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 physical phenomena of heterogeneous materials is an ongoing active research field in the structural design of buildings and roads. The failure analysis of quasi-brittle materials such as concrete is a particular topic of interest in civil engineering. This process is characterized by the initial formation of cracks on a microscopic length scale which coagulate into macroscale cracks leading up to weakening and fracture. Because the fracturing process of these materials occurs on several different length scales, care must be taken to provide an accurate description which accounts for all the relevant mechanical processes while maintaining a reasonable computation cost. With this reasoning in mind, we use a multiscale approach in our numerical simulation, switching between different meshes and material parameters depending on the local mechanical behaviour.

In this contribution, we will present a non-linear finite element computation involving a non-local damage model of a wedge-split test used for evaluating fracture mechanics in concrete-like materials. We will apply the classical FETI framework (Farhat and Roux [1991]) to the 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 damage model used in modelling concrete fracture is the gradient enhanced damage model described in detail in Peerlings et al. [1996]. It introduces a scalar, the damage parameter ω, which modifies the stress–strain relation according to

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

The damage parameter ω varies from 0 for undamaged to 1 for fully damaged material. Its evolution is a function of the history parameter κ

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

The GD model is non-local: it consists of a coupled set of equations involving the modified Helmholtz equation for the non-local equivalent strain and the classical quasi-static stress-strain relation, both having to be solved simultaneously. The damage evolution is highly non-linear, requiring the use of a loop control dividing the total displacement into small steps with an iterative Newton-Raphson (NR) scheme for each step to assure convergence.

The underlying damage formalism results in an unsymmetrical stiffness matrix. To solve the set of equations one has to use solvers supporting asymmetry, both in direct and iterative approaches.

2.2 Classical FETI method

The classical FETI method, Farhat and Roux [1991], is used to solve our multiscale system with coarse and fine grids for each domain. 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. [2012]).

Boundary conditions are also included by means of Lagrange multipliers, which implies that all domains in this framework are floating. Rigid body motion vectors are thus constructed to enforce compatibility between domains. In order 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].

2.3 Multiscale domain decomposition

In order to solve the discrete system in a reasonable amount of time, we employ domain decomposition of the numerical model into separate domains. Each domain starts out with a coarse mesh with homogeneous material parameters, and it is then checked if a certain condition, in this case a local strain difference, has reached a threshold. If this condition is met, that domain subsequently switches to the fine scale mesh and parameters. Finally the displacements and forces of the new domain mesh are matched by means of solving a boundary value problem (BVP) combined with a linear relaxation of the entire mesh.

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.

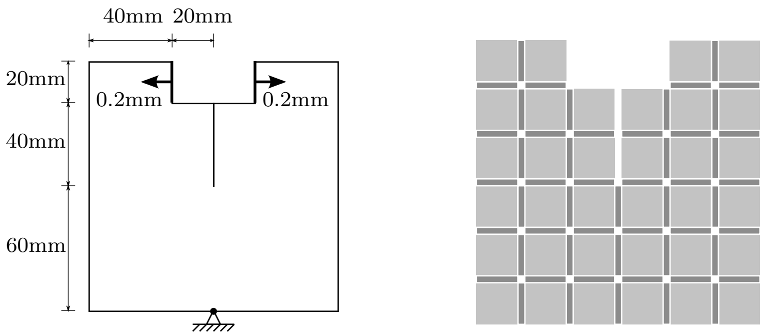

Fig. 1 Dimensions and domain decomposition of the wedge split model test. The interface is represented in dark-grey.

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, as shown in Fig. 2. The fine-scale mesh consists of three materials: the spherical aggregates, an interface transition zone (ITZ) enveloping the aggregates, and the surrounding matrix. Because of the independency of the individual domains, we are not restricted in mesh, element and material choice per domain.

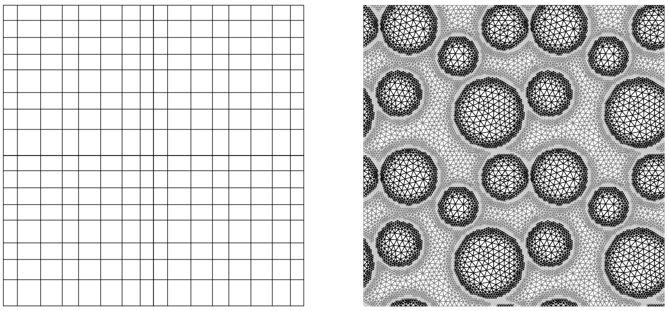

Fig. 2 Coarse (left) and fine (right) scale domain meshes. Coloring in the fine domain: aggregates in black, cement matrix in grey and ITZ in lightgrey.

The parameters are listed in table 1. Plane strain conditions are applied to the 2D model. The Young's modulus parameter for the homogeneous coarse-scale mesh is actually an effective Young's modulus for the heterogeneous mesh. This is necessary for an accurate material-averaged linear response in the coarse description of the model.

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

3.2 Software framework and solvers

The non-linear quasistatic calculation is performed by dividing the total displacement boundary condition into 200 load increments or timesteps. In each timestep the non-linear GD model is evaluated iteratively using a Newton-Raphson (NR) scheme with a convergence threshold of 1.0×10^{-6} for the relative error in energy. Usually 3-4 NR iterations are sufficient.

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 BiCGStab by van der Vorst [1992] and GMRES by Saad and Schultz [1986] are suitable as iterative solvers. 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) scalers 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 the 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 solves 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 solve of the set of equations from which the FETI method originates:

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

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 FNS and can be solved using parallel hybrid 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 full numerical solution (FNS).

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 calculation, regardless of preconditioner and scaling combination, succeed in completing the calculation within the 1000 BiCGStab iteration limit.

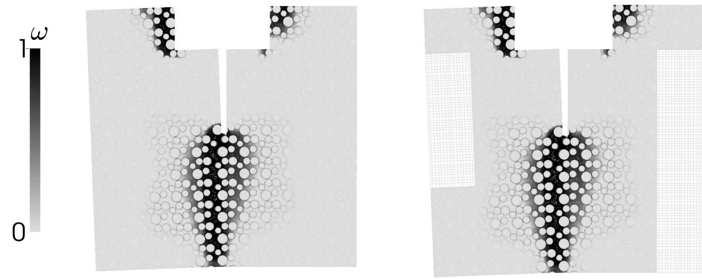

Fig. 3 Comparison of final damage profile of FNS (right) and FETI-direct 34 Domain

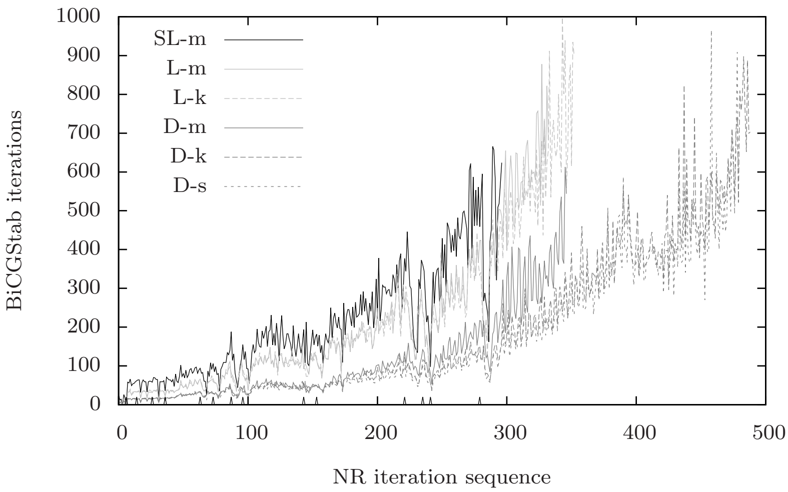

Fig. 4 BiCGStab iteration trend per NR-iteration number. Refer to Subsection 3.2 for an explanation of used preconditioner and scaler acronyms.

Figure 4 shows a significant rise in BiCGStab iterations as the damage calculation progresses. This indicates the inability of the iterative preconditioners and scalers 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 timesteps 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.

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

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

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 preconditioners and scalers 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 unsymmetric nature of the stiffness matrix in the GD model.

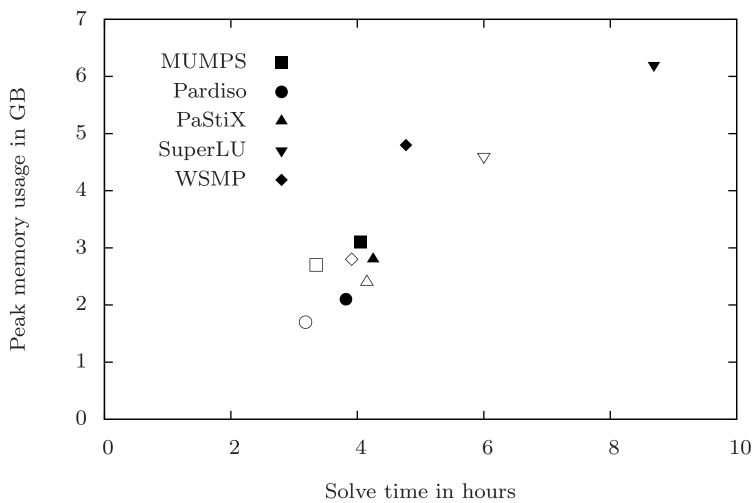

Fig. 5 Comparison of parallel direct solvers

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 (figure 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.

5 Conclusions

The multiscale framework proposed by Lloberas-Valls et al. [2011]. in combination with a classic FETI approach is shown to provide a reduction of degrees of freedom necessary to efficiently calculate damage evolution in 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 scalers. 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.

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 Com-*

puting: 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. On micro-to-macro connections in domain decomposition multiscale methods. *Comput. Methods Appl. Mech. Engrg.*, 225–228:177–196, 2012.

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.