Computing 54, 83–98 (1995)

Computing

Interpolation by Conic Model for Unconstrained Optimization*

S. Sheng, Nanjing

Received September 4, 1993

Abstract — Zusammenfassung

Interpolation by Conic Model for Unconstrained Optimization. This paper describes a method for unconstrained optimization that associates quasi-Newton methods with conic functions. The derivation is based upon the construction of a conic function so that a local nonquadratic model can interpolate two function and one gradient values of the objective function at the last two iterates as a natural extension of existing quasi-Newton methods. The new method is shown to have Q-superlinear rate of convergence under standard assumptions on the objective function, and to decrease the number of line searches for good choice of parameters. Numerical experiments verify that the new method is very successful.

AMS Subject Classifications: 65K, 49M

Key words: Conic model, quasi-Newton method, line search, superlinear convergence, unconstrained optimization.

Konische Interpolation für die unrestringierte Optimierung. Die Arbeit beschreibt eine Methode zur unrestringierten Optimierung, die konische Funktionen im quasi-Newton-Verfahren verwendet. Es wird dabei eine konische Funktion so konstruiert, daß das lokale Modell zwei Funktionswerte und einen Gradientenwert der Zielfunktion an den letzten zwei Iterierten interpoliert, was eine natürliche Erweiterung bestehender Quasi-Newton-Verfahren darstellt. Unter Standardannahmen über die Zielfunktion wird eine Q-superlineare Konvergenzgeschwindigkeit gezeigt und eine Verminderung der Anzahl der "line searches" bei guter Parameterwahl. Numerische Experimente bestätigen die Effizienz des Verfahrens.

1. Introduction

Consider the minimization problem

$$\text{minimize} \quad f(x), \quad f:\mathbb{R}^n \to \mathbb{R}, \quad f \in C^2; \tag{1.1}$$

many methods solving (1,1) are iterative, i.e., the sequence $\{x_k\}$ generated by an iterative algorithm starting with an initial point x_0 converges to a local minimizer x^* of **f(x)**. One class of typical iterations is the quasi-Newton methods which simultaneously generate a sequence of points $x_k \in \mathbb{R}^n$ and matrices $B_k \in \mathbb{R}^{n\times n}$ such that

$$x_{k+1} = x_k + \lambda_k s_k, \tag{1.2}$$

* The project was supported by the National Natural Science Foundation of China.

where

$$s_k = -B_k^{-1} f'(x_k).$$

For stability reasons, it is usually required that the matrix B_k be symmetric positive definite, so that the direction s_k is a descent direction of **f(x)** and $f(x_{k+1}) < f(x_k)$ for some $\lambda_k \in (0,1]$. Moreover, the B_k is updated according to a formula of the form $B_{k+1} = B_k + D_k$, where D_k is a symmetric matrix of rank 2 or rank 1 so that the quasi-Newton equation

$$B_{k+1}(x_{k+1} - x_k) = f'(x_{k+1}) - f'(x_k)$$

holds and B_{k+1} stays symmetric positive definite. Quasi-Newton methods are efficient in practical use, but how to rapidly find the line search step factorλ_k is still an open problem.

Davidon [3] proposed conic approximations and collinear scaling which are

$$c(x+s) = f(x) + \frac{f'(x)^T s}{1 - a^T s} + \frac{1}{2}\frac{s^T B s}{(1 - a^T s)^2} \tag{1.3}$$

and

$$S(w) = x + \frac{Jw}{1 + h^T w}, \quad 1 + h^T w > 0. \tag{1.4}$$

Davidon made a thorough investigation of the interpolating and geometrical properties of the conic function (1.3) and collinear scaling (1.4). Gourgeon and Nocedal [7] described a conic algorithm based on the conic model (1.3) that minimizes a conic function **f(x)** in **n** steps. Their algorithm can be considered a generalization of conjugate gradient method and has similar orthogonality properties. Sorensen [13] first derived an algorithm using (1.3) and (1.4), explicitly indicating the relationship to BFGS quasi-Newton methods. He referred to his algorithm as collinear scaling algorithm and proved that this algorithm has locally Q-superlinear convergence. Ariyawansa [1], Xu and Sheng [14] extended Sorensen's results to Broyden family collinear scaling algorithms with different technique, respectively. Sheng (1993) indicated that the single parameter Broyden collinear scaling class belongs to the Spedicato three-parameter family if exact collinear line search is devised. Unfortunately, according to our numerical experiments the collinear scaling methods are not better than the classical BFGS quasi-Newton method.

In this paper, we shall describe conic model algorithms based on (1.3). Our idea is that, if we can find a symmetric positive definite matrix B and choose an appropriate vector $a \in \mathbb{R}^n$ at the current iteration **x** such that the conic function **c(x + s)** satisfies some interpolating conditions and $1 - a^T s \geq 1$, then $-B^{-1} f'(x)$ is a descent direction at **x** and $(1 - a^T s)^{-1}$ is a predictive value of the line search step factor λ in (1.2). Thus it is possible to decrease the number of line searches and increase the efficiency of the conic model algorithms.

The outline of this paper is as follows. In Section 2 we state the conic model interpolation theory which is an extension of the classical quasi-Newton method, and present three algorithms in Section 3. The local and Q-superlinear convergence for these algorithms are proved in Section 4. The results of numerical experiments are presented in Section 5. Finally, in Section 6 we give some concluding remarks.

In all cases, $\|\cdot\|$ denotes the l_2-norm on $\mathbb{R}^n$ or the induced operator norm on $\mathbb{R}^{n\times n}$. Frobenius norms and weighted norms are also used and denoted by $\|\cdot\|_F$ and $\|\cdot\|_A$, respectively. Throughout this paper, f_k, g_k denote $f(x_k)$ and $f'(x_k)$. Sometimes, x_c and x_- denote the current point x_k and the previous point x_{k-1}, respectively.

2. Interpolation by Conic Model

The conic model

$$c(x_c+s)=f_c+\frac{g_c^Ts}{1-a^Ts}+\frac{1}{2}\frac{s^TB_cs}{(1-a^Ts)^2}, \tag{2.1}$$

where a and B_c are an undetermined vector and matrix respectively, is a local approximation of the objective function $f(x_c+s)$, and has the properties

$$c(x_c)=f_c,$$
$$c'(x_c)=g_c.$$

Furthermore, if B_c is positive definite, (2.1) has a unique minimizer (see [10])

$$s_c=-\frac{B_c^{-1}g_c}{1-a^TB_c^{-1}g_c}. \tag{2.2}$$

In order to determine the vector a and the matrix B_c, we introduce the following conditions into (2.1).

(1) Interpolating condition

$$c(x_-)=f_-,$$

or equivalently,

$$f_-=f_c-\frac{g_c^Ts_-}{1+a^Ts_-}+\frac{1}{2}\frac{s_-^TB_cs_-}{(1+a^Ts_-)^2}, \tag{2.3}$$

where $s_-=x_c-x_-$.

(2) Quasi-Newton condition

$$B_cs_-=\alpha y_-, \tag{2.4}$$

where $y_-=g_c-g_-$, and $\alpha>0$ is an undetermined parameter.

(3) Parameter vector condition

$$a = tg_c \tag{2.5a}$$

or

$$a = tg_-, \tag{2.5b}$$

where **t** is an undetermined real parameter.

Before choosing the parameters α and **t**, we note that the unique minimizer (2.2) should be a descent direction at x_c, which can be guaranteed by positive definite B_c and $1 - a^T B_c^{-1} g_c > 0$. Further, if B_- is symmetric positive definite and B_c is updated by B_- using (2.4), then B_c is symmetric positive defnite if and only if $\alpha s_-^T y_- = \alpha(g_c^T s_- - g_-^T s_-) > 0$ (see Lemma 9.2.1 in [5]). On the other hand, in order to decrease the number of line search, we hope that $1 - a^T B_c^{-1} g_c \geq 1$. Therefore, from (2.5) we obtain

$$t g_c^T B_c^{-1} g_c \leq 0 \tag{2.6a}$$

or

$$t g_-^T B_c^{-1} g_c \leq 0 \tag{2.6b}$$

Lemma 2.1. *Let the symmetric positive definite matrix B_c satisfy* (2.4) *and* $\alpha > 0$. *Then $g_-^T B_c^{-1} g_c > 0$ is guaranteed by any line search method.*

Proof: From (2.4) we have

$$g_c^T s_- = \alpha g_c^T B_c^{-1}(g_c - g_-).$$

Hence

$$g_-^T B_c^{-1} g_c = g_c^T B_c^{-1} g_c - \frac{1}{\alpha} g_c^T s_-.$$

Now, if $g_c^T s_- \leq 0$, then clearly $g_-^T B_c^{-1} g_c > 0$. If $g_c^T s_- > 0$, then since $g_c^T s_- = 0$ in the exact line search, we can let $(1/\alpha)\, g_c^T s_-$ be very small using any inexact line search method such that $g_-^T B^{-1} g_c > 0$.

Remark 2.1. In the numerical experiments, we restrict $-t g_-^T B_c^{-1} g_c \in [0, L]$. Therefore, we do not care about the sign of $g_-^T B_c^{-1} g_c$ in the line search.

According to Lemma 2.1 and (2.6), the parameter **t** should be nonpositive.

From (2.3), (2.4), (2.5a) or (2.5b) we obtain, respectively,

$$t = \frac{-2(f_- - f_c) - g_c^T s_- \pm \sqrt{(g_c^T s_-)^2 + 2\alpha(f_- - f_c) s_-^T y_-}}{2(f_- - f_c) g_c^T s_-} \tag{2.7a}$$

and

$$t = \frac{-2(f_- - f_c) - g_c^T s_- \pm \sqrt{(g_c^T s_-)^2 + 2\alpha(f_- - f_c) s_-^T y_-}}{2(f_- - f_c) g_-^T s_-}. \tag{2.7b}$$

We set

$$\alpha s_-^T y_- - 2 g_c^T s_- - 2(f_- - f_c) = -\zeta (g_c^T s_-)^2.$$

Then we get

$$\text{(a)}\quad \zeta=0,\quad \alpha=2(f_- - f_c + g_c^T s_-)/s_-^T y_-, \tag{2.8a}$$

$$\text{(b)}\quad \zeta=1,\quad \alpha=[2(f_- - f_c + g_c^T s_-) + (g_c^T s_-)^2]/s_-^T y_-. \tag{2.8b}$$

Now, we can determine the parameter **t** as follows:

(I). $\zeta=0$.

If $2(f_- - f_c)+g_c^T s_- \geq 0$ then we take the positive root in (2.7), otherwise the negative root. In this case, $\mathbf{t}=0$.

(II). $\zeta=1$.

(1) $a=tg_c$. We use

$$t=\begin{cases} \dfrac{-2(f_- - f_c) - g_c^T s_- + \sqrt{(g_c^T s_-)^2 + 2\alpha(f_- - f_c)s_-^T y_-}}{2(f_- - f_c)g_c^T s_-} & \text{if } g_c^T s_- \leq 0, \\[2ex] \dfrac{-2(f_- - f_c) - g_c^T s_- - \sqrt{(g_c^T s_-)^2 + 2\alpha(f_- - f_c)s_-^T y_-}}{2(f_- - f_c)g_c^T s_-} & \text{if } g_c^T s_- > 0. \end{cases}$$

With (2.8b) we get

$$t=\begin{cases} g_c^T s_- \Big/ \left[2(f_- - f_c) + g_c^T s_- + \sqrt{(2(f_- - f_c) + g_c^T s_-)^2 + 2(f_- - f_c)(g_c^T s_-)^2}\right] \\ \quad \text{if } g_c^T s_- \leq 0, \\ -g_c^T s_- \Big/ \left[-2(f_- - f_c) - g_c^T s_- \right. \\ \quad \left. + \sqrt{(2(f_- - f_c) + g_c^T s_-)^2 + 2(f_- - f_c)(g_c^T s_-)^2}\right] \quad \text{if } g_c^T s_- > 0. \end{cases} \tag{2.9}$$

(2) $a=tg_-$. We always put

$$t=\left[-2(f_- - f_c) - g_c^T s_- + \sqrt{(g_c^T s_-)^2 + 2\alpha(f_- - f_c)s_-^T y_-}\right] / 2(f_- - f_c)g_-^T s_-.$$

Using (2.8b) we have

$$t=\frac{(g_c^T s_-)^2}{g_-^T s_- \left[2(f_- - f_c) + g_c^T s_- + \sqrt{(2f_- - 2f_c + g_c^T s_-)^2 + 2(f_- - f_c)(g_c^T s_-)^2}\right]}. \tag{2.10}$$

Remark 2.2. From the above discussion, we only ensure $\mathbf{t} \leq 0$. The value of α will be controlled by the algorithm such that $\alpha > 0$.

3. Hybrid Algorithms

In Section 2 we discussed the choices of the parameter α and **t**. As long as α is chosen, **t** is uniquely determined. Then B_c^{-1} is updated by B_-^{-1} with the extension of inverse BFGS method by means of (2.4). So that the descent s_c at x_c can be given by (2.2). Unfortunately, $\alpha \le 0$, which can destroy the algorithms, may appear in the beginning of the iterations. On the other hand, very small or large values of α can disturb the stability of the method. Thus we must regulate the region of α. Let

$$0 < \rho_1 < 1 < \rho_2 .$$

If $\alpha \notin [\rho_1, \rho_2]$ then we force $\alpha = 1$ and $\mathbf{t} = 0$. In this case, the current iteration is the classical BFGS method.

Because the size of **t** only affects the predictive step factor $(1 - a^T B_c^{-1} g_c)^{-1}$, we don't hope that $|\mathbf{t}|$ is too large. Thus if $|\mathbf{t}| > \mathbf{T}$ then set $\mathbf{t} = -\mathbf{T}$. Here $\mathbf{T} > 0$ is fixed. This restriction will simplify the proof of local convergence in Section 4.

Now, we can outline the following three hybrid algorithms by means of the alternate choices of α and **t**.

Algorithm 3.1

(1) Given the initial **x**, the symmetric positive definite **H**, $\varepsilon_1, \varepsilon_2, \varepsilon_3$, **T** and $0 < \rho_1 < 1 < \rho_2 . k = 0$.

(2) Compute

$$f = f(x), \quad g = f'(x),$$
$$s = -Hg/(1 - \eta),$$

where $\eta = 0$ or $\eta = -\|g\|$, and H denotes B^{-1}.

(3) Line search. Find a $\lambda > 0$ such that

$$\bar{x} = x + \lambda s \quad \text{and} \quad \bar{f} = f(\bar{x})$$

satisfy

$$f - \bar{f} \ge -\varepsilon_1 \lambda s^T g .$$

(4) Test convergence. If $\|f'(\bar{x})\| \le \varepsilon_2$ or $\left|\dfrac{f - \bar{f}}{f}\right| \le \varepsilon_3$ then out put $\bar{x}$, $\bar{f}$. Stop.

(5) Choose α and **t**.
Compute α using (2.8b)
If $\alpha \notin [\rho_1, \rho_2]$ then set $\alpha = 1$, $\mathbf{t} = 0$ and go to (6)
Compute **t** using (2.9)
If $\mathbf{t} < -\mathbf{T}$ then $\mathbf{t} = -\mathbf{T}$.

(6) Correct **H**.
Compute $\bar{g} = f'(\bar{x})$ and

$$H = H - \frac{sy^T H + Hys^T}{s^T y} + \left(\frac{y^T Hy}{(s^T y)^2} + \frac{1}{\alpha s^T y} \right) ss^T, \tag{3.1}$$

where $s = \bar{x} - x$ and $y = \bar{g} - g$.

(7) Produce the new descent direction

$$\eta = t\bar{g}^T H\bar{g}.$$
$$s = -H\bar{g}/(1-\eta).$$
$$x = \bar{x}, \quad g = \bar{g}, \quad f = \bar{f}, \quad k = k+1.$$

Go to (3).

Algorithm 3.2

"Compute **t** using (2.9)" at step (5) and "$\eta = t\bar{g}^T H\bar{g}$" at step (7) in the Algorithm 3.1 are respectively replaced by the following

Compute **t** using (2.10)

and

$$\eta = tg^T H\bar{g}.$$

Algorithm 3.3

Step (5) in the Algorithm 3.1 is replaced by

Compute α using (2.8a).

If $\alpha \notin [\rho_1, \rho_2]$ then set $\alpha = 1$.

$\mathbf{t} = 0$.

Remark 3.1. If $\rho_1 = \rho_2 = 1$ then Algorithms 3.1, 3.2 and 3.3 are the classical BFGS algorithms.

4. Local Convergence Analysis

In this section we analyse the local convergence of Algorithms 3.1−3.3 with the direct prediction method, i.e., $\lambda \equiv 1$ in the line search. Our approach is only an application of the theory of quasi-Newton methods given in [2], [4].

In order to avoid restating hypotheses many times, we shall specify one standard assumption here.

Basic Assumption: $f:\mathbb{R}^n \to \mathbb{R}$ is twice continuously diffferentiable on an open convex set **D**, where **D** contains a strong local minimizer x^*, and there is a neighborhood **N** of x^* such that

$$\|f''(x) - f''(\bar{x})\| \le K\|x - \bar{x}\| \tag{4.1}$$

for all **x**, $\bar{x} \in N$.

From Lemma 3.1 given in [2], or Lemmata 4.1.15 and 4.1.16 given in [5], we obtain

Lemma 4.1. *Let* **f** *satisfy the Basic Assumption. Then, for any* **u**, **v**, $\in$ **D**,

$$\|f'(u) - f'(v) - f''(x^*)(u-v)\| \le \frac{K}{2}\sigma(u,v)\|v-u\|, \tag{4.2}$$

where $\sigma(u,v) = max\{\|u-x^*\|, \|v-x^*\|\}$. *Furthermore, there exist* $\varepsilon > 0$, $\beta_1 > 0$ *and* $\beta_2 > 0$ *such that if* $\sigma(u,v) \le \varepsilon$ *then* **u**, **v** $\in$ **D**, *and*

$$\frac{1}{\beta_1} \le \|f''(x)\| \le \beta_1, \quad \forall x \in N(x^*, \varepsilon) \tag{4.3}$$

$$\frac{\|u-v\|}{\beta_2} \le \|f'(u) - f'(v)\| \le \beta_2\|u-v\|, \quad \forall u,v \in N(x^*, \varepsilon). \tag{4.4}$$

Lemma 4.2. *Let f satisfy the Basic Assumption. Then there is a neighborhood* $N(x^*, \varepsilon)$ *and a constant* $\rho > 0$ *such that*

$$\left|\frac{1}{\alpha} - 1\right| \le \rho\sigma(x_-, x_c) \tag{4.5}$$

for all x_-, $x_c \in N(x^*, \varepsilon)$ *and* $x_- \ne x_c$, *where* α *is defined by* (2.8).

Proof: At the first, the inequality

$$|f_- - \left[f_c + (x_- - x_c)^T g_c + \tfrac{1}{2}(x_- - x_c)^T f''(x_c)(x_- - x_c)\right]| \le \frac{K}{6}\|x_- - x_c\|^3 \tag{4.6}$$

holds (for example, see Lemma 4.1.14 given in [5]). With estimates (4.3) and (4.1) we have

$$\begin{aligned}
&|(x_- - x_c)^T(g_c - g_-)| \\
&= \left|(x_c - x_-)^T\left[\int_0^1 (f''(x_- + \theta(x_c - x_-)) - f''(x_-))d\theta + f''(x_-)\right](x_c - x_-)\right| \\
&\ge |(x_c - x_-)^T f''(x_-)(x_c - x_-)| - \int_0^1 \|f''(x_- + \theta(x_c - x_-)) \\
&\quad - f''(x_-)\|d\theta\|x_c - x_-\|^2 \ge \left(\frac{1}{\beta_1} - K\|x_c - x_-\|\right)\|x_c - x_-\|^2.
\end{aligned}$$

Thus, if $\varepsilon > 0$ is small enough such that

$$\frac{1}{\beta_1} - K\|x_c - x_-\| \ge \frac{1}{2\beta_1},$$

then we get

$$|(x_c - x_-)^T(g_c - g_-)| \ge \frac{1}{2\beta_1}\|x_c - x_-\|^2, \quad \forall x_-, x_c \in N(x^*, \varepsilon). \tag{4.7}$$

If α is defined by (2.8a), then using (4.6), (4.7), (4.1) and $\|x_c - x_-\| \le 2\sigma(x_c, x_-)$ we have

$$\begin{aligned}
|\alpha - 1| &= |2(f_- - f_c) + (x_c - x_-)^T g_c + (x_c - x_-)^T g_-|/|(x_c - x_-)^T(g_c - g_-)| \\
&= |\left[f_- - f_c - (x_- - x_c)^T g_c - \tfrac{1}{2}(x_- - x_c)^T f''(x_c)(x_- - x_c)\right]
\end{aligned}$$

$$
\begin{aligned}
&- \left[f_c - f_- - (x_c - x_-)^T g_- - \tfrac{1}{2}(x_c - x_-)^T f''(x)(x_c - x_-)\right] \\
&+ \tfrac{1}{2}(x_c - x_-)^T [f''(x_c) - f''(x_-)](x_c - x_-)| / |(x_c - x_-)^T (g_c - g_-)| \\
&\le \left(\frac{K}{6} \|x_c - x_-\|^3 + \frac{K}{6} \|x_- - x_c\|^3 + \frac{K}{2} \|x_c - x_-\|^3 \right) \frac{2\beta_1}{\|x_c - x_-\|^2} \\
&= \tfrac{5}{3} K\beta_1 \|x_c - x_-\| \le \tfrac{10}{3} K\beta_1 \sigma(x_c, x_-). \qquad (4.8)
\end{aligned}
$$

Similarly, if α is defined by (2.8b), then using (4.8), (4.7) and (4.4) we have

$$
\begin{aligned}
|\alpha - 1| &= \left| \frac{2\left(f_- - f_c + (x_c - x_-)^T g_c\right)}{(x_c - x_-)^T (g_c - g_-)} - 1 \right| + \left| \frac{(x_c - x_-)^T (g_c - g^*)^2}{(x_c - x_-)^T (g_c - g_-)} \right| \\
&\le \tfrac{5}{3} K\beta_1 \|x_c - x_-\| + 2\beta_1 \beta_2^2 \|x_c - x^*\|^2 \\
&\le 2\beta_1\left(\tfrac{5}{3}K + \beta_2^2\right)\sigma(x_c, x_-), \qquad (4.9)
\end{aligned}
$$

where

$$g^* = f'(x^*) = 0 \quad \text{and} \quad \sigma(x_c, x_-) \le 2\varepsilon < 1.$$

Now, let $\rho = 4\beta_1\left(\frac{5}{3}K + \beta_2^2\right)$. From (4.8) and (4.9) we have

$$|\alpha - 1| \le \tfrac{1}{2}\rho\sigma(x_c, x_-).$$

Furthermore, let ε be small enough such that

$$1 - \tfrac{1}{2}\rho\sigma(x_c, x_-) \ge \tfrac{1}{2}.$$

Then we obtain

$$\left| \frac{1}{\alpha} - 1 \right| = \frac{|1 - \alpha|}{|\alpha|} \le \frac{\tfrac{1}{2}\rho\sigma(x_c, x_-)}{1 - \dfrac{\rho}{2}\sigma(x_c, x_-)} \le \rho\sigma(x_c, x_-)$$

for all $x_-, x_c \in N(x^*, \varepsilon)$.

With the above preliminary results it will be possible to give certain bounds of the rate of possible growth of the matrices $\{H_k\}$.

Lemma 4.3. *Let* **f** *satisfy the Basic Assumption. Then there exists a neighborhood* $N(x^*,\varepsilon)$ *of* x^* *and constants* $\alpha_1 > 0$, $\alpha_2 > 0$, *such that*

$$\|\bar{H} - A^{-1}\|_A \le (1 + \alpha_1 \sigma(x, \bar{x}))\|H - A^{-1}\|_A + \alpha_2 \sigma(x, \bar{x}), \qquad (4.10)$$

where $\|E\|_A = \|A^{\frac{1}{2}} E A^{\frac{1}{2}}\|_F$, $\bar{H}$, H and A denote H_k, H_{k-1} and $f''(x^*)$, respectively.

Proof: Due to (3.1) we have

$$
\begin{aligned}
\bar{H} &= \left(I - \frac{sy^T}{s^T y}\right) H \left(I - \frac{ys^T}{s^T y}\right) + \frac{ss^T}{s^T y} + \left(\frac{1}{\alpha} - 1\right)\frac{ss^T}{s^T y} \\
&= \bar{H}_{BFGS} + \left(\frac{1}{\alpha} - 1\right)\frac{ss^T}{s^T y},
\end{aligned}
$$

where $s=\bar{x}-x$, $y=\bar{g}-g$, and $\bar{H}_{\mathrm{BFGS}}$ is the classical BFGS update formula. Let ε be small enough such that Lemma 4.1 and Lemma 4.2 hold. Now, using the estimate of Broyden, Dennis, and Moré given in [2], for any $x, \bar{x} \in N(x^*, \varepsilon)$ we obtain

$$\|\bar{H}_{BFGS}-A^{-1}\|_A \leq (1+\hat{\alpha}_1\sigma(x,\bar{x}))\|H-A^{-1}\|_A+\hat{\alpha}_2\sigma(x,\bar{x}), \quad (4.11)$$

where $\hat{\alpha}_1>0$, $\hat{\alpha}_2>0$ are constants.

On the other hand, from (4.3) and (4.7) we have

$$\left\|\frac{ss^T}{s^Ty}\right\|_A = \frac{1}{|s^Ty|}\|A^{1/2}ss^TA^{1/2}\|_F = \frac{1}{|s^Ty|}\|A^{1/2}s\|^2 \leq 2\beta_1^2. \quad (4.12)$$

With (4.11), (4.12) and (4.5) we obtain

$$\begin{aligned}\|\bar{H}-A^{-1}\|_A &\leq (1+\hat{\alpha}_1\ \sigma(x,\bar{x}))\|H-A^{-1}\|_A+\hat{\alpha}_2\ \sigma(x,\hat{x})+\left|\frac{1}{\alpha}-1\right|\frac{\|ss^T\|_A}{|s^Ty|}\\ &\leq (1+\alpha_1\ \sigma(x,\bar{x}))\|H-A^{-1}\|_A+\alpha_2\ \sigma(x,\bar{x}),\end{aligned}$$

where $\alpha_1=\hat{\alpha}_1$, $\alpha_2=\hat{\alpha}_2+2\rho\beta_1^2$.

Now we analyse the local convergence of Algorithms 3.1-3.3.

Theorem 4.4. *Let* f *satisfy the Basic Assumption, and let the sequence* $\{x_k\}$ *be derived by Algorithm* 3.1, *Algorithm* 3.2, *or Algorithm* 3.3,

$$x_{k+1}=x_k-\theta_kH_kg_k, \quad k=0,1,\ldots,$$

where H_k *is updated by formula* (3.1), *and* θ_k *is given by*

$$\theta_k=\begin{cases}\dfrac{1}{1-t_kg_k^TH_kg_k} & \text{in Algorithm 3.1,}\\[2ex] \dfrac{1}{1-t_kg_{k-1}^TH_kg_k} & \text{in Algorithm 3.2,}\\[2ex] 1 & \text{in Algorithm 3.3.}\end{cases} \quad (4.13)$$

Let $N(x^*,\varepsilon)$ *be a neighborhood of* x^* *such that Lemmas 4.1–4.3 hold for all* $x, \bar{x} \in N(x^*,\varepsilon)$, *and* $A=f''(x^*)$. *Then, given any* $r\in(0,1)$, *there are positive constants* $\varepsilon=\varepsilon(r)$, $\delta=\delta(r)$ *such that if*

$$\|x_0-x^*\|<\varepsilon, \quad \|H_0-A^{-1}\|_A<\delta,$$

the sequence $\{x_k\}$ *is well-defined, converges to* x^*, *and*

$$\|x_{k+1}-x^*\|\leq r\|x_k-x^*\|, \quad k=0,1\ldots$$

Moreover, the quantities $\|H_k\|$, $\|H_k^{-1}\|$ *are uniformly bounded for all* **k**.

Proof: Let $r\in(0,1)$ be given. Due to (4.3) we have $\|A\|\leq\beta_1$ and $\|A^{-1}\|\leq\beta_1$. We remind ourselves that for fixed **A** the equivalence of matrix norms implies the existence of a constant $\eta>0$ such that

$$\|E\|\leq\eta\|E\|_A, \quad \text{for all } E\in\mathbb{R}^{n\times n}.$$

Now choose $\varepsilon > 0$ and $\delta > 0$ such that

(a) $\frac{1}{2}\beta_1 K\varepsilon + \delta(2(1+\delta)\eta + \beta_1)\beta_2 \le r$,
(b) $(2\delta\alpha_1 + \alpha_2)\varepsilon < (1-r)\delta$,
(c) $2\eta\beta_1\delta < r$,
(d) $T\eta(\sqrt{n} + 2\delta)(1+\delta)\beta_2^2\varepsilon^2 < \delta$.

Suppose $\|x_0 - x^*\| < \varepsilon$ and $\|H_0 - A^{-1}\|_A < \delta$. Because θ_0 is an arbitrary positive constant in the algorithms, without loss of generality we choose $\theta_0 > 0$ such that $|1-\theta_0| < \delta$. Then $\|H_0 - A^{-1}\| < \eta\delta < 2\eta\delta$ and

$$\begin{aligned}\|x_1 - x^*\| &= \|x_0 - x^* - \theta_0 H_0 g_0\| \\ &= \|[x_0 - x^* - A^{-1}(g_0 - g^*)] + \theta_0(A^{-1} - H_0)(g_0 - g^*) \\ &\quad + (1-\theta_0)A^{-1}(g_0 - g^*)\| \\ &\le \|A^{-1}\|\,\|g_0 - g^* - A(x_0 - x^*)\| + \theta_0\|A^{-1} - H_0\|\,\|g_0 - g^*\| \\ &\quad + |1-\theta_0|\,\|A^{-1}\|\,\|g_0 - g^*\|,\end{aligned}$$

where $g^* = f'(x^*) = 0$. Therefore, due to (4.2) and (4.4),

$$\begin{aligned}\|x_1 - x^*\| &\le \tfrac{1}{2}\beta_1 K\|x_0 - x^*\|^2 + 2\eta\delta(1+\delta)\beta_2\|x_0 - x^*\| + \delta\beta_1\beta_2\|x_0 - x^*\| \\ &\le \left(\tfrac{1}{2}\beta_1 K\varepsilon + 2(1+\delta)\delta\eta\beta_2 + \delta\beta_1\beta_2\right)\|x_0 - x^*\| \\ &\le r\|x_0 - x^*\| \quad \text{from (a)}.\end{aligned} \tag{4.14}$$

The proof now proceeds by induction on **k**. Suppose that

$$\|H_k - A^{-1}\|_A < 2\delta, \quad \|x_{k+1} - x^*\| \le r\|x_k - x^*\| \quad \text{and} \quad |1-\theta_k| < \delta$$

for $\mathbf{k} = 0,1,\ldots,\mathbf{m}-1$. Then Lemma 4.3 implies that

$$\begin{aligned}\|H_{k+1} - A^{-1}\|_A - \|H_k - A^{-1}\|_A &\le \alpha_1\sigma(x_k, x_{k+1})\|H_k - A^{-1}\|_A + \alpha_2\sigma(x_k, x_{k+1}) \\ &\le 2\alpha_1\delta\varepsilon r^k + \delta_2\varepsilon r^k \\ &= (2\delta\alpha_1 + \delta_2)\varepsilon r^k\end{aligned} \tag{4.15}$$

for $\mathbf{k} = 0,1,\ldots,\mathbf{m}-1$. Summing both sides of (4.15) from $\mathbf{k} = 0,1,\ldots,\ \mathbf{m}-1$ gives

$$\|H_m - A^{-1}\|_A - \|H_0 - A^{-1}\|_A \le \frac{\varepsilon}{1-r}(2\delta\alpha_1 + \alpha_2) < \delta \quad \text{from (b)}.$$

Thus $\|H_m - A^{-1}\|_A < 2\delta$ and $\|H_m\|_A \le \|A^{-1}\|_A + 2\delta$. Hence $\|H_m\| \le \eta(\sqrt{n} + 2\delta)$. The Banach perturbation lemma (for example, see § 2.3.2 in [9]) yields

$$\|H_k^{-1}\| \le \beta_1/(1-r), \quad k = 0,1,\ldots,m$$

due to $\|A\| \le \beta_1$, $\|H_k - A^{-1}\| \le \eta\|H_k - A^{-1}\|_A \le 2\eta\delta$ and (c).

Using $|t_m| \le T$, (4.4) and (d), it is easy to see that for Algorithm 3.1

$$|1-\theta_m| \le \frac{|t_m g_m^T H_m g_m|}{1 - |t_m g_m^T H_m g_m|} \le \frac{|t_m|\,\|H_m\|\,\|g_m\|^2}{1 - |t_m|\,\|H_m\|\,\|g_m\|^2}$$

$$\leq \frac{T\eta(\sqrt{n}+2\delta)\beta_2^2\|x_m-x\|^2}{1-T\eta(\sqrt{n}+2\delta)\beta_2^2\|x_m-x^*\|^2} \leq \frac{T\eta(\sqrt{n}+2\delta)\beta_2^2\, r^{2m}\,\varepsilon^2}{1-T\eta(\sqrt{n}+2\delta)\beta_2^2\, r^{2m}\,\varepsilon^2}$$

$$< \frac{T\eta(\sqrt{n}+2\delta)\beta_2^2\,\varepsilon^2}{1-T\eta(\sqrt{n}+2\delta)\beta_2^2\,\varepsilon^2} < \delta.$$

Similarly, for Algorithm 3.2 we have

$$|1-\theta_m| \leq \frac{|t_m g_{m-1}^T H_m g_m|}{1-|t_m g_{m-1}^T H_m g_m|} \leq \frac{T\eta(\sqrt{n}+2\delta)\beta_2^2 r^{2m-1}\varepsilon^2}{1-T\eta(\sqrt{n}+2\delta)\beta_2^2 r^{2m-1}\varepsilon^2} < \delta.$$

For Algorithm 3.3, it is obvious that $|1-\theta_m| < \delta$ for any $\delta > 0$.

Also, since $\|H_m - A^{-1}\|_A < 2\eta\delta$, $|1-\theta_m| < \delta$, it follows that

$$\|x_{m+1}-x^*\| \leq \left(\tfrac{1}{2}\beta_1 K\varepsilon + 2(1+\delta)\eta\beta_2\delta + \beta_1\beta_2\delta\right)\|x_m - x^*\|$$
$$\leq r\|x_m - x^*\| \quad \text{from (a)},$$

exactly as in (4.14) with x_m in place of x_0, x_{m+1} in place of x_1. This completes the induction and we have

$$\|H_k\| \leq \eta(\sqrt{n}+2\delta), \quad \|H_k^{-1}\| \leq \beta_1/(1-r)$$

as above for all $\mathbf{k} \geq 0$.

Corollary 4.5. Let the hypotheses of Theorem 4.4 hold. Then

$$\lim_{k\to\infty} \theta_k = 1,$$

where θ_k is defined by (4.13).

Proof: According to Theorem 4.4, the sequence $\{x_k\}$ converges with a Q-linear rate to a strong local minimizer x^*. Therefore, the sequence $\{f'(x_k)\}$ converges to $f'(x^*)=0$ by the continuity of $\mathbf{f}'(\mathbf{x})$. It is clear that

$$\lim_{k\to\infty} \theta_k = 1$$

due to $\|H_k\|$ uniformly bounded for all $\mathbf{k}$.

Note that during the above discussion we always put $\lambda_k \equiv 1$, where λ_k was the step length of the line search. Now let us regard θ_k as a new step length of the line search. Then due to Corollary 4.5, the Q-superlinear convergence of Algorithms 3.1–3.3 is only a corollary of Theorem 8.9 of Dennis and Moré given in [4]. We state it as follows.

Theorem 4.6. Let the hypotheses of Theorem 4.4 hold and consider θ_k defined by (4.13) as a step length of the line search such that $\{\theta_k\}$ converges to unity. If the sequence $\{x_k\}$ is generated by Algorithm 3.1, 3.2 or 3.3, then $\{x_k\}$ converges Q-superlinearly to x^*.

5. Numerical Experiments

Because the θ_k defined by (4.13) may be very small at the beginning of the iterations, this fact can increase the number of line searches. So in the implementation of Algorithms 3.1 and 3.2 the restricted condition $t_k \in [-T, 0]$ is changed into $\theta_k \in [\mu, 1]$. Here $\mu > 0$ is a fixed constant.

We have tested 17 functions that have been used extensively to test unconstrained optimization algorithms. These functions are all described in [9]. All the starting values are the standard starting points. It would be of interest that when $[\rho_1, \rho_2] = [0.7, 1.3]$ the numerical results of Algorithms 3.1–3.3 and the classical BFGS algorithm (i.e., $\rho_1 = \rho_2 = 1$) are about the same for the most part of testing functions. The examples are Beale, Helical valley, Biggs EXP6, Gaussian, Box three-dimensional (m = 4, 5), Variably dimensioned (n = 2, 5, 10), Watson (n = 9, 12), Penalty I (n = 10), Brown badly scaled, Brown and Dennis, Trigonometric (n = 2, 5, 10), Rosenbrock (n = 2) and Chebyquad (n = 8) functions. So we omit to tabulate their results. The other numerical results are given in Table 1.

In the implementation we put $\varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-8}$, $\varepsilon_3 = 10^{-10}$ and $\mu = 0.125$. Our algorithms use only a bisection search. All tests were done on a microcomputer 386/33 in double precision arithmitic.

In Table 1 the following notation is used:

NIT = number of iterations,
NFE = number of function evaluations,
NLS = number of line searches.

6. Final Remarks

In this paper, we only proved that the conic model interpolation with BFGS update has local and superlinear convergence. It is obvious that one can prove that the conic model interpolation with Broyden convex family update

$$H = \theta H_{\text{BFGS}} + (1-\theta) H_{\text{DFP}}, \quad \theta \in [0,1]$$

also has local and superlinear convergence using the technique of Broyden, Dennis and Moré given in [2].

An important problem is about the choice of the region $[\rho_1, \rho_2]$. In our numerical experiments we only specified the following 7 regions in the infinite cases

$$[0.05, 20],\ [0.1, 10],\ [0.5, 2],\ [0.1, 1.9],\ [0.3, 1.7],\ [0.5, 1.5],\ [0.7, 1.3].$$

In fact, the region $[\rho_1, \rho_2]$ can sensitively affect the number of function evaluations and the number of line search. As long as the good choice of region

Table 1

Function	Algorithm	$[\rho_1, \rho_2]$	NIT	NFE	NLS	Final f(x)	Final ‖f′(x)‖
Powell badly scaled F. $\mathbf{n}=2$ $\mathbf{m}=2$	BFGS		161	219	57	$.16899082 \times 10^{-27}$	$.574671 \times 10^{-9}$
	3.1	[0.5, 2]	152	190	37	$.10628152 \times 10^{-25}$	$.375959 \times 10^{-8}$
	3.2	[0.5, 2]	150	186	35	$.35401153 \times 10^{-28}$	$.329648 \times 10^{-11}$
	3.3	[0.5, 2]	155	199	43	$.48571718 \times 10^{-27}$	$.894584 \times 10^{-9}$
Box three dimensional Function $\mathbf{n}=3$ $\mathbf{m}=3$	BFGS		46	53	6	$.12138080 \times 10^{-18}$	$.230009 \times 10^{-9}$
	3.1	[0.1, 1.9]	44	49	4	$.20045003 \times 10^{-16}$	$.144546 \times 10^{-8}$
	3.2	[0.1, 1.9]	40	44	3	$.37296138 \times 10^{-15}$	$.104173 \times 10^{-8}$
	3.3	[0.1, 1.9]	24	26	1	$.37792434 \times 10^{-16}$	$.626614 \times 10^{-8}$
Watson Function $\mathbf{n}=6$ $\mathbf{m}=31$	BFGS		39	54	14	$.22876701 \times 10^{-2}$	$.446088 \times 10^{-7}$
	3.1	[0.1, 1.9]	38	46	7	$.22876701 \times 10^{-2}$	$.334181 \times 10^{-6}$
	3.2	[0.1, 1.9]	39	50	10	$.22876701 \times 10^{-2}$	$.493558 \times 10^{-6}$
	3.3	[0.1, 1.9]	39	51	11	$.22876701 \times 10^{-2}$	$.155835 \times 10^{-7}$
Penalty Function I $\mathbf{n}=4$ $\mathbf{m}=5$	BFGS		73	92	18	$.22499774 \times 10^{-4}$	$.300199 \times 10^{-8}$
	3.1	[0.1, 10]	61	63	1	$.22499774 \times 10^{-4}$	$.272200 \times 10^{-8}$
	3.2	[0.1, 10]	66	68	1	$.22499774 \times 10^{-4}$	$.789259 \times 10^{-9}$
	3.3	[0.1, 10]	91	96	4	$.22499774 \times 10^{-4}$	$.156513 \times 10^{-8}$
Penalty Function II $\mathbf{n}=4$ $\mathbf{m}=8$	BFGS		441	595	153	$.93762931 \times 10^{-5}$	$.722961 \times 10^{-8}$
	3.1	[0.1, 10]	363	399	35	$.93762931 \times 10^{-5}$	$.369483 \times 10^{-7}$
	3.2	[0.1, 10]	339	375	35	$.93762931 \times 10^{-5}$	$.115329 \times 10^{-6}$
	3.3	[0.1, 10]	372	410	37	$.93762931 \times 10^{-5}$	$.492943 \times 10^{-7}$
Penalty Function II $\mathbf{n}=10$ $\mathbf{m}=20$	BFGS		484	641	156	$.29366063 \times 10^{-3}$	$.137356 \times 10^{-6}$
	3.1	[0.1, 10]	472	464	36	$.29366056 \times 10^{-3}$	$.108975 \times 10^{-6}$
	3.2	[0.1, 10]	485	531	45	$.29366068 \times 10^{-3}$	$.303931 \times 10^{-6}$
	3.3	[0.1, 10]	461	503	41	$.29366060 \times 10^{-3}$	$.100490 \times 10^{-5}$
Wood Function $\mathbf{n}=4$ $\mathbf{m}=6$	BFGS		79	117	37	$.64619470 \times 10^{-19}$	$.496718 \times 10^{-8}$
	3.1	[0.5, 2]	83	115	31	$.34061612 \times 10^{-19}$	$.247168 \times 10^{-8}$
	3.2	[0.5, 2]	81	111	29	$.68378456 \times 10^{-20}$	$.981142 \times 10^{-10}$
	3.3	[0.5, 2]	83	111	27	$.13524181 \times 10^{-19}$	$.233128 \times 10^{-8}$
Extended Rosenbrock Function $\mathbf{n}=6$ $\mathbf{m}=6$	BFGS		70	101	30	$.25579219 \times 10^{-20}$	$.104231 \times 10^{-8}$
	3.1	[0.5, 2]	74	92	17	$.54969132 \times 10^{-19}$	$.456766 \times 10^{-8}$
	3.2	[0.5, 2]	72	96	23	$.38204581 \times 10^{-19}$	$.130123 \times 10^{-8}$
	3.3	[0.5, 2]	73	105	31	$.64044531 \times 10^{-18}$	$.957959 \times 10^{-8}$

Table 1 *(continued)*

Function	Algorithm	$[\rho_1, \rho_2]$	NIT	NFE	NLS	Final f(x)	Final $\|f'(x)\|$
Extended Rosenbrock Function $n = 10$ $m = 10$	BFGS		95	149	53	$.68416970 \times 10^{-19}$	$.458914 \times 10^{-8}$
	3.1	[0.1, 1.9]	100	133	32	$.91963239 \times 10^{-19}$	$.270758 \times 10^{-8}$
	3.2	[0.1, 1.9]	97	140	42	$.34074839 \times 10^{-18}$	$.474071 \times 10^{-8}$
	3.3	[0.1, 1.9]	95	142	46	$.40048342 \times 10^{-19}$	$.417084 \times 10^{-8}$
Extended Powell Singular Function $n = 4$ $m = 4$	BFGS		51	62	10	$.47533144 \times 10^{-14}$	$.559467 \times 10^{-8}$
	3.1	[0.1, 1.9]	41	52	10	$.72702178 \times 10^{-13}$	$.229041 \times 10^{-8}$
	3.2	[0.1, 1.9]	41	52	10	$.72686315 \times 10^{-13}$	$.232343 \times 10^{-8}$
	3.3	[0.1, 1.9]	54	65	10	$.13133898 \times 10^{-13}$	$.152569 \times 10^{-8}$
Extended Powell Singular Function $n = 8$ $m = 8$	BFGS		79	97	17	$.35032841 \times 10^{-13}$	$.498566 \times 10^{-8}$
	3.1	[0.7, 1.3]	83	100	16	$.32632772 \times 10^{-13}$	$.207912 \times 10^{-8}$
	3.2	[0.7, 1.3]	68	86	17	$.60691851 \times 10^{-12}$	$.828386 \times 10^{-8}$
	3.3	[0.7, 1.3]	50	67	16	$.94571608 \times 10^{-13}$	$.298124 \times 10^{-8}$
Extended Powell Singular Function $n = 12$ $m = 12$	BFGS		122	151	28	$.39141921 \times 10^{-15}$	$.477884 \times 10^{-8}$
	3.1	[0.5, 1.5]	99	121	21	$.28154379 \times 10^{-11}$	$.904334 \times 10^{-8}$
	3.2	[0.5, 1.5]	120	145	24	$.17121555 \times 10^{-13}$	$.388941 \times 10^{-8}$
	3.3	[0.5, 1.5]	77	103	25	$.31317460 \times 10^{-11}$	$.820248 \times 10^{-8}$
Chebyquad Function $n = 9$ $m = 9$	BFGS		67	91	23	$.48094067 \times 10^{-16}$	$.810873 \times 10^{-8}$
	3.1	[0.7, 1.3]	62	80	17	$.84896244 \times 10^{-18}$	$.138852 \times 10^{-8}$
	3.2	[0.7, 1.3]	54	89	34	$.27144198 \times 10^{-16}$	$.751233 \times 10^{-8}$
	3.3	[0.7, 1.3]	64	88	23	$.23632131 \times 10^{-17}$	$.203399 \times 10^{-8}$
Chebyquad Function $n = 10$ $m = 10$	BFGS		75	103	27	$.47727137 \times 10^{-2}$	$.152842 \times 10^{-7}$
	3.1	[0.1, 1.9]	63	80	16	$.47727137 \times 10^{-2}$	$.390727 \times 10^{-6}$
	3.2	[0.1, 1.9]	70	96	25	$.47727137 \times 10^{-2}$	$.317386 \times 10^{-6}$
	3.3	[0.1, 1.9]	81	1 11	29	$.47727137 \times 10^{-2}$	$.921132 \times 10^{-7}$

$[\rho_1, \rho_2]$, our new methods are certainly better than the classical BFGS method. Unfortunately, we do not know how to automatically determine the best region $[\rho_1, \rho_2]$. Therefore, this important project appears to be worthy of further research.

References

[1] Ariyawansa, K. A.: Deriving collinear scaling algorithms as extension of quasi-Newton methods and the local convergence of DFP and BFGS related collinear scaling algorithms. Math. Prog. *49*, 23–48 (1990).

[2] Broyden, C. G., Dennis, J. E. Moré, J. J.: On the local and superlinear convergence of quasi-Newton method. J. Inst. Math. Appl. *12*, 223–245 (1973).
[3] Davidon, W. C.: Conic approximation and collinear scalings for optimizations. SIAM J. Numer. Anal. *17*, 268–281 (1980).
[4] Dennis, J.E., Moré, J. J.: Quasi-Newton methods, motivation and theory. SIAM Rev. *19*, 46–89 (1977).
[5] Dennis, J. E., Schnabel, R. B.: Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall Series in Computational Mathematics, 1983.
[6] Dennis, J.E., Moré J.J.: A characterization of superlinear convergence and its application to quasi-Newton methods. Math. Comp. *28*, 549–560 (1979).
[7] Gourgeon, H., Nocedal, J.: A conic algorithm for optimization. SIAM J. Sci. Stat. Comput. *6*, 253–266 (1985).
[8] Moré, J. J., Garbow, B. S., Hillstron, K. E.: Testing unconstrained optimization software. ACM Trans. Math. Software *7*, 17–41 (1981).
[9] Orgeta, J. M., Rheinboldt, W. C.: Iterative solution of nonlinear equations in several variables. New York: Academic Press 1970.
[10] Schnabel, R. B.: Conic methods for unconstrained optimization and tensor methods for nonlinear equations. In: Mathematical programming: the state of the art (Bachem, A., Groschel, M., Korte, B., eds.), pp. 417–438. Bonn 1982.
[11] Songbai, S: Collinear scaling methods with a parameter vector, OR and decision making. Chengdu University of Science and Technology Press, China, pp. 426–435, 1992.
[12] Songbai, S.: A class of collinear scaling algorithms for unconstrained optimization, (1990).
[13] Sorensen, D. C.: The Q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization. SIAM J. Numer. Anal., *17*, 84–114 (1980).
[14] Xu, H., Sheng, S.: Broyden family of collinear scaling algorithm for unconstrained optimization. Numer. Math. *4*, 318–330 (1991).

S. Sheng
Department of Mathematics
Nanjing University
Nanjing 210008
The People's Republic of China

Offenlegung gem. § 25 Abs. 1 bis 3 Mediengesetz:
Unternehmensgegenstand: Verlag von wissenschaftlichen Büchern und Zeitschriften

An den Springer-Verlag KG ist beteiligt: Dr. Konrad F. Springer, Sachsenplatz 4–6, A-1201 Wien, als Kommanditist zu 52,38%. Geschäftsführer: Dr. Konrad F. Springer, Prof. Dr. Dietrich Götze, Ing. Wolfram F. Joos, Dipl.-Kfm. Claus Michaletz und Rudolf Siegle, alle Sachsenplatz 4–6, A-1201 Wien.

Verleger: Springer-Verlag KG, Sachsenplatz 4–6, A-1201 Wien. – Herausgeber: Prof. Dr. Hans J. Stetter, Institut für Angewandte und Numerische Mathematik der Technischen Universität Wien, Wiedner Hauptstrasse 6–10, A-1040 Wien. – Redaktion: Wiedner Hauptstraße 6–10, A-1040 Wien. – Satz und Umbruch: Compuscript Ltd., Shannon; Reproduktion und Offsetdruck: Novographic, Ing. W. Schmid, Maurer-Lange-Gasse 64, A-1238 Wien. – Verlagsort: Wien. – Herstellungsort: Wien. – Printed in Austria.