EXPLORING THE METHOD OF MOVING ASYMPTOTES FOR VARIOUS OPTIMIZATION APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

By

Emily Alcazar

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Civil and Environmental Engineering

Georgia Institute of Technology

August 2021

EXPLORING THE METHOD OF MOVING ASYMPTOTES FOR VARIOUS OPTIMIZATION APPLICATIONS

Thesis committee:

Dr. Glaucio Paulino, Advisor
School of Civil and
Environmental Engineering
Georgia Institute of Technology

Dr. David Rosen
The George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Yang Wang
School of Civil and Environmental
Engineering
Georgia Institute of Technology

Dr. Jonathan Russ
Plato Development Team
Sandia National Laboratories

Date approved: July 30, 2021

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. Glaucio H. Paulino for his endless support and guidance in the preparation of this thesis. His mentorship has served as a constant source of inspiration for me, both professionally and academically.

I would like to thank the members on my thesis committee, Dr. David Rosen and Dr. Yang Wang, for their valuable comments and suggestions in the preparation of this work. I would especially like to acknowledge Dr. Jonathan Russ for his participation on this committee and his ongoing support in the development of this work through productive, enlightening discussions.

I would also like to extend my gratitude to my colleagues, Emily Sanders, Tuo Zhao, Fernando Senhora, Oliver Giraldo-Londoño, Larissa Novelino, and Yang Jiang, for their always informative discussions, invaluable advice, and continuous motivation.

In recognition of the financial support of this work, I would like to thank the National Science Foundation (NSF) Graduate Research Fellowship Program (GRFP) and the Raymond Allen Jones Chair endowment through the Georgia Institute of Technology.

Finally, I would like to thank my family for their never-ending support and encouragement in all of my endeavors. I am eternally grateful for the admiration and inspiration I receive from my two sisters, Isabel and Katie. To my parents, words cannot express how fortunate I am for your unconditional love and belief in me. It is with great pleasure that I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

BFGS Broyden Fletcher Goldfard Shanno

CONLIN Convex Linearization

DFP Davidon Fletcher Powell

GBMMA Gradient Based Method of Moving Asymptotes

GCMMA Globally Convergent Method of Moving Asymptotes

KKT Karush-Kuhn-Tucker

LP Linear Programming

MMA Method of Moving Asmyptotes

NLP Nonlinear Programming

OC Optimality Criterion

PLATO PLAtform for Topology Optimization

SIMP Solid Isotropic Material with Penalization

SLP Sequential Linear Programming

SQP Sequential Quadratic Programming

SUMMARY

The development of sequential explicit, convex approximation schemes has allowed for expansion of the size of optimization problems that can now be achieved. These approximation schemes use information from the original optimization statement to generate a series of approximate subproblems allowing for an efficient solution strategy. This thesis reviews established sequential explicit, convex approximations in the literature along with a brief overview of their associated solution schemes. A primary focus is placed on the theory and application of the Method of Moving Asymptotes (MMA) approximation due to its continued regard in the field of structural topology optimization. Numerical examples explore optimization problems solved by the MMA approximation in order to demonstrate the behavior of this method and impact of the prescribed empirical parameters. Other numerical examples study structural topology optimization problems in the 2D and 3D setting to compare with alternative, competitive update schemes such as the OC and to highlight the benefit of using the MMA in more complex settings.

CHAPTER 1
INTRODUCTION

Structural topology optimization is an immensely powerful tool to aid engineers and designers alike by informing the optimal placement of material and void, inspiring designs that are both elegant and efficient, uniting the fields of art and engineering seamlessly as never before. This idea, as emphasized in this work, was introduced by Bendsoe and Kikuchi, who proposed a technique to find the optimal layout of an anisotropic material formed by void and homogeneous, isotropic material microstructures [1]. This work led to the creation of the modern field of structural topology optimization in 1988. Since then, many advancements to the literature have been made [2, 3, 4]. More specifically within the realm of solution strategies of optimization problems, significant progress has been made by the approach of convex approximations [5, 6]. This thesis will focus on the theory and application of these sequential explicit, convex approximations in solving various optimization problems.

1.1 Brief Literature Review

The concept of sequential explicit, convex approximations was proposed by the mathematicians Schmidt, Farshi, and Fleury in their approach of linearized approximations of the optimization problem coupled with dual methods for a complete and efficient solution scheme [7, 8]. This method confronted the challenges associated with solving large-scale optimization problems where the explicit solutions may be highly complex (and their implementation extremely computationally expensive) or in many cases nonexistent. The introduction of this framework led to further development of the approximation subroutines, Sequential Linear Programming (SLP) and Sequential Quadratic Programming (SQP), improving these methods for greater robustness in the context of optimization applications [9,

10]. It also influenced the development of new convex approximation subroutines such as Convex Linearization (CONLIN) [11], the Method of Moving Asymptotes (MMA) [6], and Globally Convergent Method of Moving Asymptotes (GCMMA) [12]. The MMA subroutine revolutionized the topology optimization field due to its robustness in solving problems of any form, e.g. multiple constraints, nonlinear functions, etc. and it continues to dominate as the sequential convex approximation scheme of choice. This approximation technique coupled with an efficient solution strategy (Chapter 3.0) and further developed computational capabilities has allowed for its vast expansion in the field. Such work includes topology optimization for mass minimization subject to local stress constraints handled by the augmented Lagrangian method, where the MMA was used to minimize the augmented Lagrangian function [13]. The MMA has also been applied to solve optimization problems of ultra high resolution shell structures considering over ten millions elements, demonstrating the scale by which these problems can now be achieved [14].

Due to the generality and computational efficiency of the MMA, it has been implemented into many structural topology optimization computer programs. This includes its incorporation into educational codes such as PolyStress where the MMA handles a formulation of multiple local stress constraints considering material nonlinearities in a Matlab program [15]. Another application was the use of the MMA-like function approximations implemented in a 250-line Matlab code for solving topology optimization formulations considering linearized buckling criteria [16]. The MMA procedure was also presented in an open source Matlab code in finding level set topology and shape optimization using density methods [17]. Furthermore, large-scale optimization problems have been computed in the setting of a C/C++ code coupled with parallel implementation of the MMA [18]. In addition to research endeavors, the MMA method has also proved to be an effective update scheme for industry use with software such as PLAtform for Topology Optimization (PLATO) by Sandia National Laboratories which handles topology optimization problems for compliance minimization, heat flow maximization, or stress minimization in the 2D and

3D setting [19]. Another powerful commercial software for topology optimization using the MMA is TRINITAS which is capable of generating a 2D or 3D finite element mesh for the desired design domain [20]. Note that the previously mentioned works are only a few of many that have benefited by the MMA approach. Thus this review is by no means complete and is intended just to highlight the diverse use of the MMA in optimization.

This thesis serves as a pedagogical approach in explaining and demonstrating the theory behind sequential explicit, convex approximations. With the use of these methods being so prominent in the field of structural topology optimization, it is essential to have a clear understanding on the methodology, behavior, and influence these techniques have on the final optimization solution.

1.2 Thesis Organization

The remainder of the thesis is presented in the following order. Chapter 2 will explain some widely used sequential explicit, convex approximations while also serving as a brief literature review for other existing schemes. Chapter 3 describes the duality method in which several of the sequential explicit, convex approximations can be solved. Chapter 4 demonstrates various examples illustrating the behavior of different approximation schemes, with a primary focus on the MMA scheme, in the context of both general optimization and structural topology optimization problems. Lastly, Chapter 5 concludes the notable discoveries found in the numerical results of this extensive study.

CHAPTER 2

ON SEQUENTIAL EXPLICIT, CONVEX APPROXIMATIONS

An optimization problem can be described in the following most general mathematical programming form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq \quad i = 1, ..., m \\ & h_j(\boldsymbol{x}) = 0 \quad j = 1, ..., n. \end{aligned} \tag{2.1}$$

The formulation can be translated as finding the design variable or set of design variables, $\boldsymbol{x}$, that minimize the optimization goal otherwise referred to as the objective function, $f(\boldsymbol{x})$, while satisfying the m imposed inequality and n equality constraints, $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$, respectively. In regards to structural topology optimization, the optimization formulation more commonly takes the following form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0 \quad i = 1, ..., m \\ & \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad x_j^{min} \leq x_j \leq x_j^{max}, \quad j = 1, ..., n \right\} \\ \text{with} \quad & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x}). \end{aligned} \tag{2.2}$$

Here $f(\boldsymbol{x})$ is the objective function which in the structural design setting commonly refers to compliance i.e. the inverse of stiffness, but may differ depending on the structural design target. The objective function is defined in terms of n design variables, $\boldsymbol{x}$, which may describe the area, density, thickness, etc. The design domain is restricted by m inequality constraints. These constraints could include an imposed volume limit, an allowable stress, a deflection constraint, or any restriction that may inform a more efficient design depending

on the desired structural application. In addition, $\boldsymbol{x}$ is subjected to n box constraints with lower and upper bounds of x_j^{min} and x_j^{max} to ensure feasible solutions. The formulation described in Equation 2.2 enforces a state equation describing the linear elastic equilibrium equation with $\boldsymbol{u}$ being the displacement vector, $\boldsymbol{K}(\boldsymbol{x})$ being the stiffness matrix, and $\boldsymbol{F}(\boldsymbol{x})$ being the force vector. However, the state equation and response variables may vary among topology optimization problems to describe different physical phenomena. Coupling the optimization with finite element analysis allows the equilibrium equation to be used to find the response variable as a function of the design variable $(\text{in this case } \boldsymbol{u}(\boldsymbol{x}))$, which may be a term in the objective function such as in the case of compliance minimization $(C = \boldsymbol{F}^T\boldsymbol{u}(\boldsymbol{x}))$. For the sake of simplicity, we employ a linear state equation. However, the method can also handle nonlinear state equations, which is a field of much interest - see references [21, 15].

Most structural topology optimization problems that are modeled to simulate a realistic design setting often consider various constraints and are discretized into a fine mesh for high-resolution results, thus transforming their mathematical formulation into a large-scale numerical problem. These large-scale problems can obtain upwards of thousands to millions of degrees of freedom and typically take a nonconvex form, which makes finding their explicit solution computationally expensive or in most cases either impractical or impossible [22]. To combat these limitations, an approach to approximate large scale problems as a sequence of explicit, convex subproblems was proposed [7, 8]. The idea behind this method was that a subproblem would serve as a conservative yet precise approximation to the original optimization formulation, obtaining properties of both convexity and separability ensuring a unique solution of the subproblem and an efficient solution strategy. Once one subproblem is solved, if the convergence criteria is not satisfied, it formulates a new subproblem using the solution from the previous iteration, hence achieving a final solution

by a series of subproblems:

$$\begin{aligned} \min \quad & \tilde{f}^k(\boldsymbol{x}) \\ \text{s.t.} \quad & \tilde{g}_i^k(\boldsymbol{x}) \leq 0, \quad i = 1, ..., m \\ & \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad x_j^{min} \leq x_j \leq x_j^{max}, \quad j = 1, ..., n \right\} \\ \text{with} \quad & \boldsymbol{K}(\boldsymbol{x}^k)\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x}^k). \end{aligned} \tag{2.3}$$

The sequence of subproblems are formulated using information from the original optimization formulation (see Equation 2.2) such as the objective function, inequality constraints, gradient information (for first order approximations), and even Hessian information (for second order approximations). In the optimization subproblem, as seen in Equation 2.3, the terms $\tilde{f}(\boldsymbol{x})$ and $\tilde{g}_i(\boldsymbol{x})$ represent the approximations of the objective function and the constraints. The index k represents the number of iterations or subproblems to be solved before convergence is reached. By reformulating the original optimization problem into a series of explicit, convex, separable approximations, one can take advantage of using the dual approach or other efficient mathematical techniques for solving the subproblem. The sequence of subproblems can be solved efficiently thus leading to minimal computational effort. Once the solution reaches the stopping criteria, such as a predefined tolerance of the change between two previous solutions, the solving of the sequential approximate subproblems will end resulting in a final solution to the original optimization statement. This methodology for solving optimization problems via sequential convex approximations is described more explicitly in **Algorithm 1**. Note that it is assumed in all of the approximation techniques that the sensitivity information can be computed either numerically (e.g. the adjoint method [23], automatic differentiation [24], etc.) or analytically for simple problems.

Algorithm 1 Procedure for Solving Optimization Problems using Approximation Schemes

Start with an initial design variable guess $\boldsymbol{x}^0$
Define maxIter, set $k = 0$
for $k = 0$ to maxIter **do**
    Conduct finite element analysis: Solve $\boldsymbol{K}(\boldsymbol{x}^k)\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x}^k)$
    Compute the displacement vector, $\boldsymbol{u}(\boldsymbol{x}^k)$
    Calculate the objective function $f(\boldsymbol{x}^k)$, constraints $g_i(\boldsymbol{x}^k)$, their gradients $\nabla f(\boldsymbol{x}^k)$ and $\nabla g_i(\boldsymbol{x}^k)$, and if needed the Hessian $\boldsymbol{H}(\boldsymbol{x}^k)$
    Formulate the explicit, convex approximate subproblem about $\boldsymbol{x}^k$
    Solve the subproblem to generate the next design variable update $\boldsymbol{x}^{k+1}$
    if stopping criteria is satisfied **then**
        break
    end if
end for

The solution scheme by duality will be discussed more extensively in Chapter 3. The remainder of this Chapter will discuss in detail the sequential, convex programming algorithms that are commonly used in large-scale optimization problems. These algorithms include SLP, SQP, CONLIN, MMA, and other variations of the MMA.

2.1 Sequential Linear Programming, SLP

One of the original methods for approximating optimization problems was by implementing the first order expansion of the Taylor series using first order information from the original optimization problem. This transforms the optimization problem into a set of first order, convex subproblems:

$$\begin{aligned} \min \quad & \tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T(\boldsymbol{x} - \boldsymbol{x}^k) \\ \text{s.t.} \quad & \tilde{g}_i^k(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^T(\boldsymbol{x} - \boldsymbol{x}^k) \leq 0, \quad i = 1, ..., m \\ & \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad \alpha_j^k \leq x_j - x_j^k \leq \beta_j^k, \quad j = 1, ..., n \right\}. \end{aligned} \tag{2.4}$$

The SLP algorithm approximates the original optimization formulation by linearizing the objective and constraints about the change in the design variables, $\boldsymbol{x} - \boldsymbol{x}^k$. The approximate objective function, $\tilde{f}^k(\boldsymbol{x})$, is defined using information from the original optimization prob-

lem, the function and its gradient, $f(\boldsymbol{x}^k)$ and $\nabla f(\boldsymbol{x}^k)$, evaluated at each k iteration starting with x^0 as the initial guess. The inequality constraints are formulated in a similar matter using first order information from the original inequality constraint evaluated at each sequential iteration, $g(\boldsymbol{x}^k)$ and $\nabla g(\boldsymbol{x}^k)$. Thus all the quantities in the formulation are known, explicit functions of the design variables. Due to its linearity, the subproblem can be solved through a variety of Linear Programming (LP) algorithms such as the simplex method [25].

To improve the SLP by bounding the solution domain and informing feasible changes in the design, move limits are applied as introduced by Pope [9]. The lower move limit and the upper move limit, α_j^k and β_j^k, describe the lower and upper bounds for the change in the design variables, respectively. *The selection of the move limits drastically impacts the speed of convergence and even the success of the SLP algorithm* [25]. In the case that the move limits are too large there will be oscillations in the solution, and in the case that they're too small, the solution may converge slowly or never converge. Thus it is crucial to have an understanding of the type of problem being solved in order to select reasonable move limits. To remove the random selection of move limits from the design process, techniques to calculate the move limits could be implemented. Several authors have proposed different approaches to finding move limits, some based on search direction criteria [26] and others based on gradient information [27, 28]. For example, Lamberti et al. [29] has demonstrated that the implementation of the Constraints Gradient based Move Limit [27, 28] technique in SLP has improved efficiency with comparable CPU time to other solution schemes in the literature.

The SLP technique has been proved to be an efficient design variable update scheme in several works of structural topology optimization considering compliance minimization, multi-physics problems, dynamic formulations and more [30, 31, 32]. Due to its mathematically simple form, the subproblem can be readily solved numerically and can account for various different types of optimization problems. Limitations of the SLP algorithm include the need for precise selection of move limits, which may require some trial and

error runs, to avoid unfeasible solutions (in the scenarios where techniques to calculate the move limits are not used). Although the SLP method also obtains a relatively, conservative approximate form, depending on the optimization problem, it may not be informed enough to reach a local minimum.

2.2 Sequential Quadratic Programming, SQP

To improve the robustness of the approximate subproblem generated by the SLP algorithm, the objective function approximation can be extended considering second order information transforming it into a SQP formulation [33, 10]:

$$\begin{aligned} \min \quad & \tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^k)^T\boldsymbol{H}(\boldsymbol{x}^k)(\boldsymbol{x}-\boldsymbol{x}^k) \\ \text{s.t.} \quad & \tilde{g}_i^k(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k) \leq 0, \quad i = 1, ..., m \\ & \boldsymbol{x} \in \boldsymbol{X} = \left\{\boldsymbol{x} \in \mathbb{R}^n \quad x_j^{min} \leq x_j \leq x_j^{max}, \quad j = 1, ..., n\right\}. \end{aligned} \tag{2.5}$$

The objective function is now approximated as a quadratic function by adding the second order term of the Taylor expansion where $\boldsymbol{H}(\boldsymbol{x}^k)$ denotes the Hessian of the original objective function evaluated at the design variable of the iteration. In most large-scale optimization problems, finding the exact Hessian may be computationally expensive for practical application, thus Hessian approximations are employed through techniques such as Broyden rank-one quasi Newton updates, Davidon Fletcher Powell (DFP), or Broyden Fletcher Goldfard Shanno (BFGS) [34].

The SQP algorithm has proved to be extremely useful in the optimization setting [35, 36]. Due to the quadratic nature of the approximate objective function, the problem becomes bounded and thus the need for move limits becomes less critical, which serves as an advantage over the SLP technique [25]. However, drawbacks do exist such as the subproblem not being separable and thus the dual approach no longer being an efficient solver. Because quadratic approximations are highly informative in representing certain optimiza-

tion problems, efforts have been placed into efficiently solving the Nonlinear Programming (NLP) problem [37, 38]. One of the most widely used approaches is that of Schittkowski [39], where the solution of the quadratic programming subproblem is the search direction and the corresponding Lagrange multipliers which are used to determine the next steepest descent direction.

2.3 Convex Linearization, CONLIN

CONLIN is another method of sequential explicit, convex approximations proposed by Fleury and Braibant [11]. In this method, the authors took notice of certain linear and reciprocal features found often in structural optimization problems and created an approximation that linearizes these terms with either direct or reciprocal variables, x_j or $1/x_j$, respectively:

$$\begin{aligned}
\text{min} \quad & \tilde{f}^k(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \sum_{+} \frac{\partial f(\boldsymbol{x}^k)}{\partial x_j}(x_j - x_j^k) + \sum_{-} \frac{\partial f(\boldsymbol{x}^k)}{\partial x_j} \frac{x_j^k(x_j - x_j^k)}{x_j} \\
\text{s.t.} \quad & \tilde{g}_i^k(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \sum_{+} \frac{\partial g_i(\boldsymbol{x}^k)}{\partial x_j}(x_j - x_j^k) + \sum_{-} \frac{\partial g_i(\boldsymbol{x}^k)}{\partial x_j} \frac{x_j^k(x_j - x_j^k)}{x_j} \leq 0, \quad i = 1, ..., m \\
& \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad 0 < x_j^{min} \leq x_j \leq x_j^{max}, \quad j = 1, ..., n \right\}.
\end{aligned} \tag{2.6}$$

Both the objective function and the constraints in the approximate subproblem are formulated in the same manner in (Equation 2.6) where $\sum_+$ denotes the functions, $f(\boldsymbol{x}^k)$ and $g_i(\boldsymbol{x}^k)$, with a positive derivative and where $\sum_-$ denotes the functions with a negative derivative, indicating the correct usage of either direct or reciprocal variables. The approximation is convex (as per the name) and separable which means it can be readily solved via a dual method approach. Fleury has since proposed a unique dual optimizer approach tailored specifically for the CONLIN subproblem with reduced dimensionality in the primal and dual variables for an efficient solution scheme [40]. CONLIN has been proven to

be the most conservative estimate formulated by direct and reciprocal variables [41] and thus can solve a wide array of structural optimization problems, generating solutions in the feasible domain. Such large-scale structural optimization problems have even included the designs of naval and floating hydraulic structures [42].

It is important to note, however, that some restrictions arise from this method as a result of the fixed curvature of the approximation. This may lead to poor fitting of the original optimization problem and thus slow or unstable convergence. These issues were later addressed by Svanberg in his sequential approximation technique, the MMA [6].

2.4 Method of Moving Asymptotes, MMA

The MMA technique, produced by Svanberg in 1987 [6], continues to be the most popular choice of all sequential approximation methods in the structural topology optimization field [21]. This is due to its ability in handling various types of characteristics in structural optimization functions, all types of constraints, and expensive function evaluations. Its flexibility has made it the favored approximation scheme for guaranteed convergence to the local minima.

To improve upon the CONLIN approach, which had fixed conservatism, the MMA introduces moving asymptotes, L_j^k and U_j^k, which allow the level of conservatism of each design variable at each iteration to be controlled. In the context of optimization, conservatism is defined as increasing the convexity terms to reduce the size of the trust region in order to avoid constraint violations or by the mathematical definition $\tilde{g}_i^{(k,0)}(\hat{\boldsymbol{x}}^{(k,0)}) \geq g_i(\hat{\boldsymbol{x}}^{(k,0)})$ [43].

The lower and upper asymptotes relationship with respect to the design variable is described below:

$$L_j^k < x_j^k < U_j^k. \tag{2.7}$$

The MMA subproblem is defined by the following formulation now where $1/(U_j^k - x_j)$ and $1/(x_j - L_j^k)$ serve as the intervening variables for

$$\begin{aligned} \text{min} \quad & \tilde{g_0}^k(\boldsymbol{x}) = r_0^k + \sum_{j=1}^{n} \left(\frac{p_{0j}^k}{U_j^k - x_j} + \frac{q_{0j}^k}{x_j - L_j^k} \right) \\ \text{s.t.} \quad & \tilde{g}_i^k(\boldsymbol{x}) = r_i^k + \sum_{j=1}^{n} \left(\frac{p_{ij}^k}{U_j^k - x_j} + \frac{q_{ij}^k}{x_j - L_j^k} \right) \leq 0, \quad i = 1, ..., m \\ & \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad \alpha_j^k \leq x_j \leq \beta_j^k, \quad j = 1, ..., n \right\}, \end{aligned} \tag{2.8}$$

where

$$\begin{aligned} p_{ij}^k &= \begin{cases} (U_j^k - x_j^k)^2 \dfrac{\partial g_i}{\partial x_j}, & \text{if} \quad \dfrac{\partial g_i}{\partial x_j} > 0 \\ 0, & \text{if} \quad \dfrac{\partial g_i}{\partial x_j} \leq 0 \end{cases} \\ q_{ij}^k &= \begin{cases} 0, & \text{if} \quad \dfrac{\partial g_i}{\partial x_j} \geq 0 \\ -(x_j^k - L_j^k)^2 \dfrac{\partial g_i}{\partial x_j}, & \text{if} \quad \dfrac{\partial g_i}{\partial x_j} \leq 0 \end{cases} \\ r_i^k &= g_i(\boldsymbol{x}^k) - \sum_{j=1}^{n} \left(\frac{p_{ij}^k}{U_j^k - x_j^k} + \frac{q_{ij}^k}{x_j^k - L_j^k} \right). \end{aligned} \tag{2.9}$$

The terms in Equation 2.9 will be computed for $i = 0, ..., m$. Note that the objective approximation will now be denoted as $\tilde{g_0}^k$ in place of the previously used $\tilde{f}^k$ for a more comprehensible definition in Equation 2.9. As interpreted from Equation 2.9 and Equation 2.7, $p_{ij}^k \geq 0$ and $q_{ij}^k \geq 0$. This information coupled with the evaluation of the gradient and Hessian of the approximation functions (see Equation 2.10) demonstrates that the ap-

proximation is indeed convex due to the Hessian being positive semi-definite:

$$\begin{aligned}
\frac{\partial g_i^k(\boldsymbol{x})}{\partial x_j} &= \frac{p_{ij}^k}{(U_j^k - x_j)^2} - \frac{q_{ij}^k}{(x_j - L_j^k)^2} \\
\frac{\partial^2 g_i^k(\boldsymbol{x})}{\partial x_j^2} &= \frac{2p_{ij}^k}{(U_j^k - x_j)^3} + \frac{2q_{ij}^k}{(x_j - L_j^k)^3} \\
\frac{\partial^2 g_i^k(\boldsymbol{x})}{\partial x_j \partial x_l} &= 0 \quad if \quad j \neq l.
\end{aligned} \tag{2.10}$$

By taking notice of the form of the Hessian it can be seen that as the L_j^k and U_j^k bounds move closer to the x_j^k, the curvature of the approximation increases and thus so does the level of conservatism. This is how the selection of the asymptotes, L_j^k and U_j^k, control how conservative the MMA approximation will behave. The flexibility and control of the MMA can be demonstrated by changing the magnitudes of L_j^k and U_j^k. If the asymptotes are set to $L_j^k \to -\infty$ and $U_j^k \to +\infty$ the MMA approximation is transformed into the SLP approximation. If the asymptotes are instead defined as $L_j^k = 0$ and $U_j^k \to +\infty$ the MMA subproblem will be identical to that of the CONLIN. These claims are verified by the proofs in Appendix A.

The move limits, α_j^k and β_j^k, defined in the formulation in Equation 2.8 serve to avoid division by zero. To preserve this, the following inequalities must hold

$$L_j^k < \alpha_j^k < x_j^k < \beta_j^k < U_j^k, \tag{2.11}$$

where

$$\begin{aligned}
\alpha_j^k &= max(x_j^{min}, L_j^k + \mu(x_j^k - L_j^k)) \\
\beta_j^k &= min(x_j^{max}, U_j^k - \mu(U_j^k - x_j^k))
\end{aligned} \tag{2.12}$$

and where $0 < \mu < 1$. Note that the definition of the move limits has since been slightly updated in its computational implementation by introducing a move parameter, C_4, to help control the speed of the convergence-see Appendix B for more information.

The selection of the asymptotes has a drastic impact on the behavior and consequently the efficiency of the MMA procedure. The effect of these parameters will be further explored in the numerical examples-see Chapter 4.0 for reference. For proper selection of the move asymptotes and limits, to avoid oscillations in the iterations or slow convergence, an iterative protocol is defined.

For the first two iterations, $k = 0, 1$, the moving asymptotes can be defined by the following expressions:

$$\begin{aligned} L_j^k &= x_j^k - s_{\text{init}}(x_j^{\max} - x_j^{\min}) \\ U_j^k &= x_j^k + s_{\text{init}}(x_j^{\max} - x_j^{\min}), \end{aligned} \tag{2.13}$$

where x_j^k is the value of the design variable at iteration k, $0 < s_{\text{init}} < 1$, and $x_j^{\min}, x_j^{\max}$ are the lower and upper limits of the design variable, x_j. In the later iterations, $k \geq 2$, the asymptotes are defined depending on the behavior of the design variable solutions in the previous three iterations. If the change between the previous iterations, $x_j^k - x_j^{k-1}$ and $x_j^{k-1} - x_j^{k-2}$, obtain opposite signs, this indicates that the solution is oscillating and hence the approximation is not conservative enough. To increase the degree of conservatism the asymptotes will be brought closer together by the following equations:

$$\begin{aligned} L_j^k &= x_j^k - s_{\text{slower}}(x_j^{k-1} - L_j^{k-1}) \\ U_j^k &= x_j^k + s_{\text{slower}}(U_j^{k-1} - x_j^{k-1}), \end{aligned} \tag{2.14}$$

where $0 < s_{\text{slower}} < 1$. In the case that the signs of $x_j^k - x_j^{k-1}$ and $x_j^{k-1} - x_j^{k-2}$ are instead the same sign, the approximation may be too conservative and could be sped up by the following expressions:

$$\begin{aligned} L_j^k &= x_j^k - s_{\text{faster}}(x_j^{k-1} - L_j^{k-1}) \\ U_j^k &= x_j^k + s_{\text{faster}}(U_j^{k-1} - x_j^{k-1}), \end{aligned} \tag{2.15}$$

where $s_{\text{faster}} > 1$.

In the rare occasion that the product of the difference between the the previous design variables, $(x_j^k - x_j^{k-1})(x_j^{k-1} - x_j^{k-2})$, is 0 the the move asymptotes will be defined by the following expressions:

$$\begin{aligned} L_j^k &= x_j^k - (x_j^{k-1} - L_j^{k-1}) \\ U_j^k &= x_j^k + (U_j^{k-1} - x_j^{k-1}). \end{aligned} \tag{2.16}$$

The MMA has had a few modifications for the purpose of computational implementation in efforts to improve the efficiency of the approximation scheme [43]. The equation modifications and reasoning behind the changes made is highlighted in Appendix B for reference.

While the methodology for selecting asymptotes in the MMA approximation has proven to be quite robust there still exists uncertainty on the role of the empirical parameters, $s_{init},\ s_{slower},\ s_{faster},\ \mu,$ and C_4 on the speed of convergence and how to tune such parameters according to the problem at hand. The current understanding is that, in general, the empirical parameters are problem dependent and cannot be tuned in any systematic way. One way to eliminate the uncertainty of the optimal value of finding the empirical parameters would be to remove them all together. Such an idea was introduced by Fleury who proposed calculation of the moving asymptotes using second order sensitivity information, which could determine the curvature of the original function and use it in place of the empirical parameters [5]. The results from this work indicate that either the same or fewer iterations were needed to solve the subproblem using the proposed second order MMA versus the original MMA approach. However the scalability of the second order MMA may not be as strong as the MMA due to the added cost of the Hessian calculation.

2.5 Globally Convergent Method of Moving Asymptotes, GCMMA

Although the MMA procedure has shown to be extremely effective in most optimization problems, there are occasional scenarios in which the solution never converges due to the approximation being monotonic [44]. To alleviate these occurrences, modifications to the MMA have since been proposed to ensure global convergence via non-monotonic function approximations, the most popular being the GCMMA introduced by Svanberg [12, 45].

To ensure global convergence, the GCMMA approach suggests extending the MMA method by adding another series of iterations to the subproblem, which will be referred to as inner iterations, ℓ. The inner iterations guarantee that each approximate subproblem is conservative. The indexing for the iterations will be denoted by the outer and inner iteration, (k, ℓ).

To elaborate upon the methodology of this approach let us start with an initial guess of $\boldsymbol{x}^{(k,0)}$ to form the approximate subproblem. If the solution of the approximate subproblem, $\hat{\boldsymbol{x}}^{(k,0)}$, results in the approximate function evaluations to be greater than the original optimization function, $\tilde{g}_i^{(k,0)}(\hat{\boldsymbol{x}}^{(k,0)}) \geq g_i(\hat{\boldsymbol{x}}^{(k,0)})$, the approximation is conservative and satisfies the inequality constraints, providing a solution in the feasible design domain. The sequence of approximate subproblems will then continue onto the next outer iteration, $\boldsymbol{x}^{(k+1)}$, continuing the conservative check at each iteration until convergence is reached.

In the case that the conservative check is not satisfied, $\tilde{g}_i^{(k,0)}(\hat{\boldsymbol{x}}^{(k,0)}) < g_i(\hat{\boldsymbol{x}}^{(k,0)})$, an inner iteration will be made to create a new subproblem at $\boldsymbol{x}_i^{(k,1)}$, which will be more conservative than the previous subproblem due to the conservative parameter $\rho_i^{(k,v)}$. The inner iterations will incrementally increase until the subproblem satisfies the conservative check, which normally only takes a finite, small number of iterations.

The subproblem is described by the same formulation as the MMA, as seen in Equation 2.8, however the coefficients $p_{ij}^{(k,\ell)}$, $q_{ij}^{(k,\ell)}$, and $r_{ij}^{(k,\ell)}$ are now dependent on the inner iterations as follows:

$$\begin{aligned}
\min \quad & \tilde{g_0}^{(k,\ell)}(\boldsymbol{x}) = r_0^{(k,\ell)} + \sum_{j=1}^{n} \left(\frac{p_{0j}^{(k,\ell)}}{U_j^k - x_j} + \frac{q_{0j}^{(k,\ell)}}{x_j - L_j^k} \right) \\
\text{s.t.} \quad & \tilde{g}_i^{(k,\ell)}(\boldsymbol{x}) = r_i^{(k,\ell)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k,\ell)}}{U_j^k - x_j} + \frac{q_{ij}^{(k,\ell)}}{x_j - L_j^k} \right) \leq 0, \quad i = 1, ..., m \\
& \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad \alpha_j^k \leq x_j \leq \beta_j^k, \quad j = 1, ..., n \right\}.
\end{aligned} \tag{2.17}$$

The terms p_{ij}, q_{ij}, and r_i for $i = 0, ..., m$ are calculated using gradient information and parameters, σ_j^k and $\rho_i^{(k,\ell)}$.

$$\begin{aligned}
p_{ij}^{(k,\ell)} &= (\sigma_j^k)^2 \text{max}\left\{ 0, \frac{\partial f_i}{\partial x_j}(x^k) \right\} + \frac{\rho_i^{(k,\ell)} \sigma_j^k}{4} \\
q_{ij}^{(k,\ell)} &= (\sigma_j^k)^2 \text{max}\left\{ 0, -\frac{\partial f_i}{\partial x_j}(x^k) \right\} + \frac{\rho_i^{(k,\ell)} \sigma_j^k}{4} \\
r_i^{(k,\ell)} &= f_i(x^k) - \sum_{j=1}^{n} \frac{p_{ij}^{(k,\ell)} + q_{ij}^{(k,\ell)}}{\sigma_j^k}
\end{aligned} \tag{2.18}$$

The conservative parameter at the beginning of any outer iteration (when $\ell = 0$), $\rho_i^{(k,0)}$, is calculated by the following expression:

$$\begin{aligned}
\rho_i^{(1,0)} &= 1, \\
\rho_i^{(k+1,0)} &= \text{max}\left\{ 0.1\rho_i^{(k,\hat{\ell}(k))}, \rho_i^{\text{min}} \right\},
\end{aligned} \tag{2.19}$$

where ρ_i^{min} is a strictly positive small number. Here $\hat{\ell}(k)$ denotes the required number of inner iterations. For the following calculations of the conservative parameter, $\rho_i^{(k,\ell)}$, information from the solution of the most previous subproblem is used. The approximate subproblem can be rewritten in a new form to more easily determine the the next approxi-

mate value of $\rho_i^{(k,\ell)}$.

$$\tilde{g}_i^{(k,\ell)}(\boldsymbol{x}) = h_i^k(\boldsymbol{x}) + \rho_i^{(k,\ell)} d^k(\boldsymbol{x}) \tag{2.20}$$

where

$$d^k(\boldsymbol{x}) = \sum_{j=1}^{n} \frac{(U_j^k - L_j^k)(x_j - x_j^k)^2}{(U_j^k - x_j)(x_j - L_j^k)(x_j^{max} - x_j^{min})} \tag{2.21}$$

In an attempt find a $\rho_i^{(k,\ell+1)}$ that satisfies the conservative check at the next iteration, the following equation is applied:

$$h_i^k(\hat{\boldsymbol{x}}^{(k,\ell)}) + (\rho_i^{(k,\ell)} + \delta_i^{(k,\ell)}) d^k(\hat{\boldsymbol{x}}^{(k,\ell)}) = g_i(\hat{\boldsymbol{x}}^{(k,\ell)}), \tag{2.22}$$

where

$$\delta_i^{(k,\ell)} = \frac{g_i(\hat{\boldsymbol{x}}^{(k,\ell)}) - \tilde{g}_i^{(k,\ell)}(\hat{\boldsymbol{x}}^{(k,\ell)})}{d^k(\hat{\boldsymbol{x}}^{(k,\ell)})}. \tag{2.23}$$

This relationship implies that $\rho_i^{(k,\ell)} + \delta_i^{(k,\ell)}$ will be an reasonable estimate for $\rho_i^{(k,\ell+1)}$. The following equations use this information while preserving global convergence.

$$\begin{aligned} \rho_i^{(k,\ell+1)} &= \min\{1.1(\rho_i^{(k,\ell)} + \delta_i^{(k,\ell)})\ 10\rho_i^{(k,\ell)}\} \quad \text{if} \quad \delta_i^{(k,\ell)} > 0, \\ \rho_i^{(k,\ell+1)} &= \rho_i^{(k,\ell)} \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if} \quad \delta_i^{(k,\ell)} \leq 0 \end{aligned} \tag{2.24}$$

The σ_j^k term in the approximation functions are updated in a similar fashion to that of the move asymptotes where for the first two outer iterations, $k = 1, 2$,

$$\sigma_j^k = 0.5(x_j^{\max} - x_j^{\min}). \tag{2.25}$$

For the next outer iterations, the σ_i^k parameter is updated by,

$$\sigma_j^k = \gamma_j^k \sigma_j^{(k-1)}, \tag{2.26}$$

where

$$\gamma_j^k = \begin{cases} 0.7 & \text{if} \quad (x_j^k - x_j^{k-1})(x_j^{(k-1)} - x_j^{(k-2)} < 0 \\ 1.2, & \text{if} \quad (x_j^k - x_j^{k-1})(x_j^{(k-1)} - x_j^{(k-2)} > 0 \\ 1.0, & \text{if} \quad (x_j^k - x_j^{k-1})(x_j^{(k-1)} - x_j^{(k-2)} = 0. \end{cases} \tag{2.27}$$

This term must preserve the bounds,

$$0.01(x_j^{\max} - x_j^{\min}) \leq \sigma_j^k \leq 10(x_j^{\max} - x_j^{\min}). \tag{2.28}$$

An extensive proof of global convergence of the GCMMA can be found in [45]. Note that global convergence only implies a guaranteed convergence from any starting point to a local minimum and not to the global minimum. In most scenarios, the GCMMA has shown to perform slower and less efficiently in comparison to MMA, however, in cases in which the MMA has difficulty to reach a stable solution, the GCMMA may prove to be an effective alternative choice.

2.6 Further Extensions of the MMA

Extensions of the GCMMA have since been proposed to improve the curvature of the approximation by using non mixed second order sensitivity information instead of the non-monotonic parameter, this method is known as the second order GCMMA [46, 44]. Other extensions have focused on improving the GCMMA through obtaining gradient information from the previous subproblem iteration and applying it in place of the conservative parameter otherwise referred to as the Gradient Based Method of Moving Asymptotes (GBMMA) [47]. All of the discussed MMA variations can be solved using the techniques discussed in Chapter 3.0.

CHAPTER 3

SOLVING THE MMA SUBPROBLEM: PRIMAL-DUAL RELATIONSHIPS

There are several ways in which a solution to an MMA subproblem can be achieved. The MMA approximation formulation or otherwise referred to as the primal problem, obtains properties such as convexity and separability making it an ideal candidate for many numerical solution schemes. Of these solution strategies, duality will be discussed in detail in the context of solving the MMA subproblem.

3.1 Dual Solution Scheme

The primal optimization problem, $\mathbb{P}$, of the MMA approximation as defined in section 2.4 is equal to the following:

$$\mathbb{P} \quad \begin{cases} \min \quad \tilde{g}_0^k(\boldsymbol{x}) = r_0^k + \sum_{j=1}^n \left(\dfrac{p_{0j}^k}{U_j^k - x_j} + \dfrac{q_{0j}^k}{x_j - L_j^k} \right) \\ \text{s.t.} \quad \tilde{g}_i^k(\boldsymbol{x}) = r_i^k + \sum_{j=1}^n \left(\dfrac{p_{ij}^k}{U_j^k - x_j} + \dfrac{q_{ij}^k}{x_j - L_j^k} \right) \leq 0, \quad i = 1, ..., m \\ \qquad \boldsymbol{x} \in \boldsymbol{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \quad \alpha_j^k \leq x_j \leq \beta_j^k, \quad j = 1, ..., n \right\}, \end{cases} \tag{3.1}$$

where the $\tilde{g}_i^k$ terms are convex and continuously differentiable functions. In many cases, the primal problem may be difficult to solve directly using the optimality conditions, Karush-Kuhn-Tucker (KKT) conditions [48], due to possible nonlinearity of the functions and large number of design variables and constraints which would make its implementation computationally expensive. For nonconvex problems the KKT conditions are necessary but not sufficient conditions in which the Lagrangian function obtains a local optima (minima or maxima). In the case of that the problem is convex and that Slater's constraint qualification is satisfied [22], such as in the case of the primal problem, $\mathbb{P}$, generated by the MMA ap-

proximation, the KKT conditions are both necessary and sufficient for finding the optimal point [22]. In addition, the primal problem's convexity property implies that the following dual problem, $\mathbb{D}$ is equivalent to $\mathbb{P}$.

$$\mathbb{D} \quad \begin{cases} \max\limits_{\lambda} & f(\boldsymbol{\lambda}) \\ \text{s.t.} & \lambda_i \geq 0, \quad i = 1, ..., m \end{cases} \tag{3.2}$$

where

$$f(\lambda) = \min_{x \in X} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \min_{x \in X} \left\{ \tilde{g}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \tilde{g}_i(\boldsymbol{x}) \right\} \tag{3.3}$$

The dual objective function is the Lagrangian function, $\mathcal{L}$, composed of the original objective function of the primal problem and the addition of the constraints with an associated Lagrangian multiplier, λ_i. The dual problem is subjected to the constraints that the λ_i be nonnegative.

It is beneficial to reformulate the primal subproblem into the dual subproblem because the constraints are now simpler to handle in this setting such that m is a relatively small, finite number. Now considering the n design variables the Lagrangian function takes the form:

$$\mathcal{L}_j(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{j=1}^{n} \mathcal{L}_j(x_j, \boldsymbol{\lambda}) = \sum_{j=1}^{n} \left\{ (\tilde{g}_{0j}(x_j) + \sum_{i=1}^{m} \lambda_i \tilde{g}_{ij}(x_j) \right\}. \tag{3.4}$$

Thus transforming the dual objective function into,

$$f(\lambda) = \min_{x \in X} \sum_{j=1}^{n} \mathcal{L}_j(x_j, \boldsymbol{\lambda}). \tag{3.5}$$

Due to the separability and convex characteristics of the Lagrangian function, the relationship between the Lagrangian multipliers and design variables can be determined by applying the optimality condition to the dual function:

$$\frac{\partial \mathcal{L}_j(x_j, \boldsymbol{\lambda})}{\partial x_j} = 0 = \frac{\sum_{i=0}^{m} \lambda_i p_{ij}}{(U_j - x_j)^2} - \frac{\sum_{i=0}^{m} \lambda_i q_{ij}}{(x_j - L_j)^2}, \tag{3.6}$$

where $\lambda_0 = 1$. Rearranging this expression we obtain:

$$x_j(\lambda) = \frac{U_j + \eta L_j}{\eta + 1}, \tag{3.7}$$

where

$$\eta = \sqrt{\frac{\sum_{i=0}^{m} \lambda_i p_{ij}}{\sum_{i=0}^{m} \lambda_i q_{ij}}}. \tag{3.8}$$

This term will be substituted back into the Lagrangian where the $\boldsymbol{\lambda}$ terms will be determined by using the bisection method or alternatively by a direct method [49].

The optimal values of the Lagrange multipliers $\boldsymbol{\lambda}^*$ can be used in the expression for $x_j(\lambda)$ (Equation 3.7) to determine the values of the optimal design variables. The design variables must go through a final check in order to satisfy the box constraints such that:

$$\begin{aligned} x_j(\lambda) &= x_j^* \quad \text{if} \quad \alpha_j \leq x_j^* \leq \beta_j \\ x_j(\lambda) &= \alpha_j \quad \text{if} \quad x_j^* \leq \alpha_j \\ x_j(\lambda) &= \beta_j \quad \text{if} \quad x_j^* \geq \beta_j. \end{aligned} \tag{3.9}$$

Once found, these updated design variables, $x_j(\lambda)$, will either not satisfy the convergence criteria and be used to continue the next MMA subproblem or they will satisfy the convergence criteria, completing the solution scheme.

Alternative approaches for solving the approximate subproblem include a primal-dual interior point method proposed by Svanberg in the context of the MMA and GCMMA subproblem (for an efficient Matlab implementation) [43]. This is the method that is used to solve the MMA sequence of subproblems in the sizing optimization and topology optimization examples in the following Chapter. In the commercial software, PLATO, the problems are solved using a trust region method [50].

CHAPTER 4

NUMERICAL RESULTS

In this section we will examine the MMA in several optimization settings. In Section 4.1 a one-dimensional function will be analyzed considering multiple variations of move asymptotes to explore how they impact the curvature of the approximation and how under certain values they even replicate other sequential convex approximation schemes. Section 4.2 studies a sizing optimization problem comparing the solution achieved analytically, with a Matlab package CVX [51], and by the MMA. This example also investigates the relationship between the MMA empirical parameters with the move limits and asymptotes. In Section 4.3 the MMA is examined in the compliance minimization topology optimization problem where the impact of the empirical parameters on the final topology is explored as well as comparing the MMA to other update schemes. The Section 4.4 example examines the use of the MMA in the stress constrained topology optimization framework, elaborating on the interplay between the MMA iterations and the augmented Lagrangian steps. Lastly in Section 4.5 the application of the MMA in the PLATO software is explored in the three-dimensional topology optimization setting.

4.1 One-Dimensional Function

In this first example the MMA is explored in the setting of a one-dimensional function:

$$f(x) = -x \sin x. \tag{4.1}$$

The approximation functions are explored at two different points, $x^0 = 1.5$ and $x^0 = 4.5$, with $x \in [0, 6]$. At each point, two MMA approximations are evaluated with lower and upper asymptotes of $L_1^0 \to -\infty,\ U_1^0 \to +\infty$ and $L_1^0 = 0,\ U_1^0 \to +\infty$ to replicate SLP

and CONLIN approximations respectively (see Appendix A). The first point $x^0 = 1.5$ has a negative gradient by $\partial f(x)/\partial x = -\sin x - x\cos x$ and $\partial f(1.5)/\partial x = -1.1036$, thus the MMA approximation acting as CONLIN will have the form of a reciprocal approximation as described in Section 2.3. The MMA approximation acting as an SLP function takes a linear form as expected. Both of these approximations, as seen in Figure 4.1, demonstrate the flexibility of the MMA in being able to approximate functions of many forms by modifying only the move asymptotes.

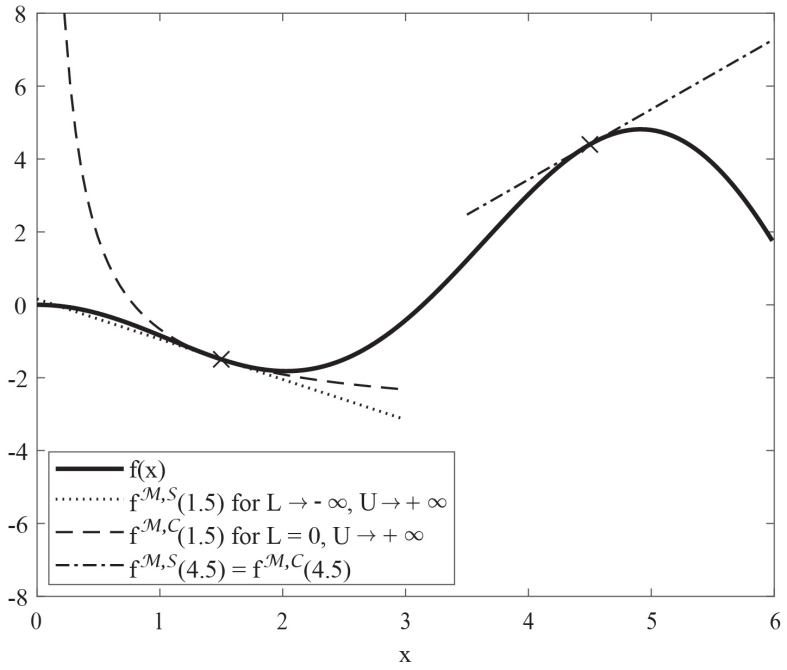


Figure 4.1: MMA approximations of a 1D function implementing asymptotes which replicate SLP and CONLIN behavior

The same MMA function approximations are evaluated at $x^0 = 4.5$ which has a positive gradient of $\partial f(4.5)/\partial x = 1.9261$. According to the CONLIN definition, because the function has a positive gradient it will be approximated by a linear function thus both MMA approximations take the same linear form (see Figure 4.1). This illustrates how CONLIN (or in this case MMA acting as CONLIN) is the most conservative possible approximation using linear and reciprocal variables [41]. This study is illustrated in Figure 4.1 where $f(x)$

represents the actual actual function, $f(x)^{\mathcal{M},\mathcal{S}}$ represents the MMA which replicates SLP behavior, and $f(x)^{\mathcal{M},\mathcal{C}}$ represents the MMA which replicates CONLIN behavior.

Using the same one-dimensional function, the MMA approximations are now made around $x^0 = 3$ with a lower asymptote of $L_1^0 = 0$ and four different upper asymptotes, $U_1^0 \to +\infty$, $U_1^0 = 10$, $U_1^0 = 4$, and $U_1^0 = 3.25$. For the first approximation ($L_1^0 = 0$, $U_1^0 \to +\infty$) the MMA takes the form of the CONLIN approximation as studied previously. Again it is seen that because the function has a positive gradient at $x^0 = 3$, $\partial f(3)/\partial x = 2.8289$, the CONLIN will act as a linear approximation. In the following approximations as the upper asymptotes are reduced so are the move asymptote intervals. As seen in Figure 4.2, as the move asymptotes move closer to each other the approximations become more conservative. This demonstrates the capability of the MMA technique for approximating functions to any degree of conservatism.

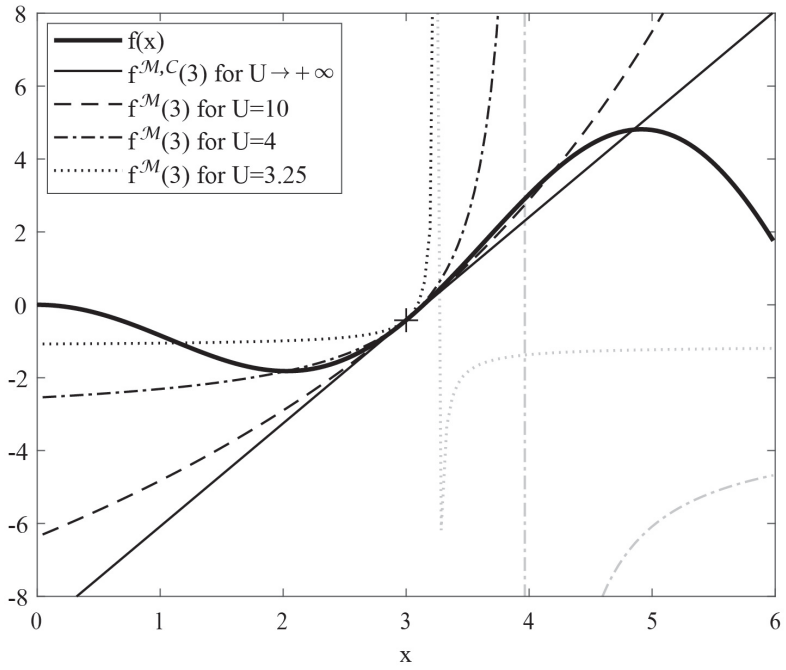


Figure 4.2: MMA approximations of a 1D function considering multiple move asymptote magnitudes

4.2 Sizing Optimization

The next problem to be analyzed is a two dimensional optimization problem. The problem is defined as a sizing optimization problem, which is a type of structural optimization in which the only design variables are cross sectional area parameters. Here the design variables represent the cross sectional width dimensions, x_1 and x_2, of a two segment cantilever beam which is fixed on the left end and subject to a point load on the right, free end [22].

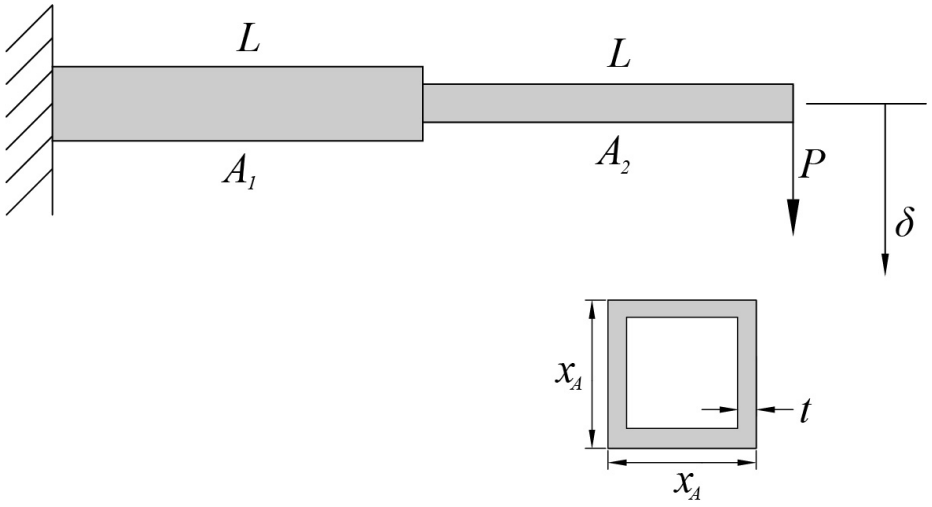


Figure 4.3: Cantilever beam design domain

We optimize the structure such that the weight is minimized considering a displacement inequality constraint and box constraints about the design variables.

$$\begin{aligned} \text{min} \quad & f(x_1, x_2) = x_1 + x_2 \\ \text{s.t.} \quad & g_1(x_1, x_2) = \frac{1}{x_1^3} + \frac{7}{x_2^3} - 1 \leq 0 \\ & 0.1 \leq x_1 \leq 10, \quad 0.1 \leq x_2 \leq 10 \end{aligned} \tag{4.2}$$

4.2.1 Analytical Solution

Due to the mathematical simplicity of the problem, the solution can be readily achieved analytically. This will be executed and used to compare against the optimal solutions obtained by the MMA subroutine. The analytical solution is achieved by assuming that the inequality constraint is active by setting it as an equality. This equality can then be rewritten by solving for one design variable as a function of the other as done for the design variable, x_2, below.

$$x_2 = \sqrt[3]{\frac{7}{1 - x_1^{-3}}} \tag{4.3}$$

Equation 4.3 will then be substituted back into the objective function where we calculate the gradient of the function and set it equal to zero to obtain an optimal solution at the stationary point,

$$\frac{\partial f}{\partial x_1} = 0 = 1 - \sqrt[3]{7} x_1^{-4} \left(\frac{1}{1 - x_1^{-3}} \right)^{4/3}. \tag{4.4}$$

Once solved, we obtain the solution to the optimization problem:

$$x_1^* = \sqrt[3]{1 + 7^{1/4}} \approx 1.3797 \quad x_2^* = \sqrt[3]{7 + 7^{3/4}} \approx 2.2442,$$

with a final objective value of

$$f(\boldsymbol{x}^*) \approx 3.6240.$$

Switching Objective and Constraint

Now let's examine the solution if the objective and constraint functions were reversed for an optimization objective to minimize deflection subject to a weight constraint, where W

is some arbitrary weight limit.

$$\begin{aligned} \min \quad & f(x_1, x_2) = \frac{1}{x_1^3} + \frac{7}{x_2^3} \\ \text{s.t.} \quad & g_1(x_1, x_2) = x_1 + x_2 \leq W \\ & x_1 > 0, \quad x_2 > 0 \end{aligned} \tag{4.5}$$

Following the same procedure as described above we reach the following solution.

$$x_1^{**} \approx 0.3807W \quad x_2^{**} \approx 0.6193W$$

It is observed from evaluating the solutions in both cases, that the ratios between the design variables are equivalent

$$\frac{x_2^{**}}{x_1^{**}} = \frac{x_2^*}{x_1^*} \approx 1.6267,$$

which brings to light a scaling property in which the solution of one problem can be found by scaling the solution of the other. This holds true for cases in which the box constraints are not active and that the optimal solutions satisfy the KKT conditions.

4.2.2 Solution by CVX

As an additional form of verification to the solution of this problem, the sizing optimization problem was solved in CVX, a Matlab software for convex programming [51, 52]. The CVX software had returned the optimal solution at $x_1 = 1.3737$ and $x_2 = 2.2442$ for a optimal objective function value of $f(\boldsymbol{x}^*) = 3.62399$ in agreement with the analytical solution previously found.

4.2.3 Solution by MMA

We now solve the same optimization formulation as in Equation 4.2 using the MMA subroutine. The optimizer parameters are set such that $s_{init} = 0.1$, $s_{faster} = 1.2$, $s_{slower} = 0.5$, $C_4 = 0.5$, and $\mu = 0.1$. The initial guess is defined by $x_1^0 = x_2^0 = 5$ and 10 iterations are run. Figure 4.4 demonstrates the convergence of the design variable, x_2, with its associated move limits and asymptotes. The convergence of the objective function and the constraint are shown in Figure 4.5.

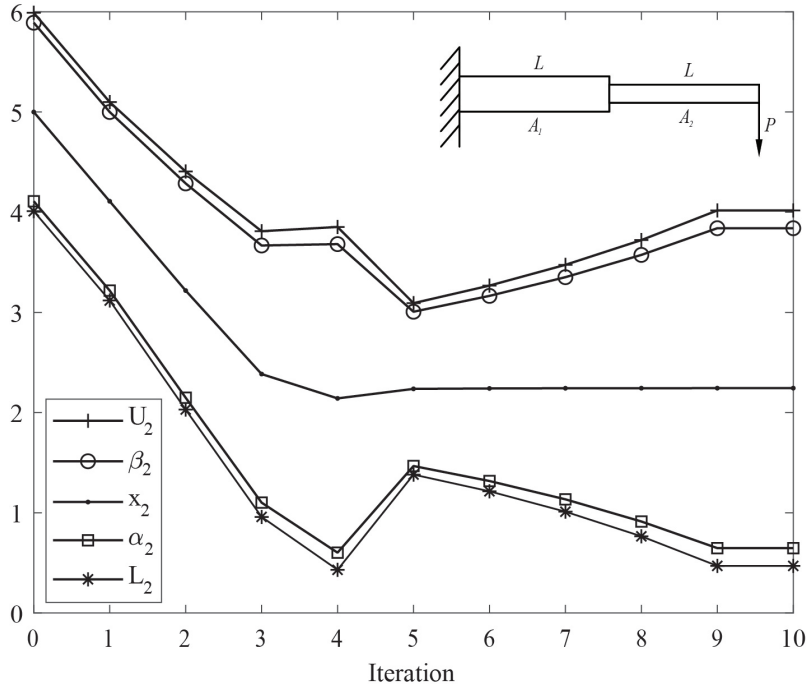


Figure 4.4: Convergence history of the design variable, x_2

The solution achieved by the MMA with this set of parameters was reported as $\boldsymbol{x}^* = (1.3799, 2.2441)$ at the end of the tenth iteration, which is extremely close to that of the analytical solution. The chosen empirical parameters set large search intervals for the design variables as per the definition of the move limit and asymptote bounds. By examining Figure 4.4, it is seen that the move asymptotes have the same range for the first two iterations as expected (see Equation 2.13). On the third iteration, the descent of the design

variable, x_2, in the past iterations results in move asymptotes that move farther apart for a less conservative approximation in effort to speed up the convergence. This continues until x_2 oscillates and thus the move asymptotes and limits are brought closer together in the fifth iteration for more conservative bounds. After this is done the design variable converges to the optimal solution.

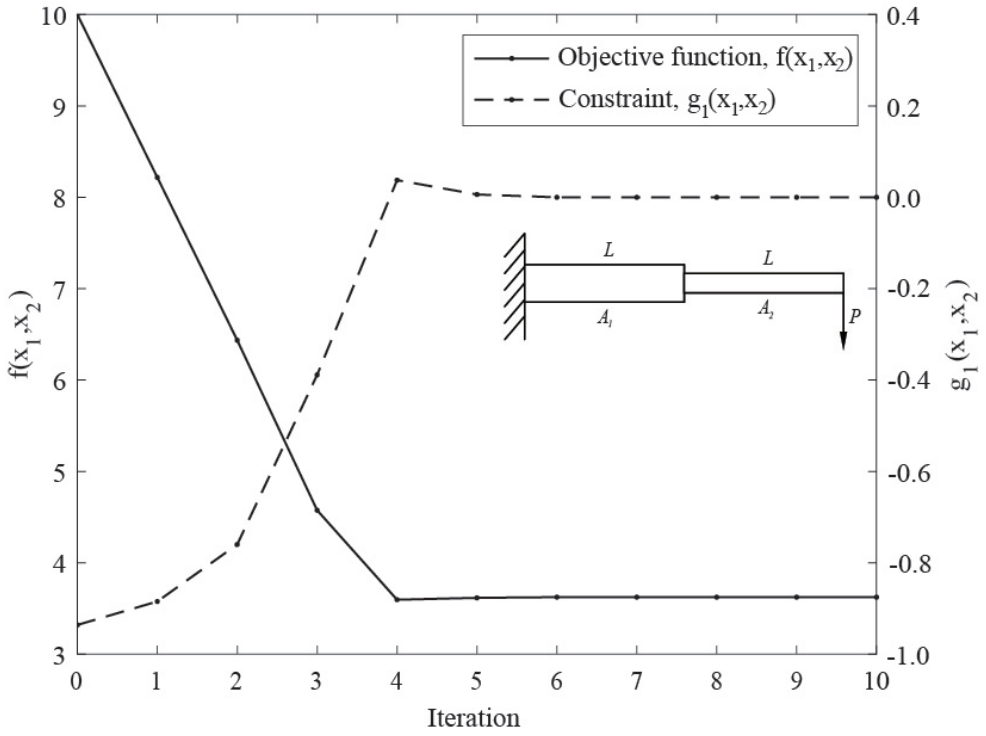


Figure 4.5: Convergence of the objective function and constraint

As seen in Figure 4.5, the history of the objective and constraint have a direct relationship to the history of the design variable. At the fourth iteration, where the move limits have a large range, the inequality constraint it violated. However once the approximation becomes more conservative at the next iteration the design variable corrects, enforcing the objective and constraint to converge smoothly with an inequality constraint that becomes active, $g_1(\boldsymbol{x}^*) = 0$, and the objective which is minimized to $f(\boldsymbol{x}^*) = 3.6240$, same as the analytical solution and solution by CVX.

The complete history on the convergence of the design variables is visualized in Figure 4.6. The initial guess of the design variables is seen at $(5, 5)$ in the feasible domain.

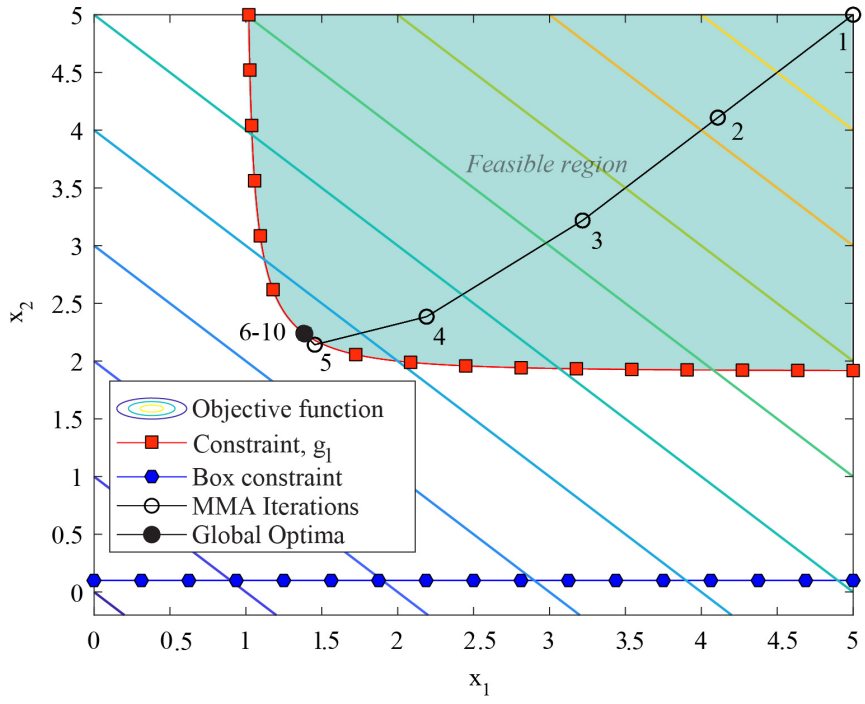


Figure 4.6: Contour plot of the sizing optimization problem

Iteratively, at each solution of the MMA subroutine, the design variables move closer toward the optimal solution. Near the fifth iteration, the solution converges in agreement with the previous convergence plots.

Next the empirical parameters are defined to achieve a more conservative approximation from the start of the subroutine. This is done by tightening the bounds for the lower and upper asymptotes through setting, $s_{init} = 0.05$, $s_{faster} = 1.1$, $s_{slower} = 0.4$, and $C_4 = 0.3$, which are all decreased from the previous MMA sizing optimization problem.

Expectedly, as demonstrated in Figure 4.7, the bounds of the move asymptotes are tighter throughout the convergence history and do not increase by the same order of magnitude as in the previous example. The more conservative move asymptotes are a result of the decreased s empirical parameters and hence the move limits are also more conservative. It is discovered that as a result of the conservative approximations, the design variables take longer to converge. In this example x_2 doesn't oscillate until the eighth iteration versus the fifth in the previous example. Also the final solution at the end of the fifteenth iteration is

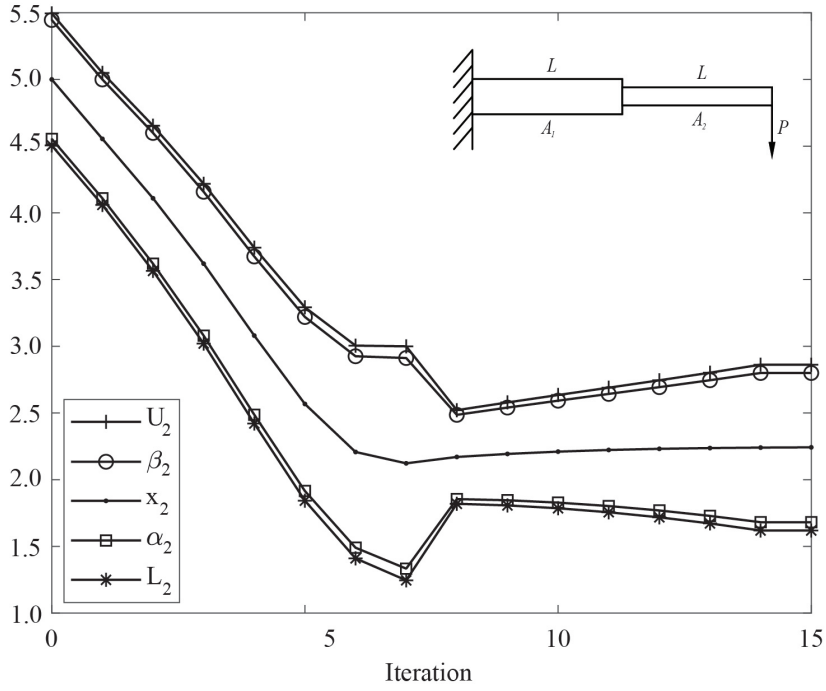


Figure 4.7: Convergence history of the design variable, x_2, by a more conservative MMA approximation

$\boldsymbol{x}^* = (1.3815, 2.2425)$ which is close to the analytical solution however the previous, less conservative approximation achieved a solution more near the analytical solution at the end of only the tenth iteration.

The convergence of the objective and constraint functions is more gradual and smooth when applying an approximation of greater conservatism, as seen in Figure 4.8 where at the final iteration $f(\boldsymbol{x}^*) = 3.6240$ and $g_1(\boldsymbol{x}^*) = 0$. When x_2 oscillated the inequality constraint was violated as in the previous MMA procedure. However, in this more conservative approximation the inequality constraint was not violated nor did the design variable oscillate to the same extent as before. Applying a more conservative approximation by the MMA will increase the computational time especially considering large scale problems, however in some cases it may be pragmatic to avoid excessive oscillation and ensure steady convergence.

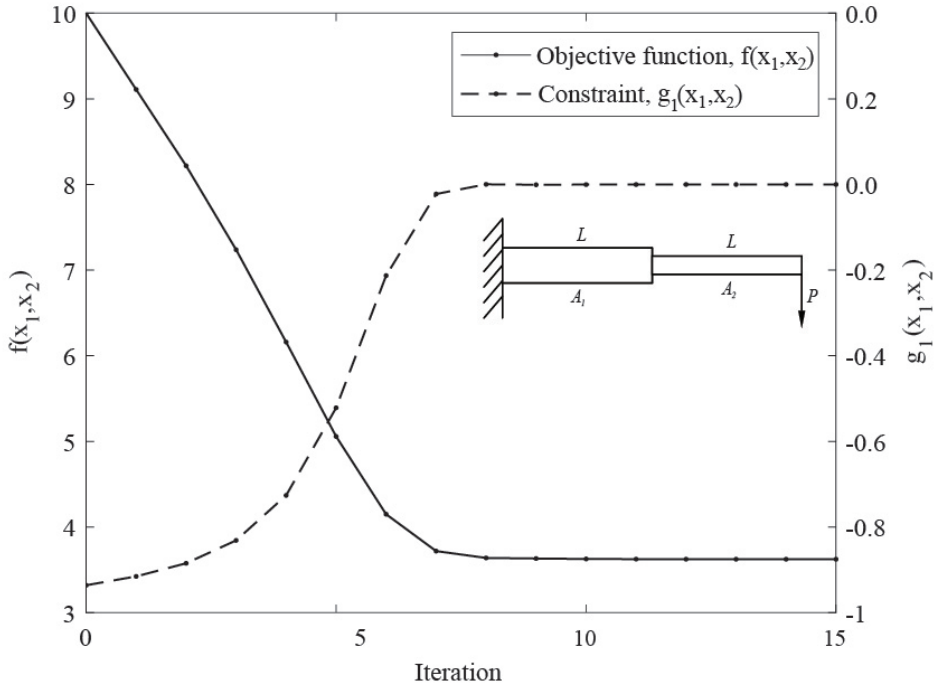


Figure 4.8: Convergence of the objective function and constraint by a more conservative MMA approximation

In the final variation of the sizing optimization problem, the MMA subproblem is defined by the same empirical parameters from the first cantilever example except with one modification, increasing s_{init} from 0.1 to 0.5. This examples serves as a reminder of the impact each empirical parameter has on the approximation and performance of the MMA subroutine, thus it is important to have a thorough understanding on how these empirical parameters influence the approximation scheme and hence the solution. The increase of s_{init} created much larger bounds of the initial move asymptotes which consequently effected the the following move asymptote bounds. The lack of conservatism led to lots of oscillation of the objective, constraint, and design variables in the initial iterations. However, this example also demonstrates the robustness of the MMA procedure such that even if poor selection of empirical parameters are chosen, in most cases the MMA will eventually reach a stable convergence and feasible solution.

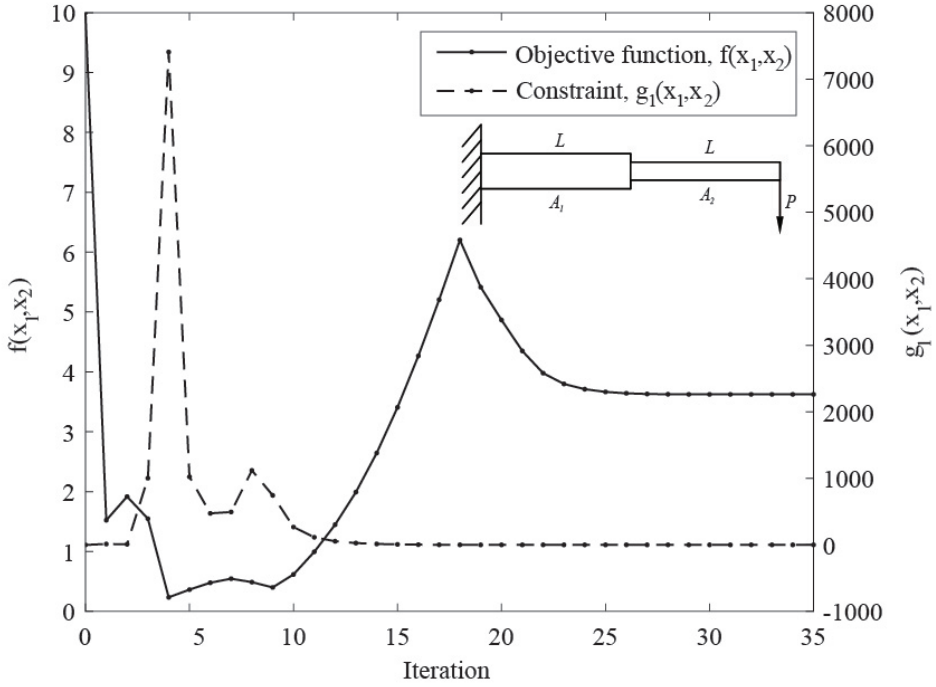


Figure 4.9: Convergence of the objective function and constraint by ill-chosen empirical parameters

4.3 Compliance Minimization Topology Optimization

In this example, sequential approximation update schemes are studied in a topology optimization problem in the continuum setting considering the classical topology optimization framework, compliance minimization subject to a single, linear volume constraint [53].

$$\begin{aligned} \min_{\rho} \quad & f(\rho, \boldsymbol{u}) = \int_{\bar{\Gamma}_N} \boldsymbol{t} \cdot \boldsymbol{u} ds \\ \text{s.t.} \quad & g(\rho) = \frac{1}{|\Omega|} \int_{\Omega} m_V(\rho) d\boldsymbol{x} - \bar{v} \leq 0 \\ & \rho_i \ \in \ [0,1] \quad i = 1, .., m \\ \text{with} \quad & \boldsymbol{K}(\rho)\boldsymbol{u} = \boldsymbol{F} \end{aligned} \tag{4.6}$$

In this formulation the objective function of compliance is defined by the design variable of element densities via a continuous parametrization function and response variable vector of displacements, ρ and $\boldsymbol{u}$. The compliance is found by integrating the dot product of the traction, $\boldsymbol{t}$, with $\boldsymbol{u}$ over the space where nonzero tractions are applied, $\bar{\Gamma}_N$. The volume constraint, $g(\boldsymbol{\rho})$, is defined about the entire domain, Ω, in which $m_V(\boldsymbol{\rho})$ represents the volume interpolation function and $\bar{v}$ the volume fraction. The densities are penalized by the material interpolation function following the Solid Isotropic Material with Penalization (SIMP) formulation to primarily take values of 0 or 1. The state equation governs linear elastic behavior.

In topology optimization, the continuum setting discretizes the design domain into thousands or millions of elements to perform finite element analysis to determine the element densities. The intermediate densities are then interpolated by a single-material interpolation function, such as the SIMP, to make the final topology continuous and discrete [54]. When considering the classical topology optimization problem, the OC method has demonstrated rapid convergence with the implementation of a damping parameter proposed in the augmented, heuristic version by Bendsoe [55]. This method was also demonstrated to obtain identical behavior to sequential convex approximations of exponential and reciprocal intervening variables [56]. Due to the efficiency of the method it has been widely popularized when solving this class of compliance minimization problems. However the OC method obtains severe limitations when solving problems considering more than one constraint or non-monotonic formulations. To accommodate for solving non-self-adjoint problems, a new update scheme, named the sensitivity-separation method, uses an approximation based on the sum of non-monotonic functions for a construction enabled by sensitivity separation [57]. However the MMA scheme remains as the most robust solver, which can handle both multiple constraints and non-self-adjoint formulations.

The following numerical example will evaluate an MBB beam using the educational code, PolyTop [53]. The first part of the example will serve to explore the impact of the

MMA empirical parameters on the final optimized topology in a compliance minimization, continuum setting. The second study will compare the topological solutions achieved by the MMA, OC, and sensitivity-separation update schemes.

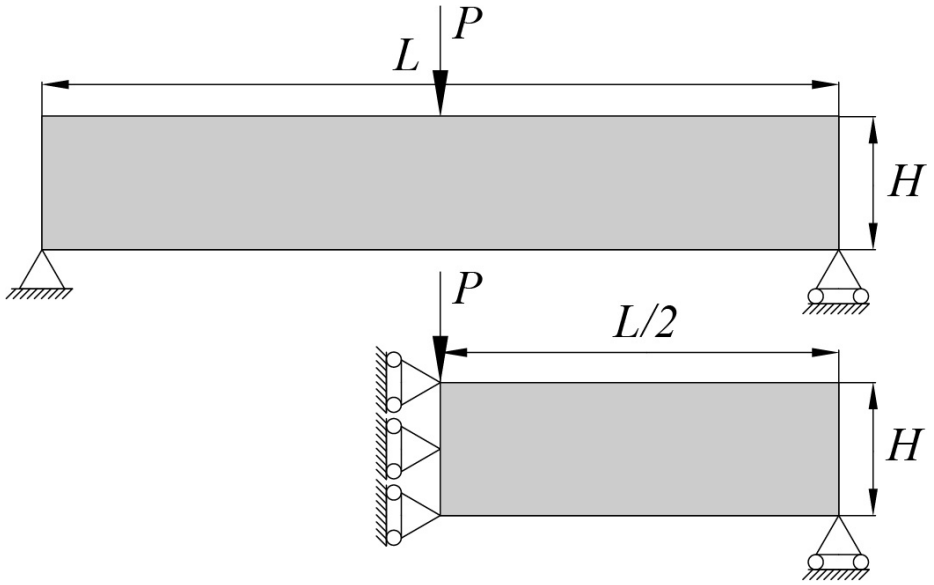


Figure 4.10: Design domain of the MBB beam

The structure is simply supported, with dimensions of $L = 6$ m and $H = 1$ m, subject to a downward load of $P = 0.5$ N located mid-span of the beam (see Figure 4.10). In the first study half of the MBB beam domain is analyzed considering 10,000 polygonal finite elements [58], a constant SIMP penalization term $p = 3$, a filter radius of $R = 0.04$ m, a volume fraction of $\bar{v} = 0.5$, a poisson's ratio of $v = 0.3$, and $k = 150$ iterations. A series of six different variations of MMA empirical parameters were evaluated, as seen in Table 4.1, to explore the affects of changing these parameters has on the topology and to what degree. All examples are run on Matlab R2019a on a desktop computer with Intel(R) Xeon(R) CPU E5-1660 v3, 3.00 GHz processor with 64.0 GB of RAM.

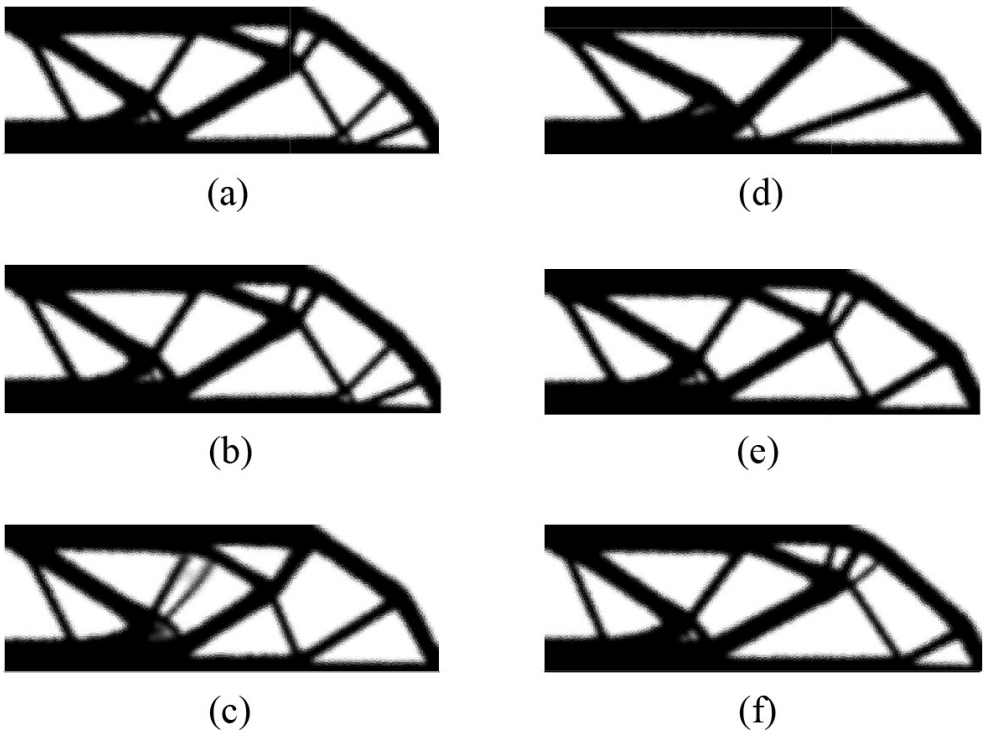


Figure 4.11: Optimized results of the MBB beam considering various MMA parameters

	s_{init}	s_{faster}	s_{slower}	C_4	$f(\boldsymbol{x}^*)$
a	0.5	1.2	0.7	0.5	50.772
b	0.5	1.2	0.55	0.5	50.685
c	0.1	1.1	0.1	0.1	51.02
d	0.99	5	0.5	0.5	50.411
e	0.1	1.2	0.55	0.2	50.364
f	-same as e, obj scaled by 10-				5.025

Table 4.1: MMA empirical parameters and final optimal objective values

In Figure 4.11a a standard set of empirical parameters were chosen resulting in an objective of 50.772 at the end of the final iteration. By slightly reducing the s_{slower} parameter in Figure 4.11b the performance improves slightly with a lower final objective of 50.685 resulting in the same topology as in the Figure 4.11a. In Figure 4.11c the empirical parameters are chosen to enforce an extremely conservative MMA approximation, as expected this leads to a higher final objective value of 51.02 with the objective function arriving at a new local minima for a different topology than the previous examples. In Figure 4.11d the empirical parameters are chosen such that the MMA scheme is a non-conservative approx-

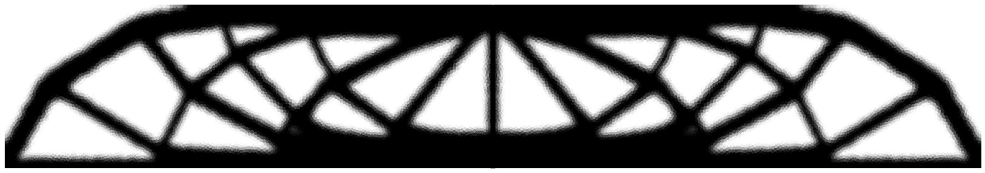
imation. Because the objective function of compliance under the SIMP penalization term $p = 3$ is highly nonlinear, the objective function yet again reaches another local minima for a different final topology achieving the lowest objective function yet of 50.411. The next selection of the empirical parameters in Figure 4.11e influences the initial bounds to be conservative but obtains customary bound update parameters, this results in a topology similar to that in the Figure 4.11a but with a slightly lower objective than all the previous examples at 50.364 indicating an increased performance. The final simulation keeps the same empirical parameters as the previous but scales the objective function and its gradients. As described by Svanberg's implementation of the MMA into Matlab [43], the algorithm works most efficiently when the objective function value is within the range $1 - 100$. Although the current objective value is within this range we consider a scaling factor of 10 to explore the effects of scaling on the optimization performance. The results of the scaling in Figure 4.11f includes a topology with a new feature in the top right corner which differentiates itself from the topology in part e. The scaled final objective value is 5.025 for an unscaled final objective of 50.25 the lowest of all the examples run. These simulations all demonstrated the impact of the selection of the MMA empirical parameters has on the topology optimization problem. It also demonstrated that there is no intuitive way for selecting these parameters for improved performance other than by trial and error and with comparison of these previous sizing optimization example in section 4.2 we see how appropriate parameter selection is a problem-dependent issue.

If we examine the topology in Figure 4.11c, it is recognized that the solution has not yet converged with grey regions remaining in the center of the beam. This is due to the problem being solved by a conservative MMA approximation, as defined by the empirical parameters, which implies a slower convergence. When increasing the iterations to $k = 200$, we notice that the topology converges to a more discrete solution, see Figure 4.12, with a final objective of 50.636.

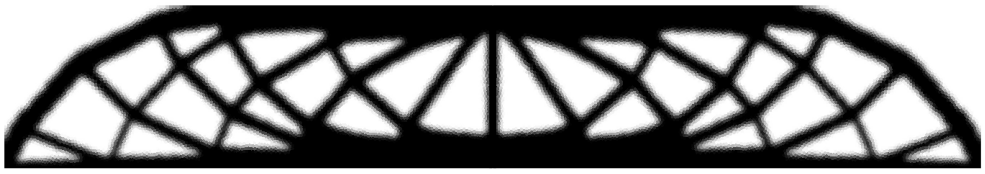
k = 150 k = 200

Figure 4.12: MBB beam topology achieved by $k = 150$ and $k = 200$

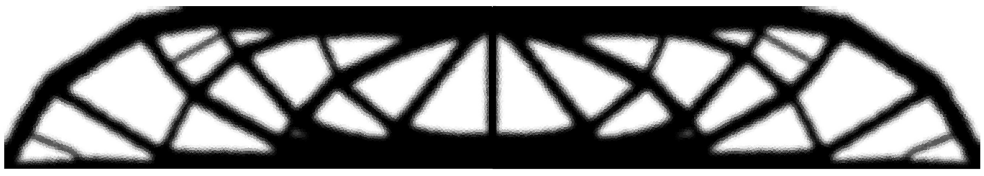
The MBB beam is now studied under SIMP continuation as p increases from 1 to 4 at increments of 0.5, holding the same parameters as the previous example, for three different update schemes; the OC, the MMA, and the sensitivity-separation scheme. The update scheme completes for each penalization term when either the change in the two most previous solutions is less than the tol $= 0.01$ or when the iteration count reaches the MaxIter $= 150$.



(a)



(b)



(c)

Figure 4.13: Topology generated by the update scheme (a) OC (b) MMA (c) sensitivity-separation

The solutions generated by the different update schemes in the continuation setting all obtain similar topologies as seen in Figure 4.13. When the OC update scheme was used it took 917 iterations to converge to the final objective of $f(\boldsymbol{x}^*) = 52.253$ with a total CPU time of 6.65 minutes with little of the time designated to the update scheme. For the MMA update scheme empirical parameters of $s_{init} = 0.5$, $s_{faster} = 1.2$, $s_{slower} = 0.55$, and $C_4 = 0.5$ were selected. The solution of $f(\boldsymbol{x}^*) = 53.157$ was achieved after 832 iterations with a final CPU time of 7.40 minutes now with a considerable percentage of the time dedicated to computing the update scheme. In the solution achieved by the sensitivity-separation scheme, a total of 951 iterations were needed to converge to a final objective of $f(\boldsymbol{x}^*) = 52.847$ with a total CPU time of 6.88 minutes, similar to the OC in total time and small CPU time required for the update scheme. When comparing the convergence history in Figure 4.14 and the CPU time in Figure 4.15 , the MMA may require less iterations in comparison to the other solution strategies however those iterations are more computationally involved during the update scheme as seen in Figure 4.15 where the CPU time per iteration is compared among the different update schemes. Although the MMA approximation may not be the most efficient solver in this classical topology optimization framework, in cases of more complex optimization formulations, the MMA may be the only feasible update scheme that leads to a solution as to be demonstrated in the following example considering a stress constrained topology optimization formulation.

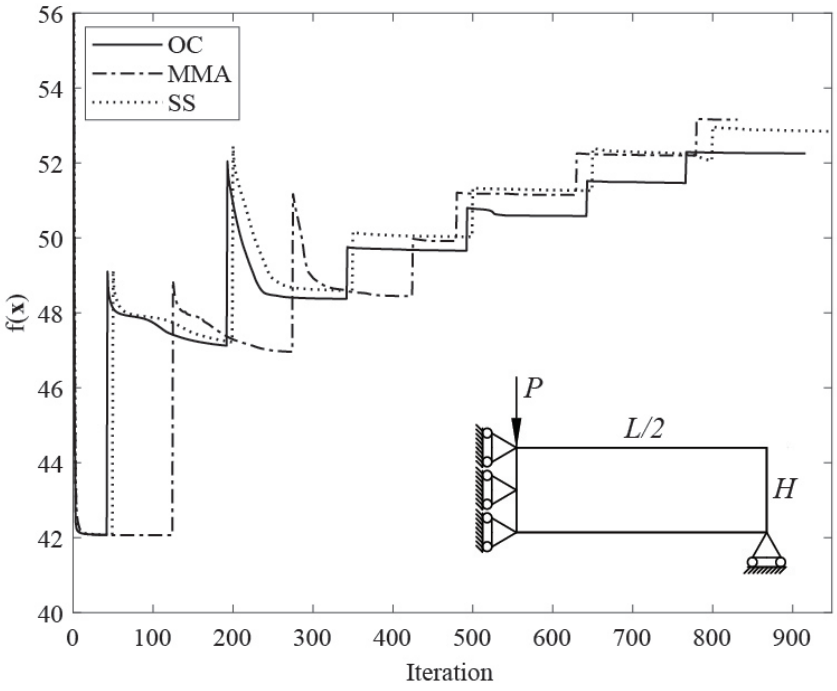


Figure 4.14: Convergence history of the objective function by solution schemes the OC, MMA, and sensitivity-separation

4.4 Stress Constrained Topology Optimization

In this example the use of the MMA approximation is explored in a more complex optimization setting, the stress-constrained problem. Here we study the stress-constrained problem solved by an augmented Lagrangian method, aggregation-free approach which handles local stress constraints [13, 15]. The optimization formulation is defined by the following:

$$\begin{aligned} \min_{\rho} \quad & f(\rho) = \frac{1}{|\Omega|}\int_{\Omega} m_V(\rho) d\boldsymbol{x} \\ \text{s.t.} \quad & g_j(\rho, \boldsymbol{u}) = m_E(\rho)\Lambda_j(\Lambda_j^2 + 1) \leq 0, \quad j = 1, .., n \\ \text{with} \quad & \Lambda_j = \sigma_j^v/\sigma_{lim} - 1 \\ & \boldsymbol{K}(\rho)\boldsymbol{u} = \boldsymbol{F}, \end{aligned} \tag{4.7}$$

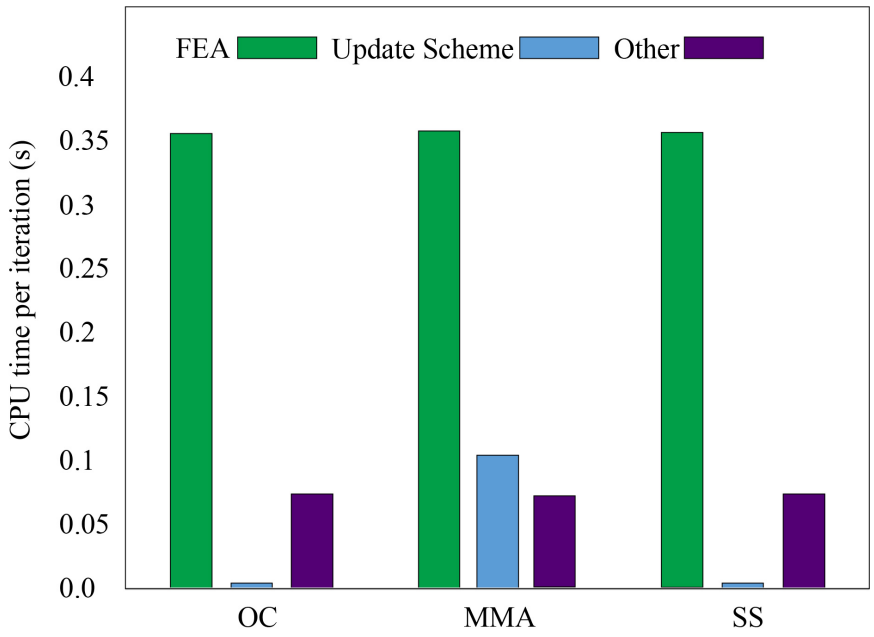


Figure 4.15: CPU time for the MBB topology optimization problem solved by OC, MMA, and the sensitivity-separation

where now the objective is to minimize the mass ratio or volume fraction subjected to n local stress constraints, one per finite element. The stress constraints are defined by the material interpolation function, $m_E(\rho)$, and Λ_j which is a function of the von Mises stress measured by the Cauchy stress tensor and the stress limit of the material. For this study only linear elastic material was considered however, the PolyStress work can also handle nonlinear material constitutive models [15].

The augmented Lagrangian based approach reformulates the optimization statement into an Lagrangian function written as the objective of mass ratio summed with all the local stress constraints and their associated Lagrange multipliers. This method, coupled with appropriate scaling of the constraints, allows us to handle the large quantity of local stress constraints. The augmented Lagrangian function is solved by a series of solutions to the MMA approximate which are found explicitly. The solution strategy considers i outer iterations for solving the augmented Lagrangian function and k inner iterations for minimizing the augmented Lagrangian by the MMA. The flowchart in Figure 4.16 depicts this procedure.

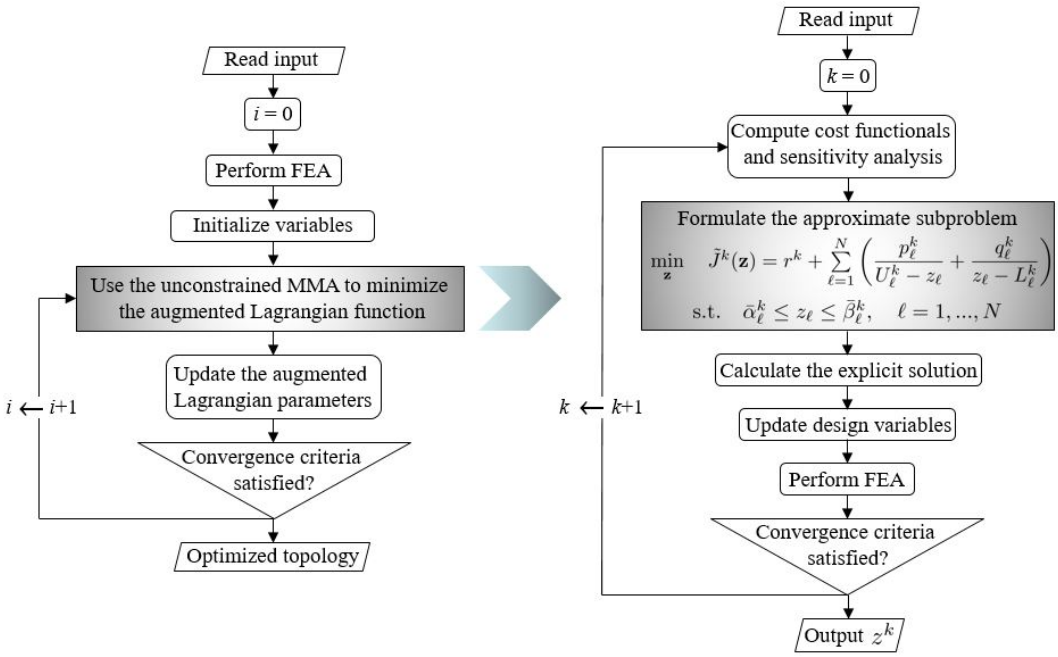


Figure 4.16: Optimization procedure solving the stress constrained problem using the augmented Lagrangian approximated by the unconstrained MMA [15]

The following stress constrained example evaluates an L-bracket domain which is pinned along the top edge and subjected to a downward point load on several nodes on the edge, see Figure 4.17a. The domain has a length of $L = 1$ m, a load $P = -0.25$ N, 10,000 elements, a filter radius of $R = 0.05$ m, a constant penalization term $p = 3.5$ m, a tol $= 0.002$, 5 maximum MMA inner iterations k, and a total augmented Lagrangian steps of $i = 150$. The MMA parameters are defined for a conservative approximation by $s_{init} = 0.2$, $s_{faster} = 1.1$, $s_{slower} = 0.5$, and $C_4 = 0.15$.

The MMA scheme served as an efficient update scheme in solving the stress constrained problem converging at 41 iterations and 42 seconds, achieving a final objective of 0.323 for the volume fraction. The von Mises map in Figure 4.17c depicts the von Mises stress map indicating areas of high stress in the red regions. While the sensitivity-separation was derived to handle non-self-adjoint problems, it was originally derived considering only one global constraint, thus under these circumstances the present version of the sensitivity-separation is not a valid update scheme. However this update scheme may work if red-

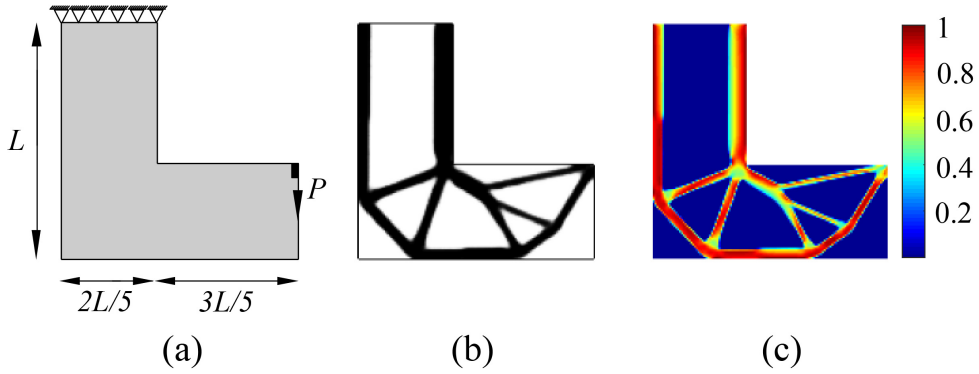


Figure 4.17: Stress constrained L-bracket featuring (a) the design domain (b) the final topology (c) the von Mises stress map

erived for the stress constrained optimization formulation. This example demonstrates how the MMA may be advantageous to other update schemes in certain problems due to its generality and robustness. It is for this reason that the MMA is typically the implemented update scheme when the optimization formulations are complex.

To explore the interplay between the number of MMA iterations, k, and augmented Lagrangian steps, several simulations are examined under different MMA iterations. We see in the previous solution that the L-bracket converges to a discrete structure using a maximum of 5 MMA iterations. It is now questioned whether the maximum number of MMA iterations can be reduced to decrease the optimization time and to what effect will this have on the final topology.

k	AL Steps	Obj	Opt time (s)
1	150	0.691	38
2	70	0.451	30
3	49	0.354	30
4	46	0.333	36

Table 4.2: Results by varying MMA iterations

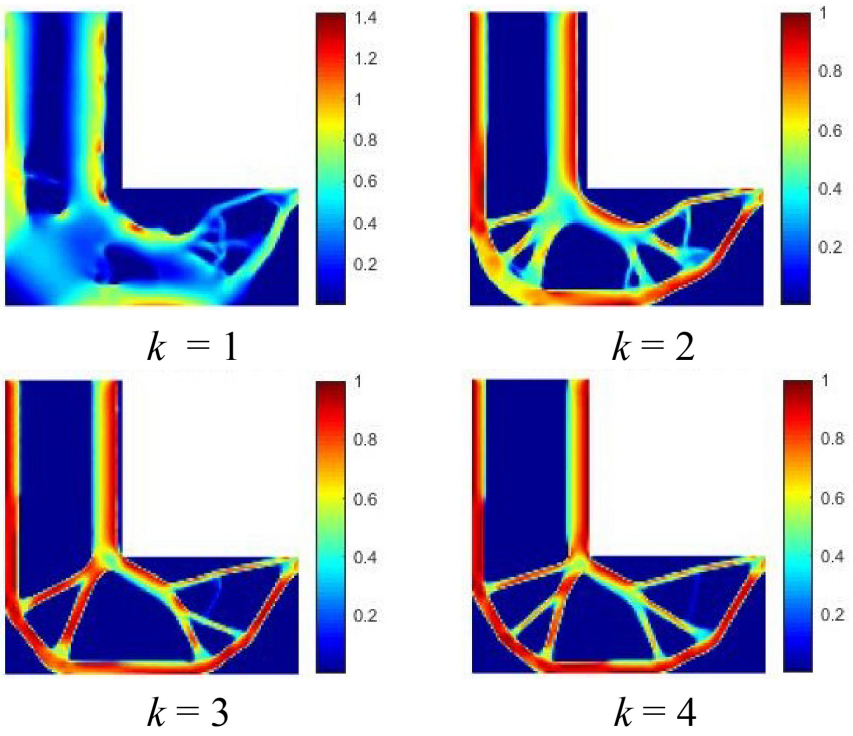


Figure 4.18: Von Mises stress map for varying MMA iterations

As seen in Figure 4.18, when $k = 4$ a structure with discrete elements, as ideal for additive manufacturing purposes, is obtained with a optimal volume fraction of $f(\rho^*) = 0.333$, higher than when $k = 5$. It is discovered that as k decreases, the topology becomes increasingly complex and the objective value increases, neither being desirable scenarios. The computational time reduces for $k = 2, 3$, however for $k = 1$, the computational time instead increases as a result of the increased augmented Lagrangian steps. From this information it is learned that a finite number of MMA iterations is necessary to minimize the augmented Lagrangian function most successfully.

4.5 Three-Dimensional Topology Optimization in PLATO

To demonstrate the translation of topology optimization academic research into industry, this example utilizes the software PLATO by Sandia National Laboratories at large [19]. This software is built off a C++ programming language which allowed for its high computational efficiency when dealing with large-scale problems. It is with this computational

efficiency that structural analysis problems for real-world applications are able to be analyzed, leading to major advancements of topology optimization implementation in the automotive, construction, and aeronautic industry.

For this example, the design domain to be evaluated is a 3D bolted bracket, as seen in Figure 4.19, which is fixed on the left end and subject to a downward y traction of $P = 1000$ N/m in the circular hole. This example will compare the topology achieved by using the MMA and OC update schemes. The bracket has dimensions of $L = 4.5$ m, $H = 3$ m, and $t = 0.5$ m. This is a widely used design component that if the weight was minimized, would lead to a more efficient structure overall. The domain is designed for a compliance minimization (stiffness maximization) objective subject to a single volume constraint, similar to the previous example. The problem is investigated considering $185,411$ linear tetrahedral elements, a filter radius of R equal to the twice the average size of the elements, a poisson's ratio of $v = 0.3$, a penalization of $p = 3$, and 150 iterations. The MMA optimizer is computed with the parameters, $s_{init} = 0.4$, $s_{faster} = 1.1$, $s_{slower} = 0.6$, and $C_4 = 0.2$.

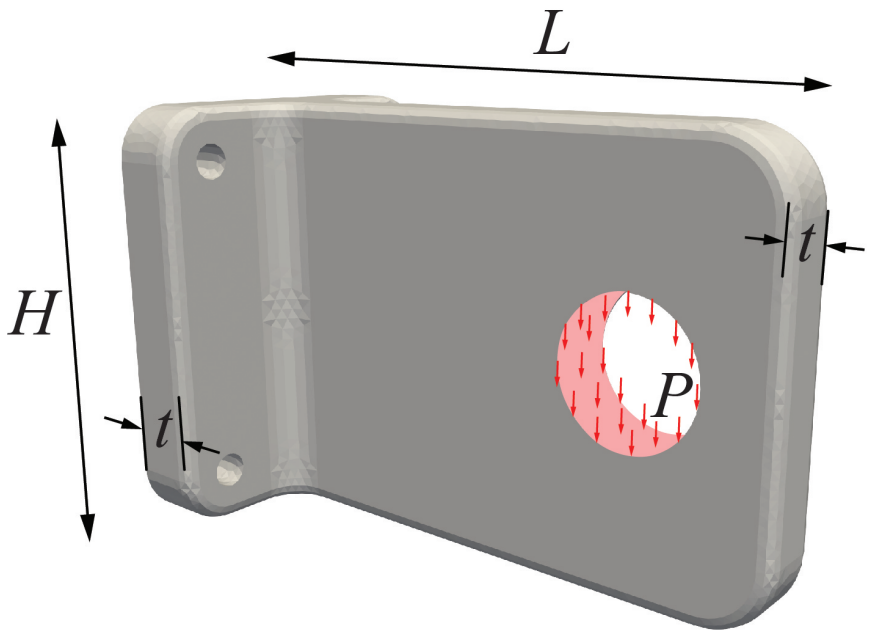


Figure 4.19: Design domain of the bolted bracket

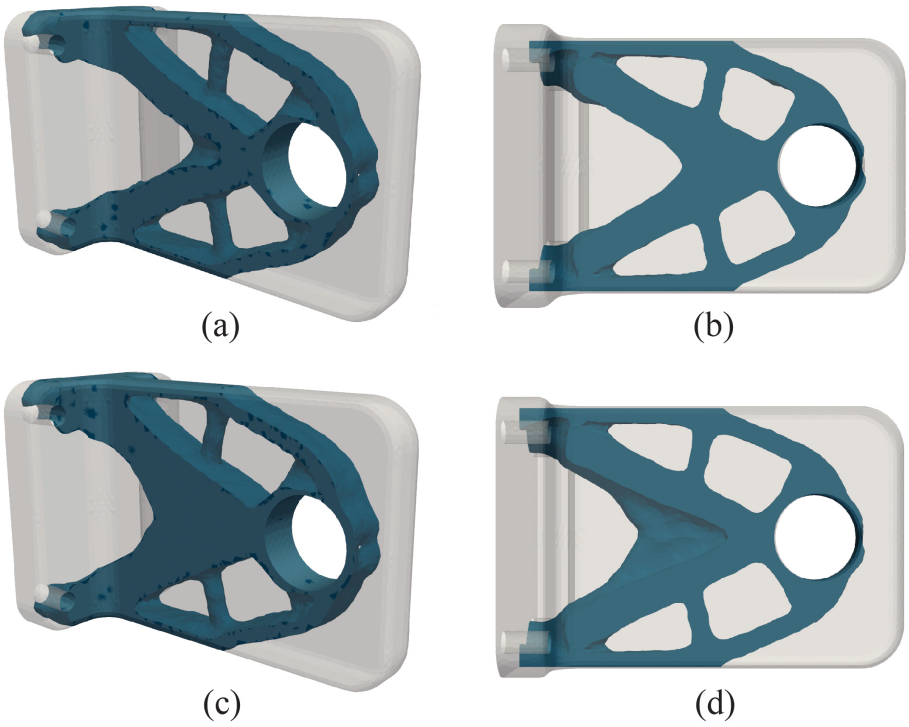


Figure 4.20: Final optimized topologies of the bolted bracket achieved by the a) MMA in an isometric view b) MMA in a side view sliced in half c) OC in an isometric view d) OC in a side view sliced in half

The optimized topologies for the bolted bracket achieved by the MMA and the OC, shown in Figure 4.20, are similar but obtain some different features, most notably near the center of the domain. Figure 4.21 demonstrates that the convergence history of the optimization obtained by the MMA and OC are nearly identical with the final objective function value at the end of the 150th iteration by the MMA being 0.8994 and for the OC being 0.9139.

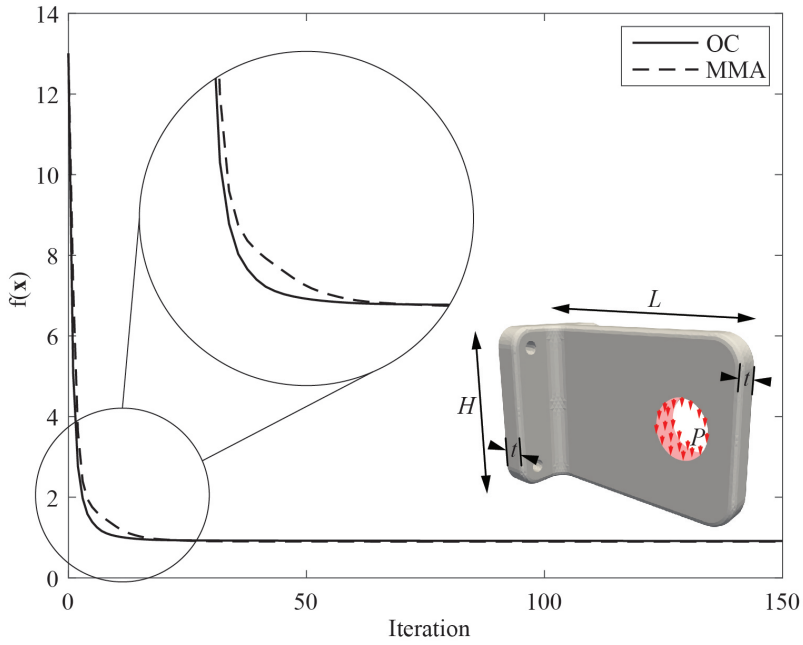


Figure 4.21: Convergence history of the bolted bracket for the OC and MMA update schemes

CHAPTER 5
CLOSING REMARKS

This thesis presented a literature review on a number of sequential explicit, convex approximation schemes with the main focus being placed on the MMA scheme [6], which since its development has been the primary sequential approximation scheme of choice in the field of structural topology optimization. The numerical examples explored the MMA from many perspectives including demonstrating the effect of the move asymptotes on the curvature of the approximation on a simple one-dimensional function; tighter bounds indicating a more conservative approximation. The results also verified how certain move asymptotes enforced in the MMA will replicate sequential approximation schemes, CONLIN and SLP, both through derivations and computational experiments. The problem of sizing optimization found that lower empirical parameters impose a tighter range of move asymptotes, leading to a more conservative approximation with slower convergence and vice versa for empirical parameters with greater magnitudes. Structural topology optimization examples were explored in the classical topology optimization framework examining the impact of the MMA empirical parameters on the final topology. These examples also compared the computational time and results of the MMA against other competing updates schemes, the OC and sensitivity-separation method. These results demonstrated that the MMA obtains greater CPU time in the update scheme subroutine in comparison to the alternative methods. The robustness of the MMA is demonstrated in the setting of solving the stress constrained topology optimization problem where in addition the interplay between the MMA iterations and augmented Lagrangian steps was explored. Lastly, the PLATO example demonstrated the application of the MMA for large-scale topology optimization problems while also highlighting the importance of the connection between academia and industry at large.

5.1 Future Work

The work of this thesis has inspired several future areas of research. Although this thesis demonstrated the behavior of the MMA as a result of the empirical parameter selection, there still remains a large amount of uncertainty regarding how to choose the most efficient set of parameters when solving a problem. Areas of future work could include using second order information to inform the approximation of the function's curvature and thus eliminating the need of empirical parameters. Another direction could be to rederive the sensitivity separation scheme to handle formulations with no inequality constraints which could expand its applications such as in the stressed constrained problem solved by the Augment Lagrangian method. The sensitivity separation technique could also be implemented into PLATO which would allow this software to solve certain non-self-adjoint formulations more efficiently, such as in the problem it was originally formulated for, the design-dependent loading optimization problem. Studying the MMA has motivated interest in deriving unique update schemes tailored to specific types of structural topology optimization frameworks to generate the most efficient solution strategy.

Appendices

APPENDIX A

MMA TRANSFORMATION PROOFS

A.1 MMA to CONLIN

By setting the move asymptotes as $L_j^k = 0$ and $U_j^k \to +\infty$ the CONLIN is obtained. This is verified by the following proof starting by rewriting the expression

$$\frac{1}{U - x_j} = \frac{1}{U(1 - x_j U^{-1})} = U^{-1}(1 + x_j U^{-1} + O(U^{-2})), \tag{A.1}$$

where U_j^k is denoted by U for simplicity. This expression resulted from expanding the term $1/(1 - x_j U^{-1})$ into a Taylor series where $O(U^{-2})$ represents the remainder of the series as $U \to +\infty$.

The MMA approximation is written as

$$g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) - \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-} x_j^k g_{i,j} + \sum_{+} \frac{(U - x_j^k)^2}{U - x_j} g_{i,j} - \sum_{-} \frac{(x_j^k)^2}{x_j} g_{i,j}, \tag{A.2}$$

following Equation 2.8 and Equation 2.9 in Section 2.4. In this approximation $g_{i,j} = \partial g_i(\boldsymbol{x}^k)/\partial x_j$ and $\sum_+$ represents summation over the terms in which $g_{i,j} > 0$, while $\sum_-$ represents summation over the terms where $g_{i,j} < 0$. By substituting the final expression of Equation A.1 into Equation A.2 we obtain

$$\begin{aligned} g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) &- \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-} x_j^k g_{i,j} \\ &+ \sum_{+} U^{-1}\left(1 + x_j U^{-1} + O(U^{-2})\right)\left(U^2 - 2U x_j^k + (x_j^k)^2\right) g_{i,j} - \sum_{-} \frac{(x_j^k)^2}{x_j} g_{i,j}, \end{aligned} \tag{A.3}$$

where further simplification leads to

$$g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) - \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-} x_j^k g_{i,j} + \sum_{+}\left(1 + x_j U^{-1} + O(U^{-2})\right)\left(U - 2x_j^k + U^{-1}(x_j^k)^2\right) g_{i,j} - \sum_{-}\frac{(x_j^k)^2}{x_j} g_{i,j}. \tag{A.4}$$

After distributing the terms and simplifying we arrive at

$$g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \sum_{+} x_j^k g_{i,j} + \sum_{-} x_j^k g_{i,j} + \sum_{+}\left(x_j + O(U^{-1}) - 2x_j^k\right) g_{i,j} - \sum_{-}\frac{(x_j^k)^2}{x_j} g_{i,j}. \tag{A.5}$$

By letting $U \to +\infty$ the MMA approximation reaches the following form

$$g_i^{M,k}(\boldsymbol{x}) \to g_i(\boldsymbol{x}^k) + \sum_{+}(x_j - x_j^k)g_{i,j} + \sum_{-}\left(x_j^k - \frac{(x_j^k)^2}{x_j}\right) g_{i,j}, \tag{A.6}$$

which is equivalent to that of the CONLIN approximation.

A.2 MMA to SLP

The SLP approximation will be obtained by setting the move asymptotes in the MMA to $L_j^k \to -\infty$ and $U_j^k \to +\infty$. Similarly to Equation A.1, the term $1/(x_j - L_j^k)$ can be rewritten in the form of a Taylor expansion

$$\frac{1}{x_j - L} = \frac{1}{L(L^{-1}x_j - 1)} = -L^{-1}\frac{1}{(1 - L^{-1}x_j)} = -L^{-1}(1 + x_j L^{-1} + O(L^{-2})), \quad \text{(A.7)}$$

where $L = L_j^k$ and $O(L^{-2})$ denotes the remainder of the expansion as $L \to -\infty$. We now write the MMA approximation by the following

$$\begin{aligned} g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) &- \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-}(x_j^k - L)g_{i,j} \\ &+ \sum_{+}\frac{(U - x_j^k)^2}{U - x_j}g_{i,j} - \sum_{-}\frac{(x_j^k - L)^2}{x_j - L}g_{i,j}. \end{aligned} \quad \text{(A.8)}$$

By substituting the expressions from Equation A.1 and Equation A.7 into Equation A.8 we arrive at

$$\begin{aligned} g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) &- \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-}(x_j^k - L)g_{i,j} \\ &+ \sum_{+} U^{-1}\left(1 + x_j U^{-1} + O(U^{-2})\right)\left(U^2 - 2Ux_j^k + (x_j^k)^2\right)g_{i,j} \\ &+ \sum_{-} L^{-1}\left(1 + x_j L^{-1} + O(L^{-2})\right)\left((x_j^k)^2 - 2x_j^k L + L^2\right)g_{i,j}. \end{aligned} \quad \text{(A.9)}$$

After distributing the U^{-1} and L^{-1} terms, the expression reduces to

$$\begin{aligned} g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) &- \sum_{+}(U - x_j^k)g_{i,j} + \sum_{-}(x_j^k - L)g_{i,j} \\ &+ \sum_{+}\left(1 + x_j U^{-1} + O(U^{-2})\right)\left(U - 2x_j^k + U^{-1}(x_j^k)^2\right)g_{i,j} \\ &+ \sum_{-}\left(1 + x_j L^{-1} + O(L^{-2})\right)\left(L^{-1}(x_j^k)^2 - 2x_j^k + L\right)g_{i,j}. \end{aligned} \quad \text{(A.10)}$$

Further simplification of the approximation leads to the following equation:

$$g_i^{M,k}(\boldsymbol{x}) = g_i(\boldsymbol{x}^k) + \sum_{+} x_j^k g_{i,j} + \sum_{-} x_j^k g_{i,j} + \sum_{+} \left(x_j + O(U^{-1}) - 2x_j^k\right) g_{i,j} + \sum_{-} \left(x_j - 2x_j^k + O(L^{-1})\right) g_{i,j}. \tag{A.11}$$

Letting $U \to +\infty$ and $L \to -\infty$ the MMA transforms into the form

$$g_i^{M,k}(\boldsymbol{x}) \to g_i(\boldsymbol{x}^k) + \sum_{+} (x_j - x_j^k) g_{i,j} + \sum_{-} (x_j - x_j^k) g_{i,j}, \tag{A.12}$$

which is identical to the SLP approximation.

APPENDIX B

MMA COMPUTATIONAL IMPLEMENTATION MODIFICATIONS

The MMA subroutine was implemented into a Matlab code (available at http://www.smoptit.se/) with several adjustments from the original formulation to provide slight improvements to the framework [43]. This appendix will address these modifications and the reasoning behind their application.

The first changes in the subproblem were made to the p_{ij}^k and q_{ij}^k expressions. Their original formulation by Equation 2.9 is defined such that if the gradient $\partial g_i/\partial x_j > 0$, p_{ij}^k will be some finite value and q_{ij}^k will be 0 and vice versa if $\partial g_i/\partial x_j < 0$. Now an additional term is added to the p_{ij}^k and q_{ij}^k expressions to improve the convergence.

The p_{ij}^k and q_{ij}^k formulas are reformulated in the following manner:

$$p_{ij}^k = (U_j^k - x_j^k)^2 \left(C_1 \left(\frac{\partial g_i}{\partial x_j}(x^k) \right)^+ + C_2 \left(\frac{\partial g_i}{\partial x_j}(x^k) \right)^- + \frac{C_3}{x_j^{max} - x_j^{min}} \right) \tag{B.1}$$

$$q_{ij}^k = (x_j^k - L_j^k)^2 \left(C_2 \left(\frac{\partial g_i}{\partial x_j}(x^k) \right)^+ + C_1 \left(\frac{\partial g_i}{\partial x_j}(x^k) \right)^- + \frac{C_3}{x_j^{max} - x_j^{min}} \right), \tag{B.2}$$

where $\left(\partial g_i/\partial x_j(x^k)\right)^+$ denotes the largest of $\left(\partial g_i/\partial x_j(x^k)\right)$ and 0 while $\left(\partial g_i/\partial x_j(x^k)\right)^-$ denotes the largest of $-\left(\partial g_i/\partial x_j(x^k)\right)$ and 0. The constants are defined as $C_1 = 1.001$, $C_2 = 0.001$, and $C_3 = 10^{-5}$.

Another slight modification was made to the move limits, α_j^k and β_j^k, to have better control over the speed of convergence.

$$\alpha_j^k = max(x_j^{min}, L_j^k + \mu(x_j^k - L_j^k), x_j^k - C_4(x_j^{max} - x_j^{min})) \tag{B.3}$$

$$\beta_j^k = min(x_j^{max}, U_j^k - \mu(U_j^k - x_j^k), x_j^k + C_4(x_j^{max} - x_j^{min}) \tag{B.4}$$

Here the α_j^k and β_j^k have an additional term which may be used to determine their value. If the introduced term does control the move limits, the constant C_4 or the 'move parameter' serves to increase or decrease the domain in which the new design variable can be chosen, where $0 < C_4 < 1$. For example, a C_4 of 0.5 is relatively large and will increase the range of the move limits, if instead $C_4 = 0.2$ the range in between the move limits will decrease. In problems where the design variables seem to be converging in the same direction, setting a larger C_4 will speed up the convergence. For nonlinear problems where the design variables are oscillating, the C_4 may need to be reduced to achieve convergence. The introduced move parameter now allows for more user freedom for selecting proper move limits.

REFERENCES

[1] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197–224, 1988.

[2] G. I. Rozvany, "A critical review of established methods of structural topology optimization," *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, pp. 217–237, 2009.

[3] O. Sigmund and K. Maute, "Topology optimization approaches," *Structural and Multidisciplinary Optimization*, vol. 48, no. 6, pp. 1031–1055, 2013.

[4] J. D. Deaton and R. V. Grandhi, "A survey of structural and multidisciplinary continuum topology optimization: Post 2000," *Structural and Multidisciplinary Optimization*, vol. 49, no. 1, pp. 1–38, 2014.

[5] C. Fleury, "First and second order convex approximation strategies in structural optimization," *Structural Optimization*, vol. 1, no. 1, pp. 3–10, 1989.

[6] K. Svanberg, "The method of moving asymptotes—a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, 1987.

[7] L. Schmit Jr and B. Farshi, "Some approximation concepts for structural synthesis," *AIAA Journal*, vol. 12, no. 5, pp. 692–699, 1974.

[8] L. A. Schmit and C. Fleury, "Structural synthesis by combining approximation concepts and dual methods," *AIAA Journal*, vol. 18, no. 10, pp. 1252–1260, 1980.

[9] G. G. Pope, "Optimum design of stressed skin structures," *AIAA Journal*, vol. 11, no. 11, pp. 1545–1552, 1973.

[10] K. Svanberg, "Some second order methods for structural optimization," in *Optimization of Large Structural Systems*, Springer, 1993, pp. 567–578.

[11] C. Fleury and V. Braibant, "Structural optimization: A new dual method using mixed variables," *International Journal for Numerical Methods in Engineering*, vol. 23, no. 3, pp. 409–428, 1986.

[12] K. Svanberg, "A globally convergent version of MMA without line search," in *Proceedings of the first World Congress of Structural and Multidisciplinary Optimization*, Goslar, Germany, vol. 28, 1995, pp. 9–16.

[13] F. V. Senhora, O. Giraldo-Londono, I. F. Menezes, and G. H. Paulino, "Topology optimization with local stress constraints: A stress aggregation-free approach," *Structural and Multidisciplinary Optimization*, vol. 62, no. 4, pp. 1639–1668, 2020.

[14] E. A. Träff, O. Sigmund, and N. Aage, "Topology optimization of ultra high resolution shell structures," *Thin-Walled Structures*, vol. 160, p. 107 349, 2021.

[15] O. Giraldo-Londoño and G. H. Paulino, "Polystress: A Matlab implementation for local stress-constrained topology optimization using the augmented lagrangian method," *Structural and Multidisciplinary Optimization*, vol. 63, no. 4, pp. 2065–2097, 2021.

[16] F. Ferrari, O. Sigmund, and J. K. Guest, "Topology optimization with linearized buckling criteria in 250 lines of matlab," *Structural and Multidisciplinary Optimization*, vol. 63, no. 6, pp. 3045–3066, 2021.

[17] C. S. Andreasen, M. O. Elingaard, and N. Aage, "Level set topology and shape optimization by density methods using cut elements with length scale control," *Structural and Multidisciplinary Optimization*, pp. 1–23, 2020.

[18] N. Aage, E. Andreassen, and B. S. Lazarov, "Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework," *Structural and Multidisciplinary Optimization*, vol. 51, no. 3, pp. 565–572, 2015.

[19] P. D. Team, "Plato 2.5 user's manual," Sandia National Labs, Tech. Rep., 2021.

[20] B. Torstenfelt, "Trinitas," *An Integrated Graphical System for Finite Element Analysis, User's Manual, Version*, vol. 2, 2012.

[21] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[22] P. W. Christensen and A. Klarbring, *An Introduction to Structural Optimization*. Springer Science & Business Media, 2008, vol. 153.

[23] Y. Cao, S. Li, L. Petzold, and R. Serban, "Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution," *SIAM Journal on Scientific Computing*, vol. 24, no. 3, pp. 1076–1089, 2003.

[24] R. D. Neidinger, "Introduction to automatic differentiation and Matlab object-oriented programming," *SIAM Review*, vol. 52, no. 3, pp. 545–563, 2010.

[25] J. S. Arora, *Introduction to Optimum Design*, 4th ed. Elsevier, 2016.

[26] K. John, C. Ramakrishnan, and K. Sharma, "Minimum weight design of trusses using improved move limit method of sequential linear programming," *Computers & Structures*, vol. 27, no. 5, pp. 583–591, 1987.

[27] T.-Y. Chen, "Calculation of the move limits for the sequential linear programming method," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 15, pp. 2661–2679, 1993.

[28] B. Wang, Y. Chang, K. Lawrence, and T. Y. Chen, "Optimum design of multiple configuration structures for frequency constraints," *AIAA Journal*, vol. 29, no. 10, pp. 1761–1762, 1991.

[29] L. Lamberti and C. Pappalettere, "Comparison of the numerical efficiency of different sequential linear programming based algorithms for structural optimisation problems," *Computers & Structures*, vol. 76, no. 6, pp. 713–728, 2000.

[30] R. Yang and C. Chuang, "Optimal topology design using linear programming," *Computers & Structures*, vol. 52, no. 2, pp. 265–275, 1994.

[31] A. Takezawa, G. H. Yoon, S. H. Jeong, M. Kobashi, and M. Kitamura, "Structural topology optimization with strength and heat conduction constraints," *Computer Methods in Applied Mechanics and Engineering*, vol. 276, pp. 341–361, 2014.

[32] Y. Maeda, S. Nishiwaki, K. Izui, M. Yoshimura, K. Matsui, and K. Terada, "Structural topology optimization of vibrating structures with specified eigenfrequencies and eigenmode shapes," *International Journal for Numerical Methods in Engineering*, vol. 67, no. 5, pp. 597–628, 2006.

[33] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–51, 1995.

[34] A. J. Shepherd, *Second-order Methods for Neural Networks: Fast and Reliable Training Methods for Multi-layer Perceptrons*. Springer Science & Business Media, 2012.

[35] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming for large-scale nonlinear optimization," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 123–137, 2000.

[36] F. E. Curtis and M. L. Overton, "A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 474–500, 2012.

[37] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer, 1984, vol. 2.

[38] P. E. Gill and E. Wong, "Sequential quadratic programming methods," in *Mixed Integer Nonlinear Programming*, Springer, 2012, pp. 147–224.

[39] K. Schittkowski, "Nlpql: A Fortran subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, no. 2, pp. 485–500, 1986.

[40] C. Fleury, "CONLIN: An efficient dual optimizer based on convex approximation concepts," *Structural Optimization*, vol. 1, no. 2, pp. 81–89, 1989.

[41] J. H. Starnes Jr and R. T. Haftka, "Preliminary design of composite wings for buckling, strength, and displacement constraints," *Journal of Aircraft*, vol. 16, no. 8, pp. 564–570, 1979.

[42] P. Rigo and C. Fleury, "Scantling optimization based on convex linearizations and a dual approach—part ii," *Marine Structures*, vol. 14, no. 6, pp. 631–649, 2001.

[43] K. Svanberg, "MMA and GCMMA - two methods for nonlinear optimization," vol. 1, pp. 1–15, 2007.

[44] P. Duysinx, "Topology optimization of structures and continua computational aspects and background," University of Liege, Tech. Rep., 2014.

[45] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 555–573, 2002.

[46] ——, "Non-mixed second order derivatives in MMA," *Dept. of Mathematics, Royal Institute of Technology, Stockholm*, 1995.

[47] M. Bruyneel, P. Duysinx, and C. Fleury, "A family of mma approximations for structural optimization," *Structural and Multidisciplinary Optimization*, vol. 24, no. 4, pp. 263–276, 2002.

[48] D. P. Bertsekas, *Convex optimization theory*. Athena Scientific Belmont, 2009.

[49] T. Kumar and K. Suresh, "Direct lagrange multiplier updates in topology optimization revisited," *Structural and Multidisciplinary Optimization*, vol. 63, no. 3, pp. 1563–1578, 2021.

[50] C. T. Kelley and E. W. Sachs, "A trust region method for parabolic boundary control problems," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1064–1081, 1999.

[51] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, http://cvxr.com/cvx, Mar. 2014.

[52] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., http://stanford.edu/~boyd/graph_dcp.html, Springer-Verlag Limited, 2008, pp. 95–110.

[53] C. Talischi, G. H. Paulino, A. Pereira, and I. F. Menezes, "Polytop: A matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes," *Structural and Multidisciplinary Optimization*, vol. 45, no. 3, pp. 329–357, 2012.

[54] G. I. Rozvany, "Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics," *Structural and Multidisciplinary optimization*, vol. 21, no. 2, pp. 90–108, 2001.

[55] M. P. Bendsøe and O. Sigmund, *Optimization of Structural Topology, Shape, and Material*. Springer, 1995, vol. 414.

[56] A. A. Groenwold and L. Etman, "On the equivalence of optimality criterion and sequential approximate optimization methods in the classical topology layout problem," *International Journal for Numerical Methods in Engineering*, vol. 73, no. 3, pp. 297–316, 2008.

[57] Y. Jiang, A. S. Ramos Jr., and G. H. Paulino, "Topology optimization with design-dependent loading: An adaptive sensitivity-separation design variable update scheme," *Structural and Multidisciplinary Optimization*, Accepted, 2020.

[58] C. Talischi, G. H. Paulino, A. Pereira, and I. F. Menezes, "Polymesher: A general-purpose mesh generator for polygonal elements written in matlab," *Structural and Multidisciplinary Optimization*, vol. 45, no. 3, pp. 309–328, 2012.