8. Root finding in higher dimensions

Summary of the previous lecture

- Convergence of iterations $x_{n+1} = g(x_n)$
- To a fixed point x^* such that $g(x^*) = x^*$
- Analysing data on log scales
- The Newton method for solving $f(x) = 0$:

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

- Order of convergence α of an iteration:

$$\delta_n := x_n - x^*; \qquad \delta_{n+1} \sim \delta_n^{\alpha}$$

Goals for today

- Solving equations in > 1 dimensions
- Implicit equations and their geometry
- Systems of nonlinear equations
- Methods for finding roots

Nonlinear equations in > 1 dimensions

- So far we have solved $f(x) = 0$ for $f : \mathbb{R} \to \mathbb{R}$

- What about functions with more variables?
- For example

$$x^2 + y^2 = 1$$

Nonlinear equations in > 1 dimensions

- So far we have solved $f(x) = 0$ for $f : \mathbb{R} \to \mathbb{R}$
- What about functions with more variables?
- For example

$$x^2 + y^2 = 1$$

- More generally:

$$f(x, y) = 0 \quad \text{with } f : \mathbb{R}^2 \to \mathbb{R}$$

Implicit equations and constraints

- $x^2 + y^2 = 1$ is an example of an **implicit** equation
- We can also think of this as a **constraint**
- It **relates** possible pairs (x, y) that solve the equation

Implicit equations and constraints

- $x^2 + y^2 = 1$ is an example of an **implicit** equation
- We can also think of this as a **constraint**
- It **relates** possible pairs (x, y) that solve the equation

- **Solving** the equation means finding the **solution set**

$$\{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}$$

Collaboration I

Implicit equations

1. What kind of object does an equation like $f(x, y) = 0$ (usually) represent?
2. Can you solve this for y?
3. Try with $x^2 + y^2 - 1 = 0$
4. When does this fail?

Can we solve implicit equations

- We can try to solve $f(x, y) = 0$ for y as $y = g(x)$:

$$x^2 + g(x)^2 - 1 = 0$$

$$g(x) = \pm\sqrt{1 - x^2}$$

Can we solve implicit equations

- We can try to solve $f(x, y) = 0$ for y as $y = g(x)$:

$$x^2 + g(x)^2 - 1 = 0$$

$$g(x) = \pm\sqrt{1 - x^2}$$

- The result is in general *non-unique* – not a function

Can we solve implicit equations

- We can try to solve $f(x, y) = 0$ for y as $y = g(x)$:

$$x^2 + g(x)^2 - 1 = 0$$

$$g(x) = \pm\sqrt{1 - x^2}$$

- The result is in general *non-unique* – not a function
- But often "locally unique" **branch** – smooth function of x

Can we solve implicit equations

- We can try to solve $f(x, y) = 0$ for y as $y = g(x)$:

$$x^2 + g(x)^2 - 1 = 0$$

$$g(x) = \pm\sqrt{1 - x^2}$$

- The result is in general *non-unique* – not a function
- But often "locally unique" **branch** – smooth function of x
- This is proved by the **implicit function theorem**

Plotting implicit functions

- We can plot implicit functions as **contours** or **level sets**
- Think of $f(x, y)$ as height of surface at (x, y)
- **Level set**: Set where the height is a given constant c

$$\{(x, y) : f(x, y) = c\}$$

Plotting implicit functions

- We can plot implicit functions as **contours** or **level sets**
- Think of $f(x, y)$ as height of surface at (x, y)
- **Level set**: Set where the height is a given constant c

$$\{(x, y) : f(x, y) = c\}$$

- `contour` function: `levels = [0, 1, 3]` specifies values of c
- This uses the marching squares algorithm
- Alternative: **numerical continuation** – "numerical version of implicit function theorem"

  – "follow the curve round"

What can go wrong

- Different types of “pathology” can occur:
- $x\,y = 0$
- $y^2 \;=\; x^2\,(x\;+a)\,(1+x)$

Higher dimensions

- For a single constraint:
- $f(x, y) = 0$ is a 1-dimensional **curve** in 2D
- $f(x, y, z) = 0$ is a 2-dimensional **surface** in 3D

Higher dimensions

- For a single constraint:
- $f(x, y) = 0$ is a 1-dimensional **curve** in 2D
- $f(x, y, z) = 0$ is a 2-dimensional **surface** in 3D

- $f(\mathbf{x}) = 0$ is an $(n - 1)$-dimensional “surface” in nD
- A **codimension**-1 **manifold**

Higher dimensions

- For a single constraint:
- $f(x, y) = 0$ is a 1-dimensional **curve** in 2D
- $f(x, y, z) = 0$ is a 2-dimensional **surface** in 3D

- $f(\mathbf{x}) = 0$ is an $(n-1)$-dimensional "surface" in nD
- A **codimension**-1 **manifold**

- What happens if we add more constraints?

Systems of nonlinear equations

- Now let's think about **systems** of nonlinear equations
- e.g. 2 equations in 2 unknowns:

$$\begin{cases} f(x,y) & := x^2 + y^2 - 3 = 0 \\ g(x,y) & := \left(\frac{x}{2}\right)^2 + (y - \frac{1}{2})^2 - 1 = 0 \end{cases}$$

Systems of nonlinear equations

- Now let's think about **systems** of nonlinear equations
- e.g. 2 equations in 2 unknowns:

$$\begin{cases} f(x,y) & := x^2 + y^2 - 3 = 0 \\ g(x,y) & := \left(\frac{x}{2}\right)^2 + (y - \frac{1}{2})^2 - 1 = 0 \end{cases}$$

- We want roots (x^*, y^*) where the equations hold *simultaneously*:

$$f(x^*, y^*) = g(x^*, y^*) = 0$$

- What will the result look like?

Systems of nonlinear equations II

- **Intersections** of curves: significantly harder than 1D
- 2 constraints in 2 dimensions
- So we expect 0-dimensional **points** as solutions

Systems of nonlinear equations II

- **Intersections** of curves: significantly harder than 1D
- 2 constraints in 2 dimensions
- So we expect 0-dimensional **points** as solutions

- How should we solve these?

Collaboration II

Roots of several functions

1. How should we formulate $f(x, y) = 0$ and $g(x, y) = 0$ to "look like" a single root-finding problem?
2. What kind of numerical methods might we try to apply?
3. How could you try to design a numerical method to solve

$$\begin{cases} x^2 + y^2 = 1 \\ x = y \end{cases}$$

Vector form of a system of equations

- Given a system of equations

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- We can rewrite the system into a **vector form**
- Rename the variables

$$x_1 := x; \qquad f_1 := f$$
$$x_2 := y; \qquad f_2 := g$$

Vector form of a system of equations

- Given a system of equations

$$f(x, y) = 0$$
$$g(x, y) = 0$$

- We can rewrite the system into a **vector form**
- Rename the variables

$$x_1 := x; \qquad f_1 := f$$
$$x_2 := y; \qquad f_2 := g$$

- We obtain the system $f_i(x_1, x_2) = 0$ for $i = 1, 2$

Vector form II

- In general introduce the vector $\mathbf{x} = (x_1, \ldots, x_n)$
- And the **vector-valued function** $\mathbf{f} = (f_1, \ldots, f_n)$

Vector form II

- In general introduce the vector $\mathbf{x} = (x_1, \dots, x_n)$
- And the **vector-valued function** $\mathbf{f} = (f_1, \dots, f_n)$

- We get the vector form

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

Numerical methods for systems of equations

- If $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, we expect the roots to be **isolated points**
- Which numerical methods could we try?

Numerical methods for systems of equations

- If $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, we expect the roots to be **isolated points**
- Which numerical methods could we try?

- Fixed-point iteration: it's difficult to find a suitable iteration
- Newton method

Multidimensional Newton

- Apply the same idea as for 1D Newton

Multidimensional Newton

- Apply the same idea as for 1D Newton

- Start from an initial guess $\mathbf{x}_0$
- Try to improve to a guess $\mathbf{x}_1 = \mathbf{x}_0 + \boldsymbol{\delta}$

Multidimensional Newton

- Apply the same idea as for 1D Newton

- Start from an initial guess $\mathbf{x}_0$
- Try to improve to a guess $\mathbf{x}_1 = \mathbf{x}_0 + \boldsymbol{\delta}$

- So impose $\mathbf{f}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{0}$
- Solve for $\boldsymbol{\delta}$

Collaboration III

Newton in higher dimensions

Suppose $\mathbf{f}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{0}$

1. What should we do with this?
2. What happens when you do that?
3. What mathematical operation will we need to be able to carry out numerically in order to do so?

Multidimensional Newton II

- What can we do with $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$?

Multidimensional Newton II

- What can we do with $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$?

- We need a **Taylor expansion in higher dimensions**:

$$\mathbf{f}(\mathbf{a} + \boldsymbol{\delta}) = \mathbf{f}(\mathbf{a}) + \mathsf{D}\mathbf{f}(\mathbf{a}) \cdot \boldsymbol{\delta} + \mathcal{O}(\|\boldsymbol{\delta}\|^2)$$

- $\mathsf{J}_f(\mathbf{a}) := \mathsf{D}\mathbf{f}(\mathbf{a})$ is the **Jacobian matrix**
- $(\mathsf{J}_f)(\mathbf{a})_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{a})$

Taylor in 2 dimensions

- Consider $f(x + \delta, y + \epsilon)$
- Let's expand to first order in δ and ϵ

Taylor in 2 dimensions

- Consider $f(x+\delta, y+\epsilon)$
- Let's expand to first order in δ and ϵ

- Apply Taylor in one variable repeatedly:

$$\begin{aligned} f(x+\delta, y+\epsilon) &\simeq f(x, y+\epsilon) + \delta \frac{\partial f}{\partial x}(x, y+\epsilon) && (1) \\ &\simeq f(x,y) + \delta \frac{\partial f}{\partial x}(x,y) + \epsilon \frac{\partial f}{\partial y}(x,y) && (2) \end{aligned}$$

Multidimensional Newton III

- So $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) \simeq \mathbf{f}(\mathbf{x}_n) + \mathsf{J}_{\mathbf{f}}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n$

Multidimensional Newton III

- So $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) \simeq \mathbf{f}(\mathbf{x}_n) + \mathsf{J}_{\mathbf{f}}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n$
- We need to solve

$$\mathsf{J}_{\mathbf{f}}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n = -\mathbf{f}(\mathbf{x}_n)$$

- What is this?

Multidimensional Newton III

- So $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) \simeq \mathbf{f}(\mathbf{x}_n) + \mathsf{J}_{\mathbf{f}}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n$

- We need to solve

$$\mathsf{J}_{\mathbf{f}}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n = -\mathbf{f}(\mathbf{x}_n)$$

- What is this?
- It's a *system of* ***linear*** *equations*:

  matrix $\times$ (unknown vector) = (known vector)

Multidimensional Newton IV

- We have reduced “solving a nonlinear system” to the *iterated* solution of *linear* systems

Multidimensional Newton IV

- We have reduced “solving a nonlinear system” to the *iterated* solution of *linear* systems
- We need to know how to **solve the linear system**

$$\mathsf{A} \cdot \mathbf{x} = \mathbf{b}$$

Multidimensional Newton IV

- We have reduced "solving a nonlinear system" to the *iterated* solution of *linear* systems
- We need to know how to **solve the linear system**

$$\mathsf{A} \cdot \mathbf{x} = \mathbf{b}$$

- Mathematically this is given by the inverse matrix

$$\mathbf{x} = \mathsf{A}^{-1}\,\mathbf{b}$$

Multidimensional Newton IV

- We have reduced "solving a nonlinear system" to the *iterated* solution of *linear* systems
- We need to know how to **solve the linear system**

$$\mathsf{A} \cdot \mathbf{x} = \mathbf{b}$$

- Mathematically this is given by the inverse matrix

$$\mathbf{x} = \mathsf{A}^{-1}\,\mathbf{b}$$

- Numerically: we prefer to directly *solve the linear system*

Towards optimization

- **Optimization**: Find the minimum of $f : \mathbb{R}^n \to \mathbb{R}$

Towards optimization

- **Optimization**: Find the minimum of $f : \mathbb{R}^n \to \mathbb{R}$
- Find zeros of the gradient ∇f!

Towards optimization

- **Optimization**: Find the minimum of $f : \mathbb{R}^n \to \mathbb{R}$

- Find zeros of the gradient ∇f!
- Use Newton on ∇f

Towards optimization

- **Optimization**: Find the minimum of $f : \mathbb{R}^n \to \mathbb{R}$

- Find zeros of the gradient ∇f!
- Use Newton on ∇f

- We need the Jacobian of the gradient
- Symmetric **Hessian matrix** H:

$$H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Implicit function theorem

- Consider $f(x, y) = 0$ again
- Suppose we have one point on the curve: $f(x_0, y_0) = 0$
- What happens close to that point?

Implicit function theorem

- Consider $f(x, y) = 0$ again
- Suppose we have one point on the curve: $f(x_0, y_0) = 0$
- What happens close to that point?
- Implicit function theorem (2D, approximate statement):

Suppose $\frac{\partial f}{\partial y}(x_0, y_0) \neq 0$.

Then for x *in a neighbourhood of* x_0, *there exists a smooth function* $g(x)$ *with* $f(x, g(x)) = 0$ *with* $g(x_0) = y_0$.

$g'(x)$ *may be calculated by implicit differentiation.*

Solving linear systems in Julia

- To solve $A \cdot \mathbf{x} = \mathbf{b}$ in Julia:

```
using LinearAlgebra   # standard library; no installation re

A = rand(2, 2)    # random matrix
b = rand(2)       # random vector

x = A \ b

residual = (A * x) - b
```

Solving linear systems in Julia

- To solve $A \cdot \mathbf{x} = \mathbf{b}$ in Julia:

```
using LinearAlgebra    # standard library; no installation re

A = rand(2, 2)    # random matrix
b = rand(2)       # random vector

x = A \ b

residual = (A * x) - b
```

- `A * x` is standard matrix–vector multiplication
- `A \ b` is a **black box** that we will open up later in the course

Summary

- Geometry of higher-dimensional functions
- Derivatives in higher dimensions
- Newton method in higher dimensions