# Performance of Conjugate Gradient Algorithms on some MINPACK-2 Unconstrained Optimization Applications

**Neculai Andrei**

National Institute for Research and Development in Informatics,

Center for Advanced Modeling and Optimization

8-10, Averescu Avenue, Bucharest 1, Romania

E-mail: nandrei@ici.ro

**Abstract:** The paper presents a survey of nonlinear conjugate gradient algorithms and the computational performances of some conjugate gradient algorithms for solving some unconstrained optimization applications from MINPACK-2 collection.

**Keywords:** Nonlinear conjugate gradient methods, Unconstrained optimization, Nonlinear programming. Applications of nonlinear programming.

**AMS subject classifications.** 90C06, 90C26, 65Y20

**Neculai Andrei** graduated in Mathematics, "A.I. Cuza" University – Iassy, and in Electrical Engineering from "Politehnica" University – Bucharest. He took his Ph.D degree for his contributions to Digraph Approach of Large-Scale Sparse Linear Dynamic Systems in 1984. He has been a senior scientific researcher at National Institute for R&D in Informatics - ICI, since 1973. He is author of 13 books and text books, as well as over 50 published papers and 300 Technical Reports in area of mathematical modeling optimization. He is the key architect of *ALLO language and compiler* for modeling in mathematical programming, as well as of some professional packages for linear programming and large-scale unconstrained optimization. His main current scientific interests centre on modeling for mathematical programming and on large-scale non-linear optimization, conjugate gradient, interior point methods, penalty and barrier approaches. Dr. Andrei is a founding member of *the Operations Research Society* of Romania; founding member of the *Center for Advanced Modeling and Optimization*; member of the Editorial Board of *Computational Optimization and Applications* – Springer Verlag (since 1992); member of *Computational Optimization and Applications – Optimization Forum*; member of the Editorial Board of *Journal of Systems Science and Complexity* – Science Press – China and Allerton Press – USA (since 2001). He is Editor-in-Chief to the *Advanced Modeling and Optimization – An Electronic International Journal* – ICI Press (since 1999). Dr. Andrei is an Alexander-von- Humboldt fellow, member ERCIM and of Academy of Romanian Scientists (since 2001).

## 1. Introduction

Conjugate gradient methods represent an important class of unconstrained optimization algorithms characterized by low memory requirements and strong local and global convergence properties. Their implementation is very easy and accessible in codes able to solve large-scale problems with millions of variables. These methods have been the subject of intense research and analysis for more than 50 years. For the very beginning they started as algorithms for solving symmetric, positive-definite linear systems of equations by works of Cornelius Lanczos and Magnus Hestenes and others (notably Forsythe, Motzkin, Rosser, Stein) at the Institute for Numerical Analysis, and with independent research of Eduard Stiefel at Technische Hochschule Zürich. The first paper was published by Hestenes and Stiefel in 1952, where an algorithm for solving symmetric, positive-definite linear systems was presented. The algorithm was soon extended to general nonlinear unconstrained optimization. Today, conjugate gradient algorithms represent a very important class of algorithms for solving large scale unconstrained optimization problems with good global convergence properties.

In this paper we survey the nonlinear conjugate gradient algorithms, emphasizing their definition and global convergence properties and present the computational performances of a number of conjugate gradient algorithms on some MINPACK-2 applications [Averick, Carter, and Moré, 1991], [Averick, Carter, Moré and Xue, 1992] All conjugate gradient algorithms can be classified in four general classes. The first class contains the algorithms based on conjugacy condition. Here the classical conjugate gradient algorithms like Hestenes and Stiefel [1952], Fletcher and Reeves [1964], Polak and Ribière [1969], Polyak [1969], Liu and Storey [1991], Dai and Yuan [1999, 2001] or Yabe and Takano [2004] are included. The second class of conjugate gradient algorithms is based on a combination of the methods from the first class, leading to the hybrid conjugate gradient algorithms. The scaled conjugate gradient algorithms form the third class of algorithms. Finally, the preconditioned conjugate gradient algorithms form the fourth class.

The structure of the paper is as follows. Section 2 contains a survey of nonlinear conjugate gradient algorithms. The other sections contain the numerical performances of 14 nonlinear conjugate gradient algorithms on 6 applications from MINPACK-2 collection.

## 2. A survey of nonlinear conjugate gradient algorithms

An excellent survey of nonlinear conjugate gradient methods, with special attention to global convergence properties, has been given by Hager and Zhang [2005]. In this section we focus on different versions of nonlinear conjugate gradient methods and their convergence properties.

Consider the problem to minimize a function of $n$ variables:

$$\min f(x), \tag{1}$$

where $f: R^n \to R$ is a continuous differentiable function with $g(x) = \nabla f(x)$ its gradient. Conjugate gradient algorithms solving (1) are iterative methods of the form:

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

where $t_k$ >0 is a step length and $d_k$ is a search direction. The search direction at the very first iteration is the steepest descent direction: $d_0 = -g_0 \equiv \nabla f(x_0)$. The directions along the iterations are computed according to:

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

where $\beta_k$ is a scalar. If the function $f(x)$ is strictly quadratic:

$$f(x) = \frac{1}{2} x^T A x - b^T x, \tag{4}$$

where $A$ is a symmetric and positive definite matrix, and $t_k$ is the exact one-dimensional minimizer given by

$$t_k = \frac{g_k^T g_k}{d_k^T A d_k}, \tag{5}$$

then the method (2)-(3) is called the linear conjugate gradient method. This method was firstly proposed by Hestenes and Stiefel [1952] for solving linear systems of equations:

$$Ax = b \tag{6}$$

where for $\beta_k$ several formulas have been proposed, which for strictly convex quadratic objective functions are equivalent. Linear conjugate gradient algorithms are characterized by the conjugacy condition:

$$d_i^T A d_j = 0, \quad i \neq j \tag{7}$$

which guarantees the finite termination of linear conjugate gradients algorithms.

The nonlinear conjugate gradient algorithm for general unconstrained optimization problem was firstly proposed by Fletcher and Reeves [1964]. In this case the conjugacy condition is replaced by

$$d_{k+1}^T y_k = 0, \tag{8}$$

where $y_k = g_{k+1} - g_k$. This is motivated by the following relations:

$$d_{k+1}^T A d_k = \frac{1}{t_k} d_{k+1}^T A(x_{k+1} - x_k) = \frac{1}{t_k} d_{k+1}^T (g_{k+1} - g_k) = \frac{1}{t_k} d_{k+1}^T y_k,$$

or by the mean value theorem

$$d_{k+1}^T y_k = t_k d_{k+1}^T \nabla^2 f(x_k + \theta t_k d_k) d_k,$$

for some $\theta \in (0,1)$.

In order to ensure the convergence of algorithm (2), it is necessary to constrain the choice of $t_k$. Usually, the step length is selected to satisfy the Wolfe line search conditions [Wolfe, 1969, 1971]:

$$f(x_k + t_k d_k) - f(x_k) \le \sigma_1 t_k g_k^T d_k, \tag{9a}$$

$$\nabla f(x_k + t_k d_k)^T d_k \ge \sigma_2 g_k^T d_k, \tag{9b}$$

where $0 < \sigma_1 \le \sigma_2 < 1$.

The general conjugate gradient algorithm can be described as follows:

**Algorithm CG**

*Step 1*. Select $x_0$, set $d_0 = -g_0$ and $k = 0$.

*Step 2*. Compute the step length $t_k$ >0 satisfying the Wolfe line search (9a) and (9b).

*Step 3*. Compute $x_{k+1} = x_k + t_k d_k$. If $\|g_{k+1}\| \le \varepsilon$, then stop.

*Step 4*. Compute $\beta_k$ and generate the direction $d_{k+1} = -g_{k+1} + \beta_k d_k$.

*Step 5*. Set $k = k + 1$ and go to step 2. ♦

Different conjugate gradient algorithms correspond to different choices for the scalar $\beta_k$. The well known formulas for $\beta_k$ are summarized in Table 1. Conjugate gradient algorithms (2)-(3) with exact line searches satisfy the equality

$$g_k^T d_k = -\|g_k\|^2, \tag{10}$$

which implies the *sufficient descent condition*

$$g_k^T d_k < -c\|g_k\|^2, \tag{11}$$

where $c > 0$ is a constant. Often, the sufficient descent condition has been used to analyze the global convergence of conjugate gradient algorithms with inexact line searches.

**Table 1: Choices for $\beta_k$ parameter in conjugate gradient algorithms.**

| | |
|---|---|
| $\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$ | The original linear conjugate gradient algorithm by Hestenes and Stiefel [1952]. |
| $\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$ | The first nonlinear conjugate gradient algorithm, proposed by Fletcher and Reeves [1964]. |
| $\beta_k^{D} = \frac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$ | Proposed by Daniel [1967]. This updating formula requires evaluation of Hessian at every iteration. |
| $\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{g_k^T g_k}$ | Proposed by Polak and Ribière [1969] and Polyak [1969]. |
| $\beta_k^{PRP+} = max\left\{0, \frac{g_{k+1}^T y_k}{g_k^T g_k}\right\}$ | Proposed by Powell [1984], and analyzed by Gilbert and Nocedal [1992]. |
| $\beta_k^{CD} = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}$ | Proposed by Fletcher [1987] as a Conjugate descent method |
| $\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$ | Proposed by Liu and Storey [1991]. |
| $\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k}$ | Proposed by Dai and Yuan [1999]. |
| $\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k}, t > 0$ | Proposed by Dai and Liao [2001]. |
| $\beta_k^{DL+} = max\left\{0, \frac{g_{k+1}^T y_k}{d_k^T y_k}\right\} - t\frac{g_{k+1}^T s_k}{d_k^T y_k}$ | Proposed by Dai and Liao [2001] |
| $\beta_k^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}$ | Proposed by Yabe and Takano [2004], where $z_k = y_k + \frac{\rho \vartheta_k}{s_k^T u_k} u_k, \ \rho \geq 0,$ $\vartheta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$ |
| $\beta_k^{YT+} = max\left\{0, \frac{g_{k+1}^T z_k}{d_k^T z_k}\right\} - t\frac{g_{k+1}^T s_k}{d_k^T z_k}$ | Proposed by Yabe and Takano [2004]. |

Observe that the FR, DY and CD methods all have in numerator the quantity $g_{k+1}^T g_{k+1}$. The first result concerning the global convergence of FR method was given by Zoutendijk [1970]. Later, Powell [1977] showed that FR method is susceptible to jamming, that is the algorithm could take many steps without making a significant progress to minimum. The CD method of Fletcher is very close to the FR method, and with an exact line search we can prove that $\beta_k^{FR} = \beta_k^{CD}$. The difference between FR and CD is that CD satisfy the sufficient descent condition. The DY method is fundamentally different from either FR or the CD method. Using standard Wolfe line search conditions, the DY method always generates descent directions. Also DY and CD methods are susceptible to jamming.

On the other hand, the HS, PRP and LS methods all have in the numerator $g_{k+1}^T y_k$. The main characteristic of these methods is that they possess a built-in restart feature that addresses the jamming problem. When $s_k = x_{k+1} - x_k$ is small, the factor $y_k = g_{k+1} - g_k$ in the numerator of $\beta_k$ tends to

zero. Therefore, $\beta_k$ becomes small and the new direction $d_{k+1}$ is essentially the steepest descent direction given by $-g_{k+1}$. These methods automatically adjust $\beta_k$ to avoid jamming. Hence, the numerical performances of these methods are better than the performance of methods with $\|g_{k+1}\|^2$ in the numerator of $\beta_k$. For algorithms related to the Polak-Ribière-Polyak method, Gilbert and Nocedal [1992] proved the first global convergence result of the conjugate gradient algorithm with $\beta_k^{PRP+}$ using inexact line search. However, numerically, under inexact line search the conjugate gradient algorithm with $\beta_k^{PRP+}$ is not an improving of its variant using $\beta_k^{PRP}$.

The DL method is very close to HS, when $g_{k+1}$ is orthogonal to $s_k$, then $\beta_k^{DL} = \beta_k^{HS}$. Like HS method, for an exact line search, the DL method may not converge. In order to ensure convergence, Dai and Liao uses the same idea like in PRP+, modifying their formula to obtain DL+. If $d_k$ satisfies the sufficient descent condition, and $f$ is Lipschitz continuous, then DL+ scheme, implementing strong Wolfe line search, is globally convergent. Very recently Yabe and Takano [2004], based on a modified secant condition, suggested another choice for the update the $\beta_k$ parameter. In their formula, $u_k \in R^n$ satisfies $u_k^T s_k \neq 0$. Similarly, the YT+ scheme is globally convergent when $d_k$ satisfies the sufficient descent condition and the strong Wolfe conditions are used, with a bounded value of parameter $\rho$. Numerical experiments show that, both DL+ and YT+ are efficient only with a proper selection of the parameters. For different choices, the performances of these methods are very different.

The numerical experiments show that although the FR, DY and CD conjugate gradients methods have strong convergence properties; however they may be affected by jamming. On the other hand, the HS, PRP and LS methods although theoretically may not converge, computationally they often are significantly better. The idea to combine these methods in order to get efficient algorithms leads to *hybrid conjugate gradient algorithms*. Table 2 contains the main hybrid conjugate gradient methods.

**Table 2. Hybrid choices for the $\beta_k$ parameter in conjugate gradient algorithms.**

| | |
|---|---|
| $\beta_k^{TAS} = \begin{cases} \beta_k^{PRP}, & 0 \le \beta_k^{PRP} \le \beta_k^{FR}, \\ \beta_k^{FR}, & \text{otherwise.} \end{cases}$ | Proposed by Touat-Ahmed and Storey [1990]. |
| $\beta_k^{PRP-FR} = max\left\{0, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$ | Proposed by Hu and Storey [1991]. |
| $\beta_k^{GN} = max\left\{-\beta_k^{FR}, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$ | Proposed by Gilbert and Nocedal [1992]. |
| $\beta_k^{HS-DY} = max\left\{0, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$ | Proposed by Dai and Yuan [2001] and Dai and Ni [2003]. |
| $\beta_k^{hDY} = max\left\{-\left(\frac{1-\sigma}{1+\sigma}\right)\beta_k^{DY}, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$ | Proposed by Dai and Yuan [2001]. $\sigma$ is the parameter used in the second Wolfe line search condition. |
| $\beta_k^{LS-CD} = max\left\{0, min\left\{\beta_k^{LS}, \beta_k^{CD}\right\}\right\}$ | |

Conjugate gradient methods can be combined to get the so called *parameter conjugate gradient algorithms*. Dai and Yuan [1998, 2003] proposed a one-parameter family of conjugate gradient algorithms with:

$$\beta_k = \frac{\|g_{k+1}\|^2}{\lambda_k \|g_k\|^2 + (1-\lambda_k) d_k^T y_k}, \tag{12}$$

where $\lambda_k \in [0,1]$ is a parameter. For $\lambda_k = 1$ we get the Fletcher-Reeves method, while the Dai-Yuan method correspond to $\lambda_k = 0$. By considering convex combinations of the numerators and denominators of $\beta_k^{FR}$ and $\beta_k^{HS}$, Nazareth [1999] suggested a two-parameter family of conjugate gradient methods:

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1-\mu_k) g_{k+1}^T y_k}{\lambda_k \|g_k\|^2 + (1-\lambda_k) d_k^T y_k}, \tag{13}$$

where $\lambda_k, \mu_k \in [0,1]$. Observe that this two-parameter family includes FR, DY, PRP and HS methods. Dai and Yuan [2001] considered a three-parameter family of hybrid conjugate gradient method; they chose:

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1-\mu_k) g_{k+1}^T y_k}{(1-\lambda_k-\omega_k)\|g_k\|^2 + \lambda_k d_k^T y_k - \omega_k d_k^T g_k}, \tag{14}$$

where $\lambda_k, \mu_k \in [0,1]$ and $\omega_k \in [0, 1-\lambda_k]$. This three-parameter family includes the six standard conjugate gradient methods, the previous one-parameter and two-parameter families, as well as many hybrid methods as special cases.

Another class of conjugate gradient algorithms is given by the so called the *scaled conjugate gradient algorithms*. For these algorithms the direction is computed as:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k d_k, \tag{15}$$

where $\theta_{k+1}$ is a positive parameter.

Observe that if $\theta_{k+1} = 1$, then we get the classical conjugate gradient algorithms according to the value of the scalar parameter $\beta_k$. On the other hand, if $\beta_k = 0$, then we get another class of algorithms according to the selection of the parameter $\theta_{k+1}$. There are two possibilities for $\theta_{k+1}$ : a positive scalar or a positive definite matrix. If $\theta_{k+1} = 1$, 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 (15) represents a combination between the quasi-Newton and conjugate gradient methods.

To determine $\beta_k$ Andrei [2004a, 2004b, 2005, 2006a, 2006b] considers the following procedure. As we know the Newton direction for solving (1) is given by $d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}$. Therefore, from the equality

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k d_k,$$

we get:

$$\beta_k = \frac{\theta_{k+1} s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k.} \tag{16}$$

Using the Taylor development, after some algebra we obtain:

$$\beta_k = \frac{\left(\theta_{k+1} y_k - s_k\right)^T g_{k+1}}{y_k^T s_k}, \tag{17}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. If $\theta_{k+1} = 1$, then (17) is the parameter corresponding to the direction considered by Perry [1977], i.e. we get the *scaled Perry* algorithm. This value of parameter $\beta_k$ is used by Birgin and Martinez [2001] in their SCG (spectral conjugate gradient) package for unconstrained optimization, where $\theta_{k+1}$ is selected in a spectral manner, as suggested by Raydan [1997]. The following particularizations can be remarked. If $s_j^T g_{j+1} = 0,\ j = 0,1,\ldots,k,$ then we get a generalization of the Polak and Ribière formula [1969], i.e. *the scaled Polak and Ribière* formula. If $s_j^T g_{j+1} = 0,\ j = 0,1,\ldots,k,$ and additionally the successive gradients are orthogonal, then we obtain a generalization of the Fletcher and Reeves formula [1964] i.e. *the scaled Fletcher and Reeves* formula. Table 3 summarizes the scaled choices for the parameter $\beta_k$.

**Table 3. Scaled choices for the $\beta_k$ parameter in conjugate gradient algorithms.**

| | |
|---|---|
| $\beta_k^{sP} = \frac{g_{k+1}^T(\theta_k y_k - s_k)}{y_k^T s_k}$ | Scaled Perry. Suggested by Birgin and Martínez [2001] and Andrei [2004a, 2004b, 2005, 2006a, 2006b]. |
| $\beta_k^{sP+} = max\left\{0, \frac{\theta_k g_{k+1}^T y_k}{y_k^T s_k}\right\} - \frac{g_{k+1}^T s_k}{y_k^T s_k}$ | Scaled Perry +. Suggested by Birgin and Martínez [2001]. |
| $\beta_k^{sPRP} = \frac{\theta_k g_{k+1}^T y_k}{t_k \theta_{k-1} g_k^T g_k}$ | Scaled Polak-Ribière-Polyak. Suggested by Birgin and Martínez [2001] and Andrei [2004a, 2004b, 2005, 2006a, 2006b]. |
| $\beta_k^{sFR} = \frac{\theta_k g_{k+1}^T g_{k+1}}{t_k \theta_{k-1} g_k^T g_k}$ | Scaled Fletcher-Reeves. Suggested by Birgin and Martínez [2001] and Andrei [2004a, 2004b, 2005, 2006a, 2006b]. |

The parameter $\theta_{k+1}$ is selected in a *spectral manner*, $\theta^s$, as the inverse of the Rayleigh quotient or in an *anticipative manner*, $\theta^a$, as given in Andrei [2004a, 2004b, 2005, 2006a, 2006b].

There is a result of Shanno [1978] that says that the conjugate gradient method is the BFGS quasi-Newton method for which at every iteration, the initial approximation to the inverse of the Hessian is taken as the identity matrix. Shanno [1978] shows how the traditional Fletcher-Reeves and Polak-Ribière conjugate gradient algorithms may be modified in a form established by Perry to a sequence which can be considered as a *memoryless BFGS preconditioned.* The algorithm is embedded into a restarting procedure based on Powell's restart criterion. The idea of the algorithm is to modify the direction in such a manner to overcome the lack of positive definiteness of the matrix defining the search direction. The CONMIN algorithm based on this technique proved to be one of the most powerful for solving large-scale unconstrained optimization problems. The extension of the preconditioning technique to the scaled conjugate gradient is very simple. Using the same methodology Andrei [2004a, 2004b, 2005, 2006a, 2006b] obtained the following direction $d_{k+1}$:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1}\left(\frac{g_{k+1}^T s_k}{y_k^T s_k}\right) y_k - \left[\left(1 + \theta_{k+1}\frac{y_k^T y_k}{y_k^T s_k}\right)\frac{g_{k+1}^T s_k}{y_k^T s_k} - \theta_{k+1}\frac{g_{k+1}^T y_k}{y_k^T s_k}\right] s_k, \tag{18}$$

Observe that if $g_{k+1}^T s_k = 0,$ then (18) reduces to:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1} \frac{g_{k+1}^T y_k}{y_k^T s_k} s_k . \tag{19}$$

Thus, in this case, the effect is simply one of multiplying the Hestenes and Stiefel [1952] search direction by a positive scalar. Thus, we get another class of scaled memoryless BFGS preconditioned conjugate gradient algorithms. The implementation of the direction (18), in the context of Powell restarts, is given by Andrei [2004a, 2004b, 2005, 2006a, 2006b] in SCALCG package. The scaling factor is selected as the spectral gradient or in an anticipative manner as it is described in [Andrei, 2004a, 2004b, 2005, 2006a, 2006b].

Recently, for solving (1), Hager and Zhang [2004a, 2004b] presented a *conjugate gradient algorithm with guaranteed descent* and the performance of the Fortran 77 package CG_DESCENT which implements it. The directions $d_k$ are computed by the following rule:

$$d_{k+1} = -g_{k+1} + \beta_k d_k , \tag{20}$$

$$\beta_k = \frac{1}{d_k^T y_k} \left( y_k - 2 \frac{y_k^T y_k}{d_k^T y_k} d_k \right)^T g_{k+1} , \tag{21}$$

$d_0 = -g_0$. We see that the above direction $d_{k+1}$ of Hager and Zhang, given in (20)-(21), denoted $d_{k+1}^{HZ}$, can be expressed as:

$$d_{k+1}^{HZ} = -g_{k+1} - \frac{y_k^T y_k}{s_k^T y_k} \left[ 2 \frac{s_k^T g_{k+1}}{s_k^T y_k} - \frac{y_k^T g_{k+1}}{y_k^T y_k} \right] s_k . \tag{22}$$

On the other hand, the direction generated by Shanno, let call it $d_{k+1}^{S}$, is given as :

$$d_{k+1}^{S} = -\frac{s_k^T y_k}{y_k^T y_k} g_{k+1} - \left[ 2 \frac{s_k^T g_{k+1}}{s_k^T y_k} - \frac{y_k^T g_{k+1}}{y_k^T y_k} \right] s_k + \frac{s_k^T g_{k+1}}{y_k^T y_k} y_k . \tag{23}$$

After some simple algebra we see that the direction $d_{k+1}^{HZ}$ of Hager and Zhang is related to the direction $d_{k+1}^{S}$ of Shanno in the following way:

$$d_{k+1}^{S} = \frac{s_k^T y_k}{y_k^T y_k} \left( d_{k+1}^{HZ} + \frac{s_k^T g_{k+1}}{s_k^T y_k} y_k \right), \quad \text{or} \quad d_{k+1}^{HZ} = \frac{y_k^T y_k}{s_k^T y_k} d_{k+1}^{S} - \frac{s_k^T g_{k+1}}{s_k^T y_k} y_k .$$

With other words, the direction $d_{k+1}^{HZ}$ is obtained from $d_{k+1}^{S}$ by deleting the term

$$\frac{s_k^T g_{k+1}}{y_k^T y_k} y_k . \tag{24}$$

In their algorithm Hager and Zhang restrict $\beta_k$ to be nonnegative. This is motivated by the work of Gilbert and Nocedal [1992] who modified the Polak and Ribière updating formula as $\beta_k^+ = max\{\beta_k, 0\}$ and proved the global convergence of this computational scheme for general nonlinear functions. Similar to the approaches considered by Gilbert and Nocedal [1992], Han, Liu, Sun and Yin [1994], and Wang, Han and Wang [2000] in their studies on the Polak-Ribière version of the conjugate gradient method, Hager and Zhang prove the convergence for general nonlinear functions by restricting the lower bound of $\beta_k$ in the following manner:

$$d_{k+1} = -g_{k+1} + \bar{\beta}_k d_k, \tag{25}$$

$$\bar{\beta}_k = \max\{\beta_k, \eta_k\}, \tag{26}$$

$$\eta_k = \frac{-1}{\|d_k\| min\{\eta, \|g_k\|\}}, \tag{27}$$

where $\beta_k$ is given by (21), and the parameter $\eta > 0$ is a user specified constant. (Suggested value: $\eta = 0.01$, considered in all numerical experiments).

An important innovation given by Hager and Zhang in their approach consists of a *new efficient and highly accurate line search procedure*. This is based on the Wolfe conditions (9a)-(9b) and on a very fine interpretation of the numerical issue concerning the first Wolfe condition (9a). Intensive numerical experiments with conjugate gradient algorithms show that the SCALCG package is top performer among conjugate gradient algorithms [Andrei, 2004a, 2004b, 2005, 2006a, 2006b],

# 3. Performance of Conjugate Gradient Algorithms on some MINPACK-2 Unconstrained Optimization Applications

In the following we present the performances of 14 conjugate gradient algorithms on 6 applications from MINPACK-2 collection [Averick, Carter and Moré, 1991], [Averick, Carter, Moré, and Xue Guo-Liang, 1992]. In this numerical study we consider only the performances of SCALCG [Andrei, 2004a, 2004b, 2005, 2006a, 2006b], SCG [Birgin and Martínez, 2001], CONMIN [Shanno and Phua, 1976], sPR, sFR, Perry-Shanno, Dai-Yuan, Dai-Liao, Dai Liao+, hHS-DY, HS, PRP, PRP+ and FR. All codes are written in Fortran 77 (default compiler settings) on a workstation Intel 4, 1.8GHz. SCALCG is authored by Andrei, CONMIN is co-authored by Shanno and Phua, and SCG by Birgin and Martínez. The rest of algorithms are implemented by Andrei. In all these numerical experiments we have considered the standard initial point, as it is recommended in MINPACK-2 and the iterations were stopped according to the following criteria:

$$\|g_k\|_\infty \le 10^{-5} \quad \text{or} \quad t_k \left| g_k^T d_k \right| \le 10^{-20} \left| f(x_{k+1}) \right|.$$

SCALCG packages uses the Powell restart criterion. SCG and scaled variants of Polak-Ribière-Polyak or Fletcher-Reeves use the angle restart criterion (see [Andrei, 2005]). All these algorithms uses the same implementation of the Wolfe line search procedure with the same values of parameters $\sigma_1$ and $\sigma_2$.

## 3.1. Elastic-Plastic Torsion

This problem consists of determination of the stress field on an infinitely long cylindrical bar. The infinite-dimensional version of the problem is as follows:

$$\min\{q(v) : v \in K\},$$

where $q : K \to R$ is the quadratic function

$$q(v) = \frac{1}{2}\int_D \|\nabla v(x)\|^2 dx - c\int_D v(x)dx$$

for some constant $c$ and $D$ is a bounded domain with smooth boundary. The convex set $K$ is defined by

$$K = \left\{ v \in H_0^1(D) : \left| v(x) \le dist(x, \partial D),\ x \in D \right| \right\},$$

where $dist(.,\partial D)$ is the distance function to the boundary of $D$, and $H_0^1(D)$ is the Hilbert space of all functions with compact support in $D$ such that $v$ and $\|\nabla v\|^2$ belong to $L^2(D)$. This formulation, as well as the physical interpretation of the torsion problem are presented by Glowinski [1984, pp.41-55]. The finite element approximation to the torsion problem is obtained by triangulating $D$ and replacing the minimization of $q$ over $H_0^1(D)$ by the minimization of $q$ over the set of picewise linear functions that satisfy the constraints specified by $K$, as is given by Averick, Carter and Moré [1991, pp.21-23]. More specifically, the finite element approximation considered by Averick, Carter and Moré is for a quadratic $q$ is of the general form:

$$q(v) = \frac{1}{2} \int_D w_q(x) \|\nabla v(x)\|^2 dx - \int_D w_l(x) v(x) dx ,$$

where $w_q : D \to R$ and $w_l : D \to R$ are functions defined on the rectangle $D$. In the torsion problem $w_q = 1$ and $w_l = c.$ This problem is solved by triangulation of $D$ by choosing $n_x \times n_y$ grid points inside of $D$.

Considering $D = (0,1) \times (0,1)$ and $c = 5$ the performances of the optimization packages considered in this numerical study are given in Tables 4 and 5.

**Table 4: Performance of Conjugate Gradient Algorithms**
***Elastic-Plastic Torsion Problem.***
$nx = 100,\ ny = 100,\ c = 5.,\ n = 10000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 217 | **284** | **12.20** | 255 | **338** | **14.45** |
| SCG | 193 | 291 | 16.04 | 233 | 356 | 19.60 |
| sPRP | 229 | 363 | 19.77 | 243 | 389 | 21.09 |
| sFR | 274 | 430 | 23.40 | 215 | 342 | 18.51 |
| Perry-Shanno | 237 | 361 | 19.83 | 259 | 396 | 21.76 |
| Dai-Yuan | 318 | 496 | 27.08 | 245 | 379 | 20.70 |
| Dai-Liao | 286 | 432 | 23.89 | 279 | 423 | 23.34 |
| Dai-Liao + | 276 | 428 | 23.35 | 274 | 422 | 23.12 |
| hHS-DY | 211 | 330 | 17.96 | 248 | 391 | 21.26 |
| HS | 262 | 412 | 22.41 | 260 | 407 | 22.25 |
| PRP | 237 | 391 | 20.98 | 264 | 443 | 23.68 |
| PRP+ | 237 | 391 | 20.98 | 264 | 443 | 23.61 |
| FR | 247 | 412 | 22.02 | 201 | 339 | 18.07 |

**Table 5. Performance of Conjugate Gradient Algorithms**
***Elastic-Plastic Torsion Problem.***
$nx = 200,\ ny = 200,\ c = 5.,\ n = 40000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 398 | 511 | **87.39** | 473 | 614 | **104.80** |
| SCG | 370 | 573 | 126.77 | 427 | 659 | 146.15 |
| sPRP | 466 | 743 | 163.13 | 419 | 669 | 146.65 |
| sFR | 387 | 620 | 135.34 | 394 | 612 | 134.73 |
| Perry-Shanno | 445 | 684 | 151.43 | 400 | 619 | 136.60 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Dai-Yuan | 439 | 677 | 149.61 | 511 | 810 | 177.68 |
| Dai-Liao | 452 | 711 | 156.76 | 326 | **510** | 113.32 |
| Dai-Liao + | 497 | 772 | 170.38 | 338 | 529 | 116.11 |
| hHS-DY | 398 | 625 | 137.75 | 379 | 591 | 130.01 |
| HS | 401 | 620 | 136.82 | 431 | 675 | 148.41 |
| PRP | 291 | **486** | 104.47 | 475 | 807 | 174.45 |
| PRP+ | 291 | **486** | 104.52 | 475 | 807 | 173.62 |
| FR | 445 | 748 | 161.76 | 518 | 886 | 190.20 |

Numerical results for the elastic-plastic torsion problem are presented, for example, by Elliot and Ockendon [1982], O'Leary and Yang [1989] and Moré and Toraldo [1991]. The solution for $nx = 10,\ ny = 10$ is illustrated in Figure 1.

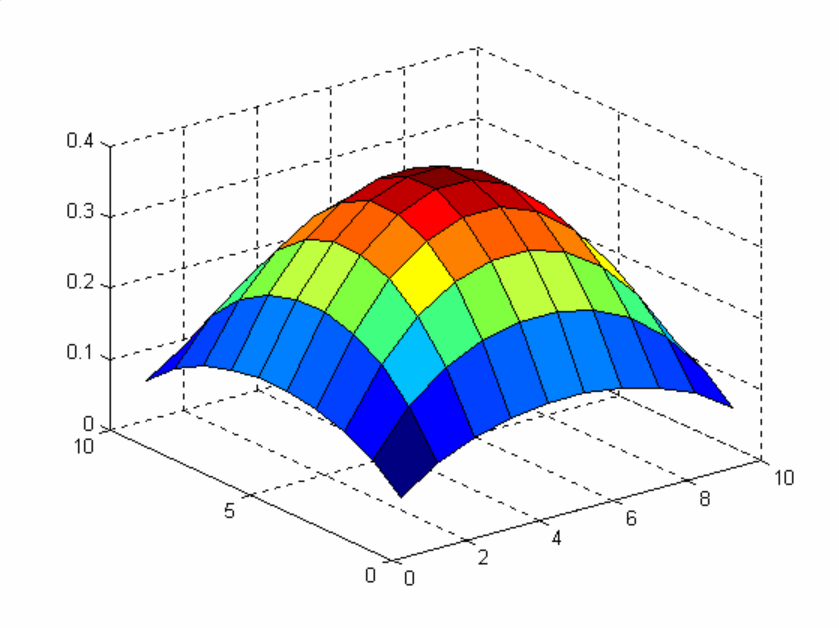


**Figure: 1. Elastic-Plastic Torsion solution.** $nx = 10,\ ny = 10$.

## 3.2. Pressure Distribution in a Journal Bearing

This problem consists of determining the pressure distribution in a thin film of lubricant between two circular cylinders. The infinite-dimensional version of the problem is of the form:

$$min\{q(v) : v \in K\}, \qquad q(v) = \frac{1}{2}\int_D w_q(x)\|\nabla v(x)\|^2 dx - \int_D w_l(x)v(x)dx$$

with

$$w_q(z_1,z_2) = (1+\varepsilon\cos z_1)^3, \quad w_l(z_1,z_2) = \varepsilon\sin z_1$$

for some constant $\varepsilon \in (0,1)$ and $D = (0,2\pi)\times(0,2b)$ for some constant $b > 0$. The convex set $K$ is defined by $K = \{v \in H_0^1(D) : v \in D\}$. Considering $b = 10$ and $\varepsilon = 0.1$, then the performances of the optimization packages considered are given in Tables 6 and 7.

**Table 6. Performance of Conjugate Gradient Algorithms**
***Pressure Distribution in a Journal Bearing***
$nx = 100,\ ny = 100,\ ecc = 0.1,\ b = 10,\ n = 10000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **Cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 433 | **567** | **23.51** | 461 | 620 | **25.71** |
| SCG | 617 | 956 | 53.28 | 443 | 692 | 38.40 |
| sPRP | 523 | 838 | 46.08 | 652 | 1035 | 57.23 |
| sFR | 495 | 777 | 42.79 | 546 | 864 | 47.62 |
| Perry-Shanno | 453 | 709 | 39.33 | 503 | 783 | 43.61 |
| Dai-Yuan | 532 | 835 | 46.08 | 506 | 792 | 43.77 |
| Dai-Liao | 443 | 696 | 38.39 | 546 | 854 | 47.41 |
| Dai-Liao + | 476 | 744 | 41.03 | 510 | 797 | 44.05 |
| hHS-DY | 485 | 755 | 41.69 | 421 | 661 | 36.36 |
| HS | 458 | 726 | 39.87 | 409 | 636 | 35.15 |
| PRP | 530 | 856 | 46.69 | 474 | 767 | 41.80 |
| PRP+ | 530 | 856 | 46.69 | 474 | 767 | 41.80 |
| FR | 590 | 898 | 49.98 | 312 | **501** | 27.24 |

**Table 7. Performance of Conjugate Gradient Algorithms**
***Pressure Distribution in a Journal Bearing***
$nx = 200,\ ny = 200,\ ecc = 0.1,\ b = 10,\ n = 40000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 876 | **1143** | **189.11** | 900 | **1157** | **191.14** |
| SCG | 1117 | 1746 | 387.00 | 1115 | 1749 | 386.41 |
| sPRP | 980 | 1556 | 341.86 | 1120 | 1775 | 390.52 |
| sFR | 874 | 1399 | 305.49 | 839 | 1326 | 290.17 |
| Perry-Shanno | 848 | 1332 | 294.78 | 1101 | 1745 | 385.35 |
| Dai-Yuan | 1098 | 1721 | 379.16 | 1206 | 1888 | 415.35 |
| Dai-Liao | 959 | 1508 | 333.50 | 1171 | 1823 | 403.81 |
| Dai-Liao + | 975 | 1530 | 336.37 | 1213 | 1907 | 419.25 |
| hHS-DY | 1096 | 1722 | 379.42 | 918 | 1449 | 317.69 |
| HS | 872 | 1386 | 303.96 | 1022 | 1609 | 353.99 |
| PRP | † | | | 1033 | 1549 | 344.82 |
| PRP+ | † | | | 1033 | 1549 | 344.11 |
| FR | 1101 | 1758 | 384.04 | 2235 | 2860 | 676.02 |

† Failure, $f(x_{207}) = 0.366117$ , $\|g_{207}\|_2 = 0.2628$ .

Numerical results for the journal bearing problem can be found, for example, in Lin and Cryer [1985], Cimatti and Menchi [1978] and Moré and Toraldo [1991]. Figure 2 illustrates the solution of the problem for $nx = 10, ny = 10$ .

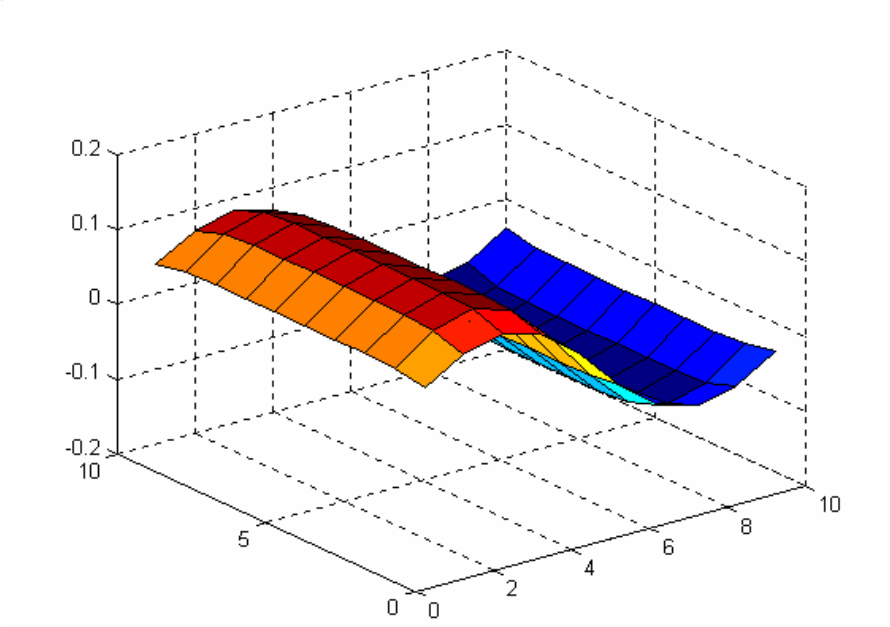


**Figure 2: Pressure Distribution in a Journal Bearing solution.** $nx = 10,\ ny = 10$ **.**

### 3.3. Optimal Design with Composite Materials

This problem requires determining the placement of two elastic materials in the cross-section of a rod with maximal torsional rigidity. The formulation of the problem is given in Goodman, Kohn and Reyna [1986] and Averick, Carter and Moré [1991].

Let $D \subset R^2$ be a bounded domain, and let $w < |D|$, where $|D|$ denotes the area of $D$. The problem is formulated as:

$$min\left\{F(v,\Omega) : v \in H_0^1(D),\ |\Omega| = w\right\},$$

where

$$F(v,\Omega) = \int_D \left\{\frac{1}{2}\mu(x)\|\nabla v(x)\|^2 + v(x)\right\}dx,$$

and

$$\mu(x) = \mu_1 \text{ for } x \in \Omega \text{, and } \mu(x) = \mu_2 \text{ for } x \notin \Omega.$$

The reciprocals of the constants $\mu_1$ and $\mu_2$ are the shear moduli of the elastic materials in the rod. It is assumed that $\mu_1 < \mu_2$.

Goodman, Kohn and Reyna [1986] give some details and formulate the optimal design problem in terms of a family of problems of the following form:

$$min\left\{f_\lambda(v) : v \in H_0^1(D)\right\},$$

where $f_\lambda : H_0^1(D) \to R$ is the functional

$$f_\lambda(v) = \int_D \left\{ \psi_\lambda\left( \left\| \nabla v(x) \right\| \right) + v(x) \right\} dx$$

and $\psi_\lambda : R \to R$ is a picewise quadratic. In this formulation $\lambda$ is the Lagrange multiplier associated to the optimal design problem, and the picewise quadratic $\psi_\lambda$ is of the form:

$$\psi_\lambda(t) = \begin{cases} \frac{1}{2}\mu_2 t^2, & 0 \le t \le t_1, \\ \mu_2 t_1 (t - \frac{1}{2} t_1), & t_1 \le t \le t_2, \\ \frac{1}{2}\mu_1 (t^2 - t_2^2) + \mu_2 t_1 (t_2 - \frac{1}{2} t_1), & t_2 \le t, \end{cases}$$

with the breakpoints $t_1$ and $t_2$ defined by

$$t_1 = \left( 2\lambda \frac{\mu_1}{\mu_2} \right)^{1/2} \text{ and } t_2 = \left( 2\lambda \frac{\mu_2}{\mu_1} \right)^{1/2}.$$

The problem considered in Averick, Carter and Moré [1991] is that of minimizing $f_\lambda$ for a fixed value of $\lambda$. A finite element approximation to this problem is obtained, in a canonical manner, by minimizing $f_\lambda$ over the space of picewise linear functions $v$ with values $v_{ij}$ at $z_{ij}$, where $z_{ij} \in R^2$ are the vertices of a triangulation of $D$ with grid spacings $h_x$ and $h_y$. Considering $\mu_1 = 1$ and $\mu_2 = 2$, the Tables 8 and 9 give the performances of the packages for different values of discretization $n_x$ and $n_y$.

**Table 8. Performance of Conjugate Gradient Algorithms**
***Optimal Desing with Composite materials.***
$nx = 100,\ ny = 100,\ c = 5.,\ n = 10000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 628 | **801** | **49.49** | 644 | **815** | **50.31** |
| SCG | 1519 | 2309 | 169.99 | 1249 | 1907 | 140.11 |
| sPRP | 1461 | 2280 | 166.65 | 1314 | 2069 | 150.72 |
| sFR | 1451 | 2295 | 166.81 | 1421 | 2229 | 162.09 |
| Perry-Shanno | 1353 | 2060 | 152.31 | 1256 | 1926 | 141.87 |
| Dai-Yuan | 1147 | 1744 | 128.26 | 1337 | 2035 | 149.67 |
| Dai-Liao | 1543 | 2355 | 173.95 | 1365 | 2074 | 152.53 |
| Dai-Liao + | 1603 | 2426 | 178.07 | 1165 | 1788 | 130.72 |
| hHS-DY | 1211 | 1835 | 134.68 | 1397 | 2140 | 156.54 |
| HS | 1452 | 2222 | 162.80 | 1265 | 1935 | 142.26 |
| PRP | † | | | ♦ | | |
| PRP+ | † | | | ♦ | | |
| FR | ‡ | | | ♣ | | |

† Failure, $f(x_{25}) = 0.00856922$, $\|g_{25}\|_2 = 0.111017$.

‡ Failure, $f(x_{68}) = 0.00427538$, $\|g_{68}\|_2 = 0.0691171$.

♦ Failure, $f(x_{68}) = -0.000109309$, $\|g_{68}\|_2 = 0.0623754$.

♣ Failure, $f(x_{267}) = -0.0096339$, $\|g_{267}\|_2 = 0.0155195$.

**Table 9. Performance of Conjugate Gradient Algorithms**
***Optimal Desing with Composite materials.***
$nx = 200,\ ny = 200,\ c = 5.,\ n = 40000.$

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 1413 | **1753** | **431.82** | 939 | **1157** | **285.06** |
| SCG | 3952 | 6003 | 1758.38 | 3184 | 4828 | 1414.88 |
| sPRP | 4528 | 7148 | 2073.66 | 4567 | 7158 | 2080.25 |
| sFR | 3957 | 6139 | 1783.10 | 4019 | 6227 | 1810.51 |
| Perry-Shanno | 4278 | 6471 | 1908.49 | 4246 | 6396 | 1887.90 |
| Dai-Yuan | 4485 | 6854 | 1999.07 | 4477 | 6819 | 1990.06 |
| Dai-Liao | 4371 | 6674 | 1950.40 | 3767 | 5779 | 1691.44 |
| Dai-Liao + | 4080 | 6157 | 1801.22 | 4027 | 6166 | 1796.72 |
| hHS-DY | 3062 | 4643 | 1356.66 | 3782 | 5723 | 1672.48 |
| HS | 3990 | 6071 | 1772.89 | 3936 | 5990 | 1746.41 |
| PRP | † | | | ♦ | | |
| PRP+ | † | | | ♦ | | |
| FR | ‡ | | | ♣ | | |

† Failure, $f(x_{79}) = 0.0122556$, $\|g_{79}\|_2 = 0.0834894$.

‡ Failure, $f(x_{111}) = 0.0026687$, $\|g_{111}\|_2 = 0.0369373$.

♦ Failure, $f(x_{354}) = -0.00748145$, $\|g_{354}\|_2 = 0.0312285$.

♣ Failure, $f(x_{56}) = 0.0105501$, $\|g_{56}\|_2 = 0.0851817$.

Figure 3 presents the solution of the problem for discretization $nx = 10,\ ny = 10$.

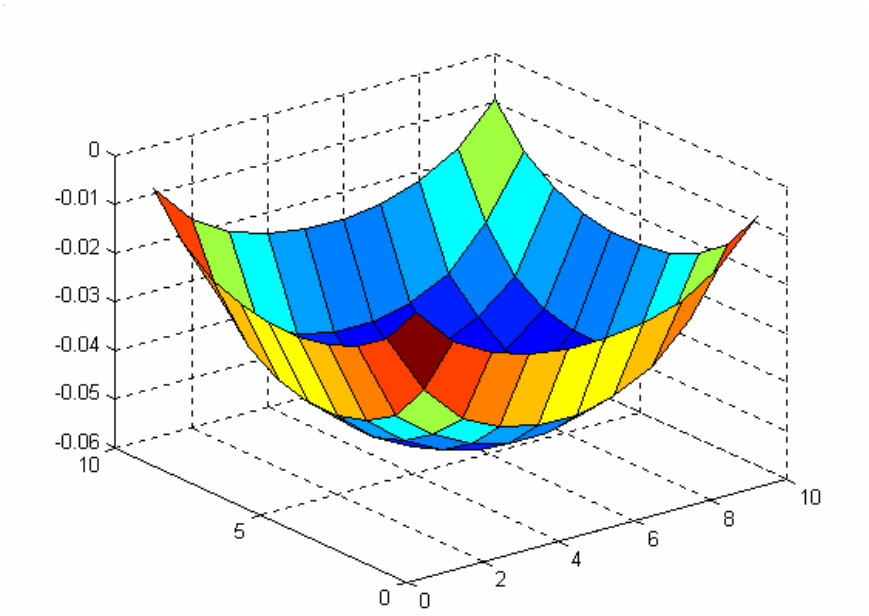


**Figure 3: Optimal Design with Composite Materials solition.** $nx = 10,\ ny = 10$ **.**

### 3.4. Inhomogeneous Superconductors

This problem appears in the solution of Ginzburg-Landau equations for inhomogeneous superconductors in the absence of a magnetic field. The one-dimensional system consists of alternating layers of lead and tin. The formulation of the problem is given by Garner and Benedek [1990] and Averick, Carter and Moré [1991].

The optimization problem is to minimize the Gibbs free energy as a function of the temperature. The infinite-dimensional version of the problem is of the form

$$min\left\{ f(v) : v(-d) = v(d),\ v \in C^1[-d,d] \right\},$$

where $2d$ is the width of the material, and $f$ is the Gibbs free energy function:

$$f(v) = \frac{1}{2d}\int_{-d}^{d}\left\{ \alpha(\xi)|v(\xi)|^2 + \frac{1}{2}\beta(\xi)|v(\xi)|^4 + \frac{\hbar}{4m}|v'(\xi)|^2 \right\} d\xi,$$

where the functions $\alpha$ and $\beta$ are picewise constant for a fixed value of the temperature, $\hbar$ is the Planck's constant (1.05459e-27 erg-sec), and $m$ is the mass of the electron (9.11e-28 grams).

The functions $\alpha$ and $\beta$ are constant in the intervals that correspond to the lead and the tin. The finite element approximation to the superconductivity problem is obtained by minimizing $f$ over the space of picewise linear functions $v$ with values $v_i$ at points $t_i$, where

$$-d = t_1 < t_2 < \cdots < t_n = d.$$

Therefore the values $v_i$ are obtained as solution of the following minimization problem

$$min\left\{\frac{1}{2d}\sum_{i=1}^{n} f_i(v) : v \in R^n\right\},$$

where

$$f_i(v) = h_i\left\{\frac{\alpha_i}{3}\frac{v_{i+1}^3 - v_i^3}{v_{i+1} - v_i} + \frac{\beta_i}{5}\frac{v_{i+1}^5 - v_i^5}{v_{i+1} - v_i} + \frac{\hbar}{4m}\left(\frac{v_{i+1} - v_i}{h_i}\right)^2\right\},$$

with $h_i = t_{i+1} - t_i$ is the length of the $i-$th subinterval, and the constants $\alpha_i$ and $\beta_i$ the values of the functions $\alpha$ and $\beta$ in the subinterval $[t_i, t_{i+1}]$.

**Table 10. Performance of Conjugate Gradient Algorithms**
***Inhomogeneous Superconductors.***
$t = 7,\ n = 1000$ **.**

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 7422 | **9499** | **141.32** | 5560 | **7107** | **108.04** |
| SCG | 10001 | 15724 | 230.02 | 10001 | 15664 | 229.92 |
| sPRP | 10001 | 15976 | 236.62 | 10001 | 16028 | 237.66 |
| sFR | 10001 | 15860 | 235.57 | 10001 | 15864 | 236.23 |
| Perry-Shanno | 10001 | 15737 | 236.51 | 10001 | 15675 | 232.78 |
| Dai-Yuan | 10001 | 15736 | 233.60 | 10001 | 15697 | 233.10 |
| Dai-Liao | 10001 | 15759 | 235.19 | 10001 | 15706 | 233.98 |
| Dai-Liao + | 10001 | 15747 | 232.34 | 10001 | 15738 | 233.66 |
| hHS-DY | 10001 | 15714 | 233.81 | 10001 | 15677 | 228.05 |
| HS | 10001 | 15710 | 234.32 | 10001 | 15720 | 233.38 |
| PRP | 10001 | 10931 | 170.71 | 10001 | 10124 | 159.50 |
| PRP+ | 10001 | 10931 | 170.15 | 10001 | 10124 | 159.45 |
| FR | 10001 | 11262 | 174.61 | 10001 | 11753 | 181.14 |

In Figure 4 it is shown the solution of the problem for $n = 100$.

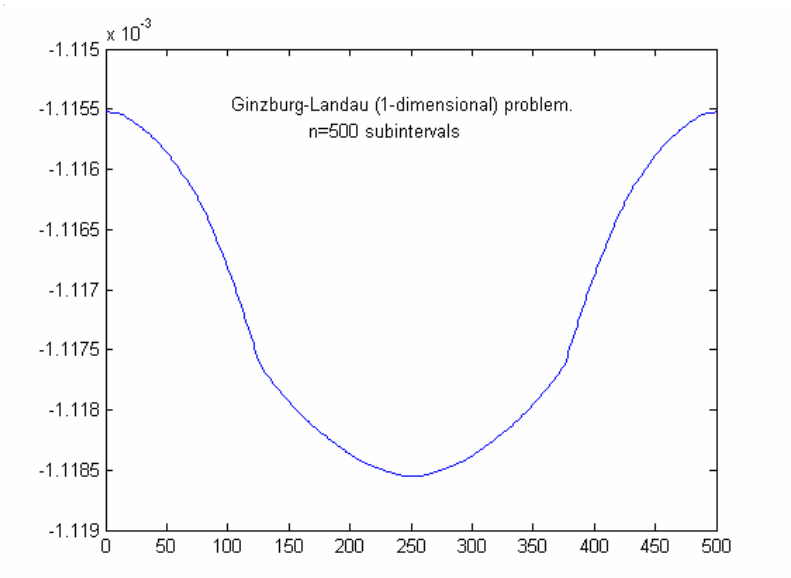


**Figure 4: Ginzburg-Landau (1-dimensional) solution. n=100.**

## 3.5. Lennard-Jones Clusters. Molecular Conformation.

The molecular conformation problem consists of determination of the minimum energy configuration of a cluster of atoms or molecules. This is a central problem in the study of cluster statics [Hoare, 1979] Given the positions $p_1, p_2, \ldots, p_n$ of $n$ molecules (points) in $R^d$, the energy potential function is defined as

$$V_d(p) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} v\left( \left\| p_j - p_i \right\|_2 \right),$$

where $v : R \to R$ is the potential function between pairs of atoms. The Lennard-Jones potential function is defined as

$$v(r) = r^{-12} - 2r^{-6}.$$

The molecular conformation problem is to determine a configuration (position for the $n$ points) such that the energy function $V_d$ is minimized. Table 11 shows the results of the conjugate gradient algorithms considered in this study.

**Table 11. Performance of Conjugate Gradient Algorithms**
***Leonard-Jones Clusters. Molecular Conformation***
**ndim=3, natoms=1000,** $n = 3000$ **.**

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 1252 | **2002** | **383.16** | 1583 | **2007** | **384.37** |
| SCG | 4552 | 7103 | 1117.24 | 4656 | 7229 | 1137.89 |
| sPRP | 2058 | 3259 | 668.33 | 2202 | 3477 | 713.37 |
| sFR | 1711 | 2915 | 595.89 | 2127 | 3325 | 680.92 |
| Perry-Shanno | 2467 | 3886 | 796.14 | 2603 | 4141 | 848.38 |
| Dai-Yuan | 1932 | 3087 | 633.07 | 2889 | 4893 | 1001.84 |
| Dai-Liao | 1719 | 2780 | 569.85 | 1888 | 3124 | 639.55 |
| Dai-Liao + | 2989 | 4806 | 985.15 | 2934 | 4709 | 965.26 |
| hHS-DY | 1593 | 2512 | 515.25 | 1806 | 2981 | 610.11 |
| HS | 1791 | 2894 | 593.25 | 1440 | 2351 | 481.42 |
| PRP | 2464 | 3694 | 758.80 | 6998 | 9019 | 1858.57 |
| PRP+ | 2464 | 3694 | 758.68 | 6998 | 9019 | 1857.80 |
| FR | 5683 | 8090 | 1662.37 | 2208 | 3428 | 702.39 |

## 3.6. Steady State Combustion. Solid fuel ignition.

The study of the steady-state in solid fuel ignition models leads to the infinite-dimensional optimization problem

$$min\left\{ f_\lambda(v) : v \in H_0^\perp(D) \right\},$$

where $f_\lambda : H_0^\perp(D) \to R$ is the functional

$$f_\lambda(v) = \int_D \left\{ \frac{1}{2} \left\| \nabla v(x) \right\|^2 - \lambda \exp[v(x)] \right\} dx,$$

and $\lambda \geq 0$ is a parameter. This problem is the variational formulation of the boundary value problem

$$-\Delta v(x) = \lambda \exp[v(x)], \quad x \in D, \quad v(x) = 0 \text{ for } x \in \partial D$$

where $\Delta$ is the Laplacian operator. Aris [1975], Bebernes and Eberly [1989] discuss this problem in the context of combustion process.

The problem is solved using a finite element approximation, by minimizing $f$ over the space of piecewise linear functions $v$ with values $v_{ij}$ at $z_{ij}$, where $z_{ij} \in R^2$ are the vertices of a triangulation of $D$ with grid spacings $h_x$ and $h_y$. The values $v_{ij}$ are obtained as solution of the following minimization problem:

$$min\left\{\sum\left(f_{ij}^{L}(v) + f_{ij}^{U}(v)\right) : v \in R^n\right\},$$

where

$$f_{ij}^{L}(v) = \frac{h_x h_y}{4}\left\{\left(\frac{v_{i+1,j} - v_{ij}}{h_x}\right)^2 + \left(\frac{v_{i,j+1} - v_{ij}}{h_y}\right)^2 - \lambda \mu_{ij}^{L}\right\},$$

$$\mu_{ij}^{L} = \frac{2}{3}\left\{\exp(v_{ij}) + \exp(v_{i+1,j}) + \exp(v_{i,j+1})\right\},$$

$$f_{ij}^{U} = \frac{h_x h_y}{4}\left\{\left(\frac{v_{i-1,j} - v_{ij}}{h_x}\right)^2 + \left(\frac{v_{i,j-1} - v_{ij}}{h_y}\right)^2 - \lambda \mu_{ij}^{U}\right\},$$

$$\mu_{ij}^{U} = \frac{2}{3}\left\{\exp(v_{ij}) + \exp(v_{i-1,j}) + \exp(v_{i,j-1})\right\}.$$

In this formulation $f_{ij}^{L}$ is defined only when $0 \leq i \leq n_x$ and $0 \leq j \leq n_y$, while $f_{ij}^{U}$ is defined when $1 \leq i \leq n_x + 1$ and $1 \leq j \leq n_y + 1$. Tables 12 and 13 show the performances of the conjugate gradient algorithms considered in this study.

**Table 12. Performance of Conjugate Gradient Algorithms**
***Steady State Combustion. Solid fuel ignition.***
$nx = 100,\ ny = 100,\ \lambda = 0.07,\ n = 10000$ .

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 312 | **408** | **33.28** | 266 | **348** | **28.40** |
| SCG | 340 | 527 | 50.75 | 288 | 445 | 42.85 |
| sPRP | 316 | 494 | 47.01 | 230 | 361 | 34.27 |
| sFR | 298 | 478 | 45.21 | 307 | 486 | 46.08 |
| Perry-Shanno | 336 | 515 | 49.38 | 389 | 601 | 57.57 |
| Dai-Yuan | 400 | 632 | 59.97 | 311 | 488 | 46.30 |
| Dai-Liao | 431 | 682 | 64.87 | 275 | 431 | 40.98 |
| Dai-Liao + | 269 | 430 | 40.70 | 299 | 469 | 44.54 |
| hHS-DY | 364 | 564 | 53.66 | 336 | 524 | 49.76 |

| | | | | | | |
|---|---|---|---|---|---|---|
| HS | 313 | 481 | 45.87 | 290 | 446 | 42.46 |
| PRP | 383 | 639 | 60.08 | 434 | 726 | 68.28 |
| PRP+ | 383 | 639 | 60.09 | 434 | 726 | 68.21 |
| FR | 430 | 737 | 68.99 | 346 | 560 | 52.89 |

**Table 13. Performance of Conjugate Gradient Algorithms**
***Steady State Combustion. Solid fuel ignition.***
$nx = 200,\ ny = 200,\ \lambda = 0.07,\ n = 40000$ **.**

| | $\theta^s$ | | | $\theta^a$ | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **# iter** | **# fg** | **cpu (s)** | **# iter** | **# fg** | **cpu (s)** |
| SCALCG | 546 | **720** | **233.65** | 460 | **586** | **190.05** |
| SCG | 589 | 927 | 354.21 | 503 | 787 | 300.88 |
| sPRP | 532 | 845 | 319.39 | 613 | 966 | 365.63 |
| sFR | 637 | 1007 | 380.58 | 540 | 853 | 322.47 |
| Perry-Shanno | 463 | 736 | 279.19 | 622 | 977 | 372.01 |
| Dai-Yuan | 675 | 1048 | 397.33 | 555 | 869 | 328.45 |
| Dai-Liao | 479 | 756 | 285.94 | 564 | 892 | 337.35 |
| Dai-Liao + | 580 | 911 | 345.32 | 640 | 988 | 374.87 |
| hHS-DY | 697 | 1084 | 410.95 | 561 | 873 | 330.49 |
| HS | 681 | 1067 | 403.71 | 426 | 673 | 253.64 |
| PRP | 553 | 941 | 350.81 | 487 | 831 | 309.39 |
| PRP+ | 553 | 941 | 351.30 | 487 | 831 | 309.78 |
| FR | 636 | 1079 | 403.26 | 663 | 1129 | 421.50 |

In the following we present the performances of CONMIN by Shanno and Phua [1976] on these unconstrained optimization applications. CONMIN implements an algorithm which is very close to SCALCG. Both of them consider a BFGS modification of the Perry updating scheme which satisfy the quasi-Newton equation in a Beale-Powell restart environment. Mainly, the difference is in the scaling of gradient parameter. SCALCG considers a scalar approximation of the inverse Hessian, which prove to be efficient and robust in numerical experiments, while CONMIN uses a value minimizing the condition number of the matrix $H_k^{-1}H_{k+1}$. CONMIN implements the Wolfe line searches in a similar manner as they are implemented in SCALCG. Table 14 shows the performance of CONMIN on these unconstrained optimization MINPACK-2 applications.

**Table 14. Performances of CONMIN on 10 MINPACK-2 applications.**

| **Application** | $n$ | **#iter** | **# fg** | **cpu(s)** | $f(x^*)$ |
|---|---|---|---|---|---|
| Elastic-Plastic Torsion | 10000 | 217 | 439 | 17.90 | -0.43916320 |
| | 40000 | 242 | 486 | 78.76 | -0.43926781 |
| Pressure Distribution in a Journal Bearing | 10000 | 396 | 801 | 32.74 | -0.2828400078 |
| | 40000 | 819 | 1657 | 270.23 | -0.282892943 |
| Optimal Design with Composite Materials | 10000 | 359 | 729 | 40.76 | -0.011377240 |
| | 40000 | 675 | 1370 | 305.28 | -0.011381291 |
| Inhomogeneous Superconductors | 1000 | 9881 | 20001 | 270.12 | 0.11079927e-03 |
| Lennard-Jones Clusters. Molecular Conformation | 3000 | 880 | 1819 | 507.07 | -6605.4586893 |
| Steady State Combustion. Solid fuel ignition. | 10000 | 171 | 346 | 28.40 | -0.070086368 |
| | 40000 | 467 | 948 | 310.60 | -0.070086374 |

# 4. Conclusion

In this study we have presented a survey and comparative numerical performances of a number of well known conjugate gradient algorithms for solving some applications from MINPACK-2 collection. The conjugate gradient algorithms classifies in four groups: conjugacy condition algorithms, hybrid conjugate gradient, scaled conjugate gradient and preconditioned conjugate gradient algorithms. All conjugate gradient algorithms are able to solve a large variety of large-scale unconstrained optimization problems. The above Tables give computational evidence that our SCALCG - s*caled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization* is the top performer among the conjugate gradient algorithms. Close to SCALCG is CONMIN algorithm by Shanno and Phua. Both these algorithms use the same philosophy based on a BFGS preconditioning of the search direction in such a way to satisfy the quasi-Newton equation. Although SCALCG and CONMIN have a lot of algebra in common, we see that SCALCG is more performant than CONMIN. We see that SCALCG is with 488.32 seconds faster than CONMIN. The other conjugate gradient algorithms considered in this study are less robust showing that the conjugacy condition alone is unable to provide an efficient searching direction. The preconditioned variants of conjugate gradient algorithms (CONMIN and SCALCG) proved to be more robust and more efficient for solving large-scale nonlinear unconstrained optimization problems.

# REFERENCES


1. ANDREI, N., (2006a). **Scaled conjugate gradient algorithms for unconstrained optimization**. Accepted: Computational Optimization and Applications.

2. ANDREI, N., (2006b). **Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization**. Accepted: Optimization Methods and Software.

3. ARIS, R., (1975). **The mathematical theory of diffusion and reaction in permeable catalysts**. Oxford, 1975.

4. AVERICK, B.M., CARTER, R.G. and MORÉ, J.J., (1991). **The MINPACK-2 test problem collection (preliminary version).** Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, Technical Memorandum No.150, May 1991.

5. AVERICK, B.M., CARTER, R.G., MORÉ, J.J. and XUE, GUO-LIANG, (1992). **The MINPACK-2 test problem collection**. Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, Preprint MCS-P153-0692, June 1992.

6. BEBERNES, J. and EBERLY, D., (1989). **Mathematical problems from combustion theory**. Applied Mathyematical Science, 83, Springer-Verlag, 1989.

7. BIRGIN, E. and MARTINEZ, J.M., (2001). **A spectral conjugate gradient method for unconstrained optimization.** Applied Math. And Optimization, 43, pp. 117-128, 2001.

8. CIMATTI, G. and MENCHI, O., (1978). **On the numerical solution of a variational inequality connected with the hydrodynamic lubrication of a complete journal bearing**. Calcolo, 15, 1978, pp.249-258.

9. DAI, Y.H. and LIAO, (2001). **New conjugacy conditions and related nonlinear conjugate gradient methods**. Appl. Math. Optim., 43, pp. 87-101, 2001.

10. DAI, Y.H. and NI, Q., (2003). **Testing different conjugate gradient methods for large-scale unconstrained optimization**. J. Comput. Math., 21, 2003, pp. 311-320.

11. DAI, Y.H. and YUAN Y., (1998). **A class of globally convergent conjugate gradient methods.** Research report ICM-98-030, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1998.

12. DAI, Y.H. and YUAN, Y., (1999). **A nonlinear conjugate gradient method with a strong global convergence property**. SIAM J. Optim., 10, pp.177-182, 1999.

13. DAI, Y.H. and YUAN, Y., (2001). **A three-parameter family of hybrid conjugate gradient method.Math**. Comp., 70, pp.1155-1167, 2001.

14. DAI, Y.H. and YUAN, Y., (2003). **A class of globally convergent conjugate gradient methods.** Sci. China Ser. A, 46 pp.251-261, 2003.

15. DANIEL, J.W., (1967). **The conjugate gradient method for linear and nonlinear operator equations**. SIAM J. Numer. Anal., 4, pp.10-26, 1967.

16. ELLIOT, C.M. and OCKENDON, J.R., (1982). **Weak and variational methods for moving boundary problems**. Research Notes in Mathematics, vol.59, Pittman, 1982.

17. FLETCHER, R. (1987). **Practical Methods of Optimization vol.1: Unconstrained Optimization**. Jhon Wiley & Sons, New York, 1987.

18. FLETCHER, R. and REEVES, C.M. (1964). **Function minimization by conjugate gradients**. Computer Journal, 7, 1964, pp.149-154.

19. GARNER, J. and BENEDEK, R. (1990). **Solution of Ginzburg-Landau equations for inhomogeneous superconductors by nonlinear optimization**. Phys. Rev., B. 42, 1990, pp.376-385.

20. GILBERT, J.C. and NOCEDAL, J., (1992). **Global convergence properties of conjugate gradient methods for optimization**. SIAM J. Optim., 2, pp.21-42, 1992.

21. GLOWINSKI, R.(1984). **Numerical Methods for Nonlinear Variational Problems**. Springer-Verlag, 1984.

22. GOODMAN, J., KOHN, R. and REYNA, L. (1986). **Numerical study of a relaxed variational problem from optimal design.** Comput. Methods Appl. Mech. Engrg., 57, 1986, pp.107-127.

23. HAGER, W.W., and ZHANG, H. (2004a). **A new conjugate gradient method with guaranteed descent and an efficient line search**. University of Florida, Department of Mathematics, November 17, 2003 (theory and comparisons), revised July 3, 2004.

24. HAGER, W.W., and ZHANG, H. (2004b). **CG-DESCENT, A conjugate gradient method with guaranteed descent (algorithm details and comparisons)**. University of Florida, Department of Mathematics, January 15, 2004.

25. HAGER, W.W. and ZHANG, H. (2005). **A survey of nonlinear conjugate gradient methods**. Department of Mathematics, University of Florida, February 7, 2005.

26. HAN, J.Y., LIU, G.H., SUN D.F., and YIN, H.X. (1994). **On the global convergence of nonlinear conjugate gradient methods**, Technical Report 94-011, Inst. of Applied Math., Chinese Academy of Science, Beijing, 1994.

27. HESTENES, M.R. and STIEFEL, E. (1952). **Methods of conjugate gradients for solving linear systems**. Journal of Research of the National Bureau of Standards, 49, 1952, pp. 409-436.

28. HOARE, M.R. (1979). **Structure and dynamics of simple microclusters**. Advances in Chemical Physics, 40, pp. 49-135, 1979.

29. HU, Y.F. and STOREY, C. (1991). **Global convergence result for conjugate gradient methods**. JOTA, 71, pp. 399-405, 1991.

30. LIN, Y. and CRYER, C.W. (1985). **An alternating direction implicit algorithm for the solution of linear complementarity problems arising from free boundary problems**. Appl. Math. Optim., 13, 1985, pp.1-17.

31. LIU, Y. and STOREY, C. (1991). **Efficient generalized conjugate gradient algorithms**, Part 1: Theory. JOTA, 69, pp. 129-137, 1991.

32. MORÉ, J.J. and TORALDO, G. (1991). **On the solution of large quadratic programming problems with bound constraints**. SIAM J. Optimization, 1, 1991, pp. 93-113.

33. NAZARETH, J.L. (1999). **Conjugate gradient methods**. Enciclopedia of Optimization, C. Floudas and P. Pardalos, (Eds.), Kluwer Academic Publishers, Boston, 1999.

34. O'LEARY, D.P. and YANG, W.H. (1978). **Elastoplastic torsion by quadratic programming**, Comput. Methods Appl. Mech. Engrg., 16, 1978, pp. 361-368.

35. TOUATI-AHMED, D. and STOREY, C. (1990). **Efficient hybrid conjugate gradient techniques**. JOTA, 64, pp.379-397, 1990.

36. PERRY, J.M. (1977). **A class of conjugate gradient algorithms with a two step variable metric memory**. Discussion paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1977.

37. POLAK, E., and RIBIÈRE, G. (1969). **Note sur la convergence de directions conjugées**. Rev. Francaise Informat. Recherche Operationelle, 3e Année 16, pp. 35-43, 1969.

38. POLYAK, B.T. (1969). **The conjugate gradient method in extreme problems**. URSS Comp. Math. Math. Phys., 9, pp. 94-112, 1969.

39. POWELL, M.J.D. (1977). **Restart procedures of the conjugate gradient method**. Math. Programm., 2, pp. 241-254, 1977.

40. POWELL, M.J.D. (1984). **Nonconvex minimization calculations and the conjugate gradient method.** Numerical Analysis (Dundee, 1983), Lecture Notes in Mathematics, vol. 1066, Springer Verlag, 1984, pp. 122-141.

41. RAYDAN, M. (1997). **The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem**. SIAM J. Opt., 7, 26-33, 1997.

42. SHANNO, D.F. (1978). **Conjugate gradient methods with inexact searches**. Mathematics of Operations Research, vol. 3, pp. 244-256, 1978.

43. SHANNO, D.F., and PHUA, K.H**. Algorithm 500, Minimization of unconstrained multivariate functions** [E4], ACM Trans. on Math. Soft., 2, pp .87-94, 1976.

44. WANG, C., HAN, J., and WANG, L. (2000). **Global convergence of the Polak-Ribière and Hestens-Stiefel conjugate gradient methods for the unconstrained nonlinear optimization**, OR Transactions, 4, pp. 1-7, 2000.

45. WOLFE, P. (1969). **Convergence conditions for ascent methods**. SIAM Rev., 11, pp. 226-235, 1969.

46. WOLFE, P. (1971). **Convergence conditions for ascent methods II: some corrections**. SIAM Rev., 13, pp. 185-188, 1971.

47. YABE, H. and TAKANO, M. (2004). **Global convergence properties of nonlinear conjugate gradient methods with modified secant conditions.** Comput. Optim. Appl., (COAP) 28, pp. 203-225, 2004.

48. ZOUTENDIJK, G. (1970). **Nonlinear programming, computational methods**. In Integer and Nonlinear Programming, J. Abadie (ed.), North-Holland, Amsterdam, pp. 37-86, 1970.