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

Johns Hopkins
Whiting School of Engineering

Motivation of the work

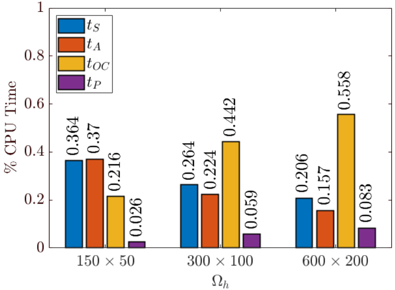
% CPU time distribution in top88[1]

t_S: linear solve

t_A: matrix assembly

t_{OC}: OC update

t_P: setup operations



(a) $r_{\min} = 4, 8, 16$, respectively

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(2)

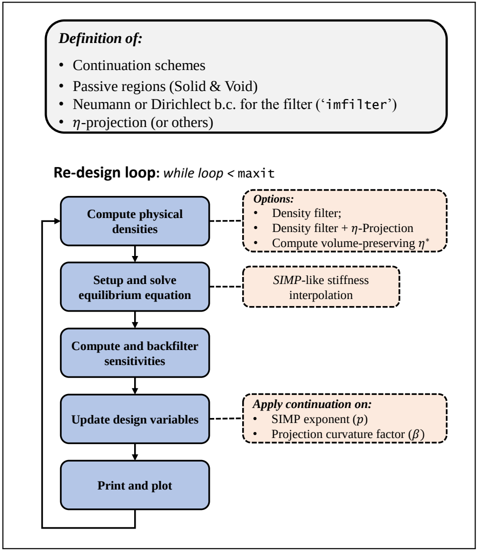
Minimum compliance

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

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

Speedups

- Stiffness matrix assembly;
- Filtering operations and OC update;
- Overall acceleration strategy(3)



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

(3) Li et al. (2020)-*Accelerated fixed-point formulation of Topology Optimization: application to compliance minimization problems*, Mech. Res. Comm.

Testing with the MBB beam example

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

(a) T1

(b) T2

(c) T3

(d) T4

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

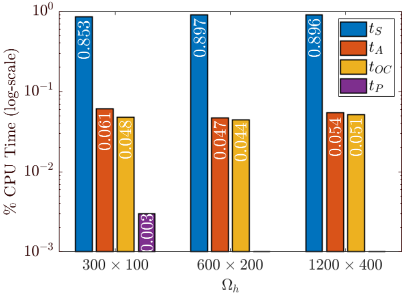


Table: Average iteration time (s) and speedup factors.

Ω_h	300×100 $r_{\min} = 4$	600×200 $r_{\min} = 8$	1200×400 $r_{\min} = 16$
top99neo(s)	0.231	1.19	5.69
top88U[4]	1.55	1.57	1.78
top88	2.66	4.09	5.51

[4] Use of sparse2 for assembly and conv2 for filtering

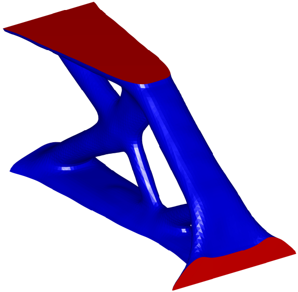
Extension to 3D domains (`top3D125`)

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

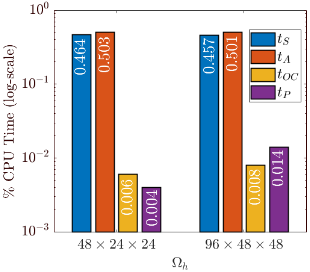
Table: Average iteration time (s) and speedup factors.

Ω_h	$48 \times 24 \times 24$ $r_{\min} = \sqrt{3}$	$96 \times 48 \times 48$ $r_{\min} = 2\sqrt{3}$
`top3D125`(s)	1.79	14.20
`top3Dmgcg`[(5)]	1.78	1.92

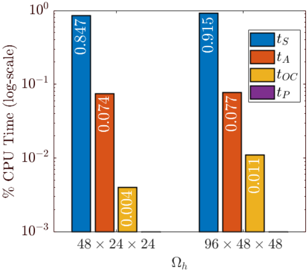
$\Omega_h = 96 \times 48 \times 48$, $V_f = 0.12$



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



(a) `top3Dmgcg`

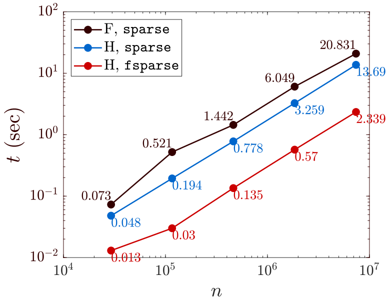


(b) `top3D125`

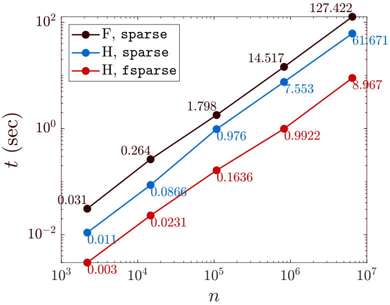
[(5)] Amir et.al. (2014)-*On multigrid-CG for efficient topology optimization*, SMO

Improving the stiffness matrix assembly

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



(a) 2D discretization



(b) 3D discretization

[6] `https://github.com/stefanengblom/stenglib`, Engblom, Lukarski (2015)-*Fast Matlab compatible sparse assembly on multicore computers*, Parallel Computing

Improving the efficiency of the OC update

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

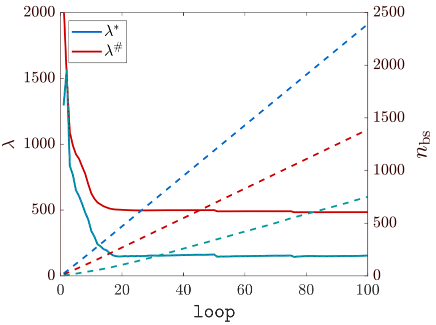
Primal & dual variables updates

Compute $(\mathbf{x}_{k+1}, \lambda_k^*)$ by iterations

$$x_{k+1,e} = \max\{\delta_-, \min\{\delta_+, x_{k,e}\left(-\frac{\partial_e c(\mathbf{x}_k)}{\lambda \partial_e V(\mathbf{x}_k)}\right)^{\frac{1}{2}}\}\}$$

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

$$\mathcal{M} = \{e \mid \delta_- < x_e < \delta_+\}$$
$$\mathcal{L} = \{e \mid x_e = \delta_-\}$$
$$\mathcal{U} = \{e \mid x_e = \delta_+\}$$



n_{bs}: cumulative number of "bisections"

Improving the efficiency of the OC update

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

Primal & dual variables updates

Compute $(\mathbf{x}_{k+1}, \lambda_k^*)$ by iterations

$$x_{k+1,e} = \max\{\delta_-, \min\{\delta_+, x_{k,e}\left(-\frac{\partial_e c(\mathbf{x}_k)}{\lambda \partial_e V(\mathbf{x}_k)}\right)^{\frac{1}{2}}\}\}$$

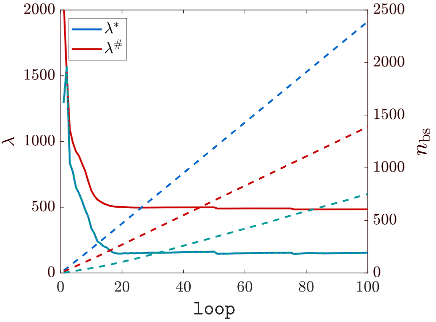
$$\lambda = \left(\frac{\sum_{e\in\mathcal{M}} x_{k,e}(-\partial_e c(\mathbf{x}_k)/\partial_e V(\mathbf{x}_k))^{1/2}}{g(\mathbf{x}_k)/\partial_e V(\mathbf{x}_k) - |\mathcal{L}|\delta_- - |\mathcal{U}|\delta_+}\right)^2$$

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

$\mathcal{L} = \{e \mid x_e = \delta_-\}$

$\mathcal{U} = \{e \mid x_e = \delta_+\}$

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



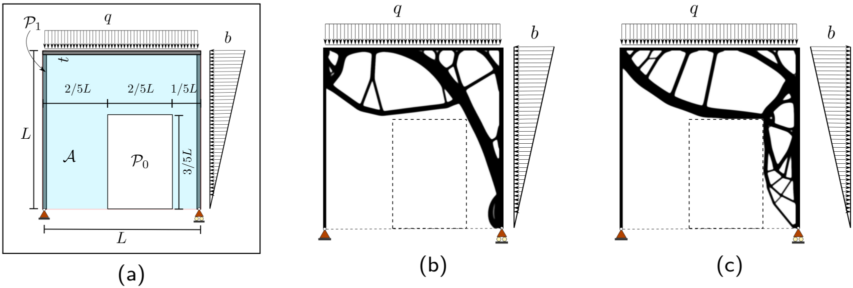
n_{bs}: cumulative number of “bisections”

	$\Lambda = [0, 10^9]$	Est.	PD
n_{bs}	2390	1389	750
$t_{\mathrm{stf}}(s)$	1.18	0.65	0.27
$t_{\mathrm{vpf}}(s)$	0.04	0.03	0.03

Introduction of passive (solid & void) elements

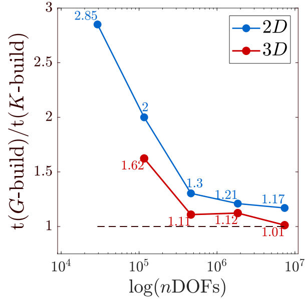
Reinforcement of a solid frame while keeping a void region

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



Extension to linearized buckling: topBuck250

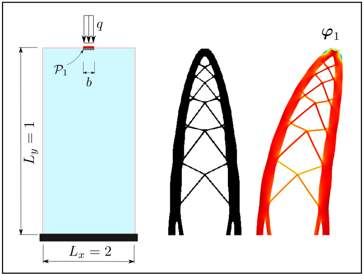
set up buckling analysis ≈ with the same cost as linear analysis



Topology Optimization with linearized buckling criteria in 250 lines of Matlab

Federico Ferrari · Ole Sigmund · James K. Guest

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



..COMING SOON!

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