Mathematics and Statistics 11(1): 117-127, 2023 http://www.hrpub.org
DOI: 10.13189/ms.2023.110113

# Two New Preconditioned Conjugate Gradient Methods for Minimization Problems

**Hussein Ageel Khatab[*], Salah Gazi Shareef**

Department of Mathematics, Faculty of Science, University of Zakho, Iraq

*Received September 12, 2022; Revised December 9, 2022; Accepted December 29, 2022*

***Cite This Paper in the Following Citation Styles***

***(a):*** *[1] Hussein Ageel Khatab, Salah Gazi Shareef , "Two New Preconditioned Conjugate Gradient Methods for Minimization Problems," Mathematics and Statistics, Vol. 11, No. 1, pp. 117 - 127, 2023. DOI: 10.13189/ms.2023.110113.*

***(b):*** *Hussein Ageel Khatab, Salah Gazi Shareef (2023). Two New Preconditioned Conjugate Gradient Methods for Minimization Problems. Mathematics and Statistics, 11(1), 117 - 127. DOI: 10.13189/ms.2023.110113.*



**Abstract** In application to general function, each of the conjugate gradient and Quasi-Newton methods has particular advantages and disadvantages. Conjugate gradient (CG) techniques are a class of unconstrained optimization algorithms with strong local and global convergence qualities and minimal memory needs. Quasi-Newton methods are reliable and efficient on a wide range of problems and they converge faster than the conjugate gradient method and require fewer function evaluations but they have the disadvantage of requiring substantially more storage and if the problem is ill-conditioned, they may take several iterations. A new class has been developed, termed preconditioned conjugate gradient (PCG) method. It is a method that combines two methods, conjugate gradient and Quasi-Newton. In this work, two new preconditioned conjugate gradient algorithms are proposed namely New PCG1 and New PCG2 to solve nonlinear unconstrained optimization problems. A new PCG1 combines conjugate gradient method Hestenes-Stiefel (HS) with new self-scaling symmetric Rank one (SR1), and a new PCG2 combines conjugate gradient method Hestenes-Stiefel (HS) with new self-scaling Davidon, Flecher and Powell (DFP). The algorithm uses the strong Wolfe line search condition. Numerical comparisons with standard preconditioned conjugate gradient algorithms show that for these new algorithms, computational scheme outperforms the preconditioned conjugate gradient.

**Keywords** Unconstrained Optimization, Quasi-Newton Method, Self-Scaling Quasi-Newton Method, Preconditioned Conjugate Gradient Method

## 1. Introduction

We are interested to consider the following nonlinear unconstrained minimization problem

$$\text{Min} f(x) \ , \ x \in R^n \tag{1}$$

where $f: R^n \to R$ is a twice continuously differentiable. Nonlinear unconstrained conjugate gradient algorithms to solve (1) are iterative technique and have the form

$$x_{m+1} = x_m + \alpha_m d_m \tag{2}$$

which is starting from an initial guess $x_0 \in R^n$, the positive step length $\alpha_m$ is calculated by one dimensional line search and $d_m$ is the search direction. The preconditioned conjugate gradient techniques have two search directions, the first one has the form:

$$d_0 = -H_0 g_0,\ H_0 = I \text{ and } g_0 = \nabla f(x_0). \tag{3}$$

and the other search direction has the form:

$$d_{m+1} = -H_{m+1}g_{m+1} + \beta_m d_m \quad (4)$$

where $g_{m+1} = \nabla f(x_{m+1})$, $\beta_m$ is a conjugate coefficient. The most well-know formulas of $\beta_m$ are called Hestenes Stiefel (HS) [1] , Polak Ribiere Polyak (PRP) [3], Dai Yuan (DY)[4] and Fletcher Reeves(FR)[5].

$H_{m+1}$ is the inverse Hessian approximation and $H_{m+1}$ is chosen to satisfy the following quasi-Newton equation

$$H_{m+1}y_m = v_m \quad (5)$$

where $y_m = g_{m+1} - g_m$ and $v_m = x_{m+1} - x_m$ . The Quasi-Newton is very similar to method of Newton, but avoids the requirement to compute Hessian matrices by repeating from iteration to iteration.

We obtain the corresponding update formula for the inverse Hessian approximation[2]

$$H_{m+1} = H_m + \frac{(v_m - H_m y_m)(v_m - H_m y_m)^T}{{y_m}^T(v_m - H_m y_m)} \quad (6)$$

And also there are other methods to find the inverse Hessian approximation like Davidon, Fletcher and Powell (DFP)

$$H_{m+1} = H_m + \frac{v_m v_m^T}{v_m^T y_m} - \frac{(\ H_m y_m)(\ H_m y_m)^T}{y_m^T\, H_m y_m} \quad (7)$$

and Broyden, Fletcher, Goldfarb and Shanno Method (BFGS)

$$H_{m+1} = H_m + (1 + \frac{y_m^T H_m y_m}{y_m^T v_m})\frac{v_m v_m^T}{v_m^T y_m} - \frac{(\ H_m y_m v_m^T) + (\ H_m y_m v_m)^T}{y_m^T\, y_m} \quad (8)$$

Self-scaling is one of the common approaches in the modification of the Quasi-Newton method. The general strategy of self-scaling Quasi-Newton method (SS) is to scale the Hessian approximation matrix before it is updated at each iteration. This is to avoid large differences in the eigenvalues of the approximated Hessian of the function. There are many self-scalings suggested, for example Oren [6] proposed some good characteristics, with a self-scaling parameter $\rho_m$, this class of updates can be written as

$$H_{m+1} = (H_m - \frac{H_m y_m y_m^T H_m}{y_m^T H_m y_m} + \emptyset_m(y_m^T H_m y_m)\bar{v}_m \bar{v}_m^T)\rho_m + \frac{v_m^T v_m}{v_m^T y_m}\ , \quad (9)$$

$$\text{with } \bar{v}_m = \frac{v_m}{v_m^T y_m} + \frac{H_m y_m}{y_m^T H_m y_m}$$

(Oren and Luenberger )[7] suggest to use the self - adaptable values for the parameter

$$\rho_m = t\frac{g_m^T v_m}{g_m^T H_m y_m} + (1-t)\frac{v_m^T y_m}{y_m^T H_m y_m}$$

and usually the value $t = 0$ is recommended for the updates in the convex class.

To improve the performance of Quasi-Newton update, Biggs [8] proposed to choose $H_{m+1}$ to satisfy the following modified equation

$H_{m+1}y_m = \gamma_m v_m$. where $\gamma_m > 0$ is a scaling parameter. Biggs showed that a modified BFGS could be derived as follows

$$H_{m+1} = H_m + \frac{H_m y_m v_m^T + v_m y_m^T H_m}{v_m^T y_m} + (\frac{1}{\tau_m} + \frac{y_m^T H_m y_m}{v_m^T y_m})\frac{v_m v_m^T}{v_m^T y_m} \quad (10)$$

where $\tau_m = \frac{1}{\gamma_m} = \frac{6}{v_m^T y_m}(f(x_m) - f(x_{m+1}) + v_m^T g_{m+1}) - 2$

also, Yang, Xu and Gao[9] made a little modification for self-scaling symmetric rank one update with Davidon's optimal condition [10] as follows

$$H_{m+1} = r_m H_m + \frac{(v_m - \gamma_m H_m \hat{y}_m)(v_m - \gamma_m H_m \hat{y}_m)^T}{{\hat{y}_m}^T(v_m - \gamma_m H_m \hat{y}_m)} \quad (11)$$

where $r_m$ is the scaling factor,

$$\hat{y}_m = \left(\frac{1+\theta_m}{v_m^T y_m}\right)y_m,\ \theta_m = 6(f_m - f_{m+1}) + 3(g_m - g_{m+1})^T v_m \text{ and } f_m = f(x_m)\ .$$

The structure of the paper is as follows. In section 2, we introduce two new self-scaling Quasi-Newton methods (new self-scaling symmetric rank one and new self-scaling Davidon, Flecher and Powell). In section 3, we present the outlines of the New PCG1 and New PCG2. In section 4, we will prove the Quasi-Newton condition and positive definite condition for the new self-scaling symmetric rank one and new self-scaling Davidon, Flecher and Powell. We give some numerical results to show the efficiency of these new algorithms in section 5.

## 2. Two New Self-Scaling Quasi-Newton Methods

In this section, we will derive two new self-scaling variable metrics for unconstrained minimization problems, the first one is new self-scaling symmetric rank one and the second one is new self-scaling DFP.
Consider the equation

$$y_m^* = y_m + \mu_m v_m \tag{12}$$

Suppose that $\mu_m = \tau * \frac{N_m}{1+N_m}$,

$$N_m = \frac{y_m^T v_m}{\|v_m\|} \text{ and } \tau \in (0,1)$$

The updating $H_{m+1}$ matrix is chosen to satisfy the following Quasi-Newton equation

$$H_{m+1} y_m^* = \delta_1 v_m \tag{13}$$

where $\delta_1 = \sqrt{\left|\frac{y_m^T H_{m+1} y_m^*}{y_m^T v_m}\right|}$

Let

$$H_{m+1} = H_m + \theta_m w_m w_m^T \tag{14}$$

Where the correction term $\theta_m w_m w_m^T$ is symmetric, $\alpha_m$ is a scalar belong to $R$ and $w_m$ is a vector belong to $R^n$ .
Give $H_m$, $y_k^*$ and $v_m$ so that the required relationship $H_{m+1} y_m^* = \delta_1 v_m$ is satisfied.
To find $\theta_m$ *and* $w_m$ , multiply both sides of equation(14) by $y_m^*$ and by using (13), we have

$$H_{m+1} y_m^* = (H_m + \theta_m w_m w_m^T) y_m^* = \delta_1 v_m \tag{15}$$

first note that $w_m^T y_m^*$ is a scalar. Thus

$$\delta_1 v_m - H_m y_m^* = (\theta_m w_m^T y_m^*) w_m \tag{16}$$

Here,

$$w_m = \frac{\delta_1 v_m - H_m y_m^*}{\theta_m (w_m^T y_m^*)}$$

Hence,

$$\theta_m w_m w_m^T = \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{\theta_m (w_m^T y_m)^2} \tag{17}$$

So,

$$H_{m+1} = H_m + \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{\theta_m (w_m^T y_m^*)^2} \tag{18}$$

Now multiply (16) by ${y_m^*}^T$ to obtain

$$\delta_1 {y_m^*}^T v_m - {y_m^*}^T H_m y_m^* = {y_m^*}^T (\theta_m w_m^T y_m^*) w_m \tag{19}$$

Observe that $\theta_m$ is scalar and
${y_m^*}^T w_m = {w_m}^T y_m^*$ , then the above equation becomes

$$\delta_1 {y_m^*}^T v_m - {y_m^*}^T H_m y_k^* = \theta_m (w_m^T y_m^*)^2 \tag{20}$$

By substituting equation (20) in equation (18) we have a new self-scaling symmetric rank one as follows,

$$H_{m+1} = H_m + \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{{y_m^*}^T (\delta_1 v_m - H_m y_m^*)} \tag{21}$$

Also we suggest new self-scaling DFP formula to find the inverse of Hessian matrix as follows:

$$H_{m+1} = H_m + \frac{v_m v_m^T}{\delta_1 v_m^T y_m^*} - \frac{(H_m y_m^*)(H_m y_m^*)^T}{{y_m^*}^T H_m y_m^*} \tag{22}$$

where $y_m^* = y_m + \mu_m v_m$

$$\mu_m = \tau * \frac{N_m}{1+N_m}, \quad N_k = \frac{y_m^T v_m}{\|v_m\|} \text{ and } \tau \in (0,1) \text{ and } \delta_1 = \sqrt{\left|\frac{y_m^T H_{m+1} y_m^*}{y_m^T v_m}\right|}$$

Now, in the algorithm below we will combine the new self-scaling symmetric rank one with parameter of HS to become New PCG1, and we will combine the new self-scaling DFP with parameter of HS to become New PCG2.

## 3. The Outlines of the New PCG1 and New PCG2

Step1: Set $m = 0$, chose $x_0$, a real symmetric positive definite $H_0 = I$ and $\varepsilon$ is a small positive integer.
Step2: Calculate $g_m = \nabla f(x_m) = \frac{\partial f}{\partial x_m}$.
Step3: Evaluate $d_m = -H_m g_m$.
Step4: Find $\alpha_m > 0$, satisfying the strong Wolfe condition

$$f(x_m + \alpha_m d_m) - f(x_m) < k_1 \alpha_m g_m^T d_m \text{ and } |g_{m+1}^T d_m| < k_2 |g_m^T d_m|$$

where, $0 < k_1 < k_2 < 1$ .
Step5: Set $v_m = \alpha_m d_m$,

$$x_{m+1} = x_m + v_m \text{ and } y_m = g_{m+1} - g_m.$$

Step6: Compute $g_{m+1} = \nabla f(x_{m+1})$,
if $\|g_{m+1}\| < \varepsilon$, then stop.
Step7: Calculate $H_{m+1}$ using (21) or $H_{m+1}$ using (22)
step8: Evaluate $d_{m+1}$ by

$$d_{m+1} = -H_{m+1} g_{k+1} + \frac{g_{m+1}^T H_{m+1} y_m}{d_m^T y_m} d_m \text{ or } d_{m+1} = -H_{m+1} g_{k+1} + \frac{g_{m+1}^T H_{m+1} y_m}{d_m^T y_m} d_m$$

Step9: If $|g_m^T g_{m+1}| \geq 0.2\|g_{m+1}\|^2$ then go to step (4) , else continue.
Set $m = m + 1$ and repeat from Step (3).

## 4. Quasi-Newton Condition and Positive Definite

In this section, we will study the Quasi-Newton condition and positive definite condition of our two new self-scaling Quasi-Newton methods.

**Theorem 1:** If the new self-scaling symmetric rank one is applied to the quadratic functions with Hessian $Q = Q^T$, we have

$$H_{m+1}\, y_i^* = \delta_1 v_i \ , 0 \leq i \leq m \, .$$

**Proof:**
Multiply both sides of $(21)$ by $y_i^*$, so, we get

$$H_{m+1} y_i^* = H_m y_i^* + \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{{y_m^*}^T (\delta_1 v_m - H_m y_m^*)} y_i^* \tag{23}$$

for $m = 0$

$$H_1 y_0^* = H_0 y_0^* + \frac{(\delta_1 v_0 - H_0 y_0^*)(\delta_1 v_0 - H_0 y_0^*)^T}{{y_0^*}^T (\delta_1 v_0 - H_0 y_0^*)} y_0^*$$

Since $(\delta_1 v_0 - H_0 y_0^*)^T y_0^* = {y_0^*}^T (\delta_1 v_0 - H_0 y_0^*)$ because they are scalars
Here we get,

$$H_1 y_0^* = H_0 y_0^* + \delta_1 v_0 - H_0 y_0^* = \delta_1 v_m \tag{24}$$

Suppose that it is true for case $m - 1$, that is, $H_m\, y_i^* = \delta_1 v_i$.
Now, for $m + 1$, if $m = i$

$$H_{m+1} y_m^* = H_m y_m^* + \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{{y_m^*}^T (\delta_1 v_m - H_m y_m^*)} y_m^* = \delta_1 v_m$$

Consider if $i < m$

$$H_{m+1} y_i^* = H_m y_i^* + \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{{y_m^*}^T (\delta_1 v_m - H_m y_m^*)} y_i^*$$

To complete the proof it is enough to show the second term on the right hand side of the above equation is equal to zero. Consider

$$\begin{aligned}(\delta_1 v_m - H_m y_m^*)^T y_i^* &= \delta_1 v_m^T y_i^* - {y_m^*}^T H_m y_i^* \\ &= \delta_1 v_m^T y_i^* - \delta_1 {y_m^*}^T v_i\end{aligned}$$

$$= \delta_1 v_m^T y_i^* - \delta_1 v_m^T Q v_i$$
$$= \delta_1 v_m^T y_i^* - \delta_1 v_m^T y_i^* = 0$$

So, we have $H_{m+1} y_i^* = H_m y_i^*$
Then, $H_{m+1} y_m^* = \delta_1 v_i$

**Theorem 2:** Suppose that $g_m \neq 0$. In equation (21), if $H_m$ is positive definite. It turns that if ${y_m^*}^T(\delta_1 v_m - H_m y_m^*) > 0$, then $H_{m+1}$ is positive definite.

**Proof:** Let $X$ be any non-zero vector, multiply both sides of equation (21) by $X^T$ from left and by $X$ from right, so, we get

$$X^T H_{m+1} X = X^T H_m X + X^T \frac{(\delta_1 v_m - H_m y_m^*)(\delta_1 v_m - H_m y_m^*)^T}{{y_m^*}^T(\delta_1 v_m - H_m y_m^*)} X \quad (25)$$

since

$$X^T(\delta_1 v_m - H_m y_m^*) = (\delta_1 v_m - H_m y_m^*)^T X$$

then, the equation (25) gives

$$X^T H_{m+1} X = X^T H_m X + \frac{(X^T(\delta_1 v_m - H_m y_m^*))^2}{(\delta_1 {y_m^*}^T v_m - {y_m^*}^T H_m y_m^*)} \quad (26)$$

a sufficient condition for $H_{m+1}$ to be positive definite is $(\delta_1 {y_m^*}^T v_m - {y_m^*}^T H_m y_m^*) > 0$
if the denominator of the correction formula becomes zero or negative, $H_{m+1}$ can be discarded and $H_m$ can be used for the subsequent iteration. However, if this problem occurs frequently, the possibility exists that $H_{m+1}$ may not converge to $H^{-1}$. Then the expected rapid convergence may not materialize.

Since $H_m$ is positive definite and

$$\delta_1 {y_m^*}^T v_m = \delta_1 (y_m + \mu_m v_m)^T v_m = \delta_1 \mu_m \|v_m\|^2 + \delta_1 y_m^T v_m > 0$$

then, the above equation becomes

$$X^T H_{m+1} X > X^T H_m X + \frac{(X^T(\delta_1 v_m - H_m y_m^*))^2}{\delta_1 {y_m^*}^T v_m} \quad (27)$$

Here we observe that the right hand side of the (27) is positive, then $H_{m+1}$ is positive definite.

**Theorem 3:** If the new self-scaling DFP is applied to the quadratic function with Hessian $Q = Q^T$, we have $H_{m+1} y_i^* = \delta_1 v_i$, $0 \le i \le m$.

**Proof**:
Multiplying the equation (22) by $y_i^*$, so, we get

$$H_{m+1} y_i^* = H_m y_i^* + \frac{\delta_1 v_m v_m^T}{v_m^T y_m^*} y_i^* - \frac{(H_m y_m^*)(H_m y_m^*)^T}{{y_m^*}^T H_m y_m^*} y_i^* \quad (28)$$

for $m = 0$, we have

$$H_1 y_0^* = H_0 y_0^* + \frac{\delta_1 v_0 v_0^T}{v_0^T y_0^*} y_0^* - \frac{(H_0 y_0^*)(H_0 y_0^*)^T}{{y_0^*}^T H_0 y_0^*} y_0^*$$

Since $(H_0 y_0^*)^T y_0^*$ and ${y^*}_0^T H_0 y_0^*$ are scalars, then $(H_0 y_0^*)^T y_0^* = {y^*}_0^T H_0 y_0^*$
Therefore,

$$H_1 y_0^* = H_0 y_0^* + \delta_1 v_0 - H_0 y_0^* = \delta_1 v_0 \quad (29)$$

Assume it is true for $m-1$, that is $H_m y_i^* = \delta_1 v_i$.
Now, we show that $H_{m+1} y_i^* = \delta_1 v_i$, if $i = m$

$$H_{m+1} y_m^* = H_m y_m^* + \frac{\delta_1 v_m v_m^T}{v_m^T y_m^*} y_m^* - \frac{(H_m y_m^*)(H_m y_m^*)^T}{{y_m^*}^T H_m y_m^*} y_m^* = \delta_1 v_m$$

Now, consider the case $i < m$

$$H_{m+1} y_i^* = H_m y_i^* + \frac{\delta_1 v_m v_m^T}{v_m^T y_m^*} y_i^* - \frac{(H_m y_m^*)({y_m^*}^T H_m)}{{y_m^*}^T H_m y_m^*} y_i^*$$
$$= \delta_1 v_i + \frac{\delta_1 v_m}{v_m^T y_m^*} v_m^T y_i^* - \frac{(H_m y_m^*)}{{y_m^*}^T H_m y_m^*} \delta_1 {y_m^*}^T v_i$$

$v_m^T y_i^* = v_m^T Q v_i^* = \alpha_m \alpha_i d_m^T Q d_i^* = 0$ and also ${y_m^*}^T v_i = 0$ then,

$$H_{m+1}y_i^* = \delta_1 v_i$$

**Theorem 4:** Suppose that $g_m \neq 0$ . In equation (22), if $H_m$ is positive definite, then so is $H_{m+1}$.
**Proof:** Let X be any non-zero vector, multiplying the equation(22) by $X^T$ from the left and by X from the right, so, we have

$$X^T H_{m+1} X = X^T H_m X + \frac{\delta_1 X^T v_m v_m^T X}{v_m^T y_m^*} - \frac{X^T (H_m y_m^*)(H_k y_m^*)^T X}{{y_m^*}^T H_m y_m^*} \tag{30}$$

implies that

$$X^T H_{m+1} X = X^T H_m X + \frac{\delta_1 (v_m^T X)^2}{v_m^T y_m^*} - \frac{(X^T H_m y_m^*)^2}{{y_m^*}^T H_m y_m^*} \tag{31}$$

Define

$$a_1 = H_m^{\frac{1}{2}} X \text{ and } a_2 = H_m^{\frac{1}{2}} y_m^*$$

where

$$H_m = H_m^{\frac{1}{2}} H_m^{\frac{1}{2}}$$

Using the definitions of $a_1$ and $a_2$, we obtain

$$X^T H_{m+1} X = X^T H_m^{\frac{1}{2}} H_m^{\frac{1}{2}} X = {a_1}^T a_1$$

$$X^T H_{m+1} y_m^* = X^T H_m^{\frac{1}{2}} H_m^{\frac{1}{2}} y_m^* = {a_1}^T a_2$$

and

$${y_m^*}^T H_{m+1} y_m^* = {y_m^*}^T H_m^{\frac{1}{2}} H_m^{\frac{1}{2}} y_m^* = {a_2}^T a_2$$

hence,

$$X^T H_{m+1} X = {a_1}^T a_1 + \frac{\delta_1 (v_m^T X)^2}{v_m^T y_m^*} - \frac{({a_1}^T a_2)^2}{{a_2}^T a_2}$$

or

$$\mathrm{X^T H_{m+1} X} = \frac{\|a_1\|^2 \|a_2\|^2 - \langle a_1, a_2 \rangle}{\|a_2\|^2} + \frac{\delta_1 (v_m^T X)^2}{v_m^T y_m^*} \tag{32}$$

We also have

$$v_m^T y_m^* = v_m^T (y_m + \mu_m v_m) = v_m^T y_m + \mu_m \|v_m\|^2 = v_m^T (g_{m+1} - g_m) + \mu_m \|v_m\|^2$$
$$= -v_m^T g_m + \mu_m \|v_m\|^2$$

Since $v_m^T g_{m+1} = \alpha_m d_m^T g_{m+1} = 0$ by (in the conjugate direction algorithm $d_i^T g_{m+1} = 0$ for all m, $0 \leq m \leq n - 1$ and $0 \leq i \leq m$)
Because $v_m = \alpha_m d_m = -\alpha_m H_m g_m$
Then, we have

$$v_m^T y_m^* = \alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2$$

So, equation (32) gives

$$X^T H_{m+1} X = \frac{\|a_1\|^2 \|a_2\|^2 - \langle a_1, a_2 \rangle}{\|a_2\|^2} + \frac{(v_m^T X)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2} \tag{33}$$

By Cauchy-Schwars the first term in the right side of the equation (33) is nonnegative and the second term is nonnegative also, because $H_m$ is positive definite.
To complete the proof, we need to show that at least one of the terms is not equal to zero.
Only when if $a_1$ and $a_2$ are proportional does the first term disappear, that is if $a_1 = \beta_1 a_2$ for some scalars $\beta_1$. Therefore, to complete the proof, it is only to prove that if $a_1 = \beta_1 a_2$, then,

$$\frac{(v_m^T X)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2} > 0.$$

observe that

$$H_m^{\frac{1}{2}}X = a_1 = \beta_1 a_2 = \beta_1 H_m^{\frac{1}{2}} y_m^* = H_m^{\frac{1}{2}}(\beta_1 y_m^*)$$

Hence, $X_m = \beta_1 y_m^*$
Here, we obtain

$$\frac{(v_m^T X)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2} = \frac{{\beta_1}^2 (v_m^T y_m^*)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2}$$

Implies that

$$\frac{(v_m^T X)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2} =$$

$$\frac{{\beta_1}^2 (\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2)^2}{\alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2} = {\beta_1}^2 \alpha_m g_m^T H_m g_m + \mu_m \|v_m\|^2 > 0.$$

Thus, for all for $X_m \neq 0$

$$X^T H_{m+1} X > 0$$

# 5. Numerical Results of the New PCG1 and New PCG2

Testing the new methods implementation is the focus of this section. We compare New PCG1 with standard PCG(SR1/HS) and New PCG2 with standard PCG(DFP/HS). Tables 1 and 2 of the findings provide specific references to the number of iterations NOI and the number of functions NOF. The results in the two tables below confirm that the New PCG1 and the New PCG2 algorithms are superior to standard PCG methods with respect to NOI and NOF.

**Table 1.** Comparing the Performance of the Two Algorithms

Standard PCG (SR1 with HS) and New PCG1 (New self-scaling SR1 with HS)

| Test Function | N. | Standard PCG (SR1 with /HS) | | New PCG1 (New self-scaling SR1 with HS) | |
|---|---|---|---|---|---|
| | | NOI | NOF | NOI | NOF |
| Cubic | 4 | 14 | 42 | 14 | 41 |
| | 50 | 15 | 48 | 16 | 49 |
| | 100 | 16 | 53 | 16 | 49 |
| | 500 | 16 | 67 | 16 | 48 |
| | 1000 | 16 | 54 | 16 | 48 |
| | 2000 | 16 | 50 | 16 | 48 |
| | 3000 | 16 | 50 | 16 | 48 |
| Dixon | 4 | 14 | 31 | 14 | 32 |
| | 50 | 420 | 2119 | 290 | 955 |
| | 100 | 444 | 4447 | 305 | 1012 |
| | 500 | 301 | 2202 | 221 | 727 |
| | 1000 | 439 | 2033 | 296 | 1030 |
| | 2000 | 414 | 1865 | 333 | 1348 |
| | 3000 | 425 | 9277 | 228 | 762 |
| G-Central | 4 | 36 | 253 | 18 | 109 |
| | 50 | 43 | 331 | 22 | 140 |
| | 100 | 43 | 331 | 25 | 176 |
| | 500 | 60 | 496 | 27 | 200 |
| | 1000 | 66 | 554 | 36 | 314 |
| | 2000 | 66 | 554 | 43 | 405 |
| | 3000 | 72 | 616 | 40 | 351 |

**Table 1 Continued**

| | | | | | |
|---|---|---|---|---|---|
| Miele | 4 | 34 | 329 | 38 | 346 |
| | 50 | 41 | 182831 | 40 | 580 |
| | 100 | 47 | 182999 | 45 | 414 |
| | 500 | 53 | 183098 | 46 | 33657 |
| | 1000 | 53 | 183098 | 47 | 4274 |
| | 2000 | 59 | 183164 | 62 | 776 |
| | 3000 | 59 | 183164 | 40 | 1515 |
| Non-Diagonal | 4 | 28 | 75 | 25 | 72 |
| | 50 | 49 | 120 | 31 | 87 |
| | 100 | 51 | 123 | 31 | 85 |
| | 500 | 44 | 108 | 31 | 87 |
| | 1000 | * | * | 31 | 89 |
| | 2000 | 49 | 118 | 33 | 97 |
| | 3000 | 48 | 116 | 32 | 92 |
| Powell | 4 | 30 | 80 | 48 | 329 |
| | 50 | 31 | 93 | 27 | 75 |
| | 100 | 32 | 95 | 27 | 74 |
| | 500 | 33 | 97 | 30 | 87 |
| | 1000 | 33 | 97 | 30 | 88 |
| | 2000 | 33 | 97 | 30 | 90 |
| | 3000 | 33 | 97 | 30 | 87 |
| Rosen | 4 | 30 | 89 | 33 | 91 |
| | 50 | 31 | 89 | 33 | 92 |
| | 100 | 31 | 89 | 33 | 92 |
| | 500 | 36 | 102 | 34 | 95 |
| | 1000 | 36 | 102 | 34 | 94 |
| | 2000 | 34 | 98 | 34 | 94 |
| | 3000 | 37 | 100 | 34 | 94 |
| Shallow | 4 | 8 | 21 | 8 | 21 |
| | 50 | 8 | 21 | 8 | 21 |
| | 100 | 9 | 24 | 8 | 21 |
| | 500 | 9 | 24 | 8 | 21 |
| | 1000 | 9 | 24 | 8 | 21 |
| | 2000 | 9 | 24 | 9 | 24 |
| | 3000 | 9 | 24 | 9 | 24 |
| Sum | 4 | 3 | 11 | 3 | 11 |
| | 50 | 12 | 76 | 12 | 76 |
| | 100 | 14 | 83 | 14 | 83 |
| | 500 | 21 | 119 | 21 | 114 |
| | 1000 | 23 | 123 | 22 | 112 |
| | 2000 | 29 | 142 | 29 | 141 |
| | 3000 | 34 | 180 | 32 | 169 |
| Total | | 4286 | 1127254 | 3224 | 52518 |

**Table 2.** Comparing the Performance of the Two Algorithms

Standard PCG (DFP with HS) and New PCG2 (New self-scaling DFP with HS)

| Test Function | N. | Standard PCG (DFP with HS) | | New PCG2 (New self-scaling DFP with HS) | |
|---|---|---|---|---|---|
| | | NOI | NOF | NOI | NOF |
| Cubic | 4 | 14 | 40 | 13 | 37 |
| | 50 | 17 | 49 | 14 | 39 |
| | 100 | 17 | 49 | 14 | 39 |
| | 500 | 16 | 46 | 14 | 39 |
| | 1000 | 17 | 49 | 14 | 39 |
| | 2000 | 16 | 47 | 14 | 39 |
| | 3000 | 16 | 46 | 14 | 39 |

**Table 2 Continued**

| | | | | | |
|---|---|---|---|---|---|
| Dixon | 4 | 14 | 31 | 14 | 30 |
| | 50 | 455 | 1809 | 377 | 1461 |
| | 100 | 487 | 1913 | 410 | 1586 |
| | 500 | 464 | 1823 | 403 | 1577 |
| | 1000 | 465 | 1829 | 421 | 1632 |
| | 2000 | 499 | 1971 | 395 | 1552 |
| | 3000 | 448 | 1759 | 391 | 1498 |
| Fred | 4 | 8 | 24 | 8 | 23 |
| | 50 | 8 | 24 | 8 | 23 |
| | 100 | 8 | 24 | 8 | 23 |
| | 500 | 8 | 24 | 8 | 23 |
| | 1000 | 8 | 24 | 8 | 23 |
| | 2000 | 8 | 24 | 8 | 23 |
| | 3000 | 8 | 24 | 8 | 23 |
| G-Central | 4 | 34 | 237 | 27 | 198 |
| | 50 | 39 | 295 | 33 | 271 |
| | 100 | 39 | 295 | 32 | 249 |
| | 500 | 49 | 423 | 48 | 411 |
| | 1000 | 54 | 479 | 46 | 364 |
| | 2000 | 59 | 542 | 22 | 145 |
| | 3000 | 65 | 610 | 51 | 489 |
| Miele | 4 | 66 | 300 | 76 | 340 |
| | 50 | 80 | 374 | 78 | 355 |
| | 100 | 86 | 411 | 85 | 398 |
| | 500 | 94 | 454 | 37 | 145 |
| | 1000 | 100 | 496 | 50 | 233 |
| | 2000 | 105 | 525 | 75 | 390 |
| | 3000 | 107 | 543 | 73 | 364 |
| Non-Diagonal | 4 | 27 | 74 | 26 | 68 |
| | 50 | 44 | 107 | 31 | 82 |
| | 100 | 53 | 128 | 32 | 84 |
| | 500 | 49 | 118 | 31 | 82 |
| | 1000 | 49 | 119 | 31 | 85 |
| | 2000 | 48 | 116 | 31 | 82 |
| | 3000 | 48 | 116 | 31 | 82 |
| OSP | 4 | 8 | 45 | 8 | 45 |
| | 50 | 43 | 198 | 34 | 143 |
| | 100 | 56 | 227 | 49 | 189 |
| | 500 | 104 | 334 | 98 | 313 |
| | 1000 | 125 | 388 | 135 | 434 |
| | 2000 | 198 | 645 | 190 | 642 |
| | 3000 | 239 | 830 | 225 | 734 |
| Rosen | 4 | 34 | 95 | 30 | 82 |
| | 50 | 34 | 94 | 30 | 82 |
| | 100 | 34 | 94 | 30 | 82 |
| | 500 | 34 | 94 | 30 | 82 |
| | 1000 | 35 | 97 | 30 | 82 |
| | 2000 | 35 | 98 | 31 | 84 |
| | 3000 | 37 | 102 | 31 | 84 |
| Wolf | 4 | 11 | 24 | 8 | 18 |
| | 50 | 42 | 85 | 40 | 81 |
| | 100 | 44 | 89 | 44 | 89 |
| | 500 | 47 | 95 | 47 | 95 |
| | 1000 | 50 | 101 | 49 | 99 |
| | 2000 | 56 | 113 | 54 | 109 |
| | 3000 | 99 | 201 | 71 | 143 |
| Wood | 4 | 20 | 50 | 22 | 52 |
| | 50 | 23 | 57 | 20 | 49 |
| | 100 | 23 | 57 | 22 | 53 |
| | 500 | 23 | 57 | 20 | 48 |
| | 1000 | 23 | 57 | 20 | 48 |
| | 2000 | 23 | 57 | 20 | 48 |
| | 3000 | 23 | 57 | 20 | 48 |
| Total | | 5819 | 22832 | 4918 | 18743 |

## 6. Conclusions

We proposed two new preconditioned conjugate gradient algorithms, New PCG1 that combines conjugate gradient method Hestenes-Stiefel (HS) with new self-scaling symmetric Rank one(SR1), and New PCG2 that combines conjugate gradient method Hestenes-Stiefel (HS) with new self-scaling Davidon, Flecher and Powell (DFP) to find the minimum of the nonlinear problems, and new self-scaling SR1 and new self-scaling DFP satisfy the Quasi-Newton condition and the positive definite condition. The numerical tests were conducted on problems with low and high dimensionality, with comparisons made between different test functions. The effectiveness of the New PCG1 algorithm and New PCG2 algorithm is seen in Tables 1 and 2.

# Appendix. The Test Functions for Unconstrained Optimization

1. Generalized Central Function:

$$f(x)=\sum_{i=1}^{n/4}(exp(x_{4i-3}+x_{4i-2})^4+100((x_{4i-2}-x_{4i-1})^6+arctan((x_{4i-1}-x_{4i})^4+x_{4i-3}),$$
$$x_0=(1,2,2,2,\dots,1,2,2,2)^T.$$

2. Generalized Cubic Function:

$$f(x)=\sum_{i=1}^{n/2}(100(x_{2i}-x_{2i-1}^3)^2+(1-x_{2i})^2)\ ,x_0=(-1.2,1,\dots,-1.2,1)^T.$$

3. Generalized Non-Diagonal Function:

$$f(x)=\sum_{i=2}^{n}(100(x_1-x_i^2)^2+(1-x_i)^2),x_0(-1,\dots,-1)^T.$$

4. Generalized Rosen Brock Banana Function:

$$f(x)=\sum_{i=1}^{n/2}(100(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2),\qquad x_0=(-1.2,1,\dots,-1.2,1)^T.$$

5. Generalized Shallow Function:

$$f(x)=\sum_{i=1}^{n/2}([x_{2i-1}^2-x_{2i}]^2+(1-x_{2i-1})^2),\qquad x_0=(-2,-2,\dots,-2,-2)^T.$$

6. Miele Function:

$$f(x)=\sum_{i=1}^{n/4}((e^{x_{4i-3}}+10x_{4i-2})^2+100(x_{4i-2}+x_{4i-1})^6+(\tan(x_{4i-1}-x_{4i}))^4+(x_{4i-3})^8+(x_{4i}-1)^2),$$
$$x_0=(1,2,2,\dots,1,2,2)^T.$$

7. Sum of Quadrics (SUM) Function:

$$f(x)=\sum_{i=1}^{n}(x_i-i)^4,x_0=(1,1,\dots,1)^T.$$

8. Powell Function:

$$f(x)=\sum_{i=1}^{n/4}((x_{4i-3}-10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4),x_0$$
$$=(3,-1,0,1,\dots,3,-1,0,1)^T.$$

9. Oren and Spedicato OSP Function:

$$f(x) = \left(\sum_{i=1}^{n} i(x_i)^2\right)^2, x_0 = (1, \dots, 1)^T.$$

10. Wolfe Function:

$$f(x) = \left(-x_1\left(3 - \frac{x_1}{2}\right) + 2x_2 - 1\right)^2 + \sum_{i=1}^{n-1}\left(x_{i-1} - x_i\left(3 - \frac{x_i}{2} + 2x_{i+1} - 1\right)\right)^2 + \left(x_{n-1} - x_n\left(3 - \frac{x_n}{2}\right) - 1\right)^2, x_0 = (-1, \dots, -1)^T.$$

11. Extended Wood Function:

$$f(x) = \sum_{i=1}^{n/4}(100(x_{4i-3}^2 - x_{4i-2})^2 + (x_{4i-3} - 1)^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1 - x_{4i-1})^2 + 10.1(x_{4i-2} - 1)^2 + (x_{4i} - 1)^2 + 19.8(x_{4i-2} - 1)(x_{4i} - 1)), x_0 = (-3, -1, \dots, -3, -1)^T.$$

12. Fred Function:

$$f(x) = \sum_{i=1}^{n/2}\left(\left(-13 + x_{2i-1} + \left((5 - x_{2i})x_{2i} - 2\right)x_{2i}\right)^2 + \left(-29 + x_{2i-1} + \left((1 + x_{2i})x_{2i} - 14\right)x_{2i}\right)^2\right), x_0 = (30,3, \dots, 30,3)^T.$$

13. Generalized Dixon Function:

$$f(x) = \sum_{i=1}^{n-1}({x_i}^2 - x_{i+1})^2, \qquad x_0 = (-1, \dots, -1)^T.$$

# REFERENCES


[1] Hestenes M. R., Stiefeln E., "Method of conjugate gradient for solving linear equations," *J.Res. Nat. Bur. Stand*, vol. 49, pp. 409–436, 1952.

[2] Jorge N., Stephen J. Wright, *Numerical Optimization*. Verlag New York, 1999.

[3] Polak E., Ribiere G., "Note sur la convergence de directions conjugees," *Rev. Fr. Inform. Rech. Oper. 3*, vol. 16, pp. 35–43, 1969.

[4] Dai Y.H., Yuan Y., "A nonlinear conjugate gradient with a strong global convergence properties," *SIAM J. Optim.*, vol. 10, no. 1, pp. 177–182, 1999.

[5] Fletcher R., Reeves C. M., "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, pp. 149–154, 1964.

[6] Shmuel S. Oren, "Self-scaling variable metric algorithms, Part II," *Manage. Sci.*, vol. 20, pp. 863–874, 1974.

[7] Shmuel S. Oren, David G. Luenberger, "Self-scaling variable metric (SSVM) Algorithms, part I, criteria and Sufficient conditions for scaling a class of Algorithms," *Manage. Sci.*, vol. 20, No. 5, pp. 845–865, 1974.

[8] Biggs M. C., "A note on minimization algorithms which make use of non-quadratic properties of objective function," *J. Inst. Maths Applics*, vol. 12, Issue 3, pp. 337–338, 1973.

[9] Yueting Y., Chengxian X., Yuelin G.,, "An optimal self-scaling strategy to the modified symmetric rank one updating," *J. Xian Jiaotong Univ.*, vol. 17, No. 1, pp. 100–103, 2005.

[10] Sun L., "Updating of the self-scaling symmetric rank one algorithm with limited memory for large-scale unconstrained optimization," *Comput. Optim. Appl.*, vol. 27, No. 1, pp. 23–29, 2004.