

# **Set 12: Unconstrained Optimization**

**Kyle A. Gallivan**

Department of Mathematics

**Florida State University**

**Foundations of Computational Math 1**

**Fall 2011**

## Unconstrained Smooth Optimization

**Problem 12.1.** Given  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ , solve

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

to find a local minimizer.

- global convergence desirable
- superlinear convergence rate very desirable
- robustness desirable
- efficiency desirable
- both  $n = 1$  and  $n > 1$  are of practical interest

## Solutions

**Definition 12.1.** The following minimizers are of interest:

- The point  $x^* \in \mathbb{R}^n$  is a global minimizer if  $f(x^*) \leq f(x)$  for all  $x \in \mathbb{R}^n$ .
- The point  $x^* \in \mathbb{R}^n$  is a local minimizer if  $f(x^*) \leq f(x)$  for all  $x \in \mathcal{N}_{x^*} \subset \mathbb{R}^n$  where  $\mathcal{N}_{x^*}$  is a neighborhood of  $x^*$ .

## Global vs. Local Convergence

- A locally convergent method converges to a local minimizer only when the initial point is sufficiently close to the local minimizer.
- The particular local minimizer reached is a complicated function of the initial point, the method, and the problem.
- A globally convergent method converges to a stationary point (hopefully also a local minimizer) for any initial point (or fails in one of a small number of detectable ways).
- Global optimization methods attempt (or make it very likely) to find a global minimizer. This is much more difficult in general.

## Approaches

There are two main ideas behind unconstrained smooth optimization algorithms:

- line search
- trust region

Two other ideas are often combined with the two above:

- majorization
- continuation/homotopy

## Definitions

**Definition 12.2.** If  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  then

- the gradient of  $f$ , denoted,  $\nabla f \in \mathbb{R}^n$ , is the vector with  $i$ -th element,

$$\frac{\partial f}{\partial \xi_i}(x)$$

- the Hessian of  $f$  denoted,  $\nabla^2 f \in \mathbb{R}^{n \times n}$ , is the symmetric matrix with  $i, j$ -element,

$$\frac{\partial^2 f}{\partial \xi_i \partial \xi_j}(x)$$

## Necessary Conditions

**Theorem 12.1. (First order necessary)** *Suppose  $f \in \mathcal{C}^1$  in a neighborhood of  $x^*$ . If  $x^*$  is a local minimizer then  $\nabla f(x^*) = 0$ , i.e.,  $x^*$  is a stationary point.*

**Theorem 12.2. (Second order necessary)** *Suppose  $f \in \mathcal{C}^2$  in a neighborhood of  $x^*$ . If  $x^*$  is a local minimizer then  $\nabla f(x^*) = 0$  and  $\nabla^2 f(x^*)$  is positive semidefinite.*

## Sufficient Condition

**Theorem 12.3. (Second order sufficient)** Suppose  $f \in \mathcal{C}^2$  in a neighborhood of  $x^*$ . If  $\nabla f(x^*) = 0$  and  $\nabla^2 f(x^*)$  is positive definite then  $x^*$  is a strict local minimizer.

*Note this means*

$$\forall p \in \mathbb{R}^n \quad p^T \nabla^2 f(x^*) p > 0$$

*which says all directions have positive curvature.*

*Note.* Theorem 12.3 is not a necessary condition. The point  $x^* = 0$  is a strict local minimizer of  $f(x) = x^4$  but  $\nabla^2 f(0) = 0$ .

**Theorem 12.4.** *If  $f$  is convex then any local minimizer is a global minimizer. If, in addition,  $f \in \mathcal{C}^1$  then any stationary point is a global minimizer.*

## Line Search

**Main Task:** Solve  $\nabla f(x) = 0$ .  $n$  nonlinear equations,  $n$  unknowns.

**Basic Idea:** Given a point  $x_k$ , choose direction  $p_k \in \mathbb{R}^n$  and stepsize  $\alpha_k \in \mathbb{R}$  such that  $f(x_k + \alpha_k p_k) < f(x_k)$ .

### Line Search Issues

- What is a good direction?
- How is stepsize chosen to be large enough to make progress toward reducing  $f$  but not so far as to cause convergence problems?

## Line Search

**Theorem 12.5.** *Let  $f \in \mathcal{C}^1$  and  $p \in \mathbb{R}^n$ . If*

$$p^T \nabla f_k = \|p\|_2 \|\nabla f_k\|_2 \cos \theta_k < 0$$

*$p$  is a descent direction at  $x$ , i.e.,  $f(x + \alpha p) < f(x)$  for all sufficiently small  $\alpha > 0$ .*

Note this says that  $p$  has a nontrivial component in the direction of the negative gradient and the vectors form an acute angle.

$f_k$  used for  $f(x_k)$  for function, gradient and Hessian expressions.

## Direction Vectors

- Steepest Descent

- (i)  $p_k = -\nabla f_k$  is the direction of steepest (instantaneous) descent.
- (ii)  $\alpha_k$  must be found.

- Newton

- (i)  $p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$  with natural scale  $\alpha_k = 1$
- (ii)  $\alpha_k < 1$  is Damped Newton.
- (iii) used as line search when  $\nabla^2 f_k$  is positive definite
- (iv)  $p_k^N$  may not exist or may not be a descent direction
- (v) line search methods modify  $p_k^N$  in these cases

## Direction Vectors

- Inexact Newton solve  $\nabla^2 f_k p_k = -\nabla f_k$  approximately
- Quasi-Newton methods
  - (i)  $B_k p_k = -\nabla f_k$
  - (ii) secant condition  $B_{k+1}(x^{(k+1)} - x^{(k)}) = (F(x^{(k+1)}) - F(x^{(k)}))$  enforced
  - (iii) often coupled with CG modified to handle semidefinite and indefinite symmetric matrices to keep cost per iteration acceptable.
- nonlinear CG methods use  $p_k = -\nabla f_k + \beta_k p_{k-1}$  with  $\beta_k$  chosen using conjugacy.

## Step Selection

- $f(x_k + p_k \alpha_k) < f(x_k)$  is not enough
- minimizing  $\phi(\alpha) = f(x_k + p_k \alpha)$  not practical usually
- need sufficient decrease (Armijo condition)
- need to avoid very small steps (curvature condition)
- Wolfe conditions (weak and strong versions)
- Alternatives possible, e.g., Goldstein conditions.
- Forms part of conditions needed to guarantee convergence

## Step via Backtracking

Complicated step selection algorithms are typically used for robustness but a simple strategy can work.

Assume  $\tilde{\alpha}$  is taken as an initial step (typically near 1 for Newton). Choose  $0 < \rho < 1$  and  $0 < \gamma < 1$  and set  $\alpha \leftarrow \tilde{\alpha}$ .

repeat until  $f(x_k + p_k \alpha) \leq f(x_k) + (\gamma_1 \nabla f_k^T p_k) \alpha$

$$\alpha \leftarrow \rho \alpha$$

terminate with  $\alpha_k = \alpha$

*Note.*  $\rho$  is typically allowed to vary on each line search iteration. The choice of  $\tilde{\alpha}$  is often critical for convergence rate and efficiency of the backtracking.

## Linear System Solving Main Ideas

- Create a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^+$  which has a unique minimum value  $f(x_{min})$  where  $x_{min} = A^{-1}b$ .
- Pick a direction  $p_k$ , choose a length of move  $\alpha_k$ :

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

- Ideally,  $\alpha_k$  that minimizes  $f(x_k + \alpha p_k)$  has simple solution.
- Fixed point iteration with fixed point  $x_{min}$ .

## Symmetric Positive Definite Systems

If  $A \in \mathbb{R}^{n \times n}$  is symmetric positive definite,

- $A$  defines an inner product on  $\mathbb{R}^n$
- Any inner product induces a vector norm:

$$\|v\|_A^2 = v^T A v$$

- We have:

Solving  $Ax = b$  is equivalent to minimizing a quadratic functional that defines a convex  $f(x)$ ..

## A-norm Minimization

$$\begin{aligned}\hat{x} &= A^{-1}b \text{ and } \phi(x) = \|x - \hat{x}\|_A^2 \\ \phi(x) &= (x - A^{-1}b)^T A(x - A^{-1}b) \\ &= (x^T A - b^T A^{-T} A)(x - A^{-1}b) \\ &= x^T Ax - x^T AA^{-1}b - b^T A^{-T} Ax + b^T A^{-T} AA^{-1}b \\ &= x^T Ax - x^T b - b^T x + b^T A^{-1}b = x^T Ax - 2b^T x + b^T A^{-1}b\end{aligned}$$

$$Q(x) = \frac{1}{2}x^T Ax - b^T x$$

$$\operatorname{argmin}_{x \in \mathbb{R}^n} \phi(x) = \operatorname{argmin}_{x \in \mathbb{R}^n} Q(x)$$

## Stepsize Selection

**Theorem 12.6.** Suppose  $x_k \in \mathbb{R}^n$  be the current guess at  $x$  and  $p_k \in \mathbb{R}^n$  is the direction in which the next step is to be taken. The scale  $\alpha_k$  that minimizes  $Q(x_{k+1})$  where

$$x_{k+1} = x_k + \alpha_k p_k \quad \text{is} \quad \alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

*Proof.*

$$\begin{aligned} Q(\alpha) &= \frac{1}{2}(x_k + \alpha p_k)^T A(x_k + \alpha p_k) - b^T(x_k + \alpha p_k) \\ &= \frac{1}{2}\alpha^2 p_k^T A p_k - \alpha r_k^T p_k - b^T x_k + \frac{1}{2}x_k^T A x_k \\ Q'(\alpha) &= \alpha p_k^T A p_k - r_k^T p_k \end{aligned}$$

□

## Steepest Descent

**Lemma 12.7.** *If  $Q(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  then the direction of most rapid descent on the surface  $Q(x)$  at some point  $x^*$  is the negative gradient*

$$-\nabla Q(x^*) = \begin{pmatrix} -\frac{\partial Q}{\partial \xi_1}(x^*) \\ \vdots \\ -\frac{\partial Q}{\partial \xi_n}(x^*) \end{pmatrix}$$

$\nabla Q(x^*)$  is the external normal of the tangent plane of the surface defined by  $Q(x)$ . We have

$$-\nabla Q(x) = b - Ax = r$$

which is the residual vector.

## Example

Let  $A = A^T$  be  $2 \times 2$ . Note  $\alpha_{12} = \alpha_{21}$ .

$$Q(x) = \frac{1}{2} \begin{pmatrix} \xi_1 & \xi_2 \end{pmatrix} \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} - \begin{pmatrix} \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$$

$$= \frac{1}{2} (\xi_1 \alpha_{11} \xi_1 + \xi_1 \alpha_{12} \xi_2 + \xi_2 \alpha_{21} \xi_1 + \xi_2 \alpha_{22} \xi_2) - (\beta_1 \xi_1 + \beta_2 \xi_2)$$

$$\frac{\partial Q}{\partial \xi_1} = \alpha_{11} \xi_1 + \frac{1}{2} (\alpha_{12} + \alpha_{21}) \xi_2 - \beta_1$$

$$\frac{\partial Q}{\partial \xi_2} = \alpha_{22} \xi_2 + \frac{1}{2} (\alpha_{12} + \alpha_{21}) \xi_1 - \beta_2$$

## Steepest Descent

- example of a line search for unconstrained optimization
- negative gradient, i.e., steepest descent direction, always used
- stepsize  $\alpha_k$  selection varies for different problems
- Direction selected is always locally fastest.
- There is often a better direction to consider based on more “global” information.

## Steepest Descent for $Ax = b$

- For  $Ax = b$  the cost function of  $\|x - A^{-1}b\|_A^2$  is used.
- The negative gradient is the residual vector.
- Steepest descent solves a series of one dimensional optimization problems, i.e., constrained to a line in  $\mathbb{R}^n$ .
- Analytical solution available and used in algorithm.
- $r_{k+1} \perp r_k$  but no global relationship is maintained between them.
- SD can converge very slowly.
- Consider level curves and iterates for example with  $n = 2$ .

## Steepest Descent

$A$  is symmetric positive definite

$x_0$  arbitrary;  $r_0 = b - Ax_0$ ;  $v_0 = Ar_0$

do  $k = 0, 1, \dots$  until convergence

$$\alpha_k = \frac{r_k^T r_k}{r_k^T v_k}$$

$$x_{k+1} \leftarrow x_k + r_k \alpha_k$$

$$r_{k+1} \leftarrow r_k - v_k \alpha_k$$

$$v_{k+1} \leftarrow Ar_{k+1}$$

end

## Preconditioned Steepest Descent

$A, M$  are symmetric positive definite

$x_0$  arbitrary;  $r_0 = b - Ax_0$ ; solve  $Mz_0 = r_0$

do  $k = 0, 1, \dots$  until convergence

$$w_k = Az_k$$

$$\alpha_k = \frac{z_k^T r_k}{r_k^T w_k}$$

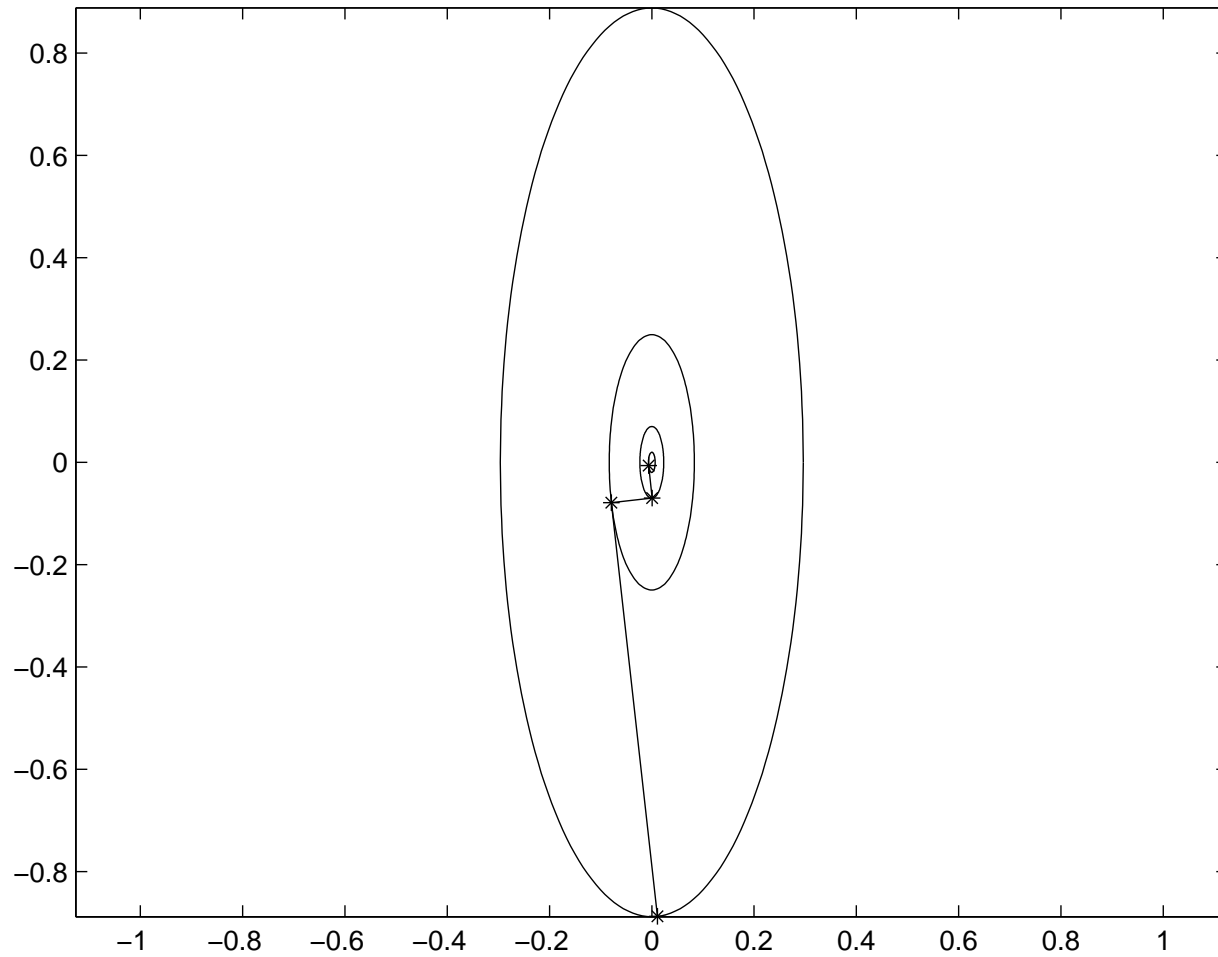
$$x_{k+1} \leftarrow x_k + z_k \alpha_k$$

$$r_{k+1} \leftarrow r_k - w_k \alpha_k$$

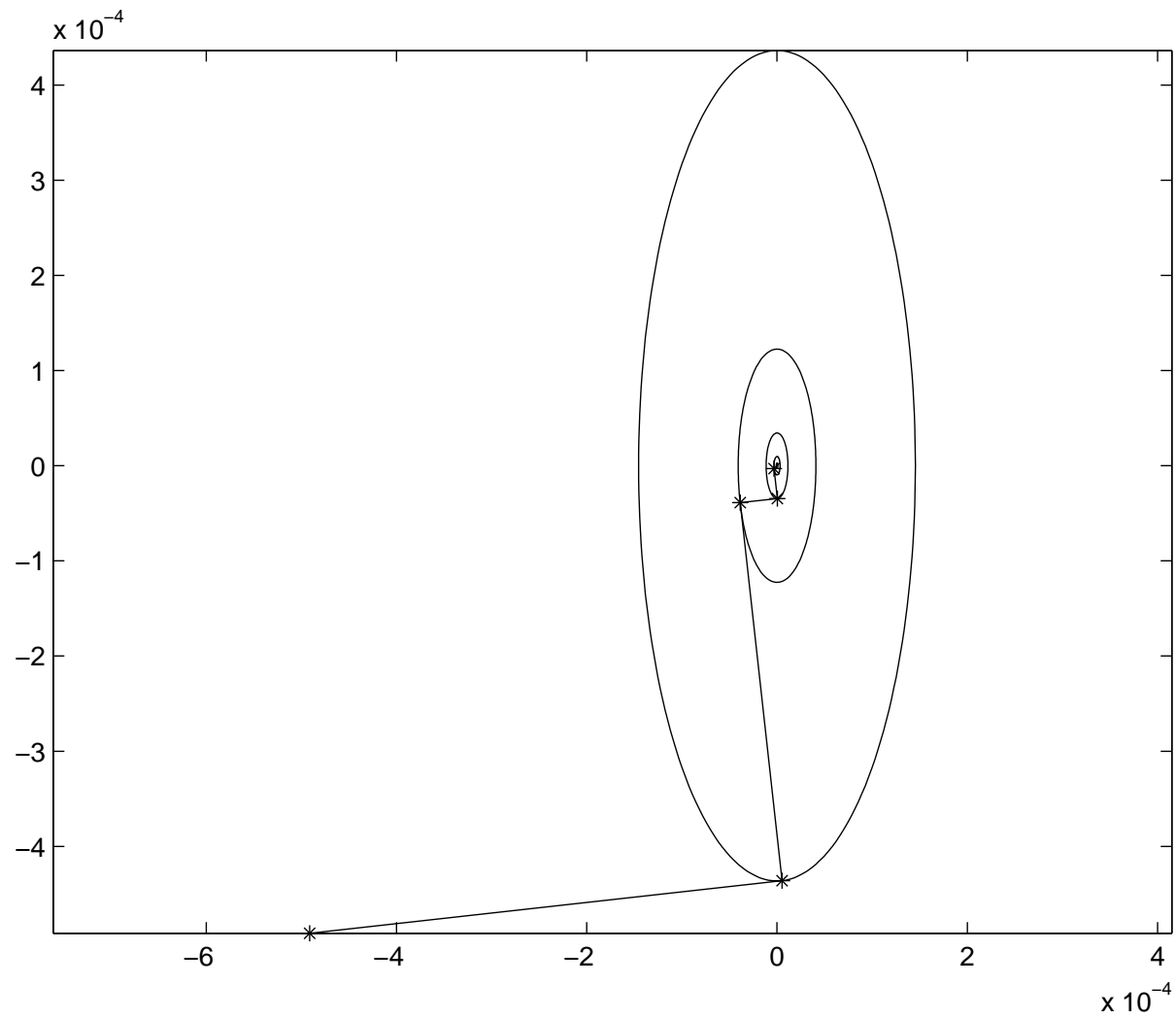
$$\text{solve } Mz_{k+1} = r_{k+1}$$

end

Iterations 2 to 5 for  $n = 2$ ,  $\lambda_1 = 9$ ,  $\lambda_2 = 1$



Iterations 7 to 11 for  $n = 2$ ,  $\lambda_1 = 9$ ,  $\lambda_2 = 1$ , note scale difference



## Convergence Rates from Level Curves

Convergence is a function of the eccentricity of the ellipses that define the level curves of  $Q$ .

You move in the direction of  $r$  until the point at which the line  $x_k + \alpha r_k$  is tangent to a level curve. This must be the minimum in that direction.

Very eccentric curves cause the search to ping-pong across the low “valley” rather than following the bottom of the valley.

## Convergence Rates

- $A$  is symmetric positive definite
- $\|A\|_2 = \lambda_{max}$  and  $\|A^{-1}\|_2 = \lambda_{min}^{-1}$
- $\kappa(A)_2 = \lambda_{max}/\lambda_{min}$
- error is damped based on eccentricity

$$\|e^{(k+1)}\|_A \leq \frac{\kappa(A)_2 - 1}{\kappa(A)_2 + 1} \|e^{(k)}\|_A$$

- note similarity to earlier analysis of stationary method
- preconditioning with some  $M$  can improve the convergence considerably by changing the eigenvalues.

## Conjugate Directions for $Ax = b$

- $\langle x, y \rangle_A = x^T Ay$  is an inner product.
- $\langle e, e \rangle_A = \|e\|_A^2$  used to define cost  $Q(x)$ , where  $e = x - A^{-1}b$ .
- $\langle x, y \rangle_A$  defines conjugacy or  $A$ -orthogonality.
- $p_0, p_1, \dots, p_k$  are  $A$ -orthogonal if

$$\langle p_i, p_j \rangle_A \begin{cases} = 0 & \text{if } i \neq j \\ \neq 0 & \text{if } i = j \end{cases}$$

## Conjugate Directions for $Ax = b$

- $A$ -orthogonality is a global constraint we relate to the underlying notion of a series of 1-dimensional optimization problems to get
  - efficiency per step
  - minimization of the error, i.e., solving the system  $Ax = b$
- Many algorithms possible based on conjugate directions.

## Conjugate Gradients

- We still have to specify how to choose a particular  $p_{k-1}$ .
- The efficiency of the method also needs to be considered and various properties used to reduce the complexity of an iteration.
- One very popular version that accomplishes all of these is the **conjugate gradient algorithm** (CG).
- There are **many** ways to derive CG.
- **Idea:** Combine steepest descent direction with  $A$ -orthogonality to get  $p_{k-1}$ .

## Conjugate Gradient Algorithm Derivation

CG chooses  $p_{k-1}$  to be the vector that minimizes

$$\|p - r_{k-1}\|_2$$

over all  $p \in \mathbb{R}^n$  that are orthogonal to  $\mathcal{R}(AP_{k-2})$  where

$$P_{k-2} = \begin{bmatrix} p_0 & p_1 & \dots & p_{k-2} \end{bmatrix}$$

The first condition is based on the notion that  $r_{k-1}$  is the steepest gradient direction but we know we must modify it to be  $A$ -orthogonal with the earlier directions to accelerate convergence by using the global perspective of  $A$ -orthogonality.

## Conjugate Gradient Algorithm Derivation

We already know how to characterize such a vector. If  $z_{min} \in \mathbb{R}^{k-1}$  solves the least squares problem

$$\min_z \|r_{k-1} - AP_{k-2}z\|_2$$

The residual at  $z_{min}$  is orthogonal to  $\mathcal{R}(AP_{k-2})$  and is the closest such vector to  $r_{k-1}$ .

$$p_{k-1} = r_{k-1} - AP_{k-2}z_{min}$$

## Efficient Production of Direction Vector

- The  $A$ -orthogonality of  $p_j$  and their relationship to the residuals can be used to show that the least squares problem defining the next direction vector  $p_k$  has significant structure.
- $p_k$  is a linear combination of  $r_k$  and only  $p_{k-1}$ .
- Specifically,

$$p_k = r_k + \beta_{k-1}p_{k-1}$$
$$\beta_{k-1} = r_k^T r_k / r_{k-1}^T r_{k-1}$$

- This is a key point in the efficiency of CG.  $x_k$ ,  $p_k$ , and  $r_k$  are all given by vector triads, i.e., there is no need to combine all previous  $p_j$  and  $r_j$ .

## Conjugate Gradient Efficient Form

### Algorithm: CG

$$x_0 \text{ arbitrary}; r_0 = b - Ax_0; p_0 = r_0$$

$$k = 0, 1, \dots$$

$$v_k = Ap_k$$

$$\alpha_k = r_k^T r_k / p_k^T v_k$$

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

$$r_{k+1} = r_k - \alpha_k v_k$$

$$\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

end

## Congugate Gradient Method

- Note the low complexity per step.
- This method can be derived from other points of view (including the view in the text).
- It is the standard algorithm to solve large sparse symmetric positive definite linear systems and sparse linear least squares solvers.
- It is the basis for two families of nonsymmetric sparse linear system solvers.
- In practice, it must be preconditioned to work well.
- It can be applied as an accerleration method to other solvers.
- It can be used in very abstract settings, e.g., optimization in vector spaces of functions when solving PDEs.

## Useful CG Facts

- The direction vectors  $p_k$ ,  $k = 0, \dots$  are mutually  $A$ -orthogonal.
- The residuals,  $r_k$ ,  $k = 0, \dots$  are mutually orthogonal
- $r_k$  is orthogonal to  $\mathcal{R}(P_{k-1})$ . (Galerkin condition)
- Two bases for the space in which  $x_k$  resides:

$$S_k = x_0 + \text{span}(p_0, \dots, p_k) = x_0 + \text{span}(r_0, \dots, r_k)$$

## Convergence of CG

- Efficient step in terms of computations and space like SD.
- In exact arithmetic it is a finite algorithm since

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

and  $(p_0, \dots, p_{n-1})$  is a basis for  $\mathbb{R}^n$ , i.e., the algorithm determines  $e^{(0)} = x - x_0$  in terms of this basis.

- This is not good enough and the main convergence results bound  $e^{(k)} = x_k - A^{-1}b$ .
- CG's behavior under finite arithmetic is fairly complicated and is also the subject of much rigorous literature and folklore.

## Convergence and the A-norm

**Theorem 12.8.** For CG,  $e^{(k)}$  is bounded in terms of,  $\kappa$ , the condition number of  $A$  and the initial error  $e^{(0)}$  by

$$\|e^{(k)}\|_A \leq 2\alpha^k \|e^{(0)}\|_A$$

$$\alpha = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$$

## Convergence and the 2-norm

**Theorem 12.9.** For CG,  $e^{(k)}$  is bounded in terms of  $\kappa$ , the condition number of  $A$  and the initial error  $e^{(0)}$  by

$$\|e^{(k)}\|_2 \leq 2\sqrt{\kappa}\alpha^k \|e^{(0)}\|_2$$

$$\alpha = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$$

## Eigenvalues and Convergence

Eigenvalues play an important role in characterizing the convergence of CG.

**Theorem 12.10.** *If  $A$  has  $m$  distinct eigenvalues, i.e., there are  $m$  values  $\mu_1, \dots, \mu_m$  such that*

- *for  $1 \leq i \leq n$  there exists  $j$  such that  $\lambda_i = \mu_j$*
- *for  $1 \leq j \leq m$  there exists at least one  $i$  such that  $\lambda_i = \mu_j$*

*then CG converges in at most  $m$  steps.*

## Rules of Thumb

Heuristically we have the following statements:

- CG converges quickly in the  $A$ -norm if  $\kappa \approx 1$ . This implies the spread of eigenvalues is getting small and therefore it “looks” like there are fewer distinct values. Alternatively, it says the steepest descent level curves are going to circles.
- If  $A$  is close to a rank  $r$  update to the identity then CG is almost converged after  $r$  steps

## Preconditioning

- Finite termination and the distinct eigenvalue convergence theorems typically do not yield satisfactory convergence in practice.
- It is necessary to alter the system in order to improve the convergence rate.
- We transform the coefficient matrix to have, effectively, fewer distinct eigenvalues.
- There is a tradeoff in the cost of transforming the system – or **preconditioning** the system – and the resulting improvement in performance.

## Preconditioned Conjugate Gradient

$A$  and  $M$  are symmetric positive definite matrices.

$x_0$  arbitrary;  $r_0 = b - Ax_0$ ;

solve  $Mz_0 = r_0$ ;  $p_0 = z_0$

$k = 0, 1, \dots$

$$v_k = Ap_k$$

$$\alpha_k = r_k^T z_k / p_k^T v_k$$

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

$$r_{k+1} = r_k - \alpha_k v_k$$

$$\text{solve } Mz_{k+1} = r_{k+1}$$

$$\beta_k = r_{k+1}^T z_{k+1} / r_k^T z_k$$

$$p_{k+1} = z_{k+1} + \beta_k p_k$$

end

## Preconditioners

- diagonal or block diagonal (Jacobi or block Jacobi)
- Symmetric Gauss-Seidel and Symmetric SOR
- Approximate inverse:  $\|I - M^{-1}A\|_F$  is minimized.
- Polynomial preconditioning:  $A^{-1} \approx P(A)$
- Incomplete Cholesky

See text Section 4.3.2 for more discussion of preconditioners.

## Local Models

**Theorem 12.11** (Taylor's Theorem). *Suppose function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuous.*

- *If  $f \in \mathcal{C}^1$  and  $p \in \mathbb{R}^n$  then for some  $0 \leq \tau \leq 1$*

$$f(x + p) = f(x) + \nabla f(x + \tau p)^T p$$

- *If  $f \in \mathcal{C}^2$  and  $p \in \mathbb{R}^n$  then for some  $0 \leq \tau \leq 1$*

$$f(x + p) = f(x) + \nabla f(x)^T p + p^T \nabla^2 f(x + \tau p) p$$

## Use of Quadratic Local Model

Let  $B_k \in \mathbb{R}^{n \times n}$  be symmetric. A quadratic local model for  $f(x)$  near  $x_k$  is of the form:

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + 0.5p^T B_k p$$

Its gradient is

$$\nabla m_k(p) = \nabla f(x_k) + B_k p$$

## Use of Quadratic Local Model

Methods based on quadratic local models take the direction  $p_k$  to be the point at which  $\nabla m_k(p) = 0$ , i.e., a stationary point of  $m_k(p)$ .

$$\nabla m_k(p) = \nabla f(x_k) + B_k p$$

$$\nabla m_k(p_k) = 0 \leftrightarrow B_k p_k = -\nabla f(x_k)$$

## Curvature and Descent Directions

Suppose  $B_k p_k = -\nabla f(x_k)$

- If  $p_k^T B_k p_k > 0$  then  $p_k$  is a direction of positive curvature and

$$-p_k^T \nabla f(x_k) > 0 \rightarrow p_k \text{ is a descent direction of } f$$

- If  $p_k^T B_k p_k < 0$  then  $p_k$  is a direction of negative curvature and

$$p_k^T B_k p_k < 0 \rightarrow -p_k^T \nabla f(x_k) < 0 \rightarrow p_k \text{ is a not descent direction of } f$$

- If  $B_k$  is symmetric positive definite then there are no directions of negative curvature.
- If  $B_k$  is symmetric indefinite then there are directions of negative curvature.

## Negative Curvature and the Local Model

Suppose  $p_k$  is a direction of negative curvature then

$$p_k^T B_k p_k < 0$$

The local model as a function of  $\alpha$  given  $p_k$

$$m_k(\alpha p_k) = f(x_k) + \alpha \nabla f(x_k)^T p_k + 0.5 \alpha^2 p_k^T B_k p_k$$

is a parabola opening downward and therefore has no bounded minimum.

## Negative Curvature and the Local Model

- local model in the directions  $\pm p_k$  unbounded below
- $f$  in the same directions is assumed to be bounded below, i.e., the minimization problem makes sense
- Some methods use these directions to move away from current neighborhood.
- All can have difficulties since the negative curvature directions may dominate the solution and have large norm, i.e., step selection lengthy
- Easy to detect and most practical Newton and Quasi-Newton methods have strategies to avoid directions of negative curvature (not so easy).

## Newton Methods

- Newton assumes  $\nabla^2 f_k$  is symmetric positive definite, i.e., close enough to local minimizer, and

$$\nabla^2 f_k p_k = -\nabla f_k$$

solved exactly, i.e., to numerical precision.

- Inexact Newton assumes  $\nabla^2 f_k$  is symmetric positive definite solves system, e.g., via CG, inexactly with

$$r_k = \nabla^2 f_k p_k + \nabla f_k \neq 0.$$

- Inexact Newton converges linearly locally if

$$\forall k \quad \|r_k\| \leq \eta_k \|\nabla f\| \quad 0 \leq \eta_k \leq \eta < 1$$

## Practical Methods

- Cannot assume problem away in practice.
- $\nabla^2 f_k$  indefinite implies directions of negative curvature must be avoided to guarantee a descent direction in the line search form.
- One approach is to modify the Hessian. Several techniques exist.
- Or, rather than modifying the matrix, modify the way the system is solved.

## Newton-CG Method

- The solution to  $\nabla^2 f_k p = -\nabla f_k$  is approximated via a CG iteration  $x_0, x_1, \dots, x_i = p_k$  using conjugate directions  $d_0, d_1, \dots, d_i$
- $x_0 = 0$  used and therefore  $d_0 = r_0 = -\nabla f_k$ .
- terminate when
  - (i)  $\|r_k\| \leq \min(0.5, \sqrt{\|\nabla f_k\|}) \|\nabla f_k\|$
  - (ii) or if  $d_i^T \nabla^2 f_k d_i < 0$  for some  $i$  when computing  $x_{i+1}$  then when  $i > 0$  take  $p_k = x_i$  otherwise  $p_k = d_0$ .
- $\alpha_k$  set to satisfy Wolfe or Goldstein with initial guess  $\alpha = 1$  to guarantee superlinear/quadratic convergence ultimately.
- Can be applied to any algorithm that solves  $B_k p_k = -\nabla f_k$ .

## Secant Condition for Scalar Equations

Secant method for a scalar equation:

$$\ell_k(x) = f_k + q_k(x - x_k) \rightarrow q_k(x_{k+1} - x_k) = q_k s_k = -f_k$$

uses slope of line connecting  $(x_{k-1}, f_{k-1})$  and  $(x_k, f_k)$

$$q_k = \frac{f_k - f_{k-1}}{x_k - x_{k-1}} \rightarrow q_k s_{k-1} = y_{k-1} \quad \text{1-D secant condition}$$

note that  $q_k s_k \neq y_k$

$$\begin{aligned} \ell_{k+1}(x) &= f_{k+1} + q_{k+1}(x - x_{k+1}) \rightarrow q_{k+1}(x_{k+2} - x_{k+1}) \\ &= q_{k+1} s_{k+1} = -f_{k+1} \end{aligned}$$

$$q_{k+1} = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} \rightarrow q_{k+1} s_k = y_k \quad \text{1-D secant condition}$$

note that  $q_{k+1} s_{k+1} \neq y_{k+1}$

## Secant Condition for Systems

- local model for systems  $M_k(x) = F(x_k) + B_k(x - x_k)$
- $x_{k+1} = x_k + \alpha_k p_k$  where  $B_k p_k = -F(x_k)$  gives the change to get to the root of  $M_k(x)$  and  $\alpha_k$  is a stepsize.
- Let  $s_k = x_{k+1} - x_k$  and  $y_k = F(x_{k+1}) - F(x_k)$
- As with  $q_k s_{k-1} = y_{k-1}$  for scalars we want  $B_k s_{k-1} = y_{k-1}$ .
- $B_k s_k \neq y_k$ . If not then take  $\alpha_k = 1 \rightarrow p_k = x_{k+1} - x_k = s_k$  and

$$B_k p_k = B_k s_k = y_k = F(x_{k+1}) - F(x_k) = F(x_{k+1}) + B_k p_k$$
$$\therefore F(x_{k+1}) = 0, \quad \text{Solution found.}$$

- On the next step we want  $B_{k+1} s_k = y_k$  etc.

## Secant Condition for Systems

Secant condition:

$$B_{k+1}s_k = y_k$$

- Note that this is underdetermined with respect to the  $n^2$  degrees of freedom in  $B_{k+1}$ .
- Many possible choices of  $B_{k+1}$  at each step.
- Suppose we look for a modification to  $B_k$  that makes  $B_{k+1}$  satisfy the secant condition, i.e.,  $B_{k+1} = B_k + E$
- May also require other conditions, e.g., symmetry.

## Quasi-Newton Methods

- Local model used:

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + p^T B_k p$$

- $B_k p_k = -\nabla f(x_k)$  solve exactly or approximately
- secant condition enforced by update  $B_{k+1} s_k = (B_k + E_k) s_k = y_k$
- symmetry of all  $B_k$  constrains the update, i.e.,  $E_k$  is symmetric.
- alternate equivalent form  $p_k = -H_k \nabla f_k$
- $H_{k+1} = H_k + G_k$  where  $G_k$  is symmetric
- method of avoiding negative curvature varies

## BFGS Method

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is the most popular Quasi-Newton method for unconstrained optimization.

- starts with  $H_0 = B_0^{-1} \approx \nabla^2 f^{-1}(x_0)$
- Secant condition is written accordingly

$$H_{k+1}y_k = s_k$$

- $H_{k+1}$  is taken to be the symmetric matrix such that the secant condition holds and that minimizes

$$\|H - H_k\|_F$$

## BFGS Method

$$\begin{aligned} H_{k+1} &= (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \\ &= H_k + G_k, \quad \rho_k = \frac{1}{y_k^T s_k} \end{aligned}$$

or equivalently 
$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

$$p_k = -H_k \nabla f(x_k) \quad \text{or} \quad B_k p_k = -\nabla f(x_k)$$

If  $H_k$  is positive definite then so is  $H_{k+1}$  (not an imposed constraint)

## BFGS Method

- $B_k p_k = -\nabla f(x_k)$  can be solved with CG version adapted to avoid negative curvature in case  $B_k$  positive definiteness does not survive numerical noise
- the step  $\alpha_k$  is chosen so  $x_{k+1} = x_k + \alpha_k p_k$  satisfies Wolfe conditions
- BFGS tends to be self-correcting when  $H_k$  is not a good approximation.
- limited memory BFGS for large sparse problems
  - $n$  is very large
  - $\nabla^2 f(x_k)$  is symmetric and sparse
  - avoids  $B_{k+1} = B_k + E_k$  or  $H_{k+1} = H_k + G_k$  due to fill-in
- member of much broader Broyden class of methods

## BFGS Method

**BFGS (  $H_k$  update):**

Choose  $H_0, x_0$

loop over  $k$  until convergence

$$\text{Solve } p_k = -H_k \nabla f(x_k)$$

Choose  $\alpha_k$  via a search that imposes Wolfe conditions

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

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\rho_k = 1/y_k^T s_k$$

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

end

## BFGS Method

**BFGS (  $B_k$  update):**

Choose  $B_0, x_0$

loop over  $k$  until convergence

$$\text{Solve } B_k p_k = -\nabla f(x_k)$$

Choose  $\alpha_k$  via a search that imposes Wolfe conditions

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

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

end

## Example – Rosenbrock Function

(Nocedal and Wright)

$$f(x) = 100(\xi_2 - \xi_1^2)^2 + (1 - \xi_1)^2$$

$x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$  is a local minimum and  $\nabla^2 f(x^*)$  is positive definite

$$x_0 = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$$

## Example – Rosenbrock Function

(Nocedal and Wright)

- Steepest Descent, Inexact Newton, and BFGS using Wolfe conditions
- Steepest Descent  $k = 5264$  and  $\|x_k - x^*\|_2 = 1.823 \times 10^{-4}$
- BFGS  $k = 34$  and  $\|x_k - x^*\|_2 = 1.01 \times 10^{-6}$
- Inexact Newton  $k = 21$  and  $\|x_k - x^*\|_2 = 1.17 \times 10^{-8}$

## Convergence

Convergence can be considered in terms of the angles defined by

$$\cos \theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|_2 \|p_k\|_2}$$

**Zoutendijk Condition:** Under mild assumptions it can be shown that if  $\forall k$   $p_k$  is a descent direction and  $\alpha_k$  satisfies the Wolfe or Goldstein conditions then

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|_2^2 < \infty$$

This in turn implies

$$\cos^2 \theta_k \|\nabla f_k\|_2^2 \rightarrow 0$$

## Convergence of Line Search

**Theorem 12.12.** *Suppose  $x_{k+1} = x_k + p_k \alpha_k$  where  $p_k$  is a descent direction and  $\alpha_k$  satisfies the Wolfe conditions and the Zoutendijk condition. If  $p_k$  is chosen so that*

$$\forall k \quad \cos \theta_k \geq \delta > 0$$

*then  $\forall x_0$  we have*

$$\lim_{k \rightarrow \infty} \|\nabla f_k\|_2 = 0.$$

*Note.* The method is therefore globally convergent to a set of stationary points. You could move around in or near the set on a continuum, i.e., no jumps to other distant stationary points.

## Convergence of Newton-like Searches

**Theorem 12.13.** *Suppose  $x_{k+1} = x_k + p_k \alpha_k$  where  $p_k = -B_k^{-1} \nabla f_k$  and  $\alpha_k$  satisfies the Wolfe conditions and the Zoutendijk condition. If  $B_k$  is symmetric positive definite and*

$$\|B_k\|_2 \|B_k^{-1}\|_2 \leq \mu$$

*then  $\forall x_0$  we have*

$$\cos \theta_k \geq \mu^{-1} \quad \text{and} \quad \lim_{k \rightarrow \infty} \|\nabla f_k\|_2 = 0.$$

*Note.* Wolfe conditions, positive definiteness and bounded condition numbers imply globally convergent Newton-like methods. Positive definiteness is the difficult bit to guarantee.

## General Line Search Convergence

**Theorem 12.14.** *Any algorithm for which*

- *every iteration produces a reduction in  $f$ ,*
- *every  $m$ -th iteration is a steepest descent step with  $\alpha_k$  satisfying the Wolfe or Goldstein conditions*

*satisfies*

$$\liminf_{k \rightarrow \infty} \|\nabla f_k\|_2 = 0$$

*Note.* This includes nonlinear CG methods and says that only a subsequence of the gradients converge to 0.

## Sufficient Decrease Condition

Suppose  $0 < \gamma_1 < 1$  is given, then sufficient decrease is achieved with  $\alpha_k$  if

$$f(x_k + p_k \alpha_k) \leq f(x_k) + \gamma_1 \alpha_k (\nabla f_k^T p_k)$$
$$\phi(\alpha_k) \leq \ell(\alpha_k)$$

- negative slope at  $\alpha_k = 0$ ,  $\gamma_1$  flattens it to define linear  $\ell(\alpha)$
- $\gamma_1 \approx 10^{-4}$  typical
- sufficient reduction requires decrease to be proportional to step and the directional derivative
- sufficiently small step sizes satisfy this trivially so the algorithm may not make progress if this is the only condition.

## Sufficient Decrease Condition Geometric View

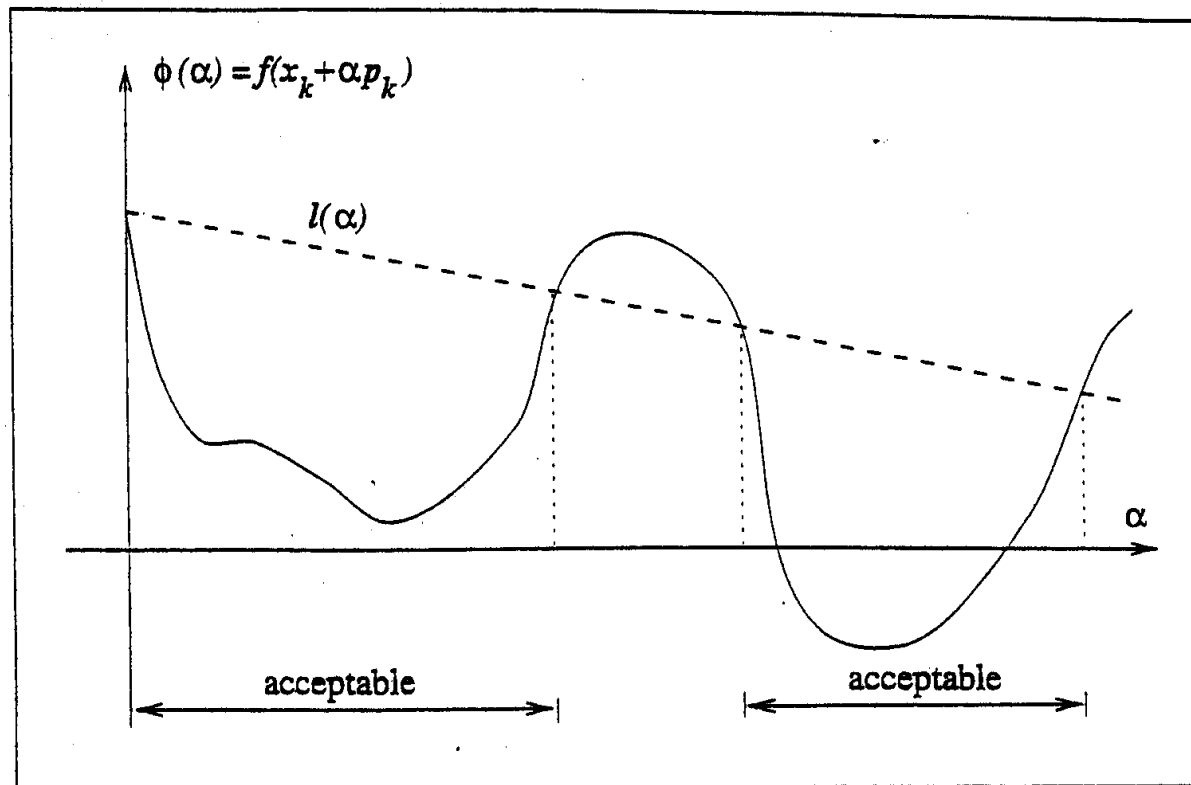


Figure 3.3 Sufficient decrease condition.

(from Nocedal and Wright)

## Curvature Condition

Suppose  $0 < \gamma_1 < \gamma_2 < 1$  is given, then  $\alpha_k$  satisfies the curvature condition if

$$\nabla f(x_k + p_k \alpha_k)^T p_k \geq \gamma_2 \nabla f_k^T p_k$$

$$\phi'(\alpha_k) \geq \gamma_2 \phi'(0)$$

- As we near minimizer the curve  $\phi(\alpha)$  should flatten, i.e., less negative slope or even slightly positive.
- If not, then a significant reduction could be expected with larger stepsizes.
- $10^{-1} \leq \gamma_2 \leq 0.9$  typical
- Note that large positive  $\phi'(\alpha_k)$  satisfies condition and are far from minimizers so a modification needed.

## Curvature Condition Geometric View

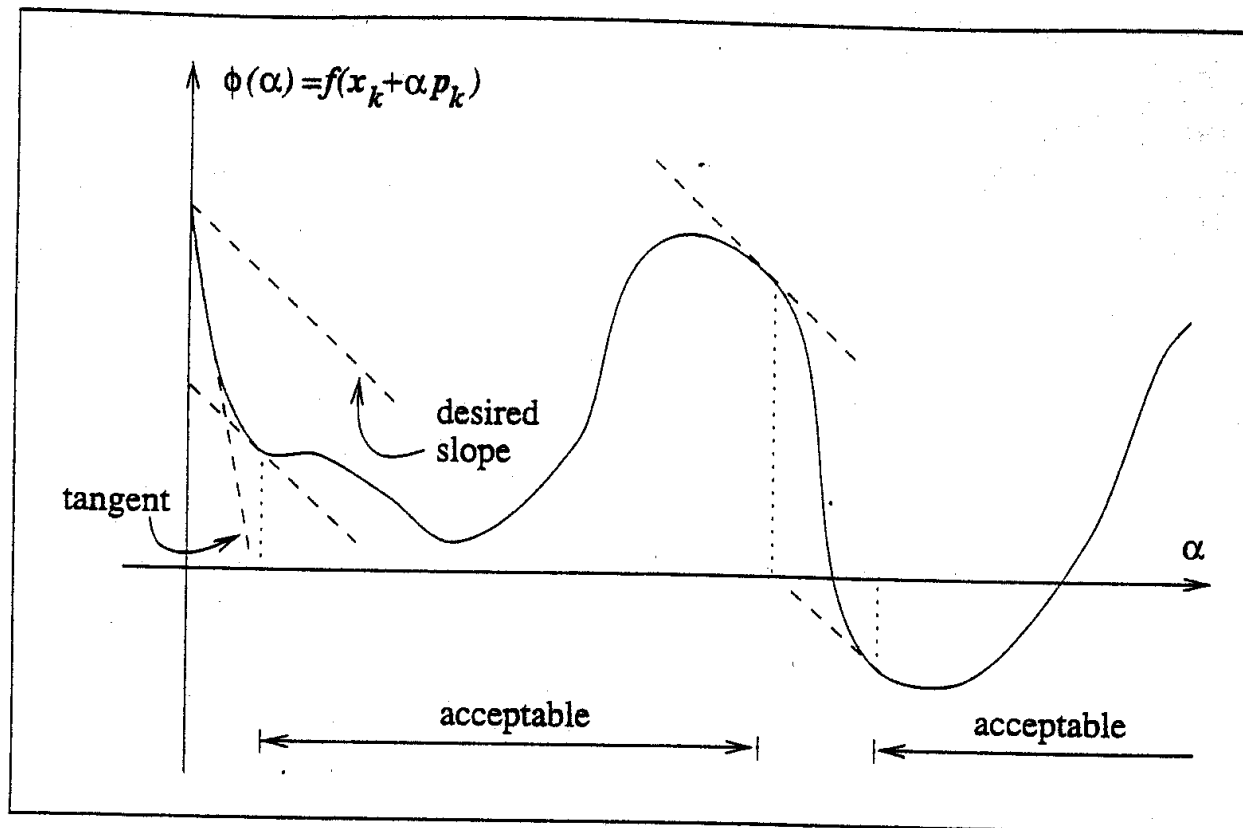


Figure 3.4 The curvature condition.

(from Nocedal and Wright)

## Wolfe Conditions

**Definition 12.3.** Suppose  $f \in \mathcal{C}^1$ ,  $0 < \gamma_1 < \gamma_2 < 1$  are given, and  $p_k \in \mathbb{R}^n$  is a descent direction at  $x_k$ . The step  $\alpha_k$  satisfies the Wolfe conditions if

$$f(x_k + p_k \alpha_k) \leq f(x_k) + \gamma_1 \alpha_k (\nabla f_k^T p_k) \quad (1)$$

$$\nabla f(x_k + p_k \alpha_k)^T p_k \geq \gamma_2 \nabla f_k^T p_k \quad (2)$$

and satisfies the strong Wolfe conditions if (2) is replaced by

$$|\nabla f(x_k + p_k \alpha_k)^T p_k| \leq \gamma_2 |\nabla f_k^T p_k|$$

*Note.* Under mild conditions,  $\alpha_k$  satisfying the Wolfe conditions always exists.

# Wolfe Conditions Geometric View

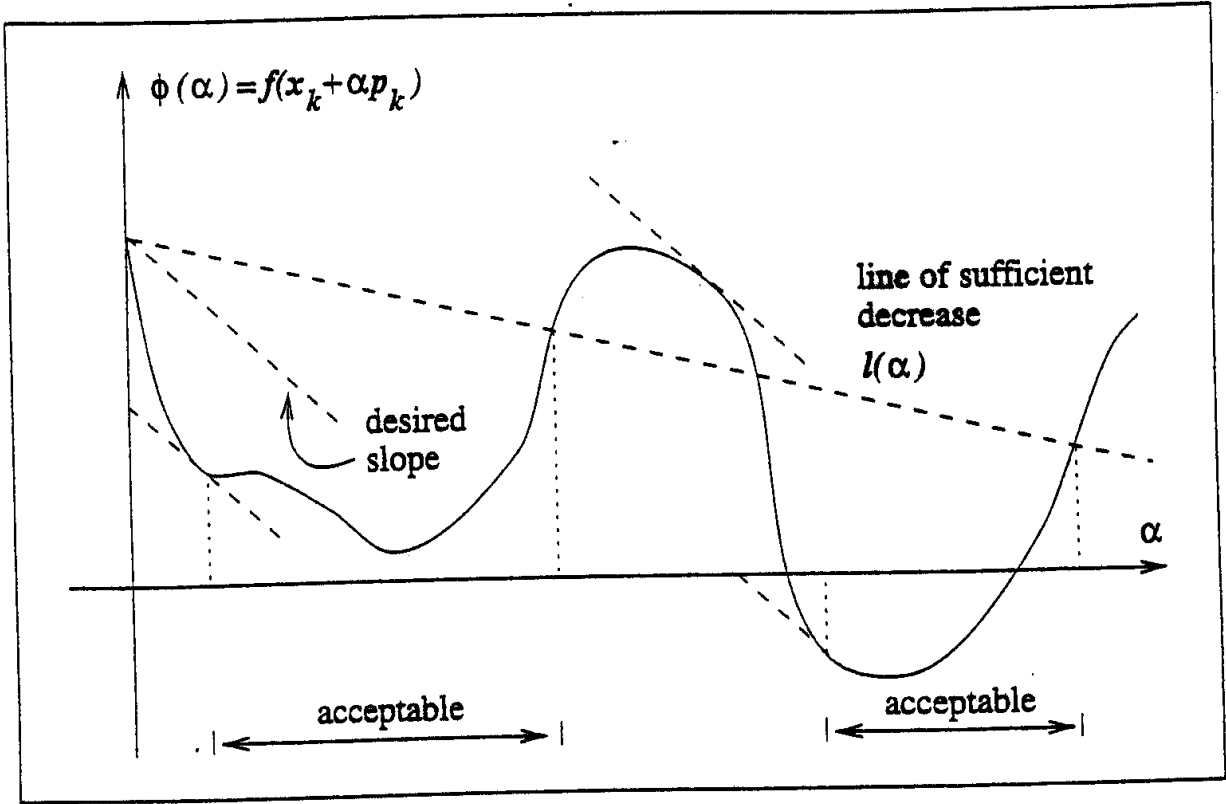


Figure 3.5 Step lengths satisfying the Wolfe conditions.

(from Nocedal and Wright)

## Observations

$$\cos \theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|_2 \|p_k\|_2}$$

- Turning  $p_k$  closer to  $-\nabla f_k$  any time  $0 < \cos \theta_k < \delta$  is a way to guarantee global convergence.
- Global convergence often conflicts with fast convergence – steepest descent can be very slow.
- Newton is fast near the solution when all of its directions are descent directions naturally.
- The key is directions of negative curvature. Sometimes you must include their influence to move rapidly lower in  $f$ . Other times they are misleading since they are purely local and their natural scaling may be excessively large.
- line searches and trust regions deal with negative curvature differently

## Rates of Convergence

**Definition 12.4.** Let  $\{x_k\}$  be a sequence in  $\mathbb{R}^n$  that converges to  $x^*$ . We say that the convergence is:

- Q-linear if  $\exists 0 < \rho < 1$  and  $k_0$  such that  $\forall k > k_0$

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \rho$$

- Q-superlinear if

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$$

- Q-order  $p$  if  $\exists 0 < \rho < 1$  and  $k_0$  such that  $\forall k > k_0$

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} \leq \rho$$

## Convergence Rates

- steepest descent converges Q-linearly and for ill-condition problems  $\rho \approx 1$ .
- Newton's method is locally Q-quadratically convergent.
- Quasi-Newton methods with  $p_k = -B_k^{-1} \nabla f_k$  a descent direction must have  $\alpha_k$  satisfying the Wolfe conditions with  $\gamma_1 \leq 1/2$  for all  $k$  greater than some  $k_0$ .
- When  $\alpha_k = 1$  for  $k > k_0$  then Quasi-Newton methods converge superlinearly to  $x^*$  if and only if

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|}$$

*Note.* We need not have  $B_k \rightarrow \nabla^2 f(x^*)$ . It need only converge in the direction of the search direction  $p_k$ !