.6117	27.2810	26.6332
$\frac{1}{80}$	54.3007	59.0695	70.4340	70.7011
$\frac{1}{100}$	137.89	145.92	165.1256	166.5808

Table 3: The comparison of  $L_2$ -error with different solvers for Example 2.

h	${\sf FEM_{LU}}$	XFEM <sub>LU</sub> (s)	$\mathrm{XFEM}_{\mathrm{GMRES}}$	XFEM <sub>iteration</sub> (s)
$\frac{1}{20}$	$4.2101 \times 10^{-2}$	$6.4292 \times 10^{-3}$	$6.3756 \times 10^{-3}$	$6.3756 \times 10^{-3}$
$\frac{1}{40}$	$2.1032 \times 10^{-2}$	$1.7108 \times 10^{-3}$	$1.7097 \times 10^{-3}$	$1.7092 \times 10^{-3}$
$\frac{1}{60}$	$1.4089 \times 10^{-2}$	$7.7599 \times 10^{-4}$	$7.7595 \times 10^{-4}$	$7.7554 \times 10^{-4}$
$\frac{1}{80}$	$1.0317 \times 10^{-2}$	$4.1781 \times 10^{-4}$	$4.1434 \times 10^{-4}$	$4.1733 \times 10^{-4}$
$\frac{1}{100}$	$8.2802 \times 10^{-3}$	$2.7344 \times 10^{-4}$	$2.7002 \times 10^{-4}$	$2.7295 \times 10^{-4}$

We denote  $r = \sqrt{x^2 + y^2}$  and  $\tan \theta = y/x$ , and the exact solution is

$$u = \frac{a}{8\mu} \left( \frac{r}{a} (\kappa + 1) \cos \theta + 2\frac{a}{r} ((1 + \kappa) \cos \theta + \cos 3\theta) - 2\frac{a^3}{r^3} \cos 3\theta \right),$$

$$v = \frac{a}{8\mu} \left( \frac{r}{a} (\kappa - 3) \sin \theta + 2\frac{a}{r} ((1 - \kappa) \sin \theta + \sin 3\theta) - 2\frac{a^3}{r^3} \sin 3\theta \right).$$
(31)

Plate with a circular hole. Figure 2(b) shows the consistent rectangular subdivision and labels the nodes which need to be extended. The CPU time we counted is the total time for the entire programm running. Table 1 shows the linear equation scale and the condition numbers of FEM and XFEM, the density of sparse matrix show the matrix is sparse. It can be easily find the condition number of XFEM is bigger than that of FEM. The CPU times are also compared with LU solvers and GMRES in Table 2. The cheapest is LU by FEM and the new iterative method is the second cheapest; however, the error for LU by FEM is the biggest in all methods. Table 3 shows an  $L_2$ -error comparison between the new iterative method and other solvers. And the numerical  $L_2$ -error is nearly  $O(h^2)$ .  $L_2$ -error, order, and iterations are given in Table 4.

h	$\mathrm{Error}_{\mathrm{XFEM}}$	Orders	CPU <sub>XFEM</sub> (s)	Iterations
$\frac{1}{20}$	$6.3754 \times 10^{-3}$	\	1.313	24
$\frac{1}{40}$	$1.7092 \times 10^{-3}$	1.8992	6.52	19
$\frac{1}{60}$	$7.7554 \times 10^{-4}$	1.9489	20.61	19
$\frac{1}{80}$	$4.1733 \times 10^{-4}$	2.1541	59.07	19
$\frac{1}{100}$	$2.7295 \times 10^{-4}$	1.9028	145.92	18

Table 4: L<sub>2</sub>-error, order, CPU time, and iterations by XFEM with the new iterative method for Example 2.

#### 5. Conclusion

In this paper, we discuss how to use XFEM for the stiff matrix ill-conditioned and came to the following conclusions. Firstly, we found why some of problems cannot be computed by XFEM; the stiff matrix is not then symmetric and illconditioned. So, a lot of storage space is wasted. The time of computing XFEM equation is longer, which cannot be computed in certain subdivision. Secondly, we use XFEM for solving two-dimensional solid test; when solving the XFEM equations, the iterative method and the standard finite element equations solver are adopted, so that we change the problems into solving the smaller scale equations and standard finite element stiffness matrix need not be changed. Finally, numerical examples show that the proposed method is more efficient than the common method. We can use this programm in many other interface problems later. Besides, we also consider combining the stable GFEM with our iterative method in the future. And it is believed that the XFEM can be widely used in more problems in fluid dynamics and material science. The calculation module in some software can be improved for users.

#### **Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

The author also thanks the anonymous reviewers for their comments in improving the paper. This research was supported by the special funds for NSFC (1127313, 11171269, 11201369, and 61163027) and the Ph.D. Programs Foundation of the Ministry of Education of China (Grant no. 20110201110027), and it was partially subsidized by the Fundamental Research Funds for the Central Universities (Grant nos. 08142003 and 08143007).

## References

[1] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, 1972.

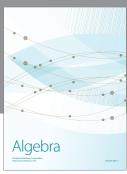
- [2] R. J. LeVeque and Z. L. Li, "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources," SIAM Journal on Numerical Analysis, vol. 31, no. 4, pp. 1019–1044, 1994.
- [3] Z. Li, K. Ito, and M.-C. Lai, "An augmented approach for Stokes equations with a discontinuous viscosity and singular forces," *Computers & Fluids*, vol. 36, no. 3, pp. 622–635, 2007.
- [4] A. L. Fogelson and J. P. Keener, "Immersed interface methods for Neumann and related problems in two and three dimensions," SIAM Journal on Scientific Computing, vol. 22, no. 5, pp. 1630–1654, 2000.
- [5] Z. Li, "The immersed interface method using a finite element formulation," *Applied Numerical Mathematics*, vol. 27, no. 3, pp. 253–267, 1998.
- [6] H. Huang and Z. Li, "Convergence analysis of the immersed interface method," *IMA Journal of Numerical Analysis*, vol. 19, no. 4, pp. 583–608, 1999.
- [7] N. Moës, J. Dolbow, and T. Belytschko, "A finite element method for crack growth without remeshing," *International Journal for Numerical Methods in Engineering*, vol. 46, no. 1, pp. 131–150, 1999
- [8] N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko, "Modeling holes and inclusions by level sets in the extended finite-element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 46-47, pp. 6183–6200, 2001.
- [9] J. Chessa, H. Wang, and T. Belytschko, "On the construction of blending elements for local partition of unity enriched finite elements," *International Journal for Numerical Methods in Engineering*, vol. 57, no. 7, pp. 1015–1038, 2003.
- [10] T.-P. Fries and T. Belytschko, "The extended/generalized finite element method: an overview of the method and its applications," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 3, pp. 253–304, 2010.
- [11] T. Belytschko, R. Gracie, and G. Ventura, "A review of extended/generalized finite element methods for material modeling," *Modelling and Simulation in Materials Science and Engineering*, vol. 17, pp. 1–24, 2009.
- [12] I. Babuška and J. M. Melenk, "The partition of unity method," International Journal for Numerical Methods in Engineering, vol. 40, no. 4, pp. 727–758, 1997.
- [13] J. M. Melenk and I. Babuška, "The partition of unity finite element method: basic theory and applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 289–314, 1996.
- [14] T.-P. Fries and T. Belytschko, "The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns,"

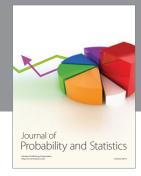
- *International Journal for Numerical Methods in Engineering*, vol. 68, no. 13, pp. 1358–1385, 2006.
- [15] J.-Y. Wu, "Unified analysis of enriched finite elements for modeling cohesive cracks," *Computer Methods in Applied Mechanics* and Engineering, vol. 200, no. 45-46, pp. 3031–3050, 2011.
- [16] H. Sauerland and T.-P. Fries, "The extended finite element method for two-phase and free-surface flows: a systematic study," *Journal of Computational Physics*, vol. 230, no. 9, pp. 3369–3390, 2011.
- [17] I. Babuška and U. Banerjee, "Stable generalized finite element method (SGFEM)," Computer Methods in Applied Mechanics and Engineering, vol. 201–204, pp. 91–111, 2012.
- [18] T. Strouboulis, I. Babuška, and K. Copps, "The design and analysis of the generalized finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 181, no. 1–3, pp. 43–69, 2000.
- [19] T. Strouboulis, K. Copps, and I. Babuška, "The generalized finite element method: an example of its implementation and illustration of its performance," *International Journal for Numerical Methods in Engineering*, vol. 47, no. 8, pp. 1401–1417, 2000.
- [20] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.
- [21] G. Meurant, "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 3, pp. 707–728, 1992.
- [22] Copyright (c), Thomas-Peter Fries, RWTH Aachen University, 2010



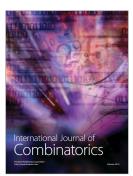






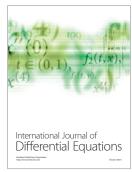




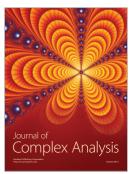


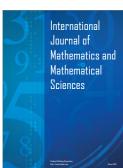


Submit your manuscripts at http://www.hindawi.com











Journal of **Discrete Mathematics** 

