Domain Decomposition and Parallel Direct Solvers as an Adaptive Multiscale Strategy for Damage Simulation in Quasi-Brittle Materials

Frank P.X. Everdij[1], Oriol Lloberas-Valls[2], Angelo Simone[1], Daniel J. Rixen[3], and Lambertus J. Sluys[1]

1 Introduction

Understanding failure processes of heterogeneous materials is an active research field in computational mechanics. The failure analysis of quasi-brittle materials such as concrete is a topic of particular interest in civil engineering. Failure in quasi-brittle materials is characterized by the initial formation of cracks at a microscopic level followed by their coalescence into macroscopic cracks leading to weakening and fracture. Because the fracturing process of these materials occurs at different length scales, care must be taken to provide an accurate description which accounts for all the relevant mechanical processes while maintaining acceptable computation costs. With this in mind, we propose a multiscale approach capable of switching between different spatial discretizations and material representations depending on the local mechanical behaviour.

In this contribution, we present a non-local damage finite element analysis of a wedge-split test used to evaluate fracture properties in concrete-like materials. We apply the classical FETI framework (Farhat and Roux [1991]) to a non-linear gradient-enhanced damage (GD) model (Peerlings et al. [1996]) using both iterative and direct solvers to the interface problem as well as using a direct solver for the entire set of equations of the fully dual assembled system.

[1] Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands, F.P.X.Everdij@tudelft.nl · [2] International Center for Numerical Methods in Engineering (CIMNE), Campus Nord UPC, Edifici C-1, C/Jordi Girona 1-3, 08034 Barcelona, Spain · [3] Faculty of Mechanical Engineering, Technische Universität München, Boltzmannstrasse 15, 85748 Garching, Germany

2 Framework

2.1 Gradient-enhanced damage model

The gradient-enhanced damage model by Peerlings et al. [1996] is employed to model concrete failure. The GD model is non-local: it consists of a coupled set of differential equations involving the modified Helmholtz equation for the non-local equivalent strain and the classical quasistatic equilibrium equations. Damage evolution is highly non-linear, requiring the use of a loop control dividing the total load into small steps with an iterative Newton-Raphson (NR) scheme for each step to assure equilibrium.

The damage parameter ω, which modifies the stress–strain relation according to

$$\boldsymbol{\sigma} = (1-\omega)\mathsf{D}^e : \boldsymbol{\varepsilon}\,, \tag{1}$$

varies from 0 for undamaged to 1 for fully damaged material. Its evolution,

$$\omega(\kappa) = \begin{cases} 0 & \kappa \leq \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa}\left(1 - \alpha\left(1 - e^{-\beta(\kappa-\kappa_0)}\right)\right) & \kappa > \kappa_0 \end{cases}, \tag{2}$$

is a function of the history parameter κ which is defined as the maximum value ever attained by the nonlocal equivalent strain. In the above equations, D^e is the elasticity fourth-order tensor, $\boldsymbol{\sigma}$ is the second-order stress tensor, $\boldsymbol{\varepsilon}$ is the second order strain tensor, κ_0, α and β are parameters governing the shape of the damage evolution law.

The underlying damage formalism results in an asymmetric stiffness matrix. To solve the set of equations, a solver supporting asymmetry, both in direct and iterative approaches, is required.

2.2 Multiscale domain decomposition

The key to solving the discrete system of equations in a reasonable amount of time is to use two different representations of the problem under examination. One numerical model has a fine mesh with a detailed representation of the mesostructure of the material. The other numerical model has a coarse mesh with homogenized material properties which have been determined to approximate the response of the 'fine' model in the linear regime. Both numerical models have been decomposed into a fixed amount of domains. Each domain in the 'fine' model has a corresponding domain in the 'coarse' model matching its shape.

The calculation starts with the 'coarse' numerical model for all domains. In each step and for each domain, a check for the condition of onset of non-linearity is performed. For every node, the non-local equivalent strain

difference is calculated from the displacement field of the current and two previous steps. Onset of non-linearity occurs if for a single node the strain difference exceeds a chosen damage initiation threshold value κ_0. The domains for which this condition is met are subsequently replaced by domains with the fine scale mesh. To preserve continuity of the displacements and forces, a boundary value problem is solved for each replaced domain followed by a global relaxation step.

Computing the strain difference for the onset of the non-linearity condition is a choice that should match the nature of the formation of non-linearities. For tensile test calculations and the gradient-enhanced damage model, our current choice yields satisfactory results (Lloberas-Valls et al. [2012a]).

2.3 Classical FETI method

In order to solve the multiscale system with a mixture of coarse and fine meshes for each domain, the classical FETI method (Farhat and Roux [1991]) is used. Lagrange multipliers ensure continuity of the solution field between interface nodes of adjacent domains. Linear multipoint constraints and full-collocation are used for fine mesh interface nodes which do not have a corresponding coarse mesh node on the adjoining domain (Lloberas-Valls et al. [2012b]).

Boundary conditions are also included by means of Lagrange multipliers, thus implying that all domains in this framework are floating. This method is known as the Total-FETI method (Dostál et al. [2006]). Rigid body motion vectors are constructed to enforce compatibility between domains. To solve the local equations for each domain, we use QR factorization of the domain stiffness matrix which can be stored for later use in computing the Lagrange multipliers by means of either the iterative or direct solve of the global interface problem as shown in Lloberas-Valls et al. [2011, 2012a].

3 Numerical computation

3.1 Model

We use a two-dimensional model of a wedge split specimen for the quasistatic damage simulation of the heterogeneous sample of concrete shown in Fig. 1.

For the multiscale framework we use two different meshes: a homogeneous mesh consisting of quadrilateral elements with four integration points for the coarse domains, and a heterogeneous mesh with triangular elements and one integration point for fine scale domains. Both meshes are shown in Fig. 2. The

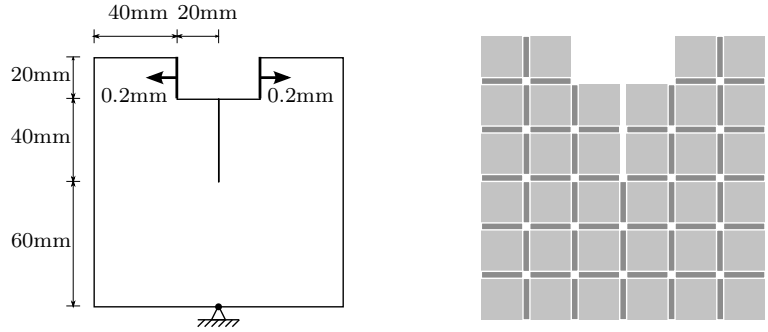

Fig. 1 Dimensions and domain decomposition of the wedge split model test. The interface is represented in dark-grey.

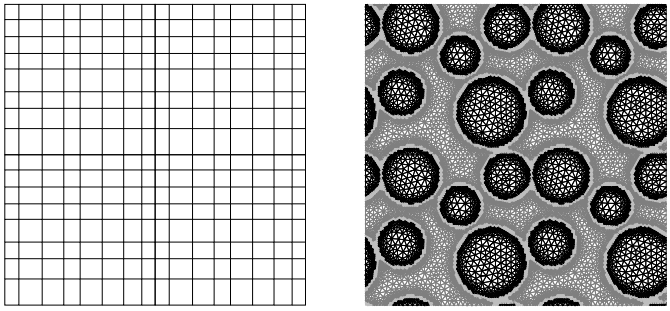

Fig. 2 Coarse (left) and fine (right) scale domain meshes. Coloring in the fine domain: aggregates in black, cement matrix in grey and ITZ in lightgrey.

fine-scale mesh is representative of a typical concrete mesostructure which consists of spherical aggregates, an interface transition zone (ITZ) surrounding the aggregates, and a cementitious matrix material in which the aggregates are embedded. Because of the independence of the individual domains, we are not restricted in mesh, element and material choice per domain provided that the solution field is continuous across the interface.

The parameters are listed in Table 1. Plane strain conditions are considered. The Young's modulus for the homogeneous coarse-scale mesh is an effective Young's modulus derived from the heterogeneous mesh. This is necessary for an accurate material-averaged linear response in the coarse description of the model.

3.2 Software framework and solvers

The non-linear quasistatic calculation is performed by dividing the total applied displacement into 200 load increments. In each load increment the non-

Table 1 Material data

Material Parameters			Aggregates	Matrix	ITZ
E	Young's Modulus	[GPa]	35.0	30.0	20.0
ν	Poisson's ratio	[-]	0.2	0.2	0.2
ε_{eq}	Non-local equivalent strain	[-]	Mazars	Mazars	Mazars
κ_0	Damage Initiation Threshold	[-]	dummy	8.5×10^{-5}	5×10^{-5}
c	Gradient parameter	[mm^2]	0.75	0.75	0.75
$\omega(\kappa)$	Damage evolution law	[-]	Exponential	Exponential	Exponential
α	Residual stress parameter	[-]	0.999	0.999	0.999
β	Softening rate parameter	[-]	150	150	150

linear GD model is evaluated iteratively using an NR scheme with a convergence threshold of 1.0×10^{-6} for the relative error in energy. Usually 3-4 NR iterations are sufficient for the solution to converge.

In the FETI calculations, all factorizations of the domain stiffness matrices are being performed by SuiteSparseQR (Davis [2011]). Solving the flexibility problem iteratively requires projection to ensure positive semi-definiteness of the matrix, allowing the iterative solvers to converge. Because of the asymmetry of the flexibility matrix, only few iterative solvers like BiCGStab by van der Vorst [1992] and GMRES by Saad and Schultz [1986] are suitable. We chose BiCGStab with projection using openMP for the product of the projected stiffness matrix and solution vector (Eqs. (9–12) in Lloberas-Valls et al. [2011]).

Superlumped (SL), lumped (L) and Dirichlet (D) type preconditioners from Rixen and Farhat [1999] are used to accelerate iterative convergence, as well as the multiplicity (m), stiffness (k) and Dirichlet (s) scaling to augment the preconditioners.

The flexibility interface problem can also be solved directly, using openMP for evaluating the flexibility matrix by distributing the domain contributions to the sum over all available parallel cores, followed by a dense matrix solver such as UMFPACK (Davis [2004]). Even though this approach was discouraged in Farhat and Roux [1991] because of the large amounts of solutions required, we have performed this direct calculation since it does provide an upper time limit for finding the Lagrange multipliers with an iterative approach.

An alternative approach is the solution of the set of equations from which the FETI method originates:

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} . \tag{3}$$

Because of the reduction in degrees of freedom, obtained by starting with all coarse domains and a simplified model description, and only substituting domains with fine, heterogeneous counterparts where it is needed, the full

dual assembled matrix is much smaller than the full numerical solution (FNS) and can be solved using parallel direct solvers.

In this contribution we have selected a couple of solvers with the requirement of being able to handle asymmetric cases: MUMPS by Amestoy et al. [2001, 2006], Pardiso by Schenk et al. [2001], PaSiX by Hénon et al. [2002], WSMP by Gupta [2006] and SuperLU by Li [2005], Li et al. [1999], Demmel et al. [1999]. These solvers can also be applied to obtain the FNS.

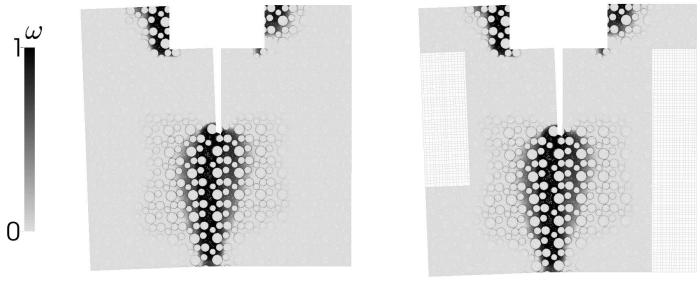

Fig. 3 Comparison of final damage profile of FNS (right) and FETI-direct 34 Domain

4 Results

The full numerical solution and the 34 domain FETI-direct calculations show identical damage patterns and displacements as shown in Fig. 3. However, none of the iterative FETI calculations, regardless of preconditioner and scaling combination, succeed in completing the calculation within the 1000 BiCGStab iteration limit.

Figure 4 shows a significant rise in BiCGStab iterations as the damage calculation progresses. This indicates the inability of the iterative preconditioners and scalings to deal with progressive damage evolution, possibly due to large differences in material stiffness. In order to ascertain this assumption we study the number of iterations for one linear elastic calculation with a domain decomposed mesh, consisting of the 26 zoomed-in domains, by choosing three different load increments i and their corresponding damage profiles ω_i from the FETI-direct calculation and substituting the Young's modulus E by $(1-\omega_i)E$. This approach enables us to observe the dependency of the damage evolution versus the number of iterative steps needed for convergence.

From Table 2 we confirm that the iterations strongly depend on the damage profile: the iterations increase dramatically upon progressively growing differences in material stiffness. This is caused by the differences of orders of magnitudes in the matrix entries. We therefore conclude that the standard

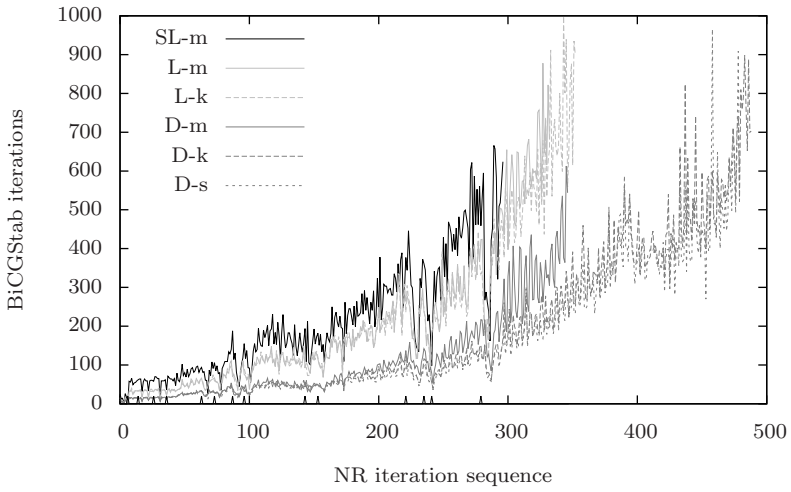

Fig. 4 BiCGStab iteration trend per NR-iteration number. Refer to Subsection 3.2 for an explanation of used preconditioner and scaling acronyms.

Table 2 Linear elastic BiCGStab iteration count as a function of damage profile for a given load increment. Two different preconditioner/scaling results are shown.

	load increment		
preconditoner + scaling	0	100	final
dirichlet + k scaling	16	233	1936
lumped + k scaling	39	781	> 5000

preconditioners and scalings fail to accelerate the BiCGStab iterative solver in situations of substantial damage.

Improving the preconditioners for these type of systems involves adapting new techniques in combination with the damage model, for instance using eigenvalue analysis in FETI-GenEO (Spillane and Rixen [2013]). This is a challenging research topic because of the asymmetric nature of the stiffness matrix in the GD model.

If we instead turn our attention to the parallel direct solvers for both the FNS and full assembly of the FETI system, we see a favourable reduction of time and used memory of the full assembly compared to the FNS for all solvers (Fig. 5). The reduction is not very large, as was expected since the used model system shows an extensive damage pattern affecting 75% of the domains. We are confident that for larger 3D model systems undergoing damage the amount of zoomed in domains will be much smaller and therefore more economic in terms of computation time.

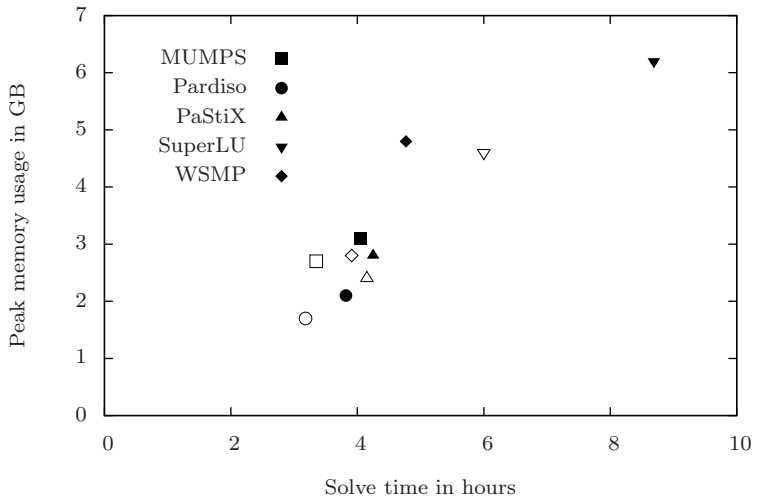

Fig. 5 Comparison of parallel direct solvers. Solid symbols denote FNS, outlined symbols denote multiscale DD.

5 Conclusions

The multiscale framework proposed by Lloberas-Valls et al. [2012a] in combination with a classic FETI approach is shown to provide a reduction of degrees of freedom necessary to efficiently simulate damage evolution in multiscale models of concrete-like materials. By using parallel direct solvers the calculation can be done in less time and memory than the FNS.

In the iterative FETI approach, a high iteration count of the iterative solver is caused by the large differences in material stiffness along domain interface boundaries because of damage evolution. This poses a challenge for existing preconditioners and scalings. We nevertheless expect the iterative FETI to become the most efficient algorithm for very large problems once suitable preconditioners have been identified.

References

P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. A.*, 23(1):15–41, 2001.

P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput.*, 32 (2):136–156, 2006.

T. A. Davis. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, 2004.

T. A. Davis. Algorithm 915: SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Trans. Math. Softw.*, 38(1): 8:1–8:22, December 2011.

J. W. Demmel, J. R. Gilbert, and X. S. Li. An asynchronous parallel supernodal algorithm for sparse gaussian elimination. *SIAM J. Matrix Anal. A.*, 20(4):915–952, 1999.

Zdeněk Dostál, David Horák, and Radek Kučera. Total FETI—an easier implementable variant of the FETI method for numerical solution of elliptic PDE. *Comm. Numer. Methods Engrg.*, 22(12):1155–1162, 2006.

C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32(6):1205–1227, 1991.

A. Gupta. A shared- and distributed-memory parallel sparse direct solver. In *Proceedings of the 7th International Conference on Applied Parallel Computing: State of the Art in Scientific Computing*, PARA'04, pages 778–787, Berlin, Heidelberg, 2006. Springer-Verlag.

P. Hénon, P. Ramet, and J. Roman. PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems. *Parallel Comput.*, 28(2):301–321, January 2002.

X. S. Li. An overview of SuperLU: Algorithms, implementation, and user interface. *ACM Trans. Math. Softw.*, 31(3):302–325, September 2005.

X. S. Li, J. W. Demmel, J. R. Gilbert, L. Grigori, M. Shao, and I. Yamazaki. SuperLU Users' Guide. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory, September 1999. `http://crd.lbl.gov/~xiaoye/SuperLU/`. Last update: August 2011.

O. Lloberas-Valls, D. J. Rixen, A. Simone, and L. J. Sluys. Domain decomposition techniques for the efficient modeling of brittle heterogeneous materials. *Comput. Methods Appl. Mech. Engrg.*, 200(13–16):1577–1590, 2011.

O. Lloberas-Valls, D. J. Rixen, A. Simone, and L. J. Sluys. Multiscale domain decomposition analysis of quasi-brittle heterogeneous materials. *Internat. J. Numer. Methods Engrg.*, 89(11):1337–1366, 2012a.

O. Lloberas-Valls, D. J. Rixen, A. Simone, and L. J. Sluys. On micro-to-macro connections in domain decomposition multiscale methods. *Comput. Methods Appl. Mech. Engrg.*, 225–228:177–196, 2012b.

R. H. J. Peerlings, R. de Borst, W. A. M. Brekelmans, and J. H. P. de Vree. Gradient enhanced damage for quasi-brittle materials. *Internat. J. Numer. Methods Engrg.*, 39(19):3391–3403, 1996.

D. J. Rixen and C. Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Internat. J. Numer. Methods Engrg.*, 44(4):489–516, 1999.

Y. Saad and M. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7(3): 856–869, 1986.

O. Schenk, K. Gärtner, W. Fichtner, and A. Stricker. PARDISO: A high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Gener. Comp. Sy.*, 18(1):69–78, 2001.

N. Spillane and D. J. Rixen. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *Internat. J. Numer. Methods Engrg.*, 95(11):953–990, 2013.

H. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13(2):631–644, 1992.