Lecture-7

Step Length Selection

Homework (Due 2/20/01)

- 3.1
- 3.2
- 3.5
- 3.6
- 3.7
- 3.9
- 3.10
- Show equation 3.44
- The last step in the proof of Theorem 3.6. (see slides)

Sufficient condition

$$f(x_k + ap_k) \le f(x_k) + c_1 a \nabla f_k^T p_k, \quad c_1 \in (0,1) \qquad c_1 = 10^{-4}$$

$$f(x_k + ap_k) - f(x_k) \le c_1 a \nabla f_k^T p_k, \quad c_1 \in (0,1)$$

The reduction should be proportional to both the step length, and directional derivative.

$$f(x_k + ap_k) \le f(x_k) + c_1 a \nabla f_k^T p_k, \quad c_1 \in (0,1)$$

$$f(x_k + ap_k) \le l(a)$$

St line

Sufficient condition

$$f(x_k + ap_k) \le l(a)$$

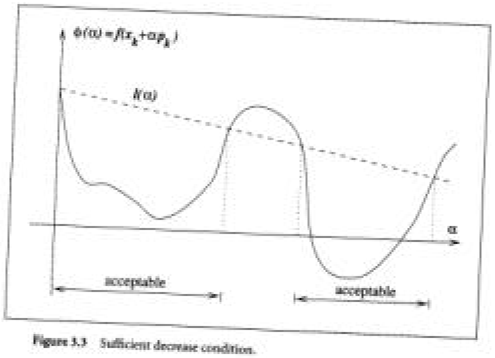


Figure 3.3 Sufficient decrease condition.

Problem:
The sufficient decrease condition is satisfied for all small values of step length

Curvature condition

$$\nabla f(x_k + \alpha p_k)^T p_k \ge c_2 \nabla f_k^T(x_k)\, p_k, \quad c_2 \in (c_1, 1)$$

$c_2 = .9$ for Newton and Quasi - Newton

$c_2 = .1$ for conjugate gradient

Derivative

The slope of is greater than times the gradient .

Curvature condition

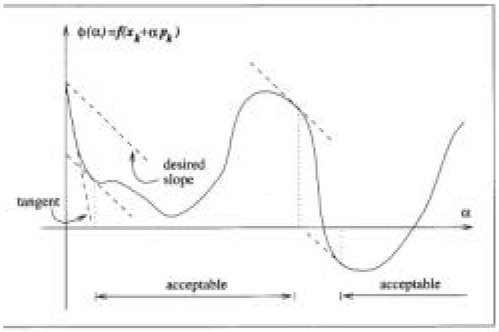


Figure 3.4 The curvature condition.

If the slope is strongly negative, that means we can reduce *f* further along the chosen direction

If the slope is positive, it indicates we can not decrease *f* further in this direction.

Wolfe conditions

$f(x_k + ap_k) \leq f(x_k) + c_1 a \nabla f_k^T p_k, \quad c_1 \in (0,1)$ Sufficient decrease

$\nabla f(x_k + ap_k)^T p_k \geq c_2 \nabla f_k^T (x_k) p_k, \quad c_2 \in (c_1, 1)$ Curvature

Backtracking Line Search

If line search method chooses its step length appropriately, we can dispense with the second condition

```
Choose $\bar{a} > 0,\ r, c \in (0,1);$ set $a \leftarrow \bar{a};$
$repeat$ until $f(x_k + a p_k) \leq f(x_k) + c a \nabla f_k^T p_k$
    $a \leftarrow r a;$
$end(repeat)$
Terminate with $a_k = a$
```

$\bar{a} = 1$, for Newton
and quasi - Newton

This ensures that the step length is short enough to satisfy the sufficient decrease condition, but not too short.

Searching Step Length Using Interpolation

$f(x_k + ap_k) \le f(x_k) + c_1 a \nabla f_k^T p_k, \quad c_1 \in (0,1)$ Sufficient decrease

$$f(x_k) \le f(0) + c_1 a_k f'(0)$$

1. Assume a_0 is the initial guess. Then if we have:

$$f(a_0) \le f(0) + c_1 a_0 f'(0)$$

Then this step length satisfies the condition, we terminate the search.

2. Otherwise, we know $[0, a_0]$ contains the acceptable step lengths.

We fit quadratic polynomial to three pieces of information:

$$f_q(0) = f(0), f_q'(0) = f'(0), f_q(a_0) = f(a_0)$$

Searching Step Length Using Interpolation

and find step length a_1 by analytically minimizing this polynomial

If the sufficient decrease condition is satisfied for this a_1 then we terminate the search.
If not we fit cubic polynomial to interpolate four pieces of information, and analytically minimize this polynomial to find .

$$f_c(0) = f(0), f_c'(0) = f'(0), f_c(a_0) = f(a_0), f_c(a_1) = f(a_1)$$

3. If not we fit cubic polynomial to interpolate four pieces of a_2 information,and analytically minimize this polynomial to find

If necessary we can repeat this process with $f(0), f'(0)$ and two Most recent values of f .

Quadratic Interpolation

$$f_q(\boldsymbol{a}) = a\boldsymbol{a}^2 + b\boldsymbol{a} + c$$

$$f_q(0) = f(0), f'_q(0) = f'(0), f_q(\boldsymbol{a}_0) = f(\boldsymbol{a}_0)$$

$$f_q(\boldsymbol{a}) = \left(\frac{f(\boldsymbol{a}_0) - f(0) - \boldsymbol{a}_0 f'(0)}{\boldsymbol{a}_0^2} \right) \boldsymbol{a}^2 + f'(0)\boldsymbol{a} + f(0)$$

$$\frac{d}{d\boldsymbol{a}} f_q(\boldsymbol{a}) = 2\left(\frac{f(\boldsymbol{a}_0) - f(0) - \boldsymbol{a}_0 f'(0)}{\boldsymbol{a}_0^2} \right) \boldsymbol{a} + f'(0)) = 0$$

$$\boldsymbol{a}_1 = -\left(\frac{f'(0)\boldsymbol{a}_0^2}{2(f(\boldsymbol{a}_0) - f(0) - \boldsymbol{a}_0 f'(0))} \right)$$

Cubic Interpolation

$$f_c(\boldsymbol{a}) = a\boldsymbol{a}^3 + b\boldsymbol{a}^2 + c\boldsymbol{a} + d$$

$$f_c(0) = f(0), f'_c(0) = f'(0), f_c(\boldsymbol{a}_0) = f(\boldsymbol{a}_0), f_c(\boldsymbol{a}_1) = f(\boldsymbol{a}_1)$$

$$f_c(\boldsymbol{a}) = a\boldsymbol{a}^3 + b\boldsymbol{a}^2 + f'(0)\boldsymbol{a} + f(0)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\boldsymbol{a}_0^2 \boldsymbol{a}_1^2 (\boldsymbol{a}_1 - \boldsymbol{a}_0)} \begin{bmatrix} \boldsymbol{a}_0^2 & -\boldsymbol{a}_1^2 \\ -\boldsymbol{a}_0^3 & \boldsymbol{a}_1^3 \end{bmatrix} \begin{bmatrix} f(\boldsymbol{a}_1) - f(0) - f'(0)\boldsymbol{a}_1 \\ f(\boldsymbol{a}_0) - f(0) - f'(0)\boldsymbol{a}_0 \end{bmatrix}$$

$$\boldsymbol{a}_2 = -\left(\frac{-b + \sqrt{b^2 - 3af'(0)}}{3a} \right)$$

Algorithm 3.2 (Line Search Algorithm)

```
Set $a_0 \leftarrow 0$, choose $a_1 > 0$, and $a_{\max}$;
$i \leftarrow 1$
repeat
    Evaulate $f(a_i)$;
    if   $f(a_i) > f(0) + c_1 a_i f'(0)$ or $[f(a_i) > f(a_{i-1}), i > 1]$          1st Wolfe's condition
        $a_* \leftarrow zoom(a_{i-1}, a_i)$, and stop;
    Evaulate $f'(a_i)$;
    if   $|f'(a_i)| \le -c_2 f'(0)$                                                2nd Wolfe's condition
        set $a_* \leftarrow a_i$, and stop;
    if   $f'(a_i) \ge 0$
        set $a_* \leftarrow zoom(a_i, a_{i-1})$, and stop;
    choose $a_{i+1} \in (a_i, a_{\max})$
    $i \leftarrow i + 1$;
end (repeat)
```

Algorithm 3.3 (Zoom)

```
repeat
    Interpolate to find a trial step length
     $a_j$ between $a_{lo}, a_{hi}$;
    Evaulate $f(a_j)$;
    if   $f(a_j) > f(0) + c_1 a_j f'(0)$ or $[f(a_j) > f(a_{lo})]$          1st Wolfe's condition
        $a_{\mathrm{hi}} \leftarrow a_{\mathrm{j}}$;
    esle
        Evaulate $f'(a_j)$;
    if   $|f'(a_i)| \le -c_2 f'(0)$                                         2nd Wolfe's condition
        set $a_* \leftarrow a_j$, and stop;
    if   $f'(a_j)(a_{hi} - a_{lo}) \ge 0$
        set $a_{hi} \leftarrow a_{lo}$
    set $a_{lo} \leftarrow a_j$
end (repeat)
```

CHAPTER 3. LINE SEARCH METHODS

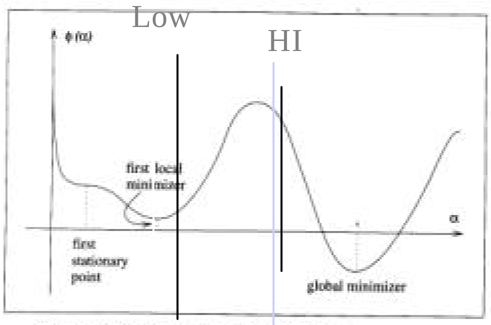


Figure 3.1 The ideal step length is the global minimizer

CHAPTER 3. LINE SEARCH METHODS

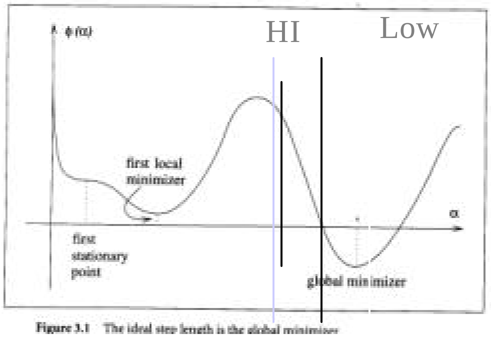


Figure 3.1 The ideal step length is the global minimizer

Theorem 3.5 (Any Descent Direction)

Suppose *f* is three times continuously differentiable. Consider iteration , where is a descent direction, Satisfies Wolfe's conditions, with . If the converges to a point such that and is pd, and if the search direction satisfies

$$\lim_{k \to 0} \frac{\| \nabla f_k + \nabla^2 f_k \, p_k \|}{\| p_k \|} = 0$$

$$\lim_{k \to 0} \frac{\| (B_k - \nabla^2 f(x^*)) p_k \|}{\| p_k \|} = 0$$

Then

(i) is admissible for all $k>k_0$ and

(ii) if for all $k>k_0$, then $\{x_k\}$ converges to x^* superlinearly.

Theorem 3.6 (Quasi-Newton)

Suppose *f* is three times continuously differentiable. Consider iteration , where is given by Quasi-Newton direction. Assume the sequence converges to a point such that and is pd, the converges superlinearly ifif the following condition holds.

$$\lim_{k \to 0} \frac{\| (B_k - \nabla^2 f(x^*)) p_k \|}{\| p_k \|} = 0$$

Order Notations

Given two non-negative infinite sequences

$$\boldsymbol{h}_k = O(\boldsymbol{n}_k)$$
$$if \quad |\boldsymbol{h}_k| \leq C\,|\boldsymbol{n}_k|, \text{ for } C > 0, \forall k$$

$$\boldsymbol{h}_k = o(\boldsymbol{n}_k)$$
$$if \ \lim_{k\to\infty} \frac{\boldsymbol{h}_k}{\boldsymbol{n}_k} = 0$$

Sketch of a Proof

$$\boldsymbol{h}_k = O(\boldsymbol{n}_k)$$
$$if \quad |\boldsymbol{h}_k| \leq C\,|\boldsymbol{n}_k|, \text{for } C > 0$$
$$h_k = o(n_k)$$
$$if \ \lim_{k\to\infty} \frac{h_k}{n_k} = 0$$

$$\begin{aligned} p_k - p_k^N &= \nabla^2 f_k^{-1}(\nabla^2 f_k\, p_k + \nabla f_k) \\ &= \nabla^2 f_k^{-1}(\nabla^2 f_k - B_k)\, p_k \\ &= O(\|(\nabla^2 f_k - B_k)\, p_k\|) \\ &= o(\|p_k\|) \end{aligned}$$

$$\lim_{k\to 0} \frac{\|(B_k - \nabla^2 f(x^*))\, p_k\|}{\|p_k\|} = 0$$

$$\lim_{k\to 0} \frac{\|\nabla f_k + \nabla^2 f_k p_k\|}{\|p_k\|} = 0$$

Norm of Hessian is bounded.

Sketch of a Proof

$$\| x_k + p_k - x^* \| = \| x_k + p_k^N - p_k^N + p_k - x^* \| \le \| x_k + p_k^N - x^* \| + \| p_k - p_k^N \|$$

$$= \mathrm{O}(\| x_k - x^* \|^2) + o(\| p_k \|)$$

$$\| x_k + p_k - x^* \| \le o(\| x_k - x^* \|)$$

$h_k = o(n_k)$

$if \ \lim_{k\to\infty} \frac{h_k}{n_k} = 0$

Theorem 3.7

Super-linear

Show this in Homework

Theorem 3.7 (Newton)

Suppose that f is twice differentiable and that Hessian is Lipschitze continuous. Consider the iteration where p_k is given by

$$p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$$

Then:

1. If the starting point x_0 is sufficiently close to x^*, the sequence converges to x^*.

2. The rate of convergence is quadratic
3. The sequence of gradient norms converges quadratically to zero.

Coordinate Descent Method

Cycle through *n* coordinate directions $e_1, e_2, \dots e_n$ using each in turn as a search direction.

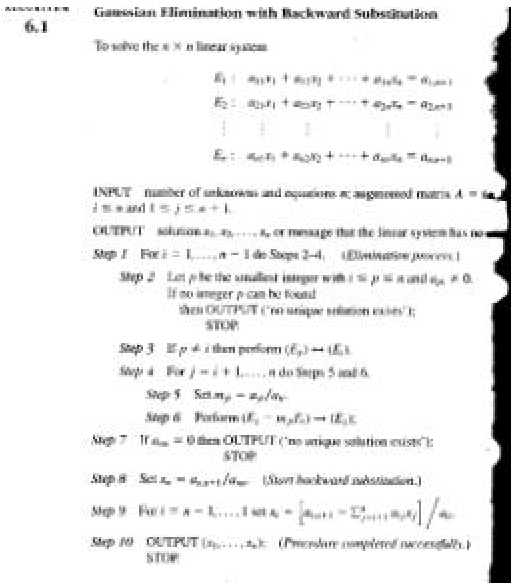
Fix all other variables except one, and minimize the function.

It is an inefficient method, it can iterate infinitely without ever approaching a point, where the gradient vanishes.

The gradient may become more and more perpendicular to search directions, making $\cos \boldsymbol{q}$ approach to zero, but not the gradient.

Solution of A linear System

- Gaussian Elimination, Backward Substitution
- Matrix Factorization
- Iterative Techniques



ALGORITHM 6.1

Gaussian Elimination with Backward Substitution

To solve the $n \times n$ linear system

$$
\begin{aligned}
E_1 &: a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1} \\
E_2 &: a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1} \\
&\ \vdots \\
E_n &: a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_{n,n+1}
\end{aligned}
$$

INPUT number of unknowns and equations n; augmented matrix $A =$
$i \le n$ and $1 \le j \le n+1$.

OUTPUT solution $x_1, x_2, \ldots, x_n$ or message that the linear system has no

```
Step 1  For $i = 1, \ldots, n-1$ do Steps 2–4.  (Elimination process.)
    Step 2  Let $p$ be the smallest integer with $i \le p \le n$ and $a_{pi} \ne 0$.
            If no integer $p$ can be found
              then OUTPUT ('no unique solution exists');
                STOP.
    Step 3  If $p \ne i$ then perform $(E_p) \leftrightarrow (E_i)$.
    Step 4  For $j = i+1, \ldots, n$ do Steps 5 and 6.
        Step 5  Set $m_{ji} = a_{ji}/a_{ii}$.
        Step 6  Perform $(E_j - m_{ji}E_i) \rightarrow (E_j)$;
Step 7  If $a_{nn} = 0$ then OUTPUT ('no unique solution exists');
                STOP.
Step 8  Set $x_n = a_{n,n+1}/a_{nn}$.  (Start backward substitution.)
Step 9  For $i = n-1, \ldots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] \Big/ a_{ii}$.
Step 10 OUTPUT $(x_1, \ldots, x_n)$;  (Procedure completed successfully.)
        STOP.
```

Iterative Methods for Solving Linear Systems

- For large sparse system Gaussian Elimination and Backward substitution is not suitable.
- Approximate solution using iterative methods

Jacobi

$$x_i^k = \frac{\sum\limits_{j=1,\ j\neq i}^{n} (-a_{ij} x_j^{k-1}) + b_i}{a_{ii}} \quad \text{for } i = 1, 2, \ldots , n$$

Gauss-Seidel

$$x_i^k = \frac{-\sum\limits_{j=1}^{i-1} (a_{ij} x_j^k) - \sum\limits_{j=i+1}^{n} (a_{ij} x_j^{k-1}) + b_i}{a_{ii}} \quad \text{for } i = 1,2, \ldots , n$$

SOR (Successive Over Relaxation)

$$x_i^{k} = (1 - w)\, x_i^{k-1} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k} - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) \right]$$

$$w > 1$$