

Chapter 5: Conjugate Gradient Methods

Outline

- 1 The Linear Conjugate Gradient Method
- 2 Nonlinear Conjugate Gradient Methods
- 3 References

Outline

1 The Linear Conjugate Gradient Method

2 Nonlinear Conjugate Gradient Methods

3 References

The Linear Conjugate Gradient (CG) Method

- The *linear* CG method was proposed by Hestenes and Stiefel in the 1950s as an iterative method for solving linear systems with positive definite coefficient matrices.
- Given A as a symmetric and positive definite matrix, the linear system of equations

$$Ax = b, \quad (1.1)$$

can be stated equivalently as the following convex minimization problem:

$$\min \phi(x) \equiv \frac{1}{2}x^T Ax - b^T x, \quad (1.2)$$

that is, both (1.1) and (1.2) have the same unique solution.

- For future reference, we note that the gradient of ϕ equals to the residual of the linear systems, that is,

$$\nabla \phi(x) = Ax - b \equiv r(x), \quad (1.3)$$

so in particular at $x = x_k$, we have

$$r_k = Ax_k - b. \quad (1.4)$$

Conjugate Direction Methods

Definition 1

A set of nonzero vectors $\{p_0, p_1, \dots, p_t\}$ is said to be *conjugate* with respect to the symmetric positive definite matrix A if

$$p_i^T A p_j = 0, \quad \text{for all } i \neq j. \quad (1.5)$$

- It is easy to show that any set of vectors satisfying this property is also linearly independent.
- The importance of conjugacy lies in the fact that we can minimize $\phi(\cdot)$ in n steps by successively minimizing it along the individual directions in a conjugate set.
- n -dim problem $\rightarrow n$ 1-dim problems, since for any $x \in \mathbb{R}^n$, $x = \sum_{i=1}^n \alpha^i p_i$, then

$$\phi(x) = \sum_{i=1}^n \frac{(\alpha^i)^2}{2} p_i^T A p_i - \alpha^i p_i^T b \Rightarrow \alpha^i = \frac{p_i^T b}{p_i^T A p_i}$$

Conjugate Direction Methods

Consider the following *conjugate direction* method. Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, \dots, p_{n-1}\}$, let us generate the sequence $\{x_k\}$ by setting

$$x_{k+1} = x_k + \alpha_k p_k, \quad (1.6)$$

where α_k is the one-dimensional minimizer of the quadratic function ϕ along $x_k + \alpha p_k$, given explicitly by

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}. \quad (1.7)$$

We have the following result.

Theorem 2

For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm (1.6), (1.7) converges to the solution x^ of the linear system (1.1) at most n steps.*

Remark: Detailed proof is referred to Theorem 5.1 in [2].

Proof Sketch

Proof Sketch: On the one hand, the linear independence of $p_i, i = 0, \dots, n-1$ indicates that

$$x^* - x_0 = \sigma_0 p_0 + \dots + \sigma_{n-1} p_{n-1}$$

Premultiplying this expression by p_k^T yields $\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}$.

On the other hand, through update (1.6),

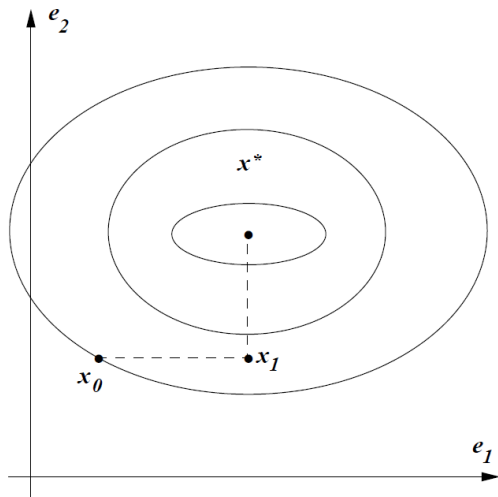
$$x_k = x_0 + \alpha_0 p_0 + \dots + \alpha_{k-1} p_{k-1}.$$

Premultiplying this expression by $p_k^T A$ yields $p_k^T A(x_k - x_0) = 0$. Then

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - Ax_k) = -p_k^T r_k$$

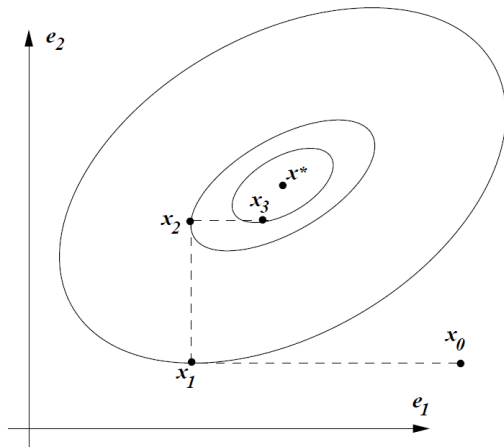
which yields $\alpha_k = \sigma_k$.

Figure 1



Successive minimizations along the coordinate directions find the minimizer of a quadratic with a diagonal Hessian in n iterations.

Figure 2



Successive minimization along coordinate axes does not find the solution in n iterations, for a general convex quadratic.

Conjugate Direction Methods

- We can recover the nice behavior of Figure 1 if we transform the problem to make A diagonal and then minimize along the coordinate directions.
- Suppose we transform the problem by defining new variables \hat{x} as

$$\hat{x} = S^{-1}x,$$

where S is the $n \times n$ matrix defined by

$$S = [p_0, p_1, \dots, p_{n-1}],$$

where $[p_0, p_1, \dots, p_{n-1}]$ is the set of conjugate directions with respect to A .

- The quadratic now becomes

$$\hat{\phi}(\hat{x}) \equiv \phi(S\hat{x}) = \frac{1}{2}\hat{x}^T(S^TAS)\hat{x} - (S^Tb)^T\hat{x},$$

By the conjugacy property, the matrix S^TAS is diagonal, so we can find the minimizing value of $\hat{\phi}$ by performing n one-dimensional minimizations along the coordinate directions of \hat{x} .

Conjugate Direction Methods

- When the Hessian matrix is diagonal, each coordinate minimization correctly determines one of the components of the solution x^* .
- In other words, after k one-dimensional minimizations, the quadratic has been minimized on the subspace spanned by e_1, e_2, \dots, e_k .

The following theorem proves this important result for the general case in which the Hessian of the quadratic is not necessarily diagonal.

Conjugate Direction Methods

Theorem 3 (Expanding Subspace Minimization)

Let $x_0 \in \mathbb{R}^n$ be any starting point and suppose that the sequence $\{x_k\}$ is generated by the conjugate direction algorithm (1.6), (1.7). Then

$$r_k^T p_i = 0, \text{ for } i = 0, 1, \dots, k-1, \quad (1.8)$$

and x_k is the minimizer of $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ over the set

$$\{x | x = x_0 + \text{span}\{p_0, p_1, \dots, p_{k-1}\}\}. \quad (1.9)$$

Remark: Detailed proof is referred to Theorem 5.2 in [2].

Conjugate Direction Methods

The fact that the current residual r_k is orthogonal to all previous search directions is a property that will be used extensively in the following.

The discussion so far has been general, in that it applies to a conjugate direction method based on any choice of the conjugate direction set $\{p_0, p_1, \dots, p_{n-1}\}$. There are many ways to choose the set of conjugate directions. For instance,

- the eigenvectors v_1, v_2, \dots, v_n of A are mutually orthogonal as well as conjugate with respect to A ;
- the Gram–Schmidt orthogonalization process can be modified to produce a set of conjugate directions rather than a set of orthogonal directions.

Basic Property of the Conjugate Gradient Method

The Conjugate Gradient (CG) method is a conjugate direction method with a very special property: In generating its set of conjugate vectors, it can compute a new vector p_k by using only the previous vector p_{k-1} . It does not need to know all the previous elements p_0, p_1, \dots, p_{k-2} of the conjugate set; p_k is automatically conjugate to these vectors.

This remarkable property implies that the method requires little storage and computation.

Details of the CG Method

- Each direction p_k is chosen to be a linear combination of the steepest descent direction $-\nabla\phi(x_k)$ (which is the same as the negative residual r_k) and the previous direction p_{k-1} . We write

$$p_k = -r_k + \beta_k p_{k-1}. \quad (1.10)$$

- The scalar β_k is to be determined by the requirement that p_{k-1} and p_k must be conjugate with respect to A . By premultiplying (1.10) by $p_{k-1}^T A$ and imposing the condition $p_{k-1}^T A p_k = 0$, we find that

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

- Choose the first search direction p_0 to be the steepest descent direction at the initial point x_0 .
- We perform successive one-dimensional minimizations along each of the search directions.

CG Method

Algorithm 1 (CG-Preliminary Version).

Given x_0 ;

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$, **do**

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k}; \quad \text{exact line search}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow Ax_{k+1} - b;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end while

Theorem 4

Suppose that the k -th iterate generated by the CG method is not the solution point x^ . The following four properties hold:*

$$r_k^T r_i = 0, \quad \forall i = 0, \dots, k-1, \quad (1.11a)$$

$$\text{span}\{r_0, r_1, \dots, r_k\} = \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \quad (1.11b)$$

$$\text{span}\{p_0, p_1, \dots, p_k\} = \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \quad (1.11c)$$

$$p_k^T A p_i = 0, \quad \forall i = 0, 1, \dots, k-1. \quad (1.11d)$$

Therefore, the sequence $\{x_k\}$ converges to x^ in at most n steps.*

Remark: Detailed proof is referred to Theorem 5.3 in [2].

CG Method

This theorem shows that

- the directions p_0, p_1, \dots, p_{n-1} are indeed conjugate, which implies termination in n steps;
- the residuals r_i are mutually orthogonal;
- each search direction p_k and residual r_k is contained in the Krylov subspace of degree k for r_0 , defined as

$$\mathcal{K}(r_0; k) \equiv \text{span}\{r_0, Ar_0, \dots, A^k r_0\}.$$

CG Method

- The proof of this theorem relies on the fact that the first direction p_0 is the steepest descent direction $-r_0$; in fact, the result does not hold for other choices of p_0 .
- Since the gradients r_k are mutually orthogonal, the term “conjugate gradient method” is actually a misnomer. It is the search directions, not the gradients, that are conjugate with respect to A .

A Practical Form of the CG Method

We can derive a slightly more economical form of the CG method by using the results of above theorems. Since

$$r_k^T p_k = r_k^T (-r_k + \beta_k p_{k-1}) = -r_k^T r_k$$

(where the conjugacy of p_i indicates $r_k^T p_i = 0$ following from Theorem 3) and

$$r_{k+1} = Ax_{k+1} - b = A(x_k + \alpha_k p_k) - b = r_k + \alpha_k A p_k,$$

we have

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k} \quad (1.12)$$

and

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{p_k^T (r_{k+1} - r_k)} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}. \quad (1.13)$$

A Practical Form of the Conjugate Gradient Method

Algorithm 2 (CG).

Given x_0 ;

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$, **do**

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}; \quad - \frac{r_k^T p_k}{p_k^T A p_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}; \quad \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end while

Rate of Convergence

Theorem 5

If A has only r distinct eigenvalues, the the CG iteration will terminate at the solution in at most r iterations.

Theorem 6

If A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, we have that

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2. \quad (1.14)$$

Remark: Detailed proofs are referred to Theorem 5.4 and 5.5 in [2].

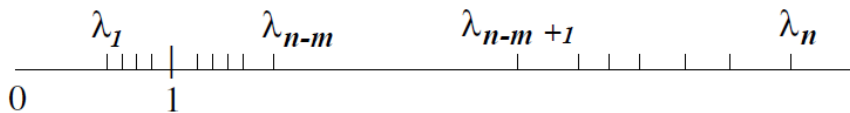
Rate of Convergence

- Above theorem can be used to predict the behavior of the CG method on specific problems. Suppose we have the situation plotted in Figure 3, where the eigenvalues of A consist of m large values, with the remaining $n-m$ smaller eigenvalues clustered around 1.
- If we define $\epsilon = \lambda_{n-m} - \lambda_1$, above theorem tells us that after $m+1$ steps of the conjugate gradient algorithm, we have

$$\|x_{m+1} - x^*\| \approx \epsilon \|x_0 - x^*\|_A.$$

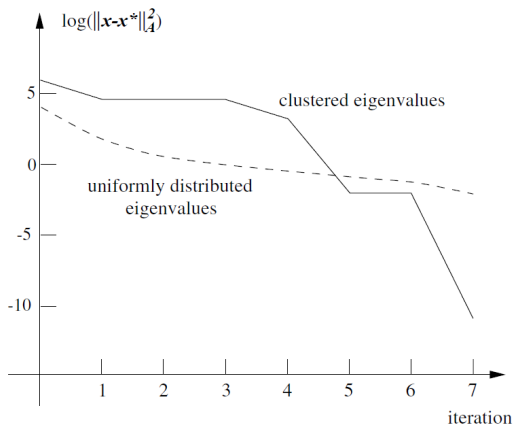
For a small value of ϵ , we conclude that the CG iterates will provide a good estimate of the solution after only $m+1$ steps.

Figure 3



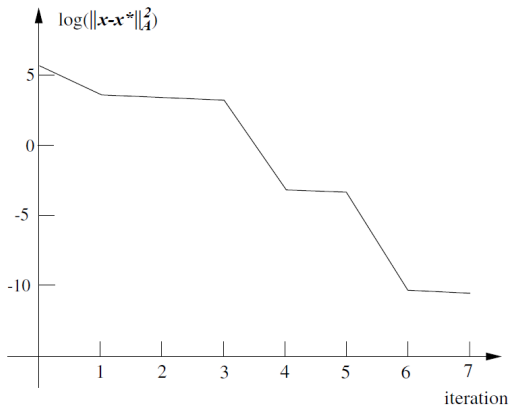
Two clusters of eigenvalues.

Demo



Performance of the conjugate gradient method on (a) a problem in which five of the eigenvalues are large and the remainder are clustered near 1, and (b) a matrix with uniformly distributed eigenvalues.

Demo



Performance of the conjugate gradient method on a matrix of $n = 14$ in which the eigenvalues occur in four distinct clusters.

Convergence Rate

- Another more approximate convergence expression for CG is based on the Euclidean condition number of A , which is defined by

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}.$$

It can be shown that

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A$$

- Comparison with steepest descent method whose convergence rate depends on $\kappa(A)$, but not on its square root.

Preconditioning

We can accelerate the CG method by transforming the linear system to improve the eigenvalue distribution of A . The key to this process, which is known as *preconditioning*, is a change of variables from x to \hat{x} via a nonsingular matrix C , that is,

$$\hat{x} = Cx.$$

The quadratic ϕ is transformed accordingly to

$$\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})^{-1}\hat{x} - (C^{-T}b)^T\hat{x}.$$

If we use CG algorithm to minimize $\hat{\phi}$ or, equivalently, to solve the linear system

$$(C^{-T}AC^{-1})\hat{x} = C^{-T}b,$$

then the convergence rate will depend on the eigenvalues of the matrix $C^{-T}AC^{-1}$ rather than those of A . Therefore, we aim to choose C such that the eigenvalues of $C^{-T}AC^{-1}$ are more favorable for the convergence theory discussed above.

Preconditioning

Algorithm 3 (Preconditioned CG).

Given x_0 , preconditioner M ;

Set $r_0 \leftarrow Ax_0 - b$;

Solve $My_0 = r_0$ for y_0 ;

$p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$, **do**

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k};$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

Solve $My_{k+1} = r_{k+1}$;

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k};$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end while

Preconditioning

- The above algorithm does not make use of C explicitly, but rather the matrix $M = C^T C$, which is symmetric and positive definite by construction.
- If we set $M = I$ in above algorithm, we recover the standard CG method.
- The orthogonality property of the successive residuals becomes

$$r_i^T M^{-1} r_j = 0 \text{ for all } i \neq j.$$

- In terms of computational effort, the main difference between the preconditioned and unpreconditioned CG methods is the need to solve systems of the form $My = r$.
- Incomplete Cholesky is one effective way to generate preconditioner.

Outline

1 The Linear Conjugate Gradient Method

2 Nonlinear Conjugate Gradient Methods

3 References

Nonlinear Conjugate Gradient Methods

- The first *nonlinear* CG method was introduced by Fletcher and Reeves in the 1960s.
- Solve nonlinear minimization problem:

$$\min f(x)$$

- In place of the choice for the step length α_k (which minimizes ϕ along the search direction p_k), we need to perform a line search that identifies an approximate minimum of the nonlinear function f along p_k .
- The residual r , which is simply the gradient of ϕ in linear CG algorithm, must be replaced by the gradient of the nonlinear objective f .

The Fletcher-Reeves Method

Algorithm 4 (FR).

Given x_0 ;

Evaluate $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$;

Set $p_0 \leftarrow -\nabla f_0$, $k \leftarrow 0$;

while $\nabla f_k \neq 0$, **do**

 Compute α_k and set $x_{k+1} = x_k + \alpha_k p_k$;

 Evaluate ∇f_{k+1} ;

$\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$; $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$ in linear CG

$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$;

$k \leftarrow k + 1$;

end while

The Fletcher-Reeves Method

If the line search is exact or any inexact line search procedure that yields an α_k satisfying the following strong wolfe conditions

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \quad (2.1a)$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k|, \quad (2.1b)$$

with $0 < c_1 < c_2 < \frac{1}{2}$ will ensure that all directions p_k are descent directions for the function f , following from

$$\nabla f_k^T p_k = -\|\nabla f_k\|^2 + \beta_k^{FR} \nabla f_k^T p_{k-1}.$$

Goldstein conditions are not suitable for CG method.

Behavior of the Fletcher-Reeves Method

Theorem 7

Suppose that FR algorithm is implemented with a step length α_k that satisfies the strong Wolfe conditions (2.1) with $0 < c_1 < c_2 < \frac{1}{2}$. Then the method generates descent directions p_k that satisfy the following inequalities:

$$-\frac{1}{1 - c_2} \leq \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2} \leq \frac{2c_2 - 1}{1 - c_2} \text{ for all } k = 0, 1, \dots$$

- Above theorem can be proved by induction.
- It can be used to explain a weakness of the Fletcher-Reeves method. We will argue that if the method generates a bad direction and a tiny step, then the next direction and next step are also likely to be poor.

Behavior of the Fletcher-Reeves Method

Let θ denote the angle between p_k and $-\nabla f_k$, defined by

$$\cos \theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\| \|p_k\|}.$$

Suppose that p_k is a poor search direction, in the sense that it makes an angle of nearly $\pi/2$ with $-\nabla f_k$, that is, $\cos \theta_k \approx 0$. By multiplying both sides of the relationship in the above theorem by $\|\nabla f_k\|/\|p_k\|$, we obtain

$$\frac{1 - 2c_2}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|} \leq \cos \theta_k \leq \frac{1}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|}, \text{ for all } k = 0, 1, \dots$$

From these inequalities, we deduce that $\cos \theta_k \approx 0$ if and only if

$$\|\nabla f_k\| \ll \|p_k\|.$$

Behavior of the Fletcher-Reeves Method

Since p_k is almost orthogonal to the gradient, it is likely that the step from x_k to x_{k+1} is tiny, that is, $x_{k+1} \approx x_k$. If so, we have $\nabla f_{k+1} \approx \nabla f_k$, and therefore

$$\beta_{k+1}^{FR} \approx 1,$$

By using this approximation together with $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$, we conclude that

$$p_{k+1} \approx p_k,$$

so the new search direction will improve little (if at all) on the previous one. It follows that if the condition $\cos \theta_k \approx 0$ holds at some iteration k and if the subsequent step is small, a long sequence of unproductive iterates will follow.

The Polak-Ribière Method

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}. \quad (2.2)$$

- PR-CG and HS-CG are identical when f is a strongly convex quadratic function and the line search is exact, since the gradients are mutually orthogonal.
- When applied to general nonlinear functions with inexact line searches, however, the behavior of the two algorithms differs markedly.
- Numerical experience indicates that Algorithm PR-CG tends to be the more robust and efficient of the two.

Behavior of the Polak-Ribière Method

If the search direction p_k satisfies $\cos \theta_k \approx 0$ for some k , and if the subsequent step is small, it follows by substituting $\nabla f_k \approx \nabla f_{k+1}$ into the PR formula, we get $\beta_{k+1}^{PR} \approx 0$. So the new search direction p_{k+1} will be close to the steepest descent direction $-\nabla f_{k+1}$, and $\cos \theta_{k+1}$ will be close to 1. Therefore, Algorithm PR-CG essentially performs a **restart** after it encounters a bad direction.

The Polak-Ribière Method's Variant: PR+

Algorithm PR-CG is that the strong Wolfe conditions (2.1) do not guarantee that p_k is always a descent direction. If we define the β parameter as

$$\beta_{k+1}^+ = \max\{\beta_{k+1}^{PR}, 0\}. \quad (2.3)$$

giving rise to an algorithm we call Algorithm PR+, then a simple adaptation of the strong Wolfe conditions ensures that the descent property holds.

The Hestenes-Stiefel Formula

There are many other choices for β_{k+1} that coincide with the Fletcher-Reeves formula β_{k+1} in the case where the objective is quadratic and the line search is exact. The Hestenes-Stiefel formula, which defines

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k}. \quad (2.4)$$

gives rise to an algorithm that is similar to Algorithm PR-CG, both in terms of its theoretical convergence properties and in its practical performance.

Dai-Yuan Formula

Another choice of β_{k+1} that possess attractive theoretical and computational properties is the Dai-Yuan formula:

$$\beta_{k+1}^{DY} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{(\nabla f_{k+1} - \nabla f_k)^T p_k}. \quad (2.5)$$

Quadratic Termination and Restarts

- A modification that is often used in nonlinear CG procedures is to restart the iteration at every n steps by setting $\beta_k = 0$, that is, by taking a steepest descent step. Restarting serves to periodically refresh the algorithm, erasing old information that may not be beneficial.
- We can even prove a strong theoretical result about restarting: It leads to n -step quadratic convergence, that is,

$$\|x_{k+n} - x^*\| = O(\|x_k - x^*\|^2). \quad (2.6)$$

- Though above result is interesting from a theoretical viewpoint, it may not be relevant in a practical context, because nonlinear conjugate gradient methods can be recommended only for solving problems with large n . In such problems restarts may never occur, since an approximate solution is often located in fewer than n steps.

Quadratic Termination and Restarts

Since the gradients are mutually orthogonal when f is a quadratic function. A restart is performed whenever two consecutive gradients are far from orthogonal, as measured by the test

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geq \nu \quad (2.7)$$

where a typical value for the parameter ν is 0.1.

Global Convergence

AS.1 The level set $\mathcal{L} := \{x | f(x) \leq f(x_0)\}$ is bounded;

AS.2 In some open neighborhood \mathcal{N} of \mathcal{L} , the objective function f is Lipschitz continuously differentiable.

Theorem 8 (Al-Baali)

Suppose that Assumptions AS.1 and AS.2 hold, and that Algorithm 4(FR) is implemented with a line search that satisfies the strong Wolfe conditions (2.1) with $0 < c_1 < c_2 < \frac{1}{2}$. Then

$$\liminf_{k \rightarrow \infty} \|\nabla f_k\| = 0. \quad (2.8)$$

Remark

For Dai-Yuan method, it is not required that $c_2 < \frac{1}{2}$.

Global Convergence

Although the PR method performs better in practice than the FR method. The following surprising result shows that the PR method can cycle infinitely without approaching a solution point, even if an ideal line search is used.

Theorem 9

Consider the Polak-Ribière method (2.2) with an ideal line search. There exists a twice continuously differential objective function $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ and starting point $x_0 \in \mathbb{R}^3$ such that the sequence of gradients $\{\nabla f_k\}$ is bounded away from zero.

Numerical Performance

The following table illustrates the performance of Algorithms FR, PR and PR₊ without restarts. The parameters in the strong Wolfe conditions (2.1) were chosen to be $c_1 = 10^{-4}$ and $c_2 = 0.1$. The iterations were terminated when

$$\|\nabla f_k\|_\infty < 10^{-5}(1 + |f_k|),$$

or after 10,000 iterations (the latter is denoted by a *).

Numerical Performance

Problem	n	Alg FR it/f-g	Alg PR it/f-g	Alg PR+ it/f-g	mod
CALCVAR3	200	2808/5617	2631/5263	2631/5263	0
GENROS	500	*	1068/2151	1067/2149	1
XPOWSING	1000	533/1102	212/473	97/229	3
TRIDIA1	1000	264/531	262/527	262/527	0
MSQRT1	1000	422/849	113/231	113/231	0
XPOWELL	1000	568/1175	212/473	97/229	3
TRIGON	1000	231/467	40/92	40/92	0

Iterations and function/gradient evaluations required by three nonlinear conjugate gradient methods on a set of test problems.

Truncated CG Method for Trust Region Subproblem

- Trust region subproblem: Given B symmetric,

$$\begin{aligned} \min \quad & m(p) = f + g^T p + \frac{1}{2} p^T B p \\ \text{s.t.} \quad & \|p\| \leq \Delta. \end{aligned}$$

Denote its solution as p^*

- Truncated CG method: Use CG method to minimize m with starting point 0. Terminate if the boundary or negative curvature direction is encountered.
- Denote s^* as the solution obtained by truncated CG method. If B is symmetric and positive definite, then

$$\frac{m(0) - m(s^*)}{m(0) - m(p^*)} \geq \frac{1}{2}. \quad (2.9)$$





Detailed proof is referred to [3].

Outline

- 1 The Linear Conjugate Gradient Method
- 2 Nonlinear Conjugate Gradient Methods
- 3 References

References

For recent surveys on CG methods see

-  N.I.M. Gould, D. Orban and P.L. Toint, Numerical methods for large-scale nonlinear optimization, Acta Numerica, 14 (2005), pp. 299-361.
-  Jorge Nocedal and Stephen J. Wright, Numerical Optimization, second edition, Springer, 2006.
-  Y. Yuan Y. On the truncated conjugate gradient method. Math. Prog., 87(2000), pp. 561 571.
-  H. Zhang and W.W. Hager, A survey of the nonlinear conjugate gradient methods, Pacific Journal of Optimization, 2(2006), pp. 35-58.

Thanks for your attention!