7. Root finding in higher dimensions

Last time

- Feedback from problem set 1
- Creating matrices in Julia

Goals for today

- Nonlinear equations in higher dimensions
- Systems of nonlinear equations
- Review of derivatives for functions $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$
- Newton in higher dimensions

Nonlinear equations in more dimensions

- Have seen numerical methods to solve $f(x) = 0$ for $f : \mathbb{R} \to \mathbb{R}$

- What about functions with more variables?
- E.g. $f(x, y) = 0$, so $f : \mathbb{R}^2 \to \mathbb{R}$
- e.g. $x^2 + y^2 = 1$

Nonlinear equations in more dimensions

- Have seen numerical methods to solve $f(x) = 0$ for $f : \mathbb{R} \to \mathbb{R}$

- What about functions with more variables?
- E.g. $f(x, y) = 0$, so $f : \mathbb{R}^2 \to \mathbb{R}$
- e.g. $x^2 + y^2 = 1$
- **Implicit equation**: relates values of x and y
- **Solving** the equation means finding **solution set**:
- $\{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}$

Constraints

- Think of $f(x, y) = 0$ as single **constraint**
- Try to solve for $y = g(x)$

Constraints

- Think of $f(x, y) = 0$ as single **constraint**
- Try to solve for $y = g(x)$

- e.g. $x^2 + g(x)^2 = 1 \Longrightarrow g(x) = \pm\sqrt{1 - x^2}$

Constraints

- Think of $f(x, y) = 0$ as single **constraint**
- Try to solve for $y = g(x)$

- e.g. $x^2 + g(x)^2 = 1 \implies g(x) = \pm\sqrt{1 - x^2}$
- Non-unique in general
- Often “locally unique”, smooth **function** of x
- Proved by **implicit function theorem**

Plotting implicit functions

- Plot as **contours** or **level sets**
- Think of $f(x, y)$ as height of surface at (x, y)
- **Level set**: Set where the height is some constant c
- `contour` function
- Uses marching squares algorithm
- Alternative: **numerical continuation** – "numerical version of implicit function theorem"

What can happen

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

Higher dimensions

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

Higher dimensions

- $f(x, y) = 0$ is a 1-dimensional **curve** in 2D
- $f(x, y, z) = 0$ is a 2-dimensional **surface** in 3D
- In general, $f(\mathbf{x}) = 0$ is an $(n-1)$-dimensional **manifold** in n dimensions
- i.e. **codimension 1** corresponding to 1 constraint

Systems of nonlinear equations

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

$$f(x,y) := x^2 + y^2 - 3 = 0$$
$$g(x,y) := \left(\frac{x}{2}\right)^2 + (y - 0.5)^2 - 1 = 0$$

Systems of nonlinear equations

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

$$f(x, y) := x^2 + y^2 - 3 = 0$$
$$g(x, y) := \left(\frac{x}{2}\right)^2 + (y - 0.5)^2 - 1 = 0$$

- Want *joint* roots $f(x^*, y^*) = g(x^*, y^*) = 0$
- What will result look like?

Systems of nonlinear equations II

- **Intersections** of curves: significantly harder than 1D
- 2 constraints in 2 dimensions $\Longrightarrow$ expect 0-dimensional **points**
- How should we solve these?

Systems of nonlinear equations II

- **Intersections** of curves: significantly harder than 1D
- 2 constraints in 2 dimensions $\Longrightarrow$ expect 0-dimensional **points**
- How should we solve these?
- If the functions are polynomials, study of **algebraic geometry**

Vector form of system

- Rewrite system of equations into vector form:
- Write $f_1 = f; f_2 = g; x_1 = x; x_2 = y$
- Get system $f_i(x_1, x_2) = 0$ for $i = 1, 2$

Vector form of system

- Rewrite system of equations into vector form:
- Write $f_1 = f$; $f_2 = g$; $x_1 = x$; $x_2 = y$
- Get system $f_i(x_1, x_2) = 0$ for $i = 1, 2$
- Write vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{f} = (f_1, \ldots, f_n)$
- Vector form: $\mathbf{f}(\mathbf{x}) = \mathbf{0}$

Numerical methods for systems of equations

- If $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, expect **isolated points** as solutions
- Which numerical method could we try?

Numerical methods for systems of equations

- If $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, expect **isolated points** as solutions
- Which numerical method could we try?
- Fixed-point iteration
- Multidimensional bisection
- Interval arithmetic

Multidimensional Newton

- Apply same idea as for 1D Newton

Multidimensional Newton

- Apply same idea as for 1D Newton
- Initial guess $\mathbf{x}_0$; try to solve $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$

Multidimensional Newton

- Apply same idea as for 1D Newton
- Initial guess $\mathbf{x}_0$; try to solve $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$
- Need **Taylor expansion for higher dimensions**:

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

Multidimensional Newton

- Apply same idea as for 1D Newton
- Initial guess $\mathbf{x}_0$; try to solve $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$
- Need **Taylor expansion for higher dimensions**:

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

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

Multidimensional Newton

- Apply same idea as for 1D Newton
- Initial guess $\mathbf{x}_0$; try to solve $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) = \mathbf{0}$
- Need **Taylor expansion for higher dimensions**:

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

- So $\mathbf{f}(\mathbf{x}_n + \boldsymbol{\delta}) \simeq \mathbf{f}(\mathbf{x}_n) + \mathsf{J}(\mathbf{x}_n) \cdot \boldsymbol{\delta}_n$
- Need to solve $\boldsymbol{\delta}_n = -\mathsf{J}(\mathbf{x}_n)^{-1}\mathbf{f}(\mathbf{x}_n)$
- $\mathsf{J} := \mathsf{Df}(\mathbf{x}_n)$ is **Jacobian matrix** of all partial derivatives

Multidimensional Newton II

- Reduced nonlinear system to iterated solution of **linear system**
- Mathematics: calculate **matrix inverse**

Multidimensional Newton II

- Reduced nonlinear system to iterated solution of **linear system**
- Mathematics: calculate **matrix inverse**
- Numerics: instead **solve linear system** (see later)

Multidimensional Newton II

- Reduced nonlinear system to iterated solution of **linear system**
- Mathematics: calculate **matrix inverse**
- Numerics: instead **solve linear system** (see later)
- Can show that multi-dimensional Newton also has quadratic convergence if close enough to root

Solving linear systems in Julia

- To solve linear system $A \cdot \mathbf{x} = \mathbf{b}$ 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

Implicit function theorem

- Consider $f(x, y) = 0$ again
- Suppose we have found one point on curve: $f(x_0, y_0) = 0$

Implicit function theorem

- Consider $f(x, y) = 0$ again
- Suppose we have found one point on 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 found one point on 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 there exists $g(x)$ *with* $f(x, g(x)) = 0$ *and* $g(x_0) = y_0$ *in a neighbourhood.*
  g *has similar smoothness to* f*. Its derivative may be calculated by implicit differentiation*

Towards optimization

- **Optimization**: Find minima (or maxima) of a function $f : \mathbb{R}^n \to \mathbb{R}$
- How can we do this?

Towards optimization

- **Optimization**: Find minima (or maxima) of a function $f : \mathbb{R}^n \to \mathbb{R}$
- How can we do this?
- Find zeros of gradient ∇f!
- Necessary condition

Towards optimization

- **Optimization**: Find minima (or maxima) of a function $f : \mathbb{R}^n \to \mathbb{R}$
- How can we do this?
- Find zeros of gradient ∇f!
- Necessary condition
- Use Newton on ∇f and $D(\nabla f)$
- (Symmetric) **Hessian matrix** H
- Matrix $\left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{i,j}$

Summary

- Geometry of higher-dimensional functions
- Higher-dimensional derivatives
- Newton method in higher dimensions