Numerical Analysis / Scientific Computing

CS450

Andreas Kloeckner

Fall 2024

Outline

What's the point of this class?

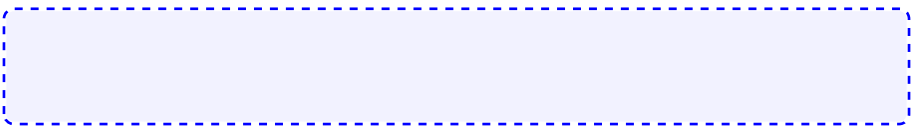
'*Scientific Computing*' describes a family of approaches to obtain approximate solutions to problems *once they've been stated mathematically*.

Name some applications:

What do we study, and how?

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

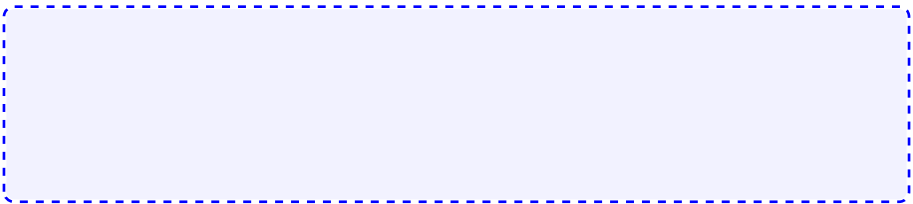
What's the general approach?



What makes for *good* numerics?

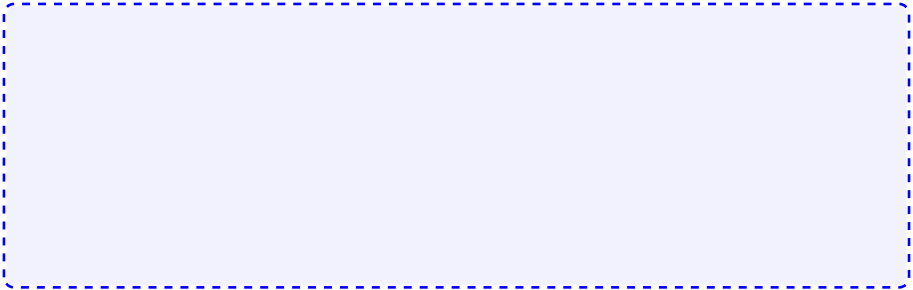
How good of an answer can we expect to our problem?

How fast can we expect the computation to complete?



Implementation concerns

How do numerical methods *get implemented*?



Class web page

https://bit.ly/cs450-f24

- Assignments
  - HW1 (soon!)
  - Pre-lecture quizzes
  - In-lecture interactive content (bring computer or phone if possible)
- Textbook
- Exams
- Class outline (with links to notes/demos/activities/quizzes)
- Discussion forum
- Policies
- Video

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?

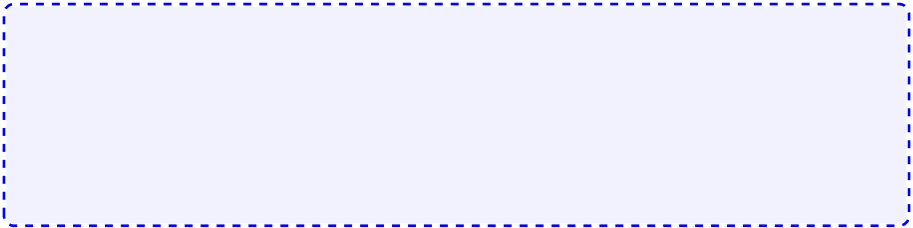
To be able to compute a solution (through a process that introduces errors), the problem...

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

Dependency on Inputs

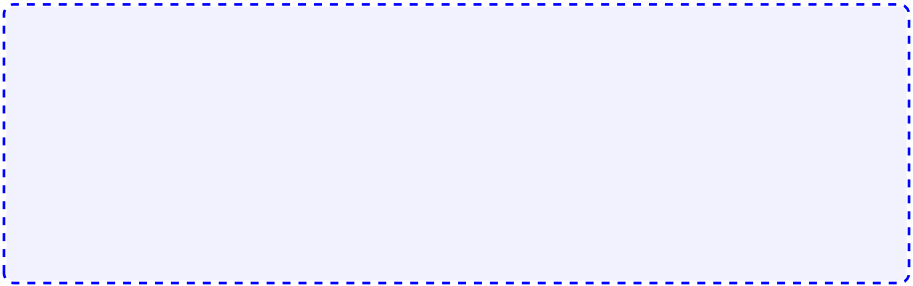
We excluded discontinuous problems–because we don't stand much chance for those.

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



Approximation

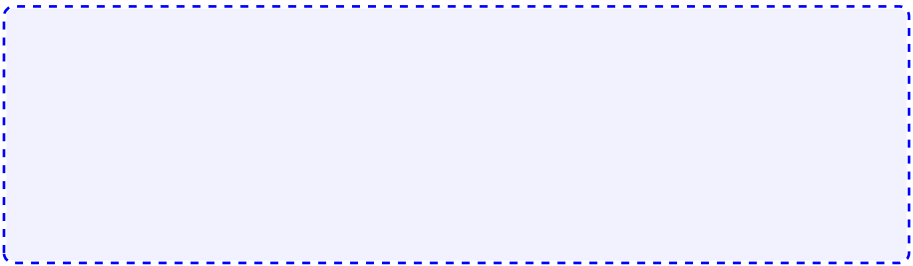
When does approximation happen?



Demo: Truncation vs Rounding [cleared]

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 [cleared]

Norms: Which one?

Does the choice of norm really matter much?

In these notes: If we write $\|\cdot\|$ without any specifics, then the statement is true for any norm. If a specific norm is needed, the notation will indicate that.

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

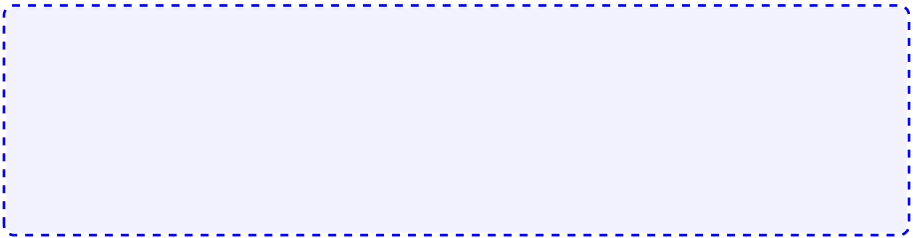
Suppose you wanted $y = \sqrt{2}$ and got $\hat{y} = 1.4$.
What's the (magnitude of) the forward error?

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 magnitude of the relative errors?



Sensitivity and Conditioning

Consider a more general setting: An input x and its perturbation $\hat{x}$.

Absolute Condition Number

Can you also define an *absolute* condition number?

Interpreting a Condition Number

What does it mean for condition numbers to be small/large?

Example: Condition Number of Evaluating a Function

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

Demo: Conditioning of Evaluating tan [cleared]

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*?

Relevance of Backward Error

What do we gain from a bound on backward error like

$$\frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|}{\|\boldsymbol{x}\|} \leq \epsilon?$$

Demo: Backward Stability by Example [cleared]

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. (via example: $(1101)_2$)

Next: Consider *implementation* of FP in hardware.

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

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 stored bits (four total) in the significand.

Unrepresentable numbers?

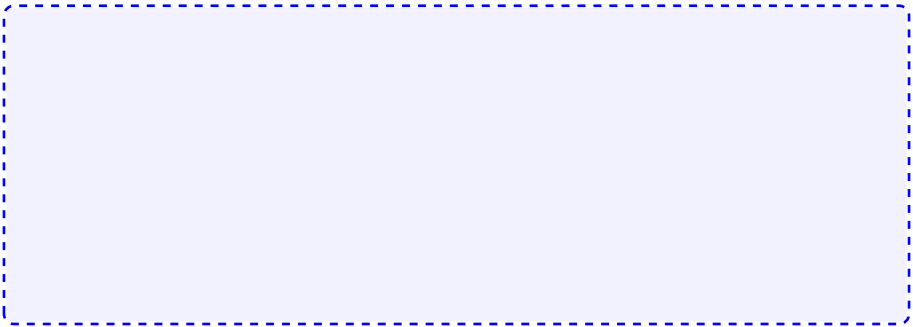
Can you think of a somewhat central number that we cannot represent as

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

Demo: Picking apart a floating point number [cleared]

Subnormal Numbers

What is the smallest representable number in an FP system with 4 stored bits (5 total) in the significand and a stored exponent range of $[-7, 8]$?



Subnormal Numbers, Attempt 2

What is the smallest representable number in an FP system with 4 stored bits in the significand and a (stored) exponent range of $[-7, 8]$?

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.
  Read a story on the epic battle about gradual underflow
- Analogously (but much more simply–no 'supernormals'): the overflow level, *OFL*.

Rounding Modes

Demo: Density of Floating Point Numbers [cleared]

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

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

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

Smallest Numbers Above...

- What is smallest FP number > 1? Assume 4 stored bits (5 total) 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...

FP: Relative Rounding Error

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

FP: Machine Epsilon

What's machine epsilon for double-precision floating point with round-to-nearest? (52 stored bits in the significand, 53 total)

Demo: Floating point and the Harmonic Series [cleared]
Demo: Floating Point vs Program Logic [cleared]

Problems with FP Addition

What happens if you subtract two numbers of very similar magnitude?
As an example, consider $a = (1.1011)_2 \cdot 2^1$ and $b = (1.1010)_2 \cdot 2^1$.

Demo: Catastrophic Cancellation [cleared]

In-Class Activity: Floating Point

In-class activity: Floating Point

Supplementary Material

- Josh Haberman, Floating Point Demystified, Part 1
- David Goldberg, What every computer programmer should know about floating point
- Evan Wallace, Float Toy
- Julia Evans, Examples of Floating Point Problems, 2022

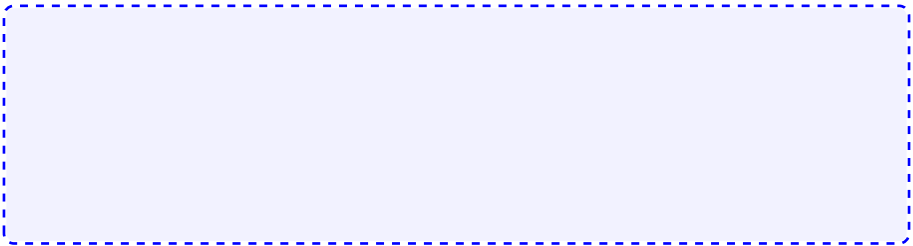
Outline

Solving a Linear System

Given:

- $m \times n$ matrix A
- m-vector $\boldsymbol{b}$

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



Next: Want to talk about conditioning of this operation. Need to measure distances of matrices.

Matrix Norms

We need norms to interact with matrix multiplication in a defined way.

Identifying Matrix Norms

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

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

Demo: Matrix norms [cleared]

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 $\|A + B\| \leq \|A\| + \|B\|$

But also some more properties that stem from our definition:

In these notes: If we write $\|\cdot\|$ (for matrix norms) without any specifics, then the statement is true for any induced norm. If a specific norm is needed, the notation will indicate that.

Conditioning

What is the condition number of solving a linear system $A\boldsymbol{x} = \boldsymbol{b}$?

Conditioning of Linear Systems: Observations

Showed $\kappa(\text{Solve } A\boldsymbol{x} = \boldsymbol{b}) \leq \left\|A^{-1}\right\| \|A\|$.
I.e. found an *upper bound* on the condition number. By finding vectors $\boldsymbol{x}$ and $\Delta\boldsymbol{b}$ that attain equality in submultiplicativity, equality in the condition bound can be achieved for all matrices, i.e. it is *sharp*.

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

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

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 [cleared]
Demo: Conditioning of 2x2 Matrices [cleared]

Residual Vector

What is the residual vector of solving the linear system

$$\boldsymbol{b} = A\boldsymbol{x}?$$

Residual and Error: Relationship

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

Changing the Matrix

So far, only discussed changing the RHS, i.e. $A\boldsymbol{x} = \boldsymbol{b} \quad \rightarrow \quad A\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{b}}$.
The matrix consists of FP numbers, too—it, too, is approximate. I.e.

$$A\boldsymbol{x} = \boldsymbol{b} \quad \rightarrow \quad \widehat{A}\widehat{\boldsymbol{x}} = \boldsymbol{b}.$$

What can we say about the error due to an approximate matrix?

Changing Condition Numbers

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

A typical case: use diagonal matrix as the preconditioner. What is the effect in each case?

In-Class Activity: Matrix Norms and Conditioning

In-class activity: Matrix Norms and Conditioning

Recap: Orthogonal Matrices

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

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

How do orthogonal matrices interact with the 2-norm?

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

Singular Value Decomposition (SVD)

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

Computing the 2-Norm

Using the SVD of A, identify the 2-norm.

Express the matrix condition number $\text{cond}_2(A)$ in terms of the SVD:

Not an Induced Norm: Frobenius

The 2-norm is very costly to compute. Can we make something simpler?

What about its properties?

Frobenius Norm: Properties

Is the Frobenius norm induced by any vector norm?

How does it relate to the SVD?

Solving Systems: Simple cases

Solve $D\boldsymbol{x} = \boldsymbol{b}$ if D is diagonal. (Computational cost?)

Solve $Q\boldsymbol{x} = \boldsymbol{b}$ if Q is orthogonal. (Computational cost?)

Given SVD $A = U\Sigma V^T$, solve $A\boldsymbol{x} = \boldsymbol{b}$. (Computational cost?)

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 [cleared]

What about non-triangular matrices?

Gaussian Elimination

Demo: Vanilla Gaussian Elimination [cleared]

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?

LU Factorization

What is the LU factorization?

Solving $A\boldsymbol{x} = \boldsymbol{b}$

Does LU help solve $A\boldsymbol{x} = \boldsymbol{b}$?

Determining an LU factorization

Demo: LU Factorization [cleared]

Computational Cost

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

- $u_{11} = a_{11}$, $\boldsymbol{u}_{12}^T = \boldsymbol{a}_{12}^T$.
- $\ell_{21} = \boldsymbol{a}_{21}/u_{11}$.
- $L_{22}U_{22} = A_{22} - \ell_{21}\boldsymbol{u}_{12}^T$.

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

Demo: Complexity of Mat-Mat multiplication and LU [cleared]

LU: Failure Cases?

Is LU/Gaussian Elimination bulletproof?

Saving the LU Factorization

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

Demo: LU Factorization with Partial Pivoting [cleared]

Cholesky: LU for Symmetric Positive Definite

LU *can* be used for SPD matrices. But can we do better?

More cost concerns

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

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

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

Demo: BLAS Level 2 vs Level 3 [cleared]

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 linear system

$$\left[\begin{array}{cc|c} A & B & \boldsymbol{b}_1 \\ C & D & \boldsymbol{b}_2 \end{array}\right],$$

can we do 'block Gaussian elimination' 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 matrix?

Round-off Error in LU without Pivoting

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

Round-off Error in LU with Pivoting

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

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 [cleared]

In-Class Activity: LU

In-class activity: LU and Cost

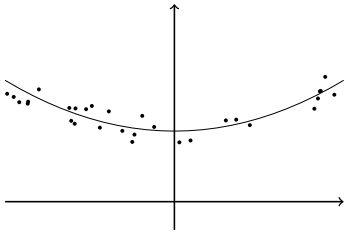
Outline

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

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

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

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

NOTE:

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

Data Fitting: Nonlinearity

Give an example of a nonlinear data fitting problem.

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

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

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

Demo: Interactive Polynomial Fit [cleared]

Properties of Least-Squares

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

Is it always unique?

What happens if A does not have full rank?

Least-Squares: Finding a Solution by Minimization

Examine the objective function, find its minimum.

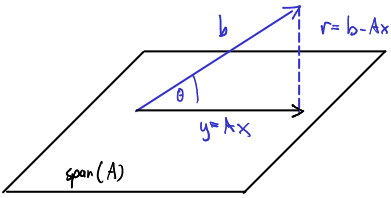
Least squares: Demos

Demo: Polynomial fitting with the normal equations [cleared]

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

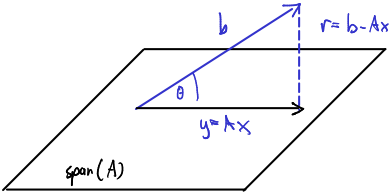
Demo: Issues with the normal equations [cleared]

Least Squares, Viewed Geometrically



Why is $\boldsymbol{r} \perp \text{span}(A)$ 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 $\operatorname{span}\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_\ell\}$ for orthonormal $\boldsymbol{q}_i$?

Least Squares and Orthogonal Projection

Check that $P = A(A^TA)^{-1}A^T$ 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?

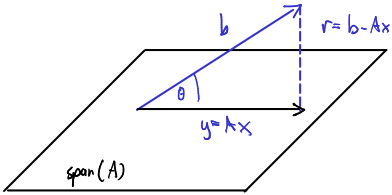
Define the condition number of a tall-and-skinny matrix.

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



Relate $\|A\boldsymbol{x}\|$ and $\boldsymbol{b}$ with θ via trig functions.

Sensitivity and Conditioning of Least Squares (II)

Derive a conditioning bound for the least squares problem.

What values of θ are bad?

Sensitivity and Conditioning of Least Squares (III)

Any comments regarding dependencies?

What about changes in the matrix?

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 $A\boldsymbol{x} \cong \boldsymbol{b}$ 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 [cleared] (shows *modified G-S*)
Demo: Gram-Schmidt and Modified Gram-Schmidt [cleared]
Demo: Keeping track of coefficients in Gram-Schmidt [cleared]
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

Computing QR: A Better Approach

Propose an alternate construction principle for a QR factorization.

Constructing Reflections

Given a plane represented by its (unit) normal vector $\boldsymbol{n}$, construct a reflection about that plane.

Householder Transformations

Find an *orthogonal* matrix Q to zero out the lower part of a vector $\boldsymbol{a}$.

Householder Reflectors: Properties

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

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

Remarks:

- Q: What if we want to zero out only the $i+1$th through nth entry?
  A: Use $\boldsymbol{e}_i$ above.
- It turns out $\boldsymbol{v}' = \boldsymbol{a} + \|\boldsymbol{a}\|_2 \, \boldsymbol{e}_1$ works out, too–just pick whichever one causes less cancellation.
- H is symmetric
- H is orthogonal

Demo: 3x3 Householder demo [cleared]
Demo: Householder in 3D [cleared]

Givens Rotations

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

Demo: 3x3 Givens demo [cleared]

Givens Rotations: Elimination Order

Given a matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

in what order can we apply Givens rotations to eliminate the nonzeros below the diagonal?

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?

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

SVD: Reduced and Full

For a matrix of shape $m \times n$ with $m > n$, what are the shapes of the factors in the SVD?

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

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

- *Low-rank Approximation*

Theorem (Eckart-Young-Mirsky)

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

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

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

$$\min_{\mathrm{rank}(B)=k} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sum_{j=k+1}^{n} \sigma_j^2}.$$

Demo: Image compression [cleared]

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

- The minimum norm solution to $A\boldsymbol{x} \cong \boldsymbol{b}$:

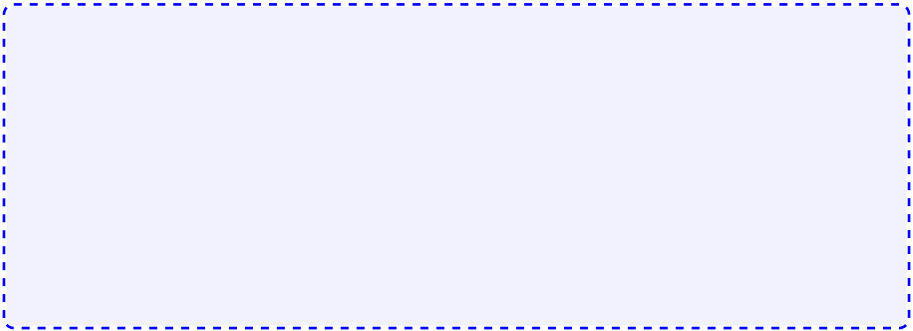
SVD: Minimum-Norm, Pseudoinverse

What is the minimum 2-norm solution to $A\boldsymbol{x} \cong \boldsymbol{b}$ and why?

Generalize the pseudoinverse to the case of a rank-deficient matrix.

Comparing the Methods

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



Demo: Relative cost of matrix factorizations [cleared]

In-Class Activity: Householder, Givens, SVD

In-class activity: Householder, Givens, SVD

Outline

Eigenvalue Problems: Setup/Math Recap

A is an $n \times n$ matrix.

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

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

- 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:

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

Eigenvalue Problems: Motivation from Mechanics

Consider mass-spring systems, e.g. as modeled in (e.g.) myphysicslab.com
What is needed to model?

Finding Eigenvalues

How do you find eigenvalues?

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

$\det(A - \lambda I)$ 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 *iterate*.

Demo: Rounding in characteristic polynomial using SymPy [cleared]

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 $A\boldsymbol{x} = \lambda\boldsymbol{x}$ do?

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

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

Power. $A \to A^k$

Eigenvalue Transformations (II)

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

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 $\left\|(\mu I - D)^{-1}\right\|^{-1} \leq \|F\|$.

Demo: Bauer-Fike Eigenvalue Sensitivity Bound [cleared]

Power Iteration

Demo: Motivating Power Iteration [cleared]

Let $A \in \mathbb{R}^{n \times n}$ and $A\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$ ($j \in \{1, 2, \ldots, n\}$) and $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$.

Pick some $\boldsymbol{x}_0$, consider $\boldsymbol{x}_{i+1} = A\boldsymbol{x}_i$ ($i \in \{0, \ldots\}$). Called Power Iteration.

Convergence of Power Iteration: Notation

- $\lambda_{\text{max}}(A)$: biggest eigenvalue by magnitude
- $\lambda_{\text{max}\,2}(A)$: second-biggest eigenvalue by magnitude.
- $\lambda_{\text{min}\,2}(A)$: second-smallest eigenvalue by magnitude
- $\lambda_{\text{min}}(A)$: smallest eigenvalue by magnitude

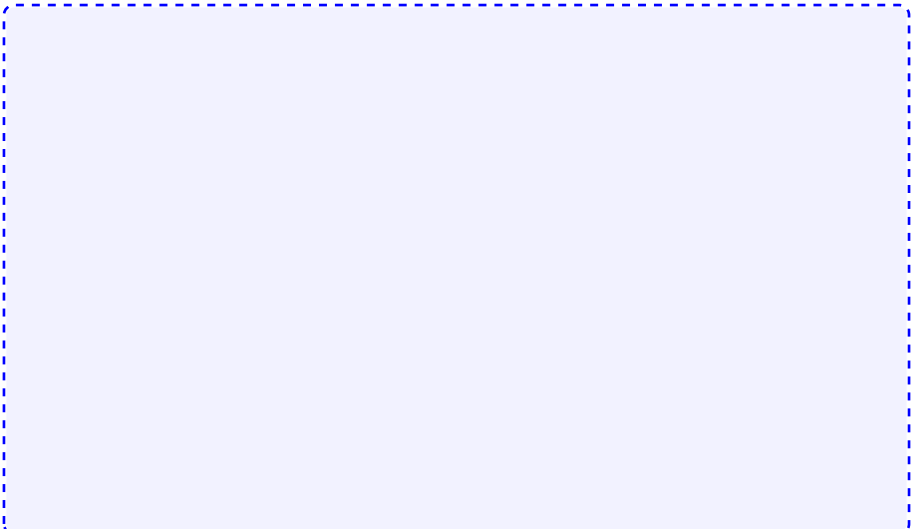
(Not well-defined if there are multiple λ with the same magnitudes. Assume that's not the case.)

Power Iteration: Shift

How does a shift $(A - \sigma I)$ change power iteration?

Power Iteration: Inversion

How does inversion (A^{-1}) change power iteration?



Power Iteration: Shift and Inversion

How does shift-invert $((A - \sigma I)^{-1})$ change power iteration?

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?

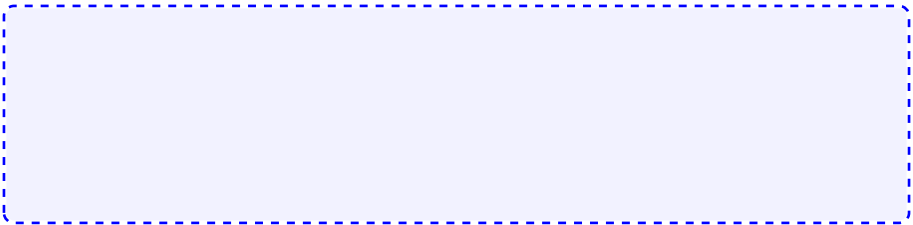
Demo: Power Iteration and its Variants [cleared]

In-Class Activity: Eigenvalues

In-class activity: Eigenvalues

Schur form: Motivation

For finding multiple eigenvalues, want factorization that allows access to all eigenvalues and eigenvectors.
Suggestions?



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 (no 2×2 blocks), 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 [cleared]

QR Iteration/QR Algorithm

Proof sketch: Equivalence of QR iteration/Orth. iteration

Orthogonal Iteration (no bars)

- $X_0 := A$
  - $Q_0 R_0 := X_0$,
  - where we may choose $Q_0 = \bar{Q}_0$
  - $\hat{X}_0 = Q_0^H A Q_0 = Q_0^H Q_0 R_0 Q_0 = R_0 Q_0$
- $X_1 := A Q_0$
  - $Q_1 R_1 := X_1$, and because of $X_1 = Q_0 Q_0^H A Q_0 = Q_0 \bar{X}_1 = Q_0 \bar{Q}_1 \bar{R}_1$ we may choose $Q_1 = Q_0 \bar{Q}_1 = \bar{Q}_0 \bar{Q}_1$.
- $\vdots$

QR Iteration (with bars)

- $\bar{X}_0 := A$
  - $\bar{Q}_0 \bar{R}_0 := A$
- $\bar{X}_1 := \bar{R}_0 \bar{Q}_0 = \hat{X}_0$
  - $\bar{Q}_1 \bar{R}_1 := \bar{X}_1$
- $\bar{X}_2 := \bar{R}_1 \bar{Q}_1$
  - $\bar{X}_2 = Q_1^H A Q_1 = \hat{X}_1$
- $\vdots$

Demo: QR Iteration [cleared]

QR Iteration: Forward *and* Inverse

QR iteration may be viewed as performing inverse iteration. How?

QR Iteration: Incorporating a Shift

How can we accelerate convergence of QR iteration using shifts?

Demo: QR Iteration [cleared] (Shifted)

QR Iteration: Computational Expense

QR at each iteration costs $O(n^3)$—can we make that cheaper?

Demo: Householder Similarity Transforms [cleared]

QR/Hessenberg: Overall procedure

Overall procedure:

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

Why does QR iteration *stay* in Hessenberg form?

What does this process look like for symmetric matrices?

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 [cleared] (Part 1)

Krylov: What about eigenvalues?

How can we use Arnoldi/Lanczos to compute eigenvalues?

Demo: Arnoldi Iteration [cleared] (Part 2)

Computing the SVD (Kiddy Version)

Demo: Computing the SVD [cleared]

"Actual"/"non-kiddy" computation of the SVD:

- Bidiagonalize $A = U \begin{bmatrix} B \\ 0 \end{bmatrix} V^T$, then diagonalize via variant of QR.
- References: Chan '82 or Golub/van Loan Sec 8.6.

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?

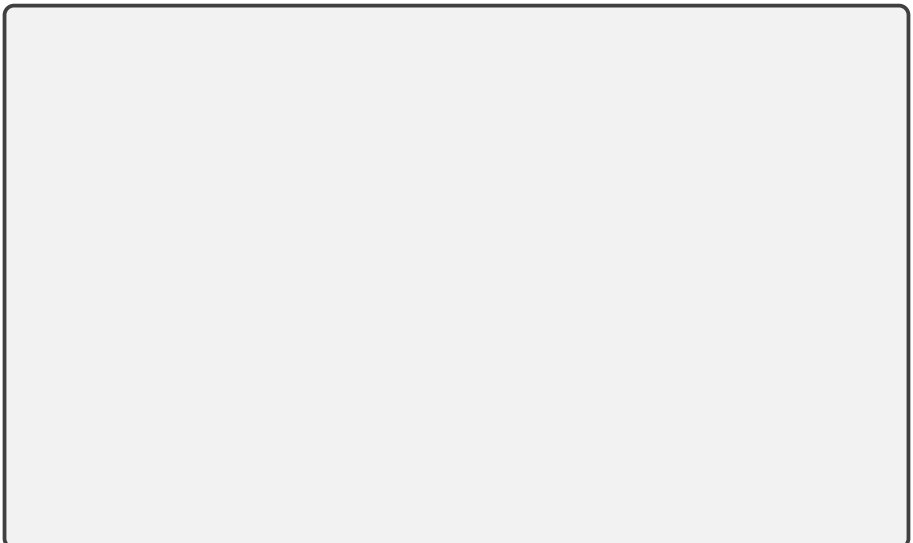
How do multiple roots interact with conditioning?

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 [cleared]

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. $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| < \varepsilon$
3. $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| / \|\boldsymbol{x}_k\| < \varepsilon$

Bisection Method

Demo: Bisection Method [cleared]

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

Mini Review: Taylor's Theorem

Fixed Point Iteration

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

Demo: Fixed point iteration [cleared]

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.

Demo: Newton's method [cleared]

Convergence and Properties of Newton

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

Drawbacks of Newton?

Demo: Convergence of Newton's Method [cleared]

Secant Method

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

Convergence of Properties of Secant

Rate of convergence is $\left(1+\sqrt{5}\right)/2 \approx 1.618$. (proof)

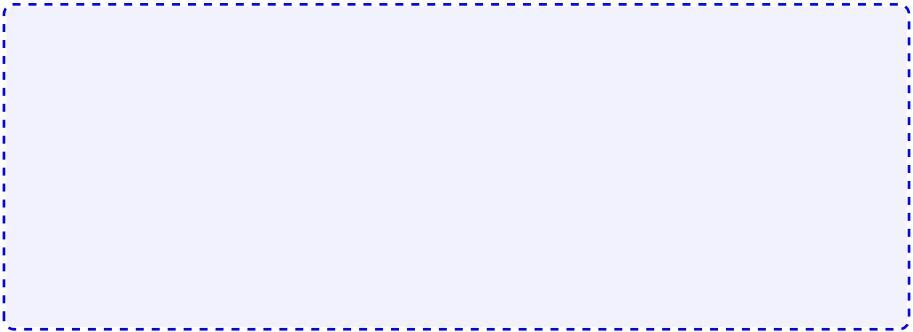
Drawbacks of Secant?

Demo: Secant Method [cleared]
Demo: Convergence of the Secant Method [cleared]

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

Improving on Newton?

How would we do "Newton + 1" (i.e. even faster, even better)?



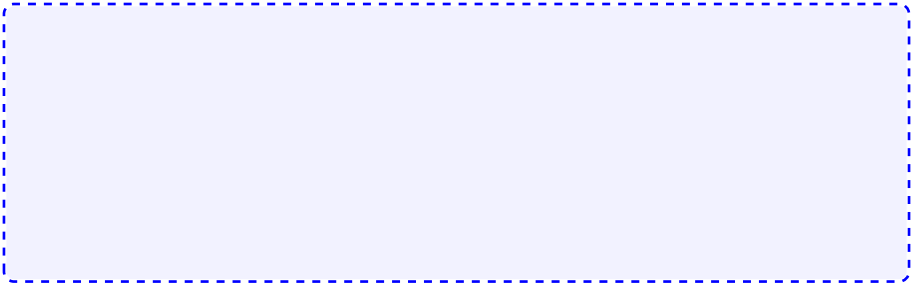
Root Finding with Interpolants

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

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 (n dimensions)

$$\begin{aligned} \boldsymbol{x}_0 &= \langle \text{starting guess} \rangle \\ \boldsymbol{x}_{k+1} &= \boldsymbol{g}(\boldsymbol{x}_k) \end{aligned}$$

When does this converge?

Newton's Method (n dimensions)

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

Downsides of n-dim. Newton?

Demo: Newton's method in n dimensions [cleared]

Secant in n dimensions?

What would the secant method look like in n dimensions?

Numerically Testing Derivatives

Getting derivatives right is important. How can I test/debug them?

Outline

Optimization: Problem Statement

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

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

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

If f is coercive and continuous and S is closed, ...

Convexity

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

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

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 $\boldsymbol{x}^*$ 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?

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

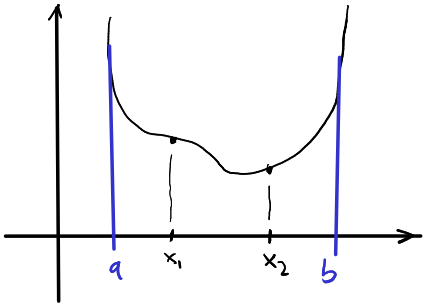
Would like a method like bisection, but for optimization.
In general: No invariant that can be preserved.
Need *extra assumption.*

In-Class Activity: Optimization Theory

In-class activity: Optimization Theory

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?

Newton's Method

Reuse the Taylor approximation idea, but for optimization.

Demo: Newton's Method in 1D [cleared]

Steepest Descent/Gradient Descent

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

Demo: Steepest Descent [cleared] (Part 1)

Steepest Descent: Convergence

Consider quadratic model problem:

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

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

Hacking Steepest Descent for Better Convergence

Extrapolation methods:

Heavy ball method:

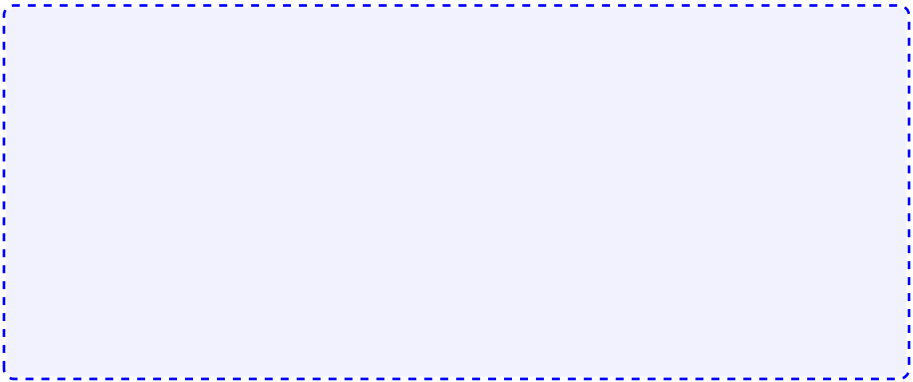
Demo: Steepest Descent [cleared] (Part 2)

Optimization in Machine Learning

What is *stochastic gradient descent* (*SGD*)?

Conjugate Gradient Methods

Can we optimize in *the space spanned* by the last two step directions?



Demo: Conjugate Gradient Method [cleared]

Nelder-Mead Method

Idea:

Demo: Nelder-Mead Method [cleared]

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 [cleared]

Quasi-Newton Methods

Secant/Broyden-type ideas carry over to optimization. How?
Come up with a way to update to update the approximate Hessian.

BFGS: Secant-type method, similar to Broyden:

$$B_{k+1} = B_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} - \frac{B_k \boldsymbol{s}_k \boldsymbol{s}_k^T B_k}{\boldsymbol{s}_k^T B_k \boldsymbol{s}_k}$$

In-Class Activity: Optimization Methods

In-class activity: Optimization Methods

Nonlinear Least Squares: Setup

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

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

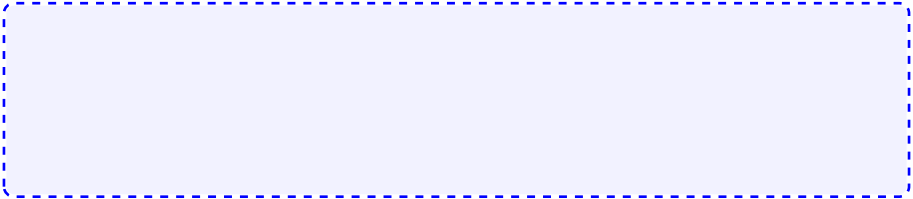
Gauss-Newton

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

Gauss-Newton: Observations?

Demo: Gauss-Newton [cleared]

Observations?



Levenberg-Marquardt

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

Constrained Optimization: Problem Setup

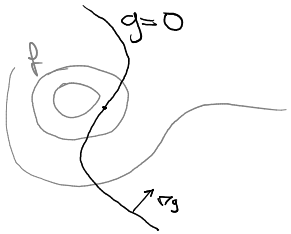
Want $\boldsymbol{x}^*$ so that

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

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

Constrained Optimization: Necessary Condition

Lagrange Multipliers



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

Idea: Turn constrained optimization problem for $\boldsymbol{x}$ into an *unconstrained* optimization problem for $(\boldsymbol{x}, \boldsymbol{\lambda})$. How?

Lagrange Multipliers: Development

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

Demo: Sequential Quadratic Programming [cleared]

Inequality-Constrained Optimization

Want $\boldsymbol{x}^*$ so that

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

Develop a necessary condition for a minimum.

Lagrangian, Active/Inactive

Put together the overall Lagrangian.

What are active and inactive constraints?

Karush-Kuhn-Tucker (KKT) Conditions

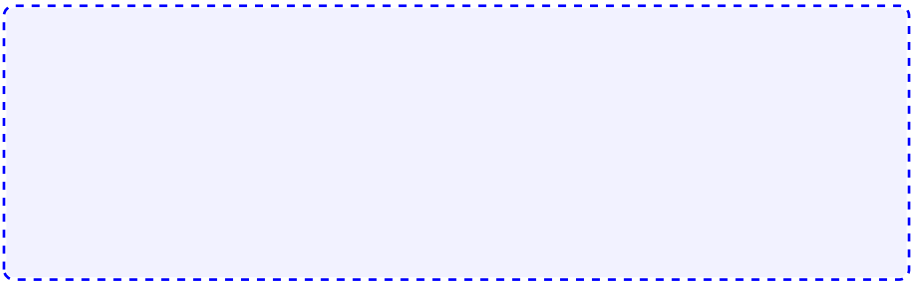
Develop a set of necessary conditions for a minimum.

Outline

Interpolation: Setup

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

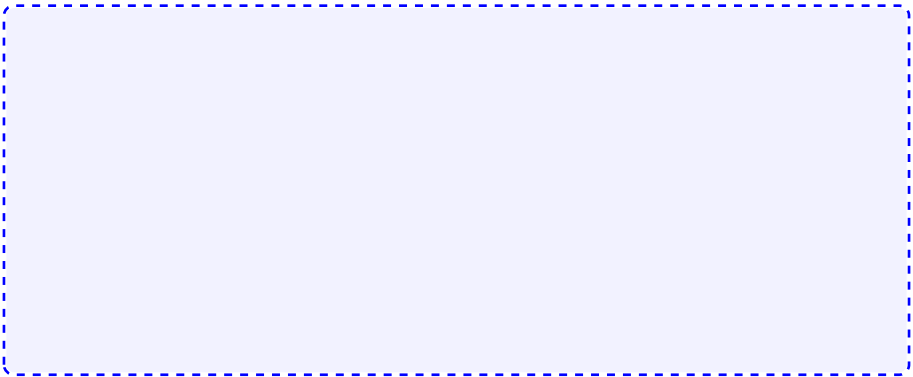
How is this not the same as function fitting? (from least squares)



Interpolation: Setup (II)

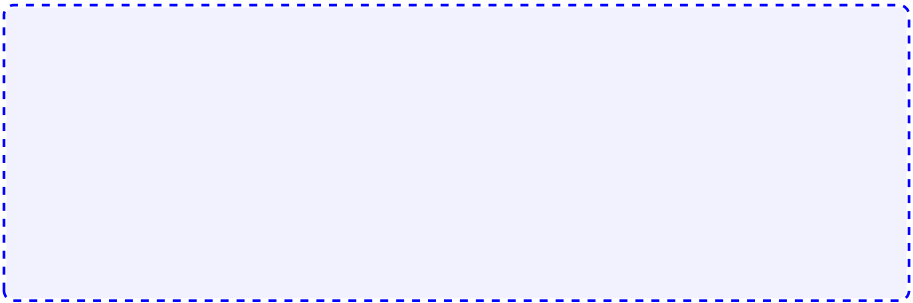
Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

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?

Demo: Lebesgue Constant [cleared]

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 $1, x, x^2, x^3, x^4, \ldots$
- 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 [cleared]
- Why not equispaced?
  Demo: Choice of Nodes for Polynomial Interpolation [cleared]

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

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

Write down the Lagrange interpolant for nodes $(x_i)_{i=1}^m$ and values $(y_i)_{i=1}^m$.

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!

Orthogonality of Functions

How can functions be orthogonal?

Constructing Orthogonal Polynomials

How can we find an orthogonal basis?

Demo: Orthogonal Polynomials [cleared] — Got: Legendre polynomials.
But how can I practically compute the Legendre polynomials?

Chebyshev Polynomials: Definitions

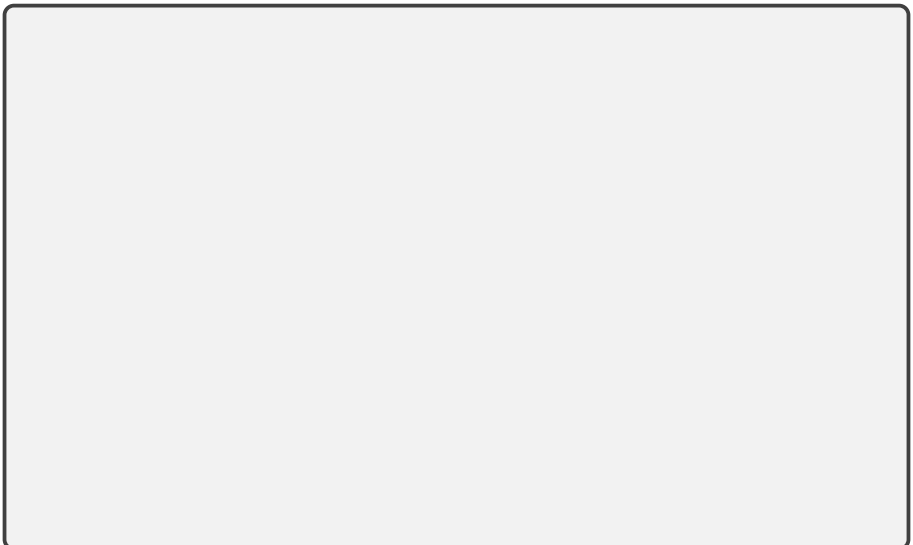
Three equivalent definitions:

- 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 $T_0 = 1,\ T_1 = x$

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.

Edge-clustering seemed like a good thing in interpolation nodes. Do these do that?

Demo: Chebyshev Interpolation [cleared] (Part I-IV)

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

Truncation Error in Interpolation

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

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

Truncation Error in Interpolation: cont'd.

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

Error Result: Connection to Chebyshev

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

Demo: Chebyshev Interpolation [cleared] (Part V)

Error Result: Simplified Form

Boil the error result down to a simpler form.

- **Demo:** Interpolation Error [cleared]
- **Demo:** Jump with Chebyshev Nodes [cleared]

Going piecewise: Simplest Case

Construct a piecewise 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 | f_1 | x_1, y_1 | f_2 | x_2, y_2 | f_3 | x_3, y_3

f_1: $a_1x^3 + b_1x^2 + c_1x + d_1$

f_2: $a_2x^3 + b_2x^2 + c_2x + d_2$

f_3: $a_3x^3 + b_3x^2 + c_3x + d_3$

Piecewise Cubic ('Splines'): Accounting

x_0, y_0 | f_1 | x_1, y_1 | f_2 | x_2, y_2 | f_3 | x_3, y_3

f_1: $a_1x^3 + b_1x^2 + c_1x + d_1$

f_2: $a_2x^3 + b_2x^2 + c_2x + d_2$

f_3: $a_3x^3 + b_3x^2 + c_3x + d_3$

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: Examples

Interpolatory Quadrature: Computing Weights

How do the weights in interpolatory quadrature get computed?

Demo: Newton-Cotes weight finder [cleared]

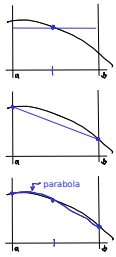
Examples and Exactness

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
S. 3/8	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-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 stability of this method?

So, what quadrature weights make for bad stability bounds?

About Newton-Cotes

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

Demo: Newton-Cotes weight finder [cleared] (again, with many nodes)

Gaussian Quadrature

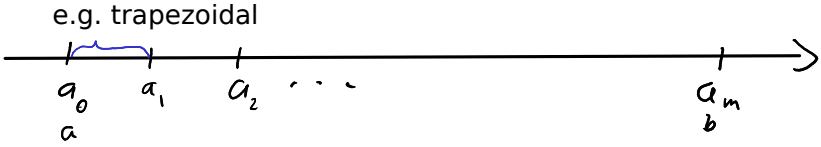
So far: nodes chosen from outside.
Can we gain something if we let the quadrature rule choose the nodes, too? Hope: More design freedom $\rightarrow$ Exact to higher degree.

Demo: Gaussian quadrature weight finder [cleared]

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.



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)

Taking Derivatives Numerically

Why *shouldn't* you take derivatives numerically?

Numerical Differentiation: How?

How can we take derivatives numerically?

Demo: Taking Derivatives with Vandermonde Matrices [cleared] (Basics)

Numerical Differentiation: Accuracy

Demo: Taking Derivatives with Vandermonde Matrices [cleared]

Differentiation Matrices

How can numerical differentiation be cast as a matrix-vector operation?

Demo: Taking Derivatives with Vandermonde Matrices [cleared] (Build D)

Properties of Differentiation Matrices

How do I find second derivatives?

Does D have a nullspace?

Demo: Taking Derivatives with Vandermonde Matrices [cleared] (Shifting and scaling the nodes)

Numerical Differentiation: Shift and Scale

Does D change if we shift the nodes $(x_i)_{i=1}^n \to (x_i + c)_{i=1}^n$?

Does D change if we scale the nodes $(x_i)_{i=1}^n \to (\alpha x_i)_{i=1}^n$?

Finite Difference Formulas from Diff. Matrices

How do the rows of a differentiation matrix relate to FD formulas?

Assume a large equispaced grid and 3 nodes w/same spacing. How to use?

Finite Differences: via Taylor

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 [cleared]
Demo: Floating point vs Finite Differences [cleared]

Richardson Extrapolation

Deriving high-order methods is hard work. Can I just do multiple low-order approximations (with different h and get a high-order one out?

Suppose we have $F = \tilde{F}(h) + O(h^p)$ and $\tilde{F}(h_1)$ *and* $\tilde{F}(h_2)$.

Richardson Extrapolation: Observations,

What are α and β for a first-order (e.g. finite-difference) method if we choose $h_2 = h_1/2$?

Demo: Richardson with Finite Differences [cleared]

Romberg Integration

Can this be used to get *even higher order* accuracy?

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) ▶ waves (time-harmonic)

Demo: Predator-Prey System [cleared]

Initial Value Problems: Problem Statement

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

- $\boldsymbol{y}^{(k)}(t) = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{y}'', \ldots, \boldsymbol{y}^{(k-1)})$ (*explicit*), or
- $\boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{y}', \boldsymbol{y}'', \ldots, \boldsymbol{y}^{(k)}) = 0$ (*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

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

Then if $\boldsymbol{f}$ is *Lipschitz continuous* (has 'bounded slope'), i.e.

$$\|\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{f}(\widehat{\boldsymbol{y}})\| \leq L \|\boldsymbol{y} - \widehat{\boldsymbol{y}}\|,$$

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 IVP is stable if and only if...

An IVP is asymptotically stable if and only if

Example I: Scalar, Constant-Coefficient

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

Solution?

When is this stable?

Example II: Constant-Coefficient System

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

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

When is this stable?

Euler's Method

Discretize the IVP

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

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

Euler's method: Forward and Backward

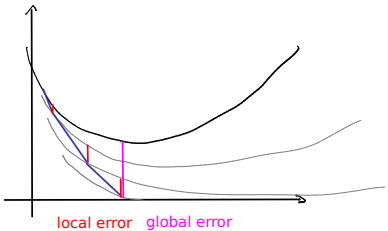
$$\boldsymbol{y}(t) = \boldsymbol{y}_0 + \int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{y}(\tau))\mathrm{d}\tau,$$

Use 'left rectangle rule' on integral:

Use 'right rectangle rule' on integral:

Demo: Forward Euler stability [cleared]

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.

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

Stability: Nonlinear ODEs

What about stability for nonlinear systems, i.e.

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

Stability for Backward Euler

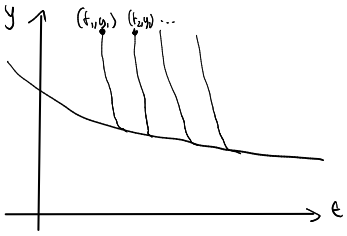
Find out when backward Euler is stable when applied to $y'(t) = \lambda y(t)$.

Demo: Backward Euler stability [cleared]

Stiff ODEs: Demo

Demo: Stiffness [cleared]

‘Stiff’ ODEs



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

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

  stiffness can arise if $J_{\boldsymbol{f}}$ 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).

Runge-Kutta/'Single-step'/'Multi-Stage' Methods

Idea: Compute intermediate 'stage values', compute new state from those:

Can summarize in a *Butcher tableau*:

Runge-Kutta: Properties

When is an RK method explicit?

When is it implicit?

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

Runge-Kutta: Embedded Pairs

How can error in RK integration be controlled?

Heun and Butcher

Stuff Heun's method into a Butcher tableau:

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

RK4

What is RK4?

Demo: Dissipation in Runge-Kutta Methods [cleared]

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):

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

Stability Regions

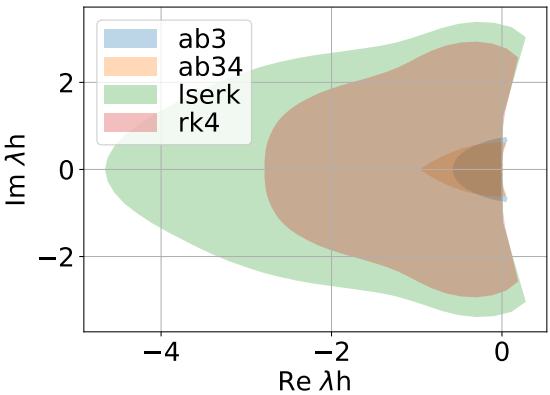
Why does the idea of stability regions still apply to more complex time integrators (e.g. RK?)

Demo: Stability regions [cleared]

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:

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

BCs:

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

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

Does a first-order, scalar BVP make sense?

Example: Linear BCs $B_a\boldsymbol{y}(a) + B_b\boldsymbol{y}(b) = \boldsymbol{c}$.
Is this Dirichlet/Neumann/...?

Do solutions 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.

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

Exploit linearity: split into multiple problems.

Solving the "Boundary" BVP

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

Solving the "Volume" BVP

$$(V)\begin{cases}\boldsymbol{y}'_V(x) = A(x)\boldsymbol{y}_V(x) + \boldsymbol{b}(x)\\ B_a\boldsymbol{y}_V(a) + B_b\boldsymbol{y}_V(b) = 0\end{cases}$$

ODE Systems: Conditioning

Altogether:

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

For perturbed problem with $\boldsymbol{b}(x) + \Delta\boldsymbol{b}(x)$ and $\boldsymbol{c} + \Delta\boldsymbol{c}$, derive a bound on $\|\Delta\boldsymbol{y}\|_\infty$.

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 [cleared]

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 [cleared]

What happens for a nonlinear ODE?

Demo: Sparse matrices [cleared]

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

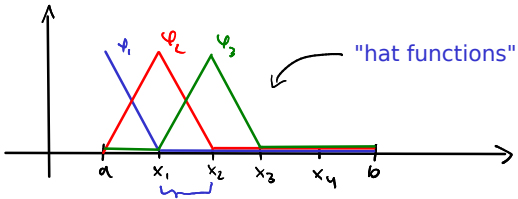
3. Solve that (LU/Newton)$\rightarrow$ done.

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.



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 $u \in \text{span}\{\text{hat functions } \varphi_i\}$
- Choose $\psi \in W = \text{span}\{\text{hat functions } \varphi_i\}$ 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 $A\boldsymbol{x} = \boldsymbol{b}$ 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" [cleared]

'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?

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

- 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 [cleared]

Conjugate Gradient Method

Assume A is symmetric positive definite.

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

Use an iterative procedure ($\boldsymbol{s}_k$ is the search direction):

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

CG: Choosing the Step Size

What should we choose for α_k (assuming we know $\boldsymbol{s}_k$)?

CG: Choosing the Search Direction

What should we choose for $\boldsymbol{s}_k$?

CG: Further Development

Introduction

Notation:

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

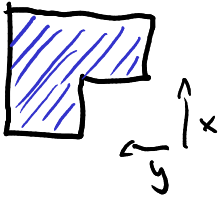
A *PDE* (*partial differential equation*) is an equation with multiple partial derivatives:

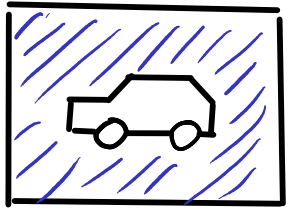
$$u_{xx} + u_{yy} = 0$$

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.



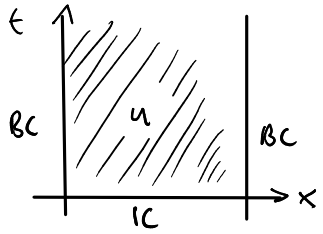


Initial and Boundary Conditions

- 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:

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

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 [cleared]

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