13. Nonlinear equations

- Newton method for nonlinear equations
- damped Newton method for unconstrained minimization
- Newton method for nonlinear least squares
Set of nonlinear equations

\( n \) nonlinear equations in \( n \) variables \( x_1, x_2, \ldots, x_n \):

\[
\begin{align*}
    f_1(x_1, \ldots, x_n) &= 0 \\
    f_2(x_1, \ldots, x_n) &= 0 \\
    & \vdots \\
    f_n(x_1, \ldots, x_n) &= 0
\end{align*}
\]

in vector notation: \( f(x) = 0 \) with

\[
x = \begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{bmatrix}, \quad f(x) = \begin{bmatrix}
    f_1(x_1, \ldots, x_n) \\
    f_2(x_1, \ldots, x_n) \\
    \vdots \\
    f_n(x_1, \ldots, x_n)
\end{bmatrix}
\]
Example: nonlinear resistive circuit

\[ g(x) - \frac{E - x}{R} = 0 \]

a nonlinear equation in the variable \( x \), with three solutions
Newton method

suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ is differentiable

**Algorithm:** choose $x^{(1)}$ and repeat for $k = 1, 2, \ldots$

$$x^{(k+1)} = x^{(k)} - Df(x^{(k)})^{-1} f(x^{(k)})$$

- each iteration requires one evaluation of $f(x)$ and $Df(x)$
- each iteration requires factorization of $n \times n$ matrix $Df(x)$
- we assume $Df(x)$ is nonsingular
Interpretation

\[ x^+ = x - Df(x)^{-1}f(x) \quad (x = x^{(k)}, \ x^+ = x^{(k+1)}) \]

- linearize \( f \) (\textit{i.e.}, make affine approximation) around current iterate \( x \)

\[
\hat{f}(y) = f(x) + Df(x)(y - x)
\]

- solve linearized equation \( \hat{f}(y) = 0 \)

\[ y = x - Df(x)^{-1}f(x) \]

- take \( y \) as new iterate \( x^+ \)
One variable

\[ \hat{f}(y) = f(x) + f'(x)(y - x) \]

- affine approximation of \( f \) around \( x \) is \( \hat{f}(y) = f(x) + f'(x)(y - x) \)
- solve the linearized equation \( \hat{f}(y) = 0 \) and take the solution \( y \) as \( x^+ \):

\[ x^+ = x - \frac{f(x)}{f'(x)} \]
Example: nonlinear resistive circuit

- nonlinear resistors with \( i-v \) characteristics \( i_1 = g_1(v_1), \ i_2 = g_2(v_2) \)
- two nonlinear equations in two variables \( v_1, v_2 \)

\[
\begin{align*}
    f_1(v_1, v_2) &= g_1(v_1) + \frac{v_1 - E}{R} + \frac{v_1 - v_2}{R} = 0 \\
    f_2(v_1, v_2) &= g_2(v_2) + \frac{v_2 - v_1}{R} = 0
\end{align*}
\]

- derivative matrix

\[
Df(v) = \begin{bmatrix}
    g_1'(v_1) + \frac{2}{R} & -\frac{1}{R} \\
    -\frac{1}{R} & g_2'(v_2) + \frac{1}{R}
\end{bmatrix}
\]
Linearized equations

• linearized equations around $\hat{v}_1$, $\hat{v}_2$:

$$g_1(\hat{v}_1) - g'_1(\hat{v}_1)\hat{v}_1 + g'_1(\hat{v}_1)v_1 + \frac{v_1 - E}{R} + \frac{v_1 - v_2}{R} = 0$$

$$g_2(\hat{v}_2) - g'_2(\hat{v}_2)\hat{v}_2 + g'_2(\hat{v}_2)v_2 + \frac{v_2 - v_1}{R} = 0$$

• describes a linear resistive circuit

\[\begin{align*}
\hat{I}_1 &= g_1(\hat{v}_1) - g'_1(\hat{v}_1)\hat{v}_1, & \hat{R}_1 &= 1/g'_1(\hat{v}_1) \\
\hat{I}_2 &= g_2(\hat{v}_2) - g'_2(\hat{v}_2)\hat{v}_2, & \hat{R}_2 &= 1/g'_2(\hat{v}_2)
\end{align*}\]
Relation to Gauss-Newton method

recall Gauss-Newton method for nonlinear least squares problem

\[
\text{minimize} \quad \| f(x) \|^2
\]

where \( f \) is a differentiable function from \( \mathbb{R}^n \) to \( \mathbb{R}^m \)

- Gauss-Newton update

\[
x^+ = x - (Df(x)^T Df(x))^{-1} Df(x)^T f(x)
\]

- if \( m = n \), then \( Df(x) \) is square and this is the Newton update

\[
x^+ = x - Df(x)^{-1} f(x)
\]
Example 1

Newton method applied to

\[ f(x) = e^x - e^{-x}, \quad x^{(1)} = 10 \]
Example 2

\[ f(x) = e^x - e^{-x} - 3x \]

- starting point \( x^{(1)} = -1 \): converges to \( x^* = -1.62 \)
- starting point \( x^{(1)} = -0.8 \): converges to \( x^* = 1.62 \)
- starting point \( x^{(1)} = -0.7 \): converges to \( x^* = 0 \)

converges to a different solution depending on the starting point
Example 3

\[ f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]

- starting point \( x^{(1)} = 0.9 \): converges very rapidly to \( x^* = 0 \)
- starting point \( x^{(1)} = 1.1 \): does not converge
Example 4

\[ f_1(x_1, x_2) = \log(x_1^2 + 2x_2^2 + 1) - 0.5 = 0 \]
\[ f_2(x_1, x_2) = x_2 - x_1^2 + 0.2 = 0 \]

two equations in two variables; two solutions \((0.70, 0.29), (-0.70, 0.29)\)
Example 4

Newton iteration

• evaluate \( g = f(x) \) and

\[
H = Df(x) = \begin{bmatrix}
2x_1/(x_1^2 + 2x_2^2 + 1) & 4x_2/(x_1^2 + 2x_2^2 + 1) \\
-2x_1 & 1
\end{bmatrix}
\]

• solve \( Hv = -g \) (two linear equations in two variables)

• update \( x := x + v \)

Results

• \( x^{(1)} = (1, 1) \): converges to \( x^* = (0.70, 0.29) \) in about 4 iterations
• \( x^{(1)} = (-1, 1) \): converges to \( x^* = (-0.70, 0.29) \) in about 4 iterations
• \( x^{(1)} = (1, -1) \) or \( x^{(0)} = (-1, -1) \): does not converge
Observations

• Newton’s method works very well if started near a solution

• may not work at all otherwise

• can converge to different solutions depending on the starting point

• does not necessarily find the solution closest to the starting point
Convergence of Newton’s method

if $f(x^*) = 0$ and $Df(x^*)$ is nonsingular, and $x^{(1)}$ is close to $x^*$, then

$$x^{(k)} \to x^*, \quad \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2$$

for some $c > 0$

• this is called quadratic convergence

• explains fast convergence when started near solution

• in practice, we don’t know what $c$ is, or how close $x^{(1)}$ has to be
Outline

• Newton’s method for sets of nonlinear equations

• damped Newton for unconstrained minimization

• Newton method for nonlinear least squares
Unconstrained minimization problem

\[ \text{minimize } g(x_1, x_2, \ldots, x_n) \]

\( g \) is a function from \( \mathbb{R}^n \) to \( \mathbb{R} \)

- \( x = (x_1, x_2, \ldots, x_n) \) is \( n \)-vector of optimization variables
- \( g(x) \) is the cost function or objective function
- to solve a maximization problem (i.e., maximize \( g(x) \)), minimize \( -g(x) \)
- we will assume that \( g \) is twice differentiable
Local and global optimum

\( x^\star \) is an optimal point (or a minimum) if

\[ g(x^\star) \leq g(x) \quad \text{for all } x \]

also called globally optimal

\( x^\star \) is a locally optimal point (local minimum) if for some \( R > 0 \)

\[ g(x^\star) \leq g(x) \quad \text{for all } x \text{ with } \|x - x^\star\| \leq R \]

Example

\( y \) is locally optimal

\( z \) is (globally) optimal
Gradient and Hessian

Gradient of $g : \mathbb{R}^n \to \mathbb{R}$ at $z \in \mathbb{R}^n$ is the $n$-vector

$$\nabla g(z) = \left( \frac{\partial g}{\partial x_1}(z), \frac{\partial g}{\partial x_2}(z), \ldots, \frac{\partial g}{\partial x_n}(z) \right)$$

Hessian of $g$ at $z$: a symmetric $n \times n$ matrix $\nabla^2 g(z)$ with elements

$$\nabla^2 g(z)_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}(z)$$

this is also the derivative matrix $Df(z)$ of $f(x) = \nabla g(x)$ at $z$

Quadratic (second order) approximation of $g$ around $z$: 

$$g_q(x) = g(z) + \nabla g(z)^T(x - z) + \frac{1}{2}(x - z)^T\nabla^2 g(z)(x - z)$$
Examples

Affine function: \( g(x) = a^T x + b \)

\[ \nabla g(x) = a, \quad \nabla^2 g(x) = 0 \]

Quadratic function: \( g(x) = x^T P x + q^T x + r \) with \( P \) symmetric

\[ \nabla g(x) = 2P x + q, \quad \nabla^2 g(x) = 2P \]

Least squares cost: \( g(x) = \| Ax - b \|^2 = x^T A^T A x - 2b^T A x + b^T b \)

\[ \nabla g(x) = 2A^T A x - 2A^T b, \quad \nabla^2 g(x) = 2A^T A \]
Properties

**Linear combination:** if \( g(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) \), then

\[
\nabla g(x) = \alpha_1 \nabla g_1(x) + \alpha_2 \nabla g_2(x)
\]
\[
\nabla^2 g(x) = \alpha_1 \nabla^2 g_1(x) + \alpha_2 \nabla^2 g_2(x)
\]

**Composition with affine mapping:** if \( g(x) = h(Cx + d) \), then

\[
\nabla g(x) = C^T \nabla h(Cx + d)
\]
\[
\nabla^2 g(x) = C^T \nabla^2 h(Cx + d) C
\]
Example

\[ g(x_1, x_2) = e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1} \]

Gradient

\[ \nabla g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} - e^{-x_1-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} \end{bmatrix} \]

Hessian

\[ \nabla^2 g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1} & e^{x_1+x_2-1} - e^{x_1-x_2-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} & e^{x_1+x_2-1} + e^{x_1-x_2-1} \end{bmatrix} \]
Gradient and Hessian via composition property

express $g$ as $g(x) = h(Cx + d)$ with $h(y_1, y_2, y_3) = e^{y_1} + e^{y_2} + e^{y_3}$ and

$$C = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}, \quad d = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Gradient: $\nabla g(x) = C^T \nabla h(Cx + d)$

$$\nabla g(x) = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1 + x_2 - 1} \\ e^{x_1 - x_2 - 1} \\ e^{-x_1 - 1} \end{bmatrix}$$

Hessian: $\nabla^2 g(x) = C^T \nabla^2 h^2(Cx + d)C$

$$\nabla^2 g(x) = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1 + x_2 - 1} & 0 & 0 \\ 0 & e^{x_1 - x_2 - 1} & 0 \\ 0 & 0 & e^{-x_1 - 1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}$$
Optimality conditions for twice differentiable $g$

**Necessary condition:** if $x^*$ is locally optimal, then

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive semidefinite}$$

**Sufficient condition:** if $x^*$ satisfies

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive definite}$$

then $x^*$ is locally optimal

**Necessary and sufficient condition for convex functions**

- $g$ is called *convex* if $\nabla^2 g(x)$ is positive semidefinite everywhere
- if $g$ is convex then $x^*$ is optimal if and only if $\nabla g(x^*) = 0$
Examples ($n = 1$)

- $g(x) = \log(e^x + e^{-x})$
  
  $$g'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad g''(x) = \frac{4}{(e^x + e^{-x})^2}$$

  $g''(x) \geq 0$ everywhere; $x^* = 0$ is the unique optimal point

- $g(x) = x^4$

  $$g'(x) = 4x^3, \quad g''(x) = 12x^2$$

  $g''(x) \geq 0$ everywhere; $x^* = 0$ is the unique optimal point

- $g(x) = x^3$

  $$g'(x) = 3x^2, \quad g''(x) = 6x$$

  $g'(0) = 0$, $g''(0) = 0$ but $x = 0$ is not locally optimal
Examples

• $g(x) = x^T P x + q^T x + r$ ($P$ is positive definite)

$$\nabla g(x) = 2P x + q, \quad \nabla^2 g(x) = 2P$$

$\nabla^2 g(x)$ is positive definite everywhere, hence the unique optimal point is

$$x^* = -(1/2) P^{-1} q$$

• $g(x) = \|Ax - b\|^2$ ($A$ is a matrix with linearly independent columns)

$$\nabla g(x) = 2 A^T A x - 2 A^T b, \quad \nabla^2 g(x) = 2 A^T A$$

$\nabla^2 g(x)$ is positive definite everywhere, hence the unique optimal point is

$$x^* = (A^T A)^{-1} A^T b$$
Examples

example of page 13-22: we can express $\nabla^2 g(x)$ as

$$\nabla^2 g(x) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} & 0 & 0 \\ 0 & e^{x_1-x_2-1} & 0 \\ 0 & 0 & e^{-x_1-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix}$$

this shows that $\nabla^2 g(x)$ is positive definite for all $x$

therefore $x^*$ is optimal if and only if

$$\nabla g(x^*) = \begin{bmatrix} e^{x_1^*+x_2^*-1} + e^{x_1^*-x_2^*-1} - e^{-x_1^*-1} \\ e^{x_1^*+x_2^*-1} - e^{x_1^*-x_2^*-1} \end{bmatrix} = 0$$

two nonlinear equations in two variables
Newton’s method for minimizing a convex function

if $\nabla^2 g(x)$ is positive definite everywhere, we can minimize $g(x)$ by solving

$$\nabla g(x) = 0$$

Algorithm: choose $x^{(1)}$ and repeat for $k = 1, 1, 2, \ldots$

$$x^{(k+1)} = x^{(k)} - \nabla^2 g(x^{(k)})^{-1} \nabla g(x^{(k)})$$

• $v = -\nabla^2 g(x)^{-1} \nabla g(x)$ is called the Newton step at $x$

• converges if started sufficiently close to the solution

• Newton step computed by a Cholesky factorization of the Hessian
Interpretations of Newton step

Affine approximation of gradient

• affine approximation of $f(y) = \nabla g(y)$ around $x$ is

$$\hat{f}(y) = \nabla g(x) + \nabla^2 g(x)(y - x)$$

• Newton update $x + v$ is solution of linear equation $\hat{f}(y) = 0$

Quadratic approximation of function

• quadratic approximation of $g(y)$ around $x$ is

$$g_q(y) = g(x) + \nabla g(x)^T(y - x) + \frac{1}{2}(y - x)^T\nabla^2 g(x)(y - x)$$

• Newton update $x + v$ is minimizer of $g_q$ (solution of $\nabla g_q(y) = 0$)
Example \((n = 1)\)

\[
g_q(y) = g(x) + g'(x)(y - x) + \frac{g''(x)}{2}(y - x)^2
\]

Nonlinear equations 13-30
Example

\[ g(x) = \log(e^x + e^{-x}), \quad g'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad g''(x) = \frac{4}{(e^x + e^{-x})^2} \]

does not converge when started at \( x^{(1)} = 1.15 \)
Damped Newton method

use update $x^+ = x + tv$ and choose $t$ so that $g(x + tv) < g(x)$

Algorithm: choose $x^{(1)}$ and repeat for $k = 1, 2, \ldots$

1. compute Newton step $v = -\nabla^2 g(x^{(k)})^{-1}\nabla g(x^{(k)})$
2. find largest $t$ in $\{1, 0.5, 0.5^2, 0.5^3, \ldots\}$ that satisfies

$$g(x^{(k)} + tv) \leq g(x^{(k)}) + \alpha t \nabla g(x^{(k)})^T v$$

and take $x^{(k+1)} = x^{(k)} + tv$

- $\alpha$ is an algorithm parameter (small and positive, e.g., $\alpha = 0.01$)
- $t$ is called the step size; step 2 in algorithm is called line search
**Interpretation of line search**

to determine a suitable step size, consider the function $h : \mathbb{R} \rightarrow \mathbb{R}$

$$h(t) = g(x + tv)$$

$x = x^{(k)}$ is the current iterate; $v$ is the Newton step at $x$

- $h'(0) = \nabla g(x)^T v$ is the *directional derivative* of $g$ at $x$ in direction $v$
- affine approximation of $h$ at $t = 0$ is

$$\hat{h}(t) = h(0) + h'(0)t = g(x) + t\nabla g(x)^T v$$

- line search accepts step size $t$ if $g(x + tv) \leq g(x) + \alpha t\nabla g(x)^T v$; i.e.,

$$h(t) - h(0) \leq \alpha(\hat{h}(t) - h(0))$$

decrease $h(t) - h(0)$ is at least $\alpha$ times what is expected based on $\hat{h}$
Interpretation of line search

start with $t = 1$; divide $t$ by 2 until $h(t) \leq h(0) + \alpha h'(0)t$

• works if $h'(0) = \nabla g(x)^T v < 0$ ($v$ is a descent direction)

• if $\nabla^2 g(x)$ is positive definite, the Newton step is a descent direction

$$h'(0) = \nabla g(x)^T v = v^T \nabla^2 g(x) v < 0$$
Example

\[ g(x) = \log(e^x + e^{-x}), \quad x^{(0)} = 4 \]

close to the solution: very fast convergence, no backtracking steps
Example

equation of page 13-22

\[ g(x_1, x_2) = e^{x_1 + x_2 - 1} + e^{x_1 - x_2 - 1} + e^{-x_1 - 1} \]

damped Newton method started at \( x = (-2, 2) \)

![Graph showing the damped Newton method](image-url)
Newton method for nonconvex functions

if $\nabla^2 g(x)$ is not positive definite, it is possible that Newton step $v$ satisfies

$$\nabla g(x)^T v = -\nabla g(x)^T \nabla^2 g(x)^{-1} \nabla g(x) > 0$$

- if Newton step is not descent direction, replace it with descent direction
- simplest choice is $v = -\nabla g(x)$; practical methods make other choices
Outline

• Newton’s method for sets of nonlinear equations
• damped Newton for unconstrained minimization
• Newton method for nonlinear least squares
Hessian of nonlinear least squares cost

\[ g(x) = \| f(x) \|^2 = \sum_{i=1}^{m} f_i(x)^2 \]

- gradient (from page 12-13):

\[ \nabla g(x) = 2 \sum_{i=1}^{m} f_i(x) \nabla f_i(x) = 2 Df(x)^T f(x) \]

- second derivatives:

\[ \frac{\partial^2 g}{\partial x_j \partial x_k}(x) = 2 \sum_{i=1}^{m} \left( \frac{\partial f_i}{\partial x_j}(x) \frac{\partial f_i}{\partial x_k}(x) + f_i(x) \frac{\partial^2 f_i}{\partial x_j \partial x_k}(x) \right) \]

- Hessian

\[ \nabla^2 g(x) = 2 Df(x)^T Df(x) + 2 \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \]
Newton and Gauss-Newton steps

(Undamped) Newton step:

\[ v_{nt} = -\nabla^2 g(x)^{-1} \nabla g(x) \]

\[ = - \left( Df(x)^T Df(x) + \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \right)^{-1} Df(x)^T f(x) \]

Gauss-Newton step (from pages 12-15):

\[ v_{gn} = - \left( Df(x)^T Df(x) \right)^{-1} Df(x)^T f(x) \]

- can be written as \( v_{gn} = - H_{gn}^{-1} \nabla g(x) \) where \( H_{gn} = 2Df(x)^T Df(x) \)
- \( H_{gn} \) is the Hessian without the term \( \sum_i f_i(x) \nabla^2 f_i(x) \)
Comparison

Newton step

• requires second derivatives of $f$
• not always a descent direction ($\nabla^2 g(x)$ not necessarily positive definite)
• fast convergence near local minimum

Gauss-Newton step

• does not require second derivatives
• a descent direction (if columns of $Df(x)$ are linearly independent):

$$\nabla g(x)^T v_{gn} = -2v_{gn}^T Df(x)^T Df(x)v_{gn} < 0 \quad \text{if } v_{gn} \neq 0$$

• local convergence to $x^*$ is similar to Newton method if

$$\sum_{i=1}^{m} f_i(x^*) \nabla^2 f_i(x^*)$$

is small (e.g., $f(x^*)$ is small, or $f$ is nearly affine around $x^*$)