

## Parallel Solution of the Linear Elasticity problem with applications in Topology Optimization

James Turner<sup>1</sup>, Michal Kocvara<sup>1</sup> and Daniel Loghin<sup>1</sup>

<sup>1</sup>School of Mathematics, University of Birmingham, UK.

[turnerj@maths.bham.ac.uk](mailto:turnerj@maths.bham.ac.uk)

### Abstract

In this paper, we aim to solve the system of equations governing linear elasticity in parallel using domain decomposition. Through a non-overlapping decomposition of the domain, our approach aims to target the resulting interface problem, allowing for the parallel computation of solutions in an efficient manner. As a major application of our work, we apply our results to the field of topology optimization, where typical solvers require repeated solutions of linear elasticity problems resulting from the use of a Picard approach.

### 1 Introduction

Consider a solid elastic body occupying an open and connected domain  $\Omega \subset \mathbb{R}^d$  with Lipschitz boundary  $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ , where clamping and traction are imposed on  $\partial\Omega_D$  and  $\partial\Omega_N$  respectively. Under the application of both body forces  $f: \Omega \rightarrow \mathbb{R}^d$  and boundary tractions  $g: \partial\Omega_N \rightarrow \mathbb{R}^d$ , the material is subject to deformation so that a given reference point  $\mathbf{x}$  of the initial undeformed material is translated to the vector  $\mathbf{x}' = \mathbf{x} + u(\mathbf{x})$  of the deformed material, with  $u$  denoting the displacement. Through the assumption of linearly elastic material behaviour, the governing equations for  $u$  correspond to the following mixed boundary value problem

$$\mathcal{L}u := -\nabla \cdot \sigma(u) = f \quad \text{in } \Omega, \quad (1a)$$

$$\sigma(u) = E: \varepsilon(u) \quad \text{in } \Omega, \quad (1b)$$

$$u = 0 \quad \text{on } \partial\Omega_D, \quad (1c)$$

$$\sigma(u) \cdot \hat{n} = g \quad \text{on } \partial\Omega_N. \quad (1d)$$

In the above, the strain caused as a result of the displacements  $u$  is characterized by the symmetric linearised strain tensor

$$\varepsilon(u) = \{\varepsilon_{ij}(u)\}_{i,j=1}^d, \quad \varepsilon_{ij}(u) = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

In the above,  $\hat{n}$  corresponds to the unit outward pointing normal vector on  $\partial\Omega$  and  $E := E(\mathbf{x})$  denotes the fourth order elasticity tensor, describing the elastic stiffness of  $\Omega$  as a result of the load placed upon it.

We will consider the case where our body consists of one or more isotropic materials (i.e: rotational and directional independence). Equation (1b) describing Hooke's Law can be written as

$$\sigma(u) = 2\mu \varepsilon(u) + \hat{\lambda} \text{tr}(\varepsilon(u))\hat{I},$$

where  $\hat{I}$  represents the identity matrix of appropriate size,  $\text{tr}(A)$  denotes the trace of a matrix  $A$  and both  $\mu$  and  $\hat{\lambda}$  correspond to Lamé constants defined in the usual manner

$$\mu := \frac{\bar{E}}{2(1+\nu)}, \quad \hat{\lambda} := \frac{\nu\bar{E}}{(1+\nu)(1-2\nu)}.$$

with  $\bar{E} > 0$  corresponding to Young's modulus and  $-1 < \nu < \frac{1}{2}$  the Poisson ratio. We now look to apply domain decomposition to the problem (1). To do this, we divide our domain  $\Omega$  into  $N$  nonoverlapping subdomains  $\Omega_i$  with local boundaries  $\partial\Omega_i$  and outer unit normals  $\hat{n}_i$ . We denote by  $\Gamma$  the resulting skeletal interface  $\Gamma = \cup_{i=1}^N \Gamma_i$  where  $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$ , and by  $I := \bar{\Omega} \setminus \Gamma$  the set of interior nodes, with  $u|_{\Omega_i} =: u_i$ . Assuming that the restriction of  $u_i$  to components of the skeletal interface  $u_i|_{\Gamma_i}$  is known, problem (1) is equivalent to the following set of subproblems

$$\begin{aligned} \mathcal{L}u_i &= f_i && \text{in } \Omega_i, \\ u_i &= 0 && \text{on } \partial\Omega_D \cap \partial\Omega_i, \\ \sigma(u_i) \cdot \hat{n}_i &= g_i && \text{on } \partial\Omega_N \cap \partial\Omega_i, \\ u_i &= \lambda_i && \text{on } \Gamma_i, \end{aligned} \quad (2)$$

where  $i = 1, \dots, N$ . By writing  $u_i = u_i^1 + u_i^2$ , we look to describe an appropriate interface operator that will allow problems to be decoupled and thus solved strictly on subdomains in parallel. Through the definition of matrix extension operators  $H_i$  that map interface data to relevant subdomains via  $u_i^2 = H_i\lambda$ , the system (2) can be decoupled into the following  $2N + 1$  subproblems

$$\begin{aligned} \mathcal{L}u_i^1 &= f_i && \text{in } \Omega_i, \\ u_i^1 &= 0 && \text{on } \partial\Omega_D \cap \partial\Omega_i, \\ \sigma(u_i^1) \cdot \hat{n}_i &= g_i && \text{on } \partial\Omega_N \cap \partial\Omega_i, \\ u_i^1 &= 0 && \text{on } \Gamma_i, \end{aligned} \quad (3a)$$

$$\sum_{i=1}^N \sigma(H_i\lambda) \cdot \hat{n}_i = - \sum_{i=1}^N \sigma(u_i^1) \cdot \hat{n}_i \quad \text{on } \Gamma, \quad (3b)$$

$$\begin{aligned} \mathcal{L}u_i^2 &= 0 && \text{in } \Omega_i, \\ u_i^2 &= 0 && \text{on } \partial\Omega_D \cap \partial\Omega_i, \\ \sigma(u_i^2) \cdot \hat{n}_i &= 0 && \text{on } \partial\Omega_N \cap \partial\Omega_i, \\ u_i^2 &= \lambda_i && \text{on } \Gamma_i, \end{aligned} \quad (3c)$$

The associated weak form to problem (3b) is referred to as the Steklov-Poincaré equation, where the so-called Steklov-Poincaré pseudo-differential operator  $S : \Lambda_\theta \rightarrow \Lambda'_\theta$  is defined in the following manner (Turner, 2014)

$$s(\lambda, \eta) = \langle S\lambda, \eta \rangle := \sum_{i=1}^N \left[ \int_{\Gamma_i} (\sigma(H_i\lambda) \cdot \hat{n}_i) \eta \, ds \right] =: \sum_{i=1}^N \langle S_i\lambda_i, \eta_i \rangle,$$

where  $\lambda, \eta \in \Lambda_\theta$  and  $\lambda|_{\Gamma_i} =: \lambda_i$ ,  $\eta|_{\Gamma_i} =: \eta_i$ . The space  $\Lambda_\theta$  is chosen to be a suitable fractional Sobolev space of index  $\theta$  based on the boundary conditions of the problem, dependent on the intersection of  $\Gamma$  with  $\partial\Omega$  (Quarteroni & Valli, 1999, Toselli & Widlund, 2005).

## 2 Matrix Formulation

Through a finite element discretisation of the weak formulation, it can be shown that the discrete formulation to the original problem (1) requires the solution of a matrix-vector system (Turner, 2014). By distributing nodes based on their location within the domain, we can view this system as follows

$$K\mathbf{u} = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{pmatrix} = \mathbf{f}, \quad (4)$$

where

$$K_{II} = \bigoplus_{i=1}^N K_{I_i I_i}, \quad \mathbf{u} = (\mathbf{u}_I, \mathbf{u}_\Gamma) \in \mathbb{R}^{n_I + n_\Gamma}. \quad (5)$$

In comparison, the corresponding matrix formulations for each of the discrete weak formulations to the problems presented in (2) can be written down as

$$K_{II}\mathbf{u}_I^1 = \mathbf{f}_I, \quad (6a)$$

$$S\mathbf{u}_\Gamma = \mathbf{f}_\Gamma - K_{\Gamma I}\mathbf{u}_I^1, \quad (S = K_{\Gamma\Gamma} - K_{\Gamma I}K_{II}^{-1}K_{I\Gamma}) \quad (6b)$$

$$K_{II}\mathbf{u}_I^2 = -K_{I\Gamma}\mathbf{u}_\Gamma, \quad (6c)$$

with global solution  $\mathbf{u} = (\mathbf{u}_I^1, \mathbf{0})^T + (\mathbf{u}_I^2, \mathbf{u}_\Gamma)^T$ . In the above, the matrix  $S$  corresponds to the Schur complement, and so the discretisation of the decoupled problem (2) can be viewed as a Schur complement approach to the discretisation of the global problem (1). Using (4) and (5), we are able to view (6) in terms of  $2N + 1$  subproblems in the following way

$$K_{I_i I_i}\mathbf{u}_{I_i}^1 = \mathbf{f}_{I_i}, \quad i = 1, \dots, N \quad (7a)$$

$$S\mathbf{u}_\Gamma = \mathbf{f}_\Gamma - K_{\Gamma I_i}\mathbf{u}_{I_i}^1, \quad (7b)$$

$$K_{I_i I_i}\mathbf{u}_{I_i}^2 = -K_{I_i \Gamma}\mathbf{u}_\Gamma, \quad i = 1, \dots, N \quad (7c)$$

We therefore look to solve (4) by exploiting the potential for parallelisation present in (7).

## 3 Preconditioning

The systems we expect to solve will typically be both sparse and large scale, due to the expected fineness of the finite element discretization required in modern design processes, allowing for the computation of resolute solutions. This is of particular importance for domains containing sharp jumps, occurring for instance due to predefined fixed or void regions. Therefore, it is appropriate to consider iterative solution techniques when solving systems of the form (4). For our problem, we will consider GMRES (Saad & Schultz, 1986) for reasons to be described below.

In order to avoid the direct construction and application of the Schur complement matrix, and to improve the spectral properties of the system matrix, we seek an appropriate preconditioner for the system (4). Through the following choice of  $P$ , we see that

$$P = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ 0 & S \end{pmatrix}, \quad KP^{-1} = \begin{pmatrix} \hat{I}_{II} & 0 \\ K_{\Gamma I}K_{II}^{-1} & \hat{I}_{\Gamma\Gamma} \end{pmatrix}.$$

The minimum polynomial of  $KP^{-1}$  is  $(\lambda - 1)^d$ , suggesting that iterative solution methods such as GMRES will converge in at most  $d$  iterations (Ipsen, 2002). Based on this, we propose to precondition from the right with an approximation  $\tilde{P}$  of  $P$  as follows

$$K\tilde{P}^{-1}\tilde{\mathbf{u}} = \mathbf{f}, \quad \tilde{\mathbf{u}} = \tilde{P}\mathbf{u},$$

where

$$\tilde{P} = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ 0 & \tilde{S} \end{pmatrix}, \quad \tilde{P}^{-1} = \begin{pmatrix} K_{II}^{-1} & 0 \\ 0 & \hat{I}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \hat{I}_{II} & -K_{I\Gamma} \\ 0 & \hat{I}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \hat{I}_{II} & 0 \\ 0 & \tilde{S}^{-1} \end{pmatrix},$$

with  $\tilde{S}$  representing an approximation to the discrete Steklov-Poincaré operator. We therefore seek a representation of  $\tilde{S}$  that is not only practical to invert but can also be seen to provide an appropriate preconditioning strategy for the resulting interface problem.

The form of  $\tilde{S}$  chosen is based on work in (Arioli & Loghin, 2009), where discrete norm representations for projections of the interpolation spaces  $\Lambda_\theta$  onto suitable finite dimensional subspaces are described and analysed. Discrete norms of the form

$$H_\theta = M + M(M^{-1}L)^{1-\theta}, \quad \theta \in [0,1], \quad (8)$$

are shown to be equivalent to their continuous counterparts on  $\Lambda_\theta$ , where  $M$  and  $L$  denote the mass and Laplacian matrices respectively assembled on the interface  $\Gamma$ . One particular example corresponds to  $\theta = 1/2$ , which is shown in (Arioli & Loghin, 2009) to adhere to the same coercivity and continuity bounds as the discrete Steklov-Poincaré operator, leading to mesh independent performance of GMRES. The norms presented in (8) can be shown to be spectrally equivalent to

$$\tilde{H}_\theta = M(M^{-1}L)^{1-\theta},$$

Both of the above can be applied component-wise to a system, suggesting an appropriate form of  $\tilde{S}$  as

$$\hat{H}_\theta = \bigoplus_1^d \tilde{H}_\theta. \quad (9)$$

From the above, it is clear that fractional powers of matrices must be determined in order to apply the discrete norms. For relatively small problems, this can be achieved through direct methods such as a generalised eigenvalue decomposition. However, the complexity involved is  $O(n_\Gamma^3)$  suggesting instead the use of iterative approaches for larger problems. In (Arioli & Loghin, 2009), approximations through the use of truncated Lanczos and inverse Lanczos algorithms are described, and will also be employed within this work through the use of flexible GMRES (Saad, 1993) to account for the changing nature of the preconditioner.

## 4 Results

We present various results in this section to illustrate our approach in practice. It should be noted that while certain examples involve symmetric system matrices, our choice of non-symmetric preconditioner suggests GMRES as an appropriate Krylov solver.

The test problem considered involves a cantilever beam over the 2D domain  $\Omega = (0,2) \times (0,1)$ , with downward force  $f = 0.75$ , and outward traction  $g = 1$ . The domain will be clamped on the right hand side through the application of homogeneous Dirichlet conditions on the relevant boundary. An illustration of the deflection as well as a pictorial example of a division of the domain (into 16 subdomains) is provided in Figure 1.

As discussed in the previous section, iterative approaches will be used for the application of  $\hat{H}_\theta$ . For this work, it was found that the inverse Lanczos approach delivered the most promising results, largely due to the relatively small number of basis vectors required to apply the discrete norms.

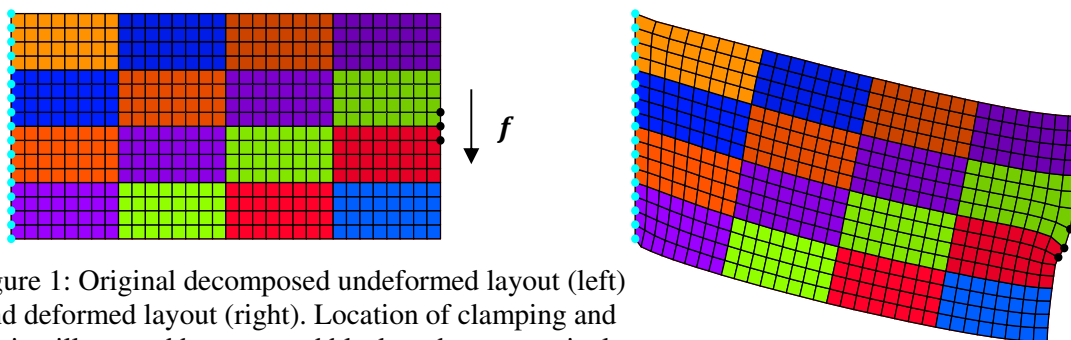


Figure 1: Original decomposed undeformed layout (left) and deformed layout (right). Location of clamping and traction illustrated by cyan and black nodes, respectively.

Table 1 illustrates our results for differing mesh parameters  $h$  and subdomains. The column labelled  $\tilde{S} = \hat{I}$  illustrates results for both test problems in the absence of interface preconditioning. By reading this column from top to bottom (for each problem), we observe a logarithmic dependence on the number of GMRES iterations for increasing mesh parameters. Reading this column from left to right also suggests a logarithmic dependence on the number of subdomains.

In comparison, the column labelled  $\tilde{S} = \hat{H}_\theta$  of the table provides results with the interface preconditioner as discussed in (9). Here, it can be seen that the number of iterations are independent of the chosen mesh parameter. Whilst there is a logarithmic dependence on the number of iterations for an increasing number of subdomains, a direct comparison with the column labelled  $\tilde{S} = \hat{I}$  suggests that our preconditioning strategy provides significant savings in the number of iterations required for convergence.

The final column labelled  $\tilde{S} = \hat{H}_{OPT}$  illustrates results for selected values of theta based on testing. It was found that the recorded values were able to provide improved results over the other two columns, suggesting that different values of theta are able to provide a closer approximation to the decay of the associated Steklov-Poincaré operator.

Table 1: Results for the test problem. Tolerance of GMRES set at  $10^{-6}$ .

| # Domains  | $\tilde{S} = \hat{I}$ |    |     | $\tilde{S} = \hat{H}_\theta$ |     |     | $\tilde{S} = \hat{H}_{OPT}$ |     |     |
|------------|-----------------------|----|-----|------------------------------|-----|-----|-----------------------------|-----|-----|
|            | 4                     | 16 | 64  | 4                            | 16  | 64  | 4                           | 16  | 64  |
| $\theta$   | -                     | -  | -   | 0.5                          | 0.5 | 0.5 | 0.5                         | 0.6 | 0.7 |
| $h = 1/32$ | 28                    | 47 | 68  | 12                           | 18  | 27  | 12                          | 17  | 22  |
| 1/64       | 41                    | 66 | 96  | 12                           | 19  | 27  | 12                          | 18  | 23  |
| 1/128      | 59                    | 93 | 137 | 12                           | 19  | 27  | 12                          | 19  | 24  |

In order to observe the computational benefits of our method, we look to provide rough estimates in order to gauge how our derived approach will perform in a parallel environment. Due to the non-overlapping nature of our approach, all subdomain solves can be carried out in parallel. As mentioned previously, the main issue surrounds the solution to the resulting interface problem. Within each application of our preconditioner to this problem, we are required to invert the discrete interface Laplacian. This issue is present in the Lanczos process, and also in the subsequent generalised eigenvalue decomposition that follows. Due to the structure of this matrix, these inversions can lead to a computational bottleneck for an increasing number of subdomains, and so we would like to consider an iterative approach to alleviate this issue.

The structure of the involved matrix suggests conjugate gradient as a suitable alternative, coupled with an appropriate preconditioning strategy (PCG). In this work, we propose to precondition by using the relevant contributions of  $L$  restricted to  $\Gamma_i$ , with the cross points removed to enable construction in parallel. The parallel CPU time taken for each GMRES iteration can then be realised by dividing the number of PCG iterations multiplied by the CPU time taken to apply the preconditioner by the total number of faces involved in the construction of  $\Gamma$ . By adding this contribution to the CPU time taken for one parallel subdomain solve, we calculate the total CPU time by multiplying the result to the total number of GMRES iterations required to achieve convergence.

The results for the investigation are displayed in Table 2 below, where CPU times (in seconds) are provided for differing mesh and subdomain sizes. A Linux machine with an Intel® Core™ i7 CPU 870 @ 2.93 GHz with 8 cores was used to obtain the data. The bracketed numbers in the table indicate the total number of GMRES iterations required to achieve convergence.

Table 2: Total CPU times (seconds) anticipated through the use of parallel computing.

|            | $\tilde{S} = \hat{H}_{OPT}$ |         |        |        |
|------------|-----------------------------|---------|--------|--------|
| # Domains  | 4                           | 16      | 64     | 256    |
| $\theta$   | 0.5                         | 0.6     | 0.7    | 0.75   |
| $h = 1/16$ | 0.0169                      | 0.0168  | 0.0176 | 0.0254 |
| 1/32       | 0.0635                      | 0.0273  | 0.0199 | 0.0232 |
| 1/64       | 0.4455                      | 0.1238  | 0.0384 | 0.0238 |
| 1/128      | 3.8716                      | 1.0804  | 0.2529 | 0.0623 |
| 1/256      | 50.2476                     | 13.2858 | 3.3204 | 0.7295 |

From the table, it can be seen that for relatively coarse meshes, we do not see a significant enough decrease in the CPU time to warrant the use of parallelism. This behaviour can be attributed to the computational complexity of sparse matrix inversion ( $O(k^2n)$ ,  $k$  is the bandwidth) for relatively small values of  $n$ , and also the efficiency of the backslash command in MATLAB. However, notable savings in CPU time equating to roughly factor 4 can be seen for finer meshes. These figures are encouraging, as they suggest that our approach is capable of significant speedup through the use of parallel architecture when compared directly to solving the problem globally on a single processor.

After collating the results in Table 2, a general increase was noted in the number of GMRES iterations when compared directly to the figures obtained in Table 1. The reason for this can be attributed to the use of inner PCG iterations. In particular, a logarithmic dependence on the mesh parameter was observed for cases involving smaller numbers of subdomains. However, the deterioration can be seen as an acceptable compromise, as the results for larger meshes suggest the use of an increasing number of subdomains for improved performance. It should be noted that the results obtained above were done so with a relatively coarse tolerance for PCG of  $10^{-3}$ .

It should not be expected that continual speedup can be gained through the use of an increasing number of subdomains, as certain factors such as inter-processor communication between each of the three steps will begin to play an important role. Therefore, in terms of a regular subdivision, this would suggest an optimal decomposition of the domain based on the mesh parameter, and also possibly other contributing factors relating to computer hardware.

## 5 Topology Optimization

As an application of our findings, we will describe how our work can be incorporated into commonly used solvers from problems arising in topology optimization. The problem we consider here is the so-called Variable Thickness Sheet problem (Bendsøe & Sigmund, 2003:p.54-57), which can be described mathematically using finite elements by the following nonlinear optimization problem

$$\begin{aligned}
 & \min_{\mathbf{u}, \boldsymbol{\rho}} && \mathbf{f}^T \mathbf{u} \\
 & \text{subject to:} && K(\boldsymbol{\rho}) \mathbf{u} = \mathbf{f}, \\
 & && K(\boldsymbol{\rho}) = \sum_{i=1}^m \rho_i K_i, \\
 & && \sum_{i=1}^m \rho_i \leq V, \\
 & && \bar{\rho} \leq \rho_i \leq \underline{\rho}, \quad i = 1, \dots, m.
 \end{aligned} \tag{10}$$

In the above, the  $n$ -vector of nodal displacement values  $\mathbf{u} = \mathbf{u}(\boldsymbol{\rho})$  denotes the solution to the elasticity equations, with  $\mathbf{f} \in \mathbb{R}^n$  representing the corresponding discretisation of the load linear form. The density  $\boldsymbol{\rho} \in \mathbb{R}^m$  is subject to upper and lower bounds  $\bar{\rho}$  and  $\underline{\rho}$  respectively, with the volume of the body being denoted by  $V$ . Additionally,  $K(\boldsymbol{\rho})$  represents the finite element stiffness matrix for the elasticity equations, with each  $K_i$ ,  $i = 1, \dots, m$  denoting elemental stiffness matrices.

By considering the method of Lagrange multipliers, minima to (10) are obtained through a nonlinear system of equations. The nonlinearities can be dealt with using a number of commonly used approaches. For instance, one could consider the use of interior point methods (Hoppe & Petrova, 2004). The fairly standard solution technique used by the community involves the consideration of fixed point type update schemes for an initial guess for the density in the following way

1. Finite Element Analysis (FEA) – solve equations of linear elasticity.
2. Density update – e.g.: OC (Bendsøe & Sigmund, 2003), MMA (Svanberg, 2002).
3. Check for convergence. If not satisfied, rerun 1 and 2 using updated density.

It can be expected that a reasonably large number of fixed point iterations are required to obtain a suitable final design. The bulk of computational effort will be concentrated on the Finite Element Analysis step, namely the repeated process of obtaining updated displacement variables through the use of the equations of linear elasticity (Borrvall & Petersson, 2001). Therefore, we propose to apply our preconditioning strategy as discussed in Section 3 to this problem coupled with the fairly straightforward Optimality Criteria (OC) method for the density update. No attempt will be made here to carry out Step 2 above in parallel; however (Borrvall & Petersson, 2001) describe an appropriate implementation using the Method of Moving Asymptotes (MMA).

In Table 3, results are provided illustrating the performance of our approach for the cantilever beam problem. The results were obtained using an adaptive tolerance for GMRES based on successive compliance values. The total number of fixed point iterations are given, along with the average number of GMRES iterations per fixed point step (bracketed). Whilst the number of fixed point iterations appears to increase for finer meshes, the average number of GMRES iterations remains roughly constant. Whilst we still see a logarithmic dependence on the average number of GMRES iterations for an increasing number of subdomains, the fixed point iterations remain roughly constant.

Table 3: Results for the cantilever beam problem solved using our preconditioning strategy for the FEA coupled with the OC method for the density update.

| # Domains  | 4       | 16      | 64      | 256     |
|------------|---------|---------|---------|---------|
| $\theta$   | 0.5     | 0.6     | 0.7     | 0.75    |
| $h = 1/16$ | 10 (10) | 10 (18) | 10 (33) | 10 (56) |
| 1/32       | 17 (11) | 17 (18) | 17 (34) | 19 (54) |
| 1/64       | 23 (11) | 23 (18) | 24 (32) | 27 (54) |
| 1/128      | 29 (12) | 30 (17) | 32 (31) | 35 (52) |
| 1/256      | 33 (13) | 36 (17) | 37 (31) | 41 (49) |

Future work involves validation of our approach on a parallel machine, as well as consideration of further problems (possibly to include 3D domains) and alternative solution methods to try to solve topology optimization problems completely in parallel. We expect our approach to adapt well in parallel, with potential speedup for 3D problems of factor 8 anticipated.

## References

- Arioli, M. and Loghin, D. (2009). Discrete Interpolation Norms with Applications. *SIAM J. Numer. Anal.*, 47(4):2924–2951.
- Bendsøe, M. P. and Sigmund, O. (2003). *Topology Optimization: Theory, Methods, and Applications*. Springer Verlag, Providence, Rhode Island.
- Borrvall, T. and Petersson, J. (2001). Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(46-47):6201–6229.
- Hoppe, R. H. W. and Petrova, S. I. (2004). Primal–Dual Newton Interior Point Methods in Shape and Topology Optimization. *Numerical Linear Algebra with Applications*, 11(5-6):413–429.
- Ipsen, I. C. F. (2002). A Note on Preconditioning Nonsymmetric Matrices. *SIAM J. Numer. Anal.*, 23(3):1050–1051.
- Quarteroni, A. and Valli, A. (1999). *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York. Oxford Science Publications.
- Saad, Y. (1993). A Flexible Inner-Outer Preconditioned GMRES Algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469.
- Saad, Y. and Schultz, M. H. (1986). GMRES: A Generalized Minimal Residual Algorithm for solving Nonsymmetric Linear Systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869.
- Svanberg, K. (2002). A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. *SIAM Journal on Optimization*, 12(2):555–573.
- Toselli, A. and Widlund, O. (2005). *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin.
- Turner, J. A. (Expected submission: 2014). Application of Domain Decomposition to problems in Topology Optimization. PhD thesis, University of Birmingham.