

Problem-solving by nonlinear mathematical programming

Dirk Munro (Hamburg, Germany)

March 13, 2022

*This note is intended as an elementary introduction to the subject of sequential approximate optimization, a general nonlinear mathematical programming paradigm. In particular, the subject is described from the solution-method's point of view, and an 'optimization problem' is taken to be a well-defined representation of any problem in a practical setting. No novelty beyond the interpretation, description and notation is introduced. The list of citations would not be reasonably complete without those referenced in the **Background**, in closure of the note.*

1 Introduction

Assume we have a problem \mathcal{P} and an array of scalar decision variables \mathbf{x} . Each decision variable is assumed to be continuous*, and packed into an array of length n ; *i.e.* $\mathbf{x} \in \mathbb{R}^n$. A particular point in the 'decision space' will henceforth be represented by a particular collection of values in \mathbf{x} . What is meant herein by 'a problem \mathcal{P} '? We will assume that \mathcal{P} is programmatic (numeric) entity, an analytic representation of which is not available, but it may be evaluated with an array of particular decision (input) variable values \mathbf{x} , $\mathcal{P}[\mathbf{x}]$ [†], and, in so doing, the problem returns the following information

$$\left\{ \begin{array}{l} \mathbf{c} = (c_0, c_1, \dots, c_m) \\ \partial \mathbf{c} = (\partial c_0, \partial c_1, \dots, \partial c_m) \\ \underline{\mathbf{x}} = (\underline{x}, \bar{x}) \end{array} \right\} = \mathcal{P}[\mathbf{x}]. \quad (1)$$

That is

- (i) an array \mathbf{c} of (scalar function) values, of length m . We will adopt the convention that the first entry c_0 is an objective (cost) to be *minimised*, and the following $m - 1$ entries denote constraint function values, with values of $c_j \leq 0$ indicating *feasibility*; *i.e.* adherence to constraint j ;
- (ii) assuming each cost-and-constraint function is (at least once) continuously differentiable, the first-order derivatives of each, to each decision variable, $\partial \mathbf{c}$, with

$$\partial c_j = \left(\frac{\partial c_j}{\partial x_1}, \frac{\partial c_j}{\partial x_2}, \dots, \frac{\partial c_j}{\partial x_n} \right) \quad \text{for } j = 0, 1, \dots, m; \quad (2)$$

- (iii) and the lower $\underline{\mathbf{x}}$ and upper $\bar{\mathbf{x}}$ bounds of the decision space, respectively, defining the entire decision space $\underline{\mathbf{x}}$.

*Discrete variables may always be relaxed to a continuous form, and driven to discrete solutions.

[†]Through-out square brackets $[\cdot]$ will denote an operation of 'to evaluate at'; and it may be used and dropped freely as is relevant to the description of a particular entity, in a particular context.

In other words, beyond the information we can collect from the quantities returned for particular evaluations $\{\mathbf{c}, \partial\mathbf{c}, \underline{\mathbf{x}}\} = \mathcal{P}[\mathbf{x}]$ we have no further information of the problem \mathcal{P} , except for what can be deduced from the upstream assumptions (and knowledge of the nature of the problem). Moreover, in general, a single evaluation of problem $\mathcal{P}[\mathbf{x}]$ is considered expensive—or inconvenient—insofar as it is non-trivially costly to evaluate. That is for example computational expense in simulation-based, multiple, potentially large-scale finite-element analyses, or computationally cheap (*e.g.* reduced-order) analyses which rely on calls to licensed applications, or problem evaluations which require significant network communications.

How to find a candidate[‡] optimum solution \mathbf{x}^* of problem \mathcal{P} ? Well, we can start by sampling the decision space $\underline{\mathbf{x}}$, to find an \mathbf{x} with a reasonable cost $c_0[\mathbf{x}]$, while all constraints are feasible $c_j[\mathbf{x}] \leq 0, \forall j > 0$. Then, we can ask the question: can this sample decision \mathbf{x} be improved if we adjust the evaluation in the decision space by a particular amount $\mathbf{x} + \Delta\mathbf{x}$? If there is a $\Delta\mathbf{x}$ for which $c_0[\mathbf{x}] > c_0[\mathbf{x} + \Delta\mathbf{x}]$, while all constraints remain feasible $c_j[\mathbf{x} + \Delta\mathbf{x}] \leq 0, \forall j > 0$, then the sample \mathbf{x} we started with was indeed not optimal[§], and we should adjust the decision to $\mathbf{x} + \Delta\mathbf{x}$.

How to determine what $\Delta\mathbf{x}$ should be? Keep in mind, we have assumed we are working with an expensive-to-evaluate problem \mathcal{P} , which only returns the information $\{\mathbf{c}, \partial\mathbf{c}, \underline{\mathbf{x}}\} = \mathcal{P}[\mathbf{x}]$ upon evaluation at \mathbf{x} . In practical terms, we will take this to mean that we want to compute a good change in the decision space $\Delta\mathbf{x}$, without re-evaluating problem \mathcal{P} .

Let us, in hope[¶], assume that the actual cost-and-constraint functions \mathbf{c} are (near) linear over the entire decision space $\underline{\mathbf{x}}$. That is, an analytic linear approximation of the cost-and-constraint functions \mathbf{c} , in terms of $\Delta\mathbf{x}$, is constructed

$$\mathbf{v}[\Delta\mathbf{x}] = \mathbf{c} + \partial\mathbf{c} \cdot \Delta\mathbf{x}, \quad (3)$$

with first derivatives following accordingly

$$\partial\mathbf{v}[\Delta\mathbf{x}] = \partial\mathbf{c}. \quad (4)$$

In general, the first-order approximation of the cost-and-constraint functions \mathbf{v} is only equal^{||} to the actual cost-and-constraint functions \mathbf{c} at \mathbf{x} —i.e., with $\Delta\mathbf{x} = \mathbf{0}$ —and representative in an infinitely small region around \mathbf{x} . In the light of this, we introduce a lower $\Delta\underline{\mathbf{x}}$ and upper $\Delta\bar{\mathbf{x}}$ bound on the change $\Delta\mathbf{x}$ we allow ourselves in the decision space $\underline{\mathbf{x}}$. That is, we have arrived at a subproblem

$$\left\{ \begin{array}{l} \mathbf{v} = (v_0, v_1, \dots, v_m) \\ \partial\mathbf{v} = (\partial v_0, \partial v_1, \dots, \partial v_m) \\ \Delta\underline{\mathbf{x}} = (\Delta\underline{\mathbf{x}}, \Delta\bar{\mathbf{x}}) \end{array} \right\} = \mathcal{S}[\Delta\mathbf{x}], \quad (5)$$

[‡]In general the problem is assumed to be multi-modal. That is, it may have many local optima, in addition to one global optimum. The topic is discussed, and a solution method is proposed, in Reference [1].

[§]Not even locally.

[¶]or desperation.

^{||}Equal in zero- and first-order information.

which is, importantly, unlike problem \mathcal{P} , very cheap to evaluate for different values of $\Delta \mathbf{x}$ due to the construction of the an analytic approximation \mathbf{v} . In general we say that a solution to problem \mathcal{S} —finding that $\Delta \mathbf{x}$ in $\Delta \bar{\mathbf{x}}$ for which v_0 is a minimum, while all $v_j \leq 0, \forall j > 0$ —is computable in polynomial time. Moreover, in practice, polynomial time solution methods are readily available.

We may thus, upon solving subproblem \mathcal{S} , update the decision to $\mathbf{x} \leftarrow \mathbf{x} + \Delta \mathbf{x}$; repeat the evaluation of problem \mathcal{P} at the new values of the decision variables $\{\mathbf{c}, \partial \mathbf{c}, \bar{\mathbf{x}}\} = \mathcal{P}[\mathbf{x}]$, and repeat the construction and solution of subproblem \mathcal{S} . The process may be repeated until there is no change in the decision $\Delta \mathbf{x}$ which improves the solution, and we may reasonable deduce that we have arrived at a candidate solution $\mathbf{x} \rightarrow \mathbf{x}^*$ of problem \mathcal{P} .

In general, however, linear approximations \mathbf{v} of the cost-and-constraint functions \mathbf{c} may result in changes $\Delta \mathbf{x}$ in the decision space which violates one or more constraints $c_j[\mathbf{x} + \Delta \mathbf{x}] > 0$, increases the cost function $c_0[\mathbf{x} + \Delta \mathbf{x}] > c_0[\mathbf{x}]$, permission of only restrictive allowable decision changes $\Delta \bar{\mathbf{x}}$, resulting in excessive, expensive evaluations of problem $\mathcal{P}[\mathbf{x}]$, and hence failure of the procedure to converge to a reasonable candidate solution \mathbf{x}^* of problem \mathcal{P} .

What can we do?

2 Decision space transformations

Let us assume that we notice, by observation, that the cost-and-constraint functions \mathbf{c} have a particular proportional relationship to decisions \mathbf{x} in the decision space $\bar{\mathbf{x}}$. For example, we might notice** that the cost-and-constraint functions \mathbf{c} have a reciprocal-like relation to our decision variables, *i.e.* $\mathbf{c} \sim 1/\mathbf{x}$. Can we exploit this to improve the change $\Delta \mathbf{x}$ in the decision space we attempt to make? Yes, we can imagine applying an analytic transformation (mapping) to the decision space $\mathbf{y} = \mathbf{y}[\mathbf{x}]$, and formulating the approximate cost-and-constraint functions \mathbf{v} in terms of it

$$\mathbf{v} = \mathbf{c} + \partial_{\mathbf{y}} \mathbf{c} \cdot \Delta \mathbf{y} . \quad (6)$$

That is, we hope that the transformation of the decisions space from $\mathbf{x} \rightarrow \mathbf{y}$ has worked to ‘linearise’ the cost-and-constraint functions \mathbf{c} . Rewriting Eq. (6) in terms of the first-order information supplied by problem \mathcal{P} , we retrieve

$$\mathbf{v} = \mathbf{c} + \partial \mathbf{c} \cdot \partial_{\mathbf{x}}^{-1} \mathbf{y} \cdot \Delta \mathbf{y} , \quad (7)$$

with the (inverted) derivative of the analytic mapping $\mathbf{y} = \mathbf{y}[\mathbf{x}]$ easy to compute

$$\partial_{\mathbf{x}}^{-1} \mathbf{y} = 1 / \frac{\partial \mathbf{y}}{\partial \mathbf{x}} . \quad (8)$$

Note that $\partial_{\mathbf{x}}^{-1} \mathbf{y}$ is an $n \times n$ square matrix (although in practice, typically, only diagonal terms are utilised).

**and/or know, based on knowledge on the nature of the problem \mathcal{P} .

How does this help us? Consider again the example of the reciprocal transformation $\mathbf{y} = 1/\mathbf{x}$. In this case

$$\partial_{\mathbf{x}}^{-1}\mathbf{y} = -\mathbf{x}_{\mathbf{I}}^2, \quad (9)$$

with \mathbf{I} indicating that the array is cast along the diagonal of the correspondingly sized square identity matrix. Hence

$$\mathbf{v} = \mathbf{c} + \partial\mathbf{c} \cdot (-\mathbf{x}_{\mathbf{I}}^2) \cdot \Delta\mathbf{y}, \quad (10)$$

and the approximate cost-and-constraint functions \mathbf{v} may be re-written in terms of the original decision space \mathbf{x} and the change $\Delta\mathbf{x}$, which yields

$$\mathbf{v} = \mathbf{c} + \partial\mathbf{c} \cdot (-\mathbf{x}_{\mathbf{I}}^2) \cdot \left(\frac{1}{\mathbf{x} + \Delta\mathbf{x}} - \frac{1}{\mathbf{x}} \right), \quad (11)$$

and, is the same as

$$\mathbf{v} = \mathbf{c} + \partial\mathbf{c} \cdot \left(\frac{\mathbf{x}}{\mathbf{x} + \Delta\mathbf{x}} \right)_{\mathbf{I}} \cdot \Delta\mathbf{x}. \quad (12)$$

Notice how the term in brackets (\dots) has introduced an analytically available nonlinearity (or ‘curvature’) in the approximate cost-and-constraint functions \mathbf{v} , with respect to $\Delta\mathbf{x}$. Keep in mind, the evaluation of problem $\mathcal{P}[\mathbf{x}]$ is constant while we evaluate (and solve) subproblem \mathcal{S} —which remains computationally cheap and convenient—to compute a change in the decision space $\Delta\mathbf{x}$.

Furthermore, a decision space transformation may be generalized to the form $\mathbf{y} = \mathbf{x}_{\mathbf{I}}^p$, with p any real number. In this case

$$\partial_{\mathbf{x}}^{-1}\mathbf{y} = \frac{1}{p}\mathbf{x}_{\mathbf{I}}^{1-p}, \quad (13)$$

and hence

$$\mathbf{v}[\Delta\mathbf{x}] = \mathbf{c} + \partial\mathbf{c} \cdot \frac{1}{p}\mathbf{x}_{\mathbf{I}}^{1-p} \cdot ((\mathbf{x} + \Delta\mathbf{x})^p - \mathbf{x}^p), \quad (14)$$

which is easily evaluated for different values of $\Delta\mathbf{x}$, with any manner of estimate for p . The first derivatives of the approximate cost-and-constraint functions \mathbf{v} follow readily

$$\partial\mathbf{v}[\Delta\mathbf{x}] = \partial\mathbf{c} \cdot (\mathbf{x}_{\mathbf{I}}^{1-p}) \cdot (\mathbf{x} + \Delta\mathbf{x})^{p-1}, \quad (15)$$

with the first-order information of problem \mathcal{P} at the current evaluation point \mathbf{x} , $\partial\mathbf{c}$, maintained

$$\partial\mathbf{v}[\mathbf{0}] = \partial\mathbf{c}. \quad (16)$$

Needs mentioning here:

$$\partial^2\mathbf{v}[\Delta\mathbf{x}] = \partial\mathbf{c} \cdot (p-1)\mathbf{x}_{\mathbf{I}}^{1-p} \cdot (\mathbf{x} + \Delta\mathbf{x})^{p-2}, \quad (17)$$

If the cost-and-constraint functions \mathbf{c} do indeed have a reciprocal relation to the decision variables \mathbf{x} , or a form captured and estimated by the generalisation above, then we can expect a more accurate approximate representation of the cost-and-constraint functions \mathbf{v} , improved estimates of the change required in the decision space $\Delta\mathbf{x}$, and, ultimately, faster and more efficient solution (fewer evaluations) of the problem \mathcal{P} .

3 Convex problems

Will have to mention duality... because CONLIN cooks it up so that analytic form of primal in terms of dual is available...

These is however a caveat. Beyond the cheap-to-evaluate analytic representation of the cost-and-constraint functions \mathbf{v} (which makes sense in its own right), we require the subproblem \mathcal{S} to be *convex* in order to solve it efficiently (in polynomial time) with commonly available computational methods. In general, the subproblem \mathcal{S} is convex if it has a convex objective (cost) function^{††} c_0 , and if the feasible region described by the constraints $c_j \leq 0$, $\forall j$, is a convex set^{‡‡}. As we will see, this is often (made) easy to guarantee via particular constructions of subproblem \mathcal{S} .

A classical and well-known construction of this form is due to Fleury *et al.*, as developed for Aerospace engineering applications in the 1980's [2]. Fleury and co-workers devised the method of *convex linearisation* (or 'CONLIN', for short) for problems of structural weight minimisation subject to displacement and stress constraints [3].

In this setting, convexity of subproblem \mathcal{S} is guaranteed by enforcing positive semi-definiteness of each cost-and-constraint function \mathbf{v} , by judicious selection^{§§} of the decision space mapping $\mathbf{y} = \mathbf{y}[\mathbf{x}]$. In this context, an individual function v_j is convex if its second derivative $\partial^2 v_j$ is greater than equal to zero over its domain of defintion $\Delta \mathbf{x} \in \Delta \bar{\mathbf{x}}$.

A convenient way to illustrate is by revisiting the generalisation $\mathbf{y} = \mathbf{x}_1^{\mathbf{p}}$. (Note bold p and mention different functions.) Consider again Equation (17). In CONLIN this is achieved by chosing \mathbf{p} according to the signs of the first-order information $\partial \mathbf{c}$, as queried from problem \mathcal{P} . If $\partial_i c_j > 0$, then $p_{ij} = 1$, otherwise $p_{ij} = -1$.

Consider the second-order approximate information, due to a mapping of a single decision variable, with respect to a single function ...

$$\partial_i^2 v_j[\Delta x_i] = \partial_i c_j \cdot (p - 1) x_i^{1-p} \cdot (x_i + \Delta x_i)^{p-2} \quad (18)$$

That is ...

One of the first generalized examples of such an approach / make use of this property... aimed specifically at weight minimization of structural configurations ... CONLIN [2,3] ... mention convexity conditions ... reciprocal and linear ...

4 Taylor series expansions and quadratically constrained quadratic programs

The analytic nonlinearity introduced by the decision space transformation is conveniently unpacked by Taylor series expansion ν , of the transformed cost-and-constraints \mathbf{v} , in

^{††}Formal definition.

^{‡‡}Formal definition

^{§§}Mention conservatism and seperability.

terms of $\Delta \mathbf{x}$ (away from the current decision point \mathbf{x} , where $\Delta \mathbf{x} = 0$)

$$\boldsymbol{\nu}[\Delta \mathbf{x}] = \mathbf{v}[\mathbf{0}] + \frac{1}{1!} \partial \mathbf{v}[\mathbf{0}] \cdot \Delta \mathbf{x} + \frac{1}{2!} \partial^2 \mathbf{v}[\mathbf{0}] \cdot \Delta \mathbf{x}^2 + \frac{1}{3!} \partial^3 \mathbf{v}[\mathbf{0}] \cdot \Delta \mathbf{x}^3 + \dots \quad (19)$$

For sake of simplicity, let us truncate after the first nonlinear (2nd) term

$$\boldsymbol{\nu}[\Delta \mathbf{x}] = \mathbf{v}[\mathbf{0}] + \partial \mathbf{v}[\mathbf{0}] \cdot \Delta \mathbf{x} + \frac{1}{2} \partial^2 \mathbf{v}[\mathbf{0}] \cdot \Delta \mathbf{x}^2, \quad (20)$$

noting that the second-derivative of the transformed cost-and-constraints \mathbf{v} is readily available

$$\partial^2 \mathbf{v}[\Delta \mathbf{x}] = \partial \mathbf{c} \cdot (p-1) \mathbf{x}_{\mathbf{I}}^{1-p} \cdot (\mathbf{x} + \Delta \mathbf{x})^{p-2}, \quad (21)$$

and, at the current decision point $\Delta \mathbf{x} = \mathbf{0}$, it reduces to

$$\partial^2 \mathbf{v}[\mathbf{0}] = \partial \mathbf{c} \cdot \left(\frac{p-1}{\mathbf{x}} \right)_{\mathbf{I}}. \quad (22)$$

This notion of ‘approximated approximations’ (approximation of decision space mappings) was first introduced, in this manner, by Groenwold *et al.* [4]. In significant practical terms, the 2nd-order Taylor expansion allows for a flexible and general interface with common, state-of-the-art polynomial time (subproblem) solvers, which are fed with the first and (approximate) second-order information of problem \mathcal{P} , as formulated in sequential subproblems \mathcal{S} .

We can therefore extend subproblem \mathcal{S} as ...

$$\left\{ \begin{array}{l} \boldsymbol{\nu} = (\nu_0, \nu_1, \dots, \nu_m) \\ \partial \boldsymbol{\nu} = (\partial \nu_0, \partial \nu_1, \dots, \partial \nu_m) \\ \Delta \bar{\mathbf{x}} = (\Delta \underline{\mathbf{x}}, \Delta \bar{\mathbf{x}}) \end{array} \right\} = \mathcal{S}[\Delta \mathbf{x}, \mathbf{v}[\mathbf{0}] = \mathbf{c}, \partial \mathbf{v}[\mathbf{0}] = \partial \mathbf{c}, \partial^2 \mathbf{v}[\mathbf{0}]], \quad (23)$$

that is ... describe input and output.... describe solution by dual

Quadratically constrained quadratic program [5] traditionally solved by dual method due to Falk [6]. Modern primal-dual second-order cone program solvers remain illusive... IBM CPLEX... [7] to the authors knowledge.

5 Quadratic programs and the necessary conditions

Introduce as simplification to arrive at common

[8]

Solution methods for the form

in this way, the notion of decision space transformations become obsolete to some extent, in practical terms. Although the notion of it may be used to estimate the second order information...

in rough drafting If we use only diagonal information. Ensure it is positive definite. and formulate the hessian of the lagrangian ...

Then we arrive at a convenient computational ‘kernel’: the quadratic program (QP)^{¶¶}. In simple terms, polynomial time solution methods for QPs are often the cheapest, fastest, simplest to implement, interface, and most readily available (to an average computational engineer), and ... for subproblem \mathcal{S} , given some straight-forward conditions. See for example the recently released implementation by Boyd and collaborators [8].

TO BE COMPLETED

Eventually we arrive rather naturally at the necessary conditions: How far can we go, before we have to evaluate the problem again..? Keeping in mind, that we can not afford to evaluate the problem an excessive number of times... etc. etc. Move limits and estimates of nonlinear information (surrogate functions/problems, second order estimates).... upon convergence / when we can stop, we get something for free: the necessary conditions are satisfied / we have arrived at the necessary conditions in a rather practical way. Done.

Example: The Van der Plaats cantilever beam

To be completed. Implementation near complete.

Background

In draft; more to follow. Fadel 1990. Duysinx. (Perhaps write in terms of ‘contributions’.)

The introduction to the ‘Elements of Structural Optimization’ by Haftka, Gürdal and Kamat [9]. The review of sensitivity analysis in structural optimization by van Keulen, Haftka and Kim [10].

Fleury [2, 3] and Svanberg [11]

The objected orientated framework present by Jacobs, Etman and van Keulen [12]. [13]. The works of Groenwold and collaborators on approximation concepts [4, 14, 15, 16, 17, 18, 19, 20, 21, 22], and the citations therein.

Simple descriptions of enforced convergence and termination and algorithmic representations [23, 24, 25, 26].

Finally, the reader is invited to peruse Chapter 3 (*Problem-solving*), in the *Fabric of Reality* (David Deutsch) for a philosophical basis in the Popperian spirit.

^{¶¶}Conceptually, the step-wise solving of QP problems in general, non-linear programming, can be seen as equivalent to the way in which a linear system-solve is the ‘kernel’ in nonlinear structural analysis. That is the ‘kernel’ of the Newton-Rhapson method.

References

- [1] An interpretation of Bayesian global optimization as proposed by Snyman and Fatti. https://github.com/dirkmunro89/SOAPs/blob/main/bayopt/bay_20220104.pdf. Accessed: 2022-02-27.
- [2] C. Fleury and V. Braibant. Structural optimization: a new dual method using mixed variables. *International journal for numerical methods in engineering*, 23(3):409–428, 1986.
- [3] C. Fleury. CONLIN: an efficient dual optimizer based on convex approximation concepts. *Structural optimization*, 1(2):81–89, 1989.
- [4] A.A. Groenwold, L.F.P. Etman, and D.W. Wood. Approximated approximations for SAO. *Structural and Multidisciplinary Optimization*, 41(1):39–56, 2010.
- [5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [6] J.E. Falk. Lagrange multipliers and nonconvex programs. *SIAM Journal on Control*, 7(4):534–545, 1969.
- [7] Cplex, IBM ILOG. V12. 1: User’s Manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.
- [8] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.
- [9] R.T. Haftka, Z. Gürdal, and M.P. Kamat. *Elements of structural optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
- [10] F. Van Keulen, R.T. Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer methods in applied mechanics and engineering*, 194(30):3213–3243, 2005.
- [11] K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.
- [12] J.H. Jacobs, L.F.P. Etman, F. Van Keulen, and J.E. Rooda. Framework for sequential approximate optimization. *Structural and Multidisciplinary Optimization*, 27(5):384–400, 2004.
- [13] A.A. Groenwold and L.F.P. Etman. SAOi: an algorithm for very large scale optimal design. In *Proceedings of the 9th World Congress on Structural and Multidisciplinary Optimization (WCSMO-9), 13-17 Juni 2011, Shizuoka, Japan*, 2011.

- [14] A.A. Groenwold and L.F.P. Etman. A quadratic approximation for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 82(4):505–524, 2010.
- [15] A.A. Groenwold and L.F.P. 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*, 73(3):297–316, 2008.
- [16] A.A. Groenwold, L.F.P. Etman, J.A. Snyman, and J.E. Rooda. Incomplete series expansion for function approximation. *Structural and Multidisciplinary Optimization*, 34(1):21–40, 2007.
- [17] L.F.P. Etman, A.A. Groenwold, and J.E. Rooda. First-order sequential convex programming using approximate diagonal qp subproblems. *Structural and Multidisciplinary Optimization*, 45(4):479–488, 2012.
- [18] S.L. Lutchman, A.A. Groenwold, P. Gauché, and S. Bode. On using a gradient-based method for heliostat field layout optimization. *Energy Procedia*, 49:1429–1438, 2014.
- [19] A.A. Groenwold and L.F.P. Etman. Sequential approximate optimization using dual subproblems based on incomplete series expansions. *Structural and Multidisciplinary Optimization*, 36(6):547–570, 2008.
- [20] L.F.P. Etman, A.A. Groenwold, and J.E. Rooda. On diagonal qp subproblems for sequential approximate optimization. Citeseer.
- [21] D.W. Wood and A.A. Groenwold. Non-convex dual forms based on exponential intervening variables, with application to weight minimization. *International journal for numerical methods in engineering*, 80(12):1544–1572, 2009.
- [22] L.F.P. Etman, A.A. Groenwold, and J. Rooda. Sequential approximate optimization in an NLP filter framework. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 7078.
- [23] A.A. Groenwold and L.F.P. Etman. On the conditional acceptance of iterates in sao algorithms based on convex separable approximations. *Structural and Multidisciplinary Optimization*, 42(2):165–178, 2010.
- [24] A.A. Groenwold, D.W. Wood, L.F.P. Etman, and S. Tosserams. Globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. *AIAA journal*, 47(11):2649, 2009.
- [25] S. Park, S.H. Jeong, G.H. Yoon, A.A. Groenwold, and D-H. Choi. A globally convergent sequential convex programming using an enhanced two-point diagonal quadratic approximation for structural optimization. *Structural and Multidisciplinary Optimization*, 50(5):739–753, 2014.

- [26] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM journal on optimization*, 12(2):555–573, 2002.