Another Conjugate Gradient Algorithm with Guaranteed Descent and Conjugacy Conditions for Large-scale Unconstrained Optimization

Neculai Andrei

Journal of Optimization Theory and Applications

ISSN 0022-3239

J Optim Theory Appl
DOI 10.1007/s10957-013-0285-9

Springer

J Optim Theory Appl
DOI 10.1007/s10957-013-0285-9

Another Conjugate Gradient Algorithm with Guaranteed Descent and Conjugacy Conditions for Large-scale Unconstrained Optimization

Neculai Andrei

Received: 27 April 2012 / Accepted: 13 February 2013

Abstract In this paper, we suggest another accelerated conjugate gradient algorithm for which both the descent and the conjugacy conditions are guaranteed. The search direction is selected as a linear combination of the gradient and the previous direction. The coefficients in this linear combination are selected in such a way that both the descent and the conjugacy condition are satisfied at every iteration. The algorithm introduces the modified Wolfe line search, in which the parameter in the second Wolfe condition is modified at every iteration. It is shown that both for uniformly convex functions and for general nonlinear functions, the algorithm with strong Wolfe line search generates directions bounded away from infinity. The algorithm uses an acceleration scheme modifying the step length in such a manner as to improve the reduction of the function values along the iterations. Numerical comparisons with some conjugate gradient algorithms using a set of 75 unconstrained optimization problems with different dimensions show that the computational scheme outperforms the known conjugate gradient algorithms like Hestenes and Stiefel; Polak, Ribière and Polyak; Dai and Yuan or the hybrid Dai and Yuan; CG_DESCENT with Wolfe line search, as well as the quasi-Newton L-BFGS.

Keywords Conjugate gradient · Wolfe line search · Descent condition · Conjugacy condition · Unconstrained optimization

N. Andrei (✉)
Research Institute for Informatics, Center for Advanced Modeling and Optimization, 8-10, Averescu Avenue, Bucharest 1, Romania
e-mail: nandrei@ici.ro

N. Andrei
Academy of Romanian Scientists, Splaiul Independenţei, Nr. 54, Sector 5, Bucharest, Romania

Published online: 21 February 2013

1 Introduction

Conjugate gradient algorithm represents an important computational innovation for continuously differentiable large-scale nonlinear unconstrained optimization, with strong local and global convergence properties and modest memory requirements. A history of these algorithms has been given by Golub and O'Leary [1], as well as by O'Leary [2]. An excellent survey of development of different versions of nonlinear conjugate gradient methods, with special attention to global convergence properties, is presented by Hager and Zhang [3]. This family of algorithms includes a lot of variants, well known in the literature, with important convergence properties and numerical efficiency. Different from the Newton or quasi-Newton methods, the descent condition plays a crucial role in convergence of the conjugate gradient algorithms. The searching directions in conjugate gradient algorithms are selected in such a way that, when applied to minimize a strongly quadratic convex function, two successive directions are conjugate, subject to the Hessian of the quadratic function. Therefore, to minimize a convex quadratic function in a subspace spanned by a set of mutually conjugate directions is equivalent to minimize this function along each conjugate direction in turn. This is a very good idea, but the performance of these algorithms is dependent on the accuracy of the line search. When applied to general nonlinear functions, often, the searching directions in conjugate gradient algorithms are computed using some formulas which do not satisfy the conjugacy condition. However, by extension we call them conjugate gradient algorithms.

In this paper, we propose a new nonlinear conjugate gradient algorithm where, at every iteration, both the descent and the conjugacy conditions are satisfied, independent by the line search. The structure of the paper is as follows. Section 2 contains some preliminaries. The search direction, presented in Sect. 3, is selected as a linear combination of the negative gradient and the previous searching direction, where the coefficients in this linear combination are selected in such a way that both the descent and the conjugacy condition are satisfied. In Sect. 4, the modified Wolfe line search conditions are introduced. Mainly the second Wolfe condition is modified by changing its parameter, at each iteration, through a specified formula. Some properties of the algorithm are presented in Sect. 5. The acceleration scheme of the algorithm is described in Sect. 6. The idea of this computational scheme is to take advantage that the step lengths in conjugate gradient algorithms are very different from 1. Therefore, we suggest modifying the step length in such a manner as to improve the reduction of the function values along the iterations. Section 7 is devoted to presentation of the algorithm. In Sect. 8, we prove the convergence of the algorithm. It is shown that both for uniformly convex functions and for general nonlinear functions, the corresponding algorithm with modified strong Wolfe line search generates directions bounded away from infinity. In Sect. 9, some numerical experiments and performance profiles of Dolan–Moré [4] corresponding to this new conjugate gradient algorithm are given. The performance profiles correspond to a set of 75 unconstrained optimization problems presented in [5]. Each problem was tested 10 times, for a gradually increasing number of variables: 1000, 2000, ..., 10000. It is shown that this new conjugate gradient algorithm outperforms the classical Hestenes and Stiefel [6], Dai and Yuan [7], Polak, Ribière and Polyak [8, 9], hybrid Dai and Yuan [7] (hDY) conjugate gradient algorithms, the CG_DESCENT conjugate gradient algorithm with Wolfe line search [10] and also L-BFGS [11]. To see the performances of the algorithm, in Sect. 10, a sensitivity study subject to variation of scalar parameters in linear combination defining the searching direction is presented. Numerical experiments prove that the algorithm is very little sensitive to the variation of these parameters. Lastly, in Sect. 11, a comparison between our algorithm and CG_DESCENT on some applications from MINPACK-2 test problems collection [12] is illustrated. All these various numerical experiments show that our algorithm is one of the fastest and more robust conjugate gradient algorithms.

2 Preliminaries

For solving large-scale unconstrained optimization problems

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

where $f:\mathbb{R}^n\to\mathbb{R}$ is a continuously differentiable function, bounded from below, one of the most elegant and probably the simplest is the conjugate gradient method. For solving this problem, starting from an initial guess $x_0\in\mathbb{R}^n$, a nonlinear conjugate gradient method generates a sequence $\{x_k\}$ as:

$$x_{k+1}=x_k+\alpha_k d_k, \tag{2}$$

where $\alpha_k>0$ is obtained by line search, and the directions d_k are generated as:

$$d_{k+1}=-g_{k+1}+\beta_k d_k,\, d_0=-g_0. \tag{3}$$

In (3) β_k is known as the conjugate gradient parameter and $g_k:=\nabla f(x_k)$. The search direction d_k, assumed to be descent, plays the main role in these methods. On the other hand, the step size α_k guarantees the global convergence in some cases and is crucial in efficiency. Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k. Line search in the conjugate gradient algorithms often is based on the standard Wolfe conditions [13],

$$f(x_k+\alpha_k d_k)-f(x_k)\le\rho\alpha_k g_k^T d_k, \tag{4}$$

$$g(x_k+\alpha_k d_k)^T d_k\ge\sigma g_k^T d_k, \tag{5}$$

where d_k is supposed to be a descent direction and $0<\rho<\sigma<1$. In our developments, the following *basic assumptions* are necessary:

(i) *Boundedness Assumption: The level set* $S:=\{x\in\mathbb{R}^n : f(x)\le f(x_0)\}$ *is bounded, i.e. there exists a positive constant* $B>0$ *such that for all* $x\in S$, $\|x\|\le B$.

(ii) *Lipschitz Continuity Assumption: In a neighborhood* N *of* S, *the function* f *is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant* $L>0$ *such that* $\|\nabla f(x)-\nabla f(y)\|\le L\|x-y\|$, *for all* $x,y\in N$.

Under these assumptions on f, there exists a constant $\Gamma\ge 0$ such that $\|\nabla f(x)\|\le\Gamma$ for all $x\in S$. Besides, $\|s_k\|=\|x_{k+1}-x_k\|\le\|x_{k+1}\|+\|x_k\|\le 2B$.

If the initial direction d_0 is selected as $d_0 = -g_0$, and the objective function to be minimized is a strictly convex quadratic function $f(x) = \frac{1}{2}x^T Ax + b^T x + c$ and the exact line searches are used, that is, $\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k)$, then the conjugacy condition $d_i^T Ad_j = 0$ holds for all $i \neq j$. This relation is the original condition used by Hestenes and Stiefel [6] to derive the conjugate gradient algorithms, mainly for solving symmetric positive-definite systems of linear equations. Let us denote $y_k := g_{k+1} - g_k$. For a general nonlinear twice differential function f, by the mean value theorem, there exists some $\xi \in (0, 1)$ such that $d_{k+1}^T y_k = \alpha_k d_{k+1}^T \nabla^2 f(x_k + \xi\alpha_k d_k)d_k$. Therefore, it seems reasonable to replace the original conjugacy condition $d_i^T Ad_j = 0 (i \neq j)$ with the following one:

$$d_{k+1}^T y_k = 0. \tag{6}$$

In order to accelerate the conjugate gradient algorithm, Perry [14] (see also Shanno [15]) extended this conjugacy condition by incorporating the second order information. He used the secant condition $H_{k+1}y_k = s_k$, where H_k is a symmetric approximation to the inverse Hessian and, as usual, $s_k := x_{k+1} - x_k$. Since for quasi-Newton method the search direction d_{k+1} is computed as $d_{k+1} = -H_{k+1}g_{k+1}$, it follows that $d_{k+1}^T y_k = -(H_{k+1}g_{k+1})^T y_k = -g_{k+1}^T (H_{k+1}y_k) = -g_{k+1}^T s_k$, thus obtaining a new conjugacy condition. This condition can be extended as

$$d_{k+1}^T y_k = -v\left(g_{k+1}^T s_k\right), \tag{7}$$

where $v \geq 0$ is a scalar [16]. In conjugate gradient algorithms we always use inexact line search. Therefore, it seems more reasonable to consider the conjugacy condition (7). The conjugate gradient algorithm (2) and (3) with exact line search always will satisfy the condition $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2$, which is in a direct connection with the sufficient descent condition

$$g_{k+1}^T d_{k+1} \leq -w\|g_{k+1}\|^2, \tag{8}$$

for some arbitrary positive constant $w > 0$. The sufficient descent condition has been used often in the literature to analyze the global convergence of the conjugate gradient algorithms with inexact line search based on the strong Wolfe conditions. Using (7), Dai and Liao [16] obtained a new conjugate gradient algorithm

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - vs_k)}{y_k^T s_k}. \tag{9}$$

For an exact line search, we see that g_{k+1} is orthogonal to s_k. Therefore, for an exact line search, the DL method reduces to the Hestenes and Stiefel (HS) method. Hence, the DL method may not converge for an exact line search. To overcome this and to ensure convergence, the following formula has been suggested [16]:

$$\beta_k^{DL+} = \max\left\{\frac{g_{k+1}^T y_k}{y_k^T s_k}, 0\right\} - v\frac{g_{k+1}^T s_k}{y_k^T s_k}. \tag{10}$$

In this paper, based on these developments, we suggest a new conjugate gradient algorithm in which both the conjugacy condition (7) and the sufficient descent condition (8) are satisfied, independent of the line search.

3 Conjugate Gradient Algorithm with Guaranteed Descent and Conjugacy Conditions

For solving the minimization problem (1) let us consider the following conjugate gradient algorithm:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{11}$$

where $\alpha_k > 0$ is obtained by a variant of the Wolfe line search below discussed, and the directions d_k are generated as

$$d_{k+1} = -\theta_k g_{k+1} + \beta_k s_k, \tag{12}$$

$$\beta_k = \frac{y_k^T g_{k+1} - t_k s_k^T g_{k+1}}{y_k^T s_k}, \tag{13}$$

$d_0 = -g_0$, where θ_k and t_k are scalar parameters which follows to be determined. Algorithms of this form, or variations of them, have been studied by many authors. For example, Andrei [17, 18] considers a preconditioned conjugate gradient algorithm where the preconditioner is a scaled memoryless BFGS matrix and the parameter scaling the gradient is selected as the spectral gradient. On the other hand, Birgin and Martínez [19] suggested a spectral conjugate gradient method, where $\theta_k = s_k^T s_k / s_k^T y_k$. Yuan and Stoer [20] studied the conjugate gradient algorithm on a subspace, where the search direction d_{k+1} is taken from the subspace $\mathrm{span}\{g_{k+1}, d_k\}$. Observe that, if for every $k \geq 1, \theta_k = 1$ and $t_k = v$, then (12) reduces to the Dai and Liao direction (9).

In our algorithm, for all $k \geq 0$, the scalar parameters θ_k and t_k in (12) and (13), respectively, are determined in such a way that both the descent and the conjugacy conditions are satisfied. Therefore, from the *descent condition* (8) we have

$$-\theta_k \|g_{k+1}\|^2 + \frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k} - t_k \frac{(s_k^T g_{k+1})^2}{y_k^T s_k} = -w \|g_{k+1}\|^2, \tag{14}$$

and from the *conjugacy condition* (7)

$$-\theta_k y_k^T g_{k+1} + y_k^T g_{k+1} - t_k s_k^T g_{k+1} = -v\big(s_k^T g_{k+1}\big), \tag{15}$$

where $v > 0$ and $w > 0$ are *known* scalar parameters. Observe that in (14) we modified the classical sufficient descent condition (8) with equality. If $v = 0$, then (15) is the "pure" conjugacy condition. However, in our algorithm, in order to improve the algorithm and to incorporate the second order information, we take $v > 0$. Now, let us define

$$\bar{\Delta}_k := \big(y_k^T g_{k+1}\big)\big(s_k^T g_{k+1}\big) - \|g_{k+1}\|^2 \big(y_k^T s_k\big), \tag{16}$$

$$\Delta_k := \big(s_k^T g_{k+1}\big)\bar{\Delta}_k, \tag{17}$$

$$a_k := v\big(s_k^T g_{k+1}\big) + y_k^T g_{k+1}, \tag{18}$$

$$b_k := w\|g_{k+1}\|^2 \big(y_k^T s_k\big) + \big(y_k^T g_{k+1}\big)\big(s_k^T g_{k+1}\big). \tag{19}$$

Supposing that $\Delta_k \neq 0$ and $y_k^T g_{k+1} \neq 0$,then, from the linear algebraic system given by (14) and (15), we get

$$t_k = \frac{b_k(y_k^T g_{k+1}) - a_k(y_k^T s_k)\|g_{k+1}\|^2}{\Delta_k}, \tag{20}$$

$$\theta_k = \frac{a_k - t_k(s_k^T g_{k+1})}{y_k^T g_{k+1}}, \tag{21}$$

with which the parameter β_k and the direction d_{k+1} can immediately be computed. Observe that, using (20) in (21), we get

$$\theta_k = \frac{a_k}{y_k^T g_{k+1}}\left[1 + \frac{(y_k^T s_k)\|g_{k+1}\|^2}{\bar{\Delta}_k}\right] - \frac{b_k}{\bar{\Delta}_k}. \tag{22}$$

Again, using (20) in (13), we have

$$\beta_k = \frac{y_k^T g_{k+1}}{y_k^T s_k}\left(1 - \frac{b_k}{\bar{\Delta}_k}\right) + a_k \frac{\|g_{k+1}\|^2}{\bar{\Delta}_k}. \tag{23}$$

Therefore, our conjugate gradient algorithm with guaranteed descent and conjugacy condition is defined by (11) and (12), where the scalar parameters θ_k and β_k are given by (22) and (23), respectively, and α_k is computed by a variant of the Wolfe line search we present in the next section.

4 Modified Wolfe Line Search Conditions

In the following, in order to define the algorithm, we shall consider a small modification of the second Wolfe line search condition (5) as

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_k g_k^T d_k, \tag{24}$$

where σ_k is a sequence of parameters satisfying the condition $0 < \rho < \sigma_k < 1$, for all k. Therefore, in our algorithm we consider that the rate of decrease of f in the direction d_k at x_{k+1} is larger than a fraction σ_k, which is modified at every iteration, of the rate of decrease of f in the direction d_k at x_k. The condition $\rho < \sigma_k$, for all $k \geq 0$, guarantees that (4) and (24) can be satisfied simultaneously. We call (4) and (24) *the modified Wolfe line search conditions*. The following propositions can be proved.

Proposition 4.1 *If*

$$\frac{1}{2} < \sigma_k \leq \frac{\|g_{k+1}\|^2}{|y_k^T g_{k+1}| + \|g_{k+1}\|^2}, \tag{25}$$

then, for all $k \geq 1$, $\bar{\Delta}_k < 0$.

Proof Observe that

$$s_k^T g_{k+1} = s_k^T y_k + s_k^T g_k < s_k^T y_k. \tag{26}$$

The modified Wolfe condition (24) gives

$$g_{k+1}^T s_k \geq \sigma_k g_k^T s_k = -\sigma_k y_k^T s_k + \sigma_k g_{k+1}^T s_k. \tag{27}$$

Since $\sigma_k < 1$, we can rearrange (27) to obtain

$$g_{k+1}^T s_k \geq \frac{-\sigma_k}{1-\sigma_k} y_k^T s_k. \tag{28}$$

Now, combining this lower bound for $g_{k+1}^T s_k$ with the upper bound (26), since $y_k^T s_k > 0$ (if $\|g_k\| \neq 0$), we get

$$\left|g_{k+1}^T s_k\right| \leq \left|y_k^T s_k\right| \max\left\{1, \frac{\sigma_k}{1-\sigma_k}\right\}. \tag{29}$$

Since $\sigma_k > 1/2$, from (29) we can write

$$\left|g_{k+1}^T s_k\right| < \frac{\sigma_k}{1-\sigma_k}\left|y_k^T s_k\right|. \tag{30}$$

If (25) is true, then

$$\frac{\sigma_k}{1-\sigma_k}\left|y_k^T g_{k+1}\right| \leq \|g_{k+1}\|^2. \tag{31}$$

Since $y_k^T s_k > 0$ it follows that

$$\frac{\sigma_k}{1-\sigma_k}\left|y_k^T s_k\right|\left|g_{k+1}^T y_k\right| \leq \left|y_k^T s_k\right| \|g_{k+1}\|^2. \tag{32}$$

From (30) and (32) we can write

$$\left|s_k^T g_{k+1}\right|\left|y_k^T g_{k+1}\right| < \frac{\sigma_k}{1-\sigma_k}\left|y_k^T s_k\right|\left|y_k^T g_{k+1}\right| \leq \left|y_k^T s_k\right| \|g_{k+1}\|^2, \tag{33}$$

i.e. $\bar{\Delta}_k < 0$ for all $k \geq 1$. □

In our algorithm we consider

$$\sigma_k = \frac{\|g_{k+1}\|^2}{|y_k^T g_{k+1}| + \|g_{k+1}\|^2}. \tag{34}$$

If $g_k \neq 0$ for all $k \geq 0$,then $0 < \sigma_k < 1$ for all $k \geq 0$.

Proposition 4.2 *Suppose that $\|g_k\| \geq \gamma > 0$ for all $k \geq 0$, i.e. the norm of the gradient is bounded away from zero for all $k \geq 0$. Then the sequence $\{\sigma_k\}$ is uniformly bounded away from zero, independent of k.*

Proof From the basic assumptions observe that $|y_k^T g_{k+1}| \leq \|y_k\|\|g_{k+1}\| \leq L\|s_k\|\Gamma \leq L\Gamma(2B)$. Therefore, $|\sigma_k| = \frac{\|g_{k+1}\|^2}{|y_k^T g_{k+1}|+\|g_{k+1}\|^2} \geq \frac{\gamma^2}{2BL\Gamma+\Gamma^2} \equiv \eta > 0$. Since $|\sigma_k| \geq \eta$ for any $k \geq 0$ it follows that $\{\sigma_k\}$ is uniformly bounded away from zero. □

Proposition 4.3 *Suppose that d_k satisfies the descent condition $g_k^T d_k = -w\|g_k\|^2$, where $w > 0$, and ∇f satisfies the Lipschitz condition $\|\nabla f(x) - \nabla f(x_k)\| \le L\|x - x_k\|$ for all x on the line segment connecting x_k and x_{k+1}, where L is a positive constant. Besides, assume that $\|g_k\| \ge \gamma > 0$ for all $k \ge 0$. If the line search satisfies the modified Wolfe conditions (4) and (24), where $0 < \sigma_k < 1$ for all $k \ge 0$, then*

$$\alpha_k \ge \frac{(1-\sigma_k)}{L} \frac{w\gamma^2}{\|d_k\|^2} \equiv \omega_k. \tag{35}$$

Proof To prove (35) subtract $g_k^T d_k$ from both sides of (24) and, using the Lipschitz condition, we get $(\sigma_k - 1)g_k^T d_k \le (g_{k+1} - g_k)^T d_k \le \alpha_k L\|d_k\|^2$. However, d_k is a descent direction and $\sigma_k < 1$. From the descent condition we immediately get

$$\alpha_k \ge \frac{(1-\sigma_k)}{L} \frac{|g_k^T d_k|}{\|d_k\|^2} = \frac{(1-\sigma_k)}{L} \frac{w\|g_k\|^2}{\|d_k\|^2} \ge \frac{(1-\sigma_k)}{L} \frac{w\gamma^2}{\|d_k\|^2} > 0. \qquad \square$$

Consider $\omega = \inf\{\omega_k\}$, where ω_k is defined in (35).

5 Some Properties of the Algorithm

In the following, we shall present some properties of the elements which define the algorithm. We assume that the step length α_k is computed by the modified Wolfe line search conditions.

Proposition 5.1 *Suppose that d_k satisfies the descent condition $g_k^T d_k = -w\|g_k\|^2$, where $w > 0$, and $\nabla f(x)$ is Lipschitz continuous on the level set S. Besides, assume that $\|g_k\| \ge \gamma > 0$ for all $k \ge 0$. Then the sequence $\{\bar{\Delta}_k\}$ given by (16) is uniformly bounded away from zero, independent of k.*

Proof Since $g_k \ne 0$ for all $k \ge 0$, from (34) it follows that $\sigma_k < 1$ for all $k \ge 1$. Observe that with this value for σ_k, from (30) it follows that $\bar{\Delta}_k < 0$ for all $k \ge 1$. Now, from Proposition 4.3, the modified Wolfe condition (24) and the descent condition $g_k^T d_k = -w\|g_k\|^2$, since $\sigma_k < 1$, for all $k \ge 1$, we have

$$y_k^T s_k = \alpha_k y_k^T d_k \ge \alpha_k(\sigma_k - 1)g_k^T d_k = -\alpha_k(\sigma_k - 1)w\|g_k\|^2 \ge \omega_k(1-\sigma_k)w\gamma^2 > 0.$$

Therefore, $|y_k^T s_k|\|g_{k+1}\|^2 \ge \omega_k(1-\sigma_k)w\gamma^4 > 0$, for all $k \ge 1$, i.e. $(y_k^T s_k)\|g_{k+1}\|^2$ is uniformly bounded away from zero independent of k. We know that d_k is a descent direction for any $k \ge 0$, therefore, even that the line search is not exact; however, the line search based on the modified Wolfe conditions is enough accurate to ensure that $s_k^T g_{k+1}$ tends to zero along the iterations. Therefore, since $|y_k^T g_{k+1}|$ is bounded as $|y_k^T g_{k+1}| \le 2BL\Gamma$, it follows that $(y_k^T g_{k+1})(s_k^T g_{k+1}) \to 0$. Since $\bar{\Delta}_k < 0$ for all $k \ge 1$, we find that the sequence $\{\bar{\Delta}_k\}$ is uniformly bounded away from zero independent of k. $\square$

Proposition 5.2 *Suppose that d_k satisfies the descent condition $g_k^T d_k = -w\|g_k\|^2$, where $w > 0$, and $\|g_k\| \geq \gamma > 0$ for all $k \geq 0$. Then the parameter θ_k defined in (22) tends to $w > 0$, i.e. $\theta_k \to w$.*

Proof From (12), using the descent condition $g_k^T d_k = -w\|g_k\|^2$, we get $\beta_k(s_k^T g_{k+1}) = (\theta_k - w)\|g_{k+1}\|^2 \geq (\theta_k - w)\gamma^2$. Since d_k is a descent direction and the step length α_k is computed by the modified Wolfe line search conditions, it follows that $s_k^T g_{k+1}$ tends to zero. Therefore, θ_k tends to $w > 0$, and hence $\theta_k > 0$. □

Observe that, since w is a real positive and finite constant, and $\theta_k \to w$, there exist real arbitrary and positive constants $0 < c_1 \leq w$ and $c_2 \geq w$, such that, for any $k \geq 1$, $c_1 \leq \theta_k \leq c_2$.

Proposition 5.3 *Suppose that d_k satisfies the descent condition $g_k^T d_k = -w\|g_k\|^2$, $\|g_k\| \geq \gamma > 0$ for all $k \geq 0$ and $w > 1$. Then the scalar parameter b_k given by (19) is positive, i.e. $b_k > 0$.*

Proof By the second Wolfe condition (24) we have $y_k^T s_k = (g_{k+1} - g_k)^T s_k \geq (\sigma_k - 1)g_k^T s_k$. However, from the descent condition it follows that $g_k^T s_k = \alpha_k g_k^T d_k = -\alpha_k w\|g_k\|^2$. From Proposition 4.3 we have $y_k^T s_k \geq (\sigma_k - 1)g_k^T s_k = -\alpha_k(\sigma_k - 1)w\|g_k\|^2 \geq \omega_k w(1 - \sigma_k)\|g_k\|^2 > \omega_k w(1 - \sigma_k)\gamma^2 > 0$. Therefore, by the modified second Wolfe condition (24), for all $k \geq 0$, $y_k^T s_k > 0$. On the other hand, since $w > 1$, from (33) it follows that $w\|g_{k+1}\|^2(y_k^T s_k) \geq |y_k^T g_{k+1}||s_k^T g_{k+1}|$. Since d_k is a descent direction and the step length α_k is computed by the modified Wolfe line search conditions, it follows that $s_k^T g_{k+1}$ tends to zero along the iterations. Therefore, from (19), $b_k > 0$ for all $k \geq 0$. □

6 Acceleration Scheme

We know that in conjugate gradient algorithms the search directions tend to be poorly scaled, and as a consequence, the line search must perform more function evaluations in order to obtain a suitable step length α_k. Therefore, the research effort was directed to design procedures for direction computation, which takes the second order information. For example, the algorithms implemented in SCALCG by Andrei [17, 18] and CONMIN by Shanno and Phua [21] use the BFGS preconditioning with remarkable results. Basically, the acceleration scheme modifies the step length α_k in a multiplicative manner to improve the reduction of the function values along the iterations. As in [22], in the accelerated algorithm, instead of (11), the new estimation of the minimum point is computed as

$$x_{k+1} = x_k + \xi_k \alpha_k d_k, \tag{36}$$

where

$$\xi_k = -\frac{\bar{a}_k}{\bar{b}_k}, \tag{37}$$

$\bar{a}_k := \alpha_k g_k^T d_k$, $\bar{b}_k := -\alpha_k (g_k - g_z)^T d_k$, $g_z := \nabla f(z)$, and $z = x_k + \alpha_k d_k$. Hence, if $\bar{b}_k > 0$, then the new estimation of the solution is computed as $x_{k+1} = x_k + \xi_k \alpha_k d_k$, otherwise $x_{k+1} = x_k + \alpha_k d_k$. Observe that $\bar{b}_k = \alpha_k (g_z - g_k)^T d_k = \alpha_k (d_k^T \nabla^2 f(\bar{x}_k) d_k)$, where $\bar{x}_k$ is a point on the line segment connecting x_k and z. Since $\alpha_k > 0$, it follows that, for convex functions, $\bar{b}_k \geq 0$. Hence, for convex functions, from the sufficient descent condition $g_k^T d_k = -w\|g_k\|^2$ we get

$$\xi_k = -\frac{\bar{a}_k}{\bar{b}_k} = \frac{-\alpha_k (g_k^T d_k)}{\alpha_k (d_k^T \nabla^2 f(\bar{x}_k) d_k)} = \frac{w\|g_k\|^2}{d_k^T \nabla^2 f(\bar{x}_k) d_k} \geq 0. \tag{38}$$

For convex functions there exist constants $m > 0$ and $M < \infty$ such that $m\|u\|^2 \leq u^T \nabla^2 f(x) u \leq M\|u\|^2$, for any $u \neq 0$. Supposing that $\|g_k\| \geq \gamma > 0$ for all $k \geq 0$, (otherwise a stationary point is obtained), then in (36) the step length α_k is modified by a finite and positive value ξ_k. Consequently, with this modification of the step length, by Proposition 5.1, the sequence $\{\bar{\Delta}_k\}$ continues to be uniformly bounded away from zero, independent of k.

7 DESCON Algorithm

Therefore, using the definitions of g_k, s_k, y_k and the above acceleration scheme (36) and (37), we can present the following conjugate gradient algorithm.

Step 1.	Select a starting point $x_0 \in \operatorname{dom} f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Select some positive values for ρ and σ_0, and for v and w. Set $d_0 = -g_0$ and $k = 0$. Select a small positive value: ε_m
Step 2.	Test a criterion for stopping the iterations. If the test is satisfied, then stop; otherwise continue with step 3
Step 3.	Determine the step length α_k by the modified Wolfe line search conditions (4) and (24)
Step 4.	Acceleration scheme. Compute: $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_k = g_k - g_z$
Step 5.	Compute: $\bar{a}_k = \alpha_k g_k^T d_k$, and $\bar{b}_k = -\alpha_k y_k^T d_k$
Step 6.	If $\bar{b}_k > 0$, then compute $\xi_k = -\bar{a}_k/\bar{b}_k$ and update the variables as $x_{k+1} = x_k + \xi_k \alpha_k d_k$, otherwise 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 7.	Compute $\bar{\Delta}_k$ as in (16)
Step 8.	If $\lvert\bar{\Delta}_k\rvert \geq \varepsilon_m$, then determine θ_k and β_k as in (22) and (23), respectively, else set $\theta_k = 1$ and $\beta_k = 0$
Step 9.	Compute the search direction as: $d_{k+1} = -\theta_k g_{k+1} + \beta_k s_k$
Step 10.	Compute $\sigma_k = \|g_{k+1}\|^2/(\lvert y_k^T g_{k+1}\rvert + \|g_{k+1}\|^2)$
Step 11.	Restart criterion. If $\lvert g_{k+1}^T g_k\rvert > 0.2\|g_{k+1}\|^2$ then set $d_{k+1} = -g_{k+1}$
Step 12.	Take $k = k + 1$ and go to step 2

If f is bounded along the direction d_k, then there exists a step size α_k satisfying the modified Wolfe line search conditions (4) and (24). In our algorithm, when the

Powell restart condition is satisfied (step 11), then we restart the algorithm with the negative gradient $-g_{k+1}$. Under reasonable assumptions, the modified Wolfe line search conditions and the Powell restart criterion are sufficient to prove the global convergence of the algorithm. The first trial of the step length crucially affects the practical behavior of the algorithm. At every iteration $k \geq 1$ the starting guess for the step α_k in the line search is computed as $\alpha_{k-1}\|d_{k-1}\|/\|d_k\|$. This selection was used for the first time by Shanno and Phua in CONMIN [21] and in SCALCG by Andrei [17, 18].

The DESCON algorithm can be implemented in some other variants. For example in step 8, when $|\bar{\Delta}_k| \geq \varepsilon_m$ is not satisfied, we can set $\theta_k = 1$ and compute β_k as in classical conjugate gradient algorithms like Hestenes and Stiefel [6], Dai and Yuan [7], Polak, Ribière and Polyak [8, 9], etc.

8 Convergence Analysis

In this section, under the basic assumptions, we analyze the convergence of the algorithm (11) and (12), where θ_k and β_k are given by (22) and (23), respectively, and $d_0 = -g_0$. In the following, we consider that $g_k \neq 0$ for all $k \geq 1$, otherwise a stationary point is obtained. In order to prove the global convergence, often we assume that the step size α_k in (11) is obtained by the strong Wolfe line search, that is,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{39}$$

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \leq \sigma_k g_k^T d_k, \tag{40}$$

where ρ and σ_k are arbitrary positive constants such that $0 < \rho < \sigma_k < 1$. Observe that, since ρ in (39) is small enough, the parameter σ_k in (40) can be selected at each iteration as in (34), thus satisfying the above condition, $0 < \rho < \sigma_k < 1$.

Lemma 8.1 *Suppose that the basic assumptions* (i) *and* (ii) *hold. Consider that the descent condition* $g_k^T d_k < 0$ *hold for all* $k \geq 1$ *and* α_k *satisfies the first Wolfe line search* (4). *Then*

$$\sum_{k=1}^{\infty} -\alpha_k \left(g_k^T d_k\right) < \infty. \tag{41}$$

Proof By (4) and the descent condition we have

$$f_{k+1} - f_k \leq \rho \alpha_k \left(g_k^T d_k\right) \leq 0, \tag{42}$$

i.e. $\{f_k\}$ is a decreasing sequence. Therefore, the basic assumptions imply that there exists a constant f^*such that $\lim_{k\to\infty} f_k = f^*$. With this

$$\sum_{k=1}^{\infty}(f_k - f_{k+1}) = \lim_{n\to\infty} \sum_{k=1}^{n}(f_k - f_{k+1}) = \lim_{n\to\infty}(f_1 - f_{n+1}) = f_1 - f^* < \infty.$$

This, together with (42), implies (41). □

Lemma 8.2 *Suppose that the basic assumptions* (i) *and* (ii) *hold. Consider the conjugate gradient algorithm* (11) *and* (12), *where θ_k and β_k are given by* (22) *and* (23), *respectively*; *the descent condition $g_k^T d_k < 0$ is satisfied for any $k \geq 0$ and α_k is obtained by the modified Wolfe line search conditions* (4) *and* (24), *where $1/2 \leq \sigma_k < 1$. Then*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{43}$$

Proof From (24) and the basic assumptions we have $(\sigma_k - 1)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq L\alpha_k \|d_k\|^2$. Since $1/2 \leq \sigma_k < 1$,it follows that

$$\alpha_k \geq \frac{-(1-\sigma_k)}{L} \frac{g_k^T d_k}{\|d_k\|^2} \geq -\frac{1}{2L} \frac{g_k^T d_k}{\|d_k\|^2}.$$

Combining this with the descent condition $g_k^T d_k < 0$ we get

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \leq 2L \sum_{k=1}^{\infty} \left(-\alpha_k g_k^T d_k\right),$$

which from (41) implies that (43) holds. □

Lemma 8.3 *Suppose that the basic assumptions* (i) *and* (ii) *hold. Consider the conjugate gradient algorithm* (11) *and* (12), *where θ_k and β_k are given by* (22) *and* (23), *respectively*; *for all $k \geq 1 d_k$ is a descent direction satisfying $d_{k+1}^T g_{k+1} = -w\|g_{k+1}\|^2 < 0$, where $w > 0$, and α_k is obtained by the strong Wolfe line search* (39) *and* (40), *where $0 < \sigma_k < 1$. Then either*

$$\liminf_{k \to \infty} \|g_k\| = 0 \tag{44}$$

or

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \tag{45}$$

Proof Observe that in Proposition 5.2 we proved that $\theta_k > 0$ and $\theta_k \to w$. Now, squaring the both terms of $d_{k+1} + \theta_k g_{k+1} = \beta_k s_k$ we obtain $\|d_{k+1}\|^2 + \theta_k^2 \|g_{k+1}\|^2 + 2\theta_k d_{k+1}^T g_{k+1} = \beta_k^2 \|s_k\|^2$. However, $d_{k+1}^T g_{k+1} = -w\|g_{k+1}\|^2$. Therefore,

$$\|d_{k+1}\|^2 = -\left(\theta_k^2 - 2\theta_k w\right)\|g_{k+1}\|^2 + \beta_k^2 \|s_k\|^2. \tag{46}$$

Using Proposition 5.2, observe that for $\theta_k \in]0, 2w]$, $\theta_k^2 - 2\theta_k w \leq 0$ is bounded below by $-w^2$. On the other hand, from (12) we have $g_{k+1}^T d_{k+1} - \beta_k g_{k+1}^T s_k = -\theta_k \|g_{k+1}\|^2$. Now, using the strong Wolfe line search we have $|g_{k+1}^T d_{k+1}| + \sigma_k |\beta_k| |g_k^T s_k| \geq \theta_k \|g_{k+1}\|^2$. At this time we apply the following inequality: $(a + \sigma b)^2 \leq (1 + \sigma^2) \times$

(a^2+b^2), true for all $a, b, \sigma \geq 0$, with $a = |g_{k+1}^T d_{k+1}|$ and $b = |\beta_k||g_k^T s_k|$. After some algebra we get

$$\left(g_{k+1}^T d_{k+1}\right)^2 + \beta_k^2 \left(g_k^T s_k\right)^2 \geq \frac{\theta_k^2}{1+\sigma_k^2} \|g_{k+1}\|^4. \tag{47}$$

However, from Proposition 5.2 $\theta_k \geq c_1$. Besides $0 < \sigma_k < 1$. Therefore $\theta_k^2/(1+\sigma_k^2) \geq c_1^2/2$. Hence

$$\left(g_{k+1}^T d_{k+1}\right)^2 + \beta_k^2 \left(g_k^T s_k\right)^2 \geq e\|g_{k+1}\|^4, \tag{48}$$

where $e = c_1^2/2$ is a positive constant. Using (46) and (48) we can write

$$\begin{aligned}
&\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} + \frac{(g_k^T s_k)^2}{\|s_k\|^2} \\
&\quad = \frac{1}{\|d_{k+1}\|^2}\left[\left(g_{k+1}^T d_{k+1}\right)^2 + \frac{\|d_{k+1}\|^2}{\|s_k\|^2}\left(g_k^T s_k\right)^2\right] \\
&\quad = \frac{1}{\|d_{k+1}\|^2}\left[\left(g_{k+1}^T d_{k+1}\right)^2 + \frac{(g_k^T s_k)^2}{\|s_k\|^2}\left(-\left(\theta_k^2 - 2\theta_k w\right)\|g_{k+1}\|^2 + \beta_k^2\|s_k\|^2\right)\right] \\
&\quad \geq \frac{1}{\|d_{k+1}\|^2}\left[e\|g_{k+1}\|^4 - \left(\theta_k^2 - 2\theta_k w\right)\frac{(g_k^T s_k)^2}{\|s_k\|^2}\|g_{k+1}\|^2\right] \\
&\quad = \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}\left[e - \left(\theta_k^2 - 2\theta_k w\right)\frac{(g_k^T s_k)^2}{\|s_k\|^2}\frac{1}{\|g_{k+1}\|^2}\right].
\end{aligned} \tag{49}$$

From Lemma 8.2 observe that the left side of (49) is finite. Now, from Lemma 8.2 we know that $\lim_{k\to\infty}(g_k^T s_k)^2/\|s_k\|^2 = 0$. On the other hand, for $\theta_k \in]0, 2w]$, $\theta_k^2 - 2\theta_k w$ is finite. Therefore, if (44) is not true, it follows that

$$\lim_{k\to\infty} \frac{(g_k^T s_k)^2}{\|s_k\|^2}\frac{(\theta_k^2 - 2\theta_k w)}{\|g_{k+1}\|^2} = 0. \tag{50}$$

Hence, from (49) we have

$$\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} + \frac{(g_k^T s_k)^2}{\|s_k\|^2} \geq e\frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}, \tag{51}$$

holds for all sufficiently large k. Therefore, by Lemma 8.2 it follows that (45) is true. □

Using Lemma 8.3 we can prove the following proposition, which has a crucial role in proving the convergence of our algorithm.

Proposition 8.1 *Suppose that the basic assumptions* (i) *and* (ii) *hold. Consider the conjugate gradient algorithm* (11) *and* (12), *where* θ_k *and* β_k *are given by* (22) *and*

(23), *respectively, and* α_k *is obtained by the strong Wolfe line search* (39) *and* (40), *where* $0<\sigma_k<1$. *If*

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=\infty, \tag{52}$$

then

$$\liminf_{k\to\infty}\|g_k\|=0. \tag{53}$$

Proof Suppose by contradiction that there is a positive constant γ such that $\|g_k\|\geq\gamma>0$ for all $k\geq 1$. Then, from Lemma 8.3 it follows that $\sum_{k\geq 1}1/\|d_k\|^2\leq\frac{1}{\gamma^4}\sum_{k\geq 1}\|g_k\|^4/\|d_k\|^2<\infty$,which is in contradiction with (52). □

Convergence for Uniformly Convex Functions For uniformly convex functions we can prove that the norm of the direction d_k generated by (12), where θ_k and β_k are given by (22) and (23), respectively, is bounded. Using Proposition 8.1 we can prove the following result.

Theorem 8.1 *Suppose that the assumptions* (i) *and* (ii) *hold. Consider the method* (11)–(13) *and* (16)–(21), *where* α_k *is obtained by the strong Wolfe line search* (39) *and* (40), *where* $1/2\leq\sigma_k<1$. *If there exists a constant* $\mu>0$ *such that*

$$\big(\nabla f(x)-\nabla f(y)\big)^T(x-y)\geq\mu\|x-y\|^2 \tag{54}$$

for all $x,y\in S$, *then*

$$\lim_{k\to\infty}g_k=0. \tag{55}$$

Proof From (54) it follows that f is a uniformly convex function on S and therefore $y_k^Ts_k\geq\mu\|s_k\|^2$. Again, by Lipschitz continuity $\|y_k\|\leq L\|s_k\|$. Using (18) and (19) in (20) we get

$$t_k=\frac{(w-1)(y_k^Ts_k)\|g_{k+1}\|^2(y_k^Tg_{k+1})}{(s_k^Tg_{k+1})\bar{\Delta}_k}+\frac{(y_k^Tg_{k+1})^2-v(y_k^Ts_k)\|g_{k+1}\|^2}{\bar{\Delta}_k}.$$

Observe that since $\{\bar{\Delta}_k\}$ is uniformly bounded away from zero independent of k and $\bar{\Delta}_k<0$ for all $k\geq 1$, there exists a positive constant c_3 such that $|\bar{\Delta}_k|>c_3$. Now, using (28), since $1/2\leq\sigma_k<1$, we get

$$|t_k|\leq\frac{|1-w|\|g_{k+1}\|^2|y_k^Tg_{k+1}|+|y_k^Tg_{k+1}|^2+v|y_k^Ts_k|\|g_{k+1}\|^2}{c_3}.$$

From the basic assumptions, observe that $|y_k^Tg_{k+1}|\leq\|y_k\|\|g_{k+1}\|\leq L\|s_k\|\Gamma\leq L\Gamma(2B)$ and $|y_k^Ts_k|\leq\|y_k\|\|s_k\|\leq L\|s_k\|^2\leq L(2B)^2$. With this we have

$$|t_k|\leq\frac{2BL\Gamma^2[|1-w|\Gamma+2B(L+v)]}{c_3}\equiv t,$$

where $t>0$ is a constant. Now, from (13), using the Lipschitz continuity, we have

$$|\beta_k| = \left|\frac{y_k^T g_{k+1}}{y_k^T s_k} - t_k \frac{s_k^T g_{k+1}}{y_k^T s_k}\right| \le \frac{\|y_k\|\|g_{k+1}\|}{\mu\|s_k\|^2} + |t_k| \frac{\|s_k\|\|g_{k+1}\|}{\mu\|s_k\|^2}$$
$$\le \frac{L\|s_k\|\|g_{k+1}\|}{\mu\|s_k\|^2} + t\frac{\|s_k\|\|g_{k+1}\|}{\mu\|s_k\|^2} = \frac{L+t}{\mu}\frac{\Gamma}{\|s_k\|}. \tag{56}$$

Hence, from (12) and Proposition 5.2:

$$\|d_{k+1}\| \le c_2\Gamma + \frac{L+t}{\mu}\frac{\Gamma}{\|s_k\|}\|s_k\| = \left(c_2 + \frac{L+t}{\mu}\right)\Gamma, \tag{57}$$

which implies that (52) is true. Therefore, by Proposition 8.1 we have (53), which for uniformly convex functions is equivalent to (55). □

Convergence for General Nonlinear Functions Firstly we prove that under very mild conditions the direction d_k generated by (12), where θ_k and β_k are given by (22) and (23), respectively, is bounded. Again, by Proposition 8.1 we can prove the following result.

Theorem 8.2 *Suppose that the basic assumptions* (i) *and* (ii) *hold and* $\|g_k\| \ge \gamma > 0$ *for all* $k \ge 0$. *Consider the conjugate gradient algorithm* (11), *where the direction* d_{k+1} *given by* (12) *and* (13) *satisfies the descent condition* $g_k^T d_k = -w\|g_k\|^2$, *where* $w > 1$, *and the step length* α_k *is obtained by the strong Wolfe line search* (39) *and* (40), *where* $1/2 \le \sigma_k < 1$. *Then* $\liminf_{k\to\infty} \|g_k\| = 0$.

Proof From (13), using (20) after some algebra, we have

$$\beta_k = \frac{y_k^T g_{k+1}}{y_k^T s_k}\left(1 - \frac{b_k}{\bar{\Delta}_k}\right) + a_k \frac{\|g_{k+1}\|^2}{\bar{\Delta}_k}. \tag{58}$$

From Proposition 4.3, the definition of ω, the modified Wolfe condition (24) and the descent condition $g_k^T d_k = -w\|g_k\|^2$, since $\|g_k\| \ge \gamma > 0$ and $\sigma_k < 1$, for all $k \ge 0$, we have $y_k^T s_k \ge w\omega_k(1-\sigma_k)\gamma^2 > w\omega(1-\sigma_k)\gamma^2 > 0$. However, from the basic assumptions we have $|y_k^T g_{k+1}|\|s_k\| \le \|y_k\|\|g_{k+1}\|\|s_k\| \le L\|s_k\|^2\Gamma \le L\Gamma(2B)^2$. Therefore,

$$\frac{|y_k^T g_{k+1}|}{|y_k^T s_k|} \le \frac{L\Gamma(2B)^2}{w\omega(1-\sigma_k)\gamma^2}\frac{1}{\|s_k\|} = \frac{\bar{c}}{\|s_k\|}, \tag{59}$$

where $\bar{c} = L\Gamma(2B)^2/w\omega(1-\sigma_k)\gamma^2$. Now, observe that since for all $k \ge 0$, $\bar{\Delta}_k < 0$ (by Proposition 5.1) and $b_k > 0$ (by Proposition 5.3), it follows that $-b_k/\bar{\Delta}_k > 0$. Besides, from (16) and (19) we can write

$$-\frac{b_k}{\bar{\Delta}_k} = w + (1+w)\frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{-\bar{\Delta}_k}. \tag{60}$$

Since $-\bar{\Delta}_k > 0$ and $s_k^T g_{k+1}$ tends to zero along the iterations, it follows that $-b_k/\bar{\Delta}_k$ tends to $w > 0$. Hence $1 - b_k/\bar{\Delta}_k$ tends to $1 + w$. Therefore, there exists a positive constant $c_4 > 1$ such that $1 < 1 - b_k/\bar{\Delta}_k \le c_4$.

Again, from the basic assumptions we have $|y_k^T s_k|\|s_k\| \le \|y_k\|\|s_k\|^2 \le L\|s_k\|^3 \le L(2B)^3$. Therefore, $|y_k^T s_k| \le L(2B)^3/\|s_k\|$. Now, from (18) and (29) we have

$$\begin{aligned}
|a_k| &= \left|v\left(s_k^T g_{k+1}\right) + \left(y_k^T g_{k+1}\right)\right| \le v\left|s_k^T g_{k+1}\right| + \left|y_k^T g_{k+1}\right| \\
&\le v\left|y_k^T s_k\right| \max\left\{1, \frac{\sigma_k}{1-\sigma_k}\right\} + \left|y_k^T g_{k+1}\right| \\
&\le v\frac{L(2B)^3}{\|s_k\|} \max\left\{1, \frac{\sigma_k}{1-\sigma_k}\right\} + \frac{L\Gamma(2B)^2}{\|s_k\|}.
\end{aligned} \tag{61}$$

Since $1/2 \le \sigma_k < 1$, there exists a positive constant $c_5 > 0$ such that $\max\{1, \sigma_k/(1-\sigma_k)\} \le c_5$. Hence,

$$|a_k| \le \left(vLc_5(2B)^3 + L\Gamma(2B)^2\right)\frac{1}{\|s_k\|} = \frac{\hat{c}}{\|s_k\|}, \tag{62}$$

where $\hat{c} = vLc_5(2B)^3 + L\Gamma(2B)^2$. With these, from (58) we can write

$$\begin{aligned}
|\beta_k| &\le \left|\frac{y_k^T g_{k+1}}{y_k^T s_k}\right|\left|1 - \frac{b_k}{\bar{\Delta}_k}\right| + |a_k|\frac{\|g_{k+1}\|^2}{|\bar{\Delta}_k|} \le \frac{\bar{c}c_4}{\|s_k\|} + \frac{\hat{c}\Gamma^2}{c_3}\frac{1}{\|s_k\|} \\
&= \left[\bar{c}c_4 + \frac{\hat{c}\Gamma^2}{c_3}\right]\frac{1}{\|s_k\|}.
\end{aligned} \tag{63}$$

From (12) we have

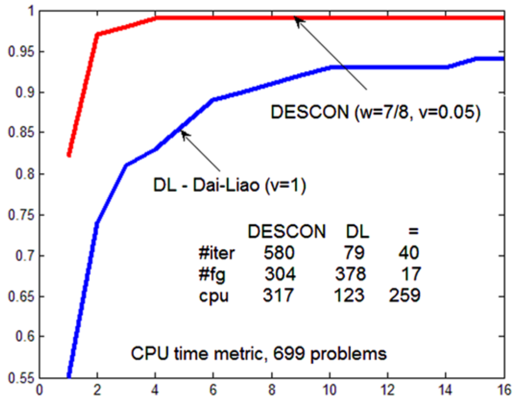
$$\|d_{k+1}\| \le |\theta_k|\|g_{k+1}\| + |\beta_k|\|s_k\| \le c_2\Gamma + \left[\bar{c}c_4 + \frac{\hat{c}\Gamma^2}{c_3}\right]\frac{1}{\|s_k\|}\|s_k\| \equiv E, \tag{64}$$

where E is a positive constant. Therefore, for all $k \ge 0$, $\|d_k\| \le E$, which implies (52). Therefore, by Proposition 8.1, since d_k is a descent direction, we have $\liminf_{k\to\infty} \|g_k\| = 0$. □

9 Numerical Results and Comparisons

In this section, we report some numerical results obtained with an implementation of the DESCON algorithm. The code is written in Fortran and compiled with f77 (default compiler settings) on a Workstation Intel Pentium 4 with 1.8 GHz. DESCON and the other algorithms considered in this numerical study use the loop unrolling to a depth of 5. We selected a number of 75 large-scale unconstrained optimization test functions in generalized or extended form [5]. For each test function we have taken ten numerical experiments with the number of variables increasing as $n = 1000, 2000, \ldots, 10000$. The algorithm implements the Wolfe line search conditions with $\rho = 0.0001$, $\sigma = \|g_{k+1}\|^2/(|y_k^T g_{k+1}| + \|g_{k+1}\|^2)$, and the same stopping

Fig. 1 DESCON versus DL($v = 1$)



criterion $\|g_k\|_\infty \le 10^{-6}$. In DESCON we set $w = 7/8$ and $v = 0.05$. In our numerical experiments θ_k is not restricted in the interval $[0, 2w]$. In all the algorithms we considered in this numerical study the maximum number of iterations is limited to 10000.

The comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2, for problem $i = 1, \dots, 750$, respectively. We say that in the particular problem i, the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{\mathrm{ALG1}} - f_i^{\mathrm{ALG2}} \right| < 10^{-3} \tag{65}$$

and the number of iterations (#iter), or the number of function-gradient evaluations (#fg), or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively.

In the first set of numerical experiments we compare DESCON versus Dai and Liao ($v = 1$) conjugate gradient algorithm (9). Figure 1 shows the Dolan and Moré CPU performance profile of DESCON versus DL($v = 1$).

When comparing DESCON with DL($v = 1$) conjugate gradient algorithm subject to CPU time metric we see that DESCON is top performer. Comparing DESCON with DL($v = 1$) (see Fig. 1), subject to the number of iterations, we see that DESCON was better in 580 problems (i.e. it achieved the minimum number of iterations in 580 problems). DL($v = 1$) was better in 79 problems and they achieved the same number of iterations in 40 problems, etc. Out of 750 problems, only for 699 problems does the criterion (65) hold.

In the second set of numerical experiments we compare DESCON versus Hestenes and Stiefel (HS) ($\beta_k^{\mathrm{HS}} = y_k^T g_{k+1}/y_k^T s_k$) [6], versus Dai and Yuan (DY) ($\beta_k^{\mathrm{DY}} = g_{k+1}^T g_{k+1}/y_k^T s_k$) [7] and versus Polak–Ribière–Polyak (PRP) ($\beta_k^{\mathrm{PRP}} = y_k^T g_{k+1}/g_k^T g_k$) [8, 9], conjugate gradient algorithms. Figures 2, 3 and 4 present the Dolan and Moré CPU performance profile of DESCON versus HS, DY, and PRP, respectively.

In the third set of numerical experiments we compare DESCON versus hybrid Dai–Yuan [7], ($\beta_k^{\mathrm{hDY}} = \max\{-c\beta_k^{\mathrm{DY}}, \min\{\beta_k^{\mathrm{HS}}, \beta_k^{\mathrm{DY}}\}\}, c = (1-\sigma)/(1+\sigma)$,

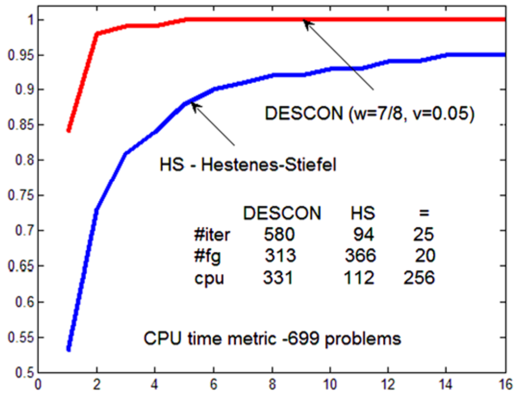


Fig. 2 DESCON versus Hestenes–Stiefel

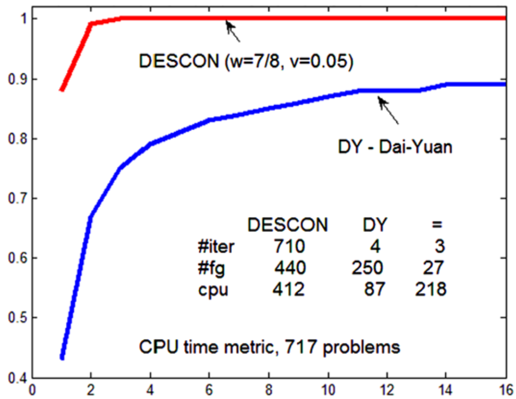


Fig. 3 DESCON versus Dai–Yuan

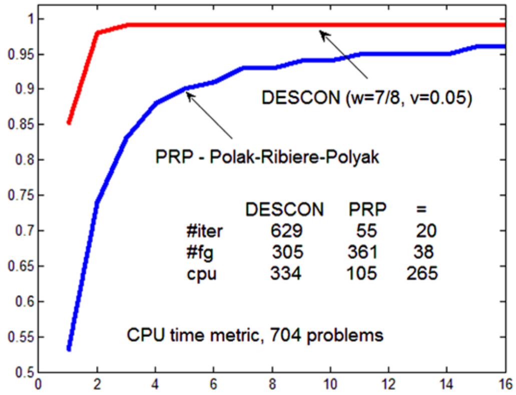


Fig. 4 DESCON versus Polak–Ribière–Polyak

$\sigma = 0.8$). Figure 5 presents the Dolan and Moré CPU time performance profile of DESCON versus hDY. The best performance, relative to the CPU time metric, again was obtained by DESCON, the top curve in Fig. 5.

In the fourth set of numerical experiments we compare DESCON versus CG_DESCENT. In CG_DESCENT, at every iteration, the direction d_k satisfies the suffi-

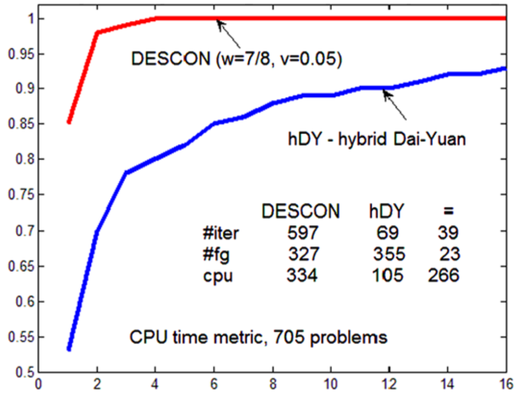


Fig. 5 DESCON versus hybrid Dai–Yuan

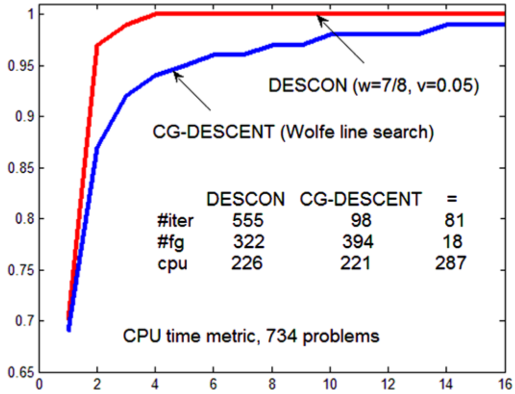


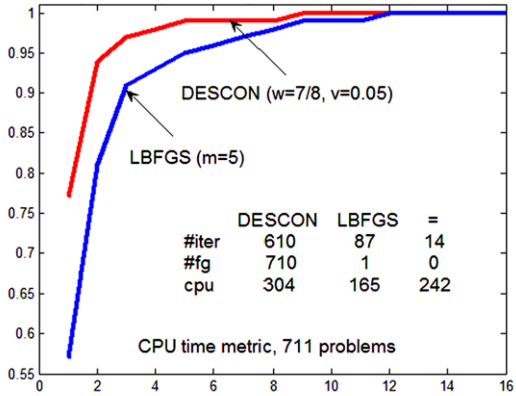
Fig. 6 DESCON versus CG_DESCENT

cient descent condition $g_k^T d_k \le -(7/8)\|g_k\|^2$. This is the main reason we considered $w = 7/8$ in all our numerical experiments. Figure 6 presents the Dolan and Moré CPU time performance profile of DESCON versus CG_DESCENT with Wolfe line search. Again, the best performance, relative to the CPU time metric, was obtained by DESCON, the top curve in Fig. 6.

Finally, we compare DESCON versus L-BFGS ($m = 5$) by Liu and Nocedal [11] as in Fig. 7, where m is the number of pairs (s_k, y_k) used. Observe that DESCON is top performer again. The differences are significant. The linear algebra in the L-BFGS code to update the search direction is very different from the linear algebra used in DESCON. On the other hand, the step length in L-BFGS is determined at each iteration by means of the line search routine MCVSRCH, which is a slight modification of the routine CSRCH written by Moré and Thuente [23].

In the following, in Fig. 8, we present the performance profile of DESCON ($w = 7/8, v = 0.05$) versus HS, PRP, CG_DESCENT and L-BFGS ($m = 5$), subject to CPU time metric. We see that, among these algorithms, DESCON is top performer. Concerning the robustness close to DESCON there are CG_DESCENT with Wolfe line search and L-BFGS ($m = 5$). In this context HS and PRP have similar performances, PRP being slightly more robust.

Fig. 7 DESCON versus L-BFGS ($m = 5$)



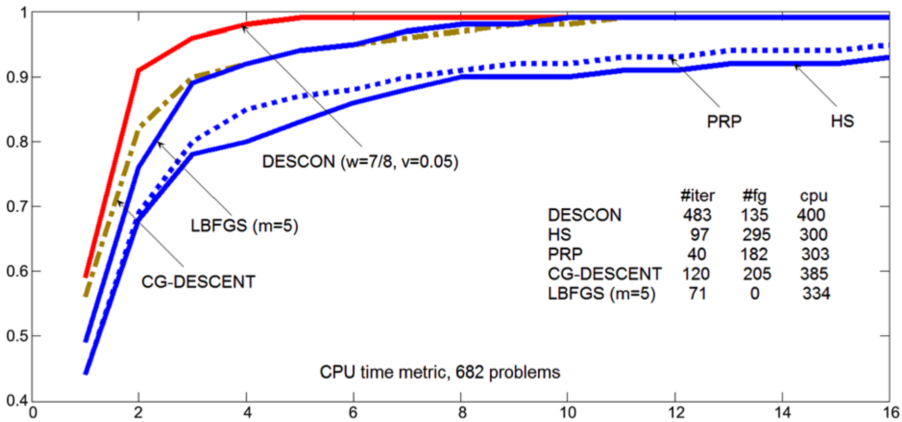


Fig. 8 DESCON versus HS, PRP, CG_DESCENT and L-BFGS ($m = 5$)

As a final remark observe that the DESCON algorithm can be implemented in different versions. For example, in step 8 for θ_k and β_k computation, one version can implement a truncation mechanism suggested by Hager and Zhang [10] as $\beta_k^+ = \max\{\beta_k, \eta_k\}$, where β_k is computed as in (23) and $\eta_k = -1/(\|d_k\| \min\{0.1, \|g_k\|\})$. In this case, subject to CPU time metric, DESCON using (22) and (23) was fastest in 113 problems. On the other hand, DESCON, using (22) and β_k^+, was fastest in 107 problems, showing that the truncation mechanism is not very much effective.

10 Sensitivity Analysis

In order to see the performances of the algorithm, we present a *sensitivity study* of DESCON subject to the variation of v and w parameters. Both these parameters emphasize the importance of the conjugacy condition and the sufficient descent condi-

Table 1 Sensitivity of the DESCON subject to $v \cdot w = 7/8$

v	#itert	#fgt	cput
0	247557	584091	130.35
0.001	248268	582814	129.69
0.005	247696	581850	132.16
0.01	248590	586607	133.66
0.02	249868	585260	138.75
0.05	248580	589644	138.71
0.07	254988	612957	141.33
0.1	246473	580293	133.54
0.2	256726	599135	131.78
0.5	249513	590716	133.38
0.7	254423	591242	128.25
1	247704	580790	133.45

tion, respectively. From (12), (13), and (16)–(21) we have

$$\frac{\partial d_{k+1}}{\partial w} = \frac{(y_k^T s_k)\|g_{k+1}\|^2}{\bar{\Delta}_k}\left(g_{k+1} - \frac{y_k^T g_{k+1}}{y_k^T s_k} s_k\right), \tag{66}$$

$$\frac{\partial d_{k+1}}{\partial v} = -\frac{(s_k^T g_{k+1})}{\bar{\Delta}_k}\left(\left(s_k^T g_{k+1}\right) g_{k+1} - \|g_{k+1}\|^2 s_k\right). \tag{67}$$

Observe that if the line search is exact ($s_k^T g_{k+1} = 0$), then from (67) we see that the algorithm is not sensitive to the variation of v. However, in our algorithm the line search is not exact.

Table 1 presents the total number of iterations (#itert), the total number of function and its gradient evaluations (#fgt) and the total CPU time (cput) for solving the above set of 750 unconstrained optimization test problems for $w = 7/8$ and for different values of v. For example, for solving the set of 750 problems with $w = 7/8$ and $v = 0$, the total number of iteration is 247557, the total number of function and its gradient evaluations is 584091 and the total CPU time is 130.35 seconds, etc. In Table 1 we have the computational evidence concerning the sensitivity of DESCON, corresponding to a set of 12 numerical experiments, subject to the variation of v parameter. Subject to the CPU time metric the average of the total CPU time corresponding to these 12 numerical experiments, for solving 750 problems in each experiment, is $1605.0/12 = 133.75$ seconds. The largest deviation is 7.58 seconds and corresponds to the numerical experiment in which $v = 0.07$. Therefore, in all these 12 numerical experiments the maximum deviation is of $7.58/750 = 0.01$ seconds per problem.

In the following, we present the sensitivity of DESCON subject to the variation of w parameter. Table 2 presents the total number of iterations, the total number of function and its gradient evaluations, and the total CPU time for solving the above set of 750 unconstrained optimization test problems for $v = 0.7$ and for six different values of w.

The best results corresponding to this set of six numerical experiments are obtained for $w = 0.9$. Subject to CPU time metric for solving 750 problems in each

Table 2 Sensitivity of the DESCON subject to $w \cdot v = 0.7$

w	#itert	#fgt	Cput
0.5	264322	631141	155.45
0.6	263076	615079	141.80
0.7	257098	603704	138.01
0.8	261982	626266	147.05
0.9	248710	586730	**134.21**
1	260475	616134	148.99

Table 3 Applications from MINPACK-2 collection

A1	*Elastic-Plastic Torsion* [24, pp. 41–55], $c = 5$
A2	*Pressure Distribution in a Journal Bearing* [25], $b = 10$, $\varepsilon = 0.1$
A3	*Optimal Design with Composite Materials* [26], $\lambda = 0.008$
A4	*Steady-State Combustion* [27, pp. 292–299], [28], $\lambda = 5$
A5	*Minimal Surfaces with Enneper conditions* [29, pp. 80–85]

of these six numerical experiments, the total CPU time difference is of $155.45 - 134.21 = 21.24$ seconds. Therefore, in all these six numerical experiments the maximum deviation is of $21.24/750 = 0.028$ seconds per problem. Observe that the average of the total CPU time corresponding to these six numerical experiments is $865.51/6 = 144.25$ seconds. The largest deviation is of $155.45 - 144.25 = 11.20$ seconds. Therefore, in all these six numerical experiments the maximum deviation is of $11.20/750 = 0.0149$ seconds per problem. Practically, DESCON is very little sensitive to the variation of w.

11 Solving MINPACK-2 Applications

We now present comparisons between DESCON and CG_DESCENT conjugate gradient algorithms for solving some applications from MINPACK-2 test problem collection [12]. In Table 3, we present these applications, as well as the values of their parameters. The infinite-dimensional version of these problems is transformed into a finite element approximation by triangulation. The discretization steps are $nx = 1000$ and $ny = 1000$, thus obtaining minimization problems with 1,000,000 variables.

A comparison between DESCON ($v = 0.05$, $w = 0.875$, Powell restart criterion, $\|\nabla f(x_k)\|_\infty \le 10^{-6}$, $\rho = 10^{-4}$) and CG_DESCENT (Wolfe line search, default settings, $\|\nabla f(x_k)\|_\infty \le 10^{-6}$) for solving these applications is given in Table 4.

Form Table 4 we see that subject to the CPU time metric the DESCON algorithm is top performer again, and the difference is significant, about **2807.65** seconds for solving all these five applications. Observe that DESCON is faster and more robust than CG_DESCENT for solving real large-scale unconstrained optimization applications.

Table 4 Performance of DESCON and CG_DESCENT. 1,000,000 variables. cpu seconds

	DESCON			CG_DESCENT		
	#iter	#fg	cpu	#iter	#fg	cpu
A1	1113	2257	**324.45**	1145	2291	450.08
A2	2833	5694	**930.37**	3368	6737	1462.38
A3	4734	9506	**2069.76**	4841	9684	2975.02
A4	1413	2864	**1282.27**	1806	3613	2358.35
A5	1279	2580	**516.39**	1226	2453	685.06
Total	11372	22901	**5123.24**	12386	24778	7930.89

12 Conclusions

For solving large scale unconstrained optimization problems we have presented an accelerated conjugate gradient algorithm that, for all $k \geq 0$, both the descent and the conjugacy conditions are guaranteed. In our algorithm the search direction is selected as a linear combination of $-g_{k+1}$ and s_k, where the coefficients in this linear combination are selected in such a way that both the descent and the conjugacy condition are satisfied at every step. The algorithm uses the modified Wolfe line search, where in the second Wolfe condition the parameter σ is modified at every iteration. Besides, the step length is modified by an acceleration scheme, which proved to be very efficient in reducing the values of the minimizing function along the iterations. For a test set consisting of 750 problems with dimensions ranging between 1000 and 10,000, the CPU time performance profiles of DESCON was higher than those of HS, PRP, DY, hDY, CG_DESCENT with Wolfe line search and limited memory quasi-Newton method L-BFGS ($m = 5$). A number of five applications from MINPACK2 problems collection, with 10^6 variables, illustrate the performances of DESCON versus CG_DESCENT. At present, from the above test problems and applications we have computational evidence that DESCON is one of the fastest and the most robust conjugate gradient algorithm.

References

1. Golub, G.H., O'Leary, D.P.: Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. SIAM Rev. **31**, 50–102 (1989)
2. O'Leary, D.P.: Conjugate gradients and related KMP algorithms: the beginnings. In: Adams, L., Nazareth, J.L. (eds.) Linear and Nonlinear Conjugate Gradient—Related Methods, pp. 1–8. SIAM, Philadelphia (1996)
3. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**, 35–58 (2006)
4. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
5. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**, 147–161 (2008)
6. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)

7. Dai, Y.H., Yuan, Y.: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. **103**, 33–47 (2001)
8. Polak, E., Ribière, G.: Note sur la convergence de directions conjuguée. Rev. Fr. Inf. Rech. Oper. **16**, 35–43 (1969)
9. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **9**, 94–112 (1969)
10. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
11. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization methods. Math. Program. **45**, 503–528 (1989)
12. Averick, B.M., Carter, R.G., Moré, J.J., Xue, G.L.: The MINPACK-2 test problem collection. Mathematics and Computer Science Division, Argonne National Laboratory, Preprint MCS-P153-0692 (1992)
13. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1969)
14. Perry, A.: A modified conjugate gradient algorithm. Oper. Res. **26**, 1073–1078 (1978)
15. Shanno, D.F.: Conjugate gradient methods with inexact searches. Math. Oper. Res. **3**, 244–256 (1978)
16. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**, 87–101 (2001)
17. Andrei, N.: Scaled conjugate gradient algorithms for unconstrained optimization. Comput. Optim. Appl. **38**, 401–416 (2007)
18. Andrei, N.: Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optim. Methods Softw. **22**, 561–571 (2007)
19. Birgin, E., Martínez, J.M.: A spectral conjugate gradient method for unconstrained optimization. Appl. Math. Optim. **43**, 117–128 (2001)
20. Yuan, Y., Stoer, J.: A subspace study on conjugate gradient algorithms. Z. Angew. Math. Mech. **75**, 69–77 (1995)
21. Shanno, D.F., Phua, K.H.: Algorithm 500. Minimization of unconstrained multivariate functions. ACM Trans. Math. Softw. **2**, 87–94 (1976)
22. Andrei, N.: Acceleration of conjugate gradient algorithms for unconstrained optimization. Appl. Math. Comput. **213**, 361–369 (2009)
23. Moré, J.J., Thuente, D.J.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Softw. **20**, 286–307 (1994)
24. Glowinski, R.: Numerical Methods for Nonlinear Variational Problems. Springer, Berlin (1984)
25. Cimatti, G.: On a problem of the theory of lubrication governed by a variational inequality. Appl. Math. Optim. **3**, 227–242 (1977)
26. Goodman, J., Kohn, R., Reyna, L.: Numerical study of a relaxed variational problem from optimal design. Comput. Methods Appl. Mech. Eng. **57**, 107–127 (1986)
27. Aris, R.: The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts. Oxford University Press, London (1975)
28. Bebernes, J., Eberly, D.: Mathematical Problems from Combustion Theory. Applied Mathematical Sciences, vol. 83. Springer, Berlin (1989)
29. Nitsche, J.C.C.: Lectures on Minimal Surfaces vol. 1. Cambridge University Press, Cambridge (1989)