FETI-DP Solvers and Deal.II for Problems in Dislocation Mechanics

Stephan Köhler\textsuperscript{1-\textdagger}, Oliver Rheinbach\textsuperscript{1-**}, Stefan Sandfeld\textsuperscript{2}, and Dominik Steinberger\textsuperscript{2}

\textsuperscript{1} Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg
\textsuperscript{2} Fakultät für Maschinenbau, Verfahrens- und Energietechnik, Technische Universität Bergakademie Freiberg

FETI-DP (Finite Element Tearing and Interconnecting Dual-Primal) solvers and the deal.II adaptive finite element library are combined to solve dislocation eigenstrain problems in micromechanics. Computational results using adaptive finite elements with millions of unknowns and up to 3072 cores of the Taurus supercomputer at ZIH in Dresden are presented.

1 Mechanical Model

To compute the stresses associated with dislocations within a specimen for the characterization of the microstructure \cite{1,2}, we start by considering a linear elastic model described by

\[
\text{div} \, \sigma = 0, \quad \sigma = \sigma^T, \quad \sigma = C : \varepsilon^{\text{el}}, \quad \text{and} \quad \varepsilon^{\text{el}} = \frac{1}{2} (\nabla u + (\nabla u)^T)
\]

to solve for the displacements \(u\). Here, \(\sigma\) is the stress tensor, \(\varepsilon^{\text{el}}\) the elastic strain tensor, and \(C\) the stiffness tensor. Dislocations are one-dimensional defects present in crystalline materials. They are the boundary of an area over which two crystal halves have been displaced by the so-called Burgers vector \(b\).

Within the linear elastic model, dislocations may be modeled by an eigenstrain approach \cite{3,4} by expressing the total strain by

\[
\varepsilon^{\text{tot}} = \varepsilon^{\text{el}} + \varepsilon^{\text{eig}},
\]

where \(\varepsilon^{\text{eig}}\) is the eigenstrain contribution due to the dislocation microstructure. The area enclosed by a dislocation is described by a vector \(A\) that is perpendicular to it. The eigenstrain contributions

\[
d\varepsilon^{\text{eig}} = \frac{1}{2} (\vec{b} \otimes d \vec{A} + d \vec{A} \otimes \vec{b}),
\]

where \(\otimes\) denotes the outer product, are regularized using the non-singular formulation proposed by Cai et al. \cite{5}, similarly to the work by Jamond et al. \cite{6}. The eigenstrain of a dislocation is a contribution to the right hand side of a linear elasticity problem.

2 Implementation of FETI-DP in Deal.II and Parallel Results

The Finite Element Tearing and Interconnecting Dual-Primal method \cite{7} is a nonoverlapping domain decomposition method for the solution of partial differential equations. We decompose our computational domain \(\Omega\) into nonoverlapping subdomains \(\Omega_1, \ldots, \Omega_N\) and define the interface \(\Gamma\) between the subdomains as \(\Gamma := (\partial \Omega_i \setminus \partial D) \setminus \partial \Omega_D\), where \(\partial \Omega_D\) denotes the Dirichlet boundary of \(\Omega\). The degrees of freedom (dof) of the interface are partitioned into dual variables (\(\Delta\)) and primal variables (\(\Pi\)). All remaining dofs are denoted as inner dofs (\(I\)). The global problem is obtained by assembling the local subdomain problems (\(K^{(1)}, f^{(1)}\)) only in the primal dofs. To enforce continuity of the dual dofs of the solution a constraint \(Bu = 0\) and Lagrange multipliers \(\lambda\) are used. This leads to the global saddle point problem

\[
\begin{bmatrix}
\bar{K} & B^T & \bar{u} \\
B & O & \lambda
\end{bmatrix} = \begin{bmatrix}
f \\
0
\end{bmatrix}, \quad \text{where} \quad \bar{K} = \begin{bmatrix}
K_{BB}^{(1)} & \ldots & K_{B\Pi}^{(1)} \\
\vdots & \ddots & \vdots \\
K_{\Pi B}^{(N)} & \ldots & K_{\Pi\Pi}^{(N)}
\end{bmatrix}, \quad \bar{f} = \begin{bmatrix}
f_B^{(1)} \\
\vdots \\
f_B^{(N)}
\end{bmatrix}
\]

and subscript \(B\) denotes the union of \(\Delta\) and \(I\). The assembly in the primal dofs ensures the invertibility of \(\bar{K}\). The system (1) can be solved by eliminating \(\bar{u}\) and solving the remaining system \(F\lambda = d\) using a Krylov method and a preconditioner; for more details, see, e.g., \cite{8,9}. We interface our parallel FETI-DP implementation \cite{9} based on PETSc \cite{10} to deal.II.

\textsuperscript{\dagger} Corresponding author: e-mail stephan.koehler@math.tu-freiberg.de, phone +49 (0)3731 39 3173

\textsuperscript{**} Corresponding author: e-mail oliver.rheinbach@math.tu-freiberg.de, phone +49 (0)3731 39 3279

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{28 dislocations in \([-10^{-7}, 10^{-7}]^3\), 96 MPI ranks. Left: Mesh after second adaptive refinement. Right: Displacement.}
\end{figure}
Deal.II [11] is a C++ finite element library making use of adaptive mesh refinement and hanging node constraints. The parallelization in deal.II is based on MPI, and, for parallel linear algebra, PETSc or Trilinos can be interfaced. For the FETI-DP method, all subdomain matrices $K^{(i)}$ and load vectors $f^{(i)}$ must be held unassembled.

For a parallel distributed triangulation, each grid cell belongs to exactly one MPI rank. Each rank has information about its locally owned cells and one ghost layer of neighboring cells. In deal.II, only a global numbering of the dofs is used. Each dof belongs also to exactly one rank, given by the locally owned dofs. Additionally, each rank has the index set locally active dofs (dofs of locally owned cells) and locally relevant dofs (locally active dofs plus the dofs of ghost cells). Deal.II uses the class ConstraintMatrix for the handling of the hanging node constraints and the Dirichlet boundary values. To set up the local sparsity patterns and local constraint matrices, a global instance is built and the entries are copied according to a local numbering of the locally relevant dofs. This local numbering of the locally relevant dofs is also used for the computation of the interface dofs. The interface dofs are defined as all non-hanging node dofs which are shared between a locally owned cell and a ghost cell. For the hanging node dofs which are shared between a locally owned cell and a ghost cell, all dofs by which the hanging node dofs are constrained are selected as interface dofs. The primal dofs can be selected from the interface dofs. Also, faces, edges and vertices can be computed [9]. Note, that the primal constraints have to ensure the invertibility of all matrices $K^{-1}_{BB}$, $i = 1, \ldots, N$. Furthermore, there has to be some special care in the treatment of the Dirichlet boundary conditions and in the case of non-connected subdomains.

For our numerical experiments in Figure 2 and Table 1, a problem with 28 dislocations in the cube $[-5\varepsilon - 7, 5\varepsilon - 7]^3$ was considered. The grid refinement was based on the KellyErrorEstimator in deal.II. The partitioning is performed by p4est [12]. We use a relatively large number of nodal coarse dofs to ensure the invertibility of $K$ for very complex grid decompositions. The solve time is therefore, up to refinement level 4, dominated by the coarse factorization and forward-backward substitution using MUMPS. A better approach would be to use edge or face averages which would also lead to a fewer number of coarse dofs and also better conditioning, see, e.g. [9]. The stopping criterion was a relative residual reduction to $1e - 9$. Our results show that we can solve adaptive finite element problems from dislocation mechanics with up to 20M unknowns.

Acknowledgements The computations were performed on a Bull Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden. We like to thank Guido Kanschat and Daniel Arndt for their support with deal.II. The third and the last author acknowledge funding from the European Research Council starting Grant, ”A Multiscale Dislocation Language for Data-Driven Materials Science,” ERC Grant Agreement No. 759419 MuDiLingo.

### References

1. S. Sandfeld, M. Monavari, and M. Zaiser, Modelling and Simulation in Materials Science and Engineering 21(8), 085006 (2013).
2. D. Steinberger, R. Gatti, and S. Sandfeld, JOM 68(8), 2065–2072 (2016).
3. J. D. Eshelby and R. E. Peierls, Proc. of the Royal Society of London. Series A, Math. and Phys. Sciences 241(1226), 376–396 (1957).
4. J. D. Eshelby and R. E. Peierls, Proc. of the Royal Society of London. Series A, Math. and Phys. Sciences 252(1271), 561–569 (1959).
5. W. Cai, A. Arsenlis, C. R. Weinberger, and V. V. Bulatov, Journal of the Mechanics and Physics of Solids 54(3), 561–587 (2006).
6. O. Jamond, R. Gatti, A. Roos, and B. Devincere, International Journal of Plasticity 80, 19–37 (2016).
7. C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen, Int. J. Numer. Meth. Engng. 50(7), 1523–1544 (2001).
8. A. Toselli and O. Widlund, Domain decomposition methods-algorithms and theory (Springer Science & Business Media, 2006).
9. A. Klawonn and O. Rheinbach, SIAM Journal on Scientific Computing 28(5), 1886–1906 (2006).
10. J. D. Eshelby and R. E. Peierls, Proc. of the Royal Society of London. Series A, Math. and Phys. Sciences 252(1271), 561–569 (1959).
11. G. Alzetta, D. Arndt, W. Bangerth, V. Boddt, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J. P. Pelteret, B. Turcksin, and D. Wells, Journal of Numerical Mathematics 26(4), 173–183 (2018).
12. C. Burstedde, L. C. Wilcox, and O. Ghattas, SIAM Journal on Scientific Computing 33(3), 1103–1133 (2011).

© 2019 The Authors. Proceedings in Applied Mathematics & Mechanics published by Wiley-VCH Verlag GmbH & Co. KGaA Weinheim www.gamm-proceedings.com