RAIRO-Oper. Res. 53 (2019) 657–666
https://doi.org/10.1051/ro/2017043

RAIRO Operations Research
www.rairo-ro.org

RAPIDLY CONVERGENT STEFFENSEN-BASED METHODS FOR UNCONSTRAINED OPTIMIZATION

Mohammad Afzalinejad[1,*]

Abstract. A problem with rapidly convergent methods for unconstrained optimization like the Newton's method is the computational difficulties arising specially from the second derivative. In this paper, a class of methods for solving unconstrained optimization problems is proposed which implicitly applies approximations to derivatives. This class of methods is based on a modified Steffensen method for finding roots of a function and attempts to make a quadratic model for the function without using the second derivative. Two methods of this kind with non-expensive computations are proposed which just use first derivative of the function. Derivative-free versions of these methods are also suggested for the cases where the gradient formulas are not available or difficult to evaluate. The theory as well as numerical examinations confirm the rapid convergence of this class of methods.

Mathematics Subject Classification. 65K10, 90C53.

Received December 24, 2016. Accepted May 23, 2017.

1. Introduction

Consider the following problem

$$\min f(\mathbf{x}) \qquad \text{s.t. } \mathbf{x} \in R^{\mathbf{n}} \tag{1.1}$$

where $f : R^n \to R$ is a smooth function. If the derivatives of function f are available, there are plenty of methods to solve this problem. The Newton, trust region and conjugate gradients are among the most successful derivative-based methods for solving the above problem (for example see [2,5]). The Newton's method applies the following iteration formula

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k),$$

where $\mathbf{g}(\mathbf{x}_k)$ and $H(\mathbf{x}_k)$ are the gradient vector and the Hessian matrix of f at $\mathbf{x}_k$, respectively. This method has quadratic convergence rate if the starting point is sufficiently close to the optimal solution and the Hessian matrix is nonsingular at that solution. The conjugate gradient method does not need the second derivative. Instead, it has slower super-linear convergence. The trust region method is in fact a step length selection scheme and can be based on a linear or quadratic model. This method has global convergence properties but is slower than the Newton's method even using the quadratic model.

Evaluation of derivatives, especially the second derivative, is a major drawback for derivative-based methods because in many practical situations, derivative formulas are unavailable or difficult to evaluate. In such cases,

Keywords. Unconstrained optimization, derivative-free, newton's method, Steffensen's method.

[1] Department of Mathematics, Tafresh University, Tafresh, Iran.
*Corresponding author: afzalinejad@tafreshu.ac.ir

Article published by EDP Sciences

we can employ either the direct search methods (for a review see [4]) or the derivative–approximation techniques. An important class of direct search methods considers a grid of points and selects a point with the minimum function value and then applies finer grids to achieve adequate precision (for example see [1]). Other direct search methods build a search direction using local information of function value. Unfortunately, the current direct search methods are not capable of solving all practical problems efficiently. Their performance decreases quickly as the size of problem increases. In practice, the best derivative-free methods for solving problem (1.1) are ones that apply the approximation of derivatives. Furthermore, using derivative-approximation techniques makes it possible to exploit the available improvements in derivate-based methods (For example see [8]). A general technique for derivative approximation is finite-difference method. There is also a special technique for implicit approximation of the second derivative in optimization. This technique updates Hessian matrix using the rank-1 or 2 formulas. The quasi-Newton methods apply this technique to become free of the second derivative, but the convergence rates in these methods reduce to super-linear. In recent years, great efforts have been made to improve the convergence speed of quasi-Newton methods [3, 7, 9].

In this paper, a modified Steffensen method [6, 10, 11] is applied for approximating the Hessian matrix in Newton's method. The Steffensen method is an iterative approach for finding roots of a function. The use of Steffensen technique here leads to very accurate approximation of Hessian matrix especially near a stationary point and hence gives rapid convergence methods. Two methods of this kind are proposed. In the first method, the Steffensen formula is embedded in another Steffensen formula and the second method is a two-point method. In problems with a single variable, the first method has two functions evaluations per step and at least quadratic convergence while the second method has three functions evaluations per step and cubic convergence. In problems with several variables, the convergence of both methods are at least of second order. The strategy employed to avoid the second derivative can be used to obtain derivative-free versions of the methods. The efficiency of the proposed class of methods is examined on test functions for unconstrained optimization. The results show rapid convergence of the methods as expected from the theory.

The rest of the paper is organized as follows. Section 2 describes the new methods and Section 3 discusses some practical remarks. The results of numerical examinations are given in Section 4. Conclusions and further directions follow in Section 5.

2. A new class of methods for unconstrained optimization

In this section, two methods for solving unconstrained optimization problems are proposed which are based on the Steffensen method. Other methods of this kind can also be developed in a similar manner.

2.1. Method A: Several use of the Steffensen method

At first, a derivation of the method for functions of a single variable is presented and then the method is extended to the general n-variable case. Suppose that g is a real-valued function of a single variable and $g \in C^3$. Consider the following modified Steffensen method for solving $g(x) = 0$ [10].

$$x_{k+1} = x_k - \frac{\alpha_k g^2(x_k)}{g\left(x_k + \alpha_k g\left(x_k\right)\right) - g(x_k)} = x_k - \frac{g(x_k)}{\frac{g(x_k+\alpha_k g(x_k))-g(x_k)}{\alpha_k g(x_k)}} \qquad k = 0, 1, 2, \ldots \tag{2.1}$$

where $\alpha_k = 1$ reduces the above formula to the classical Steffensen's method. Determining an appropriate value for α_k can improve the asymptotic convergence rate of the method. Suppose that x^* is a simple root of g. The error $\epsilon_k = x_k - x^*$ in approximating x^* by x_k can be estimated as follows.

$$(x_{k+1} - x^*) = (x_k - x^*) - \frac{g\left((x_k - x^*) + x^*\right)}{\frac{g\left((x_k-x^*)+x^*+\alpha_k g((x_k-x^*)+x^*)\right)-g((x_k-x^*)+x^*)}{\alpha_k g((x_k-x^*)+x^*)}}$$

Therefore, noting that $g(x^*)=0$, we have

$$\epsilon_{k+1}=\epsilon_k-\frac{g'(x^*)\,\epsilon_k+\frac{1}{2}g''(x^*)\,\epsilon_k^2+O(\epsilon_k^3)}{\frac{g'(x^*+\epsilon_k)\alpha_k g(x^*+\epsilon_k)+\frac{1}{2}g''(x^*+\epsilon_k)\alpha_k^2 g^2(x^*+\epsilon_k)+O(\alpha_k^3 g^3(x^*+\epsilon_k))}{\alpha_k g(x^*+\epsilon_k)}}$$

Simplifying the above formula leads to the following error estimation (refer to [10], Thm. 2.1, for the details).

$$\epsilon_{k+1}=\frac{\frac{1}{2}g''(x^*)\,(1+\alpha_k g'(x^*))\epsilon_k^2+O(\epsilon_k^3)}{g'(x^*)+O(\epsilon_k)}$$

Obviously, if α_k is chosen as $\frac{-1}{g'(x^*)}$, the convergence rate is at least of third order. It can be shown that the local convergence rate is still of third order if α_k is chosen as $\frac{-1}{g'(x_k)}$. (See [10] for details regarding local convergence properties of the method described by (2.1)). If g' is adequately approximated (like what is done in the denominator of right term in (2.1)), a derivative-free method for root-finding is achieved. These findings suggest the following iteration formula for solving the problem $g(x)=0$.

$$\begin{aligned} x_{k+1}&=x_k-\frac{g(x_k)}{l_{k+1}} \qquad k=0,1,2,\ldots,\\ l_{k+1}&=-\frac{g\left(x_k-\frac{1}{l_k}g(x_k)\right)-g(x_k)}{\frac{1}{l_k}g(x_k)} \qquad k=0,1,2,\ldots, \end{aligned} \tag{2.2}$$

where l_0 is $g'(x_0)$ or an approximation for that. Note that, the above strategy for avoiding calculation of the derivatives leads to a reduction in local convergence rate. The number of function evaluations in each iteration of the method is just 2. Now assume that $g(x)=f'(x)$ and consider the above method employed for solving $\min\{f(x)|x\in R\}$. In this case, some considerations are needed to provide the descent property of the method. These considerations are presented in Section 3.

If the first derivative of f, *i.e.*, g is also unavailable, the approximation l_{k+1} for $g'(x)$ can be applied this time to$f'(x)$. For example, $g(x_k)$ can be estimated as follows:

$$g(x_k)\approx\frac{f\left(x_k-\frac{g(x_{k-1})}{l_k}\right)-f(x_k)}{-\frac{g(x_{k-1})}{l_k}}$$

where l_k is available from (2.2) and does not need to be calculated. Now, the following extension of the method is proposed to handle the general n-dimensional case. Suppose that $f:R^n\to R$ is a smooth function and $\mathbf{g}(\mathbf{x})=\nabla f(\mathbf{x})$. In this case, $\mathbf{g}(\mathbf{x})=0$ is a homogenous system of nonlinear equations. Let L_k denote an approximation for the Hessian matrix by the above technique. Therefore the iteration formula is

$$\mathbf{x}_{k+1}=\mathbf{x}_k-L_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) \qquad k=0,1,2,\ldots, \tag{2.3}$$

where L_0 is the Hessian matrix of f at $\mathbf{x}_0$ or an approximation for that. The entries of matrix L_{k+1} denoted by l_{ij}^{k+1}, can be expressed as

$$l_{ij}^{k+1}=-\frac{g_i\left(\mathbf{x}_k-\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j\mathbf{e}_j\right)-g_i(\mathbf{x}_k)}{\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j} \qquad i=1,\ldots,n, \qquad j=1,\ldots,n \tag{2.4}$$

In the above formula $\mathbf{g}$ is the gradient vector, g_i is the ith component of $\mathbf{g}$, $\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j$ is the jth component of $L_k^{-1}\mathbf{g}(\mathbf{x}_k)$ and $\mathbf{e}_j$ is the jth unit vector. The following theorem guarantees the quadratic convergence rate of the method.

Theorem 2.1. *Suppose that $f \in C^3$ and $\mathbf{x}_k$ is sufficiently close to $\mathbf{x}^*$ for some k's where $\mathbf{x}^*$ is a simple root of $\mathbf{g}$, i.e., $H(\mathbf{x}^*)$ is nonsingular. If L_k is nonsingular for each k and the sequence $\{\|L_k^{-1}\|$ is bounded then the iteration formula (2.3) converges to $\mathbf{x}^*$ at least at second order.*

Proof. Subtracting $\mathbf{x}^*$ from both sides of (2.3) gives

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - L_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) \qquad k = 0, 1, 2, \ldots$$

Using $\epsilon_k = \mathbf{x}_k - \mathbf{x}^*$ to denote the error vector in $\mathbf{x}_k$ we have

$$\begin{aligned} L_{k+1}\epsilon_{k+1} &= L_{k+1}\epsilon_k - \mathbf{g}(\mathbf{x}^*+\epsilon_k) \\ &= L_{k+1}\epsilon_k - \mathbf{g}(\mathbf{x}^*) - H(\mathbf{x}^*)\epsilon_k + \mathbf{O}\left(\|\epsilon_k\|^2\right) \\ &= (L_{k+1} - H(\mathbf{x}^*))\epsilon_k + \mathbf{O}\left(\|\epsilon_k\|^2\right) \end{aligned} \tag{2.5}$$

where $H(\mathbf{x}^*)$ is the Hessian matrix of f at $\mathbf{x}^*$. The element (ij) of matrix $(L_{k+1} - H(\mathbf{x}^*))$ is

$$(L_{k+1} - H(\mathbf{x}^*))_{ij} = \frac{g_i\left(\mathbf{x}_k - \left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j \mathbf{e}_j\right) - g_i(\mathbf{x}_k)}{-\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j} - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j}$$

Using Taylor theorem for g_i about $\mathbf{x} = \mathbf{x}_k$ we get

$$\begin{aligned} (L_{k+1} - H(\mathbf{x}^*))_{ij} &= \frac{g_i(\mathbf{x}_k) - \nabla g_i(\mathbf{x}_k)^t\left[\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j \mathbf{e}_j\right] + O\left(\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j^2\right) - g_i(\mathbf{x}_k)}{-\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j} - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} \\ &= \nabla g_i(\mathbf{x}_k)\mathbf{e}_j + O\left(\left|\left(L_k^{-1}\mathbf{g}(\mathbf{x}_k)\right)_j\right|\right) - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} \\ &= \frac{\partial g_i(\mathbf{x}^*+\epsilon_k)}{\partial x_j} + O\left(\|\mathbf{g}(\mathbf{x}_k)\|\right) - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} \\ &= O\left(\|\epsilon_k\|\right) + O\left(\|\mathbf{g}(\mathbf{x}^*+\epsilon_k)\|\right) \\ &= O\left(\|\epsilon_k\|\right) + O\left(\|\epsilon_k\|\right) \\ &= O\left(\|\epsilon_k\|\right) \end{aligned}$$

Therefore, from (2.5) we derive $L_{k+1}\epsilon_{k+1} = \mathbf{O}\left(\|\epsilon_k\|^2\right)$ or $\epsilon_{k+1} = \mathbf{O}(\|\epsilon_k\|^2)$. Assuming that $\mathbf{x}_k$ is sufficiently close to $\mathbf{x}^*$, this result also proves the convergence of the method. □

The proposed method is well-defined if the Hessian matrix $H(\mathbf{x}^*)$ is positive definite and satisfies a Lipschitz condition $\|H(\mathbf{x}) - H(\mathbf{y})\| \leqslant \lambda\|\mathbf{x} - \mathbf{y}\|$ in a neighborhood of $\mathbf{x}^*$. By continuity of $\mathbf{g}$, $\|\mathbf{g}(\mathbf{x})\|$ is small in a neighborhood of $\mathbf{x}^*$. Hence if $\mathbf{x}_k$ is sufficiently close to $\mathbf{x}^*$ and L_k is a good approximation for $H(\mathbf{x}_{k-1})$ then $\|L_k^{-1}\|$ is bounded and L_{k+1} would be a better approximation for the Hessian matrix. Note that, the error of using L_{k+1} as a difference approximation for $H(\mathbf{x}_k)$ is of $O\left(\|L_k^{-1}\mathbf{g}(\mathbf{x}_k)\|\right)$. This implies that $\epsilon_{k+1} \leqslant \epsilon_k$ and $\|\mathbf{g}(\mathbf{x}_{k+1})\| \leqslant \|\mathbf{g}(\mathbf{x}_k)\|$. By induction, we have $\epsilon_k \to \mathbf{0}$, $\mathbf{g}(\mathbf{x}_k) \to \mathbf{0}$ and $L_k \to H(\mathbf{x}^*)$ and therefore the method is well defined under these conditions.

If the gradient of function is also unavailable, the above method can easily be extended to a non-derivative method. It should be noted that in this case, the convergence order is generally lower. The above technique for derivative approximation (Eqs. (2.2) and (2.4)) can also be employed to build a model for the trust region methods.

2.2. Method B: A two-point Steffensen-based method

Another way of avoiding the computation of second derivatives is employing a two-point method [11]. At first, consider the single variable case and suppose that $f \in C^3$ is a real function of a single variable and $g(x) = f'(x)$. The following two-point method is proposed:

$$\begin{cases} x_{k+1} = x_k - \dfrac{g(x_k)(x_k - y_k)}{g(x_k) - g(y_k)} & k = 0, 1, \ldots \\ y_k = x_k - \dfrac{g^2(x_k)}{g(x_k + g(x_k)) - g(x_k)} & k = 0, 1, \ldots \end{cases} \tag{2.6}$$

where x_0 is the initial guess. The number of function evaluations in each iteration of the above method is 3. If x_k is sufficiently close to x^* for some k, the sequence has at least third order convergence. To show this, let $\epsilon_k = x_k - x^*$ and $\epsilon'_k = y_k - x^*$ be error terms in iteration k. First, note that the second equation in (2.6) is identical to equation (2.1) if α_k is fixed to 1 in (2.1). Therefore from the discussion in Section 2.1 we obtain

$$\epsilon'_k = \frac{\frac{1}{2}g''(x^*)(1 + g'(x^*))\epsilon_k^2 + O(\epsilon_k^3)}{g'(x^*) + O(\epsilon_k)}$$

and hence $\epsilon'_k = O(\epsilon_k^2)$. Second from the first equation in (2.6) we have

$$\begin{aligned} \epsilon_{k+1} &= \epsilon_k - \frac{g(x^* + \epsilon_k)(x_k - y_k)}{g'(y_k)(x_k - y_k) + \frac{1}{2}g''(y_k)(x_k - y_k)^2 + O\left((x_k - y_k)^3\right)} \\ &= \epsilon_k - \frac{g'(x^*)\epsilon_k + \frac{1}{2}g''(x^*)\epsilon_k^2 + O\left(\epsilon_k^3\right)}{g'(y_k) + \frac{1}{2}g''(y_k)(x_k - y_k) + O\left((x_k - y_k)^2\right)} \end{aligned} \tag{2.7}$$

The denominator in the last expression can be simplified as

$$\begin{aligned} &g'(x^* + \epsilon'_k) + \frac{1}{2}g''(x^* + \epsilon'_k)(\epsilon_k - \epsilon'_k) + O\left((\epsilon_k - \epsilon'_k)^2\right) \\ &= g'(x^*) + g''(x^*)\epsilon'_k + \frac{1}{2}g''(x^*)(\epsilon_k - \epsilon'_k) + O(\max\{\epsilon_k^2, \epsilon_k'^2\}) \end{aligned}$$

Replacing the above in (2.7) gives

$$\begin{aligned} \epsilon_{k+1} &= \frac{g'(x^*)\epsilon_k + g''(x^*)\epsilon_k\epsilon'_k + \frac{1}{2}g''(x^*)\left(\epsilon_k^2 - \epsilon_k\epsilon'_k\right) - g'(x^*)\epsilon_k - \frac{1}{2}g''(x^*)\epsilon_k^2 + O\left(\epsilon_k^3\right)}{g'(x^*) + O\left(\max\{\epsilon_k, \epsilon'_k\}\right)} \\ &= \frac{\frac{1}{2}g''(x^*)\epsilon_k\epsilon'_k + O\left(\epsilon_k^3\right)}{g'(x^*) + O\left(\max\{\epsilon_k, \epsilon'_k\}\right)} \end{aligned}$$

which proves that $\epsilon_{k+1} = O(\epsilon_k^3)$ because $\epsilon'_k = O(\epsilon_k^2)$.

Now, consider the case where $f : R^n \to R$ is a real function of n variables. Assume that f is a smooth function and $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$. The iteration formulas in (2.6) can be extended as follows.

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - B_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) & k = 0, 1, 2, \ldots, \\ \mathbf{y}_k = \mathbf{x}_k - W_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) & k = 0, 1, 2, \ldots, \end{cases} \tag{2.8}$$

where entries of matrices B_{k+1} and W_{k+1} denoted by b_{ij}^{k+1} and w_{ij}^{k+1}, respectively, can be expressed as follows.

$$b_{ij}^{k+1} = \frac{g_i\left(\mathbf{x}_k + (\mathbf{y}_k - \mathbf{x}_k)_j \mathbf{e}_j\right) - g_i(\mathbf{x}_k)}{(\mathbf{y}_k - \mathbf{x}_k)_j} \quad i = 1, \ldots, n, j = 1, \ldots, n$$
$$w_{ij}^{k+1} = \frac{g_i(\mathbf{x}_k + g_j(\mathbf{x}_k)\mathbf{e}_j) - g_i(\mathbf{x}_k)}{g_j(\mathbf{x}_k)} \quad i = 1, \ldots, n, j = 1, \ldots, n \tag{2.9}$$

Note that $(\mathbf{y}_k - \mathbf{x}_k)_j$ denotes jth component of vector $\mathbf{y}_k - \mathbf{x}_k$ and $\mathbf{e}_j$ is the jth unit vector. The following theorem shows that the convergence of $\{\mathbf{x}_k\}$ is at least of second order.

Theorem 2.2. *Suppose that $f \in C^3$ and $\mathbf{x}_k$ is sufficiently close to $\mathbf{x}^*$ for some k's where $\mathbf{x}^*$ is a simple root of $\mathbf{g}$. If B_k and W_k are nonsingular for each k and the sequences $\{\|B_k^{-1}\|\}$ and $\{\|W_k^{-1}\|\}$ are bounded then the sequence $\{\mathbf{x}_k\}$ generated by (2.8) converges to $\mathbf{x}^*$ at least at a quadratic rate.*

Proof. Subtracting $\mathbf{x}^*$ from both sides of the first system of equations in (2.8) gives

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* - B_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) \qquad k = 0, 1, 2, \ldots$$

Using $\epsilon_k = \mathbf{x}_k - \mathbf{x}^*$ to denote the error vector in $\mathbf{x}_k$ we have

$$\begin{aligned} B_{k+1}\epsilon_{k+1} &= B_{k+1}\epsilon_k - \mathbf{g}(\mathbf{x}^* + \epsilon_k) \\ &= B_{k+1}\epsilon_k - \mathbf{g}(\mathbf{x}^*) - H(\mathbf{x}^*)\epsilon_k + \mathbf{O}\left(\|\epsilon_k\|^2\right) \\ &= (B_{k+1} - H(\mathbf{x}^*))\epsilon_k + \mathbf{O}\left(\|\epsilon_k\|^2\right) \end{aligned} \tag{2.10}$$

The element (ij) of matrix $(B_{k+1} - H(\mathbf{x}^*))$ is as follows.

$$\begin{aligned} (B_{k+1} - H(\mathbf{x}^*))_{ij} &= \frac{g_i\left(\mathbf{x}_k + (\mathbf{y}_k - \mathbf{x}_k)_j \mathbf{e}_j\right) - g_i(\mathbf{x}_k)}{(\mathbf{y}_k - \mathbf{x}_k)_j} - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} \\ &= \frac{(\mathbf{y}_k - \mathbf{x}_k)_j \nabla g_i(\mathbf{x}_k)^t \mathbf{e}_j + O((\mathbf{y}_k - \mathbf{x}_k)_j^2)}{(\mathbf{y}_k - \mathbf{x}_k)_j} - \frac{\partial g_i(\mathbf{x}^*)}{\partial x_j} \\ &= O(\|\epsilon_k\|) + O((\mathbf{y}_k - \mathbf{x}_k)_j) \end{aligned}$$

Form the second system of equations in (2.8) we have

$$\mathbf{y}_k - \mathbf{x}_k = -W_{k+1}^{-1}\mathbf{g}(\mathbf{x}_k) = -W_{k+1}^{-1}\mathbf{g}(\mathbf{x}^* + \epsilon_k) = -W_{k+1}^{-1}\left(H(\mathbf{x}^*)\epsilon_k + \mathbf{O}\left(\|\epsilon_k\|^2\right)\right).$$

Therefore $B_{k+1} - H(\mathbf{x}^*) = O(\|\epsilon_k\|)$ and hence from (2.10) $B_{k+1}\epsilon_{k+1} = \mathbf{O}(\|\epsilon_k\|^2)$, which means $\epsilon_{k+1} = \mathbf{O}(\|\epsilon_k\|^2)$. □

The conditions under which matrices B_k and W_k are nonsingular and the above method is well defined, are similar to method A.

3. Some practical remarks

Every numerical approach requires some considerations to be successful in practice. The methods discussed in Section 2 are variants of Newton's method and therefore the conditions for the new methods are quite similar to Newton's method. In the following, we suggest some important remarks and modifications.

(1) Since the components of $\mathbf{g}(\mathbf{x}_k)$ could be large where $\mathbf{x_k}$ is distant from the solution $\mathbf{x}^*$, the derivatives approximation in formulas (2.2), (2.4), (2.6) and (2.9) could be inexact. Using an appropriate multiplier for $\mathbf{g}(\mathbf{x}_k)$ in these formulas is suggested. For example, a sequence of constants ϑ_k's such that $0 < \vartheta_k < 1$ and $\vartheta_k \to 1$ can be chosen. Another possible strategy here is using a global-convergent method like the steepest descent method at the first stage to reach the vicinity of $\mathbf{x}^*$ and then performing one of the proposed rapid convergent methods.
(2) To have symmetry for matrices L_k, B_k and W_k we can add them to their transposes and divide the result by 2. For example, the transformation for L_k is $L_k^{new} = \frac{1}{2}(L_k + L_k^t)$. Another strategy is to avoid calculating upper (or lower) diagonal elements to reduce the computational efforts.
(3) Applying the Cholesky factorization for L_{k+1} in (2.3) can be advantageous. Firstly, the Cholesky factorization can simplify solving $L_k\mathbf{y} = \mathbf{g}(\mathbf{x}_k)$ for $\mathbf{y}$. Secondly, the positive definiteness of the matrix L_k can be checked. If L_k is positive definite, the descent property and therefore the convergence of the method to a minimum point is guaranteed. If it is found that L_k is not positive definite, we can modify L_k (for example, by adding positive values to the diagonal elements) to make it positive definite. This also guarantees the non-singularity of L_k. The same strategy can be adopted for B_{k+1} and W_{k+1} in (2.8).
(4) Performing a line search in each step is another modification. This can add the global convergence property to the proposed methods.

4. Numerical examination

To test the efficiency of the proposed methods, some of well-known test functions for unconstrained optimization are used. These functions have various behaviors like highly oscillating, flattening far from minimum, having badly conditioned Hessian matrix and so on. No line search or trust region approaches are used in the employed algorithms. The termination criterion $\|\mathbf{g}(\mathbf{x}_k)\| \leqslant 10^{-7}$ is considered and if the method is derivative-free, an approximation of $\mathbf{g}(\mathbf{x}_k)$ is applied instead. The list of selected test functions is as follows.

(1) $f(x) = \cos(x) + (x-2)^2$,
(2) $f(x) = x^4 - 8.5x^3 - 31.0625x^2 - 7.5x + 45$,
(3) $f(x) = ((x+2)^2)(x+4)(x+5)(x+8)(x-16)$,
(4) $f(x) = e^x - 3x^2$,
(5) $f(x_1, x_2) = x_1^4 + x_1x_2 + (1+x_2)^2$,
(6) Freudenstein & Roth's function:

$$f(x_1, x_2) = \frac{1}{2\left((x_1 - x_2(2 - x_2(5 - x_2)) - 13)^2 + (x_1 - x_2(14 - x_2(1 + x_2)) - 29)^2\right)}$$

(7) Levy's function:

$$f(x_1x_2) = \sin\left(\pi\left(1 + \frac{x_1 - 1}{4}\right)\right) + \left(\left(\left(1 + \frac{x_1 - 1}{4}\right) - 1\right)^2 \left(1 + 10\left(\pi\left(1 + \frac{x_1 - 1}{4}\right) + 1\right)^2\right)\right)$$

$$+ \left(\left(\left(1 + \frac{x_2 - 1}{4}\right) - 1\right)^2 \left(1 + 10\left(\sin\left(2\pi\left(1 + \frac{x_2 - 1}{4}\right)\right)\right)^2\right)\right)$$

(8) Shubert's function:

$$f(x_1, x_2) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$$

(9) Rastrigin's function:

$$f(x_1,\ldots,x_n)=10n+\sum_{i=1}^{n}\left(x_i^2-10\cos\left(2\pi x_i\right)\right)$$

(10) Schwefel's function:

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{|x_i|}\right)$$

(11) Zakharov's function:

$$f(x_1,x_2,x_3,x_4,x_5)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}\frac{i}{2}x_i^2\right)^2+\left(\sum_{i=1}^{n}\frac{i}{2}x_i^2\right)^4$$

(12) Rosenbrock's function:

$f(x_1,\ldots,x_n)=\sum_{i=1}^{n-1}\left(100\left(x_i^2-x_{i+1}\right)^2+\left(x_i-1\right)^2\right)$

(13) Griewangk's function:

$f(x_1,\ldots,x_n)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$

(14) Easom's function:

$f(x_1,\ldots,x_{2k})=\left(-\prod_{i=1}^{2k}\cos\left(x_i\right)\right)\exp\left(\sum_{i=1}^{2k}-\left(x_i-\pi\right)^2\right),k\epsilon N$

(15) Sum of different powers:

$f(x_1,\ldots,x_n)=\sum_{i=1}^{n}|x_i|^{i+1}$

(16) Langermann's function: $f(x_1x_2)=\sum_{i=1}^{m}c_i\exp\left(-\frac{(x_1-a_i)^2}{\pi}-\frac{(x_2-b_i)^2}{\pi}\right)\cos(\pi\left(x_1-a_i\right)^2+\pi\left(x_2-b_i\right)^2)$

(m,a_i,b_i and c_i are arbitrary constants.)

(17) Drop wave function:

$f(x_1,x_2)=-\frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{0.5\left(x_1^2+x_2^2\right)+2}$

(18) Trid's function:

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{n}\left(x_i-1\right)^2-\sum_{i=2}^{n}x_ix_{i-1}$$

(19) Goldstein−Price's function:

$$\begin{aligned}f(x_1,x_2)=&\left(1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right)\\&\times\left(30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right)\end{aligned}$$

(20) Styblinski−Tang's function:

$$f(x_1,\ldots,x_n)=\frac{1}{2}\sum_{i=1}^{n}\left(x_i^4-16x_i^2+5x_i\right).$$

First, we employed formulas (2.3) and (2.4) and the exact values of first derivative. To study the theoretical results, none of the remarks stated in Section 3 are applied. Table 1 demonstrates the results for 10 test functions. In this table, n is the size of the problem and #Itr denotes the number of iterations required by the method to satisfy the termination criterion. Error values for the sequence generated by the algorithm as well as the values of gradients have been reported in this table. As can be seen from the results, the reduction in error values are very rapid and the convergence is at least of second order in problems that satisfy the conditions of Theorem 1. The number of required iterations are small except for Rosenbrock's function which has badly-conditioned second derivative near to its minimum.

Second, both the first and second derivatives are approximated in the algorithm. A controlling multiplier ϑ_k for $\mathbf{g}(\mathbf{x}_k)$ is considered as noted in remark (1) of Section 3. The matrix L_k is also modified in some problems by adding constant positive values to the diagonal entries (see remark (3) of Sect. 3). This helps the resulted matrix to be invertible and possibly positive definite. Note that, this action reduces the convergence speed.

TABLE 1. Analysis of the algorithm (second-derivative-free version): Error values of the first 9 terms in the sequence generated by the algorithm.

Problem	n	$\mathbf{x}^*$	$\mathbf{x}_0$	$\|\epsilon_1\|$, $\|g_1\|$	$\|\epsilon_2\|$, $\|g_2\|$	$\|\epsilon_3\|$, $\|g_3\|$	$\|\epsilon_4\|$, $\|g_4\|$	$\|\epsilon_5\|$, $\|g_5\|$	$\|\epsilon_6\|$, $\|g_6\|$	$\|\epsilon_7\|$, $\|g_7\|$	$\|\epsilon_8\|$, $\|g_8\|$	$\|\epsilon_9\|$, $\|g_9\|$	#Itr
#5	2	(0.6958843861, -1.347942193)	(1,-1)	0.0321944, 0.159955	0.0111534, 0.0538179	0.000484383, 0.00230256	7.51206×10^{-6}, 0.0000356851	5.1571×10^{-9}, 2.42966×10^{-8}	-	-	-	-	5
Rastrigin's function	10	(0,0,...,0)	(0.2,0.2,..., 0.2)	0.123835, 48.6439	0.0301737, 11.9653	0.000230256, 0.091362	1.36249×10^{-7}, 0.0000540614	4.72621×10^{-15}, 1.87529×10^{-12}	-	-	-	-	5
Schwefel's function	10	(420.9687,..., 420.9687)	(400,400,..., 400)	0.422305, 0.106604	0.0254966, 0.00639753	0.00144606, 5.04117×10^{-7}	0.000146603, 2.27081×10^{-12}	-	-	-	-	-	4
Zakharov's function	10	(0,0,...,0)	(0.4,0.4,..., 0.4)	0.634749, 30219.0	0.418243, 8726.48	0.301772, 3327.69	0.212644, 1201.19	0.149217, 441.148	0.102879, 163.175	0.0685053, 61.3133	0.0422325, 23.3852	0.0216189, 8.63298	13
Rosenbrock's function	2	(1,1)	(1.5,1.5)	1.33571, 0.993378	0.816116, 16.4178	0.831611, 0.960284	0.508553, 7.43087	0.511243, 0.624851	0.311909, 3.25938	0.308358, 0.394259	0.186788, 1.38817	0.180593, 0.239557	23
Griewangk's function	10	(0,0,...,0)	(0.5,0.5,..., 0.5)	0.346732, 0.107075	0.0330411, 0.0119372	0.0000252688, 4.98377×10^{-6}	8.70521×10^{-11}, 3.23695×10^{-11}	-	-	-	-	-	4
Drop wave function	2	(0,0)	(0.5,0.5)	0.0274758, 0.0335043	0.00428207, 0.00522407	7.39041×10^{-7}, 9.0163×10^{-7}	-	-	-	-	-	-	3
Trid's function	6	(6,10,12,12,10,6)	(1,1,1,1,1,1)	3.55271×10^{-15}, 3.55271×10^{-15}	-	-	-	-	-	-	-	-	1
Goldstein-Price's function	2	(0,-1)	(0.5,-0.5)	0.741451, 1895.26	0.147723, 52.4029	0.135532, 60.6227	0.0206374, 20.368	0.00606536, 3.52843	0.000125444, 0.113697	6.665×10^{-7}, 0.000643194	1.33016×10^{-10}, 1.28381×10^{-7}	-	8
Styblinski-Tang's function	10	(-2.903534,..., -2.903534)	(-4,-4,...,-4)	0.367154, 13.4498	0.136823, 4.83541	0.00753513, 0.260898	0.000161452, 0.00558061	2.81327×10^{-7}, 6.69206×10^{-6}	8.78251×10^{-8}, 1.72003×10^{-10}	-	-	-	6

TABLE 2. The numerical results of derivative-free version of the proposed method.

Problem	n	A local minimum	Optimal Value	x_0	#Itr	x_k	$f(x_k)$
#1	1	2.354243	-0.580237	2	4	2.35423	-0.580237
#2	1	8.278462	-2271.58	8	5	8.27846	-2271.58
#3	1	12.679120	0	12	5	12.5791	-4.3633×10^{-6}
#4	1	2.833148	-7.08129	12.6	4	2.83315	-7.08129
#5	2	(0.695884, -1.347942)	-0.582445	(0.75,-1.25)	2	(0.695899,-1.34795)	-0.582445
Freudenstein & Roth	2	(11.4128,-0.896805)	11.4921	(12,-0.5)	10	(11.4127, -0.896807)	11.4921
Levy	2	(1.03568,1)	-0.014107	(0.75,0.75)	9	(1.03568, 1)	-0.014107
Shubert	2	(0.821528,0.341121)	-0.143767	(1,0.5)	10	(0.821534, 0.341131)	-0.143767
Rastrigin	2	(0,0)	0	(0.2,0.2)	7	(-1.35467×10^{-8}, 0.42966×10^{-7})	4.0927×10^{-12}
Rastrigin	10	(0,0,...,0)	0	(0.2, ... ,0.2)	7	(-6×10^{-7}, -6×10^{-7}, -5×10^{-7}, -5×10^{-7},-5×10^{-7}, 5×10^{-7},5×10^{-7}, 7×10^{-7}, -1×10^{-7}, -1×10^{-7})	4.8586×10^{-10}
Schwefel	2	(420.9687, 420.9687)	-841.9158	(400,400)	4	(420.969, 420.969)	-841.916
Schwefel	10	(420.9687, ..., 420.9687)	-4189.829	(400, ... ,400)	5	(420.969, ..., 420.969)	-4189.83
Zakharov	5	(0,0,0,0,0)	0	(0.25,... ,0.25)	10	(1.9×10^{-8}, 1.2×10^{-8}, 4.7×10^{-9}, -2.3×10^{-9}, -9.3×10^{-9})	5.9654×10^{-16}
Rosenbrock	5	(1,1,1,1,1)	0	(0.75, ... ,0.75)	37	(1,1,1,1,1)	2.3220×10^{-17}
Griewangk	2	(0,0)	0	(0.5,-0.5)	4	(-2.75201×10^{-8}, 1.25965×10^{-7})	4.3299×10^{-15}
Easom	4	(π, π, π, π)	-1	(2.5,2.5,2.5,2.5)	10	(3.14159, ..., 3.14159)	-1
Some of different power	2	(0,0)	0	(0.5,-0.5)	10	(4.99071×10^{-8}, -0.0170621)	4.9670×10^{-6}
Langermann	2	(1.01572,1.00518)	-3.77964	(1,1)	7	(1.01572, 1.00518)	-3.77964
Drop wave	2	(0.367848,0.367848)	-0.936245	(1,1)	7	(0.367859, 0.367828)	-0.936245
Trid	6	(6,10,12,12,10,6)	-50	(1,1,1,1,1,1)	8	(6,10,12,12,10,6)	-50
Goldstein-Price	2	(0,-1)	3	(0.5,-0.5)	15	(-3.59803×10^{-8},-1)	3
Styblinski-Tang	10	(-2.903534,..., -2.903534)	-391.6617	(-4,-4,...,-4)	6	(-2.90353,...,-2.90353)	-391.662

Table 2 summarizes the results. The number of iterations are small in most cases but they are slightly more than the number of iterations required by the second-derivative-free version of the algorithm that applies exact first derivatives. In some test functions like problems #5 and #11, however, the convergence speed in derivative-free method is observed to be more than the second-derivative-free method by the help of strategies stated in Section 3.

5. Conclusion

In this paper, a class of Steffensen-based methods for solving unconstrained optimization problems has been proposed. Two methods of this class have been introduced. The first method has 2 function evaluations per step and at least quadratic convergence for functions of a single variable. This method has also quadratic convergence for functions of several variables. The second method is a two-point method and has 3 function evaluations per step and third order convergence for problems with a single variable. The convergence rate of this method for solving problems with several variables is at least quadratic. Both methods are free from second derivatives. The strategy employed in these methods is similar to Newton method for unconstrained optimization but derivatives of functions f are approximated adequately using a rapid Steffensen technique. This Steffensen technique can be applied to approximate the first derivatives as well to obtain derivative-free methods. The theoretical and computational results show that the new methods have rapid convergence without performing expensive computations.

Based on the proposed approximation technique, modifications for other optimization methods like the conjugate gradient and the trust region methods can be considered for further research.

Acknowledgements. The author would like to thank the anonymous referee for valuable comments and suggestions which are very helpful in improving the paper.

References

[1] A. Burmen and T. Tuma, Unconstrained derivative-free optimization by successive approximation. *J. Comput. Appl. Math.* **223** (2009) 62–74.
[2] R. Fletcher, Practical Methods of Optimization, 2nd ed. Wiley (2000).
[3] E. Kahya, Modified Secant-type methods for unconstrained optimization. *Appl. Math. Comput* **181** (2006) 1349–1356.
[4] T.G. Kolda, R.M. Lewis and V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review* **45** (2003) 385–482.
[5] J. Nocedal and S. Wright, Numerical Optimization. Springer Verlag, New York (1999).
[6] J.M. Ortega and W.C. Rheinboldt, Iterative Solutions of Nonlinear Equations in Several Variables, Academic Press (1970).
[7] H. Ren and Q. Wu, A class of modified Secant methods for unconstrained optimization. *App. Math. Comput.* **206** (2008) 716–720.
[8] C. Sainvitu, How much do approximate derivatives hurt filter methods? *RAIRO: OR* **43** (2009) 309–329.
[9] Z. Weia, G. Lia and L. Qib, New quasi-Newton methods for unconstrained optimization problems. *Appl. Math. Comput.* **175** (2006) 1156–1188.
[10] X. Wu and J. Xia, Some substantial modifications and improvements for derivative-free iterative methods and derivative-free transformation for multiple zeros. *Appl. Math. Comput.* **181** (2006) 1585–1599.
[11] Q. Zheng, P. Zhao, L. Zhang and W. Ma, Variants of Steffensen-secant method and applications. *Appl. Math. Comput.* **216** (2010) 3486–3496.