6. Root finding

Summary of last lecture

- Taylor series
- Taylor polynomials + Lagrange remainder

- Solving equations as root finding
- Roots of polynomials: Fundamental Theorem of Algebra

Goals for today

- Iterative algorithms
- Root-finding algorithms using iteration

Root finding as a problem

- Suppose we want to solve the problem $\mathcal{Y} = \mathcal{F}(\mathcal{X})$

Root finding as a problem

- Suppose we want to solve the problem $\mathcal{Y} = \mathcal{F}(\mathcal{X})$
- Input $\mathcal{X} = f$ – a function
- Problem $\mathcal{F} =$ "calculate a root"
- Output $\mathcal{Y} =$ position x^* of the root

- $\mathcal{Y} = \mathcal{F}(\mathcal{X})$

- $\mathcal{Y} = \mathcal{F}(\mathcal{X})$

- Suppose we start from an initial guess y_0 of the *output*
- We hope to find an algorithm g (computational function) that *approximates* “solving $\mathcal{F}$ with input x”

- $\mathcal{Y} = \mathcal{F}(\mathcal{X})$

- Suppose we start from an initial guess y_0 of the *output*
- We hope to find an algorithm g (computational function) that *approximates* “solving $\mathcal{F}$ with input x”
- Some algorithms can *refine* an approximation to give a *better* approximation

Collaboration I

Cosine-ing

1. Open the scientific calculator app on your computer (or online) and make sure it is in “radians” mode
2. Start with an initial value and keep hitting the “cosine” key.
3. What happens after you do this many times?
4. Write down a formula that describes the *process*.
5. What is the *end result* of the process? Can you write down a formula for it?
6. How could we generalise this?

Visualising the cosine iteration

Cosine iteration

- What we have just done is an **iteration** or **recurrence**:

$$x_{n+1} = \cos(x_n)$$

starting from an initial value x_0

Cosine iteration

- What we have just done is an **iteration** or **recurrence**:

$$x_{n+1} = \cos(x_n)$$

starting from an initial value x_0

- We observe that the sequence $x_0, x_1, \ldots$ **converges**

Cosine iteration

- What we have just done is an **iteration** or **recurrence**:

$$x_{n+1} = \cos(x_n)$$

starting from an initial value x_0

- We observe that the sequence $x_0, x_1, \ldots$ **converges**
- I.e. there is a **limiting value** x^* with

$$x_n \to x^* \quad \text{as } n \to \infty$$

Cosine iteration

- What we have just done is an **iteration** or **recurrence**:

$$x_{n+1} = \cos(x_n)$$

starting from an initial value x_0

- We observe that the sequence $x_0, x_1, \ldots$ **converges**
- I.e. there is a **limiting value** x^* with

$$x_n \to x^* \quad \text{as } n \to \infty$$

Cosine iteration II

$$x_{n+1} = \cos(x_n)$$

- What can we say about the limit $n \to \infty$?

Cosine iteration II

$$x_{n+1} = \cos(x_n)$$

- What can we say about the limit $n \to \infty$?
- Taking the limit $n \to \infty$ gives

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

Cosine iteration II

$$x_{n+1} = \cos(x_n)$$

- What can we say about the limit $n \to \infty$?
- Taking the limit $n \to \infty$ gives

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

- So x^* *is a solution* of the transcendental equation

$$\cos(x) = x$$

- The iteration gives an approximation algorithm to calculate x^* as closely as we want!

Questions about iterations

- Some natural questions arise:

Questions about iterations

- Some natural questions arise:

- Which functions g give iterations that converge?

Questions about iterations

- Some natural questions arise:
- Which functions g give iterations that converge?
- How fast is the convergence?
  - I.e. how does $|x_n - x^*|$ depend on n?

Questions about iterations

- Some natural questions arise:
- Which functions g give iterations that converge?
- How fast is the convergence?
  - I.e. how does $|x_n - x^*|$ depend on n?
- To solve $h(x) = 0$, how should we choose g?

Fixed-point iteration

- Consider a general iteration $x_{n+1} = g(x_n)$
  - for an algorithm (function) g and initial guess x_0

Fixed-point iteration

- Consider a general iteration $x_{n+1} = g(x_n)$
  - for an algorithm (function) g and initial guess x_0
- It produces a sequence $x_0, x_1, x_2, \ldots$

Fixed-point iteration

- Consider a general iteration $x_{n+1} = g(x_n)$
  - for an algorithm (function) g and initial guess x_0
- It produces a sequence $x_0, x_1, x_2, \ldots$
- *If* the iteration converges, $x_n \to x^*$ as $n \to \infty$, then

$$g(x^*) = x^*$$

provided g is **continuous**

Fixed-point iteration

- Consider a general iteration $x_{n+1} = g(x_n)$
  - for an algorithm (function) g and initial guess x_0
- It produces a sequence $x_0, x_1, x_2, \ldots$
- *If* the iteration converges, $x_n \to x^*$ as $n \to \infty$, then

$$g(x^*) = x^*$$

provided g is **continuous**

- We say that x^* is a **fixed point** of g
- Hence it solve $f(x) := g(x) - x = 0$

Existence of fixed point

- Given a function g, does it *have* a fixed point?

Existence of fixed point

- Given a function g, does it *have* a fixed point?
- A *sufficient* condition is given by the Banach fixed-point / contraction mapping theorem:

If g *is* continuous *and maps* $[a, b]$ *into itself, then there* exists *a fixed point*

Existence of fixed point

- Given a function g, does it *have* a fixed point?
- A *sufficient* condition is given by the Banach fixed-point / contraction mapping theorem:

  If g is continuous *and maps $[a, b]$ into itself, then there* exists *a fixed point*

- If $|g'(x)| \leq k \; \forall x \in [a, b]$ with $k < 1$, then the fixed point is *unique*

Existence of fixed point

- Given a function g, does it *have* a fixed point?
- A *sufficient* condition is given by the Banach fixed-point / contraction mapping theorem:

  If g *is* continuous *and maps* $[a, b]$ *into itself, then there* exists *a fixed point*

- If $|g'(x)| \leq k \; \forall x \in [a, b]$ with $k < 1$, then the fixed point is *unique*

- g is called a **contraction mapping**
- Since it contracts distances: $|g(x) - g(y)| \leq k|x - y|$

Iterations as dynamical systems

- An iteration is a **discrete-time dynamical system**

Iterations as dynamical systems

- An iteration is a **discrete-time dynamical system**
- We are solving a stationary (time-independent) problem by solving a time-*dependent* problem!
- By introducing a **fictitious** (discrete) time n

Iterations as dynamical systems

- An iteration is a **discrete-time dynamical system**
- We are solving a stationary (time-independent) problem by solving a time-*dependent* problem!
- By introducing a **fictitious** (discrete) time n

- This is a common trick
  - e.g. Markov chains, elliptic PDEs

Collaboration II

Convergence

Suppose that x_0 is *close* to x^*. How does the iteration $x_{n+1} = g(x_n)$ behave?

1. Define the distance $\delta_n := x_n - x^*$
2. Can you find the approximate dynamics of δ?
3. When will the dynamics converge to the fixed point?

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

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

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

$$\delta_{n+1} = \alpha^n\, \delta_0$$

Convergence to a fixed point II

- Asymptotically

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

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

- So

$$\delta_{n+1} = \alpha^n\,\delta_0$$

- δ_n **decays** if $|\alpha| < 1$ – **stable** fixed point
- δ_n **grows** 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**

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**

- If we can find a different algorithm which *converges faster*, we will need to do *less work* to solve the problem

Collaboration III

Designing fixed-point iterations

Let's try to solve the equation $x^2 - x + 1 = 0$ using fixed-point iteration.

1. Can you come up with a fixed-point iteration that does this?
2. Which root does it find?
3. Can you find the other one?

Summary

- Iterations *may* converge
- If they converge, they converge to a **fixed point**
- Fixed points solve equations

- We can calculate a **convergence rate** to the fixed point
- This tells us **how good** the algorithm is