

# First-order sequential convex programming using approximate diagonal QP subproblems

**Citation for published version (APA):**

Etman, L. F. P., Groenwold, A. A., & Rooda, J. E. (2012). First-order sequential convex programming using approximate diagonal QP subproblems. *Structural and Multidisciplinary Optimization*, 45(4), 479-488. <https://doi.org/10.1007/s00158-011-0739-3>

**DOI:**

[10.1007/s00158-011-0739-3](https://doi.org/10.1007/s00158-011-0739-3)

**Document status and date:**

Published: 01/01/2012

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# First-order sequential convex programming using approximate diagonal QP subproblems

L. F. P. Etman · Albert A. Groenwold · J. E. Rooda

Received: 5 July 2011 / Revised: 29 September 2011 / Accepted: 27 October 2011  
© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** Optimization algorithms based on convex separable approximations for optimal structural design often use reciprocal-like approximations in a dual setting; CONLIN and the method of moving asymptotes (MMA) are well-known examples of such sequential convex programming (SCP) algorithms. We have previously demonstrated that replacement of these nonlinear (reciprocal) approximations by their own second order Taylor series expansion provides a powerful new algorithmic option within the SCP class of algorithms. This note shows that the quadratic treatment of the original nonlinear approximations also enables the restatement of the SCP as a series of Lagrange-Newton QP subproblems. This results in a diagonal trust-region SQP type of algorithm, in which the second order diagonal terms are estimated from the nonlinear (reciprocal) intervening variables, rather than from historic information using an exact or a quasi-Newton Hessian approach. The QP

formulation seems particularly attractive for problems with far more constraints than variables (when pure dual methods are at a disadvantage), or when both the number of design variables and the number of (active) constraints is very large.

**Keywords** Diagonal quadratic approximation · Sequential approximate optimization (SAO) · Sequential quadratic programming (SQP) · Sequential convex programming (SCP) · Reciprocal intervening variables · Trust region method

## 1 Introduction

Gradient-based optimization algorithms that rely on nonlinear but convex separable approximation functions have proven to be very effective for large-scale structural optimization. Well-known examples are the convex linearization (CONLIN) algorithm (Fleury and Braibant 1986) and its generalization, the method of moving asymptotes (MMA) (Svanberg 1987, 2002). These algorithms—and some related variants, e.g. see Borrvall and Petersson (2001), Bruyneel et al. (2002) and Zillober et al. (2004)—are also known as sequential convex programming (SCP) methods (Fleury 1993; Zillober et al. 2004; Duysinx et al. 2009).

The aforementioned SCP algorithms are all based on reciprocal or reciprocal-like approximations. They generate a series of convex separable nonlinear programming (NLP) subproblems. The derivation of the reciprocal-like approximations typically starts from the substitution of reciprocal intervening variables into a first-order (linear) Taylor series expansion, which is subsequently convexified, and

---

Loosely based on the paper ‘On diagonal QP subproblems for sequential approximate optimization’, presented at the 8-th World Congress on Structural and Multidisciplinary Optimization, 1–5 June, 2009, Lisbon, Portugal, paper 1065.

---

L. F. P. Etman (✉) · J. E. Rooda  
Department of Mechanical Engineering, Eindhoven University  
of Technology, Eindhoven, The Netherlands  
e-mail: l.f.p.etman@tue.nl

J. E. Rooda  
e-mail: j.e.rooda@tue.nl

A. A. Groenwold  
Department of Mechanical Engineering, University of Stellenbosch,  
Stellenbosch, South Africa  
e-mail: albertg@sun.ac.za

possibly enhanced using historic information. The resulting approximations provide for reasonably accurate subproblem approximations, while separability and convexity of the objective and constraint function approximations allows for the development of efficient solution approaches for the approximate subproblems. In many of the cited references, the dual formulation of Falk is used (Falk 1967; Fleury 1979), but other efficient subproblem solvers have also been proposed, see e.g. Zillober (2001) and Zillober et al. (2004).

In the spirit of the early efforts by Schmit and Farshi (1974), the last decades have seen the development of a variety of separable and non-separable local approximations based on intervening variables for use in sequential approximate optimization (SAO), e.g. see Haftka and Gürdal (1991), Barthelemy and Haftka (1993), Vanderplaats (1993), Groenwold et al. (2007) and Kim and Choi (2008). The intervening variables yield approximations that are nonlinear and possibly non-convex in terms of the original or direct variables. The resulting subproblems are typically solved in their primal form by means of an appropriate mathematical programming algorithm. Even though the subproblem may be separable, either the non-convexity or the inability to arrive at an analytical primal-dual relation may hinder the utilization of the dual formulation. This partly explains why these often highly accurate nonlinear approximations are not as widely used in large-scale structural optimization as the reciprocal type of approximations.

Recently, we have reported that SAO based on the replacement of approximations using specific nonlinear intervening variables *by their own* convex diagonal quadratic Taylor series expansions, may perform equally well, or sometimes even better, than the original (convex) nonlinear approximations. In Groenwold and Etman (2010b), we have demonstrated that the diagonal quadratic approximation to the reciprocal and exponential approximate objective functions can successfully be used in topology optimization. We have generalized this observation even further in Groenwold et al. (2010), where diagonal quadratic approximations to *arbitrary* nonlinear approximate objective and constraint functions were constructed; the resulting subproblems are convexified (when necessary), and cast into the dual statement. This gives an SCP type of algorithm which uses diagonal quadratic instead of reciprocal type of approximations. However, these truly quadratic approximations behave reciprocal-like. In addition, the form of the dual subproblem does not depend on which nonlinear approximations or intervening variables are selected for quadratic treatment: all the approximated approximations can be used simultaneously in a single dual statement. In Groenwold et al. (2010), diagonal quadratic replace-

ments for the reciprocal, exponential, CONLIN, MMA, and TANA (Xu and Grandhi 1998) approximations were presented. We have described this approach with the term “*approximated-approximations*”.

In this note, we convey the observation that the approximated-approximations approach also allows for the development of an SCP method that consists of series of diagonal QP subproblems, in the spirit of the well-known SQP algorithms.

SQP methods construct Hessian matrices using second-order derivative information of the Lagrangian function, often through an approximate quasi-Newton updating scheme such as BFGS or SR1, e.g. see Nocedal and Wright (2006), and many others. The storage and update of the Hessian matrix may become burdensome for large-scale structural optimization problems with very many design variables, such as those encountered in topology optimization. (Note that BFGS updates, etc. result in dense matrices, even if the original problem is sparse. While the dense matrices need not be stored in limited memory implementations, the computational effort required for the matrix-vector multiplications is significant.) For this reason, Fleury (1989) proposed an SQP method that uses diagonal Hessian information only. He developed methods to efficiently calculate the diagonal second order derivatives in a finite element environment.

Our approximated-approximations approach also allows for the estimation of diagonal Hessian information without using historic information or exact Hessian information. The diagonal curvature estimates follow directly from the selected intervening variables. For reciprocal intervening variables, this means that at every given iteration point, only function value and gradient information evaluated at the *current* iterate is required, similar to existing gradient-based SCP methods. Various (two- or multipoint) extensions to this are of course possible, but not discussed herein for the sake of brevity. A related but different approach was presented by Fleury (2009), who used an inner loop of QP subproblems generated using second order Taylor series to solve each nonlinear intervening-variables based approximate subproblem generated by the outer loop.

This note is arranged as follows. Section 2 presents the optimization problem statement, and Section 3 summarizes selected aspects of importance for SQP methods in mathematical programming. Subsequently, we present the basic concepts of SAO methods commonly used in structural optimization in Section 4, with a focus on SCP algorithms. In Section 5, we develop a first-order SCP algorithm based on diagonal QP subproblems. Numerical results are offered in Section 6, followed by selected conclusions in Section 7.

## 2 Optimization problem statement

We consider the nonlinear inequality constrained (structural) optimization problem

$$\begin{aligned} & \min_{\mathbf{x}} f_0(\mathbf{x}), \\ & \text{subject to } f_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, m, \\ & \quad \mathbf{x} \in \mathcal{C} \subseteq \mathbb{R}^n, \\ & \text{with } \mathcal{C} = \{\mathbf{x} \mid \check{x}_i \leq x_i \leq \hat{x}_i, \quad i = 1, \dots, n\}, \end{aligned} \quad (1)$$

where  $f_0(\mathbf{x})$  is a real valued scalar objective function, and the  $f_j(\mathbf{x})$  are  $m$  inequality constraint functions.  $\check{x}_i$  and  $\hat{x}_i$  respectively indicate lower and upper bounds of continuous real variable  $x_i$ . The functions  $f_j(\mathbf{x})$ ,  $j = 0, 1, \dots, m$  are assumed to be (at least) once continuously differentiable. Of particular importance here is that we assume that the evaluation of (part of) the  $f_j(\mathbf{x})$ ,  $j = 0, 1, \dots, m$ , requires an expensive numerical analysis, for instance a finite element structural analysis. We furthermore assume that gradients  $\partial f_j / \partial x_i$  can be efficiently and accurately be calculated, see e.g. van Keulen et al. (2005).

Herein, we are in particular interested in solving the large scale variant of problem (1), with a large number of design variables and constraints. We consider algorithms based on gradient-based approximations for which the approximation functions  $\tilde{f}_j^{[k]}$  developed at iterate  $\mathbf{x}^{[k]}$  satisfy (at least) the first-order conditions  $\tilde{f}_j^{[k]}(\mathbf{x}^{[k]}) = f_j(\mathbf{x}^{[k]})$ , and  $\partial \tilde{f}_j^{[k]} / \partial x_i(\mathbf{x}^{[k]}) = \partial f_j^{[k]} / \partial x_i(\mathbf{x}^{[k]})$ , e.g. see Alexandrov et al. (1998). We distinguish between two classes of algorithms: general purpose mathematical programming approaches, and the domain specific SCP approaches based on convex separable approximations. General purpose nonlinear programming algorithms aim to robustly and efficiently solve optimization problem (1), irrespective of the origin of the optimization problem. SCP approaches on the other hand use specialized (e.g. reciprocal type) approximation models motivated by (engineering) knowledge about the application at hand.

## 3 Sequential quadratic programming

Recently, Gould et al. (2005) reviewed mathematical programming methods for large-scale nonlinear optimization. Two established classes of algorithms are sequential quadratic programming (SQP) methods and interior-point methods. SQP methods generate steps by solving a sequence of quadratic subproblems. Interior point algorithms avoid the combinatorial difficulty of finding the

active inequality constraints for every subproblem by transforming the original problem into an equality constrained barrier problem. Refer to Nocedal and Wright (2006) for an extensive introduction and overview of mathematical programming methods for (large-scale) numerical optimization. In this section, we briefly summarize some aspects of Sequential Quadratic Programming needed to develop our approach.

### 3.1 Line search versus trust region SQP

There are two fundamental strategies to move from some current iteration point  $\mathbf{x}^{[k]}$  to a new iterate  $\mathbf{x}^{[k+1]}$ , namely line search strategies and trust region strategies. In line search SQP methods, a quadratic programming approximation subproblem is constructed at  $\mathbf{x}^{[k]}$ . The solution to this QP subproblem provides the line search direction. The new iterate is subsequently obtained by an (inexact) one-dimensional minimization of some merit function which balances the competing goals of reducing the objective function and satisfying the constraints. In trust region SQP methods, no line search is carried out, but instead, the QP subproblem with an additional trust region is solved. The solution to the QP subproblem with trust region provides the step to the new iterate, under the condition that it leads to a sufficient reduction in a merit function. Otherwise, the subproblem is re-solved with a reduced trust region. Alternative mechanisms for step acceptance are filter techniques, which use ideas from multiobjective optimization to replace the merit function. In certain cases, subproblems may be solved in an approximate sense only, rather than exactly.

### 3.2 Approximation subproblem in SQP

Given the inequality constrained programming problem (1), the QP subproblem at  $\mathbf{x}^{[k]}$  is written as

$$\begin{aligned} & \min_{\mathbf{s}} f_0(\mathbf{x}^{[k]}) + \nabla f_0^T(\mathbf{x}^{[k]})\mathbf{s} + \frac{1}{2}\mathbf{s}^T \nabla_{xx}^2 L(\mathbf{x}^{[k]})\mathbf{s} \\ & \text{subject to } \nabla f_j^T(\mathbf{x}^{[k]})\mathbf{s} + f_j(\mathbf{x}^{[k]}) \leq 0, \quad j = 1, \dots, m, \\ & \quad (\|\mathbf{s}\|_\infty \leq \Delta^{[k]}), \end{aligned} \quad (2)$$

with  $\mathbf{s} \equiv \mathbf{x} - \mathbf{x}^{[k]}$  the trial step,  $\nabla f_j(\mathbf{x}^{[k]})$  the column vector with gradients  $\partial f_j / \partial x_i$  evaluated at  $\mathbf{x}^{[k]}$ , and  $\nabla_{xx}^2 L(\mathbf{x}^{[k]})$  the Hessian of the Lagrange function

$$L(\mathbf{x}, \lambda) = f_0(\mathbf{x}) + \sum_{j=1}^m \lambda_j f_j(\mathbf{x}), \quad (3)$$

evaluated at  $\mathbf{x}^{[k]}$ . Note that the last equation line in (2) represents the trust region, which is only included in the trust

region SQP algorithm. Updates of the Lagrange multipliers follow along with the solution of the QP subproblem or from least-squares estimation based on the Jacobian matrix of the active constraints. For large-scale problems it may be advantageous to follow a sequential *equality*-constrained quadratic programming approach (SEQP) where for each EQP subproblem first an LP subproblem is solved to determine which inequality constraints are included as equality constraints. To overcome the possible difficulty of infeasible subproblems, relaxation procedures are used, which include the introduction of (elastic) slack variables.

### 3.3 Hessian approximations

Often it is advantageous to replace the true Hessian of the Lagrangian by an approximation, which is updated after each step. This update is based on the change in  $\mathbf{x}$  and the change in the gradient of the Lagrange function from iteration  $k$  to iteration  $k + 1$ . Well-known quasi Newton methods are BFGS or SR1, e.g. see Nocedal and Wright (2006) for details. Updating the Hessian of the Lagrangian may be problematic if the Hessian contains negative eigenvalues. Certain modifications to the original update formulas may then be necessary, again see Nocedal and Wright (2006). To avoid explicit storage of the complete Hessian (or inverse Hessian) limited memory updating can be applied, such as L-BFGS. A limited number of vectors of length  $n$  only are stored, which represent the approximate Hessian implicitly.

## 4 Sequential approximate optimization

In the mid-seventies, Schmit and his coworkers (Schmit and Farshi 1974; Schmit and Miura 1976) argued that applications of nonlinear programming methods to large structural design problems could prove cost effective, provided that suitable approximation concepts were introduced. As mentioned, key to the approximation concepts approach is the construction of high quality approximations for the objective function and constraints by incorporating application specific non-linearities through so-called intervening variables, e.g. see Barthelemy and Haftka (1993) and Vanderplaats (1993). This means that a series of approximate NLP optimization subproblems are generated, which are to be solved using a suitable solver. Barthelemy and Haftka (1993) identified three classes of function approximations: local, mid-range, and global. Herein, we consider gradient-based local approximations. We begin by summarizing the main aspects of SAO required for the development in sections to come. For a more detailed overview we refer the interested reader to Chapter 6 of the book by Haftka and Gürdal (1991) and the report by Duysinx et al. (2009).

### 4.1 Series of approximate optimization subproblems

Sequential approximate optimization as a solution strategy for problem (1) seeks to construct successive approximate subproblems  $P[k]$ ,  $k = 1, 2, 3, \dots$  at successive iteration points  $\mathbf{x}^{[k]}$ . That is, we seek suitable (analytical) approximation functions  $\tilde{f}_j$  that are inexpensive to evaluate. We write approximate optimization subproblem  $P[k]$  for problem (1) as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \tilde{f}_0^{[k]}(\mathbf{x}) \\ \text{subject to} \quad & \tilde{f}_j^{[k]}(\mathbf{x}) \leq 0, \quad j = 1, \dots, m, \\ & \mathbf{x} \in \mathcal{C}^{[k]} \\ & \text{with } \mathcal{C} = \{\mathbf{x} \mid \check{x}_i \leq x_i \leq \hat{x}_i, \quad i = 1, \dots, n\} \end{aligned} \quad (4)$$

which has  $n$  unknowns,  $m$  constraints, and  $2n$  side or bound constraints.

To guarantee that the sequence of iteration points approaches the solution  $\mathbf{x}^*$ , one may cast the SAO method into a trust region framework (Alexandrov et al. 1998), or in a framework of variable conservatism (Svanberg 2002). In the trust region framework, an allowed search domain is defined around  $\mathbf{x}^{[k]}$ , and incorporated into closed set  $\mathcal{C}$ . That is, for infinity-norm move limits,  $\mathcal{C}$  becomes:

$$\begin{aligned} \mathcal{C}^{[k]} = \{ \mathbf{x} \mid & -\delta_i^{[k]} \leq x_i - x_i^{[k]} \leq \delta_i^{[k]}, \\ & \check{x}_i \leq x_i \leq \hat{x}_i, \quad i = 1, \dots, n \}. \end{aligned} \quad (5)$$

The size of the search subregion may be manipulated to enforce termination. In the conservatism framework, no trust region is introduced. Instead, the conservatism of the objective and constraint function approximations is adjusted to enforce convergence. In both frameworks,  $\mathbf{x}^{[k^*]}$  is only conditionally accepted to become the new iterate  $\mathbf{x}^{[k+1]}$ . If  $\mathbf{x}^{[k^*]}$  is rejected, the subproblem is re-solved with a reduced trust region, or with increased conservatism. An alternative framework, which aims to combine the salient features of trust regions and conservatism, is discussed in Groenwold and Etman (2010a).

### 4.2 Intervening variables

For several structural optimization applications, it is known that intervening variables may yield nonlinear approximations of significantly better accuracy than the (linear) Taylor series in the original (direct) variables  $\mathbf{x}$ . The intervening variables concept in SAO is best explained by departing with the first order Taylor series expansion, and expressing

the Taylor series in terms of intervening variables  $y_i(x_i)$ :

$$\tilde{f}_1(\mathbf{y}) = f(\mathbf{y}^{(k)}) + \sum_{i=1}^n \frac{\partial f^{(k)}}{\partial y_i} (y_i - y_i^{(k)}). \quad (6)$$

It is typically assumed that  $y_i(x_i)$  is a continuous and monotonic expression over the interval  $[y_i(\hat{x}_i), y_i(\check{x}_i)]$ , to ensure that the mappings are bijective. For example, reciprocal intervening variables are expressed as

$$y_i = x_i^{-1}, \quad i = 1, \dots, n. \quad (7)$$

Substitution into (6) yields (Haftka and Gürdal 1991)

$$\tilde{f}_R(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \sum_{i=1}^n \left( x_i - x_i^{(k)} \right) \frac{x_i^{(k)}}{x_i} \left( \frac{\partial f}{\partial x_i} \right)^{(k)}. \quad (8)$$

Following this principle, many other intervening variables have been developed, see the references mentioned in the introduction section.

### 4.3 Sequential convex programming algorithms

It is not necessarily advantageous to treat all design variables with the same intervening variables. A popular hybrid variant is to treat reciprocally only those design variables that have a negative gradient value, and select the linear approximation for the other design variables (Starnes and Haftka 1979). Fleury and Braibant (1986) approximate the objective function and all the constraint functions using this hybrid reciprocal-linear approximation. They show that the resulting approximate subproblem is convex and separable and can be efficiently solved by means of the dual (Falk 1967):

$$\begin{aligned} \max_{\lambda} \quad & \min_{\mathbf{x} \in \mathcal{C}} L(\mathbf{x}, \lambda) \\ \text{s.t.} \quad & \lambda_j \geq 0, \quad j = 1, 2, \dots, m, \end{aligned} \quad (9)$$

where, given primal approximate subproblem (4), the corresponding Lagrange function becomes:

$$L^{(k)}(\mathbf{x}, \lambda) = \tilde{f}_0^{(k)}(\mathbf{x}) + \sum_{j=1}^m \lambda_j \tilde{f}_j^{(k)}(\mathbf{x}). \quad (10)$$

Lower and upper bounds  $\check{x}_i$  and  $\hat{x}_i$  are part of convex set  $\mathcal{C}$  in (9), and do not introduce Lagrange multipliers. The dimensionality of the dual problem is therefore determined by the number of inequality constraints  $m$ . Clearly, this is advantageous if  $m \ll n$ . For the separable reciprocal-linear approximations, the nested minimization problem can be analytically solved. The resulting maximization problem in terms of the dual variables may be solved using any suitable first order or (apparently) second order method able to handle discontinuous second derivatives. If the primal

subproblems happens to be infeasible, relaxation may for instance be used (see e.g. Svanberg 2002).

The dual method based on the mixed reciprocal-linear approximations is known as convex linearization (CONLIN). The concept to generate a series of convex separable subproblems has been further generalized in the method of moving asymptotes (MMA), introduced by Svanberg (1987). The reciprocal intervening variable is augmented with an additional asymptote that allows to adjust the curvature of the approximation. The MMA approach is again hybrid in the sense that design variables with positive gradients are treated different to design variables with negative gradient values. The intervening variables become:

$$\frac{1}{x_i - L_i} \quad \text{if} \quad \frac{\partial f}{\partial x_i} < 0 \quad \text{and} \quad \frac{1}{U_i - x_i} \quad \text{if} \quad \frac{\partial f}{\partial x_i} > 0. \quad (11)$$

The adjustment of the asymptotes  $U_i$  and  $L_i$  may be done heuristically as the optimization progresses, or guided by function value and gradient information evaluated at the previous iterate.

Several variants of, and extensions to, the original MMA algorithm have been developed during the last decades, e.g. see Borrvall and Petersson (2001), Bruyneel et al. (2002) and Zillober (2002). Various approaches for enforcing global convergence have also been proposed. These include line search variants of MMA (Zillober 1993), and MMA cast in a framework of variable conservatism (Svanberg 2002). Despite the risk of divergence, the simple ‘always accept’ strategy is nevertheless frequently adopted and found to be effective and efficient in many a structural optimization application (Groenwold and Etman 2010a).

## 5 Diagonal QP subproblems for SCP

The SCP class of algorithms based on convex separable nonlinear approximations has become very popular in large-scale structural optimization, and in topology optimization in particular. In structural topology optimization, SCP methods are almost exclusively used, in particular when optimality criteria update formulas do not apply. SQP type of methods are far less frequently used, due to the high dimensionality of many structural optimization problems.

We now show how an SQP type method can be developed from the SCP class of methods by means of the approximated approximations concept presented in Groenwold et al. (2010).

### 5.1 Diagonal quadratic function approximations

We depart with the diagonal quadratic Taylor approximations  $\tilde{f}_j$  to some objective and/or constraint function  $f_j$ , given by

$$\begin{aligned} \tilde{f}_j(\mathbf{x}) &= f_j(\mathbf{x}^{(k)}) + \sum_{i=1}^n \left( \frac{\partial f_j}{\partial x_i} \right)^{(k)} (x_i - x_i^{(k)}) \\ &+ \frac{1}{2} \sum_{i=1}^n c_{2i_j}^{(k)} (x_i - x_i^{(k)})^2, \quad j = 0, 1, \dots, m, \end{aligned} \tag{12}$$

but with the  $c_{2i_j}^{(k)}$  approximate second order diagonal Hessian terms or curvatures. We neglect the off-diagonal Hessian terms. So, our departing point is a separable form of the sequential all-quadratic programming method, in which quadratic approximations are simultaneously used for all constraint functions (Svanberg 1993; Zhang and Fleury 1997; Fleury and Zhang 2000).

Since we assume that the user only provides function value and gradient information, the second order coefficients  $c_{2i_j}^{(k)}$  are to be estimated in some manner. To guarantee strictly convex forms of the approximate subproblems, we may, for example, enforce

$$\begin{aligned} c_{2i_0}^{(k)} &= \max(\varepsilon_0 > 0, c_{2i_0}^{(k)}), \\ c_{2i_j}^{(k)} &= \max(0, c_{2i_j}^{(k)}), \quad j = 1, 2, \dots, m. \end{aligned} \tag{13}$$

We estimate the approximate second order curvatures  $c_{2i_j}^{(k)}$  by building a Taylor series expansion to the nonlinear intervening variables based approximations used in the SCP methods. If we start from first-order expansion (6), we obtain

$$c_{2i}^{(k)} = \left( \frac{\partial^2 \tilde{f}_1}{\partial x_i^2} \right)^{(k)} = \left( \frac{\partial f}{\partial x_i} \right)^{(k)} \left( \frac{\partial x_i}{\partial y_i} \right)^{(k)} \left( \frac{\partial^2 y_i}{\partial x_i^2} \right)^{(k)}. \tag{14}$$

To illustrate: for the popular reciprocal intervening variables  $y_i = 1/x_i$ , this results in

$$c_{2i}^{(k)} = \frac{-2}{x_i^{(k)}} \left( \frac{\partial f}{\partial x_i} \right)^{(k)}, \tag{15}$$

e.g. see Zhang and Fleury (1997) and Bruyneel et al. (2002). In other words: using the approximate curvatures (15) in (12) implies that (12) becomes the quadratic approximation to the reciprocal approximation in the point  $\mathbf{x}^{(k)}$  (Haftka

2007, personal communication). For many a structural optimization problem, it seems to be advantageous to replace (15) by

$$c_{2i}^{(k)} = \frac{2}{x_i^{(k)}} \left| \frac{\partial f}{\partial x_i} \right|^{(k)}, \tag{16}$$

which yields a more conservative approximation than the combination of (15) with (13) for positive gradients.

For other diagonal quadratic approximations to a number of different well known nonlinear approximations, including the exponential, CONLIN, MMA and TANA approximations, see Groenwold et al. (2010).

### 5.2 Approximate subproblem in dual form

Given primal approximate optimization subproblem (4), with the approximate functions  $\tilde{f}_j$ ,  $j = 0, 1, \dots, m$  given by the (convex) diagonal quadratic approximations (12), the approximate dual subproblem  $P_D[k]$  becomes

$$\max_{\lambda} \left\{ \gamma(\lambda) = \tilde{f}_0^{(k)}(\mathbf{x}(\lambda)) + \sum_{j=1}^m \lambda_j \tilde{f}_j^{(k)}(\mathbf{x}(\lambda)) \right\}, \tag{17}$$

subject to  $\lambda_j \geq 0$ ,  $j = 1, 2, \dots, m$ ,

with the primal-dual relationship between variables  $x_i$ ,  $i = 1, 2, \dots, n$  and  $\lambda_j$ ,  $j = 1, 2, \dots, m$ , given by

$$x_i(\lambda) = \begin{cases} \beta_i(\lambda) & \text{if } \check{x}_i^{(k)} < \beta_i(\lambda) < \hat{x}_i^{(k)}, \\ \check{x}_i^{(k)} & \text{if } \beta_i(\lambda) \leq \check{x}_i^{(k)}, \\ \hat{x}_i^{(k)} & \text{if } \beta_i(\lambda) \geq \hat{x}_i^{(k)}, \end{cases} \tag{18}$$

and

$$\begin{aligned} \beta_i(\lambda) &= x_i^{(k)} - \left( c_{2i_0}^{(k)} + \sum_{j=1}^m \lambda_j c_{2i_j}^{(k)} \right)^{-1} \\ &\times \left( \frac{\partial f^{(k)}}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial f_j^{(k)}}{\partial x_i} \right). \end{aligned} \tag{19}$$

For details, the reader is referred to our previous efforts (Groenwold and Etman 2008). A globally convergent algorithm may be obtained by casting dual subproblems  $P_D[k]$  in a framework of variable conservatism, or in a trust region framework, see Groenwold et al. (2009) and Groenwold and Etman (2010a).

### 5.3 Approximate subproblem in QP form

Since the approximations (12) are (diagonal) quadratic, the subproblems are easily transformed into a quadratic program  $P_{QP}[k]$ , written as

$$\begin{aligned} \min_{\mathbf{s}} \tilde{f}_0^{(k)}(\mathbf{s}) &= f_0(\mathbf{x}^{(k)}) + \nabla f_0^T(\mathbf{x}^{(k)})\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{Q}^{(k)}\mathbf{s} \\ \text{subject to } \tilde{f}_j^{(k)}(\mathbf{s}) &= f_j(\mathbf{x}^{(k)}) + \nabla f_j^T(\mathbf{x}^{(k)})\mathbf{s} \leq 0, \quad (20) \\ \| \mathbf{s} \|_{\infty} &\leq \Delta^{(k)}, \end{aligned}$$

with  $j = 1, 2, \dots, m$ ,  $\mathbf{s} = (\mathbf{x} - \mathbf{x}^{(k)})$  and  $\mathbf{Q}^{(k)}$  the Hessian matrix of the approximate Lagrangian at  $\mathbf{x}^{(k)}$ .

Using the diagonal quadratic objective function and constraint function approximations  $\tilde{f}_j$ ,  $j = 0, 1, \dots, m$ , the approximate Lagrangian  $L^{(k)}$  equals

$$L^{(k)} = \tilde{f}_0(\mathbf{x}^{(k)}) + \sum_{j=1}^m \lambda_j^{(k)} \tilde{f}_j(\mathbf{x}^{(k)}), \quad (21)$$

with

$$Q_{ii}^{(k)} = c_{2i_0}^{(k)} + \sum_{j=1}^m \lambda_j^{(k)} c_{2i_j}^{(k)}, \quad (22)$$

and  $Q_{il} = 0 \forall i \neq l, i, l = 1, 2, \dots, n$ . The Lagrange multiplier values  $\lambda_j^{(k)}$  follow from the multiplier estimates due to the solution of the QP subproblem at the previous iterate  $\mathbf{x}^{(k-1)}$ . This approach is very similar to the well-known SQP method discussed in Section 3; the fundamental difference being the way the Hessian matrix of the Lagrangian function is determined. Instead of using the exact or approximate quasi-Newton Hessian matrices commonly used in classical SQP algorithms, we use only approximate diagonal terms, estimated from suitable intervening variable expressions for the objective function and all the constraints. As we did for the dual subproblems, we again apply convexity conditions (13), to arrive at a strictly convex QP subproblem with a unique minimizer.

The quadratic programming problem requires the determination of the  $n$  unknowns  $x_i$ , subject to  $m$  linear inequality constraints and the trust region bounds. Efficient QP solvers can typically solve problems with very large numbers of design variables  $n$  and constraints  $m$ . Obviously, it is imperative that the diagonal structure of  $\mathbf{Q}$  is exploited when the QP subproblems are solved.

To arrive at a globally convergent algorithm, the use of trust regions in SQP algorithms is of course well known, e.g. see Nocedal and Wright (2006), Conn et al. (2000), and many others.

### 6 Numerical example

We now illustrate the QP-based SCP approach as proposed in Section 5.3 using a numerical example, and compare this method with the dual SCP method presented in Section 5.2. We use the quadratic approximation to the reciprocal approximation (16) for the objective function and all the constraints, which allows for the unconditional acceptance of the iterates; convergence is obtained without the use of a global convergence strategy.

We consider the optimal sizing design of the tip-loaded multi-segmented cantilever beam proposed by Vanderplaats (1984, 2004). The beam is of fixed length  $l$ , is divided into  $p$  segments, and is subject to geometric, stress and a single displacement constraint. The geometry has been chosen such that a very large number of the constraints are active or ‘near-active’ at the optimum (Fig. 1).

The objective function is formulated in terms of the design variables  $b_i$  and  $h_i$  as

$$\min f_0(\mathbf{b}, \mathbf{h}) = \sum_{i=1}^p b_i h_i l_i,$$

with  $l_i$  constant for given  $p$ . We enforce the bound constraints  $1.0 \leq b_i \leq 80$ , and  $5.0 \leq h_i \leq 80$  (the upper bounds were arbitrarily chosen; they are needed to allow for the notion of a ‘move limit’). The stress constraints are

$$\frac{\sigma(\mathbf{b}, \mathbf{h})}{\bar{\sigma}} - 1 \leq 0, \quad i = 1, 2, \dots, p,$$

while the linear geometric constraints are written as

$$h_i - 20b_i \leq 0, \quad i = 1, 2, \dots, p.$$

The tip displacement constraint is

$$\frac{u_{\text{tip}}(\mathbf{b}, \mathbf{h})}{\bar{u}} - 1 \leq 0.$$

The constraints are rather easily written in terms of the design variables  $\mathbf{b}$  and  $\mathbf{h}$ , e.g. see Vanderplaats (1984). Note that the constraints are normalized; this is sound practice in primal algorithms. However, this may not be desirable

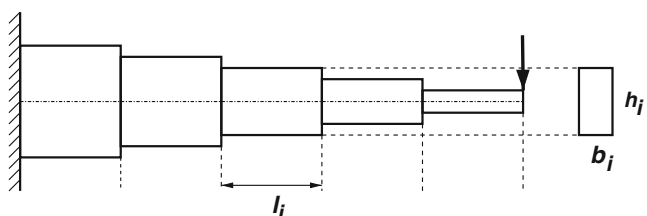


Fig. 1 Vanderplaats’ beam



**Table 1** Vanderplaats-beam with displacement constraint

$p$	$n$	$k^*$	$f_0^*$	$h^*$	CPU
5	10	8	65,419.64	$5.38 \cdot 10^{-06}$	0.01
$5 \cdot 10$	$10^2$	10	63,704.47	$3.86 \cdot 10^{-06}$	0.01
$5 \cdot 10^2$	$10^3$	11	63,665.62	$4.32 \cdot 10^{-07}$	0.04
$5 \cdot 10^3$	$10^4$	12	63,665.11	$3.77 \cdot 10^{-06}$	0.96
$5 \cdot 10^4$	$10^5$	12	63,665.10	$6.99 \cdot 10^{-05}$	50.40
$5 \cdot 10^5$	$10^6$	13	63,665.10	$2.06 \cdot 10^{-04}$	2,878.66

Computational effort for dual-SCP ( $m = n + 1$ )

in algorithms based purely on dual statements. We therefore scale the last constraint by  $10^3$  (to have all the dual variables of roughly the same order; this is easily determined for the problems of low dimensionality).

Using consistent units, the geometric and problem data are as follows (Vanderplaats 1984): we use a tip load of  $P = 50,000$ , a modulus of elasticity  $E = 2 \times 10^7$ , a beam length  $l = 500$ , while  $\bar{\sigma} = 14,000$ , and  $\bar{u} = 2.5$ . The starting point is  $b_i = 5.0$  and  $h_i = 60$  for all  $i$ .

We consider two cases of the Vanderplaats' beam problem: the original problem with tip displacement constraint and the problem without tip displacement constraint. The problems are both expressed in terms of  $n = 2p$  design variables; the number of constraints  $m$  may be found in the results tables.

We present computational results for two rudimentary SCP algorithms, denoted dual-SCP and QP-SCP, respectively. Algorithm dual-SCP implements dual subproblems (17), the dual subproblems are solved using the limited memory bound constrained L-BFGS-B solver (Zhu et al. 1994). Algorithm QP-SCP implements QP subproblem (20), this time we use the diagonal Galahad LSQP solver (Gould et al. 2004) to solve the QP subproblems. In the case of an infeasible subproblem the solver finds a subproblem solution with minimum constraint violation while respecting the imposed move limits. For both dual-SCP and QP-SCP we have used a 20% move limit (with respect to  $\hat{x}_i - \check{x}_i$ ) throughout (but it often is advantageous for QP

**Table 2** Vanderplaats-beam with displacement constraint

$p$	$n$	$k^*$	$f_0^*$	$h^*$	CPU
5	10	9	65,419.66	$1.38 \cdot 10^{-07}$	0.00
$5 \cdot 10$	$10^2$	11	63,704.47	$7.59 \cdot 10^{-09}$	0.04
$5 \cdot 10^2$	$10^3$	12	63,665.62	$3.30 \cdot 10^{-10}$	0.46
$5 \cdot 10^3$	$10^4$	12	63,665.11	$3.29 \cdot 10^{-10}$	5.50
$5 \cdot 10^4$	$10^5$	13	63,665.11	$1.70 \cdot 10^{-11}$	83.15
$5 \cdot 10^5$	$10^6$	14	63,665.11	$1.62 \cdot 10^{-11}$	1,193.77

Computational effort for QP-SCP ( $m = n + 1$ )

**Table 3** Vanderplaats-beam without displacement constraint

$p$	$n$	$k^*$	$f_0^*$	$h^*$	CPU
5	10	7	61,914.79	$1.02 \cdot 10^{-06}$	0.01
$5 \cdot 10$	$10^2$	9	54,605.11	$2.54 \cdot 10^{-06}$	0.08
$5 \cdot 10^2$	$10^3$	9	53,827.75	$5.78 \cdot 10^{-06}$	0.88
$5 \cdot 10^3$	$10^4$	10	53,749.44	$8.71 \cdot 10^{-06}$	16.77
$5 \cdot 10^4$	$10^5$	11	53,741.61	$7.02 \cdot 10^{-04}$	440.42
$5 \cdot 10^5$	$10^6$	–	–	–	–

Computational effort for dual-SCP ( $m = n$ )

problems to not do so). We terminate the iterations when  $\|\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]}\|_2 \leq \varepsilon_x$ , with  $\varepsilon_x = 10^{-3}$ . The algorithms are considered to have failed when the CPU time exceeds 5,000 s.

The numerical results for dual-SCP and QP-SCP are presented in Tables 1, 2, 3 and 4, for  $p$  ranging from 5 to  $5 \cdot 10^5$ . Each line in the tables represents the outcome of one optimization run. We have used compressed sparse row (CSR) data representation. We define  $h = \max\{f_j\}$ ,  $j = 1, 2, \dots, m$ ;  $h^*$  is the same at optimality, while  $l^*$  and  $u^*$  respectively indicate the number of design variables on the lower and upper bounds at the solution  $\mathbf{x}^*$ .  $k^*$  denotes the required number of iterations. For each optimal design found, we have verified that the first-order KKT conditions are satisfied to a reasonable accuracy (not explicitly shown). The reported CPU effort is in seconds.

For both cases considered, algorithm QP-SCP is superior to algorithm dual-SCP, albeit that algorithm dual-SCP is preferable for the case with a displacement constraint and low dimensionality. For  $p = 5 \cdot 10^5$  the dual algorithm timed out for the case without the displacement constraint, whereas algorithm QP-SCP converged rather nicely. However, not shown is that the computational effort required with QP-SCP is rather insensitive to the scaling of the constraints; for the Falk dual subproblems, ineffective scaling can increase the computational effort by up to an order of magnitude(!) Finally, the steep increase in CPU time

**Table 4** Vanderplaats-beam without displacement constraint

$p$	$n$	$k^*$	$f_0^*$	$h^*$	CPU
5	10	6	61,914.79	$1.21 \cdot 10^{-09}$	0.02
$5 \cdot 10$	$10^2$	8	54,605.12	$3.45 \cdot 10^{-13}$	0.04
$5 \cdot 10^2$	$10^3$	9	53,827.75	$3.83 \cdot 10^{-08}$	0.42
$5 \cdot 10^3$	$10^4$	10	53,749.44	$5.37 \cdot 10^{-14}$	5.05
$5 \cdot 10^4$	$10^5$	10	53,741.61	$0.00 \cdot 10^{+00}$	57.29
$5 \cdot 10^5$	$10^6$	10	53,740.83	$0.00 \cdot 10^{+00}$	706.20

Computational effort for QP-SCP ( $m = n$ )

with growing problem size is noteworthy. This has also been observed and discussed in further detail by Fleury (2009).

## 7 Conclusions

Through the concept of approximated-approximations, the sequence of nonlinear subproblems that are solved in SAO algorithms may be replaced by the solution of a sequence of diagonal Lagrange-Newton QP subproblems, in the spirit of classical SQP methods. This specializes to any SCP method that employs a sequence of convex separable nonlinear subproblems. That is, a SCP can be transformed into a SQP that uses an approximate diagonal Hessian matrix. The diagonal Hessian information may be obtained by replacing an arbitrary separable nonlinear intervening variable-based approximation by its own quadratic Taylor series expansion. Hence, only function values and gradient information are used; Hessian information is neither stored nor updated.

The resulting first-order SCP method based on diagonal QP subproblems is particularly promising for structural optimization problems where both the number of design variables and constraints are high. This may provide new opportunities for the use and development of algorithms for large-scale structural optimization. We have herein restricted ourselves to the quadratic approximation to the reciprocal approximation, but only for the sake of brevity; in principle any other nonlinear intervening variable based approximation (e.g. MMA and the exponential approximation) may be considered for quadratic treatment and cast into the QP format. What is more, different approximations may be used for the objective function, and for any or all of the constraint functions.

Finally, we have herein not addressed equality constraints, but equalities are included rather trivially in SQP and interior point methods.

**Acknowledgments** This research was supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs of the Netherlands.

During the finalization of this document, we have learned about the passing away of Professor Claude Fleury, whose efforts are the basis for much of our work. We would like to express our sincerest condolences to his family, his students, his colleagues at the University of Liège, and his colleagues at Samtech. We would like to thank Professor Fleury for his contributions to the field of structural optimization. His work will inspire us always.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Alexandrov AA, Dennis JE, Lewis RM, Torczon V (1998) A trust region framework for managing the use of approximation models in optimization. *Struct Multidisc Optim* 15:16–23
- Barthelemy JFM, Haftka RT (1993) Approximation concepts for optimum structural design - a review. *Struct Optim* 5:129–144
- Borrval T, Petersson J (2001) Large scale topology optimization in 3D using parallel computing. *Comput Methods Appl Mech Eng* 190:6201–6229
- Bruyneel M, Duysinx P, Fleury C (2002) A family of MMA approximations for structural optimization. *Struct Multidisc Optim* 24:263–276
- Conn AR, Gould NIM, Toint PL (2000) Trust-region methods. MPS/SIAM Series on Optimization, SIAM, Philadelphia
- Duysinx P, Bruyneel M, Fleury C (2009) Solution of large scale optimization problems with sequential convex programming. Tech. rep., LTAS - Aerospace and Mechanical Engineering, University of Liège
- Falk JE (1967) Lagrange multipliers and nonlinear programming. *J Math Anal Appl* 19:141–159
- Fleury C (1979) Structural weight optimization by dual methods of convex programming. *Int J Numer Methods Eng* 14:1761–1783
- Fleury C (1989) Efficient approximation concepts using second order information. *Int J Numer Methods Eng* 28:2041–2058
- Fleury C (1993) Mathematical programming methods for constrained optimization: dual methods. In: Kamat M (ed) *Structural optimization: status and promise of progress in astronautics and aeronautics*. Progress in Astronautics and Aeronautics, vol 150, AIAA, chap 7, pp 123–150
- Fleury C (2009) Structural optimization methods for large scale problems: computational time issues. In: Proc. 8th world congress on structural and multidisciplinary optimization, Lisbon, paper 1184
- Fleury C, Braibant V (1986) Structural optimization: a new dual method using mixed variables. *Int J Numer Methods Eng* 23:409–428
- Fleury C, Zhang WH (2000) Selection of appropriate approximation schemes in multi-disciplinary engineering optimization. *Adv Eng Softw* 31:385–389
- Gould N, Orban D, Toint PL (2004) GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Trans Math Softw* 29:353–372
- Gould N, Orban D, Toint P (2005) Numerical methods for large-scale nonlinear optimization. *Acta Numerica* 14:299–361
- Groenwold AA, Etman LFP (2008) Sequential approximate optimization using dual subproblems based on incomplete series expansions. *Struct Multidisc Optim* 36:547–570
- Groenwold AA, Etman LFP (2010a) On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations. *Struct Multidisc Optim* 42:165–178
- Groenwold AA, Etman LFP (2010b) A quadratic approximation for structural topology optimization. *Int J Numer Methods Eng* 82:505–524
- Groenwold AA, Etman LFP, Snyman JA, Rooda JE (2007) Incomplete series expansion for function approximation. *Struct Multidisc Optim* 34:21–40
- Groenwold AA, Wood DW, Etman LFP, Tisserams S (2009) A globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. *AIAA J* 47:2649–2657
- Groenwold AA, Etman LFP, Wood DW (2010) Approximated approximations in SAO. *Struct Multidisc Optim* 41:39–56
- Haftka RT, Gürdal Z (1991) *Elements of structural optimization*, 3rd edn. Kluwer Academic Publishers
- Kim JR, Choi DH (2008) Enhanced two-point diagonal quadratic approximation methods for design optimization. *Comput Methods Appl Mech Eng* 197:846–856

- Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Series in Operations Research, Springer, Berlin
- Schmit L, Farshi B (1974) Some approximation concepts for structural synthesis. *AIAA J* 12:692–699
- Schmit LA, Miura H (1976) Approximation concepts for efficient structural synthesis. Tech. Rep. Report NASA-CR 2552, NASA
- Starnes Jr JH, Haftka RT (1979) Preliminary design of composite wings for buckling, stress and displacement constraints. *J Aircr* 16:564–570
- Svanberg K (1987) The method of moving asymptotes - a new method for structural optimization. *Int J Numer Methods Eng* 24:359–373
- Svanberg K (1993) Some second order methods for structural optimization. In: Rozvany G (ed) *Optimization of Large Structural Systems*, NATA ASI series, vol 231, Kluwer Academic Publishers, pp 567–578
- Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J Optim* 12:555–573
- van Keulen F, Haftka RT, Kim NH (2005) Review of options for structural design sensitivity analysis. part 1: linear systems. *Comput Methods Appl Mech Engrg* 194:3213–3243
- Vanderplaats G (1993) Thirty years of modern structural optimization. *Adv Eng Softw* 16:81–88
- Vanderplaats GN (1984) Numerical optimization techniques for engineering design. McGraw-Hill, New York
- Vanderplaats GN (2004) Very large scale continuous and discrete variable optimization. In: Proc. 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, New York, aIAA-2004-4458
- Xu S, Grandhi RV (1998) Effective two-point function approximation for design optimization. *AIAA J* 36:2269–2275
- Zhang WH, Fleury C (1997) A modification of convex approximation methods for structural optimization. *Comput Struct* 64:89–95
- Zhu C, Byrd RH, Lu P, Nocedal J (1994) L-bfgs-b: Fortran subroutines for large scale bound constrained optimization. Tech. Rep. Report NAM-11, Northwestern University, EECS Department
- Zillober C (1993) A globally convergent version of the method of moving asymptotes. *Struct Optim* 6:166–174
- Zillober C (2001) A combined convex approximation - interior point approach for large scale nonlinear programming. *Optim Eng* 2:51–73
- Zillober C (2002) SCIP - an efficient software tool for the solution of structural optimization problems. *Struct Multidisc Optim* 24:362–371
- Zillober C, Schittkowski K, Moritzen K (2004) Very large scale optimization by sequential convex programming. *Optim Methods Softw* 19:103–120