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

Instructor: Thomas Shores
Department of Mathematics

Lecture 26, April 18, 2006
AvH 10

Outline

Basic Problems

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

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

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- 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

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

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

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- 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

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

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

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- 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

Root Finding:

Solve the system of equations represented in vector form as

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}.$$

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

- Here $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ and $x = (x_1, \ldots, x_m)$
- 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 $f(\mathbf{x})$, where $\mathbf{x}$ ranges over a feasible set Ω.

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x})\right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

Basic Problems

Optimization:

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

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x})\right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

Basic Problems

Optimization:

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

- Set $\mathbf{F}(\mathbf{x}) = \nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_m}(\mathbf{x})\right)$
- Hessian of f: $\nabla(\nabla f(\mathbf{x})) \equiv \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

Taylor Theorems

First Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous second partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^2\right), \ \mathbf{x} \to \mathbf{x}.$$

Second Order

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ has continuous third partials and $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then $f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^3\right), \ \mathbf{x} \to \mathbf{x}.$
(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 $\mathbf{x}^*, \mathbf{x} \in \mathbb{R}^n$. Then
$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \mathcal{O}\left(\|\mathbf{x} - \mathbf{x}^*\|^2\right), \mathbf{x} \to \mathbf{x}.$$

Second Order

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

Newton Algorithms

Root Finding

Input $\mathbf{F}$, $\nabla\mathbf{F}$, $\mathbf{x}^0$, N_{max}
for $k = 0, \ldots, N_{max}$
  $\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla\mathbf{F}\left(\mathbf{x}^k\right)^{-1}\mathbf{F}\left(\mathbf{x}^k\right)$
  if $\mathbf{x}^{k+1}, \mathbf{x}^k$ pass a convergence test
    return$(\mathbf{x}^k)$
  end
end
return$(\mathbf{x}^{N_{max}})$

Convergence Result

Theorem

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

$$\left\|\mathbf{x}^{k+1} - \mathbf{x}^*\right\| \leq K \left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2.$$

Newton for Optimization

Bright Idea:

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

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

Newton for Optimization

Bright Idea:

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

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

Newton for Optimization

Bright Idea:

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

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

Newton for Optimization

Bright Idea:

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

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

Newton for Optimization

Bright Idea:

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

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

Newton for Optimization

Bright Idea:

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

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

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 $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇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 $f(\mathbf{x})$, 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 $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇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 $f(\mathbf{x})$, 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 $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇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 $f(\mathbf{x})$, 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 $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇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 $f(\mathbf{x})$, 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 $-\nabla \mathbf{F}\left(\mathbf{x}^k\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right)$ for a point that (approximately) minimizes a merit function like $m(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|^2$.
- Optimization is NOT a special case of root finding. There are special characteristics of the min $f(\mathbf{x})$ problem that get lost if one only tries to find a zero of ∇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 $f(\mathbf{x})$, in any search direction. Using this helps avoid saddle points, maxima.

Outline

Gauss-Newton and Levenberg-Marquardt

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.

Gauss-Newton and Levenberg-Marquardt

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

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

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

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

- There's a problem here. See it?

The Problem:

Given a function $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, minimize
$f(\mathbf{x}) = \sum_{k=0}^{m} f_k(\mathbf{x})^2 = \|\mathbf{F}(\mathbf{x})\|^2$.

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

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

- There's a problem here. See it?

The Problem:

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

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

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

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

- There's a problem here. See it?

The Problem:

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

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

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

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

- There's a problem here. See it?

The Problem:

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

- A remedy: regularize the Newton problem to
$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right)\mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$$
with λ suitably chosen positive number for $\mathbf{p} = \mathbf{x} - \mathbf{x}^k$
- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing $\left\|\nabla \mathbf{F}\left(\mathbf{x}^k\right)\mathbf{p} + \mathbf{F}\left(\mathbf{x}^k\right)\right\|^2$ subject to a constraint $\|\mathbf{p}\| \leq \delta_k$. Of course, δ_k determines λ_k and vice-versa.
- The idea is to choose λ_k at each step: Increase it if the reduction in $f(\mathbf{x})$ was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.

The Problem:

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

- A remedy: regularize the Newton problem to
$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$$
with λ suitably chosen positive number for $\mathbf{p} = \mathbf{x} - \mathbf{x}^k$
- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing $\left\|\nabla \mathbf{F}\left(\mathbf{x}^k\right)\mathbf{p} + \mathbf{F}\left(\mathbf{x}^k\right)\right\|^2$ subject to a constraint $\|\mathbf{p}\| \leq \delta_k$. Of course, δ_k determines λ_k and vice-versa.
- The idea is to choose λ_k at each step: Increase it if the reduction in $f(\mathbf{x})$ was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.

The Problem:

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

- A remedy: regularize the Newton problem to
$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$$
with λ suitably chosen positive number for $\mathbf{p} = \mathbf{x} - \mathbf{x}^k$
- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing $\left\|\nabla \mathbf{F}\left(\mathbf{x}^k\right) \mathbf{p} + \mathbf{F}\left(\mathbf{x}^k\right)\right\|^2$ subject to a constraint $\|\mathbf{p}\| \leq \delta_k$. Of course, δ_k determines λ_k and vice-versa.
- The idea is to choose λ_k at each step: Increase it if the reduction in $f(\mathbf{x})$ was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.

The Problem:

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

- A remedy: regularize the Newton problem to
$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$$
with λ suitably chosen positive number for $\mathbf{p} = \mathbf{x} - \mathbf{x}^k$
- In fact, Lagrange multipliers show we are really solving a constrained problem of minimizing $\left\|\nabla \mathbf{F}\left(\mathbf{x}^k\right) \mathbf{p} + \mathbf{F}\left(\mathbf{x}^k\right)\right\|^2$ subject to a constraint $\|\mathbf{p}\| \leq \delta_k$. Of course, δ_k determines λ_k and vice-versa.
- The idea is to choose λ_k at each step: Increase it if the reduction in $f(\mathbf{x})$ was not as good as expected, and decrease it if the reduction was better than expected. Otherwise, leave it alone.

LM

More on LM:

$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right).$$

- For small λ_k, LM becomes approximately $\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$ which is GN with its favorable convergence rate.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, which is a steepest-descent step, slow but convergent.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, 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:

$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right)\mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right).$$

- For small λ_k, LM becomes approximately $\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right)\mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$ which is GN with its favorable convergence rate.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, which is a steepest-descent step, slow but convergent.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, 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:

$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right)\mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right).$$

- For small λ_k, LM becomes approximately $\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right)\mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$ which is GN with its favorable convergence rate.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, which is a steepest-descent step, slow but convergent.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, 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.

LM

More on LM:

$$\left(\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) + \lambda_k I\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right).$$

- For small λ_k, LM becomes approximately $\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla \mathbf{F}\left(\mathbf{x}^k\right) \mathbf{p} = -\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$ which is GN with its favorable convergence rate.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, which is a steepest-descent step, slow but convergent.
- For large λ_k, LM becomes approximately $\mathbf{p} = -\frac{1}{\lambda_k}\left(\nabla \mathbf{F}\left(\mathbf{x}^k\right)\right)^T \mathbf{F}\left(\mathbf{x}^k\right)$, 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:

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

- For small λ_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 λ_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 λ_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.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

LM

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Another Perspective on LM:

- NB: λ_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 $\left\|\mathbf{F}(\mathbf{x}) - \mathbf{d}^\delta\right\|^2$.
- In this case, LM amounts to cycles of these three steps:
- Forward-solve: compute $\mathbf{d}^k = \mathbf{F}\left(\mathbf{x}^k\right)$.
- Linearize: $\nabla\mathbf{F}\left(\mathbf{x}^k\right)\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = \mathbf{d}^\delta - \mathbf{d}^k$.
- Regularize: $\left(\left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T \nabla\mathbf{F}\left(\mathbf{x}^k\right) + \alpha_k I\right)\mathbf{p} = \left(\nabla\mathbf{F}\left(\mathbf{x}^k\right)\right)^T\left(\mathbf{d}^\delta - \mathbf{d}^k\right)$
- This is a regularization technique for nonlinear problems and is called **output least squares**.

Outline

Statistics

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

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

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

- We minimize $\|\mathbf{F}(\mathbf{m})\|^2 = \sum_{i=1}^{n} \frac{(G(\mathbf{m}) - d_i)^2}{\sigma_i^2}$
- Treat the linear model as locally accurate, so misfit is $\nabla\mathbf{F} = \mathbf{F}(\mathbf{m} + \Delta\mathbf{m}) - \mathbf{F}(\mathbf{m}^*) \approx \nabla\mathbf{F}(\mathbf{m}^*)\nabla\mathbf{m}$
- Obtain covariance matrix $\mathrm{Cov}(\mathbf{m}^*) = \left(\nabla\mathbf{F}(\mathbf{m}^*)^T \nabla\mathbf{F}(\mathbf{m}^*)\right)^{-1}$
- If σ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for σ in $\frac{1}{\sigma^2}\left(\nabla\mathbf{F}(\mathbf{m}^*)^T \nabla\mathbf{F}(\mathbf{m}^*)\right)^{-1}$ the estimate

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

- Do confidence intervals, χ^2 statistic and p-value as in Chapter 2.

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

- We minimize $\|\mathbf{F}(\mathbf{m})\|^2 = \sum_{i=1}^{n} \frac{(G(\mathbf{m}) - d_i)^2}{\sigma_i^2}$
- Treat the linear model as locally accurate, so misfit is $\nabla \mathbf{F} = \mathbf{F}(\mathbf{m} + \Delta \mathbf{m}) - \mathbf{F}(\mathbf{m}^*) \approx \nabla \mathbf{F}(\mathbf{m}^*) \nabla \mathbf{m}$
- Obtain covariance matrix $\mathrm{Cov}(\mathbf{m}^*) = \left(\nabla \mathbf{F}(\mathbf{m}^*)^T \nabla \mathbf{F}(\mathbf{m}^*)\right)^{-1}$
- If σ is unknown but constant across measurements, take $\sigma_i = 1$ above and use for σ in $\frac{1}{\sigma^2}\left(\nabla \mathbf{F}(\mathbf{m}^*)^T \nabla \mathbf{F}(\mathbf{m}^*)\right)^{-1}$ the estimate

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

- Do confidence intervals, χ^2 statistic and p-value as in Chapter 2.

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

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

Statistics

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

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

Statistics

Problem is $G(\mathbf{m}) = \mathbf{d}$ with least squares solution $\mathbf{m}^*$:

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

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

Outline

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 fiting 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 fiting 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 fiting 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 fiting 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 fiting 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 fiting 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 fiting 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 fiting from Great Britian population data.