The Method of Moving Asymptotes – Modelling aspects and solution schemes

by

Krister Svanberg

Optimization and Systems Theory
Royal Institute of Technology (KTH)
SE-10044 Stockholm, Sweden

Lecture notes for the DCAMM course
Advanced Topics in Structural Optimization
Lyngby, June 25 – July 3, 1998

MMA – modelling aspects and solution schemes[*]

Krister Svanberg[†]

Abstract

This paper deals with modelling aspects and solution schemes when using the Method of Moving Asymptotes (MMA) for solving different optimization problems. First, a convenient and useful general form for inequality-constrained optimization problems is suggested and analyzed. After that, MMA for solving problems on this general form is dealt with. In particular, a primal-dual interior point method for solving the subproblems is developed. Further, an option for the user to provide non-mixed second derivatives is described. Finally, an implementation of MMA in Matlab is presented and illustrated on a simple example. The reader is invited to obtain the Matlab-files free for academic usage.

1. Considered optimization problem.

Throughout this paper, optimization problems on the following form are considered, where the variables are $x = (x_1, \dots, x_n)^T \in I\!R^n$, $y = (y_i, \dots, y_m)^T \in I\!R^m$ and $z \in I\!R$.

$$\begin{array}{lll}
\text{minimize} & f_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \frac{1}{2}d_i y_i^2) & \\
\text{subject to} & f_i(x) - a_i z - y_i \leq 0, & i = 1, \dots, m \\
 & x_j^{min} \leq x_j \leq x_j^{max}, & j = 1, \dots, n \\
 & y_i \geq 0, & i = 1, \dots, m \\
 & z \geq 0, &
\end{array} \tag{1.1}$$

Here, $f_0, f_1, \dots, f_m$ are given, continuously differentiable, real-valued functions.

x_j^{min} and x_j^{max} are given real numbers such that, for each j, $x_j^{min} < x_j^{max}$.

a_i, c_i and d_i are given non-negative real numbers such that, for each i, $c_i + d_i > 0$.

Motivations for considering problems on this particular form will be given in the following sections.

[*] Research supported by the Swedish Research Council for the Engineering Sciences (TFR).

[†] Optimization and Systems Theory, KTH, Stockholm, Sweden: krille@math.kth.se

2. Some modelling aspects.

In this section, we give some examples of how the above formulation (1.1) can be used in some particular cases which often appear in application.

2.1. Ordinary NLP problems.

Assume that one wants to solve a problem on the following "standard" form for nonlinear programming.

$$\begin{aligned} &\text{minimize} && f_0(x) && \\ &\text{subject to} && f_i(x) \le 0, && i = 1, \dots, m \\ & && x_j^{min} \le x_j \le x_j^{max}, && j = 1, \dots, n \end{aligned} \tag{2.1}$$

The parameters a_i, c_i and d_i should then be chosen as follows in problem (1.1).

For each i, let $a_i = 0$. Then $z = 0$ in any optimal solution of (1.1). Further, for each i, let $d_i = 0$ and $c_i =$ "a large number", so that the variables y_i become "expensive". Then typically $y_i = 0$ in any optimal solution of (1.1), and the corresponding x is an optimal solution of (2.1).

It should be noted that the problem (1.1) always has feasible solutions, and in fact also at least one optimal solution (see below). This holds even if the user's problem (2.1) does not have any feasible solutions, in which case some $y_i > 0$ in the optimal solution of (1.1). This is one advantage of the formulation (1.1) compared to the formulation (2.1). Other advantages will be discussed later.

Now follows some practical considerations and recommendations concerning the above situation.

In many structural optimization applications, the constraints are typically on the form $\sigma_i(x) \le \sigma_i^{max}$, where $\sigma_i(x)$ stands for e.g. a certain stress in the structure, while σ_i^{max} is the largest permitted value on this stress. This means that $f_i(x) = \sigma_i(x) - \sigma_i^{max}$ (in (1.1) as well as in (2.1)). The user should then preferably scale the constraints in such a way that $1 \le \sigma_i^{max} \le 100$ for each i (and not e.g. $\sigma_i^{max} = 10^9$). The objective function $f_0(x)$ should preferably be scaled such that $1 \le f_0(x) \le 100$ for reasonable values on the variables. The variables should preferably be scaled such that typical values satisfy $0.1 \le |x_j| \le 100$ for all j.

Concerning the "large numbers" above, the user should (for numerical reasons) try to avoid very large values on the coefficients c_i (like 10^{12}). It is better to start with "reasonably large" values and then, if it turns out that not all $y_i = 0$ in the optimal solution of (1.1), increase these c_i-values by e.g. a factor 100 and solve the problem again, etc. If the functions and the variables have been scaled according to above, then "resonably large" values on the parameters c_i could be, say, $c_i = 1000$.

Finally, concerning the simple bound constraints $x_j^{min} \le x_j \le x_j^{max}$, it may sometimes be the case that some variables x_j do not have any prescribed upper and/or lower bounds. In that case, it is in practice always possible to choose "artificial" bounds x_j^{min} and x_j^{max} such that every realistic solution x satisfies the corresponding bound constraints. The user should then preferably avoid choosing $x_j^{max} - x_j^{min}$ unnecessarily large. It is better to try some reasonable bounds and then, if it turns out that some variable x_j becomes equal to such an "artificial" bound in the optimal solution of (1.1), change this bound and solve the problem again (starting from the recently obtained solution), etc.

2.2. Least squares problems. (Minimum l_2-norm problems.)

Assume that one wants to solve a constrained least squares problem on the form

$$\begin{array}{rll} \text{minimize} & \sum_{k=1}^{p}(h_k(x))^2 & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, q \\ & x_j^{min} \leq x_j \leq x_j^{max}, & j = 1, \ldots, n \end{array} \tag{2.2}$$

where h_k and g_i are given differentiable functions.

The functions f_i and the parameters a_i, c_i and d_i should then be chosen as follows in problem (1.1).

$$\begin{array}{lcll} m & = & 2p+q, & \\ f_0(x) & = & 0, & \\ f_i(x) & = & h_i(x), & i = 1, \ldots, p \\ f_{p+i}(x) & = & -h_i(x), & i = 1, \ldots, p \\ f_{2p+i}(x) & = & g_i(x), & i = 1, \ldots, q \\ a_i & = & 0, & i = 1, \ldots, m \\ d_i & = & 2, & i = 1, \ldots, 2p \\ d_{2p+i} & = & 0, & i = 1, \ldots, q \\ c_i & = & 0, & i = 1, \ldots, 2p \\ c_{2p+i} & = & \text{large number}, & i = 1, \ldots, q \end{array}$$

2.3. Minimum l_1-norm problems.

Assume that one wants to solve a minimum l_1-norm problem on the form

$$\begin{array}{rll} \text{minimize} & \sum_{k=1}^{p}|h_k(x)| & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, q \\ & x_j^{min} \leq x_j \leq x_j^{max}, & j = 1, \ldots, n \end{array} \tag{2.3}$$

where h_k and g_i are given differentiable functions.

The functions f_i and the parameters a_i, c_i and d_i should then be chosen as follows in problem (1.1).

$$\begin{array}{lll}
m & = & 2p+q, \\
f_0(x) & = & 0, \\
f_i(x) & = & h_i(x), \qquad i=1,\ldots,p \\
f_{p+i}(x) & = & -h_i(x), \qquad i=1,\ldots,p \\
f_{2p+i}(x) & = & g_i(x), \qquad i=1,\ldots,q \\
a_i & = & 0, \qquad i=1,\ldots,m \\
d_i & = & 0, \qquad i=1,\ldots,m \\
c_i & = & 1, \qquad i=1,\ldots,2p \\
c_{2p+i} & = & \text{large number}, \quad i=1,\ldots,q
\end{array}$$

2.4. Minimum l_∞-norm problems.

Assume that one wants to solve an l_∞-norm problem on the form

$$\begin{array}{ll}
\text{minimize} & \max\limits_{k=1,..,p} \{|h_k(x)|\} \\
\text{subject to} & g_i(x) \le 0, \qquad i=1,\ldots,q \\
 & x_j^{min} \le x_j \le x_j^{max}, \quad j=1,\ldots,n
\end{array} \tag{2.4}$$

where h_k and g_i are given differentiable functions.

The functions f_i and the parameters a_i, c_i and d_i should then be chosen as follows in problem (1.1).

$$\begin{array}{lll}
m & = & 2p+q, \\
f_0(x) & = & 0, \\
f_i(x) & = & h_i(x), \qquad i=1,\ldots,p \\
f_{p+i}(x) & = & -h_i(x), \qquad i=1,\ldots,p \\
f_{2p+i}(x) & = & g_i(x), \qquad i=1,\ldots,q \\
a_i & = & 1, \qquad i=1,\ldots,2p \\
a_{2p+i} & = & 0, \qquad i=1,\ldots,q \\
d_i & = & 0, \qquad i=1,\ldots,m \\
c_i & = & \text{large number}, \quad i=1,\ldots,m
\end{array}$$

3. Some theoretical properties of the considered problem.

In this section, some theoretical properties of the problem (1.1) will be proved.

To shorten the notation, let $X = \{x \in I\!R^n \mid x_j^{min} \leq x_j \leq x_j^{max},\ j = 1, \ldots, n\}$.

First, we note the simple fact that there are always feasible solutions of problem (1.1). This follows since for any $x \in X$ it is possible to find y and z such that (x, y, z) is a feasible solution of (1.1). One may take, e.g., $z = 0$ and $y_i = \max\{0, f_i(x)\}$. A perhaps less obvious fact is the following.

Proposition 3.1. *There is always at least one optimal solution of the problem (1.1).*

Proof. Since X is a compact set and f_i is a continuous function for each i, it follows from Weierstrass' theorem that the numbers $\overline{f}_i = \max\{f_i(x) \mid x \in X\}$, $i = 1, \ldots, m$, are well defined.

Let $y_i^{max} = 1 + \max\{0, \overline{f}_i\}$ and $Y = \{y \in I\!R^m \mid 0 \leq y_i \leq y_i^{max},\ i = 1, \ldots, m\}$.

If $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution of the problem (1.1) then $\hat{y}_i < y_i^{max}$ for all i, since if any $\hat{y}_i \geq y_i^{max}$ then $f_i(\hat{x}) - a_i\hat{z} - \hat{y}_i \leq -1 < 0$, and then a feasible solution with strictly lower objective value would be obtained by simply decreasing y_i.

Thus, no optimal solutions of the problem (1.1) are lost or introduced if the restriction $y \in Y$ is included in the formulation.

Next, let $Z = \{z \in I\!R \mid 0 \leq z \leq z^{max}\}$,
where $z^{max} = 1$ if $a_1 = \cdots = a_m = 0$, while $z^{max} = 1 + \max_i\{\frac{y_i^{max}}{a_i} \mid a_i > 0\}$ otherwise.

By similar arguments as above, it follows that no optimal solutions of the problem (1.1) are lost or introduced if the restriction $z \in Z$ is included in the formulation.

The above discussion implies that $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution of the problem (1.1) if and only if $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution of the following problem.

$$\begin{aligned} &\text{minimize} && f_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2}d_i y_i^2) && \\ &\text{subject to} && f_i(x) - a_i z - y_i \leq 0, && i = 1, \ldots, m \\ & && x_j^{min} \leq x_j \leq x_j^{max}, && j = 1, \ldots, n \\ & && 0 \leq y_i \leq y_i^{max}, && i = 1, \ldots, m \\ & && 0 \leq z \leq z^{max}. && \end{aligned} \tag{3.1}$$

In this problem (3.1), the feasible set is nonempty and compact while the objective function is continuous. Thus, according to Weierstrass' theorem, there is always at least one optimal solution. But any such optimal solution of (3.1) is an optimal solution also of the problem (1.1). ∎

The next result deals with the Kuhn-Tucker optimality conditions. In order to state these, we first form the Lagrange function for problem (1.1):

$$\begin{aligned} L(x,y,z,\lambda,\xi,\eta,\mu,\zeta) = \;& f_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2} d_i y_i^2) + \\ & + \sum_{i=1}^{m} \lambda_i (f_i(x) - a_i z - y_i) + \\ & + \sum_{j=1}^{n} \xi_j (x_j^{min} - x_j) + \\ & + \sum_{j=1}^{n} \eta_j (x_j - x_j^{max}) - \\ & - \sum_{i=1}^{m} \mu_i y_i - \zeta z \,, \end{aligned} \tag{3.2}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$, $\xi = (\xi_1, \ldots, \xi_n)^T$, $\eta = (\eta_1, \ldots, \eta_n)^T$, $\mu = (\mu_1, \ldots, \mu_m)^T$ and ζ are non-negative Lagrange multipliers for the different constraints in (1.1).

The Kuhn-Tucker conditions for problem (1.1), which are stated below, consist of 4 types of conditions, namely

- Stationarity of the Lagrange function L with respect to (x, y, z).
- Primal feasibility, which means that (x, y, z) should be a feasible point.
- Dual feasiblity, which means that the Lagrange multipliers should be non-negative.
- Complementary slackness, which means that Lagrange multipliers corresponding to inactive constraints should be zero.

$$\frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial f_i}{\partial x_j} - \xi_j + \eta_j = 0, \quad j = 1, \ldots, n \quad (\partial L / \partial x_j = 0) \tag{3.3a}$$

$$c_i + d_i y_i - \lambda_i - \mu_i = 0, \quad i = 1, \ldots, m \quad (\partial L / \partial y_i = 0) \tag{3.3b}$$

$$1 - \zeta - \sum_{i=1}^{m} \lambda_i a_i = 0, \quad (\partial L / \partial z = 0) \tag{3.3c}$$

$$f_i(x) - a_i z - y_i \leq 0, \quad i = 1, \ldots, m \quad \text{(primal feasibility)} \tag{3.3d}$$

$$\lambda_i (f_i(x) - a_i z - y_i) = 0, \quad i = 1, \ldots, m \quad \text{(compl slackness)} \tag{3.3e}$$

$$\xi_j (x_j^{min} - x_j) = 0, \quad j = 1, \ldots, n \quad \text{(compl slackness)} \tag{3.3f}$$

$$\eta_j (x_j - x_j^{max}) = 0, \quad j = 1, \ldots, n \quad \text{(compl slackness)} \tag{3.3g}$$

$$-\mu_i y_i = 0, \quad i = 1, \ldots, m \quad \text{(compl slackness)} \tag{3.3h}$$

$$-\zeta z = 0, \quad \text{(compl slackness)} \tag{3.3i}$$

$$x_j^{min} - x_j \leq 0 \;\text{ and }\; x_j - x_j^{max} \leq 0, \quad j = 1, \ldots, n \quad \text{(primal feasibility)} \tag{3.3j}$$

$$-z \leq 0 \;\text{ and }\; -y_i \leq 0, \quad i = 1, \ldots, m \quad \text{(primal feasibility)} \tag{3.3k}$$

$$\xi_j \geq 0 \;\text{ and }\; \eta_j \geq 0, \quad j = 1, \ldots, n \quad \text{(dual feasibility)} \tag{3.3l}$$

$$\zeta \geq 0 \;\text{ and }\; \mu_i \geq 0, \quad i = 1, \ldots, m \quad \text{(dual feasibility)} \tag{3.3m}$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, m \quad \text{(dual feasibility)} \tag{3.3n}$$

The next proposition says that these Kuhn-Tucker conditions are in fact necessary conditions for an optimal solution of problem (1.1).

Proposition 3.2. *If $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution (local or global) of problem (1.1) then there are Lagrange multipliers $(\lambda, \xi, \eta, \mu, \zeta)$ which, together with $(\hat{x}, \hat{y}, \hat{z})$, satisfy the above conditions (3.3a)–(3.3n).*

Proof. It is well known that if $\hat{x}$ is an optimal solution (local or global) of a problem on the form

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \le 0, \quad i = 1, \ldots, m \\ & && x \in I\!R^n, \end{aligned} \tag{3.4}$$

and if there is a vector $\Delta x \in I\!R^n$ such that $\nabla f_i(\hat{x})^T \Delta x < 0$ for all i such that $f_i(\hat{x}) = 0$ (i.e. the inner product of Δx and the gradient vector of any active constraint in problem (3.4) is strictly negative) then there are Lagrange multipliers λ_i which together with $\hat{x}$ satisfy the Kuhn-Tucker conditions, which in this case are

$$\frac{\partial f_0}{\partial x_j}(\hat{x}) + \sum_{i=1}^{m} \lambda_i \frac{\partial f_i}{\partial x_j}(\hat{x}) = 0, \quad j = 1, \ldots, n \quad (\partial L / \partial x_j = 0) \tag{3.5a}$$

$$f_i(\hat{x}) \le 0, \quad i = 1, \ldots, m \quad \text{(primal feasibility)} \tag{3.5b}$$

$$\lambda_i \ge 0, \quad i = 1, \ldots, m \quad \text{(dual feasibility)} \tag{3.5c}$$

$$\lambda_i f_i(\hat{x}) = 0, \quad i = 1, \ldots, m \quad \text{(compl slackness)} \tag{3.5d}$$

This result shall now be applied to problem (1.1).

Assume that $(\hat{x}, \hat{y}, \hat{z})$ is an optimal solution of problem (1.1) and construct a corresponding vector $(\Delta x, \Delta y, \Delta z)$ as follows.

For $j = 1, \ldots, n$, let $\Delta x_j = 1$ if $\hat{x}_j = x_j^{min}$, $\Delta x_j = -1$ if $\hat{x}_j = x_j^{max}$, $\Delta x_j = 0$ otherwise.

For $i = 1, \ldots, m$, let $\Delta y_i = 1 + \sum_{j=1}^{n} |\frac{\partial f_i}{\partial x_j}(\hat{x})|$. Finally, let $\Delta z = 1$.

Then it is easily checked that the inner product of $(\Delta x, \Delta y, \Delta z)$ and the gradient vector, calculated at $(\hat{x}, \hat{y}, \hat{z})$, of any active constraint in problem (1.1) is strictly negative. Thus, there are Lagrange multipliers which together with $(\hat{x}, \hat{y}, \hat{z})$ satisfy the Kuhn-Tucker conditions (3.3a)–(3.3n). ∎

Corollary 3.1. *If, in problem (1.1), $d_i = 0$ for all $i \in \{1, \ldots, m\}$ and $\sum_{i=1}^{m} a_i c_i < 1$ then $z = 0$ in every optimal solution.*

Proof. (3.3b) implies that $\lambda_i \le c_i$ for each i. Together with (3.3c) this implies that $\zeta \ge 1 - \sum_i a_i c_i > 0$. But then (3.3i) implies that $z = 0$. ∎

Corollary 3.2. *If, in problem (1.1), $a_i c_i > 1$ for some $i \in \{1, \ldots, m\}$ then the corresponding $y_i = 0$ in every optimal solution.*

Proof. (3.3c) implies that $\lambda_i a_i \le 1$ for each i. Together with (3.3b) this implies that $\mu_i a_i \ge c_i a_i - 1 > 0$, so that $\mu_i > 0$. But then (3.3h) implies that $y_i = 0$. ∎

4. The MMA subproblem.

MMA is a method for solving problems on the form (1.1), using the following approach: In each iteration, a current iteration point $(x^{(k)}, y^{(k)}, z^{(k)})$ is given. Then an approximating explicit subproblem is generated. In this subproblem, the functions $f_i(x)$ are replaced by approximating convex functions $\tilde{f}_i^{(k)}(x)$. These approximations are based mainly on gradient information at the current iteration point, but also (implicitly) on information from previous iteration points. The subproblem is solved, and the unique optimal solution becomes the next iteration point $(x^{(k+1)}, y^{(k+1)}, z^{(k+1)})$. Then a new subproblem is generated, etc.

The subproblem mentioned above looks as follows.

$$\begin{array}{lll}
\text{minimize} & \tilde{f}_0^{(k)}(x) + z + \sum_{i=1}^{m} (c_i y_i + \frac{1}{2} d_i y_i^2) & \\
\text{subject to} & \tilde{f}_i^{(k)}(x) - a_i z - y_i \leq 0, & i = 1, \ldots, m \\
 & \alpha_j^{(k)} \leq x_j \leq \beta_j^{(k)}, & j = 1, \ldots, n \\
 & y_i \geq 0, & i = 1, \ldots, m \\
 & z \geq 0. &
\end{array} \tag{4.1}$$

The approximating functions $\tilde{f}_i^{(k)}(x)$ are chosen as

$$\tilde{f}_i^{(k)}(x) = \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - l_j^{(k)}} \right) + r_i^{(k)}, \quad i = 0, 1, \ldots, m,$$

where

$$p_{ij}^{(k)} = (u_j^{(k)} - x_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^+ + \kappa_{ij}^{(k)} \right),$$

$$q_{ij}^{(k)} = (x_j^{(k)} - l_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^- + \kappa_{ij}^{(k)} \right),$$

$$r_i^{(k)} = f_i(x^{(k)}) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - l_j^{(k)}} \right),$$

$$\alpha_j^{(k)} = \max\{x_j^{min}, 0.9 l_j^{(k)} + 0.1 x_j^{(k)}\},$$

$$\beta_j^{(k)} = \min\{x_j^{max}, 0.9 u_j^{(k)} + 0.1 x_j^{(k)}\}.$$

Here,

$$\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^+ = \max\{0, \frac{\partial f_i}{\partial x_j}(x^{(k)})\} \text{ and } \left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^- = \max\{0, -\frac{\partial f_i}{\partial x_j}(x^{(k)})\}.$$

The default rules for updating the lower asymptotes $l_j^{(k)}$ and the upper asymptotes $u_j^{(k)}$ are as follows. The first two iterations, when $k = 1$ and $k = 2$,

$$l_j^{(k)} = x_j^{(k)} - 0.5(x_j^{max} - x_j^{min}),$$

$$u_j^{(k)} = x_j^{(k)} + 0.5(x_j^{max} - x_j^{min}).$$

In later iterations, when $k \geq 3$,

$$l_j^{(k)} = x_j^{(k)} - \gamma_j^{(k)}(x_j^{(k-1)} - l_j^{(k-1)}),$$

$$u_j^{(k)} = x_j^{(k)} + \gamma_j^{(k)}(u_j^{(k-1)} - x_j^{(k-1)}),$$

where

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

In the original MMA, $\kappa_{ij}^{(k)} = 0$ for all i and j. Other values on $\kappa_{ij}^{(k)}$ appear in the following two sections.

5. Non-mixed second derivatives in MMA

Sometimes the user might be able to calculate some of the non-mixed second derivatives of some of the functions f_i. In this section, it will be described how this additional information can be utilized when generating the MMA subproblem.

First, the parameters $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are calculated as above with all $\kappa_{ij}^{(k)} = 0$, i.e.

$$p_{ij}^{(k)} = (u_j^{(k)} - x_j^{(k)})^2 \left(\frac{\partial f_i}{\partial x_j}(x^{(k)})\right)^+,$$

$$q_{ij}^{(k)} = (x_j^{(k)} - l_j^{(k)})^2 \left(\frac{\partial f_i}{\partial x_j}(x^{(k)})\right)^-.$$

Next, the following numbers $\delta_{ij}^{(k)}$ are calculated.

$$\delta_{ij}^{(k)} = \frac{\partial^2 f_i}{\partial x_j^2}(x^{(k)}) - \frac{2p_{ij}^{(k)}}{(u_j^{(k)} - x_j^{(k)})^3} - \frac{2q_{ij}^{(k)}}{(x_j^{(k)} - l_j^{(k)})^3},$$

where it is assumed that the second derivatives have been calculated by the user.

If $\delta_{ij}^{(k)} \leq 0$ then the above values on $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are used to define $\tilde{f}_i^{(k)}$.

If $\delta_{ij}^{(k)} > 0$ then instead the following values on $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are used to define $\tilde{f}_i^{(k)}$.

$$p_{ij}^{(k)} = (u_j^{(k)} - x_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^+ + \frac{\delta_{ij}^{(k)}(u_j^{(k)} - x_j^{(k)})(x_j^{(k)} - l_j^{(k)})}{2(u_j^{(k)} - l_j^{(k)})} \right),$$

$$q_{ij}^{(k)} = (x_j^{(k)} - l_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^- + \frac{\delta_{ij}^{(k)}(u_j^{(k)} - x_j^{(k)})(x_j^{(k)} - l_j^{(k)})}{2(u_j^{(k)} - l_j^{(k)})} \right).$$

With these latter values on $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ it holds that

$$\frac{\partial \tilde{f}_i^{(k)}}{\partial x_j}(x^{(k)}) = \frac{\partial f_i}{\partial x_j}(x^{(k)}) \quad \text{and} \quad \frac{\partial^2 \tilde{f}_i^{(k)}}{\partial x_j^2}(x^{(k)}) = \frac{\partial^2 f_i}{\partial x_j^2}(x^{(k)}).$$

When all the $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ have been calculated, the parameters $r_i^{(k)}$ are calculated as

$$r_i^{(k)} = f_i(x^{(k)}) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - l_j^{(k)}} \right).$$

Then it holds that $\tilde{f}_i^{(k)}(x^{(k)}) = f_i(x^{(k)})$.

It is possible for the user to calculate only some of the non-mixed second derivatives. The ones that are not calculated should simply be set to zero, so that the corresponding $\delta_{ij}^{(k)}$ become ≤ 0. This implies that the original MMA approximations will be used for these missing combinations of i and j.

6. A globally convergent version of MMA.

The original MMA can be modified in such a way that a globally convergent version is obtained. Again, the approximating functions $\tilde{f}_i^{(k)}(x)$ are chosen as

$$\tilde{f}_i^{(k)}(x) = \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - l_j^{(k)}} \right) + r_i^{(k)}, \quad i = 0, 1, \ldots, m,$$

but now $p_{ij}^{(k)}$ and $q_{ij}^{(k)}$ are chosen as

$$p_{ij}^{(k)} = (u_j^{(k)} - x_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^+ + \frac{\rho_i^{(k)}(u_j^{(k)} - l_j^{(k)})}{2} \right),$$

$$q_{ij}^{(k)} = (x_j^{(k)} - l_j^{(k)})^2 \left(\left(\frac{\partial f_i}{\partial x_j}(x^{(k)}) \right)^- + \frac{\rho_i^{(k)}(u_j^{(k)} - l_j^{(k)})}{2} \right),$$

where the parameters $\rho_i^{(k)}$ are chosen as follows. The first iteration, when $k = 1$, $\rho_i^{(1)} =$ a "small" but strictly positive number for all $i \in \{0, 1, \dots, m\}$.

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

$$\rho_i^{(k)} = 2\rho_i^{(k-1)} \quad \text{if} \quad \tilde{f}_i^{(k-1)}(x^{(k)}) < f_i(x^{(k)}) \, ,$$

$$\rho_i^{(k)} = \rho_i^{(k-1)} \quad \text{if} \quad \tilde{f}_i^{(k-1)}(x^{(k)}) \geq f_i(x^{(k)}) \, .$$

Further, if $\tilde{f}_i^{(k-1)}(x^{(k)}) \geq f_i(x^{(k)})$ for all $i \in \{0, 1, \dots, m\}$, the asymptotes should now instead be updated as

$$l_j^{(k)} = x_j^{(k)} - (x_j^{(k-1)} - l_j^{(k-1)}),$$

$$u_j^{(k)} = x_j^{(k)} + (u_j^{(k-1)} - x_j^{(k-1)}).$$

Even though this leads to a globally convergent method, the practical experience is that it in most cases converges slower than the original MMA (on problems where the original MMA does converge). The reason for this is that since the parameters $\rho_i^{(k)}$ are increased but never decreased, the approximations become increasingly conservative. This may eventually lead to very small steps in the iteration process.

7. An interior point method for solving the MMA subproblem.

To simplify the notations, we now skip the index k in the subproblem. Further, we let $b_i = -r_i^{(k)}$, and we drop the constant $r_0^{(k)}$ from the objective function. Then the MMA subproblem becomes

$$\begin{aligned}
&\text{minimize} && g_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2}d_i y_i^2) && \\
&\text{subject to} && g_i(x) - a_i z - y_i \leq b_i\,, && i = 1,\dots,m \\
& && \alpha_j \leq x_j \leq \beta_j\,, && j = 1,\dots,n \\
& && y_i \geq 0, && i = 1,\dots,m \\
& && z \geq 0, &&
\end{aligned} \tag{7.1}$$

where

$$g_i(x) = \sum_{j=1}^{n}\left(\frac{p_{ij}}{u_j - x_j} + \frac{q_{ij}}{x_j - l_j}\right), \quad i = 0, 1, \dots, m.$$

In the following sections, a primal-dual interior point method for solving problems on this form will be described.

7.1. Optimality conditions for the subproblem.

Since the subproblem (7.1) is a convex problem, and since the Slater's constraint qualifications are fulfilled, the Kuhn-Tucker optimality conditions are both necessary and sufficient for an optimal solution of (7.1). In order to state these conditions, we first form the Lagrange function corresponding to (7.1).

$$\begin{aligned}
L(x,y,z,\lambda,\xi,\eta,\mu,\zeta) = \;& g_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2}d_i y_i^2) + \\
& + \sum_{i=1}^{m}\lambda_i(g_i(x) - a_i z - y_i - b_i) + \\
& + \sum_{j=1}^{n}(\xi_j(\alpha_j - x_j) + \eta_j(x_j - \beta_j)) - \sum_{i=1}^{m}\mu_i y_i - \zeta z\,,
\end{aligned} \tag{7.2}$$

where $\lambda = (\lambda_1,\dots,\lambda_m)^T$, $\xi = (\xi_1,\dots,\xi_n)^T$, $\eta = (\eta_1,\dots,\eta_n)^T$, $\mu = (\mu_1,\dots,\mu_m)^T$ and ζ are non-negative Lagrange multipliers for the different constraints in (7.1).

Let

$$\psi(x,\lambda) = g_0(x) + \sum_{i=1}^{m}\lambda_i g_i(x) = \sum_{j=1}^{n}\left(\frac{p_j(\lambda)}{u_j - x_j} + \frac{q_j(\lambda)}{x_j - l_j}\right),$$

where $p_j(\lambda) = p_{0j} + \sum_{i}^{m}\lambda_i p_{ij}$ and $q_j(\lambda) = q_{0j} + \sum_{i}^{m}\lambda_i q_{ij}$.

Then the Lagrange function can be written

$$\begin{aligned} L(x,y,z,\lambda,\xi,\eta,\mu,\zeta) = \ & \psi(x,\lambda) + \sum_{j=1}^{n}(\xi_j(\alpha_j - x_j) + \eta_j(x_j - \beta_j)) + (1-\zeta)z + \\ & + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2}d_i y_i^2 - \lambda_i a_i z - \lambda_i y_i - \lambda_i b_i - \mu_i y_i)\,, \end{aligned} \tag{7.3}$$

and then the Kuhn-Tucker optimality conditions for the subproblem (7.1) become as follows.

$$\begin{aligned} \frac{\partial\psi}{\partial x_j} - \xi_j + \eta_j &= 0, & j &= 1,\dots,n & &(\partial L/\partial x_j = 0) & &(7.4a)\\ c_i + d_i y_i - \lambda_i - \mu_i &= 0, & i &= 1,\dots,m & &(\partial L/\partial y_i = 0) & &(7.4b)\\ 1 - \zeta - \lambda^T a &= 0, & & & &(\partial L/\partial z = 0) & &(7.4c)\\ g_i(x) - a_i z - y_i - b_i &\le 0, & i &= 1,\dots,m & &\text{(primal feasibility)} & &(7.4d)\\ \lambda_i(g_i(x) - a_i z - y_i - b_i) &= 0, & i &= 1,\dots,m & &\text{(compl slackness)} & &(7.4e)\\ \xi_j(\alpha_j - x_j) &= 0, & j &= 1,\dots,n & &\text{(compl slackness)} & &(7.4f)\\ \eta_j(x_j - \beta_j) &= 0, & j &= 1,\dots,n & &\text{(compl slackness)} & &(7.4g)\\ -\mu_i y_i &= 0, & i &= 1,\dots,m & &\text{(compl slackness)} & &(7.4h)\\ -\zeta z &= 0, & & & &\text{(compl slackness)} & &(7.4i)\\ \alpha_j - x_j \le 0 \ \text{ and } \ x_j - \beta_j &\le 0, & j &= 1,\dots,n & &\text{(primal feasibility)} & &(7.4j)\\ -z \le 0 \ \text{ and } \ -y_i &\le 0, & i &= 1,\dots,m & &\text{(primal feasibility)} & &(7.4k)\\ \xi_j \ge 0 \ \text{ and } \ \eta_j &\ge 0, & j &= 1,\dots,n & &\text{(dual feasibility)} & &(7.4l)\\ \zeta \ge 0 \ \text{ and } \ \mu_i &\ge 0, & i &= 1,\dots,m & &\text{(dual feasibility)} & &(7.4m)\\ \lambda_i &\ge 0, & i &= 1,\dots,m & &\text{(dual feasibility)} & &(7.4n) \end{aligned}$$

where

$$\frac{\partial\psi}{\partial x_j} = \frac{p_j(\lambda)}{(u_j - x_j)^2} - \frac{q_j(\lambda)}{(x_j - l_j)^2} \quad \text{and} \quad \lambda^T a = \sum_{i=1}^{m}\lambda_i a_i .$$

7.2. The "ε-relaxed" optimality conditions for the subproblem

When a primal-dual interior point method is used for solving the subproblem (7.1), the zeros in right hand sides of the complementary slackness conditions (7.4e)–(7.4i) are replaced by the negative of a "small" parameter $\varepsilon > 0$. Further, slack variables s_i are introduced for the constraints (7.4d).

The "ε-relaxed" optimality conditions then become

$$\begin{aligned} \frac{\partial\psi}{\partial x_j} - \xi_j + \eta_j &= 0, & j &= 1,\dots,n & &(7.5a)\\ c_i + d_i y_i - \lambda_i - \mu_i &= 0, & i &= 1,\dots,m & &(7.5b)\\ 1 - \zeta - \lambda^T a &= 0. & & & &(7.5c) \end{aligned}$$

$$
\begin{aligned}
g_i(x) - a_i z - y_i + s_i - b_i &= 0, \quad i = 1, \ldots, m && \text{(7.5d)}\\
\xi_j(x_j - \alpha_j) - \varepsilon &= 0, \quad j = 1, \ldots, n && \text{(7.5e)}\\
\eta_j(\beta_j - x_j) - \varepsilon &= 0, \quad j = 1, \ldots, n && \text{(7.5f)}\\
\mu_i y_i - \varepsilon &= 0, \quad i = 1, \ldots, m && \text{(7.5g)}\\
\zeta z - \varepsilon &= 0, && \text{(7.5h)}\\
\lambda_i s_i - \varepsilon &= 0, \quad i = 1, \ldots, m && \text{(7.5i)}\\
x_j - \alpha_j > 0 \text{ and } \xi_j &> 0, \quad j = 1, \ldots, n && \text{(7.5j)}\\
\beta_j - x_j > 0 \text{ and } \eta_j &> 0, \quad j = 1, \ldots, n && \text{(7.5k)}\\
y_i > 0 \text{ and } \mu_i &> 0, \quad i = 1, \ldots, m && \text{(7.5l)}\\
z > 0 \text{ and } \zeta &> 0, && \text{(7.5m)}\\
s_i > 0 \text{ and } \lambda_i &> 0, \quad i = 1, \ldots, m && \text{(7.5n)}
\end{aligned}
$$

For each fixed $\varepsilon > 0$, there exists a unique solution $(x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)$ of these conditions. This follows because (7.5a)–(7.5n) are mathematically (but not numerically) equivalent to the Kuhn-Tucker conditions of the following strictly convex problem in the variables x, y, z and s.

$$
\begin{aligned}
\text{minimize} \quad & g_0(x) + z + \sum_{i=1}^{m}(c_i y_i + \tfrac{1}{2} d_i y_i^2) + \\
& + \sum_{j=1}^{n}(-\varepsilon \log(x_j - \alpha_j) - \varepsilon \log(\beta_j - x_j)) + \\
& + \sum_{i=1}^{m}(-\varepsilon \log y_i - \varepsilon \log s_i) - \varepsilon \log z \qquad \text{(7.6)}\\
\text{subject to} \quad & g_i(x) - a_i z - y_i + s_i \leq b_i, \quad i = 1, \ldots, m \\
& (\alpha_j < x_j < \beta_j,\ y_i > 0,\ z > 0,\ s_i > 0)
\end{aligned}
$$

where the strict inequalities will automatically be satisfied because of the logarithm terms in the objective function.

7.3. A Newton direction for the "ε-relaxed" optimality conditions.

Given a point $(x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)$ such that (7.5j)–(7.5n) are satisfied. If, starting from this point, Newton's method should be applied to the system of nonlinear equations (7.5a)–(7.5i), the following system of linear equations should be generated and solved.

$$\begin{pmatrix} \Psi & & & G^T & -I & I & & & \\ & \langle d\rangle & & -I & & & -I & & \\ & & & -a^T & & & & -1 & \\ G & -I & -a & & & & & & I \\ \langle \xi\rangle & & & & \langle x-\alpha\rangle & & & & \\ -\langle \eta\rangle & & & & & \langle \beta-x\rangle & & & \\ & \langle \mu\rangle & & & & & \langle y\rangle & & \\ & & \zeta & & & & & z & \\ & & & \langle s\rangle & & & & & \langle \lambda\rangle \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\lambda \\ \Delta\xi \\ \Delta\eta \\ \Delta\mu \\ \Delta\zeta \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\delta_1 \\ -\delta_2 \\ -\delta_3 \\ -\delta_4 \\ -\delta_5 \\ -\delta_6 \\ -\delta_7 \\ -\delta_8 \\ -\delta_9 \end{pmatrix}$$

where $\delta_1, \ldots, \delta_9$ are defined by the left hand sides in (7.5a)–(7.5i),

Ψ is an $n \times n$ diagonal matrix with $(\Psi)_{jj} = \dfrac{\partial^2 \psi}{\partial x_j^2} = \dfrac{2p_j(\lambda)}{(u_j - x_j)^3} + \dfrac{2q_j(\lambda)}{(x_j - l_j)^3}$,

G is an $m \times n$ matrix with $(G)_{ij} = \dfrac{\partial g_i}{\partial x_j} = \dfrac{p_{ij}}{(u_j - x_j)^2} - \dfrac{q_{ij}}{(x_j - l_j)^2}$,

I is a unit matrix, with dimensions apparent from its position,

$\langle d\rangle$ is a diagonal matrix with the vector $d = (d_1, \ldots, d_m)^T$ on the diagonal,

$\langle x-\alpha\rangle$ is a diagonal matrix with the vector $x-\alpha$ on the diagonal, etc.

In the above Newton system, $\Delta\xi$, $\Delta\eta$, $\Delta\mu$, $\Delta\zeta$ and Δs can be eliminated through

$$\Delta\xi = -\langle x-\alpha\rangle^{-1}\langle\xi\rangle\Delta x - \xi + \varepsilon\langle x-\alpha\rangle^{-1}e\,, \tag{7.7a}$$

$$\Delta\eta = \langle\beta-x\rangle^{-1}\langle\eta\rangle\Delta x - \eta + \varepsilon\langle\beta-x\rangle^{-1}e\,, \tag{7.7b}$$

$$\Delta\mu = -\langle y\rangle^{-1}\langle\mu\rangle\Delta y - \mu + \varepsilon\langle y\rangle^{-1}e\,, \tag{7.7c}$$

$$\Delta\zeta = -z^{-1}\zeta\Delta z - \zeta + \varepsilon z^{-1}\,, \tag{7.7d}$$

$$\Delta s = -\langle\lambda\rangle^{-1}\langle s\rangle\Delta\lambda - s + \varepsilon\langle\lambda\rangle^{-1}e\,, \tag{7.7e}$$

where $e = (1, \ldots, 1)^T$, with dimension apparent from the context.

Then the following partially reduced system in $(\Delta x, \Delta y, \Delta z, \Delta\lambda)$ is obtained.

$$\begin{pmatrix} D_x & & & G^T \\ & D_y & & -I \\ & & D_z & -a^T \\ G & -I & -a & D_\lambda \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} -\delta_x \\ -\delta_y \\ -\delta_z \\ -\delta_\lambda \end{pmatrix}$$

where

$D_x = \Psi + \langle x-\alpha\rangle^{-1}\langle\xi\rangle + \langle\beta-x\rangle^{-1}\langle\eta\rangle\,,$

$D_y = \langle d\rangle + \langle y\rangle^{-1}\langle\mu\rangle\,,$

$D_z = \zeta/z$ (a scalar),

$D_\lambda = -\langle\lambda\rangle^{-1}\langle s\rangle\,,$

$\delta_x = \frac{\partial\psi}{\partial x} - \varepsilon\langle x-\alpha\rangle^{-1}e + \varepsilon\langle\beta-x\rangle^{-1}e\,,$

$\delta_y = c + \langle d\rangle y - \lambda - \varepsilon\langle y\rangle^{-1}e\,,$

$\delta_z = 1 - \varepsilon z^{-1} - \lambda^T a$ (a scalar),

$\delta_\lambda = g(x) - az - y - b + \varepsilon\langle\lambda\rangle^{-1}e\,.$

In this partially reduced system, Δx, Δy and Δz can be eliminated through

$$\Delta x = -D_x^{-1}G^T\Delta\lambda - D_x^{-1}\delta_x\,, \tag{7.8a}$$

$$\Delta y = D_y^{-1}\Delta\lambda - D_y^{-1}\delta_y\,, \tag{7.8b}$$

$$\Delta z = D_z^{-1}a^T\Delta\lambda - D_z^{-1}\delta_z\,. \tag{7.8c}$$

Then, finally, the following reduced system in $\Delta\lambda$ is obtained.

$$\widehat{D}_\lambda\Delta\lambda = -\widehat{\delta}_\lambda \tag{7.9}$$

where

$\widehat{D}_\lambda = D_\lambda - GD_x^{-1}G^T - D_y^{-1} - aD_z^{-1}a^T$, and

$\widehat{\delta}_\lambda = \delta_\lambda - GD_x^{-1}\delta_x + D_y^{-1}\delta_y + aD_z^{-1}\delta_z\,.$

It should be noted that $\widehat{D}_\lambda$ is symmetric and negative definite.

7.4. Line search in the Newton direction.

When the direction $(\Delta x, \Delta y, \Delta z, \Delta\lambda, \Delta\xi, \Delta\eta, \Delta\mu, \Delta\zeta, \Delta s)$ has been calculalated, a step should be taken in that direction. A "pure" Newton step would be to take a step equal to the calculated direction. But this step might lead to a point where some of the variables are negative, which is not feasible. Therefore, we first let θ be the largest number such that $\theta \leq 1$ and

$$x_j + \theta\Delta x_j - \alpha_j \geq 0.01(x_j - \alpha_j) \text{ for all } j,$$

$$\beta_j - (x_j + \theta\Delta x_j) \geq 0.01(\beta_j - x_j) \text{ for all } j,$$

$$y_i + \theta\Delta y_i \geq 0.01 y_i \quad \text{for all } i,$$

$$z + \theta\Delta z \geq 0.01z,$$

$$\lambda_i + \theta\Delta\lambda_i \geq 0.01\lambda_i \quad \text{for all } i,$$

$$\xi_j + \theta\Delta\xi_j \geq 0.01\xi_j \quad \text{for all } j,$$

$$\eta_j + \theta\Delta\eta_j \geq 0.01\eta_j \quad \text{for all } j,$$

$$\mu_i + \theta\Delta\mu_i \geq 0.01\mu_i \quad \text{for all } i,$$

$$\zeta + \theta\Delta\zeta \geq 0.01\zeta,$$

$$s_i + \theta\Delta s_i \geq 0.01 s_i \quad \text{for all } i.$$

Now, if a step equal to θ times the Newton direction is taken, the resulting point will have all variables strictly positive. But in order to guarrantee convergence, the new point should also be in some sense better than the previous. Therefore, we next let τ be the largest of θ, $\theta/2$, $\theta/4$, $\theta/8, \ldots$ such that

$$||\delta((x, y, z, \lambda, \xi, \eta, \mu, \zeta, s) + \tau \cdot (\Delta x, \Delta y, \Delta z, \Delta\lambda, \Delta\xi, \Delta\eta, \Delta\mu, \Delta\zeta, \Delta s))|| <$$
$$< \; ||\delta(x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)||,$$

where $\delta(x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)$ is the residual vector defined by the left hand sides in the ε-relaxed KKT conditions (7.5a)–(7.5i), and $||\cdot||$ is the ordinary Euclidian norm.

This is always possible to obtain since the Newton direction is a descent direction for $||\delta(x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)||$.

7.5. The complete iteration scheme for solving the MMA subproblem.

First of all, $\varepsilon^{(1)}$ and a starting point $(x^{(1)}, y^{(1)}, z^{(1)}, \lambda^{(1)}, \xi^{(1)}, \eta^{(1)}, \mu^{(1)}, \zeta^{(1)}, s^{(1)})$ which satisfies (7.5j)–(7.5n) are chosen. The following is a simple but reasonable choice.

$\varepsilon^{(1)} = 1,\ \ x_j^{(1)} = \frac{1}{2}(\alpha_j + \beta_j),\ \ y_i^{(1)} = 1,\ \ z^{(1)} = 1,\ \ \lambda_i^{(1)} = 1,$

$\xi_j^{(1)} = 1/(x_j^{(1)} - \alpha_j),\ \ \eta_j^{(1)} = 1/(\beta_j - x_j^{(1)}),\ \ \mu_i^{(1)} = 1,\ \ \zeta^{(1)} = 1,\ \ s_i^{(1)} = 1.$

A typical iteration, leading from the ν:th iteration point to the $(\nu+1)$:th iteration point, consists of the following steps.

Step 1 (*direction finding*):

For given $\varepsilon^{(\nu)}$ and $(x^{(\nu)}, y^{(\nu)}, z^{(\nu)}, \lambda^{(\nu)}, \xi^{(\nu)}, \eta^{(\nu)}, \mu^{(\nu)}, \zeta^{(\nu)}, s^{(\nu)})$ which satisfy (7.5j)–(7.5n), calculate $(\Delta x^{(\nu)}, \Delta y^{(\nu)}, \Delta z^{(\nu)}, \Delta\lambda^{(\nu)}, \Delta\xi^{(\nu)}, \Delta\eta^{(\nu)}, \Delta\mu^{(\nu)}, \Delta\zeta^{(\nu)}, \Delta s^{(\nu)})$ by solving the system of linear equations described above.

Step 2 (*linesearch*):

Calculate a steplength $\tau^{(\nu)}$ as described in the previous section.

Step 3 (*updating*):

Let $(x^{(\nu+1)}, y^{(\nu+1)}, z^{(\nu+1)}, \lambda^{(\nu+1)}, \xi^{(\nu+1)}, \eta^{(\nu+1)}, \mu^{(\nu+1)}, \zeta^{(\nu+1)}, s^{(\nu+1)}) =$
$(x^{(\nu)}, y^{(\nu)}, z^{(\nu)}, \lambda^{(\nu)}, \xi^{(\nu)}, \eta^{(\nu)}, \mu^{(\nu)}, \zeta^{(\nu)}, s^{(\nu)}) +$
$\tau^{(\nu)} \cdot (\Delta x^{(\nu)}, \Delta y^{(\nu)}, \Delta z^{(\nu)}, \Delta\lambda^{(\nu)}, \Delta\xi^{(\nu)}, \Delta\eta^{(\nu)}, \Delta\mu^{(\nu)}, \Delta\zeta^{(\nu)}, \Delta s^{(\nu)})$.

Step 4 (*perhaps decreasing* ε):

If $\|\delta(x^{(\nu+1)}, y^{(\nu+1)}, z^{(\nu+1)}, \lambda^{(\nu+1)}, \xi^{(\nu+1)}, \eta^{(\nu+1)}, \mu^{(\nu+1)}, \zeta^{(\nu+1)}, s^{(\nu+1)})\| < \varepsilon^{(\nu)}$,
let $\varepsilon^{(\nu+1)} = \varepsilon^{(\nu)}/10$. Otherwise, let $\varepsilon^{(\nu+1)} = \varepsilon^{(\nu)}$.

Increase ν by 1 and go to Step 1.

The process is terminated when $\varepsilon^{(\nu)}$ has become sufficiently small, say $\varepsilon^{(\nu)} < 10^{-7}$.

8. An implementation of MMA in Matlab.

The method described in this paper has been implemented in Matlab 5. This section contains:

- the head of a Matlab m-file containing the function "mmasub",

- an m-file used to define a simple test problem (cantilever beam),

- an m-file used to initialize some vectors for the test problem,

- an m-file used as a "main program" for the test proplem,

- a Matlab run with the above m-files.

The testproblem is a simple cantilever beam problem which can be formulated explicitly as follows.

$$\begin{aligned} &\text{minimize} && x_1 + x_2 + x_3 + x_4 + x_5 \\ &\text{subject to} && 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 \ \leq 1 \\ & && 1 \leq x_j \leq 10, \quad j = 1, \ldots, 5 \end{aligned} \tag{8.1}$$

```
%************************************************************
%    This is the file mmasub.m
%
function [xmma,ymma,zmma,lamma,low,upp] = ...
mmasub(m,n,iter,epsimin,xval,xmin,xmax,xold1,xold2, ...
f0val,df0dx,df0dx2,fval,dfdx,dfdx2,low,upp,a,c,d);
%
%    written in June 1998 by
%
%    Krister Svanberg (krille@math.kth.se)
%    Optimization and Systems Theory, KTH,
%    SE-10044 Stockholm, Sweden.
%
%    mmasub performs one MMA-iteration, aimed at
%    solving the nonlinear programming problem:
%
%      Minimize  f_0(x) + z + sum( c_i*y_i + 0.5*d_i*(y_i)^2 )
%    subject to  f_i(x) - a_i*z - y_i <= 0,  i = 1,...,m
%                xmax_j <= x_j <= xmin_j,    j = 1,...,n
%                z >= 0,   y_i >= 0,         i = 1,...,m
%*** INPUT:
%
%   m    = The number of constraints (see above).
%   n    = The number of variables x_j (see above).
%  iter  = Current iteration number ( =1 the first time mmasub is called).
%epsimin = Tolerance parameter used in the termination criteria for
%          the MMA subproblem (smallest value of epsi).
%  xval  = Column vector with the current values of the variables x_j.
%  xmin  = Column vector with the lower bounds for the variables x_j.
%  xmax  = Column vector with the upper bounds for the variables x_j.
%  xold1 = xval, one iteration ago (provided that iter>1).
%  xold2 = xval, two iterations ago (provided that iter>2).
%  f0val = The value of the objective function f_0 at xval.
%  df0dx = Column vector with the derivatives of the objective function f_0
%          with respect to the variables x_j, calculated at xval.
% df0dx2 = Column vector with the non-mixed second derivatives of the
%          objective function f_0 with respect to the variables x_j,
%          calculated at xval.
%  fval  = Column vector with the values of the constraint functions f_i,
%          calculated at xval.
%  dfdx  = (m x n)-matrix with the derivatives of the constraint functions
%          f_i with respect to the variables x_j, calculated at xval.
%          dfdx(i,j) = the derivative of f_i with respect to x_j.
%  dfdx2 = (m x n)-matrix with the non-mixed second derivatives of the
%          constraint functions f_i with respect to the variables x_j,
%          calculated at xval.
%          dfdx2(i,j) = the second derivative of f_i with respect to x_j.
%  low   = Column vector with the lower asymptotes from the previous
%          iteration (provided that iter>1).
%  upp   = Column vector with the upper asymptotes from the previous
%          iteration (provided that iter>1).
%  a     = Column vector with the constants a_i in the terms a_i*z.
%  c     = Column vector with the constants c_i in the terms c_i*y_i.
%  d     = Column vector with the constants d_i in the terms 0.5*d_i*(y_i)^2.
%
%*** OUTPUT:
%
%  xmma  = Column vector with the optimal values of the variables x_j
%          in the current MMA subproblem.
%  ymma  = Column vector with the optimal values of the variables y_i
%          in the current MMA subproblem.
%  zmma  = Scalar with the optimal value of the variable z
%          in the current MMA subproblem.
%  lamma = Column vector with the optimal values of the dual variables
%          (Lagrange multipliers) in the current MMA subproblem.
%  low   = Column vector with the lower asymptotes, calculated and used
%          in the current MMA subproblem.
%  upp   = Column vector with the upper asymptotes, calculated and used
%          in the current MMA subproblem.
%
%***********************************************************
%   This is the file beam.m
%   which defines the cantilever beam problem.
%
function [f0val,df0dx,df0dx2,fval,dfdx,dfdx2] = beam(x);
%
%   written in June 1998 by
%
%   Krister Svanberg (krille@math.kth.se)
%   Optimization and Systems Theory, KTH,
%   SE-10044 Stockholm, Sweden.
```

```
%
e = [ 1 1 1 1 1 ]';
df0dx = e;
df0dx2 = 0*e;
f0val = df0dx'*x;
coef = [61 37 19 7 1]';
x2 = x.*x;
x3 = x2.*x;
x4 = x2.*x2;
x5 = x3.*x2;
x3inv = e./x3;
fval = coef'*x3inv - 1;
dfdx = -3*(coef./x4)';
dfdx2 = 12*(coef./x5)';
%************************************************************
%    This is the file beaminit.m
%    in which some vectors for the cantilever
%    beam problem are initialized.
%
%    written in June 1998 by
%
%    Krister Svanberg (krille@math.kth.se)
%    Optimization and Systems Theory, KTH,
%    SE-10044 Stockholm, Sweden.
%
  m = 1;
  n = 5;
  epsimin = 0.000000005;
  xval  = 5*ones(n,1);
  xold1 = zeros(n,1);
  xold2 = zeros(n,1);
  low   = zeros(n,1);
  upp   = zeros(n,1);
  xmin  = ones(n,1);
  xmax  = 10*ones(n,1);
  c = 1000*ones(m,1);
  d = zeros(m,1);
  a = zeros(m,1);
  [f0val,df0dx,df0dx2,fval,dfdx,dfdx2] = beam(xval);
  outvector = [f0val fval xval']'
  iter = 0;
%************************************************************
%    This is the file beammain.m
%    which is used as a main program for
%    the cantilever beam problem.
%
%    written in June 1998 by
%
%    Krister Svanberg (krille@math.kth.se)
%    Optimization and Systems Theory, KTH,
%    SE-10044 Stockholm, Sweden.
```

```
%
itte = 0;
while itte < maxite
  iter = iter+1
  itte = itte+1;

  [xmma,ymma,zmma,lamma,low,upp] = ...
  mmasub(m,n,iter,epsimin,xval,xmin,xmax,xold1,xold2, ...
  f0val,df0dx,df0dx2,fval,dfdx,dfdx2,low,upp,a,c,d);

  xold2 = xold1;
  xold1 = xval;
  xval = xmma;
  [f0val,df0dx,df0dx2,fval,dfdx,dfdx2] = beam(xval);
  outvector = [f0val fval xval']'
end
%*************************************************************

>> beaminit

outvector =
    25
     0
     5
     5
     5
     5
     5

>> maxite=3;
>> beammain

iter =    1
outvector =
  21.23671540968126
   0.05143688305828
   5.53199378990684
   5.19640664935817
   4.65148408913184
   3.72484970364471
   2.13198117763970

iter =    2
outvector =
  21.49392433684576
  -0.00030008977653
   5.84532737946843
   5.30615287587694
   4.58356436951021
   3.58975283225656
   2.16912687973362
```

```
iter =    3
outvector =
  21.47555761630785
  -0.00000162629643
   5.95683551645921
   5.31238291165642
   4.52326975413410
   3.52438212611257
   2.15868730794555

>> beammain

iter =    4
outvector =
  21.47382497571198
  -0.00000035796458
   5.99800703382267
   5.31118339118582
   4.50259563126298
   3.50698984583424
   2.15504907360627

iter =    5
outvector =
  21.47367102202199
  -0.00000001754855
   6.01116643542795
   5.31009073351918
   4.49635193271211
   3.50262846058030
   2.15343345978245

iter =    6
outvector =
  21.47366026272084
  -0.00000000182023
   6.01486128269035
   5.30950079917009
   4.49474808025951
   3.50167851835040
   2.15287158225050
```

The academic user can get these Matlab m-files for free. Just send an e-mail to the author, explaining the intended usage of the code.