COMPUTATIONAL MECHANICS 2025
40th conference with international participation
Srní
November 3 - 5, 2025

Solving piezoelastic problems on periodic lattices using block preconditioners

R. Cimrman[a,b]

[a] Institute of Thermomechanics, Czech Academy of Sciences, Dolejškova 1402/5, 18200, Praha 8, Czech Republic
[b] New Technologies Research Centre, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic

Polymer materials, such as polyvinylidene fluoride (PVDF), can be polarized during or after the 3D printing process, making them suitable for producing electroactive metamaterials. A 3D-printed periodic lattice made of such a piezoelectric (PZ) material can then exhibit electroactivity that differs from that of the original material. One example is flexoelectricity, in which the electric polarization is coupled with the strain gradient instead of the strain. Designing such periodic lattices requires the ability to perform numerical simulations efficiently. Towards this goal, an approach based on iterative solvers preconditioned by state-of-the-art preconditioners is presented below.

The FE discretization of a static linear piezoelasticity problem in terms of the displacements $\mathbf{u}$ and potential $\mathbf{p}$ degrees of freedom (DOFs) leads to a linear system with the following structure [5]:

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ -\mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} , \tag{1}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{B}$ is the piezoelastic coupling matrix, $\mathbf{C}$ is the electrostatic potential matrix, and $\mathbf{f}$ is the volume forces vector.

While the system (1) can be solved using a direct solver for problems of small to moderate size, for large problems an iterative solver is necessary. Here the flexible generalized minimal residual method (FGMRES) is considered, as implemented in PETSc [2] and used via the petsc4py Python interface [7], together with two preconditioners:

1. a simple block diagonal (BD) preconditioner [8]

$$\mathbf{P}_{BD} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} , \tag{2}$$

2. the cell-by-cell Schur complement (CBCS) preconditioner [3]

$$\mathbf{P}_{CBCS} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ -\mathbf{B} & \mathbf{S}_{CBC} \end{bmatrix} , \tag{3}$$

where $\mathbf{S}_{CBC} = \sum_{T_e \in \Omega_h} \mathbf{C}_e + \mathbf{B}_e^T \mathbf{K}_e^{-1} \mathbf{B}_e$, and T_e are the elements of the FE domain Ω_h.

The performance of these solvers is compared to the direct solver MUMPS [1] in terms of memory usage and elapsed time.

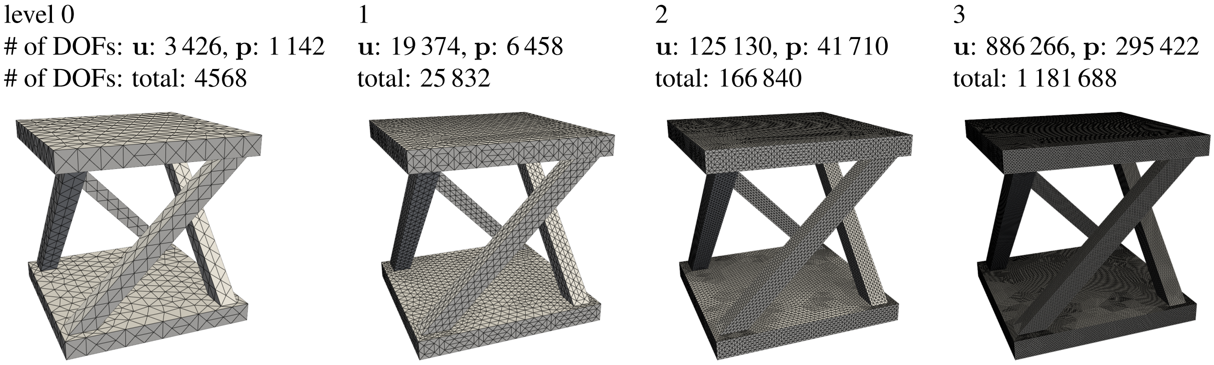


Fig. 1. Lattice geometry, uniform mesh refinement levels and their respective DOF counts

The numerical example is implemented using the open-source finite element code SfePy [4, 6]. This package was recently enhanced [5] to support preconditioners such as (3) in order to allow efficient modeling of 3D printed lattice-like electroactive metamaterials. Various lattices can be constructed by tiling a reference periodic cell (RPC) such as the one shown for various levels of uniform mesh refinement in Fig. 1.

The RPC lattice was fixed in the bottom plane, a voltage of 1 000 V was applied to the upper surfaces of the skewed bars and 0 V to the lower surfaces of these bars. The performance of the direct solver MUMPS and the preconditioned iterative solvers BD-FGMRES, CBCS-FGMRES was evaluated for the meshes in Fig. 1. The relative tolerance of FGMRES was set to 10^{-8}. The preliminary results shown in Fig. 2 indicate the following:

1. The global convergence of BD-FGMRES and CBCS-FGMRES does not depend on the refinement level, consistently with results published in [8].
2. The iterative solvers need less memory, but the problem sizes are still too small to have an advantage in speed.

Note that the solvers were limited to use only one thread/process (no automatic parallelization). The preconditioned iterative solvers are a necessity for large problems such as the ones illustrated for different tiling arrangements of the RPC in Fig. 3.

Acknowledgement

This work has been supported by the grant 23-06220S of the Czech Science Foundation within institutional support RVO:61388998.

References

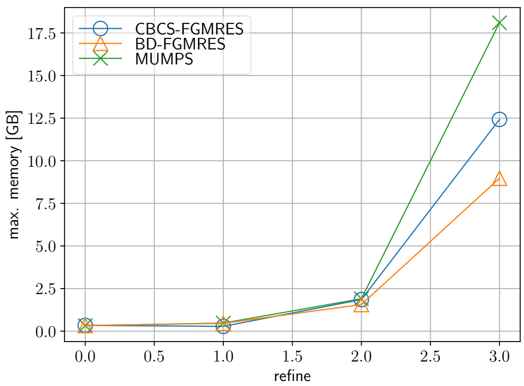
[1] Amestoy, P. R., Buttari, A., L'Excellent, J. Y., Mary, T., Performance and scalability of the block low-rank multifrontal factorization on multicore architectures, ACM Transactions on Mathematical Software 45 (1) (2019) 1–26.

[2] Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J. et al., PETSc web page, 2024. https://petsc.org/

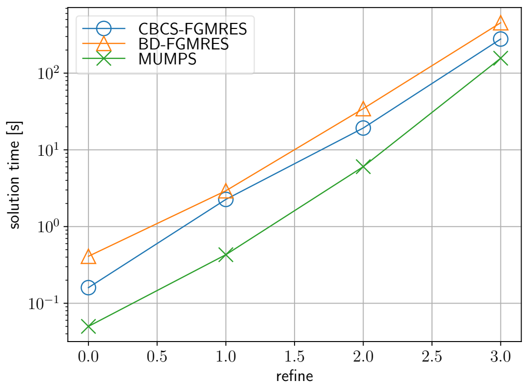
[3] Cao, Y., Neytcheva, M., Cell-by-cell approximate schur complement technique in preconditioning of meshfree discretized piezoelectric equations, Numerical Linear Algebra with Applications 28 (4) (2021) No. e2362.

[4] Cimrman, R., Fast evaluation of finite element weak forms using Python tensor contraction packages, Advances in Engineering Software 159 (2021) No. 103033.

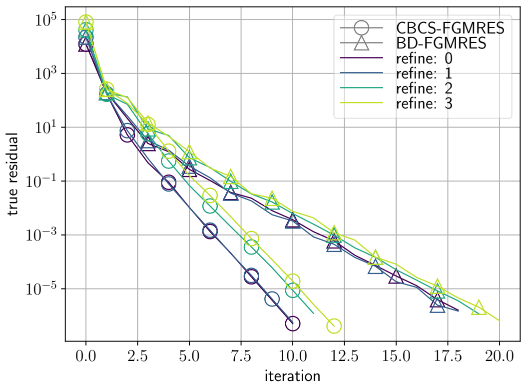
[5] Cimrman, R., Implementation of block preconditioners for piezoelasticity in sfepy, Proceedings of the 31th International Conference Engineering Mechanics 2025, Medlov, 2025, pp. 61–64.



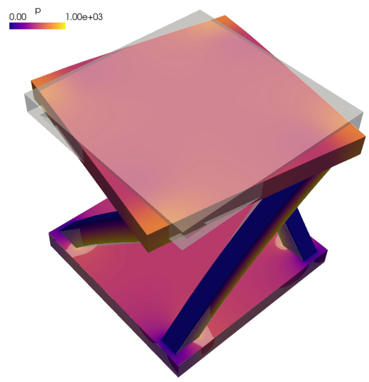
(a) Peak memory usage



(b) Solution times

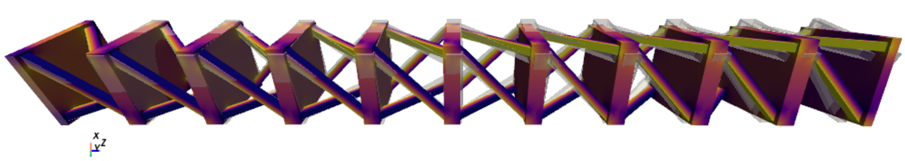


(c) Convergence curves

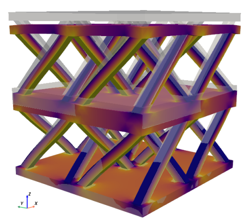


(d) The solution

Fig. 2. Dependence of (a) memory and (b) time requirements of the direct (crosses) and iterative solvers on the uniform mesh refinement level; (c) convergence curves of the iterative solvers; (d) the finest mesh solution: the displacements indicated by the magnified deformation, the potential is given by color, the original undeformed geometry is in gray



tiling (1,1,10)



tiling (2,2,2)

Fig. 3. Tiled RPC solutions: the displacements indicated by the magnified deformation, the potential is given by color; the original undeformed geometry is in gray

[6] Cimrman, R., Lukeš, V., Rohan, E., Multiscale finite element calculations in Python using SfePy, Advances in Computational Mathematics 45 (2019) 1 897–1 921.

[7] Dalcin, L. D., Paz, R. R., Kler, P. A., Cosimo, A., Parallel distributed computing using Python, Advances in Water Resources 34 (9) (2011) 1 124–1 139.

[8] Wegert, Z. J., Roberts, A. P., Challis, V. J., Level set-based inverse homogenisation of three-dimensional piezoelectric materials, 2024, doi: 10.48550/arXiv.2410.03148.