# Three-Step Fixed-Point Quasi-Newton Methods for Unconstrained Optimisation 

J. A. Ford<br>Department of Computer Science, University of Essex<br>Wivenhoe Park, Colchester, Essex, U.K.<br>fordj@essex.ac.uk<br>S. Tharmlikit<br>Department of Computer Science, Burapha University<br>Chonburi, Thailand<br>stharm@buec4.buu.ac.th


#### Abstract

Multistep quasi-Newton methods were introduced by Ford and Moghrabi [1]. They address the problem of the unconstrained minimisation of a function $$
f: \mathbb{R}^{n} \rightarrow \mathbb{R}
$$ whose gradient and Hessian are denoted by $g$ and $G$, respectively. These methods generalised the standard construction of quasi-Newton methods and were based on employing interpolatory polynomials to utilise information from more than one previous step. In a series of papers, Ford and Moghrabi [2-5] have developed various techniques for determining the parametrisation of the interpolating curves. In [2], they introduced two-step metric-based methods which determine the set of parameter values required through measuring distances between various pairs of the iterates employed in the current interpolation. One of the most successful methods in [2] was found to be in the "fixed-point" class, in which the parametrisation of the interpolating curve is determined, at each iteration, by reference to distances measured from a fixed iterate.

As suggested in [1], multistep quasi-Newton methods can be constructed for any number of steps. In this paper, we therefore extend the previous work by describing the development of some threestep methods which use the "fixed-point" approach and data derived from the latest four iterates. The experimental results provide evidence that the new methods offer a significant improvement in performance when compared with the standard BFGS method and the unit-spaced three-step method, particularly as the dimension of the test problems grows. © 2005 Elsevier Ltd. All rights reserved.


Keywords-Unconstrained optimisation, Quasi-Newton methods, Multistep quasi-Newton methods.

## 1. INTRODUCTION

Multistep quasi-Newton methods were introduced by Ford and Moghrabi [1]. These methods replaced the quasi-Newton condition [6]

$$
\begin{equation*}
B_{i+1} \mathbf{s}_{i}=\mathbf{y}_{i} \tag{1}
\end{equation*}
$$

[^0]which the Hessian approximation $B_{i+1}$ is required to satisfy, with a similar condition
\[

$$
\begin{equation*}
B_{i+1} \mathbf{r}_{i}=\mathbf{w}_{i} \tag{2}
\end{equation*}
$$

\]

where $\mathbf{r}_{i}$ and $\mathbf{w}_{i}$ are vectors determined by the specific multistep method under consideration. These methods generalised the traditional approach to quasi-Newton methods, and were based on interpolatory functions which utilise information from the $m$ most recent steps, where ordinary quasi-Newton methods correspond to $m=1$. (In equation (1), $\mathbf{s}_{i}$ is the step taken from $\mathbf{x}_{i}$ to $\mathbf{x}_{i+1}$

$$
\begin{equation*}
\mathbf{s}_{i}=\mathbf{x}_{i+1}-\mathbf{x}_{i}, \tag{3}
\end{equation*}
$$

and $\mathbf{y}_{i}$ is the corresponding change in the gradient

$$
\begin{equation*}
\left.\mathbf{y}_{i}=\mathbf{g}\left(\mathbf{x}_{i+1}\right)-\mathbf{g}\left(\mathbf{x}_{i}\right) .\right) \tag{4}
\end{equation*}
$$

Equation (1) is a particular approximation to the so-called Newton equation [7]

$$
\begin{equation*}
G\left(\mathbf{x}_{i+1}\right) \frac{d \mathbf{x}\left(\tau^{*}\right)}{d \tau}=\frac{d \mathbf{g}\left(\mathbf{x}\left(\tau^{*}\right)\right)}{d \tau} \tag{5}
\end{equation*}
$$

where $\mathbf{x}(\tau)$ is any differentiable curve (denoted by $\mathbb{X}$ ) in $\mathbb{R}^{n}$ constrained to satisfy $\mathbf{x}\left(\tau^{*}\right)=\mathbf{x}_{i+1}$. Relation (5) follows immediately from a straightforward application of the chain rule to the vector function $\mathbf{g}(\mathbf{x}(\tau))$. The quasi-Newton equation (1) may be derived from the Newton equation (5) by taking $\mathbf{x}(\tau)$ to be the simple linear function

$$
\begin{equation*}
\mathbf{x}(\tau) \equiv \mathbf{x}_{i}+\tau \mathbf{s}_{i} \tag{6}
\end{equation*}
$$

and choosing $\tau^{*}=1$. From (6), we have

$$
\mathbf{x}(0)=\mathbf{x}_{i}, \quad \mathbf{x}(1)=\mathbf{x}_{i+1}, \quad \text { and } \quad \frac{d \mathbf{x}}{d \tau}=\mathbf{s}_{i}, \quad \forall \tau
$$

If the curve $\mathbf{g}(\mathbf{x}(\tau))$ is approximated by the interpolating linear polynomial $\mathbf{g}(\mathbf{x}(0))+\tau[\mathbf{g}(\mathbf{x}(1))-$ $\mathbf{g}(\mathbf{x}(0))]$, then we can estimate the derivative $\frac{\operatorname{dg}\left(\mathbf{x}\left(\tau^{*}\right)\right)}{d \tau}$ at $\tau^{*}=1$ by simple linear differencing

$$
\frac{d g(x(1))}{d \tau} \approx g(x(1))-g(x(0))=g\left(x_{i+1}\right)-g\left(x_{i}\right)=y_{i}
$$

Thus, (5) becomes

$$
\begin{equation*}
G\left(\mathbf{x}_{i+1}\right) \mathbf{s}_{i} \approx \mathbf{y}_{i} \tag{7}
\end{equation*}
$$

The standard quasi-Newton approach uses an approximation $B_{i+1}$ to the Hessian $G\left(\mathbf{x}_{i+1}\right)$. In the light of relation (7), $B_{i+1}$ is therefore required to satisfy

$$
\begin{equation*}
B_{i+1} \mathbf{s}_{i}=\mathbf{y}_{i} \tag{8}
\end{equation*}
$$

We now focus attention on multistep quasi-Newton approaches which are based on interpolatory polynomial forms for $\mathbb{X}$ and which make use of information from the $m$ most recent steps ( $m>1$ )

$$
\begin{equation*}
\mathbf{x}\left(\tau_{k}\right)=\mathbf{x}_{i-m+k+1}, \quad \text { for } k=0,1, \ldots, m \tag{9}
\end{equation*}
$$

Therefore, one way of representing the curve (if $\mathbf{x}(\tau)$ is taken to be a polynomial of degree $m$ ) is

$$
\begin{equation*}
\mathbf{x}(\tau) \equiv \sum_{k=0}^{m} \mathcal{L}_{k}(\tau) \mathbf{x}_{i-m+k+1} \tag{10}
\end{equation*}
$$

where $\mathcal{L}_{k}(\tau)$ is the $k^{\text {th }}$ standard Lagrangian polynomial of degree $m$, based on the set $\left\{\tau_{k}\right\}_{k=0}^{m}$. By analogy, the gradient (when restricted to $\mathbb{X}$ ) may be approximated by the corresponding interpolatory polynomial

$$
\begin{equation*}
\mathbf{g}(\mathbf{x}(\tau)) \approx \sum_{k=0}^{m} \mathcal{L}_{k}(\tau) \mathbf{g}\left(\mathbf{x}_{i-m+k+1}\right) \tag{11}
\end{equation*}
$$

Having selected form (10) for $\mathbf{x}(\tau)$ and with the approximation (11) for $\mathbf{g}(\mathbf{x}(\tau)$ ), we now apply the Newton equation (5) with the choice $\tau^{*}=\tau_{m}$. We determine $\frac{d \times\left(\tau_{m}\right)}{d \tau}$ from (10) and estimate $\frac{d g\left(x\left(\tau_{m}\right)\right)}{d \tau}$ from (11), so that we may substitute for these derivatives in (5). Therefore, for multistep methods, we obtain the condition that the new Hessian approximation $B_{i+1}$ is required to satisfy equation (2), where

$$
\begin{align*}
\frac{d \mathrm{x}\left(\tau_{m}\right)}{d \tau} & =\sum_{k=0}^{m} \mathcal{L}_{k}^{\prime}\left(\tau_{m}\right) \mathbf{x}_{i-m+k+1} \\
& \triangleq \mathbf{r}_{i}  \tag{12}\\
\frac{d \mathrm{~g}\left(\mathbf{x}\left(\tau_{m}\right)\right)}{d \tau} & \approx \sum_{k=0}^{m} \mathcal{L}_{k}^{\prime}\left(\tau_{m}\right) \mathbf{g}\left(\mathbf{x}_{i-m+k+1}\right) \\
& \triangleq \mathbf{w}_{i} \tag{13}
\end{align*}
$$

Formulae (12) and (13) may be rearranged to show that $r_{i}$ and $w_{i}$ can be represented in terms of the most recent "step-vectors" $\left\{\mathbf{s}_{i-j}\right\}_{j=0}^{m-1}$ and $\left\{\mathbf{y}_{i-j}\right\}_{j=0}^{m-1}$

$$
\begin{align*}
& \mathbf{r}_{i}=\sum_{j=0}^{m-1} \mathbf{s}_{i-j}\left\{\sum_{k=m-j}^{m} \mathcal{L}_{k}^{\prime}\left(\tau_{m}\right)\right\}  \tag{14}\\
& \mathbf{w}_{i}=\sum_{j=0}^{m-1} \mathbf{y}_{i-j}\left\{\sum_{k=m-j}^{m} \mathcal{L}_{k}^{\prime}\left(\tau_{m}\right)\right\} \tag{15}
\end{align*}
$$

In multistep methods, we can obtain the new Hessian approximation $B_{i+1}$ by (for example) use of a "BFGS-type" [8-11] formula, in which we substitute the vectors $r_{i}$ and $w_{i}$ for $s_{i}$ and $y_{i}$, respectively,

$$
\begin{equation*}
B_{i+1}=B_{i}-\frac{B_{i} \mathbf{r}_{i} \mathbf{r}_{i}^{\top} B_{i}}{\mathbf{r}_{i}^{\top} B_{i} \mathbf{r}_{i}}+\frac{\mathbf{w}_{i} \mathbf{w}_{i}^{\top}}{\mathbf{r}_{i}^{\top} \mathbf{w}_{i}} \tag{16}
\end{equation*}
$$

The remainder of this paper is organised as follows: in Section 2, we give (in a form suitable for computation) expressions for the vectors $\mathbf{r}_{i}$ and $\mathbf{w}_{i}$, for the cases when $m=2$ and $m=3$. We begin our derivation of specific three-step methods in Section 3, where we show the formulae which arise when the simplest possible choice for the values $\left\{\tau_{k}\right\}_{k=0}^{3}$ is made. This is followed, in Section 4, by a brief review of the concept of "metric-based" methods, and the derivation of certain three-step algorithms based on this concept. The paper concludes with a description, in Section 5, of the numerical experiments performed to assess the new methods and an evaluation of the experimental results in Section 6.

## 2. MULTISTEP METHODS

### 2.1. Two-Step Methods

In the case of two-step methods, we substitute $m=2$ in (14) and (15). Then the expressions for $\mathbf{r}_{i}$ and $w_{i}$ may be rewritten (after removal of a common scaling factor) in the form

$$
\begin{aligned}
\mathbf{r}_{i}^{(2-\text { step })} & =\mathbf{s}_{i}-\left\{\frac{\delta^{2}}{2 \delta+1}\right\} \mathbf{s}_{i-1} \\
\mathbf{w}_{i}^{(2-\text { step })} & =\mathbf{y}_{i}-\left\{\frac{\delta^{2}}{2 \delta+1}\right\} \mathbf{y}_{i-1}
\end{aligned}
$$

where

$$
\delta=\frac{\left(\tau_{2}-\tau_{1}\right)}{\left(\tau_{1}-\tau_{0}\right)}
$$

### 2.2. Three-Step Methods

We note that three-step methods will use data derived from the latest four iterates $\mathbf{x}_{i-2}, \mathbf{x}_{i-1}$, $\mathbf{x}_{i}$, and $\mathbf{x}_{i+1}$. Substituting $m=3$ in (14) and (15), we obtain (again after removal of a common scaling factor)

$$
\begin{align*}
\mathbf{r}_{i}^{(3-\text { step })}= & \mathbf{s}_{i}+\left\{\frac{-\delta_{1}^{2}\left(\delta_{2}+1\right)^{3}}{\left(\delta_{1}-\delta_{2}\right)\left(3 \delta_{1} \delta_{2}+\delta_{1}+\delta_{2}\right)}+1\right\} \mathbf{s}_{i-1} \\
& +\left\{\frac{\left(\delta_{1} \delta_{2}\right)^{2}}{\left(3 \delta_{1} \delta_{2}+\delta_{1}+\delta_{2}\right)}\right\} \mathbf{s}_{i-2},  \tag{17}\\
\mathbf{w}_{i}^{(3 \text {-step })}= & \mathbf{y}_{i}+\left\{\frac{-\delta_{1}^{2}\left(\delta_{2}+1\right)^{3}}{\left(\delta_{1}-\delta_{2}\right)\left(3 \delta_{1} \delta_{2}+\delta_{1}+\delta_{2}\right)}+1\right\} \mathbf{y}_{i-1} \\
& +\left\{\frac{\left(\delta_{1} \delta_{2}\right)^{2}}{\left(3 \delta_{1} \delta_{2}+\delta_{1}+\delta_{2}\right)}\right\} \mathbf{y}_{i-2}, \tag{18}
\end{align*}
$$

where

$$
\begin{align*}
& \delta_{1}=\frac{\left(\tau_{3}-\tau_{1}\right)}{\left(\tau_{1}-\tau_{0}\right)},  \tag{19}\\
& \delta_{2}=\frac{\left(\tau_{3}-\tau_{2}\right)}{\left(\tau_{2}-\tau_{0}\right)} . \tag{20}
\end{align*}
$$

## 3. UNIT-SPACED THREE-STEP METHOD

In the standard single-step quasi-Newton framework ( $m=1$ ), we have seen (Section 1) that the set of values $\left\{\tau_{k}\right\}_{k=0}^{1}$ can be taken to be $\tau_{0}=0$ and $\tau_{1}=1$. Therefore, a natural generalisation for multistep methods is to define the required values $\left\{\tau_{k}\right\}_{k=0}^{m}$ by means of a unit-spacing between consecutive members of the set

$$
\begin{equation*}
\tau_{k}=k-m+1, \quad \text { for } k=0,1,2, \ldots, m . \tag{21}
\end{equation*}
$$

Therefore, from (21), we obtain

$$
\tau_{0}=-2, \quad \tau_{1}=-1, \quad \tau_{2}=0, \quad \text { and } \quad \tau_{3}=1
$$

By (19) and (20), $\delta_{1}=2$ and $\delta_{2}=1 / 2$. Thus, $\mathrm{r}_{i}$ and $\mathbf{w}_{i}$ for the unit-spaced three-step method are given by

$$
\begin{aligned}
\mathbf{r}_{i} & =\mathbf{s}_{i}-\frac{7}{11} \mathbf{s}_{i-1}+\frac{2}{11} \mathbf{s}_{i-2}, \\
\mathbf{w}_{i} & =\mathbf{y}_{i}-\frac{7}{11} \mathbf{y}_{i-1}+\frac{2}{11} \mathbf{y}_{i-2} .
\end{aligned}
$$

Following [1], we denote this method by M3.
However, Ford and Moghrabi [2] developed some alternatives to unit-spacing for defining the parameter-values. The experimental results for the two-step methods developed in this manner showed substantial gains in performance by comparison with the unit-spaced methods, so that such methods are well worth investigation. The following section is devoted to the development of three-step methods using one of these alternative approaches.

## 4. METRIC-BASED APPROACHES

Metric-based methods determine the set of parameter values $\left\{\tau_{k}\right\}_{k=0}^{m}$ required through the measurement of distances between various pairs of the iterates involved in the current interpolation. These methods employ a metric $\varphi_{M}$ which is generated by any appropriate positive-definite matrix ( $M$, say)

$$
\begin{equation*}
\varphi_{M}\left(z_{1}, z_{2}\right)=\left\{\left(z_{1}-z_{2}\right)^{\top} M\left(z_{1}-z_{2}\right)\right\}^{1 / 2} \tag{22}
\end{equation*}
$$

where $\mathbf{z}_{1}, \mathbf{z}_{2} \in \mathbb{R}^{n}$. More specifically, two metric-based approaches were developed in [2]-the accumulative and the fixed-point approaches. The accumulative approach operates by accumulating the distance (measured by the metric $\varphi_{M}$ ) between consecutive iterates as they are traversed in the natural sequence. The fixed-point approach uses the metric to compute the distance of every iterate from one selected iterate (the so-called "fixed-point"). The choices of the matrix $M$ which were considered in [2] were: $M=I, M=B_{i}$, and $M=B_{i+1}$. We will confine our attention to the first two choices and to the fixed-point approach in this paper.

### 4.1. Fixed-Point Methods

These methods fix the origin for $\tau$ at $\mathbf{x}_{i+1}$, so that $\tau_{m}=0$. Then each of the other parameter values $\tau_{j}$ is computed by measuring (using $\varphi_{M}$ ) the distance from the corresponding iterate $\mathbf{x}_{i-m+j+1}$ to $\mathbf{x}_{i+1}$

$$
\begin{equation*}
\tau_{j}=-\varphi_{M}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i-m+j+1}\right), \quad \text { for } j=0,1, \ldots, m \tag{23}
\end{equation*}
$$

In what follows, we will investigate fixed-point methods which use three steps, so that

$$
\tau_{3}=0
$$

4.1.1. Three-step fixed-point approach, $M=I$

We first choose $M$ to be the unit matrix, so that, from (23), the values for $\tau$ are

$$
\begin{aligned}
& \tau_{3}=0 \\
& \tau_{2}=-\varphi_{I}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i}\right)=-\left[\left(\mathbf{x}_{i+1}-\mathbf{x}_{i}\right)^{\top}\left(\mathbf{x}_{i+1}-\mathbf{x}_{i}\right)\right]^{1 / 2}=-\left\|\mathbf{s}_{i}\right\|_{2} \\
& \tau_{1}=-\varphi_{I}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i-1}\right)=-\left\|\mathbf{s}_{i}+\mathbf{s}_{i-1}\right\|_{2} \\
& \tau_{0}=-\varphi_{I}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i-2}\right)=-\left\|\mathbf{s}_{i}+\mathbf{s}_{i-1}+\mathbf{s}_{i-2}\right\|_{2}
\end{aligned}
$$

We denote by $F_{I}^{(3)}$ the algorithm based on the repeated use of the formulae above (with a 'start-up' procedure [for the first two iterations] of a single-step iteration followed by a two-step fixed-point iteration using $M=I$ ).

### 4.1.2. Three-step fixed-point approach, $M=B_{i}$

Now we choose the matrix $M$ to be $B_{i}$. From (23), the $\tau$-values are

$$
\begin{align*}
\tau_{3} & =0 \\
\tau_{2} & =-\varphi_{B_{i}}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i}\right)=-\left[-t_{i} \mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)\right]^{1 / 2} \\
\tau_{1} & =-\varphi_{B_{i}}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i-1}\right)  \tag{24}\\
& \approx-\left[-t_{i} \mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)-2 t_{i} \mathbf{s}_{i-1}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)+\mathbf{s}_{i-1}^{\top} \mathbf{y}_{i-1}\right]^{1 / 2} \\
\tau_{0} & =-\varphi_{B_{i}}\left(\mathbf{x}_{i+1}, \mathbf{x}_{i-2}\right)
\end{align*}
$$

(We note that, in the derivation of the expressions above, it has been assumed that the new iterate $\mathbf{x}_{i+1}$ has been determined by a line-search along the "quasi-Newton" direction $-B_{i}^{-1} \mathbf{g}\left(\mathbf{x}_{i}\right)$, so that

$$
B_{i} \mathrm{~s}_{i}=-t_{i} \mathbf{g}\left(\mathbf{x}_{i}\right),
$$

for some known scalar $t_{i}$. We have also used the approximation

$$
B_{i} \mathbf{s}_{i-1} \approx \mathbf{y}_{i} .
$$

See [2].) We now observe that the expression for $\tau_{0}$ above can be calculated cheaply by using similar techniques. Thus, $\tau_{0}$ becomes (approximately)

$$
\begin{gather*}
\tau_{0} \approx-\left[-t_{i} \mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)-2 t_{i} \mathbf{s}_{i-1}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)+\mathbf{s}_{i-1}^{\top} \mathbf{y}_{i-1}\right. \\
\left.-2 t_{i} \mathbf{s}_{i-2}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right)+2 \mathbf{s}_{i-2}^{\top} \mathbf{y}_{i-1}+\mathbf{s}_{i-2}^{\top} \mathbf{y}_{i-2}\right]^{1 / 2} \tag{25}
\end{gather*}
$$

An iteration using this three-step method will be denoted by $F_{B_{i}}^{(3)}$.
The work of Ford and Moghrabi [4] showed that improvements in the numerical performance of at least one two-step method could be obtained by an application of the technique of alternation. This involves (for two-step methods) repeated application of a cycle comprising a single-step iteration followed by a two-step iteration. The two-step iteration calculates the required values for $\left\{\tau_{k}\right\}_{k=0}^{2}$ using relations known to be true because the previous iteration was 'single-step'. In light of this experience, we propose to study the three-step methods developed above with $M=I$ and $M=B_{i}$ in the form of new methods which use an extended form of alternation. More specifically, we propose two new algorithms (to be denoted, for convenience, by $F_{I}^{(1,2,3)}$ and $F_{B_{i}}^{(1,2,3)}$ ) based on the following cycles, each of which comprises three iterations:

$$
\begin{aligned}
& F_{I}^{(1,2,3)} \equiv\left\{S ; F_{I}^{(2)} ; F_{I}^{(3)}\right\} ; \\
& F_{B_{i}}^{(1,2)} \equiv\left\{S ; F_{B_{i}}^{(2)} ; F_{B_{i}}^{(3)}\right\} ;
\end{aligned}
$$

where $S$ denotes a single-step iteration and $F_{I}^{(2)}$ and $F_{B_{i}}^{(2)}$ denote two-step fixed-point iterations utilising $M=I$ and $M=B_{i}$, respectively (see [2]). We will compare the experimental performance of the following four algorithms with the standard BFGS single-step method in the next section:

- M3,
- $F_{I}^{(3)}$,
- $F_{I}^{(1,2,3)}$,
- $F_{B_{i}}^{(1,2,3)}$.


## 5. NUMERICAL RESULTS

The three-step fixed-point methods described in Section 4 were compared with the standard single-step BFGS method and with the unit-spaced three-step method M3. The experimental work was carried out on a set of 79 test functions. These test functions are taken from standard sets which are described in [12]. The dimensions of these test functions varied from 2 to 200. For these functions, we also chose the starting-point for each function as specified in [12]. However, it also seems to be desirable to explore other parts of the variable-space in addition to the standard point, in each case. Therefore, in our implementation, we chose a total of four different startingpoints for each test function, giving, in all, 316 test cases. Further details of the test functions used may be found in [13].
In order to analyse the efficiency of the new algorithms and particularly any tendency for the performance to improve or degrade as the dimension increases, the test problems have been
divided into four groups, corresponding to lowest ( $2 \leqslant n \leqslant 15$ ) dimension, low ( $16 \leqslant n \leqslant 45$ ) dimension, medium ( $46 \leqslant n \leqslant 80$ ) dimension, and high $(81 \leqslant n \leqslant 200)$ dimension.

In the actual implementation of these methods, we have employed a line search procedure. The specific procedure for the line search used safeguarded cubic interpolation to produce a point $\mathbf{x}_{i+1}$ which was accepted if the two standard conditions given below were satisfied by the new iterate $\mathbf{x}_{i+1}$ [14]

$$
\begin{align*}
f\left(\mathbf{x}_{i+1}\right) & \leqslant f\left(\mathbf{x}_{i}\right)+10^{-4} \mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right),  \tag{26}\\
\mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i+1}\right) & \geqslant \beta \mathbf{s}_{i}^{\top} \mathbf{g}\left(\mathbf{x}_{i}\right) . \tag{27}
\end{align*}
$$

Taking account of earlier work by Ford and Moghrabi [1] on the effect of safeguarded cubic interpolation on the performance of multistep methods, $\beta$ was set to 0.9.

All the algorithms and the BFGS method tested in these experiments employed the BFGS formula to update the inverse Hessian approximations $H_{i} \triangleq B_{i}^{-1}$, but (in the case of multistep methods) with the usual vectors $s_{i}$ and $y_{i}$ replaced by the forms of $r_{i}$ and $w_{i}$ (see (17) and (18)) appropriate to that algorithm

$$
\begin{equation*}
H_{i+1}=H_{i}+\left(1+\frac{\mathbf{w}_{i}^{\top} H_{i} \mathbf{w}_{i}}{\mathbf{r}_{i}^{\top} \mathbf{w}_{i}}\right) \frac{\mathbf{r}_{i} \mathbf{r}_{i}^{\top}}{\mathbf{r}_{i}^{\top} \mathbf{w}_{i}}-\left(\frac{H_{i} \mathbf{w}_{i} \mathbf{r}_{i}^{\top}+\mathbf{r}_{i} \mathbf{w}_{i}^{\top} H_{i}}{\mathbf{r}_{i}^{\top} \mathbf{w}_{i}}\right) \tag{28}
\end{equation*}
$$

Furthermore, in the case of test functions of dimension ten or greater, the initial approximation to the inverse Hessian was scaled by the method of Shanno and Phua [15] before the first update was performed.

The basic structure of the algorithms we have used in our experiments is as follows.
STEP 1. Set $H_{0}=I$ and $i=0$; evaluate $f\left(\mathbf{x}_{0}\right)$ and $\mathbf{g}\left(x_{0}\right)$.
Repeat
STEP 2. $\mathbf{p}_{i}=-H_{i} \mathbf{g}\left(\mathbf{x}_{i}\right) ;$
If $i<n$ and $\left\|p_{i}\right\|_{2}>1$, then $\mathbf{p}_{i}:=\mathbf{p}_{i} /\left\|\mathbf{p}_{i}\right\|_{2}$.
STEP 3. Compute $x_{i+1}$ which satisfies conditions (26) and (27), by means of a line-search from $\mathbf{x}_{i}$ along $p_{i}$, using safeguarded cubic interpolation.

STEP 4. If a one-step iteration is being executed, then set $\mathbf{r}_{i}=\mathbf{s}_{\boldsymbol{i}}$ and $\mathbf{w}_{i}=\mathbf{y}_{\boldsymbol{i}}$; else \{calculate $\mathbf{r}_{i}$ and $\mathbf{w}_{i}$;

If $\mathbf{r}_{i}^{\top} \mathbf{w}_{i} \leq 10^{-4}\left\|\mathbf{r}_{i}\right\|_{2}\left\|\mathbf{w}_{i}\right\|_{2}$
then set $\mathbf{r}_{i}=\mathbf{s}_{i}$ and $\mathbf{w}_{i}=\mathbf{y}_{i}$.\}
STEP 5. If $i=0$ and $n \geq 10$, then scale $H_{0}$ by the method of Shanno and Phua.
STEP 6. Update $H_{i}$ (by use of equation (28)) to produce $H_{i+1}$ satisfying $H_{i+1} \mathbf{w}_{i}=\mathbf{r}_{i}$; increment $i$.
Until $\left\|\mathbf{g}\left(\mathbf{x}_{i}\right)\right\|<\varepsilon$ (where $\varepsilon$ is a problem-dependent tolerance).
(Remark: with regard to Step 4, in order to preserve the positive-definite property of the matrices $H_{i}$ generated by multistep methods, it is necessary and sufficient that

$$
\mathbf{r}_{i}^{\top} \mathbf{w}_{i}>0
$$

by analogy with the condition employed in standard quasi-Newton updating (such as the BFGS formula, for example)

$$
\mathbf{s}_{i}^{\top} \mathbf{y}_{i}>0
$$

We guarantee this property by using the stronger condition

$$
\begin{equation*}
\mathbf{r}_{i}^{\top} \mathbf{w}_{i}>10^{-4}\left\|\mathbf{r}_{i}\right\|_{2}\left\|\mathbf{w}_{i}\right\|_{2} \tag{29}
\end{equation*}
$$

which ensures, at the same time, that we avoid a potential source of numerical instability by not permitting a small value of $\mathbf{r}_{i}^{\top} \mathbf{w}_{i}$ to be used in the updating of $H_{i}$-compare equation (28). Therefore, if condition (29) fails, we perform a single-step update.)

We provide summaries of the performance of the algorithms in Table 1. For each method, the total number of function/gradient evaluations required to solve all the problems in the given test set is stated, followed (two rows later) by the total number of iterations. The entries in each row labelled 'ratios' give the proportions of evaluations or iterations, respectively, each expressed as a percentage of the corresponding figure for the BFGS method. The asterisks attached to the results for $F_{I}^{(1,2,3)}$ in the "low" category indicate that these figures include the results for one failure for this method. The circumstances pertaining to this failure were that a close approach to the minimum was being achieved but the line-search failed before the gradient tolerance could be satisfied.

Table 1. Comparison of three-step methods and BFGS.

|  | BFGS | $M 3$ | $F_{I}^{(3)}$ | $F_{I}^{(1,2,3)}$ | $F_{B_{i}}^{(1,2,3)}$ | Problem Set |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| Evaluations | 3,263 | 4,292 | 4,505 | 3,624 | 3,638 |  |
| Ratios | $100.0 \%$ | $131.5 \%$ | $138.1 \%$ | $111.1 \%$ | $111.5 \%$ | Lowest |
| Iterations | 2,555 | 3,105 | 3,166 | 2,735 | 2,733 |  |
| Ratios | $100.0 \%$ | $121.5 \%$ | $123.9 \%$ | $107.0 \%$ | $107.0 \%$ |  |
| Evaluations | 17,682 | 19,182 | 17,800 | $15,282^{*}$ | 16,874 |  |
| Ratios | $100.0 \%$ | $108.5 \%$ | $100.7 \%$ | $86.4 \%^{*}$ | $95.4 \%$ | Low |
| Iterations | 15,262 | 14,918 | 14,212 | $12,587^{*}$ | 13,436 |  |
| Ratios | $100.0 \%$ | $97.7 \%$ | $93.1 \%$ | $82.5 \%^{*}$ | $88.0 \%$ |  |
| Evaluations | 20,761 | 20,266 | 17,927 | 17,294 | 16,946 |  |
| Ratios | $100.0 \%$ | $97.6 \%$ | $86.3 \%$ | $83.3 \%$ | $81.6 \%$ | Medium |
| Iterations | 18,727 | 16,563 | 15,037 | 14,834 | 14,243 |  |
| Ratios | $100.0 \%$ | $88.4 \%$ | $80.3 \%$ | $79.2 \%$ | $76.1 \%$ |  |
| Evaluations | 26,250 | 24,815 | 21,813 | 21,370 | 20,661 |  |
| Ratios | $100.0 \%$ | $94.5 \%$ | $83.1 \%$ | $81.4 \%$ | $78.7 \%$ | High |
| Iterations | 23,609 | 20,123 | 18,295 | 18,332 | 17,409 |  |
| Ratios | $100.0 \%$ | $85.2 \%$ | $77.5 \%$ | $77.6 \%$ | $73.7 \%$ |  |
| Evaluations | 67,956 | 68,555 | 62,045 | $57,570^{*}$ | 58,119 |  |
| Ratios | $100.0 \%$ | $100.9 \%$ | $91.3 \%$ | $84.7 \%^{*}$ | $85.5 \%$ | Combined |
| Iterations | 60,153 | 54,709 | 50,710 | $48,486^{*}$ | 47,823 |  |
| Ratios | $100.0 \%$ | $90.9 \%$ | $84.3 \%$ | $80.6 \% 0^{*}$ | $79.5 \%$ |  |

## 6. SUMMARY AND CONCLUSIONS

The concept of multistep quasi-Newton methods has been reviewed, and the derivation of certain three-step methods (based on approaches known to have been successful in the case of two-step methods) has been described. These methods have been compared experimentally with the standard single-step BFGS method and with the unit-spaced three-step method originally introduced in [1].

Consideration of the experimental results summarised in Table 1 leads to the following conclusions.

1. None of the three-step methods considered here are competitive with the BFGS method for problems of the lowest dimension. This is a feature which confirms again what has been
demonstrated on a number of occasions elsewhere [1-4] in respect of multistep methods. The feature may be considered to be of mainly theoretical interest since, from a practical point of view, such problems generally require very little computational effort for their solution.
2. As the problem dimension and the cost of solution rises, the three-step methods become much more competitive with the BFGS method and, for problem dimensions in the 'high' range, outperform it by as much as $20-25 \%$.
3. The unit-spaced method $M 3$ is a little more effective than the BFGS methods for the higher dimensions, although the improvement is seen more in respect of iterations than of evaluations. It does not appear to be competitive with the metric-based methods.
4. All three metric-based methods considered here equal or outperform the BFGS method for all but the lowest dimension problems, with the gap in performance increasing as the dimension rises.
5. If we give greatest weighting to the functions with higher dimensions, the best of the three metric-based methods (and therefore, of all the methods considered here) is evidently $F_{B_{i}}^{(1,2,3)}$. In particular, this method shows (for problems of the highest dimension) improvements of around $21 \%$ in evaluations and $26 \%$ in iterations. Since the extra computation required on each iteration (by comparison with the BFGS method) is only a modest multiple of the dimension, it may safely be concluded (and experimental work on other multistep methods in [2] confirms this) that commensurate reductions in the computational time may also be expected.
6. Previous work [13] has shown that the alternating two-step method F21 introduced in [4] is capable, on problems of high dimension, of improvements over the BFGS method of around $30 \%$ in evaluations and around $40 \%$ in iterations. Evidently, none of the threestep methods considered here has managed to achieve that level of performance. However, further research on the development and implementation of such methods, together with the use of recurrences to reduce the dependence on estimated quantities (see, for example, equations (24) and (25)), holds promise of additional improvements in the numerical performance.

## REFERENCES

1. J.A. Ford and I.A. Moghrabi, Multi-step quasi-Newton methods for optimization, J. Comput. Appl. Math. 50, 305-323, (1994).
2. J.A. Ford and I.A. Moghrabi, Alternative parameter choices for multi-step quasi-Newton methods, Optimization Methods and Software 2, 357-370, (1993).
3. J.A. Ford and I.A. Moghrabi, Minimum curvature multistep quasi-Newton methods, Computers Math. Applic. 31 (4/5), 179-186, (1996).
4. J.A. Ford and I.A. Moghrabi, Alternating multi-step quasi-Newton methods for unconstrained optimization, J. Comput. Appl. Math. 82, 105-116, (1997).
5. J.A. Ford, Implicit updates in multistep quasi-Newton methods, Computers Math. Applic. 42 (8/9), 10831091, (2001).
6. J.E. Dennis and R.B. Schnabel, Least change secant updates for quasi-Newton methods, SIAM Review 21, 443-459, (1979).
7. J.A. Ford and A.F. Saadallah, A rational function model for unconstrained optimization, Colloquia Mathematica Societatis János Bolyai 50, 539-563, (1986).
8. C.G. Broyden, The convergence of a class of double-rank minimization algorithms-Part 2: The new algorithm, J. Inst. Math. Applic. 6, 222-231, (1970).
9. R. Fletcher, A new approach to variable metric algorithms, Comput. J. 13, 317-322, (1970).
10. D. Goldfarb, A family of variable metric methods derived by variational means, Math. Comp. 24, 23-26, (1970).
11. D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comp. 24, 647-656, (1970).
12. J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software 7, 17-41, (1981).
13. S. Tharmlikit, Further investigation of multi-step quasi-Newton methods for unconstrained optimisation, Ph.D. Thesis, University of Essex, (2001).
14. R. Fletcher, Practical Methods of Optimization, Second Edition, Wiley, New York, (1987).
15. D.F. Shanno and K.H. Phua, Matrix conditioning and nonlinear optimization, Math. Prog. 14, 149-160, (1978).

[^0]:    The support provided by the Department of Computer Science, Faculty of Science, Burapha University, Thailand, the Ministry of Science, Technology and Environment (Thailand), and the Ministry of University Affairs (Thailand) is gratefully acknowledged by one of us (S.T.).

