Lecture - 8
NONLINEAR PROGRAMMING WITH NO CONSTRAINTS

Reference: Book Chapter 12

INTRODUCTION

- Nonlinear programming is a technique to solve optimization problems that include **nonlinear expressions**, either objective functions or constraints, or both.
- The methods presented in this chapter generally deal with unconstrained optimization problems with unimodal objectives.
- Unconstrained nonlinear programming has many practical applications. Some design problems do not have constraints, or the constraints for many problems are negligible.
- Constrained optimization problems can be reformulated as unconstrained problems.

NECESSARY AND SUFFICIENT CONDITIONS

First-Order Necessary Conditions

- *If $x*$ is a <u>local minimum</u> and $f(x)$ is continuously differentiable in an open neighborhood of $x*$, then the gradient $\nabla f(x*) = 0$.*

- Taylor expansion

$$f(x^* + \Delta x) = f(x^*) + \nabla f(x^*)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^*) \Delta x + O_3(\Delta x)$$

- Ignore the higher order terms

$$\Delta f(x) = f(x^* + \Delta x) - f(x^*) = \nabla f(x^*)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^*) \Delta x$$

- If $x*$ is a local minimum, any other points in its neighborhood should produce a greater objective value.

$$\Delta f(x) = f(x^* + \Delta x) - f(x^*) \geq 0$$

- In order that the sign of f(x) be known for arbitrary values of Δx, <u>the first derivative of f(x) should be zero</u>. Otherwise, f(x) can be forced to be positive or negative by changing the sign of Δx.

Second-Order Necessary Conditions

- *If $x*$ is a local minimum of $f(x)$ and $\nabla^2 f(x)$ exists and is continuous in an open neighborhood of $x*$, then $\nabla f(x*) = 0$ and $\nabla^2 f(x*)$ is positive semidefinite.*

- $\nabla f(x*) = 0$ at the local minimum, then

$$\Delta f(x) = f(x^* + \Delta x) - f(x^*) = \frac{1}{2}\Delta x^T \nabla^2 f(x^*)\Delta x$$

- It should satisfy $\Delta f(x) \geq 0$, since $x*$ is a local minimum. <u>Then $\nabla^2 f(x*)$ should be positive semidefinite</u>.

Second-Order Sufficient Conditions

- For f(x), suppose that $\nabla^2 f(x)$ is continuous in an open neighborhood of $x*$, and that $\nabla f(x*) = 0$. If $\nabla^2 f(x*)$ is positive definite, $x*$ is a strict local minimum of f(x).

- If the Hessian of f(x) is positive definite at the point where $\nabla f(x) = 0$, then

$$\Delta f(x) = f(x^* + \Delta x) - f(x^*) = \frac{1}{2}\Delta x^T \nabla^2 f(x^*) \Delta x > 0$$

- <u>It guarantees the point, $x*$, is a strict local minimum.</u>

SINGLE VARIABLE OPTIMIZATION

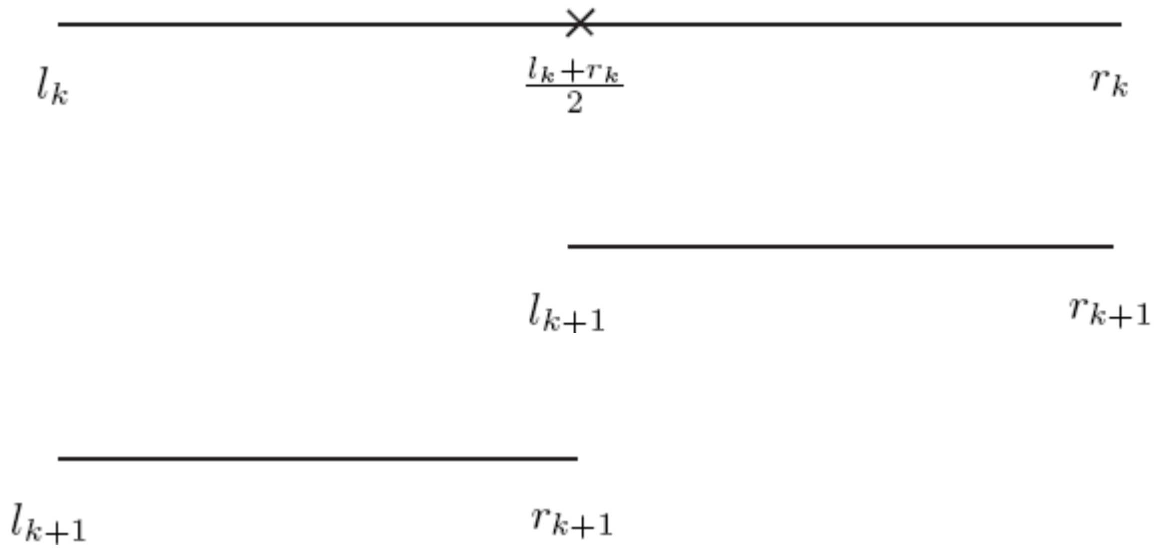
- Interval reduction Method
    - Bisection
    - Golden Section
- Polynomial Approximations
    - Quadratic Approximation

BISECTION

- The bisection method enables us to find an extreme point in a bounded region for a unimodal single-variable function.
- The function and its first derivative are assumed continuous.
- It successively halves the interval, and decides which one of the two half intervals the extreme point exists.
- The bisection method uses **the first derivative** of the function to determine which half the extreme point lies in.
- Suppose the variable x of a unimodal convex function f(x) is inside the interval [a, b]. The function f(x) and its derivative f′(x) are continuous over this interval.
- The first derivatives at the two ends satisfy the condition $f'(a)f'(b) < 0$. This condition implies that there is a minimum with its derivative at 0.

BISECTION

➢*Interval updates*



$f'(l_k)f'(x_k) > 0$

$f'(l_k)f'(x_k) < 0$

BISECTION

➢*Procedure*

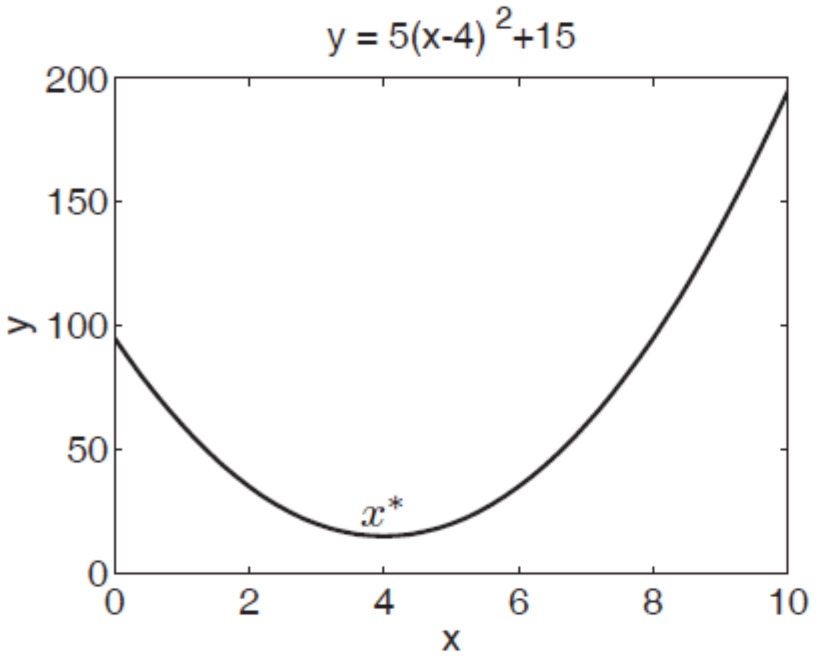
1. Specify the convergence tolerance $\varepsilon > 0$. Specify the convergence tolerance of the gradient $\gamma > 0$. Set the number of iterations, k, as 0. The bounds are $l_0 = a$, and $r_0 = b$.
2. If $\mathbf{r_k - l_k < \varepsilon}$, **stop**. The midpoint $x_k = (l_k+r_k)/2$ is taken as the minimum, $x*$, and $f(x*)$ is the optimal solution.
3. Evaluate the gradient at the middle point, $f'(x_k)=f'((l_k+r_k)/2)$.
4. If $\mathbf{||f'(x_k)|| < \gamma}$, **stop**. The x_k is taken as the minimum, $x*$, and the corresponding function value $f(x*)$ is the optimal solution.
5. Evaluate the product $f'(l_k)f'(x_k)$. If it is negative, $l_{k+1} = l_k$ and $r_{k+1} = x_k$. If it is positive, $l_{k+1} = x_k$ and $r_{k+1} = r_k$.
6. $k = k + 1$. Go to step 2.

Example:

$$\min_{x} f(x) = 5(x-4)^2 + 15$$

subject to

$$0 \le x \le 10$$



Bi section

$f(x) = 5(x-4)^2 + 15 \qquad 0 \leq x \leq 10$

$f'(x) = 10(x-4) \qquad \text{set: } \varepsilon = 0.1$

$k = 0$

$l_0 = 0 \; ; \; r_0 = 10$

$x_0 = \frac{0+10}{2} = 5 \qquad \| f'(5) \| = 10$

$f'(l_0) f'(x_0) = -40 \times 10 = -400 < 0$

$l_1 = l_0 = 0 \; ; \; r_1 = x_0 = 5$

$k = 1$

$r_1 - l_1 = 5 > \varepsilon$

$f'(x_1) = f'\left(\frac{0+5}{2}\right) = -15$

$f'(l_1) f'(x_1) = -40 \times (-15) = 600 > 0$

$l_2 = x_1 = 2.5 \; ; \; r_2 = r_1 = 5$

$k = 2$

$r_2 - l_2 = 2.5 > \varepsilon$

$f'(x_2) = f'\left(\frac{2.5+5}{2}\right) = -2.5$

$f'(l_2) f'(x_2) = -15 \times (-2.5) = \qquad > 0$

$l_3 = x_2 = 3.75 \; ; \quad r_3 = x_2 = 5$

$k = 3$

Example:

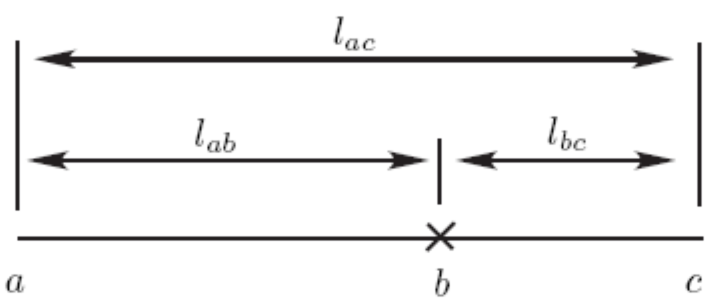
- The convergence tolerance of the gradient at middle points is set as 0.01.

	a_k	b_k	Derivative
1	0.0000	5.0000	-15.0000
2	2.5000	5.0000	-2.5000
3	3.7500	5.0000	3.7500
4	3.7500	4.3750	0.6250
5	3.7500	4.0625	-0.9375
6	3.9063	4.0625	-0.1563
7	3.9844	4.0625	0.2344
8	3.9844	4.0234	0.0391
9	3.9844	4.0039	-0.0586
10	3.9941	4.0039	-0.0098

- The optimal value is at the middle point 3.9990. The optimal function value is $f(x*) = 15.0000$.

GOLDEN SECTION

- The golden section search is a method to minimize or maximize a **<u>unimodal function of one variable</u>**.
- It does <u>not require the information of the derivatives</u>.
- Suppose we know the minimum exists inside a region of the variable, this method successively narrows the range of function values inside which the minimum exists.
- In mathematics, two quantities are in the golden ratio, <u>if the ratio of the sum of the quantities to the larger quantity is equal to the ratio of the larger quantity to the smaller</u>.



$$\varphi \equiv \frac{l_{ab} + l_{bc}}{l_{ab}} = \frac{l_{ab}}{l_{bc}}$$

$$\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.618$$

$$\tau \equiv \frac{1}{\varphi} = \frac{-1 + \sqrt{5}}{2} \approx 0.618$$

GOLDEN SECTION

- Suppose f(x) is inside the interval [a, b], and f(x) is continuous.

  1. Choose $\varepsilon > 0$ as the convergence tolerance of interval. Set $k = 0$. The bounds are $a_0 = a$ and $b_0 = b$. Set $l_0 = b_0 - \tau(b_0 - a_0)$, and $r_0 = a_0 + \tau(b_0 - a_0)$.

  2. **If $b_k - a_k < \varepsilon$**, **stop**. The midpoint $x_k = (a_k+b_k)/2$ is taken as the optimal value, x^*, and $f(x^*)$ is the optimal solution.

GOLDEN SECTION

- Suppose f(x) is inside the interval [a, b], and f(x) is continuous.

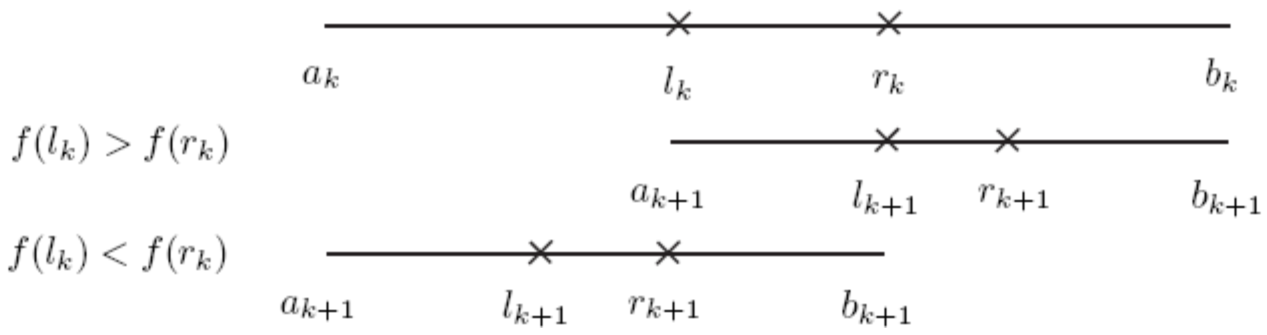
  3. **If $f(l_k) > f(r_k)$**, the parameters are updated as follows.

     $a_{k+1} = l_k$; $b_{k+1} = b_k$; $l_{k+1} = r_k$; $r_{k+1} = a_{k+1} + \tau(b_{k+1} - a_{k+1})$.

  4. **If $f(l_k) < f(r_k)$**, the parameters are updated as follows.

     $a_{k+1} = a_k$; $b_{k+1} = r_k$; $r_{k+1} = l_k$; $l_{k+1} = b_{k+1} - \tau(b_{k+1} - a_{k+1})$

  5. $k = k + 1$, go to Step 2.



- The length of the interval is reduced by τ at each iteration.

GOLDEN SECTION

Example:

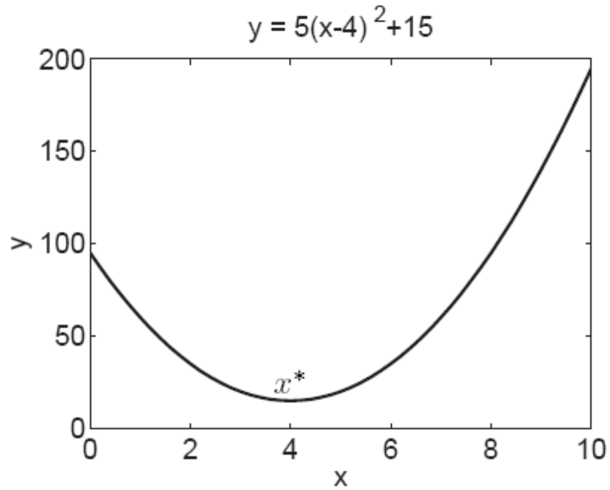
$$\min_x f(x) = 5(x-4)^2 + 15$$

subject to

$$0 \leq x \leq 10$$

- The convergence tolerance, ε, is set as 0.1.

	a_k	b_k	$f(a_k)$	$f(b_k)$	l_k	r_k	$f(l_k)$	$f(r_k)$
1	0.00	10.00	95.00	195.00	3.82	6.18	15.16	38.77
2	0.00	6.18	95.00	38.77	2.36	3.82	28.44	15.16
3	2.36	6.18	28.44	38.77	3.82	4.72	15.16	17.60
4	2.36	4.72	28.44	17.60	3.26	3.82	17.72	15.16
5	3.26	4.72	17.72	17.60	3.82	4.16	15.16	15.13
6	3.82	4.72	15.16	17.60	4.16	4.38	15.13	15.71
7	3.82	4.38	15.16	15.71	4.03	4.16	15.01	15.13
8	3.82	4.16	15.16	15.13	3.95	4.03	15.01	15.01
9	3.95	4.16	15.01	15.13	4.03	4.08	15.01	15.03
10	3.95	4.08	15.01	15.03	4.00	4.03	15.00	15.01



- At the 10th iteration, l_{10} = 4.00 and r_{10} = 4.03. The optimal value is between the points, 3.95 and 4.03.
- x^* = (3.95+4.03)/2 = 3.99. $f(x^*)$ = 15.00.

Golden Section

$f(x)=5(x-4)^2+15 \quad 0\le x\le 10$

Set: $\varepsilon=0.1$

$k=0$ $\quad a_0=0 \quad b_0=10 \quad l_0=10-0.618\times 10=3.82$

$r_0=0+0.618\times 10=6.18$

$b_0-a_0=10>\varepsilon$

$f(l_0)=5\times 0.18^2+15=15.16$ $\quad f(l_0)<f(r_0)$

$f(r_0)=5\times 2.18^2+15=38.77$

$a_1=a_0=0, \quad b_1=r_0=6.18; \quad r_1=l_0=3.82$

$l_1=b_1-0.618\times(b_1-a_1)$

$=6.18-0.618\times 6.18$

$=2.36$

$k=1.$ $\quad b_1-a_1=6.18>\varepsilon$

$f(l_1)=f(2.36)=5\times(2.36-4)^2+15=28.45$

$f(r_1)=f(3.82)=5\times(3.82-4)^2+15=15.16$

$f(l_1)>f(r_1)$

$a_2=l_1=2.36; \quad b_2=b_1=6.18$

$l_2=r_1=3.82; \quad r_2=a_2+\tau(b_2-a_2)$

$=2.36+0.618\times(6.18-2.36)$

$=4.72$

$k=2$ $\quad b_2-a_2=6.18-2.36>\varepsilon$

QUADRATIC APPROXIMATION

- Assume an objective function is **unimodal and continuous inside an interval**, then it can be approximated by a quadratic approximation.
- It consists of a sequence of interval reducing and iterative approximation in the reduced intervals.
- A quadratic approximating function is constructed, given three consecutive points and their corresponding function values.

$$\tilde{f}(x) = c_0 + c_1(x - x_1) + c_2(x - x_1)(x - x_2)$$

$$f(x_1) = \tilde{f}(x_1) = c_0 \qquad c_0 = f(x_1)$$

$$f(x_2) = \tilde{f}(x_2) = c_0 + c_1(x_2 - x_1) \qquad c_1 = \frac{f(x_2) - c_0}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$f(x_3) = \tilde{f}(x_3) = c_0 + c_1(x_3 - x_1) + c_2(x_3 - x_1)(x_3 - x_2)$$

$$c_2 = \frac{f(x_3) - c_0 - c_1(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} = \frac{1}{x_3 - x_2}\left(\frac{f(x_3) - f(x_1)}{x_3 - x_1} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}\right)$$

QUADRATIC APPROXIMATION

- Using the quadratic approximation, the minimum point, x^*, should satisfy the first-order necessary conditions.

$$\frac{d\tilde{f}(x)}{dx} = c_1 + c_2(x - x_1) + c_2(x - x_2) = 0$$

$$x^* = \frac{x_1 + x_2}{2} - \frac{c_1}{2c_2}$$

QUADRATIC APPROXIMATION

➢ *Procedure*

1. Define the tolerance of function value difference, $\gamma > 0$, and the tolerance of the variable, $\varepsilon > 0$. Set Δx. Set $k = 1$.

2. Set the initial point, x^1_1 . Compute $x^1_2 = x^1_1 + \Delta x$. Evaluate $f(x^1_1)$ and $f(x^1_2)$.

   If $f(x^1_1) > f(x^1_2)$, then $x^1_3 = x^1_2 + \Delta x$.
   If $f(x^1_1) < f(x^1_2)$, then $x^1_3 = x^1_1 - \Delta x$.
   Evaluate $f(x^1_3)$.

3. Compare the function values at the three points, x^k_1 , x^k_2 , and x^k_3. Find out the minimum $f^k_{min} = \min\{f(x^k_1), (x^k_2), f(x^k_3)\}$, and the corresponding point x^k_{min}.

QUADRATIC APPROXIMATION

➢ *Procedure*

4. Using the three points, x^k_1 , x^k_2 , and x^k_3, construct a quadratic approximation. Compute the minimum point, x^*_k. Evaluate $f(x^*_k)$.

5. If $|| f(x^*_k) - f^k_{min} || < \gamma$ and $|| x^*_k - x^k_{min} || < \varepsilon$, take x^*_k as the minimum. Terminate the iteration.

6. Take the current **best point $\mathbf{x^*_k}$** and the two points bracketing it, as the three points for the next quadratic approximation. $k = k + 1$. Go to <u>Step 3</u>.

QUADRATIC APPROXIMATION

Example:

- The variable, x, is inside (1.001, 10).

$$\text{Minimize } f(x) = x^2 - 2x + \frac{8}{x-1} + 6$$

- Define the tolerance of function value difference as 0.01 || $f(x^*_k)$ ||, and the tolerance of the variable as 0.05 || x^*_k || . Set Δ x = 1. Set the initial point, $x^1_1 = 2$.

$$\delta f^k = \frac{\|f(x_k^*) - f_{min}^k\|}{\|f(x_k^*)\|} \qquad \delta x^k = \frac{\|x_k^* - x_{min}^k\|}{\|x_k^*\|}$$

k	x_1^k	$f(x_1^k)$	x_2^k	$f(x_2^k)$	x_3^k	$f(x_3^k)$	x_{min}^k	f_{min}^k	x_k^*	$f(x_k^*)$	δf^k	δx^k
1	2	14	3	13	4	16.67	3	13	2.71	12.61	0.03	0.11
2	2	14	2.71	12.61	3	13	2.71	12.61	2.65	12.57	0.002	0.02

- At the end of the second iteration, the errors satisfy the tolerances. The optimal point is 2.65 and the optimal function value is 12.57.

Quadratic Appoximation

$$f(x) = x^2 - 2x + \frac{8}{x-1} + 6 \qquad x \in (1.001, 10)$$
$$\Delta x = 1$$

$k=1$: $x_1^1 = 2$; $x_2^1 = 3$

$f(x_1^1) = 4-4+8+6 = 14$; $f(x_2^1) = 9-6+4+6 = 13$

$f(x_1^1) > f(x_2^1)$

$\Rightarrow x_3^1 = x_2^1 + \Delta x = 3+1 = 4$

$f(x_3^1) = 16 - 8 + \frac{18}{3} + 6 = 16.67$

$f_{min}^1 = \min(f(x_1^1), f(x_2^1), f(x_3^1)) = 13$

$x_{min}^1 = 3$

$$c_1^1 = \frac{f(x_2^1) - f(x_1^1)}{x_2^1 - x_1^1} = \frac{13-14}{3-2} = -1$$

$$c_2^1 = \frac{1}{x_3^1 - x_2^1}\left(\frac{f(x_3^1) - f(x_1^1)}{x_3^1 - x_1^1} - \frac{f(x_2^1) - f(x_1^1)}{x_2^1 - x_1^1}\right)$$

$$= \frac{1}{4-3} \times \left(\frac{16.67 - 14}{4-2} - \frac{13-14}{3-2}\right)$$

$$= 2.335$$

$$x_1^* = \frac{x_1^1 + x_2^1}{2} - \frac{c_1^1}{2c_2^1} = \frac{2+3}{2} - \frac{-1}{2 \times 2.335} = 2.71$$

$$f(x_1^*) = f(2.71) = 2.71^2 - 2 \times 2.71 + \frac{8}{1.71} + 6 = 12.61$$

$$\delta f^1 = \frac{\| f(x_1^*) - f_{min}^1 \|}{\| f(x_1^*) \|} = \frac{\|12.61 - 13\|}{\|12.61\|} = 0.03 > 0.01$$

$$\delta x^1 = \frac{\| x_1^* - x_{min}^1 \|}{\| x_1^* \|} = \frac{\|2.71 - 3\|}{2.71} = 0.11 > 0.05$$

for Engineering Students and Professionals

$k=2$

$x^2_1 = 2 \quad x^2_2 = 2.71 \quad x^2_3 = 3$

$f(x^2_1) = 4-4+8+6 = 14$

$f(x^2_2) = f(2.71) = \underline{12.61}$

$f(x^2_3) = f(3) = 13$

$f^2 min = 12.61 \quad x^2 min = 2.71$

$$x^*_2 = \frac{x^2_1 + x^2_2}{2} - \frac{C^2_1}{2C^2_2} = 2.65$$

$f(x^*_2) = f(2.65) = 12.57$

$$\delta f^2 = \frac{\| f(x^*_2) - f^2 min \|}{\| f(x^*_2) \|} = \frac{12.57 - 12.61}{12.57} = 0.003 < 0.01$$

$$\delta x^2 = \frac{\| x^*_2 - x^2 min \|}{\| x^*_2 \|} = \frac{\| 2.65 - 2.71 \|}{2.65} = 0.02 < 0.05$$

STOP.

MULTIVARIABLE OPTIMIZATION

- Zeroth order methods
  - Simplex Search Method
  - Pattern Search Methods
- First order methods
  - Steepest Descent
  - Conjugate Gradient Methods
- Second order methods
  - Newton Methods
  - Quasi-Newton Methods

SIMPLEX SEARCH METHOD

- It is a **direct-search method**, and does not require function derivatives.
- In geometry, a simplex is a triangle in two-dimensional space, and a tetrahedron in three dimensional space. In n-dimensional space, it is an n-dimensional polytope.
- At each iteration, based on the comparison of function values at all vertex, one vertex is projected through the centroid of the other vertices at a suitable distance.
- At the beginning, the simplex search method sets up a regular simplex in the space of the variables.
- The objective function is evaluated at each vertex. For minimization problem, the vertex with the highest function value is the one that should be reflected through the centroid of the other vertices to generate a new point.
- The point and the remaining vertices construct the next simplex.

SIMPLEX SEARCH METHOD

- At the beginning, given a base point, x^0, and a scale factor, α. The variable vector has N dimensions. Two increments are defined:

$$\delta_1 = \left[\frac{(N+1)^{1/2}-1}{N\sqrt{2}}\right]\alpha \qquad \delta_2 = \left[\frac{(N+1)^{1/2}+N-1}{N\sqrt{2}}\right]\alpha$$

- The j^{th} dimension for the i^{th} vertex:

$$x_j^{(i)} = \begin{cases} x_j^{(0)} + \delta_1 & \text{if } j \neq i \\ x_j^{(0)} + \delta_2 & \text{if } j = i \end{cases}$$

- At each iteration, the vertex with the highest function value is selected, and is reflected through the centroid of the remaining points.

$$x_{centroid} = \frac{1}{N}\sum_{\substack{i=0 \\ i\neq j}}^{N} x^{(i)}$$

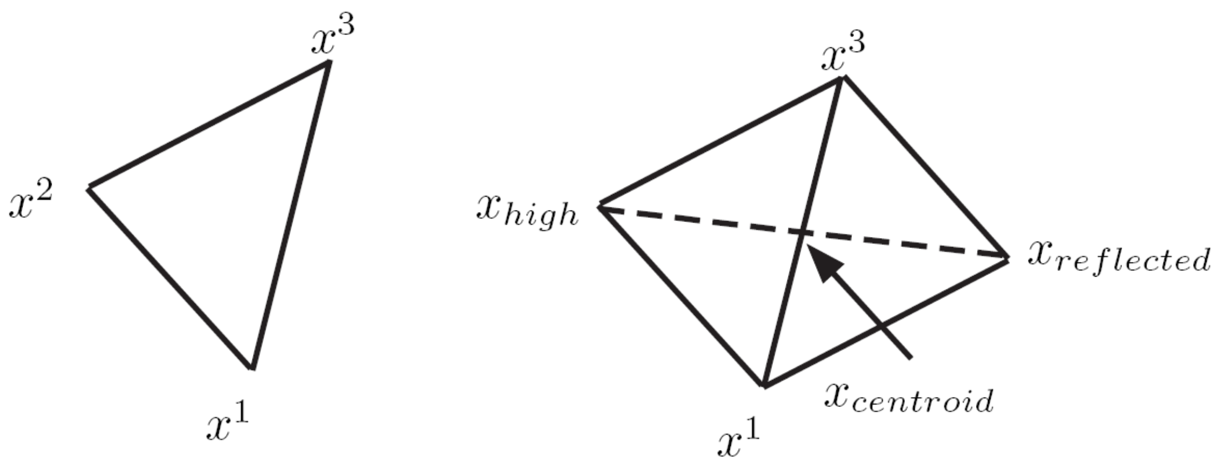
- The line passing x_{high} and $x_{centroid}$ is given by

$$x = x_{high} + \lambda(x_{centroid} - x_{high})$$

SIMPLEX SEARCH METHOD

- The selection of λ can yield the desirable point on the line. In order to generate a symmetric reflection, λ is set as 2.

$$x_{reflected} = 2x_{centroid} - x_{high}$$



SIMPLEX SEARCH METHOD

- ***During the optimization, the following two situations may happen.***

1. At the current iteration, the vertex with the highest function value is the reflected point generated in the last iteration. Choose instead the vertex with the next highest function value, and generate a reflected point.

2. The iterations can cycle between two or more simplexes. If a vertex remains in the simplex for more than M iterations, set up a new simplex with the lowest point as the base point, and reduce the size of the simplex. The number of the dimension for the variable is N. The value of M can be estimated by $M = 1.65N + 0.05N^2$, and M is rounded to the nearest integer.

It is terminated when the size of the simplex is small enough or other termination criteria are met. The vertex with the lowest function value is taken as the minimum.

SIMPLEX SEARCH METHOD

Example: $\text{Minimize } f(x) = (x_1 - 2)^2 + (x_2 - 3)^2$

- $x_0 = [1, 1]^T$. $\alpha = 2$.

$$\delta_1 = \left[\frac{(2+1)^{1/2} - 1}{2\sqrt{2}}\right] \times 2 = 0.5176 \qquad \delta_2 = \left[\frac{(2+1)^{1/2} + 2 - 1}{2\sqrt{2}}\right] \times 2 = 1.9318$$

$$x^1 = [1 + 1.9318, 1 + 0.5176]^T = [2.9318, 1.5176]^T$$

$$x^2 = [1 + 0.5176, 1 + 1.9318]^T = [1.5176, 2.9318]^T$$

$$f(x^0) = 5 \qquad f(x^1) = 3.0658 \qquad f(x^2) = 0.2374$$

- X^0 is reflected to form a new simplex.

$$x_{centroid} = \frac{1}{2}(x^1 + x^2) = [2.2247, 2.2247]^T$$

$$x_{reflected} = 2x_{centroid} - x^0 = [3.4494, 3.4494]^T$$

- $f(x_{reflected}) = 2.3027$. Iterations continue until the minimum is found.

$\boxed{x^* = [2, 3]^T.\ f(x^*) = 0}$

PATTERN SEARCH METHODS

- The pattern search methods look along certain **specified directions**, and **evaluate the objective function at a given step length along each of these directions**.
- These points form a frame around the current iteration.
- Depending on whether any point within the pattern has a lower objective function value than the current point, the frame shrinks or expands in the next iteration. The search stops after a minimum pattern size is reached.
- An important problem of the pattern search methods is to choose the search direction set, D_k, at each iteration k. A key condition is that at least one direction in this set should give a descent direction for the objective function.
- δ is some positive constant. A search direction p should satisfy the following inequality,

$$\cos\theta = \frac{-\nabla f_k^T p}{\|\nabla f_k\| \|p\|} \geq \delta$$

PATTERN SEARCH METHODS

- A **sufficient decrease functio**n ρ(l) is used to determine the convergence of the results.
- It is a function of the step length, and its domain is [0,∞). It is an increasing function of l, and the function values are positive.
- As l approaches to 0, the limit of ρ(t)/t is 0.
- <u>If the decrease of the objective function value is less than the value of the sufficient decrease function, the pattern search is converged to the minimum</u>.

Procedure

1. Define the sufficient decrease function $\rho(l)$. Choose convergence tolerance l_{tol}, contraction parameter θ_{max}, and search direction set D_k. Choose initial point x_0, initial step length l_0, and initial direction set D_0. $k = 1$.
2. Evaluate the objective function value $f(x_k)$.
3. If the step length $\mathbf{l_k \leq l_{tol}}$, take $\mathbf{x_k}$ as the optimal point. **Stop**.

➢*Procedure*

4. **If $f(x_k + l_k p_k) < f(x_k) - \rho(l_k)$** for some $p_k \in D_k$, **set x_{k+1} as $x_k + l_k p_k$** for some $p_k \in D_k$, and increase the step length for the next iteration.

5. If **$f(x_k + l_k p_k) \geq f(x_k) - \rho(l_k)$** for all **$p_k \in D_k$**, use the same point x_k as the point x_{k+1} for the next iteration. Reduce the step length for the next iteration, l_{k+1}. Set l_{K+1} as $\theta_k l_k$, where $0 < \theta_k \leq \theta_{max} < 1$.

6. Set k = k + 1. Go to step 2.

PATTERN SEARCH METHODS

Example:

$$f(x) = \frac{1}{2}x^T Qx - c^T x$$

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

- $\rho(l) = 10l^{3/2}$.
- The contraction factor is 0.5.
- The convergence tolerance is 0.001.
- The contraction factor, $\theta_{max} = 0.5$.
- The initial step length, $l_0 = 0.5$, $x_0 = [0, 0, 0]^T$.
- p_k: $\{[1,0,0]^T, [0,1,0]^T, [0,0,1]^T, [-1,0,0]^T, [0,-1,0]^T, [0,0,-1]^T\}$
- $f(x_0) = 0$. $f(x_0 + l_0 p_k)$ at ($k = 1, ..., 6$):
  4.25, 4.875, 4.625, −3.75, −4.125, and −3.375.

$$\rho(l_0) = \rho(0.5) = 10 \times 0.5^{3/2} = 3.53$$

Pattern Search.

$$f(x) = \frac{1}{2}\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} -8 & -9 & -8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} 2x_1 & 3x_2 & 5x_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 8x_1 + 9x_2 + 8x_3$$

$$= \frac{1}{2}(2x_1^2 + 3x_2^2 + 5x_3^2) + 8x_1 + 9x_2 + 8x_3$$

$$f(x_0) = 0$$

$$f(x_0 + l_0 p_1) = 0.25 + \cancel{8}\,4 = 4.25$$

$$f(x_0 + l_0 p_2) = \frac{3}{2} \times 0.25 + 4.5 = 4.875$$

$$f(x_0 + l_0 p_3) = \frac{5}{2} \times 0.25 + 4 = 4.625$$

$$f(x_0 + l_0 p_4) = 0.25 - 4 = -3.75$$

$$f(x_0 + l_0 p_5) = \frac{3}{2} \times 0.25 - \frac{9}{2} = -4.125$$

$$f(x_0 + l_0 p_6) = \frac{5}{2} \times 0.25 - 4 = -3.375$$

$$P(l_0) = 3.53$$

PATTERN SEARCH METHODS

Example:

$$f(x) = \frac{1}{2}x^T Q x - c^T x$$

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

- Two of them (k = 4, 5) satisfy the <u>sufficient decrease equation</u>,**$f(x_k+l_k p_k) < f(x_k)-\rho\ (l_k)$.**
- Set the point, $x_0+l_0 p_5$, as x_1, and increase the step length. The iterations continue. $x^*=[-4,-3,-1.6]^T$. $f(x^*)=-35.9$

Example:

$$f(x) = \frac{1}{2}x^T Q x - c^T x$$

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

- Use Matlab to solve the problem.

Main Function

```
clc
clear
X0 = [0; 0; 0];
[x,fopt]=patternsearch(@obj_fun,X0)
```

Objective Function

```
function f = obj_fun(x)
Q = [2 0 0; 0 3 0; 0 0 5];
c = [-8; -9; -8];
f = 0.5*x'*Q*x-c'*x;
```

- Result: $x^* = \begin{bmatrix} -4 \\ -3 \\ -1.6000 \end{bmatrix}$ $\qquad f(x^*) = -35.9$

MULTIVARIABLE OPTIMIZATION

- Zeroth order methods
  - Simplex Search Method
  - Pattern Search Methods
- First order methods
  - Steepest Descent
  - Conjugate Gradient Methods
- Second order methods
  - Newton Methods
  - Quasi-Newton Methods

STEEPEST DESCENT

- The steepest descent method is **a first-order optimization algorithm**.
- In each iteration, it takes the negative direction of the gradient as the descent direction of the objective function.
- The method takes steps along the descent direction to find a local minimum.

STEEPEST DESCENT

➢***Procedure***

1. Choose a starting point x_0. Set $k = 0$.
2. Check the conditions of $f(x_k)$. If x_k is the minimum, stop.
3. Calculate the descent direction. $p_k = -\nabla f(x_k)$
4. Evaluate **the step length, $\boldsymbol{\alpha_k}$**. The step length, α_k, should be an acceptable solution to the following minimization problem.

$$\min_{\alpha_k > 0} f(x_k + \alpha_k p_k)$$

5. Update the value of the variable, x_{k+1}, using <u>the descent direction and the step length</u>. $x_{k+1} = x_k + \alpha_k p_k$
6. Set $k = k + 1$, go to Step 2.

STEEPEST DESCENT

Example:

$$f(x) = \frac{1}{2}x^TQx - c^Tx$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \qquad c = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

- The steepest descent direction

$$p_k = -\nabla f(x_k) = -(Qx_k - c)$$

- The steepest descent direction step length

$$\alpha_k = -\frac{\nabla f(x_k)^T p_k}{p_k^T Q p_k}$$

STEEPEST DESCENT

Example:

$$f(x) = \frac{1}{2}x^T Q x - c^T x$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \qquad c = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

- The steepest descent direction

$$x_0 = (0,0,0)^T \qquad f(x_0) = 0 \qquad \nabla f(x_0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \|\nabla f(x_0)\| = 1.7321$$

$$x_1 = \begin{bmatrix} -0.1875 \\ -0.1875 \\ -0.1875 \end{bmatrix} \quad f(x_1) = -0.2813 \quad \nabla f(x_1) = \begin{bmatrix} 0.8125 \\ 0.0625 \\ -0.8750 \end{bmatrix} \|\nabla f(x_1)\| = 1.1957$$

- After 36 iterations:

$$x^* = \begin{bmatrix} -0.9993 \\ -0.2000 \\ -0.0999 \end{bmatrix} \qquad f(x^*) = -0.65$$

Steepest Descent

$$f(x) = \frac{1}{2} x^T Q x - c^T x$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \qquad c = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \qquad \gamma = 0.01$$

$$\nabla f(x_k) = Q x_k - c$$

$$p_k = -\nabla f(x_k) = -Q x_k + c$$

$$\alpha_k = -\frac{\nabla f(x_k)^T p_k}{p_k^T Q p_k}$$

$$x_0 = (0, 0, 0)^T \quad f(x_0) = 0 \quad \nabla f(x_0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$p_0 = -\nabla f(x_0) = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \qquad \|\nabla f(x_0)\| = 1.732 > \gamma$$

$$\alpha_0 = -\frac{\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}}{\begin{bmatrix} -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}} = \frac{3}{16} = 0.1875$$

$$x_1 = x_0 + \alpha_0 p_0 = (0, 0, 0)^T + 0.1875 \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.1875 \\ -0.1875 \\ -0.1875 \end{bmatrix}$$

$$k = 1 \quad f(x_1) = -0.2813$$

$$\nabla f(x_1) = \begin{bmatrix} 0.8125 \\ 0.0625 \\ -0.8750 \end{bmatrix} \qquad \|\nabla f(x_1)\| = 1.1957 > \gamma$$

$$\alpha_1 = 0.1715$$

$$x_2 = x_1 + \alpha_1 p_1 = \begin{bmatrix} -0.3269 \\ -0.1982 \\ -0.0374 \end{bmatrix}$$

CONJUGATE GRADIENT METHODS

- Iterative methods are also a valuable tool for solving linear equations. When **the matrix A is symmetric and positive definite**, the **conjugate gradient method can be used to solve Ax =** b.

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

- The first-order necessary conditions require $\nabla f(x) = Ax - b = 0$. The Hessian matrix $\nabla^2 f(x) = A$ is positive definite, and the sufficient conditions for a local minimum are satisfied.
- This optimization problem is **equivalent to Ax = b**.
- The vectors p_i are conjugate with respect to A.

$$p_i^T A p_j = 0, \text{ if } i \neq j$$

- Any set of conjugate vectors is also linearly independent.
- The residual is defined as **r = Ax – b**.

CONJUGATE GRADIENT METHODS

➢*Procedure*

1. Set an initial guess x_0. The residual is $r_0 = Ax_0 - b$. A vector with the same dimension of the **conjugate vector** $p_0 = -r_0$. Specify the convergence tolerance $\varepsilon > 0$. Set $i = 0$.
2. Check the value of the residual $||r_i||$. If $\boldsymbol{||r_i|| < \varepsilon}$, **stop**.
3. Set $\alpha_i = r^T_i\, r_i / p^T_i\, Ap_i$.
4. Evaluate the next point $x_{i+1} = x_i + \alpha_i p_i$.
5. Evaluate the residual $r_{i+1} = r_i + \alpha_i Ap_i$.
6. Set $\beta_{i+1} = r^T_{i+1}\, r_{i+1} / r^T_i\, r_i$.
7. Set the conjugate vector $p_{i+1} = -r_{i+1} + \beta_{i+1} p_i$.
8. Set $i = i + 1$. Go to <u>Step 2</u>

CONJUGATE GRADIENT METHODS

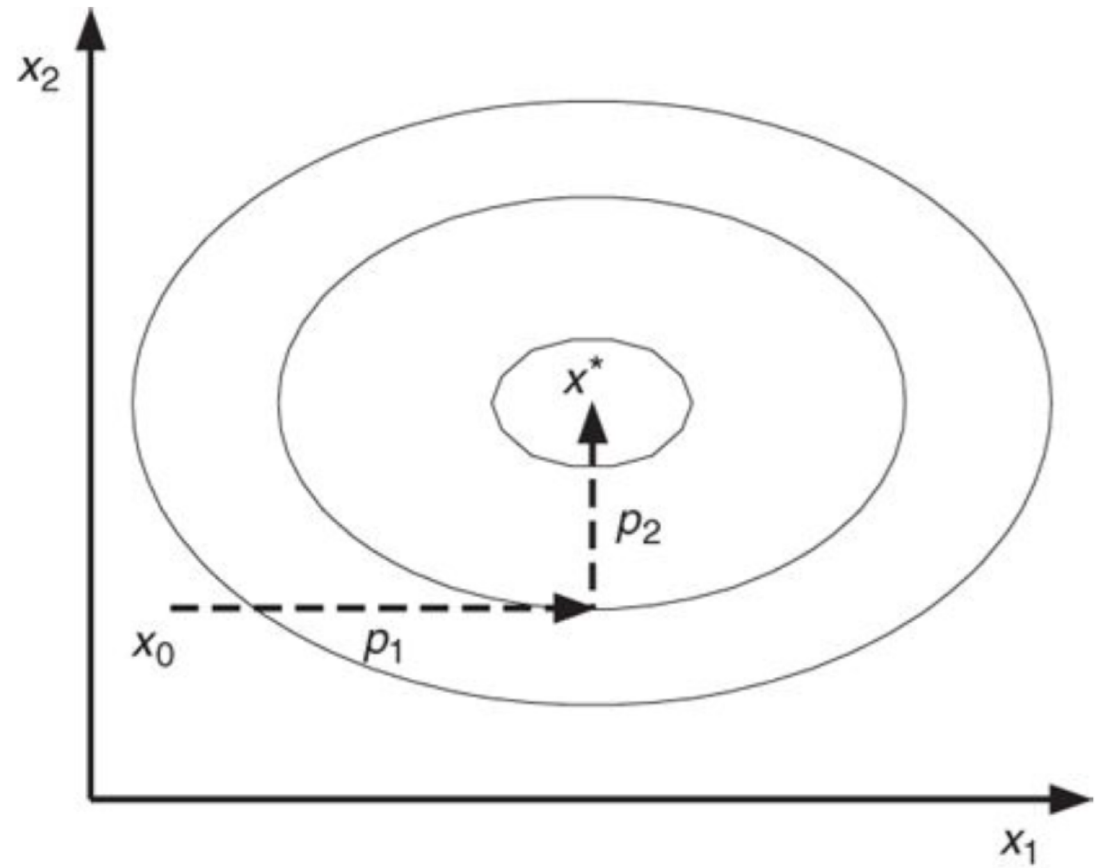


Figure 12.6. Conjugate Directions for a Diagonal Matrix A

CONJUGATE GRADIENT METHODS

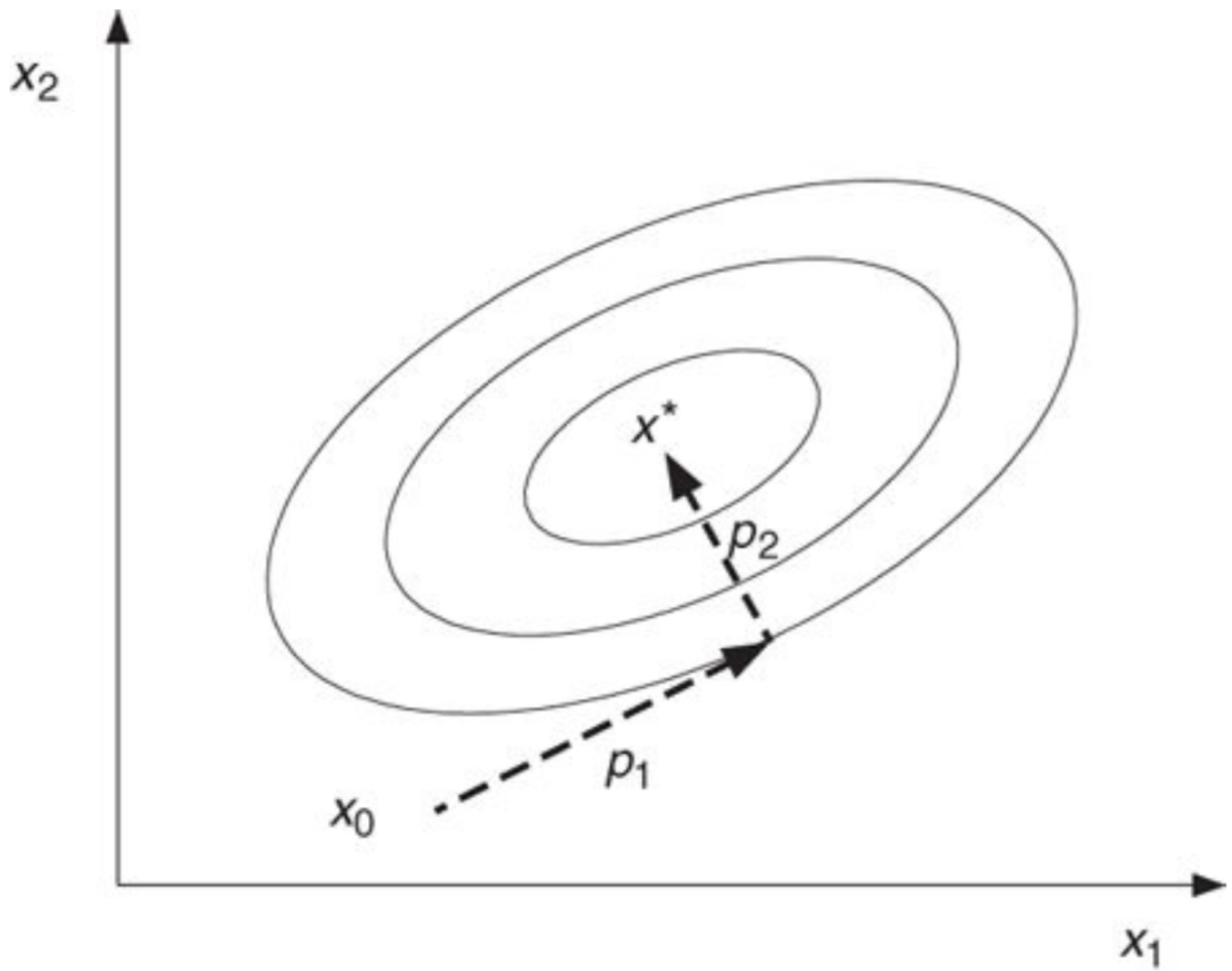


Figure 12.7. Conjugate Directions for a Non-diagonal Matrix A

Example:

- a quadratic problem $f(x) = (1/2)x^TQx - c^Tx$.

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

- $\varepsilon = 0.01$. $x_0 = (0, 0, 0)^T$. $r_0 = (8, 9, 8)^T$.

k	α_k	x_k^T	r_k^T	$\|r_k\|$	β_k	p_k^T
0		$(0,0,0)$	$(8,9,8)$	14.46		
1	0.30	$(-2.42,-2.72,-2.42)$	$(3.16,0.83,-4.10)$	5.24	0.13	$(-4.21,-2.02,3.05)$
2	0.29	$(-3.65,-3.31,-1.53)$	$(0.70,-0.93,0.35)$	1.22	0.05	$(-0.93,0.82,-0.19)$
3	0.38	$(-4.00,-3.00,-1.60)$	$(0,0,0)$	0	0	$(0,0,0)$

- The point $x_3 = (-4.00, -3.00, -1.60)^T$ is the optimal solution, and $f(x_3) = -35.9$

Conjugate Gradient

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

$$f(x) = \frac{1}{2} x^T A x - b^T x \qquad b = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

$\underline{i=0}$

$\varepsilon = 0.01 \quad x_0 = (0, 0, 0)^T, \quad r_0 = A x_0 - b = (8, 9, 8)^T$

$\|r_0\| = 14.46 > \varepsilon \qquad p_0 = -r_0 = (-8, -9, -8)^T$

$$\alpha_0 = r_0^T r_0 / p_0^T A p_0$$

$$= \frac{\cancel{(8,9,8)}\ 64+81+64}{(-8,-9,-8)\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}\begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}} = \frac{64+81+64}{\cancel{2\times 64+81}\ 2\times 64 + 3\times 81 + 5\times 64} = 0.30$$

$x_1 = x_0 + \alpha_0 p_0 = (-2.4, -2.7, -2.4)$

$$r_1 = r_0 + \alpha_0 A p_0 = (8, 9, 8)^T + 0.3 \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}\begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

$$= (8, 9, 8)^T + 0.3\,(-16, -27, -40)^T$$

$$= (3.2, 0.9, -4)^T$$

$\|r_1\| = 5.2 > \varepsilon$

$$\beta_1 = r_1^T r_1 / r_0^T r_0 = \frac{3.2^2 + 0.81 + 16}{64+81+64} = 0.13$$

$$p_1 = -r_1 + \beta_1 p_0 = -(3.2, 0.9, -4)^T + 0.13(-8, -9, -8)^T$$

$$= (-4.24, -2.07, 2.96)^T$$

$\underline{i=1} \quad \alpha_1 = r_1^T r_1 / p_1^T Q p_1 = \_\_\_\_\_\_$

FLETCHER-REEVES METHOD

➢ ***General unconstrained nonlinear optimization.***

The Fletcher-Reeves method:

(1) the one-dimensional minimum along each conjugate vector is identified by a line search;

(2) the residual is replaced by the gradient of the nonlinear function.

FLETCHER-REEVES METHOD

➢ *Procedure*

1. Set an initial guess x_0. At x_0, evaluate the function, $f(x_0)$, and its gradient, $\nabla f(x_0)$. Set $p_0 = -\nabla f(x_0)$. Specify the convergence tolerance of the residual, $\varepsilon > 0$. $i = 0$.
2. If $\mathbf{||\nabla f(x_i)||<\varepsilon}$, **stop**.
3. Use a line search along the direction of the conjugate vector to determine the minimum $x_{i+1} = x_i + \alpha_i p_i$.
4. Evaluate the gradient $\nabla f(x_{i+1})$.
5. Set $\beta_{i+1} = \nabla f(x_{i+1})^T \nabla f(x_{i+1}) / \nabla f(x_i)^T \nabla f(x_i)$.
6. Set the conjugate vector $p_{i+1} = -\nabla f(x_{i+1}) + \beta_{i+1} p_i$.
7. Set $i = i + 1$. Go to Step 2.

MULTIVARIABLE OPTIMIZATION

- Zeroth order methods
  - Simplex Search Method
  - Pattern Search Methods
- First order methods
  - Steepest Descent
  - Conjugate Gradient Methods
- Second order methods
  - Newton Methods
  - Quasi-Newton Methods

NEWTON METHOD

- Newton's method approximates the function f(x) with a quadratic function at each iteration. The approximated quadratic function is minimized exactly, and a descent direction is evaluated.

$$\tilde{f}(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

- The first derivative should satisfy $\nabla \tilde{f}(x) = 0$

$$\nabla \tilde{f}(x) = \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0$$

$$\nabla^2 f(x_k)(x - x_k) = -\nabla f(x_k).$$

$$x = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

NEWTON METHOD

- The second term is used as the descent direction to calculate x_{k+1}

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

- The descent direction

$$p_k = [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

- Look for the search direction p_k that satisfies

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

- P_k can be solved by Gaussian Elimination, or other suitable methods.

Example 1D:

Example 5.12 Find the minimum of the function

$$f(\lambda) = 0.65 - \frac{0.75}{1+\lambda^2} - 0.65\lambda \tan^{-1}\frac{1}{\lambda}$$

using the Newton–Raphson method with the starting point $\lambda_1 = 0.1$. Use $\varepsilon = 0.01$ in Eq. (5.66) for checking the convergence.

SOLUTION The first and second derivatives of the function $f(\lambda)$ are given by

$$f'(\lambda) = \frac{1.5\lambda}{(1+\lambda^2)^2} + \frac{0.65\lambda}{1+\lambda^2} - 0.65\tan^{-1}\frac{1}{\lambda}$$

$$f''(\lambda) = \frac{1.5(1-3\lambda^2)}{(1+\lambda^2)^3} + \frac{0.65(1-\lambda^2)}{(1+\lambda^2)^2} + \frac{0.65}{1+\lambda^2} = \frac{2.8-3.2\lambda^2}{(1+\lambda^2)^3}$$

Example 1D:

Iteration 1

$$\lambda_1 = 0.1, \quad f(\lambda_1) = -0.188197, \quad f'(\lambda_1) = -0.744832, \quad f''(\lambda_1) = 2.68659$$

$$\lambda_2 = \lambda_1 - \frac{f'(\lambda_1)}{f''(\lambda_1)} = 0.377241$$

Convergence check: $|f'(\lambda_2)| = |-0.138230| > \varepsilon$.

Iteration 2

$$f(\lambda_2) = -0.303279, \quad f'(\lambda_2) = -0.138230, \quad f''(\lambda_2) = 1.57296$$

$$\lambda_3 = \lambda_2 - \frac{f'(\lambda_2)}{f''(\lambda_2)} = 0.465119$$

Convergence check: $|f'(\lambda_3)| = |-0.0179078| > \varepsilon$.

Iteration 3

$$f(\lambda_3) = -0.309881, \quad f'(\lambda_3) = -0.0179078, \quad f''(\lambda_3) = 1.17126$$

$$\lambda_4 = \lambda_3 - \frac{f'(\lambda_3)}{f''(\lambda_3)} = 0.480409$$

Convergence check: $|f'(\lambda_4)| = |-0.0005033| < \varepsilon$.

Since the process has converged, the optimum solution is taken as $\lambda^* \approx \lambda_4 = 0.480409$.

NEWTON METHOD

Example 2D:

$$f(x) = 4x_1^2 + 2x_1x_2 + 2.5x_2^2$$

$$\nabla f(x) = \begin{bmatrix} 8x_1 + 2x_2 \\ 2x_1 + 5x_2 \end{bmatrix} \qquad \nabla^2 f(x) = \begin{bmatrix} 8 & 2 \\ 2 & 5 \end{bmatrix} \qquad [\nabla^2 f(x)]^{-1} = \begin{bmatrix} \frac{5}{36} & -\frac{1}{18} \\ -\frac{1}{18} & \frac{2}{9} \end{bmatrix}$$

- Choose an initial point $x_0 = (10, 10)^T$

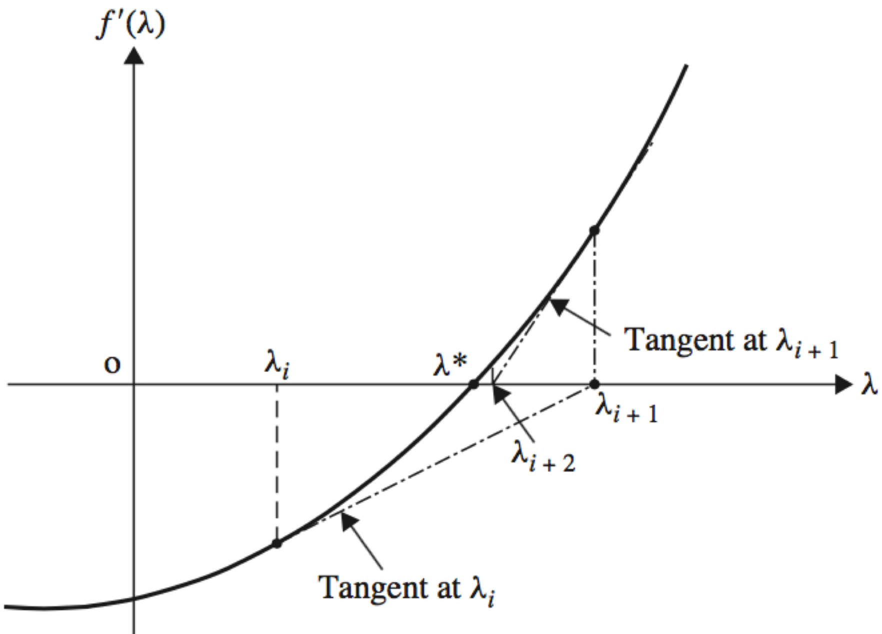
$$x_1 = x_0 - [\nabla^2 f(x_0)]^{-1} \nabla f(x_0) = (0, 0)^T$$

- Check $|\nabla f(x_i)| < \varepsilon$?

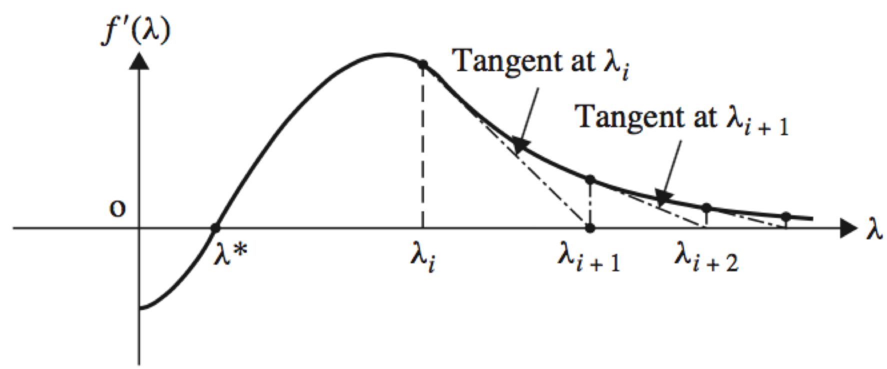
- The point $x_1 = (0, 0)^T$ is the optimal solution, and $f(x_1) = 0$.

NEWTON METHOD

- The Newton method was originally developed by Newton for solving nonlinear equations and later refined by Raphson, and hence the method is also known as Newton–Raphson method in the literature of numerical analysis.
- If the starting point for the iterative process is not close to the true solution, the Newton iterative process might diverge.



convergence



divergence

QUASI-NEWTON METHODS

- Quasi-Newton methods are among the most widely used methods for nonlinear optimization.
- If the function being minimized f (x) is not available in closed form or is difficult to differentiate, the derivatives f ′(x) and f ″(x) can be approximated.

QUASI-NEWTON METHODS

For one-dimensional nonlinear optimization problems

$$f'(\lambda_i) = \frac{f(\lambda_i + \Delta\lambda) - f(\lambda_i - \Delta\lambda)}{2\Delta\lambda}$$

$$f''(\lambda_i) = \frac{f(\lambda_i + \Delta\lambda) - 2f(\lambda_i) + f(\lambda_i - \Delta\lambda)}{\Delta\lambda^2}$$

$$\lambda_{i+1} = \lambda_i - \frac{\Delta\lambda[f(\lambda_i + \Delta\lambda) - f(\lambda_i - \Delta\lambda)]}{2[f(\lambda_i + \Delta\lambda) - 2f(\lambda_i) + f(\lambda_i - \Delta\lambda)]}$$

To test the convergence of the iterative process, the following criterion can be used:

$$|f'(\lambda_{i+1})| = \left| \frac{f(\lambda_{i+1} + \Delta\lambda) - f(\lambda_{i+1} - \Delta\lambda)}{2\Delta\lambda} \right| \leq \varepsilon$$

Example 1D:

$$f(\lambda) = 0.65 - \frac{0.75}{1+\lambda^2} - 0.65\lambda \tan^{-1}\frac{1}{\lambda}$$

using quasi-Newton method with the starting point $\lambda_1 = 0.1$ and the step size $\Delta\lambda = 0.01$ in central difference formulas. Use $\varepsilon = 0.01$ in Eq. (5.70) for checking the convergence.

SOLUTION

Iteration 1

$$\lambda_1 = 0.1, \quad \Delta\lambda = 0.01, \quad \varepsilon = 0.01, \quad f_1 = f(\lambda_1) = -0.188197,$$

$$f_1^+ = f(\lambda_1 + \Delta\lambda) = -0.195512, \quad f_1^- = f(\lambda_1 - \Delta\lambda) = -0.180615$$

$$\lambda_2 = \lambda_1 - \frac{\Delta\lambda(f_1^+ - f_1^-)}{2(f_1^+ - 2f_1 + f_1^-)} = 0.377882$$

Convergence check:

$$|f'(\lambda_2)| = \left| \frac{f_2^+ - f_2^-}{2\Delta\lambda} \right| = 0.137300 > \varepsilon$$

Example 1D:

Iteration 2

$$f_2 = f(\lambda_2) = -0.303368, \quad f_2^+ = f(\lambda_2 + \Delta\lambda) = -0.304662,$$

$$f_2^- = f(\lambda_2 - \Delta\lambda) = -0.301916$$

$$\lambda_3 = \lambda_2 - \frac{\Delta\lambda(f_2^+ - f_2^-)}{2(f_2^+ - 2f_2 + f_2^-)} = 0.465390$$

Convergence check:

$$|f'(\lambda_3)| = \left|\frac{f_3^+ - f_3^-}{2\Delta\lambda}\right| = 0.017700 > \varepsilon$$

Iteration 3

$$f_3 = f(\lambda_3) = -0.309885, \quad f_3^+ = f(\lambda_3 + \Delta\lambda) = -0.310004,$$

$$f_3^- = f(\lambda_3 - \Delta\lambda) = -0.309650$$

$$\lambda_4 = \lambda_3 - \frac{\Delta\lambda(f_3^+ - f_3^-)}{2(f_3^+ - 2f_3 + f_3^-)} = 0.480600$$

Convergence check:

$$|f'(\lambda_4)| = \left|\frac{f_4^+ - f_4^-}{2\Delta\lambda}\right| = 0.000350 < \varepsilon$$

Since the process has converged, we take the optimum solution as $\lambda^* \approx \lambda_4 = 0.480600$.

QUASI-NEWTON METHODS

- Newton's method estimates the search direction p_k that satisfies

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

- Quasi-Newton methods use an approximation of the Hessian, B_k.

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

- For one-dimensional nonlinear optimization problems, quasi-Newton methods are generalizations of the secant method.
- The secant method approximates the Hessian using

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \qquad f"(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

QUASI-NEWTON METHODS

- For multidimensional nonlinear optimization problems

$$\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1})$$

$$B_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}).$$

- Approximate and update the matrix B_k.
  - ✓ The symmetric rank-one update:

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

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

  - ✓ The Broyden–Fletcher–Goldfarb–Shanno (BFGS) update:

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

QUASI-NEWTON METHODS

➢*Procedure*

1. Choose a starting point x_0. Choose an initial Hessian approximation B_0, and it can be a diagonal matrix, I. set $k = 0$.
2. If x_k is optimal, **stop**.
3. Find out the search direction, p_k, by solving $B_k p_k = -\nabla f(x_k)$.
4. Find the step length, α_k, by a line search to minimize $f(x_k + \alpha_k p_k)$.
5. Set $x_{k+1} = x_k + \alpha_k p_k$.
6. Update B_{k+1} using the selected update formula.
7. Set $k = k + 1$, go to Step 2.

QUASI-NEWTON METHODS

Example: $f(x) = \frac{1}{2}x^T Qx - \bar{c}^T x$

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

$$x_0 = (0,0,0)^T \qquad B_0 = I \qquad \|\nabla f(x_0)\| = \|-c\| = 14.4568 \qquad p_0 = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix} \qquad \alpha_0 = 0.3025$$

$$x_1 = \begin{bmatrix} -2.4197 \\ -2.7221 \\ -2.4197 \end{bmatrix} \qquad \nabla f(x_1) = \begin{bmatrix} 3.1606 \\ 0.8336 \\ -4.0984 \end{bmatrix} \qquad \|\nabla f(x_1)\| = 5.2423$$

$$s_0 = x_1 - x_0 = \begin{bmatrix} -2.4197 \\ -2.7221 \\ -2.4197 \end{bmatrix} \qquad y_0 = \nabla f(x_1) - \nabla f(x_0) = \begin{bmatrix} -4.8394 \\ -8.1664 \\ -12.0984 \end{bmatrix}$$

$$B_1 = I + \frac{(y_0 - Is_0)(y_0 - Is_0)^T}{(y_0 - Is_0)^T s_0} = \begin{bmatrix} 1.1328 & 0.2988 & 0.5311 \\ 0.2988 & 1.6722 & 1.1950 \\ 0.5311 & 1.1950 & 3.1245 \end{bmatrix}$$

$$p_1 = \begin{bmatrix} -3.5444 \\ -1.6971 \\ 2.5633 \end{bmatrix} \qquad \alpha_1 = 0.3471$$

Example:

$$f(x) = \frac{1}{2}x^T Q x - \bar{c}^T x$$

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \qquad c = \begin{bmatrix} -8 \\ -9 \\ -8 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} -3.6499 \\ -3.3112 \\ -1.5300 \end{bmatrix} \qquad \nabla f(x_2) = \begin{bmatrix} 0.7002 \\ -0.9335 \\ 0.3501 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 1.4875 & 0.6833 & -0.2562 \\ 0.6833 & 2.0890 & 0.3416 \\ -0.2562 & 0.3416 & 4.8719 \end{bmatrix}$$

$$\|\nabla f(x_2)\| = 0.8397 \qquad p_2 = \begin{bmatrix} -0.4851 \\ 0.5749 \\ -0.2426 \end{bmatrix} \qquad \alpha_2 = 0.4145$$

$$x_3 = \begin{bmatrix} -4 \\ -3 \\ -1.6000 \end{bmatrix} \qquad \nabla f(x_3) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad f(x_3) = -35.9$$

LINE SEARCH METHODS

- The iteration is given

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

- Most line search algorithms require <u>the search direction to be a descent direction</u>

$$p_k^T \nabla_k f(x_k) < 0$$

- The **step length**, α_k, should generate a substantial reduction of f(x) along the descent direction.

$$\phi(\alpha_k) = f(x_k + \alpha_k p_k),\ \alpha_k > 0$$

- It is too expensive to find the minimum. Some practical strategies perform an inexact line search to identify a step length with adequate reductions in f(x), and require minimal cost.

LINE SEARCH METHODS

- The Armijo condition (sufficient decrease): $c_1 \in (0, 1)$

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f_k^T p_k$$

- Reasonable progress: $c_2 \in (c1, 1)$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \ge c_2 \nabla f_k^T p_k$$

- The sufficient decrease condition and the curvature condition are known collectively as the Wolfe conditions.

RATE OF CONVERGENCE

- The rate of convergence is an measure of efficiency, which describes **how quickly the estimates of the solution approach the exact solution**.
- The error at the k^{th} iteration can be defined as

$$e_k = x_k - x^*$$

- As the sequence approaches the solution x^*, the limit of e_k is 0.
- The sequence x_k converges to x^* with rate r and constant C.

$$\lim_{x \to \infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C$$

RATE OF CONVERGENCE

$$\lim_{x\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C$$

- When r = 1, it is a linear convergence. If 0 < C < 1, then the norm of the error is reduced at every iteration; and if C > 1, then the sequence diverges.
- If r = 1 and C = 0, the rate of the convergence is superlinear.
- For any r > 1, if

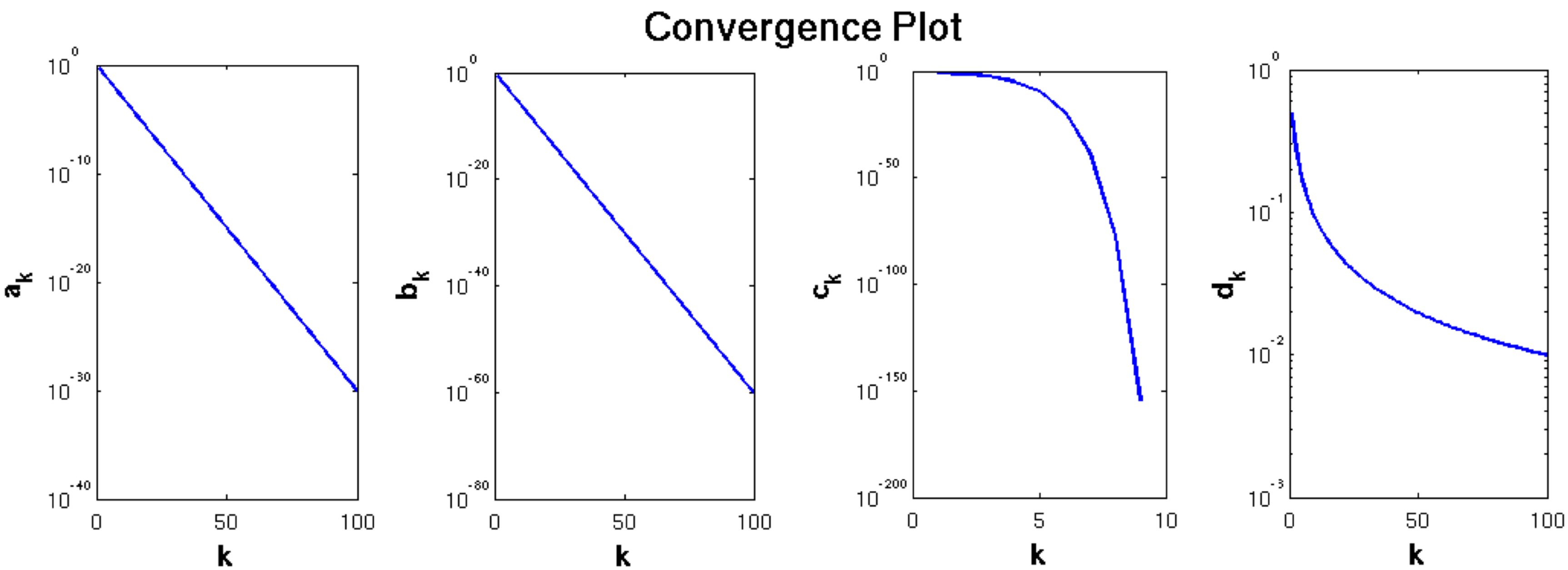
$$\lim_{x\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C < \infty$$

- Then

$$\lim_{x\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = \lim_{x\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r}\|e_k\|^{r-1} = C \times \lim_{x\to\infty} \|e_{k+1}\|^{r-1} = 0$$

- Any convergence with r > 1, it has a superlinear rate of convergence.
- When r > 2, the convergence is called quadratic.

RATE OF CONVERGENCE



Linear

Linear

Superlinear (quadratic)

Sublinear

COMPARISON OF OPTIMIZATION METHODS

Method	Converge Guarantee	Converge Rate	Compute Cost	Scale	Storage
Bisection	Yes	Linear			
Golden Section	Yes	Linear	High		
Quadratic Appro.					
Pattern Search	Yes				
Steepest Descent	Yes	Linear	Low		
Conjugate Gradient	Yes		Low	Large	Small
Newton	No	Quadratic	High	Scale	
Quasi-Newton	Yes	Superlinear	Low	Small - Mid	Big