Efficient and easily accessible Matlab codes for topology optimization

Federico Ferrari
Department of Civil and Systems Engineering, Johns Hopkins University

Acknowledgements: Ole Sigmund, Department of Mechanical Engineering, Technical University of Denmark
Jamie Guest, Department of Civil and Systems Engineering, Johns Hopkins University
Motivation of the work

% CPU time distribution in top88\(^{(1)}\)
\[t_S: \text{linear solve} \]
\[t_A: \text{matrix assembly} \]
\[t_{OC}: \text{OC update} \]
\[t_P: \text{setup operations} \]

\(t_S\) will substantially decrease when using highly efficient preconditioned solvers.

Goals

- Cut all the times other than \(t_S\), making the code highly efficient for medium-size problems (\(10^5 - 10^6\) elements);
- Keep the readability and flexibility of top88

\(^{(1)}\) Andreassen et. al. (2011)-Efficient topology optimization in MATLAB using 88 lines of code, SMO
Overview of the code and speedups

Minimum compliance

\[
\begin{align*}
\min_{\hat{x}} c(\hat{x}) &= f^T u(\hat{x}) \\
K(\hat{x})u(\hat{x}) &= f \\
g(\hat{x}) &= \sum_{e=1}^{m} \hat{x}_e v_e - V_f|\Omega_h| \leq 0 \\
0 \leq x_e \leq 1, \quad \forall e
\end{align*}
\]

\(\hat{x}_e\): physical densities, \(x_e\): design variables

Speedups

- Stiffness matrix assembly;
- Filtering operations and OC update;
- Overall acceleration strategy

\(^{(2)}\) Ferrari, Sigmund (2020)-A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D, SMO

Testing with the MBB beam example

- \( \Omega_h = 600 \times 200, \ V_f = 0.5, \ r_{\text{min}} = 8 \)
- Test cases
  - T1: Density Filter, \( p = 3 \);
  - T2: Density Filter, \( p = 1 \rightarrow 3 \);
  - T3: Density+Proj. \( p = 3, \ \beta = 2 \rightarrow 24 \);
  - T4: Density+Proj. \( p = 1 \rightarrow 3, \ \beta = 2 \rightarrow 24 \);

\[ t_S: \text{linear solve, } t_A: \text{matrix assembly,} \]
\[ t_{OC}: \text{OC update, } t_P: \text{setup operations} \]

\[ \begin{array}{cccc}
\Omega_h & 300 \times 100 & 600 \times 200 & 1200 \times 400 \\
\hline
r_{\text{min}} = 4 & t_{\text{top99neo}}(s) & 0.231 & 1.19 & 5.69 \\
r_{\text{min}} = 8 & t_{\text{top88U}}(4) & 1.55 & 1.57 & 1.78 \\
r_{\text{min}} = 16 & t_{\text{top88}} & 2.66 & 4.09 & 5.51 \\
\end{array} \]

\( ^{(4)} \) Use of sparse2 for assembly and conv2 for filtering

4  Department of Civil and Systems Engineering - JHU

6th Top Webinar  October 29th, 2020
**Extension to 3D domains (top3D125)**

- change elemental stiffness matrix (8-nodes hexahedron element);
- change reshape operations (12 lines total);

\[ \Omega_h = 96 \times 48 \times 48, \ V_f = 0.12 \]

**Table: Average iteration time (s) and speedup factors.**

<table>
<thead>
<tr>
<th>( \Omega_h )</th>
<th>48 \times 24 \times 24</th>
<th>96 \times 48 \times 48</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{\text{min}} ) = ( \sqrt{3} )</td>
<td>1.79</td>
<td>14.20</td>
</tr>
<tr>
<td>( r_{\text{min}} ) = 2( \sqrt{3} )</td>
<td>1.78</td>
<td>1.92</td>
</tr>
</tbody>
</table>

**t\(_S\):** linear solve, **t\(_A\):** matrix assembly, **t\(_{OC}\):** OC update, **t\(_P\):** setup operations

---

\( ^{(5)} \) Amir et.al. (2014)- *On multigrid-CG for efficient topology optimization*, SMO

---

5 Department of Civil and Systems Engineering - JHU

6\(^{th}\) Top Webinar October 29\(^{th}\), 2020
Improving the stiffness matrix assembly

- Assemble only lower half $K^{(s)} \rightarrow \text{cut} \approx 45\%$ of CPU time and RAM;
- Define mesh-related indices as 'int32'(6) → RAM is cut to $\approx 1/4$ and CPU time to $\approx 1/10$;
- $\text{chol}$ and similar work with $K^{(s)}$ (not "\"!!), (CG, MINRES can be adapted);

---

\[(a)\] 2D discretization  \hspace{1cm} \[(b)\] 3D discretization

Improving the efficiency of the OC update

\[ \delta_- = \max\{0, x_{k,e} - \mu\}, \quad \delta_+ = \min\{1, x_{k,e} + \mu\} \] (local lower/upper bounds).

**Primal & dual variables updates**

Compute \((x_{k+1}, \lambda^*_k)\) by iterations

\[ x_{k+1,e} = \max\{\delta_-, \min\{\delta_+, x_{k,e}\} \left( - \frac{\partial c(x_k)}{\lambda \partial V(x_k)} \right)^{1/2} \} \]

\[ \lambda = \left( \frac{\sum_{e \in \mathcal{M}} x_{k,e} \left( - \frac{\partial c(x_k) / \partial V(x_k)}{\partial e V(x_k)} \right)^{1/2}}{g(x_k) / \partial e V(x_k) - \mathcal{L} |\delta_- - \mathcal{U} |\delta_+} \right)^2 \]

\[ \mathcal{M} = \{e \mid \delta_- < x_e < \delta_+\} \]

\[ \mathcal{L} = \{e \mid x_e = \delta_-\} \]

\[ \mathcal{U} = \{e \mid x_e = \delta_+\} \]

\(n_{bs}\): cumulative number of “bisects”
Improving the efficiency of the OC update

\[ \delta_- = \max\{0, x_{k,e} - \mu\}, \quad \delta_+ = \min\{1, x_{k,e} + \mu\} \] (local lower/upper bounds).

**Primal & dual variables updates**

Compute \((x_{k+1}, \lambda^*_k)\) by iterations

\[ x_{k+1,e} = \max\{\delta_-, \min\{\delta_+, x_{k,e}\} \left(-\frac{\partial_e c(x_k)}{\lambda \partial_e V(x_k)}\right)^{1/2}\} \]

\[ \lambda = \left(\frac{\sum_{e \in \mathcal{M}} x_{k,e}(-\partial_e c(x_k)/\partial_e V(x_k))^{1/2}}{g(x_k)/\partial_e V(x_k) - |\mathcal{L}|\delta_- - |\mathcal{U}|\delta_+}\right)^2 \]

\[ \mathcal{M} = \{e \mid \delta_- < x_e < \delta_+\} \]

\[ \mathcal{L} = \{e \mid x_e = \delta_-\} \]

\[ \mathcal{U} = \{e \mid x_e = \delta_+\} \]

Repeated filtering operations are avoided if we adopt volume-preserving filters.

| \(n_{bs}\) | 2390 | 1389 | 750 |
| \(t_{stf}(s)\) | 1.18 | 0.65 | 0.27 |
| \(t_{vpf}(s)\) | 0.04 | 0.03 | 0.03 |
Introduction of passive (solid & void) elements

Reinforcement of a solid frame while keeping a void region

- $\Omega_h = 900 \times 900$ elements, $V_f = 0.2$;
- Average cost per iteration: $10.8s \ (1.62 \cdot 10^6$ DOFs)
Extension to linearized buckling: **topBuck250**

- Fully vectorized setup of the buckling eigenproblem and buckling load factors (BLFs) sensitivity analysis;
- Includes 4 design problems by default
  1. max BLF, s.t. \( \{ \text{vol, compliance} \} \) constraints
  2. min vol, s.t. \( \{ \text{BLF, compliance} \} \) constraints
  3. min Compliance, s.t. vol constraint
  4. min vol, s.t. compliance constraint
- Explicit OC update based on MMA-like approximations;

..COMING SOON!

Topo```
Thank you for your attention

top99neo and top3D125 codes can be found at www.topopt.dtu.dk

Visit also https://www.ce.jhu.edu/topopt for upcoming news and codes