

MMA and GCMMA – two methods for nonlinear optimization

Krister Svanberg
Optimization and Systems Theory,
KTH, Stockholm, Sweden.
krille@math.kth.se

This note describes the algorithms used in the author’s 2007 implementations of MMA and GCMMA in Matlab. The first versions of these methods were published in [1] and [2].

1. Considered optimization problem

Throughout this note, optimization problems of the following form are considered, where the variables are $\mathbf{x} = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$, $\mathbf{y} = (y_1, \dots, y_m)^\top \in \mathbb{R}^m$, and $z \in \mathbb{R}$.

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}) + a_0 z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) \\ & \text{subject to} && f_i(\mathbf{x}) - a_i z - y_i \leq 0, \quad i = 1, \dots, m \\ & && \mathbf{x} \in X, \quad \mathbf{y} \geq \mathbf{0}, \quad z \geq 0. \end{aligned} \tag{1.1}$$

Here, $X = \{\mathbf{x} \in \mathbb{R}^n \mid x_j^{\min} \leq x_j \leq x_j^{\max}, j = 1, \dots, n\}$, where x_j^{\min} and x_j^{\max} are given real numbers which satisfy $x_j^{\min} < x_j^{\max}$ for all j , f_0, f_1, \dots, f_m are given, continuously differentiable, real-valued functions on X , a_0, a_i, c_i and d_i are given real numbers which satisfy $a_0 > 0$, $a_i \geq 0$, $c_i \geq 0$, $d_i \geq 0$ and $c_i + d_i > 0$ for all i , and also $a_i c_i > a_0$ for all i with $a_i > 0$.

In (1.1), the “natural” optimization variables are x_1, \dots, x_n , while y_1, \dots, y_m and z are “artificial” optimization variables which should make it easier for the user to formulate and solve certain subclasses of problems, like least squares problems and minmax problems.

As a first example, assume that the user wants to solve a problem on the following “standard” form for nonlinear programming.

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & && \mathbf{x} \in X, \end{aligned} \tag{1.2}$$

where f_0, f_1, \dots, f_m are given differentiable functions and X is as above. To make problem (1.1) (almost) equivalent to this problem (1.2), first let $a_0 = 1$ and $a_i = 0$ for all $i > 0$. Then $z = 0$ in any optimal solution of (1.1). Further, for each i , let $d_i = 1$ and $c_i =$ “a large number”, so that the variables y_i become “expensive”. Then typically $\mathbf{y} = \mathbf{0}$ in any optimal solution of (1.1), and the corresponding \mathbf{x} is an optimal solution of (1.2). It should be noted

that the problem (1.1) always has feasible solutions, and in fact also at least one optimal solution. This holds even if the user's problem (1.2) does not have any feasible solutions, in which case some $y_i > 0$ in the optimal solution of (1.1).

As a second example, assume that the user wants to solve a "min-max" problem on the form

$$\begin{aligned} & \text{minimize} && \max_{i=1,\dots,p} \{h_i(\mathbf{x})\} \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \\ & && \mathbf{x} \in X. \end{aligned} \tag{1.3}$$

where h_i and g_i are given differentiable functions and X is as above. It is assumed that the functions h_i are non-negative on X (possibly after adding a sufficiently large constant C to each of them, the same C for all h_i). For each given $\mathbf{x} \in X$, the value of the objective function in problem (1.3) is the largest of the p real numbers $h_1(\mathbf{x}), \dots, h_p(\mathbf{x})$. Problem (1.3) may equivalently be written on the following form with variables $\mathbf{x} \in \mathbb{R}^n$ and $z \in \mathbb{R}$:

$$\begin{aligned} & \text{minimize} && z \\ & \text{subject to} && z \geq h_i(\mathbf{x}), \quad i = 1, \dots, p \\ & && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \\ & && \mathbf{x} \in X, \quad z \geq 0. \end{aligned} \tag{1.4}$$

To make problem (1.1) (almost) equivalent to this problem (1.4), let $m = p + q$, $f_0(\mathbf{x}) = 0$, $f_i(\mathbf{x}) = h_i(\mathbf{x})$ for $i = 1, \dots, p$, $f_{p+i}(\mathbf{x}) = g_i(\mathbf{x})$ for $i = 1, \dots, q$, $a_0 = a_1 = \dots = a_p = 1$, $a_{p+1} = \dots = a_{p+q} = 0$, $d_1 = \dots = d_m = 1$, $c_1 = \dots = c_m =$ "a large number".

As a third example, assume that the user wants to solve a constrained least squares problem on the form

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \sum_{i=1}^p (h_i(\mathbf{x}))^2 \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \\ & && \mathbf{x} \in X. \end{aligned} \tag{1.5}$$

where h_i and g_i are given differentiable functions and X is as above. Problem (1.5) may equivalently be written on the following form with variables $\mathbf{x} \in \mathbb{R}^n$ and $y_1, \dots, y_{2p} \in \mathbb{R}$:

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \sum_{i=1}^p (y_i^2 + y_{p+i}^2) \\ & \text{subject to} && y_i \geq h_i(\mathbf{x}), \quad i = 1, \dots, p \\ & && y_{p+i} \geq -h_i(\mathbf{x}), \quad i = 1, \dots, p \\ & && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q \\ & && y_i \geq 0 \text{ and } y_{p+i} \geq 0, \quad i = 1, \dots, p \\ & && \mathbf{x} \in X. \end{aligned} \tag{1.6}$$

To make problem (1.1) (almost) equivalent to this problem (1.6), let $m = 2p + q$, $f_0(\mathbf{x}) = 0$, $f_i(\mathbf{x}) = h_i(\mathbf{x})$ for $i = 1, \dots, p$, $f_{p+i}(\mathbf{x}) = -h_i(\mathbf{x})$ for $i = 1, \dots, p$, $f_{2p+i}(\mathbf{x}) = g_i(\mathbf{x})$ for $i = 1, \dots, q$, $a_0 = 1$, $a_1 = \dots = a_m = 0$, $d_1 = \dots = d_m = 1$, $c_1 = \dots = c_{2p} = 0$, $c_{2p+1} = \dots = c_{2p+q} =$ "a large number".

2. Some practical considerations

In many applications, the constraints are on the form $g_i(\mathbf{x}) \leq g_i^{\max}$, where $g_i(\mathbf{x})$ stands for e.g. a certain stress, while g_i^{\max} is the largest permitted value on this stress. This means that $f_i(\mathbf{x}) = g_i(\mathbf{x}) - g_i^{\max}$ (in (1.1) as well as in (1.2)). The user should then preferably scale the constraints in such a way that $1 \leq g_i^{\max} \leq 100$ for each i (and not $g_i^{\max} = 10^{10}$). The objective function $f_0(\mathbf{x})$ should preferably be scaled such that $1 \leq f_0(\mathbf{x}) \leq 100$ for reasonable values on the variables. The variables x_j should preferably be scaled such that $0.1 \leq x_j^{\max} - x_j^{\min} \leq 100$, for all j .

Concerning the “large numbers” on the coefficients c_i (mentioned above), the user should for numerical reasons try to avoid “extremely large” values on these coefficients (like 10^{10}). 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 the corresponding values of c_i 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 “reasonably large” values on the parameters c_i could be, say, $c_i = 1000$ or 10000 .

Finally, concerning the simple bound constraints $x_j^{\min} \leq x_j \leq x_j^{\max}$, 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 \mathbf{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.

3. The ordinary MMA

MMA is a method for solving problems on the form (1.1), using the following approach: In each iteration, the current iteration point $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, z^{(k)})$ is given. Then an approximating subproblem, in which the functions $f_i(\mathbf{x})$ are replaced by certain convex functions $\tilde{f}_i^{(k)}(\mathbf{x})$, is generated. The choice of these approximating functions is based mainly on gradient information at the current iteration point, but also on some parameters $u_j^{(k)}$ and $l_j^{(k)}$ (“moving asymptotes”) which are updated in each iteration based on information from previous iteration points. The subproblem is solved, and the unique optimal solution becomes the next iteration point $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)})$. Then a new subproblem is generated, etc.

The MMA subproblem looks as follows:

$$\begin{aligned}
& \text{minimize} && \tilde{f}_0^{(k)}(\mathbf{x}) + a_0 z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) \\
& \text{subject to} && \tilde{f}_i^{(k)}(\mathbf{x}) - a_i z - y_i \leq 0, && i = 1, \dots, m \\
& && \alpha_j^{(k)} \leq x_j \leq \beta_j^{(k)}, && j = 1, \dots, n, \\
& && y_i \geq 0, && i = 1, \dots, m \\
& && z \geq 0.
\end{aligned} \tag{3.1}$$

In this subproblem (3.1), the approximating functions $\tilde{f}_i^{(k)}(\mathbf{x})$ are chosen as

$$\tilde{f}_i^{(k)}(\mathbf{x}) = \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - l_j^{(k)}} \right) + r_i^{(k)}, \quad i = 0, 1, \dots, m, \tag{3.2}$$

where

$$p_{ij}^{(k)} = (u_j^{(k)} - x_j^{(k)})^2 \left(1.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^+ + 0.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^- + \frac{10^{-5}}{x_j^{\max} - x_j^{\min}} \right), \tag{3.3}$$

$$q_{ij}^{(k)} = (x_j^{(k)} - l_j^{(k)})^2 \left(0.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^+ + 1.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^- + \frac{10^{-5}}{x_j^{\max} - x_j^{\min}} \right), \tag{3.4}$$

$$r_i^{(k)} = f_i(\mathbf{x}^{(k)}) - \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - l_j^{(k)}} \right). \tag{3.5}$$

Here, $\left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^+$ denotes the largest of the two numbers $\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)})$ and 0, while $\left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^-$ denotes the largest of the two numbers $-\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)})$ and 0.

The bounds $\alpha_j^{(k)}$ and $\beta_j^{(k)}$ in (3.1) and (5.1) are chosen as

$$\alpha_j^{(k)} = \max \{ x_j^{\min}, l_j^{(k)} + 0.1(x_j^{(k)} - l_j^{(k)}), x_j^{(k)} - 0.5(x_j^{\max} - x_j^{\min}) \}, \tag{3.6}$$

$$\beta_j^{(k)} = \min \{ x_j^{\max}, u_j^{(k)} - 0.1(u_j^{(k)} - x_j^{(k)}), x_j^{(k)} + 0.5(x_j^{\max} - x_j^{\min}) \}, \tag{3.7}$$

which means that the constraints $\alpha_j^{(k)} \leq x_j \leq \beta_j^{(k)}$ are equivalent to the following three sets of constraints:

$$x_j^{\min} \leq x_j \leq x_j^{\max}, \tag{3.8}$$

$$-0.9(x_j^{(k)} - l_j^{(k)}) \leq x_j - x_j^{(k)} \leq 0.9(u_j^{(k)} - x_j^{(k)}), \tag{3.9}$$

$$-0.5(x_j^{\max} - x_j^{\min}) \leq x_j - x_j^{(k)} \leq 0.5(x_j^{\max} - x_j^{\min}). \tag{3.10}$$

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 $k = 2$,

$$\begin{aligned} l_j^{(k)} &= x_j^{(k)} - 0.5(x_j^{\max} - x_j^{\min}), \\ u_j^{(k)} &= x_j^{(k)} + 0.5(x_j^{\max} - x_j^{\min}). \end{aligned} \quad (3.11)$$

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

$$\begin{aligned} l_j^{(k)} &= x_j^{(k)} - \gamma_j^{(k)}(x_j^{(k-1)} - l_j^{(k-1)}), \\ u_j^{(k)} &= x_j^{(k)} + \gamma_j^{(k)}(u_j^{(k-1)} - x_j^{(k-1)}), \end{aligned} \quad (3.12)$$

where

$$\gamma_j^{(k)} = \begin{cases} 0.7 & \text{if } (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) < 0, \\ 1.2 & \text{if } (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) > 0, \\ 1 & \text{if } (x_j^{(k)} - x_j^{(k-1)})(x_j^{(k-1)} - x_j^{(k-2)}) = 0, \end{cases} \quad (3.13)$$

provided that this leads to values that satisfy

$$\begin{aligned} l_j^{(k)} &\leq x_j^{(k)} - 0.01(x_j^{\max} - x_j^{\min}), \\ l_j^{(k)} &\geq x_j^{(k)} - 10(x_j^{\max} - x_j^{\min}), \\ u_j^{(k)} &\geq x_j^{(k)} + 0.01(x_j^{\max} - x_j^{\min}), \\ u_j^{(k)} &\leq x_j^{(k)} + 10(x_j^{\max} - x_j^{\min}). \end{aligned} \quad (3.14)$$

If any of these bounds is violated, the corresponding $l_j^{(k)}$ or $u_j^{(k)}$ is put to the right hand side of the violated inequality.

Note that most of the explicit numbers in the above expressions are just default values of different parameters in the fortran code. More precisely:

The number 10^{-5} in (3.3) and (3.4) is the default value of the parameter `raa0`.

The number 0.1 in (3.6) and (3.7) is the default value of the parameter `albefa`.

The number 0.5 in (3.6) and (3.7) is the default value of the parameter `move`.

The number 0.5 in (3.11) is the default value of the parameter `asyinit`.

The number 0.7 in (3.13) is the default value of the parameter `asydecr`.

The number 1.2 in (3.13) is the default value of the parameter `asyincr`.

All these values can be carefully changed by the user. As an example, a more conservative method is obtain by decreasing `move` and/or `asyinit` and/or `asyincr`.

4. GCMMA – the globally convergent version of MMA

The globally convergent version of MMA, from now on called GCMMA, for solving problems of the form (1.1) consists of “outer” and “inner” iterations. The index k is used to denote the outer iteration number, while the index ν is used to denote the inner iteration number. Within each outer iteration, there may be zero, one, or several inner iterations. The double index (k, ν) is used to denote the ν :th inner iteration within the k :th outer iteration.

The first iteration point is obtained by first choosing $\mathbf{x}^{(1)} \in X$ and then choosing $\mathbf{y}^{(1)}$ and $z^{(1)}$ such that $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, z^{(1)})$ becomes a feasible solution of (1.1). This is easy. An outer iteration of the method, going from the k :th iteration point $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, z^{(k)})$ to the $(k+1)$:th iteration point $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)})$, can be described as follows:

Given $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, z^{(k)})$, an approximating subproblem is generated and solved. In this subproblem, the functions $f_i(\mathbf{x})$ are replaced by certain convex functions $\tilde{f}_i^{(k,0)}(\mathbf{x})$. The optimal solution of this subproblem is denoted $(\hat{\mathbf{x}}^{(k,0)}, \hat{\mathbf{y}}^{(k,0)}, \hat{z}^{(k,0)})$. If $\tilde{f}_i^{(k,0)}(\hat{\mathbf{x}}^{(k,0)}) \geq f_i(\hat{\mathbf{x}}^{(k,0)})$, for all $i = 0, 1, \dots, m$, the next iteration point becomes $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)}) = (\hat{\mathbf{x}}^{(k,0)}, \hat{\mathbf{y}}^{(k,0)}, \hat{z}^{(k,0)})$, and the outer iteration is completed (without any inner iterations needed). Otherwise, an inner iteration is made, which means that a new subproblem is generated and solved at $\mathbf{x}^{(k)}$, with new approximating functions $\tilde{f}_i^{(k,1)}(\mathbf{x})$ which are more *conservative* than $\tilde{f}_i^{(k,0)}(\mathbf{x})$ for those indices i for which the above inequality was violated. (By this we mean that $\tilde{f}_i^{(k,1)}(\mathbf{x}) > \tilde{f}_i^{(k,0)}(\mathbf{x})$ for all $\mathbf{x} \in X^{(k)}$, except for $\mathbf{x} = \mathbf{x}^{(k)}$ where $\tilde{f}_i^{(k,1)}(\mathbf{x}^{(k)}) = \tilde{f}_i^{(k,0)}(\mathbf{x}^{(k)})$.) The optimal solution of this new subproblem is denoted $(\hat{\mathbf{x}}^{(k,1)}, \hat{\mathbf{y}}^{(k,1)}, \hat{z}^{(k,1)})$. If $\tilde{f}_i^{(k,1)}(\hat{\mathbf{x}}^{(k,1)}) \geq f_i(\hat{\mathbf{x}}^{(k,1)})$, for all $i = 0, 1, \dots, m$, the next iteration point becomes $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)}) = (\hat{\mathbf{x}}^{(k,1)}, \hat{\mathbf{y}}^{(k,1)}, \hat{z}^{(k,1)})$, and the outer iteration is completed (with one inner iterations needed). Otherwise, another inner iteration is made, which means that a new subproblem is generated and solved at $\mathbf{x}^{(k)}$, with new approximating functions $\tilde{f}_i^{(k,2)}(\mathbf{x})$, etc. These inner iterations are repeated until $\tilde{f}_i^{(k,\nu)}(\hat{\mathbf{x}}^{(k,\nu)}) \geq f_i(\hat{\mathbf{x}}^{(k,\nu)})$ for all $i = 0, 1, \dots, m$, which always happens after a finite (usually small) number of inner iterations. Then the next iteration point becomes $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)}) = (\hat{\mathbf{x}}^{(k,\nu)}, \hat{\mathbf{y}}^{(k,\nu)}, \hat{z}^{(k,\nu)})$, and the outer iteration is completed (with ν inner iterations needed).

It should be noted that in each inner iteration, there is no need to recalculate the gradients $\nabla f_i(\mathbf{x}^{(k)})$, since $\mathbf{x}^{(k)}$ has not changed. Gradients of the original functions f_i are calculated only once in each outer iteration. This is an important note since the calculation of gradients is typically the most time consuming part in structural optimization.

The GCMMA subproblem looks as follows, for $k \in \{1, 2, 3, \dots\}$ and $\nu \in \{0, 1, 2, \dots\}$:

$$\begin{aligned}
& \text{minimize} && \tilde{f}_0^{(k,\nu)}(\mathbf{x}) + a_0 z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) \\
& \text{subject to} && \tilde{f}_i^{(k,\nu)}(\mathbf{x}) - a_i z - y_i \leq 0, && i = 1, \dots, m \\
& && \alpha_j^{(k)} \leq x_j \leq \beta_j^{(k)}, && j = 1, \dots, n, \\
& && y_i \geq 0, && i = 1, \dots, m \\
& && z \geq 0.
\end{aligned} \tag{4.1}$$

In this subproblem (4.1), the approximating functions $\tilde{f}_i^{(k,\nu)}(\mathbf{x})$ are chosen as

$$\tilde{f}_i^{(k,\nu)}(\mathbf{x}) = \sum_{j=1}^n \left(\frac{p_{ij}^{(k,\nu)}}{u_j^{(k)} - x_j} + \frac{q_{ij}^{(k,\nu)}}{x_j - l_j^{(k)}} \right) + r_i^{(k,\nu)}, \quad i = 0, 1, \dots, m, \tag{4.2}$$

where

$$p_{ij}^{(k,\nu)} = (u_j^{(k)} - x_j^{(k)})^2 \left(1.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^+ + 0.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^- + \frac{\rho_i^{(k,\nu)}}{x_j^{\max} - x_j^{\min}} \right), \tag{4.3}$$

$$q_{ij}^{(k,\nu)} = (x_j^{(k)} - l_j^{(k)})^2 \left(0.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^+ + 1.001 \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right)^- + \frac{\rho_i^{(k,\nu)}}{x_j^{\max} - x_j^{\min}} \right), \tag{4.4}$$

$$r_i^{(k,\nu)} = f_i(\mathbf{x}^{(k)}) - \sum_{j=1}^n \left(\frac{p_{ij}^{(k,\nu)}}{u_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k,\nu)}}{x_j^{(k)} - l_j^{(k)}} \right). \tag{4.5}$$

Note: If $\rho_i^{(k,\nu+1)} > \rho_i^{(k,\nu)}$ then $\tilde{f}_i^{(k,\nu+1)}(\mathbf{x})$ is a more *conservative* approximation than $\tilde{f}_i^{(k,\nu)}(\mathbf{x})$.

Between each outer iteration, the bounds $\alpha_j^{(k)}$ and $\beta_j^{(k)}$ and the asymptotes $l_j^{(k)}$ and $u_j^{(k)}$ are updated as in the original MMA, the formulas (3.6)–(3.14) still hold.

The parameters $\rho_i^{(k,\nu)}$ in (4.3) and (4.4) are strictly positive and updated according to below. Within a given outer iteration k , the only differences between two inner iterations are the values of some of these parameters. In the beginning of each outer iteration, when $\nu = 0$, the following default values are used:

$$\rho_i^{(k,0)} = \frac{0.1}{n} \sum_{j=1}^n \left| \frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}) \right| (x_j^{\max} - x_j^{\min}), \quad \text{for } i = 0, 1, \dots, m. \tag{4.6}$$

If any of the right hand sides in (4.6) is $< 10^{-6}$ then the corresponding $\rho_i^{(k,0)}$ is set to 10^{-6} .

In each new inner iteration, the updating of $\rho_i^{(k,\nu)}$ is based on the solution to the most recent subproblem. Note that $\tilde{f}_i^{(k,\nu)}(\mathbf{x})$ may be written on the form:

$$\tilde{f}_i^{(k,\nu)}(\mathbf{x}) = h_i^{(k)}(\mathbf{x}) + \rho_i^{(k,\nu)} d^{(k)}(\mathbf{x}),$$

where $h_i^{(k)}(\mathbf{x})$ and $d^{(k)}(\mathbf{x})$ do not depend on $\rho_i^{(k,\nu)}$. Some calculations give that

$$d^{(k)}(\mathbf{x}) = \sum_{j=1}^n \frac{(u_j^{(k)} - l_j^{(k)})(x_j - x_j^{(k)})^2}{(u_j^{(k)} - x_j)(x_j - l_j^{(k)})(x_j^{\max} - x_j^{\min})}. \quad (4.7)$$

Now, let

$$\delta_i^{(k,\nu)} = \frac{f_i(\hat{\mathbf{x}}^{(k,\nu)}) - \tilde{f}_i^{(k,\nu)}(\hat{\mathbf{x}}^{(k,\nu)})}{d^{(k)}(\hat{\mathbf{x}}^{(k,\nu)})}. \quad (4.8)$$

Then $h_i^{(k)}(\hat{\mathbf{x}}^{(k,\nu)}) + (\rho_i^{(k,\nu)} + \delta_i^{(k,\nu)})d^{(k)}(\hat{\mathbf{x}}^{(k,\nu)}) = f_i(\hat{\mathbf{x}}^{(k,\nu)})$, which shows that $\rho_i^{(k,\nu)} + \delta_i^{(k,\nu)}$ might be a natural value of $\rho_i^{(k,\nu+1)}$. In order to get a globally convergent method, this natural value is modified as follows.

$$\begin{aligned} \rho_i^{(k,\nu+1)} &= \min\{1.1(\rho_i^{(k,\nu)} + \delta_i^{(k,\nu)}), 10\rho_i^{(k,\nu)}\} & \text{if } \delta_i^{(k,\nu)} > 0, \\ \rho_i^{(k,\nu+1)} &= \rho_i^{(k,\nu)} & \text{if } \delta_i^{(k,\nu)} \leq 0. \end{aligned} \quad (4.9)$$

It follows from the formulas (4.2)–(4.5) that the functions $\tilde{f}_i^{(k,\nu)}$ are always first order approximations of the original functions f_i at the current iteration point, i.e.

$$\tilde{f}_i^{(k,\nu)}(\mathbf{x}^{(k)}) = f_i(\mathbf{x}^{(k)}) \quad \text{and} \quad \frac{\partial \tilde{f}_i^{(k,\nu)}}{\partial x_j}(\mathbf{x}^{(k)}) = \frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}). \quad (4.10)$$

Since the parameters $\rho_i^{(k,\nu)}$ are always strictly positive, the functions $\tilde{f}_i^{(k,\nu)}$ are strictly convex. This implies that there is always a unique optimal solution of the GCMMA subproblem.

There are at least two approaches for solving the subproblems in MMA and in GCMMA, the “dual approach” and the “primal-dual interior-point approach”.

In the primal-dual interior-point approach, a sequence of relaxed KKT conditions are solved by Newton’s method. We have implemented this approach in Matlab, since all the required calculations are most naturally carried out on a matrix and vector level. This approach for solving the subproblem is described next.

5. A primal-dual method for solving the subproblems in MMA and GCMMA

To simplify the notations, the iteration indices k and ν are now removed in the subproblem. Further, we let $b_i = -r_i^{(k,\nu)}$, and drop the constant $r_0^{(k,\nu)}$ from the objective function. Then the MMA/GCMMA subproblem becomes

$$\begin{aligned} & \text{minimize} && g_0(x) + a_0z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) \\ & \text{subject to} && g_i(x) - a_i z - y_i \leq b_i, && i = 1, \dots, m \\ & && \alpha_j \leq x_j \leq \beta_j, && j = 1, \dots, n \\ & && z \geq 0, \quad y_i \geq 0, && i = 1, \dots, m \end{aligned} \quad (5.1)$$

where

$$g_i(x) = \sum_{j=1}^n \left(\frac{p_{ij}}{u_j - x_j} + \frac{q_{ij}}{x_j - l_j} \right), \quad i = 0, 1, \dots, m, \quad (5.2)$$

and where $l_j < \alpha_j < \beta_j < u_j$ for all j .

5.1. Optimality conditions for the MMA/GCMMA subproblem

Since the subproblem (5.1) is a regular convex problem, the KKT optimality conditions are both necessary and sufficient for an optimal solution to (5.1). In order to state these conditions, we first form the Lagrange function corresponding to (5.1).

$$\begin{aligned} L = & g_0(x) + a_0z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) + \sum_{i=1}^m \lambda_i (g_i(x) - a_i z - y_i - b_i) + \\ & + \sum_{j=1}^n (\xi_j (\alpha_j - x_j) + \eta_j (x_j - \beta_j)) - \sum_{i=1}^m \mu_i y_i - \zeta z, \end{aligned} \quad (5.3)$$

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 (5.1).

Let

$$\psi(x, \lambda) = g_0(x) + \sum_{i=1}^m \lambda_i g_i(x) = \sum_{j=1}^n \left(\frac{p_j(\lambda)}{u_j - x_j} + \frac{q_j(\lambda)}{x_j - l_j} \right), \quad (5.4)$$

where

$$p_j(\lambda) = p_{0j} + \sum_i^m \lambda_i p_{ij} \quad \text{and} \quad q_j(\lambda) = q_{0j} + \sum_i^m \lambda_i q_{ij}. \quad (5.5)$$

Then the Lagrange function can be written

$$\begin{aligned} L = & \psi(x, \lambda) + (a_0 - \zeta)z + \sum_{j=1}^n (\xi_j (\alpha_j - x_j) + \eta_j (x_j - \beta_j)) + \\ & + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2 - \lambda_i a_i z - \lambda_i y_i - \lambda_i b_i - \mu_i y_i), \end{aligned} \quad (5.6)$$

and then the KKT optimality conditions for the subproblem (5.1) become as follows.

$$\frac{\partial \psi}{\partial x_j} - \xi_j + \eta_j = 0, \quad j = 1, \dots, n \quad (\partial L / \partial x_j = 0) \quad (5.7a)$$

$$c_i + d_i y_i - \lambda_i - \mu_i = 0, \quad i = 1, \dots, m \quad (\partial L / \partial y_i = 0) \quad (5.7b)$$

$$a_0 - \zeta - \lambda^T a = 0, \quad (\partial L / \partial z = 0) \quad (5.7c)$$

$$g_i(x) - a_i z - y_i - b_i \leq 0, \quad i = 1, \dots, m \quad (\text{primal feasibility}) \quad (5.7d)$$

$$\lambda_i (g_i(x) - a_i z - y_i - b_i) = 0, \quad i = 1, \dots, m \quad (\text{compl slackness}) \quad (5.7e)$$

$$\xi_j (\alpha_j - x_j) = 0, \quad j = 1, \dots, n \quad (\text{compl slackness}) \quad (5.7f)$$

$$\eta_j (x_j - \beta_j) = 0, \quad j = 1, \dots, n \quad (\text{compl slackness}) \quad (5.7g)$$

$$-\mu_i y_i = 0, \quad i = 1, \dots, m \quad (\text{compl slackness}) \quad (5.7h)$$

$$-\zeta z = 0, \quad (\text{compl slackness}) \quad (5.7i)$$

$$\alpha_j \leq x_j \leq \beta_j, \quad j = 1, \dots, n \quad (\text{primal feasibility}) \quad (5.7j)$$

$$-z \leq 0 \quad \text{and} \quad -y_i \leq 0, \quad i = 1, \dots, m \quad (\text{primal feasibility}) \quad (5.7k)$$

$$\xi_j \geq 0 \quad \text{and} \quad \eta_j \geq 0, \quad j = 1, \dots, n \quad (\text{dual feasibility}) \quad (5.7l)$$

$$\zeta \geq 0 \quad \text{and} \quad \mu_i \geq 0, \quad i = 1, \dots, m \quad (\text{dual feasibility}) \quad (5.7m)$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m \quad (\text{dual feasibility}) \quad (5.7n)$$

where

$$\frac{\partial \psi}{\partial x_j} = \frac{p_j(\lambda)}{(u_j - x_j)^2} - \frac{q_j(\lambda)}{(x_j - l_j)^2} \quad \text{and} \quad \lambda^T a = \sum_{i=1}^m \lambda_i a_i. \quad (5.8)$$

5.2. The relaxed optimality conditions for the MMA/GCMMA subproblem

When a primal-dual interior-point method is used for solving the subproblem (5.1), the zeros in the right hand sides of the complementary slackness conditions (5.7e)–(5.7i) are replaced by the negative of a “small” parameter $\varepsilon > 0$. Further, slack variables s_i are introduced for the constraints (5.7d). The relaxed optimality conditions then become as follows.

$$\frac{\partial \psi}{\partial x_j} - \xi_j + \eta_j = 0, \quad j = 1, \dots, n \quad (5.9a)$$

$$c_i + d_i y_i - \lambda_i - \mu_i = 0, \quad i = 1, \dots, m \quad (5.9b)$$

$$a_0 - \zeta - \lambda^T a = 0, \quad (5.9c)$$

$$g_i(x) - a_i z - y_i + s_i - b_i = 0, \quad i = 1, \dots, m \quad (5.9d)$$

$$\xi_j(x_j - \alpha_j) - \varepsilon = 0, \quad j = 1, \dots, n \quad (5.9e)$$

$$\eta_j(\beta_j - x_j) - \varepsilon = 0, \quad j = 1, \dots, n \quad (5.9f)$$

$$\mu_i y_i - \varepsilon = 0, \quad i = 1, \dots, m \quad (5.9g)$$

$$\zeta z - \varepsilon = 0, \quad (5.9h)$$

$$\lambda_i s_i - \varepsilon = 0, \quad i = 1, \dots, m \quad (5.9i)$$

$$x_j - \alpha_j > 0 \text{ and } \xi_j > 0, \quad j = 1, \dots, n \quad (5.9j)$$

$$\beta_j - x_j > 0 \text{ and } \eta_j > 0, \quad j = 1, \dots, n \quad (5.9k)$$

$$y_i > 0 \text{ and } \mu_i > 0, \quad i = 1, \dots, m \quad (5.9l)$$

$$z > 0 \text{ and } \zeta > 0, \quad (5.9m)$$

$$s_i > 0 \text{ and } \lambda_i > 0, \quad i = 1, \dots, m \quad (5.9n)$$

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 (5.9a)–(5.9n) are mathematically (but not numerically) equivalent to the KKT conditions of the following strictly convex problem in the variables x, y, z and s .

$$\begin{aligned} \text{minimize} \quad & g_0(x) + a_0 z + \sum_i (c_i y_i + \frac{1}{2} d_i y_i^2) - \\ & - \varepsilon \sum_j \log(x_j - \alpha_j) - \varepsilon \sum_j \log(\beta_j - x_j) - \\ & - \varepsilon \sum_i \log y_i - \varepsilon \sum_i \log s_i - \varepsilon \log z \end{aligned} \quad (5.10)$$

$$\text{subject to} \quad g_i(x) - a_i z - y_i + s_i \leq b_i, \quad i = 1, \dots, m$$

$$(\alpha_j < x_j < \beta_j, y_i > 0, z > 0, s_i > 0)$$

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

5.3. A Newton direction for the relaxed optimality conditions

Assume that a point $w = (x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)$ which satisfies (5.9j)–(5.9n) is given, and assume that, starting from this point, Newton’s method should be applied to (5.9a)–(5.9i). Then the following system of linear equations in $\Delta w = (\Delta x, \Delta y, \Delta z, \Delta \lambda, \Delta \xi, \Delta \eta, \Delta \mu, \Delta \zeta, \Delta s)$ should be generated and solved.

$$\begin{pmatrix} \Psi & & G^T & \langle -e \rangle & \langle e \rangle & & & & & & & \\ & \langle d \rangle & & \langle -e \rangle & & & \langle -e \rangle & & & & & \\ & & -a^T & & & & & -1 & & & & \\ G & \langle -e \rangle & -a & & & & & & \langle e \rangle & & & \\ \langle \xi \rangle & & & \langle x-\alpha \rangle & & & & & & & & \\ \langle -\eta \rangle & & & & \langle \beta-x \rangle & & & & & & & \\ & \langle \mu \rangle & & & & \langle y \rangle & & & & & & \\ & & \zeta & & & & z & & & & & \\ & & & \langle s \rangle & & & & & & \langle \lambda \rangle & & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \lambda \\ \Delta \xi \\ \Delta \eta \\ \Delta \mu \\ \Delta \zeta \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\delta_x \\ -\delta_y \\ -\delta_z \\ -\delta_\lambda \\ -\delta_\xi \\ -\delta_\eta \\ -\delta_\mu \\ -\delta_\zeta \\ -\delta_s \end{pmatrix}$$

where $\delta_x, \dots, \delta_s$ are defined by the left hand sides in (5.9a)–(5.9i),

$$\Psi \text{ is an } n \times n \text{ diagonal matrix with } (\Psi)_{jj} = \frac{\partial^2 \psi}{\partial x_j^2} = \frac{2p_j(\lambda)}{(u_j - x_j)^3} + \frac{2q_j(\lambda)}{(x_j - l_j)^3}, \quad (5.11)$$

$$G \text{ is an } m \times n \text{ matrix with } (G)_{ij} = \frac{\partial g_i}{\partial x_j} = \frac{p_{ij}}{(u_j - x_j)^2} - \frac{q_{ij}}{(x_j - l_j)^2}, \quad (5.12)$$

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

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

e is a vector $(1, \dots, 1)^T$, with dimension apparent from the context, so that

$\langle e \rangle$ is a unity matrix, with dimension apparent from the context.

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

$$\Delta \xi = -\langle x-\alpha \rangle^{-1} \langle \xi \rangle \Delta x - \xi + \varepsilon \langle x-\alpha \rangle^{-1} e, \quad (5.13a)$$

$$\Delta \eta = \langle \beta-x \rangle^{-1} \langle \eta \rangle \Delta x - \eta + \varepsilon \langle \beta-x \rangle^{-1} e, \quad (5.13b)$$

$$\Delta \mu = -\langle y \rangle^{-1} \langle \mu \rangle \Delta y - \mu + \varepsilon \langle y \rangle^{-1} e, \quad (5.13c)$$

$$\Delta \zeta = -(\zeta/z) \Delta z - \zeta + \varepsilon/z, \quad (5.13d)$$

$$\Delta s = -\langle \lambda \rangle^{-1} \langle s \rangle \Delta \lambda - s + \varepsilon \langle \lambda \rangle^{-1} e. \quad (5.13e)$$

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

$$\begin{pmatrix} D_x & & & G^T \\ & D_y & & \langle -e \rangle \\ & & \zeta/z & -a^T \\ G & \langle -e \rangle & -a & -D_\lambda \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\tilde{\delta}_x \\ -\tilde{\delta}_y \\ -\tilde{\delta}_z \\ -\tilde{\delta}_\lambda \end{pmatrix} \quad (5.14)$$

where

$$D_x = \Psi + \langle x - \alpha \rangle^{-1} \langle \xi \rangle + \langle \beta - x \rangle^{-1} \langle \eta \rangle, \quad (\text{a diagonal matrix}) \quad (5.15a)$$

$$D_y = \langle d \rangle + \langle y \rangle^{-1} \langle \mu \rangle, \quad (\text{a diagonal matrix}) \quad (5.15b)$$

$$D_\lambda = \langle \lambda \rangle^{-1} \langle s \rangle, \quad (\text{a diagonal matrix}) \quad (5.15c)$$

$$\tilde{\delta}_x = \frac{\partial \psi}{\partial x} - \varepsilon \langle x - \alpha \rangle^{-1} e + \varepsilon \langle \beta - x \rangle^{-1} e, \quad (5.15d)$$

$$\tilde{\delta}_y = c + \langle d \rangle y - \lambda - \varepsilon \langle y \rangle^{-1} e, \quad (5.15e)$$

$$\tilde{\delta}_z = a_0 - \lambda^T a - \varepsilon/z, \quad (\text{a scalar}) \quad (5.15f)$$

$$\tilde{\delta}_\lambda = g(x) - az - y - b + \varepsilon \langle \lambda \rangle^{-1} e. \quad (5.15g)$$

In (5.15d), $\frac{\partial \psi}{\partial x}$ is a vector with the n components $\frac{\partial \psi}{\partial x_j} = \frac{p_j(\lambda)}{(u_j - x_j)^2} - \frac{q_j(\lambda)}{(x_j - l_j)^2}$.

Next, Δy can be eliminated from the system (5.14) through

$$\Delta y = D_y^{-1} \Delta \lambda - D_y^{-1} \tilde{\delta}_y. \quad (5.16)$$

Then the following system in $\Delta x, \Delta z$ and $\Delta \lambda$ is obtained.

$$\begin{pmatrix} D_x & & G^T \\ & \zeta/z & -a^T \\ G & -a & -D_{\lambda y} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\tilde{\delta}_x \\ -\tilde{\delta}_z \\ -\tilde{\delta}_{\lambda y} \end{pmatrix} \quad (5.17)$$

where

$$D_{\lambda y} = D_\lambda + D_y^{-1}, \quad (\text{a diagonal matrix}) \quad (5.18a)$$

$$\tilde{\delta}_{\lambda y} = \tilde{\delta}_\lambda + D_y^{-1} \tilde{\delta}_y. \quad (5.18b)$$

In this system (5.17), either Δx or $\Delta \lambda$ can be eliminated. If Δx is eliminated through

$$\Delta x = -D_x^{-1}G^T\Delta\lambda - D_x^{-1}\tilde{\delta}_x, \quad (5.19)$$

the following system in $\Delta\lambda$ and Δz is obtained.

$$\begin{pmatrix} D_{\lambda y} + GD_x^{-1}G^T & a \\ a^T & -\zeta/z \end{pmatrix} \begin{pmatrix} \Delta\lambda \\ \Delta z \end{pmatrix} = \begin{pmatrix} \tilde{\delta}_{\lambda y} - GD_x^{-1}\tilde{\delta}_x \\ \tilde{\delta}_z \end{pmatrix} \quad (5.20)$$

If, instead, $\Delta\lambda$ is eliminated from the system (5.17) through

$$\Delta\lambda = D_{\lambda y}^{-1}G\Delta x - D_{\lambda y}^{-1}a\Delta z + D_{\lambda y}^{-1}\tilde{\delta}_{\lambda y}, \quad (5.21)$$

the following system in Δx and Δz is obtained.

$$\begin{pmatrix} D_x + G^TD_{\lambda y}^{-1}G & -G^TD_{\lambda y}^{-1}a \\ -a^TD_{\lambda y}^{-1}G & \zeta/z + a^TD_{\lambda y}^{-1}a \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z \end{pmatrix} = \begin{pmatrix} -\tilde{\delta}_x - G^TD_{\lambda y}^{-1}\tilde{\delta}_{\lambda y} \\ -\tilde{\delta}_z + a^TD_{\lambda y}^{-1}\tilde{\delta}_{\lambda y} \end{pmatrix} \quad (5.22)$$

Note that the size of the system (5.20) is $(m+1) \times (m+1)$, while the size of the system (5.22) is $(n+1) \times (n+1)$. Therefore, typically, the system (5.20) should be preferred if $n > m$, while the system (5.22) should be preferred if $m > n$.

In both the systems (5.20) and (5.22), it is of course possible to eliminate also the variable Δz (and reduce the systems to $m \times m$ and $n \times n$, respectively), but for numerical reasons it is not wise to do that. As an example, if a is dense (many non-zeros) then the system obtained after eliminating Δz will be dense, even if the original system (5.20) or (5.22) is sparse.

5.4. Line search in the Newton direction

When the Newton direction $\Delta w = (\Delta x, \Delta y, \Delta z, \Delta\lambda, \Delta\xi, \Delta\eta, \Delta\mu, \Delta\zeta, \Delta s)$ has been calculated in the given point $w = (x, y, z, \lambda, \xi, \eta, \mu, \zeta, s)$, a step should be taken in that direction.

A pure Newton step would be to take a step equal to the calculated direction, but such a step might lead to a point where some of the inequalities (5.9j)–(5.9n) are violated.

Therefore, we first let t be the largest number such that

$$t \leq 1, \quad x_j + t\Delta x_j - \alpha_j \geq 0.01(x_j - \alpha_j) \text{ for all } j, \quad \beta_j - (x_j + t\Delta x_j) \geq 0.01(\beta_j - x_j) \text{ for all } j,$$

and $(y, z, \lambda, \xi, \eta, \mu, \zeta, s) + t \cdot (\Delta y, \Delta z, \Delta\lambda, \Delta\xi, \Delta\eta, \Delta\mu, \Delta\zeta, \Delta s) \geq 0.01 \cdot (y, z, \lambda, \xi, \eta, \mu, \zeta, s)$.

Further, the new point should also be in some sense better than the previous.

Therefore, we let τ be the largest of $t, t/2, t/4, t/8, \dots$ such that

$$\|\delta(w + \tau \cdot \Delta w)\| < \|\delta(w)\|,$$

where $\delta(w)$ is the residual vector defined by the left hand sides in the relaxed KKT conditions (5.9a)–(5.9i), and $\|\cdot\|$ is the ordinary Euclidian norm. This is always possible to obtain since the Newton direction is a descent direction for $\|\delta(w)\|$.

5.5. The complete primal-dual algorithm

First of all, $\varepsilon^{(1)}$ and a starting point $w^{(1)} = (x^{(1)}, y^{(1)}, z^{(1)}, \lambda^{(1)}, \xi^{(1)}, \eta^{(1)}, \mu^{(1)}, \zeta^{(1)}, s^{(1)})$ which satisfies (5.9j)–(5.9n) are chosen. The following is a simple but reasonable choice.

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

$$\xi_j^{(1)} = \max\{1, 1/(x_j^{(1)} - \alpha_j)\}, \quad \eta_j^{(1)} = \max\{1, 1/(\beta_j - x_j^{(1)})\}, \quad \mu_i^{(1)} = \max\{1, c_i/2\}.$$

A typical iteration, leading from the ℓ :th iteration point $w^{(\ell)}$ to the $(\ell+1)$:th iteration point $w^{(\ell+1)}$, consists of the following steps.

Step 1: For given $\varepsilon^{(\ell)}$ and $w^{(\ell)}$ which satisfy (5.9j)–(5.9n), calculate $\Delta w^{(\ell)}$ as described in section 5.3 above.

Step 2: Calculate a steplength $\tau^{(\ell)}$ as described in section 5.4 above.

Step 3: Let $w^{(\ell+1)} = w^{(\ell)} + \tau^{(\ell)} \cdot \Delta w^{(\ell)}$.

Step 4: If $\|\delta(w^{(\ell+1)})\|_\infty < 0.9\varepsilon^{(\ell)}$, let $\varepsilon^{(\ell+1)} = 0.1\varepsilon^{(\ell)}$.
Otherwise, let $\varepsilon^{(\ell+1)} = \varepsilon^{(\ell)}$. Increase ℓ by 1 and go to Step 1.

The algorithm is terminated when $\varepsilon^{(\ell)}$ has become sufficiently small, say $\varepsilon^{(\ell)} \leq 10^{-7}$, and $\|\delta(w^{(\ell+1)})\|_\infty < 0.9\varepsilon^{(\ell)}$.

6. References

- [1] K. Svanberg, The method of moving asymptotes – a new method for structural optimization, *International Journal for Numerical Methods in Engineering*, 1987, 24, 359-373.
- [2] K. Svanberg, A class of globally convergent optimization methods based on conservative convex separable approximations, *SIAM Journal of Optimization*, 2002, 12, 555-573.