Numerical Methods I

MATH-GA 2010.001/CSCI-GA 2420.001

Benjamin Peherstorfer
Courant Institute, NYU

Based on slides by G. Stadler and A. Donev

Today

Last time

- Interpolation

Today

- Solving systems of nonlinear equations
- Bisection method
- Newton method

Announcements

- Homework 6 posted and is due Mon, Dec 5 before class

Solving nonlinear equations ("root finding")

We want to solve the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}.$$

We could also have $n < \infty$ equations in n unknowns with $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\boldsymbol{x}) = \mathbf{0}$$

In general, we will need an iterative approach that constructs $x_1, x_2, x_3, \ldots$ such that

$$\lim_{k\to\infty} x_k = x^*,$$

with $f(x^*) = 0$.

Solving nonlinear equations ("root finding")

We want to solve the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}.$$

We could also have $n < \infty$ equations in n unknowns with $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\boldsymbol{x}) = \boldsymbol{0}$$

In general, we will need an iterative approach that constructs $x_1, x_2, x_3, \dots$ such that

$$\lim_{k\to\infty} x_k = x^*,$$

with $f(x^*) = 0$.

What are important properties of a method for solving nonlinear equations?

Solving nonlinear equations (“root finding”)

We want to solve the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}.$$

We could also have $n < \infty$ equations in n unknowns with $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\boldsymbol{x}) = \boldsymbol{0}$$

In general, we will need an iterative approach that constructs $x_1, x_2, x_3, \ldots$ such that

$$\lim_{k\to\infty} x_k = x^*,$$

with $f(x^*) = 0$.

What are important properties of a method for solving nonlinear equations?

- Does it converge? From which starting point x_0?

Solving nonlinear equations ("root finding")

We want to solve the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}.$$

We could also have $n < \infty$ equations in n unknowns with $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\boldsymbol{x}) = \boldsymbol{0}$$

In general, we will need an iterative approach that constructs $x_1, x_2, x_3, \ldots$ such that

$$\lim_{k \to \infty} x_k = x^*,$$

with $f(x^*) = 0$.

What are important properties of a method for solving nonlinear equations?

- Does it converge? From which starting point x_0?
- How quickly does it converge?

Solving nonlinear equations ("root finding")

We want to solve the nonlinear equation

$$f(x) = 0, \quad x \in \mathbb{R}.$$

We could also have $n < \infty$ equations in n unknowns with $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\boldsymbol{x}) = \boldsymbol{0}$$

In general, we will need an iterative approach that constructs $x_1, x_2, x_3, \ldots$ such that

$$\lim_{k\to\infty} x_k = x^*,$$

with $f(x^*) = 0$.

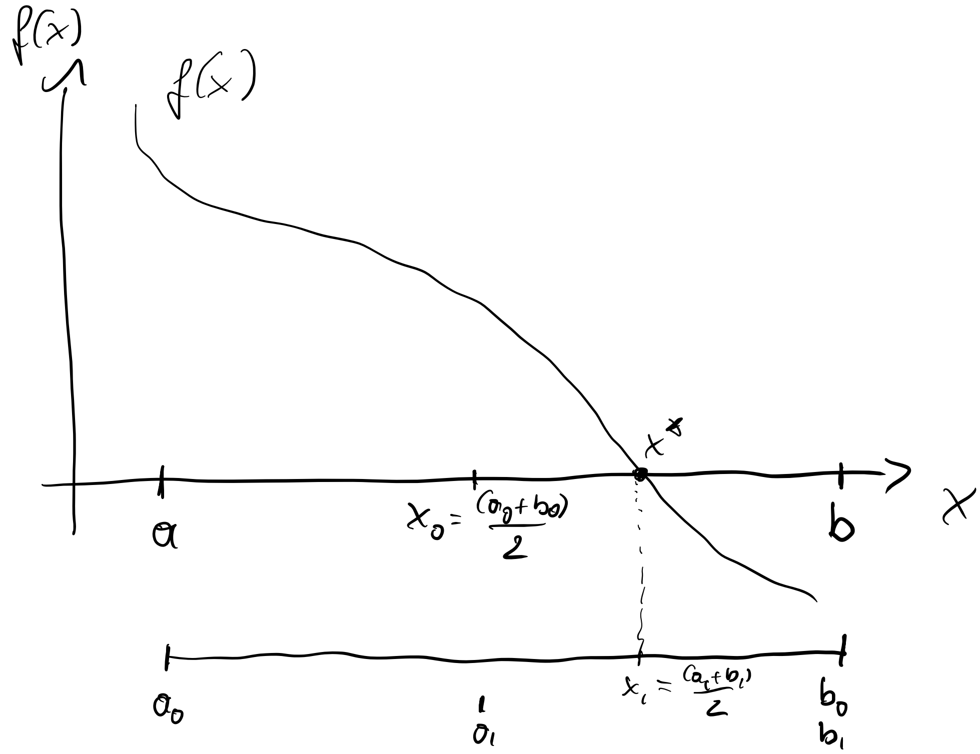
What are important properties of a method for solving nonlinear equations?

- Does it converge? From which starting point x_0?
- How quickly does it converge?
- How expensive is each step?

Bisection method

The bisection method exploits that given a continuous function $f : [a, b] \to \mathbb{R}$, such that $f(a)f(b) < 0$, there exists $x^* \in (a, b)$ with $f(x^*) = 0$

- Assumption: f is continuous over $[a, b]$ (very weak assumption!)
- We have chosen a reasonable interval $[a, b]$ so that there exists a solution $x^* \in (a, b)$ with $f(x^*) = 0$



Convergence: $I_k = [a_k, b_k]$

$e_k = x_k - x^*$ $\qquad |I_k| = b_k - a_k = \dfrac{b-a}{2^k}$

$|e_k| = |x_k - x^*| \leq \frac{1}{2}|I_k| = \dfrac{b-a}{2^{k+1}}$

$\lim\limits_{k \to \infty} |e_k| = 0$

Bisection method

The bisection method exploits that given a continuous function $f : [a, b] \to \mathbb{R}$, such that $f(a)f(b) < 0$, there exists $x^* \in (a, b)$ with $f(x^*) = 0$

- Assumption: f is continuous over $[a, b]$ (very weak assumption!)
- We have chosen a reasonable interval $[a, b]$ so that there exists a solution $x^* \in (a, b)$ with $f(x^*) = 0$

Set $a_0 = a, b_0 = b, x_0 = (a + b)/2$ and iterate for $k = 0, 1, 2, 3, \ldots$ as follows:

1. Set $a_{k+1} = a_k, b_{k+1} = x_k$ if $f(x_k)f(a_k) < 0$
2. Set $a_{k+1} = x_k, b_{k+1} = b_k$ if $f(x_k)f(b_k) < 0$
3. Set $x_{k+1} = (a_{k+1} + b_{k+1})/2$
4. Terminate if $|b_{k+1} - a_{k+1}| \leq \epsilon$

Visualization $\rightsquigarrow$ board

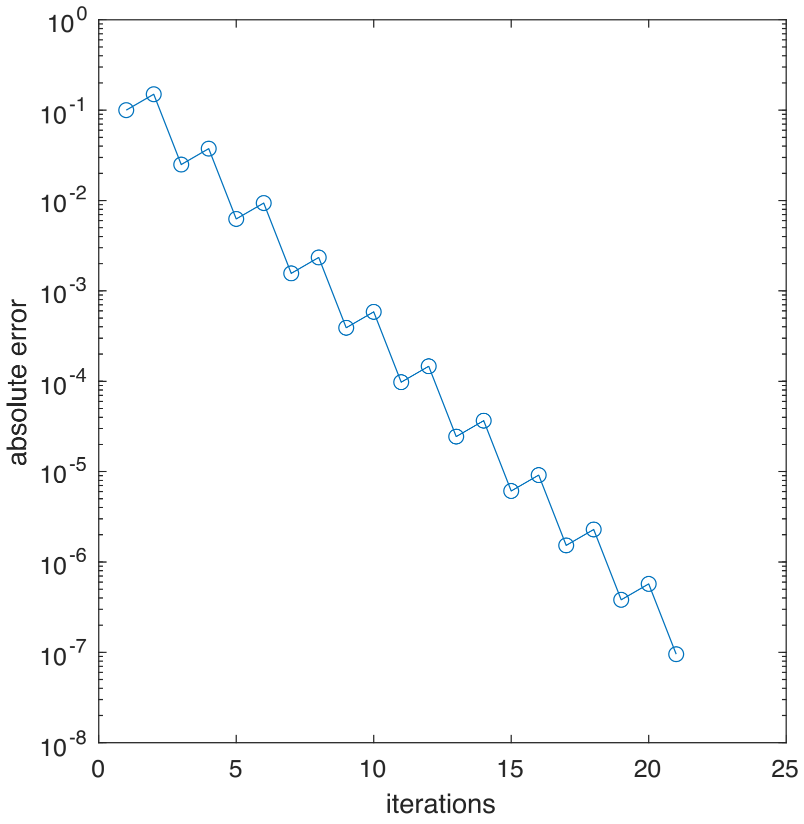
Numerical example

Experiment: Solve $f(x) = x^2 - c = 0$ over $[0.5, 1.5]$ with $c = 0.81$ and $x_0 = 1$

```
a = 0.5; b = 1.5; c = 0.81; xStar = sqrt(c);
f = @(x)x^2 - c;
x = (a + b)/2;

res = [x, xStar];
for k=1:20
    if(f(a)*f(x) < 0)
        b = x;
    else
        a = x;
    end
    x = (a + b)/2;
    res(end + 1, :) = [x, xStar];
end
```

1:	1.0000e+00	9.0000e-01
2:	7.5000e-01	9.0000e-01
3:	8.7500e-01	9.0000e-01
4:	9.3750e-01	9.0000e-01
5:	9.0625e-01	9.0000e-01
6:	8.9062e-01	9.0000e-01
7:	8.9844e-01	9.0000e-01
8:	9.0234e-01	9.0000e-01
9:	9.0039e-01	9.0000e-01
10:	8.9941e-01	9.0000e-01
11:	8.9990e-01	9.0000e-01
12:	9.0015e-01	9.0000e-01
13:	9.0002e-01	9.0000e-01
14:	8.9996e-01	9.0000e-01
15:	8.9999e-01	9.0000e-01
16:	9.0001e-01	9.0000e-01
17:	9.0000e-01	9.0000e-01



- Bisection is a slow but sure method.
- It uses no information about the value of the function or its derivatives - only the sign
- There are variants that achieve faster convergence $\rightsquigarrow$ textbook by Quarterioni
- How can we achieve faster convergence in general?

- Bisection is a slow but sure method.
- It uses no information about the value of the function or its derivatives - only the sign
- There are variants that achieve faster convergence $\rightsquigarrow$ textbook by Quarterioni
- How can we achieve faster convergence in general? $\rightsquigarrow$ need to use additional information, at least the function value instead of just the sign

More general formulation via fixed point iterations

Reformulation as fixed point method so that x^* is fixed point

$$x^* = \Phi(x^*)$$

Corresponding iteration: Choose x_0 (initialization) and compute $x_1, x_2, \ldots$ from

$$x_{k+1} = \Phi(x_k)$$

We now want to study when this iteration converges to x^* with $f(x^*) = 0$

Convergence of fixed point methods

A mapping $\Phi : [a, b] \to \mathbb{R}$ is called contractive on $[a, b]$ if there is a $0 \leq \Theta < 1$ such that

$$|\Phi(x) - \Phi(y)| \leq \Theta|x - y| \text{ for all } x, y \in [a, b].$$

If Φ is continuously differentiable on $[a, b]$, then

$$\Theta = \sup_{x,y\in[a,b]} \frac{|\Phi(x) - \Phi(y)|}{|x - y|} = \sup_{z\in[a,b]} |\Phi'(z)|$$

Convergence of fixed point methods

Let $\Phi : [a, b] \to [a, b]$ be contractive with constant $\Theta < 1$. Then:

- There exists a unique fixed point $\bar{x}$ with $\bar{x} = \Phi(\bar{x})$
- For any starting guess x_0 in $[a, b]$, the fixed point iteration converges to $\bar{x}$ and

  $$|x_{k+1} - x_k| \leq \Theta |x_k - x_{k-1}| \quad \text{(linear convergence)}$$

  and

  $$|\bar{x} - x_k| \leq \frac{\Theta^k}{1 - \Theta} |x_1 - x_0|.$$

The second expression allows to estimate the required number of iterations.

$\rightsquigarrow$ board

For all $x_0 \in I$, $x_0, x_1, x_2, \dots$

$$|x_{q+1} - x_q| = |\phi(x_q) - \phi(x_{q-1})| \leq \Theta \, |x_q - x_{q-1}|$$

By induction

$$|x_{q+1} - x_q| \leq \Theta^q \, |x_1 - x_0|$$

$$|x_{q+m} - x_q| \leq |x_{q+m} - x_{q+m-1}| + \dots + |x_{q+1} - x_q|$$

$$\leq \underbrace{(\Theta^{q+m-1} + \dots + \Theta^q)}_{\Theta^q(1 + \Theta + \dots + \Theta^{m-1})} |x_1 - x_0|$$

$\Theta < 1$

$$\leq \frac{\Theta^q}{1-\Theta} |x_1 - x_0|$$

$\Rightarrow$ convergence $\lim_{q \to \infty} x_q = x^*$

$$|x^* - \phi(x^*)| = |x^* - x_{q+1} + x_{q+1} - \phi(x^*)|$$

$$= |x^* - x_{q+1} + \phi(x_q) - \phi(x^*)|$$

$$\leq \underbrace{|x^* - x_{q+1}|}_{\longrightarrow 0} + \underbrace{|\phi(x_q) - \phi(x^*)|}_{\longrightarrow 0}$$

$$\leq 0$$

$$\Rightarrow |x^* - \phi(x^*)| = 0$$

Assume there are two fix points x^*, y^*

$$0 \le |x^* - y^*| = |\phi(x^*) - \phi(y^*)| \le \underbrace{\theta}_{<1} |x^* - y^*|$$

$$|x^* - y^*| = 0$$

Newton's method

What is the standard approach in numerics when we encounter a nonlinear problem?

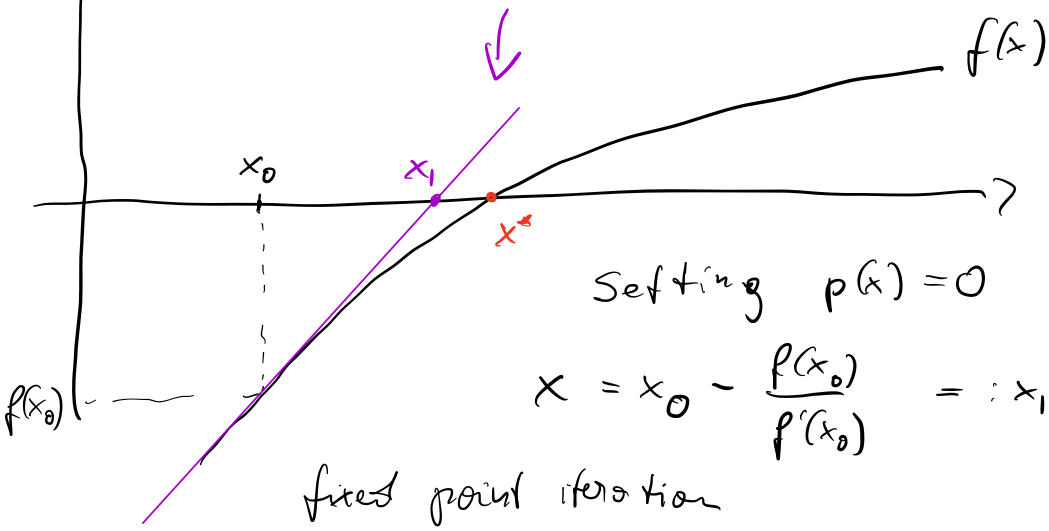
Newton's method

What is the standard approach in numerics when we encounter a nonlinear problem?

$\rightsquigarrow$ we linearize

$\rightsquigarrow$ board

$$p(x) = f(x_0) + f'(x_0)(x - x_0)$$



Setting $p(x) = 0$

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} =: x_1$$

fixed point iteration

$$\phi(x) = x - \frac{f(x)}{f'(x)}$$

In one dimension, solve $f(x) = 0$:

Start with x_0, and compute $x_1, x_2, \ldots$ from

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \ldots$$

Requires $f(x_k) \neq 0$ to be well-defined (i.e., tangent has nonzero slope).

In one dimension, solve $f(x) = 0$:

Start with x_0, and compute $x_1, x_2, \ldots$ from

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \ldots$$

Requires $f(x_k) \neq 0$ to be well-defined (i.e., tangent has nonzero slope).

Experiment: Solve $f(x) = x^2 - c = 0$ with $c = 0.81$ and $x_0 = 1$

$$\phi(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - c}{2x} = x - \frac{x}{2} + \frac{c}{2x} = \frac{1}{2}\left(x + \frac{c}{x}\right)$$

Iterations

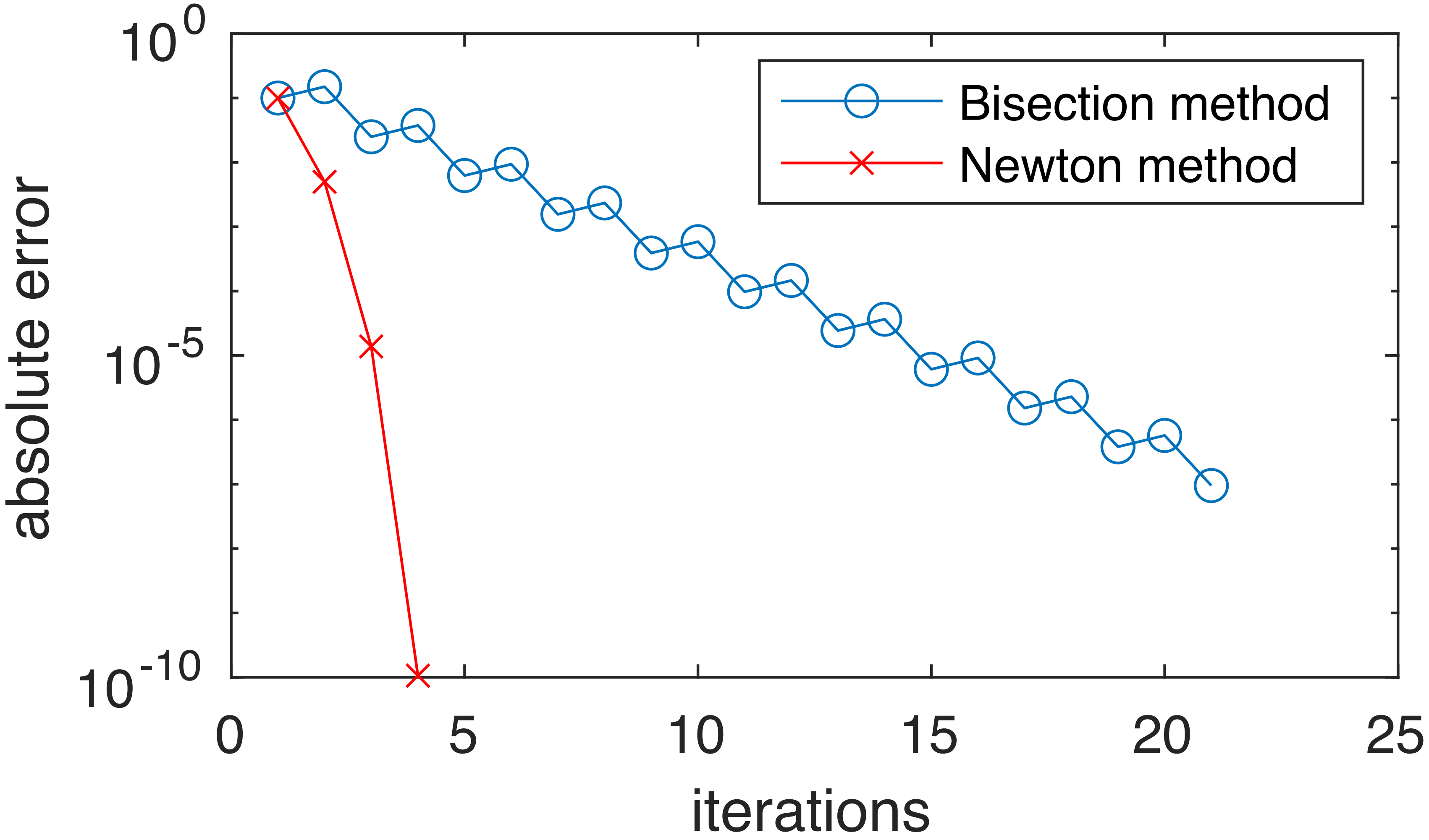
$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{c}{x_k}\right)$$

```
format longE
c = 0.81;
xStar = sqrt(c);
x = 1;
res = [x, xStar];
for i=1:4
    x = 0.5*(x + c/x);
    res(end + 1, :) = [x, xStar];
end
res
```

```
format longE
c = 0.81;
xStar = sqrt(c);
x = 1;
res = [x, xStar];
for i=1:4
    x = 0.5*(x + c/x);
    res(end + 1, :) = [x, xStar];
end
res
```

```
res =
     1.000000000000000e+00     9.000000000000000e-01
     9.050000000000000e-01     9.000000000000000e-01
     9.000138121546961e-01     9.000000000000000e-01
     9.000000001059849e-01     9.000000000000000e-01
     9.000000000000000e-01     9.000000000000000e-01
```

$\rightsquigarrow$ very quick convergence; certainly faster than linear convergence


Bisection method
Newton method
absolute error
iterations
10^0
10^-5
10^-10
0
5
10
15
20
25

Newton's method

Let $F : \mathbb{R}^n \to \mathbb{R}^n$, $n \geq 1$ and solve

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

Truncated Taylor expansion of F about starting point $\boldsymbol{x}^0$:

$$F(\boldsymbol{x}) \approx F(\boldsymbol{x}^0) + F'(\boldsymbol{x}^0)(\boldsymbol{x} - \boldsymbol{x}^0).$$

Hence:

$$\boldsymbol{x}^1 = \boldsymbol{x}^0 - F'(\boldsymbol{x}^0)^{-1} F(\boldsymbol{x}^0)$$

Newton iteration: Start with $\boldsymbol{x}^0 \in \mathbb{R}^n$, and for $k = 0, 1, \ldots$ compute

$$F'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k = -F(\boldsymbol{x}^k), \quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k$$

Requires that $F'(\boldsymbol{x}^k) \in \mathbb{R}^{n\times n}$ is invertible.

Newton's method

Newton iteration: Start with $\boldsymbol{x}^0 \in \mathbb{R}^n$, and for $k = 0, 1, \ldots$ compute

$$F'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k = -F(\boldsymbol{x}^k), \quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k$$

Equivalently:

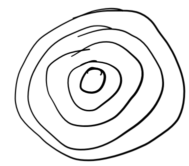
$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - F'(\boldsymbol{x}^k)^{-1}F(\boldsymbol{x}^k)$$

Newton's method is affine invariant, that is, the sequence is invariant to affine transformations $\rightsquigarrow$ board

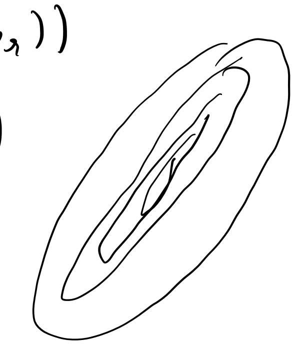
Solve $F(x)=0$ equivalent to solving

$$AF(x)=0 \quad , \quad A \in \mathbb{R}^{n\times n}, \text{ regular}$$

$$G(x) = AF(x)$$



$$\begin{aligned}
x_{k+1} &= x_k - G'(x_k)^{-1} G(x_k) \\
&= x_k - (AF'(x_k))^{-1} (AF(x_k)) \\
&= x_k - F'(x_k)^{-1} A^{-1} A F(x_k) \\
&= x_k - F'(x_k)^{-1} F(x_k)
\end{aligned}$$



Convergence of Newton's method

Assumptions on F: $D \subset \mathbb{R}^n$ open and convex, $F : D \to \mathbb{R}^n$ continuously differentiable with $F'(\boldsymbol{x})$ invertible for all $\boldsymbol{x}$, and there exists $\omega \geq 0$ such that

$$\|F'(\boldsymbol{x})^{-1}(F'(\boldsymbol{x} + s\boldsymbol{v}) - F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

for all $s \in [0, 1]$, $\boldsymbol{x} \in D, \boldsymbol{v} \in \mathbb{R}^n$ with $\boldsymbol{x} + \boldsymbol{v} \in D$.

Assumptions on $\boldsymbol{x}^*$ and $\boldsymbol{x}^0$: There exists a solution $\boldsymbol{x}^* \in D$ and a starting point $\boldsymbol{x}^0 \in D$ such that

$$\rho := \|\boldsymbol{x}^* - \boldsymbol{x}^0\| \leq \frac{2}{\omega} \text{ and } B_\rho(\boldsymbol{x}^*) \subset D$$

where

$$B_\rho(\boldsymbol{x}^*) = \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{y} - \boldsymbol{x}^*\| < \rho\}$$

Q: Meaning of ω?

Theorem: Under the assumptions of the previous slide, the Newton sequence $\boldsymbol{x}^k$ stays in $B_\rho(\boldsymbol{x}^*)$ and $\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x}^*$, and

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\| \leq \frac{\omega}{2}\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2$$

Moreover, the solution x^* is unique in $B_{2/\omega}(x^*)$.

Proof: $\rightsquigarrow$ board

- $\| F'(x)^{-1} (F'(x + sv) - F'(x)) v \|_2 \leq s\omega \|v\|_2^2$

$\forall s \in [0, 1]$, $x + v \in D$

- $\int_0^1 [F'(x + s(y - x)) - F'(x)](y - x)\, ds$ $\quad y \in D$

$$= F(y) - F(x) - F'(x)(y - x)$$

$$\| F'(x)^{-1} [F(y) - F(x) - F'(x)(y-x)] \| =$$

$$= \| \int_0^1 F'(x)^{-1} [F'(x + s(y-x)) - F'(x)](y-x)\, ds \| \qquad (*)$$

$$\leq \int_0^1 \| \dots \| \, ds$$

$$\leq \int_0^1 s\omega \|y - x\|^2 ds = \omega \frac{s^2}{2} \|y - x\|^2 \Big|_0^1 = \frac{\omega}{2} \|y - x\|^2$$

Convergence:

$$x^{q+1} - x^* = x^q - F'(x^q)^{-1} F(x^q) - x^*$$

$$= x^q - x^* - F'(x^q)^{-1} (F(x^q) - \underline{F(x^*)})$$

$$= F'(x^q)^{-1} [-F(x^q) + F(x^*) - F'(x^q)(x^* - x^q)]$$

Use $(*)$

$$\|x^{q+1} - x^*\| \leq \frac{\omega}{2} \|x^* - x^q\|^2$$

Now show that $\{x_q\}$ remains in $B_\rho(x^*)$

$$0 < \|x^q - x^*\| \leq \|x^* - x^0\| =: \rho$$

$$\|x^{q+1} - x^*\| \leq \underbrace{\frac{\omega}{2}\|x^q - x^*\|}_{\leq \rho \frac{\omega}{2}} \|x^q - x^*\|$$

Assumptions

$$\rho = \|x^* - x^0\| < \frac{2}{\omega}$$

$$\Rightarrow \rho \frac{\omega}{2} < 1$$

$$\|x^* - x^{q+1}\| < \|x^q - x^*\| \leq \rho$$

$\{x_q\}$ remains in $B_\rho(x^*) \subseteq D$

Solution x^* is unique in $B_{2/\omega}(x^*)$

Let $x^{**} \in B_{2/\omega}(x^*)$: $F(x^{**}) = 0$

$\|x^* - x^{**}\| < \frac{2}{\omega}$

$$\|x^{**} - x^*\| = \| 0 - 0 - F'(x^*)^{-1} F'(x^*)(x^{**} - x^*)\|$$

$$= \| F'(x^*)^{-1}\big(F(x^{**}) - F(x^*) - F'(x^*)(x^{**} - x^*)\big)\|$$

use(*)

$$\le \underbrace{\frac{\omega}{2}\|x^{**} - x^*\|}_{<1} \|x^{**} - x^*\|$$

because x^{**} in $B_{2/\omega}(x^*)$

$\|x^{**} - x^*\| = 0$

Theorem: Under the assumptions of the previous slide, the Newton sequence $\boldsymbol{x}^k$ stays in $B_\rho(\boldsymbol{x}^*)$ and $\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x}^*$, and

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\| \leq \frac{\omega}{2}\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2$$

Moreover, the solution x^* is unique in $B_{2/\omega}(x^*)$.

Proof: $\rightsquigarrow$ board

Summary: The Newton method converges locally and quadratically.

Role of initialization

Choice of initialization $\boldsymbol{x}^0$ is critical. Depending on the initialization, the Newton iteration might

- not converge (it could "blow up" or "oscillate" between two points)
- converge to different solutions
- fail cause it hits a point where the Jacobian is not invertible (this cannot happen if the conditions of the convergence theorem are satisfied)
- ...

Sometimes, continuation ideas must be used to find good initializations: Solve simpler problems first and use solution as starting point for harder problems.

General comments

The "more nonlinear" a problem, the harder it is to solve.

$$\|F'(\boldsymbol{x})^{-1}(F'(\boldsymbol{x}+s\boldsymbol{v})-F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

Very nonlinear $\rightsquigarrow$ $F'(x)$ changes a lot $\rightsquigarrow$ ω large (need x_0 closers to x^* required)

General comments

The “more nonlinear” a problem, the harder it is to solve.

$$\|F'(\boldsymbol{x})^{-1}(F'(\boldsymbol{x}+s\boldsymbol{v}) - F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

Very nonlinear $\rightsquigarrow$ $F'(x)$ changes a lot $\rightsquigarrow$ ω large (need x_0 closers to x^* required)

Computation of Jacobian can be costly/complicated
$\rightsquigarrow$ sometimes approximate $F'(x^k)$

General comments

The "more nonlinear" a problem, the harder it is to solve.

$$\|F'(\boldsymbol{x})^{-1}(F'(\boldsymbol{x}+s\boldsymbol{v})-F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

Very nonlinear $\rightsquigarrow$ $F'(x)$ changes a lot $\rightsquigarrow$ ω large (need x_0 closers to x^* required)

Computation of Jacobian can be costly/complicated
$\rightsquigarrow$ sometimes approximate $F'(x^k)$

There's no reliable black-box solver for nonlinear problems; at least for higher-dimensional problems, the structure of the problem must be taken into account.

General comments

The "more nonlinear" a problem, the harder it is to solve.

$$\|F'(\boldsymbol{x})^{-1}(F'(\boldsymbol{x}+s\boldsymbol{v}) - F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

Very nonlinear $\rightsquigarrow$ $F'(x)$ changes a lot $\rightsquigarrow$ ω large (need x_0 closers to x^* required)

Computation of Jacobian can be costly/complicated
$\rightsquigarrow$ sometimes approximate $F'(x^k)$

There's no reliable black-box solver for nonlinear problems; at least for higher-dimensional problems, the structure of the problem must be taken into account.

"Classification of mathematical problems as linear and nonlinear is like classification of the Universe as bananas and non-bananas."

Overview

Nonlinear least squares—Gauss-Newton

Nonlinear least-squares problems

Assume a least squares problem, where the parameters $\boldsymbol{x}$ do *not* enter linearly into the model.

Instead of

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|^2,$$

we have with $F : D \to \mathbb{R}^n$, $D \subset \mathbb{R}^n$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} g(\boldsymbol{x}) := \frac{1}{2}\|F(\boldsymbol{x})\|^2, \quad \text{where } F(\boldsymbol{x})_i = \varphi(t_i, \boldsymbol{x}) - b_i, 1 \leq i \leq m$$

The (local) minimum $\boldsymbol{x}^*$ of this optimization problem satisfies:

$$g'(\boldsymbol{x}) = 0, \quad g''(\boldsymbol{x}) \text{ is positive definite.}$$

Nonlinear least-squares problems

The derivative of $g(\cdot)$ is

$$G(\boldsymbol{x}) := g'(\boldsymbol{x}) = F'(\boldsymbol{x})F(\boldsymbol{x})$$

Setting $G(\boldsymbol{x}) = 0$ gives a nonlinear system in $\boldsymbol{x}$, $G : D \to \mathbb{R}^n$.

Let's try to solve it $G(\boldsymbol{x}) = 0$ using Newton's method:

$$G'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k = -G(\boldsymbol{x}^k), \quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k$$

where

$$G'(\boldsymbol{x}) = F'(\boldsymbol{x})^T F'(\boldsymbol{x}) + F''(\boldsymbol{x})^T F(\boldsymbol{x}) \qquad \text{Hessian of } g \text{ (objective)}$$

$\rightsquigarrow$ second-order information of F enters through $F''(X)^T$

Nonlinear least-squares problems

If the data is compatible with the model, which means that the model can perfectly fit the data with zero training error, then $F(\boldsymbol{x}^*) = 0$

Then, term involving $F''(\boldsymbol{x}^*)$ drops out in $G'(\boldsymbol{x}^*)$ anyway as we move towards $\boldsymbol{x}^*$.

If $\|F(\boldsymbol{x}^*)\|$ is small, and thus data *almost* compatible with model, then neglecting that term might not make the convergence much slower.

Also, it's expensive to compute $F''(\boldsymbol{x})$

Nonlinear least-squares problems—Gauss-Newton

The resulting Newton method for the nonlinear least squares problem is called Gauss-Newton method: Initialize $\boldsymbol{x}^0$ and for $k = 0, 1, \ldots$ solve

$$F'(\boldsymbol{x}^k)^T F'(\boldsymbol{x}^k) \Delta \boldsymbol{x}^k = -F'(\boldsymbol{x}^k)^T F(\boldsymbol{x}^k) \quad \text{(solve)} \tag{1}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k. \quad \text{(update step)}$$

Nonlinear least-squares problems—Gauss-Newton

The resulting Newton method for the nonlinear least squares problem is called Gauss-Newton method: Initialize $\boldsymbol{x}^0$ and for $k = 0, 1, \ldots$ solve

$$F'(\boldsymbol{x}^k)^T F'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k = -F'(\boldsymbol{x}^k)^T F(\boldsymbol{x}^k) \quad \text{(solve)} \tag{1}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k. \quad \text{(update step)}$$

The solve step is the normal equation for the linear least squares problem

$$\min_{\Delta\boldsymbol{x}} \|F'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k + F(\boldsymbol{x}^k)\|. \tag{2}$$

so we better solve (2) rather than directly (1)

Convergence of Gauss-Newton method

Assumptions on F: $D \subset \mathbb{R}^n$ open and convex, $F : D \to \mathbb{R}^m$, $m \geq n$ continuously differentiable with $F'(\boldsymbol{x})$ has full rank for all $\boldsymbol{x}$, and let $\omega \geq 0, 0 \leq \kappa^* < 1$ such that

$$\|F'(\boldsymbol{x})^+(F'(\boldsymbol{x} + s\boldsymbol{v}) - F'(\boldsymbol{x}))\boldsymbol{v}\| \leq s\omega\|\boldsymbol{v}\|^2$$

for all $s \in [0,1]$, $\boldsymbol{x} \in D, \boldsymbol{v} \in \mathbb{R}^n$ with $\boldsymbol{x} + \boldsymbol{v} \in D$.

Assumptions on $\boldsymbol{x}^*$ and $\boldsymbol{x}^0$: Assume there exists a solution $\boldsymbol{x}^* \in D$ of the least squares problem and a starting point $\boldsymbol{x}^0 \in D$ such that

$$\|F'(\boldsymbol{x})^+ F(\boldsymbol{x}^*)\| \leq \kappa^* \|\boldsymbol{x} - \boldsymbol{x}^*\|$$

$$\rho := \|\boldsymbol{x}^* - \boldsymbol{x}^0\| \leq \frac{2(1-\kappa^*)}{\omega} := \sigma$$

Theorem: Then, the sequence $\boldsymbol{x}^k$ stays in $B_\rho(\boldsymbol{x}^*)$ and $\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x}^*$, and

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\| \leq \frac{\omega}{2}\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2 + \underbrace{\kappa^*\|\boldsymbol{x}^k - \boldsymbol{x}^*\|}_{\rightsquigarrow \text{ linear convergence if } \kappa^* > 0!}$$

$\rightsquigarrow$ we usually want to choose models that are "almost compatible" which means κ^* is often very small

Conclusions

- Solving nonlinear systems of equations ("root finding") is iterative in nature in general
- The order of convergence matters; quadratic is good enough but mind costs per step
- Newton's method is second order but requires derivatives/Jacobian evaluations.
- In higher dimensions, a good initial guess is critical for Newton's method
- There are many variants of Newton's method (e.g., Quasi-Newton methods) that avoid the computational costs of computing the Jacobian
- (Machine learning is using first-order methods only anyway...)