BULLETIN *of the*
Malaysian Mathematical
Sciences Society
http://math.usm.my/bulletin

Bull. Malays. Math. Sci. Soc. (2) **34**(2) *(2011), 319–330*

Open Problems in Nonlinear Conjugate Gradient Algorithms for Unconstrained Optimization

Neculai Andrei
Research Institute for Informatics, Center for Advanced Modeling and Optimization,
8-10, Averescu Avenue, Bucharest 1, Romania
Academy of Romanian Scientists, 54, Splaiul Independentei, Bucharest 5, Romania
nandrei@ici.ro

Abstract. The paper presents some open problems associated to the nonlinear conjugate gradient algorithms for unconstrained optimization. Mainly, these problems refer to the initial direction, the conjugacy condition, the step length computation, new formula for conjugate gradient parameter computation based on function's values, the influence of accuracy of line search procedure, how we can take the problem's structure on conjugate gradient algorithms, how we can consider the second order information in these algorithms, what the most convenient restart procedure is, what the best hybrid conjugate gradient algorithm is, scaled conjugate gradient algorithms, what the most suitable stopping criterion in conjugate gradient algorithms is, etc.

2010 Mathematics Subject Classification: 49M07, 49M10, 90C06, 65K

Keywords and phrases: Unconstrained optimization, conjugate gradient method, Newton method, quasi-Newton methods.

1. Introduction

The conjugate gradient method represents a major contribution to the panoply of methods for solving large-scale unconstrained optimization problems. They are characterized by low memory requirements and have strong local and global convergence properties. The popularity of these methods is remarkable partially due to their simplicity both in their algebraic expression and in their implementation in computer codes, and partially due to their efficiency in solving large-scale unconstrained optimization problems.

The conjugate gradient method has been devised by Magnus Hestenes (1906–1991) and Eduard Stiefel (1909–1978) in their seminal paper where an algorithm for solving symmetric, positive-definite linear algebraic systems has been presented [41]. After a relatively short period of stagnation, the paper by Reid [55] brought the conjugate gradient method as a main active area of research in unconstrained

Communicated by Lee See Keong.
Received: December 4, 2008; *Revised:* June 15, 2009.

optimization. In 1964 the method has been extended to nonlinear problems by Fletcher and Reeves [35], which is usually considered as the first nonlinear conjugate gradient algorithm. Since then a large number of variants of conjugate gradient algorithms have been suggested. A survey on their definition including 40 nonlinear conjugate gradient algorithms for unconstrained optimization is given by Andrei [13]. Even if the conjugate gradient methods are now over 50 years old, they continue to be of a considerable interest particularly due to their convergence properties, a very easy implementation effort in computer programs and due to their efficiency in solving large-scale problems. For general unconstrained optimization problem:

$$\min_{x\in R^n} f(x), \tag{1.1}$$

where $f : R^n \to R$ is a continuously differentiable function, bounded from below, starting from an initial guess, a nonlinear conjugate gradient algorithm generates a sequence of points $\{x_k\}$, according to the following recurrence formula:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

where α_k is the step length, usually obtained by the Wolfe line search,

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho\alpha_k g_k^T d_k, \tag{1.3}$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k, \tag{1.4}$$

with $0 < \rho < 1/2 \le \sigma < 1$, and the directions d_k are computed as:

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad d_0 = -g_0. \tag{1.5}$$

Here β_k is a scalar known as the conjugate gradient parameter, $g_k = \nabla f(x_k)$ and $s_k = x_{k+1} - x_k$. In the following $y_k = g_{k+1} - g_k$. Different conjugate gradient algorithms correspond to different choices for the parameter β_k. Therefore, a crucial element in any conjugate gradient algorithm is the formula definition of β_k. Any conjugate gradient algorithm has a very simple general structure as illustrated below.

Table 1. The prototype of Conjugate Gradient Algorithm

Step 1	Select the initial starting point $x_0 \in dom\, f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Set for example $d_0 = -g_0$ and $k = 0$.
Step 2	Test a criterion for stopping the iterations. For example, if $\|g_k\|_\infty \le \varepsilon$, then stop; otherwise continue with step 3.
Step 3	Determine the step length α_k.
Step 4	Update the variables as: $x_{k+1} = x_k + \alpha_k d_k$. Compute f_{k+1} and g_{k+1}. Compute $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.
Step 5	Determine β_k.
Step 6	Compute the search direction as: $d_{k+1} = -g_{k+1} + \beta_k s_k$.
Step 7	Restart criterion. For example, if the restart criterion of Powell $\left\|g_{k+1}^T g_k\right\| > 0.2\,\|g_{k+1}\|^2$ is satisfied, then set $d_{k+1} = -g_{k+1}$.
Step 8	Compute the initial guess $\alpha_k = \alpha_{k-1}\,\|d_{k-1}\| / \|d_k\|$, set $k = k + 1$ and continue with step 2.

This is a prototype of the conjugate gradient algorithm, but some more sophisticated variants are also known (CONMIN [56, 57], SCALCG [2–5], ASCALCG [12], ACGHES [19], ACGMSEC [11], CG_DESCENT [39, 40]). These variants focus on parameter β_k computation and on the step length determination.

2. The open problems

In the following we shall present some open problems in conjugate gradient algorithms. These problems refer to the initial direction selection, to the conjugacy condition, to the step length computation, new formula for conjugate parameter computation based on function's values, the influence of accuracy of line search procedure on the efficiency of conjugate gradient algorithm, how we can consider the problem's structure on conjugate gradient algorithms, how we can take the second order information in these algorithms, what the best restart procedure is, what the best hybrid conjugate gradient algorithm is, scaled conjugate gradient algorithms, what the best stopping criterion in conjugate gradient algorithms is, how these algorithms can be modified for solving simple bounded optimization problems etc.

Problem 1. *Why is the initial search direction $d_0 = -g_0$ critical?*
Crowder and Wolfe [28] presented a 3-dimensional strongly convex quadratic example showing that if the initial search direction is not the steepest descent, then the convergence rate of conjugate gradient is linear. On the other hand, Beale [24] showed that if

$$d_{k+1} = -g_{k+1} + \frac{y_0^T g_{k+1}}{y_0^T d_0} d_0 + \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} d_k \tag{2.1}$$

then if $d_0 \neq -g_0$, then conjugate directions are still obtained. This approach given by (2.1) allows a set of conjugate directions to be generated starting from any initial direction d_0. However, since d_0 remains in the formula for d_{k+1} along the iterations, it may be undesirable [33].

Later, Powell [53] showed that if $f(x)$ is a convex quadratic function, then using an arbitrary initial search direction d_0 the solution is obtained at a linear rate of convergence. Nazareth [47] suggested a conjugate gradient algorithm with a complicated three-term recurrence for d_{k+1} as

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_{k-1}} d_{k-1}, \tag{2.2}$$

and $d_0 = 0$. In this form, apart from a scalar multiplier, the new direction given by (2.2) does not depend on the step length. He proved that if $f(x)$ is a convex quadratic, then for any step length α_k the search directions are conjugate relatively to the Hessian of f. However, if $d_0 \neq -g_0$, then d_k can become zero away from the minimum. Although interesting, this innovation has not been profitable in practice. An alternative way of allowing an arbitrary initial direction d_0 for quadratic functions was suggested by Allwright [1] who introduced a change of variable based on a factorization of the Hessian of the function f. Observe that all these remarks address only to the convex quadratic functions; for the general nonlinear function we have

no results on this problem.

Problem 2. *What is the best conjugacy condition?*
The conjugacy condition is expressed as $y_k^T d_{k+1} = 0$. Recently, Dai and Liao [29] introduced the new conjugacy condition $y_k^T d_{k+1} = -ts_k^T g_{k+1}$, where $t \geq 0$ is a scalar. This is indeed very reasonable since in real computation the inexact line search is generally used. However, this condition is very dependent on the nonnegative parameter t, for which we do not know any formula to choose in an optimal manner.

Problem 3. *Why does the sequence of step length* $\{\alpha_k\}$*tend to vary in a totally unpredictable manner and differ from 1 by two order of magnitude?*
Intensive numerical experiments with different variants of conjugate gradient algorithms proved that the step length may differ from 1 up to two orders of magnitude, being larger or smaller than 1, depending on how the problem is scaled. Moreover, the sizes of the step length tend to vary in a totally unpredictable way. This is in sharp contrast with the Newton and quasi-Newton methods, as well as with the limited memory quasi-Newton methods, which usually admit the unit step length for most of the iterations, thus requiring only very few function evaluations for step length determination. Numerical experiments with the limited memory quasi Newton method by Liu and Nocedal [45] show that it is successful [10, 21]. One explanation of the efficiency of the limited memory quasi-Newton method is given by its ability to accept unity step lengths along the iterations.

In an attempt to take the advantage of this behavior of conjugate gradient algorithms Andrei [14, 15] suggested an acceleration procedure by modifying the step length α_k (computed by means of the Wolfe line search conditions) through a positive parameter η_k, in a multiplicative manner, like $x_{k+1} = x_k + \eta_k \alpha_k d_k$, in such a way as to improve the reduction of the function's values along the iterations. It is shown that the acceleration scheme is linear convergent, but the reduction in function value is significantly improved. Intensive numerical comparisons with different accelerated conjugate gradient algorithms are documented in [10,15]. An acceleration of the gradient descent algorithm with backtracking for unconstrained optimization is given in [9].

Problem 4. *What is the influence of the accuracy of line search procedure on the performances of conjugate gradient algorithms?*
For any unconstrained optimization algorithm one of the crucial elements is the stepsize computation. Many procedures have been suggested. In the *exact line search* the step α_k is selected as:

$$\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k), \tag{2.3}$$

where d_k is a descent direction. In some very special cases (quadratic problems, for example) it is possible to compute the step α_k analytically, but for the vast majority of cases it is computed to approximately minimize f along the ray $\{x_k + \alpha d_k : \alpha \geq 0\}$, or at least to reduce f sufficiently. In practice the most used are the *inexact procedures*. A lot of inexact line search procedures have been proposed: Goldstein [37], Armijo [23], Wolfe [61], Powell [52], Dennis and Schnabel [32], Potra and Shi [51],

Lemaréchal [44], Moré and Thuente [46], Hager and Zhang [39], and many others. The most used is based on the Wolfe line search conditions (1.3) and (1.4). An important contribution in understanding the behavior of Wolfe conditions was given by Hager and Zhang [39, 40] by introducing the approximate Wolfe conditions

$$(2\rho - 1)g_k^T d_k \geq g_{k+1}^T d_k \geq \sigma g_k^T d_k. \tag{2.4}$$

The first inequality in (2.4) is an approximation to the first Wolfe condition (1.3). When the iterates are near a local optimum this approximation can be evaluated with greater accuracy than the original condition, since the approximate Wolfe conditions are expressed in terms of a derivative, not as the difference of function values. It is worth saying that the first Wolfe condition (1.3) limits the accuracy of a conjugate gradient algorithm to the order of the square root of the machine precision, while the approximate Wolfe conditions (2.4) achieve accuracy on the order of the machine precision [39].

It seems that the higher accuracy of the step length, the faster convergence of a conjugate gradient algorithm. For example the CG_DESCENT algorithm by Hager and Zhang which implement (2.4) is the fastest known conjugate gradient variant.

In this context another interesting open question is whether the non-monotone line search [38] is more effective than the Wolfe line search.

Another open problem, more interesting, is to design conjugate gradient algorithms without line search, the idea being to save computation. Such conjugate gradient algorithms could be faster because there is no loss of accuracy related to checking the Wolfe conditions.

Problem 5. *How can we use the function values in β_k to generate new conjugate gradient algorithms?*

This problem is taken from Yuan [63]. Generally, in conjugate gradient algorithms the parameter β_k is computed using $\|g_k\|$, $\|g_{k+1}\|$, $\|y_k\|$, $\|s_k\|$, $y_k^T s_k$, $g_k^T g_{k+1}$, $y_k^T g_{k+1}$ and $s_k^T g_{k+1}$ [6,13]. As we can see in the formula for β_k the difference $f(x_k) - f(x_{k+1})$ is not used at all. In [62] Yabe and Takano, using a result of Zhang, Deng and Chen [66], suggest the following formula for β_k

$$\beta_k^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}, \tag{2.5}$$

where $z_k = y_k + \frac{\delta \eta_k}{s_k^T u_u} u_k$, $\eta_k = 6(f(x_k) - f(x_{k+1})) + 3(g_k + g_{k+1})^T s_k$, $\delta > 0$ is a constant and $u_k \in R^n$ satisfies $s_k^T u_k \neq 0$; for example $u_k = s_k$. In the same context based on the modified secant condition of Zhang, Deng and Chen [66], with $u_k = s_k$, Andrei [11] proposed the following formula for β_k

$$\beta_k = \left(\frac{\delta \eta_k}{\|s_k\|^2} - 1 \right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k} + \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k}, \tag{2.6}$$

where $\delta \geq 0$ is a scalar parameter. Another possibility is presented by Yuan [63] as

$$\beta_k^Y = \frac{y_k^T g_{k+1}}{(f_k - f_{k+1})/\alpha_k - d_k^T g_k / 2}. \tag{2.7}$$

Problem 6. *Can we take advantage of problem structure to design more effective nonlinear conjugate gradient algorithms?*

This problem was formulated by Nocedal [48]. When the problem is partially separable, i.e. it can be expressed as a sum of element functions, each of which does have a large invariant subspace [26], can we formulate a partitioned updating of parameter β_k to obtain a powerful conjugate gradient algorithm? This idea of decomposition of partially separable functions in the context of large-scale optimization was considered in quasi-Newton methods by Conn, Gould and Toint [27]. The advantage of this approach is that the information contained in the partially separable description of the function is so detailed that it can be used in exploring the objective function only along some relevant directions. The idea is to ignore some invariant subspace of the function and only consider its complement. The question is whether we can use this type of invariant subspace information to design new formula forβ_k.

Problem 7. *How can we consider the second order information in conjugate gradient algorithms?*

In [3, 4] Andrei suggested the following formula for β_k:

$$\beta_k = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k}. \tag{2.8}$$

Observe that if the line search is exact, then we get the Daniel method [31]. The salient point with this formula for β_k computation is the presence of the Hessian matrix. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian.

A direct possibility to use the second order information given by the Hessian matrix is to compute the Hessian/vector product $\nabla^2 f(x_{k+1}) s_k$. However, our numerical experiments proved that even though the Hessian is partially separable (block diagonal) or it is a multi-diagonal matrix, the Hessian/vector product $\nabla^2 f(x_{k+1}) s_k$ is time consuming, especially for large-scale problems. Besides, what happens when $s_k \in Ker \nabla^2 f(x_{k+1})$? In an effort to use the Hessian in β_k Andrei [19] suggested a nonlinear conjugate gradient algorithm in which the Hessian/vector product $\nabla^2 f(x_{k+1}) s_k$ is approximated by finite differences:

$$\nabla^2 f(x_{k+1}) s_k = \frac{\nabla f(x_{k+1} + \delta s_k) - \nabla f(x_{k+1})}{\delta}, \tag{2.9}$$

where

$$\delta = \frac{2\sqrt{\varepsilon_m}(1 + \|x_{k+1}\|)}{\|s_k\|}, \tag{2.10}$$

and ε_m is epsilon machine.

As we know, for quasi-Newton methods an approximation matrix B_k to the Hessian $\nabla^2 f(x_k)$ is used and updated so that the new matrix B_{k+1} satisfies the secant condition $B_{k+1} s_k = y_k$. Therefore, as it is explained in [3–5] in order to have an algorithm for solving large-scale problems we can assume that the pair (s_k, y_k) satisfies

the secant condition. Using this assumption we get:

$$\beta_k = \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{y_k^T s_k}, \tag{2.11}$$

where $\theta_{k+1}$is a parameter. Birgin and Martínez [25] arrived at the same formula for β_k, but using a geometric interpretation of quadratic function minimization.

Further in [11] we experienced another nonlinear conjugate gradient algorithm in which the Hessian/vector product $\nabla^2 f(x_{k+1}) s_k$ is approximated by the modified secant condition introduced by Zhang, Deng and Chen [66] and by Zhang and Xu [67], obtaining β_k as in (2.6).

Problem 8. *What is the best scaled conjugate gradient algorithm?*
This is the preconditioning of conjugate gradient algorithms, which is a very active area. Some authors suggested the search direction of the following form

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k, \tag{2.12}$$

where θ_{k+1} is a positive scalar or a symmetric and positive definite matrix [2, 25]. The formula (2.12) is known as the scaled conjugate gradient algorithm. Observe that if $\theta_{k+1} = 1$, then we get the classical conjugate gradient algorithms according to the value of the scalar parameter β_k. On the other hand, if $\beta_k = 0$, then we get another class of algorithms according to the selection of the parameter θ_{k+1}. Considering $\beta_k = 0$, there are two possibilities for θ_{k+1}: a positive scalar or a positive definite matrix. If $\theta_{k+1} = 1$, then we have the steepest descent algorithm. If $\theta_{k+1} = \nabla^2 f(x_{k+1})^{-1}$, or an approximation of it, then we get the Newton or the quasi-Newton algorithms, respectively. Therefore, we see that in the general case, when $\theta_{k+1} \neq 0$is selected in a quasi-Newton manner, and $\beta_k \neq 0$, then (2.12) represents a combination between the quasi-Newton and the conjugate gradient methods. However, if $\theta_{k+1}$is a matrix containing some useful information about the inverse Hessian of function f, we are better off using $d_{k+1} = -\theta_{k+1} g_{k+1}$ since the addition of the term $\beta_k s_k$in (2.12) may prevent the direction d_{k+1} from being a descent direction unless the line search is sufficiently accurate. In [2, 25] θ_{k+1} is selected as the inverse of the Rayleigh quotient. Another selection based on the values of the minimizing function in two successive points is presented in [2, 5]. A diagonal Hessian preconditioner is considered by Fessler and Booth [34]. For linear conjugate gradient see [43].

Problem 9. *Which is the best hybrid conjugate gradient algorithm?*
Hybrid conjugate gradient algorithms have been devised to use and combine the attractive features of the classical conjugate gradient algorithms. Touati-Ahmed and Storey [58], Hu and Storey [42], Gilbert and Nocedal [36] suggested hybrid conjugate gradient algorithms using projections of Fletcher-Reeves [35], Polak-Ribière [49] and Polyak [50] conjugate gradient algorithms. Another source of hybrid conjugate gradient algorithms is based on the concept of convex combination of classical conjugate gradient algorithms. Thus in [7, 8, 20] Andrei introduced a new class of the hybrid conjugate gradient algorithm based on a convex combination of Hestenes-Stiefel [41] and Dai-Yuan [30]. In [16] other hybrid conjugate gradient algorithms are designed

as convex combination of Polak-Ribière-Polyak [49, 50], and Dai-Yuan [30]. Generally, the performance of the hybrid variants based on the concept of convex combination is better than that of the constituents [17, 18]. Some other variants are considered in [64,65]. New nonlinear conjugate gradient formulas for unconstrained optimization, including the global convergence of the corresponding algorithms are given in [59,60]. But, finding the best convex combination of the classical conjugate gradient algorithms remains for further study.

Problem 10. *What is the most convenient restart procedure of conjugate gradient algorithms?*
In the early conjugate gradient algorithms, the restarting strategy was usually to restart whenever $k = n$ or $k = n + 1$. When n is very large and the number of clusters of similar eigenvalues of the Hessian is very small, this strategy can be very inefficient. Powell [54] has suggested restarting whenever

$$\left|g_k^T g_{k+1}\right| \geq 0.2 \left\|g_{k+1}\right\|^2. \tag{2.13}$$

On quadratic functions the left-hand side of (2.13) is an indicator of the nonconjugacy of the search directions and therefore a signal that the current cycle must be terminated and another one must be started with negative of the current gradient. It is also desirable to restart if the direction is not effectively downhill. Powell suggested restarting if

$$-1.2 \left\|g_k\right\|^2 < d_k^T g_k < -0.8 \left\|g_k\right\|^2 \tag{2.14}$$

is not satisfied. Another criterion for restarting the iterations in conjugate gradient algorithms was designed by Birgin and Martínez [25]

$$d_{k+1}^T g_{k+1} > -10^{-3} \left\|d_{k+1}\right\|_2 \left\|g_{k+1}\right\|_2. \tag{2.15}$$

In (2.15) when the angle between d_{k+1} and $-g_{k+1}$ is not acute enough then restart the algorithm with $-g_{k+1}$. Clearly, more sophisticated restarting procedures can be imagined, but which one is the best remains to be seen.

Problem 11. *What is the most suitable criterion for stopping the conjugate gradient iterations?*
In infinite precision, a necessary condition for x^* to be the exact minimizer of function f is $\nabla f(x^*) = 0$. In an iterative and finite precision algorithm, we must modify this condition as $\nabla f(x^*) \cong 0$. Although $\nabla f(x^*) = 0$ can also occur at a maximum or at a saddle point, the line search strategy makes the convergence of the algorithm virtually impossible to maxima or saddle points. Therefore, $\nabla f(x^*) = 0$ is considered a necessary and sufficient condition for x^* to be a local minimizer of f.

For linear conjugate gradient algorithms different stopping criteria were analyzed by Arioli and Loghin [22]. For nonlinear conjugate gradient algorithms the following stopping criteria were suggested

$$\left\|\nabla f(x_k)\right\|_\infty \leq \varepsilon_g, \tag{2.16}$$

$$\alpha_k g_k^T d_k \leq \varepsilon_f \left|f(x_{k+1})\right|, \tag{2.17}$$

$$\left\|\nabla f(x_k)\right\|_\infty \leq \varepsilon_g (1 + \left|f(x_k)\right|), \tag{2.18}$$

$$\|\nabla f(x_k)\|_\infty \leq \max\{\varepsilon_g, \varepsilon_0 \|\nabla f(x_0)\|_\infty\}, \tag{2.19}$$

$$\|\nabla f(x_k)\|_2 \leq \varepsilon_g, \tag{2.20}$$

where, for example $\varepsilon_f = 10^{-20}$, $\varepsilon_g = 10^{-6}$ and $\varepsilon_0 = 10^{-12}$. For large-scale problems $\|\nabla f(x_k)\|_\infty$is more suitable to be used to stop the algorithm, but for small problems it is better to use $\|\nabla f(x_k)\|_2$.

Problem 12. *Affine components of the gradient.*
The Newton method has a very nice property. If any component functions of the gradient $\nabla f(x)$ are affine, then each iterate generated by the Newton method will be a solution of these components, since the affine model associated to the system $\nabla f(x) = 0$ will always be exact for these functions. Is there an equivalent property for conjugate gradient algorithms?

Problem 13. *What is the interrelationship between conjugate gradient and quasi-Newton algorithms, including here the limited memory quasi-Newton algorithms?*
Both these algorithms have some maturity with very well established theoretical results and strong computational experience. The question is that we don't have any significant progress in designing efficient and robust algorithms for large-scale problems using concepts from both these two classes of algorithms.

Problem 14. *Can the nonlinear conjugate gradient algorithms be extended to solve simple bounded constrained optimization?*
Consider the problem

$$\min_{x \in R^n} \{f(x) \,|\, l \leq x \leq u\}, \tag{2.21}$$

where l and u are known vectors from R^n. How can we adapt the conjugate gradient algorithms to solve equation (2.21)? A possible idea is to consider the techniques from the interior point methods and devise a nonlinear conjugate gradient algorithm in which the bounds on variables are not dealt with explicitly [48].

Conclusion

For more than 50 years the conjugate gradient algorithms have been under an intensive theoretical and computational analysis. Today, they represent an important component of optimization algorithms. In this paper we have presented some interesting open problems concerning the design and implementation in computing codes of nonlinear conjugate gradient algorithms.

References

[1] J. C. Allwright, Improving the conditioning of optimal control problems using simple methods, In O. J. Bell (Ed.), *Recent Mathematical Developments in Control*, Academic Press, London, 1972.

[2] N. Andrei, Scaled conjugate gradient algorithms for unconstrained optimization, *Comput. Optim. Appl.* **38** (2007), no. 3, 401–416.

[3] N. Andrei, Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, *Optim. Methods Softw.* **22** (2007), no. 4, 561–571.

[4] N. Andrei, A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, *Appl. Math. Lett.* **20** (2007), no. 6, 645–650.
[5] N. Andrei, A scaled nonlinear conjugate gradient algorithm for unconstrained optimization, *Optimization* **57** (2008), no. 4, 549–570.
[6] N. Andrei, A Dai-Yuan conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization, *Appl. Math. Lett.* **21** (2008), no. 2, 165–171.
[7] N. Andrei, A hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization, *ICI Technical Report*, February 6, 2008.
[8] N. Andrei, A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan, *Studies in Informatics and Control* **17** (2008), 55–70.
[9] N. Andrei, An acceleration of gradient descent algorithm with backtracking for unconstrained optimization, *Numer. Algorithms* **42** (2006), no. 1, 63–73.
[10] N. Andrei, Numerical comparison of conjugate gradient algorithms for unconstrained optimization, *Studies in Informatics and Control* **16** (2007), no. 4, 333–352.
[11] N. Andrei, Accelerated conjugate gradient algorithm with modified secant condition for unconstrained optimization, *ICI Technical Report*, March 3, 2008.
[12] N. Andrei, Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, *ICI Technical Report*, March 24, 2008.
[13] N. Andrei, 40 conjugate gradient algorithms for unconstrained optimization. A survey on their definition. *ICI Technical Report No. 13/08*, March 14, 2008.
[14] N. Andrei, Acceleration of conjugate gradient algorithms for unconstrained optimization, *Appl. Math. Comput.* **213** (2009), no. 2, 361–369.
[15] N. Andrei, New accelerated conjugate gradient algorithms for unconstrained optimization, *ICI Technical Report*, June 18, 2008.
[16] N. Andrei, New hybrid conjugate gradient algorithms as a convex combination of PRP and DY for unconstrained optimization, *ICI Technical Report*, October 1, 2007.
[17] N. Andrei, Hybrid conjugate gradient algorithm for unconstrained optimization, *J. Optim. Theory Appl.* **141** (2009), no. 2, 249–264.
[18] N. Andrei, Another nonlinear conjugate gradient algorithm for unconstrained optimization, *Optim. Methods Softw.* **24** (2009), no. 1, 89–104.
[19] N. Andrei, Accelerated conjugate gradient algorithm with finite difference Hessian/vector product approximation for unconstrained optimization, *J. Comput. Appl. Math.* **230** (2009), no. 2, 570–582.
[20] N. Andrei, Hybrid conjugate gradient algorithm for unconstrained optimization, *J. Optim. Theory Appl.* **141** (2009), no. 2, 249–264.
[21] N. Andrei, Performance profiles of conjugate gradient algorithms for unconstrained optimization. Encyclopedia of Optimization, second edition, C. A. Floudas and P. Pardalos (Eds.), Springer Science + Business Media, New York, 2009, vol. P, pp. 2938–2953.
[22] M. Arioli and D. Loghin, Stopping criteria for mixed finite element problems, *Electron. Trans. Numer. Anal.* **29** (2007/08), 178–192.
[23] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific J. Math.* **16** (1966), 1–3.
[24] E. M. L. Beale, A derivation of conjugate gradients, in *Numerical Methods for Non-linear Optimization* (*Conf., Univ. Dundee, Dundee, 1971*), 39–43, Academic Press, London.
[25] E. G. Birgin and J. M. Martínez, A spectral conjugate gradient method for unconstrained optimization, *Appl. Math. Optim.* **43** (2001), no. 2, 117–128.
[26] A. R. Conn, N. Gould and P. L. Toint, Improving the decomposition of partially separable functions in the context of large-scale optimization: a first approach, in *Large scale optimization* (*Gainesville, FL, 1993*), 82–94, Kluwer Acad. Publ., Dordrecht.
[27] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *LANCELOT*, Springer Series in Computational Mathematics, 17, Springer, Berlin, 1992.
[28] H. Crowder and P. Wolfe, Linear convergence of the conjugate gradient method, *IBM J. Res. Develop.* **16** (1972), 431–433.

[29] Y.-H. Dai and L.-Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.* **43** (2001), no. 1, 87–101.
[30] Y.-H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.* **10** (1999), no. 1, 177–182 (electronic).
[31] J. W. Daniel, The conjugate gradient method for linear and nonlinear operator equations, *SIAM J. Numer. Anal.* **4** (1967), 10–26.
[32] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ, 1983.
[33] L. C. W. Dixon, P. G. Ducksbury and P. Singh, A new three-term conjugate gradient method, *J. Optim. Theory Appl.* **47** (1985), no. 3, 285–300.
[34] J. A. Fessler and S. D. Booth, Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction, *IEEE Trans. Image Process.* **8** (1999), no. 5, 688–699.
[35] R. Fletcher and C. M. Reeves, Function minimization by conjugate gradients, *Comput. J.* **7** (1964), 149–154.
[36] J. C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optim.* **2** (1992), no. 1, 21–42.
[37] A. A. Goldstein, On steepest descent, *J. Soc. Indust. Appl. Math. Ser. A Control* **3** (1965), 147–151.
[38] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.* **23** (1986), no. 4, 707–716.
[39] W. W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM J. Optim.* **16** (2005), no. 1, 170–192.
[40] W. W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.* **2** (2006), no. 1, 35–58.
[41] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Research Nat. Bur. Standards* **49** (1952), 409–436 (1953).
[42] Y. F. Hu and C. Storey, Global convergence result for conjugate gradient methods, *J. Optim. Theory Appl.* **71** (1991), no. 2, 399–405.
[43] A. V. Knyazev and I. Lashuk, Steepest descent and conjugate gradient methods with variable preconditioning, *SIAM J. Matrix Anal. Appl.* **29** (2007), no. 4, 1267–1280.
[44] C. Lemarechal, A view of line-searches, in *Optimization and Optimal Control* (*Proc. Conf., Math. Res. Inst., Oberwolfach, 1980*), 59–78, Lecture Notes in Control and Information Sci., 30, Springer, Berlin.
[45] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Programming* **45** (1989), no. 3, (Ser. B), 503–528.
[46] J. J. Moré and D. J. Thuente, Line search algorithms with guaranteed sufficient decrease, *ACM Trans. Math. Software* **20** (1994), no. 3, 286–307.
[47] L. Nazareth, A conjugate direction algorithm without line searches, *J. Optimization Theory Appl.* **23** (1977), no. 3, 373–387.
[48] J. Nocedal, Conjugate gradient methods and nonlinear optimization, in *Linear and Nonlinear Conjugate Gradient-related Methods* (*Seattle, WA, 1995*), 9–23, SIAM, Philadelphia, PA.
[49] E. Polak and G. Ribière, Note sur la convergence de directions conjuguée, *Rev. Francaise Informat Recherche Operationelle, 3e Année* **16** (1969), 35–43.
[50] B. T. Polyak, The conjugate gradient method in extreme problems, *USSR Comp. Math. Math. Phys.* **9** (1969), 94–112.
[51] F. A. Potra and Y. Shi, Efficient line search algorithm for unconstrained optimization, *J. Optim. Theory Appl.* **85** (1995), no. 3, 677–704.
[52] M. J. D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in *Nonlinear Programming* (*Proc. Sympos., New York, 1975*), 53–72. SIAM-AMS Proc., IX, Amer. Math. Soc., Providence, RI.
[53] M. J. D. Powell, Some convergence properties of the conjugate gradient method, *Math. Programming* **11** (1976), no. 1, 42–49.
[54] M. J. D. Powell, Restart procedures for the conjugate gradient method, *Math. Programming* **12** (1977), no. 2, 241–254.

[55] J. K. Reid, On the method of conjugate gradients for the solution of large sparse systems of linear equations, in *Large Sparse Sets of Linear Equations* (*Proc. Conf., St. Catherine's Coll., Oxford, 1970*), 231–254, Academic Press, London.

[56] D. F. Shanno, Conjugate gradient methods with inexact searches, *Math. Oper. Res.* **3** (1978), no. 3, 244–256.

[57] D. F. Shanno, K. H. Phua, Algorithm 500, Minimization of unconstrained multivariate functions, *ACM Trans. on Math. Software* **2** (1976), 87–94.

[58] D. Touati-Ahmed and C. Storey, Efficient hybrid conjugate gradient techniques, *J. Optim. Theory Appl.* **64** (1990), no. 2, 379–397.

[59] Z. Wei, G. Li and L. Qi, New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems, *Appl. Math. Comput.* **179** (2006), no. 2, 407–430.

[60] Z. X. Wei, G. Y. Li and L. Q. Qi, Global convergence of the Polak-Ribière-Polyak conjugate gradient method with an Armijo-type inexact line search for nonconvex unconstrained optimization problems, *Math. Comp.* **77** (2008), no. 264, 2173–2193.

[61] P. Wolfe, Convergence conditions for ascent methods, SIAM Rev. **11** (1969), 226–235.

[62] H. Yabe and M. Takano, Global convergence properties of nonlinear conjugate gradient methods with modified secant condition, *Comput. Optim. Appl.* **28** (2004), no. 2, 203–225.

[63] Y. Yuan, Some problems in nonlinear programming, *Numerical Linear Algebra and Optimization*, Science Press, Beijing/New York, 2003, pp. 90.

[64] G. Yuan and X. Lu, A modified PRP conjugate gradient method, *Ann. Oper. Res.* **166** (2009), 73–90.

[65] G. Yuan, Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems, *Optimization Letters* (DOI:10.1007/s11590-008-0086-5).

[66] J. Z. Zhang, N. Y. Deng and L. H. Chen, New quasi-Newton equation and related methods for unconstrained optimization, *J. Optim. Theory Appl.* **102** (1999), no. 1, 147–167.

[67] J. Zhang and C. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, *J. Comput. Appl. Math.* **137** (2001), no. 2, 269–278.