A nonmonotone inexact MFR method for solving symmetric nonlinear equations

Bo Zou and Li Zhang
Department of Mathematics, Changsha University of Science and Technology, Changsha 410004, China

Abstract

In [3], Li and Wang proposed a derivative-free modified Fletcher-Reeves (MFR) method for solving symmetric nonlinear equations, which determines the stepsize and the search direction simultaneously by the use of some norm descent backtracking type line search. This method is an extension of the MFR method proposed by Zhang, Zhou and Li [4] for general optimization. In this paper, based on the idea of the alternate direction method, we present a nonmonotone inexact MFR method for symmetric nonlinear equations with global convergence, which can reduce the computational cost of function values. Some preliminary numerical results are reported to show its efficiency.

Keywords. The MFR method; symmetric nonlinear equations; global convergence; line search.

AMS subject classification 2010. 90C30, 65K05.

1 Introduction

In this paper, we consider the symmetric nonlinear system

$$F(x) = 0,$$  \hspace{1cm} (1.1)

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable mapping, and its Jacobian $J(x) \triangleq J'(x)$ is symmetric. There are many practical problems with symmetric Jacobian such as the KKT systems of equality constrained optimization problems, the discretized two-point boundary value problem, and etc. [2].

The symmetric nonlinear problem (1.1) has been considered by some authors. Li and Fukushima [2] proposed a globally and superlinearly convergent Gauss-Newton-based BFGS method for such problems. This method has been extended to the norm descent case and the symmetric nonlinear least squares by Gu et al. [1] and Zhou [5], respectively. Recently, Li and Wang [3] introduced a derivative-free method for (1.1), which is an extension of the modified Fletcher-Reeves (MFR) method proposed by Zhang et al. [4] for general unconstrained optimization.

---

\*This work was supported by the NSF (11371073) of China, the Project of the Scientific Research Fund (13B137) of the Hunan Provincial Education Department, and the NSF (14JJ3084) of Hunan Province.
In this paper, we will further study the MFR method in [4] and modify it to solve the symmetric problem (1.1) using a different way from that of [3]. Consider the following smooth unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x).$$

(1.2)

The search direction $d_k$ generated by the MFR method in [4] is given by

$$d_k = \begin{cases} -\nabla f(x_k), & \text{if } k = 0, \\ -\theta_k^{MFR} \nabla f(x_k) + \beta_k^{FR} d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

(1.3)

where $\nabla f(x_k)$ is the gradient of $f$ at $x_k$ and

$$\theta_k^{MFR} = \frac{d_{k-1}^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{\|\nabla f(x_{k-1})\|^2}, \quad \beta_k^{FR} = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}.$$  

(1.4)

An important feature of the MFR method is that $d_k^T \nabla f(x_k) = -\|\nabla f(x_k)\|^2$.

Throughout the paper, we denote $s_k = x_{k+1} - x_k = \alpha_k d_k$, $F_k = F(x_k)$ and $J_k = J(x_k)$. We assume that the problem (1.1) is symmetric and $f$ in (1.2) is specified by

$$f(x) \triangleq \frac{1}{2} \|F(x)\|^2.$$  

(1.5)

Then the problem (1.1) is equivalent to the global optimization problem (1.2). However, when $f(x)$ is given by (1.5), $\nabla f(x) = J(x)^T F(x) = J(x) F(x)$, which requires the exact computation of Jacobian or the exact gradient. Hence the MFR method (1.3) is not suitable for such problems whose Jacobian is not available or very difficult to compute.

To overcome this difficulty, Li and Wang [3] proposed the following derivative-free MFR method for solving (1.1). They consider the search direction with a parameter $\alpha > 0$ as follows:

$$d_k(\alpha) = \begin{cases} -g_k(\alpha), & \text{if } k = 0, \\ -\theta_k^{MFR}(\alpha) g_k(\alpha) + \beta_k^{FR}(\alpha) d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

(1.6)

with

$$g_k(\alpha) = \frac{F(x_k + \alpha F_k) - F(x_k)}{\alpha}.$$  

(1.7)

and

$$\theta_k^{MFR}(\alpha) = \frac{d_{k-1}^T (g_k(\alpha) - g_{k-1})}{\|g_{k-1}\|^2}, \quad \beta_k^{FR}(\alpha) = \frac{\|g_k(\alpha)\|^2}{\|g_{k-1}\|^2},$$

(1.8)

where $g_{k-1}$ is an estimation to $\nabla f(x_{k-1})$ to be determined. Then they use the following procedures to compute the stepsize $\alpha_k$ and $d_k$ simultaneously.

**Procedure 1.** Let $g_k(\alpha)$ be defined by (1.7) and $d_k(\alpha)$ be computed by (1.6) and (1.8). Given constants $\sigma_1, \sigma_2, \sigma_3 > 0$. Let $i_k$ be the smallest nonnegative integer such that the following inequality holds with $\alpha = \rho^i, i = 0, 1, \cdots$,

$$f(x_k + \alpha d_k(\alpha)) \leq f(x_k) + \sigma_1 (F(x_k + \alpha F_k) - F(x_k))^T d_k(\alpha) - \sigma_2 \|\alpha F_k\|^2 - \sigma_3 \|\alpha d_k(\alpha)\|^2.$$  

Let $d_k = d_k(\rho^{i_k})$ and $g_k = g_k(\rho^{i_k})$.

**Procedure 2.** Let $d_k$ be generated by Procedure 1. Let constants $\sigma_i, i = 1, 2, 3$ and $\rho$
be the same as those in Procedure 1. If \( i_k = 0 \), we let \( \alpha_k = 1 \). Otherwise, we let \( j_k \) be the largest positive integer \( j_k \in \{0, 1, 2, \ldots, i_k - 1\} \) satisfying
\[
f(x_k + \rho^{i_k-j_k} d_k) \leq f(x_k) + \sigma_1 (F(x_k + \rho^{i_k-j_k} F_k) - F(x_k))^T d_k - \sigma_2 \|\rho^{i_k-j_k} F_k\|^2 - \sigma_3 \|\rho^{i_k-j_k} d_k\|^2.
\]
Let \( \alpha_k = \rho^{i_k-j_k} \).

However, we note that Procedures 1-2 requires many computations on function values since it need to compute \( F(x_k + \alpha F_k) \) and \( F(x_k + \alpha d_k(\alpha)) \). Based on this observation, we use the idea of the alternate direction method to reduce the computational cost on function values, that is, we first give a stepsize \( \alpha_{k-1} \), then determine the search direction \( d_k \) by \( \alpha_{k-1} \) and use some line search to compute the next stepsize \( \alpha_k \). In [2], Li and Fukushima used the term
\[
g_k = \frac{F(x_k + \alpha_{k-1} F_k) - F(x_k)}{\alpha_{k-1}},
\]
to approximate \( \nabla f(x_k) \). Hence, if we replace the terms \( \nabla f(x_k) \) and \( \nabla f(x_{k-1}) \) in (1.3) and (1.4) by the terms \( g_k \) and \( g_{k-1} \), respectively, then we obtain an inexact MFR method for solving (1.1), that is,
\[
d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -\theta_k g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases}
\]
where \( g_k \) is defined by (1.9) and
\[
\theta_k = \frac{d_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad \beta_k = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad y_{k-1} = g_k - g_{k-1}.
\]
It is easy to verify that
\[
g_k^T d_k = -\|g_k\|^2,
\]
which implies that
\[
\|g_k\| \leq \|d_k\|. \tag{1.13}
\]

Since \( d_k \) given by (1.10) may be not a descent direction of (1.5), the standard Wolfe and Armijo line searches can not be used to compute the stepsize directly. Hence, we adopt the following nonmonotone line search, which was proposed by Li and Fukushima in [2], to compute the next stepsize \( \alpha_k \). Let \( \sigma_1 > 0, \sigma_2 > 0, \eta > 0, r \in (0, 1) \) be constants and \( \{\eta_k\} \) be a given positive sequence such that
\[
\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty.
\]
Let \( \alpha_k = \max\{1, r, r^2, \ldots\} \) satisfy
\[
f(x_k + \alpha d_k) \leq f(x_k) - \sigma_1 \|\alpha d_k\|^2 - \sigma_2 \|\alpha F_k\|^2 + \eta_k f(x_k). \tag{1.15}
\]
It is clear that the line search (1.15) is well-defined. This line search can not guarantee that the function value sequence \( \{f(x_k)\} \) is decreasing. Therefore, we obtain the following nonmonotone inexact MFR method for (1.1).

**Algorithm 1.**
Step 0. Choose \( x_0 \in \mathbb{R}^n, \sigma_1 > 0, \sigma_2 > 0, \alpha_- > 0, r \in (0, 1) \) and a positive sequence \( \{\eta_k\} \) satisfying (1.14). Let \( k := 0 \).

Step 1. Compute \( d_k \) by (1.9)-(1.11).

Step 2. Compute \( \alpha_k \) by the line search (1.15).

Step 3. Set \( x_{k+1} = x_k + \alpha_k d_k \).

Step 4. Let \( k := k + 1 \) and go to Step 1.

In the next section, we show global convergence of Algorithm 1 under some assumptions. In Section 3, we report some numerical results.

2 Global convergence

In this section, we prove global convergence of Algorithm 1. To this end, we use the following assumption.

Assumption 1.

(i) The level set \( \Omega = \{x \mid f(x) \leq e^\eta f(x_0)\} \) is bounded.

(ii) In some neighborhood \( N \) of \( \Omega \), the Jacobian is Lipschitz continuous, namely, there exists a constant \( L > 0 \) such that

\[
\|J(x) - J(y)\| \leq L\|x - y\|, \quad \forall x, y \in N. \tag{2.1}
\]

It is clear that the sequence \( \{x_k\} \subset \Omega \). Moreover, Assumption 1 implies that there exist positive constants \( M_1, M_2 \) and \( L_1 \) such that

\[
\|F(x)\| \leq M_1, \quad \|J(x)\| \leq M_2, \quad \forall x \in N, \tag{2.2}
\]

\[
\|\nabla f(x) - \nabla f(y)\| \leq L_1\|x - y\|, \quad \forall x, y \in N. \tag{2.3}
\]

Lemma 2.1. Let Assumption 1 hold. Then we have

\[
\sum_{k=0}^{\infty} \|\alpha_k d_k\|^2 < \infty, \quad \sum_{k=0}^{\infty} \|\alpha_k F_k\|^2 < \infty. \tag{2.4}
\]

Proof. It follows from (1.15) and (1.14) directly. \( \square \)

Lemma 2.1 implies that

\[
\lim_{k \to \infty} \|\alpha_k d_k\| = \lim_{k \to \infty} \|s_k\| = 0, \quad \lim_{k \to \infty} \|\alpha_k F_k\| = 0. \tag{2.5}
\]

Lemma 2.2. Let the sequence \( \{x_k\} \) be generated by Algorithm 1. Then we have

\[
\frac{\|d_k\|^2}{\|g_k\|^4} \leq \sum_{i=0}^{k} \frac{1}{\|g_i\|^2}. \tag{2.6}
\]

Proof. From (1.10), (1.11) and (1.12), we have

\[
\|d_k\|^2 = \beta_k^2 \|d_{k-1}\|^2 - 2\theta_k g_k^T d_k - \theta_k^2 \|g_k\|^2
\]

\[
= \frac{\|g_k\|^4}{\|g_{k-1}\|^4} \|d_{k-1}\|^2 + 2\theta_k \|g_k\|^2 - \theta_k^2 \|g_k\|^2
\]

\[
= \frac{\|g_k\|^4}{\|g_{k-1}\|^4} \|d_{k-1}\|^2 - (1 - \theta_k)^2 \|g_k\|^2 + \|g_k\|^2
\]

\[
\leq \frac{\|g_k\|^4}{\|g_{k-1}\|^4} \|d_{k-1}\|^2 + \|g_k\|^2.
\]
This implies that
\[
\frac{\|d_k\|^2}{\|g_k\|^4} \leq \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} + \frac{1}{\|g_k\|^2},
\]
which together with \(d_0 = -g_0\) yields (2.6).

The following result shows that Algorithm 1 is globally convergent.

**Theorem 2.1.** Let Assumption 1 hold. Then the sequence \(\{x_k\}\) generated by Algorithm 1 converges globally, that is,
\[
\lim_{k \to \infty} \|\nabla f(x_k)\| = 0. \tag{2.7}
\]

**Proof.** We prove this theorem by contradiction. Suppose that (2.7) is not true, then there exists a positive constant \(\tau\) such that
\[
\|\nabla f(x_k)\| \geq \tau, \quad \forall k \geq 0. \tag{2.8}
\]

Since \(\nabla f(x_k) = J_k^TF_k\), (2.8) implies that there exists a positive constant \(\tau_1\) satisfying
\[
\|F_k\| \geq \tau_1, \quad \forall k \geq 0. \tag{2.9}
\]

Case (i): \(\limsup_{k \to \infty} \alpha_k > 0\). Then by (2.5), we have \(\liminf_{k \to \infty} \|F_k\| = 0\), which contradicts (2.9).

Case (ii): \(\limsup_{k \to \infty} \alpha_k = 0\). Since \(\alpha_k \geq 0\), this case implies that
\[
\lim_{k \to \infty} \alpha_k = 0. \tag{2.10}
\]

Moreover, by the definition of \(g_k\) in (1.9) and the symmetry of the Jacobian, we have
\[
\|g_k - \nabla f(x_k)\| = \left\| F(x_k + \alpha_{k-1}F_k) - F(x_k) - J_k^TF_k \right\|
\]
\[
= \left\| \int_0^1 (J(x_k + t\alpha_{k-1}F_k) - J_k)dtF_k \right\|
\]
\[
\leq L\alpha_{k-1}\|F_k\|^2 \leq LM_2\alpha_{k-1}, \tag{2.11}
\]
where we use (2.1) and (2.2) in the last inequality. (2.10), (2.11) and (2.8) show that there exists a constant \(\tau_2 > 0\) such that
\[
\|g_k\| \geq \tau_2, \quad \forall k \geq 0. \tag{2.12}
\]

By (1.9) and (2.2), we know
\[
\|g_k\| = \left\| \int_0^1 J(x_k + t\alpha_{k-1}F_k)F_k dt \right\| \leq M_1M_2, \quad \forall k \geq 0. \tag{2.13}
\]

Since \(\lim_{k \to \infty} \alpha_k = 0\), then \(\alpha'_k = \alpha_k/r\) does not satisfy (1.15), namely,
\[
f(x_k + \alpha'_k d_k) > f(x_k) - \sigma_1\|\alpha'_kd_k\|^2 - \sigma_2\|\alpha'_kF_k\|^2 + \eta_kf(x_k),
\]
which means that
\[
\frac{f(x_k + \alpha'_k d_k) - f(x_k)}{\alpha'_k} > -\sigma_1\alpha'_k\|d_k\|^2 - \sigma_2\alpha'_k\|F_k\|^2. \tag{2.14}
\]
By the mean-value theorem and (2.3), there exists $\theta_k \in (0, 1)$ such that
\[
\frac{f(x_k + \alpha'_k d_k) - f(x_k)}{\alpha_k} = \nabla f(x_k + \theta_k \alpha'_k d_k)^T d_k
\]
\[
= \nabla f(x_k)^T d_k + (\nabla f(x_k + \theta_k \alpha'_k d_k) - \nabla f(x_k))^T d_k
\]
\[
\leq \nabla f(x_k)^T d_k + L_1 \alpha'_k \|d_k\|^2.
\] (2.15)

Then by (2.14)-(2.15) and (1.12), we know
\[
\alpha_k \geq \frac{-\nabla f(x_k)^T d_k}{(L_1 + \sigma_k) \|d_k\|^2 + \sigma_2 \|F_k\|^2} = \frac{-g_k^T d_k + (g_k - \nabla f(x_k))^T d_k}{(L_1 + \sigma_k) \|d_k\|^2 + \sigma_2 \|F_k\|^2}.
\] (2.16)

From (2.11), (1.10), (2.13), (2.12) and (2.5), we get
\[
\| (g_k - \nabla f(x_k))^T d_k \| \leq L \alpha_{k-1} M_2^2 (\|g_k\| + \beta_k \|d_{k-1}\|)
\]
\[
= L \alpha_{k-1} M_2^2 \|g_k\| + L \alpha_{k-1} M_2^2 \|g_{k-1}\|^2 \|d_{k-1}\|
\]
\[
\leq L \alpha_{k-1} M_2^2 M_1 M_2 + \frac{L M_1^2 (M_1 M_2)^2}{r_2^2} \|s_{k-1}\| \to 0,
\]
which together with (2.16) and (2.12) shows that there exist two positive constants $C_1$ and $C_2$ such that
\[
\alpha_k \geq \frac{C_1}{\|d_k\|^2 + C_2}.
\] (2.17)

On the other hand, by (2.6), (2.12) and (2.13), we obtain
\[
\|d_k\|^2 \leq \frac{(M_1 M_2)^4}{r_2^4} (k + 1).
\] (2.18)

By (2.4) and (2.17), we get that
\[
\sum_{k=0}^{\infty} \frac{\|d_k\|^2}{(\|d_k\|^2 + C_2)^2} = \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2 + 2C_2 + \frac{C_2^2}{\|d_k\|^2}} < \infty.
\]

This together with (2.18), (1.13) and (2.12) yields that
\[
\sum_{k=0}^{\infty} \frac{1}{\frac{(M_1 M_2)^4}{r_2^4} (k + 1) + 2C_2 + \frac{C_2^2}{r_2^2}} \leq \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2 + 2C_2 + \frac{C_2^2}{\|d_k\|^2}} < \infty,
\]
which leads to a contradiction since
\[
\sum_{k=0}^{\infty} \frac{1}{\frac{(M_1 M_2)^4}{r_2^4} (k + 1) + 2C_2 + \frac{C_2^2}{r_2^2}} = \infty.
\]
The proof is then completed. \hfill \Box
3 Numerical experiments

In this section, we compare the performance of the following two methods for solving nonlinear equations (1.1).

- **DF-MFR:** the derivative-free modified Fletcher-Reeves method in [3]. We set \( \sigma_1 = \sigma_2 = \sigma_3 = 10^{-4}, \rho = 0.1 \).
- **Algorithm 1:** we set parameters \( \eta_k = \frac{1}{(k+1)^2}, \sigma_1 = \sigma_2 = 10^{-4}, \alpha_1 = 0.01 \) and \( r = 0.1 \).

The codes were written in Matlab 7.4 and run on a personal computer 2.66 GHz CPU processor and 1 GB RAM memory. We stopped the iteration if the total number of iterations exceeds \( 3 \times 10^3 \) or \( \| F_k \| \leq 10^{-3} \). We tested both methods on the following two test problems with different initial points and \( n \) values.

**Problem 1.** The discretized two-point boundary value problem [2]:

\[
F(x) = \begin{pmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & -1 & \\
& & & -1 & 2
\end{pmatrix} x + \frac{1}{(n+1)^2} (\sin x_1 - 1, \cdots, \sin x_n - 1)^T.
\]

**Problem 2.** The gradient of the Engval function [2]:

\[
F_1(x) = x_1(x_1^2 + x_2^2) - 1,
F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1, \quad i = 2, 3, \cdots, n - 1,
F_n(x) = x_n(x_{n-1}^2 + x_n^2).
\]

Table 1 lists numerical results of the two methods on the problems with the initial points \( x_0 = (-1, \cdots, -1)^T, x_0 = (1, \cdots, 1)^T \) and \( x_0 = (10, \cdots, 10)^T \). In Table 1, "P" indicates the problem; "Iter" and "Time" stand for the total number of iterations and the CPU time in seconds, respectively; "Fcnt" is the total number of function values; "-" means that the method failed to find the solution of the problem within \( 3 \times 10^3 \) iterations; "\( \| F_k \| \)" is the norm of the residual at the stopping point.

From Table 1, we can see that Algorithm 1 performed better than the DF-MFR method for both problems. The DF-MFR method needs more computations on function values and CPU time. Moreover, both methods are very sensitive to the initial points.

References

Table 1: Test results for the two methods on the test problems with different initial points and different sizes.

| P  |  x_0  | n   | Iter | Fcnt | Time | ||F_k|| | Iter | Fcnt | Time | ||F_k|| |
|----|-------|-----|------|------|------|---------|------|------|------|---------|------|---------|
| 1  | -1    | 10  | 781  | 5369 | 2.5211 | 9.744e-004 | 81 | 322 | 1.4576 | 9.466e-004 |
| 20 | 488   | 292 | 1.8019 | 9.834e-004 | 496 | 2235 | 1.6506 | 9.974e-004 |
| 30 | 610   | 3044 | 1.8014 | 9.975e-004 | 610 | 2437 | 1.75 | 9.973e-004 |
| 40 | 627   | 3129 | 1.991 | 9.969e-004 | 627 | 2505 | 1.8013 | 9.968e-004 |
| 50 | 844   | 4214 | 2.0508 | 9.999e-004 | 844 | 3373 | 1.9645 | 9.999e-004 |
| 1  | 1     | 10  | –    | –    | –    | –    | –    | –    | –    | –    | –    |
| 20 | 290   | 1444 | 3.5201 | 9.928e-004 | 289 | 1153 | 1.5069 | 9.824e-004 |
| 30 | 400   | 1994 | 1.9499 | 9.967e-004 | 400 | 1597 | 1.5236 | 9.967e-004 |
| 40 | 594   | 2964 | 2.0149 | 9.939e-004 | 594 | 2373 | 1.9458 | 9.939e-004 |
| 50 | 828   | 4134 | 2.1502 | 9.953e-004 | 828 | 3309 | 2.0345 | 9.953e-004 |
| 1  | 10    | 10  | 1077 | 7439 | 2.2202 | 9.892e-004 | 948 | 4691 | 1.9056 | 9.766e-004 |
| 20 | 579   | 2889 | 2.2852 | 9.999e-004 | 579 | 2313 | 1.6085 | 9.980e-004 |
| 30 | 1696  | 8472 | 2.4308 | 9.999e-004 | 1705 | 6816 | 2.1426 | 9.962e-004 |
| 40 | 1268  | 6334 | 2.3106 | 9.994e-004 | 1258 | 5029 | 2.3295 | 9.987e-004 |
| 50 | 2432  | 12152 | 3.4121 | 9.976e-004 | 2469 | 9872 | 3.2187 | 9.919e-004 |
| 2  | -1    | 10  | –    | 32004 | 4.4082 | 4.007e-002 | 152 | 750 | 2.9937 | 9.167e-004 |
| 100 | 43082 | 6.0406 | 6.904e+000 | 155 | 765 | 2.1315 | 9.760e-004 |
| 500 | 55274 | 10.1622 | 2.204e+001 | 128 | 630 | 2.6829 | 9.442e-004 |
| 1000 | 55188 | 14.8387 | 3.131e+001 | 129 | 635 | 2.9196 | 9.820e-004 |
| 2  | 1     | 10  | 155  | 1069 | 1.4685 | 9.808e-004 | 121 | 597 | 1.2991 | 9.821e-004 |
| 100 | 157   | 1081 | 2.8263 | 9.804e-004 | 195 | 966 | 1.7088 | 9.987e-004 |
| 500 | 154   | 1064 | 2.0754 | 9.585e-004 | 192 | 953 | 2.1407 | 9.816e-004 |
| 1000 | 217  | 1507 | 2.2584 | 9.750e-004 | 219 | 1089 | 2.3849 | 9.788e-004 |
| 3000 | 205  | 1423 | 2.5669 | 9.660e-004 | 220 | 1094 | 2.8989 | 9.834e-004 |
| 2  | 10    | 10  | 184  | 1296 | 1.5225 | 9.965e-004 | 244 | 1223 | 1.5723 | 1.000e+003 |
| 50  | 144   | 1030 | 1.5929 | 9.901e-004 | 192 | 966 | 1.4715 | 9.741e-004 |
| 100 | 1530  | 13800 | 3.4638 | 9.985e-004 | 155 | 781 | 1.7848 | 9.975e-004 |
| 200 | –     | 27708 | 4.9951 | 1.886e+003 | 828 | 4936 | 2.5067 | 9.935e-004 |
| 300 | 907   | 8065 | 3.0297 | 9.933e-004 | 853 | 5086 | 2.5759 | 9.952e-004 |
| 500 | 891   | 7937 | 3.0257 | 9.975e-004 | 127 | 641 | 1.6858 | 9.390e-004 |
| 1000 | 912  | 8084 | 3.6847 | 9.950e-004 | 717 | 3620 | 2.5784 | 9.991e-004 |
| 3000 | 176  | 1286 | 2.5952 | 9.950e-004 | 448 | 2280 | 2.9903 | 9.891e-004 |