

Accelerated fixed-point formulation of Topology Optimization: application to compliance minimization problems

Weichen Li^a, Phanish Suryanarayana^{a,*}, Glaucio H. Paulino^a

^a*School of Civil and Environmental Engineering, Georgia Institute of Technology
Atlanta, Georgia, 30332*

Abstract

We present a simple, effective, and scalable approach for significantly accelerating the convergence in Topology Optimization simulations. Specifically, treating the design process as a fixed-point iteration, we propose employing a recently developed acceleration technique in which Anderson extrapolation is applied periodically, with simple weighted relaxation used for the remaining steps. Through selected examples in compliance minimization, we show that the proposed approach is able to accelerate the overall simulation several fold, while maintaining the quality of the solution.

Keywords: Topology optimization, Fixed-point iteration, Optimality Criteria, Anderson extrapolation, Compliance minimization

1. Introduction

Topology optimization is finding increasingly widespread use in a number of different fields, including aerospace engineering, biomedical engineering, and architecture [1, 2, 3]. It consists of a nonlinear programming problem, which can be solved, for example, by means of sequential convex programming schemes [1, 4], such as the Optimality Criteria (OC) update [1] and the Method of Moving Asymptotes (MMA) [5]. However, the large computational cost associated with such simulations severely restricts the system sizes that can be studied, and ultimately, the resolution that can be achieved in the final designs [6].

In order to obtain converged designs, the aforementioned techniques can require hundreds or even thousands of optimization steps, with each step involving the solution of an ill-conditioned linear system. To ac-

celerate the convergence of the design process, a number of techniques have been proposed to incorporate second-order information. These include variants of the MMA [7, 8, 9, 10, 11], sequential quadratic programming (SQP) [12, 13], and interior point algorithms [14]. In spite of significant advances, these techniques are generally challenging to implement, have relatively poor scaling with system size, and are associated with larger computational time, which makes them unattractive compared to first-order methods.

In this work, we present a simple, effective, and scalable approach for accelerating convergence in Topology Optimization simulations. Specifically, in order to accelerate the standard fixed-point iteration employed in such computations, we adopt a recently developed extrapolation method [15] that has found application in large-scale linear [15, 16, 17] as well as nonlinear [18, 19, 20] problems. In this technique, Anderson extrapolations [21] are applied periodically in the fixed-point iteration, with standard weighted relaxations used for the remaining steps. Through se-

*Corresponding author

Email address:

phanish.suryanarayana@ce.gatech.edu (Phanish Suryanarayana)

lected examples in compliance minimization, we demonstrate that the proposed approach can significantly accelerate the design process in the framework of the SIMP (Solid Isotropic Material with Penalization) approach and the OC update [1]. Notably, we find that the proposed approach is able to not only achieve significant speedup, but also achieve lower objective function values.

The remainder of this paper is organized as follows. We present the accelerated fixed-point formulation of Topology Optimization in Section 2, verify its accuracy and efficiency through selected examples in Section 3, and finally conclude in Section 4.

2. Accelerated fixed-point formulation of Topology Optimization

In Topology Optimization, the solution of the design problem is typically achieved via a fixed-point iteration of the form

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k), \quad k = 0, 1, \dots \quad (1)$$

where $\mathbf{g}$ denotes the mapping of the density vector $\mathbf{x} \in \mathbb{R}^{N \times 1}$ between consecutive iterations. This mapping is typically comprised of the solution of the equilibrium equation, filtering of the sensitivity/density, and the application of an update scheme (see Algorithm 1). In the vicinity of the solution, a necessary condition for ensuring convergence of the above fixed-point iteration is

$$\sigma(\mathbf{I} - \mathbf{J}^*) < 1, \quad \mathbf{J}^* = -\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*}, \quad (2)$$

where $\sigma(\cdot)$ denotes the spectral radius of the associated matrix, $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix, and $\mathbf{J}^* \in \mathbb{R}^{N \times N}$ is the negative Jacobian of the residual function:

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \mathbf{x}, \quad (3)$$

evaluated at the solution $\mathbf{x}^*$. In general, the condition given in Eqn. (2) is not necessarily satisfied and even in cases where it holds, the convergence can be extremely slow, particularly when $\sigma(\mathbf{I} - \mathbf{J}^*) \approx 1$. Indeed, the convergence is faster as the value of $\sigma(\mathbf{I} - \mathbf{J}^*)$ becomes smaller.

In order to enhance convergence of the design process in Topology Optimization, we propose generalizing the fixed-point iteration in Eqn. (1) to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}_k \mathbf{f}_k, \quad k = 0, 1, \dots \quad (4)$$

where $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_k)$, and $\mathbf{B}_k \in \mathbb{R}^{N \times N}$ are appropriately chosen matrices. The necessary condition for ensuring convergence of the fixed-point iteration now becomes

$$\sigma(\mathbf{I} - \mathbf{B}_k \mathbf{J}^*) < 1, \quad (5)$$

with faster convergence again achieved for smaller values of the spectral radius. Therefore, the ideal choice would be $\mathbf{B}_k \approx \mathbf{J}^{*-1}$, which unfortunately requires knowledge of the solution.¹ More importantly, the calculation of such a matrix and its inverse in topology optimization is prohibitively expensive, even for small to moderately sized problems. This provides the motivation for the use of an extrapolation technique that is not only able to accelerate the convergence of the fixed-point iteration, but at the same time does not require Jacobian related information.

In view of the above discussion, we propose using a recently developed fixed-point acceleration method [15] in which Anderson extrapolation [21] is applied periodically within the fixed-point iteration, while a simple weighted relaxation is used in the remaining steps. Mathematically, this translates to the matrix $\mathbf{B}_k$ taking the form [16]

$$\mathbf{B}_k = \begin{cases} \alpha \mathbf{I} & \text{if } (k+1)/q \notin \mathbb{N}, \\ \beta \mathbf{I} - (\mathbf{X}_k + \beta \mathbf{F}_k)(\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T & \text{if } (k+1)/q \in \mathbb{N}, \end{cases} \quad (6)$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are relaxation parameters, $q \in \mathbb{N}$ is the frequency of Anderson extrapolation, and $\mathbf{X}_k \in \mathbb{R}^{N \times m}$ and $\mathbf{F}_k \in \mathbb{R}^{N \times m}$ are matrices containing the iter-

¹It is possible to utilize the Jacobian of the residual function at the current iterate with the hope that it is sufficiently close to $\mathbf{J}^*$.

ation and residual histories:

$$\mathbf{X}_k = [\Delta \mathbf{x}_{k-m} \quad \Delta \mathbf{x}_{k-m+1} \quad \dots \quad \Delta \mathbf{x}_{k-1}] , \quad (7)$$

$$\mathbf{F}_k = [\Delta \mathbf{f}_{k-m} \quad \Delta \mathbf{f}_{k-m+1} \quad \dots \quad \Delta \mathbf{f}_{k-1}] . \quad (8)$$

Above, $(m+1)$ is the number of iterates used for Anderson extrapolation, $\Delta \mathbf{x}_j = \mathbf{x}_{j+1} - \mathbf{x}_j$, and $\Delta \mathbf{f}_j = \mathbf{f}_{j+1} - \mathbf{f}_j$. Anderson extrapolation can be understood as taking the weighted average of the previous $(m+1)$ iterates to generate the next iterate, with the weights chosen so as to minimize the ℓ_2 norm of the vector resulting from the same weighted average of the previous $(m+1)$ residuals. Note that the matrix $\mathbf{B}_k$ never needs to be calculated, but rather only its multiplication with $\mathbf{f}_k$ is required. Therefore, the acceleration step is associated with low computational cost and computer memory requirements.

In Algorithm 1, we summarize the above described fixed-point formulation for accelerating Topology Optimization simulations. Each design cycle (i.e., fixed-point map $\mathbf{g}$) is comprised of solving the equilibrium equation, computing and filtering the sensitivity², and updating the density field. As discussed above, the design process is accelerated by employing Anderson extrapolations periodically, with weighted relaxation in the remaining steps. The acceleration is performed starting from step number s , since we have observed that applying the acceleration from the very beginning can sometimes stagnate the design process. Note that, although the acceleration step conserves the total volume, some elements of the density vector can occasionally fall outside the range $[0, 1]$ after applying Anderson extrapolation.³ Therefore, we restrict the elements of the density vector to $[0, 1]$ after this step, which can result in small violations of the volume constraint. However, based on the tens of simulations performed in this work, we have found that such violations fade out as the iteration heads

towards convergence.

Algorithm 1 Periodic Anderson accelerated topology optimization update

Initialize: $\mathbf{x}_0, s, q, m, \alpha, \beta, \epsilon, \text{iter}_{\max}$
for $k = 0, 1, \dots, \text{iter}_{\max}$ **do**
 Solve: $\mathbf{K}(\mathbf{x}_k) \mathbf{u} = \mathbf{d}$
 Compute and filter sensitivity
 Use an update scheme to compute: $\tilde{\mathbf{x}}_k$ and $\mathbf{f}_k$
 if $\|\mathbf{f}_k\| < \epsilon$ **then**
 quit
 end if
 if $k \geq s$ **then**
 if $k/q \in \mathbb{N}$ **then**
 Obtain $\mathbf{x}_{k+1}$: Anderson extrapolation (β, m)
 Restrict $\mathbf{x}_{k+1}$ to $[0, 1]$
 else
 Obtain $\mathbf{x}_{k+1}$: Weighted relaxation (α)
 end if
 else
 Obtain $\mathbf{x}_{k+1}$: $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}_k$
 end if
 Form $\mathbf{K}(\mathbf{x}_{k+1})$
end for

At first glance, it would appear that setting $q = 1$, i.e., performing Anderson extrapolation every iteration is likely to be the optimal choice. However, similar to previous observations in the context of both linear [15, 16] and nonlinear problems [18], we have found that applying the Anderson extrapolation periodically provides substantially faster convergence, an observation that can be attributed to the better subspace over which the residual is minimized.⁴ As is to be expected for nonlinear problems, both large and small values of the mixing history (i.e., m) can negatively impact the convergence. In addition, larger values of the relaxation parameters (i.e., α and β) can result in faster convergence, but at the cost of the method being less stable/robust. Overall, the proposed approach has the potential to significantly accelerate Topology Optimization sim-

²An alternative to filtering the sensitivity is the filtering of the density.

³In Topology Optimization, the density typically takes values in the interval $[0, 1]$.

⁴We have also found that the proposed approach demonstrates superior performance compared to other extrapolation techniques such as Broyden mixing [22].

ulations, as demonstrated by selected examples in the next section.

3. Results and discussion

In this section, we verify the accuracy and efficiency of the proposed fixed-point formulation in accelerating Topology Optimization simulations. Specifically, we consider the following compliance minimization problem in the context of the finite-element discretization:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{u}^T \mathbf{K}(\mathbf{x}) \mathbf{u} \\ & \text{s.t. } 0 \leq x_e \leq 1 \text{ and } \sum_{e=1}^{n_e} x_e \leq V, \quad (9) \\ & \text{with } \mathbf{K}(\mathbf{x}) \mathbf{u} = \mathbf{d}, \end{aligned}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x})$ denotes the displacement vector, $\mathbf{K}$ is the stiffness matrix, x_e denotes the component of the density vector $\mathbf{x}$ corresponding to element number e , n_e is the total number of elements, V represents the volume fraction, and $\mathbf{d}$ is the force vector. We implement the proposed acceleration scheme within the top88 [23] and top3D [24] codes, which are used to study 2D MBB (Messerschmitt-Bölkow-Blohm) and 3D cantilever beams with concentrated loads, respectively. In both examples, we employ the modified SIMP approach [25] with the Optimality Criteria (OC) update [1]. In this context, we refer to the proposed acceleration scheme as PAOC (Periodic Anderson accelerated OC), an abbreviation we will use henceforth. We have found the following parameters to work well in PAOC: $m = 3 \sim 5$, $q = 3 \sim 5$, $\alpha = 0.8 \sim 0.95$, and $\beta = 2 \sim 9$. In addition, we have found that continuously increasing the relaxation parameter β with design cycles helps in accelerating convergence.

3.1. 2D MBB beam with concentrated load

We first consider a 2D MBB beam with a concentrated load, as shown in Fig. 1a. We choose a 600×300 mesh consisting of 4-node quad bilinear elements, a volume fraction of $V = 0.3$, penalization of $p = 3$, and sensitivity filter radius of $r = 12$. In PAOC, we select

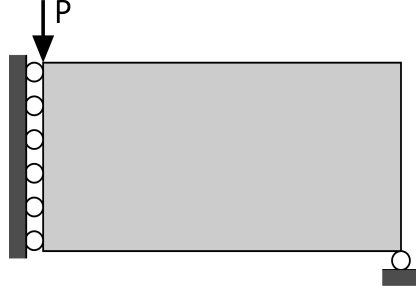
$m = 4$, $q = 4$, $\alpha = 0.9$, $\beta = 4 + (k - s)/50$, and $s = 50$. In Figs. 1b and 1c, we present the final designs obtained by the OC and PAOC methods, which are practically indistinguishable. In fact, PAOC is able to achieve lower values of the compliance (i.e., objective function) and the residual⁵, as demonstrated by the results in Figs. 2a and 2b, respectively. Furthermore, it is clear from the timings presented in Fig. 2c that PAOC is able to significantly accelerate the convergence, demonstrating larger speedup for tighter tolerances. In particular, PAOC requires a factor of 5 lower time compared to OC for achieving a tolerance of 10^{-6} in the residual.

3.2. 3D cantilever beam with concentrated load

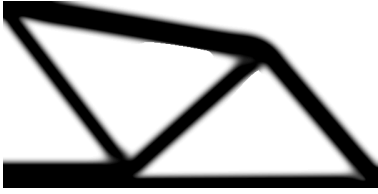
We now consider a 3D cantilever beam with a end concentrated load, as shown in Fig. 3a. We discretize the design domain with a $160 \times 80 \times 80$ mesh consisting of 8-node hexahedral trilinear elements, resulting in over 1 million elements. In addition, we choose a volume fraction of $V = 0.12$, penalization of $p = 3$, and sensitivity filter radius of $r = 8$. In PAOC, we select $m = 4$, $q = 4$, $\alpha = 0.9$, $\beta = 3 + (k - s)/50 \leq 5$, and $s = 25$. Given the relatively large size of the problem, particularly for simulations in Matlab, instead of using a direct solver, we use the conjugate gradient (CG) method [26] with IC(0) preconditioner for solving the equilibrium equation. To further reduce the computational time, we use a continuation strategy for the tolerance of the relative residual, wherein it is made stricter by one order of magnitude every 50 design cycles, starting from 10^{-4} up to 10^{-8} . Although this leads to larger displacement errors in the early stages of the design process, it has been found that the design sensitivity is insensitive to the accuracy of the linear system's solution [6].

In Figs. 3b and 3c, we present the final designs so obtained by the OC and PAOC methods. As in the previous example, the designs are indistinguishable and PAOC is

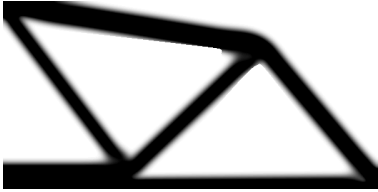
⁵We are denoting the l_2 norm of the residual vector as the residual.



(a)



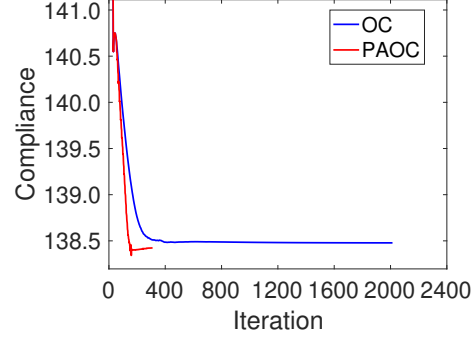
(b)



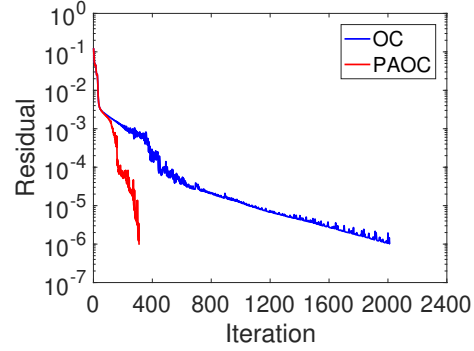
(c)

Figure 1: 2D MBB beam with concentrated load. (a) Design domain. (b) Design generated by OC. (c) Design generated by PAOC.

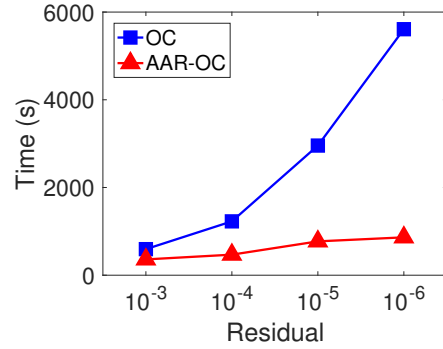
able to achieve lower values of the compliance (i.e., objective function) and the residual, as demonstrated by the results in Figs. 4a and 4b, respectively. Furthermore, it is clear from the CPU times presented in Fig. 4c that PAOC is able to significantly accelerate the convergence, again demonstrating larger speedup for tighter tolerances, requiring a factor of greater than 3 lower time compared to OC for achieving a tolerance of 10^{-5} in the residual. Note that since the residual of the



(a)



(b)



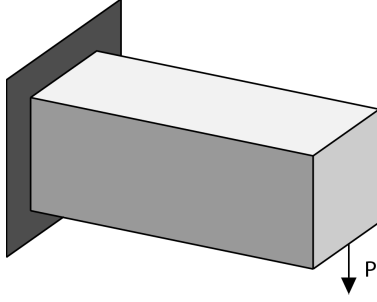
(c)

Figure 2: Comparison of the performance of OC and PAOC for the 2D MBB beam with a concentrated load. (a) Objective function history. (b) Residual history. (c) Timing.

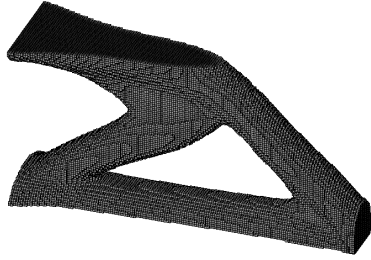
OC did not reach 10^{-6} in even 400 cycles, the corresponding time is estimated by adopting a linear fit to the data in Fig. 4b. Indeed, the linear fit provides a lower bound on the speedup, since the increase in time as a function of residual is worse than linear, as shown in Fig. 2c.

3.3. Discussion

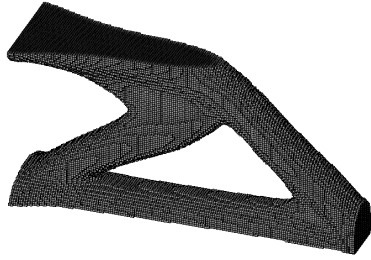
We have shown that PAOC significantly accelerates the convergence of the design pro-



(a)



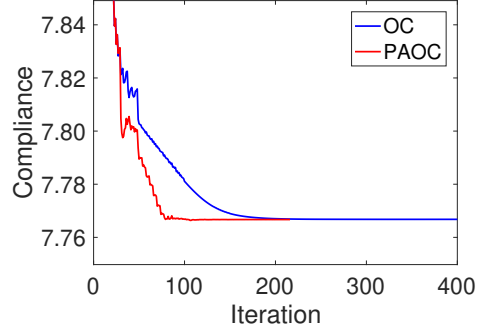
(b)



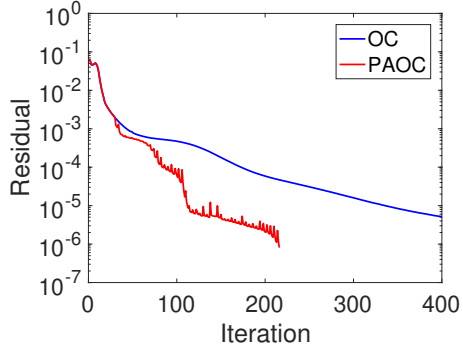
(c)

Figure 3: 3D cantilever beam with concentrated load. (a) Design domain. (b) Design generated by OC. (c) Design generated by PAOC.

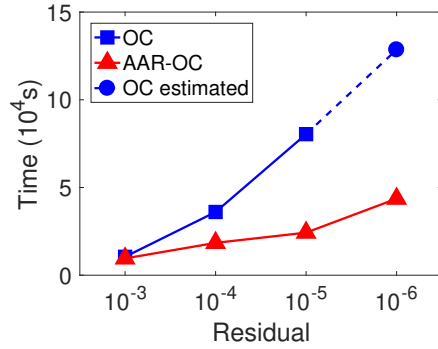
cess in Topology Optimization simulations. Since the additional time associated with the extrapolation is negligible (less than 0.1% of the total time), the reduction in the number of iterations required directly translates to the observed speedup. Note that the speedup of PAOC over OC reduces as the sensitivity filter radius becomes smaller, which is to be expected due to the reduced smoothness of the solution. Interestingly, PAOC is also able to achieve smaller values of the objective



(a)



(b)



(c)

Figure 4: Comparison of the performance of OC and PAOC for the 3D cantilever beam with a concentrated load. (a) Objective function history. (b) Residual history. (c) Timing.

function, an observation that warrants further investigation. Due to the non-convex nature of the problems, PAOC can indeed converge to different designs compared to OC, even though this was not the case for the examples studied here. A limitation of PAOC that warrants further investigation is the apparent stagnation for the choice of a density filter. It is worth noting that the extrapolation technique employed here can be used to

accelerate the Richardson/Jacobi fixed-point iteration [16], resulting in the AAR linear solver. AAR can outperform state-of-the-art Krylov subspace solvers like CG and GMRES [27], with larger speedups as the number of processors increase in parallel computing. This has the potential to further increase the size of problems that can be studied using Topology Optimization.

4. Conclusions

We have presented a new strategy for accelerating the convergence in Topology Optimization simulations. Specifically, viewing the design process as a fixed-point iteration, we have proposed employing a recently developed acceleration technique in which Anderson extrapolations are applied periodically, with simple weighted relaxation used for the remaining steps. For the specific problem of compliance minimization, we have shown through selected examples that the proposed approach is able to not only accelerate the complete simulation several fold, but also achieve lower values of the objective function. Overall, the proposed approach is simple, effective, and scalable. Topics for future investigation include addressing larger-scale problems through parallel computing, and the study of problems other than compliance minimization, e.g. compliant mechanisms and multiphysics problems.

Acknowledgements

The authors gratefully acknowledge the support provided by the National Science Foundation (NSF) under grant number 1663244.

References

- [1] M. P. Bendsøe, O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer-Verlag Berlin Heidelberg, 2004.
- [2] A. Sutradhar, J. Park, D. Carrau, T. H. Nguyen, M. J. Miller, G. H. Paulino, Designing patient-specific 3d printed craniofacial implants using a novel topology optimization method, *Medical & Biological Engineering & Computing* 54 (7) (2016) 1123–1135.

- [3] L. L. Stromberg, A. Beghini, W. F. Baker, G. H. Paulino, Application of layout and topology optimization using pattern gradation for the conceptual design of buildings, *Structural and Multidisciplinary Optimization* 43 (2) (2011) 165–180.
- [4] P. Christensen, A. Klarbring, *An Introduction to Structural Optimization*, Springer Netherlands, 2009.
- [5] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International Journal for Numerical Methods in Engineering* 24 (2) (1987) 359–373.
- [6] O. Amir, O. Sigmund, On reducing computational effort in topology optimization: how far can we go?, *Structural and Multidisciplinary Optimization* 44 (1) (2011) 25–29.
- [7] K. Svanberg, A class of globally convergent optimization methods based on conservative convex separable approximations 12 (2) (2002) 555–573.
- [8] M. Bruyneel, C. Fleury, Composite structures optimization using sequential convex programming, *Advances in Engineering Software* 33 (7) (2002) 697 – 711, *engineering Computational Technology & Computational Structures Technology*.
- [9] H. Smaoui, C. Fleury, L. A. Schmit, Advances in dual algorithms and convex approximation methods, in: *29th Structures, Structural Dynamics and Materials Conference*, California Univ., 1988, p. 9.
- [10] C. Fleury, First and second order convex approximation strategies in structural optimization, *Structural optimization* 1 (1) (1989) 3–10.
- [11] P. Duisinx, W. Zhang, C. Fleury, V. H. Nguyen, S. Haubridge, A new separable approximation scheme for topological problems and optimization problems characterized by a large number of design variables, *Proceedings of the First World Congress of Structural and Multidisciplinary Optimization (WCSMO1)* (1995) 1 – 8.
- [12] S. Rojas-Labanda, M. Stolpe, An efficient second-order SQP method for structural topology optimization, *Structural and Multidisciplinary Optimization* 53 (6) (2016) 1315–1333.
- [13] P. Gill, W. Murray, M. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, *SIAM Review* 47 (1) (2005) 99–131.
- [14] S. Rojas-Labanda, M. Stolpe, Solving large-scale structural topology optimization problems using a second-order interior point method.
- [15] P. P. Pratapa, P. Suryanarayana, J. E. Pask, Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems, *Journal of Computational Physics* 306 (2016) 43 – 54.
- [16] P. Suryanarayana, P. P. Pratapa, J. E. Pask, Alternating anderson-richardson method: An efficient alternative to preconditioned krylov methods for large, sparse linear systems, *Computer Physics Communications* 234 (2019) 278–285.
- [17] M. Lupo Pasini, Convergence analysis of anderson-type acceleration of richardson’s iter-

- ation, Numerical Linear Algebra with Applications (2019) e2241.
- [18] A. S. Banerjee, P. Suryanarayana, J. E. Pask, Periodic Pulay method for robust and efficient convergence acceleration of self-consistent field iterations, *Chemical Physics Letters* 647 (2016) 31 – 35.
 - [19] P. Suryanarayana, P. P. Pratapa, A. Sharma, J. E. Pask, Sqdft: Spectral quadrature method for large-scale parallel $O(n)$ kohn–sham calculations at high temperature, *Computer Physics Communications* 224 (2018) 288–298.
 - [20] Q. Xu, P. Suryanarayana, J. E. Pask, Discrete discontinuous basis projection method for large-scale electronic structure calculations, *The Journal of chemical physics* 149 (9) (2018) 094104.
 - [21] D. G. Anderson, Iterative procedures for nonlinear integral equations, *Journal of the Association for Computing Machinery* 12 (4) (1965) 547 – 560.
 - [22] D. D. Johnson, Modified broyden’s method for accelerating convergence in self-consistent calculations, *Physical Review B* 38 (18) (1988) 12807.
 - [23] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, O. Sigmund, Efficient topology optimization in matlab using 88 lines of code, *Structural and Multidisciplinary Optimization* 43 (1) (2011) 1–16.
 - [24] K. Liu, A. Tovar, An efficient 3d topology optimization code written in matlab, *Structural and Multidisciplinary Optimization* 50 (6) (2014) 1175–1196.
 - [25] O. Sigmund, Morphology-based black and white filters for topology optimization, *Structural and Multidisciplinary Optimization* 33 (4-5) (2007) 401–424.
 - [26] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards* 49 (6) (1952) 409–436.
 - [27] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* 7 (3) (1986) 856–869.