Chapter

Unconstrained Optimization Methods: Conjugate Gradient Methods and Trust-Region Methods

Snezana S. Djordjevic

Abstract

Here, we consider two important classes of unconstrained optimization methods: conjugate gradient methods and trust region methods. These two classes of methods are very interesting; it seems that they are never out of date. First, we consider conjugate gradient methods. We also illustrate the practical behavior of some conjugate gradient methods. Then, we study trust region methods. Considering these two classes of methods, we analyze some recent results.

Keywords: conjugate gradient method, hybrid conjugate gradient method, three-term conjugate gradient method, modified conjugate gradient method, trust region methods

1. Introduction

Remind to the unconstrained optimization problem which we can present as

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function.

Here, we consider two classes of unconstrained optimization methods: conjugate gradient methods and trust region methods. Both of them are made with the aim to solve the unconstrained optimization problem (1).

In this chapter, at first, we consider the conjugate gradient methods. Then, we study trust region methods. Also, we try to give some of the most recent results in these areas.

2. Conjugate gradient method (shortly *CG*)

The conjugate gradient method is the method between the steepest descent method and the Newton method.

The conjugate gradient method in fact deflects the direction of the steepest descent method by adding to it a positive multiple of the direction used in the last step.

The restarting and the preconditioning are very important to improve the conjugate gradient method [47].

Some of well-known *CG* methods are [12, 19, 20, 23, 24, 31, 39, 40, 49]:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$$

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$$

$$\beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}$$

$$\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$$

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}$$

$$\beta_k^{N} = \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k}\right)^T \frac{g_{k+1}}{d_k^T y_k}$$

$$\beta_k^{WYL} = \frac{g_k^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1}\right)}{\|g_{k-1}\|^2}$$

Consider positive definite quadratic function

$$f(x) = \frac{1}{2} x^T G x + b^T x + c, \tag{2}$$

where G is an $n \times n$ symmetric positive definite matrix, $b \in \mathbb{R}^n$, and c is a real number.

Theorem 1.2.1. *[47] (Property theorem of conjugate gradient method) For positive definite quadratic function (2), FR conjugate gradient method with the exact line search terminates after $m \le n$ steps, and the following properties hold for all i, $0 \le i \le m$:*

$$d_i^T G d_j = 0, j = 0, 1, \ldots, i-1;$$

$$g_i^T g_j = 0, j = 0, 1, \ldots, i-1;$$

$$d_i^T g_i = -g_i^T g_i;$$

$$[g_0, g_1, \ldots, g_i] = [g_0, Gg_0, \ldots, G^i g_0];$$

$$[d_0, d_1, \ldots, d_i] = [g_0, Gg_0, \ldots, G^i g_0],$$

where m is the number of distinct eigenvalues of G.

Now, we give the algorithm of conjugate gradient method.

Algorithm 1.2.1. *(CG method).*

Assumptions: $\varepsilon < 0$ and $x_0 \in \mathbb{R}^n$. *Let* $k = 0$, $t_0 = 0$, $d_{-1} = 0$, $d_0 = -g_0$, $\beta_{-1} = 0$, and $\beta_0 = 0$.

Step 1. If $\|g_k\| \le \varepsilon$, then STOP.
Step 2. Calculate the step-size t_k by a line search.
Step 3. Calculate β_k by any of the conjugate gradient method.
Step 4. Calculate $d_k = -g_k + \beta_{k-1} d_{k-1}$.
Step 5. Set $x_{k+1} = x_k + t_k d_k$.
Step 6. Set $k = k + 1$ and go to Step 1.

2.1 Convergence of conjugate gradient methods

Theorem 1.2.2. *[47] (Global convergence of FR conjugate gradient method) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on a bounded level set*

$$L = \{x \in \mathbb{R}^n | f(x) \le f(x_0)\},$$

and let FR method be implemented by the exact line search. Then, the produced sequence $\{x_k\}$ has at least one accumulation point, which is a stationary point, i.e.:

1. *When $\{x_k\}$ is a finite sequence, then the final point x^* is a stationary point of f.*

2. *When $\{x_k\}$ is an infinite sequence, then it has a limit point, and it is a stationary point.*

In [35], a comparison of two methods, the steepest descent method and the conjugate gradient method which are used for solving systems of linear equations, is illustrated. The aim of the research is to analyze, which method is faster in solving these equations and how many iterations are needed by each method for solving.

The system of linear equations in the general form is considered:

$$Ax = B, \tag{3}$$

where matrix A is symmetric and positive definite.

The conclusion is that the *SD* method is a faster method than the *CG*, because it solves equations in less amount of time.

By the other side, the authors find that the *CG* method is slower but more productive than the *SD*, because it converges after less iterations.

So, we can see that one method can be used when we want to find solution very fast and another can converge to maximum in less number of iterations.

Again, we consider the problem (1), where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function and its gradient is available.

A hybrid conjugate gradient method is a certain combination of different conjugate gradient methods; it is made to improve the behavior of these methods and to avoid the jamming phenomenon.

An excellent survey of hybrid conjugate gradient methods is given in [5].

Three-term conjugate gradient methods were studied in the past (e.g., see [8, 32, 34], etc.); but, from recent papers about *CG* methods, we can conclude that maybe the mainstream is made by three-term and even four-term conjugate gradient methods. An interesting paper about a five-term hybrid conjugate gradient method is [1]. Also, from recent papers we can conclude that different modifications of the existing *CG* methods are made, as well as different hybridizations of *CG* and *BFGS* methods.

Consider unconstrained optimization problem (1), where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded from below. Starting from an initial point $x_0 \in \mathbb{R}^n$, the three-term conjugate gradient method with line search generates a sequence $\{x_k\}$, given by the next iterative scheme:

$$x_{k+1} = x_k + t_k d_k, \tag{4}$$

where t_k is a step-size which is obtained from the line search, and

$$d_0 = -g_0, d_{k+1} = -g_{k+1} + \delta_k s_k + \eta_k y_k.$$

In the last relation, δ_k and η_k are the conjugate gradient parameters, $s_k = x_{k+1} - x_k, g_k = \nabla f(x_k)$, and $y_k = g_{k+1} - g_k$. We can see that the search direction d_{k+1} is computed as a linear combination of $-g_{k+1}$, s_k, and y_k.

In [6], the author suggests another way to get three-term conjugate gradient algorithms by minimization of the one-parameter quadratic model of the function f. The idea is to consider the quadratic approximation of the function f in the current point and to determine the search direction by minimization of this quadratic model. It is assumed that the symmetrical approximation of the Hessian matrix B_{k+1} satisfies the general quasi-Newton equation which depends on a positive parameter:

$$B_{k+1} s_k = \omega^{-1} y_k, \omega \neq 0. \tag{5}$$

In this paper the quadratic approximation of the function f is considered:

$$\Phi_{k+1}(d) = f_{k+1} + g_{k+1}^T d + \frac{1}{2} d^T B_{k+1} d.$$

The direction d_{k+1} is computed as

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \tag{6}$$

where the scalar β_k is determined as the solution of the following minimizing problem:

$$\min_{\beta_k \in \mathbb{R}} \Phi_{k+1}(d_{k+1}). \tag{7}$$

From (6) and (7), the author obtains

$$\beta_k = \frac{g_{k+1}^T B_{k+1} s_k - g_{k+1}^T s_k}{s_k^T B_{k+1} s_k}. \tag{8}$$

Using (5), from (7), the next expression for β_k is obtained:

$$\beta_k = \frac{g_{k+1}^T y_k - \omega g_{k+1}^T s_k}{y_k^T s_k}. \tag{9}$$

Using the idea of Perry [36], the author obtains

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1} - \omega s_k^T g_{k+1}}{y_k^T s_k} s_k - \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k.$$

In fact, in this approach the author gets a family of three-term conjugate gradient algorithms depending of a positive parameter ω.

Next, in [52], the *WYL* conjugate gradient (*CG*) formula, with $\beta_k^{WYL} \geq 0$, is further studied. A three-term *WYL CG* algorithm is presented, which has the sufficiently descent property without any conditions. The global convergence and the linear convergence are proven; moreover, the n-step quadratic convergence with a restart strategy is established if the initial step length is appropriately chosen.

The first three-term Hestenes-Stiefel (*HS*) method (*TTHS* method) can be found in [55].

Baluch et al. [7] describe a modified three-term Hestenes-Stiefel (*HS*) method. Although the earliest conjugate gradient method *HS* achieves global convergence using an exact line search, this is not guaranteed in the case of an inexact line search. In addition, the *HS* method does not usually satisfy the descent property. The modified three-term conjugate gradient method from [7] possesses a sufficient descent property regardless of the type of line search and guarantees global convergence using the inexact Wolfe-Powell line search [50, 51]. The authors also prove the global convergence of this method. The search direction, which is considered in [7], has the next form:

$$d_k = \begin{cases} -g_k, \text{ if } k = 0, \\ -g_k + \beta_k^{BZA} d_{k-1} - \theta_k^{BZA} y_{k-1}, \text{ if } k \geq 1, \end{cases}$$

where $\beta_k^{BZA} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}$, $\theta_k^{BZA} = \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1} + \mu |g_k^T d_{k-1}|}$, $\mu > 1$.

In [13], an accelerated three-term conjugate gradient method is proposed, in which the search direction satisfies the sufficient descent condition as well as extended Dai-Liao conjugacy condition:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \ t \geq 0.$$

This method seems different from the existent methods.

Next, Li-Fushikuma quasi-Newton equation is

$$\nabla^2 f(x_k) s_{k-1} = z_{k-1}, \tag{10}$$

where

$$z_{k-1} = y_{k-1} + C \|g_{k-1}\|^r s_{k-1} + \max\left\{ -\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}, 0 \right\} s_{k-1},$$

where C and r are two given positive constants. Based on (10), Zhou and Zhang [56] propose a modified version of *DL* method, called *ZZ* method in [13].

In [30], some new conjugate gradient methods are extended, and then some three-term conjugate gradient methods are constructed. Namely, the authors remind to [41, 42], with its conjugate gradient parameters, respectively:

$$\beta_k^{RMIL} = \frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}, \tag{11}$$

$$\beta_k^{MRMIL} = \frac{g_k^T (g_k - g_{k-1} - d_{k-1})}{\|d_{k-1}\|^2}, \tag{12}$$

wherefrom it is obvious that $\beta_k^{MRMIL} = \beta_k^{RMIL}$ for the exact line search. Let us say that these methods, presented in [41, 42], are *RMIL* and *MRMIL* methods.

The three-term *RMIL* and *MRMIL* methods are introduced in [30].

The search direction d_k can be expressed as

$$d_0 = -g_0, d_k = -g_k + \beta_k d_{k-1} + \theta_k y_{k-1},$$

where β_k is given by (11) or (12), and

$$\theta_k = -\frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2}.$$

An important property of the proposed methods is that the search direction always satisfies the sufficient descent condition without any line search, that is, the next relation always holds

$$g_k^T d_k \leq -\|g_k\|^2.$$

Under the standard Wolfe line search and the classical assumptions, the global convergence properties of the proposed methods are proven.

Having in view the conjugate gradient parameter suggested in [49], in [45] the next two conjugate gradient parameters are presented:

$$\beta_k^{MHS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}, \tag{13}$$

$$\beta_k^{MLS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{-d_{k-1}^T g_{k-1}}. \tag{14}$$

Motivated by [49], as well as by [45], in [1], a new hybrid nonlinear *CG* method is proposed; it combines the features of five different *CG* methods, with the aim of combining the positive features of different non-hybrid methods. The proposed method generates descent directions independently of the line search. Under some assumptions on the objective function, the global convergence is proven under the standard Wolfe line search. Conjugate gradient parameter, proposed in [1], is

$$\beta_k^{hAO} = \frac{\|g_k\|^2 - \max\left\{0, \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}\right\}}{\max\left\{\|g_{k-1}\|^2, d_{k-1}^T (g_k - g_{k-1}), -d_{k-1}^T g_{k-1}\right\}}. \tag{15}$$

Let's note that the proposed method is hybrid of *FR*, *DY*, *WYL*, *MHS*, and *MLS*.

The behaviors of the methods *BZA*, *TTRMIL*, *MRMIL*, *MHS*, *MLS*, and *hAO* are illustrated by the next tables.

The test criterion is CPU time.

The tests are performed on the computer Workstation Intel Celeron CPU 1,9 GHz.

The experiments are made on the test functions from [3].

Each problem is tested for a number of variables $n = 1000$ and $n = 5000$.

The average CPU time values are given in the last rows of these tables (**Tables 1**–**4**).

In [2], based on the numerical efficiency of Hestenes-Stiefel (*HS*) method, a new modified *HS* algorithm is proposed for unconstrained optimization. The new direction independent of the line search satisfies the sufficient descent condition. Motivated by theoretical and numerical features of three-term conjugate gradient (*CG*) methods proposed by [33], similar to the approach in [10], the new direction is computed by minimizing the distance between the *CG* direction and the direction of the three-term *CG* methods proposed by [33]. Under some mild conditions, the global convergence of the new method for general functions is established when the standard Wolfe line search is used. In this paper the conjugate gradient parameter is given by

$$\beta_k = \beta_k^{HS} \theta_k, \tag{16}$$

function	*BZA*	TTRMIL	MRMIL	MHS	MLS	hAO
Ext.Pen.	21.793340	20.966534	16.036903	19.812127	21.933741	20.326930
Pert.Quad.	21.855740	22.542144	15.506499	20.904134	22.230142	18.954121
Raydan1	6.801644	7.066845	6.349241	7.098045	7.066845	7.332047
Raydan2	0.608404	0.592804	0.577204	0.592804	0.608404	0.639604
Diag.1	0.608404	0.608404	0.577204	0.608404	0.514803	0.577204
Diag.2	5.163633	5.600436	4.695630	4.758031	5.662836	4.851631
Diag.3	5.616036	5.694037	5.241634	5.756437	5.584836	5.506835
Gen.Tridiag.-1	3.042019	2.932819	2.683217	2.948419	2.792418	2.808018
Hager	2.917219	2.932819	2.620817	3.042019	2.917219	2.886019
Ext.Tridiag.-1	2.886019	2.932819	2.761218	2.932819	2.730018	2.917219
Ext.ThreeExp.	2.979619	2.964019	2.605217	2.886019	3.042019	2.714417
Diag.4	2.901619	2.870418	2.574016	2.792418	2.948419	2.652017
Diag.5	2.792418	2.917219	2.574016	2.901619	3.026419	2.901619
Ext.Himm.	2.761218	2.714417	2.667617	2.964019	2.995219	2.854818
Ext.PSC1	2.932819	2.745618	2.714417	2.511616	3.026419	2.792418
FullHess.FH2	2.870418	2.948419	2.886019	2.839218	3.010819	2.948419
Ext.Bl.Diag.BD1	2.979619	2.886019	2.948419	2.886019	2.901619	2.542816
Quad.QF1	2.854818	2.870418	3.057620	2.964019	2.964019	2.886019
Ext.Quad.Pen.QP1	2.948419	2.808018	2.605217	2.964019	2.823618	2.542816
Quad.QF2	2.839218	2.620817	2.886019	2.979619	2.901619	2.683217
Ext.EP1	2.730018	2.402415	2.932819	2.698817	2.792418	2.652017
Ext.Tridiag.-2	2.683217	2.605217	2.839218	2.870418	2.886019	2.542816
Tridia	2.683217	2.511616	2.964019	2.823618	2.823618	2.511616
Arwhead	2.917219	2.995219	2.745618	2.823618	2.745618	2.012413
Dqdrtic	2.761218	2.995219	2.901619	2.823618	2.730018	2.589617
Quartc(Cute)	2.886019	2.776818	2.886019	2.776818	2.870418	2.839218
Dixon3dq(Cute)	2.808018	2.948419	2.948419	2.839218	2.917219	2.605217

Table 1.
n = *1000.*

function	BZA	TTRMIL	MRMIL	MHS	MLS	hAO
Biggsb1(Cute)	2.792418	2.870418	2.870418	2.917219	2.979619	2.901619
Gen.quart.	2.917219	2.932819	2.464816	2.948419	2.808018	2.620817
Diag.7	2.574016	2.589617	2.870418	2.620817	3.026419	2.698817
Diag.8	2.730018	2.979619	2.839218	2.964019	2.792418	2.979619
Full Hess.FH3	2.948419	2.574016	2.698817	3.026419	2.636417	2.745618
Himmelbg	2.854818	3.010819	2.901619	2.854818	2.995219	2.730018
Ext.Pow.	2.901619	2.854818	2.761218	2.808018	2.870418	2.995219
Ext.Maratos	2.854818	2.948419	2.870418	2.995219	2.870418	2.917219
Ext.Cliff	2.964019	3.042019	2.854818	2.932819	2.886019	2.854818
Pert.quad.diag.	2.714417	3.104420	2.683217	2.964019	2.667617	2.901619
Ext.Wood	2.995219	2.932819	2.948419	2.948419	2.964019	2.948419

function	BZA	TTRMIL	MRMIL	MHS	MLS	hAO
Ext.Trigon.	2.792418	2.995219	2.839218	3.010819	2.995219	2.745618
Ext.Rosenbr.	2.964019	2.839218	2.948419	2.932819	2.995219	2.776818
Average	3.915625	3.928105	3.533423	3.868045	3.973345	3.722184

Table 2.
n = *1000.*

function	BZA	TTRMIL	MRMIL	MHS	MLS	hAO
Ext.Pen.	46.160696	46.831500	48.656712	66.284825	65.863622	63.695208
Pert.Quad.	48.375910	45.801894	52.307135	66.612427	66.113224	65.551620
Raydan1	12.994883	12.105678	13.759288	16.972909	16.598506	16.754507
Raydan2	1.170008	1.029607	1.076407	1.154407	1.092007	1.107607
Diag.1	8.845257	0.904806	1.076407	1.123207	1.170008	1.092007
Diag.2	8.658055	7.831250	7.924851	9.094858	10.358466	10.327266
Diag.3	8.361654	9.141659	8.673656	10.686068	10.358466	10.514467
Gen.Tridiag.-1	5.616036	5.382034	5.865638	6.021639	6.489642	6.364841
Hager	5.241634	4.851631	5.881238	6.286840	5.304034	6.021639
Ext.Tridiag.-1	5.007632	4.804831	5.740837	5.787637	6.224440	5.803237
Ext.ThreeExp.	4.882831	4.820431	5.522435	6.115239	6.333641	5.834437
Diag.4	4.929632	4.898431	5.179233	5.803237	6.177640	6.427241
Diag.5	5.694037	4.851631	5.538036	5.709637	5.896838	6.115239
Ext.Himm.	5.834437	5.116833	5.382034	6.099639	5.772037	6.411641
Ext.PSC1	5.023232	5.054432	5.163633	6.411641	6.115239	5.990438
FullHess.FH2	5.210433	4.929632	4.851631	6.068439	6.349241	6.349241
Ext.Bl.Diag.BD1	4.851631	5.007632	5.226033	6.364841	6.364841	5.569236
Quad.QF1	5.475635	5.662836	6.302440	6.177640	6.146439	6.286840
Ext.Quad.Pen.QP1	5.226033	5.163633	4.929632	6.130839	5.818837	5.943638
Quad.QF2	5.335234	4.836031	5.990438	6.084039	6.084039	6.084039
Ext.EP1	5.070032	5.038832	6.052839	6.115239	4.992032	6.177640
Ext.Tridiag.-2	4.851631	4.976432	4.851631	6.349241	5.990438	6.099639
Tridia	5.413235	4.820431	5.475635	5.569236	5.818837	6.021639
Arwhead	4.867231	4.882831	5.023232	6.099639	6.380441	6.177640
Dqdrtic	5.163633	4.945232	5.023232	5.428835	6.006038	5.850038
Quartc(Cute)	5.912438	5.350834	5.834437	5.787637	5.896838	6.193240
Dixon3dq(Cute)	5.428835	4.789231	5.163633	6.162039	5.616036	5.881238

Table 3.
n = *5000.*

where

$$\theta_k = 1 - \frac{\left(g_k^T d_{k-1}\right)^2}{\|g_k\|^2 \|d_{k-1}\|^2}.$$

But this new *CG* direction does not fulfill a descent condition, so further modification is made, namely, having in view [53], the authors [2] introduce

function	BZA	TTRMIL	MRMIL	MHS	MLS	hAO
Biggsb1(Cute)	5.148033	4.695630	5.413235	5.912438	6.052839	6.349241
Gen.quart.	5.288434	4.758031	5.023232	6.349241	6.052839	4.960832
Diag.7	5.163633	4.664430	5.054432	5.959238	6.193240	6.255640
Diag.8	5.787637	4.742430	4.898431	6.099639	5.600436	6.208840
Full Hess.FH3	5.444435	4.789231	5.569236	6.177640	6.162039	6.224440
Himmelbg	5.584836	6.130839	5.475635	5.475635	6.006038	5.912438
Ext.Pow.	5.569236	4.789231	4.773631	5.990438	5.772037	6.162039
Ext.Maratos	5.148033	5.740837	4.976432	6.021639	6.286840	6.130839
Ext.Cliff	5.943638	5.850038	4.976432	5.990438	5.304034	6.286840
Pert.quad.diag.	5.912438	6.427241	4.976432	6.318041	6.115239	6.068439
Ext.Wood	5.584836	5.647236	4.789231	6.255640	5.350834	6.021639
Ext.Trigon.	5.366434	5.709637	4.773631	6.115239	6.021639	5.787637
Ext.Rosenbr.	6.177640	5.319634	4.617630	6.333641	6.021639	6.021639
Average	7.79302995	7.327367	7.694749	9.287519525	9.206789	9.225899

Table 4.
n = *5000.*

$$\overline{\beta}_k = \beta_k - \lambda \left(\frac{\|y_{k-1}\| \theta_k}{d_{k-1}^T y_{k-1}} \right)^2 g_k^T d_{k-1},$$

where $\lambda > \frac{1}{4}$ is a parameter. Also, the global convergence is proven under standard conditions.

It is worth to mention the next papers about this theme, which can be interesting [4, 14–17, 25–27].

3. Trust region methods

We remind that the basic idea of Newton method is to approximate the objective function $f(x)$ around x_k by using a quadratic model:

$$q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s_k^T G_k s,$$

where $g_k = \nabla f(x_k)$, $G_k = \nabla^2 f(x_k)$, and also use the minimizer s_k of $q^{(k)}(s)$ to set $x_{k+1} = x_k + s_k$.

Also, remind that Newton method can only guarantee the local convergence, i.e., when s is small enough and the method is convergent locally.

Further, Newton method cannot be used when Hessian is not positive definite.

There exists another class of methods, known as trust region methods. It does not use the line search to get the global convergence, as well as it avoids the difficulty which is the consequence of the nonpositive definite Hessian in the line search.

Furthermore, it produces greater reduction of the function f than line search approaches.

Here, we define the region around the current iterate:

$$\Omega_k = \{x : \|x - x_k\| \le \Delta_k\},$$

where Δ_k is the radius of Ω_k, inside which the model is trusted to be adequate to the objective function.

Our further intention is to choose a step which should be the approximate minimizer of the quadratic model in the trust region. In fact, $x_k + s_k$ should be the approximately best point on the sphere:

$$\{x_k + s \,|\, \|s\| \le \Delta_k\},$$

with the center x_k and the radius Δ_k.

In the case that this step is not acceptable, we reduce the size of the step, and then we find a new minimizer.

This method has the rapid local convergence rate, and that's the property of Newton method and quasi-Newton method, too, but the important characteristic of trust region method is also the global convergence.

Since the step is restricted by the trust region, this method is also called the restricted step method.

The model subproblem of the trust region method is

$$\min q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s, \tag{17}$$

$$\text{s.t.} \|s\| \le \Delta_k, \tag{18}$$

where Δ_k is the trust region radius and B_k is a symmetric approximation of the Hessian G_k.

In the case that we use the standard l_2 norm $\| \cdot \|_2$, s_k is the minimizer of $q^{(k)}(s)$ in the ball of radius Δ_k. Generally, different norms define the different shapes of the trust region.

Setting $B_k = G_k$ in (17)–(18), the method becomes a Newton-type trust region method.

The problem by itself is the choice of Δ_k at each single iteration.

If the agreement between the model $q^{(k)}(s)$ and the objective function $f(x_k + s)$ is satisfactory enough, the value Δ_k should be chosen as large as it is possible. The expression $Ared_k = f(x_k) - f(x_k + s_k)$ is called the *actual reduction*, and the expression $Pred_k = q^{(k)}(0) - q^{(k)}(s_k)$ is called the *predicted reduction*; here, we emphasize that

$$r_k = \frac{Ared_k}{Pred_k}$$

measures the agreement between the model function $q^{(k)}(s)$ and the objective function $f(x_k + s)$.

If r_k is close to 0 or it is negative, the trust region is going to shrink; otherwise, we do not change the trust region.

The conclusion is that r_k is important in making the choice of new iterate x_{k+1} as well as in updating the trust region radius Δ_k. Now, we give the trust region algorithm.

Algorithm 1.3.1. *(Trust region method).*

Assumptions: x_0, $\overline{\Delta}$, $\Delta_0 \in (0, \overline{\Delta})$, $\varepsilon \ge 0$, $0 < \eta_1 \le \eta_2 < 1$, and $0 < \gamma_1 < 1 < \gamma_2$.

Let $k = 0$.

Step 1. If $\|g_k\| \le \varepsilon$, *then STOP.*

Step 2. Approximately solve the problem (17)–(18) for s_k.

Step 3. Compute $f(x_k + s_k)$ and r_k. Set

$$x_{k+1} = \begin{cases} x_k + s_k, & \text{if } r_k \geq \eta_1, \\ x_k, & \text{otherwise}. \end{cases}$$

Step 4. If $r_k < \eta_1$, then $\Delta_{k+1} \in (0, \gamma_1 \Delta_k)$.
If $r_k \in [\eta_1, \eta_2)$, then $\Delta_{k+1} \in (\gamma_1 \Delta_k, \Delta_k)$.
If $r_k \geq \eta_2$ and $\|s_k\| = \Delta_k$, then $\Delta_{k+1} \in \left[\Delta_k, \min\left\{\gamma_2 \Delta_k, \overline{\Delta}\right\}\right]$.
Step 5. Generate B_{k+1}, update $q^{(k)}$, set $k = k + 1$, and go to Step 1.

In Algorithm 1.3.1, $\overline{\Delta}$ is a bound for all Δ_k. Those iterations with the property $r_k \geq \eta_2$ (and so those for which $\Delta_{k+1} \geq \Delta_k$) are called *very successful iterations*; the iterations with the property $r_k \geq \eta_1$ (and so those for which $x_{k+1} = x_k + s_k$) are called *successful iterations*; and the iterations with the property $r_k < \eta_1$ (and so those for which $x_{k+1} = x_k$) are called *unsuccessful iterations*. Generally, the iterations from the two first cases are called *successful iterations*.

Some choices of parameters are $\eta_1 = 0, 01$, $\eta_2 = 0, 75$, $\gamma_1 = 0, 5$, $\gamma_2 = 2$, $\Delta_0 = 1$, and $\Delta_0 = \frac{1}{10}\|g_0\|$. The algorithm is insensitive to change of these parameters.

Next, if $r_k < 0, 01$, then Δ_{k+1} can be chosen from $(0.01, 0.5)\|s_k\|$ on the basis of a polynomial interpolation.

In the case of quadratic interpolation, we set

$$\Delta_{k+1} = \lambda \|s_k\|,$$

where

$$\lambda = \frac{-g_k^T s_k}{2\left(f(x_k + s_k) - f(x_k) - g_k^T s_k\right)}.$$

3.1 Convergence of trust region methods

Assumption 1.3.1 *(Assumption A_0).*

We assume that the approximations of Hessian $\{B_k\}$ are uniformly bounded in norm and the level set $L = \{x | f(x) \leq f(x_0)\}$ is bounded, as well as $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on L. We allow the length of the approximate solution s_k of the subproblem (17)–(18) to exceed the bound of the trust region, but we also assume that

$$\|s_k\| \leq \tilde{\eta} \Delta_k,$$

where $\tilde{\eta}$ is a positive constant.

In this kind of trust region way of thinking, generally we do not seek an accurate solution of the subproblem (17)–(18); we are satisfied by finding a nearly optimal solution of the subproblem (17)–(18).

Strong theoretical as well as numerical results can be obtained if the step s_k, produced by Algorithm 1.3.1, satisfies

$$q_k(0) - q_k(s_k) \geq \beta_1 \|g_k\|_2 \min\left\{\Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2}\right\}, \beta_1 \in (0, 1).$$

Theorem 1.3.1 *[47] Under Assumption A_0, if Algorithm 3.1 has finitely many successful iterations, then it converges to the first-order stationary point.*

Theorem 1.3.2 *[47] Under Assumption A_0, if Algorithm 3.1 has infinitely many successful iterations, then*

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

In [44], it is emphasized that trust region methods are very effective for optimization problems and a new adaptive trust region method is presented. This method combines a modified secant equation with the *BFGS* update formula and an adaptive trust region radius, where the new trust region radius makes use of not only the function information but also the gradient information. Let $\hat{B}_k$ be a positively definite matrix based on modified Cholesky factorization [43]. Under suitable conditions, in [44] the global convergence is proven; also, the local superlinear convergence of the proposed method is demonstrated. Motivated by the adaptive technique, the proposed method possesses the following nice properties:

1. The trust region radius uses not only the gradient value but also the function value.

2. Computing the matrix $\hat{B}_k$ of the inverse and the value of $\|\hat{B}_k{}^{-1}\|$, at each iterative point x_k, is not required.

3. The computational time is reduced.

A modified secant equation is introduced:

$$B_{k+1}d_k = q_k, \tag{19}$$

where $q_k = y_k + h_k d_k, f_k = f(x_k)$, and $h_k = \frac{(g_{k+1}+g_k)^T d_k + 2(f_k - f_{k+1})}{\|d_k\|^2}$.

When f is twice continuously differentiable and B_{k+1} is generated by the *BFGS* formula, where $B_0 = I$, this modified secant Eq. (19) possesses the following nice property:

$$f_k = f_{k+1} - g_{k+1}^T d_k + \frac{1}{2} d_k^T B_{k+1} d_k,$$

and this property holds for all k.

Under classical assumptions, the global convergence of the method presented in [44] is also proven in this paper.

In [28], the hybridization of monotone and non-monotone approaches is made; a modified trust region ratio is used, in which more information is provided about the agreement between the exact and the approximate models. An adaptive trust region radius is used, as well as two accelerated Armijo-type line search strategies to avoid resolving the trust region subproblem whenever a trial step is rejected. It is shown that the proposed algorithm is globally and locally superlinearly convergent. In this paper trust region methods are denoted shortly by *TR*; it is emphasized that in *TR* method, having in view that the iterative scheme is

$$x_0 \in \mathbb{R}^n, x_{k+1} = x_k + s_k, k = 0, 1, \ldots,$$

and it often happens that s_k is an approximate solution of the following quadratic subproblem:

$$\min_{s\in\mathbb{R}^n,\ \|s_k\|\le\Delta_k} m_k(s) = g_k^T s + \frac{1}{2} s_k^T B_k s. \tag{20}$$

Performance of the *TR* methods is much influenced by the strategy of choosing the *TR* radius at each iteration. To determine the radius Δ_k, in the standard *TR* method, the agreement between $f(x_k + s)$ and $m_k(s)$ is evaluated by the so-called *TR* ratio ρ_k:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}.$$

When ρ_k is negative or a small positive number near to zero, the quadratic model is a poor approximation of the objective function. In such situation, Δ_k should be decreased and, consequently, the subproblem (20) should be solved again. However, when ρ_k is close to 1, it is reasonable to use the quadratic model as an approximation of the objective function. So, the step s_k should be accepted and Δ_k can be increased. Here, the authors use the modified version of ρ_k:

$$\overline{\rho}_k = \frac{R_k - f(x_k + s_k)}{P_k - m_k(s_k)},$$

where $R_k = \eta_k f_{l(k)} + (1 - \eta_k) f_k$, $\eta_k \in [\eta_{min}, \eta_{max}]$, $\eta_{min} \in [0, 1)$, and $\eta_{max} \in [\eta_{min}, 1]$. Also,

$$f_{l(k)} = \max_{0 \le j \le q(k)} \left\{ f_{k-j} \right\}, f_i = f(x_i), q(0) = 0, 0 \le q(k) \le \min\{q(k-1) + 1, N\},$$

where $N \in \mathbb{N}$ which is originally used by Toint [48].

Something more about trust region methods can be found in [9, 18, 21, 22, 54].

4. Conclusion

The conjugate gradient methods and trust region methods are very popular now.

Many scientists consider these methods.

Namely, different modifications of these methods are made, with the aim to improve them.

Next, the scientists try to make not only new methods but also whole new classes of methods. For the specific values of the parameters, individual methods are distinguished from these classes. It is always more desirable to make a class of methods instead of individual methods.

Hybrid conjugate gradient methods are made in many different ways; this class of conjugate gradient methods is always actual.

Further, one of the contemporary trends is to use *BFGS* update in constructions of new conjugate gradient methods (e.g., see [46]).

Finally, let us emphasize that contemporary papers often use the Picard-Mann-Ishikawa iterative processes and they make the connection of these kinds of processes with the unconstrained optimization (see [29, 37, 38]).

Author details

Snezana S. Djordjevic
Faculty of Technology, University of Nis, Leskovac, Serbia

*Address all correspondence to: snezanadjordjevicle@gmail.com

References

[1] Adeleke OJ, Osinuga IA. A five-term hybrid conjugate gradient method with global convergence and descent properties for unconstrained optimization problems. Asian Journal of Scientific Research. 2018;**11**:185-194

[2] Amini K, Faramarzi P, Pirfalah N. A modified Hestenes-Stiefel conjugate gradient method with an optimal property. Optimization Methods and Software. In press. 2018. DOI: 10.1080/10556788.2018.1457150

[3] Andrei N. An unconstrained optimization test functions. Advanced Modeling and Optimization. An Electronic International Journal. 2008; **10**:147-161

[4] Andrei N. Acceleration of conjugate gradient algorithms for unconstrained optimization. Applied Mathematics and Computation. 2009;**213**:361-369

[5] Andrei N. 40 Conjugate Gradient Algorithms For Unconstrained Optimization. A Survey on Their Definition. ICI Technical Report No. 13/08, March 14, 2008

[6] Andrei N. A new three-term conjugate gradient algorithm for unconstrained optimization. Numerical Algorithms. 2015;**68**:305-321

[7] Baluch B, Salleh Z, Alhawarat A. A new modified three-term Hestenes-Stiefel conjugate gradient method with sufficient descent property and its global convergence. Hindawi Journal of Optimization. 2017;**2017**:1-13

[8] Beale EML. A derivative of conjugate gradients. In: Lootsma FA, editor. Numerical Methods for Nonlinear Optimization. London: Academic; 1972. pp. 39-43

[9] Curtis FE, Lubberts Z, Robinson DP. Concise complexity analyses for trust-region methods. Optimization Letters. 2018;**12**(8):1713-1724

[10] Dai YH, Kou CX. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. SIAM Journal on Optimization. 2013;**23**(1):296-320

[11] Dai Y, Yuan Y. Alternate minimization gradient method. IMA Journal of Numerical Analysis. 2003;**23**: 377-393

[12] Dai YH, Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on Optimization. 1999;**10**: 177-182

[13] Dong X-L, Han D, Dai Z, Li L, Zhu J. An accelerated three-term conjugate gradient method with sufficient descent condition and conjugacy condition. Journal of Optimization Theory and Applications. 2018;**179**:944-961

[14] Du S, Chen M. A new smoothing modified three-term conjugate gradient method for l_1-norm minimization problem. Journal of Inequalities and Applications. 2018;**2018**(1):1-14. SpringerOpen

[15] Djordjević SS. New hybrid conjugate gradient method as a convex combination of FR and PRP methods. Univerzitet u Nišu. 2016;**30**(11): 3083-3100

[16] Djordjević SS. New hybrid conjugate gradient method as a convex combination of LS and CD methods. Univerzitet u Nišu. 2016;**31**(6): 1813-1825

[17] Djordjevic S. New hybrid conjugate gradient method as a convex combination of Ls and Fr methods. Acta Mathematica Scientia. 2019;**39**(1): 214-228

[18] El-Sobky B, Abo-Elnaga Y. A penalty method with trust-region mechanism for nonlinear bilevel optimization problem. Journal of Computational and Applied Mathematics. 2018;**340**:360-374

[19] Fletcher R, Reeves C. Function minimization by conjugate gradients. The Computer Journal. 1964;7:149-154

[20] Fletcher R. Practical methods of optimization. In: Unconstrained Optimization. Vol. 1. New York: John Wiley & Sons; 1987

[21] Gao J, Cao J. A class of derivative-free trust-region methods with interior backtracking technique for nonlinear optimization problems subject to linear inequality constraints. Journal of Inequalities and Applications. 2018; **2018**(1):108; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5942389/

[22] Gertz EM, Gill PE. A primal-dual trust region algorithm for nonlinear optimization. Mathematical Programming, Series B. 2004;**100**:49-94

[23] Hager WW, Zhang H. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on Optimization. 2003;**16**(1):170-192

[24] Hestenes MR, Stiefel EL. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards. 1952;**49**: 409-436

[25] Huang H, Lin S. A modified Wei-Yao-Liu conjugate gradient method for unconstrained optimization. Applied Mathematics and Computation. 2014; **231**:179-186

[26] Huang H et al. The proof of the sufficient descent condition of the Wei-Yao-Liu conjugate gradient method under the strong Wolfe-Powell line search. Applied Mathematics and Computation. 2007;**189**(2):1241-1245

[27] Jiang Z, Xu N. Hot spot thermal floor plan solver using conjugate gradient to speed up. Mobile Information Systems. 2018;**2018**:1-8

[28] Babaie-Kafaki S, Rezaee S. Two accelerated nonmonotone adaptive trust region line search methods. Numerical Algorithms. 2018;**78**:911-928

[29] Khan SH. A Picard-Mann hybrid iterative process. Fixed Point Theory and Applications. 2013;**2013**:69. DOI: 10.1186/1687-1812-2013-69

[30] Liu JK, Feng YM, Zou LM. Some three-term conjugate gradient methods with the inexact line search condition. Calcolo. 2018;**55**:16

[31] Liu Y, Storey C. Efficient generalized conjugate gradient algorithms, Part 1: Theory. Journal of Optimization Theory and Applications. 1991;**69**:129-137

[32] McGuire MF, Wolfe P. Evaluating a Restart Procedure for Conjugate Gradients, Report RC-4382. Yorktown Heights: IBM Research Center; 1973

[33] Narushima Y, Yabe H, Ford JA. A three-term conjugate gradient method with sufficient descent property for unconstrained optimization. SIAM Journal on Optimization. 2011;**21**: 212-230

[34] Nazareth L. A conjugate direction algorithm without line search. Journal of Optimization Theory and Applications. 1977;**23**:373-387

[35] Osadcha O, Marszaek Z. Comparison of steepest descent method and conjugate gradient method. In: CEUR Workshop Proceedings, SYSTEM 2017 - Proceedings of the Symposium for Young Scientists in Technology, Engineering and Mathematics. 2017; **1853**:1-4

[36] Perry A. Technical Note - A modified conjugate gradient algorithm.

Operations Research. 1978;**26**(6): 1073-1078

[37] Petrovic M, Rakocevic V, Kontrec N, et al. Hybridization of accelerated gradient descent method. Numerical Algorithms. 2018;**79**:769-786. DOI: 10.1007/s11075-017-0460-4

[38] Petrović MJ, Stanimirović PS, Kontrec N, Mladenov J. Hybrid modification of accelerated double direction method. Mathematical Problems in Engineering. 2018;**2018**:1-8

[39] Polak E, Ribiére G. Note sur la convergence de de directions conjugées. In: Rev. Francaise Informat Recherche Opertionelle, 3e Année. Vol. 16. 1969. pp. 35-43

[40] Polyak BT. The conjugate gradient method in extreme problems. USSR Computational Mathematics and Mathematical Physics. 1969;**9**:94-112

[41] Rivaie M, Mamat M, June LW, Mohd I. A new class of nonlinear conjugate gradient coefficient with global convergence properties. Applied Mathematics and Computation. 2012; **218**:11323-11332

[42] Rivaie M, Mamat M, Abashar A. A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches. Applied Mathematics and Computation. 2015; **268**:1152-1163

[43] Schnabel RB, Eskow E. A new modified Cholesky factorization. SIAM Journal on Scientific Computing. 1990; **11**:1136-1158

[44] Sheng Z, Yuan G, CUI Z. A new adaptive trust region algorithm for optimization problems. Acta Mathematica Scientia. 2018;**38B**(2): 479-496

[45] Shengwei Y, Wei Z, Huang H. A note about Wyl's conjugate gradient method and its applications. Applied Mathematics and Computation. 2007; **191**:381-388

[46] Stanimirović PS, Ivanov B, Djordjević S, Brajević I. New hybrid conjugate gradient and Broyden–Fletcher–Goldfarb–Shanno Conjugate Gradient Methods. Journal of Optimization Theory and Applications. 2018;**178**:860-884

[47] Sun W, Yuan Y-X. Optimization theory and methods: Nonlinear programming. Springer Optimization and Its Applications. 2006;**1**

[48] Toint PhL. Nonmonotone trust region algorithms for nonlinear optimization subject to convex constraints. Mathematical Programming. 1997;**77**:69. DOI: 10.1007/BF02614518

[49] Wei Z, Yao S, Liu L. The convergence properties of some new conjugate gradient methods. Applied Mathematics and Computation. 2006; **183**(2):1341-1350

[50] Wolfe P. Convergence conditions for ascent methods. SIAM Review. 1969; **11**:226-235

[51] Wolfe P. Convergence conditions for ascent methods II: Some corrections. SIAM Review. 1969;**11**:226-235

[52] Wu G, Li Y, Yuan G. A three-term conjugate gradient algorithm with quadratic convergence for unconstrained optimization problems. Hindawi, Mathematical Problems in Engineering. 2018, Article ID: 4813030, 15 p. DOI: 10.1155/2018/4813030

[53] Yu GH, Guan LT, Chen WF. Spectral conjugate gradient methods with sufficient descent property for large scale unconstrained optimization. Optimization Methods and Software. 2008;**23**(2):275-293

[54] Zhang X, Zhang J, Liao L. An adaptive trust region method and its

convergence. Science in China, Series A: Mathematics. 2002;**45A**:620-631

[55] Zhang L, Zhou W, Li D. Some descent three-term conjugate gradient methods and their global convergence. Optimization Methods and Software. 2007;**22**(4):697-711

[56] Zhou W, Zhang L. A nonlinear conjugate gradient method based on the *MBFGS* secant condition. Optimization Methods and Software. 2006;**21**(5): 707-714