SOLUTION OF LARGE SCALE OPTIMIZATION PROBLEMS WITH SEQUENTIAL CONVEX PROGRAMMING

Pierre DUYSINX, Michael BRUYNEEL, and Claude FLEURY

LTAS- Dept of Aerospace and Mechanical Engineering
Institute of Mechanics and Civil Engineering
University of Liège,
Chemin des Chevreuils 1, Building B52
B-4000 Liège, Belgium
tel +32.4.3669194, fax +32.4.3669505,
P.Duysinx@ulg.ac.be

1 INTRODUCTION

Structural optimization consists in formulating the design problem of structural components as an optimization problem in order to take advantage of mathematical programming tools.

From a mathematical point of view, a quite general statement of the optimization problem is given as following

$$(P) \quad \begin{cases} \min\limits_{\mathbf{x}} & g_0(\mathbf{x}) & \\ \text{s.t.:} & g_j(\mathbf{x}) \leq \bar{g}_j & j = 1 \dots m \\ & \underline{x}_i \leq x_i \leq \overline{x}_i & i = 1 \dots n \end{cases} \tag{1}$$

The function g_0 is the objective function of the problem, i.e. a cost function or a performance index that has to be minimized in order to have a better design. In topology optimization, this is for instance the compliance of the structure under the considered load case.

The set of constraint functions g_j (in number m) expresses the restrictions the design is subject to in order to be *feasible*. For example these functions are some bounds upon a stress measure to have resistance, restricted displacements, a volume resource or perimeter bound...

The n variables x_i are the design variables of the problem, that is, the parameters, which can be modified, to improve the design. In the topology optimization context, the x_i variables are the element densities and the orientation parameters of the local microstructure. In optimization of composite structures, the x_i variables may be the ply thicknesses

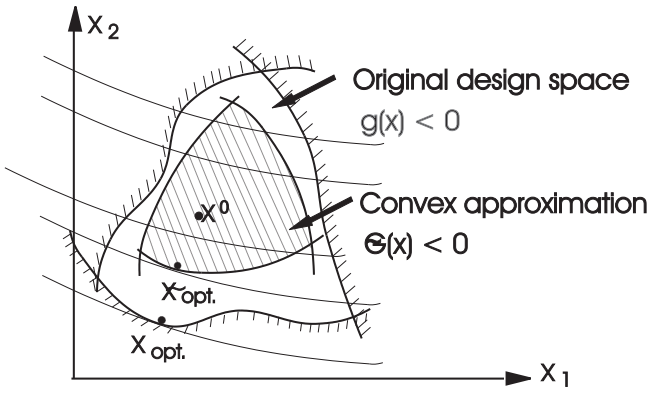

Figure 1: Approximation concept

and orientations. The design variables are also subject to some very simple restrictions, because of some physical or mathematical reasons (the density lies between 0 and 1, and the orientation can be searched between 0 and π because of periodicity reasons). Other constraints on design variables can come from technological or manufacturing arguments like prescribed density regions in topology optimization. These special constraints on design variables are called *side constraints*. They generally call for a special treatment in the algorithm because of their very simple structure.

The direct solution of optimization problem (P) is totally prohibitive in structural optimization, and furthermore in topology optimization, because of the computational cost of the structural and sensitivity analysis of the problem. Indeed the problem of structural optimization is highly non-linear, and implicit in terms of the design variables so that each function evaluation would require a Finite Element (F.E.) analysis.

As soon as the seventies, Schmidt and his co-workers (e.g. Schmidt and Farshi [40], Schmidt and Fleury [41]) proposed an interesting way to circumvent the problem while using mathematical programming tools. The *approximation concepts approach* replaces the primary optimization problem (P) with a sequence of explicit approximate sub-problems having a simple algebraic structure, and built from the available information (function values, first and second order derivatives) at the current design point or at the former iteration points.

$$(\tilde{P}) \quad \left\{ \begin{array}{lll} \min\limits_{\mathbf{x}} & \tilde{g_0}(\mathbf{x}) & \\ \text{s.t.:} & \tilde{g_j}(\mathbf{x}) \ \leq \ \bar{g}_j & j = 1 \dots m \\ & \underline{x}_i \ \leq \ x_i \ \leq \overline{x}_i & j = 1 \dots n \end{array} \right. \tag{2}$$

where $\tilde{g_0}$ and $\tilde{g_j}$ denote respectively the approximations of objective function g_0 and of the constraint functions g_j. These approximations can be regarded as some kind of Taylor expansions of the response functions around the current design point $\mathbf{x}^{(k)}$. Different

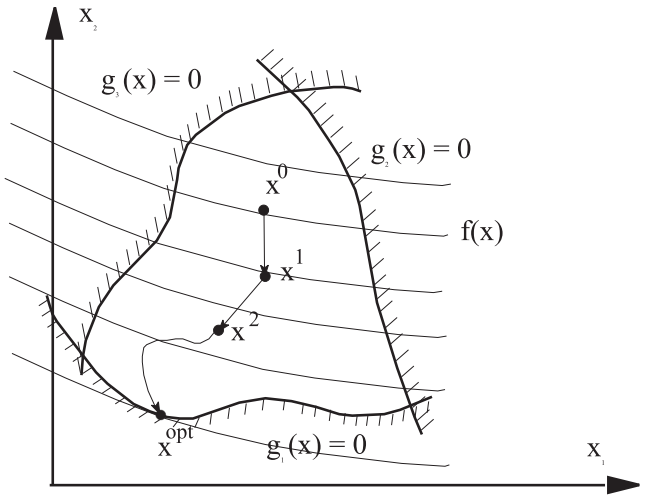

Figure 2: Iterative scheme with SCP approach

techniques have been proposed, but the most famous ones remain CONLIN by Fleury and Braibant [27] and MMA (Method of Moving Asymptotes) by Svanberg [44]

However the success of the approximation strategy comes from the fact that the sub-problems ($\tilde{P}$) can be solved efficiently with relevant mathematical programming algorithms. Up to now the most efficient strategy is the dual method proposed initially by Fleury [21] but used by many other authors like Svanberg [44]. Dual methods are well adapted to structural problems, because the dimensionality of the dual solution space is generally much lower than the primal design space. With efficient algorithms, dual solvers are able to solve sub-problems within a reasonable computational time. For sizing and shape optimization, solution time is less than 1 percent of the F.E. computation time. When dealing with compliance topology problems, the same character is preserved.

To summarize, the solution strategy of optimization problems is based on the following steps:

1. For the current design characterized by the design variables $\mathbf{x^{(k)}}$, perform a Finite Element analysis and its sensitivity analysis.

2. From the results of the current structural analysis, generate an approximate ($\tilde{P}^{(k)}$).

3. Solve the sub-problem ($\tilde{P}^{(k)}$) with an efficient solver, like a dual solver.

4. Adopt the solution of the approximate sub-problem $\mathbf{x}^\star$ as a the new design $\mathbf{x^{(k+1)}}$ and go back to 1 until convergence (convergence critrion).

The strategy combines both concepts of approximation and dual solution and it is now generally known as the *sequential convex programming approach* (SCP) as suggested by Fleury in Ref. [25, 26].

The SCP approach have demonstrated its efficiency in academic and industrial applications. One can come to a nearly stationary solution generally within 20 iteration steps, independently of the number of design variables. For topology optimization convergence speed can be a bit slower, because of the higher complexity of the problem; it is not rare to have to wait for 50 or 100 iterations for having a stable solution.

Before studying the numerical solution techniques into details, it is important to remind the main characteristics of optimal material distribution problems in order to be able to select the most suited algorithms to solve these problems. Topology optimization problems are *large scale optimization problems*. It results that extension of traditional SCP methods must be considered carefully.

- The number of design variables is large or very large, i. e. in usual topology problems, one has to consider from 1.000 up to 1.000.000 density variables nowadays in industrial applications. In large composite structures, the number of design variables is proportional to the number of plies times the number of pannels and generally rises several hundreds of variables.

- The number of constraints is generally quite small when the problem formulation of topology relies on *global constraints*. For example, in compliance type problems, one has to take into account as many compliance responses as the number of load cases plus volume and perimeter constraints. This means in turn that one has to consider around 10 restrictions. The number of restrictions being much smaller than the number of design variables, we are in a favorable situation for applying dual algorithms.

- This situation is totally different when considering *local constraints* like stress constraints in the design problem. Indeed one has to cope with one constraint per finite element, which means a number of constraint that is of the same order of magnitude as the number of design variables. This could also be the case if slope constraints are included in the design as in Sigmund and Petersson [35]. For composite structures, the number of restrictions is generally rather large, because one failure criterion has to be considered in each ply. Further more local buckling restrictions are also usually considered in each panel so that the number of restrictions is generally over several thousand. Thus in these cases the problem is a large scale problem both in the design variable space *and* in the dual space. Under these conditions, advantages of dual algorithms are lower, but work done in Ref. [16] showed that dual solvers are still able to produce solutions within a computation time that is of the same order of magnitude as the F.E. analysis even when there are around a thousand of constraints.

There is also another characteristic related to stress constraints in topology optimization, which complicates the task of the optimizer: the singularity phenomenon of stress constraints. This difficulty is alleviated by using a perturbation technique (ϵ-relaxation technique as proposed by Cheng and Guo [11]). So to consider stress constraints, one has to tailor a additional strategy to manage the extra perturbation parameter.

This lecture is devoted to explain the basic elements of SCP method and in particular its application to topology and composite structure optimization problems, which are very large scale problems. Despite the high level of integration of the different concepts in available implementations of SCP, this study will still distinguish in this presentation the two main concepts as pointed out by [25, 26]:

- The *solution aspect* based on dual methods, which will be explored in section 2.
- The *approximation concept* and different approximation schemes that will be reviewed in section 3.

Finally as our goal is to apply Sequential Convex programming to large scale problems such as topology optimization and composite structures, different issues particular to topology problems will be reviewed in section 4 and 5. At first, we will see how an efficient approximation can be built for the perimeter constraint (section 4). Then we will have a careful study of the management of the ϵ-relaxation technique of stress constraints (section 5).

2 DUAL SOLUTION ALGORITHMS

2.1 Lagrange function

Suppose that we have to solve the following non linear optimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g_j(\mathbf{x}) \leq 0 \qquad j = 1, \ldots, m \end{array} \tag{3}$$

where functions $f(\mathbf{x})$ and $g_j(\mathbf{x})$ are assumed to be continuous and differentiable.

It is classic to define the Lagrange function associated to problem (3):

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x}) \tag{4}$$

where the $\lambda_j \geq 0$ are Lagrange multipliers associated to each constraint g_j (and so in number m as the constraints). The new variables λ_j are generally called *dual variables* since there is a one-to-one association with the design constraints. Conversely the original design variables $\mathbf{x}$ of the problem are said *primal variables* of the problem.

One can remark that the Lagrange transformation replaces the constraints $g_j(\mathbf{x}) \leq 0$ by a linear term $\lambda_j \, g_j(\mathbf{x})$ in the objective function. This can be interpreted as adding to objective function $f(\mathbf{x})$ a linear cost, with marginal price λ_j, which has to be paid whenever the constraint is violated.

Lagrange function transforms the optimization constrained problem into an unconstrained problem. The objective function of this new optimization problem is precisely the Lagrangian function $L(\mathbf{x}, \lambda)$ and the design variables are both $\mathbf{x}$ and λ variables.

$$\min_{\mathbf{x}} \max_{\lambda \geq 0} \quad L(\mathbf{x}, \lambda) \tag{5}$$

Maximization over λ affects an infinite cost (penalty) to Lagrange function when constraints are violated $g_j(\mathbf{x}) > 0$. The price of this transformation is that the dimension of the optimization growths from n variables to $n + m$ ones.

2.2 Karush-Kuhn-Tucker conditions

Theorem 1 *Necessary conditions of optimality of contrained problems.*
If $x^\star$ is an optimum of problem (3), and if $x^\star$ is a regular point, Then one can find a vector of Langrange multipliers $\lambda^\star = (\lambda_1^\star, \ldots, \lambda_m^\star)$ such that

$$\frac{\partial f(\mathbf{x}^*)}{\partial x_i} + \sum_{j=1}^{m} \lambda_j^* \frac{\partial g_j(\mathbf{x}^*)}{\partial x_i} \;=\; 0 \quad \forall\, i \tag{6}$$

$$g_j(\mathbf{x}^*) \leq 0 \tag{7}$$

$$\lambda_j^* \geq 0 \tag{8}$$

$$\lambda_j^*\; g_j(\mathbf{x}^*) = 0 \quad \forall\, j \tag{9}$$

The Karush-Kuhn-Tucker (KKT) conditions stated as below consist of four types of conditions, namely

- Stationarity of the Lagrange function $L(\mathbf{x}, \lambda)$ with respect to $\mathbf{x}$ (6),
- Primal feasibility, which means that $\mathbf{x}^\star$ is a feasible point (7),
- Dual feasibility, which means that the Lagrange multipliers $\lambda^\star$ are non negative (8),
- Complementary slackness, which means that the Lagrange multipliers corresponding to inactive constraints are zero (9).

Remark 1 *A point $\mathbf{x}^\star$ is a regular point of the problem, if all the gradient vectors ∇g_j of active constraints (i.e. $g_j(\mathbf{x}^\star) = 0$) are linearly independent.*

2.3 Introduction to duality

As noticed above, the first KKT condition (6) implies the solution of the system:

$$\nabla_x\, L(\mathbf{x}, \lambda^\star) \;=\; 0 \tag{10}$$

This condition is equivalent to say that $\mathbf{x}^\star$ is the solution of the minimization problem:

$$\min_{\mathbf{x}}\; L(\mathbf{x}, \lambda^\star)$$

Imagine now that, for any Lagrange vector, i.e.

$$\lambda = (\lambda_1, \ldots, \lambda_m) \quad \text{such that} \quad \lambda_j \geq 0 \quad j = 1 \ldots m$$

one can find the solution of the minimization problem (called *Lagrangian problem*)

$$\min_{\mathbf{x}} \; L(\mathbf{x}, \lambda) \tag{11}$$

This gives rise to a functional relationship of dependency of primal variables $\mathbf{x}$ in terms of the new dual variables λ.

$$\mathbf{x} \;=\; \mathbf{x}(\lambda) \;=\; \arg\min_{\mathbf{x}} \; L(\mathbf{x}, \lambda) \tag{12}$$

Substituting primal variables in terms of function (12), it is possible to rewrite Lagrange function $L(\mathbf{x}, \lambda)$ in terms of dual variables λ solely

$$\begin{aligned} \ell(\lambda) &= L(\mathbf{x}(\lambda), \lambda) \\ &= f(\mathbf{x}(\lambda)) + \sum_{j=1}^{m} \lambda_j \, g_j(\mathbf{x}(\lambda)) \end{aligned} \tag{13}$$

This function $\ell(\lambda)$ is called *dual function* of the problem.

The optimization problem that is related to dual function is a maximization problem, called the *dual problem* (while the original problem is called *primal problem*):

$$\begin{aligned} \max_{\lambda_j} \quad & \ell(\lambda) \\ \text{s.t.} \quad & \lambda_j \;\geq 0 \qquad j = 1, \ldots, m \end{aligned} \tag{14}$$

The solution of the dual problem determines the optimal Lagrange multipliers $\lambda^\star$, which satisfy KKT conditions. One can show that dual problem has the following properties:

- If the primal problem is a minimization problem, the dual problem is a maximization problem;
- The dual problem possesses a solution if the primal problem does;
- A solution of the dual problem also provides a solution to the primal problem.

Remark 2 *When defining the Lagrangian problem (11), we have implicitly assumed that we were allowed to switch the* $\max_{\lambda_j}$ *and* $\min_{\mathbf{x}}$ *operations in original problem (5) , which is not trivial. In fact this can be mathematically correct under strict mathematical assumptions such as a convexity. This will be clarified later in the lesson.*

2.4 Weak duality

With very little assumptions on f and g_j, one can state the most general definition of the dual function as follows:

$$\ell(\lambda) \;=\; \inf_{\mathbf{x}} \quad L(\mathbf{x}, \lambda) \tag{15}$$

So a dual function exists even for non convex problems or discrete valued design variable problems, etc.

With very weak assumptions on f and g_j, one can prove that $\ell(\lambda)$ is *concave*. However, the dual function is *non-smooth*, so that derivatives must be replaced by sub-gradients.

The *dual problem* is defined as the maximization problem

$$\begin{array}{ll} \max\limits_{\lambda_j} & \ell(\lambda) \\ \text{s.t.} & \lambda_j \ \geq 0 \qquad j = 1, \ldots, m \end{array} \tag{16}$$

Since the dual function $\ell(\lambda)$ is a concave function of λ, it is possible to state that every local optimum of $\ell(\lambda)$ is a global optimum. Therefore the dual problem will in general be easier to solve than the primal problem (for example when dealing with integer programming). This is one of the reasons why the concept of duality has been found to be very useful in mathematical programming. However because of the non-smoothness of the dual function, one has to resort to non smooth maximization algorithms, which are generally less efficient and more complex.

General properties of dual function

For any feasible $\mathbf{x}$ and for any Lagrange vector λ, there holds:

$$f(\mathbf{x}) \ \geq \ \ell(\lambda) \tag{17}$$

Going along, if $\lambda^\star$ is the solution of dual problem, and if $\mathbf{x}^\star$ is the optimum of the primal problem, one has

$$f(\mathbf{x}^\star) \ \geq \ \ell(\lambda^\star) \tag{18}$$

This means that the dual function is always a lower bound of primal objective function value. The gap $G = f(\mathbf{x}^\star) - \ell(\lambda^\star)$ is called *duality gap*. In addition, the primal point that is associated to the optimal Lagrange multiplier through the Lagragian problem

$$\arg \inf_{\mathbf{x}} \quad L(\mathbf{x}, \lambda^\star) \tag{19}$$

may not be realized or may not be a feasible point.

2.5 Strong duality

Now suppose that we want to solve the *convex problem*:

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in X} & f(\mathbf{x}) \\ \text{s.t.} & g_j(\mathbf{x}) \ \leq 0 \qquad j = 1, \ldots, m \end{array} \tag{20}$$

where functions $f(\mathbf{x})$ and $g_j(\mathbf{x})$ are assumed to be C^1 (i.e. continuous and differentiable), and *convex*. The set X of feasible design variables is also *convex*.

The set X is usually made of side constraints on design variables, which is obviously a convex set (box constraints):

$$X \;=\; \{x_i \quad | \quad \underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i \quad i = 1 \ldots n\} \tag{21}$$

In addition, it is assumed that the Slater condition (or qualification of the constraint) is fulfilled:

$$\exists\, \tilde{\mathbf{x}} \;:\; g_j(\tilde{\mathbf{x}}) \;<\; 0 \;\forall\, j \tag{22}$$

Theorem 2 *If the problem (20) is a convex problem and if the Slater condition is satisfied, then, there is at least one Lagrange multiplier vector $\lambda^\star$ such that*

$$f(\mathbf{x}^\star) \;=\; \min_{\mathbf{x}} \; L(\mathbf{x}, \lambda^\star) \tag{23}$$

and there is no duality gap

$$f(\mathbf{x}^\star) \;=\; \ell(\lambda^\star) \tag{24}$$

Corollary 1 *Necessary and sufficient conditions of optimality:*
If the problem (20) is convex, if functions $f(\mathbf{x})$ and $g_j(\mathbf{x})$ are differentiable and if the Slater condition is satisfied, then $\mathbf{x}^\star$ is the primal optimal point and $\lambda^\star$ is the optimal dual point if and only if Karush-Kuhn-Tucker conditions are satisfied.

Corollary 2 *If the problem (20) is convex and if the Slater condition is satisfied, solving the primal problem (20) is completely equivalent to solve the dual problem:*

$$\begin{array}{lll} \min\limits_{\lambda_j} & \ell(\lambda) & \\ s.t. & \lambda_j \;\geq 0 & \quad j = 1, \ldots, m \end{array} \tag{25}$$

with the dual function

$$\ell(\lambda) \;=\; \min_{\mathbf{x} \in X} \quad L(\mathbf{x}, \lambda) \tag{26}$$

This is a consequence of the definition of the Lagrange function. If $f(\mathbf{x})$ and $g_j(\mathbf{x})$ are C^1 and convex functions, the Lagrange function $L(\mathbf{x}, \lambda)$ has a saddle point in $(\mathbf{x}^\star,\ \lambda^\star)$, which means that:

$$L(\mathbf{x}^\star, \lambda) \;\leq\; L(\mathbf{x}^\star, \lambda^\star) \;\leq\; L(\mathbf{x}, \lambda^\star) \tag{27}$$

Thence it is possible to show the equivalence between the following optimization problems:

$$\min_{\mathbf{x} \in X} \; \max_{\lambda \geq 0} \; L(\mathbf{x}, \lambda) \qquad \Leftrightarrow \qquad \max_{\lambda \geq 0} \; \min_{\mathbf{x} \in X} \; L(\mathbf{x}, \lambda) \tag{28}$$

In other words it is possible to switch the $\min_{\mathbf{x} \in X}$ and the $\max_{\lambda \geq 0}$. As the left hand side problem is an equivalent to the primal problem, while the right problem is the dual maximization, one gets the equivalence between primal and dual problems.

2.6 Dual function properties in strong duality [25]

Concavity

Dual function $\ell(\lambda)$ is concave.

Lower bound and duality gap

Generally speaking, the dual function is a lower bound of the primal function, but for convex problems, there is no duality gap:

$$\ell(\lambda^\star) \;=\; f(\mathbf{x}^\star)$$

This means that the dual solution is equivalent to the primal solution.

Gradient of dual function

The first derivatives of dual function with respect to a Lagrange multiplier is simply the constraint value associated to that multiplier λ_k in point $\mathbf{x}(\lambda)$:

$$\frac{\partial \ell(\lambda)}{\partial \lambda_k} \;=\; g_k(\mathbf{x}(\lambda)) \tag{29}$$

The proof of this formula can be obtained easily when deriving the dual function:

$$\begin{aligned}
\frac{\partial \ell(\lambda)}{\partial \lambda_k} &= \frac{\partial}{\partial \lambda_k} \left\{ f(\mathbf{x}(\lambda)) + \sum_{j=1}^{m} \lambda_j \, g_j(\mathbf{x}(\lambda)) \right\} \\
&= \sum_{i=1}^{n} \left[\frac{\partial f(\mathbf{x}(\lambda))}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j(\mathbf{x}(\lambda))}{\partial x_i} \right] \frac{\partial x_i}{\partial \lambda_k} \;+\; g_k(\mathbf{x}(\lambda))
\end{aligned}$$

The first contribution vanishes because it is the optimality conditions of the Lagrangian problem leading to definition of $\mathbf{x}(\lambda)$.

From result (29), its comes that the evaluation of the gradient of the dual function is very easy to calculate.

Optimality conditions of dual problem

It comes from the previous property (29) that the optimality conditions of the quasi-unconstrained dual problem can be written as follows:

$$\frac{\partial \ell(\lambda)}{\partial \lambda_k} \;=\; 0 \quad \text{if} \quad \lambda_k \;>\; 0 \tag{30}$$

$$\frac{\partial \ell(\lambda)}{\partial \lambda_k} \;<\; 0 \quad \text{if} \quad \lambda_k \;=\; 0 \tag{31}$$

A first algorithm for dual maximization

These conditions (30-31) indicate that the maximum of the dual function is attained when the constraints are satisfied exactly (as equalities) for $\lambda_j > 0$ and as inequalities if $\lambda_j = 0$.

Going a bit further, we can propose a first algorithm based on the steepest ascent method to maximize the dual function:

$$\lambda^+ \;=\; \lambda \;+\; \alpha\,\nabla\ell \tag{32}$$

or

$$\lambda_j^+ \;=\; \lambda_j \;+\; \alpha\,g_j(\mathbf{x}(\lambda)) \tag{33}$$

where α is a step size. Thus a dual variables λ_j increases if the corresponding constraint g_j is violated, whereas it decreases (possibly reaching zero) if g_j is negative (and satisfied). From these considerations, it results in an intuitive interpretation of the dual method approach: the approach attempts to satisfy the inequality constraints by adjusting the values of the dual variables.

Since non negativity constraints are very easy to take into account, classical algorithms for unconstrained maximization can be readily adapted to solve quasi-unconstrained dual problems. In particular, the conjugate gradient method is well suited since the computation of the gradient is straight forward: it needs solely computing the constraint values $g_j(\mathbf{x}(\lambda))$.

Separable case

A central issue of the dual approach is to compute the relationships between primal and dual methods $\mathbf{x}(\lambda)$:

$$\mathbf{x}(\lambda) \;=\; \arg\min_{\mathbf{x}\,\in\,X}\; L(\mathbf{x},\lambda) \tag{34}$$

Solving the minimization problem has to be repeated a lot of times, and this might lead to a prohibitive computational cost. However if the problem has some special structures, as separable problems, this is not cumbersome. In addition to convexity, separability is an essential property for dual formulation to be efficient.

A function $f(\mathbf{x})$ is said to be separable if the function be written as a sum of functions $f_i(x_i)$, which depends only on the single variable x_i.

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{n}\; f_i(x_i) \tag{35}$$

Separable functions benefit from some computationally important properties. In particular, the Hessian matrix of such functions is diagonal.

The constrained problem (20) is a separable programming problem if every functions $f(\mathbf{x})$, $g_j(\mathbf{x})$ are themselves separable. This implies that the Lagrangian function $L(\mathbf{x},\lambda)$ is separable too.

$$L(\mathbf{x},\lambda) \;=\; \sum_{i=1}^{n}\; L_i(x_i,\lambda)$$

As a result, the n-dimensional Lagrangian problem can be broken up into 'n' one-dimensional Lagrangian problems

$$\min_{\mathbf{x}\,\in\,X}\; L(\mathbf{x},\lambda) \;=\; \sum_{i=1}^{n}\; \min_{x_i\,\in\,X_i} L_i(x_i,\lambda) \tag{36}$$

where each one-dimensional problem can be solved separately. Thus the dual function can be written as follows:

$$\ell(\lambda) \;=\; \sum_{i=1}^{n} \ell_i(\lambda) \;=\; \sum_{i=1}^{n} \{ \min_{x_i \,\in\, X_i} L_i(x_i, \lambda) \} \tag{37}$$

In may cases the single-variable minimization problem has a simple algebraic structure and it can be solved in closed-form, thus yielding an explicit dual function.

Second order derivatives of dual function

To get second order derivatives of dual function, one derives a second time the expression of first order derivatives given in (29), which gives:

$$\frac{\partial^2 \ell(\lambda)}{\partial \lambda_k \partial \lambda_l} \;=\; \frac{\partial}{\partial \lambda_l}\, g_k(\mathbf{x}(\lambda)) \;=\; \sum_{i=1}^{n} \frac{\partial g_k}{\partial x_i}\, \frac{\partial x_i(\lambda)}{\partial \lambda_l} \tag{38}$$

One remarks immediately that the second order derivatives can be *discontinuous* because of the presence of term $\frac{\partial x_i(\lambda)}{\partial \lambda_l}$. Indeed, when side constraints are taken into account in Lagrangian problem (11) in order to be treated explicitly, the function relationship $\mathbf{x}(\lambda)$ can be first order discontinuous. This is explained in more details in the following.

Evaluation of this term $\frac{\partial x_i(\lambda)}{\partial \lambda_l}$ can be made from KKT conditions. Deriving the KKT conditions (6) with respect to λ_l gives:

$$\sum_{k=1}^{n} \left[\frac{\partial^2 f}{\partial x_i \partial x_k} + \sum_{j=1}^{m} \lambda_j \frac{\partial^2 g_j}{\partial x_i \partial x_k} \right] \frac{\partial x_k(\lambda)}{\partial \lambda_l} \;+\; \frac{\partial g_l(\lambda)}{\partial x_i} \;=\; 0$$

Going to matrix notations, one recognizes the Hessian matrix of the Lagrange function and gradient vector of constraint g_l with respect to primal variables:

$$\nabla^2_{xx}\, L(\mathbf{x}, \lambda)\, \frac{\partial \mathbf{x}(\lambda)}{\partial \lambda_l} \;+\; \nabla_x\, g_l \;=\; 0$$

If side-constraints are treated explicitly, one has to restrict this relation to free variables (variables, which are not fixed at their lower or upper bounds). So one has to consider the matrix $\mathbf{G}$ instead of $\nabla^2_{xx}\, L$. The matrix $\mathbf{G}$ is obtained from $\nabla^2_{xx}\, L$ by deleting rows and columns corresponding to fixed variables. One gets now:

$$\mathbf{G}\, \frac{\partial \mathbf{x}(\lambda)}{\partial \lambda_l} \;+\; \nabla_x g_l \;=\; 0 \tag{39}$$

If one also notes by $\mathbf{N}$ the matrix whose columns are made with the gradient vectors of all constraints

$$\mathbf{N} \;=\; [\nabla g_1 \ldots \nabla g_m]$$

and if one introduces the result (39) into (38), one gets the final expression of Hessian of dual function:

$$\left[\frac{\partial^2 \ell(\lambda)}{\partial \lambda_k \partial \lambda_l} \right] \;=\; -\, \mathbf{N}^T\, \mathbf{G}^{-1}\, \mathbf{N} \tag{40}$$

We must repeat that this Hessian is only continuous by pieces. Discontinuity of dual Hessian occurs along hyper-planes of equations

$$x_i(\lambda) \;=\; \underline{x}_i \quad \text{and} \quad x_i(\lambda) \;=\; \overline{x}_i \tag{41}$$

This has a major impact on dual maximization, since the greatest care has to given to that property to built second order algorithms for dual maximization. Classical Newton or quasi-Newton methods can only be applied in subregions where the set of free and fixed design variables is frozen. Building an efficient strategy to find the optimal set of free and fixed design variables is one of the major issues of dual optimizer construction.

2.7 Application to quadratic problems with linear constraints

We illustrate the dual approach on the quadratic problem with linear inequality constraints. The problem statement is the following:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \dfrac{1}{2}\,\mathbf{x}^T\,\mathbf{x} \\ \text{s.t.} & \mathbf{C}^T\,\mathbf{x} \;\geq\; \mathbf{d} \end{array}$$

where $\mathbf{C}$ is a $n \times m$ matrix of the constraint gradients (which are assumed to be linearly dependent). The Lagrangian function of the problem is:

$$L(\mathbf{x}, \lambda) \;=\; \frac{1}{2}\,\mathbf{x}^T\,\mathbf{x} \;-\; \lambda^T\,(\mathbf{C}^T\mathbf{x} \;-\; \mathbf{d})$$

Optimality condition (KKT conditions) are:

$$\begin{aligned} &\nabla_x L \;= \mathbf{x} \;-\; \mathbf{C}\,\lambda \;=\; 0 \\ &-\,\mathbf{C}^T\,\mathbf{x} \;+\; \mathbf{d} \;\leq\; 0 \\ &\lambda \;\geq\; 0 \\ &\lambda^T\,(\mathbf{C}^T\,\mathbf{x} \;-\; \mathbf{d}) \;=\; 0 \end{aligned}$$

The first condition is the solution of the Lagrangian problem, which leads to the relationship between primal and dual variables:

$$\mathbf{x}(\lambda) \;=\; \mathbf{C}\,\lambda$$

This relation is *fully explicit*.

Replacing primal variables into the Lagrangian function, one gets the dual function:

$$\begin{aligned} \ell(\lambda) \;&=\; \frac{1}{2}\,\mathbf{x}(\lambda)^T\,\mathbf{x}(\lambda) \;-\; \lambda^T\,(\mathbf{C}^T\mathbf{x}(\lambda) \;-\; \mathbf{d}) \\ &=\; \frac{1}{2}\lambda^T\mathbf{C}^T\mathbf{C}\lambda \;-\; \lambda^T\,(\mathbf{C}^T\mathbf{C}\lambda \;-\; \mathbf{d}) \\ &=\; -\frac{1}{2}\lambda^T\mathbf{C}^T\mathbf{C}\lambda \;+\; \lambda^T\,\mathbf{d} \end{aligned}$$

The gradient of the dual function is given by:

$$\begin{aligned} \nabla \ell(\lambda) &= -\mathbf{C}^T \mathbf{C} \lambda + \mathbf{d} \\ &= \mathbf{d} - \mathbf{C}^T \mathbf{x}(\lambda) \end{aligned}$$

which is the value of primal constraints, in agreement with the general theory.

From this expression, one gets also easily the Hessian matrix of the dual function:

$$\nabla^2 \ell(\lambda) = -\mathbf{C}^T \mathbf{C}$$

This Hessian matrix is constant. Thus dual function of a quadratic problem is a quadratic function. It has the fully explicit form:

$$\ell(\lambda) = -\frac{1}{2} \lambda^T \mathbf{A} \lambda + \lambda^T \mathbf{d}$$

where $\mathbf{A} = \mathbf{C}^T \mathbf{C}$ denotes the negative of the dual Hessian matrix.

In the case of equality constraints, dual variables are unrestricted in sign. Therefore the maximum of dual function can simply be obtained by stating that its gradient must vanish:

$$\nabla \ell(\lambda) = -\mathbf{A} \lambda + \mathbf{d} = 0$$

This leads to the solution:

$$\lambda^\star = \mathbf{A}^{-1} \mathbf{d}$$

From the dual solution, one recovers the optimal primal solution with the help of the primal-dual relationships:

$$\mathbf{x}^\star = \mathbf{C} \lambda^\star$$

In the case of inequality constraints, the dual variables are subject to non negativity side constraints, and one has to solve dual maximization problem:

$$\begin{aligned} \max_{\lambda} \quad & -\frac{1}{2} \lambda^T \mathbf{A} \lambda + \lambda^T \mathbf{d} \\ \text{s.t.} \quad & \lambda_j \geq 0 \end{aligned}$$

After solving this quasi-unconstrained maximization problem in the dual space, one recovers the primal solution from the same primal-dual relationships as before:

$$\mathbf{x}^\star = \mathbf{C} \lambda^\star$$

Dual maximization is the most natural and rigorous way to select automatically the optimum set of Lagrange multipliers. In addition, as the number of dual variables is often smaller than the number of primal variables, the dimensionality of the optimization problem we actually solve is smaller.

Numerical example of a quadratic separable problem

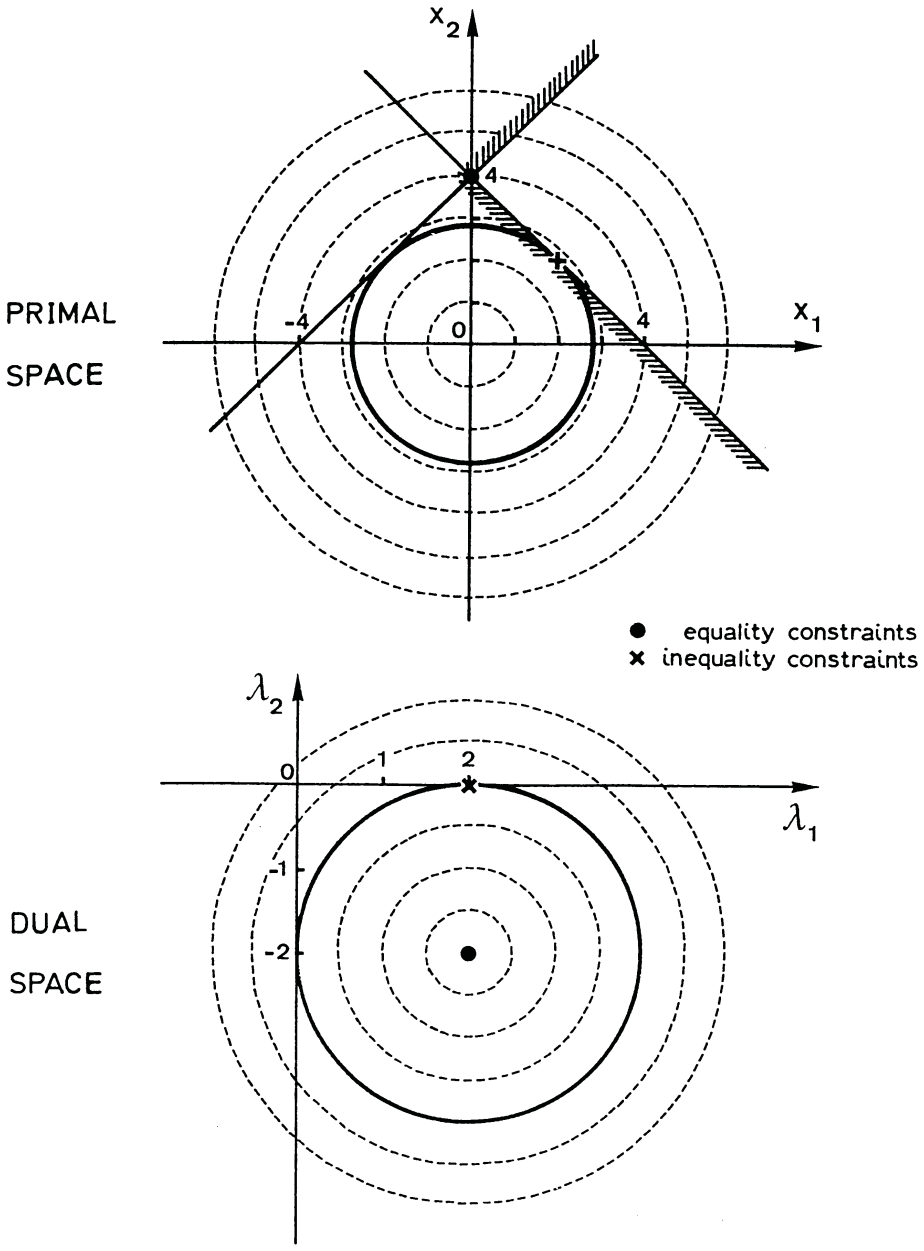

Figure 3: Numerical application of a quadratic separable problem [25]

$$\begin{array}{ll} \min\limits_{x_1\,,\,x_2} & \frac{1}{2}\,x_1^2 \,+\, \frac{1}{2}\,x_2^2 \\ \text{s.t.} & x_1 \,+\, x_2 \,\geq\, 4 \\ & x_1 \,-\, x_2 \,\geq\, -4 \end{array}$$

From the previous definitions, one can identify the following matrix and vector:

$$\mathbf{C} \,=\, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{d} \,=\, \begin{bmatrix} 4 \\ -4 \end{bmatrix}$$

The Lagrange function writes:

$$L(\mathbf{x}, \lambda) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 - \lambda_1(x_1 + x_2 - 4) - \lambda_2(x_1 - x_2 + 4)$$

The optimality conditions of the Lagrange function with respect to the primal variables give rise to the primal-dual relationships:

$$\begin{aligned} x_1(\lambda_1, \lambda_2) &= \lambda_1 + \lambda_2 \\ x_2(\lambda_1, \lambda_2) &= \lambda_1 - \lambda_2 \end{aligned}$$

The dual function then takes an explicit expression:

$$\ell(\lambda_1, \lambda_2) = -(\lambda_1 - 2)^2 - (\lambda_2 + 2)^2 + 8$$

and the dual problem is:

$$\begin{array}{ll} \max\limits_{\lambda_1, \lambda_2} & \ell(\lambda_1, \lambda_2) = -(\lambda_1 - 2)^2 - (\lambda_2 + 2)^2 + 8 \\ \text{s.t.} & \lambda_1 \geq 0 \\ & \lambda_2 \geq 0 \end{array}$$

The first order partial derivatives of the dual function are given by:

$$\begin{aligned} \frac{\partial \ell}{\partial \lambda_1} &= -2\lambda_1 + 4 = 4 - x_1 - x_2 \\ \frac{\partial \ell}{\partial \lambda_2} &= -2\lambda_2 - 4 = -4 - x_1 + x_2 \end{aligned}$$

Thus one can verify that the partial first order derivatives are given by minus the value of the associated primal constraints.

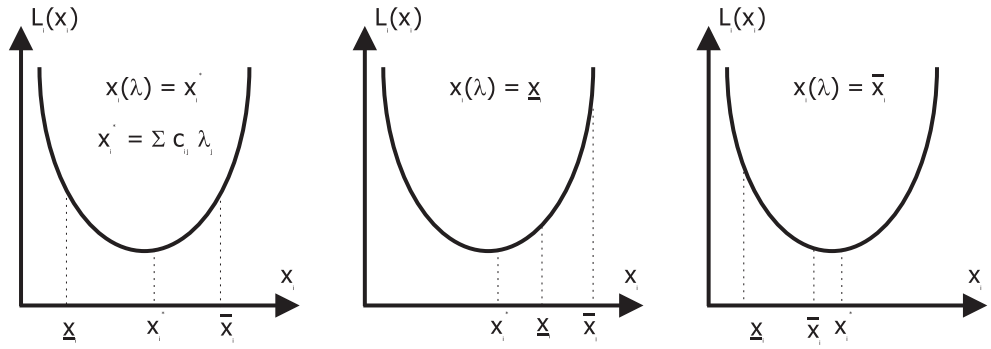

Figure 4: Illustration of Lagrangian problem with side constraints [25]

2.8 Treatment of side constraints

Let us consider the following separable, quadratic problem, with linear inequality and side-constraints:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2}\sum_{i=1}^{n} x_i^2 \\ \text{s.t.} \quad & \sum_{i=1}^{n} c_{ij}\, x_i \;\geq\; d_j \quad j = 1 \ldots m \\ & \underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i \quad i = 1 \ldots n \end{aligned} \tag{42}$$

The side constraints, which impose lower and upper bounds on the design variables can, of course, be considered as linear inequality constraints. However this would increase dramatically the number of dual variables, and thus this would reduce the potential advantage of dual method by increasing the dimensionality of dual workspace. Therefore, the efficiency calls for a particular and separate treatment of these very simple constraints, apart from the general constraints.

The Lagragian problem is

$$\min_{\underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i} \quad \frac{1}{2}\sum_{i=1}^{n} x_i^2 \;-\; \sum_{j=1}^{m} \lambda_j \,(\sum_{i=1}^{n} c_{ij}\, x_i \;-\; d_j) \tag{43}$$

Because of the separability property, the n-dimensional problem can be split into n one-dimensional problems relative to each variable "i":

$$\min_{\underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i} \quad L_i(x_i, \lambda) \;=\; \frac{1}{2}\, x_i^2 \;-\; (\sum_{j=1}^{m} \lambda_j\, c_{ij})\; x_i \tag{44}$$

The solutions of these problems are obtained by expressing the optimality conditions,

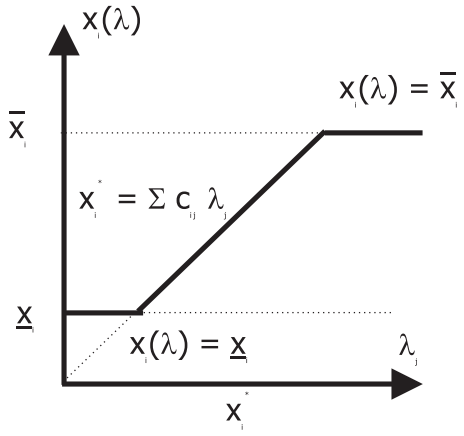

Figure 5: Primal-dual relations when side-constraints are taken into account

i.e. vanishing the first order derivative with respect to x_i:

$$\frac{\partial L_i}{\partial x_i} \;=\; x_i - (\sum_{j=1}^{m} \lambda_j \, c_{ij}) \;=\; 0 \tag{45}$$

This yields the primal-dual relationships in closed form:

$$x_i^\star \;=\; (\sum_{j=1}^{m} \lambda_j \, c_{ij}) \tag{46}$$

This expression holds if the side constraint are ignored. Now to enforce side-constraints, one has to consider three situations (see figure 4):

$$x_i(\lambda) = x_i^\star \quad \text{if} \quad \underline{x}_i \;\leq\; x_i^\star \;\leq \overline{x}_i, \tag{47}$$

$$x_i(\lambda) \;=\; \underline{x}_i \quad \text{if} \quad x_i^\star \;\leq\; \underline{x}_i, \tag{48}$$

$$x_i(\lambda) \;=\; \overline{x}_i \quad \text{if} \quad x_i^\star \geq\; \overline{x}_i. \tag{49}$$

Even if the solution introduces different conditions, the primal-dual relations $\mathbf{x} = \mathbf{x}(\lambda)$ are still available under closed form. Nonetheless considering different cases leads to a non-smooth expression. Indeed the derivation of this expression is discontinuous when changing from one condition to another, that is when a fixed variable becomes free or conversely.

The dual function is formed by inserting these primal-dual relationships into the Lagrangian function, which leads to the following statement of the dual problem (one cannot find anymore an explicit expression of dual function):

$$\begin{array}{ll} \max\limits_{\lambda} & \frac{1}{2}\sum\limits_{i=1}^{n} x_i^2(\lambda) \;-\; \sum\limits_{j=1}^{m} \lambda_j \, (\sum\limits_{i=1}^{n} c_{ij} \, x_i(\lambda) \;-\; d_j) \\ \text{s.t.} & \lambda_j \;\geq\; 0 \end{array}$$

First derivatives of the dual function are given by the value of primal variable constraint, in which primal variables has been replaced by their expression in terms of the Lagrange multipliers:

$$\frac{\partial \ell}{\partial \lambda_j} \;=\; d_j \;-\; (\sum_{i=1}^{n} c_{ij}\, x_i(\lambda)) \tag{50}$$

Second derivatives of dual function, giving Hessian matrix, are easily available too:

$$A_{jk} \;=\; \frac{\partial^2 \ell}{\partial \lambda_j \partial \lambda_k} \;=\; \frac{\partial}{\partial \lambda_k} \left[d_j - \sum_{i=1}^{n} c_{ij}\, x_i(\lambda) \right] \;=\; -\sum_{i=1}^{n} c_{ij}\, \frac{\partial x_i}{\partial \lambda_k} \tag{51}$$

Now the situation is much more complicated, because of the separate treatment of side constraints, which introduces non smooth primal-dual relations. On the one hand for free variables, we have:

$$\frac{\partial x_i}{\partial \lambda_k} \;=\; c_{ik} \quad \text{if} \quad \underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i \tag{52}$$

On the other hand, for fixed variables, it is obvious that

$$\frac{\partial x_i}{\partial \lambda_k} \;=\; 0 \quad \text{if} \quad x_i \;= \overline{x}_i \quad \text{or} \quad x_i \;=\; \underline{x}_i \tag{53}$$

Therefore the Hessian matrix takes the form:

$$A_{jk} \;=\; \frac{\partial^2 \ell}{\partial \lambda_j \partial \lambda_k} \;=\; =\; -\sum_{i \in F} c_{ij}\, c_{ik} \tag{54}$$

where set F is the set of index of free variables.

It turns out that the second derivatives of the dual function becomes discontinuous each time that a free primal variable becomes fixed or conversely, since one changes the set F of free variables in formula (54). From the primal dual relationships it is clear that the dual space is partitioned in several regions separated by second-order discontinuity planes. These planes are defines here by their equation:

$$(\sum_{j=1}^{m} \lambda_j\, c_{ij}) \;=\; \underline{x}_i \qquad (\sum_{j=1}^{m} \lambda_j\, c_{ij}) \;=\; \overline{x}_i \tag{55}$$

Numerical application example of quadratic problem with separable treatment of side constraints

$$\begin{array}{ll} \min\limits_{x_1\,,\,x_2} & x_1^2 \;+\; x_2^2 \\ \text{s.t.} & x_1 \;+\; x_2 \;\geq\; 4 \\ & x_1 \;-\; x_2 \;\geq\; -4 \\ & 1 \;\leq\; x_i \;\leq 4 \quad i = 1, 2 \end{array}$$

The Langrange function writes:

$$L(\mathbf{x}, \lambda) = x_1^2 \, + \, x_2^2 \, - \, \lambda_1 \, (x_1 \, + \, x_2 \, - \, 4) \, - \, \lambda_2 \, (x_1 \, - \, x_2 \, + \, 4)$$

The stationnarity of the dual function provides the primal dual relationships:

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= x_1 - \lambda_1 - \lambda_2 \quad \Longrightarrow \quad x_1 = \lambda_1 + \lambda_2 \\ \frac{\partial L}{\partial x_2} &= x_2 - \lambda_1 + \lambda_2 \quad \Longrightarrow \quad x_2 = \lambda_1 - \lambda_2 \end{aligned}$$

If one wants to treat the side-constraints, one has to invoque the fact that the functions are separable and solve the two one-dimensional minimisation problems with side-constraints:

$$\begin{aligned} \min_{1 \le x_1 \le 4} \quad & L_1(x_1) = 1/2 \, x_1^2 - \lambda_1 \, x_1 - \lambda_2 \, x_2 \\ \min_{1 \le x_2 \le 4} \quad & L_2(x_2) = 1/2 \, x_2^2 - \lambda_1 \, x_2 + \lambda_2 \, x_2 \end{aligned}$$

They admit a closed-form solution and they result in fully explicit primal-dual relationships $\mathbf{x} = \mathbf{x}(\lambda)$:

$$\begin{aligned} x_1 = \lambda_1 + \lambda_2 \quad & \text{if} \quad 1 \le \lambda_1 + \lambda_2 \le 4 \\ x_1 = 1 \quad & \text{if} \quad \lambda_1 + \lambda_2 \le 1 \\ x_1 = 4 \quad & \text{if} \quad \lambda_1 + \lambda_2 \ge 4 \end{aligned}$$

$$\begin{aligned} x_2 = \lambda_1 - \lambda_2 \quad & \text{if} \quad 1 \le \lambda_1 - \lambda_2 \le 4 \\ x_2 = 1 \quad & \text{if} \quad \lambda_1 - \lambda_2 \le 1 \\ x_2 = 4 \quad & \text{if} \quad \lambda_1 - \lambda_2 \ge 4 \end{aligned}$$

The discontinuity planes

$$\begin{aligned} \lambda_1 + \lambda_2 &= 4 \\ \lambda_1 + \lambda_2 &= 1 \\ \lambda_1 - \lambda_2 &= 4 \\ \lambda_1 - \lambda_2 &= 1 \end{aligned}$$

divide the dual space into six regions as shown in figure 6.

The value of the dual function is changing from one region to another since the value of the primal dual relationship is changing (see figure 7). One can find the dual function expression by inserting the current primal-dual relationship into the Lagrange function.

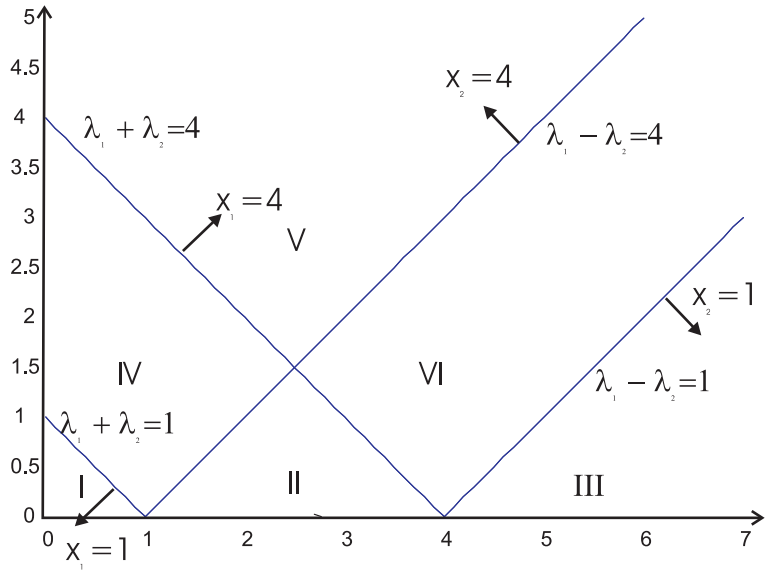

Figure 6: Structure of the dual space with its discontinuity regions

Region	x_1	x_2	$\ell(\lambda_1, \lambda_2)$
I	1	1	$1 + 2\lambda_1 + 2\lambda_2$
II	$\lambda_1 + \lambda_2$	$\lambda_1 - \lambda_2$	$-\lambda_1^2 - \lambda_2^2 + 4\lambda_1 - 4\lambda_2$
III	4	1	$16 - 4\lambda_1 - 4\lambda_2$
IV	$\lambda_1 + \lambda_2$	1	$0.5\lambda_1^2 - 0.5\lambda_2^2 - \lambda_1\lambda_2 + 3\lambda_1 - 3\lambda_2 + 0.5$
V	4	1	$17/2 - \lambda_1 - 7\lambda_2$
VI	4	$\lambda_1 - \lambda_2$	$-0.5\lambda_1 - 0.5\lambda_2^2 + \lambda_1\lambda_2 - 8\lambda_2 + 8$

However the dual function and its first derivatives are discontinuous across the discontinuity planes. Only the second order derivatives are discontinuous.

2.9 Dual solution scheme of MMA sub-problems

Anticipating a little bit on structural approximations, it is rather interesting to study the principles of dual solutions applied to MMA approximated problems. After normalization, the MMA sub-problem can be written in the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{n} \frac{p_{i0}}{U_{ij} - x_i} + \sum_{i=1}^{n} \frac{q_{i0}}{x_i - L_{ij}} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \frac{p_{ij}}{U_{ij} - x_i} + \sum_{i=1}^{n} \frac{q_{ij}}{x_i - L_{ij}} \;\leq\; d_j \quad j = 1 \ldots m \\ & \underline{x}_i \;\leq\; x_i \;\leq \overline{x}_i \quad i = 1 \ldots n \end{aligned} \tag{56}$$

One must notice that in this expression the asymptotes may depend on both variable and constraint indices: U_{ij} and L_{ij}. One can also notice that for the sake of simplicity the

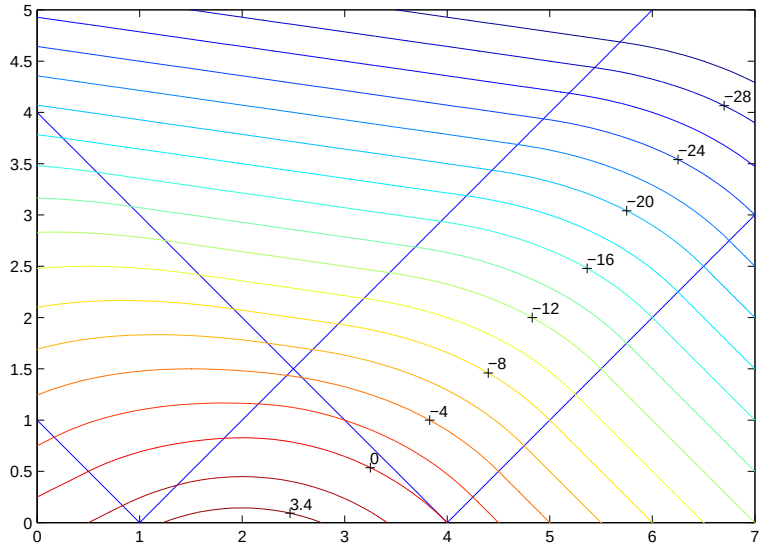

Figure 7: Dual space with discontinuity regions

objective function has been denoted by $g_0(\mathbf{x})$ instead of $f(\mathbf{x})$. In order to simplify the notations, we adopt the following notation: $\lambda_0 = 1$, so that Lagrangian function is given by:

$$L(\mathbf{x}, \lambda) \;=\; \sum_{j=0}^{m} \lambda_j \,(\sum_{i=1}^{n} \frac{p_{ij}}{U_{ij} - x_i} \;+\; \sum_{i=1}^{n} \frac{q_{ij}}{x_i - L_{ij}} \;-\; d_j) \tag{57}$$

The Lagragian problem is

$$\min_{\underline{x}_i \,\leq\, x_i \,\leq \overline{x}_i} \quad L(\mathbf{x}, \lambda) \tag{58}$$

Because of the separability property, the n-dimensional problem can be split into n one-dimensional problems relative to each variable "i":

$$\min_{\underline{x}_i \,\leq\, x_i \,\leq \overline{x}_i} \quad L_i(x_i, \lambda) \;=\; \sum_{j=0}^{m} \frac{\lambda_j \, p_{ij}}{U_{ij} - x_i} \;+\; \sum_{j=0}^{m} \frac{\lambda_j \, q_{ij}}{x_i - L_{ij}} \tag{59}$$

For a pure MMA [44] approximation, the asymptotes are the same for each constraint and they do not depend on the approximation index j: $U_{ij} = U_i$ and $L_{ij} = L_j$. The solution of this problem can be solved explicitly for each variable and gives rise to the primal-dual relations. Optimality conditions of the Lagrangian problem are:

$$\frac{\sum_{j=0}^{m} \lambda_j \, p_{ij}}{(U_i - x_i)^2} \;-\; \frac{\sum_{j=0}^{m} \lambda_j \, q_{ij}}{(x_i - L_i)^2} \;=\; 0 \tag{60}$$

From this expression, one extracts the candidate optimal value of variable x_i in terms of dual variables:

$$x_i^\star(\lambda) \;=\; \frac{U_i \;+\; \eta L_i}{\eta + 1} \tag{61}$$

where

$$\eta \;=\; \sqrt{\frac{\sum_{j=0}^{m} \lambda_j \, p_{ij}}{\sum_{j=0}^{m} \lambda_j \, q_{ij}}} \tag{62}$$

For an approximation of the GMMA family [42], the primal-dual relationships are no longer explicit since each asymptote depends now on both the primal variables (index "i") and on the constraint (index "j"). As the Lagrangian problem no longer admits a closed solution form, a Newton-Raphson scheme is adopted (as in Ref. [51]). However the approximation is separable and n one-dimensional numerical minimizations are performed. The iteration scheme for primal-dual relationships $x_i(\lambda)$ is given by:

$$x_i(\lambda^+) \;=\; x_i(\lambda) \;-\; \frac{\partial L_i / \partial x_i}{\partial^2 L_i / \partial x_i^2} \tag{63}$$

where

$$\frac{\partial L_i}{\partial x_i} \;=\; \sum_{j=0}^{m} \frac{\lambda_j \, p_{ij}}{(U_{ij} - x_i)^2} \;-\; \sum_{j=0}^{m} \frac{\lambda_j \, q_{ij}}{(x_i - L_{ij})^2} \tag{64}$$

$$\frac{\partial^2 L_i}{\partial x_i^2} \;=\; 2 \sum_{j=0}^{m} \frac{\lambda_j \, p_{ij}}{(U_{ij} - x_i)^3} \;-\; 2 \sum_{j=0}^{m} \frac{\lambda_j \, q_{ij}}{(x_i - L_{ij})^3} \tag{65}$$

For both MMA and GMMA schemes, we have to take care of the side-constraints and we consider three situations:

$$x_i(\lambda) = x_i^\star \quad \text{if} \quad \underline{x}_i \;\leq\; x_i^\star \;\leq \overline{x}_i, \tag{66}$$

$$x_i(\lambda) \;=\; \underline{x}_i \quad \text{if} \quad x_i^\star \;\leq\; \underline{x}_i, \tag{67}$$

$$x_i(\lambda) \;=\; \overline{x}_i \quad \text{if} \quad x_i^\star \geq\; \overline{x}_i. \tag{68}$$

Values of primal variables in terms of Lagrange multipliers are then inserted in the Lagrange function to calculate dual function value. Gradient of dual function is also given by the value of primal constraints.

When primal-dual relationships are not available in closed form, each evaluation requires the solution of non linear problems, which needs an additional numerical effort. In order to circumvent the problem, Fleury [25] suggested to break the solution of the convex subproblem itself into a sequence of quadratic explicit separable sub-subproblems. As explained in the former section, these problems can be solved efficiently by dual methods. So the non-explicit character of primal-dual relationships is not a real obstacle in practice.

3 STRUCTURAL APPROXIMATIONS

3.1 Characteristics of approximations

When building approximation schemes, one pursues several (and sometimes antagonist) goals. It is good to explicitly give them before reviewing the different schemes.

The first constraint one has to keep in mind is that the solution of the generated sub-problems can be carried out by efficient and and small computationally expensive methods. In order to be able to resort to dual methods, one would like, at first, that the solution of the dual problem is the same as the solution of primal sub-problem. Then the efficiency of the dual method holds if the computational effort to move to the dual space is small to save the benefit of the dual solution. To match these conditions, the sub-problems ($\tilde{P}$) and the related structural approximations must be:

- *Convex*. The convexity of the sub-problems insures that there is a unique solution and that the solution of the dual problem is the solution of the original problem. In addition the dual problem is concave and efficient algorithms can exploit this property.

- *Separable*. The separability is essential to arrive at relations between the primal design variables and the Lagrange multipliers that are easy to compute. Primal variables are given by an uncoupled system of equations in terms of the Lagrange multipliers, which is possible to solve independently for each primal variable. Furthermore for most of the approximation schemes, it is even possible to express the solution in closed form. Separability is an essential property to reduces the computational effort with dual methods.

Besides those two basic properties one would like that the procedure leads to a stationary solution within a minimum number of iterations without constraint violations. Obviously the number of structural analyses can be largely reduced when appropriate schemes are used. To this end one would like that the structural approximations are:

- *Precise* in order to give the best fit to the real responses in the largest neighborhood possible. Then for higher quality approximations one experiences faster convergence rate to the solution.

- *Sufficiently conservative* in order to generate a sequence of steadily improved iterations which provide feasible solutions at any stage of the optimization process. Conservative approximations are obtained by increasing the value of the convexity terms in order to reduce the size of the trust region and to avoid constraint violations. This argument is generally antagonist with the precision property.

Furthermore the overall computational time to come to an optimum depends also upon the numerical work that is necessary to build the approximation. To this end one wants to look for

- *A minimum computational effort* to generate the approximation, but also to calculate the necessary information that is required to built it. For example computing the true second order sensitivity is generally expensive so that it is generally avoided and replaced by an approximated second order information e.g. with Quasi-Newton techniques. Another example is the cost of the solution of the primal-dual relationships: Too complex approximation schemes are generally highly penalized because they require elaborated and expensive numerical solutions of the primal-dual system, which, in its turn, increases the numerical effort to formulate the dual problem. Therefore, it comes that reducing the overall computational effort is also antagonist to the accuracy and precision criteria, so that a trade-off has to be found between the two criteria.

This list is non extensive, one must also keep in mind that the approximation scheme and the approximated sub-problems must be

- *Robust*, in order to be able to construct the approximation in an automatic and reliable manner and in a lot of situations.
- *Flexible*, to be able to be used for various kinds of structural and geometric responses.
- *Overall convergence* in order to be able to end up in an optimum solution (may be a local optimum) from any starting point design.
- ...

The approximation schemes are generated through first or second order Taylor expansion of the design function $g_j(\mathbf{x})$ around current design point $\mathbf{x^{(k)}}$. The different schemes are expressed in terms of specific intermediate linearization variables, e.g. reciprocal variables $1/x_i$. Because of the Taylor expansion procedure, the approximations are *local* schemes, which means that the precision of the approximation is restricted to a neighborhood about the current design point.

We are now going to review the most important schemes and see the improvements (or drawbacks) that each one brings.

3.2 Linear approximation and sequential linear programming

When thinking about local approximation, the most natural and simple approximation scheme is the Taylor expansion around the current design point $\mathbf{x}^0$ restricted to linear terms. The linear approximation writes:

$$\tilde{g}_j(\mathbf{x}) \;=\; g_j(\mathbf{x^0}) + \sum_{i=1}^{n} \frac{\partial g(\mathbf{x^0})}{\partial x_i} \, (x_i - x_i^0) \tag{69}$$

The choice of linear approximation is natural when it corresponds to the nature of the restriction. For example it is exact for the volume restriction when using SIMP material. It is also mandatory when dealing with equality constraints.

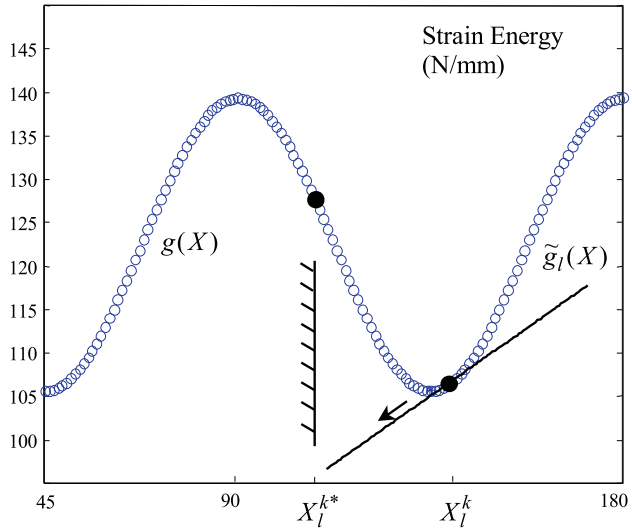

Figure 8: Linear approximation scheme applied to the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

When the linear approximation is applied to every restrictions of the problem, one transforms the original problem into a *sequence of linear programming problems*:

$$\begin{aligned} \min_{x_i} \quad & g_0(\mathbf{x^0}) + \nabla g_0(\mathbf{x^0})^T(\mathbf{x}-\mathbf{x^0}) \\ \text{s.t.} \quad & g_j(\mathbf{x^0}) + \nabla g_j(\mathbf{x^0})^T(\mathbf{x}-\mathbf{x^0}) \ \leq \ 0 \\ & \underline{x_i} \ \leq \ x_i \ \leq \ \bar{x}_i \end{aligned} \tag{70}$$

With the general formulation of linear programming, it is possible to treat a wide spectrum of problems independently of the nature of the restrictions. But, probably the most interesting advantage of sequential linear programming, comes from the fact that this kind of sub-problems can be readily solved efficiently with the help of linear programming algorithms like a *SIMPLEX* algorithm or a primal-dual interior point method [33] which is very well adapted to very large scale problems.

However the sequential linear approximation strategy has also some drawbacks. Due to the lack of convexity of the sub-problem, one can have some problems to stabilize the convergence process. One can experience some oscillations of the convergence or generate a sequence of un-feasible designs during the iteration history. To overcome the difficulty, one has to add some *move-limits* to play the role of a 'trust region':

$$max(\underline{x_i}, x_i^0 - \alpha_i) \ \leq \ x_i \ \leq \ min(\bar{x}_i, x_i^0 + \alpha_i) \tag{71}$$

For topology problems, we suggest an adaptive move-limit strategy which reduces the design interval when the design variable oscillates and which enlarges it when the convergence

process is stable.

- For iteration $k = 1$ and 2, take an initial 10 percent move-limits

$$\alpha_i \;=\; 0.1(\bar{x_i} - \underline{x_i}) \tag{72}$$

- For iterations $k > 2$, update move-limits according to the following rule:

$$\alpha_i^{(k+1)} = 0.7\alpha_i^{(k+1)} \quad \text{if} \quad \Delta x_i^{(k+1)} \Delta x_i^{(k)} < 0 \tag{73}$$

$$\alpha_i^{(k+1)} = 1.2\alpha_i^{(k+1)} \quad \text{if} \quad \Delta x_i^{(k+1)} \Delta x_i^{(k)} \geq 0 \tag{74}$$

When the original problem is very non-linear, it is sometimes necessary to adopt very close move-limits, which reduces a lot the convergence speed. It is usual that 100 iterations may be necessary to come to a stable solution in topology optimization.

3.3 Reciprocal variable expansion

As soon as the beginning of the seventies, Schmidt and his co-authors (see for example Ref. [40]) showed that an approximation scheme in terms of reciprocal variables $y_i = 1/x_i$ is favorable to reduce the non-linearity of responses in sizing problems. Thus a good local approximation is realized when expanding the responses $g_j(\mathbf{x})$ in terms of intermediate variables $y_i = 1/x_i$ to better capture the non-linear character of the function:

$$\tilde{g}_j(\mathbf{x}) \;=\; g_j\;(\mathbf{x^0}) + \sum_{i=1}^{n} \; -(x_i^0)^2 \; \frac{\partial g(\mathbf{x^0})}{\partial x_i} \; (\frac{1}{x_i} - \frac{1}{x_i^0}) \tag{75}$$

In a lot of structural applications, the reciprocal scheme (75) leads to successful results with an impressive reduction of the number of stages to arrive at a stationary solution of the design problem. In fact the scheme is efficient when all the first derivatives of the function are negative (like in determinate structures), since in this case the approximation (75) has only positive curvature terms and is conservative. However when the problem is highly non-linear or when dealing with other kinds of problems like shape or composite problems, Fleury and Braibant [27] observed that the derivatives can have mixed signs and that convergence process is not stable anymore, because the approximation (75) is no longer conservative for the variables which have positive derivatives.

3.4 Mixed linearization approximation: CONLIN

To overcome the difficulty of the reciprocal variable and linear expansions, Fleury and Braibant [27] expressed the idea to combine reciprocal expansion (when the derivative is negative) and linear approximation (when the derivative has a positive value) in a single mixed approximation scheme:

$$\tilde{g}_j(\mathbf{x}) = g_j(\mathbf{x^0}) + \sum_{+} \frac{\partial g_j}{\partial x_i}(x_i - x_i^0) - \sum_{-} (x_i^0)^2 \frac{\partial g_j}{\partial x_i}(\frac{1}{x_i} - \frac{1}{x_i^0}) \tag{76}$$

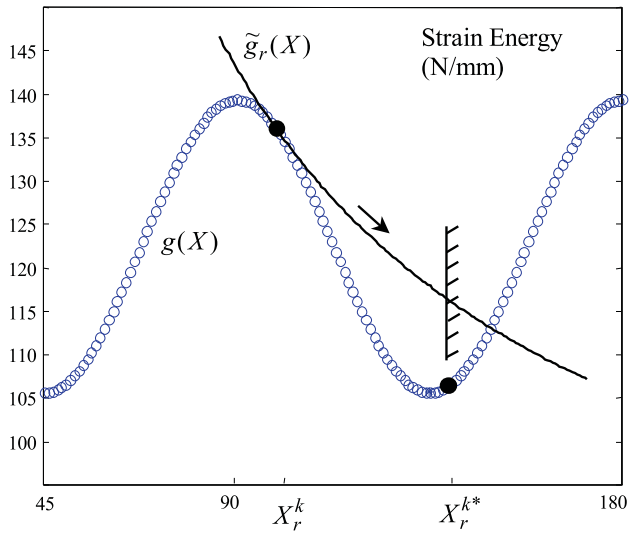

Figure 9: Reciprocal expansion scheme applied to the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

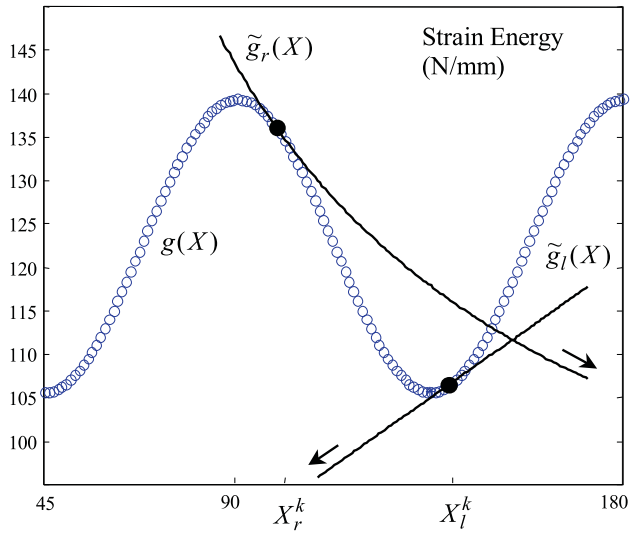

Figure 10: CONLIN approximation of the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

where $\sum_+$ is the sum over all the terms for which the derivative is positive and $\sum_-$ is the sum over all the terms for which the derivative is negative.

The scheme (76) is unconditionally convex since all its second derivatives are positive or nul, that's why it is named CONLIN as Convex Linearization. Moreover, it was demonstrated by Starnes and Haftka [43] that this scheme is the most conservative approximation that can be generated with linear and reciprocal variables. This convex character gives rise to the conservative character of CONLIN, which means that the approximation (76) tends to lie in the feasible domain of the constraint. It follows that the CONLIN method mostly tends to generate feasible new solutions. In numerous applications - mostly related to sizing, shape but also topology optimisation -, CONLIN technique leads to a stable convergence and a feasible sequence of steadily improved designs. Convergence speed is generally fast: 10 to 20 F.E. analyses are generally necessary to reach a stationary solution in sizing optimization.

Another strength of CONLIN comes from the fact that it is very well suited to a use dual optimizers, because of the convex and separable character of the approximated functions, the sub-problem lends itself to an efficient solution procedure based on dual method and second order maximization algorithm (see Fleury [23]).

The main drawback of CONLIN is related to the fact that the curvature of the approximation is fixed and there is no way for the user to modify it except by using tricky changes of design variables. This also sometimes introduces a bad fitting of the approximation to the real function. So despite numerous successes, the mixed linearization leads sometimes to too slow or unstable convergence histories as well. Famous example of divergence processes were reported by Svanberg [44].

3.5 Method of Moving Asymptotes: MMA

To be able to adjust the curvature of the approximation and to have a better fitting to the real function, one has to use the Method of Moving Asymptotes (MMA) from Svanberg [44] which extends the convex linearization scheme.

$$\tilde{g}_j(\mathbf{x}) = r_j^0 + \sum_{i=1}^{n} \frac{p_{ij}}{U_i - x_i} \; + \; \sum_{i=1}^{n} \frac{q_{ij}}{x_i - L_i} \tag{77}$$

with

$$\begin{aligned} p_{ij} &= \max\{0, (U_i - x_i^0)^2 \frac{\partial g_j}{\partial x_i}\} \\ q_{ij} &= \max\{0, -(x_i^0 - L_i)^2 \frac{\partial g_j}{\partial x_i}\} \end{aligned} \tag{78}$$

and where r_j^0 collects all zero order terms that are adjusted to fit to the constraint value in $\mathbf{x}^0$.

Once again the scheme can be regarded as a first order Taylor expansion in terms of intermediate variables $1/(U_i - x_i)$ or $1/(x_i - L_i)$ depending upon the sign of the derivatives. Of course, the asymptotes values are such that $L_i^{(k)} < x_i^{(k)} < U_i^{(k)}$.

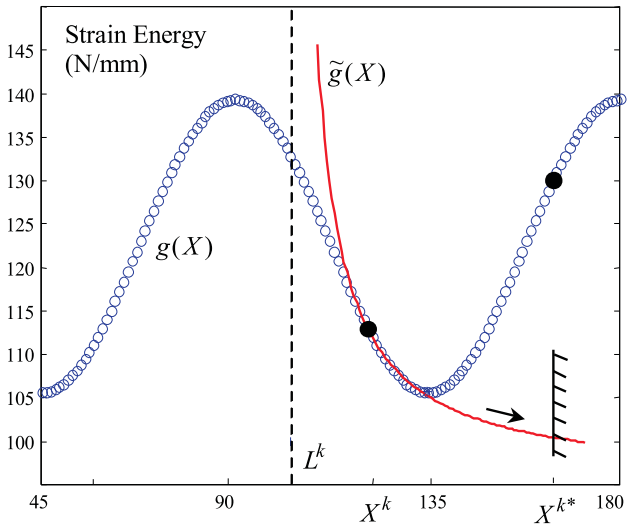

Figure 11: MMA approximation of the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

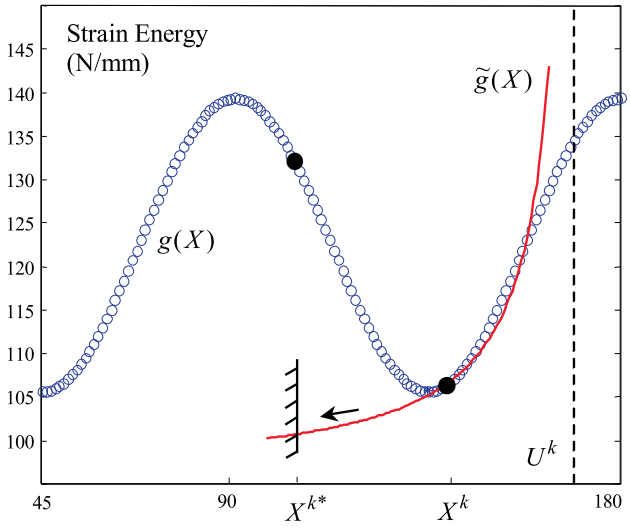

Figure 12: MMA approximation of the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

The two sets of vertical asymptotes U_i and L_i generalize the vertical asymptotes introduced in CONLIN with the change of variables $y_i = 1/x_i$. In fact the CONLIN approximation can be recovered from the general statement by assuming $L_i = 0$ for the lower bound and $U_i \to \infty$ for the lower bound.

The two sets of asymptotes also play a role of move-limits. This analysis has been even more reinforced with the recent work that has been done in primal-dual interior point methods in which side-constraints (here lower bound) are taken into account through barrier functions of the type [6, 31]:

$$\ln(x_i - \underline{x}_i) \quad \text{or} \quad \frac{1}{x_i - \underline{x}_i} \tag{79}$$

To adjust the MMA approximation, one has to play with the parameters of the approximation i.e. the position of the vertical asymptotes. One could calculate that the second order derivatives (and the convexity as well) increase when the asymptotes are pushed closer to the design point. Conversely the convexity of the approximation is reduced when they are moved away from the design point $\mathbf{x^{(k)}}$.

One difficulty of the method comes from the fact that the automatic choice of these curvature parameters remains empirical and mostly problem dependent. Svanberg [44] proposed a heuristic strategy for asymptotes update based on the design variable oscillations. For the iterations $k = 1$ and 2, the default values of the asymptotes are adopted:

$$\begin{aligned} L_i^{(k)} &= x_i^{(k)} - s_0(\bar{x_i} - \underline{x_i}) \\ U_i^{(k)} &= x_i^{(k)} + s_0(\bar{x_i} - \underline{x_i}) \end{aligned} \tag{80}$$

with $s_0 = 0.5$ is suggested by Svanberg. Then asymptotes are updated in the following ways.

For $k > 2$, the update scheme is the following. When the convergence process is smooth $(x_i^{(k-2)} - x_i^{(k-1)}).(x_i^{(k-1)} - x_i^{(k)}) \geq 0$, the convexity can be reduced not to slow down the convergence rate. So the asymptotes can be moved away from design point.

$$\begin{aligned} L_i^{(k)} &= x_i^{(k)} - s_1(x_i^{(k-1)} - L_i^{(k-1)}) \\ U_i^{(k)} &= x_i^{(k)} + s_1(U_i^{(k-1)} - x_i^{(k-1)}) \end{aligned} \tag{81}$$

with s_1 less than 1 and generally chosen as $s_1 = 1.2$. When the problem is difficult, for instance when a p-norm of the local stress constraints is considered like in [18], it is better to reduce this factor up to $s_1 = 1.05$ for example.

While when the convergence history oscillates $(x_i^{(k-2)} - x_i^{(k-1)}).(x_i^{(k-1)} - x_i^{(k)}) < 0$, one has to move the asymptotes closer to the design point to increase the convexity. So one gets the following update rules

$$\begin{aligned} L_i^{(k)} &= x_i^{(k)} - s_2(x_i^{(k-1)} - L_i^{(k-1)}) \\ U_i^{(k)} &= x_i^{(k)} + s_2(U_i^{(k-1)} - x_i^{(k-1)}) \end{aligned} \tag{82}$$

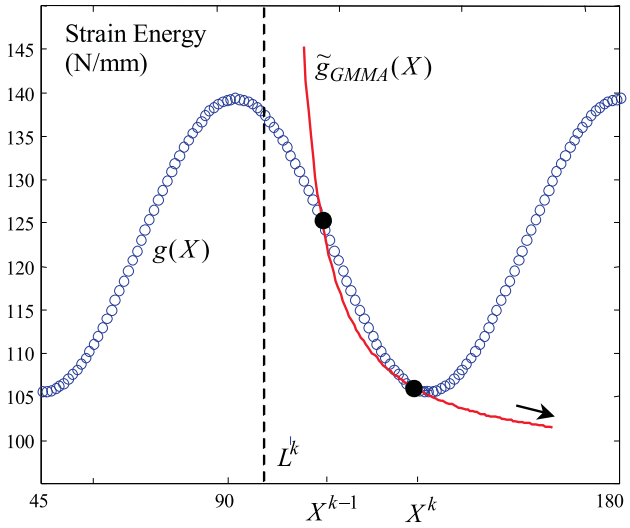

Figure 13: Asymptote update based on fitting the approximation to the previous point value [50]

with s_2 inferior to 1 and generally chosen as $s_2 = 1/\sqrt{s_1}$. Svanberg [44] suggests $s_2 = 0.7$, but when the convergence is difficult as in [18], $s_2 = 0.65$ gives good results.

Alternatively as in [50], they can also be defined by the value of the considered structural response at the previous iteration, $g_j(\mathbf{x}^{(k-1)})$. In this case, the asymptotes for each constraint are computed by

$$\begin{aligned} L_{ij}^{(k)} &= x_i^{(k)} - s_j\,(x_i^{(k-1)} - L_{ij}^{(k-1)}) \\ U_{ij}^{(k)} &= x_i^{(k)} + s_j\,(U_{ij}^{(k-1)} - x_i^{(k-1)}) \end{aligned} \tag{83}$$

where the s_j factor is adjusted to fit the approximation to the value of the function at the previous iteration (see Figure 13). This needs solving a one-dimensional line search (84) with a Newton-Raphson procedure:

$$\begin{aligned} & \tilde{g}_j(\mathbf{x}^{(k-1)}) = g_j(\mathbf{x}^{(k-1)}) \\ \Leftrightarrow \quad & \tilde{g}_j(s_j) - g_j(\mathbf{x}^{(k-1)}) = 0 \end{aligned} \tag{84}$$

3.6 Globally Convergent Method of Moving Asymptotes: GCMMA

As one can remark in figures 11 and 12 that MMA (but also COLNLIN) is made of sum of monotonous functions. When one is faced to non-monotonous problems (which is often the case with composite structures for example), it is mandatory to restrict the design variable motion in both directions. Therefore, as suggested by Svanberg [44], a move-limit strategy

is generally adopted in combination with MMA.

$$max(\underline{x}_i, 0.9L_j + 0.1x_i^0) \ \leq \ x_i \ \leq \ min(\bar{x}_i, 0.1x_i^0 + 0.9U_i) \tag{85}$$

More recently, Svanberg [46] proposed a new extension of MMA method which uses simultaneously both asymptotes in order to create a non monotonous approximation function. The general statement of the approximation is similar to MMA formula:

$$\tilde{g}_j(\mathbf{x}) = r_j^0 + \sum_{i=1}^{n} \frac{p_{ij}^{(k)}}{U_i - x_i} \ + \ \sum_{i=1}^{n} \frac{q_{ij}^{(k)}}{x_i - L_i} \tag{86}$$

but coefficients $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are both non zero in general. They are as chosen as follows:

$$p_{ij}^{(k)} \ = \ (U_i^{(k)} - x_i^{(k)})^2 \ \left(\max\{0, \frac{\partial g_j(\mathbf{x^{(k)}})}{\partial x_i}\} \ + \ \frac{\rho_j^{(k)}}{2} \, (U_i^{(k)} - L_i^{(k)}) \right) \tag{87}$$

$$q_{ij}^{(k)} \ = \ (x_i^{(k)} - L_i^{(k)})^2 \ \left(\max\{0, -\frac{\partial g_j(\mathbf{x^{(k)}})}{\partial x_i}\} \ + \ \frac{\rho_j^{(k)}}{2} \, (U_i^{(k)} - L_i^{(k)}) \right) \tag{88}$$

where $\rho_j^{(k)}$ are strictly positive parameters (to insure the convexity of the approximation) and they are updated together with the asymptotes $L_i^{(k)}$ and $U_i^{(k)}$. This is precisely the presence of the complementary term in $\rho_j^{(k)}$ that allows the approximation to be non-monotonous by using in the same time the two asymptotes $L_i^{(k)}$ and $U_i^{(k)}$.

Mobile asymptotes are updated with the same rule as in MMA (see formula (80-82)). The additional parameters $\rho_j^{(k)}$ are chosen as follows. For the two first iterations ($k = 1$), one choose

$$\rho_j^{(1)} = \varepsilon \quad \forall \, j \in \{0, 1 \ldots, m\} \quad 0 < \varepsilon \ll 1 \tag{89}$$

In the later iterations ($k \geq 2$), the parameters $\rho_i^{(k)}$ are updated according to

$$\begin{aligned} \rho_j^{(k)} &= 2 \, \rho_j^{(k-1)} \qquad \text{if} \qquad \tilde{g}_j^{(k-1)}(\mathbf{x^{(k)}}) \ < \ g_j(\mathbf{x^{(k)}}) \\ \rho_j^{(k)} &= \rho_j^{(k-1)} \qquad \text{if} \qquad \tilde{g}_j^{(k-1)}(\mathbf{x^{(k)}}) \ \geq \ g_j(\mathbf{x^{(k)}}) \end{aligned} \tag{90}$$

Further (in order to prove the convergence!), if

$$\tilde{g}_j^{(k-1)}(\mathbf{x^{(k)}}) \ \geq \ g_j(\mathbf{x^{(k)}}) \quad \forall \, j \in \{0, 1 \ldots, m\}$$

the asymptotes should now instead be updated as

$$\begin{aligned} L_i^{(k)} &= x_i^{(k)} \ - \ (x_i^{(k-1)} - L_i^{(k-1)}) \\ U_i^{(k)} &= x_i^{(k)} \ + \ (U_i^{(k-1)} - x_i^{(k-1)}) \end{aligned} \tag{91}$$

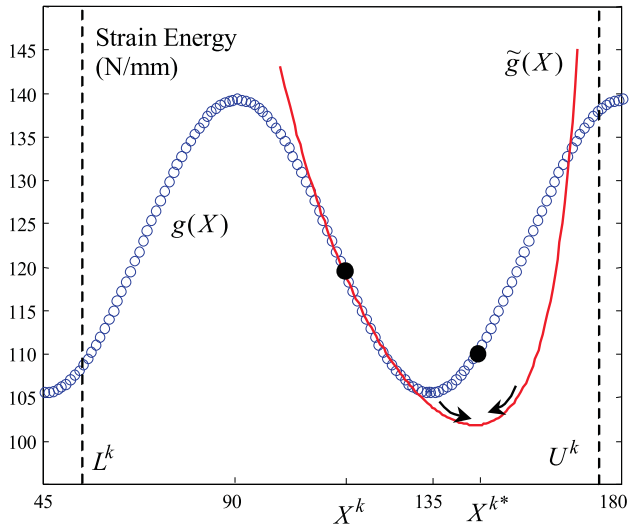

Figure 14: GCMMA approximation of the strain energy in a two plies symmetric laminate subject to shear and torsion loads [9]

Even though that for this scheme, one can prove the globally convergent character of the method (i.e. it converges to a stationary point from any starting point; which not means that it converges to THE global optimum of the problem!), the practical experience is that, in most cases, it converges more slowly than the original MMA (on problems where MMA does converge). The reason for this is that since the parameters $\rho_i^{(k)}$ are increased, and never decreased, the approximations become increasingly conservative. This may eventually lead to very small steps in the iteration process.

3.7 Example

To illustrate the difference between the various approximation schemes, we now plot the contours $g(\mathbf{x}) = 0$ of the real and approximated constraints for a simple analytic example. The function under study is

$$g(\mathbf{x}) = 5\,x_2 \;-\; x_1^2$$

First order derivatives of function g are:

$$\frac{\partial g}{\partial x_1} \;=\; -2\,x_1 \qquad \frac{\partial g}{\partial x_2} \;=\; 5$$

The current approximation point is $(x_1^0, x_2^0) \;=\; (4, 4)$ where the function and derivative values are:

$$g(\mathbf{x^0}) \;=\; 4 \qquad \nabla g(\mathbf{x^0}) \;=\; [-8\,,\,5]^T$$

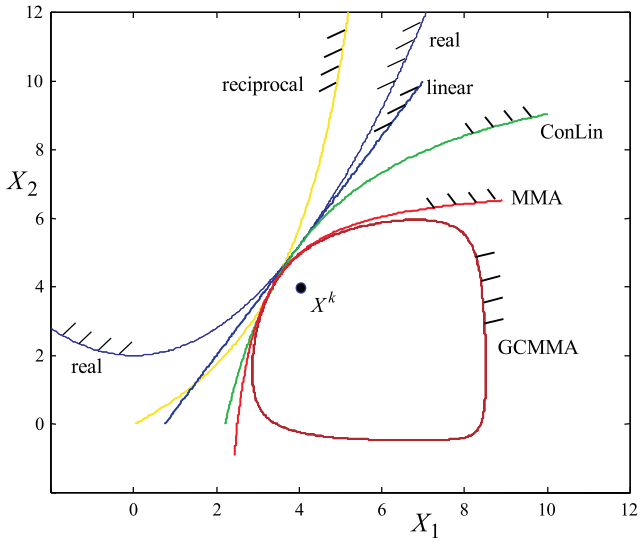

Figure 15: Comparison between the different approximation schemes for the function $g(\mathbf{x}) = 5\,x_2 - x_1^2$ at point $(x_1, x_2) = (4, 4)$

The reciprocal scheme is not conservative and lies in the un-feasible region of the true constraint. Linear, CONLIN, MMA and GCMMA are gradually more conservative and lie in the feasible part of the design space.

Remark that as CONLIN, MMA, etc. are monotonous approximations, so that the contours are open curves, while GCMMA which is a non monotonous approximation leads to a closed contour curve.

3.8 The Sequential Quadratic Programming method SQP

An alternative strategy to remedy to CONLIN's problems consists in improving the quality of the approximation and in introducing second order derivative information. Of course the most famous approach using second order derivatives is the Sequential Quadratic Programming approach or SQP (see for instance [38]). This scheme is however not separable and dual maximization approach is not efficient in this case.

3.8.1 Introduction: SQP for problems with equality constraints

In order to solve the optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & h_j(\mathbf{x}) = 0 \qquad j = 1, \ldots, m \end{array} \tag{92}$$

the basic idea is to formulate and solve a quadratic programming subproblem in each iteration which is obtained by linearizing the constraints and approximating quadratically the Lagrange function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda^T \, \mathbf{h}(\mathbf{x})$$

Assume that $\mathbf{x}^k$ is the current primal variable and that λ^k are the actual Lagrange multiplier vector. The optimality conditions of the problem (92) writes:

$$\nabla_x L(\mathbf{x}, \lambda) = \nabla_x f(\mathbf{x}) + \lambda^T \, \nabla_x \mathbf{h}(\mathbf{x}) = 0 \tag{93}$$

$$\mathbf{h}(\mathbf{x}) = 0 \tag{94}$$

In order to solve this non linear system of equations in variables $\mathbf{x}^0$ and λ^0 and finding new primal and dual variables $\mathbf{x}^+ = \mathbf{x}^0 + \mathbf{d}$ and $\lambda^+ = \lambda^0 + \Delta\lambda$, we use the usual Newton method that consists in linearizing the equations and we get:

$$[\nabla_x f(\mathbf{x}^0) + \lambda^{0\,T} \nabla_x \mathbf{h}(\mathbf{x}^0)] + [\nabla^2_{xx} f(\mathbf{x}^0) + \lambda^T \, \nabla^2_{xx} \mathbf{h}(\mathbf{x}^0)] \, \mathbf{d} + \Delta\lambda^T \nabla_x \mathbf{h}(\mathbf{x}^0) = 0$$
$$\mathbf{h}(\mathbf{x}^0) + \nabla_x \mathbf{h}(\mathbf{x}^0) \, \mathbf{d} = 0$$

When reordering the terms in the first equation, one gets the final system of equations:

$$\nabla^2_{xx} L(\mathbf{x}^0, \lambda^0) \, \mathbf{d} + \lambda^{+\,T} \nabla_x \mathbf{h}(\mathbf{x}^0) = -\nabla_x f(\mathbf{x}^0) \tag{95}$$

$$\nabla_x \mathbf{h}(\mathbf{x}^0) \, \mathbf{d} = -\mathbf{h}(\mathbf{x}^0) \tag{96}$$

Or as we see quite often under the following matrix form:

$$\begin{bmatrix} \nabla^2_{xx} L(\mathbf{x}^0, \lambda^0) & \nabla_x \mathbf{h}(\mathbf{x}^0)^T \\ \nabla_x \mathbf{h}(\mathbf{x}^0) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \lambda^+ \end{bmatrix} = \begin{bmatrix} -\nabla_x f(\mathbf{x}^0) \\ -\mathbf{h}(\mathbf{x}^0) \end{bmatrix} \tag{97}$$

As the Hessian matrix $\nabla^2_{xx} L(\mathbf{x}^0, \lambda^0)$ is generally positive definite on the nullspace of $\nabla_x \mathbf{h}(\mathbf{x}^0)$ (which is usually the case in the neighborhood of a local minimum point) then the equations (95-96) or (97) are the optimality conditions for the following *quadratic programming problem* (see Svanberg [45]):

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{1}{2} \, \mathbf{d}^T \, \nabla^2_{xx} L^{(k)} \, \mathbf{d} + \nabla f^{(k)} . \mathbf{d} \\ \text{s.t.} \quad & \mathbf{h}^{(k)} + \nabla \mathbf{h}^{(k)} \, \mathbf{d} = 0 \end{aligned} \tag{98}$$

Thus the Lagrange-Newton method for solving optimization problems subject to equality constraints follows the iterative procedure:

- $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)}$ where $\mathbf{d}^{(k)}$ is the solution of the quadratic programming problem (98);
- $\lambda^{(k+1)}$ is the Langrage multiplier vector of the quadratic programming problem (98).

It is further possible to stabilize the method and insure its global convergence character by using a line-search so that:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \, \mathbf{d}^{(k)} \tag{99}$$

with α_k is such that $0 < \alpha_k \leq 1$. One can show that an inexact line-search procedure like the Armijo-Goldstein is sufficient to guarantee the global convergence character.

When it is not possible to calculate the second order derivatives or when they are too expensive to compute, it is possible to replace the Hessian of the Lagrange function $\nabla^2_{xx} L^{(k)}$ by an approximation $\mathbf{B}^{(k)}$ which is updated according to some quasi-Newton schemes on the basis of the accumulated gradient information.

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{1}{2} \, \mathbf{d}^T \, \mathbf{B}^{(k)} \, \mathbf{d} + \nabla_x f^{(k)} . \mathbf{d} \\ \text{s.t.} \quad & \mathbf{h}^{(k)} + \nabla_x \mathbf{h}^{(k)} \, \mathbf{d} = 0 \end{aligned} \tag{100}$$

3.8.2 Sequential Quadratic Programming of inequality constrained problems

Inasmuch the interpretation of the Lagrange-Newton method as a sequential quadratic programming method, it is natural to generalize the method also to inequality constrained problems:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0 \qquad j = 1, \ldots, m \end{aligned} \tag{101}$$

This problem can be solved as an iterative process in which the following quadratic programming problems are solved:

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{1}{2} \, \mathbf{d}^T \, \nabla^2_{xx} L^{(k)} \, \mathbf{d} + \nabla f^{(k)} . \mathbf{d} \\ \text{s.t.} \quad & \mathbf{g}_j^{(k)} + \nabla g_j^{(k)} \, \mathbf{d} \leq 0 \qquad j = 1, \ldots, m \end{aligned} \tag{102}$$

or if the second order derivative matrix is approximated using quasi-Newton updates $\mathbf{B}^{(k)}$ by

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{1}{2} \, \mathbf{d}^T \, \mathbf{B}^{(k)} \, \mathbf{d} + \nabla f^{(k)} . \mathbf{d} \\ \text{s.t.} \quad & \mathbf{g}_j^{(k)} + \nabla g_j^{(k)} \, \mathbf{d} \leq 0 \qquad j = 1, \ldots, m \end{aligned} \tag{103}$$

As for the equality constrained problem, the SQP iterative process consists in solving problem (102) or alternatively (103). Let $\mathbf{d}^{(k)}$ and $\mu^{(k)}$ be respectively the optimal solution of the quadratic problem and the associated Lagrange mutliplier vector. Then

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \, \mathbf{d}^{(k)} \tag{104}$$

and

$$\lambda^{(k+1)} = \mu^{(k)} \tag{105}$$

In (104) the line search has to be carried out to insure the global convergence of the method. The line search is based on a merit function that includes not only the objectif function but also some penalty terms (for instance) for the violated constraints:

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{m} r_j \, |\max(0, g_j(\mathbf{x}))| \tag{106}$$

The active set of constraints is updated using different strategies, which are particular to different implementations.

Sequential quadratic programming methods are standard general purpose algorithms for solving smooth nonlinear optimization problems, at least under the following assumptions:

- The problem is not too big.
- The functions and gradients can be evaluated with sufficiently high precision.
- The problem is smooth and well-scaled.

One of the most famous implement (and one of the most efficient) remains the NLPQLP Fortran routine by Schittkowski [39].

3.8.3 Estimation of second order derivatives: the Quasi Newton updates

The quasi Newton equation

In order to obtain an approximation to the Hessian matrix (matrix of the second order derivatives) of a given function $f(\mathbf{x})$, we consider the Taylor expansion of the gradient of $f(\mathbf{x})$: $\mathbf{g} = \nabla f$ in the vicinity of $\mathbf{x}^{(k+1)}$:

$$\mathbf{g}(\mathbf{x}^{(k)}) = \mathbf{g}(\mathbf{x}^{(k+1)}) + \mathbf{H}(\mathbf{x}^{(k+1)})[\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}] + \mathbf{\Delta} \tag{107}$$

with $\mathbf{\Delta} \to 0$ if $\mathbf{x}^{(k)} \to \mathbf{x}^{(k+1)}$.

If we define

$$\mathbf{y}^{(k)} = \mathbf{g}(\mathbf{x}^{(\mathbf{k+1})}) - \mathbf{g}(\mathbf{x}^{(k)}) \tag{108}$$

$$\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \tag{109}$$

and if we neglect the second order term $\mathbf{\Delta}$, we obtain the *quasi Newton equation*:

$$\mathbf{y}^{(k)} = \mathbf{H}^{(k+1)} \, \mathbf{s}^{(k)} \tag{110}$$

It is sometimes easier to work with the $\mathbf{S}^{(k+1)} = [\mathbf{H}^{(k+1)}]^{-1}$ inverse of the Hessian matrix. We construct the related quasi-Newton equation for the S matrix.

$$\mathbf{s}^{(k)} = \mathbf{S}^{(k+1)} \, \mathbf{y}^{(k)} \quad k \geq 0 \tag{111}$$

The matrix $\mathbf{S}^{(k+1)}$ is easily computable from $\mathbf{S}^{(k)}$ if it is obtained by adding to $\mathbf{S}^{(k)}$ a correction term $\mathbf{C}^{(k)}$, which depends only upon $\mathbf{S}^{(k)}$, $\mathbf{y}^{(k)}$ and $\mathbf{s}^{(k)}$.

$$\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} + \mathbf{C}^{(k)} \quad k \geq 0 \tag{112}$$

At the same time the correction $\mathbf{C}^{(k)}$ should be constructed to preserve the symmetry and the positive definiteness of $\mathbf{S}^{(k)}$.

Rank-one update

To obtain a symmetric correction $\mathbf{C}^{(k)}$ we investigate a correction in the form of a rank-one matrix:

$$\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} + \beta^{(k)}\, \mathbf{z}^{(k)}\, \mathbf{z}^{(k)\,T} \tag{113}$$

The vector $\mathbf{z}^{(k)}$ and the coefficient $\beta^{(k)}$ are selected so that the condition (112) is satisfied. After some algebraic manipulation in order to identify these coefficients, one gets the *Broyden rank-one quasi Newton updates*:

$$\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} + \frac{[\mathbf{s}^{(k)} - \mathbf{S}^{(k)}\mathbf{y}^{(k)}]\,[\mathbf{s}^{(k)} - \mathbf{S}^{(k)}\mathbf{y}^{(k)}]^T}{\mathbf{y}^{(k)\,T}\,[\mathbf{s}^{(k)} - \mathbf{S}^{(k)}\mathbf{y}^{(k)}]} \tag{114}$$

One can demonstrate that in the case of a quadratic function, the quasi-Newton update converge to the exact matrix of the second order derivatives in n iterations at most (n is the number of design variables).

For non quadratic functions, the positive definiteness of the update is only preserved along the iteration process at the price of using a line search. Therefore the rank-two updates are generally preferred because they preserve the positive definiteness of $\mathbf{S}^{(k)}$.

Davidon-Fletcher-Powell (DFP) rank-two updates

In the Davidon-Fletcher-Powell (DFP) method the updating formula is taken of the form:

$$\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} + \beta\, \mathbf{s}^{(k)}\, \mathbf{s}^{(k)\,T} + \gamma\, [\mathbf{S}^{(k)}\, \mathbf{y}^{(k)}]\, [\mathbf{S}^{(k)}\, \mathbf{y}^{(k)}]^T \tag{115}$$

Once again the coefficient identification starts with satisfying the quasi Newton equation (112). The equality holds for a particular choice of the coefficients β and γ, which gives the DFP updating formula:

$$\mathbf{S}_{DFP}^{(k+1)} = \mathbf{S}_{DFP}^{(k)} + \frac{\mathbf{s}^{(k)}\, \mathbf{s}^{(k)\,T}}{\mathbf{s}^{(k)\,T}\, \mathbf{y}^{(k)}} - \frac{[\mathbf{S}_{DFP}^{(k)}\, \mathbf{y}^{(k)}]\, [\mathbf{S}_{DFP}^{(k)}\, \mathbf{y}^{(k)}]^T}{\mathbf{y}^{(k)\,T} \mathbf{S}_{DFP}^{(k)}\, \mathbf{y}^{(k)}} \tag{116}$$

We can prove that if $\mathbf{S}_{DFP}^{(k)}$ is positive definite, then $\mathbf{S}_{DFP}^{(k+1)}$ is also positive definite if a line-search is used.

When applied to a quadratic function, it is possible to prove (see Fleury [22]) that DFP updates have the following properties

- It generates conjugate directions

$$\mathbf{s}^{(i)\,T} \mathbf{A} \mathbf{s}^{(j)} = 0 \quad i < j \leq k \tag{117}$$

- and

$$\mathbf{S}^{(k+1)}\, \mathbf{A}\, \mathbf{s}^{(i)} = \mathbf{s}^{(i)} \quad 0 \leq i \leq k \tag{118}$$

Which implies that $\mathbf{S}^{(n)} = \mathbf{A}^{-1}$ so that convergence is obtained after n steps.

It is also a simple matter to show that the updating formula (116) may be also inverted to approximate the Hessian matrix itself rather than its inverse. The resulting matrix $\mathbf{B}_{DFP}^{(k)}$ is given by the DFP update:

$$\mathbf{B}_{DFP}^{(k+1)} = \left(\mathbf{I} - \frac{\mathbf{y}^{(k)}\,\mathbf{s}^{(k)\,T}}{\mathbf{y}^{(k)\,T}\,\mathbf{s}^{(k)}}\right)\mathbf{B}_{DFP}^{(k)}\left(\mathbf{I} - \frac{\mathbf{s}^{(k)}\,\mathbf{y}^{(k)\,T}}{\mathbf{y}^{(k)\,T}\,\mathbf{s}^{(k)}}\right) + \frac{\mathbf{y}^{(k)}\,\mathbf{y}^{(k)\,T}}{\mathbf{y}^{(k)\,T}\,\mathbf{s}^{(k)}} \tag{119}$$

Broyden-Fletcher-Goldfard-Shanno (BFGS) rank-two updates

The complementary formula to equation (116) giving $\mathbf{S}_{DFP}^{(k)}$ and to equation (119) giving $\mathbf{B}_{DFP}^{(k)}$ are easily calculated by adopting for $\mathbf{B}_{BFGS}^{(k)}$ an updating formula analogous to $\mathbf{S}_{DFP}^{(k)}$ and vice versa.

If we adopt for $\mathbf{B}_{BFGS}^{(k)}$ a rank-two updating formula of the form (115), the coefficient are selected to satisfy the direct quasi-Newton equation:

$$\mathbf{B}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$$

It comes the famous Quasi-Newton update formula by Broyden-Fletcher-Goldfard-Shanno (BFGS):

$$\mathbf{B}_{BFGS}^{(k+1)} = \mathbf{B}_{BFGS}^{(k)} + \frac{\mathbf{y}^{(k)}\mathbf{y}^{(k)\,T}}{\mathbf{s}^{(k)\,T}\mathbf{y}^{(k)}} - \frac{[\mathbf{B}_{BFGS}^{(k)}\mathbf{s}^{(k)}][\mathbf{B}_{BFGS}^{(k)}\mathbf{s}^{(k)}]^T}{\mathbf{s}^{(k)\,T}\mathbf{B}_{BFGS}^{(k)}\mathbf{s}^{(k)}} \tag{120}$$

This scheme satisfies simultaneously the symmetry property, the positive definiteness of the update, and the Quasi-Newton condition (161). But it doesn't preserve sparse or diagonal structure of the previous estimation. Such updates preserving diagonal pattern can be derived from Thapa's theorems [48].

In inverse form, it is similarly obtained from the DFP formula (119)

$$\mathbf{S}_{BFGS}^{(k+1)} = \left(\mathbf{I} - \frac{\mathbf{s}^{(k)}\,\mathbf{y}^{(k)\,T}}{\mathbf{s}^{(k)\,T}\,\mathbf{y}^{(k)}}\right)\mathbf{S}_{BFGS}^{(k)}\left(\mathbf{I} - \frac{\mathbf{y}^{(k)}\,\mathbf{s}^{(k)\,T}}{\mathbf{s}^{(k)\,T}\,\mathbf{y}^{(k)}}\right) + \frac{\mathbf{s}^{(k)}\,\mathbf{s}^{(k)\,T}}{\mathbf{s}^{(k)\,T}\,\mathbf{y}^{(k)}} \tag{121}$$

One can notice that the role of $\mathbf{y}^{(k)}$ and $\mathbf{s}^{(k)}$ have been interchanged between the DFP and the corresponding BFGS formula.

According to several authors, there is growing evidence that the BFGS is the best current update formula for use in unconstrained optimization. This is due to the fact that the eigenvalues of the BFGS update (121) are systematically larger than those of (116).

3.9 Second order separable approximation schemes

Second order derivatives can also be used with separable approximation schemes. These ones are more generally derived from previously exiting first order schemes. Smaoui, Fleury and Schmit [42] proposed a generalized MMA in which the asymptotes are automatically

selected to match the second order derivatives. A bit later Fleury [24] generalized the use of second order derivatives to several other convex approximations (like the quadratic diagonal scheme and the expansion in terms of a power of the design variables) and showed there is a big pay-off related to the use of the second order information in structural approximations.

3.9.1 Quadratic Separable Approximations

The quadratic approximation is the second order Taylor's expansion of the given structural response. Nevertheless, the full second order expansion introduces a coupling terms and the approximation is not separable anymore. Furthermore, the full second order sensitivity analysis can be expensive for structural problems. So Fleury [24] proposed to neglect the off-diagonal second order terms and to admit that structural constraints are sufficiently well approximated by the following local expansion:

$$\tilde{g}(\mathbf{x}) = g(\mathbf{x_0}) \; + \; \sum_{i=1}^{n} \frac{\partial g(\mathbf{x_0})}{\partial x_i}(x_i - x_i^0) \; + \; \frac{1}{2}\sum_{i=1}^{n}(\frac{\partial^2 g(\mathbf{x_0})}{\partial x_i^2} + \delta_{ii})(x_i - x_i^0)^2 \tag{122}$$

The additional terms δ_{ii} are generally added in order to reinforce the convexity of the separable approximation and to play the role of 'move limits' in establishing a trust region around the design point. For example, the penalty terms δ_{ii} can be estimated so that the unconstrained optimum is kept within a given distance of the expansion point.

$$|x_i^* - x_i^0| \; = \; |\frac{\partial g}{\partial x_i}/(\frac{\partial^2 g}{\partial x_i^2} + \delta_{ii})| \; < \; \alpha \; x_i^0 \tag{123}$$

3.9.2 The Generalized Method of Moving Asymptotes (GMMA)

The approximation of the generalized method of moving asymptotes can be written in the following form:

$$\tilde{g}(\mathbf{x^0}) = c_0 \; + \; \sum_{i=1}^{n} \frac{a_i}{x_i - b_i} \tag{124}$$

As the original MMA [44], this approximation can easily match the function value and its first derivatives at current design point $\mathbf{x^0}$. If the second order information is available, Smaoui et al. [42] showed that the asymptotes could be selected automatically too. The parameters of the expansion are given by:

$$\begin{aligned} a_i &= -(x_i^0 - b_i)^2 \; \frac{\partial g(\mathbf{x^0})}{\partial x_i} \\ b_i &= x_i^0 \; + \; 2 \; \frac{\partial g(\mathbf{x^0})}{\partial x_i} / \max(\epsilon, \frac{\partial^2 g(\mathbf{x^0})}{\partial x_i^2}) \qquad 0 < \epsilon \ll 1 \end{aligned} \tag{125}$$

while the parameter c_0 is defined to meet the constraint value at the design point. The small positive number ϵ guarantees the convexity of the approximation. This approximation is a generalization of the pure Method of Moving Asymptotes of Svanberg [44], since, here, each constraint has got its own set of asymptotes.

3.9.3 Second order GCMMA

As suggested in Svanberg [47], the curvature of the GCMMA approximation can be improved by considering non mixed second order derivatives instead of the non-monotonic parameter.

$$p_{ij}^{(k)} = \frac{(U_i^{(k)} - x_i^{(k)})^3}{2(U_i^{(k)} - L_i^{(k)})} \times \left(2\frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} + (x_i^{(k)} - L_i^{(k)})\frac{\partial^2 g_j(\mathbf{x}^{(k)})}{\partial x_i^2} \right) \tag{126}$$

$$q_{ij}^{(k)} = \frac{(x_i^{(k)} - L_i^{(k)})^3}{2(U_i^{(k)} - L_i^{(k)})} \times \left(-2\frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} + (U_i^{(k)} - x_i^{(k)})\frac{\partial^2 g_j(\mathbf{x}^{(k)})}{\partial x_i^2} \right) \tag{127}$$

In the later, this approximation is called GCMMA2.

3.10 Approximation procedure of the diagonal second derivatives

Several approaches have been developed to find efficient estimation procedures of the diagonal second order terms. For example in Duysinx et al. [19], a fast estimation procedure of second order terms is derived from Thapa's theory [48] of quasi-Newton update preserving sparse patterns by particularizing it to diagonal structures. However the result is a bit disappointing from a theoretical point of view since one comes to the conclusion that the formula leads to making finite differences between the first derivatives at the current and the previous design points.

In fact the basic problem stems from the fact that there was no real mathematical (and rigorous) theory to deal with the estimation of a diagonal Hessian matrix. An important break-through was recently realized with the work of Zhu, Nazareth and Wolkowicz [52] and their theory of *quasi-Cauchy diagonal updatings*. Adaptation to structural and multidisciplinary problems has been drawn in [17].

3.10.1 Diagonal quasi-Newton updates

A study [19, 20] based on the theory of Quasi-Newton updates preserving the diagonal structure has shown that the best approximation of second order diagonal terms is simply given by:

$$B_{ii} \simeq \frac{\dfrac{\partial g(\mathbf{x^{(k)}})}{\partial x_i} - \dfrac{\partial g(\mathbf{x^{(k-1)}})}{\partial x_i}}{x_i^{(k)} - x_i^{(k-1)}} \tag{128}$$

This rather intuitive result consists in "making finite differences between the computed first derivatives in two successive iteration points and in ignoring the cross derivatives".

This theoretical result was adapted to the characteristics of structural optimization problems to yield quickly convergent estimates of the Hessian. This adaptation relies on the key role of the reciprocal design variables to reduce the non-linearity of the structural responses. The Hessian is updated in the space of reciprocal design variables and then converted into curvatures in terms of the direct variables to be used in the approximation. The initial guess of the Hessian is also very important. Starting in the reciprocal design space from a diagonal matrix of small terms restores the curvatures of CONLIN which is generally a good starting point.

As in Ref. [19, 20] this approximate second order information can be introduced in two high quality approximations: the second order Method of Moving Asymptotes and the quadratic separable approximation. The performances of this procedure are very close from the results obtained while using the same approximations with analytic second order derivatives [24]. In every cases these performances are at least as good as the best results with MMA [44] or CONLIN [27].

3.10.2 Quasi-cauchy updates

Compared to quasi-Newton updates which are quite well known in the engineering community, the main characteristics of the quasi-Cauchy theory are the following:

- The update must satisfy the quasi-Cauchy equation which is a diagonal version of the weak-quasi-Newton equation (developed in Dennis and Wolkowicz [12]) -which is itself a weak version of the well-known quasi-Newton relation-:

  $$\mathbf{s}^T \mathbf{D}_+ \mathbf{s} = \mathbf{s}^T \mathbf{y}$$

  where $\mathbf{D}_+$ is the updated approximation of the (true) Hessian matrix we look for, $\mathbf{s} = \mathbf{x}_+ - \mathbf{x}$ is the step between the two design points and $\mathbf{y} = \mathbf{g}_+ - \mathbf{g}$ is the gradient change between these two points.

- The update $\mathbf{D}_+$ is required to be *a priori* a diagonal matrix.

According to theoretical results presented in Zhu et al. [52], two quasi-Cauchy updating schemes are implemented and compared: one is based on the update of the matrix $\mathbf{D}$ itself and the other one is based on the updating of the matrix $\mathbf{D}^{1/2}$.

Updating D

Updating scheme of D is based on the variational problem:

$$\begin{array}{ll} \min & ||\mathbf{D}_+ - \mathbf{D}||_F \\ \text{s.t.} & \mathbf{s}^T \mathbf{D}_+ \mathbf{s} = \mathbf{s}^T \mathbf{y} \end{array} \tag{129}$$

where $\mathbf{s} \neq \mathbf{0}$, $\mathbf{s}^T\mathbf{y} > 0$, and $\mathbf{D} > 0$. Let $\mathbf{D}_+ = \mathbf{D} + \mathbf{\Lambda}$, $a = \mathbf{s}^T\mathbf{D}\mathbf{s}$, $b = \mathbf{s}^T\mathbf{y}$. According to [52], the solution to this problem writes:

$$\mathbf{\Lambda} = \frac{(b-a)}{\mathrm{tr}(\mathbf{E}^2)}\,\mathbf{E} \quad \text{with} \quad \mathbf{E} = \mathrm{diag}\{s_1^2, s_2^2 \dots s_n^2\} \tag{130}$$

When $b < a$, it is interesting to notice that the update matrix $\mathbf{D}_+ = \mathbf{D}+\mathbf{\Lambda}$ is not necessarily positive definite. This may be a difficulty if $\mathbf{D}$ is used within a metric-based algorithm, but when used in a structural approximation, this is not really an obstacle because one replaces the negative second order terms by a very small positive number to stay with a convex approximation.

Updating $\mathbf{D}^{1/2}$

An alternative approach is similar to the principle used to derive the BFGS update in the quasi-Newton setting. It consists in updating the square root or Cholesky factor $\mathbf{D}^{1/2}$ to give the corresponding $\mathbf{D}_+^{1/2}$. The update is defined by $\mathbf{D}_+^{1/2} = \mathbf{D}^{1/2} + \mathbf{\Omega}$. The $\mathbf{\Omega}$ update is calculated via the solution of the minimization problem:

$$\begin{array}{ll} \min & ||\mathbf{\Omega}||_F \\ \text{s.t.} & \mathbf{s}^T(\mathbf{D}^{1/2} + \mathbf{\Omega})\mathbf{s} = \mathbf{s}^T\mathbf{y} > 0 \end{array} \tag{131}$$

Let $\mathbf{D} > 0$, and $\mathbf{s} \neq 0$. There is a unique solution to the minimization problem (131) (see theorem 2.2.1 of Zhu et al. [52]) and it is given by:

$$\mathbf{\Omega} \;=\; \left\{ \begin{array}{ll} 0 & \text{if } b = a \\ \mu^* \mathbf{E}\,(\mathbf{I} + \mu^*\,\mathbf{E})^{-2}\,\mathbf{D}^{1/2} & \text{if } b \neq a \end{array} \right. \tag{132}$$

where μ^* is the largest solution of the non linear equation $F(\mu) \;=\; b$ with

$$F(\mu) \;=\; \mathbf{s}^T\left(\mathbf{D}\,(\mathbf{I} + \mu\mathbf{E})^{-2}\right)\mathbf{s} \;=\; \sum_{\{i \mid s_i \neq 0\}} \frac{d_i s_i}{(1 + \mu s_i^2)^2} \tag{133}$$

One can demonstrate that the solution $\mathbf{D}_+$ is always definite positive. However this updating scheme requires the solution of a one-dimensional non-linear equation.

The solution of the one-dimensional equation $F(\mu) \;=\; b$ is not too difficult to realize numerically for example with a bisection iteration scheme. However, one may notice that the solution of the one-dimensional equation goes into trouble when the function is non convex during the step, i.e. $b = \mathbf{s}^T\mathbf{y} < 0$. Indeed $F(\mu)$ is strictly positive on the search interval. In this case the QC update can not be applied anymore and the update procedure is not possible. In this case, our parade is to restart the update procedure.

Oren-Luenberger scaling

The simplest relation derived from QC relation is known from a long time [32]. It is the unique matrix that is obtained from QC relation with further restriction that the diagonal matrix is a scalar multiple of identity matrix $\mathbf{I}$, that is $\mathbf{D} \;=\; d\,\mathbf{I}$. So it comes that $\mathbf{s}^T\mathbf{D}\mathbf{s} \;=\; d\,\mathbf{s}^T\mathbf{s}$. QC relation allows to identify d, which leads to the Oren-Luenberger scaling matrix :

$$\mathbf{D}_+ \;=\; \frac{\mathbf{s}^T\mathbf{y}}{\mathbf{s}^T\mathbf{s}}\,\mathbf{I} \tag{134}$$

Properties of quasi-Cauchy updatings

Up to now, the second scheme (based on updating of the matrix $\mathbf{D}^{1/2}$) has shown to be better in our numerical testings, mainly because it preserves the positive definiteness of the diagonal estimate, but it also requires the solution of a one-dimensional non-linear equation. However, both quasi-Cauchy updating schemes present the following major advantages:

- They require very little storage (O(n)) so that they are very well adapted to the solution of large scale problems such as topology problems in structural optimization.
- The update provides a rigorous way, to avoid off-diagonal second order terms and to replace them by a weighted effect over the diagonal.

Nonetheless, the weakness of the theory up to now is that, to the authors' knowledge, there are no demonstrated convergence properties for quasi-Cauchy techniques like for quasi-Newton techniques.

Moreover the first conclusions drawn by study in Ref. [17] in structural optimization are quite surprising. If it was expected that exact second order sensitivities are better than quasi-Cauchy estimations, it is more surprising to see that Quasi-Newton estimation based on implementation given in Ref. [19, 20] is generally better than quasi-Cauchy updates. The surprise is that making finite differences of the first derivatives is quite competitive compared to quasi-Cauchy techniques, which have a rigorous and elaborated mathematical background. Despite what seems to be suggested in [52], updating scheme based on $\mathbf{D}$ is not less good than the updating scheme based on $\mathbf{D}^{1/2}$ in structural optimization.

3.11 Improvement of the MMA family schemes

3.11.1 GBMMA approximations

In the Globally Convergent version of MMA ([46]), each function $g_j(\mathbf{x})$ is approximated according to the following expansion:

$$\begin{aligned}\tilde{g}_j(\mathbf{x}) &= g_j(\mathbf{x}^{(k)}) + \sum_{i=1}^{n} p_{ij}^{(k)} \left(\frac{1}{U_i^{(k)} - x_i} - \frac{1}{U_i^{(k)} - x_i^{(k)}}\right) \\ &\quad + \sum_{i=1}^{n} q_{ij}^{(k)} \left(\frac{1}{x_i - L_i^{(k)}} - \frac{1}{x_i^{(k)} - L_i^{(k)}}\right) \end{aligned} \tag{135}$$

$g_j(\mathbf{x}^{(k)})$ is the function value at the current iteration k, whereas the parameters $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are computed based on the first order derivatives, on the asymptotes $L_i^{(k)}$ and $U_i^{(k)}$, and on a non monotonic parameter $\rho_j^{(k)}$. At each iteration k, the asymptotes $L_i^{(k)}$ and $U_i^{(k)}$ are updated according to a heuristic rule that is the same as for the classical MMA, while the parameter $\rho_j^{(k)}$ is updated on the basis of a rule proposed by Svanberg to ensure the globally convergent character of the approximation. If the parameters $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ in

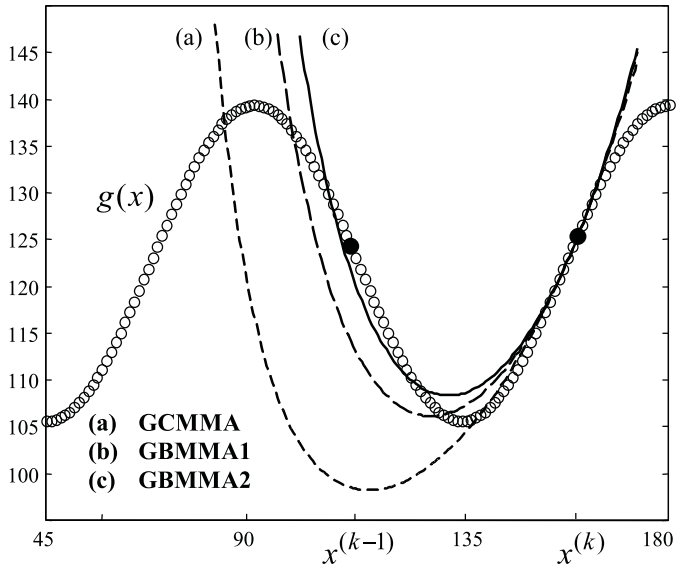

Figure 16: Non monotonous approximations around $\mathbf{x}^{(k)}$: GCMMA, GBMMA1 and GBMMA2

(135) are positive, the approximation is convex. Because of the presence of the parameter $\rho_j^{(k)}$, the approximation is non monotonous as illustrated in Fig. 16.

As shown in Bruyneel et al. [8], the original Svanberg's GCMMA scheme can be much improved when exploiting the information at previous iteration points. In the Gradient Based MMA approximation schemes (or GBMMA) proposed in Bruyneel et al. [8], the gradient information from the previous iteration $k-1$ is used in place of $\rho_j^{(k)}$ to build (135).

For GBMMA1, $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ in (135) are determined by matching the first partial derivatives at the current and previous design points. They are analytically computed from the following set of equations:

$$\begin{aligned} \frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} &= \frac{p_{ij}^{(k)}}{(U_i^{(k)} - x_i^{(k)})^2} - \frac{q_{ij}^{(k)}}{(x_i^{(k)} - L_i^{(k)})^2} \\ \frac{\partial g_j(\mathbf{x}^{(k-1)})}{\partial x_i} &= \frac{p_{ij}^{(k)}}{(U_i^{(k)} - x_i^{(k-1)})^2} - \frac{q_{ij}^{(k)}}{(x_i^{(k-1)} - L_i^{(k)})^2} \end{aligned} \tag{136}$$

In GBMMA2, the quality of the approximation (135) is improved by using an estimation of the diagonal second order derivatives (137) introduced for the first time in Duysinx et al. [19]. Determining parameters $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ of the scheme then relies on the first partial derivatives at the current design points and on the estimated second order diagonal

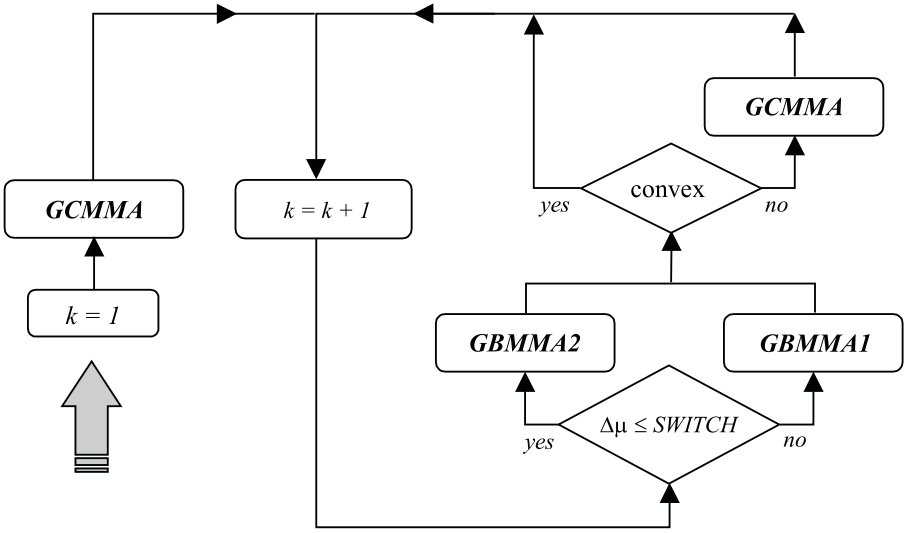

Figure 17: Selection of the non monotonous approximation based on GCMMA, GBMMA1 and GBMMA2

derivatives (137).

$$\frac{\partial^2 g_j(\mathbf{x}^{(k)})}{\partial x_i^2} \simeq \frac{\dfrac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} - \dfrac{\partial g_j(\mathbf{x}^{(k-1)})}{\partial x_i}}{x_i^{(k)} - x_i^{(k-1)}} \tag{137}$$

It was observed on numerical tests that it is interesting to use GBMMA2 when the current design point is in the vicinity of the optimum, that is at the end of the optimization process. Indeed, it make sense that in the final convergence stages, the use of second order information, even if estimated, improves the convergence speed. Based on this observation, the contribution of a given design variable x_i in a given design function $g_j(\mathbf{x})$ can be approximated by GBMMA2 when the criterion (138) is verified:

$$\frac{|x_i^{(k)} - x_i^{(k-1)}|}{\overline{x}_i - \underline{x}_i} \leq SWITCH \tag{138}$$

Otherwise, GBMMA1 is used. This leads to consider the mixed non monotonous GBMMA1 - GBMMA2 approximation, for $SWITCH \in]0, 1[$.

When $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ computed by GBMMA1 or GBMMA2 are not positive, the approximation procedure switches automatically back to a classical GCMMA to keep a convex approximation. The automatic selection of the non monotonous convex approximation based on (138) is illustrated in Fig. 17.

Monotonous approximations like MMA or CONLIN can also been recovered as special cases of the more general non monotonous approximation GCMMA. For these approximations, only one asymptote is used at a time, which means that depending on the sign of

the first derivatives, either $p_{ij}^{(k)}$ or $q_{ij}^{(k)}$ is set to zero. Furthermore, by forcing $L_i^{(k)} = 0$ and $U_i^{(k)} \to \infty$, MMA is reduced to Conlin ([27]):

3.11.2 Mixed approximations

When dealing with structural optimization problems including design variables of two different natures, for example in problems mixing sizing and shape variables or ply thickness and orientation variables in composite design, one is faced to a difficult task because of the simultaneous presence of monotonous and non-monotonous behaviors with respect to the design variables. In these conditions, most of usual approximation schemes have a poor convergence properties or even fail to solve these kinds of problems. This fact was noticed by Zhang et al. [49] for truss configuration optimization. Those authors put forward the idea that a mixed approximation scheme was interesting in this case and they developed such an approximation in which *a priori* sizing variables are approximated by a GMMA scheme, whereas a diagonal quadratic approximation is used for shape variables.

In this work, we are continuing along this idea and we propose a mixed approximation based on monotonous and non-monotonous schemes from the MMA family we presented before. Using approximations of the same family is an advantage for the numerical implementation. We also provide a strategy to select the monotonous or the non-monotonous approximation schemes for each variable in each function. This insures the efficiency and the robustness of the procedure.

In order to generate the most general mixed approximation scheme of the MMA family, GBMMA and GMMA are combining in (139)

$$\begin{aligned}
\tilde{g}_j(\mathbf{X}) \;=\; & g_j(\mathbf{x}^{(k)}) + \sum_{i\in A}^{n} p_{ij}^{(k)} \left(\frac{1}{U_{ij}^{(k)} - x_i} - \frac{1}{U_{ij}^{(k)} - x_i^{(k)}}\right) \\
& + \sum_{i\in A}^{n} q_{ij}^{(k)} \left(\frac{1}{x_i - L_{ij}^{(k)}} - \frac{1}{x_i^{(k)} - L_{ij}^{(k)}}\right) \\
& + \sum_{+,i\in B}^{n} p_{ij}^{(k)} \left(\frac{1}{U_{ij}^{(k)} - x_i} - \frac{1}{U_{ij}^{(k)} - x_i^{(k)}}\right) \\
& + \sum_{-,i\in B}^{n} q_{ij}^{(k)} \left(\frac{1}{x_i - L_{ij}^{(k)}} - \frac{1}{x_i^{(k)} - L_{ij}^{(k)}}\right)
\end{aligned} \tag{139}$$

In this new GBMMA-GMMA approximation, the design variables are partitioned into two sets, namely A and B. For the design variables belonging to the set A the GBMMA scheme (135) is used, which introduces a non-monotonous contribution, whereas for variables from set B, a GMMA scheme (124) is considered, which gives rise to monotonous terms in the approximation.

At any stage $k \geq 2$ of the iterative optimization process, an automatic strategy selects the partition of the design variables between the two sets A and B. The tests are based on the gradient values at two successive iterations, or more exactly on the variation of the

sign of the first derivatives between the two design points. For each structural response $g_j(\mathbf{x})$ and each design variable x_i, one performs the following tests:

$$\frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} \times \frac{\partial g_j(\mathbf{x}^{(k-1)})}{\partial x_i} > 0 \quad \Rightarrow \quad \text{GMMA} \tag{140}$$

$$\frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} \times \frac{\partial g_j(\mathbf{x}^{(k-1)})}{\partial x_i} < 0 \quad \Rightarrow \quad \text{GBMMA} \tag{141}$$

$$\frac{\partial g_j(\mathbf{x}^{(k)})}{\partial x_i} - \frac{\partial g_j(\mathbf{x}^{(k-1)})}{\partial x_i} = 0 \quad \Rightarrow \quad \text{LINEAR} \tag{142}$$

As suggested for the first time in Bruyneel and Fleury [9], tests (140) to (142) are performed on a given number of iterations defined by the user parameter ICHECK to be sure to capture the true structural behaviors. During this checking phase, the use of monotonous approximations is forbidden for avoiding the risk of approximating a non-monotonous function with a monotonous one. This is illustrated in Fig. 18: between the iterations $k-2$ and $k-1$, the non-monotonous GBMMA3 scheme is selected based on the test (141). If the detected non-monotonous behavior is not stored, a monotonous approximation could be built at the next step according to the relation (140), because the gradients in $\theta^{(k-1)}$ and $\theta^{(k)}$ are now of the same sign. This would reject the new design point $\theta^{(k)\star}$ far from the current one and could slow down the overall optimization process.

If the test (140) is verified during the ICHECK iterations, the behavior of the function $g_j(\mathbf{x})$ is considered to be monotonous with respect to x_i. This variable is then associated to the set B in (139) and its contribution in $g_j(\mathbf{x})$ is given by a monotonous approximation.

In practice, the choice of the value for the ICHECK parameter results from a compromise between security and speed: if this value is low, a non-monotonous structural behavior could be approximated using a monotonous approximation, and if its value is high, one then works mainly with non-monotonous (and perhaps too conservative) approximations. A typical value for ICHECK is 2.

A simpler scheme combining the GBMMA and the MMA approximations (called GBMMA -MMA) can be derived from (139). One needs just to define one set of lower asymptotes $L_i^{(k)}$ and one set of upper asymptotes $U_i^{(k)}$ for all the functions instead of $L_{ij}^{(k)}$ and $U_{ij}^{(k)}$. In this approximation scheme, the fitting procedure (83-84) is replaced by the simpler update procedure (80-82), so the function value at the previous iteration is then not used anymore. This 'less expensive' scheme was used by [9] for the optimization of laminates over plies thickness and fibres

4 COMPARISON OF FIRST AND SECOND ORDER SCHEMES

4.1 Comparison of first and second order approximations in topology optimization [13]

First order convex approximations

In Refs. [13, 14], we tested first order approximations in solving material distribution problems and we observed good results. These approximations give rise to quick descent

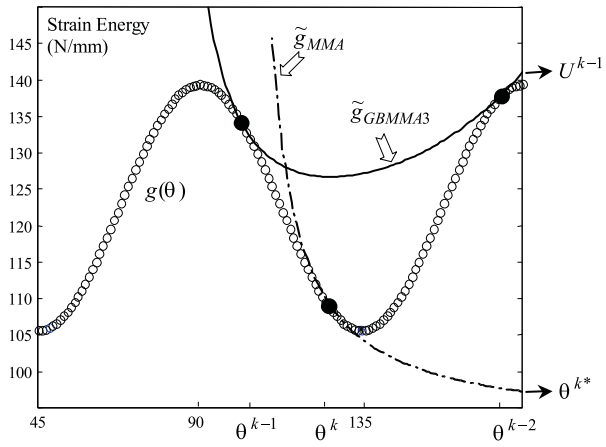

Figure 18: Selection of monotonous or non-monotonous approximations in a mixed scheme

rate during the first 10 iterations, but the convergence is slower around the optimum, so that 20 to 40 iterations more are generally necessary to arrive to a stationary solution. Nevertheless, if these performances (in terms of the number of iterations) are compared to results obtained with standard implementations of optimality criteria (like in Ref. [3, 5]), CONLIN or MMA give equal or better performances. Among the approximations, the expansion in terms of power p of the design variables is often too conservative and the convergence rate is the slowest. Performance of CONLIN is generally very satisfactory. Due to the weaker curvatures of CONLIN and MMA in the first iterations, it is worth using move-limits to have a smooth convergence history.

From our experiences, the CONLIN (Fleury and Braibant [27]) approximations give rise to good results in topology design. For the compliances that are self-adjoint, all the derivatives are negative and CONLIN restores the reciprocal design variables expansion that is well known to reduce the non-linearity of the structural responses. But convexity and conservativity properties of the approximation in CONLIN are important when treating eigenfrequencies or constraints whose first derivatives have mixed signs. The main disadvantage of CONLIN is that the approximations introduce fixed curvatures, so that the approximation may be too much or too little conservative. This can give rise to a slow or unstable convergence towards the optimum. To remedy this, we select the MMA [44] approximation scheme which generalizes and improves the CONLIN scheme by introducing two sets of asymptotes. The choice of the moving asymptotes provides the way to modify the curvature and to fit better to the characteristics of the problem.

Because of the flexibility introduced by the moving asymptotes, MMA fits better to the convexity of the problem and MMA is often a bit quicker than CONLIN. It was also observed that for very difficult problems, MMA was more stable than CONLIN. Therefore it is generally chosen as default algorithm.

Nevertheless, we can conclude that both CONLIN and MMA lead to satisfactory results for topology design and improve often greatly the performance of the solution procedure. In many problems we observed that a solution is often achieved in 30 to 50 iterations depending on the difficulty of the problem and the precision of the stopping criterion. One strong advantage of CONLIN and MMA arises from the very reliable dual solvers that are used to solve the associated convex sub-problems. On another hand, one major drawback of first order approximations is that we can observe a deceleration of the progression towards the optimum once the algorithm arrives in the neighborhood of the optimum. To accelerate the convergence rate in the final stage, one needs better approximations based on curvature information (Fleury [24]).

We should also briefly discuss the selection of a termination criteria. We prefer not using termination criteria based on limited improvement of the objective function since the optimum is very flat and the value of the objective function decreases very slowly during more than half of the iterations. So, such a method will lead termination of the optimization process at a too early stage. When all is said and done, one looks for the optimal material distribution, so we think that it is better to adopt a termination criteria based on the design variable motion. This can be based on any norm of the difference of the design variable vectors between two iterations. We often use arithmetic mean modification (order 1 norm) or the maximum modification (infinite norm). As the problem under consideration is a constrained optimization problem, one can also use Kuhn-Tucker conditions. Satisfaction of any one of these last termination criteria avoids premature stopping.

Second order convex approximations [14]

Second order are high quality approximations that are indeed more precise and that lead to faster convergence rates. Nevertheless, second order sensitivity is very onerous to compute and to store so that the overall cost of the optimization run with second order sensitivity can be similar to the cost of an optimization run that would be made with first order approximations even if the number of iteration is greater [34]. When the size of the problem increases, computing and storing second order derivatives becomes quickly cumbersome and the problem becomes impossible to manage.

To be able to use second order approximation schemes with large scale optimization problems, we developed a new procedure to generate an estimation of the curvature information with a small computation cost [19, 20]. As separable approximations needs only diagonal second derivatives, the idea is to built an estimation of the curvature information with a quasi-Newton update able to preserve diagonal structure of the Hessian estimates. This update scheme is derived form the general theory of quasi-Newton update with sparse Hessian estimates made by Thapa [48]. The diagonal version of the BFGS update [19, 20] that we implemented is very un-expensive even for large scale problems since it introduces only vectors manipulations. In [13], it was observed that for a given topology problem, the time spent in the diagonal BFGS update is only 3 % of the time spent in the optimizer CONLIN [23, 25] and only 0.01 % of the time needed for sensitivity analysis with a commercial finite element package.

The estimated second order information is introduced into two well known second order

approximations. The first one is a second order version of MMA proposed by Smaoui et al. [42]. The second approximation is the separable quadratic approximation suggested by Fleury [24]. Combining diagonal BFGS update with both these approximations gives very interesting results that leads to important savings in terms of number of iterations and of computation time. This conclusion can be explained as follow. Firstly, the estimation of the curvature improves greatly the quality of the approximation with only the help of the accumulated first order information. Secondly, instead of ignoring the second order coupling terms, diagonal BFGS provides a way to take them into account by correction terms on the diagonal coming from the diagonal update. Due to our initial guess of the Hessian, one can observe, in the first iterations, a convergence history that is very similar to first order approximations. But after some iterations, the update procedure improves the estimation of the Hessian and one can see a real advantage in the convergence speed. Around an accumulation point satisfying the optimality conditions, we could observe a convergence speed superior to first order methods, sometimes closed from super-linear behaviour.

As a conclusion, second order approximations can advantageously be used for large scale optimization problems like topology design. Starting from an initial choice of curvatures which is close to the reciprocal design variable expansion results in similar characteristics as first order approximations, during the first iterations. This choice generally yields a good descent rate of the objective function in the beginning. Then, since the Hessian estimate is improved and the approximation is enriched by this curvature information, it leads to a better convergence rate during final convergence and the number of iterations to reach a stationary solution is reduced.

Attention must nevertheless be paid to the well-known fact that second order algorithms are more sensitive to local optima. This fact was observed also with our procedure and particularly with the quadratic separable approximation. This drawback can be attenuated by adding move-limits or by adding additional convex terms in the quadratic approximation.

4.2 Topological optimization of short cantilever beam

Finally we can illustrate the application of four approximations -CONLIN, MMA, MMA second order with diagonal BFGS (DQNMMA), and quadratic separable approximation with diagonal BFGS (DQNQUA)-, in the context of a topology optimization problem, the short cantilever beam problem whose geometry of the problem is given at figure 19. This design is a classical bench-mark of topology optimization. For the sake of simplicity, the material law is simply given by a cubic relation between the rigidity and the relative density [4]: $E = \mu^3 E_0$ and $\rho = \mu\rho_0$. The Poisson's ratio and the Young's modules of the solid are: $\nu_0 = 0.3$ and $E_0 = 100\, GPa$. The compliance under the given load case is minimized while the volume is bounded to 37.5% of the volume of the design domain. The design domain is discretized by a regular mesh of 1040 finite elements of degree 2. The finite element analysis and the sensitivity analysis are performed with the SAMCEF package.

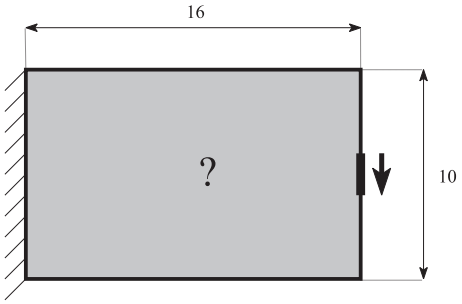

Figure 19: Short cantilever beam problem

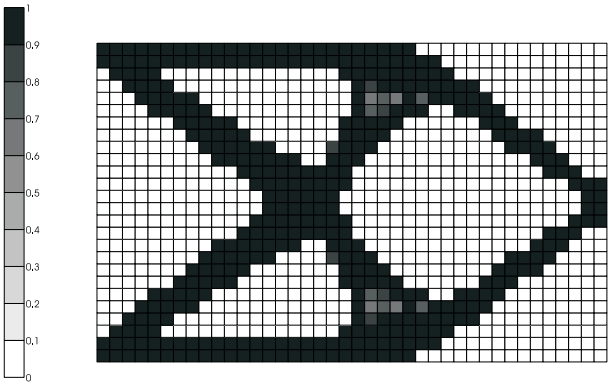

Figure 20: Distribution of density (CONLIN)

The problem is solved with the four different approximations for compliance (the volume is linearized). Since all the first derivatives of compliance are negative, CONLIN and the reciprocal variables expansion are the same. We also implemented a MMA scheme similar to Svanberg's one. Then, we use the high quality approximations DQNMMA and DQNQUA.

Histories of compliance are given at figure 21. At first glance, the different convergence curves are very similar and the different algorithms tend towards local optima with nearly the same compliances. Nevertheless, when material distributions are visualized, the optimal distributions reveal a very similar topology, except in the quadratic approximation (Figure 20 presents the material distribution obtained with CONLIN). Thus several optima exist when intermediate densities are highly penalized and attention must be paid to local configurations. Also, second order approximations are partly more sensitive to local optima.

First order approximations give smooth and monotone history curves. At the beginning, the descent rate is good but, after more or less 10 iterations, compliance reduction is seriously slowed down when close to the optimum. Progression becomes much slower.

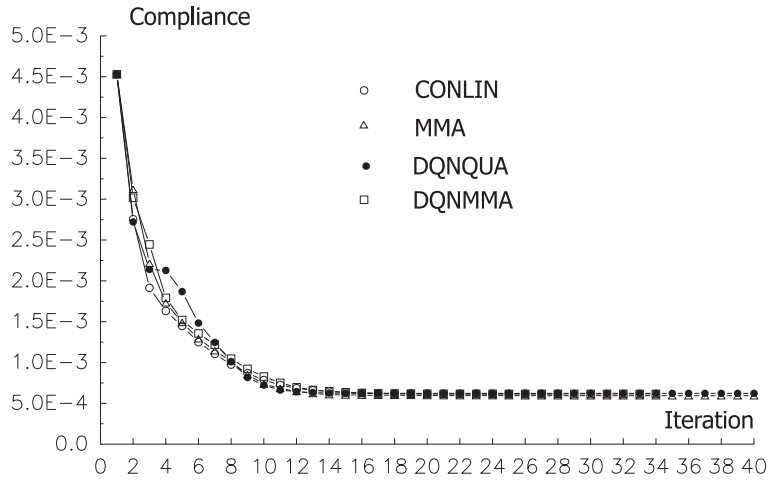

Figure 21: Compliance history

One can note that the modulus of the KKT test (figure 22) and the mean modification of the design variables between two iterations (figure 23) diminish slowly, even with small oscillations.

Second order approximations also results in convergence curves with a very good descent rate. The progression towards the optimum is not handicapped too much by the non monotone behaviour of the first iterations (iterations 3 and 4). That fact may be due to an uncertain value of the estimation of the curvatures by the diagonal BFGS. After this phase which is necessary to stabilize the estimations of the curvatures, second order information gives rise to a very good descent rate. When in the flat part of the compliance curve, second order information preserves a good convergence speed and continues to accelerate the progression towards a stationary solution. This can be clearly noted on the mean modifications of the design variables or on the KKT modulus. Stationarity is reached much faster with second order schemes and diagonal BFGS than with first order approximations. As predicted by Thapa's theory [48], we recover in fact the asymptotic superlinear descent rate of quasi-Newton methods in the neighborhood of the optimum. This characteristic saves a high number of iterations. The quadratic scheme and the MMA second order method come to stationarity in 30 to 40 iterations while CONLIN and MMA needs more than 40 iterations to satisfy a weaker termination criteria.

Finally one can also compare the two proposed termination criteria: the modulus of the Lagrangian function (in figure 22) and the mean modification of the design variables between two iterations (in figure 23). One can observe that both have a parallel evolution and they are in perfect correlation. As predicted by the theory they give an equivalent information and they be both used to predict convergence in topology optimization problems.

With the DQNMMA and DQNQUA approximations, the objective function is com-

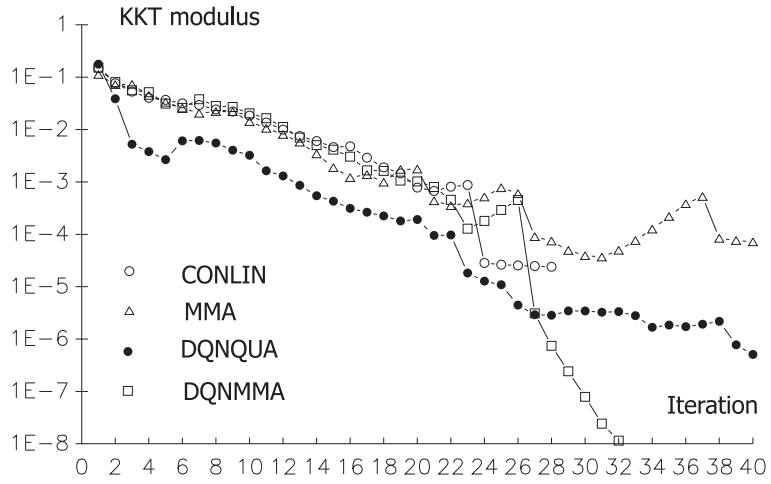

Figure 22: History of modulus of KKT vector

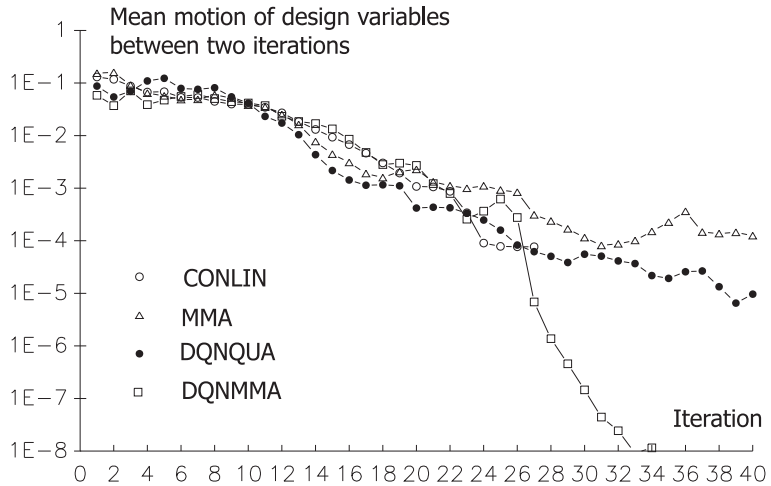

Figure 23: History of the mean motion of the design variables between two successive iterations

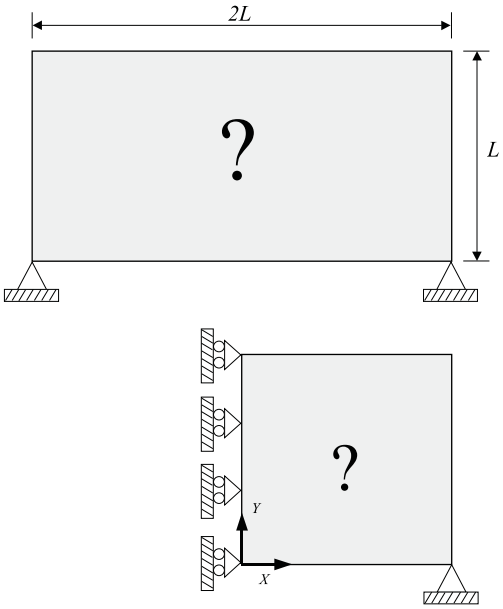

Figure 24: Design domain and supports.

pletely stationary after 26 iterations and 40 iterations. At this moment, no design modifications can be observed, whereas the convergence of the other first order schemes is not achieved yet. CONLIN and MMA's final convergence rates are much slower.

4.3 Advantage of using advanced MMA schemes in topology optimization [7]

The test case illustrated in Fig. 24 consists in designing a structure that relies on two supports, while supporting its own weight. A non structural mass is placed on is placed at the top to load the structure. Intuitively an arch type structure is expected (see figure 25 the solution obtained with GCMMA). The reference length L is $L = 1m$. Due to symmetry conditions, only one half of the design domain is studied and is discretized with 20×20 4-node quadrangular finite elements of 8 degrees of freedom. The mechanical properties of the base material to be distributed in the domain are: $E_o = 1N/m^2$, $\nu = 0.3$ and $\rho_o = 1kg/m^3$, while the gravitational acceleration a_g is $9.81kgm/s^2$. The exponent p in SIMP interpolation is equal to 2. The maximum available amount of material $\overline{V}$ at the solution is 80%, while the minimum amount of material $\underline{V}$ is set to 1%.

The stopping criteria adopted is based on the maximum variation of the design variables over two design steps where $TOL = 0.0001$:

$$\max_{i=1...n} |\mu_i^{(k)} - \mu_i^{(k-1)}| \leq TOL \tag{143}$$

At first we can observe that when CONLIN is used, no optimal topology can be obtained. The topology changes from one iteration to an other (Figs. 26 and 27) and there

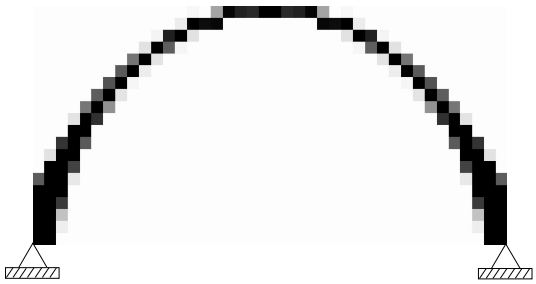

Figure 25: Optimal topology for the arch problem when using GCMMA and the modified SIMP law

Figure 26: Topology obtained by CONLIN at iteration 199

are oscillations of the design variables during the optimization process. Such a monotonous approximation is definitively not efficient for solving this non monotonous problem.

For the other approximations: MMA, GCMMA, GBMMA1, GBMMA2 and GBMMA1-GBMMA2, a solution can always be reached (similar to Fig. 25). Although MMA gives rise to monotonous approximations of the design functions, it is able to come to an optimal topology, thanks to a robust move-limits strategy suggested by [44]. However, as reported in Table 1 for different values of the precision TOL in (143), MMA requires a lot of iterations: twice more than GCMMA and nearly four times more than the best GBMMA. For self-weight problems, the non monotonous approximations are obviously much more efficient, especially when gradients from the previous iteration are used as in GB-

Figure 27: Topology obtained by CONLIN at iteration 200

Table 1: Number of iterations needed for solving the arch problem under selfweight for different values of TOL in (143). GBMMA1-GBMMA2 with automatic selection strategy of Fig. 17($SWITCH = 0.2$)

Approximations	0.01	0.001	0.0001
MMA	130	402	438
GCMMA	80	200	253
GBMMA1	51	98	130
GBMMA2	73	109	139
GBMMA1-GBMMA2	54	91	112

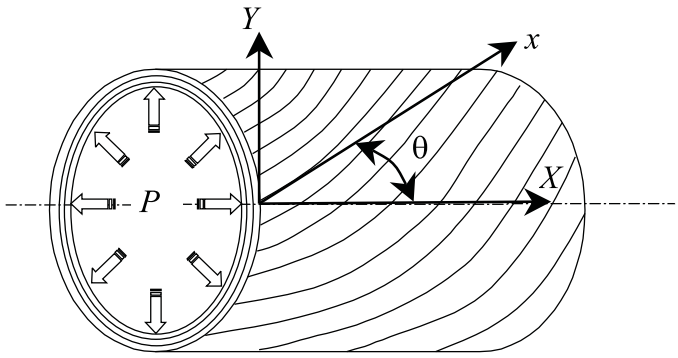

Figure 28: Composite cylinder problem

MMA approximations. According to the results of Table 1, GBMMA is always faster than GCMMA. The best results are obtained with the automatic strategy combining GBMMA1 and GBMMA2. In this case, the mixed GBMMA1-GBMMA2 scheme is nearly twice faster than GCMMA.

4.4 Advantage of using advanced MMA schemes in composite structure optimization [8]

This numerical application will demonstrate the efficiency of the novel GCMMA family approximation with respect to the original MMA one. The design problem is a composite optimization including both thickness and orientation variables.

The design of a closed composite cylindrical container (see Fig. 28) subject to an internal

Table 2: Starting point for the composite cylinder optimization problem

Initial Orientations $(\theta_1, \theta_2, \theta_3, \theta_4)$ $(\theta_5, \theta_6, \theta_7, \theta_8)$ in degrees	Initial thicknesses (t_1, t_2, t_3, t_4) (t_5, t_6, t_7, t_8) in mm
$(0°, 135°, 45°, 90°)$	$(0.1; 0.2; 0.1; 0.2)$
$(90°, 45°, 135°, 0°)$	$(0.1; 0.2; 0.1; 0.2)$

pressure of $P = 10$ bars is considered (144).

$$\begin{array}{lll} \min & \frac{1}{2}\varepsilon^T \mathbf{A}\varepsilon & \\ \theta, \mathbf{t} & & \\ \text{s.t.:} & \mathrm{TW}(\theta_i, t_i) \ \leq \ 1 & i = 1 \ldots 8 \\ & \sum_{i=1}^{8} 2\, t_i \ \leq \ 10\text{mm} & \\ & 0° \ \leq \theta_i \ \leq \ 180° & i = 1 \ldots 8 \\ & 0 \ < \ t_i \ \leq \ 10\,\text{mm} & i = 1 \ldots 8 \end{array} \tag{144}$$

The radius of this thin-walled cylinder is 1 meter. The initial design is given in Table 2.

The number of structural analyses performed to reach a feasible local optimum are given in Table 3 when different types of approximations are used. Two differents values of the stopping precision parameter TOL are used in criterion

$$\left| \frac{g_0(\mathbf{x}^{(k)}) - g_0(\mathbf{x}^{(k-1)})}{g_0(\mathbf{x}^{(k-1)})} \right| \ < \ TOL$$

In Figs. 29 and 30, convergence curves are provided for MMA and GBMMA-GMMA. In these figures, the value of the constraints is normalized so that they are violated when they take a value larger than unity. The maximum violation at each iteration is plotted.

When MMA is used, large oscillations are observed for the successive fibres orientations values (Fig. 29). This is due to the bad approximation of the structural responses in terms of those design variables. GCMMA2 is quite slow (Table 3) because it is degenerated to the first order approximation when second derivatives are negative.

From Table 3, it is clear that resorting to Gradient Based MMA approximations and using information from previous design point can bring a major reduction of the number of iterations in this kind of composite design. Moreover, the mixed schemes can even further improve the convergence speed. From Fig. 30, one can see that the mixed GBMMA-GMMA scheme leads to monotonous convergence curves (in terms of the objective function as well as in terms of the design variables evolutions).

5 PERIMETER APPROXIMATION [14]

Perimeter constraint has a major place in topology design, because it is a very interesting alternative to optimal relaxation using optimal microstructures as in the homogenization

Table 3: Iterations versus approximation type for the optimization of the composite cylinder

Approximation	Number of iterations	
	$TOL = 0.05$	$TOL = 0.01$
MMA	29	51
GCMMA	16	18
GCMMA2	19	24
GBMMA1	9	9
GBMMA2	9	9
GBMMA3	8	9
GBMMA4	8	9
GBMMA-MMA	6	7
GBMMA-GMMA	6	7

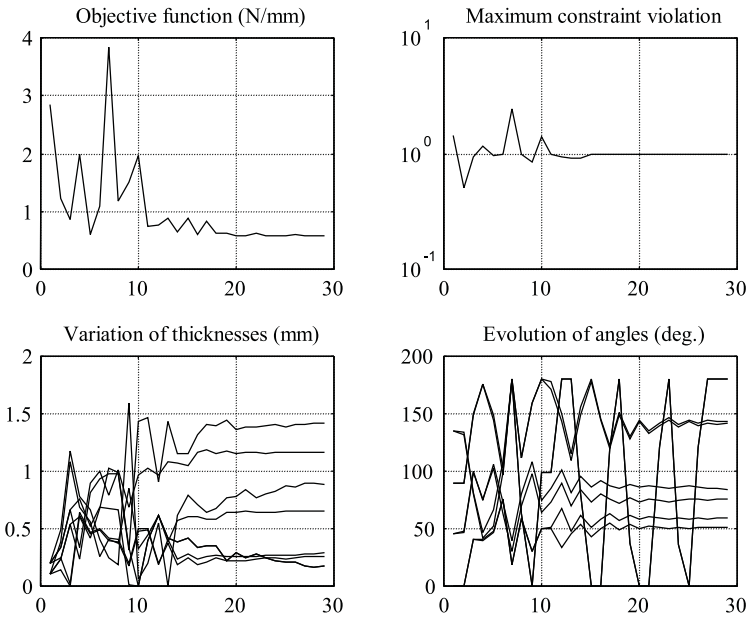

Figure 29: Iteration history for MMA

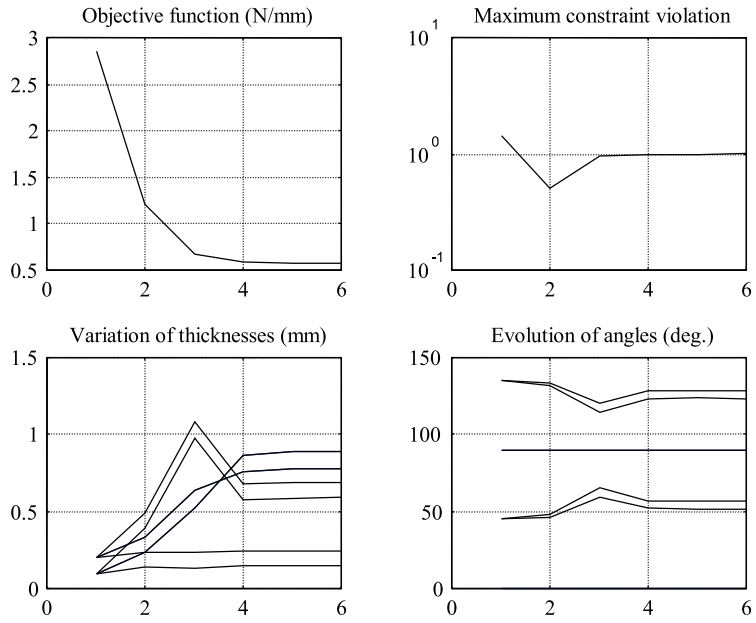

Figure 30: Iteration history for GBMMA-GMMA (ICHECK =2, SWITCH = 0.01)

method [1, 5]. In engineering applications, perimeter control also seems further attractive than the rigorous homogenization method because it allows to use penalization of intermediate densities and to generate clear density distributions with well separated voids and solids zones, so that an unambiguous macroscopic topology often appear. Ambrosio and Buttazzo [2] demonstrated that the design problem with perimeter penalization is well-posed. Application of perimeter control to topology design of structures was presented by Haber et al. [28]. The work presented here aims at remedying some difficulties that occur in the numerical implementations of the perimeter control. We focus on providing an efficient numerical strategy to take perimeter constraint into account in order to use perimeter control as a real practical design tool. The perimeter is a rather difficult constraint to satisfy and to approximate, as it will be seen.

We based our numerical experiences on power law models (also called SIMP materials) to provide in the same time a continuous approximation of the design distribution problem as well as a penalization of the intermediate densities. SIMP material is also easy to implement in any industrial code. Finally, SIMP material introduces only one variable per element, so that the size of the problem is kept at a minimum.

Although in the original work of Haber et al. [28] the perimeter control was treated with a interior penalty function, we propose to generalize the solution by solving the problem as a constrained problem in which perimeter is one of the inequality constraints. If one wants to control the perimeter by prescribing a target value with a penalization, one can use a relaxation technique and introduce an additional variable δ, which is quadratically penalized in the objective function (the *pds* factor is a tuning parameter to control the

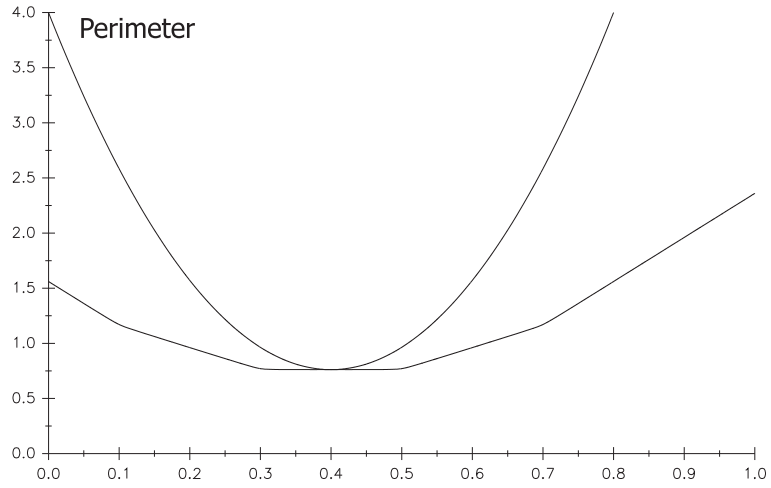

Figure 31: Example of a perimeter measure and its quadratic approximation

relative weight of the penalization compared to the magnitude of the objective function).

$$\begin{array}{rl} \min & \mathbf{f}^T\,\mathbf{u} \;+\; pds\,\delta^2 \\ \underline{\mu} \le \mu \le 1 & \\ 1 \le \delta \le 2 & \\ \text{s.t.} & V \;\le\; \bar{V} \\ & P \;\le\; \bar{P} \;+\; (\delta - 1)\,\Delta P \end{array} \tag{145}$$

ΔP is an allowable maximum violation of the target bound $\bar{P}$ give by the user. Standard versions of solvers like CONLIN [26] or MMA [44] are able to handle the solution of this kind of optimization problems with relaxation.

When the density varies continuously, Haber, et al. [28] propose to replace the geometrical measure of the perimeter by the total variation of the density ρ. For a density field which is element by element constant, we can write:

$$P \;=\; \sum_k \; l_k \; \left(\sqrt{<\rho>_k^2 + \epsilon^2} \;-\; \epsilon \right) \tag{146}$$

where $<\rho>_k$ is the jump of material density trough the element interface k of length l_k. The parameter ϵ is a small positive number to guarantee the differentiability of the measure. Values of ϵ are generally taken between 10^{-2} to 10^{-4}.

Figure 31 sketches the perimeter measure of a square element of density μ surrounded by four elements of density μ_1, μ_2, μ_3, and μ_4. Perimeter is nearly a piecewise linear function even if it is globally non monotonous. It turns out that perimeter constraint is not easy to approximate with classical schemes. Monotonous approximations like CONLIN or MMA give rise to oscillatory behaviours. Furthermore, the perimeter is globally non linear and

there are important couplings between neighboring finite element (F.E.) densities. Thus, the trust region of separable approximations is narrow.

Nevertheless, in order to treat problems with a large number of variables and to use dual solvers, we need a convex and separable approximation. From our numerical experience, good results are expected with a quadratic separable approximation of the general form:

$$\tilde{P}(\mu) = P(\mu_0) \; + \; \sum_{i=1}^{n} \frac{\partial P}{\partial \mu_i}(\mu_i - \mu_i^0) \; + \; \frac{1}{2} \sum_{i=1}^{n} a_i \, (\mu_i - \mu_i^0)^2 \tag{147}$$

The main problem is now to choose the second order terms a_i carefully. Their values must be a compromise between precision and conservativity. Too small values of a_i would imply important constraint violations while too large values of these second order terms would lead to a freeze the motion of the variables. On one hand, by selecting a_i, one will try to fit the true shape of the constraint while on the other hand, these coefficients will play the role of move-limits that restrict the validity of the approximation. Also, the analytic second order derivatives are not useful since they are zero, except in angular points where they are very large (or do not exist). So the choice of the artificial curvatures is based on a heuristic rule, which is explained in the next section.

5.1 A heuristic estimation of curvatures for perimeter approximation

In the following, we develop a heuristic estimation of curvatures for approximating the perimeter when there is one density variable μ_i per element as it is for SIMP materials. These estimates of the curvatures are based on a bound over individual contributions of each element. According to the quadratic approximation, the contribution of element "i" to global perimeter is:

$$\tilde{P}_i(\mu) = P_i(\mu_i^0) \; + \; \frac{\partial P}{\partial \mu_i}(\mu_i - \mu_i^0) \; + \; \frac{1}{2} \, a_i \, (\mu_i - \mu_i^0)^2 \tag{148}$$

$P_i(\mu_i^0)$ is the contribution of element "i" to the perimeter with the current distribution of density. This contribution of element "i" is maximum when the density jump across the element interfaces becomes equal to unity. Suppose now that this situation happens at point $\mu_i^\star$. Then one can write the second order coefficients in term of the new parameter :

$$a_i \; = \; 2 \, \frac{\sum_{k \, \in \, K_i} l_k \, (\sqrt{1+\epsilon^2} \; - \; \sqrt{|\mu_i^0 - \mu_k^0|^2 + \epsilon^2}) \; - \; \frac{\partial P}{\partial \mu_i}(\mu_i^\star - \mu_i^0)}{(\mu_i^\star \; - \; \mu_i^0)^2} \tag{149}$$

where the sum is realized over the set K_i of the interfaces of element "i".

The question is now to find the point $\mu_i^\star$ where this situation could probably happen. If the separability hypothesis were true, the point $\mu_i^\star$ could be chosen at the boundary of the admissible set of μ_i, that is when μ_i touch its side-constraints. But, because of the neglected coupling effects, which lead to ignore the modification of neighboring element densities, this situation happens sooner, so that this choice leads to approximations that

are not conservative enough. Instead, we propose to play with the point $\mu_i^\star$ as a move limit. From our numerical experience, we propose to take:

$$\mu_i^\star \;=\; \mu_i^0 \;\pm\; \alpha\,(\overline{\mu}_i - \underline{\mu}_i) \quad \text{with} \quad \alpha \;\in\; [0.33, 0.44] \tag{150}$$

This choice prevents an oscillation of the neighboring design variables. The proposed approximation as applied is illustrated in figure 31.

5.2 An internal loop procedure for perimeter approximation

Even if the approximation procedure of the perimeter is efficient, the perimeter constraint remains difficult to approximate and we generally note that the number of iterations increases drastically when a perimeter constraint is considered. It often takes more than 100 iterations. This effect can be imparted to the fact that we need to take a convex approximation with a high curvature to have a sufficiently conservative approximation. Unfortunately, this has the drawback of slowing down the optimization process. On the other hand, perimeter constraint is a geometrical constraint, and, thus, contrary to structural responses, perimeter is easy and inexpensive to evaluate.

The idea is thus to create an internal loop over the optimization algorithm with several updates of geometrical constraints as the perimeter. The strategy is given at figure 32. The outer loop is usual: it includes the finite element analysis, the sensitivity analysis and the optimization procedure. The optimization procedure encloses an inner loop that is repeated with updated approximations of the perimeter until the perimeter approximations coincide with its real calculated value within a sufficiently high precision at the proposed new optimum. Since structural constraints are expensive to evaluate (it requires one finite element run and a sensitivity analysis) and since they are sufficiently well approximated by high quality schemes, the structural approximations are kept unchanged during this inner loop. As the inner loop is repeated until the perimeter has a given precision, the perimeter constraint does not slow the optimization process and we noted a spectacular acceleration to reach optimal solutions. The number of iterations is often divided by a factor 3 or 4.

Let denote by l the sub-iteration index. To implement the update procedure of the approximations for geometrical constraints, we need writing, around the reference design point $\mathbf{x}^0$ where the structural approximations are expressed, an approximation of the perimeter that matches the true perimeter value and its derivatives in an other sub-iteration point $\mathbf{x}^l$. This is possible by defining fictitious parameters (that are denoted with a 'check').

For a quadratic approximation given in (122), one can define the dummy parameters:

$$\check{a}_i \;=\; a_{ii}(\mathbf{x^l}) \qquad \check{b}_i \;=\; \frac{\partial g(\mathbf{x}^l)}{\partial x_i} \;+\; a_{ii}\,(x_i^0 - x_i^l)$$
$$\check{g}(\mathbf{x}^0) = g(\mathbf{x}^l) \;+\; \sum_{i=1}^{n} \frac{\partial g(\mathbf{x}^l)}{\partial x_i}(x_i^0 - x_i^l) \;+\; \frac{1}{2}\sum_{i=1}^{n} a_i\,(x_i - x_i^l)^2$$

so that the approximation is exact in $\mathbf{x}^l$, but it is written around $\mathbf{x}^0$.

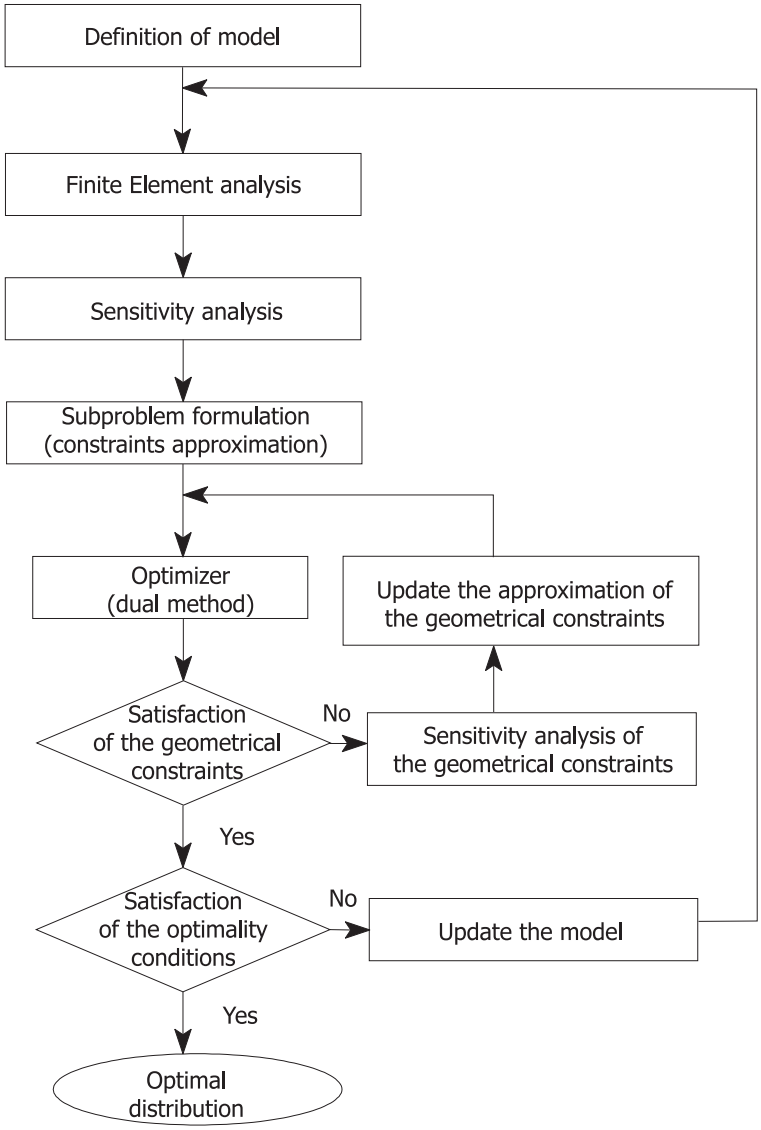

Figure 32: Optimization process with 2 loops

The quality of the approximation curvatures is also improved with the information collected during the inner loop. If the variable "i" tends to oscillate, the second order terms a_i can be multiplied by a given factor to increase the conservativity. If the process is monotone in this variable, the convexity is decreased. We adopt a similar procedure to the update strategy of the moving asymptotes of Svanberg [44].

For the two first iterations $l = 1$ and 2, the default heuristic quadratic coefficients a_i (149) are adopted. After two iterations, if the process oscillates, one has to increase the curvature of the approximation :

$$\text{if } (x_i^{l-1} - x_i^{l-2})(x_i^l - x_i^{l-1}) \leq 0 \quad a_i^l \;=\; a_i^{l-1} \cdot s_1 \quad s_1 > 1 \tag{151}$$

If the process is stable and monotone, the approximation curvature is decreased:

$$\text{if } (x_i^{l-1} - x_i^{l-2})(x_i^l - x_i^{l-1}) > 0 \quad a_i^l \;=\; a_i^{l-1}/s_2 \quad s_2 > 1 \tag{152}$$

Parameter s_2 is generally chosen equal to s_1 or better to $\sqrt{s_1}$ to stabilize the process.

The use of an accurate approximation of perimeter constraints combined with the inner loop strategy proved its great efficiency. This procedure generally leads to optimal distributions in less than 50 iterations and the optimization process is stable and reliable.

6 MANAGING STRESS CONSTRAINTS

6.1 Relaxation of stress constraint

6.1.1 Singularity of stress constraint

When including stress constraints in topology optimization, a major difficulty comes from the so-called 'singularity phenomenon' (see for example Kirch [30]). It results in the impossibility for the optimization algorithms to create or to remove holes in the material distribution during the optimization process. The physics of the phenomenon is now understood [10]: Low density regions sometimes remain highly strained. When the density decreases to zero in these regions, the limit of the stress state in the microstructure tends to a non-zero value and remains even higher than the stress limit. Therefore, the optimization procedure cannot remove the material in this region. The paradox is, that if the material is totally removed, the stress constraint would obviously not be active. This discontinuity in the stress constraint at zero density is the origin of the problem.

As remarked by Rozvany and Birker [37], these discontinuities create complex design domains: They can have several non connected parts and they often include regions of zero measure i.e. parts whose dimensionality is smaller than the dimensionality of the design space.

From a mathematical point of view, the 'singularity' phenomenon for topology design with stress constraints should rather be called a 'degeneracy' or a 'irregularity' of the design space since the key effect is that the design space contains degenerated appendices where the qualification of constraints (the Slater condition) is not verified. This means that

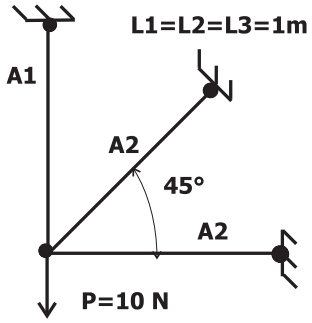

Figure 33: 3-bar truss problem

classical optimization algorithms based on Kuhn-Tucker conditions are unable to reach the optima that are located in these regions. It follows that the optimization algorithm is not able to remove totally some low density regions and then to reach the true optimal topologies.

Singularity phenomenon is illustrated by a very simple truss example reported by Hoback [29] (see figure 33). The problem is a 3-bar-truss. Cross sections of bars number 2 and 3 are the same. The weight is minimized while the stress in the three bars are kept below a prescribed stress limit of 20 N/m^2. Design problem is given as:

$$\begin{aligned} \min_{A_1,\ A_2 \geq 0} \quad & W_T \ = \ \alpha\, A_1 \ + \ A_2 \\ \text{s.t.} \quad & g_1 = A_1 + A_2/3 - 0.5 \leq 0 \\ & g_2 = A_1 + A_2/3 - 0.236 \leq 0 \\ & g_3 = A_1 + A_2/3 - 0.167 \leq 0 \end{aligned}$$

where α is a cost parameter. The design space is presented in figure 34. Topology optimization of the truss is a performed by allowing zero cross sections. However for zero cross section the stress constraint has not to be considered anymore and there is a discontinuity of the restriction. The stress constraint contours are stopped along the coordinate axes and the piece of line from 'B' to 'C' is still part of the feasible domain. This linear part is a degenerated part of the design space. Obviously optimum is located in point 'C', which corresponds to remove bar A_1. However mathematical programming algorithms cannot reach point 'C' and get stocked in 'B' because of the degenerated nature of the appendix part running form 'B' to 'C'.

6.1.2 ϵ-relaxation technique

To circumvent the difficulty, Cheng and Guo [11] applied a perturbation method, called the ϵ-relaxation technique. The strategy replaces the solution of the 'singular' problem with a sequence of perturbated non-singular problems which can be solved with classical optimization algorithms.

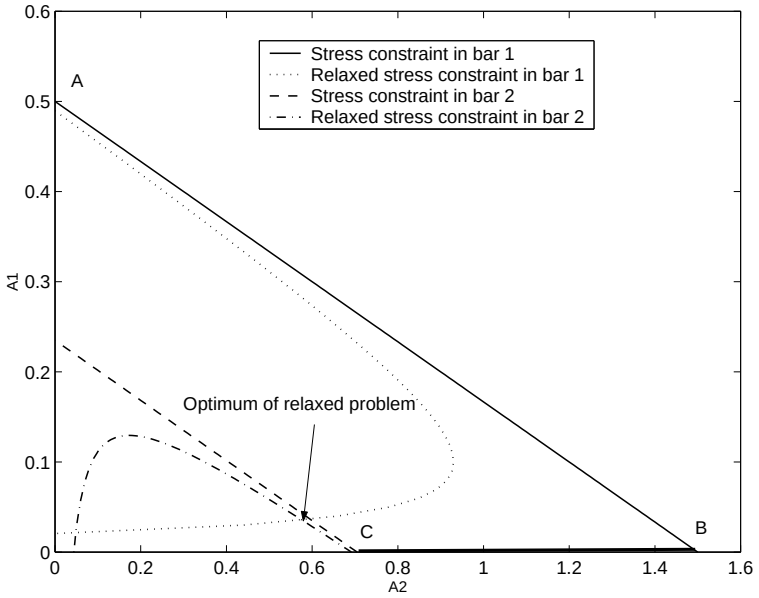

Figure 34: Design space of the 3-bar truss problem

At first, one has to reformulate the stress constraint. If $\|\sigma\|$ is a relevant stress criterion and if ρ is the density parameter, then the stress constraints in their original form are

$$\|\sigma^{eq}\| \ \leq \ \sigma_l \quad \text{if } \rho > 0 \tag{153}$$

One has to eliminate the condition $\rho > 0$ and to normalize the constraint, which gives the equivalent formulation:

$$\rho \ (\ \|\sigma^{eq}\|/\sigma_l - 1 \ ) \ \leq 0 \tag{154}$$

For bars in a truss, this is equivalent to considering forces instead of stresses, as proposed in Ref. [30] or to the adopting a quality function to replace the stress constraints like in Ref. [10]. However, this reformulation does not change the qualification of the constraints and it does not remove the algorithmic problems.

To circumvent the singularity of the design space, one has to use a perturbation of the stress constraints by using the ϵ-relaxation (in the sense of mathematical programming) approach proposed initially by Cheng and Guo [11] for truss topology. Given an additional parameter ϵ, the original stress constraints are replaced by the following relaxed stress constraints and side constraints:

$$\begin{aligned} \rho \ (\ \|\sigma^{eq}\|/\sigma_l - 1 \ ) \ &\leq \ \epsilon \\ \epsilon^2 \ &\leq \ \rho \end{aligned} \tag{155}$$

Of course for $\epsilon = 0$, this formulation renders the original problem with stress constraints. But for any $\epsilon > 0$, the ϵ-relaxed problem with the constraints (155) is characterized by

a design space W_ϵ that is not any longer degenerate, i.e. the optima are now placed in regions of the design space with non zero measure. It is thus possible to reach the optimum, denoted by ρ^*_ϵ, with classic optimization algorithms based on Karush-Kuhn-Tucker conditions. The mathematical rigor of the method stems from the possibility to prove (see Cheng and Guo [11]) that this perturbation this relaxation creates continuous point-to-set maps between the parameter ϵ and the relaxed design domains as well as to their optimal solutions. This means that when $\epsilon \to 0$, the sequence of domains $\{W_\epsilon\}$ and their optimal solutions $\{\rho^*_\epsilon\}$ converge continuously towards the original degenerate problem and its associated optimal solution. Nevertheless, the solution of every relaxed problem is regular and can be found with classical mathematical programming algorithms. Then the idea is to solve numerically a sequence of perturbated problems with decreasing values of ϵ to come to the singular solution.

A recent study [18] showed the classical implementation (155) of the ϵ-relaxation technique, which initially was developed for truss topology optimization problems, is not totally satisfactory for continuum-type topology optimization problems. Indeed with continuum topology, the problem is the following. The influence of the perturbation disappears only for $\rho = \infty$, whereas the perturbation has still a non negligible effect on the stress limit for solid material ($\rho = 1$). (The situation is illustrated in Fig. 35). Therefore, a feasible design for a given ϵ is no longer feasible for a smaller value of $\epsilon' < \epsilon$. This slows down the convergence of the optimization procedure, because the optimal and feasible solution for a problem with parameter ϵ violates some stress constraints when ϵ is reduced. To overcome the problem, one can adopt the following modified set of perturbated constraints:

$$\begin{aligned} \frac{\|\sigma^{eq}\|}{\sigma_l} - \frac{\epsilon}{\rho} + \epsilon &\leq 1 \\ \epsilon^2 &\leq \rho \end{aligned} \tag{156}$$

This new relaxation function is similar to the original one (155) in the sense that the mapping between ϵ and the perturbated problems and their optimal solutions are still continuous. However, one can easily see that the perturbation vanishes for $\rho = 1$ such that the solution remains feasible when ϵ is reduced.

On the basis of Fig. 35, one can interpret the physical mechanism of the relaxation by rewriting this constraint. If $\epsilon > 0$ the perturbated stress constraint can be rewritten as

$$\|\sigma^{eq}\| \leq \sigma_l \, (1 - \epsilon + \epsilon/\rho) \tag{157}$$

The left hand side of this constraint gives a clear physical understanding of the relaxation technique. The permissible stress is increased for low densities as illustrated in Fig. 35. This opens the degenerated parts of the design space and this gives the possibility to create or remove holes without violating the stress constraint. It should be mentioned that the figure suggests there is an equivalence between the ϵ-relaxation technique, which is a rigorous mathematical programming technique, and an intuitive idea that was independently suggested by Rozvany [36].

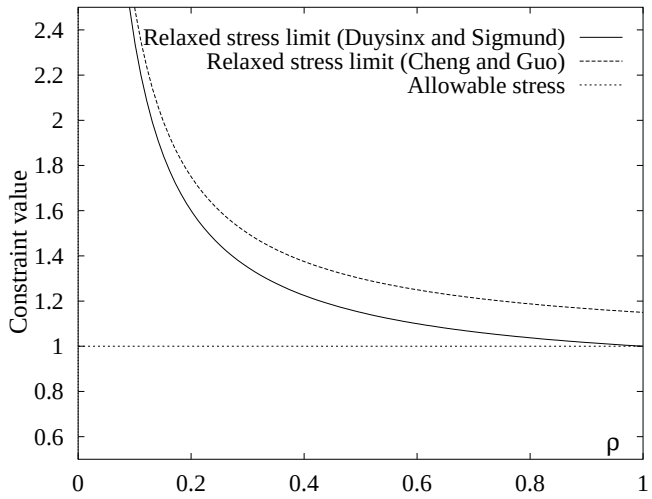

Figure 35: Relaxed stress limit as a function of the density

6.1.3 Algorithm for ϵ parameter reduction

Thus the solution procedure consists in solving a sequence of optimization problems relative to decreasing ϵ parameters. One uses a *continuation* approach similar to what is done with barrier and penalty functions. In our implementation the process is driven by the minimum density $\rho_{min} = \epsilon^2$. We typically decrease progressively the minimum density from 10^{-1} to 10^{-4} or 10^{-6}. The choice of a quite large initial minimum density is necessary to open the degenerated parts of the design domain and to be able to find the 'singular' optima from most of the initial starting points of the design space.

The reduction of the perturbation parameter ϵ is ruled by an automatic and systematic strategy. Based on numerical experience, the highly perturbated problems need not to be solved with a high precision. The perturbation parameter can be reduced as soon as the solution of the optimization problem satisfies a mild convergence criterion. The convergence criterion is based on the Euclidean norm of the gradient of the Lagrangian function (where only components that correspond to free variables are considered). If the objective function $g_0(\mathbf{x})$ and the constraint $g_j(\mathbf{x})$ are normalized with a target value of the objective function $\overline{g_0}$ and the constraint bounds $\overline{g_j}$, the following criterion gives good results

$$\|\nabla L\|_2 \;=\; \|\nabla g_0/\overline{g_0} - \sum_j \lambda_j \nabla g_j/\overline{g_j}\|_2. \tag{158}$$

The reduction algorithms is

$$\text{If} \quad \|\nabla L\|_2 \leq \alpha \qquad \text{Then} \quad \epsilon := \epsilon/\beta. \tag{159}$$

In the examples, we use $\alpha = 0.005$ and $\beta = 1.05$. A precise convergence optimization is performed when $\rho_{min} = \epsilon^2$ is sufficiently small (e.g. $\rho_{min} = 10^{-3}$). With this algorithm

the parameter ϵ is reduced automatically without any interaction from the user.

6.2 Solving optimization problems with a large number of constraints

Solving the continuum topology design problem with local stress constraints by numerical techniques results in a very large scale optimization problem. For the displacement based analysis and for the approximation of the density, the structure is discretized by finite elements, using the standard approach of continuum topology design (see e.g. Ref. [3]). In order to achieve a reasonable resolution of the optimal structure, i.e., a reasonable definition of topology and shape through the density, we need to use a fairly fine discretization. Thus in the optimization problem we have a large number of design variables coming from the discretization of the material distribution. Moreover, we have here also to treat a large number of stress constraints. If the optimal distribution of material is made up of only voids and solid, we can by analogy to fully stressed design estimate the percentage of active constraints at the optimum to be approximately proportional to the ratio between the volume of the structure and the volume of the design domain. This conclusion remains roughly the same with the ϵ-relaxed formulation because relaxation leads to a removal of a stress from the active constraint set as soon as the density of an element is close to its lower bound. This is a conservative estimate on the number of active constraints, particularly during the first steps of the optimization process where large zones of intermediate density leads to huge number of active stress constraints.

Once again, the solution procedure based on sequential mathematical programming, which was elaborated in Refs. [15, 16, 18], gives full satisfaction.

Different choices of convex approximations for compliance and eigenvalues in topology design have been discussed above. However, for stress constraints, we simply use a CONLIN approximation scheme [27] of the constraints. For an efficient use of the classical structural approximations, the stress constraints have to be written in a more convenient way (observing that the density variables are strictly positive for $\epsilon > 0$):

$$\frac{\|\sigma\|}{\sigma_l} - \frac{\epsilon}{\rho} + \epsilon \;\leq\; 1 \tag{160}$$

Numerical experiments showed that the mixed approximations of CONLIN were sufficiently conservative and precise when applied to this statement, the explanation being that the relaxation term $-\epsilon/\rho$ is a concave term which makes a convex approximation more conservative. An advantage of using the standard CONLIN approximations is that the solution of the convex and separable subproblems can be performed with a robust second order dual algorithm designed by Fleury [23, 25, 26]. This algorithm is able to deal with the huge dimensions of the problem and provides a solution within a reasonable computation time.

In addition, any strategy that aims at reducing the number of constraints to be handled by the optimizer is favorable to diminish the computational effort. If a priori, constraints are likely to be satisfied, one can save some effort by removing them from the set restrictions submitted to the optimization algorithms. So, we also implemented an active constraint selection (deletion) strategy in order to do a preselection of the potentially dangerous stress

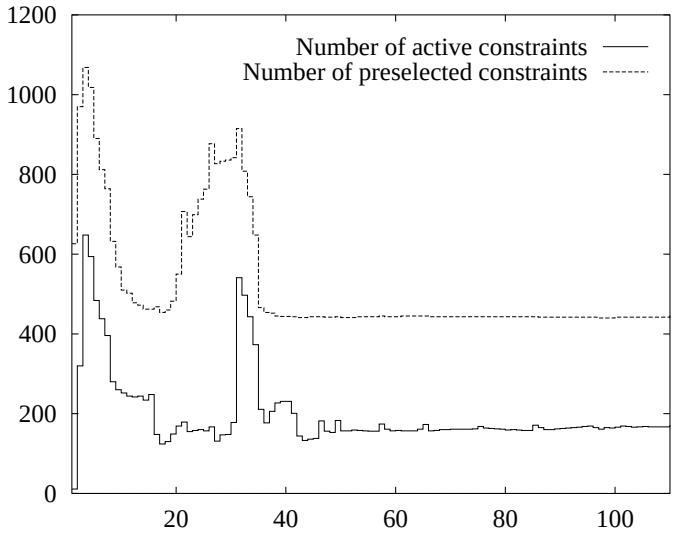

Figure 36: Number of active and preselected constraints (iterations 1-110)

constraints. At the beginning of the optimization process, the selection is large, because large modifications of the design variables occur and a lot of constraints can become active or not activate. But at the end of the optimization the set of active constraints is stable and it can be restricted to a little fraction of the dangerous elements, whose stress level lies within a small margin from stress limit. Moreover, the non active stress constraints can be kept as side constraints in the sub-problems of the iterative procedure by a zero order approximation.

Figure 36, which is related to a typical stress constrained problem, illustrates the iteration history for the number of active stress constraints (active after dual maximisation) and for the number of preselected potentially dangerous constraints retained for sensitivity analysis. During the first design steps the number of active stress constraints is quite large: 1112 potentially dangerous stresses are retained for sensitivity analysis and 648 of them are active in the CONLIN dual optimizer. Then progressively these number are reduced and become stable with the convergence of the design variables. In the final iterations, the number of active constraints is reduced to 446 potentially dangerous stresses and 180 really active constraints.

7 CONCLUSIONS

As a conclusion, we can remind the advantages of Sequential Convex Programming approach:

- Dual solvers allows to solve efficiently and with a minimum computation time optimization sub-problems even with a large number of design variables.

- The solution procedure showed itself to be robust even for problems with a large number of constraints like in stress constraints.
- One can greatly improve convergence rates and reduce the number of re-analyses to come to a stationary solution when using appropriate approximation schemes. For first order schemes, MMA is generally more efficient than CONLIN because of the capability to adapt conservativeness of the approximation to problem characteristics. Good results can also be expected with high quality approximation schemes based on second order expansions and estimated curvature information. However, this kind of procedure is sometimes more fragile.
- One major advantage of SCP approach compared to Optimality Criteria stems from the inherent flexibility and generality of the approach to solve various kinds of problems in topology: compliance, eigenfrequency, stress constraints, design of materials, design of compliant mechanism ...
- Finally a very important characteristic of SCP approach is that one has in hands mathematical foundations to attack special problems: relaxing unfeasible constraints, perturbation of non-regular problem ('singularity' phenomenon of local constraints), etc. This gives a rigorous framework for future developments and research.

8 ACKNOWLEDGMENTS

The author gratefully acknowledge the support of project ARC MEMS, Action de recherche concertée 03/08-298 funded by the Communaute Francaise de Belgique.

References

[1] G. Allaire and R. Kohn. Optimal design of minimum compliance. *European Journal of Mechanics A*, 12:839–878, 1993.

[2] L. Ambrosio and G. Buttazzo. An optimal design problem with perimeter penalization. *Calculus and Variation*, 1:55–69, 1993.

[3] M. Bendsøe. *Optimization of Structural Topology, Shape, and Material*. Springer Verlag, Heidelberg, 1995.

[4] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1:193–202, 1989.

[5] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[6] D. Bertsekas. *Non Linear Programming*. Athena Scientific Publishers, MA, USA, 2nd edition edition, 1991.

[7] M. Bruyneel and P. Duysinx. Note on note on topology optimization of continuum structures including self-weight. *Structural and Multidisciplinary Optimization*, 29(4):245–256, 2005.

[8] M. Bruyneel, P. Duysinx, and C. Fleury. A family of MMA approximations for structural optimization. *Structural and Multidisciplinary Optimization*, 24(4):263–276, 2004.

[9] M. Bruyneel and C. Fleury. Composite structures optimization using sequantial convex programming. In B. Toppings, editor, *5th International Conference on Computational Structures Technology*, volume Techniques for Materials, Composites, and Composite Structures, pages 243–254. Civil Comp Press, September 2000.

[10] G. Cheng and Z. Jiang. Study on topology optimization with stress constraints. *Engineering Optimization*, 20:129–148, 1992.

[11] G. D. Cheng and X. Guo. ϵ-relaxed approach in structural topology optimization. *Structural Optimization*, 13:258–266, 1997.

[12] J. Dennis and H. Wolkowicz. Sizing and least-change secant methods. *SIAM Journal of Numerical Analysis*, 30(5):1291–1314, October 1993.

[13] P. Duysinx. *Optimisation Topologique : du Milieu Continu à la Structure Elastique*. PhD thesis, Institut de Mécanique, Université de Liège, Belgium, February 1996. (in French).

[14] P. Duysinx. Layout optimization: a mathematical programming approach. Danish Center for Applied Mathematics and Mechanics, DCAMM Report No. 540, March 1997.

[15] P. Duysinx. Topology optimization with different stress limits in tension and compression. In C. Bloebaum, editor, *Third World Congress of Structural and Multidisciplinary Optimization*, May 1999. (CD-Rom).

[16] P. Duysinx and M. Bendsøe. Control of local stresses in topology optimization of continuum structures. *International Journal for Numerical Methods in Engineering*, 43:1453–1478, 1998.

[17] P. Duysinx, V. Nguyen, M. Bruyneel, and C. Fleury. Estimating diagonal second order terms in structural approximations with quasi-cauchy techniques. In G. Cheng, editor, *Fourth World Congress of Structural and Multidisciplinary Optimization*, May 2001. (CD-Rom).

[18] P. Duysinx and O. Sigmund. New developments in handling stress constraints in optimal material distribution. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Design and Optimization*. AIAA, September 1998.

[19] P. Duysinx, W. Zhang, C. Fleury, V. Nguyen, and S. Haubruge. A new separable approximation scheme for topological problems and optimization problems characterized by a large number of design variables. In G. Rozvany and N. Olhoff, editors, *First World Congress of Structural and Multidisciplinary Optimization*, pages 1–8. Pergamon, 1995.

[20] P. Duysinx, W. Zhang, C. Fleury, V. Nguyen, and S. Haubruge. A fast estimation procedure of diagonal second order terms in structural approximations. Report OA-57, LTAS, University of Liège, June 2000. Submited for publication.

[21] C. Fleury. Structural weight optimization by dual methods of convex programming. *International Journal of Numerical Methods in Engineering*, 14:1761–1783, 1979.

[22] C. Fleury. *Unconstrained and linearly constrained minimization*, volume Foundations of Structural Optimization: A Unified Approach, chapter 8, pages 298–334. John Wiley & Sons, 1982.

[23] C. Fleury. CONLIN: An efficient dual optimizer based on convex approximation concepts. *Structural Optimization*, 1:81–89, 1989.

[24] C. Fleury. Efficient approximation concepts using second order information. *International Journal for Numerical Methods in Engineering*, 28:2041–2058, 1989.

[25] C. Fleury. *Mathematical Programming Methods for Constrained Optimization: Dual Methods*, volume Structural Optimization: Status and Promise of *Progress in Astronautics and Aeronautics*, chapter 7, pages 123–150. AIAA, 1993.

[26] C. Fleury. *Recent Developments in Structural Optimization Methods*, volume Structural Optimization: Status and Promise of *Progress in Astronautics and Aeronautics*, chapter 9, pages 183–208. AIAA, 1993.

[27] C. Fleury and V. Braibant. Structural optimization: A new dual method using mixed variables. *International Journal for Numerical Methods in Engineering*, 23:409–428, 1986.

[28] R. Haber, C. Jog, and M. Bendsøe. A new approach to variable-topology shape design using a constraint on perimeter. *Structural Optimization*, 11:1–12, 1996.

[29] A. Hoback. Optimization of singular problems. *Structural Optimization*, 12:93–97, 1996.

[30] U. Kirsch. On singular topologies in optimum structural design. *Structural Optimization*, 2:133–142, 1990.

[31] A. Kumar. A sequential optimization algorithm using logarithmic barriers: Application to structural optimization. *Journal of Mechanical Design*, 2000. to be published.

[32] D. Luenberger. *Linear and Non-Linear Programming*. Addison Wesley, second edition, 1994.

[33] S. Mehrotra. On the implementation of a Primal-Dual Interior Point method. *SIAM Journal on Optimization*, 2:575–601, 1992.

[34] H. Mlejnek, U. Jehle, and R. Schirrmacher. Second order approximations in structural genesis and shape finding. *International Journal of Numerical Methods in Engineering*, 34:853–872, 1992.

[35] J. Petersson and O. Sigmund. Slope constrained topology optimization. *International Journal of Numerical Methods in Engineering*, 41:1417–1434, 1998.

[36] G. Rozvany. Difficulties in truss topology optimization with stress, local buckling, and system stability constraints. *Structural Optimization*, 11:213–217, 1996.

[37] G. Rozvany and T. Birker. On singular topologies in exact layout optimization. *Structural Optimization*, 8:228–235, 1994.

[38] K. Schittkowski. The non-linear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function, Part 1: Converge analysis, Part 2: An efficient implementation with least-sqaure problems. *Numerical Mathematics*, 38:38–114, 115–127, 1981.

[39] K. Schittkowski. NLPQL: A Fortran subroutine solving constrained non-linear programming problems. *Annals of Operation Research*, pages 485–500, 1985.

[40] L. Schmit and B. Farshi. Some approximation concepts for structural synthesis. *AIAA Journal*, 12:692–699, 1974.

[41] L. Schmit and C. Fleury. Structural synthesis by combining approximation concepts and dual methods. *AIAA Journal*, 18:1252–1260, 1980.

[42] H. Smaoui, C. Fleury, and L. Schmit. Advances in dual algorithms and convex approximations methods. In *AIAA/ASME/ASCE 29 Structures, Structural Dynamics and Material Conference*, pages 1339–1347, 1988.

[43] J. Starnes and R. Haftka. Preliminary design of composite wings for buckling, stress, and displacement constraints. *Journal of Aircraft*, 16(8):564–570, 1979.

[44] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

[45] K. Svanberg. Some second order methods for structural optimization. In G. Rozvany, editor, *Optimization of Large Structural Systems*, volume I of *NATO ASI series*, pages 567–578, The Netherlands, 1993. Kluwer Academic Publishers.

[46] K. Svanberg. A globally convergent version of MMA without linesearch. In G. Rozvany and N. Olhoff, editors, *First World Congress of Structural and Multidisciplinary Optimization*, pages 9–16. Pergamon, 1995.

[47] K. Svanberg. Non-mixed second order derivatives in MMA. Dpt. of Mathematics, Royal Institute of Technology, Stockholm, 1995.

[48] K. Thapa. *Optimization of Unconstrained Functions with Sparse Hessian Matrices*. PhD thesis, Stanford University, Department of Operations Research, 1981.

[49] W. Zhang, M. Domaszewski, and C. Fleury. A new mixed convex approximation method with applications for truss configuration optimization. *Structural Optimization*, 15:237–241, 1998.

[50] W. Zhang and C. Fleury. Recent advances in convex approximation methods for structural optimization. In B. Toppings and M. Papadrakakis, editors, *Advances in Structural Optimization*, pages 83–90, Edinburgh, 1994. CIVIL-COMP Press. Proceedings of the Second International Conference on Computational Structures Technology, Athens.

[51] W. Zhang, C. Fleury, P. Duysinx, V. Nguyen, and I. Laschet. A generalized method of moving asymptotes (gmma) including equality constraints. *Structural Optimization*, 12:143–146, 1996.

[52] M. Zhu, J. Nazareth, and H. Wolkowicz. The Quasi-Cauchy relations and diagonal updating. *SIAM Journal of Optimization*, 9(4):1192–1204, 1999.

A DIAGONAL HESSIAN ESTIMATES

To avoid the second order sensitivity analysis, the idea is to use the available first order information and to build an approximation of the Hessian matrix of the response with a Quasi-Newton update procedure. The problem is that the "full" Quasi-Newton becomes also expensive as the number of design variables increases. Furthermore, only diagonal terms are useful since only separable approximations are considered in structural optimization. That's why we present here a modified BFGS updating scheme able to generate a sequence of diagonal Hessian estimates. The algorithm is the adaptation to diagonal matrices of more general results established by Thapa [48] for Quasi-Newton updates preserving the sparsity pattern of the Hessian estimates.

A.1 Adaptation of sparse Quasi-Newton updates to diagonal structure

Let $\mathbf{B}$ be a diagonal approximation of the Hessian matrix of a given structural response $g(\mathbf{x})$ at the current design point $\mathbf{x}$. If the new design $\mathbf{x}^+$ doesn't satisfy convergence criteria, one seeks to enrich the estimation of the Hessian with a Quasi-Newton update procedure.

Update formulae are based on the Quasi-Newton equation:

$$\mathbf{B}^+\mathbf{s} = \mathbf{y} \quad \text{where} \quad \mathbf{s} = \mathbf{x}^+ - \mathbf{x} \quad \text{and} \quad \mathbf{y} = \nabla g(\mathbf{x}^+) - \nabla g(\mathbf{x}) \tag{161}$$

The most famous Quasi-Newton update formula is the Broyden-Fletcher-Goldfard-Shanno (BFGS) one:

$$\hat{\mathbf{B}}^+ = \mathbf{B} + \mathbf{U_{BFGS}} \quad \text{with} \quad \mathbf{U_{BFGS}} = \frac{\mathbf{y}\mathbf{y}^T}{\mathbf{s}^T\mathbf{y}} - \frac{\mathbf{B}\mathbf{s}\mathbf{s}^T\mathbf{B}}{\mathbf{s}^T\mathbf{B}\mathbf{s}} \tag{162}$$

This scheme satisfies simultaneously the symmetry property, the positive definiteness of the update, and the Quasi-Newton condition (161). But it doesn't preserve sparse or diagonal structure of the previous estimation. Such updates preserving diagonal pattern can be derived from Thapa's theorems [48]. The results of adaptation of this general theory to diagonal structure are summarized here.

$\hat{\mathbf{B}}_{\mathbf{D}}^{+}$ and $\hat{\mathbf{B}}_{\mathbf{ND}}^{+}$ stand for the matrices which are formed respectively with the diagonal and the off-diagonal terms of the "full" update $\hat{\mathbf{B}}^{+}$, while the diagonal Quasi-Newton update we look for is $\mathbf{B}^{+}$.

According to Thapa, the diagonal update can be found by adding a diagonal correction matrix $\mathbf{E}$ to the diagonal part of the classic updated matrix $\hat{\mathbf{B}}_{\mathbf{D}}^{+}$:

$$\mathbf{B}^{+} = \hat{\mathbf{B}}_{\mathbf{D}}^{+} + \mathbf{E} \tag{163}$$

This diagonal correction matrix $\mathbf{E}$ is such one that the diagonal updated matrix $\mathbf{B}^{+}$ is the closest to the classic updated matrix $\hat{\mathbf{B}}_{\mathbf{D}}^{+}$ in the Frobenius norm and still satisfies the Quasi-Newton condition $\mathbf{B}^{+}\mathbf{s} = \mathbf{y}$. If $||\mathbf{E}||_F$ is the Frobenius norm of matrix $\mathbf{E}$, the correction matrix $\mathbf{E}$ is the solution of the minimum problem:

$$\begin{array}{ll} \min & ||\mathbf{E}||_F \\ \text{s.t.} & \mathbf{E}\mathbf{s} = \hat{\mathbf{B}}_{\mathbf{ND}}^{+}\mathbf{s} \\ & E_{ij} = 0 \qquad (i \neq j) \end{array} \tag{164}$$

The solution of this problem writes

$$\mathbf{E} = \mathrm{diag}\{2\lambda_i s_i\} \tag{165}$$

where the vector $\lambda = (\lambda_1, \ldots, \lambda_n)$, itself, is the solution of the linear diagonal system :

$$\mathbf{Q}\lambda = \hat{\mathbf{B}}_{\mathbf{ND}}^{+}\mathbf{s} \quad \text{with} \quad \mathbf{Q} = \mathrm{diag}\{2s_i^2\} \tag{166}$$

The computation of the diagonal BFGS update requires solving this last diagonal system, which is very simple. It is interesting to note that computing the off-diagonal terms of the classical update correction is not necessary. Only the diagonal terms are useful, since it can be easily shown with the Quasi-Newton equation and the constraint of the problem (164) that:

$$\hat{\mathbf{B}}_{\mathbf{ND}}^{+}\mathbf{s} = \mathbf{y} - \mathbf{B}\mathbf{s} - \mathbf{U}_{\mathbf{D}}\,\mathbf{s} \tag{167}$$

where $\mathbf{U}_{\mathbf{D}}$ is the diagonal part of the update matrix $\mathbf{U}$ given in (162).

A.2 Computation of second order diagonal terms

Pursuing the developments will lead to the practical formula that is used to compute the second order terms. Equation (163) writes:

$$\mathbf{B}^{+} = \mathbf{B}_{\mathbf{D}} + \mathbf{U}_{\mathbf{D}} + \mathbf{E}$$

When writing this last equation component by component and taking into account that $[\mathbf{E}]_{ii} = 2\lambda_i \ s_i$, one has:

$$\left[\mathbf{B}^+\right]_{ii} = [\mathbf{B_D}]_{ii} + [\mathbf{U_D}]_{ii} + 2\lambda_i \ s_i \tag{168}$$

On another hand, from the solution of system (166) and from equation (167), one has:

$$2\lambda_i \ s_i^2 = y_i - [\mathbf{B_D}]_{ii} \ s_i - [\mathbf{U_D}]_{ii} \ s_i$$

From which one extracts the value of $2\lambda_i \ s_i$:

$$2\lambda_i \ s_i = y_i/s_i - [\mathbf{B_D}]_{ii} - [\mathbf{U_D}]_{ii} \quad \text{if} \quad s_i \ \neq \ 0 \tag{169}$$

Combining equations (168) and (169), one comes to the very simple equation:

$$\left[\mathbf{B}^+\right]_{ii} = y_i/s_i \quad \text{if} \quad s_i \ \neq \ 0 \tag{170}$$

which is the result announced in the introduction:

$$\left[\mathbf{B}^+\right]_{ii} \ = \ \frac{\dfrac{\partial g(\mathbf{x}^+)}{\partial x_i} - \dfrac{\partial g(\mathbf{x})}{\partial x_i}}{x_i^+ - x_i} \tag{171}$$

When $s_i = 0$, which means that $x_i^+ = x_i$, one can obviously keep the former estimation of the second order term:

$$\left[\mathbf{B}^+\right]_{ii} \ = \ [\mathbf{B}]_{ii} \tag{172}$$

One thus comes to the conclusion that in the framework of diagonal Quasi-Newton updates, the best diagonal approximation of Hessian matrix is given by the "finite difference" between the partial derivatives evaluated in the two last iteration points.

The finite difference formula (171) is here justified by the mathematical theory of sparse (diagonal) Quasi-Newton updates and this rather intuitive technique finds here a mathematical foundation.

Unfortunately, as we will see later a "brutal" implementation of this estimation formula generally leads to unsatisfactory results in structural optimization. Our explanation of this comes from the very non-linear character of structural optimization problems. But we now present how the theoretical result can be turned into an efficient strategy.