User Manual for BESO79 (2D) and BESO94 (3D) MATLAB Codes

Overview

The BESO (Bi-directional Evolutionary Structural Optimization) method is an iterative approach to optimize the topology of structures by removing and adding material based on sensitivity analysis. This manual provides detailed instructions for using the MATLAB codes BESO79 (for 2D problems) and BESO94 (for 3D problems).

System Requirements

- **MATLAB Version**: MATLAB R2017a or later.
- **Memory**: At least 8 GB RAM for 3D problems (BESO94).
- **Processor**: Multi-core CPU recommended for better performance.
- **Toolboxes**: Image Processing Toolbox for 3D problems.

File Description

1. BESO79.m (2D)

- Implements 2D structural topology optimization using the BESO method.
- Includes stiffness matrix generation, finite element analysis, sensitivity filtering, and design update.

2. BESO94.m (3D)

- Implements 3D structural topology optimization using the BESO method.
- Extends the functionality of BESO79 to three dimensions.

Inputs and Parameters

General Inputs (Both 2D and 3D)

Input Parameter	Description	Recommended Value
nelx	Number of elements along the x-direction	≥ 0
nely	Number of elements along the y-direction	≥ 0
nelz	Number of elements along the z-direction (for 3D only)	≥ 0
vol_con	Target volume fraction for the structure	0–1
er	Evolution rate for material addition/removal	0.005–0.05
rmin	Filter radius for sensitivity analysis	> 1
E0	Young's modulus of material	1
Emin	Minimum stiffness for void elements	1e-9
nu	Poisson's ratio	0.3

Boundary Conditions and Loads

Users must define boundary conditions and loads in the script:

- F: Load vector. (Line 25 in 2D, Line 41 in 3D)
- fixeddofs: Degrees of freedom that are fixed. (Line 27 in 3D, Line 42 in 3D)
- freedofs: Automatically determined as the remaining degrees of freedom.

Material Properties

- The material properties are isotropic and linear elastic by default.

Outputs

Output	Description
x	Final topology design (density matrix).
c	Objective function history (compliance values for each iteration).
vol	Volume fraction history over iterations.
change	Change in objective function between iterations.

Running the Codes

1. Setup

- Open the MATLAB script (BESO79.m or BESO94.m).
- Modify the input parameters at the beginning of the script.
- Define boundary conditions and loads in the following section.

2. Execution

- Run the script by typing the name of the file in the MATLAB command window:

  >> BESO79

  or

  >> BESO94

3. Visualization

- During execution, the code will display the iterative evolution of the topology.
- Final results are saved in the workspace for further post-processing.

Customization

1. Filter Radius (rmin)

- Controls the smoothing of sensitivity. Larger values produce simpler designs but may slow convergence.

2. Evolution Rate (er)

- Controls the speed of material addition/removal. Smaller values improve stability but increase runtime.

3. Objective Function

- The default objective function is compliance minimization. Modify this by editing the sensitivity analysis and objective evaluation sections.

Notes and Tips

1. **Convergence:**
    - Monitor the change value to ensure convergence. The default stopping criterion is change < 0.001.
2. **Performance:**
    - Use sparse matrices and efficient solvers to handle large problems.
    - Decrease rmin or mesh density (nelx, nely, nelz) if the process is too slow.
3. **Boundary Conditions:**
    - Ensure proper boundary conditions are applied to avoid singular matrices.
4. **Recommendations:**
    - Due to the robustness of compliance problems, FEA computational precision can be reduced to improve algorithm efficiency.
    - Consider using the fsparse function to accelerate matrix assembly. (https://github.com/stefanengblom/stenglib)
    - The BESO algorithm also has well-developed implementations in C and Python. Refer to the Ameba software for high-performance alternatives. (https://ameba.xieym.com/)

Troubleshooting

Issue	Possible Cause	Solution
Singular matrix error	Improper boundary conditions	Check and revise boundary condition setup.
Slow convergence	Small er or large problem size	Increase er or decrease mesh density.
Checkerboard problem in topology	Small rmin or improper filtering	Increase rmin for smoother results.

References

[1] Y.M. Xie, G.P. Steven, A simple evolutionary procedure for structural optimization, Computers & Structures, 49 (1993) 885–896.

[2] X. Huang, Y.M. Xie, Evolutionary Topology Optimization of Continuum Structures: Methods and Applications, Wiley, Chichester, 2010.

Contact

For questions or further assistance, contact the code developers.

zhuanginhongkong@outlook.com (Zicheng Zhuang)

zicheng.zhuang@rmit.edu.au (Zicheng Zhuang)

mike.xie@rmit.edu.au (Yi Min Xie)

End of Manual

CISM@2025