SI152: Numerical Optimization

Lecture 1: Equations

Hao Wang
Email: haw309@gmail.com

ShanghaiTech University

September 18, 2025

Outline

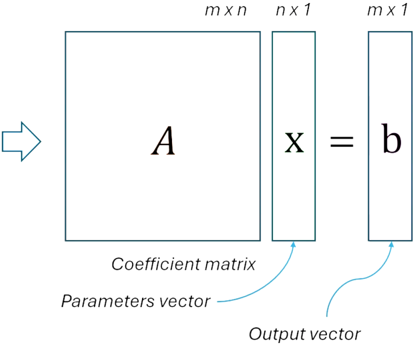
Linear equations are fundamental in many fields:

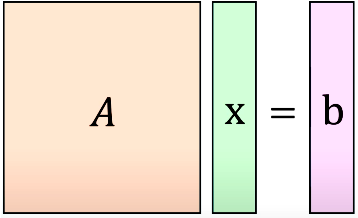
$$Ax = b, \quad A \in \mathbb{R}^{m \times n}$$

Equations: m

Parameters: n

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$





- Direct methods: Gaussian elimination
  Complexity $O(n^3)$: can be inefficient for large systems.
- Iterative methods
- Conjugate gradient method (will see this method later)

Many real-world applications that require solving systems of linear equations with a large number of variables, sometimes even on the scale of millions.

- Computer Graphics and Image Processing
- Machine Learning,
- Scientific Simulations
- Economics and Finance
- ....

They appear in two ways:

- Reconstruct the original signal via a linear observation: $b = Ax$
- Linear equations are the subproblems of many other algorithms: e.g., Newton method for solving nonlinear equations.

An iterative technique to solve the $n \times n$ linear system $Ax = b$

- Start with an initial guess $x^{(0)}$.
- Generate a sequence of approximations $\{x^{(k)}\}$ using an iteration formula

$$x^{(k+1)} = \mathscr{M}(x^{(k)})$$

- Stop when a convergence criterion is met $\|Ax^{(k)} - b\| < \text{tol}$
- Different algorithm has different Iteration Formula $\mathscr{M}$

Why Iterative Methods?

- Very efficient for large, sparse systems.
- Can be parallelized easily.
- Provide approximations with controlled accuracy.

- Splits the matrix into diagonal and off-diagonal parts.
- Simple iteration formula.
- Convergence depends on the spectral radius of the iteration matrix.

$$A = D + L + U$$

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_{nn} \end{bmatrix},$$

$$L = \begin{bmatrix} 0 & \dots & \dots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & a_{n,n-1} & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ \dots & \ddots & \ddots & \vdots \\ \dots & & \ddots & a_{n-1,n} \\ 0 & \dots & \dots & 0 \end{bmatrix}$$

The equation $Ax = b$, or $(D + L + U)x = b$, is then transformed into

$$Dx = b - (L + U)x$$

and, if D^{-1} exists, that is, if $a_{ii} \neq 0$ for each i, then

$$x = D^{-1}b - D^{-1}(L + U)x$$

This results in the matrix form of the Jacobi iterative technique (let $c = D^{-1}b, M = -D^{-1}(L + U)$):

$$x^{(k+1)} = D^{-1}b - D^{-1}(L + U)x^{(k)} = Mx^{(k)} + c$$

Elementwisely:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

Fixed point of an operator: if D is invertible,

$$Ax^* = b \iff x^* = D^{-1}b - D^{-1}(L+U)x^* = Mx^* + c$$

Fixed-point iteration:

$$x^{(k+1)} = Mx^{(k)} + c$$

Residual

$$x^{(k+1)} - x^* = M(x^{(k)} - x^*) \implies \|x^{(k+1)} - x^*\| \leq \|M\|\|x^{(k)} - x^*\|$$

$\|M\| < 1 \implies$ linear convergence.
Sufficient condition:

- $\rho(D^{-1}(L+U)) < 1$

Weighted Jacobi method:

$$\begin{aligned} x^{(k+1)} &= (1-\omega)x^{(k)} + \omega D^{-1}(b - Lx^{(k)} - Ux^{(k)}) \\ &= x^{(k)} + \omega[D^{-1}(b - Lx^{(k)} - Ux^{(k)}) - x^{(k)}] \end{aligned}$$

Fixed point:

$$[D - D + \omega(D + L + U)]x = \omega b \iff (D + \omega L)x = \omega b - [\omega U + (\omega - 1)D]x$$

Fixed-point iteration:

$$x^{(k+1)} = (D + \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]x^{(k)})$$

Weighted Jacobi method:

$$(D + \omega L)x^{(k+1)} = \omega b - [\omega U + (\omega - 1)D]x^{(k)}$$

Lower-triangle equations:

$$\begin{aligned} a_{11}x_1 &= b_1, \\ a_{21}x_1 + a_{22}x_2 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \implies \begin{aligned} x_1 &= \frac{b_1}{a_{11}} \\ x_2 &= \frac{b_2 - a_{21}x_1}{a_{22}} \\ x_3 &= \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}. \end{aligned}$$

- Similar to Jacobi but uses the most recent approximations: $A = L + U$

$$L = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- $(L + U)x = b \iff Lx = b - Ux$
- $x^{(k+1)} = L^{-1}(b - Ux^{(k)})$,
- Often converges faster than Jacobi.
- Still depends on the spectral radius but with improved convergence properties.

Outline

1. Linear equations
   - Iterative methods for linear equations

2. Univariate Nonlinear Equations
   - Bisection method
   - Newton's method
   - Secant Method

3. Newton's method for Multivariable cases

4. Equations and Optimization

We are familiar with equation

$$f(x) = 0$$

for a given function f.
Finding the root of this equation may not be trivial:

$$f(x) = 2x^2 - e^x + \sin x$$

Analysis

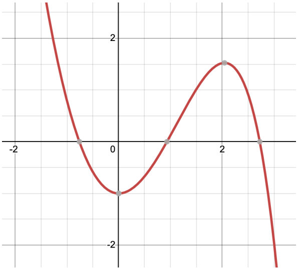
python

```python
import numpy as np
from scipy.optimize import fsolve

# Define the function f(x) = 2x^2 - e^x + sin(x)
def f(x):
    return 2*x**2 - np.exp(x) + np.sin(x)

# Use fsolve to find a root, starting with an initial guess
initial_guess = 0.5
root = fsolve(f, initial_guess)

root[0]  # Return the first root found
```

Always show details | Copy code

Result

0.9317317338916016



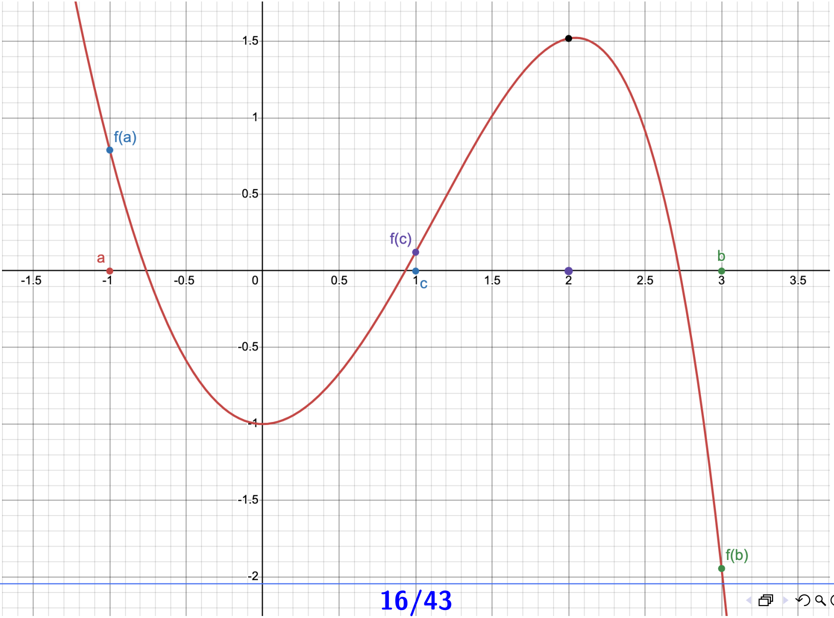
Bisection method

Initialization: a, b such that $f(a) \cdot f(b) < 0$.
Let $c = \frac{a+b}{2}$
If $f(c) = 0$ or $|a - b| < \epsilon$, stop! c is the root.
if $f(a) \cdot f(c) < 0$: **then**
    set $b \leftarrow c$
else
    set $a \leftarrow c$
end if

Theorem 1 (Intermediate Value Theorem)

If a function $f(x)$ is continuous on an interval $[a, b]$ and $f(a) \cdot f(b) < 0$, then a value $c \in (a, b)$ exists for which $f(c) = 0$.

$$f(x) = 2x^2 - e^x + \sin x = 0$$



Termination

Given tolerance $\epsilon > 0$, terminate the algorithm if

$$|a-b| \leq \epsilon.$$

- This implies $|c - c^*| < \epsilon$, where c^* is a root ($f(c^*) = 0$).
- Terminate with an approximate root.
- How many iterations at most (worst case) needed to reduce initial $|a-b|$ to ϵ?

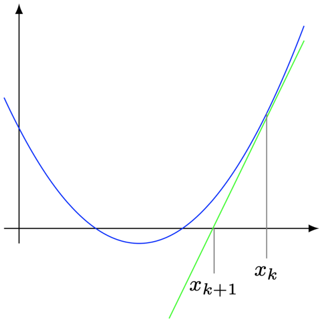
$$\frac{|a-b|}{2^k} \leq \epsilon \implies \log \frac{|a-b|}{\epsilon} \leq k \log 2 \implies k \geq \frac{\log \frac{|a-b|}{\epsilon}}{\log 2}.$$

We can motivative Newton's method in 3 ways (that are basically all the same).

- At the current point $(x_k, f(x_k))$, draw a tangent line until it hits the x-axis; call that point x_{k+1}.
- Create an affine model of f at x_k:

  $$m_k(x) = f(x_k) + f'(x_k)(x - x_k);$$

  call x_{k+1} the solution to $m_k(x) = 0$.



- Write the Taylor series of F at x_k:

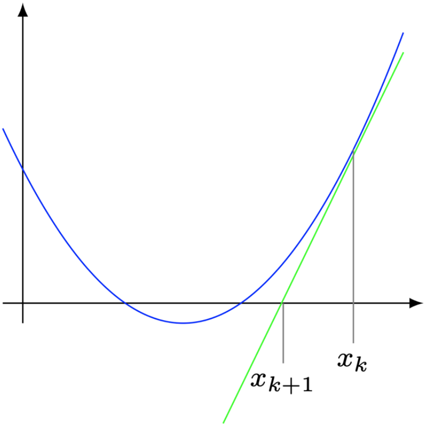
  $$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2 + \ldots;$$

  approximate $f(x)$ with the affine portion, solve the resulting affine equation, and call the solution x_{k+1}.

Newton's method

Solving the "affine approximation" yields the formula:

$$\boxed{x_{k+1} \leftarrow x_k - \frac{f(x_k)}{f'(x_k)}}$$



Quadratic convergence (Local)

Letting $e = x - x_*$,

$$f(x) = f(x_*) + f'(x_*)e + \frac{f''(x_*)}{2}e^2 + O(e^3) = f'(x_*)e + \frac{f''(x_*)}{2}e^2 + O(e^3)$$

$$f'(x) = f'(x_*) + f''(x_*)e + O(e^2)$$

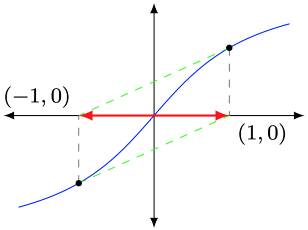
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{f'(x_*)e_k + \frac{f''(x_*)}{2}e_k + O(e_k^3)}{f'(x_*) + f''(x_*)e_k + O(e_k^2)}$$

For small x, we have $(1+x)^{-1} \approx 1 - x$,

$$\begin{aligned} e_{k+1} &\approx e_k - e_k \left(1 + \frac{f''(x_*)}{2f'(x_*)} + O(e_k^2)\right)\left(1 - \frac{f''(x_*)}{f'(x_*)}e_k + O(e_k^2)\right) \\ &\approx e_k - e_k \left(1 - \frac{f''(x_*)}{2f'(x_*)}e_k\right) = \frac{f''(x_*)}{2f'(x_*)}e_k^2 \end{aligned}$$

Newton's method can fail in many ways:

- Certain starting point can lead to cycling and even divergence.
- May have $f'(x_k) = 0$. (So what?)
- $f(x_k)$ may be undefined.



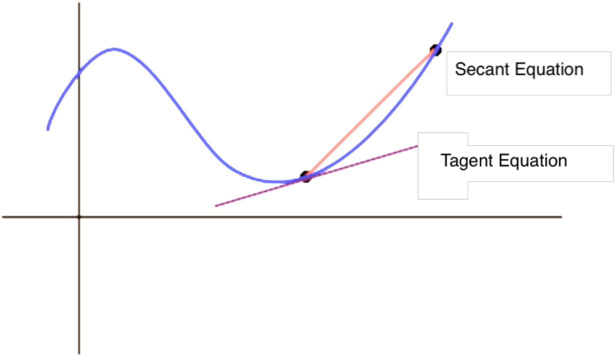
Also, it might not fail, but in some situatioins it can converge very sloooowly...

Still, it is very powerful.

We will investigate it in the multivariable cases.

Issues with Newton's method:

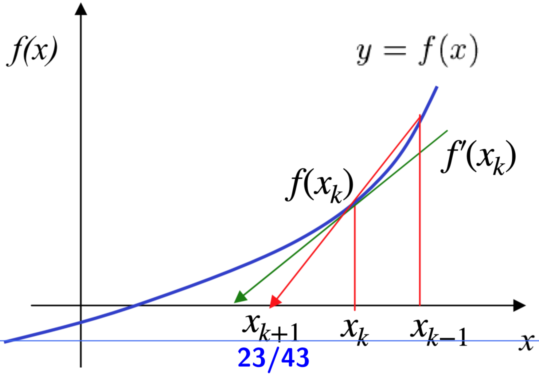
- Newton's method is an extremely powerful technique, but it has a major weakness: the need to know the value of the derivative of f at each approximation.
- Frequently, $f'(x)$ is far more difficult and needs more arithmetic operations to calculate than $f(x)$.



$$f'(x_k) \approx \frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k} = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Substitute into Newton's method, or solving a linear equation:

$$\boxed{x_{k+1} \leftarrow x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)}$$



- Initialization? x_0, x_1
- Only one function evaluation is needed per step. In contrast, each step of Newton's method requires an evaluation of both the function and its derivative.
- Local convergence rate: superline!

$$|x_{n+1} - x_*| \approx C|x_n - x_*|^p, p \in (1, 2)$$

Letting $e_k := x_k - x_*$ and $M = \frac{f''(x_*)}{2f'(x_*)}$. We know
$f(x_* + e) \approx f'(x_*)e + \frac{f''(x_*)}{2}e^2$

$$\begin{aligned}
\implies f(x_* + e_k) &\approx e_k f'(x_*)(1 + Me_k) \\
f(x_* + e_k) - f(x_* + e_{k-1}) &\approx f'(x_*)(e_k - e_{k-1})(1 + M(e_k + e_{k-1})) \\
e_{k+1} &\approx e_k - \frac{e_k(1 + Me_k)}{1 + M(e_k + e_{k-1})} \\
&= \frac{e_{k-1}e_k M}{1 + M(e_k + e_{k-1})} \approx Me_{k-1}e_k.
\end{aligned}$$

If $|e_{k+1}| \approx C|e_k|^p$, then

$$C|e_k|^p \approx |M||e_{k-1}||e_k|$$

$$|e_k| \approx \left(\frac{|M|}{C}\right)^{\frac{1}{p-1}} |e_{k-1}|^{\frac{1}{p-1}}.$$

So we have $p = 1/(p-1)$. Solving $p^2 - p - 1 = 0$, we end up with

$$p = \frac{1+\sqrt{5}}{2} \approx 1.618.$$

We also have that

$$C = \left(\frac{|M|}{C}\right)^{\frac{1}{p-1}} \implies C = |M|^{1/p} = \left|\frac{f''(x_*)}{2f'(x_*)}\right|^{p-1}.$$

$$|x_{k+1} - x_*| \approx \left|\frac{f''(x_*)}{2f'(x_*)}\right|^{\frac{\sqrt{5}-1}{2}} |x_k - x_*|^{\frac{\sqrt{5}+1}{2}}.$$

$$e_{k+1} \approx \frac{f''(x_*)}{2f'(x_*)} e_{k-1} e_k \approx \left| \frac{f''(x_*)}{2f'(x_*)} \right|^{\frac{\sqrt{5}-1}{2}} e_k^{\frac{\sqrt{5}+1}{2}}.$$

$$e_{k+1} \approx \frac{f''(x_*)}{2f'(x_*)} e_k^2.$$

Example: $f(x) = \cos x - x$. $x_0 = 0.5$ for Newton's method and $p_1 = \pi/4$ for Secant method.

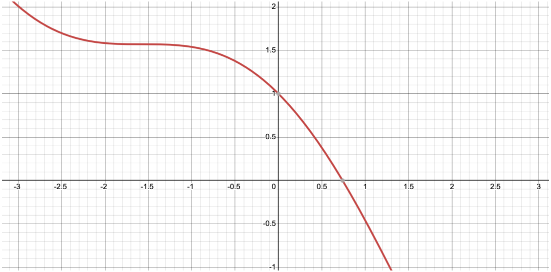


Table: Test results for Newton's method

k	x_{k-1}	$f(x_{k-1})$	$f'(x_{k-1})$	x_k	$\lvert x_k - x_{k-1}\rvert$
1	0.78539816	−0.078291	−1.707107	0.73953613	0.04586203
2	0.73953613	−0.000755	−1.673945	0.73908518	0.00045096
3	0.73908518	−0.000000	−1.673612	0.73908513	0.00000004
4	0.73908513	−0.000000	−1.673612	0.73908513	0.00000000

Table: Test results for Secant method

k	x_{k-2}	x_{k-1}	x_k	$\lvert x_k - x_{k-1}\rvert$
2	0.500000000	0.785398163	0.736384139	0.0490140246
3	0.785398163	0.736384139	0.739058139	0.0026740004
4	0.736384139	0.739058139	0.739085149	0.0000270101
5	0.739058139	0.739085149	0.739085133	0.0000000161

Outline

Suppose we have $F:\mathbb{R}^m \to \mathbb{R}^n$ and solve $F(x)=0$.

- Form an affine model of the function at x_k:

$$F(x) \approx F(x_k) + \nabla F(x_k)^T (x - x_k).$$

- Solve for the step/displacement d_k (i.e., let $d = x - x_k$):

$$\boxed{\nabla F(x_k)^T d = -F(x_k).}$$

- Update the iterate

$$x_{k+1} = x_k + d_k.$$

Example

Suppose we aim to solve

$$F(x) = \begin{bmatrix} x_1^2 + x_1 x_2 \\ e^{x_1} - x_2 \end{bmatrix} = 0.$$

(Can we solve this analytically? How many solutions does it have?)

- Pick a starting point, say $x_0 = (1, 1)$.
- Evaluate $F(x)$ and $\nabla F(x)^T$:

$$F(x_0) = \begin{bmatrix} 2 \\ e - 1 \end{bmatrix}, \nabla F(x)^T = \begin{bmatrix} 2x_1 + x_2 & x_1 \\ e^{x_1} & -1 \end{bmatrix}, \nabla F(x_0)^T = \begin{bmatrix} 3 & 1 \\ e & -1 \end{bmatrix}$$

- Solve the Newton system:

$$\begin{bmatrix} 3 & 1 \\ e & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = - \begin{bmatrix} 3 \\ e - 1 \end{bmatrix}.$$

- Update $x_1 = x_0 + d_0$, reevaluate, solve, update, reevaluate, solve...

Example results

```
>> x = newton('example',[1;1]);
================================
  k      ||F(x)||        ||d||
================================
   0  2.6368e+000  6.5211e-001
   1  6.5261e-001  2.9418e-001
   2  8.5037e-002  5.8346e-002
   3  4.1779e-003  4.2638e-003
   4  2.2801e-005  2.6147e-005
   5  8.2119e-010  -----------
================================
>> x

x =

    0.0000
    1.0000
```

```
>> x = newton('example',[-1;1]);
================================
  k      ||F(x)||        ||d||
================================
   0  6.3212e-001  6.5353e-001
   1  4.6100e-002  4.1159e-002
   2  2.4495e-004  2.2103e-004
   3  6.9278e-009  -----------
================================
>> x

x =

   -0.5671
    0.5671
```

Suppose we aim to solve

$$F(x) = \arctan(x) = 0.$$

- Pick a starting point, say $x_0 = 1$.
- Evaluate $F(x)$ and $\nabla F(x)^T$:

$$F(x_0) \approx 0.7854 \text{ and } \nabla F(x)^T = \frac{1}{1+x^2} = \frac{1}{2}.$$

- Solve the Newton system: $\frac{1}{2}d = -0.7854$.
- Update $x_1 = x_0 + d_0$, reevaluate, solve, update, reevaluate, solve ...

Example results

```
>> x = newton('example2',1.391);
===============================
  k      ||F(x)||        ||d||
===============================
   0  9.4749e-001  2.7808e+000
   1  9.4708e-001  2.7763e+000
   2  9.4598e-001  2.7647e+000
   3  9.4308e-001  2.7342e+000
   4  9.3539e-001  2.6555e+000
   5  9.1489e-001  2.4597e+000
   6  8.5946e-001  2.0165e+000
   7  7.0810e-001  1.2272e+000
   8  3.5527e-001  4.0417e-001
   9  3.3148e-002  3.3184e-002
  10  2.4303e-005  2.4303e-005
  11  9.5692e-015  -----------
===============================
>> x

x =

 -9.5692e-015
```

```
>> x = newton('example2',1.392);
===============================
  k      ||F(x)||        ||d||
===============================
   0  9.4783e-001  2.7844e+000
   1  9.4798e-001  2.7859e+000
   2  9.4835e-001  2.7899e+000
   3  9.4934e-001  2.8006e+000
   4  9.5194e-001  2.8288e+000
   5  9.5878e-001  2.9047e+000
   6  9.7661e-001  3.1160e+000
   7  1.0221e+000  3.7576e+000
   8  1.1304e+000  6.2188e+000
   9  1.3314e+000  2.3681e+001
  10  1.5198e+000  5.8438e+002
  11  1.5690e+000  5.0052e+005
  12  1.5708e+000  3.9262e+011
  13  1.5708e+000  2.4214e+023
  14  1.5708e+000  9.2101e+046
  15  1.5708e+000  1.3324e+094
  16  1.5708e+000  2.7888e+188
  17  1.5708e+000  Inf
  18  1.5708e+000  Inf
  19  NaN  -----------
===============================
>> x

x =

   NaN
```

Newton's method converges quadratically! (under nice assumptions) We will go through the proof ($m = n$).

Assumption 2

For some point $x_* \in \mathbb{R}^n$ such that $F(x_*) = 0$, the following hold:

- F *is continuously differentiable in an open convex set* $\mathcal{X} \subset \mathbb{R}^n$ *with* $x_* \in \mathcal{X}$.
- *The Jacobian of* F *at* x_* *is invertible and is bounded in norm by* $M > 0$, *i.e.,*

$$\|(\nabla F(x_*)^T)^{-1}\|_2 \leq M.$$

- *For some neighborhood of* x_* *with radius* $r > 0$ *contained in* $\mathcal{X}$, *i.e.,*

$$\mathbb{B}(x_*, r) := \{x \in \mathbb{R}^n \mid \|x - x_*\|_2 \leq r\} \in \mathcal{X},$$

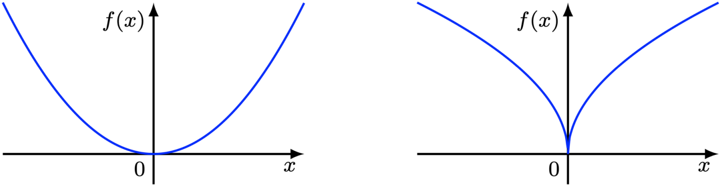
the Jacobian of $F(x)$ *is Lipschitz continuous with constant* L *in* $\mathbb{B}(x_*, r)$.

Definition 3

A function G is Lipschitz continuous with constant $L \geq 0$ in $\mathcal{X}$ if

$$\|G(x_1) - G(x_2)\|_2 \leq L\|x_1 - x_2\|_2 \quad \forall x_1, x_2 \in \mathcal{X}.$$

- What does it mean? There is a limit to how fast G changes.



- Thus, in our proof, we assume there is a limit to how fast the Jacobian of $F(x)$ changes (in norm) in a neighborhood of x_*.

Theorem 4

Quadratic convergence of Newton's method There exists $\epsilon > 0$ such that for all $x_0 \in \mathbb{B}(x_*, \epsilon)$, the sequence defined by

$$\begin{aligned} \nabla F(x_k)^T d_k &= -F(x_k) \\ x_{k+1} &= x_k + d_k, \quad k = 0, 1, 2, \ldots \end{aligned}$$

is well-defined, converges to x_*, and for some $c > 0$ satisfies

$$\|x_{k+1} - x_k\|_2 \leq c\|x_k - x_*\|_2^2.$$

Proof, part 1.

Consider $\bar{\epsilon} > 0$ such that, with $\|x_0 - x_*\|_2 \le \bar{\epsilon}$, the Jacobian $\nabla F(x_0)^T$ is nonsingular. This guarantees that the iteration is well-defined at x_0. Let

$$\epsilon := \min\{\bar{\epsilon}, \tfrac{1}{2ML}\} > 0.$$

Recall that if A is nonsingular and $\|A^{-1}(B-A)\|_2 < 1$, then B is nonsingular and

$$\|B^{-1}\|_2 \le \frac{\|A^{-1}\|_2}{1 - \|A^{-1}(B-A)\|_2}.$$

Thus, since $\nabla F(x_*)^T$ is nonsingular and

$$\begin{aligned} \|\nabla F(x_*)^{-T}(\nabla F(x_0)^T - \nabla F(x_*)^T)\|_2 \le\ & \|\nabla F(x_*)^{-T}\|_2 \|\nabla F(x_0)^T - \nabla F(x_*)^T\|_2 \\ \le\ & ML\|x_0 - x_*\|_2 \le ML\epsilon \le \tfrac{1}{2}, \end{aligned}$$

we know that $\nabla F(x_0)^T$ is nonsingular and

$$\|\nabla F(x_0)^{-T}\|_2 \le \frac{\|\nabla F(x_*)^{-T}\|_2}{1 - \|\nabla F(x_*)^{-T}(\nabla F(x_0)^T - \nabla F(x_*)^T)\|_2} \le 2M.$$

Proof, part 2.

We now show that for some $c > 0$ we have $\|x_1 - x_*\|_2 \leq c\|x_0 - x_*\|_2^2$. (This is the relationship we need for quadratic convergence, which will be nice if we can show that we actually converge!) The difference between x_1 and x_* can be written as

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - \nabla F(x_0)^{-T} F(x_0) \\ &= x_0 - x_* - \nabla F(x_0)^{-T}(F(x_0) - F(x_*)) \\ &= \nabla F(x_0)^{-T}(F(x_*) - \underbrace{F(x_0) - \nabla F(x_0)^T (x_* - x_0)}_{\text{affine model of } F(x) \text{ at } x_0}). \end{aligned}$$

Recalling a result from multivariable calculus, we then find

$$\begin{aligned} \|x_1 - x_*\|_2 &\leq \|\nabla F(x_0)^{-T}\|_2 \|F(x_*) - F(x_0) - \nabla F(x_0)^T (x_* - x_0)\|_2 \\ &\leq (2M)(\tfrac{1}{2} L \|x_0 - x_*\|_2^2) \\ &\leq ML\|x_0 - x_*\|_2^2. \end{aligned}$$

Thus, the result holds for $c = ML$.

Proof, part 3.

Finally, we show that $\|x_1 - x_*\|_2 \leq \frac{1}{2}\|x_0 - x_*\|_2$. (This shows that $x_1 \in \mathbb{B}(x_*, \epsilon)$, so all of our results so far will continue to hold for $k = 1, 2, \ldots,$ meaning that we converge and do so quadratically!) We have shown already that

$$\|x_1 - x_*\|_2 \leq ML\|x_0 - x_*\|_2^2,$$

and since $\|x_0 - x_*\|_2 \leq \epsilon \leq (2ML)^{-1}$ (by definition of ϵ), we have

$$\|x_1 - x_*\|_2 \leq \frac{1}{2}\|x_0 - x_*\|_2.$$

- Naturally, we need to be close enough to x_* so that the function will be differentiable at all iterates; otherwise, the algorithm won't be well-defined.
- The other two constants, L and M, play crucial roles in the proof, but also have great significance in terms of
- If L is large, then the gradients of the functions change rapidly.
- If M is large, then following the gradient may send us far away.

Equations $\Longrightarrow$ Optimization

- $F(x) = 0 \implies \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2}\|F(x)\|_2^2$

Optimization $\Longrightarrow$ Equations

- For any C^1-smooth,

$$\min_{x \in \mathbb{R}^n} f(x)$$

- What is the equivalent system of equations we need to solve?
- We will learn in the future: $\nabla f(x) = 0$.