TDT4127 Programming and Numerics Week 39

Functions

Equation solving

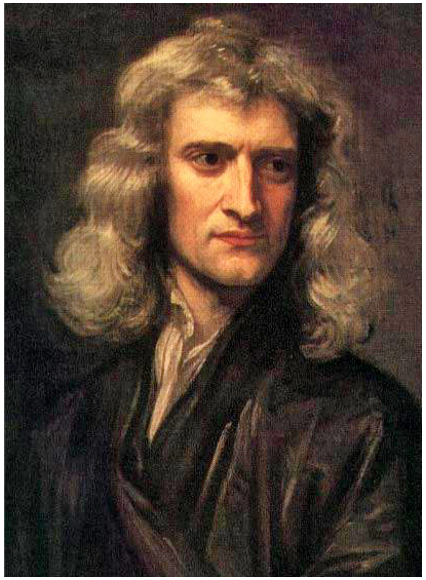
Convergence of Newton’s method and secant method

Important note

- Next week: **auditorium exercise 1**
  - **Bring your own computer**, or borrow one from NTNU!
  - **No** computer lab hours next week
  - Instead, do some exercises live in an **auditorium** (2 hours)
  - Days: *Tuesday-Friday*.
    - Due to space constraints, you are given a **fixed time/place**
    - Check this link to see where/when you should show up:
    https://www.ntnu.no/wiki/pages/viewpage.action?pageId=122881206
    - If you cannot make it, there will be an extra session the week after
  - Install **Safe Exam Browser** (link above)
- One of the two auditorium exercises **must be approved** in order to take the exam
  - Also, we use **Inspera** to get acquainted before the exam

Learning goals

- Goals
  - Solving equations numerically
    - Algorithms:
      - ***Newton's method, secant method***
    - **Convergence analysis**
- Curriculum
  - Exercise 5



NTNU

Functions

- This week in programming: **functions**
  - Some understand them quickly, others take more time
- A connection to mathematical functions
  - A *mathematical function*: A rule that connects inputs to outputs

$$f(x) = 3x^2 + 5x^4$$

  - A *programming function*: A rule that connects inputs to outputs

```
def f(x):
    return 3*x**2 + 5*x**4
```

  - It's almost as if programming is inspired by mathematics 🤔
    - To be fair, programming functions can take different inputs/outputs than we are used to from mathematics, such as **text strings**.

Equation solving

- Newton's method is stated as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \qquad k = 0,1,2,3, \ldots$$

- Typical stopping conditions are

$$|x_{k+1} - x_k| < \delta \text{ or } |f(x_k)| < \epsilon$$

- Last week:
  - Statement of the algorithm
  - Some tips for implementation
- This week, we take a look at the **user manual**
  - How *quickly* does it provide a root?
  - What *conditions* do we need for the method to work?
  - For both Newton's method and the secant method
    - » And the bisection method

Convergence order

- If the algorithm is working as intended, then the x_k should approach the solution z with $f(z) = 0$.
  - That is, the *error* $E_k = |x_k - z|$ should be *decreasing*
- Can we say anything about how *fast* it is decreasing?
  - Typical strategy: compare iteration k to iteration $k-1$
- A root-finder is said to ***converge*** to z with ***order*** q if

$$|x_k - z| < C|x_{k-1} - z|^q$$

for some (unspecified) constant C

Convergence order

- A root-finder is said to ***converge*** to z with **order** q if
$$|x_k - z| < C|x_{k-1} - z|^q$$
  - Example (Newton): $q = 2$ If $C = 1, |x_0 - z| = 0.1$:
    - $|x_1 - z| < 0.01$
    - $|x_2 - z| < 0.0001$
    - $|x_3 - z| < 0.00000001$
  - Example (Secant): $q = 1.618$ If $C = 1, |x_0 - z| = 0.1$:
    - $|x_1 - z| < 0.02$
    - $|x_2 - z| < 0.0006$
    - $|x_3 - z| < 0.00000033$
  - Example (Bisection): $q = 1, C = 1/2.$ If $|x_0 - z| = 0.1$:
    - $|x_1 - z| < 0.05$
    - $|x_2 - z| < 0.025$
    - $|x_3 - z| < 0.01$

Convergence of Newton's method

- Newton's method will converge quadratically (with $q = 2$) to a root z, $(f(z) = 0)$ if all of the following hold:
  - f is continuous, f' is continuous *and* f'' is continuous in an interval $[a, b]$ around z
  - $f''(z) \neq 0$
  - $\frac{|f''(x)|}{|f'(y)|} \leq A$ for some A, for all x and y in $[a, b]$
    - Meaning $f''(x) < \infty$ and $f'(y) \neq 0$
  - The starting point x_0 is in $[a, b]$ and $|x_0 - z| < 1/A$
- Note: This is a *guarantee*; it could work in other cases, but if these conditions hold, you are sure to find a root

Convergence of Newton's method

- Translated into a more understandable form:

  Newton's method will eventually solve $f(z) = 0$ if:

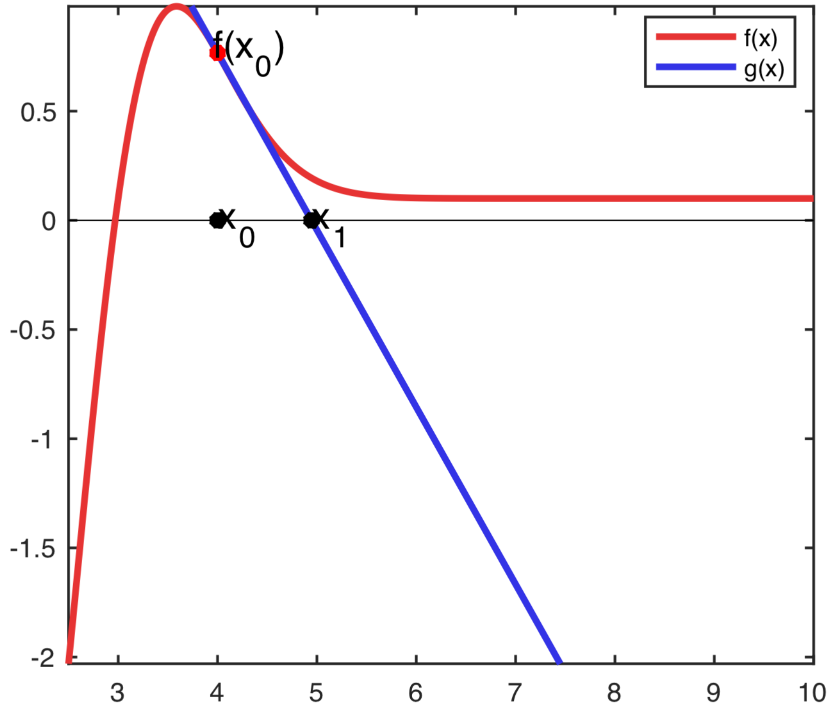
  – The graph of f is sufficiently smooth (no corners!)
  – $f''(z) \neq 0$
  – $f''(x)$ does not blow up
  – $f'(x) \neq 0$ (No minimum or maximum points)
  – The starting point x_0 is sufficiently close to the solution

- However, it can work in other cases, but at a slower rate!

Failure to converge

$$f(x) = x(x^2-1)(x-3)e^{-\frac{(x-1)^2}{2}} + 0.1$$

$$x_1 = x_0 - f(x_0)/f'(x_0)$$



$x_0 = 4.000$, $x_1 = 4.9452$

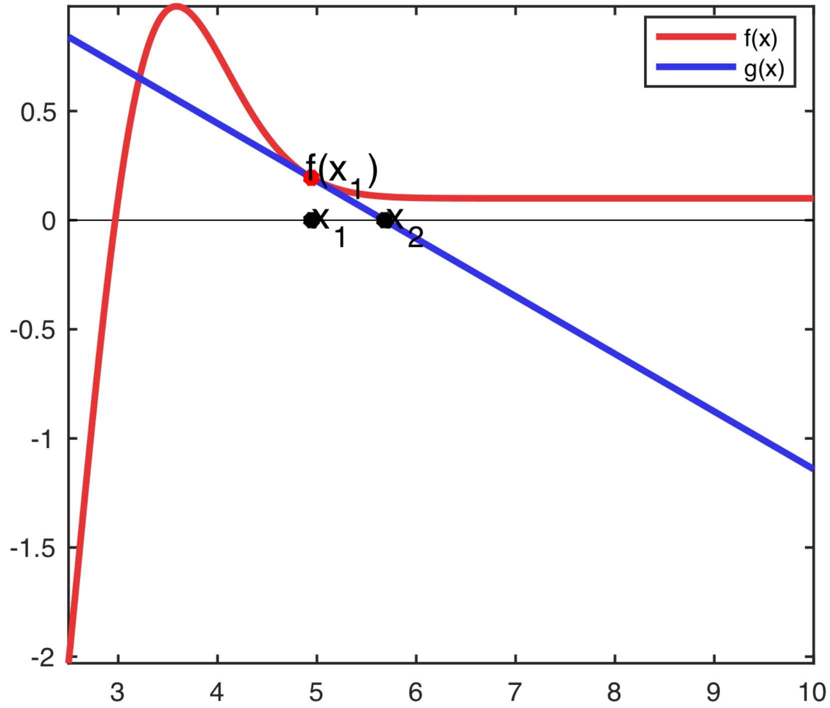
$f(x_0) = 0.7665$, $f(x_1) = 0.1941$

NTNU

Failure to converge

$$f(x) = x(x^2-1)(x-3)e^{-\frac{(x-1)^2}{2}} + 0.1$$

$$x_2 = x_1 - f(x_1)/f'(x_1)$$



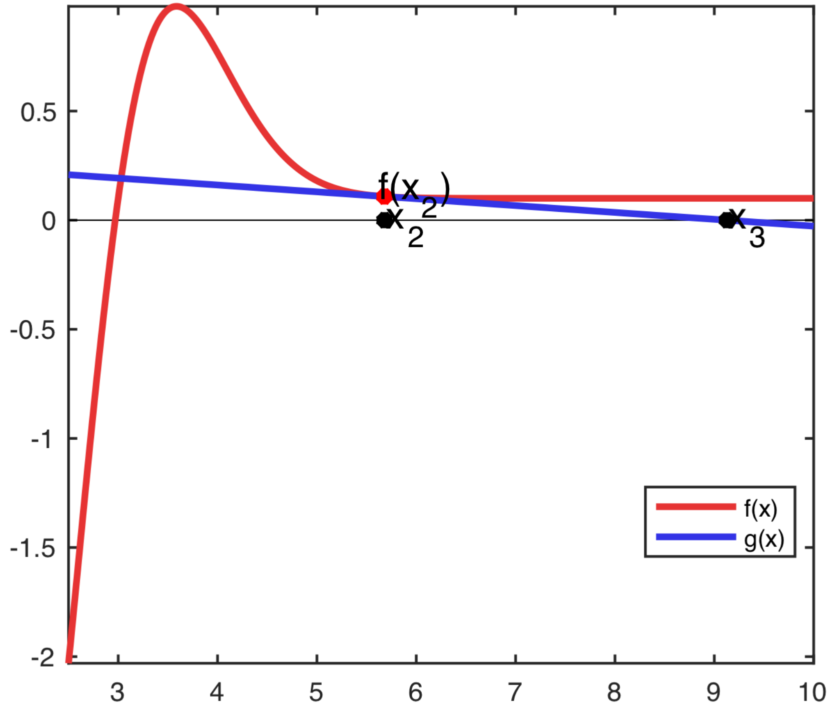
$x_1 = 4.9452, \qquad x_2 = 5.6800$

$f(x_1) = 0.1941, f(x_2) = 0.1083$

Failure to converge

$$f(x) = x(x^2 - 1)(x - 3)e^{-\frac{(x-1)^2}{2}} + 0.1$$

$$x_3 = x_2 - f(x_2)/f'(x_2)$$



$$x_2 = 5.6800, \qquad x_3 = 9.1265$$

$$f(x_2) = 0.1083, \; f(x_3) = 0.1000$$

Convergence of the secant method

- The secant method will converge at least linearly (in fact, with $q = 1.618$) to a root z if all of the following hold:
  - f is continuous and f' is continuous in an interval around z
  - $f'(z) \neq 0$
  - The starting points x_0 and x_1 are sufficiently close to z
- In plain language, the secant method will work if
  - f is smooth enough, less restrictively so than Newton's method
  - The root is not also a maximum/minimum
  - The starting guesses are good enough
- Note: This is also a *guarantee*; it could work in other cases, too.

Comparison of conditions

In terms of restrictive properties, Newton > Secant > Bisection

Property type	Newton's method	Secant method	Bisection method
Continuity	f''	f'	f
Nonzero	$f''(z) \neq 0, f'(x) \neq 0$	$f'(z) \neq 0$	None
Extra bounds	$\frac{\lvert f''(x)\rvert}{\lvert f'(y)\rvert} \leq A$	None	None
Starting point	Close enough	Close enough	Initial interval must enclose z

Note: If f' is continuous, then so is f, so just one continuity requirement is posted here

Combining algorithms

- Newton's method is good when we are *close enough* to a solution
- The bisection method works, but slower, in more situations than Newton's method
- Bright idea: combine the two
  - Some bisection iterations first, then let Newton take the wheel
- This is a smart tactic, and often used in practice
- Doesn't just apply to root finding, many other tasks use this kind of thinking as well.

In practice – tips for use

- Check if your function has continuous derivatives
  - If not, **don't** use Newton's method or the secant method
  - Other parts of the convergence guarantee is harder to verify
- If f is twice continuously differentiable, try Newton
  - And check function values to verify that you get a zero!
- If you don't have derivatives or f is only continuously differentiable
  - Try the secant method
- If you're out of luck with any of the above
  - Try the bisection method
- Same tips apply for built-in solvers such as those found in MATLAB or libraries: **Check the conditions**!

Summary

- We are now **done** with equation solving. We have seen:
  - Newton's method
  - The secant method
  - The bisection method
- We have seen how they work and what is required for them to work
  - Conditions for use
  - Convergence rates, which translate to algorithmic speed
- Next two weeks: linear systems
  - Solving matrix-vector equations

Questions?