MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Civil and Environmental Engineering

1.731 Water Resource Systems

Lecture 12 & 13, Nonlinear Programming, Oct. 19, 24 2006

Unconstrained Nonlinear Programming
General statement of unconstrained nonlinear programming (NLP) problem:

$$\underset{x_1,x_2,\ldots,x_n}{\text{Maximize}} \quad F(x_1, x_2, \ldots, x_n)$$

All candidate solutions are feasible for unconstrained problems and optimum always lies inside feasible region.

Necessary Conditions:

1. Feasibility
Any x^* is feasible

2. Stationarity
If x^* is a local maximum then:

$$\frac{\partial F(x^*)}{\partial x_j} = 0$$

3. Inequality Lagrange multiplier
Not applicable since there are no constraints

4. Curvature
If x^* is local maximum then:.

$$W_{kl} = \frac{\partial^2 L(x^*, \lambda)}{\partial x_j \partial x_k} Z_{ik} Z_{lj} = \frac{\partial^2 F(x^*)}{\partial x_j \partial x_k} \leq 0$$

In NLP problems local maxima are usually not necessarily global maxima (convexity conditions usually do not hold).

Finding a Local Maximum.

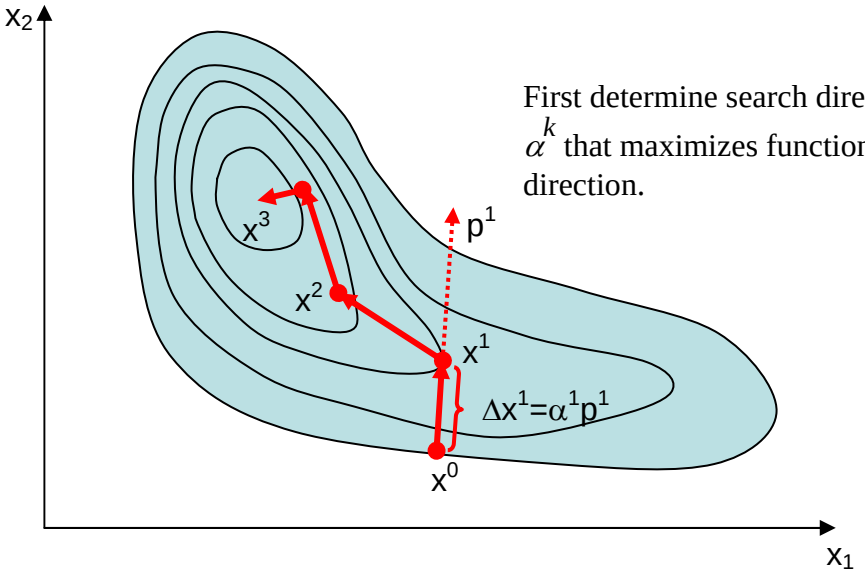
Use an iterative search that moves from one candidate solution $x^k = x_i^k$ on iteration k to a new candidate solution $x^{k+1} = x_i^{k+1}$ on iteration $k + 1$:

Search procedure:

Start by selecting an initial solution x^0, set $k = 0$.

1. Test for **convergence**, if converged set $x^* = x^k$ and exit.
2. Compute a **search direction** vector $p^k = p_i^k$
3. Compute a **search distance** $\alpha^k > 0$ along p^k. For some search methods α^k is set to 1.
4. Compute the new solution $x^{k+1} = x^k + \Delta x^k = x^k + \alpha^k p^k$ and go to Step 1

If search is an **ascent method** then $F(x^k + 1) \geq F(x)$ for α^k sufficiently small.



First determine search direction p^k. Then find α^k that maximizes function along this direction.

There are a number of different search alternatives for UNLP..

Steepest Ascent/Gradient Search

Derive p^k by expanding $F(x^k + 1)$ in a **first-order Taylor series** in $\Delta x^k = \alpha^k p_i^k$:

$$F(x^{k+1}) = F(x^k) + \frac{\partial F(x^k)}{\partial x_i}\alpha^k p_i^k + \ldots \qquad \text{where} \quad \frac{\partial F(x^k)}{\partial x_i} = \left.\frac{\partial F(x)}{\partial x_i}\right|_{x=x^k} = \text{Gradient}$$

The increase in $F(x)$ is largest when p_i^k is aligned with the objective function gradient $\partial F(x^k)/\partial x_j$:

$$p_i^k = \frac{\partial F(x^k)}{\partial x_i}$$

If the step Δx^k is too large the first-order expansion may not be valid. Control the step with α^k, which is derived by maximizing $F(x^{k+1})$ with respect to α^k (with p^k given):

$$\underset{\alpha^k}{\text{Maximize}} \quad F(x^k + \alpha^k p_i^k)$$

Use an iterative **univariate search algorithm** to solve this problem (see Gill et al. for alternatives). This requires **multiple evaluations of *F(x)*** on each iteration.

Usually the gradient $\partial F(x^k)/\partial x_j$ is derived **numerically**, using finite difference or adjoint methods that require **multiple evaluations of *F(x)*** on each iteration.

Steepest ascent characteristics:

- **Converges very slowly** (or not at all).
- **Univariate search** for α^k is essential
- **Many function evaluations** required

Newton's Method

Derive p^k by expanding $F(x^k + 1)$ in a **second-order Taylor series** about x^k (with $\alpha^k = 1$, so $\Delta x^k = p^k$):

$$F(x^{k+1}) = F(x^k) + \frac{\partial F(x^k)}{\partial x_i} p_i^k + \frac{1}{2} \frac{\partial^2 F(x^k)}{\partial x_i \partial x_j} p_i^k p_j^k + \ldots$$

where: $\left. \frac{\partial^2 F(x)}{\partial x_i \partial x_j} \right|_{x=x^k} = \frac{\partial^2 F(x^k)}{\partial x_i \partial x_j} = H_{ij}^k$ = objective Hessian

Maximize increase in $F(x)$ with respect to p^k:

$$\underset{p^k}{\text{Maximize}} \quad \frac{\partial F(x^k)}{\partial x_i} p_i^k + \frac{1}{2} H_{ij}^k p_i^k p_j^k$$

Necessary conditions for this quadratic optimization problem imply that:

$$H_{ij}^k p_j^k = -\gamma_i^k \quad \text{where} \quad \gamma^k = \gamma_i^k = \partial F(x^k)/\partial x_i \text{ = objective gradient}$$

The Newton direction $\Delta x^k = p^k$ is solution to this set of linear eqs. It is an ascent direction if $H_{ij}^k < 0$ (negative definite). Modifications are needed at points where $\partial F(x^k)/\partial x_i = 0$ and/or $H_{ij}^k = 0$. **Modified Newton methods** replace indefinite Hessian by a positive definite approximation (e.g. by using an eigenvalue decomposition).

Usually the gradient and Hessian need to be computed **numerically**, using finite difference or adjoint methods. Exact Hessian computation may be infeasible for large problems because of the number of function evaluations required.

Newton's method characteristics:

- Hessian must be **negative definite**
- **Converges faster than steepest ascent**
- **Many function evaluations** required
- **Very expensive** to compute H_{ij}^k if numerical differentiation is required.

Quasi-Newton Methods

Avoid computational disadvantages of Newton's method and retain good convergence properties by gradually constructing a negative definite approximation to Hessian.

Basic idea is to use an iterative algorithm to update an approximate negative definite Hessian matrix $B^k = B_{ij}^k$., using only information about gradients and previous search steps. The **search direction** p^k is found from an equation having the same form as the Newton equation:

$$B_{ij}^k p_j^k = -\gamma_i^k$$

The new solution is usually obtained by using a search distance a^k obtained from a univariate search:

$$x^{k+1} = x^k + \Delta x^k = x^k + \alpha^k p^k$$

The iteration is often initialized with the negative of the identity matrix $B^0 = -I = -\delta_{ij}$.

Consequently, the first step is a steepest ascent step.

The approximate Hessian is obtained from:

$$B^{k+1} = B^k + U^k \qquad \text{where } U^k = U_{ij}^k \text{ is an update matrix}$$

How do we choose U^k? Many options !

The **curvature** of $F(x)$ in the direction $\Delta x_i^k = x_i^{k+1} - x_i^k$ at x^k is the quadratic form $\Delta x_i^k H_{ij}^k \Delta x_j^k$.

This can be expressed in terms of the objective gradient since:

$$\Delta x_i^k H_{ij}^k \Delta x_j^k \approx \Delta x_i^k (\gamma_i^{k+1} - \gamma_i^k) = \Delta x_i^k \Delta \gamma_i^k \quad \text{since } \gamma_i^{k+1} = \gamma_i^k + H_{ij}^k \Delta x_i^k + \ldots$$

We wish to obtain the same curvature when we replace H^k by B^{k+1}. This implies:

$$B_{ij}^{k+1} \Delta x_j^k \approx \Delta \gamma_i^k \qquad \textbf{Quasi-Newton condition}.\text{for } k \to k+1$$

In most quasi-Newton methods U^k is chosen subject to following requirements:

1. B^{k+1} should satisfy the **quasi-Newton condition**:

2. U^k should be **rank 1**, which implies that $U_{ij}^k = u_i^k v_i^k$ for some vectors u^k and v^k.
3. U^k should be **symmetric** (since B_{k+1} should be symmetric)

There are various methods that satisfy these requirements asymptotically (for large k). One of the most widely used is the **Broyden-Fltecher-Goldfarb-Shanno (BFGS)** algorithm:

$$U_{ij}^k = -\frac{1}{\Delta x_l^k B_{lm}^k \Delta x_m^k} B_{il}^k \Delta x_l^k B_{mj}^k \Delta x_m^k + \frac{1}{\Delta \gamma_l^k \Delta x_l^k} \Delta \gamma_i^k \Delta \gamma_j^k$$

When we substitute $\Delta x^k = \alpha^k p^k$ and $B_{ij}^k p_j^k = -\gamma_i^k$ this simplifies to:

$$U_{ij}^k = \frac{1}{\gamma_l^k p_l^k} \gamma_i^k \gamma_j^k + \frac{1}{\alpha^k \Delta \gamma_l^k p_l^k} \Delta \gamma_i^k \Delta \gamma_j^k$$

The BFGS update maintains the negative definiteness of the approximate Hessian from iteration to iteration.

As in steepest ascent, the univariate search and gradient evaluation together require many function evaluations on each iteration.

Quasi-Newton characteristics:

- No need to derive **Hessian**
- Approximate Hessian is **negative definite**
- **Univariate search** for α^k is desirable
- **Many function evaluations** required
- **Good convergence** properties
- Somewhat more expensive than steepest ascent but much more **reliable**.

Gauss-Newton Methods

Unconstrained nonlinear least-squares problems have a special structure that leads to efficient iterative search algorithms. These problems seek to **minimize** the weighted sum of squared errors between a set of measurements y_t (t = 1,…, m) and a nonlinear model $h_t(x_j)$ that depends on a set of decision variables (or parameters) x_j (j = 1,…, n).

$$\underset{x_1,x_2,\ldots,x_n}{\text{Minimize}} \quad F(x_1, x_2, \ldots, x_n)$$

where $F(x) = \frac{1}{2}\sum_{t=1}^{m} K_t[y_t - h_t(x)]^2$ = = sum of weighted squared errors, K_t = positive weight

Solve iteratively, starting with an initial estimate x^0. Obtain search step as follows:

1. Set gradient at new iterate x^{k+1} equal to zero (from stationarity condition):

$$\frac{\partial F(x^{k+1})}{\partial x_j} = \sum_{t=1}^{N} K_t[y_t - h_t(x^{k+1})] \frac{\partial h_t(x^{k+1})}{\partial x_j} = 0$$

2. Approximate $h_t(x_j)$ by a **first-order Taylor series expansion** around previous iteration:

$$h_t(x^{k+1}) = h_t(x^k) + \frac{\partial h_t(x^k)}{\partial x_j}\Delta x_j^k + \ldots$$

This is valid only if step Δx_j^k is **sufficiently small**.

3. Assume $\partial h_t(x^{k+1})/\partial x_j = \partial h_t(x^k)/\partial x_j$.

Then stationarity condition reduces to:

$$B_{ij}^k \Delta x_j^k = \gamma_i^k$$

where:

$$\frac{\partial F(x^k)}{\partial x_i} = \sum_{t=1}^{N} K_t[y_t - h_t(x^k)]\frac{\partial h_t(x^k)}{\partial x_i} \qquad B_{ij}^k = \sum_{t=1}^{N} K_t \frac{\partial h_t(x^{k+1})}{\partial x_i}\frac{\partial h_t(x^{k+1})}{\partial x_j}$$

This is the **Gauss-Newton** search algorithm. It has same form as Quasi-Newton algorithm, with a positive definite approximate Hessian B^k computed from the sensitivity derivatives $\partial h_t(x^k)/\partial x_j$ and with $\Delta x^k = p^k(\alpha^k = 1)$. Sensitivity derivatives are usually be computed numerically, from multiple evaluations of the function $h(x)$.

Algorithm performance often improved by adding a positive constant λ^k to diagonals of B^k so:

$$[B_{ij}^k + \lambda^k \delta_{ij}]\Delta x_j^k = \gamma_i^k$$

This gives **Levenberg-Marquardt** search algorithm.

Small $\lambda^k \rightarrow$ unmodified Gauss-Newton (steps may be too large)

Large $\lambda^k \rightarrow$ Steepest descent (steps may be too small)

It is helpful to adjust λ^k dynamically to improve convergence. Univariate search along Δx^k is an alternative.

Gauss-Newton characteristics:

- Specialized algorithm for **least-squares (minimization) problems**
- Approximate positive definite Hessian is derived from **sensitivity derivatives**
- **Many function evaluations** required
- **Good convergence** properties
- Most **efficient** option for least-squares problems (depending on effort required to evaluate sensitivity derivatives).