Math 4/896: Seminar in Mathematics
Topic: Inverse Theory

Instructor: Thomas Shores
Department of Mathematics

Lecture 26, April 18, 2006
AvH 10
Root Finding:

Solve the system of equations represented in vector form as

\[ \mathbf{F}(\mathbf{x}) = 0. \]

for point(s) \( \mathbf{x}^* \) for which \( \mathbf{F}(\mathbf{x}^*) = 0. \)

- Here \( \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \) and \( \mathbf{x} = (x_1, \ldots, x_m) \)
- Gradient notation: \( \nabla f_j(\mathbf{x}) = \left( \frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x}) \right) \).
- Jacobian notation:
  \[ \nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[ \frac{\partial f_i}{\partial x_j} \right]_{i,j=1,...,m}. \]
Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

\[ \mathbf{F}(\mathbf{x}) = 0. \]

for point(s) \( \mathbf{x}^* \) for which \( \mathbf{F}(\mathbf{x}^*) = 0. \)

- Here \( \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \) and \( \mathbf{x} = (x_1, \ldots, x_m) \)
- Gradient notation: \( \nabla f_j(\mathbf{x}) = \left( \frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x}) \right) \).
- Jacobian notation:
  \[ \nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[ \frac{\partial f_i}{\partial x_j} \right]_{i,j=1,...,m}. \]
Root Finding:

Solve the system of equations represented in vector form as

\[ F(x) = 0. \]

for point(s) \( x^* \) for which \( F(x^*) = 0. \)

- Here \( F(x) = (f_1(x), \ldots, f_m(x)) \) and \( x = (x_1, \ldots, x_m) \)
- Gradient notation: \( \nabla f_j(x) = \left( \frac{\partial f_j}{\partial x_1}(x), \ldots, \frac{\partial f_j}{\partial x_m}(x) \right) \).
- Jacobian notation:
  \[
  \nabla F(x) = [\nabla f_1(x), \ldots, \nabla f_m(x)]^T = \left[ \frac{\partial f_i}{\partial x_j} \right]_{i,j=1,\ldots,m}.
  \]
Root Finding:

Solve the system of equations represented in vector form as

\[ \mathbf{F}(\mathbf{x}) = 0. \]

for point(s) \( \mathbf{x}^* \) for which \( \mathbf{F}(\mathbf{x}^*) = 0 \).

- Here \( \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \) and \( \mathbf{x} = (x_1, \ldots, x_m) \)
- Gradient notation: \( \nabla f_j(\mathbf{x}) = \left( \frac{\partial f_j}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f_j}{\partial x_m}(\mathbf{x}) \right) \).
- Jacobian notation:

\[
\nabla \mathbf{F}(\mathbf{x}) = [\nabla f_1(\mathbf{x}), \ldots, \nabla f_m(\mathbf{x})]^T = \left[ \frac{\partial f_i}{\partial x_j} \right]_{i,j=1,\ldots,m}.
\]
Basic Problems

Optimization:

Find the minimum value of scalar valued function \( f(x) \), where \( x \) ranges over a feasible set \( \Omega \).

- Set \( F(x) = \nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_m}(x) \right) \)

- Hessian of \( f \): \( \nabla (\nabla f(x)) \equiv \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_i \partial x_j} \end{bmatrix} \).
Optimization:

Find the minimum value of scalar valued function \( f(x) \), where \( x \) ranges over a feasible set \( \Omega \).

- Set \( F(x) = \nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_m}(x) \right) \)

- Hessian of \( f \): \( \nabla (\nabla f(x)) \equiv \nabla^2 f(x) = \left[ \frac{\partial^2 f}{\partial x_i \partial x_j} \right] \).
Optimization:

Find the minimum value of scalar valued function $f(x)$, where $x$ ranges over a feasible set $\Omega$.

- Set $\mathbf{F}(x) = \nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_m}(x) \right)$

- Hessian of $f$: $\nabla (\nabla f(x)) \equiv \nabla^2 f(x) = \left[ \frac{\partial^2 f}{\partial x_i \partial x_j} \right]$. 

Instructor: Thomas Shores Department of Mathematics

Math 4/896: Seminar in Mathematics Topic: Inverse Theory
Taylor Theorems

First Order
Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous second partials and $x^*, x \in \mathbb{R}^n$. Then
\[ f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \mathcal{O}\left(\|x - x^*\|^2\right), \quad x \to x. \]

Second Order
Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous third partials and $x^*, x \in \mathbb{R}^n$. Then
\[ f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*) (x - x^*) + \mathcal{O}\left(\|x - x^*\|^3\right), \quad x \to x. \]
(See Appendix C for versions of Taylor’s theorem with weaker hypotheses.)
Taylor Theorems

First Order
Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous second partials and $x^*, x \in \mathbb{R}^n$. Then
$$f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + O\left(\|x - x^*\|^2\right), \ x \to x.$$ 

Second Order
Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous third partials and $x^*, x \in \mathbb{R}^n$. Then
$$f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*) (x - x^*) + O\left(\|x - x^*\|^3\right), \ x \to x.$$ 
(See Appendix C for versions of Taylor’s theorem with weaker hypotheses.)
Newton Algorithms

Root Finding

Input $F$, $\nabla F$, $x^0$, $N_{\text{max}}$

for $k = 0, \ldots, N_{\text{max}}$

\[ x^{k+1} = x^k - \nabla F \left( x^k \right)^{-1} F \left( x^k \right) \]

if $x^{k+1}, x^k$ pass a convergence test

return($x^k$)

end

end

return($x^{N_{\text{max}}}$)
Theorem

Let \( x^* \) be a root of the equation \( \mathbf{F}(x) = 0 \), where \( \mathbf{F}, \mathbf{x} \) are \( m \)-vectors, \( \mathbf{F} \) has continuous first partials in some neighborhood of \( x^* \) and \( \nabla \mathbf{F}(x^*) \) is non-singular. Then Newton’s method yields a sequence of vectors that converges to \( x^* \), provided that \( x^0 \) is sufficiently close to \( x^* \). If, in addition, \( \mathbf{F} \) has continuous second partials in some neighborhood of \( x^* \), then the convergence is quadratic in the sense that for some constant \( K > 0 \),

\[
\|x^{k+1} - x^*\| \leq K \|x^k - x^*\|^2.
\]
Bright Idea:

We know from calculus that where \( f(x) \) has a local minimum, \( \nabla f = 0 \). So just let \( F(x) = \nabla f(x) \) and use Newton’s method.

- Result is iteration formula: \( x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \)

- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving \( F(x) = 0 \) is the same as minimizing \( f(x) = ||F(x)||^2 \).

- Downside of root finding point of view of optimization: saddle points and local maxima \( x \) also satisfy \( \nabla f(x) = 0 \).

- Upside of optimization view of root finding: if \( F(x) = 0 \) doesn’t have a root, minimizing \( f(x) = ||F(x)||^2 \) finds the next best solutions – least squares solutions!

- In fact, least squares problem for \( ||Gm - d||^2 \) is optimization!
**Bright Idea:**

We know from calculus that where $f(x)$ has a local minimum, $\nabla f = 0$. So just let $F(x) = \nabla f(x)$ and use Newton’s method.

- Result is iteration formula: $x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $F(x) = 0$ is the same as minimizing $f(x) = \|F(x)\|^2$.
- Downside of root finding point of view of optimization: saddle points and local maxima $x$ also satisfy $\nabla f(x) = 0$.
- Upside of optimization view of root finding: if $F(x) = 0$ doesn’t have a root, minimizing $f(x) = \|F(x)\|^2$ finds the next best solutions – least squares solutions!
- In fact, least squares problem for $\|Gm - d\|^2$ is optimization!
Newton for Optimization

Bright Idea:

We know from calculus that where \( f(x) \) has a local minimum, \( \nabla f = 0 \). So just let \( F(x) = \nabla f(x) \) and use Newton’s method.

- Result is iteration formula: \( x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \)
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving \( F(x) = 0 \) is the same as minimizing \( f(x) = \|F(x)\|^2 \).

- Downside of root finding point of view of optimization: saddle points and local maxima also satisfy \( \nabla f(x) = 0 \).
- Upside of optimization view of root finding: if \( F(x) = 0 \) doesn’t have a root, minimizing \( f(x) = \|F(x)\|^2 \) finds the next best solutions – least squares solutions!
- In fact, least squares problem for \( \|Gm - d\|^2 \) is optimization!
Newton for Optimization

Bright Idea:

We know from calculus that where \( f(x) \) has a local minimum, \( \nabla f = 0 \). So just let \( F(x) = \nabla f(x) \) and use Newton’s method.

- Result is iteration formula: \( x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \)
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving \( F(x) = 0 \) is the same as minimizing \( f(x) = \|F(x)\|^2 \).
- Downside of root finding point of view of optimization: saddle points and local maxima \( x \) also satisfy \( \nabla f(x) = 0 \).
- Upside of optimization view of root finding: if \( F(x) = 0 \) doesn’t have a root, minimizing \( f(x) = \|F(x)\|^2 \) finds the next best solutions — least squares solutions!
- In fact, least squares problem for \( \|Gm - d\|^2 \) is optimization!
Bright Idea:

We know from calculus that where $f(x)$ has a local minimum, $\nabla f = 0$. So just let $F(x) = \nabla f(x)$ and use Newton’s method.

- Result is iteration formula: $x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$

- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving $F(x) = 0$ is the same as minimizing $f(x) = \|F(x)\|^2$.

- Downside of root finding point of view of optimization: saddle points and local maxima also satisfy $\nabla f(x) = 0$.

- Upside of optimization view of root finding: if $F(x) = 0$ doesn’t have a root, minimizing $f(x) = \|F(x)\|^2$ finds the next best solutions – least squares solutions!

- In fact, least squares problem for $\|Gm - d\|^2$ is optimization!
Newton for Optimization

Bright Idea:

We know from calculus that where \( f(x) \) has a local minimum, \( \nabla f = 0 \). So just let \( F(x) = \nabla f(x) \) and use Newton’s method.

- Result is iteration formula: \( x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \)
- We can turn this approach on its head: root finding is just a special case of optimization, i.e., solving \( F(x) = 0 \) is the same as minimizing \( f(x) = \|F(x)\|^2 \).
- Downside of root finding point of view of optimization: saddle points and local maxima also satisfy \( \nabla f(x) = 0 \).
- Upside of optimization view of root finding: if \( F(x) = 0 \) doesn’t have a root, minimizing \( f(x) = \|F(x)\|^2 \) finds the next best solutions – least squares solutions!
- In fact, least squares problem for \( \|Gm - d\|^2 \) is optimization!
About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla F(x^k)^{-1} F(x^k)$ for a point that (approximately) minimizes a merit function like $m(x) = \|F(x)\|^2$.

- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(x)$ problem that get lost if one only tries to find a zero of $\nabla f$.

- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.

- There is an automatic merit function, namely $f(x)$, in any search direction. Using this helps avoid saddle points, maxima.
About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla F(x^k)^{-1}F(x^k)$ for a point that (approximately) minimizes a merit function like $m(x) = \|F(x)\|^2$.

- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(x)$ problem that get lost if one only tries to find a zero of $\nabla f$.

- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.

- There is an automatic merit function, namely $f(x)$, in any search direction. Using this helps avoid saddle points, maxima.
About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla F(x^k)^{-1} F(x^k)$ for a point that (approximately) minimizes a merit function like $m(x) = \|F(x)\|^2$.

- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(x)$ problem that get lost if one only tries to find a zero of $\nabla f$.

- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.

- There is an automatic merit function, namely $f(x)$, in any search direction. Using this helps avoid saddle points, maxima.
Remarks on Newton

About Newton:

This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction \(-\nabla F(x_k)^{-1} F(x_k)\) for a point that (approximately) minimizes a merit function like \(m(x) = \|F(x)\|^2\).

- Optimization is NOT a special case of root finding. There are special characteristics of the \(\min f(x)\) problem that get lost if one only tries to find a zero of \(\nabla f\).

- For example, \(-\nabla f\) is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.

- There is an automatic merit function, namely \(f(x)\), in any search direction. Using this helps avoid saddle points, maxima.
About Newton:
This barely scratches the surface of optimization theory (take Math 4/833 if you can!!).

- Far from a zero, Newton does not exhibit quadratic convergence. It is accelerated by a line search in the Newton direction $-\nabla F(x^k)^{-1} F(x^k)$ for a point that (approximately) minimizes a merit function like $m(x) = \|F(x)\|^2$.

- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(x)$ problem that get lost if one only tries to find a zero of $\nabla f$.

- For example, $-\nabla f$ is a search direction that leads to the method of steepest descent. This is not terribly efficient, but well understood.

- There is an automatic merit function, namely $f(x)$, in any search direction. Using this helps avoid saddle points, maxima.
Chapter 9: Nonlinear Regression

Newton's Method

Gauss-Newton and Levenberg-Marquardt Methods

Section 9.3: Statistical Aspects

Implementation Issues

Outline

Instructor: Thomas Shores
Department of Mathematics
Math 4/896: Seminar in Mathematics
Topic: Inverse Theory
The Problem:

Given a function \( F(x) = (f_1(x), \ldots, f_m(x)) \), minimize

\[
f(x) = \sum_{k=0}^{m} f_k(x)^2 = \|F(x)\|^2.
\]

- Newton’s method can be very expensive, due to derivative evaluations.
- For starters, one shows \( \nabla f(x) = 2(\nabla F(x))^T F(x) \)
- Then, \( \nabla^2 f(x) = 2(\nabla F(x))^T \nabla F(x) + Q(x) \), where \( Q(x) = \sum_{k=1}^{m} f_k(x) \nabla^2 f_k(x) \) contains all the second derivatives.
The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2.$$  

- Newton’s method can be very expensive, due to derivative evaluations.

- For starters, one shows $\nabla f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$

- Then, $\nabla^2 f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x}) = \sum_{k=1}^{m} f_k(\mathbf{x}) \nabla^2 f_k(\mathbf{x})$ contains all the second derivatives.
The Problem:

Given a function \( \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \), minimize

\[
f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \| \mathbf{F}(\mathbf{x}) \|^2.
\]

- Newton’s method can be very expensive, due to derivative evaluations.
- For starters, one shows \( \nabla f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x}) \)
- Then, \( \nabla^2 f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x}) \), where \( Q(\mathbf{x}) = \sum_{k=1}^{m} f_k(\mathbf{x}) \nabla^2 f_k(\mathbf{x}) \) contains all the second derivatives.
The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize

$$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \| \mathbf{F}(\mathbf{x}) \|^2.$$

- Newton’s method can be very expensive, due to derivative evaluations.
- For starters, one shows $\nabla f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x})$
- Then, $\nabla^2 f(\mathbf{x}) = 2 (\nabla \mathbf{F}(\mathbf{x}))^T \nabla \mathbf{F}(\mathbf{x}) + Q(\mathbf{x})$, where $Q(\mathbf{x}) = \sum_{k=1}^{m} f_k(\mathbf{x}) \nabla^2 f_k(\mathbf{x})$ contains all the second derivatives.
The Problem:

Given a function $F(x) = (f_1(x), \ldots, f_m(x))$, minimize

$$f(x) = \sum_{k=0}^{m} f_k(x)^2 = \|F(x)\|^2.$$ 

- This inspires a so-called quasi-Newton method, which approximates the Hessian as $\nabla^2 f(x) \approx 2(\nabla F(x))^T \nabla F(x)$.
- Thus, Newton’s method morphs into the Gauss-Newton (GN) method

$$x^{k+1} = x^k - \left(\left(\nabla F(x^k)\right)^T \nabla F(x^k)\right)^{-1} \left(\nabla F(x^k)\right)^T F(x^k)$$

- There’s a problem here. See it?
The Problem:

Given a function $F(x) = (f_1(x), \ldots, f_m(x))$, minimize

$$f(x) = \sum_{k=0}^{m} f_k(x)^2 = \|F(x)\|^2.$$

- This inspires a so-called quasi-Newton method, which approximates the Hessian as $\nabla^2 f(x) \approx 2 (\nabla F(x))^T \nabla F(x)$.
- Thus, Newton’s method morphs into the Gauss-Newton (GN) method

$$x^{k+1} = x^k - \left((\nabla F(x^k))^T \nabla F(x^k)\right)^{-1} \left((\nabla F(x^k))^T F(x^k)\right)$$

- There’s a problem here. See it?
The Problem:

Given a function \( F(x) = (f_1(x), \ldots, f_m(x)) \), minimize

\[
    f(x) = \sum_{k=0}^{m} f_k(x)^2 = \|F(x)\|^2.
\]

- This inspires a so-called quasi-Newton method, which approximates the Hessian as \( \nabla^2 f(x) \approx 2 (\nabla F(x))^T \nabla F(x) \).
- Thus, Newton’s method morphs into the Gauss-Newton (GN) method

\[
    x^{k+1} = x^k - \left( (\nabla F(x^k))^T \nabla F(x^k) \right)^{-1} (\nabla F(x^k))^T F(x^k)
\]

- There’s a problem here. See it?
The Problem:

Given a function \( F(x) = (f_1(x), \ldots, f_m(x)) \), minimize

\[
f(x) = \sum_{k=0}^{m} f_k(x)^2 = \| F(x) \|^2.
\]

- This inspires a so-called quasi-Newton method, which approximates the Hessian as \( \nabla^2 f(x) \approx 2 (\nabla F(x))^T \nabla F(x) \).
- Thus, Newton’s method morphs into the Gauss-Newton (GN) method

\[
x^{k+1} = x^k - \left( (\nabla F(x^k))^T \nabla F(x^k) \right)^{-1} (\nabla F(x^k))^T F(x^k)
\]

- There’s a problem here. See it?
The Problem:

$\nabla F (x)$ may not have full column rank.

- A remedy: regularize the Newton problem to
  \[
  \left( \left( \nabla F (x^k) \right)^T \nabla F (x^k) + \lambda_k I \right) p = - \left( \nabla F (x^k) \right)^T F (x^k)
  \]
  with $\lambda$ suitably chosen positive number for $p = x - x^k$

- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing
  \[
  \left\| \nabla F (x^k) p + F (x^k) \right\|^2
  \]
  subject to a constraint $\|p\| \leq \delta_k$. Of course, $\delta_k$ determines $\lambda_k$ and vice-versa.

- The idea is to choose $\lambda_k$ at each step: Increase it if the reduction in $f (x)$ was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.
The Problem:
\[\nabla F (x)\] may not have full column rank.

- A remedy: regularize the Newton problem to
  \[
  \left( \left( \nabla F (x^k) \right)^T \nabla F (x^k) + \lambda_k I \right) p = - \left( \nabla F (x^k) \right)^T F (x^k)
  \]
  with \(\lambda\) suitably chosen positive number for \(p = x - x^k\)

- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing
  \[\left\| \nabla F (x^k) p + F (x^k) \right\|^2\]
  subject to a constraint \(\|p\| \leq \delta_k\). Of course, \(\delta_k\) determines \(\lambda_k\) and vice-versa.

- The idea is to choose \(\lambda_k\) at each step: Increase it if the reduction in \(f (x)\) was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.
The Problem:
\( \nabla F (x) \) may not have full column rank.

- A remedy: regularize the Newton problem to
  
  \[
  \left( \left( \nabla F (x^k) \right)^T \nabla F (x^k) + \lambda_k I \right) p = - \left( \nabla F (x^k) \right)^T F (x^k)
  \]

  with \( \lambda \) suitably chosen positive number for \( p = x - x^k \)

- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing
  
  \[
  \| \nabla F (x^k) p + F (x^k) \|^2 \]

  subject to a constraint \( \| p \| \leq \delta_k \). Of course, \( \delta_k \) determines \( \lambda_k \) and vice-versa.

- The idea is to choose \( \lambda_k \) at each step: Increase it if the reduction in \( f (x) \) was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.
The Problem:

\( \nabla F(x) \) may not have full column rank.

- A remedy: regularize the Newton problem to
  
  \[
  \left( \left( \nabla F(x^k) \right)^T \nabla F(x^k) + \lambda_k I \right) p = - \left( \nabla F(x^k) \right)^T F(x^k)
  \]
  
  with \( \lambda \) suitably chosen positive number for \( p = x - x^k \)

- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing
  
  \[
  \| \nabla F(x^k) p + F(x^k) \|^2 \]
  
  subject to a constraint \( \| p \| \leq \delta_k \). Of course, \( \delta_k \) determines \( \lambda_k \) and vice-versa.

- The idea is to choose \( \lambda_k \) at each step: Increase it if the reduction in \( f(x) \) was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.
More on LM:

\[
\left( \left( \nabla F \left( x^k \right) \right)^T \nabla F \left( x^k \right) + \lambda_k I \right) p = - \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right).
\]

- For small \( \lambda_k \), LM becomes approximately
  \[
  \left( \nabla F \left( x^k \right) \right)^T \nabla F \left( x^k \right) p = - \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right)
  \]
  which is GN with its favorable convergence rate.

- For large \( \lambda_k \), LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right),
  \]
  which is a steepest-descent step, slow but convergent.

- For large \( \lambda_k \), LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right),
  \]
  which is a steepest-descent step, slow but convergent.

- For small residuals, LM (and GN, when stable) converge superlinearly. They tend to perform poorly on large residual problems, where the dropped Hessian terms are significant.
More on LM:

\[
\begin{pmatrix}
\nabla F(x^k) \\
\n\end{pmatrix}
^T \nabla F(x^k) + \lambda_k I \bigg] \mathbf{p} = - \begin{pmatrix}
\nabla F(x^k) \\
\n\end{pmatrix}
^T \nabla F(x^k) F(x^k).
\]

- For small \(\lambda_k\), LM becomes approximately
  \[
  \begin{pmatrix}
  \nabla F(x^k) \\
  \n\end{pmatrix}
^T \nabla F(x^k) \mathbf{p} = - \begin{pmatrix}
  \nabla F(x^k) \\
  \n\end{pmatrix}
^T \nabla F(x^k) \frac{1}{\lambda_k} (\nabla F(x^k) \nabla F(x^k))^{-1} F(x^k)
\]
  which is GN with its favorable convergence rate.

- For large \(\lambda_k\), LM becomes approximately
  \[
  \mathbf{p} = - \frac{1}{\lambda_k} \begin{pmatrix}
  \nabla F(x^k) \\
  \n\end{pmatrix}
^T \nabla F(x^k) F(x^k),
\]
  which is a steepest-descent step, slow but convergent.

- For large \(\lambda_k\), LM becomes approximately
  \[
  \mathbf{p} = - \frac{1}{\lambda_k} \begin{pmatrix}
  \nabla F(x^k) \\
  \n\end{pmatrix}
^T \nabla F(x^k) F(x^k),
\]
  which is a steepest-descent step, slow but convergent.

- For small residuals, LM (and GN, when stable) converge superlinearly. They tend to perform poorly on large residual problems, where the dropped Hessian terms are significant.
More on LM:

\[
\left( \left( \nabla F \left( x^k \right) \right)^T \nabla F \left( x^k \right) + \lambda_k I \right) p = - \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right).
\]

- For small \( \lambda_k \), LM becomes approximately
  \[\left( \nabla F \left( x^k \right) \right)^T \nabla F \left( x^k \right) p = - \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right)\] which is GN with its favorable convergence rate.

- For large \( \lambda_k \), LM becomes approximately
  \[p = - \frac{1}{\lambda_k} \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right),\] which is a steepest-descent step, slow but convergent.

- For large \( \lambda_k \), LM becomes approximately
  \[p = - \frac{1}{\lambda_k} \left( \nabla F \left( x^k \right) \right)^T F \left( x^k \right),\] which is a steepest-descent step, slow but convergent.

- For small residuals, LM (and GN, when stable) converge superlinearly. They tend to perform poorly on large residual problems, where the dropped Hessian terms are significant.
More on LM:

\[
\begin{pmatrix}
(\nabla F(x^k)) \quad \nabla F(x^k) + \lambda_k I
\end{pmatrix}
\begin{pmatrix}
p
\end{pmatrix}
= - (\nabla F(x^k)) \quad F(x^k).
\]

- For small $\lambda_k$, LM becomes approximately
  \[
  (\nabla F(x^k)) \quad \nabla F(x^k) \quad p = - (\nabla F(x^k)) \quad F(x^k)
  \]
  which is GN with its favorable convergence rate.

- For large $\lambda_k$, LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} (\nabla F(x^k)) \quad F(x^k),
  \]
  which is a steepest-descent step, slow but convergent.

- For large $\lambda_k$, LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} (\nabla F(x^k)) \quad F(x^k),
  \]
  which is a steepest-descent step, slow but convergent.

- For small residuals, LM (and GN, when stable) converge superlinearly. They tend to perform poorly on large residual problems, where the dropped Hessian terms are significant.
More on LM:

\[
\left( \left( \nabla F(x^k) \right)^T \nabla F(x^k) + \lambda_k I \right) p = - \left( \nabla F(x^k) \right)^T F(x^k).
\]

- For small \( \lambda_k \), LM becomes approximately
  \[
  \left( \nabla F(x^k) \right)^T \nabla F(x^k) p = - \left( \nabla F(x^k) \right)^T F(x^k)
  \]
  which is GN with its favorable convergence rate.

- For large \( \lambda_k \), LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} \left( \nabla F(x^k) \right)^T F(x^k),
  \]
  which is a steepest-descent step, slow but convergent.

- For large \( \lambda_k \), LM becomes approximately
  \[
  p = - \frac{1}{\lambda_k} \left( \nabla F(x^k) \right)^T F(x^k),
  \]
  which is a steepest-descent step, slow but convergent.

- For small residuals, LM (and GN, when stable) converge superlinearly. They tend to perform poorly on large residual problems, where the dropped Hessian terms are significant.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $d^k = F(x^k)$.
  - Linearize: $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - Regularize: $\begin{pmatrix}(\nabla F(x^k))^T \nabla F(x^k) + \alpha_k I\end{pmatrix}p = \begin{pmatrix}(\nabla F(x^k))^T \left(d^\delta - d^k\right)\end{pmatrix}$

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- **NB:** $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - **Forward-solve:** compute $d^k = F(x^k)$.
  - **Linearize:** $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - **Regularize:** $\left(\begin{pmatrix} \nabla F(x^k) \end{pmatrix}^T \nabla F(x^k) + \alpha_k I \right) p = \begin{pmatrix} \nabla F(x^k) \end{pmatrix}^T (d^\delta - d^k)$

- This is a regularization technique for nonlinear problems and is called **output least squares**.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $d^k = F(x^k)$.
  - Linearize: $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - Regularize: $\left(\begin{pmatrix} \nabla F(x^k) \end{pmatrix}^T \nabla F(x^k) + \alpha_k I\right) p = \begin{pmatrix} \nabla F(x^k) \end{pmatrix}^T (d^\delta - d^k)$

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $d^k = F(x^k)$.
  - Linearize: $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - Regularize: $\left(\left(\nabla F(x^k)\right)^T \nabla F(x^k) + \alpha_k I\right)p = 

\left(\nabla F(x^k)\right)^T \left(d^\delta - d^k\right)$

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularizaton parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $\mathbf{F}(\mathbf{x}) = \mathbf{d}$, given output data $\mathbf{d}$ with error, in the form of $\mathbf{d}^\delta$, i.e., to minimize $\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $\mathbf{d}^k = \mathbf{F}(\mathbf{x}^k)$.
  - Linearize: $\nabla \mathbf{F}(\mathbf{x}^k) (\mathbf{x}^{k+1} - \mathbf{x}^k) = \mathbf{d}^\delta - \mathbf{d}^k$.
  - Regularize: 
    \[
    \begin{pmatrix}
    \nabla \mathbf{F}(\mathbf{x}^k) \\
    \nabla \mathbf{F}(\mathbf{x}^k) + \alpha_k \mathbf{I}
    \end{pmatrix} \mathbf{p} = \\
    \begin{pmatrix}
    \nabla \mathbf{F}(\mathbf{x}^k) \\
    \nabla \mathbf{F}(\mathbf{x}^k)
    \end{pmatrix}^T
    \begin{pmatrix}
    \mathbf{d}^\delta - \mathbf{d}^k
    \end{pmatrix}
    \]

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $\mathbf{F}(\mathbf{x}) = \mathbf{d}$, given output data $\mathbf{d}$ with error, in the form of $\mathbf{d}^{\delta}$, i.e., to minimize $\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^{\delta}\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $\mathbf{d}^k = \mathbf{F}(\mathbf{x}^k)$.
  - Linearize: $\nabla \mathbf{F}(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = \mathbf{d}^{\delta} - \mathbf{d}^k$.
  - Regularize: $\left((\nabla \mathbf{F}(\mathbf{x}^k))^T \nabla \mathbf{F}(\mathbf{x}^k) + \alpha_k I\right)p = \left((\nabla \mathbf{F}(\mathbf{x}^k))^T(\mathbf{d}^{\delta} - \mathbf{d}^k)\right)$

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $d^k = F(x^k)$.
  - Linearize: $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - Regularize: $\left(\left(\nabla F(x^k)\right)^T \nabla F(x^k) + \alpha_k I\right)p = \left(\nabla F(x^k)\right)^T (d^\delta - d^k)$

- This is a regularization technique for nonlinear problems and is called output least squares.
Another Perspective on LM:

- NB: $\lambda_k$ is not a regularization parameter in usual sense, but rather a tool for efficiently solving a nonlinear system which itself may or may not be regularized.

- However: suppose our objective is to find a least squares solution to the problem $F(x) = d$, given output data $d$ with error, in the form of $d^\delta$, i.e., to minimize $\|F(x) - d^\delta\|^2$.

- In this case, LM amounts to cycles of these three steps:
  - Forward-solve: compute $d^k = F(x^k)$.
  - Linearize: $\nabla F(x^k)(x^{k+1} - x^k) = d^\delta - d^k$.
  - Regularize: $\left(\left(\nabla F(x^k)\right)^T\nabla F(x^k) + \alpha_k I\right)p = \left(\nabla F(x^k)\right)^T(d^\delta - d^k)$

- This is a regularization technique for nonlinear problems and is called output least squares.
Problem is $G(m) = d$ with least squares solution $m^*$:

Now what? What statistics can we bring to bear on the problem?

- We minimize $\|F(m)\|^2 = \sum_{i=1}^{n} \left( \frac{(G(m) - d_i)^2}{\sigma_i^2} \right)$
- Treat the linear model as locally accurate, so misfit is $\nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m$
- Obtain covariance matrix $\text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$
- If $\sigma$ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for $\sigma$ in $\frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$ the estimate $s^2 = \frac{1}{m-n} \sum_{i=1}^{m} (G(m) - d_i)^2$.
- Do confidence intervals, $\chi^2$ statistic and $p$-value as in Chapter 2.
Problem is $G(m) = d$ with least squares solution $m^*$.

Now what? What statistics can we bring to bear on the problem?

- We minimize $\| F(m) \|^2 = \sum_{i=1}^{n} \frac{(G(m) - d_i)^2}{\sigma_i^2}$

- Treat the linear model as locally accurate, so misfit is $\nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m$

- Obtain covariance matrix $\text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$

- If $\sigma$ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for $\sigma$ in $\frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$ the estimate $s^2 = \frac{1}{m-n} \sum_{i=1}^{m} (G(m) - d_i)^2$.

- Do confidence intervals, $\chi^2$ statistic and $p$-value as in Chapter 2.
Problem is $G(m) = d$ with least squares solution $m^*$:

Now what? What statistics can we bring to bear on the problem?

- We minimize $\|F(m)\|^2 = \sum_{i=1}^{n} \frac{(G(m) - d_i)^2}{\sigma_i^2}$

- Treat the linear model as locally accurate, so misfit is $\nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m$

- Obtain covariance matrix $\text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$

- If $\sigma$ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for $\sigma$ in $\frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$ the estimate

$$s^2 = \frac{1}{m-n} \sum_{i=1}^{m} (G(m) - d_i)^2 .$$

- Do confidence intervals, $\chi^2$ statistic and $p$-value as in Chapter 2.
Problem is $G(m) = d$ with least squares solution $m^*$:

Now what? What statistics can we bring to bear on the problem?

- We minimize $\|F(m)\|^2 = \sum_{i=1}^{n} \frac{(G(m) - d_i)^2}{\sigma_i^2}$

- Treat the linear model as locally accurate, so misfit is
  $\nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m$

- Obtain covariance matrix
  $\text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$

- If $\sigma$ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for $\sigma$ in $\frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$ the estimate
  $s^2 = \frac{1}{m-n} \sum_{i=1}^{m} (G(m) - d_i)^2$.

- Do confidence intervals, $\chi^2$ statistic and $p$-value as in Chapter 2.
Problem is $G(m) = d$ with least squares solution $m^*$:

Now what? What statistics can we bring to bear on the problem?

- We minimize $\|F(m)\|^2 = \sum_{i=1}^{n} \frac{(G(m) - d_i)^2}{\sigma_i^2}$

- Treat the linear model as locally accurate, so misfit is
  \[ \nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m \]

- Obtain covariance matrix
  \[ \text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1} \]

- If $\sigma$ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for $\sigma$ in $\frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1}$ the estimate
  \[ s^2 = \frac{1}{m-n} \sum_{i=1}^{m} (G(m) - d_i)^2. \]

- Do confidence intervals, $\chi^2$ statistic and $p$-value as in Chapter 2.
Problem is \( G(m) = d \) with least squares solution \( m^* \):

Now what? What statistics can we bring to bear on the problem?

- We minimize \( \| F(m) \|^2 = \sum_{i=1}^{n} \frac{(G(m) - d_i)^2}{\sigma_i^2} \)

- Treat the linear model as locally accurate, so misfit is \( \nabla F = F(m + \Delta m) - F(m^*) \approx \nabla F(m^*) \nabla m \)

- Obtain covariance matrix \( \text{Cov}(m^*) = \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1} \)

- If \( \sigma \) is unknown but constant across measurements, take \( \sigma_i = 1 \) above and use for \( \sigma \) in \( \frac{1}{\sigma^2} \left( \nabla F(m^*)^T \nabla F(m^*) \right)^{-1} \) the estimate

\[
s^2 = \frac{1}{m - n} \sum_{i=1}^{m} (G(m) - d_i)^2.
\]

- Do confidence intervals, \( \chi^2 \) statistic and \( p \)-value as in Chapter 2.
What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.
What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.
What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.
What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britain population data.
Implementation Issues

What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.
Implementation Issues

What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britain population data.
Implementation Issues

What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.
What could go wrong?

- Problem may have many local minima.
- Even if it has a unique solution, it might lie in a long flat basin.
- Analytical derivatives may not be available. This presents an interesting regularization issue not discussed by the authors. We do so at the board.
- One remedy for first problem: use many starting points and statistics to choose best local minimum.
- One remedy for second problem: use a better technique than GN or LM.
- Do Example 9.2 from the CD to illustrate some of these ideas.
- If time permits, do data fitting from Great Britian population data.