Numerical Analysis / Scientific Computing

CS450

Andreas Kloeckner

Spring 2019

Outline

What's the point of this class?

'*Scientific Computing*' describes a family of approaches to obtain approximate solutions to problems *once they've been stated mathematically*.

Name some applications:

What do we study, and how?

Problems with real numbers (i.e. *continuous* problems)

What's the general approach?

What makes for *good* numerics?

How good of an answer can we expect to our problem?

How fast can we expect the computation to complete?

Implementation concerns

How do numerical methods *get implemented*?

Class web page

https://bit.ly/cs450-s19

- Assignments
  - HW0!
  - Pre-lecture quizzes
  - In-lecture interactive content (bring computer or phone if possible)
- Textbook
- Exams
- Class outline (with links to notes/demos/activities/quizzes)
- Virtual Machine Image
- Piazza
- Policies
- Video
- Inclusivity Statement

Programming Language: Python/numpy

- Reasonably readable
- Reasonably beginner-friendly
- Mainstream (top 5 in 'TIOBE Index')
- Free, open-source
- Great tools and libraries (not just) for scientific computing
- Python 2/3? 3!
- `numpy`: Provides an array datatype
  Will use this and `matplotlib` all the time.
- See class web page for learning materials

Demo: Sum the squares of the integers from 0 to 100. First without numpy, then with numpy.

Supplementary Material

- Numpy (from the SciPy Lectures)
- 100 Numpy Exercises
- Dive into Python3

Sources for these Notes

- M.T. Heath, Scientific Computing: An Introductory Survey, Revised Second Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA. 2018.
- CS 450 Notes by Edgar Solomonik
- Various bits of prior material by Luke Olson

Open Source <3

These notes (and the accompanying demos) are open-source!

Bug reports and pull requests welcome:
https://github.com/inducer/numerics-notes

What problems *can* we study in the first place?

To be able to compute a solution (through a process that introduces errors), the problem...

If it satisfies these criteria, the problem is called *well-posed*. Otherwise, *ill-posed*.

Dependency on Inputs

We excluded discontinuous problems–because we don't stand much chance for those.

. . . what if the problem's input dependency is just *close to discontinuous*?

Approximation

When does approximation happen?

Demo: Truncation vs Rounding

Example: Surface Area of the Earth

Compute the surface area of the earth.
What parts of your computation are approximate?

Measuring Error

How do we measure error?

Idea: Consider all error as being *added onto* the result.

Recap: Norms

What's a norm?

Define *norm*.

Norms: Examples

Examples of norms?

Demo: Vector Norms

Norms: Which one?

Does the choice of norm really matter much?

Norms and Errors

If we're computing a vector result, the error is a vector.
That's not a very useful answer to 'how big is the error'.
What can we do?

Forward/Backward Error

Suppose *want* to compute $y = f(x)$, but *approximate* $\hat{y} = \hat{f}(x)$.

What are the forward error and the backward error?

Forward/Backward Error: Example

Suppose you wanted $y = \sqrt{2}$ and got $\hat{y} = 1.4$.
What's the (magnitude of) the forward error?

Forward/Backward Error: Example

Suppose you wanted $y = \sqrt{2}$ and got $\hat{y} = 1.4$.
What's the (magnitude of) the backward error?

Forward/Backward Error: Observations

What do you observe about the relative manitude of the relative errors?

Sensitivity and Conditioning

What can we say about amplification of error?

Example: Condition Number of Evaluating a Function

$y = f(x)$. Assume f differentiable.

Demo: Conditioning of Evaluating tan

Stability and Accuracy

Previously: Considered *problems* or *questions*.

Next: Considered *methods*, i.e. computational approaches to find solutions.

When is a method *accurate*?

When is a method *stable*?

Getting into Trouble with Accuracy and Stability

How can I produce inaccurate results?

In-Class Activity: Forward/Backward Error

In-class activity: Forward/Backward Error

Wanted: Real Numbers... in a computer

Computers can represent *integers*, using bits:

$$23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (10111)_2$$

How would we represent fractions?

Fixed-Point Numbers

Suppose we use units of 64 bits, with 32 bits for exponents $\geqslant 0$ and 32 bits for exponents < 0. What numbers can we represent?

How many ‘digits’ of relative accuracy (think relative rounding error) are available for the smallest vs. the largest number?

Floating Point Numbers

Convert $13 = (1101)_2$ into floating point representation.

What pieces do you need to store an FP number?

Floating Point: Implementation, Normalization

Previously: Consider *mathematical* view of FP.

Next: Consider *implementation* of FP in hardware.

Do you notice a source of inefficiency in our number representation?

Unrepresentable numbers?

Can you think of a somewhat central number that we cannot represent as

$$x = (1.\_\_\_\_\_\_\_\_\_\_\_\_)_2 \cdot 2^{-p}?$$

Demo: Picking apart a floating point number

Subnormal Numbers

What is the smallest representable number in an FP system with 4 stored bits in the significand and an exponent range of $[-7, 7]$?

Subnormal Numbers II

What is the smallest representable number in an FP system with 4 stored bits in the significand and an exponent range of $[-7, 7]$? (Attempt 2)

Why learn about subnormals?

Underflow

- FP systems without subnormals will *underflow* (return 0) as soon as the exponent range is exhausted.
- This smallest representable *normal* number is called the *underflow level*, or *UFL*.
- Beyond the underflow level, subnormals provide for *gradual underflow* by 'keeping going' as long as there are bits in the significand, but it is important to note that subnormals don't have as many accurate digits as normal numbers.
- Analogously (but much more simply–no 'supernormals'): the overflow level, *OFL*.

Rounding Modes

How is rounding performed? (Imagine trying to represent π.)

$$(\underbrace{1.1101010}_{\text{representable}}11)_2$$

What is done in case of a tie? $0.5 = (0.1)_2$ ("Nearest"?)

Demo: Density of Floating Point Numbers

Demo: Floating Point vs Program Logic

Smallest Numbers Above...

- What is smallest FP number > 1? Assume 4 bits in the significand.

What's the smallest FP number > 1024 in that same system?

Can we give that number a name?

Unit Roundoff

Unit roundoff or *machine precision* or *machine epsilon* or $\varepsilon_{\text{mach}}$ is the smallest number such that

$$\text{float}(1+\varepsilon) > 1.$$

- Assuming round-to-nearest, in the above system, $\varepsilon_{\text{mach}} = (0.00001)_2$.
- Note the extra zero.
- Another, related, quantity is *ULP*, or *unit in the last place*. ($\varepsilon_{\text{mach}} = 0.5\,\text{ULP}$)

FP: Relative Rounding Error

What does this say about the relative error incurred in floating point calculations?

FP: Machine Epsilon

What's that same number for double-precision floating point? (52 bits in the significand)

Demo: Floating Point and the Harmonic Series

In-Class Activity: Floating Point

In-class activity: Floating Point

Implementing Arithmetic

How is floating point addition implemented?
Consider adding $a = (1.101)_2 \cdot 2^1$ and $b = (1.001)_2 \cdot 2^{-1}$ in a system with three bits in the significand.

Problems with FP Addition

What happens if you subtract two numbers of very similar magnitude?
As an example, consider $a = (1.1011)_2 \cdot 2^0$ and $b = (1.1010)_2 \cdot 2^0$.

Demo: Catastrophic Cancellation

Supplementary Material

- Josh Haberman, Floating Point Demystified, Part 1
- David Goldberg, What every computer programmer should know about floating point

Outline

Solving a Linear System

Given:

- $m \times n$ matrix A
- m-vector b

What are we looking for here, and when are we allowed to ask the question?

Matrix Norms

What norms would we apply to matrices?

Matrix Norm Properties

What is $\|A\|_1$? $\|A\|_\infty$?

How do matrix and vector norms relate for $n \times 1$ matrices?

Demo: Matrix norms

Properties of Matrix Norms

Matrix norms inherit the vector norm properties:

- $\|A\| > 0 \Leftrightarrow A \neq 0$.
- $\|\gamma A\| = |\gamma| \, \|A\|$ for all scalars γ.
- Obeys triangle inequality $\|A + B\| \leqslant \|A\| + \|B\|$

But also some more properties that stem from our definition:

Conditioning

What is the condition number of solving a linear system $Ax = b$?

Conditioning of Linear Systems: Observations

Showed $\kappa(\text{Solve } A\mathsf{x} = \mathsf{b}) \le \left\|A^{-1}\right\| \|A\|$.
I.e. found an *upper bound* on the condition number. With a little bit of fiddling, it's not too hard to find examples that achieve this bound, i.e. that it is *sharp*.

So we've found the *condition number of linear system solving*, also called the condition number of the matrix A:

$$\operatorname{cond}(A) = \kappa(A) = \|A\| \left\|A^{-1}\right\|.$$

Conditioning of Linear Systems: More properties

- cond is relative to a given norm. So, to be precise, use

$$\text{cond}_2 \quad \text{or} \quad \text{cond}_\infty .$$

- If A^{-1} does not exist: $\text{cond}(A) = \infty$ by convention.

What is $\kappa(A^{-1})$?

What is the condition number of matrix-vector multiplication?

Demo: Condition number visualized

Demo: Conditioning of 2x2 Matrices

Residual Vector

What is the residual vector of solving the linear system

$$b = Ax?$$

Residual and Error: Relationship

How do the (norms of the) residual vector r and the error $\Delta x = x - \widehat{x}$ relate to one another?

Changing the Matrix

So far, all our discussion was based on changing the right-hand side, i.e.

$$A\mathrm{x} = \mathrm{b} \quad \to \quad A\widehat{\mathrm{x}} = \widehat{\mathrm{b}}.$$

The matrix consists of FP numbers, too–it, too, is approximate. I.e.

$$A\mathrm{x} = \mathrm{b} \quad \to \quad \hat{A}\widehat{\mathrm{x}} = \mathrm{b}.$$

What can we say about the error now?

Changing Condition Numbers

Once we have a matrix A in a linear system $A\mathsf{x} = \mathsf{b}$, are we stuck with its condition number? Or could we improve it?

What is this called as a general concept?

In-Class Activity: Matrix Norms and Conditioning

In-class activity: Matrix Norms and Conditioning

Solving Systems: Triangular matrices

Solve

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Demo: Coding back-substitution

What about non-triangular matrices?

Gaussian Elimination

Demo: Vanilla Gaussian Elimination

What do we get by doing Gaussian Elimination?

How is that different from being upper triangular?

What if we do not just eliminate downward but also upward?

Elimination Matrices

What does this matrix do?

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ -\frac{1}{2} & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

About Elimination Matrices

Are elimination matrices invertible?

More on Elimination Matrices

Demo: Elimination matrices I

Idea: With enough elimination matrices, we should be able to get a matrix into row echelon form.

So what do we get from many combined elimination matrices like that?

Demo: Elimination Matrices II

Summary on Elimination Matrices

- El.matrices with off-diagonal entries in a single column just “merge” when multiplied by one another.
- El.matrices with off-diagonal entries in different columns merge when we multiply (left-column) * (right-column) but not the other way around.
- Inverse: Flip sign below diagonal

LU Factorization

Can build a *factorization* from elimination matrices. How?

Solving $Ax = b$

Does LU help solve $Ax = b$?

Demo: LU factorization

LU: Failure Cases?

Is LU/Gaussian Elimination bulletproof?

Saving the LU Factorization

What can be done to get something *like* an LU factorization?

Recap: Permuation Matrices

How do we capture 'row switches' in a factorization?

$$\underbrace{\begin{bmatrix}1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1\end{bmatrix}}_{P}\begin{bmatrix}A & A & A & A\\ B & B & B & B\\ C & C & C & C\\ D & D & D & D\end{bmatrix} = \begin{bmatrix}A & A & A & A\\ C & C & C & C\\ B & B & B & B\\ D & D & D & D\end{bmatrix}.$$

P is called a *permutation matrix*.
Q: What's P^{-1}?

Fixing nonexistence of LU

What does LU with permutations process look like?

Demo: LU with Partial Pivoting (Part I)

What about the L in LU?

Sort out what LU with pivoting looks like. Have: $M_3 P_3 M_2 P_2 M_1 P_1 A = U$.

Demo: LU with Partial Pivoting (Part II)

Computational Cost

What is the computational cost of multiplying two $n \times n$ matrices?

What is the computational cost of carrying out LU factorization on an $n \times n$ matrix?

Demo: Complexity of Mat-Mat multiplication and LU

More cost concerns

What's the cost of solving $A\mathrm{x} = \mathrm{b}$?

What's the cost of solving $A\mathrm{x} = \mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_n$?

What's the cost of finding A^{-1}?

Cost: Worrying about the Constant, BLAS

$O(n^3)$ really means

$$\alpha \cdot n^3 + \beta \cdot n^2 + \gamma \cdot n + \delta.$$

All the non-leading and constants terms swept under the rug. But: at least the leading constant ultimately matters.

Shrinking the constant: surprisingly hard (even for 'just' matmul)

Idea: Rely on library implementation: *BLAS* (Fortran)

Level 1	$\mathrm{z} = \alpha \mathrm{x} + \mathrm{y}$	vector-vector operations
		$O(n)$
		`?axpy`
Level 2	$\mathrm{z} = A\mathrm{x} + \mathrm{y}$	matrix-vector operations
		$O(n^2)$
		`?gemv`
Level 3	$C = AB + \beta C$	matrix-matrix operations
		$O(n^3)$
		`?gemm`, `?trsm`

Show (using perf): `numpy` matmul calls BLAS `dgemm`

LAPACK

LAPACK: Implements ‘higher-end’ things (such as LU) using BLAS
Special matrix formats can also help save const significantly, e.g.

- banded
- sparse
- symmetric
- triangular

Sample routine names:

- `dgesvd`, `zgesdd`
- `dgetrf`, `dgetrs`

LU on Blocks: The Schur Complement

Given a matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

can we do 'block LU' to get a *block triangular matrix*?

LU: Special cases

What happens if we feed a non-invertible matrix to LU?

What happens if we feed LU an $m \times n$ non-square matrices?

Round-off Error in LU

Consider factorization of $\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}$ where $\epsilon < \epsilon_{\text{mach}}$:

- Without pivoting: $L = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix}$, $U = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - 1/\epsilon \end{bmatrix}$
- Rounding: $\text{fl}(U)) = \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}$
- This leads to $L\,\text{fl}(U)) = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix}$, a backward error of $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

Permuting the rows of A in partial pivoting gives $PA = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix}$

- We now compute $L = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}$, $U = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix}$, so $\text{fl}(U) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$
- This leads to $L\,\text{fl}(U) = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 + \epsilon \end{bmatrix}$, a backward error of $\begin{bmatrix} 0 & 0 \\ 0 & \epsilon \end{bmatrix}$.

Changing matrices

Seen: LU cheap to re-solve if RHS changes. (Able to keep the expensive bit, the LU factorization) What if the *matrix* changes?

Demo: Sherman-Morrison

In-Class Activity: LU

In-class activity: LU and Cost

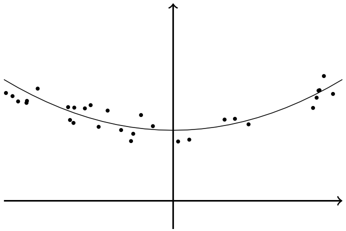
Outline

What about non-square systems?

Specifically, what about linear systems with ‘tall and skinny’ matrices? (A: $m \times n$ with $m > n$) (aka *overdetermined* linear systems)

Specifically, any hope that we will solve those exactly?

Example: Data Fitting



Have data: (x_i, y_i) and model:

$$y(x) = \alpha + \beta x + \gamma x^2$$

Find data that (best) fit model!

Data Fitting Continued

Rewriting Data Fitting

Rewrite in matrix form.

Least Squares: The Problem In Matrix Form

$$\|A\mathrm{x} - \mathrm{b}\|_2^2 \to \min!$$

is cumbersome to write.
Invent new notation, defined to be equivalent:

$$A\mathrm{x} \cong \mathrm{b}$$

NOTE:

- Data Fitting is *one example* where LSQ problems arise.
- Many other application lead to $A\mathrm{x} \cong \mathrm{b}$, with different matrices.

Data Fitting: Nonlinearity

Give an example of a nonlinear data fitting problem.

$$\begin{aligned}&\left|\exp(\alpha)+\beta x_1+\gamma x_1^2-y_1\right|^2\\ &\qquad +\cdots+\\ &\left|\exp(\alpha)+\beta x_n+\gamma x_n^2-y_n\right|^2 \quad\to\quad \text{min!}\end{aligned}$$

But that would be easy to remedy: Do linear least squares with $\exp(\alpha)$ as the unknown. More difficult:

$$\begin{aligned}&\left|\alpha+\exp(\beta x_1+\gamma x_1^2)-y_1\right|^2\\ &\qquad +\cdots+\\ &\left|\alpha+\exp(\beta x_n+\gamma x_n^2)-y_n\right|^2 \quad\to\quad \text{min!}\end{aligned}$$

Demo: Interactive Polynomial Fit

Properties of Least-Squares

Consider LSQ problem $A\mathrm{x} \cong \mathrm{b}$ and its associated *objective function* $\varphi(x) = \|\mathrm{b} - A\mathrm{x}\|_2^2$. Does this always have a solution?

Is it always unique?

Examine the objective function, find its minimum.

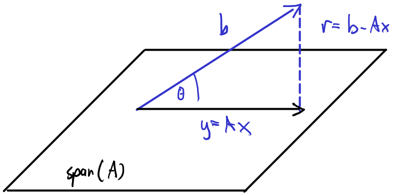
Least squares: Demos

Demo: Polynomial fitting with the normal equations

What's the shape of $A^T A$?

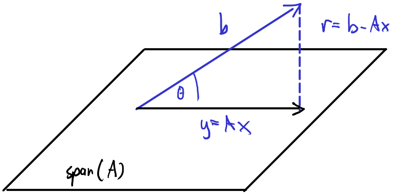
Demo: Issues with the normal equations

Least Squares, Viewed Geometrically



Why is r $\perp$ span(A) a good thing to require?

Least Squares, Viewed Geometrically (II)



Phrase the Pythagoras observation as an equation.

Write that with an orthogonal projection matrix P.

About Orthogonal Projectors

What is a *projector*?

What is an *orthogonal projector*?

How do I make one projecting onto $\operatorname{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_\ell\}$ for orthogonal $\mathbf{q}_i$?

Least Squares and Orthogonal Projection

Check that $P = A(A^T A)^{-1} A^T$ is an orthogonal projector onto colspan(A).

What assumptions do we need to define the P from the last question?

Pseudoinverse

What is the pseudoinverse of A?

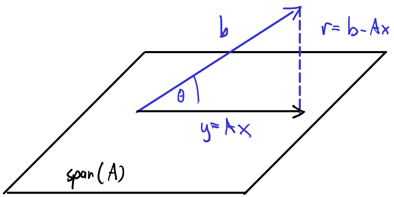
What can we say about the condition number in the case of a tall-and-skinny, full-rank matrix?

What does all this have to do with solving least squares problems?

In-Class Activity: Least Squares

In-class activity: Least Squares

Sensitivity and Conditioning of Least Squares



What values of θ are bad?

Sensitivity and Conditioning of Least Squares (II)

Any comments regarding dependencies?

What about changes in the matrix?

Recap: Orthogonal Matrices

What's an *orthogonal (=orthonormal) matrix*?

One that satisfies $Q^TQ = I$ and $QQ^T = I$.

How do orthogonal matrices interact with the 2-norm?

$$\|Q\mathrm{v}\|_2^2 = (Q\mathrm{v})^T(Q\mathrm{v}) = \mathrm{v}^TQ^TQ\mathrm{v} = \mathrm{v}^T\mathrm{v} = \|\mathrm{v}\|_2^2.$$

Transforming Least Squares to Upper Triangular

Suppose we have $A = QR$, with Q square and orthogonal, and R upper triangular. This is called a QR factorization.

How do we transform the least squares problem $A\mathrm{x} \cong \mathrm{b}$ to one with an upper triangular matrix?

Simpler Problems: Triangular

What do we win from transforming a least-squares system to upper triangular form?

How would we minimize the residual norm?

Computing QR

- Gram-Schmidt
- Householder Reflectors
- Givens Rotations

Demo: Gram-Schmidt–The Movie
Demo: Gram-Schmidt and Modified Gram-Schmidt
Demo: Keeping track of coefficients in Gram-Schmidt
Seen: Even modified Gram-Schmidt still unsatisfactory in finite precision arithmetic because of roundoff.

NOTE: Textbook makes further modification to 'modified' Gram-Schmidt:

- Orthogonalize *subsequent* rather than *preceding* vectors.
- Numerically: no difference, but sometimes algorithmically helpful.

Economical/Reduced QR

Is QR with square Q for $A \in \mathbb{R}^{m\times n}$ with $m > n$ efficient?

In-Class Activity: QR

In-class activity: QR

Householder Transformations

Find an *orthogonal* matrix Q to zero out the lower part of a vector a.

Householder Reflectors: Properties

Seen from picture (and easy to see with algebra):

$$H\mathrm{a} = \pm \|\mathrm{a}\|_2 \mathrm{e}_1.$$

Remarks:

- Q: What if we want to zero out only the $i + 1$th through nth entry?
  A: Use e_i above.
- A product $H_n \cdots H_1 A = R$ of Householders makes it easy (and quite efficient!) to build a QR factorization.
- It turns out $\mathrm{v}' = \mathrm{a} + \|\mathrm{a}\|_2 \mathrm{e}_1$ works out, too–just pick whichever one causes less cancellation.
- H is symmetric
- H is orthogonal

Demo: 3x3 Householder demo

Givens Rotations

If reflections work, can we make rotations work, too?

Demo: 3x3 Givens demo

Rank-Deficient Matrices and QR

What happens with QR for rank-deficient matrices?

Rank-Deficient Matrices and Least-Squares

What happens with Least Squares for rank-deficient matrices?

$$A\mathbf{x} \cong \mathbf{b}$$

- QR still finds a solution with minimal residual
- By QR it's easy to see that least squares with a short-and-fat matrix is equivalent to a rank-deficient one.
- But: No longer unique. $\mathbf{x} + \mathbf{n}$ for $\mathbf{n} \in N(A)$ has the same residual.
- In other words: Have more freedom
  Or: Can demand another condition, for example:
  - Minimize $\|\mathbf{b} - A\mathbf{x}\|_2^2$, *and*
  - minimize $\|\mathbf{x}\|_2^2$, simultaneously.
    Unfortunately, QR does not help much with that $\rightarrow$ Need better tool.

Singular Value Decomposition (SVD)

What is the *Singular Value Decomposition* of an $m \times n$ matrix?

SVD: What's this thing good for? (I)

SVD: What's this thing good for? (II)

- *Low-rank Approximation*

Theorem (Eckart-Young-Mirsky)

If $k < r = \operatorname{rank}(A)$ *and*

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

then

$$\min_{\operatorname{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

Demo: Image compression

SVD: What's this thing good for? (III)

- The minimum norm solution to $Ax \cong b$:

SVD: Minimum-Norm, Pseudoinverse

$y = \Sigma^+ U^T b$ is the minimum norm-solution to $\Sigma y \cong U^T b$.
Observe $\|x\|_2 = \|y\|_2$.

$$x = V\Sigma^+ U^T b$$

solves the minimum-norm least-squares problem.

Define $A^+ = V\Sigma^+ U^T$ and call it the pseudoinverse of A.
Coincides with prior definition in case of full rank.

In-Class Activity: Householder, Givens, SVD

In-class activity: Householder, Givens, SVD

Comparing the Methods

Methods to solve least squares with A an $m \times n$ matrix:

- Form: $A^T A$: $n^2 m/2$
  Solve with $A^T A$: $n^3/6$
- Solve with Householder: $mn^2 - n^3/3$
- If $m \approx n$, about the same
- If $m \gg n$: Householder QR requires about twice as much work as normal equations
- SVD: $mn^2 + n^3$ (with a large constant)

Demo: Relative cost of matrix factorizations

Outline

Eigenvalue Problems: Setup/Math Recap

A is an $n \times n$ matrix.

- $\mathsf{x} \neq 0$ is called an *eigenvector* of A if there exists a λ so that

$$A\mathsf{x} = \lambda \mathsf{x}.$$

- In that case, λ is called an *eigenvalue*.
- The set of all eigenvalues $\lambda(A)$ is called the *spectrum*.
- The *spectral radius* is the magnitude of the biggest eigenvalue:

$$\rho(A) = \max\{|\lambda| : \lambda(A)\}$$

Finding Eigenvalues

How do you find eigenvalues?

$$A\mathrm{x} = \lambda \mathrm{x} \Leftrightarrow (A - \lambda I)\mathrm{x} = 0$$
$$\Leftrightarrow A - \lambda I \text{ singular} \Leftrightarrow \det(A - \lambda I) = 0$$

$\det(A - \lambda I)$ is called the *characteristic polynomial*, which has degree n, and therefore n (potentially complex) roots.

Does that help algorithmically? Abel-Ruffini theorem: for $n \geqslant 5$ is no general formula for roots of polynomial. IOW: no.

- For LU and QR, we obtain *exact* answers (except rounding).
- For eigenvalue problems: not possible—must *approximate*.

Demo: Rounding in characteristic polynomial using SymPy

Multiplicity

What is the *multiplicity* of an eigenvalue?

Actually, there are two notions called multiplicity:

- *Algebraic Multiplicity*: multiplicity of the root of the characteristic polynomial
- *Geometric Multiplicity*: #of lin. indep. eigenvectors

In general: AM $\geqslant$ GM.
If AM $>$ GM, the matrix is called *defective*.

An Example

Give characteristic polynomial, eigenvalues, eigenvectors of

$$\begin{bmatrix} 1 & 1 \\ & 1 \end{bmatrix}.$$

Diagonalizability

When is a matrix called *diagonalizable*?

Similar Matrices

Related definition: Two matrices A and B are called similar if there exists an invertible matrix X so that $A = XBX^{-1}$.

In that sense: "Diagonalizable" = "Similar to a diagonal matrix".

Observe: Similar A and B have same eigenvalues. (Why?)

Eigenvalue Transformations (I)

What do the following transformations of the eigenvalue problem $Ax = \lambda x$ do?

Shift. $A \to A - \sigma I$

Inversion. $A \to A^{-1}$

Power. $A \to A^k$

Eigenvalue Transformations (II)

Polynomial $A \rightarrow aA^2 + bA + cI$

Similarity $T^{-1}AT$ with T invertible

Sensitivity (I)

Assume A not defective. Suppose $X^{-1}AX = D$. Perturb $A \to A + E$. What happens to the eigenvalues?

Sensitivity (II)

$X^{-1}(A+E)X = D + F$. Have $\left\|(\mu I - D)^{-1}\right\|^{-1} \leqslant \|F\|$.

Demo: Bauer-Fike Eigenvalue Sensitivity Bound

Power Iteration

What are the eigenvalues of A^{1000}?

Assume $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$ with eigenvectors $\mathrm{x}_1, \ldots, \mathrm{x}_n$.
Further assume $\|\mathrm{x}_i\| = 1$.

Power Iteration: Issues?

What could go wrong with Power Iteration?

What about Eigenvalues?

Power Iteration generates eigenvectors. What if we would like to know eigenvalues?

Convergence of Power Iteration

What can you say about the convergence of the power method?
Say $\mathsf{v}_1^{(k)}$ is the kth estimate of the eigenvector x_1, and

$$e_k = \left\| \mathsf{x}_1 - \mathsf{v}_1^{(k)} \right\| .$$

Rayleigh Quotient Iteration

Describe *inverse iteration*.

Describe *Rayleigh Quotient Iteration*.

In-Class Activity: Eigenvalues

In-class activity: Eigenvalues

Schur form

Show: Every matrix is orthonormally similar to an upper triangular matrix, i.e. $A = QUQ^T$. This is called the Schur form or Schur factorization.

Schur Form: Comments, Eigenvalues, Eigenvectors

$A = QUQ^T$. For complex λ:

- Either complex matrices, or
- 2×2 blocks on diag.

If we had a Schur form of A, how can we find the eigenvalues?

And the eigenvectors?

Computing Multiple Eigenvalues

All Power Iteration Methods compute one eigenvalue at a time.
What if I want *all* eigenvalues?

Simultaneous Iteration

What happens if we carry out power iteration on multiple vectors simultaneously?

Orthogonal Iteration

Toward the QR Algorithm

Demo: Orthogonal Iteration

QR Iteration/QR Algorithm

QR Iteration: Incorporating a Shift

How can we accelerate convergence of QR iteration using shifts?

QR Iteration: Computational Expense

A full QR factorization at each iteration costs $O(n^3)$–can we make that cheaper?

Demo: Householder Similarity Transforms

QR/Hessenberg: Overall procedure

Overall procedure:

1. Reduce matrix to Hessenberg form
2. Apply QR iteration using Givens QR to obtain Schur form

For symmetric matrices:

- Use Householders to attain tridiagonal form
- Use QR iteration with Givens to attain diagonal form

Krylov space methods: Intro

What subspaces can we use to look for eigenvectors?

Krylov for Matrix Factorization

What matrix factorization is obtained through Krylov space methods?

Conditioning in Krylov Space Methods/Arnoldi Iteration (I)

What is a problem with Krylov space methods? How can we fix it?

Conditioning in Krylov Space Methods/Arnoldi Iteration (II)

Demo: Arnoldi Iteration (Part 1)

Krylov: What about eigenvalues?

How can we use Arnoldi/Lanczos to compute eigenvalues?

Demo: Arnoldi Iteration (Part 2)

Computing the SVD (Kiddy Version)

How can I compute an SVD of a matrix A?

Demo: Computing the SVD

Outline

Solving Nonlinear Equations

What is the goal here?

Showing Existence

How can we show existence of a root?

Sensitivity and Multiplicity

What is the sensitivity/conditioning of root finding?

What are multiple roots?

How do multiple roots interact with conditioning?

In-Class Activity: Krylov and Nonlinear Equations

In-class activity: Krylov and Nonlinear Equations

Rates of Convergence

What is *linear convergence*? *quadratic convergence*?

About Convergence Rates

Demo: Rates of Convergence

Characterize linear, quadratic convergence in terms of the 'number of accurate digits'.

Stopping Criteria

Comment on the 'foolproof-ness' of these stopping criteria:

1. $|f(x)| < \varepsilon$ ('residual is small')
2. $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$
3. $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \,/\, \|\mathbf{x}_k\| < \varepsilon$

Bisection Method

Demo: Bisection Method

What's the rate of convergence? What's the constant?

Fixed Point Iteration

$$\begin{aligned} x_0 &= \langle\text{starting guess}\rangle \\ x_{k+1} &= g(x_k) \end{aligned}$$

Demo: Fixed point iteration

When does fixed point iteration converge? Assume g is smooth.

Fixed Point Iteration: Convergence cont'd.

Error in FPI: $e_{k+1} = x_{k+1} - x^* = g(x_k) - g(x^*)$

Newton's Method

Derive Newton's method.

Convergence and Properties of Newton

What's the rate of convergence of Newton's method?

Drawbacks of Newton?

Demo: Newton's method

Demo: Convergence of Newton's Method

Secant Method

What would Newton without the use of the derivative look like?

Convergence of Properties of Secant

Rate of convergence (not shown) is $\left(1+\sqrt{5}\right)/2 \approx 1.618$.

Drawbacks of Secant?

Demo: Secant Method
Demo: Convergence of the Secant Method

Secant (and similar methods) are called Quasi-Newton Methods.

Root Finding with Interpolants

Secant method uses a linear interpolant based on points $f(x_k)$, $f(x_{k-1})$, could use more points and higher-order interpolant:

What about existence of roots in that case?

Achieving Global Convergence

The linear approximations in Newton and Secant are only good locally. How could we use that?

In-Class Activity: Nonlinear Equations

In-class activity: Nonlinear Equations

Fixed Point Iteration

$$\begin{aligned} x_0 &= \langle\text{starting guess}\rangle \\ x_{k+1} &= g(x_k) \end{aligned}$$

When does this converge?

Newton's Method

What does Newton's method look like in *n* dimensions?

Downsides of *n*-dim. Newton?

Demo: Newton's method in n dimensions

Secant in n dimensions?

What would the secant method look like in n dimensions?

Outline

Optimization: Problem Statement

Have: Objective function $f : \mathbb{R}^n \to \mathbb{R}$
Want: Minimizer $\mathrm{x}^* \in \mathbb{R}^n$ so that

$$f(\mathrm{x}^*) = \min_{\mathrm{x}} f(\mathrm{x}) \quad \text{subject to} \quad \mathrm{g}(\mathrm{x}) = 0 \quad \text{and} \quad \mathrm{h}(\mathrm{x}) \leqslant 0.$$

- $\mathrm{g}(\mathrm{x}) = 0$ and $\mathrm{h}(\mathrm{x}) \leqslant 0$ are called constraints.
  They define the set of feasible points $\mathrm{x} \in S \subseteq \mathbb{R}^n$.
- If g or h are present, this is constrained optimization.
  Otherwise unconstrained optimization.
- If f, g, h are *linear*, this is called linear programming.
  Otherwise nonlinear programming.

Optimization: Observations

Q: What if we are looking for a *maximizer* not a minimizer?
Give some examples:

What about multiple objectives?

Existence/Uniqueness

Terminology: global minimum / local minimum

Under what conditions on f can we say something about existence/uniqueness?

If $f : S \to \mathbb{R}$ is continuous on a closed and bounded set $S \subseteq \mathbb{R}^n$, then

$f : S \to \mathbb{R}$ is called *coercive* on $S \subseteq \mathbb{R}^n$ (which must be unbounded) if

If f is coercive, ...

Convexity

$S \subseteq \mathbb{R}^n$ is called convex if for all $x, y \in S$ and all $0 \leqslant \alpha \leqslant 1$

$f : S \to \mathbb{R}$ is called convex on $S \subseteq \mathbb{R}^n$ if for $\setminus$ $x, y \in S$ and all $0 \leqslant \alpha \leqslant 1$

Q: Give an example of a convex, but not strictly convex function.

Convexity: Consequences

If f is convex, . . .

If f is strictly convex, . . .

Optimality Conditions

If we have found a candidate x^* for a minimum, how do we know it actually is one? Assume f is smooth, i.e. has all needed derivatives.

Optimization: Observations

Q: Come up with a hypothetical approach for finding minima.

Q: Is the Hessian symmetric?

Q: How can we practically test for positive definiteness?

In-Class Activity: Optimization Theory

In-class activity: Optimization Theory

Sensitivity and Conditioning (1D)

How does optimization react to a slight perturbation of the minimum?

Sensitivity and Conditioning (nD)

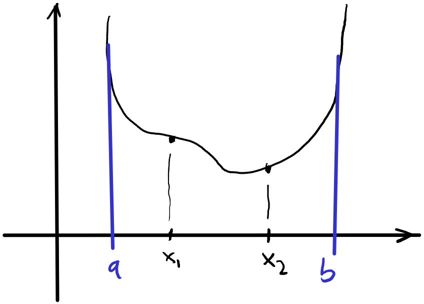
How does optimization react to a slight perturbation of the minimum?

Unimodality

Would like a method like bisection, but for optimization.
In general: No invariant that can be preserved.
Need *extra assumption.*

Golden Section Search

Suppose we have an interval with f unimodal:



Would like to maintain unimodality.

Golden Section Search: Efficiency

Where to put x_1, x_2?

Convergence rate?

Demo: Golden Section Proportions

Newton's Method

Reuse the Taylor approximation idea, but for optimization.

Demo: Newton's Method in 1D

In-Class Activity: Optimization Methods

In-class activity: Optimization Methods

Steepest Descent

Given a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ at a point x, which way is down?

Demo: Steepest Descent

Steepest Descent: Convergence

Consider quadratic model problem:

$$f(\mathrm{x}) = \frac{1}{2}\mathrm{x}^T A\mathrm{x} + \mathrm{c}^T\mathrm{x}$$

where A is SPD. (A good model of f near a minimum.)

Define error $\mathrm{e}_k = \mathrm{x}_k - \mathrm{x}^*$. Then

$$||\mathrm{e}_{k+1}||_A = \sqrt{\mathrm{e}_{k+1}^T A\mathrm{e}_{k+1}} = \frac{\sigma_{\mathrm{max}}(A) - \sigma_{\mathrm{min}}(A)}{\sigma_{\mathrm{max}}(A) + \sigma_{\mathrm{min}}(A)}||\mathrm{e}_k||_A$$

$\rightarrow$ confirms linear convergence.

Convergence constant related to conditioning:

$$\frac{\sigma_{\mathrm{max}}(A) - \sigma_{\mathrm{min}}(A)}{\sigma_{\mathrm{max}}(A) + \sigma_{\mathrm{min}}(A)} = \frac{\kappa(A) - 1}{\kappa(A) + 1}.$$

Hacking Steepest Descent for Better Convergence

Extrapolation methods: Look back a step, maintain '*momentum*'.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

Heavy ball method: constant $\alpha_k = \alpha$ and $\beta_k = \beta$. Gives:

$$||e_{k+1}||_A = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} ||e_k||_A$$

Conjugate gradient method:

$$(\alpha_k, \beta_k) = \text{argmin}_{\alpha_k, \beta_k} \left[f\left(x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1}) \right) \right]$$

- Will see in more detail later (for solving linear systems)
- Provably optimal first-order method for the quadratic model problem
- Turns out to be closely related to Lanczos (A-orthogonal search directions)

Nelder-Mead Method

Idea:

Demo: Nelder-Mead Method

Newton's method (n D)

What does Newton's method look like in n dimensions?

Newton's method (n D): Observations

Drawbacks?

Demo: Newton's method in n dimensions

Quasi-Newton Methods

Secant/Broyden-type ideas carry over to optimization. How?

BFGS: Secant-type method, similar to Broyden:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$$

where

- $s_k = x_{k+1} - x_k$
- $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

Nonlinear Least Squares: Setup

What if the f to be minimized is actually a 2-norm?

$$f(\mathrm{x}) = \|\mathrm{r}(\mathrm{x})\|_2, \qquad \mathrm{r}(\mathrm{x}) = \mathrm{y} - \mathrm{a}(\mathrm{x})$$

Gauss-Newton

For brevity: $J := J_\mathrm{r}(\mathrm{x})$.

Gauss-Newton: Observations?

Demo: Gauss-Newton

Observations?

Levenberg-Marquardt

If Gauss-Newton on its own is poorly, conditioned, can try Levenberg-Marquardt:

Constrained Optimization: Problem Setup

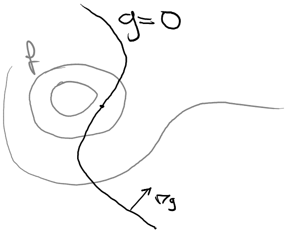
Want x^* so that

$$f(x^*) = \min_x f(x) \quad \text{subject to} \quad g(x) = 0$$

No inequality constraints just yet. This is *equality-constrained optimization*. Develop a necessary condition for a minimum.

Constrained Optimization: Necessary Condition

Lagrange Multipliers



Seen: Need $-\nabla f(\mathrm{x}) = J_{\mathrm{g}}^T \lambda$ at the (constrained) optimum.

Idea: Turn constrained optimization problem for x into an *unconstrained* optimization problem for (x, λ). How?

Lagrange Multipliers: Development

$$\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x}).$$

Demo: Sequential Quadratic Programming

Inequality-Constrained Optimization

Want x^* so that

$$f(x^*) = \min_x f(x) \quad \text{subject to} \quad g(x) = 0 \quad \text{and} \quad h(x) \leqslant 0$$

This is *inequality-constrained optimization*. Develop a necessary condition for a minimum.

Inequality-Constrained Optimization (cont'd)

Develop a set of necessary conditions for a minimum.

Outline

Interpolation: Setup

Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

How is this not the same as function fitting? (from least squares)

Interpolation: Setup (II)

Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

Does this problem have a unique answer?

Interpolation: Importance

Why is interpolation important?

Making the Interpolation Problem Unique

Existence/Sensitivity

Solution to the interpolation problem: Existence? Uniqueness?

Sensitivity?

Modes and Nodes (aka Functions and Points)

Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

- Monomials $1, x, x^2, x^3, x^4, \dots$
- Functions that make $V = I \rightarrow$ 'Lagrange basis'
- Functions that make V triangular $\rightarrow$ 'Newton basis'
- *Splines* (piecewise polynomials)
- *Orthogonal polynomials*
- Sines and cosines
- 'Bumps' (*'Radial Basis Functions'*)

Ideas for points:

- Equispaced
- *'Edge-Clustered'* (so-called Chebyshev/Gauss/... nodes)

Specific issues:

- Why *not* monomials on equispaced points? **Demo:** Monomial interpolation
- Why not equispaced? **Demo:** Choice of Nodes for Polynomial Interpolation

Lagrange Interpolation

Find a basis so that $V = I$, i.e.

$$\varphi_j(x_i) = \begin{cases} 1 & i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Lagrange Polynomials: General Form

$$\varphi_j(x) = \frac{\prod_{k=1,k\neq j}^{m}(x - x_k)}{\prod_{k=1,k\neq j}^{m}(x_j - x_k)}$$

Newton Interpolation

Find a basis so that V is triangular.

Why not Lagrange/Newton?

Better conditioning: Orthogonal polynomials

What caused monomials to have a terribly conditioned Vandermonde?

What's a way to make sure two vectors are *not* like that?

But polynomials are functions!

Constructing Orthogonal Polynomials

How can we find an orthogonal basis?

Demo: Orthogonal Polynomials — Obtained: Legendre polynomials.
But how can I practically compute the Legendre polynomials?

Chebyshev Polynomials: Definitions

Three equivalent definitions:

- Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$. What is that weight?

  (Like for Legendre, you won't exactly get the standard normalization if you do this.)
- $T_k(x) = \cos(k \cos^{-1}(x))$
- $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ plus $T_0 = 1$, $T_0 = x$

Demo: Chebyshev Interpolation (Part 1)

Chebyshev Interpolation

What is the Vandermonde matrix for Chebyshev polynomials?

Chebyshev Nodes

Might also consider roots (instead of extrema) of T_k:

$$x_i = \cos\left(\frac{2i-1}{2k}\pi\right) \quad (i = 1\ldots,k).$$

Vandermonde for these (with T_k) can be applied in $O(N \log N)$ time, too.
It turns out that we were still looking for a good set of interpolation nodes.
We came up with the criterion that the nodes should bunch towards the ends. Do these do that?

Demo: Chebyshev Interpolation (Part 2)

Chebyshev Interpolation: Summary

- Chebyshev interpolation is fast and works extremely well
- `http://www.chebfun.org/` and: ATAP
- In 1D, they're a very good answer to the interpolation question
- But sometimes a piecewise approximation (with a specifiable level of smoothness) is more suited to the application

In-Class Activity: Interpolation

In-class activity: Interpolation

Interpolation Error

If f is n times continuously differentiable on a closed interval I and $p_{n-1}(x)$ is a polynomial of degree at most n that interpolates f at n distinct points $\{x_i\}$ $(i = 1, ..., n)$ in that interval, then for each x in the interval there exists ξ in that interval such that

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\xi)}{n!}(x - x_1)(x - x_2)\cdots(x - x_n).$$

Interpolation Error: Proof cont'd

$$Y(t) = R(t) - \frac{R(x)}{W(x)} W(t) \quad \text{where} \quad W(t) = \prod_{i=1}^{n} (t - x_i)$$

Error Result: Connection to Chebyshev

What is the connection between the error result and Chebyshev interpolation?

Error Result: Simplified From

Boil the error result down to a simpler form.

Demo: Interpolation Error

Going piecewise: Simplest Case

Construct a piecweise linear interpolant at four points.

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
	$f_1 = a_1 x + b_1$		$f_2 = a_2 x + b_2$		$f_3 = a_3 x + b_3$	
	2 unk.		2 unk.		2 unk.	
	$f_1(x_0) = y_0$		$f_2(x_1) = y_1$		$f_3(x_2) = y_2$	
	$f_1(x_1) = y_1$		$f_2(x_2) = y_2$		$f_3(x_3) = y_3$	
	2 eqn.		2 eqn.		2 eqn.	

Why three intervals?

Piecewise Cubic ('Splines')

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
	f_1		f_2		f_3	
	$a_1x^3 + b_1x^2 + c_1x + d_1$		$a_2x^3 + b_2x^2 + c_2x + d_2$		$a_3x^3 + b_3x^2 + c_3x + d_3$	

Piecewise Cubic (‘Splines’): Accounting

x_0, y_0		x_1, y_1		x_2, y_2		x_3, y_3
	f_1		f_2		f_3	
	$a_1x^3 + b_1x^2 + c_1x + d_1$		$a_2x^3 + b_2x^2 + c_2x + d_2$		$a_3x^3 + b_3x^2 + c_3x + d_3$	

Outline

Numerical Integration: About the Problem

What is numerical integration? (Or quadrature?)

What about existence and uniqueness?

Conditioning

Derive the (absolute) condition number for numerical integration.

Interpolatory Quadrature

Design a quadrature method based on interpolation.

Interpolatory Quadrature: Examples

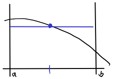
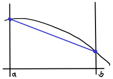
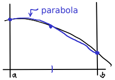
Interpolatory Quadrature: Computing Weights

How do the weights in interpolatory quadrature get computed?

Demo: Newton-Cotes weight finder

Examples and Exactness

To what polynomial degree are the following rules exact?

Midpoint rule	$(b-a)f\left(\frac{a+b}{2}\right)$	
Trapezoidal rule	$\frac{b-a}{2}(f(a)+f(b))$	
Simpson's rule	$\frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)$	

Interpolatory Quadrature: Accuracy

Let p_{n-1} be an interpolant of f at nodes $x_1, \ldots, x_n$ (of degree $n-1$)
Recall

$$\sum_i \omega_i f(x_i) = \int_a^b p_{n-1}(x)\mathrm{d}x.$$

What can you say about the accuracy of the method?

Quadrature: Overview of Rules

	n	Deg.	Ex.Int.Deg. (w/odd)	Intp.Ord.	Quad.Ord. (regular)	Quad.Ord. (w/odd)
		$n-1$	$(n-1)+1_{\text{odd}}$	n	$n+1$	$(n+1)+1_{\text{odd}}$
Midp.	1	0	1	1	2	3
Trapz.	2	1	1	2	3	3
Simps.	3	2	3	3	4	5
—	4	3	3	4	5	5

- n: number of points
- "Deg.": Degree of polynomial used in interpolation ($= n-1$)
- "Ex.Int.Deg.": Polynomials of up to (and including) this degree *actually* get integrated exactly. (including the odd-order bump)
- "Intp.Ord.": Order of Accuracy of Interpolation: $O(h^n)$
- "Quad.Ord. (regular)": Order of accuracy for quadrature predicted by the error result above: $O(h^{n+1})$
- "Quad.Ord. (w/odd):" Actual order of accuracy for quadrature given 'bonus' degrees for rules with odd point count

Observation: Quadrature gets (at least) 'one order higher' than interpolation–even more for odd-order rules. (i.e. more accurate)

Interpolatory Quadrature: Stability

Let p_n be an interpolant of f at nodes $x_1, \ldots, x_n$ (of degree $n-1$)
Recall

$$\sum_i \omega_i f(x_i) = \int_a^b p_n(x) \mathrm{d}x$$

What can you say about the stability of this method?

About Newton-Cotes

What's not to like about Newton-Cotes quadrature?

Gaussian Quadrature

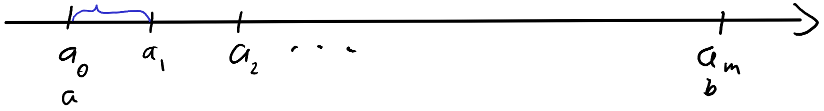
So far: nodes chosen from outside.
Can we gain something if we let the quadrature rule choose the nodes, too? Hope: More design freedom $\rightarrow$ Exact to higher degree.

Demo: Gaussian quadrature weight finder

Composite Quadrature

High-order polynomial interpolation requires a high degree of smoothness of the function.
Idea: Stitch together multiple lower-order quadrature rules to alleviate smoothness requirement.

e.g. trapezoidal



Error in Composite Quadrature

What can we say about the error in the case of composite quadrature?

Composite Quadrature: Notes

Observation: Composite quadrature loses an order compared to non-composite.

Idea: If we can estimate errors on each subinterval, we can shrink (e.g. by splitting in half) only those contributing the most to the error. (adaptivity, $\rightarrow$ hw)

Taking Derivatives Numerically

Why *shouldn't* you take derivatives numerically?

Demo: Taking Derivatives with Vandermonde Matrices

Finite Differences

More Finite Difference Rules

Similarly:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

(Centered differences)

Can also take higher order derivatives:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

Can find these by trying to match Taylor terms.
Alternative: Use linear algebra with interpolate-then-differentiate to find FD formulas.
Demo: Finite Differences vs Noise
Demo: Floating point vs Finite Differences

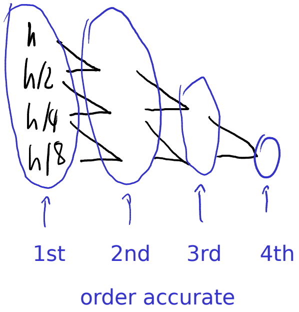
Richardson Extrapolation

If we have two estimates of something, can we get a third that's more accurate? Suppose we have an approximation $F = \tilde{F}(h) + O(h^p)$ and we know $\tilde{F}(h_1)$ *and* $\tilde{F}(h_2)$.

Richardson Extrapolation: Observations, Romberg Integration

Important observation: Never needed to know a.

Idea: Can repeat this for even higher accuracy.



Carrying out this process for quadrature is called Romberg integration.

Demo: Richardson with Finite Differences

In-Class Activity: Differentiation and Quadrature

In-class activity: Differentiation and Quadrature

Outline

What can we solve already?

- Linear Systems: yes
- Nonlinear systems: yes
- Systems with derivatives: no

Some Applications

IVPs	BVPs
▶ Population dynamics $y_1' = y_1(\alpha_1 - \beta_1 y_2)$ (prey) $y_2' = y_2(-\alpha_2 + \beta_2 y_1)$ (predator) ▶ chemical reactions ▶ equations of motion	▶ bridge load ▶ pollutant concentration (steady state) ▶ temperature (steady state)

Initial Value Problems: Problem Statement

Want: Function $\mathbf{y} : [0, T] \to \mathbb{R}^n$ so that

- $\mathbf{y}^{(k)}(t) = \mathbf{f}(t, \mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots, \mathbf{y}^{(k-1)})$ (*explicit*)

  or

- $\mathbf{f}(t, \mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots, \mathbf{y}^{(k)}) = 0$ (*implicit*)

are called explicit/implicit *kth-order ordinary differential equations* (*ODEs*). Give a simple example.

Not uniquely solvable on its own. What else is needed?

Reducing ODEs to First-Order Form

A kth order ODE can always be reduced to first order. Do this in this example:

$$y''(t) = f(y)$$

Properties of ODEs

What is a linear ODE?

What is a linear and homogeneous ODE?

What is a constant-coefficient ODE?

Properties of ODEs (II)

What is an autonomous ODE?

Existence and Uniqueness

Consider the perturbed problem

$$\begin{cases} y'(t) = f(y) \\ y(t_0) = y_0 \end{cases} \qquad \begin{cases} \widehat{y}'(t) = f(\widehat{y}) \\ \widehat{y}(t_0) = \widehat{y}_0 \end{cases}$$

Then if f is *Lipschitz continuous* (has ‘bounded slope’), i.e.

$$\|f(y) - f(\widehat{y})\| \leqslant L \|y - \widehat{y}\|$$

(where L is called the *Lipschitz constant*), then...

What does this mean for uniqueness?

Conditioning

Unfortunate terminology accident: “Stability” in ODE-speak
To adapt to conventional terminology, we will use ‘Stability’ for

- the conditioning of the IVP, *and*
- the stability of the methods we cook up.

Some terminology:

An ODE is stable if and only if. . .

An ODE is asymptotically stable if and only if

Example I: Scalar, Constant-Coefficient

$$\begin{cases} y'(t) = \lambda y \\ y(0) = y_0 \end{cases} \qquad \text{where} \quad \lambda = a + ib$$

Solution?

When is this stable?

Example II: Constant-Coefficient System

$$\begin{cases} \mathrm{y}'(t) = A\mathrm{y}(t) \\ \mathrm{y}(t_0) = \mathrm{y}_0 \end{cases}$$

Assume $V^{-1}AV = D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ diagonal.

How do we find a solution?

When is this stable?

Euler's Method

Discretize the IVP

$$\begin{cases} y'(t) = f(y) \\ y(t_0) = y_0 \end{cases}$$

- Discrete times: $t_1, t_2, \ldots$, with $t_{i+1} = t_i + h$
- Discrete function values: $y_k \approx y(t_k)$.

Euler's method: Forward and Backward

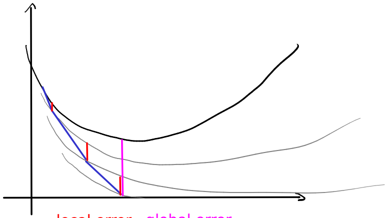
$$\mathrm{y}(t) = \mathrm{y}_0 + \int_{t_0}^{t} \mathrm{f}(\mathrm{y}(\tau))\mathrm{d}\tau,$$

Use 'left rectangle rule' on integral:

Use 'right rectangle rule' on integral:

Demo: Forward Euler stability

Global and Local Error



Let $u_k(t)$ be the function that solves the ODE with the initial condition $u_k(t_k) = y_k$.

Define the local error at step k as...

Define the global error at step k as...

About Local and Global Error

Is global error $= \sum$local errors?

A time integrator is said to be *accurate of order p* if…

ODE IVP Solvers: Order of Accuracy

A time integrator is said to be *accurate of order p* if $\ell_k = O(h^{p+1})$

This requirement is one order higher than one might expect–why?

Stability of a Method

Find out when forward Euler is stable when applied to $y'(t) = \lambda y(t)$.

Stability: Systems

What about stability for systems, i.e.

$$\mathrm{y}'(t) = A\mathrm{y}(t)?$$

Stability: Nonlinear ODEs

What about stability for nonlinear systems, i.e.

$$\mathrm{y}'(t) = \mathrm{f}(\mathrm{y}(t))?$$

Stability for Backward Euler

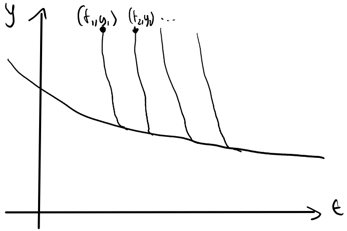
Find out when backward Euler is stable when applied to $y'(t) = \lambda y(t)$.

Demo: Backward Euler stability

Stiff ODEs: Demo

Demo: Stiffness

‘Stiff’ ODEs



- Stiff problems have *multiple time scales*.
  (In the example above: Fast decay, slow evolution.)
- In the case of a stable ODE system

$$\mathrm{y}'(t) = \mathrm{f}(\mathrm{y}(t)),$$

stiffness can arise if J_f has eigenvalues of very different magnitude.

Stiffness: Observations

Why not just ‘small’ or ‘large’ magnitude?

What is the problem with applying explicit methods to stiff problems?

Stiffness vs. Methods

Phrase this as a conflict between accuracy and stability.

Can an implicit method take arbitrarily large time steps?

Predictor-Corrector Methods

Idea: Obtain intermediate result, improve it (with same or different method).

For example:

1. *Predict* with forward Euler: $\tilde{y}_{k+1} = y_k + hf(y_k)$
2. *Correct* with the trapezoidal rule: $y_{k+1} = y_k + \frac{h}{2}(f(y_k) + f(\tilde{y}_{k+1}))$.

This is called Heun's method.

Runge-Kutta/‘Single-step’/‘Multi-Stage’ Methods

Idea: Compute intermediate ‘stage values’:

$$\begin{aligned} r_1 &= f(t_k + c_1 h, y_k + (a_{11} \cdot r_1 + \cdots + a_{1s} \cdot r_s) h) \\ \vdots &\quad \vdots \\ r_s &= f(t_k + c_s h, y_k + (a_{s1} \cdot r_1 + \cdots + a_{ss} \cdot r_s) h) \end{aligned}$$

Then compute the new state from those:

$$y_{k+1} = y_k + (b_1 \cdot r_1 + \cdots + b_s \cdot r_s) h$$

Can summarize in a *Butcher tableau*:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

Runge-Kutta: Properties

When is an RK method explicit?

When is it implicit?

When is it *diagonally implicit*? (And what does that mean?)

Heun and Butcher

Stuff Heun's method into a Butcher tableau:

1. $\tilde{y}_{k+1} = y_k + hf(y_k)$
2. $y_{k+1} = y_k + \frac{h}{2}(f(y_k) + f(\tilde{y}_{k+1}))$.

What is RK4?

Demo: Dissipation in Runge-Kutta Methods

Multi-step/Single-stage/Adams Methods/Backward Differencing Formulas (BDFs)

Idea: Instead of computing stage values, use *history* (of either values of f or y–or both):

$$y_{k+1} = \sum_{i=1}^{M} \alpha_i y_{k+1-i} + h \sum_{i=1}^{N} \beta_i f(y_{k+1-i})$$

Extensions to implicit possible.
Method relies on existence of history. What if there isn't any? (Such as at the start of time integration?)

Stability Regions

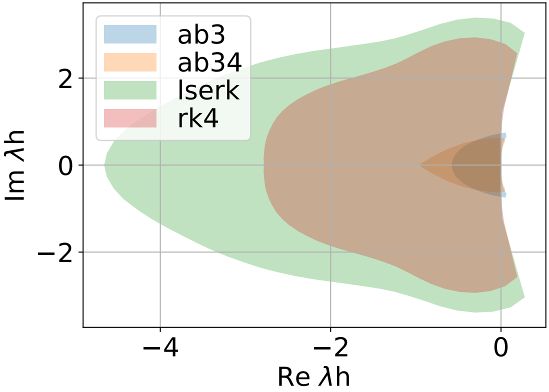
Why does the idea of stability regions still apply to more complex time integrators (e.g. RK?)

Demo: Stability regions

More Advanced Methods

Discuss:

- What is a good cost metric for time integrators?
- AB3 vs RK4
- Runge-Kutta-Chebyshev
- LSERK and AB34
- IMEX and multi-rate
- Parallel-in-time ("Parareal")



In-Class Activity: Initial Value Problems

In-class activity: Initial Value Problems

Outline

BVP Problem Setup: Second Order

Example: Second-order linear ODE

$$u''(x) + p(x)u'(x) + q(x)u(x) = r(x)$$

with *boundary conditions ('BCs')* at a:

- *Dirichlet* $u(a) = u_a$
- or *Neumann* $u'(a) = v_a$
- or *Robin* $\alpha u(a) + \beta u'(a) = w_a$

and the same choices for the BC at b.

Note: BVPs in time are rare in applications, hence x (not t) is typically used for the independent variable.

BVP Problem Setup: General Case

ODE:

$$\mathrm{y}'(x) = \mathrm{f}(\mathrm{y}(x)) \quad \mathrm{f} : \mathbb{R}^n \to \mathbb{R}^n$$

BCs:

$$\mathrm{g}(\mathrm{y}(a), \mathrm{y}(b)) = 0 \quad \mathrm{g} : \mathbb{R}^{2n} \to \mathbb{R}^n$$

(Recall the rewriting procedure to first-order for any-order ODEs.)

Does a first-order, scalar BVP make sense?

Example: Linear BCs

$$B_a \mathrm{y}(a) + B_b \mathrm{y}(b) = \mathrm{c}$$

Is this Dirichlet/Neumann/...?

Does a solution even exist? How sensitive are they?

General case is harder than root finding, and we couldn't say much there.
$\rightarrow$ Only consider linear BVP.

$$(*) \begin{cases} \mathrm{y}'(x) = A(x)\mathrm{y}(x) + \mathrm{b}(x) \\ B_a\mathrm{y}(a) + B_b\mathrm{y}(b) = \mathrm{c} \end{cases}$$

To solve that, consider *homogeneous IVP*

$$\mathrm{y}_i'(x) = A(x)\mathrm{y}_i(x)$$

with initial condition

$$\mathrm{y}_i(a) = \mathrm{e}_i.$$

Note: $\mathrm{y} \neq \mathrm{y_i}$. e_i is the *i*th unit vector. With that, build the fundamental solution matrix

$$Y(x) = \begin{bmatrix} | & & | \\ \mathrm{y}_1 & \cdots & \mathrm{y}_n \\ | & & | \end{bmatrix}$$

ODE Systems: Existence

Let

$$Q := B_a Y(a) + B_b Y(b)$$

Then $(*)$ has a unique solution if and only if Q is invertible. Solve to find coefficients:

$$Q\alpha = \mathrm{c}$$

Then $Y(x)\alpha$ solves $(*)$ with $\mathrm{b}(x) = 0$.

Define $\Phi(x) := Y(x)Q^{-1}$. So $\Phi(x)\mathrm{c}$ solves $(*)$ with $\mathrm{b}(x) = 0$.
Define *Green's function*

$$G(x,y) := \begin{cases} \Phi(x)B_a\Phi(a)\Phi^{-1}(y) & y \leqslant x, \\ -\Phi(x)B_b\Phi(b)\Phi^{-1}(y) & y > x. \end{cases}$$

Then

$$\mathrm{y}(x) = \Phi(x)\mathrm{c} + \int_a^b G(x,y)\mathrm{b}(y)\mathrm{d}y.$$

Can verify that this solves $(*)$ by plug'n'chug.

ODE Systems: Conditioning

For perturbed problem with $\mathrm{b}(x) + \Delta\mathrm{b}(x)$ and $\mathrm{c} + \Delta\mathrm{c}$:

$$\|\Delta\mathrm{y}\|_\infty \leqslant \max\left(\|\Phi\|_\infty, \|G\|_\infty\right)\left(\|\Delta\mathrm{c}\|_1 + \int \|\Delta\mathrm{b}(y)\|_1 \,\mathrm{d}y\right).$$

- Did not prove uniqueness. (But true.)
- Also get continuous dependence on data.

Shooting Method

Idea: Want to make use of the fact that we can already solve IVPs.
Problem: Don't know *all* left BCs.

Demo: Shooting method

What about systems?

What are some downsides of this method?

What's an alternative approach?

Finite Difference Method

Idea: Replace u' and u'' with finite differences.

For example: second-order centered

$$\begin{aligned} u'(x) &= \frac{u(x+h)-u(x-h)}{2h} + O(h^2) \\ u''(x) &= \frac{u(x+h)-2u(x)+u(x-h)}{h^2} + O(h^2) \end{aligned}$$

Demo: Finite differences

What happens for a nonlinear ODE?

Demo: Sparse matrices

Collocation Method

$$(*) \begin{cases} y'(x) = f(y(x)), \\ g(y(a), y(b)) = 0. \end{cases}$$

1. Pick a basis (for example: Chebyshev polynomials)

   $$\hat{y}(x) = \sum_{i=1}^{n} \alpha_i T_i(x)$$

   Want $\hat{y}$ to be close to solution y. So: plug into $(*)$.

   Problem: $\hat{y}$ won't satisfy the ODE at all points at least.
   We do not have enough unknowns for that.

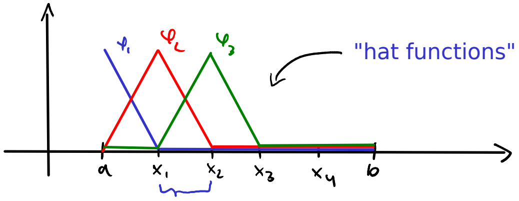
2. Idea: Pick n points where we would like $(*)$ to be satisfied.
   $\rightarrow$ Get a big (non-)linear system
3. Solve that (LU/Newton)$\rightarrow$ done.

Galerkin/Finite Element Method

$$u''(x) = f(x), \qquad u(a) = u(b) = 0.$$

Problem with collocation: Big dense matrix.

Idea: Use piecewise basis. Maybe it'll be sparse.



one "finite element"

What's the problem with that?

Weak solutions/Weighted Residual Method

Idea: Enforce a 'weaker' version of the ODE.

Galerkin: Choices in Weak Solutions

Make some choices:

- Solve for $u \in \operatorname{span}\{\text{hat functions } \varphi_i\}$
- Choose $\psi \in W = \operatorname{span}\{\text{hat functions } \varphi_i\}$ with $\psi(a) = \psi(b) = 0$.
  $\rightarrow$ Kills boundary term $[u'(x)\psi(x)]_a^b$.

These choices are called the Galerkin method. Also works with other bases.

Discrete Galerkin

Assemble a matrix for the Galerkin method.

Outline

Advertisement

Remark: Both PDEs and Large Scale Linear Algebra are big topics. Will only scratch the surface here. Want to know more?

- CS555 $\rightarrow$ Numerical Methods for PDEs
- CS556 $\rightarrow$ Iterative and Multigrid Methods
- CS554 $\rightarrow$ Parallel Numerical Algorithms

We would love to see you there! :)

Solving Sparse Linear Systems

Solving $A\mathbf{x} = \mathbf{b}$ has been our bread and butter.

Typical approach: Use factorization (like LU or Cholesky)
Why is this problematic?

Idea: Don't factorize, iterate.
Demo: Sparse Matrix Factorizations and "Fill-In"

‘Stationary’ Iterative Methods

Idea: Invert only part of the matrix in each iteration. Split

$$A = M - N,$$

where M is the part that we are actually inverting. Convergence?

$$\begin{aligned} A\mathrm{x} &= \mathrm{b} \\ M\mathrm{x} &= N\mathrm{x} + \mathrm{b} \\ M\mathrm{x}_{k+1} &= N\mathrm{x}_k + \mathrm{b} \\ \mathrm{x}_{k+1} &= M^{-1}(N\mathrm{x}_k + \mathrm{b}) \end{aligned}$$

- These methods are called *stationary* because they do the same thing in every iteration.
- They carry out fixed point iteration.
  $\rightarrow$ Converge if contractive, i.e. $\rho(M^{-1}N) < 1$.
- Choose M so that it's easy to invert.

Choices in Stationary Iterative Methods

What could we choose for M (so that it's easy to invert)?

Name	M	N
Jacobi	D	$-(L+U)$
Gauss-Seidel	$D+L$	$-U$
SOR	$\frac{1}{\omega}D+L$	$\left(\frac{1}{\omega}-1\right)D-U$

where L is the below-diagonal part of A, and U the above-diagonal.

Demo: Stationary Methods

Conjugate Gradient Method

Assume A is symmetric positive definite.

Idea: View solving $A\mathrm{x} = \mathrm{b}$ as an optimization problem.

$$\text{Minimize} \quad \varphi(\mathrm{x}) = \frac{1}{2}\mathrm{x}^T A\mathrm{x} - \mathrm{x}^T\mathrm{b} \quad \Leftrightarrow \quad \text{Solve} \quad A\mathrm{x} = \mathrm{b}.$$

Observe $-\nabla\varphi(\mathrm{x}) = \mathrm{b} - A\mathrm{x} = \mathrm{r}$ (residual).

Use an iterative procedure (s_k is the search direction):

$$\begin{aligned} \mathrm{x}_0 &= \langle\text{starting vector}\rangle \\ \mathrm{x}_{k+1} &= \mathrm{x}_k + \alpha_k \mathrm{s}_k, \end{aligned}$$

CG: Choosing the Step Size

What should we choose for α_k (assuming we know s_k)?

CG: Choosing the Search Direction

What should we choose for s_k?

CG: Further Development

Demo: Conjugate Gradient Method

Introduction

Notation:

$$\frac{\partial}{\partial x} u = \partial_x u = u_x.$$

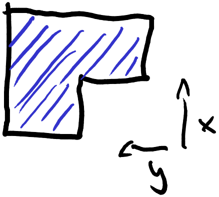
A *PDE* (*partial differential equation*) is an equation with multiple partial derivatives:

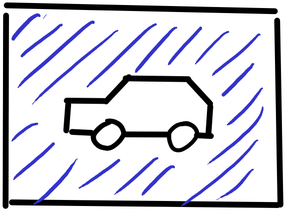
$$u_{xx} + u_{yy} = 0$$

Here: solution is a function $u(x, y)$ of two variables.

Examples: Wave propagation, fluid flow, heat diffusion

- Typical: Solve on domain with complicated geometry.



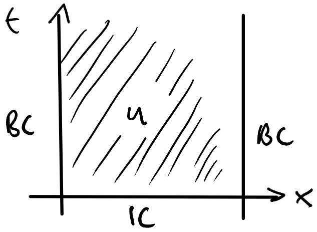


Initial and Boundary Conditions

- Sometimes one variable is time-like.
  What makes a variable time-like?
  - Causality
  - No geometry

Have:

- PDE
- Boundary conditions
- Initial conditions (in t)



Time-Dependent PDEs

Time-dependent PDEs give rise to a *steady-state* PDE:

$$u_t = f(u_x, u_y, u_{\mathrm{xx}}, u_{\mathrm{yy}}) \quad \to \quad 0 = f(u_x, u_y, u_{\mathrm{xx}}, u_{\mathrm{yy}})$$

Idea for time-dep problems (Method of Lines):

- Discretize spatial derivatives first
- Obtain large (semidiscrete) system of ODEs
- Use ODE solver from Chapter 9

Demo: Time-dependent PDEs

Notation: Laplacian

Laplacian (dimension-independent)

$$\Delta u = \operatorname{div}\operatorname{grad} u = \nabla \cdot (\nabla u) = u_{xx} + u_{yy}$$

Classifying PDEs

Three main types of PDEs:

- hyperbolic (wave-like, conserve energy)
  - first-order conservation laws: $u_t + f(u)_x = 0$
  - second-order wave equation: $u_{tt} = \Delta u$
- parabolic (heat-like, dissipate energy)
  - heat equation: $u_t = \Delta u$
- elliptic (steady-state, of heat and wave eq. for example)
  - Laplace equation $\Delta u = 0$
  - Poisson equation $\Delta u = f$
    (Pure BVP, similar to 1D BVPs, same methods apply–FD, Galerkin, etc.)

Outline

Outline