COMPDYN 2019
7th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Crete, Greece, 24-26 June 2019

# DOMAIN DECOMPOSITION METHODS FOR CRACK GROWTH PROBLEMS USING XFEM

**Serafeim Bakalakos, Manolis Georgioudakis and Manolis Papadrakakis**

Institute of Structural Analysis & Antiseismic Research
School of Civil Engineering
National Technical University of Athens
GR 15780, Zografou Campus, Athens
e-mail: serafeim.bakalakos@outlook.com, geoem@mail.ntua.gr, mpapadra@centra.ntua.gr

**Keywords:** Crack Propagation, Domain Decomposition, FETI, XFEM.

**Abstract.** *The extended finite element method (XFEM) enriches the polynomial basis functions of standard finite elements with specialized non-smooth functions. The resulting approximation space can be used to solve problems with moving discontinuities, such as cracks, while avoiding the computational cost of remeshing. As the crack propagates, many artificial degrees of freedom are introduced near the crack tip, which can inflate the size of resulting linear system to be solved. In addition, the stiffness matrix of the cracked body may become ill-conditioned, causing slow convergence of iterative solvers. To overcome this, domain decomposition methods for solving the resulting linear systems of crack propagation problems is combined with XFEM to improve its performance. A suitable decomposition is proposed to avoid inter-subdomain boundaries near crack tip area. It is shown that choosing the proper FETI method, offers significant speedup compared to a direct solver that uses the common finite element solution techniques by means of Cholesky/LDL factorization, even if the execution is performed on a single-core systems.*

## 1 INTRODUCTION

The extended finite element method (XFEM) enriches the polynomial basis functions of standard finite elements with specialized non-smooth functions [1]. The resulting approximation space can be used to solve problems with moving discontinuities, such as cracks, while avoiding the computational cost of remeshing. Although the local enrichment strategy of XFEM is computationally efficient, it can rapidly increase the size of the resulting linear system in large scale problems. In addition, the stiffness matrix may become ill-conditioned, causing slow convergence of iterative solvers.

In this paper, the application of domain decomposition methods for solving the resulting linear systems is proposed. Among such methods we will focus to the family of finite element tearing and interconnecting (FETI) methods which were originally developed for the standard finite element method. In a FETI method, the finite element model is decomposed into several subdomains and the continuity at the boundary freedom degrees is enforced by applying Lagrange multipliers. These are the unknowns of an equivalent dual linear system, which can be solved by processing each subdomain independently. To speed up the solution process of each subdomain the standard preconditioners (e.g. preconditioned conjugate gradient method - PCG) are used. When FETI methods are combined with XFEM, the identification of rigid body motions of floating subdomains must be purely identified. Round-off errors in the computation of the rigid body motions can increase the PCG iterations that are required to solve the interface problem. Thus, a FETI method and can be combined with an effective mesh partitioning algorithm to ensure a decomposition into balanced subdomains that does not slow down the convergence.

A benchmark structure under fatigue crack growth is analyzed to demonstrate the capabilities of the proposed algorithm. A suitable decomposition is proposed to avoid inter-subdomain boundaries near crack tip area. It is shown that choosing the proper FETI method, offers significant speedup compared to a direct solver that uses the common finite element solution techniques by means of Cholesky/LDL factorization, even if the execution is performed on a single-core systems.

The paper is organized as follows: a brief description of modeling crack propagation with XFEM is given first, with special emphasis to fatigue crack growth. Subsequently, the FETI methods are discussed and their basic concepts are presented. Finally numerical tests are provided for illustrating the capabilities of the proposed methodology.

## 2 MODELING CRACK PROPAGATION USING XFEM

This paper focuses on the 2D mixed-mode linear elastic fracture mechanics (LEFM) formulation, where the size of cracked plastic zone is sufficiently small and it is embedded within an elastic zone around the crack tip. The inelastic behavior at the crack tip can cause a fatigue crack growth, meaning that the crack will grow under conditions of cyclic applied loading. The basic assumptions of LEFM in order to quantify crack growth around the crack tip in the presence of constant amplitude cyclic stress intensity, are given in the next sections.

### 2.1 XFEM basic ingredients

According to XFEM, special functions are added to the finite element approximation based on the partition of unity (PU) [2] and additional degrees of freedom are introduced to selected nodes of the finite element mesh near to discontinuities providing a higher level of accuracy. The quasi-static crack propagation simulations can be carried out without the need of remesh-

ing, modeling the domain with standard finite elements without explicitly meshing the crack surfaces.

Two basic types of enrichment functions are used: (i) the Heaviside function (step function) and (ii) the asymptotic crack-tip enrichment functions derived from LEFM [3]. The displacement field for a single "nominal" element can be expressed as a superposition of the standard $\vec{u}^{\text{std}}$, the crack-split $\vec{u}^{\text{H}}$ and the crack-tip $\vec{u}^{\text{tip}}$ fields as:

$$\vec{u}(\vec{x}) = \vec{u}^{\text{std}} + \vec{u}^{\text{enr}} = \vec{u}^{\text{std}} + \vec{u}^{\text{H}} + \vec{u}^{\text{tip}} \tag{1}$$

or more explicitly:

$$\vec{u}(\vec{x}) = \underbrace{\sum_{j=1}^{n} N_j(\vec{x})\vec{u}_j}_{\vec{u}^{\text{std}}} + \underbrace{\sum_{h=1}^{n_h} N_h(\vec{x})H(\vec{x})\vec{a}_h + \sum_{k=1}^{n_t} N_k(\vec{x})\Big(\sum_{l=1}^{n_f} F_l(\vec{x})\vec{b}_k^l\Big)}_{\vec{u}^{\text{enr}}} \tag{2}$$

The first contributing part ($\vec{u}^{\text{std}}$) on the right-hand side of Eq. (2) corresponds to the classical finite element approximation of the displacement field, while the second part ($\vec{u}^{\text{enr}}$) refers to the enrichment approximation which takes into account the existence of discontinuities. This second contributing part utilizes additional degrees of freedom to facilitate modeling of the discontinuous field, without modeling it explicitly. $n$ is the number of nodes in each finite element model with standard degrees of freedom $\vec{u}_j$ and shape functions $N_j(\vec{x})$, $n_h$ is the number of nodes in the elements containing the crack face (but not the crack tip), $\vec{a}_h$ is the vector of additional degrees of freedom for modeling crack faces defined by the Heaviside function $H(\vec{x})$, $n_t$ is the number of nodes associated with the crack tip in its influence domain and $\vec{b}_k^l$ is the vector of additional degrees of freedom for modeling crack tips.

Furthermore, $F_l(\vec{x})$ are the crack-tip enrichment functions, given by:

$$\{F_l(r,\theta)\}_{l=1}^4 = \left\{\sqrt{r}\sin(\frac{\theta}{2}); \sqrt{r}\cos(\frac{\theta}{2}); \sqrt{r}\sin(\frac{\theta}{2})\sin(\theta); \sqrt{r}\cos(\frac{\theta}{2})\sin(\theta)\right\} \tag{3}$$

The elements which are cut entirely by the crack, are enriched with the Heaviside (step) function $H$. The Heaviside function is a discontinuous function across the crack surface and is constant on each side of the crack. Splitting the domain by the crack causes a displacement jump and the Heaviside function provides the desired behavior to approximate the real displacement field.

### 2.2 Crack representation by level sets

In association to XFEM, the level set method (LSM) [4] is adopted to describe the geometry of the crack. The LSM was originally proposed for tracking moving interfaces, and became a key ingredient of XFEM for describing complicated geometrical interfaces by computing their motion on a fixed finite element mesh. Thus, level set functions offer an elegant way of modeling cracks which take the form of signed distance functions. In order to fully characterize a crack discontinuity, two level set functions are defined as signed distance functions: (i) a normal level set function $\phi$ and (ii) a tangent level set function $\psi$. To define the signed distance functions, the interface $\Gamma$ can be regarded as the zero level contour of the level set function $\phi(\vec{x})$. The distance $d$ from a point $\vec{x}$ to the interface $\Gamma$ is defined as:

$$d = \|\vec{x} - \vec{x}_\Gamma\| \tag{4}$$

where $\vec{x}_\Gamma$ is the normal projection of $\vec{x}$ on $\Gamma$. Mathematically, the signed distance function $\phi(\vec{x})$ can be defined as,

$$\phi(\vec{x}) = \min \|\vec{x} - \vec{x}_\Gamma\| \, \text{sign}(\vec{n_1} \cdot (\vec{x} - \vec{x}_\Gamma)) \tag{5}$$

where $\vec{n_1}$ is the unit normal vector to the crack at point $\vec{x_\Gamma}$ (see Fig. 1).

For the case of an edge crack, to construct the two level set functions, the approach proposed by Stolarska et al. [5] is adopted. The crack tip segment is extended to "meet" the boundary of the domain and the normal level set function $\phi$ is then defined using Eq. (5), from the original crack segment $\Gamma$ (at crack tip $j$) and the virtual segments (extensions) ($\phi = 0$ and $\psi = 0$) (see Fig. 1). Similar, the tangent level set function $\psi$ as the signed distance to the plane including

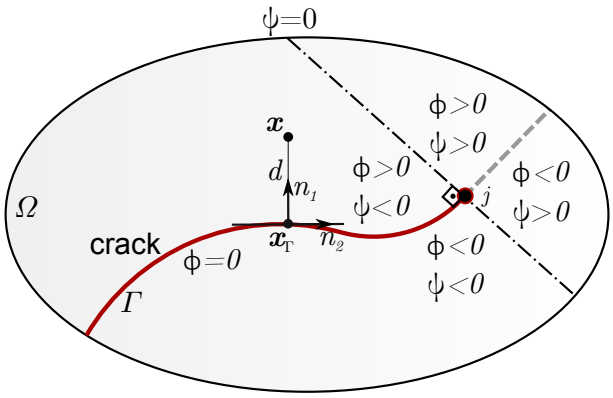



Figure 1: Signed distance definition, and corresponding level set functions $\phi$ and $\psi$ in crack tip point $\vec{x}_\Gamma$.

the crack front and perpendicular to the crack surface, is defined as follows:

$$\psi(\vec{x}) = \min \|\vec{x} - \vec{x}_\Gamma\| \, \text{sign}(\vec{n_2} \cdot (\vec{x} - \vec{x}_\Gamma)) \tag{6}$$

where $\vec{n_2}$ is the unit vector tangent to the crack at its tip. The crack can now be represented by the two level set functions $\phi$ and $\psi$ such that $\phi = 0$ for the crack surface $\Gamma$. Consequently, the crack tip $j$ is identified on an intersection of normal and tangent zero level set functions, i.e. intersection of $\phi = 0$ and $\psi = 0$. In the remaining domain, $\phi$ has a positive value above the crack, while $\psi$ has a positive value to the right of the unit normal vector $\vec{n_1}$ at the crack tip $j$. The crack tip level set $\psi$ is generally assumed to be orthogonal to $\phi$, i.e. $\phi\psi = 0$. It is also assumed that once a part of the crack has formed, that part will no longer change its shape or move.

### 2.3 Computation of the crack growth direction

The accuracy and reliability of the method primarily on the continuity and the determination of the crack path. Among the existing crack growth criteria, the maximum hoop stress criterion [7], is adopted. According to this criterion: (i) the crack initiation will occur when the maximum hoop stress reaches a critical value and (ii) the crack will grow along direction $\theta_{cr}$ in which circumferential stress $\sigma_{\theta\theta}$ is maximum.

The circumferential stress $\sigma_{\theta\theta}$ in a polar coordinate system $(r, \theta)$ (see Fig. 2) along the direction of the crack propagation is a principal stress. Hence, the critical crack propagation direction $\theta_{cr}$ is determined by setting the shear stress $\sigma_{r\theta}$ equal to zero, i.e.:

$$\sigma_{r\theta} = \frac{1}{2\sqrt{(2\pi r)}} cos\frac{\theta_{cr}}{2}\Big(K_I sin\theta_{cr} + K_{II}(3cos\theta_{cr} - 1)\Big) = 0 \tag{7}$$

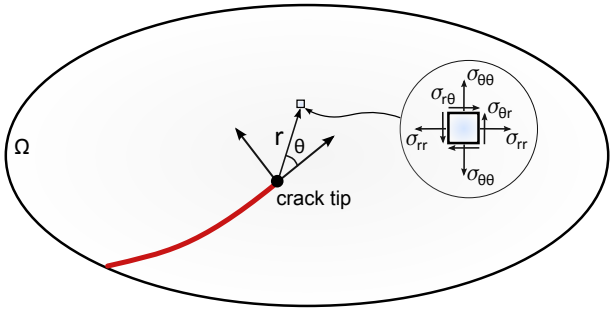



Figure 2: Polar coordinates in the crack tip coordinate system.

leading to the following expression of the crack growth direction $\theta_c$, in terms of local crack tip coordinate system:

$$\theta_c = \theta_{cr} = 2arctan\frac{1}{4}\Big(\frac{K_I}{K_{II}} \pm \sqrt{\frac{K_I^2}{K_{II}} + 8}\Big) \tag{8}$$

### 2.4 Computation of fatigue cycles

Fatigue crack growth is quantified using Paris law [8], which is originally proposed for single mode deformation cases, and correlates crack propagation rate under fatigue loading with the stress intensity factors (SIFs). For the case of mixed-mode loading, a modified Paris law is defined using the effective stress intensity factor range $\Delta K_{\text{eff}} = K_{\max} - K_{\min}$. For a certain fatigue loading level, where the crack grows by length $\Delta a$ in $\Delta N_c$ cycles, Paris law reads:

$$\frac{\Delta a}{\Delta N_c} \approx \frac{da}{dN_c} = C(\Delta K_{\text{eff}})^m \tag{9}$$

where $C$ and $m$ are empirical material constants. $m$ is often called as the *Paris exponent* and is typically defined in the range of $3 \div 4$ for common steel and aluminium alloys. Eq. (9) represents a linear relationship between $log(\Delta K_{\text{eff}})$ and $log(\frac{da}{dN_c})$ which is used to describe fatigue crack propagation in Region II (Fig. 3). For calculating the effective mixed-mode stress

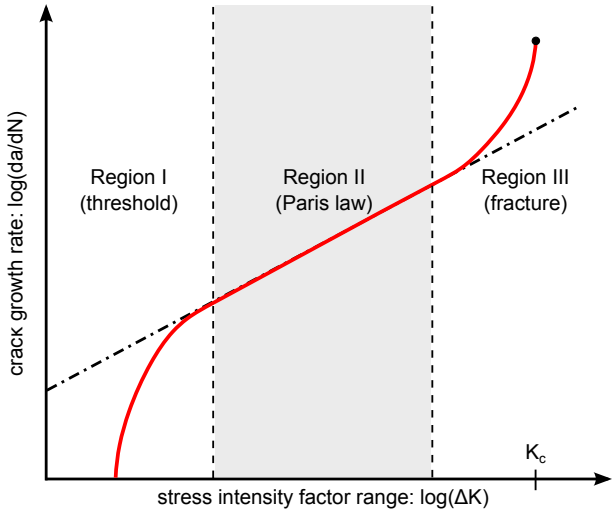



Figure 3: Logarithmic crack growth rate and effective region of Paris law.

intensity factor $\Delta K_{\text{eff}}$, the energy release rate model is adopted, leading to:

$$\Delta K_{\text{eff}} = \sqrt{\Delta K_I^2 + \Delta K_{II}^2} \tag{10}$$

Subsequently, the number of the corresponding cycles for a crack growth $\Delta a$ is computed according to [3]:

$$\Delta N_c = \frac{\Delta a}{C(\Delta K_{\text{eff}})^m} \tag{11}$$

Similar to strength of materials theory where the calculated stress is compared with an allowable stress defining the material strength, LEFM assumes that unstable fracture occurs when SIF $K$ reaches a critical value $K_c$, called *fracture toughness*, which represents the capacity of a material to withstand a given stress field at the crack tip and to resist into the progressive tensile crack extension. In other words, $K_c$ is a material constant and is used as a threshold value for SIFs.

## 3 FETI Methods

Domain decomposition methods (DDM) split the domain into smaller subdomains which can be processed independently. Among such methods, the finite element tearing and in interconnecting (FETI) DDMs exhibit the highest scalability, when executed in distributed computing environments. The first FETI method (FETI-1) was introduced by Farhat et al. [11], and is the first implementation of FETI methods which uses Lagrange multipliers to describe the forces between the subdomains. Later, dual-primal FETI (FETI-DP) was introduced by Farhat et al. [12], as a simpler and in many cases more efficient, alternative.

### 3.1 FETI-1

In FETI-1 method, the domain is divided into disconnected subdomains as in Fig. 4 and the continuity between them is retained by enforcing equal displacements for instances of the same boundary DOF (degree of freedom).

$$\left[ \begin{array}{cc} 1 & -1 \end{array} \right] \left[ \begin{array}{c} u_k^{(1)} \\ u_k^{(2)} \end{array} \right] = 0 \tag{12}$$

By gathering the 1, -1 and 0 coefficients in a signed boolean matrix $\boldsymbol{B}^s$ for all boundary DOFs of a subdomain:

$$\sum_{s=1}^{N_s} \boldsymbol{B}^s \boldsymbol{u}^s = \boldsymbol{0} \tag{13}$$

To solve the initial linear system in the presence of these constraints, we apply Lagrange multipliers at boundary DOFs. These are dual quantities and can be viewed as forces, while displacements are primal quantities. Using the superscript e to denote the aggregation of all subdomain matrices/vectors into an expanded matrix/vector, the resulting linear system is written as:

$$\left[ \begin{array}{cc} \boldsymbol{K}^e & (\boldsymbol{B}^e)^T \\ \boldsymbol{B}^e & \boldsymbol{0} \end{array} \right] \left[ \begin{array}{c} \boldsymbol{u}^e \\ \boldsymbol{\lambda} \end{array} \right] = \left[ \begin{array}{c} \boldsymbol{f}^e \\ \boldsymbol{0} \end{array} \right] \tag{14}$$

A subdomain of FETI-1 can be floating, meaning there are not enough constrained DOFs and its stiffness matrix $\boldsymbol{K}^s$ is singular. To overcome this, we use a generalized inverse $(\boldsymbol{K}^s)^+$

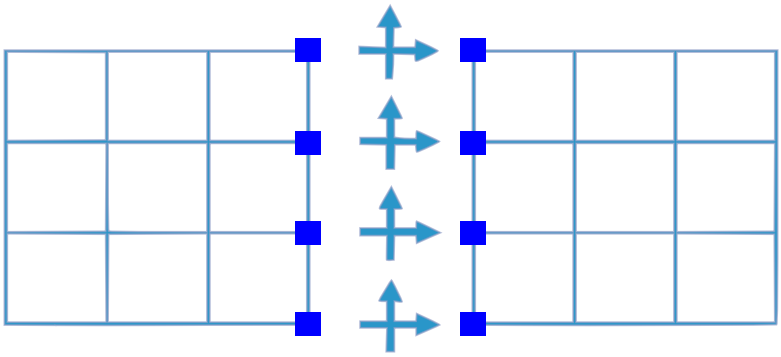


Figure 4: Langrange multipliers applied to boundary DOFs of subdomains.

and a normalized basis for its nullspace $\boldsymbol{R}^s$, which corresponds to the rigid body motions of the floating subdomain.

$$(\boldsymbol{K}^s)^+ = \begin{bmatrix} (\boldsymbol{K}^s_{11})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{15}$$

$$\boldsymbol{R}^s = \begin{bmatrix} (\boldsymbol{K}^s_{11})^{-1}\boldsymbol{K}^s_{12} \\ \boldsymbol{I} \end{bmatrix} \tag{16}$$

where the subscripts 1/2 denote linearly independent/dependent rows and columns, respectively. If the subdomain has sufficient constraints then $\boldsymbol{R}^s = \emptyset$ and $(\boldsymbol{K}^s)^{-1}$. The interface problem of FETI-1 is as follows:

$$\begin{bmatrix} \boldsymbol{F}_I & -\boldsymbol{G}_I \\ -\boldsymbol{G}_I^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ -\boldsymbol{e} \end{bmatrix} \tag{17}$$

where $\boldsymbol{a}$ expresses linear combinations of the normalized rigid body motions in $\boldsymbol{R}$, $\boldsymbol{F}_I$ is the flexibility matrix, $\boldsymbol{G}$ aggregates the rigid body motions, $\boldsymbol{d}$ are the displacements due to applied forces (boundary conditions and loading) and $\boldsymbol{e}$ is the work of rigid body motions due to these applied forces.

$$\boldsymbol{F}_I = \sum_{s=1}^{N_s} \boldsymbol{F}_I^s = \sum_{s=1}^{N_s} \boldsymbol{B}^s (\boldsymbol{K}^s)^+ (\boldsymbol{B}^s)^T \tag{18}$$

$$\boldsymbol{d} = \sum_{s=1}^{d_s} \boldsymbol{F}_I^s = \sum_{s=1}^{N_s} \boldsymbol{B}^s (\boldsymbol{K}^s)^+ \boldsymbol{f}^s \tag{19}$$

$$\boldsymbol{G}_I = \begin{bmatrix} \boldsymbol{B}^{(1)}\boldsymbol{R}^{(1)} & \boldsymbol{B}^{(2)}\boldsymbol{R}^{(2)} & \dots & \boldsymbol{B}^{(N_s)}\boldsymbol{R}^{(N_s)} \end{bmatrix} \tag{20}$$

The linear system of Eq. (17) is not symmetric positive definite. To solve it using the preconditioned conjugate gradient (PCG) method, we must project it into another space using a projection matrix $\boldsymbol{P}$ and then solve the following system:

$$\boldsymbol{P}^T \boldsymbol{F}_I \boldsymbol{P} \bar{\boldsymbol{\lambda}} = \boldsymbol{P}^T (\boldsymbol{d} - \boldsymbol{F}_I \boldsymbol{P} \boldsymbol{\lambda}_0) \tag{21}$$

where

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{G}_I (\boldsymbol{G}_I^T \boldsymbol{G}_I)^{-1} \boldsymbol{G}_I^T \tag{22}$$

$$\boldsymbol{\lambda}_0 = \boldsymbol{G}_I (\boldsymbol{G}_I^T \boldsymbol{G}_I)^{-1} \boldsymbol{e} \tag{23}$$

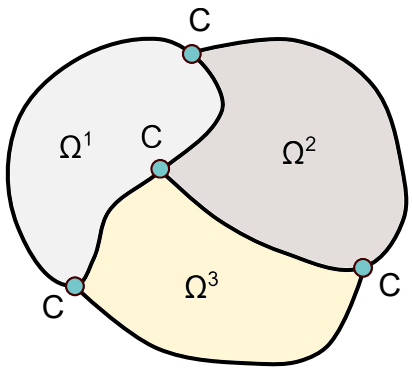



Figure 5: Definition of corner nodes (C).

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{P}\bar{\boldsymbol{\lambda}} \tag{24}$$

$$\boldsymbol{a} = (\boldsymbol{G}_I^T \boldsymbol{G}_I)^{-1} \boldsymbol{G}_I^T (\boldsymbol{F}_I \boldsymbol{\lambda} - \boldsymbol{d}) \tag{25}$$

The coarse problem of FETI-1 is expressed in the linear system $\boldsymbol{G}_I^T \boldsymbol{G}_I x = y$ of Eq. (22) and it helps speed up convergence by globally distributing the error at each PCG iteration. Finally we calculate the displacements depending on if the subdomain is floating or not:

$$\boldsymbol{u}^s = (\boldsymbol{K}^s)^+ (\boldsymbol{f}^s - (\boldsymbol{B}^s)^T \boldsymbol{\lambda}) + \boldsymbol{R}^s \boldsymbol{a}^s \implies \boldsymbol{u}^s = (\boldsymbol{K}^s)^{-1} (\boldsymbol{f}^s - (\boldsymbol{B}^s)^T \boldsymbol{\lambda}) \tag{26}$$

### 3.2 FETI-DP

The FETI-DP method is based on a dual-primary formulation of the problem to be solved. The main difference between the FETI-DP method and the FETI-1 method is that in FETI-DP, in every iteration of the PCG method, the compatibility of displacements is imposed to a small subset of primal *corner* DOFs, while the compatibility of displacements to the remaining boundary DOFs is only achieved after convergence.

We define *corner* nodes and corner or primal DOFs (Fig. 5). All other DOFs are called *remainder* and are further separated into boundary-remainder and boundary-internal. We then perform static condensation of the remainder DOFs in each subdomain and assemble the resulting matrices/vectors into global ones:

$$\boldsymbol{K}_{cc}^* = \sum_{s=1}^{N_s} (\boldsymbol{L}_c^s)^T \left( \boldsymbol{K}_{cc}^s - (\boldsymbol{K}_{rc}^s)^T (\boldsymbol{K}_{rr}^s)^{-1} \boldsymbol{K}_{rc}^s \right) \boldsymbol{L}_c^s \tag{27}$$

$$\boldsymbol{f}_c^* = \sum_{s=1}^{N_s} (\boldsymbol{L}_c^s)^T \left( \boldsymbol{f}_c^s - (\boldsymbol{K}_{rc}^s)^T (\boldsymbol{K}_{rr}^s)^{-1} \boldsymbol{f}_r^s \right) \tag{28}$$

where the subscripts r and c denote the rows/columns corresponding to remainder and corner DOFs respectively. $\boldsymbol{L}_c$ are unsigned boolean matrices (entries are 0 or 1) mapping global corner displacements to subdomain corner displacements. With an appropriate selection of corner nodes, the matrices $\boldsymbol{K}_{rr}$ is invertible, thus there are no floating subdomains. For the signed boolean matrices $\boldsymbol{B}_r$, we will use the columns of $\boldsymbol{B}$ that correspond to remainder DOFs.

The interface problem of FETI-DP can be written as follows:

$$\left(\boldsymbol{F}_{Irr} + \boldsymbol{F}_{Irc}(\boldsymbol{K}_{cc})^{-1}(\boldsymbol{F}_{Irr})^{T}\right)\boldsymbol{\lambda} = \boldsymbol{d}_r - \boldsymbol{F}_{Irc}(\boldsymbol{K}_{cc})^{-1}\boldsymbol{f}_c^* \tag{29}$$

where $\boldsymbol{d}_r$ are the displacements due to boundary conditions and loading and $\boldsymbol{F}_{Irr}$ is the flexibility matrix. The linear system of Eq. (29) can be solved using the PCG method.

$$\boldsymbol{d}_r = \sum_{s=1}^{N_s} \boldsymbol{B}_r^s(\boldsymbol{K}_{rr}^s)^{-1}\boldsymbol{f}_r^s \tag{30}$$

$$\boldsymbol{F}_{Irr} = \sum_{s=1}^{N_s} \boldsymbol{B}_r^s(\boldsymbol{K}_{rr}^s)^{-1}(\boldsymbol{B}_r^s)^{T} \tag{31}$$

$$\boldsymbol{F}_{Irc} = \sum_{s=1}^{N_s} \boldsymbol{B}_r^s(\boldsymbol{K}_{rc}^s)^{-1}\boldsymbol{K}_{rc}^s\boldsymbol{L}_c^s \tag{32}$$

The coarse problem of FETI-DP is expressed in the linear system $\boldsymbol{K}_{cc} * x = y$ of Eq. (29) and it helps speed up convergence by globally distributing the error at each PCG iteration. After solving the interface problem the displacements can be calculated using:

$$\boldsymbol{u}_c = \left(\boldsymbol{K}_{cc}^*\right)^{-1}\left(\boldsymbol{f}_c^* + (\boldsymbol{F}_{Irc})^{T}\boldsymbol{\lambda}\right) \tag{33}$$

$$\boldsymbol{u}_r^s = (\boldsymbol{K}_{rr}^s)^{-1}\left(\boldsymbol{f}_r^s - \left(\boldsymbol{B}_r^s\right)^{T}\boldsymbol{\lambda} - \boldsymbol{K}_{rc}^s\boldsymbol{L}_c^s\boldsymbol{u}_c\right) \tag{34}$$

### 3.3 Preconditioning

Both FETI-1 and FETI-DP can use the same preconditioners to speed up the convergence of PCG when solving the interface problem. The idea is to approximate the inverse of the flexibility matrix, which is defined for the boundary DOFs. The most common preconditioners used are Dirichlet and lumped ones.

#### Dirichlet preconditioner

A Dirichlet preconditioner is defined as follows:

$$\boldsymbol{F}_I^{-1} = \sum_{s=1}^{N_s} \boldsymbol{B}_{pb}^s\boldsymbol{S}^s(\boldsymbol{B}_{pb}^s)^{T} \tag{35}$$

where $\boldsymbol{S}^s$ is the Schur complement of internal dofs (internal-remainder in FETI-DP)

$$\boldsymbol{S}^s = \boldsymbol{K}_{bb}^s - (\boldsymbol{K}_{ib}^s)^{T}(\boldsymbol{K}_{ii}^s)^{-1}\boldsymbol{K}_{ib}^s \tag{36}$$

$\boldsymbol{B}_{pb}$ are sparse matrices that perform mapping (like $\boldsymbol{B}$) and scaling, e.g. for a homogeneous stiffness distribution among subdomains:

$$\boldsymbol{B}_{pb}^s = \boldsymbol{B}_b^s(\boldsymbol{M}_b^s)^{-1} \tag{37}$$

where $\boldsymbol{B}_b$ are the columns of $\boldsymbol{B}$ corresponding to boundary DOFs (boundary remainder in FETI-DP) and $\boldsymbol{M}_b$ is a diagonal matrix whose diagonal entries are the multiplicity of the corresponding DOFs, i.e. how many subdomains have instances of the DOFs.

### Lumped preconditioner

A lumped preconditioner is defined as follows:

$$\boldsymbol{F}_I^{-1} = \sum_{s=1}^{N_s} \boldsymbol{B}_{pb}^s \boldsymbol{K}_{bb}^s (\boldsymbol{B}_{pb}^s)^T \tag{38}$$

In general, a lumped preconditioner is faster to compute and implement, but does not speed up the convergence of PCG as much as Dirichlet.

### Diagonal Dirichlet preconditioner

A Diagonal Dirichlet preconditioner is defined as follows:

$$\boldsymbol{F}_I^{-1} = \sum_{s=1}^{N_s} \boldsymbol{B}_{pb}^s \hat{\boldsymbol{S}}^s (\boldsymbol{B}_{pb}^s)^T \tag{39}$$

where $\hat{\boldsymbol{S}}^s$ is an approximation of the Schur complement of internal dofs (internal-remainder in FETI-DP).

$$\hat{\boldsymbol{S}}^s = \boldsymbol{K}_{bb}^s - (\boldsymbol{K}_{ib}^s)^T (\boldsymbol{D}_{ii}^s)^{-1} \boldsymbol{K}_{ib}^s \tag{40}$$

where $\boldsymbol{D}_{ii}^s$ is the diagonal of $\boldsymbol{K}_{ii}^s$. In general, the computational cost of computing and implementing a diagonal Dirichlet preconditioner, as well as the resulting reduction of PCG iterations falls between that of Dirichlet and lumped preconditioner.

## 4 Combining XFEM with FETI

Using FETI methods to solve the linear systems resulting from XFEM during crack propagation can speed up the process. FETI methods are inherently parallel and can be executed on distributed computing environments concurrently. Compared to other domain decomposition methods, FETI methods scale extremely well with the number of processors in a network environment.

Even on a single-core system, FETI methods can result in a speedup over direct solvers. Due to domain decomposition, the stiffness matrix of each subdomain will have lower bandwidth than the stiffness matrix of the global domain. Non zero entries of the stiffness matrix correspond to DOFs of the same element. These DOFs are numbered closer to each other in a subdomain that has fewer DOFs than the whole model. In contrast, direct solvers rely on factorizing the stiffness matrix and their main bottleneck is its bandwidth. The more subdomains a domain decomposition solvers uses, the smaller the bandwidth of the resulting stiffness matrices.

To use FETI-1 for XFEM one needs to correctly identify the rigid body motions of the floating subdomains defined in Eq. (16). Round-off errors in the computation of the rigid body motions can increase PCG iterations required to solve the interface problem. FETI-DP on the other hand requires an appropriate selection of corner nodes to avoid singular matrices. Both methods can use the standard preconditioners described in Section 3.3. They should also be combined with an effective mesh partitioning algorithm to ensure a decomposition into balanced subdomains that does not slow down convergence.

## 5 NUMERICAL EXAMPLE

The test case deals with a fatigue crack growth in a double cantilever beam (DCB) [10]. The geometry of the DCB are $h = 3.94$ in and $L = 3h = 11.82$ in, as shown in Fig. 6. The thickness of the beam is equal to 1 in. The first crack segment is horizontal at half the beam's height and has length $a = 3.95$ in. An extra crack segment is added with length $da = 0.5$ in with an inclination angle $d\theta = 5.71^o$, to avoid pure Mode I propagation. The right side of the DCB is fixed, while a cyclic load $P = 197$ lbs is applied on the top left and bottom left corners. The material properties are $E = 3 \cdot 10^7$ psi and $v = 0.3$, while plane stress conditions are considered.

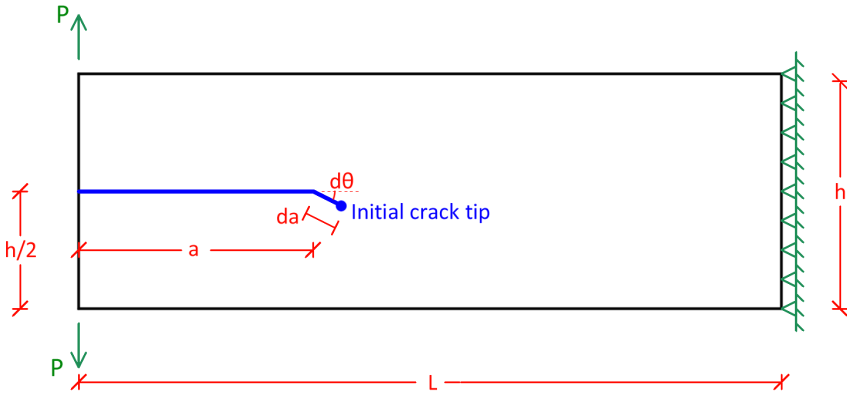



Figure 6: Geometry characteristics of the double cantilever beam.

The beam is discretized with standard QUAD4 finite elements, forming a uniform mesh. XFEM analysis is performed at 7 propagation steps, with topological tip enrichment. All analyses were conducted on a single-core system (Intel Core i7-6700 @3.40GHz CPU / 8 GB RAM) using three different types of solvers: i) a direct solver (with LDL factorization), ii) a FETI-1 solver and iii) a FETI-DP solver. In all analyses approximate minimum degree (AMD) reordering was used to reduce the bandwidth of the resulting stiffness matrices. Fig. 7 shows the crack path at the last step of the XFEM analysis.

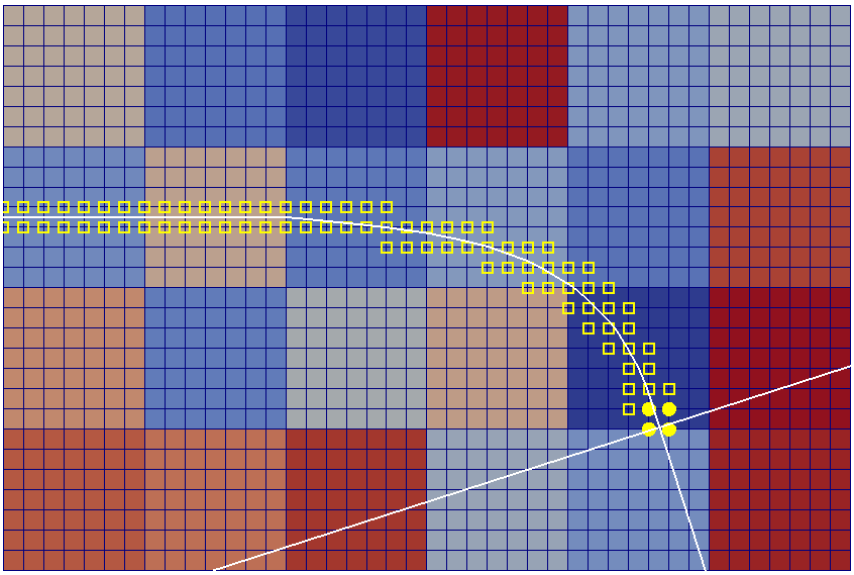
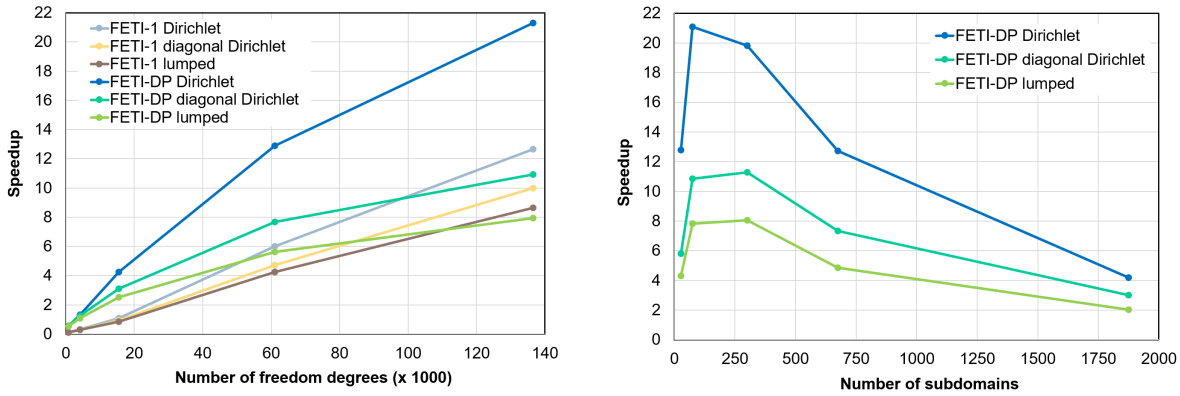


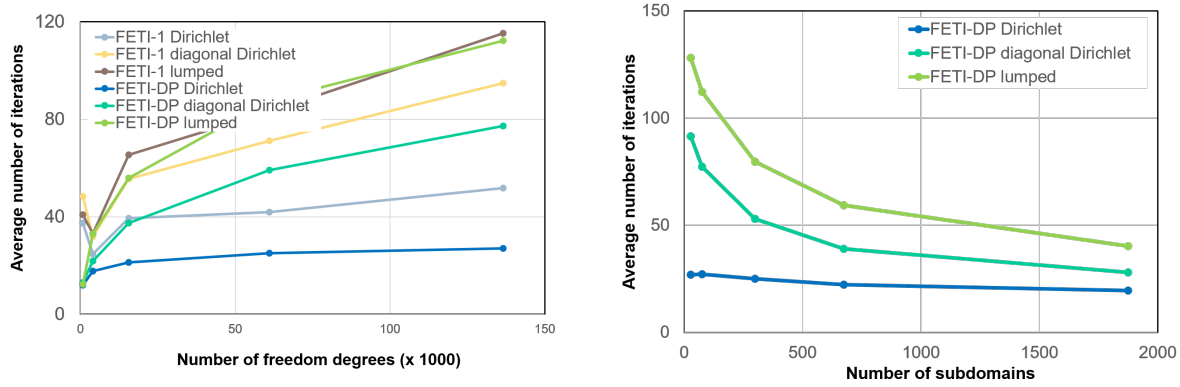
Figure 7: Crack propagation in a double cantilever beam.

The speedup of FETI-1 and FETI-DP solver over the direct solver is depicted in Fig. 8 versus the number of DOFs and the number of subdomains. It is shown that FETI-DP exhibits excellent performance compared to other two solvers. More specific, it offers significant speedup (up to 22X) compared to the direct solver for all meshes examined. Furthmore, the Dirichlet preconditioner outperforms lumped and diagonal Dirichlet (up to a factor of 10X). The average number of PCG iterations is depicted in Fig. 9 versus the number of DOFs and the number of subdomains. The number of required PCG iterations remained low throughout the crack propagation analysis and was not affected by the ill-conditioned due to the enriched DOFs, while the optimal number of subdomains depends on the mesh size. On the other hand, FETI-1 required more PCG iterations to converge, as more enriched DOFs are added to the system. Although FETI-1 is inferior to FETI-DP, it still results in a speedup of the solution phase compared to the direct solver (up to 12X). Again Dirichlet precondition outperforms the other two in terms of average number of iterations required for the convergence.




(a) (b)

Figure 8: Speedup over direct solver versus number of (a) dofs and (b) subdomains.




(a) (b)

Figure 9: Average number of PCG iterations vs number of (a) dofs and (b) subdomains.

## 6 CONCLUSIONS

- FETI-1 and FETI-DP can be used for the solution of the linear systems resulting from 2D crack propagation in the XFEM framework.
- They offer significant speedup compared to a direct solver that uses Cholesky/LDL factorization and reordering, even when run on a single-core system.
- Potentially they can be run on distributed computing environments to further reduce the computational time.
- FETI-DP is simpler to use than FETI-1 and more efficient, since the number of PCG iterations is marginally increased by the ill-conditioning caused by enriched dofs.
- For both methods, Dirichlet preconditioner performs the best, while lumped preconditioner performs the worst.

## REFERENCES

[1] Moes, N., Dolbow, J. and Belytschko, T. A Finite Element Method for Crack Growth without Remeshing. *International Journal for Numerical Methods in Engineering*, **46**, 131–150, 1999.

[2] Babuška, I. and Melenk, J. M., The Partition of Unity Method. *International Journal for Numerical Methods in Engineering*, **40**, 727–758, 1997.

[3] Anderson, T. L., *Fracture Mechanics: Fundamentals and Applications*. 3rd Edition, CRC Press, 2004.

[4] Osher, S. and Sethian, J. A, The Partition of Unity Method. *Journal of Computational Physics*, **79**, 12–49, 1988.

[5] Stolarska, M. and Chopp, D. L. and Moës, N. and Belytschko, T., Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, **51**, 943–960, 2001.

[6] Rice, J. R., A path independent integral and the approximate analysis of strain concentrations by notches and cracks. *Journal of Applied Mechanics*, **35**, 379–386, 1968.

[7] Erdogan, F. and Sih, G. C., On the Crack Extension in Plates Under Plane Loading and Transverse Shear. *Journal of Fluids Engineering*, **51**, 519–525, 1963.

[8] Paris, P.C. and Gomez, M.P. and Anderson, W.E., On the Crack Extension in Plates Under Plane Loading and Transverse Shear. *The Trend in Engineering*, **13**, 9–14, 1961.

[9] Yau, J. F. and Wang, S. S. and Corten, H. T., A Mixed-Mode Crack Analysis of Isotropic Solids Using Conservation Laws of Elasticity. *Journal of Applied Mechanics*, **47**, 335–341, 1980.

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

[11] Farhat, C. and Roux, F.-X., A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, **32**, 1205–1227, 1991.

[12] Farhat, C., Lesoinne, M. and Pierson, K., A scalable dualprimal domain decomposition method. *Numerical Linear Algebra with Applications*, **32**, 687-714, 2000.