Conjugate gradient Algorithms for Molecular Formation under pairwise Potential Minimization

Neculai Andrei
Research Institute for 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 the molecular conformation application, i.e. the so called the Lennard-Jones clusters problem. The conjugate gradient algorithms classifies in four groups: conjugacy condition algorithms, hybrid conjugate gradient, scaled conjugate gradient and preconditioned conjugate gradient algorithms. In this paper we present the main characteristics of these conjugate gradient algorithms emphasizing their convergence properties. We present the performances of 25 conjugate gradient algorithms to solve the protein folding problem. We show that the conjugate gradient algorithms are able to solve this difficult problem, and our s*caled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization* is the top performer among the conjugate gradient algorithms.

Key words. Molecular formation, potential minimization, nonlinear conjugate gradient methods, unconstrained optimization, nonlinear programming.
AMS subject classifications. 90C06, 90C26, 65Y20

1. Introduction

One of the very important problem with a crucial role in structure determination and anlaysis of proteins, peptides and other organic molecules is energy minimization. The interatomic interactions of biomolecules such as proteins and nucleic acids are described by an empirical potential energy function (force field) which depends on the structure i.e. on the geometry of the molecule, and typically leads to an energy surface with a tremendous number of local minima. Identifying the lowest energy minima is the goal of protein folding problem. This problem received a great deal of attention both from its importance in computational structural biology in analysis of organic aggregates and in testing the performances of unconstrained optimization algorithms.

The algorithmic and computational studies given by Navon *et al* [1992a, 1992b], Wang, *et al* [1995] and Zou *et al* [1993] show that truncated Newton (TN, [Nash, 1984]) and limited BFGS (L-BFGS, [Liu and Nocedal, 1989]) are powerful optimization methods that are more efficient than other techniques. Truncated Newton method combine the quadratic convergence rate of the classic Newton method with feasible storage and computational requirements. On the other hand, the L-BFGS algorithm is simple to use mainly because it does not require the knowledge of the sparsity structure of the Hessian or knowledge of the separability of the minimization function. The conclusion is that the TN performs better than L-BFGS for functions that are nearly quadratic, while for highly nonlinear complex functions L-BFGS outperforms TN.

An excellent review on minimization methods has been considered by Schlick [1992] where some details on TNPACK package co-authored with Fogelson, which implements a TN algorithm, are presented [Schlick and Fogelson, 1992]. The main characteristic of

TNPACK is that the user has the possibility to supply a sparse preconditioning matrix that cluster the eigenvalues of the preconditioned Hessian, thus accelerating the convergence of the optimization process. The second implementation of TN algorithm is that given by Nash [1984] which uses an automatic preconditioning. The Nash's implementation TN has the advantage of portability because the preconditioner does not have to be tailored to a specific problem. Some computational studies and comparisons between several algorithms applied to the molecule deoxycytine and to cluster of water molecules are presented by Schlick. It is shown that for molecule deoxycytine the TN with preconditioning is the most efficient requiring about half of the CPU time corresponding to L-BFGS with preconditioning.

In this paper we present another approach to models of protein by using the conjugate gradient algorithms for unconstrained optimization. Conjugate gradient methods represent a class of unconstrained optimization algorithms characterized by low memory requirements and strong local and global convergence properties. The structure of the paper is as follows. In section 2 we present a survey of nonlinear conjugate gradient algorithms. Section 3 is dedicated to present the Lennard-Jones Clusters, i.e. the molecular conformation problem and its solution given by 25 nonlinear conjugate gradient algorithms. It is shown that our s*caled memoryless BFGS preconditioned conjugate gradient algorithm* is the top performer among the conjugate gradient algorithms [Andrei, 2006a]. Some conclusions are given in section 4.

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:

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

where $\alpha_k > 0$ 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 β_k is a scalar. 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 $d_{k+1}^T y_k = 0,$ where $y_k = g_{k+1} - g_k$. This is motivated by the following relations:

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

or by the mean value theorem

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

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

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

$$f(x_k + \alpha_k d_k) - f(x_k) \le \sigma_1 \alpha_k g_k^T d_k, \tag{4a}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \ge \sigma_2 g_k^T d_k, \tag{4b}$$

where $0 < \sigma_1 \le \sigma_2 < 1$. For some conjugate gradient algorithms the stronger versions of the Wolfe line search conditions are needed to ensure the convergence and the stability. The strong Wolfe conditions consists of (4a) and the following strengthened version of (4b)

$$\left|\nabla f(x_k + \alpha_k d_k)^T d_k\right| \le \sigma_2 g_k^T d_k . \tag{4c}$$

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 $\alpha_k$ >0 satisfying the Wolfe line search (4a) and (4b).
Step 3. Compute $x_{k+1} = x_k + \alpha_k d_k$. 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. ♦
```

2.1. Classical Conjugate Gradient Algorithms

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

$$g_k^T d_k = -\|g_k\|^2, \tag{5}$$

which implies the *sufficient descent condition*

$$g_k^T d_k < -c\|g_k\|^2, \tag{6}$$

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 β_k parameter in conjugate gradient algorithms.

$\beta_k^{HS} = \dfrac{g_{k+1}^T y_k}{d_k^T y_k}$	The original linear conjugate gradient algorithm by Hestenes and Stiefel [1952].
$\beta_k^{FR} = \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$	The first nonlinear conjugate gradient algorithm, proposed by Fletcher and Reeves [1964].
$\beta_k^{D} = \dfrac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$	Proposed by Daniel [1967]. This updating formula requires evaluation of Hessian at every iteration.
$\beta_k^{PRP} = \dfrac{g_{k+1}^T y_k}{g_k^T g_k}$	Proposed by Polak and Ribière [1969] and Polyak [1969].
$\beta_k^{PRP+} = max\left\{0, \dfrac{g_{k+1}^T y_k}{g_k^T g_k}\right\}$	Proposed by Powell [1984], and analyzed by Gilbert and Nocedal [1992].
$\beta_k^{CD} = \dfrac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}$	Proposed by Fletcher [1987] as a Conjugate descent method
$\beta_k^{LS} = \dfrac{g_{k+1}^T y_k}{-d_k^T g_k}$	Proposed by Liu and Storey [1991].
$\beta_k^{DY} = \dfrac{g_{k+1}^T g_{k+1}}{d_k^T y_k}$	Proposed by Dai and Yuan [1999].

$\beta_k^{DL} = \dfrac{g_{k+1}^T(y_k - ts_k)}{d_k^T y_k}, t > 0$	Proposed by Dai and Liao [2001].
$\beta_k^{DL+} = max\left\{0, \dfrac{g_{k+1}^T y_k}{d_k^T y_k}\right\} - t\,\dfrac{g_{k+1}^T s_k}{d_k^T y_k}$	Proposed by Dai and Liao [2001]
$\beta_k^{YT} = \dfrac{g_{k+1}^T(z_k - ts_k)}{d_k^T z_k}$	Proposed by Yabe and Takano [2004], where $z_k = y_k + \dfrac{\rho\vartheta_k}{s_k^T u_k} u_k, \ \ \rho \geq 0,$ $\vartheta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$
$\beta_k^{YT+} = max\left\{0, \dfrac{g_{k+1}^T z_k}{d_k^T z_k}\right\} - t\,\dfrac{g_{k+1}^T s_k}{d_k^T z_k}$	Proposed by Yabe and Takano [2004].

2.2. Hybrid and parametrized conjugate gradient algorithms

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 β_k parameter in conjugate gradient algorithms.

$\beta_k^{TAS} = \begin{cases} \beta_k^{PRP}, & 0 \leq \beta_k^{PRP} \leq \beta_k^{FR}, \\ \beta_k^{FR}, & \text{otherwise.} \end{cases}$	Proposed by Touat-Ahmed and Storey [1990].
$\beta_k^{HuSt} = max\left\{0, min\left\{\beta_k^{PRP}, \beta_k^{FR}\right\}\right\}$	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].
$\beta_k^{hDY} = max\left\{-\left(\dfrac{1-\sigma}{1+\sigma}\right)\beta_k^{DY}, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$	Proposed by Dai and Yuan [2001]. σ is the parameter used in the second Wolfe line search condition.
$\beta_k^{hDYz} = max\left\{0, min\left\{\beta_k^{HS}, \beta_k^{DY}\right\}\right\}$	Proposed by Dai and Yuan [2001]. σ 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:

$$\beta_k = \frac{\left\|g_{k+1}\right\|^2}{\lambda_k \left\|g_k\right\|^2 + (1-\lambda_k) d_k^T y_k}, \tag{7}$$

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 β_k^{FR} and β_k^{HS}, Nazareth [1999] suggested a two-parameter family of conjugate gradient methods:

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1-\mu_k) g_{k+1}^T y_k}{\lambda_k \|g_k\|^2 + (1-\lambda_k) d_k^T y_k}, \tag{8}$$

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:

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1-\mu_k) g_{k+1}^T y_k}{(1-\lambda_k - \omega_k)\|g_k\|^2 + \lambda_k d_k^T y_k - \omega_k d_k^T g_k}, \tag{9}$$

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.

2.3. Scaled Conjugate Gradient Algorithms

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:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k d_k, \tag{10}$$

where θ_{k+1} is a positive parameter (the scaling parameter).

Observe that if $\theta_{k+1} = 1,$ then we get the classical conjugate gradient algorithms according to the value of the scalar parameter β_k. On the other hand, if $\beta_k = 0,$ then we get another class of algorithms according to the selection of the parameter θ_{k+1}. There are two possibilities for θ_{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 (10) represents a combination between the quasi-Newton and conjugate gradient methods.

To determine β_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:

$$\beta_k = \frac{s_k^T \nabla^2 f(x_{k+1}) \theta_{k+1} g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k.} \tag{11}$$

Using the Taylor development, after some algebra we obtain:

$$\beta_k = \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{y_k^T s_k}, \tag{12}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

If $\theta_{k+1} = 1$, then (12) is the parameter corresponding to the direction considered by Perry [1977], i.e. we get the *scaled Perry* algorithm. This value of parameter β_k is used by Birgin and Martinez [2001] in their SCG (spectral conjugate gradient) package for unconstrained optimization, where θ_{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 β_k.

Table 3. Scaled choices for the β_k parameter in conjugate gradient algorithms.

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

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

2.4. BFGS preconditioned conjugate gradient algorithm. (SCALCG Algorithm) [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}:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1}\left(\frac{g_{k+1}^T s_k}{y_k^T s_k}\right) y_k - \left[\left(1 + \theta_{k+1}\frac{y_k^T y_k}{y_k^T s_k}\right)\frac{g_{k+1}^T s_k}{y_k^T s_k} - \theta_{k+1}\frac{g_{k+1}^T y_k}{y_k^T s_k}\right] s_k, \quad (13)$$

Thus, we get another class of scaled memoryless BFGS preconditioned conjugate gradient algorithms. The implementation of the direction (13), in the context of Powell restarts, is given by Andrei [2006a, 2006b] in SCALCG package. In SCALCG algorithm θ_{k+1} is defined as a scalar approximation of the inverse Hessian. The following procedures can be used: θ_{k+1} spectral or θ_{k+1} anticipative [Andrei 2004a].

Algorithm SCALCG

Step 1. Select $x_0 \in R^n$, and the parameters $0 < \sigma_1 \le \sigma_2 < 1$. Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$. Set $k = 0$.

Step 2. Line search. Compute α_k satisfying the Wolfe conditions (4a) and (4b). Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 3. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k+1$.

Step 4. Compute θ_k using a spectral $\theta_{k+1} = s_k^T s_k / y_k^T s_k$ or an anticipative $\theta_{k+1} = 1/\gamma_{k+1}$ approach, where $\gamma_{k+1} = \frac{2}{d_k^T d_k} \frac{1}{\alpha_k^2}\left[f(x_{k+1}) - f(x_k) - \alpha_k g_k^T d_k \right]$.

Step 5. Compute the (restart) direction d_k as in (13).

Step 6. Line search. Compute the initial guess of the step length as $\alpha_k = \alpha_{k-1}\left\|d_{k-1}\right\|_2 / \left\|d_k\right\|_2$. Using this initialization compute α_k satisfying the Wolfe conditions (4a) and (4b). Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 7. Store $\theta = \theta_k$, $s = s_k$ and $y = y_k$.

Step 8. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k+1$.

Step 9. If the Powell restart criterion $\left|g_{r+1}^T g_r\right| \geq 0.2\left\|g_{r+1}\right\|^2$, or the angle restart criterion $d_r^T g_{r+1} > -10^{-3}\left\|d_r\right\|_2 \left\|g_{r+1}\right\|_2$, is satisfied, then go to step 4 (a restart step); otherwise continue with step 10 (a standard step).

Step 10. Compute

$$v = \theta g_k - \theta\left(\frac{g_k^T s}{y^T s}\right) y + \left[\left(1 + \theta\frac{y^T y}{y^T s}\right)\frac{g_k^T s}{y^T s} - \theta\frac{g_k^T y}{y^T s}\right] s,$$

$$w = \theta y_k - \theta\left(\frac{y_{k-1}^T s}{y^T s}\right) y + \left[\left(1 + \theta\frac{y^T y}{y^T s}\right)\frac{y_{k-1}^T s}{y^T s} - \theta\frac{y_{k-1}^T y}{y^T s}\right] s,$$

and

$$d_k = -v + \frac{(g_k^T s_{k-1})w + (g_k^T w)s_{k-1}}{y_{k-1}^T s_{k-1}} - \left(1 + \frac{y_{k-1}^T w}{y_{k-1}^T s_{k-1}}\right)\frac{g_k^T s_{k-1}}{y_{k-1}^T s_{k-1}} s_{k-1}. \qquad (14)$$

Step 11. Line search. Compute the initial guess of the step length as $\alpha_k = \alpha_{k-1}\left\|d_{k-1}\right\|_2 / \left\|d_k\right\|_2$. Using this initialization compute α_k satisfying the Wolfe conditions (4a) and (4b). Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 12. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k+1$ and go to step 9. ■

It is well known that if f is bounded below along the direction d_k, then there exists a step length α_k satisfying the Wolfe conditions. The initial selection 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 $\alpha_{k-1}\left\|d_{k-1}\right\|_2 / \left\|d_k\right\|_2$. This selection, was considered for the first time by Shanno and Phua in CONMIN [1976].

2.5. Conjugate Gradient Algorithms with Sufficient Descent Condition

For solving the unconstrained optimization problem (1) where $f : R^n \to R$ is continuously differentiable, recently Andrei [2006c] consider a nonlinear conjugate gradient algorithm:

$$x_{k+1} = x_k + \alpha_k d_k, \qquad (15)$$

where the stepsize α_k is positive and the directions d_k are computed by the rule:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^a s_k, \quad d_0 = -g_0, \qquad (16)$$

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}}, \tag{17}$$

$$\beta_k^{aCGSD} = \frac{1}{y_k^T s_k} \left(g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k \right)^T g_{k+1}, \tag{18}$$

$$\delta_k = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}}. \tag{19}$$

Here $g_k = \nabla f(x_k)$ and $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$. Observe that if f is a quadratic function and α_k is selected to achieve the exact minimum of f in the direction d_k, then $s_k^T g_{k+1} = 0$ and the formula (18) for β_k^{aCGSD} reduces to the Dai and Yuan computational scheme. However, in this paper we consider general nonlinear functions and inexact line search.

This computational scheme has been obtained by modifying the Dai and Yuan scheme $\beta_k^{DY} = g_{k+1}^T g_{k+1} / y_k^T s_k$, in order to satisfy both the sufficient descent and conjugacy conditions in the frame of conjugate gradient methods. To determine the parameters θ_{k+1} and δ_k, from (16) and (18), after some algebra we get:

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

where

$$Q_{k+1} = \theta_{k+1} I - \frac{s_k g_{k+1}^T}{y_k^T s_k} + \delta_k \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2} (s_k s_k^T). \tag{21}$$

Now, by symmetrization of Q_{k+1} as:

$$\overline{Q}_{k+1} = \theta_{k+1} I - \frac{s_k g_{k+1}^T + g_{k+1} s_k^T}{y_k^T s_k} + \delta_k \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2} (s_k s_k^T), \tag{22}$$

we can consider $d_{k+1} = -\overline{Q}_{k+1} g_{k+1}$. Therefore, from the conjugacy condition, $y_k^T d_{k+1} = 0$, i.e.

$$y_k^T \overline{Q}_{k+1} = 0, \tag{23}$$

it follows that:

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}} \quad \text{and} \quad \delta_k = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}} = \frac{1}{\theta_{k+1}}. \tag{24}$$

To conclude the sufficient descent condition from (20), the quantity $\theta_{k+1} - 1/(4\delta_k)$ is required to be nonnegative. Supposing that $\theta_{k+1} - 1/(4\delta_k) > 0$, then the direction given by (16) and (18) is a descent direction. Dai and Yuan [1999, 2003] present conjugate gradient schemes with the property that $g_k^T d_k < 0$ when $y_k^T s_k > 0$. In our algorithm observe that, if for all k, $\theta_{k+1} \geq 1/4\delta_k$, and the line search satisfies the Wolfe conditions, then for all k the search direction (16) and (18) satisfy the sufficient descent condition.

In Andrei [2006d] another algebraic interpretation of the conjugacy condition $y_k^T d_{k+1} = 0$, (where d_{k+1} is given as in (20)) is suggested as:

$$y_k^T \overline{Q}_{k+1} g_{k+1} = 0, \tag{25}$$

where $\overline{Q}_{k+1}$ is given in (22). After some algebra the following algorithm is obtained:

$$d_{k+1} = -g_{k+1} + \beta_k^{ACGSD} s_k, \quad d_0 = -g_0, \tag{26}$$

where

$$\beta_k^{ACGSD} = \frac{1}{y_k^T s_k}\left(y_k - \frac{g_{k+1}^T y_k}{y_k^T s_k} s_k \right)^T g_{k+1}. \tag{27}$$

With these the following variant of conjugate gradient with sufficient descent and conjugacy conditions can be presented:

ACGSD Algorithm

Step 1. Initialization. Select $x_0 \in R^n$ and the parameters $0 < \sigma_1 < \sigma_2 < 1.$ Compute $f(x_0)$ and g_0. Consider $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|.$ Set $k = 0.$

Step 2. Test for continuation of iterations. If $\|g_k\|_\infty \le 10^{-6}$, then stop, else set $k = k+1.$

Step 3. Line search. Compute α_k satisfying the Wolfe line search conditions (4a)-(4b) and update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}),$ g_{k+1} and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 4. Direction computation. Compute $d = -g_{k+1} + \beta_k^{ACGSD} s_k$, where β_k^{ACGSD} is computed as in (27). If $g_{k+1}^T d \le -10^{-3}\|d\|_2 \|g_{k+1}\|_2$, then define $d_{k+1} = d,$ otherwise set $d_{k+1} = -g_{k+1}$. Compute the initial guess $\alpha_k = \alpha_{k-1}\|d_{k-1}\| / \|d_k\|,$ set $k = k+1$ and continue with step 2. ■

Another variant of this conjugate gradient algorithm with sufficient descent and conjugacy conditions is given by:

$$\beta_k^{ACGSDz} = \max\left\{0, \frac{g_{k+1}^T y_k}{y_k^T s_k}\right\}\left(1 - \frac{g_{k+1}^T s_k}{y_k^T s_k}\right). \tag{28}$$

Finally, using the same arguments as above, but this time on the Polak-Ribière-Polyak parameter β_k^{PRP} we get another conjugate gradient algorithm where

$$\beta_k^{CGSD/PRP} = \frac{1}{y_k^T s_k}\left(y_k - \frac{\|y_k\|^2}{\|g_k\|^2} s_k \right)^T g_{k+1}. \tag{29}$$

For all these algorithms some convergence results can be proved (see [Andrei 2006c, d]

2.6. A Conjugate Gradient Algorithm with guaranteed descent and an efficient line search

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:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{30}$$

$$\beta_k = \frac{1}{d_k^T y_k}\left(y_k - 2\frac{y_k^T y_k}{d_k^T y_k} d_k \right)^T g_{k+1}, \tag{31}$$

$d_0 = -g_0$. In their algorithm Hager and Zhang restrict β_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 β_k in the following manner:

$$d_{k+1} = -g_{k+1} + \overline{\beta}_k d_k, \tag{32}$$

$$\overline{\beta}_k = \max\{\beta_k, \eta_k\}, \tag{33}$$

$$\eta_k = \frac{-1}{\|d_k\| min\{\eta, \|g_k\|\}}, \tag{34}$$

where β_k is given by (31), 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 (4a)-(4b) and on a very fine interpretation of the numerical issue concerning the first Wolfe condition (4a).

3. 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 static [Hoare, 1979]. Given the positions $p_1, p_2, \ldots, p_n$ of n atoms (points) in R^3, the energy potential function is defined as

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

where $\|.\|$ denotes the Euclidian norm and $v : R \to R$ is the potential energy function between pairs of atoms. In classical molecular dynamics simulations the interaction between atoms in the sample are described with an interatomic potential $v(r)$, generally assumed to depend only on the distance r between two atoms. Generally, the pairwise potential energy function satisfies the following requirements: it is twice continuously differentiable on R_+, $v(1) = -1$, $(r-1)\dot{v}(r) > 0$ for $r \neq 1$, and

$$\int_1^\infty r^2 |v(r)| \, dr < \infty .$$

The most common choices for the interatomic potential are the Lennard-Jones and Morse potential. The Lennard-Jones is more suitable for closed shell systems such as the noble gases, while the Morse interatomic potential is more appropriate for metals [Lennard-Jones, 1931], [Morse, 1929].

The Lennard-Jones potential function we consider in this study is defined as

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

The Lennard-Jones potential function $v(.)$ is a non-convex function that is bounded below. Indeed, $v(r) \to +\infty$ as $r \to 0$, while $v(r)$ converges to zero as r approaches $+\infty$. The global minimum of v occurs at $r = 1$. The potential function V is invariant with respect to permutations, translations and rotations of the n atoms. Invariance with respect to translations can be eliminated by translating the cluster so that the center of gravity is at the origin.

The molecular conformation problem is to determine a configuration (position for the n points in R^3) such that the energy function V is minimized. Therefore, we determine the configurations that are minimal in the sense that they correspond to a local minimizer of the molecular conformation problem with the least known value of the energy function. Of course, we get only a local minimum. Finding the global minimum of V is a difficult problem because the number of local minima grows exponentially with the number of atoms n.

In the following we present the performances of 25 conjugate gradient algorithms on Lennard-Jones clusters, i.e. molecular conformation application (see 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 all these algorithms. All codes are written in Fortran 77 (default compiler settings) on a workstation Intel 4, 1.8GHz.

SCALCG is authored by Andrei, SCG by Birgin and Martínez [2001] and CG_DESCENT by Hager and Zhang [2004a]. The rest of algorithms are implemented by Andrei. In all these numerical experiments we have considered the standard initial point and the iterations were stopped according to the following criteria:

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

The SCALCG package 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 σ_1 and σ_2.

Table 4 shows the results of the conjugate gradient algorithms considered in this study (#iter is the number of iterations, #fg is the number of function and its gradient evaluations and cpu(s) is the central processor time in seconds).

Table 4. Performance of Conjugate Gradient Algorithms
Lennard-Jones Clusters. Molecular Conformation
$p_i \in R^3$, natoms=1000, $n = 3000$, Powell restart.

Nr.	Algorithm	#iter	#fg	CPU	f*
1.	HS (Hestenes-Stiefel)	1636	1988	169.15	-6603.499965
2.	FR (Fletcher-Reeves)	4001	5254	446.03	-5313.018443
3.	PRP (Polak-Ribiére-Polyak)	3344	4632	392.91	-6629.091442
4.	PRP+ (Polak-Ribiére-Polyak +)	1869	2554	216.70	-6607.394780
5.	CD (Conjugate Descent)	3152	3584	305.22	-6618.251234
6.	LS (Liu-Storey)	2129	2909	246.84	-6588.156816
7.	DY (Dai-Yuan)	4001	4015	343.08	-5245.489809
8.	DL(t=1) (Dai-Liao)	3427	4157	353.67	-6645.128435
9.	DL+(t=1) (Dai-Liao +)	1320	1594	135.61	-6605.076969
10.	aCGSD (Andrei)	1007	1148	97.75	-6628.424341
11.	hDY (hybrid Dai-Yuan)	1797	2175	185.03	-6624.332112
12.	hDYz (hybrid Dai-Yuan zero)	1402	1677	142.70	-6609.111685
13.	GN (Gilbert-Nocedal)	1559	2103	178.50	-6619.710430
14.	HuSt (Hu-Storey)	1345	1804	153.11	-6574.545172
15.	TAS (Touat-Ahmed-Storey)	2202	3007	255.14	-6613.138435
16.	LS-CD (hybrid LS-CD)	2333	3082	261.68	-6614.265345
17.	sP (Scaled Perry) (Birgin-Martínez)	1414	1660	141.34	-6600.920689
18.	sP+ (Scaled Perry+) (Birgin-Martínez+)	675	787	67.02	-6590.794367
19.	sPRP (scaled Polak-Ribiére-Polyak)	2606	3555	301.68	-6629.087628
20.	sFR (scaled Fletcher-Reeves)	4001	5256	446.38	-5312.929509
21.	CGSD/PRP (Andrei)	2208	2672	227.34	-6591.417318
22.	ACGSD (Andrei)	813	915	**77.97**	-6621.069654
23.	ACGSDz (Andrei)	1066	1194	101.76	**-6662.955559**

For different number of atoms in R^3, the SCALCG package has the performances presented in Table 5. Table 6 shows the optimal value of the potential function V for different number of atoms.

Table 5. Performance of SCALCG. θ_{k+1} spectral. Powell restart.

#atoms	n	#iter	#fg	cpu(s)
1000	3000	1252	2002	383.16
2000	6000	1019	1388	752.52
3000	9000	1931	2444	2984.55
4000	12000	3542	4452	9667.37
5000	15000	2642	3415	11590.63

Table 6. Optimal value of the potential function - SCALCG.

#atoms	n	V^*
1000	3000	-6602.762142
2000	6000	-13932.284675
3000	9000	-21135.347220
4000	12000	-28797.188265
5000	15000	-35902.028520

Table 7 shows the performances of CG_DESCENT package by Hager and Zhang [2004a, 2004b] using the Wolfe line search and the approximate Wolfe line search, respectively.

Table 7. Performance of CG_DESCENT. (tol=10^{-5})

	Wolfe line search			Approximate Wolfe line search		
n	#iter	#fg	cpu(s)	#iter	#fg	cpu(s)
3000	1259	2586	631.65	1294	2526	619.58
6000	1721	3506	3393.11	1793	3480	3424.91
9000	1428	2924	6413.15	1494	2901	6496.64

In Table 8 we present the optimal value of the potential function V for different number of atoms, as given by CG_DESCENT.

Table 8. Optimal value of the potential function - CG_DESCENT.

#atoms	n	V^*
1000	3000	-6615.4140837
2000	6000	-14082.394671
3000	9000	-21562.291265

We see that the scaled conjugate gradient algorithm SCALCG is top performer among these conjugate gradient algorithms. However, as we can see the optimal value given by CG_DESCENT is better than that obtained by SCALCG. This is due to the accurate implementation of the Wolfe line search.

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 the molecular conformation problem. 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 including this difficult one which refers to the Lennard-Jones clusters problem. 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. To get a conclusion some more numerical experiments must be considered, especially by comparing the conjugate gradient algorithms (mainly SCALCG and CG_DESCENT) with the limited memory BFGS (LBFGS by Nocedal) and truncated Newton (TN by Nash) algorithms.

References

Andrei, N., (2004a) *Scaled conjugate gradient algorithms for unconstrained optimization.* ICI Technical Report, December 2004

Andrei, N., (2004b) *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* ICI Technical Report, September 2005.

Andrei, N., (2005) SCALCG: *A nonlinear scaled conjugate gradient algorithm for unconstrained optimization.* ICI Technical Report, July 2005.

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

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

Andrei, N., (2006c) *A conjugate gradient algorithm with conjugacy and sufficient descent conditions for unconstrained optimization*. Submitted to ETNA, 2006.

Andrei, N., (2006d) *Another nonlinear conjugate gradient algorithm with conjugacy and sufficient descent conditions for unconstrained optimization*. Submitted SIOPT, 2006.

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.

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.

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

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

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.

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.

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.

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

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

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

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

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

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

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.

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.

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

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.

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.

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

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

Lennard- Jones, J.E., (1931) Proc. Comb. Phil. Soc., 1931, 469.

Liu, D.C. and Nocedal, J., (1989) *On the limited memory BFGS method for large scale optimization.* Mathematical Programming, 45, 1989, pp.503-528.

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

Morse, P.M., (1929) *Diatomic molecules according to the wave mechanics II. Vibrational levels.* Phys. Rev., 34, 1929, 57.

Nash, S.G., (1984) User Guide for TN/TNBC: Fortran routines for nonlinear optimization. Report 397, Mathematical Sciences Department, Johns Hopkins University, Baltimore, 1984.

Navon, I.M., Zou, X., Berger, M., Phua, K.H., Schlick, T., LeDimet, F.X., (1992a) in Optimization Technique and Applications, vol.1, Phua K.H. (Ed.), World Scientific, Singapore, 1992

Navon, I.M., Zou, X., Berger, M., Phua, K.H., Schlick, T., LeDimet, F.X., (1992b) in Optimization Technique and Applications, vol.1, Phua K.H. (Ed.), World Scientific, Singapore, 1992

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

Touati-Ahmed, D., Storey, C., (1990) *Efficient hybrid conjugate gradient techniques.* JOTA, 64, pp.379-397, 1990.

Perry, A., (1977) *A class of conjugate gradient algorithms with a two-step variable metric memory.* Discussion Paper No. 269. Center for Mathematical Studies in Economics and Management Science. Northwestern University, Evanston, Illinois, 1977.

Polak E., G. Ribière, (1969) *Note sur la convergence de méthodes de directions conjuguée,* Revue Francaise Informat. Recherche Opérationnelle, 3e Année 16 (1969), pp.35-43.

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

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

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.

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

Schlick, T., (1992) Rev. Comput. Chem., 1992, 3,1.

Schlick, T., Fogelson, A., (1992) *A Truncated Newton Minimization Package for Large-Scale Problems: I Algorithm and Usage,* ACM TOMS, 1992, vol.18, 46-70

Schlick, T., Fogelson, A., (1992) *A Truncated Newton Minimization Package for Large-Scale Problems: II Implementation Examples.* ACM TOMS, 1992, vol.18, 71-111.

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

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

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.

Wang, Z., Navon, I.M., Zou, X., LeDimet, F.X., (1995) Comput. Opt. Applications, 1995, vol.4

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

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

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.

Zou, X., Navon, I.M., Berger, M., Phua, K.H., Schilck, T., LeDimet, F.X., (1993) SIAM J. Opt., 1993, vol3, 582.

January 6, 2007