

## ASSEMBLY-FREE BUCKLING ANALYSIS FOR TOPOLOGY OPTIMIZATION

Xiang Bian<sup>1</sup>

<sup>1</sup>Department of Mechanical Engineering,  
Northwestern Polytechnic University, Xi'an, China

Praveen Yadav<sup>2</sup>, Krishnan Suresh<sup>2</sup>

suresh@engr.wisc.edu  
<sup>2</sup>Department of Mechanical Engineering  
UW-Madison, Madison, Wisconsin 53706, USA

### ABSTRACT

Linear buckling analysis entails the solution of a generalized eigenvalue problem. Popular methods for solving such problems tend to be memory-hungry, and therefore slow for large degrees of freedom.

The main contribution of this paper is a low-memory assembly-free linear buckling analysis method. In particular, we employ the classic inverse iteration, in conjunction with an assembly-free deflated linear solver. The resulting implementation is simple, fast and particularly well suited for parallelization. The proposed method is used here to solve large scale 3D topology optimization problems subject to buckling constraints, where buckling problems must be solved repeatedly.

### INTRODUCTION

Buckling is the sudden failure of a structural member to carry compressive load. For example, Figure 1 illustrates the classic buckling of a pinned-pinned beam. Structural elements, such as those found in high-rise buildings are typically subjected to compressive loads, and must be analyzed and designed to prevent such buckling failures.

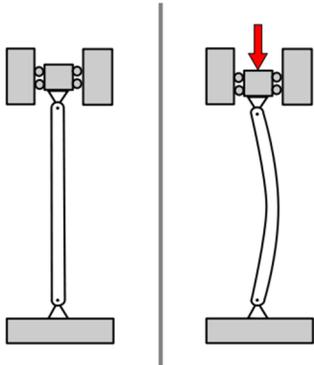


Figure 1 BUCKLING OF A PINNED-PINNED BEAM.

Finite element analysis of linear buckling is typically carried out in two stages. In the first stage, the structural member is subject to a unit load. A finite element mesh of the domain is constructed, and the corresponding static linear-elasticity problem is posed and solved; this amounts to solving a linear system of equations:

$$Kd = f \quad (1)$$

where  $K$  is the stiffness matrix that is sparse and positive definite [1].

In the second stage, the linear displacement field  $d$  is post-processed to obtain the stress tensor within each of the finite elements [1]:

$$\sigma_{elem} = \begin{Bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{Bmatrix}_{elem} \quad (2)$$

Then the stress tensor is used to define an element-level stress stiffness matrix [1]:

$$K_{\sigma}^{elem} = \int G^T \begin{bmatrix} \sigma_{elem} & & \\ & \sigma_{elem} & \\ & & \sigma_{elem} \end{bmatrix} G dV \quad (3)$$

where  $G$  is shape function gradient matrix described in [1]. This is then assembled to construct the global stress stiffness matrix [1]:

$$K_{\sigma} = \prod_{assemble} (K_{\sigma}^{elem}) \quad (4)$$

Finally, the following generalized eigenvalue problem is posed and solved [1]:

$$(K + \lambda K_{\sigma})w = 0 \quad (5)$$

While there are multiple pairs of solutions to the above problem, only the lowest few are typically important. In particular, the lowest eigenvalue of Eqn. (5) determines the buckling safety factor [1], i.e., the load at which buckling will occur (assuming a unit load has been applied initially). The vector  $w$  in Eqn. (5) represents the associated buckling mode.

The computational bottleneck in buckling analysis lies in solving Eqn. (1) and Eqn. (5). Typical methods to solve these two equations, in particular Eqn. (5), are reviewed in Section 2. Briefly, while there are well established methods, they tend to be memory-hungry, leading to high computational costs for problems with large degrees of freedom. This is further exacerbated in applications such as buckling constrained topology optimization, where one must solve buckling

problems repeatedly. Further, in topology optimization, the domain must necessarily be discretized using a large number of elements [2].

In this paper, we propose a simple inverse iteration driven, assembly-free deflated conjugate-gradient method for solving Eqn. (5); this is described in Section 3. Then, in Section 4, we use the proposed method to solve buckling constrained topology optimization problems. Conclusions and future work are summarized in Section 5.

## LITERATURE REVIEW

### Buckling Analysis

As stated earlier, linear buckling analysis entails solving Eqn. (1) followed by Eqn. (5). We focus here on methods to solve Eqn. (5). Observe that this equation is similar to the generalized eigenvalue problem associated with modal analysis as described by:

$$(K - \lambda M)w = 0 \quad (6)$$

Thus many of the methods developed for modal analysis (for example, block-Lanczos) can be used to solve Eqn. (5) as well; see [3],[4],[5],[6]. However, there are three differences between Eqns (5) and (6) that should be noted:

- (1) The mass matrix  $M$  is positive definite, but the stress matrix  $K_\sigma$  need not be positive definite.
- (2) The mass matrix depends only on the material and the underlying mesh, while the stress matrix depends on the stresses as well. This has implications in assembly-free analysis (to be discussed in a later section).
- (3) From (2), it follows that the sensitivity expressions for modal and buckling are quite different (to be discussed in a later section).

One of the established methods for solving the generalized buckling eigenvalue problem is the block-Lanczos algorithm [3],[5] that requires repeated solution of a linear system of equations where the matrix is a linear combination of  $K$  and  $K_\sigma$  that is determined dynamically.

Since both  $K$  and  $K_\sigma$  are large, and since the linearly combined matrix is constantly changing, explicit factorization can be expensive. Alternative strategies use preconditioned iterative solver. However, these can be slow to converge, while accuracy is severely compromised with early termination [3], [7]. Alternatively, computing an approximate inverse over Krylov sub-space has been proposed in [7].

Other algorithms for solving Eqn. (5) include ‘locally optimal block preconditioned conjugate gradient’, ‘Davidson-Jacobi’, etc. that have been demonstrated to be competitive for large-scale eigenvalue problem in [3]. One such algorithm is the subspace-augmented Rayleigh-Ritz conjugate gradient (RCG) that exploits the assembly-free aspect presented for solving linear systems [8]. While RCG is efficient for large-scale *modal* analysis, as discussed in [9], it cannot be applied here effectively for reasons described later in Section 3.

### Buckling Constrained Optimization

Topology optimization is a systematic method of generating designs to meet specific engineering requirements. It has numerous applications including optimization of aircraft components [10], [11], spacecraft modules [12], automobiles components [13], and compliant mechanisms [14]–[17].

In many of these applications, buckling failure must be accounted for during topology optimization [18], leading to a buckling-constrained topology optimization problem:

$$\begin{aligned} \text{Min} & |\Omega| \\ \Omega & \in D \\ J & \leq J_{\max} \\ \sigma & \leq \sigma_{\max} \\ \lambda_c & \geq \lambda_{\min} \end{aligned} \quad (7)$$

Where

- $\Omega$  : domain of objective topology
- $D$  : allowable design space
- $J$  : compliance of structure
- $J_{\max}$  : maximum allowable compliance
- $\sigma$  : maximum von-mises for current design (8)
- $\sigma_{\max}$  : maximum allowable von-mises stress
- $\lambda_c$  : critical buckling load
- $\lambda_{\min}$  : minimum buckling load

Different topology optimization methods have been proposed to solve such problems, including Solid Isotropic Material with Penalization (SIMP), evolutionary and level-set.

SIMP uses pseudo-densities assigned to elements, and they vary between 0 and 1. These pseudo-densities are then used as continuous relaxation parameter [19]. However, when continuous relaxation method is used in the context of buckling modes, undesirable numerical effects are observed [20], Pedersen [21] and Neves et al. [22] discuss the spurious modes computed in such methods. They consider assigning zero stiffness to such elements to overcome these issues, but this results in inconsistencies in the model.

The variability in densities from element to element also causes ill-conditioning of the stiffness matrices [23],[24]. Additionally for stress related analysis, the accuracy over gray elements is poor.

As an alternative to SIMP, a free material optimization (FMO) was proposed in [25]. FMO considers the entire stiffness tensor as a continuous design variable. The sensitivity computation for compliance and stress field becomes more expensive. A binary programming method is discussed in [26], where the bottleneck is in computing the derivatives of buckling constraints.

The other strategy for solving buckling-constrained topology optimization relies on defining the evolving topology through a level-set [27]. Level-set allows the domain to be well defined at all times; thus overcoming the issue of ill-conditioned stiffness matrices.

Numerous examples are provided in the literature to illustrate the effectiveness of the level set based methods [28],[29],[30]. In this paper, we extend the formulation for buckling-constraints in a semi-analytical manner. In section 4, we discuss the technical aspects of computing such level-set.

## ASSEMBLY-FREE BUCKLING ANALYSIS

In the present paper, an accelerated buckling finite element analysis is developed by implementing and merging four distinct but complementary concepts (see Figure 2). Each

of these concepts is discussed in the following sections, but briefly:

1. **Assembly-Free:** Assembly-free finite element analysis was proposed by Hughes and others in 1983[31], but has resurfaced due to the surge in fine-grain parallelization. The basic concept here is that the stiffness matrix is never assembled; instead, the fundamental matrix operations such as the Sparse-Matrix Vector multiplications (SpMV) are performed in an assembly-free elemental level as:

$$Kw = \prod_{assemble} (K_e w_e) \quad (9)$$

Assembly-free SpMV is particularly advantageous if memory foot-print can be reduced by storing limited data. Exploiting element congruency helps reduce memory footprint [8]. Secondly, assembly-free iterative analysis is effective only if an assembly-free acceleration/preconditioning can be exploited; here we rely on assembly-free deflation, discussed below.

2. **Voxelization:** Voxelization is a special form of spatial discretization where all finite elements are identical hexahedral elements; the most important benefits of voxelization is meshing-robustness and significantly reduced memory foot-print, especially in conjunction with assembly-free analysis. This ensures a faster SpMV through parallel implementation on multi-core architectures.
3. **Deflation:** Deflation is a powerful acceleration technique for conjugate gradient [32], and is more amenable to an assembly-free implementation, than classic preconditioners such as incomplete Cholesky. The particular method of deflation exploited in this paper is based on rigid-body agglomeration discussed in [33]. The rigid-body agglomeration has a simple assembly-free implementation and offers significant advantage in parallel computing [8].
4. **Inverse Iteration:** Focusing specifically on buckling problem, we propose the use of inverse iteration for reasons discussed later in the paper.

Finally, given the above infrastructure, fine-grain parallelization is achieved in this paper on multi-core CPUs using OpenMP.

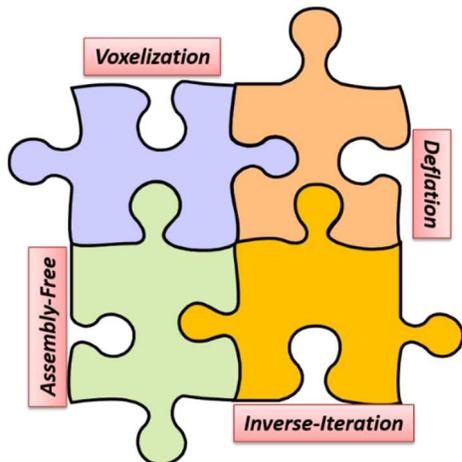


Figure 2: OVERVIEW OF PROPOSED METHOD.

## Voxelization

The proposed method of assembly-free deflated conjugate-gradient is applicable to any finite-element discretization. However, in this paper, we consider a simple discretization, where the geometry is approximated via uniform hexahedral elements or ‘voxels’; the voxel-approach has gained significant popularity recently due to its robustness and low memory foot-print. The voxelization of a complex geometry is illustrated in Figure 3; it has over 300,000 elements. Fortunately, even such a large-sized problem is easily handled via the proposed method.

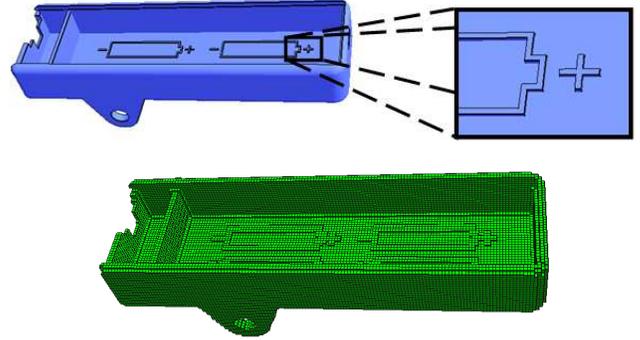


Figure 3: VOXELIZATION OF BATTERY HOLDER.

The voxelization process is straight-forward, and is discussed, for example, in [34]. The most significant benefits of voxelization are: (1) it is robust in that it rarely fails (unlike traditional meshing), (2) the mesh storage is compact, (3) computational cost of voxelization is usually negligible and is relatively insensitive to geometric complexity, and (4) it promotes assembly-free-analysis.

## Assembly-Free Deflation for Static Analysis

The first step in assembly-free buckling analysis is solving Eqn. (1). This is accomplished here using the deflated conjugate gradient method discussed in [8]. Deflated conjugate gradient uses several different agglomeration groups described in [8] to accelerate the solver. The solution of Eqn. (1) yields the displacement and stress fields.

## Rayleigh Ritz CG for Buckling Analysis

As discussed in the literature review, the generalized eigenvalue problem for buckling is similar to modal analysis. Therefore, based on our earlier work [9], we attempted to use Rayleigh-Ritz CG (RCG) algorithm that requires repeated operations  $Kw$  and  $K_\sigma w$ .

Unfortunately, RCG is efficient only if we can exploit congruency and limit the number of unique elements for both operations. In the case of stiffness matrix  $K$ , one can certainly exploit the congruency in the voxel mesh.

However, for the stress matrix  $K_\sigma$ , each element’s stress stiffness depends on its own stress tensor. Exploiting congruency is therefore not an option. Furthermore, storing every element stress stiffness matrix  $K_\sigma$  will create a large memory footprint. This was observed to significantly slow down the computation.

## Inverse Iteration

This draws attention towards another method known as inverse iteration [35]. The basic principle is to carry out:

$$y = -K^{-1}K_\sigma w \quad (10)$$

and recycle the solution. The number of  $K_\sigma w$  operations is considerably reduced, and the computational burden falls on solving an equivalent static problem [8].

Using inverse iteration, the algorithm to solve Eqn. (11) is thus:

1. Initialize  $w^{(1)} \neq 0$  such that  $\|w^{(1)}\| = 1$
2. Set  $i = 1$
3. Compute  $z^{(i)} = K_\sigma w^{(i)}$
4. Solve  $Ky^{(i+1)} = z^{(i)}$  for  $y^{(i+1)}$
5. Update  $w^{(i+1)} = y^{(i+1)} / \|y^{(i+1)}\|$
6. Compute  $g^{(i)} = Kw^{(i+1)} + K_\sigma w^{(i+1)}$
7. If  $\|g^{(i)}\| \leq \varepsilon$ , terminate; else, increment  $i$ , and go to step 3

Once the algorithm converges to a mode shape  $w$ , the eigenvalue can be computed through:

$$\lambda = \frac{w^T K w}{w^T K_\sigma w} \quad (11)$$

The number of iterations required to converge to the mode shape is far smaller than RCG as the numerical error is primarily eliminated in the linear solution in step 4.

The numerical results in Section 5 illustrate the advantage of using assembly-free inverse iteration with DCG for buckling analysis. An efficient buckling analysis creates an opportunity to apply buckling constraints during topology optimization. We discuss the formulation in the next section.

## TOPOLOGY OPTIMIZATION

Consider the structural problem illustrated in Figure 4.



Figure 4: A STRUCTURE SUBJECT TO COMPRESSIVE LOAD.

The sensitivity expression for buckling is derived, for example, in [36], and is given by:

$$\lambda' = -\frac{w^T (K' - \lambda K_\sigma') w}{w^T K_\sigma w} \quad (12)$$

It is assumed that the eigenvectors have been  $K_\sigma$  orthogonalized, such that

$$-w^T K_\sigma w = 1 \quad (13)$$

Thus:

$$\lambda' = w^T (K' - \lambda K_\sigma') w \quad (14)$$

Unlike SIMP methods, where the sensitivity is obtained with respect to pseudo-density variables, here the sensitivity is

discrete addition and subtraction of element; for example, the discrete sensitivity of the stiffness matrix to element deletion is given by:

$$K' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -k_e & 0 \\ 0 & 0 & 0 \end{bmatrix}_{N \times N} \quad (15)$$

where  $k_e$  is the elemental stiffness matrix. The second part of the sensitivity can be simplified using the adjoint as per the derivation presented in [36]. While the derivative is computed in [36] with respect to element density, the same can be extended for a discrete element variable. The element-by-element sensitivity can be projected to the nodes to obtain a continuous field.

## Topological Level-set

A straightforward approach to exploit topological sensitivity, is to use the information to delete elements with lower sensitivity values. However, this method would lead to same issues of creating checker board pattern and instability in the mesh. Sensitivity field, however, can be used as a level-set [2]. Consider an example of a topological sensitivity field illustrated in Figure 5.

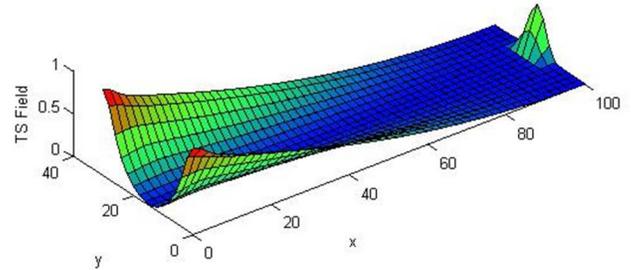
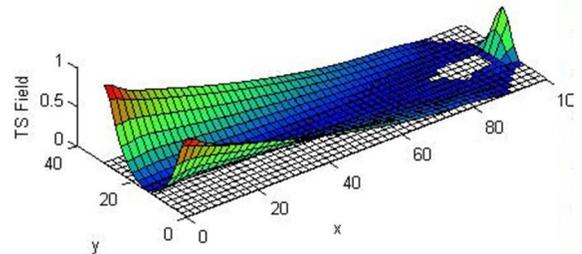


Figure 5: THE SENSITIVITY FIELD.

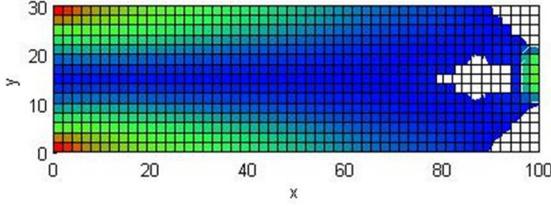
Given the field and a cutting manifold corresponding to an arbitrary cut-off value  $r = 0.03$ , one can define a domain  $\Omega^r$  per:

$$\Omega^r = \{p | T_j(p) > r\} \quad (16)$$

This will determine the set of points with sensitivity values greater than an arbitrary value of  $r = 0.03$ . The resulting domain is illustrated in Figure 6.



(a) Topological Sensitivity field with cutting manifold



(b) Domain  $\Omega^r$  for  $r = 0.03$

Figure 6: SENSITIVITY FIELD AS LEVEL SET.

The sensitivity field provides a direct ‘pseudo-optimal’ domain for a specific volume reduction that can be determined by cutting-manifold. The computed domain, however, may not be optimal [2], i.e., it may not be the best possible design for objective function with given volume fraction. Reducing the volume fraction may change the sensitivity field and therefore, one must repeat the following steps: 1) solve the finite element problem over  $\Omega$  2) re-compute the sensitivity field, and 3) reset the cutting-manifold for desired volume fraction.

Once convergence has been achieved for the desired volume fraction, one can move forward with the next step of volume reduction, repeating the above process.

### Algorithm

Typically, the sensitivity field is well defined for an unconstrained problem. When constraints are involved, the sensitivity field of the objective (compliance) must be combined with those of the constraints (in this case, buckling) through weighting factors. The weighting factors are determined along the lines described in [37]. The complete algorithm is described below.

1. The allowable domain is initialized and discretized.
2. The initial FEA requires a static solve and a buckling modal analysis by solving Eqn. (1) and (5) Hence, FEA would refer to solving both equations.
3. Based on the FEA, topological sensitivity for the objective (compliance or p-norm von Mises stress) and buckling are computed.
4. Based on proximity to imposed constraints, weight parameters (multipliers) are computed as described in [37].
5. The desired volume fraction is used to determine the cutting-manifold.
6. FEA is used to compute the constraints parameter.
7. If constraints are met, we return to step 3 and repeat the process. Else, volume fraction is reduced and we return to step 5.

Figure 7 illustrates the algorithm described above.

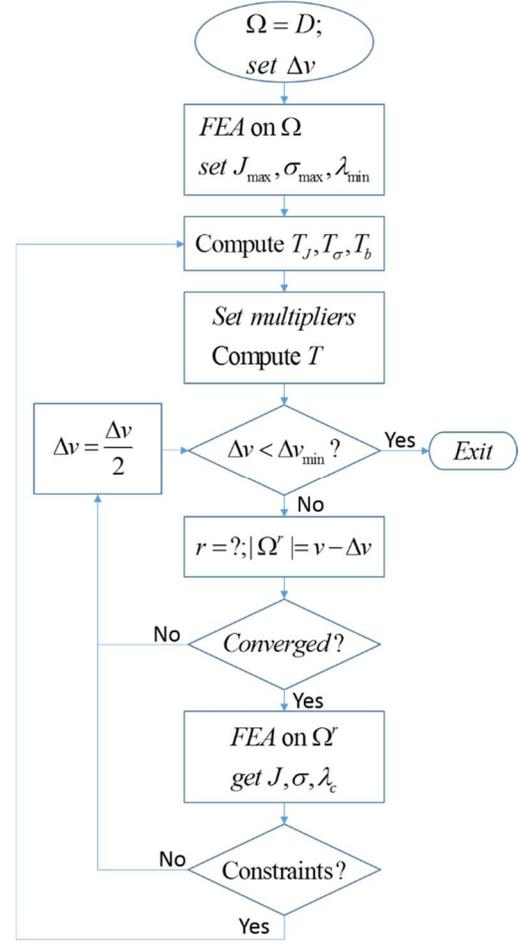


Figure 7: PROPOSED ALGORITHM.

## NUMERICAL RESULTS

In this section, we compare the results of buckling analysis using the proposed method, against those obtained through SolidWorks [38]. The material properties for all examples are those of steel with  $E = 2.1 \times 10^{11}$  Pa and  $\nu = 0.33$ .

### Buckling of a Square Beam

The first example is that of a beam of 1 meter in length, and 10 mm by 10 mm cross-section. The beam is fixed at one end, and a compressive unit load is applied at the other. The classic fixed-free Euler-beam analysis yields a critical load of

$$P_{cr} = \frac{\pi^2 EI}{(2L)^2} = 431.8 \quad (17)$$

The results obtained through the proposed Assembly-Free Buckling Analysis (AFBA) and those obtained from SolidWorks using the same number of degrees of freedom (DOF) are illustrated in Figure 8. Both methods converge to a critical load of 430.03. Note that we do not expect 3-D FEA results to converge to the exact Euler-buckling result in Eqn. (17) however, we do expect similar results.

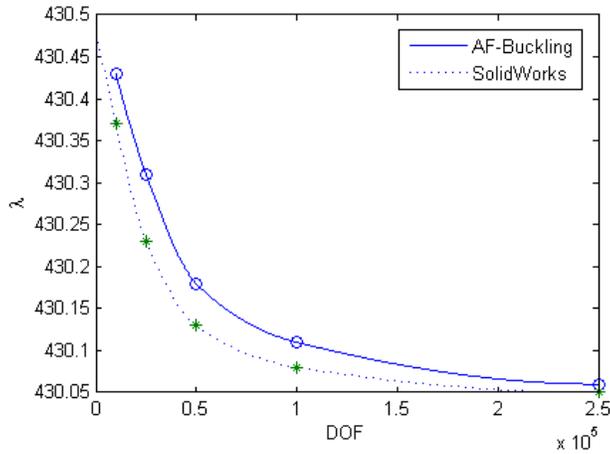


Figure 8: PREDICTED CRITICAL LOAD USING PROPOSED AFBA AND SOLIDWORKS.

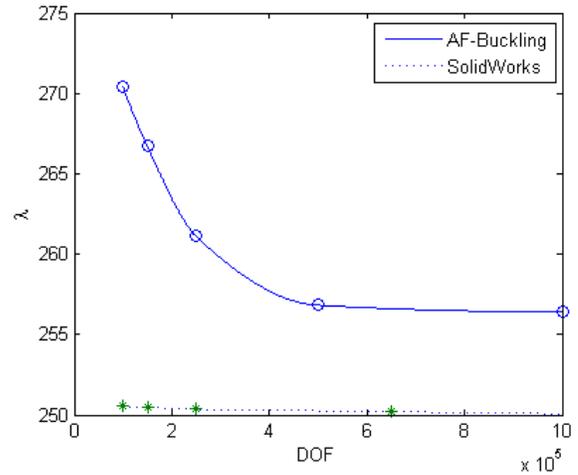


Figure 10: ACCURACY PLOT FOR CYLINDRICAL COLUMN.

The real advantage of AFBA is in speed. Figure 9 illustrates the computing time for AFBA versus SolidWorks. The quadratic growth in computation in SolidWorks can be attributed to the quadratic growth in memory consumption with increasing degrees of freedom.

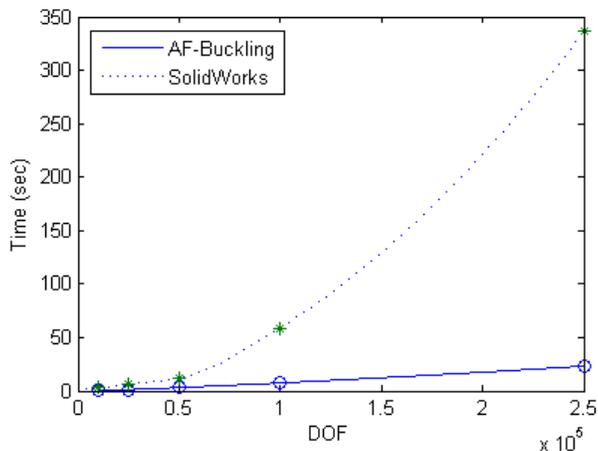


Figure 9: COMPUTING TIME FOR AFBA AND SOLIDWORKS.

The time taken to solve the problem follows a similar trend as illustrated in Figure 11. Thus, if one can tolerate a few percent error, the voxelized AFBA method can be significantly faster.

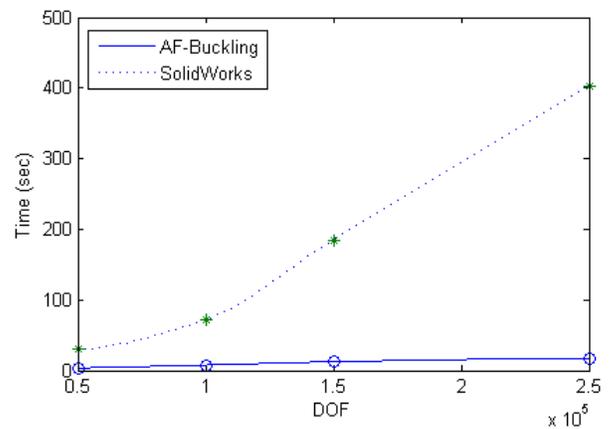


Figure 11: COMPUTING TIME VS DOF FOR CYLINDRICAL COLUMN

### Buckling Analysis of Cylindrical Column

To illustrate the potential deficiency of AFBA, we consider an example of a circular cylinder of 1 meter in length, and a diameter of 10 mm. The classic fixed-free Euler-beam analysis yields a critical load of

$$P_{cr} = \frac{\pi^2 EI}{(2L)^2} = 254.3 \quad (18)$$

The predicted buckling loads are illustrated in Figure 10. The two results differ by 2.5%. The difference can be attributed to the voxelization in AFBA. Local stress variation in the voxelized mesh is an issue that will be addressed in the future.

However, for topology optimization, the relative sensitivity of the buckling field is of greater importance than the accuracy of the field *per se*. Further, in topology optimization, the domain must necessarily be discretized using a large number of elements [2], and thus speed becomes an important issue.

### Buckling of a Rectangular Column with a hole

In this example, we consider the structure shown in Figure 12. The dimensions of the column are 5x30x100 (mm), and the hole is of diameter 10 mm.

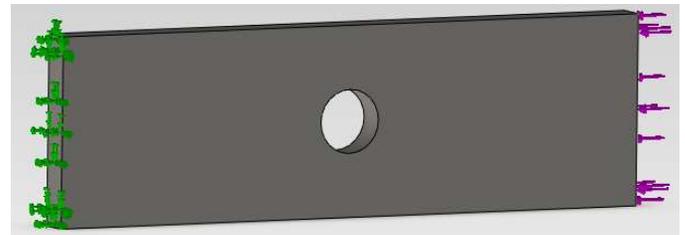


Figure 12: RECTANGULAR COLUMN WITH HOLE.

The results for the load factor computed using different mesh sizes are plotted in Figure 13. Here we observe a 0.3% error in the solution. The computation time is plotted in Figure 14.

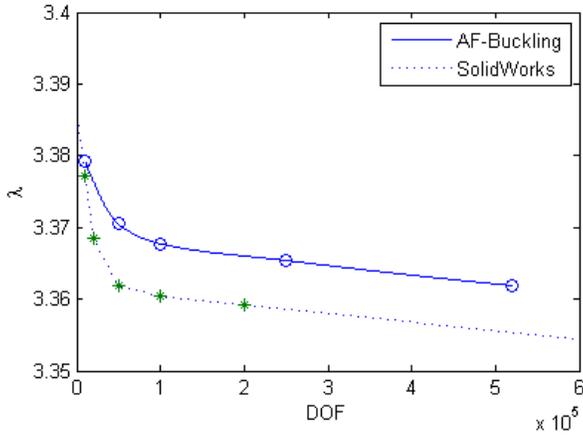


Figure 13: PREDICTED CRITICAL LOAD USING PORPOSED METHOD AND SOLIDWORKS FOR RECTANGULAR BEAM WITH HOLE.

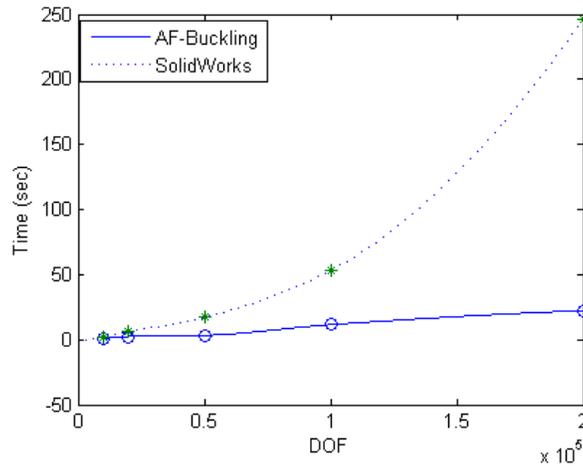


Figure 14: COMPUTING TIME FOR RECTANGULAR BEAM WITH HOLE

### Optimizing a Thin Column

We now consider minimizing the volume of a thin column with compressive load, illustrated in Figure 15. Specifically, the objective is to solve the topology optimization problem:

$$\begin{aligned}
 & \underset{\Omega \in D}{\text{Min}} |\Omega| \\
 & J \leq 5J_0 \\
 & \sigma \leq 5\sigma_0 \\
 & \lambda_c \geq (\text{SF})\lambda_0
 \end{aligned} \tag{19}$$

In other words, the maximum allowable von Mises stress and compliance is 5 times their initial values, respectively. For the buckling constraint, a safety factor (SF) was prescribed with respect to the initial buckling load.

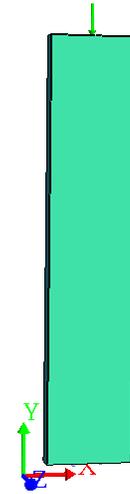
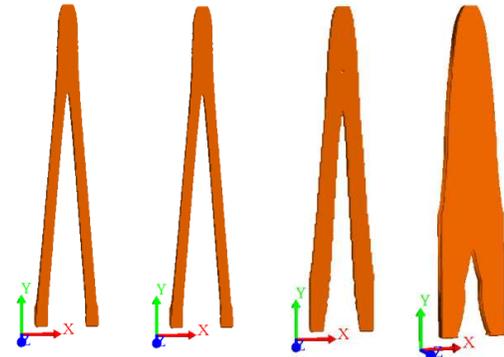


Figure 15: THIN COLUMN WITH COMPRESSIVE LOAD

The structure was voxelized with 500,000 DOF, and the time taken for buckling analysis was 46 sec. As the safety factor (SF) is increased in Eqn. (19), the buckling constraint begins to dominate, resulting in topologies illustrated in Figure 16.



a) No buckling constraint; b) SF= 1.1; c) SF = 1.5; d) SF = 2  
Figure 16: STIFF DESIGNS WITH DIFFERENT SAFETY FACTORS

Further, as the safety factor is increased, the optimization terminates at a higher volume fraction (see Table 1), as expected.

Table 1: MINIMIZING VOLUME FOR STIFF STRUCTURE

Prescribed S.F.	Final Volume Fraction	Time (in min)	#FEA
0	0.3	15	64
1.1	0.3	38.3	86
1.5	0.42	41.5	98
2	0.52	24	74

### Optimizing a Thin Plate

The structural problem considered next is illustrated in Figure 17. The plate is of dimensions 100x100x10 (mm); the lower face is fixed while a uniform load is applied on the top. The topology optimization problem is defined per Eqn. (19).

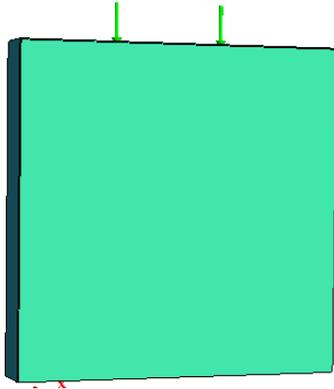


Figure 17: PLATE WITH COMPRESSIVE LOAD.

The structure was again voxelized with 500,000 DOF. The time taken for one FEA to run including buckling analysis was 7.1 seconds.

With various buckling safety factor imposed, the resulting topologies are illustrated in Figure 18. As one can observe, as the safety factor is increased, additional ribs are introduced.

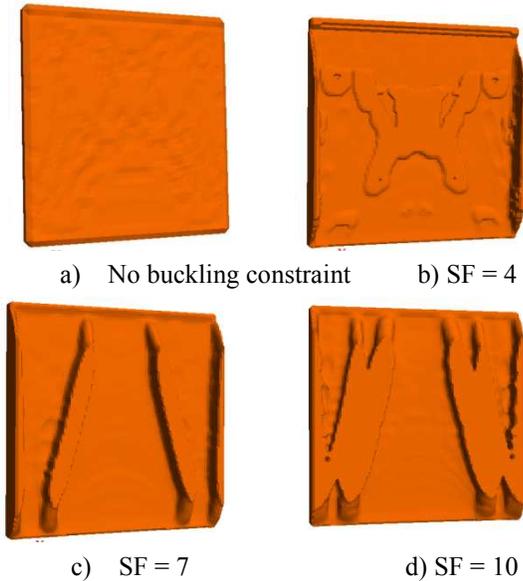


Figure 18: OPTIMIZED TOPOLOGIES FOR VARIOUS SAFETY FACTORS.

The final volume fractions and time taken are summarized in Table 2. Once again, as the buckling safety factor is increased, the optimization process converges to a higher volume fraction, as expected.

Table 2: OPTIMIZING PLATE WITH BUCKLING CONSTRAINTS.

Buckling Safety Factor	Time (in seconds)	Volume fraction	#FEA
No constraint	300	0.3	61
4	432	0.35	98
7	332	0.42	85
10	230	0.61	62

## CONCLUSION AND FUTURE WORK

In this paper, we proposed an assembly-free buckling analysis (AFBA) method that is well suited for computationally intensive tasks such as topology optimization. Future work will focus on post-buckling analysis that is critical for topology optimization.

## REFERENCES

- [1] R. D. Cook, *Concepts and Applications of Finite Element Analysis*. New York: John Wiley, 1989.
- [2] K. Suresh, "Efficient Generation of Large-Scale Pareto-Optimal Topologies," *Struct. Multidiscip. Optim.*, vol. 47, no. 1, pp. 49–61, 2013.
- [3] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro, "A Comparison of Eigensolvers for Large-scale 3D Modal Analysis using AMG-Preconditioned Iterative Methods," pp. 204–236, 2005.
- [4] Y. Saad, *Numerical methods for large eigenvalue problems*. Manchester University Press.
- [5] R. G. Grimes, J. G. Lewis, and H. D. Simon, "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," p. 228, 1994.
- [6] D. C. Sorensen, "Numerical methods for large eigenvalue problems," pp. 519–584, 2002.
- [7] G. H. Golub and Q. Ye, "An Inverse free Preconditioned Krylov Subspace method for Symmetric Generalized Eigenvalue problems," pp. 312–334, 2002.
- [8] P. Yadav and K. Suresh, "Large Scale Finite Element Analysis Via Assembly-Free Deflated Conjugate Gradient," *J. Comput. Inf. Sci. Eng.*, vol. 14, no. 4, pp. 041008–041008, Oct. 2014.
- [9] K. Suresh and P. Yadav, "Large-Scale Modal Analysis on Multi-Core Architectures," presented at the IDETC/CIE 2012, Chicago, IL, 2012.
- [10] E. Kessler, "Multidisciplinary design analysis and multi-objective optimisation applied to aircraft wing," *WSEAS Trans. Syst. Control Cybern.*, vol. 1, no. 2, p. 221–227, 2006.
- [11] J. J. Alonso, "Aircraft design optimization," *Math. Comput. Simul.*, vol. 79, no. 6, pp. 1948–1958, 2009.
- [12] V. H. Coverstone-Carroll, "Optimal multi-objective low-thrust spacecraft trajectories," *Comput Methods Appl Mech Eng*, vol. 186, pp. 387–402, 2000.
- [13] L. Wang, "Automobile body reinforcement by finite element optimization," *Finite Elem. Anal. Des.*, vol. 40, no. 8, pp. 879–893, 2004.
- [14] G. K. Ananthasuresh, S. Kota, and Y. Gianchandani, "A methodical approach to the design of compliant micromechanisms," in *Solid State Sensor and Actuator Workshop*, 1994, pp. 189–192.
- [15] S. Nishiwaki, "Topology Optimization of Compliant Mechanisms using the Homogenization Method," *Int. J. Numer. Methods Eng.*, vol. 42, pp. 535–559, 1998.
- [16] T. E. Bruns and D. A. Tortorelli, "Topology optimization of non-linear elastic structures and compliant mechanisms," *Comput. Methods Appl. Mech. Eng.*, vol. 190, no. 26–27, pp. 3443–3459, 2001.
- [17] Z. Luo, "Compliant mechanism design using multi-objective topology optimization scheme of continuum structures," *Struct. Multidiscip. Optim.*, vol. 30, pp. 142–154, 2005.

- [18] M. M. Neves, H. Rodrigues, and J. M. Guedes, "Generalized topology design of structures with a buckling load criterion," *Struct. Optim.*, vol. 10, no. 2, pp. 71–78, Oct. 1995.
- [19] O. Sigmund, "A 99 line topology optimization code written in Matlab," *Struct. Multidiscip. Optim.*, vol. 21, no. 2, pp. 120–127, 2001.
- [20] L. H. Tenek, "Eigenfrequency maximization of plates by optimization of topology using homogenization and mathematical programming," *JSME Int J*, vol. 37, pp. 667–677, 1994.
- [21] N. L. Pedersen, "Maximization of eigenvalues using topology optimization," in *Structural and Multidisciplinary Optimization*, vol. 20, 2000, pp. 2–11.
- [22] M. M. Neves, O. Sigmund, and M. P. Bendsøe, "Topology optimization of periodic microstructures with a penalization of highly localized buckling modes," *Int. J. Numer. Methods Eng.*, vol. 54, no. 6, pp. 809–834, 2002.
- [23] X. Guo, W. S. Zhang, Y. U. Wang, and P. Wei, "Stress-related topology optimization via level set approach," *Comput Methods Appl Mech Eng*, vol. 200, pp. 3439–3452, 2011.
- [24] X. Guo and G. D. Cheng, "Epsilon-continuation approach for truss topology optimization," *Acta Mech. Sin.*, vol. 20, no. 5, pp. 526–533, 2004.
- [25] M. Kocvara and M. Stingl, "Solving stress constrained problems in topology and material optimization," *Struct. Multidiscip. Optim.*, vol. 46, no. 1, pp. 1–15, 2012.
- [26] P. A. Browne, C. Budd, N. I. M. Gould, H. A. Kim, and J. A. Scott, "A fast method for binary programming using first-order derivatives, with application to topology optimization with buckling constraints," *Int. J. Numer. Methods Eng.*, vol. 92, no. 12, pp. 1026–1043, 2012.
- [27] G. Allaire, "A level-set method for vibration and multiple loads structural optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 3269–3290, 2005.
- [28] G. Allaire and F. Jouve, "Minimum stress optimal design with the level set method," *Eng. Anal. Bound. Elem.*, vol. 32, no. 11, pp. 909–918, 2008.
- [29] K. Suresh and M. Takaloozadeh, "Stress-Constrained Topology Optimization: A Topological Level-Set Approach," *Struct. Multidiscip. Optim.*, vol. Submitted, 2012.
- [30] X. Wang, "Structural shape and topology optimization in a level-set-based framework of region representation," *Struct. Multidiscip. Optim.*, vol. 27, no. 1–2, pp. 1–19, 2004.
- [31] T. J. R. Hughes, I. Levit, and J. Winget, "An element-by-element solution algorithm for problems of structural and solid mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 36, no. 2, pp. 241–254, Feb. 1983.
- [32] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, "A Deflated Version of the Conjugate Gradient Algorithm," pp. 1909–1926, 2000.
- [33] R. Aubry, F. Mut, S. Dey, and R. Lohner, "Deflated Preconditioned Conjugate Gradient Solvers for Linear Elasticity," *Int. J. Numer. Methods Eng.*, vol. 88, pp. 1112–1127.
- [34] E. A. Karabassi, G. Papaioannou, and T. Theoharis, "A Fast Depth-Buffer-Based Voxelization Algorithm," *J. Graph. Tools*, vol. 4, no. 4, pp. 5–10, 1999.
- [35] I. C. F. Ipsen, "Computing an Eigenvector with Inverse Iteration," pp. 254–291, 1997.
- [36] S. J. van den Boom, "Topology Optimisation Including Buckling Analysis," Delft University of Technology, Delft, 2014.
- [37] K. Suresh, A. Ramani, and A. Kaushik, "An Adaptive Weighting Strategy for Multi-Load Topology Optimization," in *ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, 2012.
- [38] SolidWorks, *SolidWorks*; [www.solidworks.com](http://www.solidworks.com). 2005.