最佳化方法與應用
MA5037-*

Chapter 5. Conjugate Gradient Methods

§5.1 The Linear Conjugate Gradient Method

§5.2 Nonlinear Conjugate Gradient Methods

Introduction

Our interest in conjugate gradient methods is twofold. First, they are among the most useful techniques for solving large linear systems of equations. Second, they can be adapted to solve nonlinear optimization problems. The remarkable properties of both linear and nonlinear conjugate gradient methods will be described in this chapter.

Introduction

The linear conjugate gradient method was proposed by Hestenes and Stiefel in the 1950s as an **iterative method** for solving linear systems with positive definite coefficient matrices. It is an alternative to Gaussian elimination that is well suited for solving large problems. The performance of the linear conjugate gradient method is determined by the distribution of the eigenvalues of the coefficient matrix. By transforming, or preconditioning, the linear system, we can make this distribution more favorable and improve the convergence of the method significantly. Preconditioning plays a crucial role in the design of practical conjugate gradient strategies. Our treatment of the linear conjugate gradient method will highlight those properties of the method that are important in optimization.

Introduction

The first nonlinear conjugate gradient method was introduced by Fletcher and Reeves in the 1960s. It is one of the earliest known techniques for solving large-scale nonlinear optimization problems. Over the years, many variants of this original scheme have been proposed, and some are widely used in practice. The key features of these algorithms are that they require no matrix storage and are faster than the steepest descent method.

§5.1 The Linear Conjugate Gradient Method

The conjugate gradient method is an iterative method for solving a linear system of equations

$$Ax = b, \tag{1}$$

where A is an $n \times n$ symmetric positive definite matrix. The problem (1) can be stated equivalently as the following minimization problem:

$$\min \varphi(x) = \frac{1}{2} x^{\mathrm{T}} A x - b^{\mathrm{T}} x; \tag{2}$$

that is, both (1) and (2) have the same unique solution. This equivalence will allow us to interpret the conjugate gradient method either as an algorithm for solving linear systems or as a technique for minimizing convex quadratic functions.

§5.1 The Linear Conjugate Gradient Method

For future reference, we note that the gradient of φ equals the **residual** of the linear system; that is,

$$(\nabla\varphi)(x) = Ax - b = r(x)\,, \tag{3}$$

so in particular at $x = x_k$ we have

$$r_k = Ax_k - b\,. \tag{4}$$

§5.1 The Linear Conjugate Gradient Method

- **Conjugate Direction Methods**

One of the remarkable properties of the conjugate gradient method is its ability to generate, in a very economical fashion, a set of vectors with a property known as conjugacy. A set of nonzero vectors $\{p_0, p_1, \cdots, p_\ell\}$ is said to be conjugate with respect to the symmetric **positive definite** matrix A if

$$p_i^{\mathrm{T}} A p_j = 0 \quad \text{for all } i \neq j. \tag{5}$$

It is easy to show that any set of vectors satisfying this property is also linearly independent (for a geometrical illustration of conjugate directions see Section 9.4).

§5.1 The Linear Conjugate Gradient Method

The importance of conjugacy lies in the fact that we can minimize $\varphi(\cdot)$ in n steps by successively minimizing it along the individual directions in a conjugate set. To verify this claim, we consider the following **conjugate direction method** (the distinction between the conjugate gradient method and the conjugate direction method will become clear as we proceed). Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, \cdots, p_{n-1}\}$, let us generate the sequence $\{x_k\}$ by setting

$$x_{k+1} = x_k + \alpha_k p_k \,, \tag{6}$$

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

$$\alpha_k = -\frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k} \,. \tag{7}$$

§5.1 The Linear Conjugate Gradient Method

The importance of conjugacy lies in the fact that we can minimize $\varphi(\cdot)$ in n steps by successively minimizing it along the individual directions in a conjugate set. To verify this claim, we consider the following **conjugate direction method** (the distinction between the conjugate gradient method and the conjugate direction method will become clear as we proceed). Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, \cdots, p_{n-1}\}$, let us generate the sequence $\{x_k\}$ by setting

$$x_{k+1} = x_k + \alpha_k p_k \,, \tag{6}$$

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

$$\alpha_k = -\frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k} \,. \tag{7}$$

§5.1 The Linear Conjugate Gradient Method

Theorem

For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm (6), (7) converges to the solution x_* of the linear system (1) in at most n steps.

Proof.

Since the directions $\{p_i\}$ are linearly independent, they must span the whole space $\mathbb{R}^n$. Hence, we can write the difference between x_0 and the solution x_* in the following way:

$$x_* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \cdots + \sigma_{n-1} p_{n-1}$$

for some choice of scalars σ_k. By premultiplying this expression by $p_k^{\mathrm{T}} A$ and using the conjugacy property (5), we obtain

$$\sigma_k = \frac{p_k^{\mathrm{T}} A (x_* - x_0)}{p_k^{\mathrm{T}} A p_k}. \tag{8}$$

□

§5.1 The Linear Conjugate Gradient Method

Theorem

For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm (6), (7) *converges to the solution x_* of the linear system* (1) *in at most n steps.*

Proof.

Since the directions $\{p_i\}$ are linearly independent, they must span the whole space $\mathbb{R}^n$. Hence, we can write the difference between x_0 and the solution x_* in the following way:

$$x_* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \cdots + \sigma_{n-1} p_{n-1}$$

for some choice of scalars σ_k. By premultiplying this expression by $p_k^{\mathrm{T}} A$ and using the conjugacy property (5), we obtain

$$\sigma_k = \frac{p_k^{\mathrm{T}} A (x_* - x_0)}{p_k^{\mathrm{T}} A p_k}. \tag{8}$$

□

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

We now establish the result by showing that these coefficients σ_k coincide with the step lengths α_k generated by the formula (7). If x_k is generated by algorithm (6), (7), then we have

$$x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{k-1} p_{k-1} .$$

By premultiplying this expression by $p_k^{\mathrm{T}} A$ and using the conjugacy property, we have that

$$p_k^{\mathrm{T}} A(x_k - x_0) = 0 ,$$

and therefore

$$p_k^{\mathrm{T}} A(x_* - x_0) = p_k^{\mathrm{T}} A(x_* - x_k) = p_k^{\mathrm{T}}(b - A x_k) = -p_k^{\mathrm{T}} r_k .$$

By comparing this relationwith (7) and (8), we find that $\sigma_k = \alpha_k$, giving the result. □

§5.1 The Linear Conjugate Gradient Method

There is a simple interpretation of the properties of conjugate directions. If the matrix A in (2) is diagonal, the contours of the function $\varphi(\cdot)$ are ellipses whose axes are aligned with the coordinate directions, as illustrated in Figure 1 (in the next slide). We can find the minimizer of this function by performing one-dimensional minimizations along the coordinate directions e_1, e_2, $\cdots$, e_n in turn.

When A is not diagonal, its contours are still elliptical, but they are usually no longer aligned with the coordinate directions. The strategy of successive minimization along these directions in turn no longer leads to the solution in n iterations (or even in a finite number of iterations). This phenomenon is illustrated in the two-dimensional example of Figure 2.

§5.1 The Linear Conjugate Gradient Method

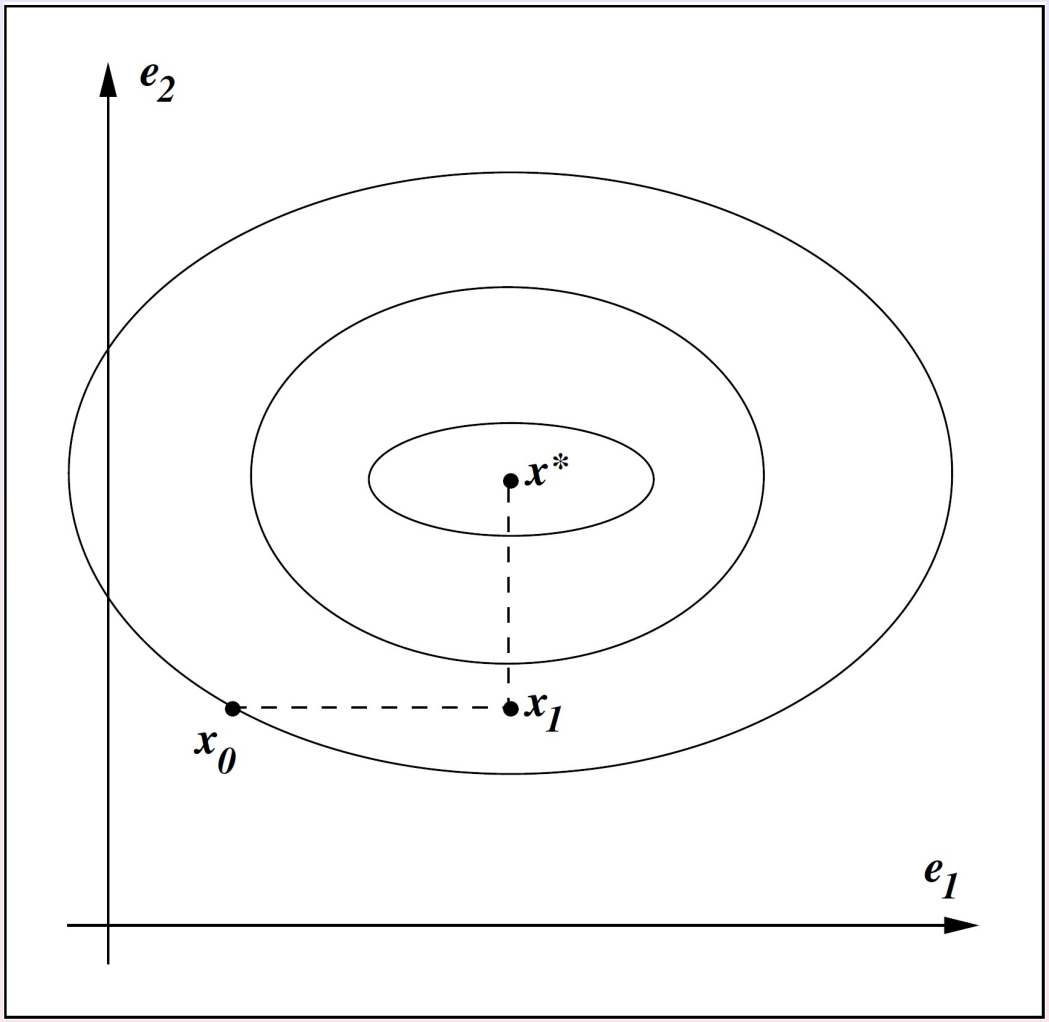

Figure 1: Successive minimizations along the coordinate directions find the minimizer of a quadratic with a diagonal Hessian in n iterations.

§5.1 The Linear Conjugate Gradient Method

There is a simple interpretation of the properties of conjugate directions. If the matrix A in (2) is diagonal, the contours of the function $\varphi(\cdot)$ are ellipses whose axes are aligned with the coordinate directions, as illustrated in Figure 1. We can find the minimizer of this function by performing one-dimensional minimizations along the coordinate directions e_1, e_2, $\cdots$, e_n in turn.

When A is not diagonal, its contours are still elliptical, but they are usually no longer aligned with the coordinate directions. The strategy of successive minimization along these directions in turn no longer leads to the solution in n iterations (or even in a finite number of iterations). This phenomenon is illustrated in the two-dimensional example of Figure 2.

§5.1 The Linear Conjugate Gradient Method

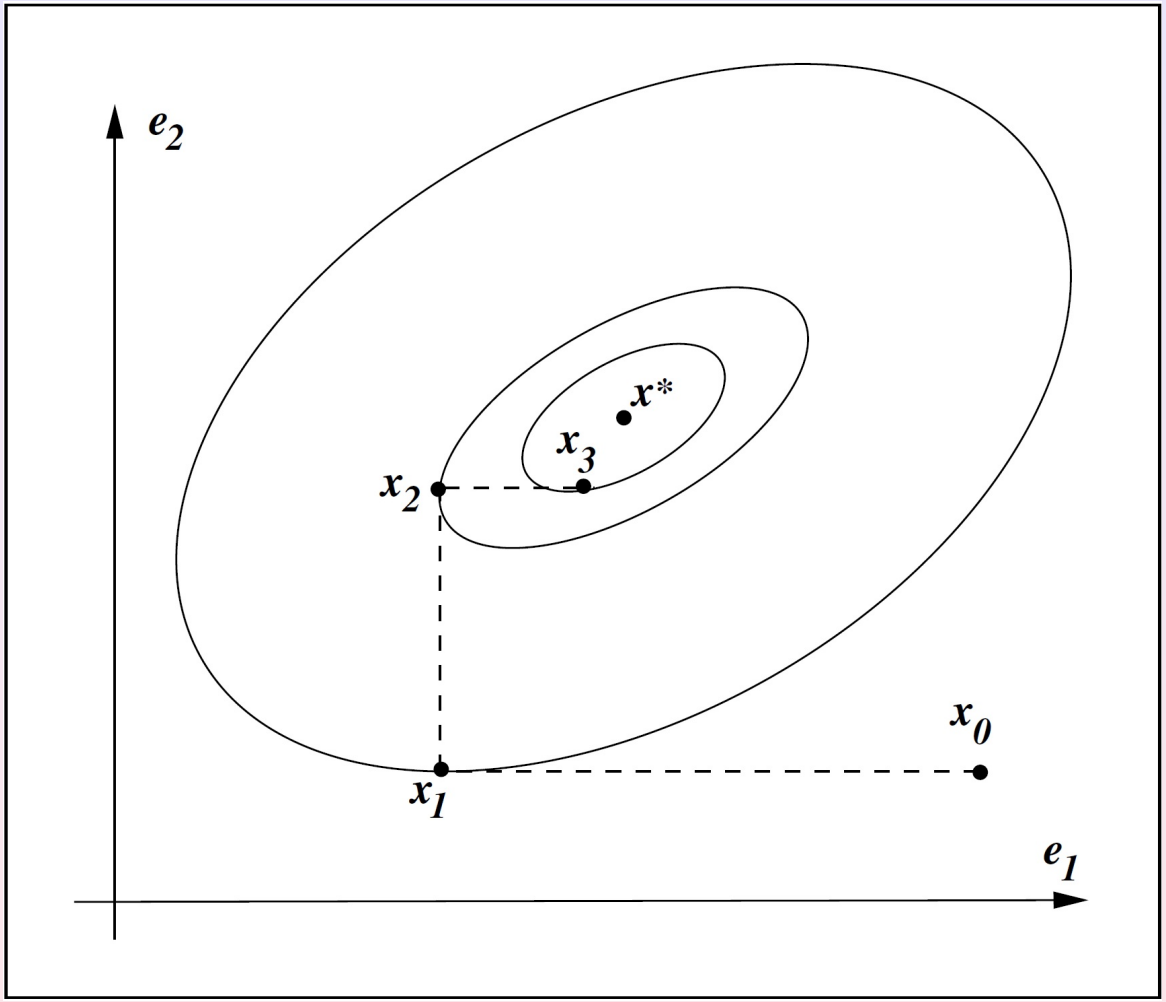

Figure 2: Successive minimization along coordinate axes does not find the solution in n iterations, for a general convex quadratic.

§5.1 The Linear Conjugate Gradient Method

We can, however, 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

$$\hat{x} = S^{-1}x, \tag{9}$$

where, with $\{p_0, p_1, \cdots, p_{n-1}\}$ denoting a set of conjugate directions with respect to A, S is the $n \times n$ matrix given by $S = \left[p_0 \vdots p_1 \vdots \cdots \vdots p_{n-1}\right]$.

The quadratic φ defined by (2) now becomes

$$\hat{\varphi}(\hat{x}) = \varphi(S\hat{x}) = \frac{1}{2}\hat{x}^{\mathrm{T}}(S^{\mathrm{T}}AS)\hat{x} - (S^{\mathrm{T}}b)^{\mathrm{T}}\hat{x}.$$

By the conjugacy property (5), the matrix $S^{\mathrm{T}}AS$ is diagonal, so we can find the minimizing value of $\hat{\varphi}$ by performing n one-dimensional minimizations along the coordinate directions of $\hat{x}$.

§5.1 The Linear Conjugate Gradient Method

We can, however, 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

$$\hat{x} = S^{-1}x, \tag{9}$$

where, with $\{p_0, p_1, \cdots, p_{n-1}\}$ denoting a set of conjugate directions with respect to A, S is the $n \times n$ matrix given by $S = \left[p_0 \vdots p_1 \vdots \cdots \vdots p_{n-1}\right]$. The quadratic φ defined by (2) now becomes

$$\hat{\varphi}(\hat{x}) = \varphi(S\hat{x}) = \frac{1}{2}\hat{x}^{\mathrm{T}}(S^{\mathrm{T}}AS)\hat{x} - (S^{\mathrm{T}}b)^{\mathrm{T}}\hat{x}.$$

By the conjugacy property (5), the matrix $S^{\mathrm{T}}AS$ is diagonal, so we can find the minimizing value of $\hat{\varphi}$ by performing n one-dimensional minimizations along the coordinate directions of $\hat{x}$.

§5.1 The Linear Conjugate Gradient Method

Because of the relation (9), the i-th coordinate direction in $\hat{x}$-space corresponds to the direction p_i in x-space. Hence, the coordinate search strategy applied to $\hat{\varphi}$ is equivalent to the conjugate direction algorithm (6), (7). We then conclude that the conjugate direction algorithm terminates in at most n steps.

Returning to Figure 1, we note another interesting property: 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 $e_1, e_2, \cdots, e_k$. The following theorem proves this important result for the general case in which the Hessian of the quadratic is not necessarily diagonal.

§5.1 The Linear Conjugate Gradient Method

Because of the relation (9), the i-th coordinate direction in $\hat{x}$-space corresponds to the direction p_i in x-space. Hence, the coordinate search strategy applied to $\hat{\varphi}$ is equivalent to the conjugate direction algorithm (6), (7). We then conclude that the conjugate direction algorithm terminates in at most n steps.

Returning to Figure 1, we note another interesting property: 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 $e_1, e_2, \cdots, e_k$. The following theorem proves this important result for the general case in which the Hessian of the quadratic is not necessarily diagonal.

§5.1 The Linear Conjugate Gradient Method

From now on, we use the notation $\text{span}\{p_0, p_1, \cdots, p_k\}$ to denote the set of all linear combinations of the vectors p_0, p_1, $\cdots$, p_k. In proving the result we will make use of the following expression, which is easily verified from the relations (4) and (6):

$$r_{k+1} = r_k + \alpha_k A p_k \,. \tag{10}$$

Theorem

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 (6), (7). Then

$$r_k^{\mathrm{T}} p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1 \,, \tag{11}$$

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

$$\left\{ x \,\middle|\, x = x_0 + \text{span}\{p_0, p_1, \cdots, p_{k-1}\} \right\} . \tag{12}$$

§5.1 The Linear Conjugate Gradient Method

From now on, we use the notation $\text{span}\{p_0, p_1, \cdots, p_k\}$ to denote the set of all linear combinations of the vectors p_0, p_1, $\cdots$, p_k. In proving the result we will make use of the following expression, which is easily verified from the relations (4) and (6):

$$r_{k+1} = r_k + \alpha_k A p_k \,. \tag{10}$$

Theorem

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 (6), (7). Then

$$r_k^{\mathrm{T}} p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1 \,, \tag{11}$$

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

$$\left\{ x \,\middle|\, x = x_0 + \text{span}\{p_0, p_1, \cdots, p_{k-1}\} \right\} . \tag{12}$$

§5.1 The Linear Conjugate Gradient Method

Proof.

We begin by showing that a point $\widehat{x}$ minimizes φ over the set (12) if and only if $r(\widehat{x})^{\mathrm{T}} p_i = 0$, for each $i = 0, 1, \cdots, k-1$. Let us define $h(\sigma) = \varphi(x_0 + \sigma_0 p_0 + \cdots + \sigma_{k-1} p_{k-1})$, where $\sigma = (\sigma_0, \sigma_1, \cdots, \sigma_{k-1})^{\mathrm{T}}$. Since $h(\cdot)$ is a strictly convex quadratic, it has a unique minimizer σ^* that satisfies

$$\frac{\partial h}{\partial \sigma_i}(\sigma^*) = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1 .$$

By the chain rule, this equation implies that

$$(\nabla \varphi)(x_0 + \sigma_0^* p_0 + \cdots + \sigma_{k-1}^* p_{k-1})^{\mathrm{T}} p_i = 0 \quad \text{for } i = 0, 1, \cdots, k-1 .$$

By recalling the definition (3), we have for the minimizer $\widehat{x} = x_0 + \sigma_0^* p_0 + \sigma_1^* p_1 + \cdots + \sigma_{k-1}^* p_{k-1}$ on the set (12) that $r(\widehat{x})^{\mathrm{T}} p_i = 0$, as claimed. □

§5.1 The Linear Conjugate Gradient Method

Proof.

We begin by showing that a point $\widehat{x}$ minimizes φ over the set (12) if and only if $r(\widehat{x})^{\mathrm{T}} p_i = 0$, for each $i = 0, 1, \cdots, k-1$. Let us define $h(\sigma) = \varphi(x_0 + \sigma_0 p_0 + \cdots + \sigma_{k-1} p_{k-1})$, where $\sigma = (\sigma_0, \sigma_1, \cdots, \sigma_{k-1})^{\mathrm{T}}$. Since $h(\cdot)$ is a strictly convex quadratic, it has a unique minimizer σ^* that satisfies

$$\frac{\partial h}{\partial \sigma_i}(\sigma^*) = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1 .$$

By the chain rule, this equation implies that

$$(\nabla \varphi)(x_0 + \sigma_0^* p_0 + \cdots + \sigma_{k-1}^* p_{k-1})^{\mathrm{T}} p_i = 0 \quad \text{for } i = 0, 1, \cdots, k-1 .$$

By recalling the definition (3), we have for the minimizer $\widehat{x} = x_0 + \sigma_0^* p_0 + \sigma_1^* p_1 + \cdots + \sigma_{k-1}^* p_{k-1}$ on the set (12) that $r(\widehat{x})^{\mathrm{T}} p_i = 0$, as claimed. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

We now use induction to show that x_k satisfies (11). For the case $k = 1$, we have from the fact that $x_1 = x_0 + \alpha_0 p_0$ minimizes φ along p_0 that $r_1^{\mathrm{T}} p_0 = 0$. Let us now make the induction hypothesis that $r_{k-1}^{\mathrm{T}} p_i = 0$ for $i = 0, 1, \cdots, k-2$. By (10), we have

$$r_k = r_{k-1} + \alpha_{k-1} A p_{k-1} ,$$

so that

$$p_{k-1}^{\mathrm{T}} r_k = p_{k-1}^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_{k-1}^{\mathrm{T}} A p_{k-1} = 0 ,$$

by the definition of α_{k-1}. For $i = 0, 1, \cdots, k-2$, we have

$$p_i^{\mathrm{T}} r_k = p_i^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_i^{\mathrm{T}} A p_{k-1} = 0 ,$$

where $p_i^{\mathrm{T}} r_{k-1} = 0$ by the induction hypothesis and $p_i^{\mathrm{T}} A p_{k-1} = 0$ by the conjugacy of p_i's. We have shown that $r_k^{\mathrm{T}} p_i = 0$, for $i = 0, 1, \cdots, k-1$, so the proof is complete. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

We now use induction to show that x_k satisfies (11). For the case $k = 1$, we have from the fact that $x_1 = x_0 + \alpha_0 p_0$ minimizes φ along p_0 that $r_1^{\mathrm{T}} p_0 = 0$. Let us now make the induction hypothesis that $r_{k-1}^{\mathrm{T}} p_i = 0$ for $i = 0, 1, \cdots, k-2$. By (10), we have

$$r_k = r_{k-1} + \alpha_{k-1} A p_{k-1} ,$$

so that

$$p_{k-1}^{\mathrm{T}} r_k = p_{k-1}^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_{k-1}^{\mathrm{T}} A p_{k-1} = 0 ,$$

by the definition of α_{k-1}. For $i = 0, 1, \cdots, k-2$, we have

$$p_i^{\mathrm{T}} r_k = p_i^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_i^{\mathrm{T}} A p_{k-1} = 0 ,$$

where $p_i^{\mathrm{T}} r_{k-1} = 0$ by the induction hypothesis and $p_i^{\mathrm{T}} A p_{k-1} = 0$ by the conjugacy of p_i's. We have shown that $r_k^{\mathrm{T}} p_i = 0$, for $i = 0, 1, \cdots, k-1$, so the proof is complete. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

We now use induction to show that x_k satisfies (11). For the case $k = 1$, we have from the fact that $x_1 = x_0 + \alpha_0 p_0$ minimizes φ along p_0 that $r_1^{\mathrm{T}} p_0 = 0$. Let us now make the induction hypothesis that $r_{k-1}^{\mathrm{T}} p_i = 0$ for $i = 0, 1, \cdots, k-2$. By (10), we have

$$r_k = r_{k-1} + \alpha_{k-1} A p_{k-1} ,$$

so that

$$p_{k-1}^{\mathrm{T}} r_k = p_{k-1}^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_{k-1}^{\mathrm{T}} A p_{k-1} = 0 ,$$

by the definition of α_{k-1}. For $i = 0, 1, \cdots, k-2$, we have

$$p_i^{\mathrm{T}} r_k = p_i^{\mathrm{T}} r_{k-1} + \alpha_{k-1} p_i^{\mathrm{T}} A p_{k-1} = 0 ,$$

where $p_i^{\mathrm{T}} r_{k-1} = 0$ by the induction hypothesis and $p_i^{\mathrm{T}} A p_{k-1} = 0$ by the conjugacy of p_i's. We have shown that $r_k^{\mathrm{T}} p_i = 0$, for $i = 0, 1, \cdots, k-1$, so the proof is complete. □

§5.1 The Linear Conjugate Gradient Method

The fact that the current residual r_k is orthogonal to all previous search directions, as expressed in (11), is a property that will be used extensively in this chapter.

The discussion so far has been general, in that it applies to a conjugate direction method (6), (7) based on any choice of the conjugate direction set $\{p_0, p_1, \cdots, p_{n-1}\}$. There are many ways to choose the set of conjugate directions. For instance, the eigenvectors v_1, v_2, $\cdots$, v_n of A are mutually orthogonal as well as conjugate with respect to A, so these could be used as the vectors $\{p_0, p_1, \cdots, p_{n-1}\}$. For large-scale applications, however, computation of the complete set of eigenvectors requires an excessive amount of computation.

§5.1 The Linear Conjugate Gradient Method

The fact that the current residual r_k is orthogonal to all previous search directions, as expressed in (11), is a property that will be used extensively in this chapter.

The discussion so far has been general, in that it applies to a conjugate direction method (6), (7) based on any choice of the conjugate direction set $\{p_0, p_1, \cdots, p_{n-1}\}$. There are many ways to choose the set of conjugate directions. For instance, the eigenvectors v_1, v_2, $\cdots$, v_n of A are mutually orthogonal as well as conjugate with respect to A, so these could be used as the vectors $\{p_0, p_1, \cdots, p_{n-1}\}$. For large-scale applications, however, computation of the complete set of eigenvectors requires an excessive amount of computation.

§5.1 The Linear Conjugate Gradient Method

An alternative approach is to modify the Gram-Schmidt orthogonalization process to produce a set of conjugate directions rather than a set of orthogonal directions (this modification is easy to produce, since the properties of conjugacy and orthogonality are closely related in spirit). However, the Gram-Schmidt approach is also expensive, since it requires us to store the entire direction set.

§5.1 The Linear Conjugate Gradient Method

- **Basic Properties of the Conjugate Gradient Method**

The conjugate gradient 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, $\cdots$, 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.

In the conjugate gradient method, each direction p_k is chosen to be a linear combination of the negative residual $-r_k$ (which, by (3), is the steepest descent direction for the function φ at x_k) and the previous direction p_{k-1}.

§5.1 The Linear Conjugate Gradient Method

- **Basic Properties of the Conjugate Gradient Method**

The conjugate gradient 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, $\cdots$, 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.

In the conjugate gradient method, each direction p_k is chosen to be a linear combination of the negative residual $-r_k$ (which, by (3), is the steepest descent direction for the function φ at x_k) and the previous direction p_{k-1}.

§5.1 The Linear Conjugate Gradient Method

We write

$$p_k = -r_k + \beta_k p_{k-1}\,, \tag{13}$$

where 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 (13) by $p_{k-1}^{\mathrm{T}} A$ and imposing the condition $p_{k-1}^{\mathrm{T}} A p_k = 0$, we find that

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

We choose the first search direction p_0 to be the steepest descent direction at the initial point x_0. As in the general conjugate direction method, we perform successive one-dimensional minimizations along each of the search directions.

§5.1 The Linear Conjugate Gradient Method

We write

$$p_k = -r_k + \beta_k p_{k-1}\,, \tag{13}$$

where 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 (13) by $p_{k-1}^{\mathrm{T}} A$ and imposing the condition $p_{k-1}^{\mathrm{T}} A p_k = 0$, we find that

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

We choose the first search direction p_0 to be the steepest descent direction at the initial point x_0. As in the general conjugate direction method, we perform successive one-dimensional minimizations along each of the search directions.

§5.1 The Linear Conjugate Gradient Method

Algorithm 5.1 (CG - Preliminary version)

Given x_0;

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

while $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k}; \tag{14a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{14b}$$

$$r_{k+1} \leftarrow Ax_{k+1} - b; \tag{14c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^{\mathrm{T}} A p_k}{p_k^{\mathrm{T}} A p_k}; \tag{14d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{14e}$$

$$k \leftarrow k+1; \tag{14f}$$

end (while)

§5.1 The Linear Conjugate Gradient Method

This version is useful for studying the essential properties of the conjugate gradient method, but we present a more efficient version later. We show first that the directions $p_0, p_1, \cdots, p_{n-1}$ are indeed conjugate, which by the previous theorem implies termination in n steps. The theorem in the next slide establishes this property and two other important properties. First, the residuals r_i are mutually orthogonal. Second, each search direction p_k and residual r_k is contained in the Krylov subspace of degree k for r_0, defined as

$$\mathcal{K}(r_0; k) = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}. \tag{15}$$

§5.1 The Linear Conjugate Gradient Method

Theorem

Suppose that the k-th iterate generated by the conjugate gradient method is not the solution point x_*. Then

$$p_k^{\mathrm{T}} A p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1. \tag{16}$$

$$r_k^{\mathrm{T}} r_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1, \tag{17}$$

$$\mathrm{span}\{r_0, r_1, \cdots, r_k\} = \mathrm{span}\{r_0, Ar_0, \cdots, A^k r_0\}, \tag{18}$$

$$\mathrm{span}\{p_0, p_1, \cdots, p_k\} = \mathrm{span}\{r_0, Ar_0, \cdots, A^k r_0\}, \tag{19}$$

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

Proof.

The proof is by induction. The expressions (18) and (19) hold trivially for $k = 0$, while (16) holds by construction for $k = 1$. Assuming now that these three expressions are true for some k (the induction hypothesis), we show that they continue to hold for $k + 1$. □

§5.1 The Linear Conjugate Gradient Method

Theorem

Suppose that the k-th iterate generated by the conjugate gradient method is not the solution point x_*. Then

$$p_k^{\mathrm{T}} A p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1. \tag{16}$$

$$r_k^{\mathrm{T}} r_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1, \tag{17}$$

$$\mathrm{span}\{r_0, r_1, \cdots, r_k\} = \mathrm{span}\{r_0, Ar_0, \cdots, A^k r_0\}, \tag{18}$$

$$\mathrm{span}\{p_0, p_1, \cdots, p_k\} = \mathrm{span}\{r_0, Ar_0, \cdots, A^k r_0\}, \tag{19}$$

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

Proof.

The proof is by induction. The expressions (18) and (19) hold trivially for $k = 0$, while (16) holds by construction for $k = 1$. Assuming now that these three expressions are true for some k (the induction hypothesis), we show that they continue to hold for $k + 1$. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

To prove (18), we show first "$\subseteq$" that the set on the left-hand side is contained in the set on the right-hand side. Because of the induction hypothesis, we have from (18) and (19) that $r_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$, $p_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$, while by multiplying the second of these expressions by A, we obtain

$$Ap_k \in \text{span}\{Ar_0, \cdots, A^{k+1} r_0\}. \tag{20}$$

By the identity $r_{k+1} = r_k + \alpha_k A p_k$, we find that

$$r_{k+1} \in \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}.$$

By combining this expression with the induction hypothesis for (18), we conclude that

$$\text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\} \subseteq \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}. \qquad \square$$

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

To prove (18), we show first "$\subseteq$" that the set on the left-hand side is contained in the set on the right-hand side. Because of the induction hypothesis, we have from (18) and (19) that $r_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$, $p_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$, while by multiplying the second of these expressions by A, we obtain

$$Ap_k \in \text{span}\{Ar_0, \cdots, A^{k+1} r_0\}. \tag{20}$$

By the identity $r_{k+1} = r_k + \alpha_k A p_k$, we find that

$$r_{k+1} \in \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}.$$

By combining this expression with the induction hypothesis for (18), we conclude that

$$\text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\} \subseteq \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}. \qquad \square$$

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

To prove that the reverse inclusion "$\supseteq$" holds as well, we use the induction hypothesis on (19) to deduce that

$$A^{k+1} r_0 = A(A^k r_0) \in \text{span}\{Ap_0, Ap_1, \cdots, Ap_k\}.$$

Since by the identity $r_{k+1} = r_k + \alpha_k A p_k$ we have $Ap_i = \dfrac{r_{i+1} - r_i}{\alpha_i}$ for $i = 0, 1, \cdots, k$, it follows that

$$A^{k+1} r_0 \in \text{span}\{r_0, r_1, \cdots, r_{k+1}\}.$$

By combining this expression with the induction hypothesis for (18), we find that

$$\text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\} \subseteq \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\}.$$

Therefore, the relation (18) continues to hold when k is replaced by $k+1$, as claimed. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

We show that (19) continues to hold when k is replaced by $k+1$ by the following argument:

$$
\begin{aligned}
&\text{span}\{p_0, p_1, \cdots, p_k, p_{k+1}\} \\
&\quad = \text{span}\{p_0, p_1, \cdots, p_k, r_{k+1}\} \quad \text{by (14e): } p_{k+1} = -r_{k+1} + \beta_{k+1} p_k \\
&\quad = \text{span}\{r_0, Ar_0, \cdots, A^k r_0, r_{k+1}\} \quad \text{by induction hypothesis for (19)} \\
&\quad = \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\} \quad \text{by (18)} \\
&\quad = \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\} \quad \text{by (18) for } k+1. \qquad \square
\end{aligned}
$$

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

Next, we prove the conjugacy condition (16) with k replaced by $k+1$. By multiplying (14e) by Ap_i, $i = 0, 1, \cdots, k$, we obtain

$$p_{k+1}^T A p_i = -r_{k+1}^{\mathrm{T}} A p_i + \beta_{k+1} p_k^{\mathrm{T}} A p_i . \tag{21}$$

By the definition (14d) of β_k $\left(\beta_{k+1} = \dfrac{r_{k+1}^{\mathrm{T}} A p_k}{p_k^{\mathrm{T}} A p_k}\right)$, the right-hand side of (21) vanishes when $i = k$; thus $p_{k+1}^T A p_k = 0$. For $i \leqslant k-1$ we need to collect a number of observations. Note first that our induction hypothesis for (16) implies that the directions p_0, p_1, $\cdots$, p_k are conjugate, so we can apply the previous theorem to deduce that

$$r_{k+1}^{\mathrm{T}} p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k . \tag{22}$$

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

Next, we prove the conjugacy condition (16) with k replaced by $k+1$. By multiplying (14e) by Ap_i, $i = 0, 1, \cdots, k$, we obtain

$$p_{k+1}^T A p_i = -r_{k+1}^{\mathrm{T}} A p_i + \beta_{k+1} p_k^{\mathrm{T}} A p_i . \tag{21}$$

By the definition (14d) of β_k $\left(\beta_{k+1} = \dfrac{r_{k+1}^{\mathrm{T}} A p_k}{p_k^{\mathrm{T}} A p_k}\right)$, the right-hand side of (21) vanishes when $i = k$; thus $p_{k+1}^T A p_k = 0$. For $i \leqslant k-1$ we need to collect a number of observations. Note first that our induction hypothesis for (16) implies that the directions p_0, p_1, $\cdots$, p_k are conjugate, so we can apply the previous theorem to deduce that

$$r_{k+1}^{\mathrm{T}} p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k . \tag{22}$$

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

Second, by repeatedly applying (19), we find that for $i = 0, 1, \cdots, k-1$, the following inclusion holds:

$$\begin{aligned} A p_i &\in A\text{span}\{r_0, Ar_0, \cdots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \cdots, A^{i+1} r_0\} \\ &\subseteq \text{span}\{p_0, p_1, \cdots, p_{i+1}\}. \end{aligned} \tag{23}$$

By combining (22) and (23), we deduce that

$$r_{k+1}^{\mathrm{T}} A p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1,$$

so the first term in the right-hand side of (21) vanishes for $i = 0, 1, \cdots, k-1$. Because of the induction hypothesis for (16), the second term vanishes as well, and we conclude that $p_{k+1}^{\mathrm{T}} A p_i = 0$, $i = 0, 1, \cdots, k$. Hence, the induction argument holds for (16) also.

□

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

Second, by repeatedly applying (19), we find that for $i = 0, 1, \cdots, k-1$, the following inclusion holds:

$$\begin{aligned} A p_i &\in A\text{span}\{r_0, Ar_0, \cdots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \cdots, A^{i+1} r_0\} \\ &\subseteq \text{span}\{p_0, p_1, \cdots, p_{i+1}\}. \end{aligned} \tag{23}$$

By combining (22) and (23), we deduce that

$$r_{k+1}^{\mathrm{T}} A p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1,$$

so the first term in the right-hand side of (21) vanishes for $i = 0, 1, \cdots, k-1$. Because of the induction hypothesis for (16), the second term vanishes as well, and we conclude that $p_{k+1}^{\mathrm{T}} A p_i = 0$, $i = 0, 1, \cdots, k$. Hence, the induction argument holds for (16) also.

□

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

Second, by repeatedly applying (19), we find that for $i = 0, 1, \cdots, k-1$, the following inclusion holds:

$$\begin{aligned} A p_i &\in A\text{span}\{r_0, Ar_0, \cdots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \cdots, A^{i+1} r_0\} \\ &\subseteq \text{span}\{p_0, p_1, \cdots, p_{i+1}\}. \end{aligned} \tag{23}$$

By combining (22) and (23), we deduce that

$$r_{k+1}^{\mathrm{T}} A p_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1,$$

so the first term in the right-hand side of (21) vanishes for $i = 0, 1, \cdots, k-1$. Because of the induction hypothesis for (16), the second term vanishes as well, and we conclude that $p_{k+1}^{\mathrm{T}} A p_i = 0$, $i = 0, 1, \cdots, k$. Hence, the induction argument holds for (16) also.

□

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

It follows that the direction set generated by the conjugate gradient method is indeed a conjugate direction set, so the algorithm terminates in at most n iterations.

Finally, we prove (17) by a non-inductive argument. Because the direction set is conjugate, we have from (11) that $r_k^{\mathrm{T}} p_i = 0$ for all $i = 0, 1, \cdots, k-1$ and any $k = 1, 2, \cdots, n-1$. By rearranging (14e), we find that

$$r_i = -p_i + \beta_i p_{i-1},$$

so that $r_i \in \mathrm{span}\{p_i, p_{i-1}\}$ for all $i = 1, \cdots, k-1$. We conclude that $r_k^{\mathrm{T}} r_i = 0$ for all $i = 1, \cdots, k-1$. To complete the proof, we note that $r_k^{\mathrm{T}} r_0 = -r_k^{\mathrm{T}} p_0 = 0$, by definition of p_0 in Algorithm 5.1 and by (11). □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

It follows that the direction set generated by the conjugate gradient method is indeed a conjugate direction set, so the algorithm terminates in at most n iterations.

Finally, we prove (17) by a non-inductive argument. Because the direction set is conjugate, we have from (11) that $r_k^{\mathrm{T}} p_i = 0$ for all $i = 0, 1, \cdots, k-1$ and any $k = 1, 2, \cdots, n-1$. By rearranging (14e), we find that

$$r_i = -p_i + \beta_i p_{i-1},$$

so that $r_i \in \operatorname{span}\{p_i, p_{i-1}\}$ for all $i = 1, \cdots, k-1$. We conclude that $r_k^{\mathrm{T}} r_i = 0$ for all $i = 1, \cdots, k-1$. To complete the proof, we note that $r_k^{\mathrm{T}} r_0 = -r_k^{\mathrm{T}} p_0 = 0$, by definition of p_0 in Algorithm 5.1 and by (11). □

§5.1 The Linear Conjugate Gradient 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.

§5.1 The Linear Conjugate Gradient Method

- **A Practical Form of the Conjugate Gradient Method**

Recall **Algorithm 5.1** (CG - Preliminary version)

Given x_0;

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

while $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k}; \tag{14a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{14b}$$

$$r_{k+1} \leftarrow Ax_{k+1} - b; \tag{14c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^{\mathrm{T}} A p_k}{p_k^{\mathrm{T}} A p_k}; \tag{14d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{14e}$$

$$k \leftarrow k+1; \tag{14f}$$

end (while)

§5.1 The Linear Conjugate Gradient Method

We can derive a slightly more economical form of the conjugate gradient method by using the established results. First, we can use (14e) and (11) ($r_k^{\mathrm{T}} p_i = 0$ for $0 \leqslant i \leqslant k-1$) to replace the formula (14a) for α_k by

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}.$$

Second, we have from (10) that $\alpha_k A p_k = r_{k+1} - r_k$, so by applying (14e) and (11) once again we can simplify the formula for β_{k+1} to

$$\beta_{k+1} = \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}.$$

By using these formulae together with (10), we obtain the following standard form of the conjugate gradient method.

§5.1 The Linear Conjugate Gradient Method

We can derive a slightly more economical form of the conjugate gradient method by using the established results. First, we can use (14e) and (11) ($r_k^{\mathrm{T}} p_i = 0$ for $0 \leqslant i \leqslant k-1$) to replace the formula (14a) for α_k by

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}.$$

Second, we have from (10) that $\alpha_k A p_k = r_{k+1} - r_k$, so by applying (14e) and (11) once again we can simplify the formula for β_{k+1} to

$$\beta_{k+1} = \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}.$$

By using these formulae together with (10), we obtain the following standard form of the conjugate gradient method.

§5.1 The Linear Conjugate Gradient Method

We can derive a slightly more economical form of the conjugate gradient method by using the established results. First, we can use (14e) and (11) ($r_k^{\mathrm{T}} p_i = 0$ for $0 \leqslant i \leqslant k-1$) to replace the formula (14a) for α_k by

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}.$$

Second, we have from (10) that $\alpha_k A p_k = r_{k+1} - r_k$, so by applying (14e) and (11) once again we can simplify the formula for β_{k+1} to

$$\beta_{k+1} = \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}.$$

By using these formulae together with (10), we obtain the following standard form of the conjugate gradient method.

§5.1 The Linear Conjugate Gradient Method

Algorithm 5.2 (CG)

Given x_0;

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

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}; \tag{24a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{24b}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \tag{24c}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}; \tag{24d}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{24e}$$

$$k \leftarrow k+1; \tag{24f}$$

end (while)

§5.1 The Linear Conjugate Gradient Method

At any given point in Algorithm 5.2 we never need to know the vectors x, r, and p for more than the last two iterations. Accordingly, implementations of this algorithm overwrite old values of these vectors to save on storage. The major computational tasks to be performed at each step are computation of the matrix-vector product Ap_k, calculation of the inner products $p_k^{\mathrm{T}} A p_k$ and $r_{k+1}^{\mathrm{T}} r_{k+1}$, and calculation of three vector sums. The inner product and vector sum operations can be performed in a small multiple of n floating-point operations, while the cost of the matrix-vector product is, of course, dependent on the problem. The CG method is recommended only for large problems; otherwise, Gaussian elimination or other factorization algorithms such as the singular value decomposition are to be preferred, since they are less sensitive to rounding errors.

§5.1 The Linear Conjugate Gradient Method

At any given point in Algorithm 5.2 we never need to know the vectors x, r, and p for more than the last two iterations. Accordingly, implementations of this algorithm overwrite old values of these vectors to save on storage. The major computational tasks to be performed at each step are computation of the matrix-vector product Ap_k, calculation of the inner products $p_k^{\mathrm{T}} A p_k$ and $r_{k+1}^{\mathrm{T}} r_{k+1}$, and calculation of three vector sums. The inner product and vector sum operations can be performed in a small multiple of n floating-point operations, while the cost of the matrix-vector product is, of course, dependent on the problem. The CG method is recommended only for large problems; otherwise, Gaussian elimination or other factorization algorithms such as the singular value decomposition are to be preferred, since they are less sensitive to rounding errors.

§5.1 The Linear Conjugate Gradient Method

At any given point in Algorithm 5.2 we never need to know the vectors x, r, and p for more than the last two iterations. Accordingly, implementations of this algorithm overwrite old values of these vectors to save on storage. The major computational tasks to be performed at each step are computation of the matrix-vector product Ap_k, calculation of the inner products $p_k^{\mathrm{T}} A p_k$ and $r_{k+1}^{\mathrm{T}} r_{k+1}$, and calculation of three vector sums. The inner product and vector sum operations can be performed in a small multiple of n floating-point operations, while the cost of the matrix-vector product is, of course, dependent on the problem. The CG method is recommended only for large problems; otherwise, Gaussian elimination or other factorization algorithms such as the singular value decomposition are to be preferred, since they are less sensitive to rounding errors.

§5.1 The Linear Conjugate Gradient Method

At any given point in Algorithm 5.2 we never need to know the vectors x, r, and p for more than the last two iterations. Accordingly, implementations of this algorithm overwrite old values of these vectors to save on storage. The major computational tasks to be performed at each step are computation of the matrix-vector product Ap_k, calculation of the inner products $p_k^{\mathrm{T}} A p_k$ and $r_{k+1}^{\mathrm{T}} r_{k+1}$, and calculation of three vector sums. The inner product and vector sum operations can be performed in a small multiple of n floating-point operations, while the cost of the matrix-vector product is, of course, dependent on the problem. The CG method is recommended only for large problems; otherwise, Gaussian elimination or other factorization algorithms such as the singular value decomposition are to be preferred, since they are less sensitive to rounding errors.

§5.1 The Linear Conjugate Gradient Method

For large problems, the CG method has the advantage that it does not alter the coefficient matrix and (in contrast to factorization techniques) does not produce fill in the arrays holding the matrix. Another key property is that the CG method sometimes approaches the solution quickly, as we discuss next.

§5.1 The Linear Conjugate Gradient Method

- **Rate of Convergence**

We have seen that in exact arithmetic the conjugate gradient method will terminate at the solution in at most n iterations. What is more remarkable is that when the distribution of the eigenvalues of A has certain favorable features, the algorithm will identify the solution in many fewer than n iterations. To explain this property, we begin by viewing the expanding subspace minimization property proved in the previous theorem in a slightly different way, using it to show that Algorithm 5.2 is optimal in a certain important sense. From (24b) and (19), we have that

$$\begin{aligned} x_{k+1} &= x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0 \,, \end{aligned} \tag{25}$$

for some constants γ_i.

§5.1 The Linear Conjugate Gradient Method

- **Rate of Convergence**

We have seen that in exact arithmetic the conjugate gradient method will terminate at the solution in at most n iterations. What is more remarkable is that when the distribution of the eigenvalues of A has certain favorable features, the algorithm will identify the solution in many fewer than n iterations. To explain this property, we begin by viewing the expanding subspace minimization property proved in the previous theorem in a slightly different way, using it to show that Algorithm 5.2 is optimal in a certain important sense. From (24b) and (19), we have that

$$\begin{aligned} x_{k+1} &= x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0 \,, \end{aligned} \tag{25}$$

for some constants γ_i.

§5.1 The Linear Conjugate Gradient Method

We now define $p_k^*(\cdot)$ to be a polynomial of degree k with coefficients $\gamma_0, \gamma_1, \cdots, \gamma_k$. Like any polynomial, p_k^* can take either a scalar or a square matrix as its argument. For the matrix argument A, we have

$$p_k^*(A) = \gamma_0 \mathrm{I} + \gamma_1 A + \cdots + \gamma_k A^k \,,$$

which allows us to express (25) as follows:

$$x_{k+1} = x_0 + p_k^*(A) r_0 \,. \tag{26}$$

We now show that among all possible methods whose first k steps are restricted to the Krylov subspace $\mathcal{K}(r_0; k) \equiv \mathrm{span}\{r_0, A r_0, \cdots, A^k r_0\}$, Algorithm 5.2 does the best job of minimizing the distance to the solution after k steps, when this distance is measured by the weighted norm measure $\|\cdot\|_A$ defined by

$$\|z\|_A^2 = z^{\mathrm{T}} A z. \tag{27}$$

§5.1 The Linear Conjugate Gradient Method

We now define $p_k^*(\cdot)$ to be a polynomial of degree k with coefficients $\gamma_0, \gamma_1, \cdots, \gamma_k$. Like any polynomial, p_k^* can take either a scalar or a square matrix as its argument. For the matrix argument A, we have

$$p_k^*(A) = \gamma_0 \mathrm{I} + \gamma_1 A + \cdots + \gamma_k A^k \,,$$

which allows us to express (25) as follows:

$$x_{k+1} = x_0 + p_k^*(A) r_0 \,. \tag{26}$$

We now show that among all possible methods whose first k steps are restricted to the Krylov subspace $\mathcal{K}(r_0; k) \equiv \mathrm{span}\{r_0, A r_0, \cdots, A^k r_0\}$, Algorithm 5.2 does the best job of minimizing the distance to the solution after k steps, when this distance is measured by the weighted norm measure $\|\cdot\|_A$ defined by

$$\|z\|_A^2 = z^{\mathrm{T}} A z. \tag{27}$$

§5.1 The Linear Conjugate Gradient Method

Using this norm and the definition of φ, and the fact that x_* minimizes φ, it is easy to show that

$$\frac{1}{2}\|x - x_*\|_A^2 = \frac{1}{2}(x - x_*)^{\mathrm{T}} A(x - x_*) = \varphi(x) - \varphi(x_*)\,. \tag{28}$$

One of previous theorems states that x_{k+1} minimizes φ, and hence $\|x-x_*\|_A^2$, over the set $x_0+\text{span}\{p_0, p_1, \cdots, p_k\}$, which by (19) is the same as $x_0+\text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$. It follows from (5.26) that the polynomial p_k^* solves the following problem in which the minimum is taken over the space of all possible polynomials of degree k:

$$\min_{P_k} \|x_0 + P_k(A) r_0 - x_*\|_A. \tag{29}$$

We exploit this optimality property repeatedly in the remainder of the section. Since $r_0 = Ax_0 - b = Ax_0 - Ax_* = A(x_0 - x_*)$,

$$x_{k+1} - x_* = x_0 + p_k^*(A) r_0 - x_* = \left[\mathrm{I} + p_k^*(A) A\right](x_0 - x_*)\,. \tag{30}$$

§5.1 The Linear Conjugate Gradient Method

Using this norm and the definition of φ, and the fact that x_* minimizes φ, it is easy to show that

$$\frac{1}{2}\|x - x_*\|_A^2 = \frac{1}{2}(x - x_*)^{\mathrm{T}} A(x - x_*) = \varphi(x) - \varphi(x_*)\,. \tag{28}$$

One of previous theorems states that x_{k+1} minimizes φ, and hence $\|x-x_*\|_A^2$, over the set $x_0+\text{span}\{p_0, p_1, \cdots, p_k\}$, which by (19) is the same as $x_0+\text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$. It follows from (5.26) that the polynomial p_k^* solves the following problem in which the minimum is taken over the space of all possible polynomials of degree k:

$$\min_{P_k} \left\|x_0 + P_k(A)r_0 - x_*\right\|_A. \tag{29}$$

We exploit this optimality property repeatedly in the remainder of the section. Since $r_0 = Ax_0 - b = Ax_0 - Ax_* = A(x_0 - x_*)$,

$$x_{k+1} - x_* = x_0 + p_k^*(A)r_0 - x_* = \left[\mathrm{I} + p_k^*(A)A\right](x_0 - x_*)\,. \tag{30}$$

§5.1 The Linear Conjugate Gradient Method

Let $0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$ be the eigenvalues of A, and let v_1, v_2, $\cdots$, v_n be the corresponding orthonormal eigenvectors, so that

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^{\mathrm{T}} .$$

Since the eigenvectors span the whole space $\mathbb{R}^n$, we can write

$$x_0 - x_* = \sum_{i=1}^{n} \xi_i v_i , \tag{31}$$

for some coefficients ξ_i. It is easy to show that any eigenvector of A is also an eigenvector of $P_k(A)$ for any polynomial P_k. For our particular matrix A and its eigenvalues λ_i and eigenvectors v_i,

$$P_k(A)v_i = P_k(\lambda_i)v_i \quad \text{for} \quad i = 1, 2, \cdots, n .$$

By substituting (31) into (30) we have

$$x_{k+1} - x_* = \sum_{i=1}^{n} \left[1 + \lambda_i p_k^*(\lambda_i)\right] \xi_i v_i .$$

§5.1 The Linear Conjugate Gradient Method

By using the fact that $\|z\|_A^2 = z^{\mathrm{T}}Az = \sum_{i=1}^{n} \lambda_i (v_i^{\mathrm{T}} z)^2$, we have

$$\|x_{k+1} - x_*\|_A^2 = \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i p_k^*(\lambda_i)\big]^2 \xi_i^2 \,. \tag{32}$$

Since the polynomial p_k^* generated by the CG method is optimal with respect to this norm, we have

$$\|x_{k+1} - x_*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \xi_i^2 \,.$$

By extracting the largest of the terms $\big[1 + \lambda_i P_k(\lambda_i)\big]^2$ from this expression, we obtain that

$$\begin{aligned} \|x_{k+1} - x_*\|_A^2 &\leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \Big(\sum_{j=1}^{n} \lambda_j \xi_j^2 \Big) \\ &= \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \|x_0 - x_*\|_A^2 \,, \end{aligned} \tag{33}$$

where we have used the fact that $\|x_0 - x_*\|_A^2 = \sum_{j=1}^{n} \lambda_j \xi_j^2$.

§5.1 The Linear Conjugate Gradient Method

By using the fact that $\|z\|_A^2 = z^{\mathrm{T}}Az = \sum_{i=1}^{n} \lambda_i (v_i^{\mathrm{T}} z)^2$, we have

$$\|x_{k+1} - x_*\|_A^2 = \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i p_k^*(\lambda_i)\big]^2 \xi_i^2 \,. \tag{32}$$

Since the polynomial p_k^* generated by the CG method is optimal with respect to this norm, we have

$$\|x_{k+1} - x_*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \xi_i^2 \,.$$

By extracting the largest of the terms $\big[1 + \lambda_i P_k(\lambda_i)\big]^2$ from this expression, we obtain that

$$\begin{aligned} \|x_{k+1} - x_*\|_A^2 &\leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \Big(\sum_{j=1}^{n} \lambda_j \xi_j^2 \Big) \\ &= \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \|x_0 - x_*\|_A^2 \,, \end{aligned} \tag{33}$$

where we have used the fact that $\|x_0 - x_*\|_A^2 = \sum_{j=1}^{n} \lambda_j \xi_j^2$.

§5.1 The Linear Conjugate Gradient Method

By using the fact that $\|z\|_A^2 = z^{\mathrm{T}}Az = \sum_{i=1}^{n} \lambda_i (v_i^{\mathrm{T}} z)^2$, we have

$$\|x_{k+1} - x_*\|_A^2 = \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i p_k^*(\lambda_i)\big]^2 \xi_i^2 \,. \tag{32}$$

Since the polynomial p_k^* generated by the CG method is optimal with respect to this norm, we have

$$\|x_{k+1} - x_*\|_A^2 = \min_{P_k} \sum_{i=1}^{n} \lambda_i \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \xi_i^2 \,.$$

By extracting the largest of the terms $\big[1 + \lambda_i P_k(\lambda_i)\big]^2$ from this expression, we obtain that

$$\begin{aligned} \|x_{k+1} - x_*\|_A^2 &\leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \Big(\sum_{j=1}^{n} \lambda_j \xi_j^2 \Big) \\ &= \min_{P_k} \max_{1 \leqslant i \leqslant n} \big[1 + \lambda_i P_k(\lambda_i)\big]^2 \|x_0 - x_*\|_A^2 \,, \end{aligned} \tag{33}$$

where we have used the fact that $\|x_0 - x_*\|_A^2 = \sum_{j=1}^{n} \lambda_j \xi_j^2$.

§5.1 The Linear Conjugate Gradient Method

The expression (33) allows us to quantify the convergence rate of the CG method by estimating the non-negative scalar quantity

$$\min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2 . \tag{34}$$

In other words, we search for a polynomial P_k that makes this expression as small as possible. In some practical cases, we can find this polynomial explicitly and draw some interesting conclusions about the properties of the CG method. The following result is an example.

§5.1 The Linear Conjugate Gradient Method

Theorem

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

Proof.

Suppose that the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ take on the r distinct values $\tau_1 < \tau_2 < \cdots < \tau_r$. We define a polynomial $Q_r(\lambda)$ by

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r}(\lambda - \tau_1)(\lambda - \tau_2)\cdots(\lambda - \tau_r)\,.$$

Since $Q_r(\lambda_i) = 0$ for $i = 1, 2, \cdots, n$ and $Q_r(0) = 1$, we deduce that $Q_r(\lambda) - 1$ is a polynomial of degree r with a root at $\lambda = 0$, so the function $\bar{P}_{r-1}$ defined by

$$\bar{P}_{r-1}(\lambda) = \frac{Q_r(\lambda) - 1}{\lambda}$$

is a polynomial of degree $r - 1$. □

§5.1 The Linear Conjugate Gradient Method

Theorem

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

Proof.

Suppose that the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ take on the r distinct values $\tau_1 < \tau_2 < \cdots < \tau_r$. We define a polynomial $Q_r(\lambda)$ by

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r}(\lambda - \tau_1)(\lambda - \tau_2)\cdots(\lambda - \tau_r)\,.$$

Since $Q_r(\lambda_i) = 0$ for $i = 1, 2, \cdots, n$ and $Q_r(0) = 1$, we deduce that $Q_r(\lambda) - 1$ is a polynomial of degree r with a root at $\lambda = 0$, so the function $\bar{P}_{r-1}$ defined by

$$\bar{P}_{r-1}(\lambda) = \frac{Q_r(\lambda) - 1}{\lambda}$$

is a polynomial of degree $r - 1$. □

§5.1 The Linear Conjugate Gradient Method

Theorem

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

Proof.

Suppose that the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ take on the r distinct values $\tau_1 < \tau_2 < \cdots < \tau_r$. We define a polynomial $Q_r(\lambda)$ by

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r}(\lambda - \tau_1)(\lambda - \tau_2)\cdots(\lambda - \tau_r)\,.$$

Since $Q_r(\lambda_i) = 0$ for $i = 1, 2, \cdots, n$ and $Q_r(0) = 1$, we deduce that $Q_r(\lambda) - 1$ is a polynomial of degree r with a root at $\lambda = 0$, so the function $\bar{P}_{r-1}$ defined by

$$\bar{P}_{r-1}(\lambda) = \frac{Q_r(\lambda) - 1}{\lambda}$$

is a polynomial of degree $r - 1$. □

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

By setting $k = r - 1$ in

$$\min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2, \tag{34}$$

we have

$$\begin{aligned} 0 \leqslant \min_{P_{r-1}} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_{r-1}(\lambda_i)\right]^2 &\leqslant \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i \bar{P}_{r-1}(\lambda_i)\right]^2 \\ = \max_{1 \leqslant i \leqslant n} Q_r^2(\lambda_i) &= 0\,. \end{aligned}$$

Hence, the constant in (34) is zero for the value $k = r - 1$, so we have by substituting into (33) that $\|x_r - x_*\|_A^2 = 0$, and therefore $x_r = x_*$, as claimed. □

By using similar reasoning (choosing a particular Q), Luenberger [195] establishes the estimate in the next slide, which gives a useful characterization of the behavior of the CG method.

§5.1 The Linear Conjugate Gradient Method

Proof (cont'd).

By setting $k = r - 1$ in

$$\min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2, \tag{34}$$

we have

$$\begin{aligned} 0 \leqslant \min_{P_{r-1}} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_{r-1}(\lambda_i)\right]^2 &\leqslant \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i \bar{P}_{r-1}(\lambda_i)\right]^2 \\ = \max_{1 \leqslant i \leqslant n} Q_r^2(\lambda_i) &= 0\,. \end{aligned}$$

Hence, the constant in (34) is zero for the value $k = r - 1$, so we have by substituting into (33) that $\|x_r - x_*\|_A^2 = 0$, and therefore $x_r = x_*$, as claimed. □

By using similar reasoning (choosing a particular Q), Luenberger [195] establishes the estimate in the next slide, which gives a useful characterization of the behavior of the CG method.

§5.1 The Linear Conjugate Gradient Method

Theorem

If A has eigenvalues $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$, we have that

$$\|x_{k+1} - x_*\|_A^2 \leqslant \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x_0 - x_*\|_A^2 . \tag{35}$$

Without giving details of the proof, we describe how this result is obtained from

$$\|x_{k+1} - x_*\|_A^2 \leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2 \|x_0 - x_*\|_A^2 . \tag{33}$$

One selects a polynomial $\bar{P}_k$ of degree k such that the polynomial $Q_{k+1}(\lambda) = 1 + \lambda \bar{P}_k(\lambda)$ has roots at the k largest eigenvalues $\lambda_n, \lambda_{n-1}, \cdots, \lambda_{n-k+1}$, as well as at the midpoint between λ_1 and λ_{n-k}. It can be shown that the maximum value attained by Q_{k+1} on the remaining eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{n-k}$ is precisely $(\lambda_{n-k} - \lambda_1)/(\lambda_{n-k} + \lambda_1)$.

§5.1 The Linear Conjugate Gradient Method

Theorem

If A has eigenvalues $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$, we have that

$$\|x_{k+1} - x_*\|_A^2 \leqslant \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x_0 - x_*\|_A^2 . \tag{35}$$

Without giving details of the proof, we describe how this result is obtained from

$$\|x_{k+1} - x_*\|_A^2 \leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2 \|x_0 - x_*\|_A^2 . \tag{33}$$

One selects a polynomial $\bar{P}_k$ of degree k such that the polynomial $Q_{k+1}(\lambda) = 1 + \lambda \bar{P}_k(\lambda)$ has roots at the k largest eigenvalues $\lambda_n, \lambda_{n-1}, \cdots, \lambda_{n-k+1}$, as well as at the midpoint between λ_1 and λ_{n-k}. It can be shown that the maximum value attained by Q_{k+1} on the remaining eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{n-k}$ is precisely $(\lambda_{n-k} - \lambda_1)/(\lambda_{n-k} + \lambda_1)$.

§5.1 The Linear Conjugate Gradient Method

Theorem

If A has eigenvalues $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$, we have that

$$\|x_{k+1} - x_*\|_A^2 \leqslant \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x_0 - x_*\|_A^2 . \tag{35}$$

Without giving details of the proof, we describe how this result is obtained from

$$\|x_{k+1} - x_*\|_A^2 \leqslant \min_{P_k} \max_{1 \leqslant i \leqslant n} \left[1 + \lambda_i P_k(\lambda_i)\right]^2 \|x_0 - x_*\|_A^2 . \tag{33}$$

One selects a polynomial $\bar{P}_k$ of degree k such that the polynomial $Q_{k+1}(\lambda) = 1 + \lambda \bar{P}_k(\lambda)$ has roots at the k largest eigenvalues $\lambda_n, \lambda_{n-1}, \cdots, \lambda_{n-k+1}$, as well as at the midpoint between λ_1 and λ_{n-k}. It can be shown that the maximum value attained by Q_{k+1} on the remaining eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{n-k}$ is precisely $(\lambda_{n-k} - \lambda_1)/(\lambda_{n-k} + \lambda_1)$.

§5.1 The Linear Conjugate Gradient Method

We now illustrate how estimate (35) 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 and $n-m$ smaller eigenvalues clustered around 1.

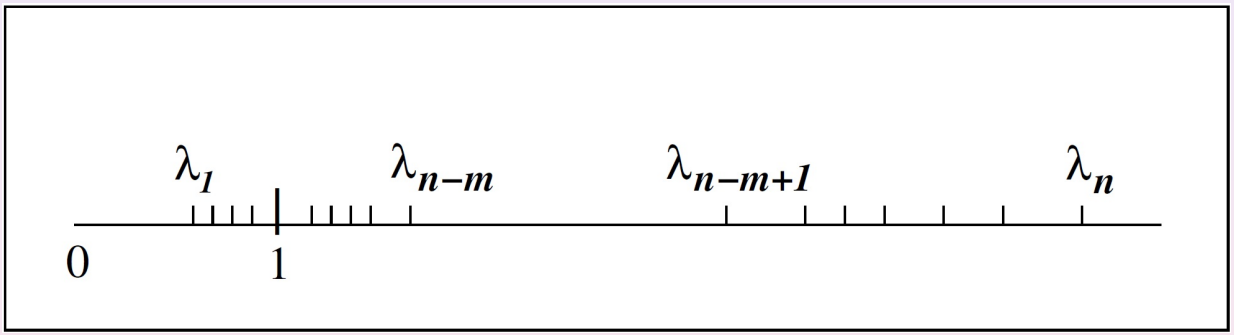

Figure 3: Two clusters of eigenvalues.

If we let $k = m$ in (35) and define $\varepsilon = \lambda_{n-m} - \lambda_1$, we have

$$\|x_{m+1} - x_*\|_A \approx \varepsilon \|x_0 - x_*\|_A .$$

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.

§5.1 The Linear Conjugate Gradient Method

We now illustrate how estimate (35) 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 and $n-m$ smaller eigenvalues clustered around 1.

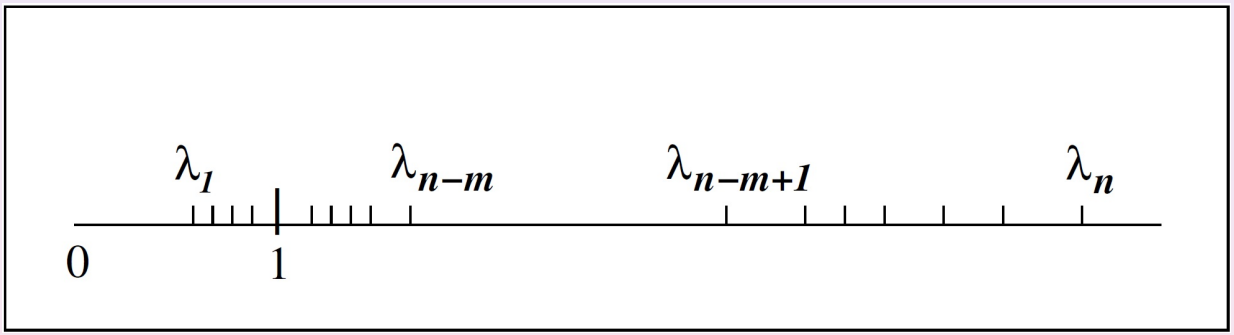

Figure 3: Two clusters of eigenvalues.

If we let $k = m$ in (35) and define $\varepsilon = \lambda_{n-m} - \lambda_1$, we have

$$\|x_{m+1} - x_*\|_A \approx \varepsilon \|x_0 - x_*\|_A .$$

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.

§5.1 The Linear Conjugate Gradient Method

Figure 4 (in the next slide) shows the behavior of CG on a problem of this type, which has five large eigenvalues with all the smaller eigenvalues clustered between 0.95 and 1.05, and compares this behavior with that of CG on a problem in which the eigenvalues satisfy some random distribution. In both cases, we plot the log of φ after each iteration. For the problem with clustered eigenvalues, estimate (35) predicts a sharp decrease in the error measure at iteration 6. Note, however, that this decrease was achieved one iteration earlier, illustrating the fact that estimate (35) gives only an upper bound, and that the rate of convergence can be faster. By contrast, we observe in Figure 4 that for the problem with randomly distributed eigenvalues (dashed line), the convergence rate is slower and more uniform.

§5.1 The Linear Conjugate Gradient Method

Figure 4 (in the next slide) shows the behavior of CG on a problem of this type, which has five large eigenvalues with all the smaller eigenvalues clustered between 0.95 and 1.05, and compares this behavior with that of CG on a problem in which the eigenvalues satisfy some random distribution. In both cases, we plot the log of φ after each iteration. For the problem with clustered eigenvalues, estimate (35) predicts a sharp decrease in the error measure at iteration 6. Note, however, that this decrease was achieved one iteration earlier, illustrating the fact that estimate (35) gives only an upper bound, and that the rate of convergence can be faster. By contrast, we observe in Figure 4 that for the problem with randomly distributed eigenvalues (dashed line), the convergence rate is slower and more uniform.

§5.1 The Linear Conjugate Gradient Method

Figure 4 (in the next slide) shows the behavior of CG on a problem of this type, which has five large eigenvalues with all the smaller eigenvalues clustered between 0.95 and 1.05, and compares this behavior with that of CG on a problem in which the eigenvalues satisfy some random distribution. In both cases, we plot the log of φ after each iteration. For the problem with clustered eigenvalues, estimate (35) predicts a sharp decrease in the error measure at iteration 6. Note, however, that this decrease was achieved one iteration earlier, illustrating the fact that estimate (35) gives only an upper bound, and that the rate of convergence can be faster. By contrast, we observe in Figure 4 that for the problem with randomly distributed eigenvalues (dashed line), the convergence rate is slower and more uniform.

§5.1 The Linear Conjugate Gradient Method

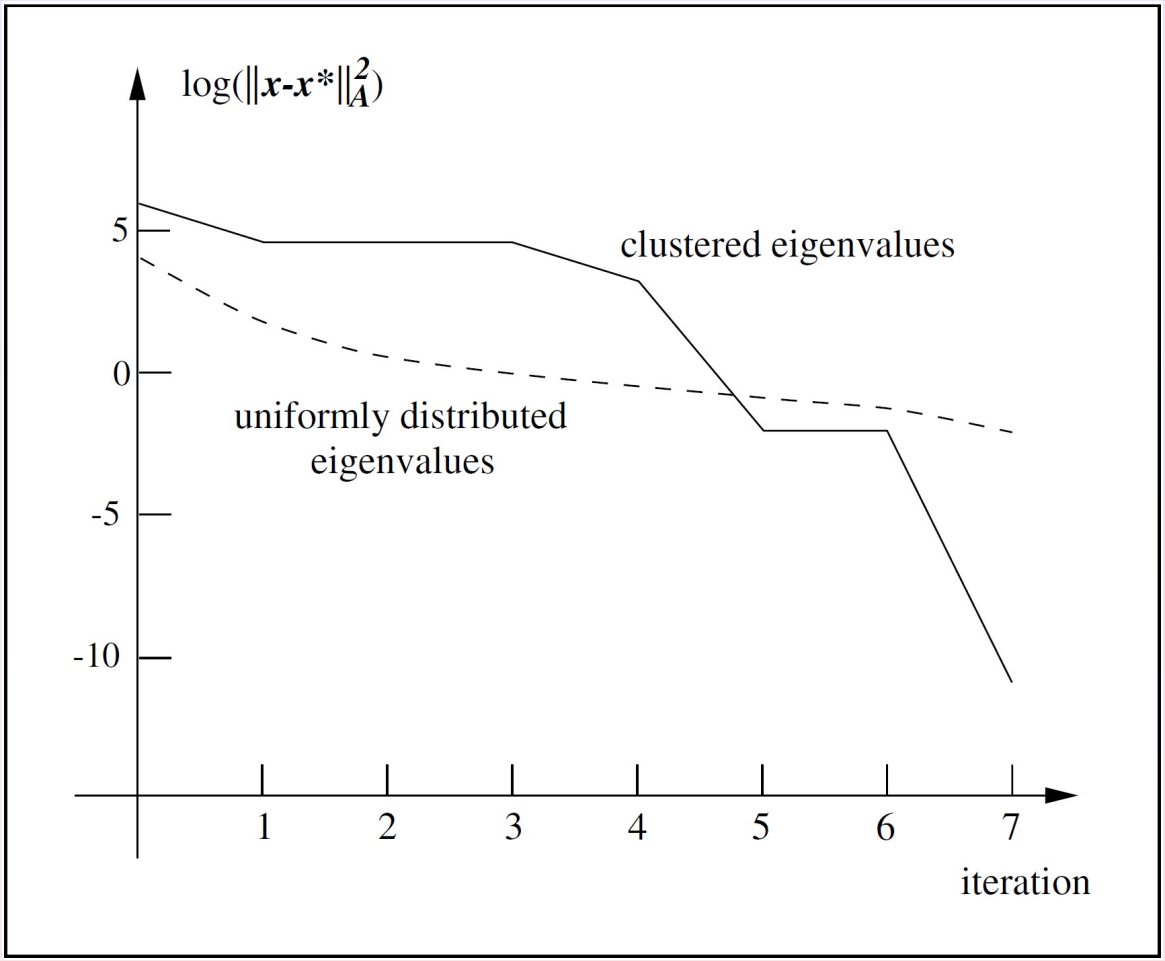

Figure 4: Performance of the conjugate gradient method on (a) a problem in which five of the eigenvalues are large and the remainder are clustered near 1, and (b) a matrix with uniformly distributed eigenvalues.

§5.1 The Linear Conjugate Gradient Method

Figure 4 illustrates another interesting feature: After one more iteration (a total of seven) on the problem with clustered eigenvalues, the error measure drops sharply. An extension of the arguments leading to one of previous theorems explains this behavior: it is almost true to say that the matrix A has just six distinct eigenvalues: the five large eigenvalues and 1. Then we would expect the error measure to be zero after six iterations. Because the eigenvalues near 1 are slightly spread out, however, the error does not become very small until iteration 7.

§5.1 The Linear Conjugate Gradient Method

Figure 4 illustrates another interesting feature: After one more iteration (a total of seven) on the problem with clustered eigenvalues, the error measure drops sharply. An extension of the arguments leading to one of previous theorems explains this behavior: it is almost true to say that the matrix A has just six distinct eigenvalues: the five large eigenvalues and 1. Then we would expect the error measure to be zero after six iterations. Because the eigenvalues near 1 are slightly spread out, however, the error does not become very small until iteration 7.

§5.1 The Linear Conjugate Gradient Method

To state this claim more precisely, it is generally true that if the eigenvalues occur in r distinct clusters, the CG iterates will approximately solve the problem in about r steps (see [136]). This result can be proved by constructing a polynomial $\bar{P}_{r-1}$ such that $(1 + \lambda \bar{P}_{r-1}(\lambda))$ has zeros inside each of the clusters. This polynomial may not vanish at the eigenvalues λ_i, $i = 1, 2, \cdots, n$, but its value will be small at these points, so the constant defined in (34) will be small for $k \geqslant r - 1$. We illustrate this behavior in Figure 5, which shows the performance of CG on a matrix of dimension $n = 14$ that has four clusters of eigenvalues: single eigenvalues at 140 and 120, a cluster of 10 eigenvalues very close to 10, with the remaining eigenvalues clustered between 0.95 and 1.05. After four iterations, the error has decreased significantly. After six iterations, the solution is identified to good accuracy.

§5.1 The Linear Conjugate Gradient Method

To state this claim more precisely, it is generally true that if the eigenvalues occur in r distinct clusters, the CG iterates will approximately solve the problem in about r steps (see [136]). This result can be proved by constructing a polynomial $\bar{P}_{r-1}$ such that $(1 + \lambda \bar{P}_{r-1}(\lambda))$ has zeros inside each of the clusters. This polynomial may not vanish at the eigenvalues λ_i, $i = 1, 2, \cdots, n$, but its value will be small at these points, so the constant defined in (34) will be small for $k \geqslant r - 1$. We illustrate this behavior in Figure 5, which shows the performance of CG on a matrix of dimension $n = 14$ that has four clusters of eigenvalues: single eigenvalues at 140 and 120, a cluster of 10 eigenvalues very close to 10, with the remaining eigenvalues clustered between 0.95 and 1.05. After four iterations, the error has decreased significantly. After six iterations, the solution is identified to good accuracy.

§5.1 The Linear Conjugate Gradient Method

To state this claim more precisely, it is generally true that if the eigenvalues occur in r distinct clusters, the CG iterates will approximately solve the problem in about r steps (see [136]). This result can be proved by constructing a polynomial $\bar{P}_{r-1}$ such that $(1 + \lambda \bar{P}_{r-1}(\lambda))$ has zeros inside each of the clusters. This polynomial may not vanish at the eigenvalues λ_i, $i = 1, 2, \cdots, n$, but its value will be small at these points, so the constant defined in (34) will be small for $k \geqslant r - 1$. We illustrate this behavior in Figure 5, which shows the performance of CG on a matrix of dimension $n = 14$ that has four clusters of eigenvalues: single eigenvalues at 140 and 120, a cluster of 10 eigenvalues very close to 10, with the remaining eigenvalues clustered between 0.95 and 1.05. After four iterations, the error has decreased significantly. After six iterations, the solution is identified to good accuracy.

§5.1 The Linear Conjugate Gradient Method

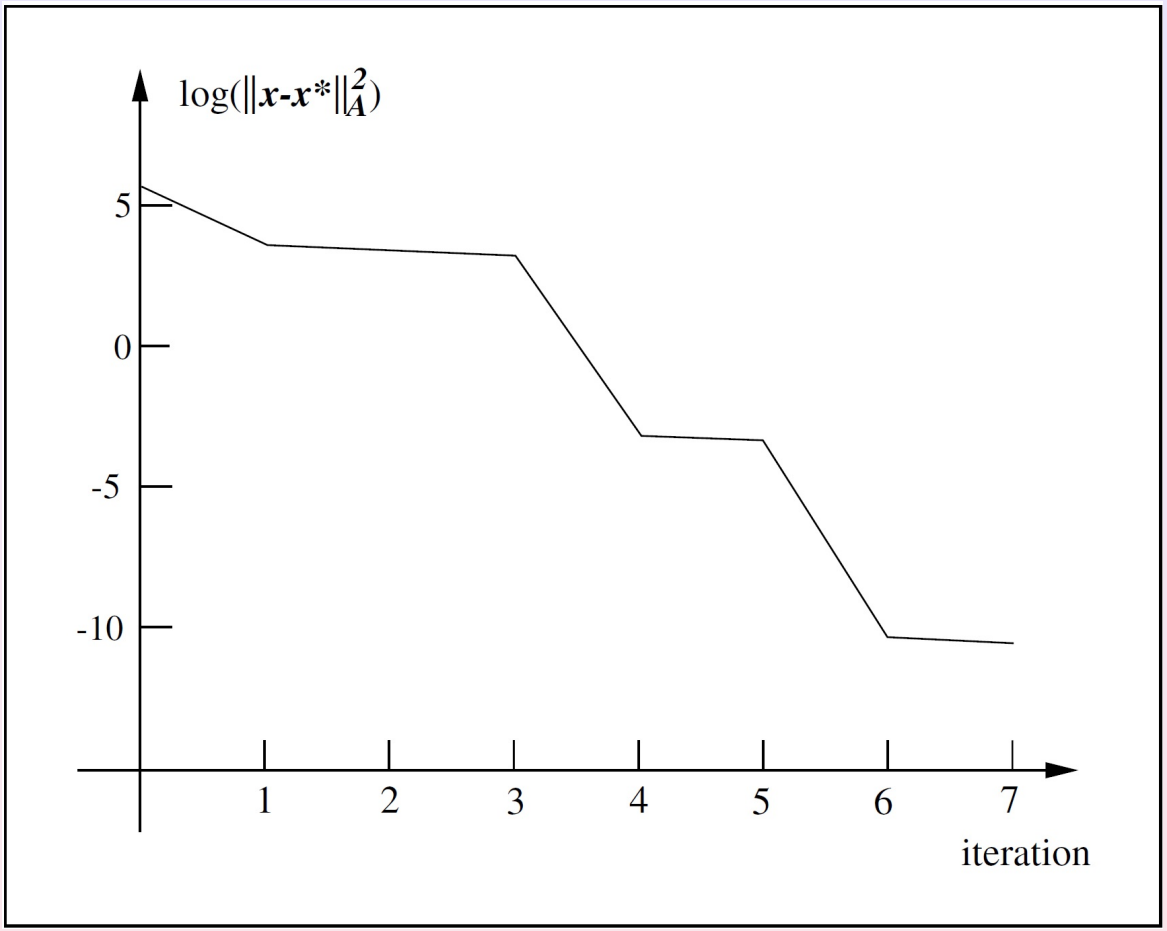

Figure 5: Performance of the conjugate gradient method on a matrix in which the eigenvalues occur in four distinct clusters.

§5.1 The Linear Conjugate Gradient Method

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

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

It can be shown that

$$\|x_k - x_*\|_A \leqslant 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x_*\|_A \,. \tag{36}$$

This bound often gives a large overestimate of the error, but it can be useful in cases where the only information we have about A is estimates of the extreme eigenvalues λ_1 and λ_n. This bound should be compared with that of the steepest descent method given by

$$\|x_{k+1} - x_*\|_Q^2 \leqslant \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \|x_k - x_*\|_Q^2$$

in Chapter 3, which is identical in form but which depends on the condition number $\kappa(A)$, and not on its square root $\sqrt{\kappa(A)}$.

§5.1 The Linear Conjugate Gradient Method

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

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

It can be shown that

$$\|x_k - x_*\|_A \leqslant 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x_*\|_A \,. \tag{36}$$

This bound often gives a large overestimate of the error, but it can be useful in cases where the only information we have about A is estimates of the extreme eigenvalues λ_1 and λ_n. This bound should be compared with that of the steepest descent method given by

$$\|x_{k+1} - x_*\|_Q^2 \leqslant \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \|x_k - x_*\|_Q^2$$

in Chapter 3, which is identical in form but which depends on the condition number $\kappa(A)$, and not on its square root $\sqrt{\kappa(A)}$.

§5.1 The Linear Conjugate Gradient Method

- **Preconditioning**

We can accelerate the conjugate gradient 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 non-singular matrix C; that is,

$$\hat{x} = Cx. \tag{37}$$

The quadratic φ defined by (2) is transformed accordingly to

$$\hat{\varphi}(\hat{x}) = \frac{1}{2}\hat{x}^{\mathrm{T}}(C^{-\mathrm{T}}AC^{-1})\hat{x} - (C^{-\mathrm{T}}b)^{\mathrm{T}}\hat{x}. \tag{38}$$

If we use Algorithm 5.2 to minimize $\hat{\varphi}$ or, equivalently, to solve the linear system $(C^{-\mathrm{T}}AC^{-1})\hat{x} = C^{-\mathrm{T}}b$, then the convergence rate will depend on the eigenvalues of the matrix $C^{-\mathrm{T}}AC^{-1}$ rather than those of A.

§5.1 The Linear Conjugate Gradient Method

Therefore, we aim to choose C such that the eigenvalues of $C^{-\mathrm{T}}AC^{-1}$ are more favorable for the convergence theory discussed above. We can try to choose C such that the condition number of $\widehat{A} \equiv C^{-\mathrm{T}}AC^{-1}$ is much smaller than the original condition number of A, for instance, so that the constant in

$$\|\widehat{x}_k - \widehat{x}_*\|_{\widehat{A}} \leqslant 2\left(\frac{\sqrt{\kappa(\widehat{A})}-1}{\sqrt{\kappa(\widehat{A})}+1}\right)^k \|\widehat{x}_0 - \widehat{x}_*\|_{\widehat{A}}\,. \tag{36'}$$

is smaller. We could also try to choose C such that the eigenvalues of $C^{-\mathrm{T}}AC^{-1}$ are clustered, which by the discussion of the previous section ensures that the number of iterates needed to find a good approximate solution is not much larger than the number of clusters.

§5.1 The Linear Conjugate Gradient Method

Therefore, we aim to choose C such that the eigenvalues of $C^{-\mathrm{T}}AC^{-1}$ are more favorable for the convergence theory discussed above. We can try to choose C such that the condition number of $\widehat{A} \equiv C^{-\mathrm{T}}AC^{-1}$ is much smaller than the original condition number of A, for instance, so that the constant in

$$\|\widehat{x}_k - \widehat{x}_*\|_{\widehat{A}} \leqslant 2\left(\frac{\sqrt{\kappa(\widehat{A})}-1}{\sqrt{\kappa(\widehat{A})}+1}\right)^k \|\widehat{x}_0 - \widehat{x}_*\|_{\widehat{A}}\,. \qquad (36')$$

is smaller. We could also try to choose C such that the eigenvalues of $C^{-\mathrm{T}}AC^{-1}$ are clustered, which by the discussion of the previous section ensures that the number of iterates needed to find a good approximate solution is not much larger than the number of clusters.

§5.1 The Linear Conjugate Gradient Method

It is not necessary to carry out the transformation (37) explicitly. Rather, we can apply Algorithm 5.2 to the problem

$$\widehat{\varphi}(\widehat{x}) = \frac{1}{2}\widehat{x}^{\mathrm{T}}(C^{-\mathrm{T}}AC^{-1})\widehat{x} - (C^{-\mathrm{T}}b)^{\mathrm{T}}\widehat{x} \tag{38}$$

in terms of the variables $\widehat{x}$, and then invert the transformations to re-express all the equations in terms of x. This process of derivation results in Algorithm 5.3 (Preconditioned Conjugate Gradient), which we now define. It happens that Algorithm 5.3 does not make use of C explicitly, but rather the matrix $M = C^{\mathrm{T}}C$, which is symmetric and positive definite by construction.

§5.1 The Linear Conjugate Gradient Method

It is not necessary to carry out the transformation (37) explicitly. Rather, we can apply Algorithm 5.2 to the problem

$$\widehat{\varphi}(\widehat{x}) = \frac{1}{2}\widehat{x}^{\mathrm{T}}(C^{-\mathrm{T}}AC^{-1})\widehat{x} - (C^{-\mathrm{T}}b)^{\mathrm{T}}\widehat{x} \tag{38}$$

in terms of the variables $\widehat{x}$, and then invert the transformations to re-express all the equations in terms of x. This process of derivation results in Algorithm 5.3 (Preconditioned Conjugate Gradient), which we now define. It happens that Algorithm 5.3 does not make use of C explicitly, but rather the matrix $M = C^{\mathrm{T}}C$, which is symmetric and positive definite by construction.

§5.1 The Linear Conjugate Gradient Method

Algorithm 5.3 (Preconditioned CG)

Given x_0, preconditioner M;

Set $r_0 \leftarrow Ax_0 - b$, solve $My_0 = r_0$ for y_0;

Set $p_0 = -y_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^{\mathrm{T}} y_k}{p_k^{\mathrm{T}} A p_k}; \tag{39a}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{39b}$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \tag{39c}$$

$$\text{Solve } My_{k+1} = r_{k+1}; \tag{39d}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^{\mathrm{T}} y_{k+1}}{r_k^{\mathrm{T}} y_k}; \tag{39e}$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k; \tag{39f}$$

$$k \leftarrow k+1; \tag{39g}$$

end (while)

§5.1 The Linear Conjugate Gradient Method

If we set $M = \mathrm{I}$ in Algorithm 5.3, we recover the standard CG method, Algorithm 5.2. The properties of Algorithm 5.2 generalize to this case in interesting ways. In particular, the orthogonality property (17) of the successive residuals becomes

$$r_i^{\mathrm{T}} M^{-1} r_j = 0 \quad \forall\, i \neq j. \tag{40}$$

In terms of computational effort, the main difference between the preconditioned and unpreconditioned CG methods is the need to solve systems of the form $My = r$ (step (39d)).

§5.1 The Linear Conjugate Gradient Method

- **Practical Preconditioners**

No single preconditioning strategy is "best" for all conceivable types of matrices: The tradeoff between various objectives – effectiveness of M, inexpensive computation and storage of M, inexpensive solution of $My = r$ – varies from problem to problem. Often, the preconditioner is defined in such a way that the system $My = r$ amounts to a simplified version of the original system $Ax = b$. As in many other areas of optimization and numerical analysis, knowledge about the structure and origin of a problem is the key to devising effective techniques for solving the problem.

§5.1 The Linear Conjugate Gradient Method

- **Practical Preconditioners**

No single preconditioning strategy is "best" for all conceivable types of matrices: The tradeoff between various objectives – effectiveness of M, inexpensive computation and storage of M, inexpensive solution of $My = r$ – varies from problem to problem. Often, the preconditioner is defined in such a way that the system $My = r$ amounts to a simplified version of the original system $Ax = b$. As in many other areas of optimization and numerical analysis, knowledge about the structure and origin of a problem is the key to devising effective techniques for solving the problem.

§5.1 The Linear Conjugate Gradient Method

- **Practical Preconditioners**

No single preconditioning strategy is "best" for all conceivable types of matrices: The tradeoff between various objectives – effectiveness of M, inexpensive computation and storage of M, inexpensive solution of $My = r$ – varies from problem to problem. Often, the preconditioner is defined in such a way that the system $My = r$ amounts to a simplified version of the original system $Ax = b$. As in many other areas of optimization and numerical analysis, knowledge about the structure and origin of a problem is the key to devising effective techniques for solving the problem.

§5.1 The Linear Conjugate Gradient Method

General-purpose preconditioners have also been proposed, but their success varies greatly from problem to problem. The most important strategies of this type include symmetric successive over-relaxation (SSOR), incomplete Cholesky, and banded preconditioners (see [272], [136], and [72] for discussions of these techniques). Incomplete Cholesky is probably the most effective in general. The basic idea is simple: We follow the Cholesky procedure, but instead of computing the exact Cholesky factor L that satisfies $A = LL^{\mathrm{T}}$, we compute an approximate factor $\bar{L}$ that is sparser than L (Usually, we require $\bar{L}$ to be no denser, or not much denser, than the lower triangle of the original matrix A).

§5.1 The Linear Conjugate Gradient Method

General-purpose preconditioners have also been proposed, but their success varies greatly from problem to problem. The most important strategies of this type include symmetric successive over-relaxation (SSOR), incomplete Cholesky, and banded preconditioners (see [272], [136], and [72] for discussions of these techniques). Incomplete Cholesky is probably the most effective in general. The basic idea is simple: We follow the Cholesky procedure, but instead of computing the exact Cholesky factor L that satisfies $A = LL^{\mathrm{T}}$, we compute an approximate factor $\bar{L}$ that is sparser than L (Usually, we require $\bar{L}$ to be no denser, or not much denser, than the lower triangle of the original matrix A).

§5.1 The Linear Conjugate Gradient Method

We then have $A \approx \bar{L}\bar{L}^{\mathrm{T}}$, and by choosing $C = \bar{L}^{\mathrm{T}}$, we obtain $M = \bar{L}\bar{L}^{\mathrm{T}}$ and $C^{-\mathrm{T}}AC^{-1} = \bar{L}^{-1}A\bar{L}^{-\mathrm{T}} \approx \mathrm{I}$, so the eigenvalue distribution of $C^{-\mathrm{T}}AC^{-1}$ is favorable. We do not compute M explicitly, but rather store the factor $\bar{L}$ and solve the system $My = r$ by performing two triangular substitutions with $\bar{L}$. Because the sparsity of $\bar{L}$ is similar to that of A, the cost of solving $My = r$ is similar to the cost of computing the matrix-vector product Ap.

§5.1 The Linear Conjugate Gradient Method

There are several possible pitfalls in the incomplete Cholesky approach. One is that the resulting matrix may not be (sufficiently) positive definite, and in this case one may need to increase the values of the diagonal elements to ensure that a value for $\bar{L}$ can be found. Numerical instability or breakdown can occur during the incomplete factorization because of the sparsity conditions we impose on the factor $\bar{L}$. This difficulty can be remedied by allowing additional fill-in in $\bar{L}$, but the denser factor will be more expensive to compute and to apply at each iteration.

§5.2 Nonlinear Conjugate Gradient Methods

We have noted that the CG method, Algorithm 5.2, can be viewed as a minimization algorithm for the convex quadratic function φ defined by

$$\varphi(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x. \tag{2}$$

It is natural to ask whether we can adapt the approach to minimize general convex functions, or even general nonlinear functions f. In fact, as we show in this section, nonlinear variants of the conjugate gradient are well studied and have proved to be quite successful in practice.

§5.2 Nonlinear Conjugate Gradient Methods

- **The Fletcher-Reeves Method**

Fletcher and Reeves [107] showed how to extend the conjugate gradient method to nonlinear functions by making two simple changes in Algorithm 5.2. First, in place of the formula

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k} \tag{24a}$$

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. Second, the residual r, which is simply the gradient of φ in Algorithm 5.2, must be replaced by the gradient of the nonlinear objective f. These changes give rise to the following algorithm for nonlinear optimization.

§5.2 Nonlinear Conjugate Gradient Methods

- **The Fletcher-Reeves Method**

Fletcher and Reeves [107] showed how to extend the conjugate gradient method to nonlinear functions by making two simple changes in Algorithm 5.2. First, in place of the formula

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k} \tag{24a}$$

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. Second, the residual r, which is simply the gradient of φ in Algorithm 5.2, must be replaced by the gradient of the nonlinear objective f. These changes give rise to the following algorithm for nonlinear optimization.

§5.2 Nonlinear Conjugate Gradient Methods

Algorithm 5.4 (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 $\nabla f_k \neq 0$

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

Evaluate ∇f_{k+1};

$$\beta_{k+1}^{\mathrm{FR}} \leftarrow \frac{\nabla f_{k+1}^{\mathrm{T}} \nabla f_{k+1}}{\nabla f_k^{\mathrm{T}} \nabla f_k}; \tag{41a}$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{FR}} p_k; \tag{41b}$$

$$k \leftarrow k+1; \tag{41c}$$

end (while)

§5.2 Nonlinear Conjugate Gradient Methods

If we choose f to be a strongly convex quadratic and α_k to be the exact minimizer, this algorithm reduces to the linear conjugate gradient method, Algorithm 5.2. Algorithm 5.4 is appealing for large nonlinear optimization problems because each iteration requires only evaluation of the objective function and its gradient. No matrix operations are required for the step computation, and just a few vectors of storage are required.

To make the specification of Algorithm 5.4 complete, we need to be more precise about the choice of line search parameter α_k. Because of the second term in (41b), the search direction p_k may fail to be a descent direction unless α_k satisfies certain conditions.

§5.2 Nonlinear Conjugate Gradient Methods

If we choose f to be a strongly convex quadratic and α_k to be the exact minimizer, this algorithm reduces to the linear conjugate gradient method, Algorithm 5.2. Algorithm 5.4 is appealing for large nonlinear optimization problems because each iteration requires only evaluation of the objective function and its gradient. No matrix operations are required for the step computation, and just a few vectors of storage are required.

To make the specification of Algorithm 5.4 complete, we need to be more precise about the choice of line search parameter α_k. Because of the second term in (41b), the search direction p_k may fail to be a descent direction unless α_k satisfies certain conditions.

§5.2 Nonlinear Conjugate Gradient Methods

By taking the inner product of (41b) (with k replacing $k+1$) with the gradient vector ∇f_k, we obtain

$$\nabla f_k^{\mathrm{T}} p_k = -\|\nabla f_k\|^2 + \beta_k^{\mathrm{FR}} \nabla f_k^{\mathrm{T}} p_{k-1}\,. \tag{42}$$

If the line search is exact, so that x_k is a local minimizer of f along the direction p_{k-1}, we have that $\nabla f_k^{\mathrm{T}} p_{k-1} = 0$. In this case we have from (42) that $\nabla f_k^{\mathrm{T}} p_k < 0$, so that p_k is indeed a descent direction. If the line search is not exact, however, the second term in (42) may dominate the first term, and we may have $\nabla f_k^{\mathrm{T}} p_k > 0$, implying that p_k is actually a direction of ascent.

§5.2 Nonlinear Conjugate Gradient Methods

By taking the inner product of (41b) (with k replacing $k+1$) with the gradient vector ∇f_k, we obtain

$$\nabla f_k^{\mathrm{T}} p_k = -\|\nabla f_k\|^2 + \beta_k^{\mathrm{FR}} \nabla f_k^{\mathrm{T}} p_{k-1}\,. \tag{42}$$

If the line search is exact, so that x_k is a local minimizer of f along the direction p_{k-1}, we have that $\nabla f_k^{\mathrm{T}} p_{k-1} = 0$. In this case we have from (42) that $\nabla f_k^{\mathrm{T}} p_k < 0$, so that p_k is indeed a descent direction.

If the line search is not exact, however, the second term in (42) may dominate the first term, and we may have $\nabla f_k^{\mathrm{T}} p_k > 0$, implying that p_k is actually a direction of ascent.

§5.2 Nonlinear Conjugate Gradient Methods

By taking the inner product of (41b) (with k replacing $k+1$) with the gradient vector ∇f_k, we obtain

$$\nabla f_k^{\mathrm{T}} p_k = -\|\nabla f_k\|^2 + \beta_k^{\mathrm{FR}} \nabla f_k^{\mathrm{T}} p_{k-1}\,. \tag{42}$$

If the line search is exact, so that x_k is a local minimizer of f along the direction p_{k-1}, we have that $\nabla f_k^{\mathrm{T}} p_{k-1} = 0$. In this case we have from (42) that $\nabla f_k^{\mathrm{T}} p_k < 0$, so that p_k is indeed a descent direction. If the line search is not exact, however, the second term in (42) may dominate the first term, and we may have $\nabla f_k^{\mathrm{T}} p_k > 0$, implying that p_k is actually a direction of ascent.

§5.2 Nonlinear Conjugate Gradient Methods

Fortunately, we can avoid this situation by requiring the step length α_k to satisfy the **strong** Wolfe conditions, which we restate here:

$$f(x_k + \alpha_k p_k) \leqslant f(x_k) + c_1 \alpha_k \nabla f_k^{\mathrm{T}} p_k \,, \tag{43a}$$

$$\left|\nabla f(x_k + \alpha_k p_k)^{\mathrm{T}} p_k\right| \leqslant -c_2 \nabla f_k^{\mathrm{T}} p_k \,, \tag{43b}$$

where $0 < c_1 < c_2 < \frac{1}{2}$. Note that we impose $c_2 < \frac{1}{2}$ here, in place of the looser condition $c_2 < 1$ that was used in the earlier statement. By applying the lemma stated two slides later, we can show that condition (43b) implies that $\nabla f_k^{\mathrm{T}} p_k < 0$ (so that p_k is a descent direction), and we conclude that any line search procedure that yields an α_k satisfying (43) will ensure that all directions p_k are descent directions for the function f.

§5.2 Nonlinear Conjugate Gradient Methods

Fortunately, we can avoid this situation by requiring the step length α_k to satisfy the **strong** Wolfe conditions, which we restate here:

$$f(x_k + \alpha_k p_k) \leqslant f(x_k) + c_1 \alpha_k \nabla f_k^{\mathrm{T}} p_k \,, \tag{43a}$$

$$\left|\nabla f(x_k + \alpha_k p_k)^{\mathrm{T}} p_k\right| \leqslant -c_2 \nabla f_k^{\mathrm{T}} p_k \,, \tag{43b}$$

where $0 < c_1 < c_2 < \frac{1}{2}$. Note that we impose $c_2 < \frac{1}{2}$ here, in place of the looser condition $c_2 < 1$ that was used in the earlier statement. By applying the lemma stated two slides later, we can show that condition (43b) implies that $\nabla f_k^{\mathrm{T}} p_k < 0$ (so that p_k is a descent direction), and we conclude that any line search procedure that yields an α_k satisfying (43) will ensure that all directions p_k are descent directions for the function f.

§5.2 Nonlinear Conjugate Gradient Methods

In fact, for algorithms similar to Algorithm 5.4 but updating p_k by

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k; \tag{41b'}$$

with some sequence β_k satisfying $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$, we will always have $\nabla f_k^{\mathrm{T}} p_k < 0$ for all $k \in \mathbb{N}$.

Algorithm 5.4'

Given x_0, evaluate $f_0 = f(x_0))$, $\nabla f_0 = (\nabla f)(x_0))$;
Set $p_0 = -\nabla f_0$, $k \leftarrow 0$;
while $\nabla f_k \neq 0$
  Compute α_k and set $x_{k+1} = x_k + \alpha_k p_k$;
  Evaluate ∇f_{k+1};
  Update β_{k+1} (using $\nabla f_{k+1}, \nabla f_k$) with $|\beta_{k+1}| \leqslant \beta_{k+1}^{\mathrm{FR}}$; (41a')
  $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k$; (41b')
  $k \leftarrow k+1$; (41c)
end (while)

§5.2 Nonlinear Conjugate Gradient Methods

In fact, for algorithms similar to Algorithm 5.4 but updating p_k by

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k; \qquad (41b')$$

with some sequence β_k satisfying $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$, we will always have $\nabla f_k^{\mathrm{T}} p_k < 0$ for all $k \in \mathbb{N}$.

Algorithm 5.4'

Given x_0, evaluate $f_0 = f(x_0))$, $\nabla f_0 = (\nabla f)(x_0))$;
Set $p_0 = -\nabla f_0$, $k \leftarrow 0$;
while $\nabla f_k \neq 0$
    Compute α_k and set $x_{k+1} = x_k + \alpha_k p_k$;
    Evaluate ∇f_{k+1};
    Update β_{k+1} (using $\nabla f_{k+1}, \nabla f_k$) with $|\beta_{k+1}| \leqslant \beta_{k+1}^{\mathrm{FR}}$; $(41a')$
    $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k;$ $(41b')$
    $k \leftarrow k+1;$ $(41c)$
end (while)

§5.2 Nonlinear Conjugate Gradient Methods

Lemma

Suppose that Algorithm 5.4' is implemented with a step length α_k satisfying the strong Wolfe conditions (43b) with $0 < c_2 < 1/2$ and β_k satisfying $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$. Then the method generates descent directions p_k that satisfy the following inequalities:

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2} \leqslant \frac{2c_2-1}{1-c_2} \quad \text{for all } k \in \mathbb{N} \cup \{0\}. \tag{44}$$

Proof.

Note first that the function $t(\xi) = (2\xi - 1)/(1-\xi)$ is monotonically increasing on the interval $[0, 1/2]$. Since $c_2 \in (0, 1/2)$, we have

$$-1 = t(0) < \frac{2c_2-1}{1-c_2} < t(1/2) = 0\,. \tag{45}$$

The descent condition $\nabla f_k^{\mathrm{T}} p_k < 0$ holds once we establish (44). □

§5.2 Nonlinear Conjugate Gradient Methods

Lemma

Suppose that Algorithm 5.4' is implemented with a step length α_k satisfying the strong Wolfe conditions (43b) with $0 < c_2 < 1/2$ and β_k satisfying $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$. Then the method generates descent directions p_k that satisfy the following inequalities:

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2} \leqslant \frac{2c_2-1}{1-c_2} \quad \text{for all } k \in \mathbb{N} \cup \{0\}. \tag{44}$$

Proof.

Note first that the function $t(\xi) = (2\xi - 1)/(1-\xi)$ is monotonically increasing on the interval $[0, 1/2]$. Since $c_2 \in (0, 1/2)$, we have

$$-1 = t(0) < \frac{2c_2-1}{1-c_2} < t(1/2) = 0\,. \tag{45}$$

The descent condition $\nabla f_k^{\mathrm{T}} p_k < 0$ holds once we establish (44). □

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

The proof is by induction. For $k = 0$, the middle term in (44) is -1, so by using (45), we see that both inequalities in (44) are satisfied.

Next, assume that (44) holds for some $k \geqslant 1$. From

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k; \tag{41b'}$$

we have

$$\frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1} \frac{\nabla f_{k+1}^{\mathrm{T}} p_k}{\|\nabla f_{k+1}\|^2}. \tag{46}$$

Using the condition $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$ and the line search condition (43b), (46) shows that

$$\begin{aligned} \left| \frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} + 1 \right| &= |\beta_{k+1}| \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2} \leqslant \beta_{k+1}^{\mathrm{FR}} \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2} \\ &= \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_k\|^2} \leqslant -c_2 \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2}. \end{aligned}$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

The proof is by induction. For $k = 0$, the middle term in (44) is -1, so by using (45), we see that both inequalities in (44) are satisfied. Next, assume that (44) holds for some $k \geqslant 1$. From

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k; \tag{41b'}$$

we have

$$\frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1} \frac{\nabla f_{k+1}^{\mathrm{T}} p_k}{\|\nabla f_{k+1}\|^2}. \tag{46}$$

Using the condition $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$ and the line search condition (43b), (46) shows that

$$\left| \frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} + 1 \right| = |\beta_{k+1}| \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2} \leqslant \beta_{k+1}^{\mathrm{FR}} \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2}$$

$$= \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_k\|^2} \leqslant -c_2 \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2}. \qquad \square$$

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

The proof is by induction. For $k = 0$, the middle term in (44) is -1, so by using (45), we see that both inequalities in (44) are satisfied. Next, assume that (44) holds for some $k \geqslant 1$. From

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k; \tag{41b'}$$

we have

$$\frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} = -1 + \beta_{k+1} \frac{\nabla f_{k+1}^{\mathrm{T}} p_k}{\|\nabla f_{k+1}\|^2}. \tag{46}$$

Using the condition $|\beta_k| \leqslant \beta_k^{\mathrm{FR}}$ for all $k \in \mathbb{N}$ and the line search condition (43b), (46) shows that

$$\begin{aligned} \left| \frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} + 1 \right| &= |\beta_{k+1}| \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2} \leqslant \beta_{k+1}^{\mathrm{FR}} \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_{k+1}\|^2} \\ &= \frac{|\nabla f_{k+1}^{\mathrm{T}} p_k|}{\|\nabla f_k\|^2} \leqslant -c_2 \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2}. \end{aligned}$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Substituting for the term $\dfrac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2}$ from the left-hand-side of the induction hypothesis

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2} \leqslant \frac{2c_2-1}{1-c_2}, \tag{44}$$

we obtain

$$\left| \frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} + 1 \right| \leqslant \frac{c_2}{1-c_2},$$

which shows that

$$-1 - \frac{c_2}{1-c_2} \leqslant \frac{\nabla f_{k+1}^{\mathrm{T}} p_{k+1}}{\|\nabla f_{k+1}\|^2} \leqslant -1 + \frac{c_2}{1-c_2}.$$

Therefore, (44) holds for $k+1$ as well. □

§5.2 Nonlinear Conjugate Gradient Methods

This result used only the second strong Wolfe condition (43b); the first Wolfe condition (43a) will be needed in the next section to establish global convergence. The bounds on

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2} \leqslant \frac{2c_2-1}{1-c_2} \quad \text{for all } k \in \mathbb{N} \cup \{0\}. \tag{44}$$

impose a limit on how fast the norms of the steps $\|p_k\|$ can grow, and they will play a crucial role in the convergence analysis given below.

§5.2 Nonlinear Conjugate Gradient Methods

The established lemma can also 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. As in Chapter 3, we let θ_k denote the angle between p_k and the steepest descent direction $-\nabla f_k$, defined by

$$\cos\theta_k = -\frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|\|p_k\|}. \tag{47}$$

Suppose that p_k is a poor search direction, in the sense that it makes an angle of nearly 90° with $-\nabla f_k$; that is, $\cos\theta_k \approx 0$. By multiplying both sides of (44) by $\|\nabla f_k\|/\|p_k\|$ and using (47), we obtain

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k = 0, 1, \cdots. \tag{48}$$

§5.2 Nonlinear Conjugate Gradient Methods

The established lemma can also 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. As in Chapter 3, we let θ_k denote the angle between p_k and the steepest descent direction $-\nabla f_k$, defined by

$$\cos\theta_k = -\frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|\|p_k\|}. \tag{47}$$

Suppose that p_k is a poor search direction, in the sense that it makes an angle of nearly 90° with $-\nabla f_k$; that is, $\cos\theta_k \approx 0$. By multiplying both sides of (44) by $\|\nabla f_k\|/\|p_k\|$ and using (47), we obtain

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k = 0, 1, \cdots. \tag{48}$$

§5.2 Nonlinear Conjugate Gradient Methods

From these inequalities, we deduce that

$$\cos\theta_k \approx 0 \quad \text{if and only if} \quad \|\nabla f_k\| \ll \|p_k\|.$$

Since p_k is almost orthogonal to the gradient, it is likely that the step from x_k to $x_{k+1}(= x_k + \alpha_k p_k)$ is tiny; that is, $x_{k+1} \approx x_k$. If so, we have $\nabla f_{k+1} \approx \nabla f_k$, and therefore by Definition (41a),

$$\beta_{k+1}^{\text{FR}} \equiv \frac{\nabla f_{k+1}^{\mathrm{T}} \nabla f_{k+1}}{\nabla f_k^{\mathrm{T}} \nabla f_k} \approx 1\,. \tag{49}$$

Using this approximation together with $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$ in

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\text{FR}} p_k\,, \tag{41b}$$

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.

§5.2 Nonlinear Conjugate Gradient Methods

From these inequalities, we deduce that

$$\cos\theta_k \approx 0 \quad \text{if and only if} \quad \|\nabla f_k\| \ll \|p_k\|.$$

Since p_k is almost orthogonal to the gradient, it is likely that the step from x_k to $x_{k+1}(= x_k + \alpha_k p_k)$ is tiny; that is, $x_{k+1} \approx x_k$. If so, we have $\nabla f_{k+1} \approx \nabla f_k$, and therefore by Definition (41a),

$$\beta_{k+1}^{\text{FR}} \equiv \frac{\nabla f_{k+1}^{\mathrm{T}} \nabla f_{k+1}}{\nabla f_k^{\mathrm{T}} \nabla f_k} \approx 1\,. \tag{49}$$

Using this approximation together with $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$ in

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\text{FR}} p_k\,, \tag{41b}$$

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.

§5.2 Nonlinear Conjugate Gradient Methods

- **The Polak-Ribière Method and Variants**

There are many variants of the Fletcher-Reeves method that differ from each other mainly in the choice of the parameter β_k. An important variant, proposed by Polak and Ribière, defines this parameter as follows:

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}. \tag{50}$$

We refer to the algorithm in which (50) replaces (41a) as Algorithm PR. It is identical to Algorithm FR when f is a strongly convex quadratic function and the line search is exact, since by (17) the gradients are mutually orthogonal, and so $\beta_{k+1}^{\mathrm{PR}} = \beta_{k+1}^{\mathrm{FR}}$. When applied to general nonlinear functions with inexact line searches, however, the behavior of the two algorithms differs markedly.

§5.2 Nonlinear Conjugate Gradient Methods

- **The Polak-Ribière Method and Variants**

There are many variants of the Fletcher-Reeves method that differ from each other mainly in the choice of the parameter β_k. An important variant, proposed by Polak and Ribière, defines this parameter as follows:

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}. \tag{50}$$

We refer to the algorithm in which (50) replaces (41a) as Algorithm PR. It is identical to Algorithm FR when f is a strongly convex quadratic function and the line search is exact, since by (17) the gradients are mutually orthogonal, and so $\beta_{k+1}^{\mathrm{PR}} = \beta_{k+1}^{\mathrm{FR}}$. When applied to general nonlinear functions with inexact line searches, however, the behavior of the two algorithms differs markedly.

§5.2 Nonlinear Conjugate Gradient Methods

The Polak-Ribière method behaves quite differently from the Fletcher-Reeves one. If, as in the previous case, 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

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \tag{50}$$

that $\beta_{k+1}^{\mathrm{PR}} \approx 0$. From the formula

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{PR}} p_k\,, \tag{41b'}$$

we find that 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 essentially performs a **restart** after it encounters a bad direction.

§5.2 Nonlinear Conjugate Gradient Methods

The Polak-Ribière method behaves quite differently from the Fletcher-Reeves one. If, as in the previous case, 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

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \tag{50}$$

that $\beta_{k+1}^{\mathrm{PR}} \approx 0$. From the formula

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{PR}} p_k\,, \tag{41b'}$$

we find that 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 essentially performs a **restart** after it encounters a bad direction.

§5.2 Nonlinear Conjugate Gradient Methods

The Polak-Ribière method behaves quite differently from the Fletcher-Reeves one. If, as in the previous case, 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

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \tag{50}$$

that $\beta_{k+1}^{\mathrm{PR}} \approx 0$. From the formula

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{PR}} p_k\,, \tag{41b'}$$

we find that 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 essentially performs a **restart** after it encounters a bad direction.

§5.2 Nonlinear Conjugate Gradient Methods

The undesirable behavior of the Fletcher-Reeves method predicted by the arguments given above can be observed in practice. For example, the paper [123] describes a problem with $n = 100$ in which $\cos\theta_k$ is of order 10^{-2} for hundreds of iterations and the steps $\|x_k - x_{k-1}\|$ are of order 10^{-2}. Algorithm FR requires thousands of iterations to solve this problem, while Algorithm PR requires just 37 iterations. In this example, the Fletcher-Reeves method performs much better if it is periodically restarted along the steepest descent direction, since each restart terminates the cycle of bad steps. In general, Algorithm FR should not be implemented without some kind of restart strategy.

§5.2 Nonlinear Conjugate Gradient Methods

Numerical experience also indicates that Algorithm PR tends to be the more robust and efficient of the two. A surprising fact about Algorithm PR is that the strong Wolfe conditions (43) do not guarantee that p_k is always a descent direction. If we define the β parameter as

$$\beta_{k+1}^{+} = \max\left\{\beta_{k+1}^{\mathrm{PR}}, 0\right\}, \tag{51}$$

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

§5.2 Nonlinear Conjugate Gradient Methods

Numerical experience also indicates that Algorithm PR tends to be the more robust and efficient of the two. A surprising fact about Algorithm PR is that the strong Wolfe conditions (43) do not guarantee that p_k is always a descent direction. If we define the β parameter as

$$\beta_{k+1}^{+} = \max\left\{\beta_{k+1}^{\mathrm{PR}}, 0\right\}, \tag{51}$$

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

§5.2 Nonlinear Conjugate Gradient Methods

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

$$\beta_{k+1}^{\mathrm{HS}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^{\mathrm{T}} p_k}, \tag{52}$$

gives rise to an algorithm (called Algorithm HS) that is similar to Algorithm PR, both in terms of its theoretical convergence properties and in its practical performance. Formula (52) can be derived by demanding that consecutive search directions be conjugate with respect to the average Hessian over the line segment $[x_k, x_{k+1}]$, which is defined as

$$\bar{G}_k = \int_0^1 \left[(\nabla^2 f)(x_k + \tau \alpha_k p_k)\right] d\tau .$$

§5.2 Nonlinear Conjugate Gradient Methods

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

$$\beta_{k+1}^{\mathrm{HS}} = \frac{\nabla f_{k+1}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^{\mathrm{T}} p_k}, \tag{52}$$

gives rise to an algorithm (called Algorithm HS) that is similar to Algorithm PR, both in terms of its theoretical convergence properties and in its practical performance. Formula (52) can be derived by demanding that consecutive search directions be conjugate with respect to the average Hessian over the line segment $[x_k, x_{k+1}]$, which is defined as

$$\bar{G}_k = \int_0^1 \left[(\nabla^2 f)(x_k + \tau \alpha_k p_k)\right] d\tau .$$

§5.2 Nonlinear Conjugate Gradient Methods

Recalling from Taylor's theorem that $\nabla f_{k+1} = \nabla f_k + \alpha_k \bar{G}_k p_k$, we see that for any direction of the form $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$, the condition $p_{k+1}^{\mathrm{T}} \bar{G}_k p_k = 0$ requires β_{k+1} to be given by (52): since $\alpha_k \neq 0$ (for otherwise p_k is not a descent direction),

$$
\begin{aligned}
0 = p_{k+1}^{\mathrm{T}} \bar{G}_k p_k &= \left[-\nabla f_{k+1} + \beta_{k+1} p_k \right]^{\mathrm{T}} \alpha_k^{-1} (\nabla f_{k+1} - \nabla f_k) \\
&\Rightarrow \beta_{k+1} (\nabla f_{k+1} - \nabla f_k)^{\mathrm{T}} p_k = \nabla f_{k+1}^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k) \\
&\Rightarrow \beta_{k+1} = \frac{\nabla f_{k+1}^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^{\mathrm{T}} p_k} \equiv \beta_{k+1}^{\mathrm{HS}} .
\end{aligned}
$$

§5.2 Nonlinear Conjugate Gradient Methods

Later, we see that it is possible to guarantee **global convergence** for any parameter β_k satisfying the bound

$$|\beta_k| \leqslant \beta_k^{\mathrm{FR}}, \tag{53}$$

for all $k \geqslant 2$. This fact suggests the following modification of the PR method, which has performed well on some applications. For all $k \geqslant 2$ let

$$\beta_k = \begin{cases} -\beta_k^{\mathrm{FR}} & \text{if } \beta_k^{\mathrm{PR}} < -\beta_k^{\mathrm{FR}}, \\ \beta_k^{\mathrm{PR}} & \text{if } |\beta_k^{\mathrm{PR}}| \leqslant \beta_k^{\mathrm{FR}}, \\ \beta_k^{\mathrm{FR}} & \text{if } \beta_k^{\mathrm{PR}} > \beta_k^{\mathrm{FR}}. \end{cases} \tag{54}$$

The algorithm based on this strategy will be denoted by FR-PR.

§5.2 Nonlinear Conjugate Gradient Methods

Other variants of the CG method have recently been proposed. Two choices for β_{k+1} that possess attractive theoretical and computational properties are

$$\beta_{k+1} = \frac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^{\mathrm{T}} p_k} \tag{55}$$

and

$$\beta_{k+1} = \left(\hat{y}_k - 2p_k \frac{\|\hat{y}_k\|^2}{\hat{y}_k^{\mathrm{T}} p_k}\right)^{\mathrm{T}} \frac{\nabla f_{k+1}}{\hat{y}_k^{\mathrm{T}} p_k} \quad \text{with} \quad \hat{y}_k = \nabla f_{k+1} - \nabla f_k\,. \tag{56}$$

These two choices guarantee that p_k is a descent direction, provided the step length α_k satisfies the Wolfe conditions. The CG algorithms based on (55) or (56) appear to be competitive with the Polak-Ribière method, and the global convergence can be established without introducing any modification to a line search based on the Wolfe conditions.

§5.2 Nonlinear Conjugate Gradient Methods

- **Quadratic Termination and Restarts**

Implementations of nonlinear conjugate gradient methods usually preserve their close connections with the linear conjugate gradient method. Usually, a quadratic (or cubic) interpolation along the search direction p_k is incorporated into the line search procedure; see Chapter 3. This feature guarantees that when f is a strictly convex quadratic, the step length α_k is chosen to be the exact one-dimensional minimizer, so that the nonlinear conjugate gradient method reduces to the linear method, Algorithm 5.2.

§5.2 Nonlinear Conjugate Gradient Methods

Another modification that is often used in nonlinear conjugate gradient procedures is to **restart** the iteration at **every** n **steps** by setting $\beta_k = 0$ in (41a); 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,

$$\|x_{k+n} - x_*\| = \mathcal{O}(\|x_k - x_*\|^2)\,. \tag{57}$$

§5.2 Nonlinear Conjugate Gradient Methods

After a little thought, this result is not so surprising. Consider a function f that is strongly convex quadratic in a neighborhood of the solution, but is non-quadratic everywhere else. Assuming that the algorithm is converging to the solution in question, the iterates will eventually enter the quadratic region. At some point, the algorithm will be restarted in that region, and from that point onward, its behavior will simply be that of the linear conjugate gradient method, Algorithm 5.2. In particular, finite termination will occur within n steps of the restart. The restart is important, because the finite-termination property and other appealing properties of Algorithm 5.2 hold only when its initial search direction p_0 is equal to the negative gradient.

§5.2 Nonlinear Conjugate Gradient Methods

After a little thought, this result is not so surprising. Consider a function f that is strongly convex quadratic in a neighborhood of the solution, but is non-quadratic everywhere else. Assuming that the algorithm is converging to the solution in question, the iterates will eventually enter the quadratic region. At some point, the algorithm will be restarted in that region, and from that point onward, its behavior will simply be that of the linear conjugate gradient method, Algorithm 5.2. In particular, finite termination will occur within n steps of the restart. The restart is important, because the finite-termination property and other appealing properties of Algorithm 5.2 hold only when its initial search direction p_0 is equal to the negative gradient.

§5.2 Nonlinear Conjugate Gradient Methods

After a little thought, this result is not so surprising. Consider a function f that is strongly convex quadratic in a neighborhood of the solution, but is non-quadratic everywhere else. Assuming that the algorithm is converging to the solution in question, the iterates will eventually enter the quadratic region. At some point, the algorithm will be restarted in that region, and from that point onward, its behavior will simply be that of the linear conjugate gradient method, Algorithm 5.2. In particular, finite termination will occur within n steps of the restart. The restart is important, because the finite-termination property and other appealing properties of Algorithm 5.2 hold only when its initial search direction p_0 is equal to the negative gradient.

§5.2 Nonlinear Conjugate Gradient Methods

Even if the function f is not exactly quadratic in the region of a solution, Taylor's theorem implies that it can still be approximated quite closely by a quadratic, provided that it is smooth. Therefore, while we would not expect termination in n steps after the restart, it is not surprising that substantial progress is made toward the solution, as indicated by the expression

$$\|x_{k+n} - x_*\| = \mathcal{O}(\|x_k - x_*\|^2)\,. \tag{57}$$

Though the result (57) 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. Restarts may never occur in such problems because an approximate solution may be located in fewer than n steps.

§5.2 Nonlinear Conjugate Gradient Methods

Even if the function f is not exactly quadratic in the region of a solution, Taylor's theorem implies that it can still be approximated quite closely by a quadratic, provided that it is smooth. Therefore, while we would not expect termination in n steps after the restart, it is not surprising that substantial progress is made toward the solution, as indicated by the expression

$$\|x_{k+n} - x_*\| = \mathcal{O}(\|x_k - x_*\|^2)\,. \tag{57}$$

Though the result (57) 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. Restarts may never occur in such problems because an approximate solution may be located in fewer than n steps.

§5.2 Nonlinear Conjugate Gradient Methods

Hence, nonlinear CG method are sometimes implemented without restarts, or else they include strategies for restarting that are based on considerations other than iteration counts. The most popular restart strategy makes use of the observation

$$r_k^{\mathrm{T}} r_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1, \tag{17}$$

which is that 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

$$\frac{|\nabla f_k^{\mathrm{T}} \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geqslant \nu\,, \tag{58}$$

where a typical value for the parameter ν is 0.1.

§5.2 Nonlinear Conjugate Gradient Methods

Hence, nonlinear CG method are sometimes implemented without restarts, or else they include strategies for restarting that are based on considerations other than iteration counts. The most popular restart strategy makes use of the observation

$$r_k^{\mathrm{T}} r_i = 0 \quad \text{for} \quad i = 0, 1, \cdots, k-1, \tag{17}$$

which is that 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

$$\frac{|\nabla f_k^{\mathrm{T}} \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geqslant \nu\,, \tag{58}$$

where a typical value for the parameter ν is 0.1.

§5.2 Nonlinear Conjugate Gradient Methods

We could also think of formula

$$\beta_{k+1}^{+} = \max\left\{\beta_{k+1}^{\mathrm{PR}}, 0\right\} \tag{51}$$

as a restarting strategy, because p_{k+1} will revert to the steepest descent direction whenever β_k^{PR} is negative. In contrast to (58), these restarts are rather infrequent because β_k^{PR} is positive most of the time.

§5.2 Nonlinear Conjugate Gradient Methods

The same argument can be applied to Algorithms PR+ and HS. For the FR-PR variant, defined by

$$\beta_k = \begin{cases} -\beta_k^{\mathrm{FR}} & \text{if } \beta_k^{\mathrm{PR}} < -\beta_k^{\mathrm{FR}}, \\ \beta_k^{\mathrm{PR}} & \text{if } |\beta_k^{\mathrm{PR}}| \leqslant \beta_k^{\mathrm{FR}}, \\ \beta_k^{\mathrm{FR}} & \text{if } \beta_k^{\mathrm{PR}} > \beta_k^{\mathrm{FR}}, \end{cases} \tag{54}$$

we have noted already that $\beta_{k+1}^{\mathrm{FR}} \approx 1$, and $\beta_{k+1}^{\mathrm{PR}} \approx 0$. The formula (54) thus sets $\beta_{k+1} = \beta_{k+1}^{\mathrm{PR}}$, as desired. Thus, the modification (54) seems to avoid the inefficiencies of the FR method, while falling back on this method for global convergence.

§5.2 Nonlinear Conjugate Gradient Methods

- **Global Convergence**

Unlike the linear conjugate gradient method, whose convergence properties are well understood and which is known to be optimal as described above, nonlinear conjugate gradient methods possess surprising, sometimes bizarre, convergence properties. We now present a few of the main results known for the Fletcher-Reeves and Polak-Ribière methods using practical line searches.

§5.2 Nonlinear Conjugate Gradient Methods

For the purposes of this section, we make the following (nonrestrictive) assumptions on the objective function.

Assumptions 5.1.

1. The level set $S = \{x \mid f(x) \leqslant f(x_0)\}$ is bounded;
2. In some open neighborhood $\mathcal{N}$ of S, the objective function f is Lipschitz continuously differentiable; that is, ∇f is Lipschitz continuous on $\mathcal{N}$ or

$$\|(\nabla f)(x) - (\nabla f)(y)\| \leqslant L\|x - y\| \qquad \forall\, x, y \in \mathcal{N}\,.$$

These assumptions imply that there is a constant $\bar{\gamma}$ such that

$$\|(\nabla f)(x)\| \leqslant \bar{\gamma} \quad \text{for all } x \in S.$$

§5.2 Nonlinear Conjugate Gradient Methods

For the purposes of this section, we make the following (nonrestrictive) assumptions on the objective function.

Assumptions 5.1.

1. The level set $S = \{x \mid f(x) \leqslant f(x_0)\}$ is bounded;
2. In some open neighborhood $\mathcal{N}$ of S, the objective function f is Lipschitz continuously differentiable; that is, ∇f is Lipschitz continuous on $\mathcal{N}$ or

$$\|(\nabla f)(x) - (\nabla f)(y)\| \leqslant L\|x - y\| \qquad \forall\, x, y \in \mathcal{N}\,.$$

These assumptions imply that there is a constant $\bar{\gamma}$ such that

$$\|(\nabla f)(x)\| \leqslant \bar{\gamma} \quad \text{for all } x \in S.$$

§5.2 Nonlinear Conjugate Gradient Methods

Our main analytical tool in this section is Zoutendijk's theorem. It states, that under Assumptions 5.1, any line search iteration of the form $x_{k+1} = x_k + \alpha_k p_k$, where p_k is a descent direction and α_k satisfies the Wolfe conditions (43) gives the limit

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty \,.$$

We can use this result to prove global convergence for algorithms that are periodically restarted by setting $\beta_k = 0$. If k_1, k_2, and so on denote the iterations on which restarts occur, we have from Zoutendijk's condition that

$$\sum_{k=k_1,k_2,\cdots} \|\nabla f_k\|^2 < \infty \,. \tag{59}$$

§5.2 Nonlinear Conjugate Gradient Methods

If we allow no more than $\bar{n}$ iterations between restarts, the sequence $\{k_j\}_{j=1}^{\infty}$ is infinite, and from (59) we have that $\lim_{j\to\infty} \|\nabla f_{k_j}\| = 0$. That is, a subsequence of gradients approaches zero, or equivalently,

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0\,. \tag{60}$$

This result applies equally to restarted versions of all the algorithms discussed in this chapter.

It is more interesting, however, to study the global convergence of **unrestarted** conjugate gradient methods, because for large problems (say $n \geqslant 1000$) we expect to find a solution in many fewer than n iterations – the first point at which a regular restart would take place. Our study of large sequences of unrestarted conjugate gradient iterations reveals some surprising patterns in their behavior.

§5.2 Nonlinear Conjugate Gradient Methods

If we allow no more than $\bar{n}$ iterations between restarts, the sequence $\{k_j\}_{j=1}^{\infty}$ is infinite, and from (59) we have that $\lim_{j\to\infty} \|\nabla f_{k_j}\| = 0$. That is, a subsequence of gradients approaches zero, or equivalently,

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0\,. \tag{60}$$

This result applies equally to restarted versions of all the algorithms discussed in this chapter.

It is more interesting, however, to study the global convergence of **unrestarted** conjugate gradient methods, because for large problems (say $n \geqslant 1000$) we expect to find a solution in many fewer than n iterations – the first point at which a regular restart would take place. Our study of large sequences of unrestarted conjugate gradient iterations reveals some surprising patterns in their behavior.

§5.2 Nonlinear Conjugate Gradient Methods

We can build on Zoutendijk's theorem and the lemma showing that

$$-\frac{1}{1-c_2} \leqslant \frac{\nabla f_k^{\mathrm{T}} p_k}{\|\nabla f_k\|^2} \leqslant \frac{2c_2-1}{1-c_2} \quad \text{for all } k \in \mathbb{N} \cup \{0\}. \tag{44}$$

to prove a global convergence result for the Fletcher-Reeves method. While we cannot show that the limit of the sequence of gradients $\{\nabla f_k\}$ is zero, the following result shows that this sequence is not bounded away from zero.

Theorem

Suppose that Assumptions 5.1 holds, and that Algorithm 5.4' is implemented with a line search that satisfies the strong Wolfe conditions (43)*, with* $0 < c_1 < c_2 < \frac{1}{2}$*. Then*

$$\liminf_{k \to \infty} \|\nabla f_k\| = 0\,. \tag{60}$$

§5.2 Nonlinear Conjugate Gradient Methods

Proof.

The proof is by contradiction. It assumes that the opposite of (60) holds; that is, there exist $\gamma > 0$ such that

$$\|\nabla f_k\| \geqslant \gamma \quad \text{for all } k \in \mathbb{N} \cup \{0\}.$$

By substituting the left inequality of

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k \in \mathbb{N} \cup \{0\} \tag{48}$$

into Zoutendijk's condition, we obtain

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \leqslant \sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} \leqslant \left(\frac{1-c_2}{1-2c_2}\right)^2 \sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{62}$$

Using (43b) and (44), we obtain that

$$|\nabla f_k^{\mathrm{T}} p_{k-1}| \leqslant -c_2 \nabla f_{k-1}^{\mathrm{T}} p_{k-1} \leqslant \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2.$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof.

The proof is by contradiction. It assumes that the opposite of (60) holds; that is, there exist $\gamma > 0$ such that

$$\|\nabla f_k\| \geqslant \gamma \quad \text{for all } k \in \mathbb{N} \cup \{0\}.$$

By substituting the left inequality of

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k \in \mathbb{N} \cup \{0\} \tag{48}$$

into Zoutendijk's condition, we obtain

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \leqslant \sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} \leqslant \left(\frac{1-c_2}{1-2c_2}\right)^2 \sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{62}$$

Using (43b) and (44), we obtain that

$$|\nabla f_k^{\mathrm{T}} p_{k-1}| \leqslant -c_2 \nabla f_{k-1}^{\mathrm{T}} p_{k-1} \leqslant \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2.$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof.

The proof is by contradiction. It assumes that the opposite of (60) holds; that is, there exist $\gamma > 0$ such that

$$\|\nabla f_k\| \geqslant \gamma \quad \text{for all } k \in \mathbb{N} \cup \{0\}.$$

By substituting the left inequality of

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k \in \mathbb{N} \cup \{0\} \tag{48}$$

into Zoutendijk's condition, we obtain

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \leqslant \sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} \leqslant \left(\frac{1-c_2}{1-2c_2}\right)^2 \sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{62}$$

Using (43b) and (44), we obtain that

$$|\nabla f_k^{\mathrm{T}} p_{k-1}| \leqslant -c_2 \nabla f_{k-1}^{\mathrm{T}} p_{k-1} \leqslant \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2.$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof.

The proof is by contradiction. It assumes that the opposite of (60) holds; that is, there exist $\gamma > 0$ such that

$$\|\nabla f_k\| \geqslant \gamma \quad \text{for all } k \in \mathbb{N} \cup \{0\}.$$

By substituting the left inequality of

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k \in \mathbb{N} \cup \{0\} \tag{48}$$

into Zoutendijk's condition, we obtain

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \leqslant \sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} \leqslant \left(\frac{1-c_2}{1-2c_2}\right)^2 \sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{62}$$

Using (43b) and (44), we obtain that

$$|\nabla f_k^{\mathrm{T}} p_{k-1}| \leqslant -c_2 \nabla f_{k-1}^{\mathrm{T}} p_{k-1} \leqslant \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2.$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof.

The proof is by contradiction. It assumes that the opposite of (60) holds; that is, there exist $\gamma > 0$ such that

$$\|\nabla f_k\| \geqslant \gamma \quad \text{for all } k \in \mathbb{N} \cup \{0\}.$$

By substituting the left inequality of

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leqslant \cos\theta_k \leqslant \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{for all } k \in \mathbb{N} \cup \{0\} \tag{48}$$

into Zoutendijk's condition, we obtain

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \leqslant \sum_{k=0}^{\infty} \frac{\|\nabla f_k\|^4}{\|p_k\|^2} \leqslant \left(\frac{1-c_2}{1-2c_2}\right)^2 \sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty. \tag{62}$$

Using (43b) and (44), we obtain that

$$|\nabla f_k^{\mathrm{T}} p_{k-1}| \leqslant -c_2 \nabla f_{k-1}^{\mathrm{T}} p_{k-1} \leqslant \frac{c_2}{1-c_2}\|\nabla f_{k-1}\|^2.$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Thus, from (41b) and recalling the definition (41a) of β_k^{FR}, by defining $c_3 = \dfrac{1+c_2}{1-c_2} \geqslant 1$ we obtain

$$\begin{aligned}\|p_k\|^2 &\leqslant \|\nabla f_k\|^2 + 2\beta_k^{\mathrm{FR}} |\nabla f_k^{\mathrm{T}} p_{k-1}| + (\beta_k^{\mathrm{FR}})^2 \|p_{k-1}\|^2 \\ &\leqslant \|\nabla f_k\|^2 + \frac{2c_2}{1-c_2} \beta_k^{\mathrm{FR}} \|\nabla f_{k-1}\|^2 + (\beta_k^{\mathrm{FR}})^2 \|p_{k-1}\|^2 \\ &= c_3 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4} \|p_{k-1}\|^2 .\end{aligned}$$

Applying this relation repeatedly,

$$\|p_k\|^2 \leqslant c_3 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4} \left[c_3 \|\nabla f_{k-1}\|^2 + \frac{\|\nabla f_{k-1}\|^4}{\|\nabla f_{k-2}\|^4} \|p_{k-2}\|^2 \right]$$

$$= c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_k\|^2} + c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^2} + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-2}\|^4} \|p_{k-2}\|^2$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Thus, from (41b) and recalling the definition (41a) of β_k^{FR}, by defining $c_3 = \dfrac{1+c_2}{1-c_2} \geqslant 1$ we obtain

$$\begin{aligned}\|p_k\|^2 &\leqslant \|\nabla f_k\|^2 + 2\beta_k^{\mathrm{FR}} |\nabla f_k^{\mathrm{T}} p_{k-1}| + (\beta_k^{\mathrm{FR}})^2 \|p_{k-1}\|^2 \\ &\leqslant \|\nabla f_k\|^2 + \frac{2c_2}{1-c_2} \beta_k^{\mathrm{FR}} \|\nabla f_{k-1}\|^2 + (\beta_k^{\mathrm{FR}})^2 \|p_{k-1}\|^2 \\ &= c_3 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4} \|p_{k-1}\|^2 .\end{aligned}$$

Applying this relation repeatedly,

$$\begin{aligned}\|p_k\|^2 &\leqslant c_3 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4} \left[c_3 \|\nabla f_{k-1}\|^2 + \frac{\|\nabla f_{k-1}\|^4}{\|\nabla f_{k-2}\|^4} \|p_{k-2}\|^2 \right] \\ &= c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_k\|^2} + c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^2} + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-2}\|^4} \|p_{k-2}\|^2 \end{aligned}$$

□

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Thus, from (41b) and recalling the definition (41a) of β_k^{FR}, by defining $c_3 = \dfrac{1+c_2}{1-c_2} \geqslant 1$ we obtain

$$\|p_k\|^2 \leqslant c_3 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4} \|p_{k-1}\|^2 .$$

Applying this relation repeatedly,

$$\begin{aligned}
\|p_k\|^2 &\leqslant c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_k\|^2} + c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^2} + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-2}\|^4} \|p_{k-2}\|^2 \\
&\leqslant c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_k\|^2} + c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^2} + c_3 \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-2}\|^2} + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-3}\|^4} \|p_{k-3}\|^2 \\
&\leqslant \cdots\cdots \leqslant c_3 \|\nabla f_k\|^4 \sum_{j=0}^{k} \|\nabla f_j\|^{-2} , \qquad (64)
\end{aligned}$$

where the fact that $p_0 = -\nabla f_0$ is used to conclude the inequality. □

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Using the bounds

$$\|\nabla f_k\| \geqslant \gamma \text{ for all } k \in \mathbb{N} \cup \{0\} \quad \text{and} \quad \|(\nabla f)(x)\| \leqslant \bar{\gamma} \text{ for all } x \in S$$

in (64), we obtain

$$\|p_k\|^2 \leqslant c_3 \frac{\bar{\gamma}^4}{\gamma^2}(k+1) \quad \text{for all } k \in \mathbb{N} \cup \{0\}$$

which implies that

$$\sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \geqslant \frac{\gamma^2}{c_3 \bar{\gamma}^4} \sum_{k=0}^{\infty} \frac{1}{k+1} = \infty ,$$

a contradiction to (62). □

In general, if there exist constants c_4, $c_5 > 0$ such that

$$\cos\theta_k \geqslant c_4 \frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{and} \quad \frac{\|\nabla f_k\|}{\|p_k\|} \geqslant c_5 > 0 \quad \text{for all } k = 1, 2, \cdots,$$

it follows from Zoutendijk's condition that $\lim_{k\to\infty} \|\nabla f_k\| = 0$.

§5.2 Nonlinear Conjugate Gradient Methods

Proof (cont'd).

Using the bounds

$$\|\nabla f_k\| \geqslant \gamma \text{ for all } k \in \mathbb{N} \cup \{0\} \quad \text{and} \quad \|(\nabla f)(x)\| \leqslant \bar{\gamma} \text{ for all } x \in S$$

in (64), we obtain

$$\|p_k\|^2 \leqslant c_3 \frac{\bar{\gamma}^4}{\gamma^2}(k+1) \quad \text{for all } k \in \mathbb{N} \cup \{0\}$$

which implies that

$$\sum_{k=0}^{\infty} \frac{1}{\|p_k\|^2} \geqslant \frac{\gamma^2}{c_3 \bar{\gamma}^4} \sum_{k=0}^{\infty} \frac{1}{k+1} = \infty ,$$

a contradiction to (62). □

In general, if there exist constants c_4, $c_5 > 0$ such that

$$\cos\theta_k \geqslant c_4 \frac{\|\nabla f_k\|}{\|p_k\|} \quad \text{and} \quad \frac{\|\nabla f_k\|}{\|p_k\|} \geqslant c_5 > 0 \quad \text{for all } k = 1, 2, \cdots,$$

it follows from Zoutendijk's condition that $\lim_{k\to\infty} \|\nabla f_k\| = 0$.

§5.2 Nonlinear Conjugate Gradient Methods

In fact, the result

$$\lim_{k\to\infty} \|\nabla f_k\| = 0$$

can be established for the Polak-Ribière method under the assumption that f is strongly convex and that an exact line search is used. For general (non-convex) functions, however, it is **not** possible to prove a result $\left(\liminf_{k\to\infty} \|\nabla f_k\| = 0\right)$ like the previous theorem for Algorithm PR. This fact is unexpected, since the Polak-Ribière method performs better in practice than the Fletcher-Reeves method. The following surprising result shows that the Polak-Ribière method can cycle infinitely without approaching a solution point, even if an **ideal** line search is used. (By "ideal" we mean that line search returns a value α_k that is the first positive stationary point for the function $t(\alpha) = f(x_k + \alpha p_k)$.)

§5.2 Nonlinear Conjugate Gradient Methods

In fact, the result

$$\lim_{k\to\infty} \|\nabla f_k\| = 0$$

can be established for the Polak-Ribière method under the assumption that f is strongly convex and that an exact line search is used. For general (non-convex) functions, however, it is **not** possible to prove a result $\left(\liminf_{k\to\infty} \|\nabla f_k\| = 0\right)$ like the previous theorem for Algorithm PR. This fact is unexpected, since the Polak-Ribière method performs better in practice than the Fletcher-Reeves method. The following surprising result shows that the Polak-Ribière method can cycle infinitely without approaching a solution point, even if an **ideal** line search is used. (By "ideal" we mean that line search returns a value α_k that is the first positive stationary point for the function $t(\alpha) = f(x_k + \alpha p_k)$.)

§5.2 Nonlinear Conjugate Gradient Methods

In fact, the result

$$\lim_{k\to\infty} \|\nabla f_k\| = 0$$

can be established for the Polak-Ribière method under the assumption that f is strongly convex and that an exact line search is used. For general (non-convex) functions, however, it is **not** possible to prove a result $\left(\liminf_{k\to\infty} \|\nabla f_k\| = 0\right)$ like the previous theorem for Algorithm PR. This fact is unexpected, since the Polak-Ribière method performs better in practice than the Fletcher-Reeves method. The following surprising result shows that the Polak-Ribière method can cycle infinitely without approaching a solution point, even if an **ideal** line search is used. (By "ideal" we mean that line search returns a value α_k that is the first positive stationary point for the function $t(\alpha) = f(x_k + \alpha p_k)$.)

§5.2 Nonlinear Conjugate Gradient Methods

Theorem

Consider the Polak-Ribière method (50) *with an ideal line search. There exists a twice continuously differentiable objective function* $f : \mathbb{R}^3 \to \mathbb{R}$ *and a starting point* $x_0 \in \mathbb{R}^3$ *such that the sequence of gradients* $\{\|\nabla f_k\|\}$ *is bounded away from zero.*

The proof of this result, given in [253], is quite complex. It demonstrates the existence of the desired objective function without actually constructing this function explicitly. The result is interesting, since the step length assumed in the proof – the first stationary point – may be accepted by any of the practical line search algorithms currently in use.

§5.2 Nonlinear Conjugate Gradient Methods

Theorem

Consider the Polak-Ribière method (50) *with an ideal line search. There exists a twice continuously differentiable objective function* $f : \mathbb{R}^3 \to \mathbb{R}$ *and a starting point* $x_0 \in \mathbb{R}^3$ *such that the sequence of gradients* $\{\|\nabla f_k\|\}$ *is bounded away from zero.*

The proof of this result, given in [253], is quite complex. It demonstrates the existence of the desired objective function without actually constructing this function explicitly. The result is interesting, since the step length assumed in the proof – the first stationary point – may be accepted by any of the practical line search algorithms currently in use.

§5.2 Nonlinear Conjugate Gradient Methods

The proof of this theorem requires that some consecutive search directions become almost negatives of each other. In the case of ideal line searches, this happens only if $\beta_k < 0$, so the analysis suggests Algorithm PR+, in which we reset β_k to zero whenever it becomes negative. We mentioned earlier that a line search strategy based on a slight modification of the Wolfe conditions guarantees that all search directions generated by Algorithm PR+ are descent directions. Using these facts, it is possible to a prove global convergence result for Algorithm PR+.

§5.2 Nonlinear Conjugate Gradient Methods

The proof of this theorem requires that some consecutive search directions become almost negatives of each other. In the case of ideal line searches, this happens only if $\beta_k < 0$, so the analysis suggests Algorithm PR+, in which we reset β_k to zero whenever it becomes negative. We mentioned earlier that a line search strategy based on a slight modification of the Wolfe conditions guarantees that all search directions generated by Algorithm PR+ are descent directions. Using these facts, it is possible to a prove global convergence result for Algorithm PR+.

§5.2 Nonlinear Conjugate Gradient Methods

- **Numerical Performance**

Table 6 (in the next page) illustrates the performance of Algorithms FR, PR, and PR+ without restarts. For these tests, the parameters in the strong Wolfe conditions (43) were chosen to be $c_1 = 10^{-4}$ and $c_2 = 0.1$. The iterations were terminated when

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

If this condition was not satisfied after 10,000 iterations, we declare failure (indicated by a $*$ in the table). The final column, headed "mod," indicates the number of iterations of Algorithm PR+ for which the adjustment (51) was needed to ensure that $\beta_k^{\mathrm{PR}} \geqslant 0$.

§5.2 Nonlinear Conjugate Gradient Methods

- **Numerical Performance**

Table 6 (in the next page) illustrates the performance of Algorithms FR, PR, and PR+ without restarts. For these tests, the parameters in the strong Wolfe conditions (43) were chosen to be $c_1 = 10^{-4}$ and $c_2 = 0.1$. The iterations were terminated when

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

If this condition was not satisfied after 10,000 iterations, we declare failure (indicated by a $*$ in the table). The final column, headed "mod," indicates the number of iterations of Algorithm PR+ for which the adjustment (51) was needed to ensure that $\beta_k^{\mathrm{PR}} \geqslant 0$.

§5.2 Nonlinear Conjugate Gradient Methods

- **Numerical Performance**

Table 6 (in the next page) illustrates the performance of Algorithms FR, PR, and PR+ without restarts. For these tests, the parameters in the strong Wolfe conditions (43) were chosen to be $c_1 = 10^{-4}$ and $c_2 = 0.1$. The iterations were terminated when

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

If this condition was not satisfied after 10,000 iterations, we declare failure (indicated by a $*$ in the table). The final column, headed "mod," indicates the number of iterations of Algorithm PR+ for which the adjustment (51) was needed to ensure that $\beta_k^{\mathrm{PR}} \geqslant 0$.

§5.2 Nonlinear Conjugate Gradient Methods

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

Figure 6: Iterations and function/gradient evaluations required by three nonlinear conjugate gradient methods on a set of test problems; see [123]

Note that the Polak-Ribière algorithm, or its variation PR+, are not always more efficient than Algorithm FR, and it has the slight disadvantage of requiring one more vector of storage.