7. Root finding II: Newton method

Summary of last lecture

- Iterative algorithms:

$$x_{n+1} = g(x_n)$$

- *If* an iteration converges to x^* then $x^* = g(x^*)$

- x^* solves the equation $x = g(x)$
- x^* solves root-finding problem $f(x) = g(x) - x = 0$

- *Sufficient* condition for existence + stability of fixed point:

  $|g'(x)| < k < 1$ for all x in an interval

Goals for today

- Rate of convergence
- Data analysis
- Constructing fixed-point methods to solve equations
- Newton method

Convergence to a fixed point

- Assume there is a fixed point x^*
- Let's look at the distance $\delta_n := x_n - x^*$

Convergence to a fixed point

- Assume there is a fixed point x^*
- Let's look at the distance $\delta_n := x_n - x^*$
- We have

$$
\begin{aligned}
\delta_{n+1} &= x_{n+1} - x^* \\
&= g(x_n) - x^* \\
&= g(x^* + \delta_n) - x^* \\
&\simeq g(x^*) + g'(x^*)\,\delta_n - x^* \qquad \text{Taylor} \\
&\simeq \delta_n\, g'(x^*) \qquad \text{since } g(x^*) = x^*
\end{aligned}
$$

Convergence to a fixed point II

- Asymptotically

$$\delta_{n+1} = \alpha\,\delta_n$$

with $\alpha := g'(x^*)$

Convergence to a fixed point II

- Asymptotically

$$\delta_{n+1} = \alpha\,\delta_n$$

with $\alpha := g'(x^*)$

- So

$$\delta_n = \alpha^n\,\delta_0$$

Convergence to a fixed point II

- Asymptotically

$$\delta_{n+1} = \alpha\, \delta_n$$

with $\alpha := g'(x^*)$

- So

$$\delta_n = \alpha^n\, \delta_0$$

- $|\delta_n| \to 0$ if $|\alpha| < 1$ **stable** fixed point
- $|\delta_n| \to \infty$ if $|\alpha| > 1$ **unstable** fixed point

Meaning of speed of convergence

- The fixed-point iteration *solves an equation*
- It is an *approximation algorithm* for solving that equation!

Meaning of speed of convergence

- The fixed-point iteration *solves an equation*
- It is an *approximation algorithm* for solving that equation!
- What does the speed of convergence mean?

Meaning of speed of convergence

- The fixed-point iteration *solves an equation*
- It is an *approximation algorithm* for solving that equation!
- What does the speed of convergence mean?

- The speed of convergence tells us **how good the approximation algorithm is**
- An algorithm that *converges faster* does *less work* to solve the problem

Collaboration I: Data analysis using logarithms

Data analysis using logarithms

Suppose we have data pairs (x_i, y_i).

1. On a semi-log graph, with a logarithmic scale for y, what are the variables we're plotting?
2. If we see a straight line on a semi-log graph, how are these new variables related?
3. So what is the relationship between the original variables x and y?
4. Repeat this if instead we have a straight line on a log–log graph, i.e. with logarithmic axes in both x and y.

Data analysis: Semi-log graphs

- Suppose we have some data that "look exponential"
- I.e. points (x_i, y_i) that we *think* satisfy

$$y \sim \exp(-\alpha x)$$

- How could we become "more confident" about this by plotting the data in a different way?

Data analysis: Semi-log graphs

- Suppose we have some data that “look exponential”
- I.e. points (x_i, y_i) that we *think* satisfy

$$y \sim \exp(-\alpha x)$$

- How could we become “more confident” about this by plotting the data in a different way?

- Plot the data with a logarithmic axis in the y direction
- Effectively we are plotting $\log(y_i)$ against x_i

Data analysis: Semi-log graphs II

- If plotting $\log(y_i)$ against x_i gives a straight line then

$$\log(y_i) \sim -Ax_i + B$$

for some constants A and B

Data analysis: Semi-log graphs II

- If plotting $\log(y_i)$ against x_i gives a straight line then

$$\log(y_i) \sim -Ax_i + B$$

  for some constants A and B

- So

$$y_i \sim \exp(B - Ax_i) = C\exp(-Ax_i)$$

Data analysis: Log-log graphs

- Suppose we have some data that “decay but not fast enough to be exponential”
- Suppose we plot the data with logarithmic axes in the x and y directions
- What can we conclude if we find a straight line?

Data analysis: Log-log graphs

- Suppose we have some data that "decay but not fast enough to be exponential"
- Suppose we plot the data with logarithmic axes in the x and y directions
- What can we conclude if we find a straight line?

$$\log(y_i) \sim -\alpha \log(x_i) + \beta$$

- So

$$y_i \sim \exp[\beta - \alpha \log(x_i) + \beta] = C x_i^{-\alpha}$$

– a **power law**

Cobweb diagrams

- A nice visualisation of fixed-point iteration is given by **cobweb diagrams**
- We can think of the fixed-point iteration $x_{n+1} = g(x_n)$ as the two successive operations

$$y_n = g(x_n)$$

$$x_{n+1} = y_n$$

Cobweb diagrams

- A nice visualisation of fixed-point iteration is given by **cobweb diagrams**
- We can think of the fixed-point iteration $x_{n+1} = g(x_n)$ as the two successive operations

$$y_n = g(x_n)$$

$$x_{n+1} = y_n$$

- So we plot a trajectory that visits the points $(x_n, g(x_n))$ and (x_{n+1}, x_{n+1})

Cobweb diagrams

- A nice visualisation of fixed-point iteration is given by **cobweb diagrams**
- We can think of the fixed-point iteration $x_{n+1} = g(x_n)$ as the two successive operations

$$y_n = g(x_n)$$

$$x_{n+1} = y_n$$

- So we plot a trajectory that visits the points $(x_n, g(x_n))$ and (x_{n+1}, x_{n+1})
- We see how the condition |derivative| < 1 leads to convergence

Constructing a useful iteration

- Given a root-finding problem $f(x) = 0$ to solve
- How can we *construct* a g such that the iteration $x_{n+1} = g(x_n)$ converges to a root of h?

Constructing a useful iteration

- Given a root-finding problem $f(x) = 0$ to solve
- How can we *construct* a g such that the iteration $x_{n+1} = g(x_n)$ converges to a root of h?

- We know that *if* it converges to x^*, then $x^* = g(x^*)$
- So we need to rearrange $f(x) = 0$ into $g(x) = x$

Constructing a useful iteration II

- We need to rearrange $h(x) = 0$ into $g(x) = x$
- One possibility is $g(x) = f(x) + x$

Constructing a useful iteration II

- We need to rearrange $h(x) = 0$ into $g(x) = x$
- One possibility is $g(x) = f(x) + x$

- In general

$$g(x) = \phi(x)\, f(x) + x$$

for *any* ϕ!

Constructing a useful iteration II

- We need to rearrange $h(x) = 0$ into $g(x) = x$
- One possibility is $g(x) = f(x) + x$
- In general

$$g(x) = \phi(x)\, f(x) + x$$

for *any* ϕ!

- We need $|g(x^*)| < 1$ at the fixed point
- This is tricky: *we don't know where the fixed point is*

Towards the Newton method

- So far we have seen that the convergence in fixed-point iterations has $\delta_{n+1} \simeq \alpha \delta_n$
- Thus the convergence is exponential
- How could we try to *accelerate* convergence?

Towards the Newton method

- So far we have seen that the convergence in fixed-point iterations has $\delta_{n+1} \simeq \alpha \delta_n$
- Thus the convergence is exponential
- How could we try to *accelerate* convergence?

- Look at the cobweb plot
- What would happen if $g'(x^*) = 0$?

Collaboration: A special fixed-point iteration

A special fixed-point iteration

Let's look for roots of f, i.e. x^* such that $f(x^*) = 0$.

Define $g(x) := x - \phi(x)f(x)$ and look for fixed points of g.

If we impose $g'(x^*) = 0$, what should ϕ satisfy?

Newton method as a fixed-point iteration

- Let's look for a fixed-point algorithm to solve $f(x) = 0$
- Define $g(x) := x - \phi(x) f(x)$

  where ϕ is as-yet unknown

Newton method as a fixed-point iteration

- Let's look for a fixed-point algorithm to solve $f(x) = 0$
- Define $g(x) := x - \phi(x) f(x)$

  where ϕ is as-yet unknown

- Let's *impose* that $g'(x^*) = 0$ at the root $f(x^*) = 0$, in other words at the fixed point $g(x^*) = x^*$

Newton method as a fixed-point iteration II

- We have $g(x) := x - \phi(x) f(x)$
- So $g'(x) = 1 - \phi'(x) f(x) - \phi(x) f'(x)$
- So $g'(x^*) = 1 - \phi(x^*) f'(x^*) = 0$ since $\phi'(x^*) = 0$

Newton method as a fixed-point iteration II

- We have $g(x) := x - \phi(x)f(x)$
- So $g'(x) = 1 - \phi'(x)f(x) - \phi(x)f'(x)$
- So $g'(x^*) = 1 - \phi(x^*)f'(x^*) = 0 \quad$ since $\phi'(x^*) = 0$

- So we must take $\phi(x^*) = \frac{1}{f'(x^*)}$
- Take $\phi(x) := \frac{1}{f'(x)}$

Newton method as a fixed-point iteration II

- We have $g(x) := x - \phi(x) f(x)$
- So $g'(x) = 1 - \phi'(x) f(x) - \phi(x) f'(x)$
- So $g'(x^*) = 1 - \phi(x^*) f'(x^*) = 0 \quad$ since $\phi'(x^*) = 0$

- So we must take $\phi(x^*) = \frac{1}{f'(x^*)}$
- Take $\phi(x) := \frac{1}{f'(x)}$
- This gives the **Newton method**:

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

Newton method: Taylor series derivation

- We want to solve $f(x) = 0$, starting at a guess x_0

Newton method: Taylor series derivation

- We want to solve $f(x) = 0$, starting at a guess x_0
- Let $x_1 = x_0 + \delta$ be an unknown, better approximation
- Impose $f(x_0 + \delta) = 0$

Newton method: Taylor series derivation

- We want to solve $f(x) = 0$, starting at a guess x_0
- Let $x_1 = x_0 + \delta$ be an unknown, better approximation
- Impose $f(x_0 + \delta) = 0$
- Taylor to first order:

$$f(x_0) + f'(x_0)\,\delta \simeq 0$$

- So $\delta = -f(x_0)/f'(x_0)$
- And $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

Newton method

- The Newton method is a *general* method to try to find a root for *any* function f:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newton method

- The Newton method is a *general* method to try to find a root for *any* function f:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Iteration with $g(x) = x - f(x)/f'(x)$

Newton method

- The Newton method is a *general* method to try to find a root for *any* function f:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Iteration with $g(x) = x - f(x)/f'(x)$

- Sufficiently close to a root, it converges *very fast*
- But it *may fail to converge*!

- The Babylonian algorithm is a special case

Convergence of Newton method

- There are known conditions for Newton to converge:
  - Smale α theory
  - Radii polynomials
- Interval Newton method: Use interval arithmetic to guarantee convergence

Rate of convergence

- How can we describe the rate of convergence?

Rate of convergence

- How can we describe the rate of convergence?

- If there are constants λ and α such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^\alpha} = \lambda$$

then the iteration is of **order** α

Rate of convergence

- How can we describe the rate of convergence?

- If there are constants λ and α such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^\alpha} = \lambda$$

  then the iteration is of **order** α

- $\delta_{n+1} \sim \delta_n^\alpha$ as $n \to \infty$

Rate of convergence

- How can we describe the rate of convergence?

- If there are constants λ and α such that

$$\lim_{n\to\infty} \frac{\left|x_{n+1} - x^*\right|}{\left|x_n - x^*\right|^\alpha} = \lambda$$

  then the iteration is of **order** α

- $\delta_{n+1} \sim \delta_n^\alpha$ as $n \to \infty$
- $\alpha = 1$: **“linear” convergence**
- $\alpha = 2$: **“quadratic” convergence**

Summary

- Rate of convergence
- Some data analysis
- Newton method:
    - General fixed-point method to solve $f(x) = 0$