1 Preliminaries

Note that

$$\begin{aligned} np.dot(A, x) &= Ax \\ np.dot(x, A) &= x^T A \\ A@B = np.matmul(A, B) &= AB \\ np.dot(x, y) &= x^T y \end{aligned}$$

2 Lecture 1

Rounding, Approximation, Well Posedness

Definition 2.1. We say that a problem is well posed if its solution exists, is unique and depends continuously on the inputs.

Definition 2.2. Rounding error refers to our representation of real numbers as floating point numbers.

Definition 2.3. Truncation error refers to our representation of infinite collections as finite ones. For example, the truncation of a Taylor series to a finite quantity of terms is an example of truncation.

Definition 2.4. Approximating a solution can be divided into two stages: approximation before the calculation and approximation during the calculation. Approximation before the calculation includes the following:

- How we model the problem.
- The accuracy of our input data.
  - How we measure the input data
  - How we compute the input data

Approximation during the calculation includes:

- Rounding Error
- Truncation Error

Remark 2.5. Recall that norms allow for constant pullout, satisfy the triangle inequality and output 0 if and only the input to the norm is also 0.

Example 2.6. It is easy to undestand rounding and truncation error. The first set of approximations can be understood as follows:

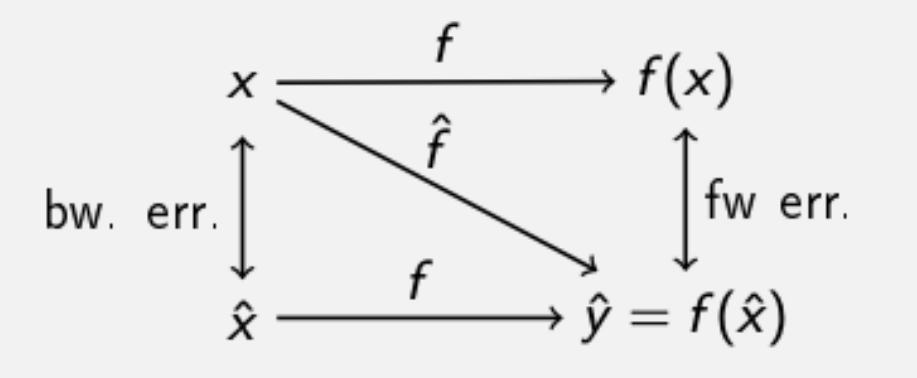
Suppose we use the surface area of the earth as an input. To calculate the surface area, we first assume that the earth is a sphere, then we somehow measure the radius and then we compute that area by multiplying the radius squared by π.

3 Lecture 2

Rounding and Error

Definition 3.1. Let x be some input, f some true function and $\hat{f}$ some estimated function. Absolute forward error is the difference $\left|f(x) - \hat{f}(x)\right|$. Relative forward error is $\frac{\left|f(x)-\hat{f}(x)\right|}{|f(x)|}$.

Definition 3.2. Backward error is the difference between the original input and the input that, under the true function f, would have given you $\hat{f}(x)$, the output of the original input under the approximate function.



Remark 3.3. Suppose I present to you two problems p_1 and p_2 that have the same forward error but the backward error of p_1, $b(p_1)$ is such that $b(p_1) > b(p_2)$. Is problem 1 or 2 more favorable? Is any more favorable? We can generally say that p_1 is more favorable, because it takes a larger input perturbation in p_1 to produce, under a true function f, the same output perturbation as p_2. This would suggest that p_1 is more *well conditioned*.

Remark 3.4. Truncation and rounding error make up the ways in which an operation is approximate. By truncation error, we refer to the process of replacing infinite sums and expressions with finite ones and by rounding error, we refer to the error introduced in representing real numbers with computer based number systems.

Lemma 3.5. This is sort of a lemma:

The relative error in the forward error tells us how many digits are significant in our estimate of the solution to a problem.

Example 3.6. Suppose we estimate $\sqrt{2} = 1.414\ldots$ as 1.4. Then the negative log of the relative error is $- \approx \log(0.01) = 2$, meaning that (correctly) our estimate is accurate up to 2 digits.

Definition 3.7. The condition number is

$$\sup_{x \in \text{domain}} \frac{\text{Relative Forward Error}}{\text{Relative backward error}}$$

Theorem 3.8. Assuming that a true function f is differentiable, then its condition number is given by

$$\sup_{x \in \text{Domain}} \frac{xf'(x)}{f(x)}$$

Proof.

$$\begin{aligned} \Delta y = \frac{f(x+\Delta x) - f(x)}{1} &= f'(x)\Delta(x) \\ \implies \frac{\Delta y}{y} / \frac{\Delta x}{x} & \\ = \frac{f'(x)\Delta x}{y} / \frac{\Delta x}{x} & \end{aligned}$$

and the latter simplies to the claimed expression. □

Problem Forward Error. Approximate $f(x) = 1/x$ by the taylor expansion $1 - (x-1)$. What is the forward error when we let $x = 0.5$?

It is 0.5

Problem Backward Error. What about the backward error if we use $x = 0.5$

We need to find the value x' such that $f(x') = 1.5$. That is $x' = \frac{2}{3}$. Therefore, $|x - x'| = 0.16$.

Problem Condition Number. Determine the condition number of $\sin(x)$ on the interval $[0, \pi/2]$.

Proof. The formula gives us that $k(f) = \sup_{x \in [0,\pi/2]} \frac{x \cos x}{\sin x}$. Note that this is a decreasing function on. the given interval, so that the largest value is taken on at $x = 0$. By L'Hopital's rule, that gives us:

$$\frac{\cos x - x \sin x}{\cos x}.$$

which evaluates to 1 at $x = 0$. Thus, the condition number is 1. □

Problem Larger Condition Number. Over which interval is the condition number larger: $[0, 2\pi]$ or $[10^4\pi, 10^4\pi + 2\pi]$.

Proof. Recall that the formula is $\frac{xf'(x)}{f(x)}$, and that $f'(x)$ and $f(x)$ will assume the same values up to equivalence classes of $\sin$. Thus, the x in the numerator accounts for any potential difference, making the second interval have a greater condition number. □

Definition 3.9. A problem is well conditioned if, when considering a problem, small perturbations in an input do not result in large perturbations in the problem's output.

Definition 3.10. A method is stable if, when using it to solve a problem, the solution produced by the method is an exact solution to a nearby problem (ie the same problem applied to a different input). In other words, the error introduced by the method is no worse than the introduction of a little error into the input data.

Remark 3.11. We see that conditioning describes a problem, whereas stability describes a method used to solve that problem.

Definition 3.12. A method is accurate if it produces solutions that are very close to the true answer. A solution is accurate if both the method is stable and the problem well conditioned. If a problem has a stable algorithm but is not itself well conditioned, then the solution may not necessarily be accurate.

4 Lecture 3: Floating Point

The following questions were prelecture questions:

Problem Operator Conditioning. Let $x > 1$. Which of $f(x) = 3x, f(x) = 3/x, f(x) = 3 - x, f(x) = \sqrt{x}$ is ill posed (here, make the restrictive definition that a well posed problem must have a finite condition number.?

Proof. We can compute the condition numbers using the formula $xf'(x)/f(x)$ since all functions are differentiable over the domain $x > 1$. This gives:

- $$\frac{3x}{3x} = 1$$
- $$\frac{x(-3/x^2)}{3/x} = \frac{(-3/x)}{3/x} = -1$$
- $$\frac{x(-1)}{3-x} \to \infty \text{ as } x \to 3$$
- $$\frac{-1/2}{\sqrt{x}}$$

□

Problem Forward and Backward Errors. Approximate $\sqrt{1+x}$ by $1+x/2$. What is the absolute backward error using $x = 8$.

Proof. To get $1 + 8/2 = 5$ using f, we need to let $x = 24$. Hence, the absolute backward error is $|8 - 24| = 16$. □

Problem Why choose a stable algorithm?. A stable algorithm always:

- produces an accurate result
  - False. If a problem is additionally well conditioned, then a stable algorithm applied to it may result in an accurate solution; otherwise, no

- insensitive to data error
  - No, a stable algorithm only assures us that an answer is a solution to a nearby problem
- improves the conditioning of the problem
  - No, this is distinct from stability. See the first retort.
- gives the correct answer to a nearby problem
  - This is exactly the definition.

Problem Condition Number of Square Root. At $x = 7$?

Proof.

$$\frac{1/2}{\sqrt{x}}x \Big/ \sqrt{x} = 1/2$$

□

Problem Backward Error. Given $x = 0.5$, approximate $\cos x$ by $1 - \frac{x^2}{2}$. What is the backward error?

Proof. Find $\hat{x}$ such that $\cos \hat{x} = 0.875$. Subtract that from 0.5. □

Problem Estimate Condition Number. Estimate $f(x) = \log(1 + x)$ (remember that log means natural log) with $x - \frac{x^2}{2} + \frac{x^3}{3}$. Suppose we know that $f(0.5168967) = \hat{f}(0.5)$. Then we know that the relative backward error is $\frac{0.516897-0.5}{0.5}$. Now use this information to approximate the condition number of f.

Proof. We already know the relative backward error, and the condition number is relative forward error over relative backward error. Hence, the condition number estimate is

$$\frac{\hat{f}(0.5) - f(0.5)}{f(0.5)} \Big/ \text{Relative Backward Error}$$

□

Definition 4.1. Suppose that we represent numbers using 64 bits. 32 bits will be dedicated to the whole portion of any number – we will progress under this system, going from rightmost bit to left most bit, from 2^0 to 2^32; similarly, 32 bits will be dedicated to the fractionalpart of any number – we will progress from rightmost bit to leftmost bit from $2^{-}32$ to $2^{-}1$.

Remark 4.2. Suppose that we wish to represent 2^{32}. Observe that largest number representable using this scheme is $2^{32} - 2^{-32}$.

It follows that the relative error of our represntation of 2^{32} is

$$\frac{2^{-32}}{2^{32}} = 2^{-64}$$

Now suppose that we wish to represent some number x in the range $(2^{-32}, 2^{-31})$. Assuming that 2^{-32} is closer to this number then 2^{-32}, the relative error of this representation is

$$\frac{\left|x - 2^{-32}\right|}{|x|} \leq \frac{\left|2^{-33}\right|}{|x|} \leq \frac{\left|2^{-33}\right|}{2^{-32}} = 2^{-1}$$

Observe that whereas the relative error in the former is good, in the latter it is poor. This motivates our definition of the double precision floating point system.

Example 4.3. Represent 13 in scientific notation in a binary number system. Require that the first and only digit before the decimal point (remember that this is scientific notation) is a 1; further require that 1 must occupy this position. You will find that the representation is uniquely $(1.101)_2 * 2^3$.

Remark 4.4. Suppose I told you that we we will represent all numbers as we did above. Leaving aside how we would represent 0, observe that since we require that the first and only digit before the decimal point is a 1, if we want to store such representations, there is no point in storing the 1. This allows us to say the following:

Definition 4.5.

- The double precision floating point system allocates 52 bits to store the fractional part of a number represented using the scheme above, where a 1 in the units place implicitly precedes the fractional part.
- 11 bits are allocated to store an exponent and 1 bit is allocated to store a sign bit.
  - The 11 exponent bits allow us to get integers in the range $[0, 2047]$. Assuming that we begin with an offest of -1023, meaning that 0 maps to -1023, we can represent an exponent in the range $[-1023, 1024]$.
- When the exponent becomes 0, instead of mapping to -1023, we to -1022 (yes, this contradicts the previous line, but introducting the contradiction and then correcting it is pedagogically more gradual). We also do away with the implicit 1 that precedes the fractional part of the significand (recall tha the fractional part of the significand is the only part which is stored in our representation). 0 is represented if we let all bits of the fractional part be 0.
  - Since we are no longer required to have an implicit, leading 1, notice that we can now drop down past $1 * 2^{-1022}$. Represent 2^{-1023} as $0.100\cdots \times 2^{-1022}$; represent 2^{-1024} as $0.0100\cdots \times 2^{-1022}$ and so forth. UNRESOLVED
- We map 2^{1024} to ∞ and if the fractional part contains any non-zero digit, we call that values ∞ instead. UNRESOLVED
- Define the UFL and OFL to be the smallest and greatest normal numbers that are not subnormal nor ∞.

Definition 4.6. Suppose that we have a number $x = b.\underbrace{b_1 b_2 \ldots b_{52}}_{\text{all useable digits consumed}} d_1 d_2 d_3.$

We need to somehow represent x in our floating point system. To do so, we

round on the basis of $d_1d_2d_3$. Now we could make the scheme round to nearest, meaning that add 1 to b_{52} if $d_1d_2d_3 > 100$ and do nothing if $d_1d_2d_3 < 100$, but we must then choose what to do when $d_1d_2d_3 = 100$. What then? If we make the consistent, arbitrary decision to round up, we will produce statistical noise. Thus, the scheme employed is to round to even:

If b_{52} is 0, then do nothing, so that x is even; if b_{52} is 1, then add 1 to b_{52}.

Remark 4.7. Some floating point exercises `https://relate.cs.illinois.edu/course/cs450-s19/flow/inclass-floating-point/start/`

Definition 4.8. Let ϵ be the smallest number such that $\text{fl}(x(1+\epsilon)) > x$. If we use round to even, it is 2^{-52}; if we use round to nearest, with rounding up in case of a tie, it is 2^{-53} – in which case we also distinguish 2^{-52} with some title.

Lemma 4.9. The relative error in representing any number x within the normal range is at most ϵ.

Proof. Given x, there is an a such that

$$2^a \leq x \leq 2^{a+1}$$

Since we can represent any number $y = 2^a + \epsilon * n$ up to $y = 2^{a+1}$, it follows that there is at least one representable number in the range $[x, x + \epsilon]$. This allows us to say that the absolute error between this number and x is at most ϵ, which is formalized below:

$$\frac{|x - (1+\epsilon)x|}{|x|} = \frac{|x|\epsilon}{|x|} = \epsilon$$

□

5 Lecture 4

5.1 Quiz

Problem 1.

Estimating Output Error with Condition Numbers A function f(x) has a condition number of 2.3×10^2. For x=65 with an absolute error Δx=5×10−5, what is an upper bound on the relative error of the output?

TIP: On any quiz/exam question where you are asked to enter a number, you can also enter an expression, which will be evaluated for you. No need to bring out the calculator. (I.e. in the box below, you could simply enter "100*13", which would count the same as entering "1300".)

Make sure your answer has at least two accurate digits.

Solution 5.1. Relative Error <= Relative Input Error * Condition Number Relative Error <= (Machine Epsilon or some problem specific relative input error) * condition Number <= $(5 * 10^-5)/(65) * 2.3 * 10^2$

Problem 2.

Floating Point Rounding What is the value of $2\sqrt{\times}$ when calculated in a decimal floating point system with 3 digits that uses rounding? (Apply rounding to both the input data and the result of the arithmetic.)

Solution 5.2. $\sqrt{2} \approx 1.41$ and $\pi \approx 3.14$. Multiply both and round to 3 decimal places.

Problem 3.

In a normalized floating point system that supports subnormal numbers, which of the following binary operations on two positive numbers could lead to overflow?

Solution 5.3. Addition, Multiplication and Division can all lead numbers to explode outside of the normalized floating number range.

If two addends are already normalized floating point numbers, however, then there is no chance that subtracting these numbers can lead to a result that is not a normalized floating point[1]

Problem 4.

1 point Machine Epsilon and UFL Which of the following statements are true?

Select all that apply: Machine epsilon places a lower bound on the magnitude of normal floating point numbers. Machine epsilon is determined by the number of bits in the significand. Machine epsilon places an upper bound on the relative error of representing a real number in a floating point format. The UFL is determined by the number of bits in the exponent.

Solution 5.4. The bottom 3 are correct; the first is wrong, because the magnitude of normal floating point numbers is determined by the exponent range; the second is correct, because machine epsilon is $\beta^{-(p-1)}$ where β is a base and p a precision; the third is correct, because given a floating point number y, the smallest perturbation to y that results in a different floating point number is $\epsilon|y|$. The fourth is correct, because the order of magnitude of UFL is β^{L+1} where the exponent range falls in $[L, U]$.

Problem 5.

Floating Point Relative Error Which of the following is closest to the relative error from representing $e \times 2 - 145$ in IEEE single precision.

Solution 5.5. Notice that $e \times 2^{-145}$ is subnormal number. The underflow limit for single precision IEEE floating point numbers is 2^{-126}. Anything smaller requires that the leading 1 be shifted along the fractional part of a number. Since $e \times 2^{-145} \approx 2^{-144}$, it follows that we need to shift the leading 1 18 units to the right of the decimal. This will leave 5 digits (bits) in the fractional component for futher reprsentation. If we have n digits, then our relative error is approximately β^{-n}. Whence, the relative error here is 2^{-5}.

Problem 6.

[1]Recall that a normalized floating point number has 1 in the fractional part before the significand.

Properties of Floating Point Operations Which of the following is true for floating point multiplication and addition? Select all that apply: Floating point addition is associative. Floating point multiplication is commutative. Floating point multiplication is associative. Floating point addition is commutative.

Solution 5.6. Floating point multiplication and addition are commutative but not associative.

I skip problem 7 – if you understood the solution to an earlier problem, you'll be okay.

Example 5.7. To perform floating point addition, suppose that you have $b_0.b_1 \dots b_n \times 2^x$ and $d_0.d_1 \dots d_n \times 2^y$ where $x > y$; make sure that you multiply the second number by 2^{x-y} and then perform grade school addition.

Definition 5.8. Catastrophic cancellation takes place when the number of accurate digits in the result of an operation are far less than the number of accurate digits in the members of that operation.

Example 5.9. Supposing that we opereate in double precision and that a and b both have 12 digits of accuracy. Suppose that a and b agree up to their first 10 digits.

When a and b are subtracted, the first 10 digits will be cancelled and the final 2 digits will shift leftwards, becoming the 2, sole accurate digits. The remaining digits will be filled with garbage. From this, we see that whereas we had 12 digits of accuracy, we now only have 2 (ie our relative error is 10^-2).

Definition 5.10. Given a vector norm $\|\cdot\|$, we define the matrix norm to be

$$\sup_{x \in \mathcal{D}, x \neq 0} \frac{\|Ax\|}{\|x\|}$$

This is equivalent to

$$\sup_{x \in \mathcal{D}, x \neq 0} \frac{\|Ax\|}{\|x\|}$$
$$\sup_{x \in \mathcal{D}, x \neq 0} \frac{\left\|A \frac{x}{\|x\|}\right\|}{1}$$
$$\sup_{x \in \mathcal{D}, \|x\| = 1} \frac{\|Ax\|}{1}$$

We denote this norm by $\|A\|$.

Proposition 5.11.

$$\|A\|\|x\| \geq \|Ax\|$$

for all x that A can multiply.

Proof. Follows by definition. □

Proposition 5.12.

$$\|A\|_1 = \max_j \sum_i |A_{i,j}|$$

$$\|A\|_\infty = \max_i \sum_j |A_{i,j}|$$

An easy way to remember the 1 norm and ∞ norm is to note that these matrix norms also coincide for a vector, for which the 1 norm is the sum of absolute values, and the ∞ norm is the maximum value.

Definition 5.13. Suppose that A is invertible and that we wish to find the condition number of the problem of solving $Ax = b$ where b is the input and x the ouput; then the condition number is given by

$$\frac{\|\Delta x\|}{\|x\|} \Big/ \frac{\|\Delta b\|}{\|b\|}$$

$$\frac{\|\Delta x\|}{\|x\|} \Big/ \frac{\|\Delta b\|}{\|b\|}$$

Let $x = A^{-1}b$ and $\Delta x = A^{-1}\Delta b$

$$\frac{\|A^{-1}\Delta b\|}{\|A^{-1}b\|} \Big/ \frac{\|\Delta b\|}{\|b\|}$$

Now apply submultiplicativity

$$\leq \|A\|\|A^{-1}\|$$

This inequality can be shown to be sharp for every matrix: for every matrix, there is a vector b for which this bound is attained, and so the condition number of the problem is $\|A\|\|A^{-1}\|$

We also call this quantity the condition number of the matrix.

Definition 5.14. If a matrix is not invertible, then we say that $k(A) = \infty$.

Proposition 5.15. $k(A) = k(A^{-1})$ and matrix vector multiplication has the same conditioning as solving a system of linear equations.

Proof. The first is easy. To find the y such that $Ax = y$, take x as the input of $A^{-1}y = x$. □

Proposition 5.16.

- The following properties hold:
  - $\|AB\| \leq \|A\|\|B\|$

    * For $\|ABx\| \leq \|A\|\|Bx\| \leq \|A\|\|B\|\|x\|$. The previous was true for all x; in particular, for all $x \neq 0$, this implies that

$$\|AB\| = \frac{\|ABx\|}{\|x\|} \leq \|A\|\|B\|$$

  - $\|A + B\| \leq \|A\| + \|B\|$.
  - $\left\|AA^{-1}\right\| = 1 \leq \|A\|\left\|A^{-1}\right\| \implies 1 \leq k(A)$

- $k(\gamma A) = k(A)$.
- $\|\gamma A\| = \gamma\|A\|$.
- $k(A) = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|}(\min_{x\neq 0} \frac{\|Ax\|}{\|x\|})^{-1}$
  -

    By definition:

$$\left\|A^{-1}\right\| = \max_{x\neq 0} \frac{\left\|A^{-1}x\right\|}{\|x\|}$$

    Let $y = A^{-1}x$, meaning that $Ay = x$. Then

$$= \max_{x\neq 0} \frac{\|y\|}{\|Ay\|}$$

    Since $y = A^{-1}y$, we can consider the max over y instead of x:

$$= \max_{y\neq 0} \frac{\|y\|}{\|Ay\|}$$
$$= (\min_{y\neq 0} \frac{\|Ay\|}{\|y\|})^{-1}$$

- If D is a diagonal matrix, then $k(D) = \frac{\max d_{i,i}}{\min_{d_{i,i}}}$
- An orthogonal matrix has norm 1 in any norm.

Remark 5.17. A computational technique to evaluating a condition number is multiply a matrix A by many vectors that have unit norm and take the maximum norm of the resultant vectors divided by the minimum norm of the resultant vectors.

6 Lectures 5 and 6

6.1 Quiz Lecture 5

Problem 1.

Floating Point Cancellation For which of the following reasons is cancellation in floating-point computation usually bad?

Choice* The digits lost are the least significant. The digits lost are the most significant. The result is usually not exactly representable. Subsequent operations are likely to underflow or overflow.

Solution 6.1. The digits lost during catastrophic cancellation are the most significant ones (ie the left most ones). Since we're losing digits in the resultant answer, our answer is often exactly representable (remember that we have about $-\log(2^{-52}) \approx 16 = -\log(10^{-16})$ digits of relative accuracy).

Problem 2.

Cancellation and Rounding Which of the following statements is true regarding the relationship between rounding and cancellation?

Choice* In computation that is done with exact (not rounded) values, cancellation cannot occur. Catastrophic cancellation occurs when cancellation amplifies rounding errors in the input. Subtracting two rounded numbers will always amplify the error. Cancellation from subtracting two rounded inputs of similar magnitude and sign can be reduced by first converting the rounded inputs to higher precision.

Solution 6.2. Even with exact values, we can find that leading digits are chopped off, if the two numbers agree to a high number of decimal places. Yes, catastrophic cancellation is the phenemonon that takes places when we subtract so much from a number that whatever is shifted upwards to replace the nullified digits is inaccurate, rounding error. This does not mean that subtracting any two numbers will always amplify rounding error, however. Even if you convert numbers to higher precision, this will not change the fact that cancellation will take place – several initial numbers will still match; it will reduce the chances of catastrophic cancellation taking place, however.

Problem 3.

Cancellation and Relative Error Suppose a and b are stored in single precision and agree to four decimal digits. Assume a is known to seven decimal digits and b is known to five decimal digits.

Let c=b−a.

Accounting only for cancellation error, how many decimal digits of accuracy are in c?

How many decimal digits are in c after accounting for the relative error in a and b?

Solution 6.3. In single precision, we have $-\log(2^{-22}) \approx 7$ digits of accuracy. Thus, a and b are (ignoring their relative accuracies initially) known really to 7 digits, so that if subtract them and nullify their first 4 digist, then there are only 3 digits that remain. If we account for relative error in the subtraction, however, then there are only $\min 7-4, 5-4 = 1$ digits that are accurate.

Problem 4.

Changing the RHS You just solved a linear system Ax=b. Unfortunately, the RHS b that you solved it with was wrong.

Worried, you compute ‖Δb‖‖b‖10−12. The condition number of your matrix is about 10000.

What could your worst-case relative error in the solution x be due to your use of the wrong RHS?

Solution 6.4. Just multiply condition number by relative input error.

Problem 5.

Distance to Singularity Which of the following is a good indicator that a matrix is nearly (as measured by the matrix norm) singular?

Choice* Its norm is small. Its determinant is small. Its condition number is large. Its norm is large.

Solution 6.5. $k(\gamma A) = k(A)$. That is, no scaling of a matrix can change its condition number; all the other quantities can change very much, however.

Problem 6.

What is the 2 norm of a diagonal matrix:

Solution 6.6. It is the maximum of teh diagonal entries. The condition number is the ratio of the maximum diagonal entry in absolute value to the minimum one in absolute value.

6.2 Quiz for Lecture 6

Relative Residual Consider a matrix $A = \begin{bmatrix} -9 & -5 \\ 8 & 3 \end{bmatrix}$ and right-hand side vector b=[3−2]. Using the infinity norm, calculate the relative residual if elements of the solution vector xˆ are rounded to one significant digit. Include at least three significant digits in your answer.

Solution 6.7. $\|A\| = 14$. Let $\hat{x} = A^{-1}b = \begin{bmatrix} -0.08 \\ 0.5 \end{bmatrix}$ if you round to 1 digit. Then $r = A\hat{x} - b = \begin{bmatrix} -0.22 \\ 0.14 \end{bmatrix}$

Meaning that $\|r\| = 0.22$. And $\|x\| = 0.5$

$$\frac{\|r\|}{\|x\|\|A\|} = \frac{0.22}{0.5 * 15}$$

which is correct.

Problem 2.

A stupid definition question.

Problem 3.

Gaussian Elimination In the following questions, consider Gaussian elimination with the prescribed pivoting strategy to generate the lower (L) and upper (U) triangular factors for the following matrix,

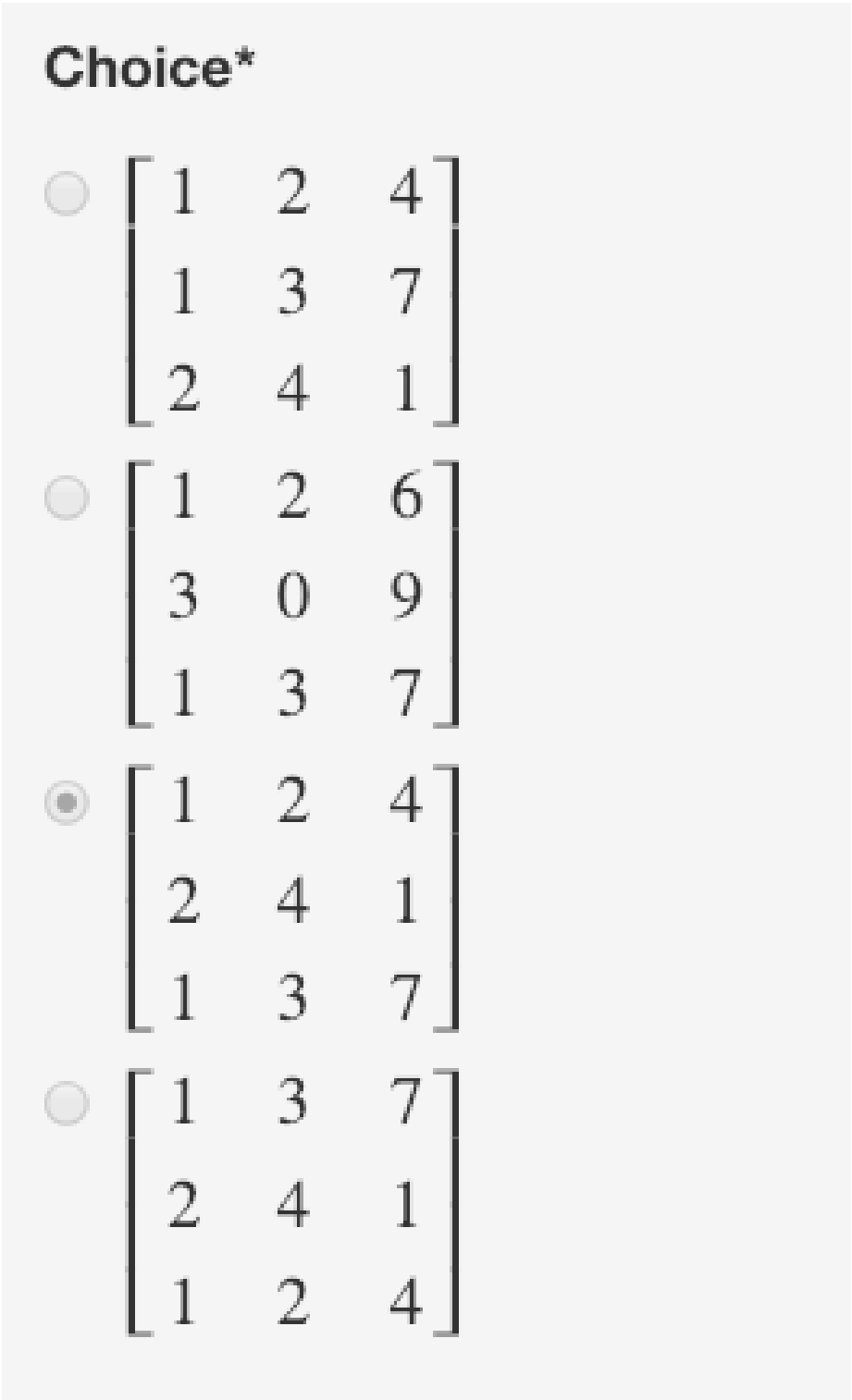
$$A = \begin{bmatrix} 2 & 1 & 3 \\ 2 & 4 & 8 \\ 4 & -7 & 4 \end{bmatrix}$$

Know that with pivoting, we must first swap rows 1 and 3 and then perform gaussian elimination.

Problem 4.

Existence of LU Decomposition with no Pivoting For which of the following matrices does a LU factorization without pivoting not exist?

Solution 6.8. I have



Sufficient Condition: If a pivot is 0 (assuming that we don't pivot the matrix when performing gaussian elimination), then the matrix will fail to provide an

LU factorization.

Problem 5.

Just apply

$$\text{rel error} \leq k(A)\frac{\|r\|}{\|A\|\|x\|}$$

Problem 6.

Elimination Matrices

1 point

Consider two 10×10 elimination matrices M_4 and M_7.

- M_4 only has off-diagonal entries (below the diagonal) in column 4.
- M_7 only has off-diagonal entries (below the diagonal) in column 7.

Which of the following is true?

Choice*

- $M_4M_7 = M_4 + M_7 - I$ (where I is the identity matrix)
- $M_7M_4 = M_4 + M_7 - I$ (where I is the identity matrix)
- $M_4 = M_7$
- None of these

Note that elimination matrices must progress from left to right, meaning that elimination matrices M_1 and M_2 must be such that the off diagonal entry of M_1 is left of the off diagonal entry of M_2, if we want M_1M_2 to merge.

The following is the definition of the residual.

$$\boldsymbol{r = b - A\hat{x}.}$$

Remark 6.9. The residual itself does not reveal much. Suppose we calculate $r = b - Ax$. Now solve for $kAx = kb$ and the residual required to solve that is k times as great. This is why we define the relative residual:

$$\frac{\|r\|}{\|A\| \cdot \|\hat{x}\|}$$

We can obtain a bound on the relative forward error required to solve $Ax = b$ in terms of r.

$$\|\Delta \boldsymbol{x}\| = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\| = \|\boldsymbol{A}^{-1}(\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b})\| = \| - \boldsymbol{A}^{-1}\boldsymbol{r}\| \leq \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{r}\|.$$

Dividing both sides by $\|\hat{\boldsymbol{x}}\|$ and using the definition of $\text{cond}(\boldsymbol{A})$, we then have

$$\frac{\|\Delta \boldsymbol{x}\|}{\|\hat{\boldsymbol{x}}\|} \leq \text{cond}(\boldsymbol{A}) \frac{\|\boldsymbol{r}\|}{\|\boldsymbol{A}\| \cdot \|\hat{\boldsymbol{x}}\|}.$$

Remark 6.10. This bound tells us that if the residual is small and the matrix and well conditioned, then the relative error is low.

Example 2.8 Small Residual. Consider the linear system

$$\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 0.913 & 0.659 \\ 0.457 & 0.330 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.254 \\ 0.127 \end{bmatrix} = \boldsymbol{b},$$

whose matrix we saw in Example 2.7. Consider two approximate solutions

$$\hat{\boldsymbol{x}}_1 = \begin{bmatrix} 0.6391 \\ -0.5 \end{bmatrix} \quad \text{and} \quad \hat{\boldsymbol{x}}_2 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}.$$

The norms of their respective residuals are

$$\|\boldsymbol{r}_1\|_1 = 7.0 \times 10^{-5} \quad \text{and} \quad \|\boldsymbol{r}_2\|_1 = 2.4 \times 10^{-2}.$$

So which is the better solution? We are tempted to say $\hat{\boldsymbol{x}}_1$ because of its much smaller residual. But the exact solution to this system is $\boldsymbol{x} = [1, \ -1]^T$, as is easily confirmed, so $\hat{\boldsymbol{x}}_2$ is actually much more accurate than $\hat{\boldsymbol{x}}_1$. The reason for this surprising behavior is that the matrix $\boldsymbol{A}$ is ill-conditioned, as we saw in Example 2.7, and because of its large condition number, a small residual does not imply a small error in the solution. To see how $\hat{\boldsymbol{x}}_1$ was obtained, see Example 2.17.

Demo: Vanilla Gaussian Elimination

What do we get by doing Gaussian Elimination?

> *Row Echelon Form.*

How is that different from being upper triangular?

> Zeros allowed on and above the diagonal.

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

> That's called *Gauss-Jordan elimination*. Turns out to be computationally inefficient. We won't look at it.

Remark 6.11. Also note that a matrix is in row echelon form if the first non-zero entry of each row (what was the pivot during gaussian elimination) is to the right of the first non-zero entry of any preceding row; moreover, entries in rows above the pivot (but in the same column) must be 0.

Elimination Matrices

What does this matrix do?

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

- Add $(-1/2)\times$ the first row to the third row.
- One elementary step in Gaussian elimination
- Matrices like this are called *Elimination Matrices*

Remark 6.12. If we add k to the identity matrix at entry i, j, and left multiply the resultant matrix C by some matrix of interest A, then the result is to take the j th row of A multiply it by k and then add it to i. We can undo this process by using the same matrix but, in place of k, using $-k$. This second matrix is the inverse to the elimination matrix C.

Elimination Matrices

What does this matrix do?

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

- Add $(-1/2)\times$ the first row to the third row.
- One elementary step in Gaussian elimination
- Matrices like this are called *Elimination Matrices*

Remark 6.13. Suppose that we multiply A by an elimination matrix M_1, then by M_2 up to M_l, where M_l is the last matrix required to turn A into Row Echelon Form. Eventually, we will have

$$(M_l \dots M_1)A = U \implies A = (M_l \dots M_1)^{-1}U$$

At first glance, this is okay, because it turns out that left multiplication of an elimination matrix X by Y such that X has a non-zero off diagonal at column i and Y has a non-zero off diagonal at column j where $i < j$ results in an elimination matrix that just merges X and Y [2]

For whatever reason, pivoting foils this attempt:

No, very much not:

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}.$$

Q: Is this a problem with the process or with the entire *idea* of LU?

$$\begin{bmatrix} 1 & \\ \ell_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ & u_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \rightarrow u_{11} = 0$$

$$\underbrace{u_{11} \cdot \ell_{21}}_{0} + 1 \cdot 0 = 2$$

It turns out to be that A doesn't *have* an LU factorization.

The solution is to repeatedly apply permutations to A (in the form of permutation matrices) so that the pivot is the largest element in terms of absolute value in its column.

Thus, we now have

$$(M_l P_l \dots M_1 P_1)A = U \implies A = (M_l P_l \dots M_1 P_1)^{-1} U$$

However, what should be L above is not always left triangular. It can be shown that a factorization of $(M_l P_l \dots M_1 P_1)^{-1}$ does, however, give us a lower triangular system.

[2] Note that merging also takes place if we multiply two elimination matrices that have their off diagonal non-zero entry in the same column as each other.

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

Define: $L_3 := M_3$
Define $L_2 := P_3 M_2 P_3^{-1}$
Define $L_1 := P_3 P_2 M_1 P_2^{-1} P_3^{-1}$

$$\begin{aligned}&(L_3 L_2 L_1)(P_3 P_2 P_1)\\ =&M_3 (P_3 M_2 P_3^{-1})(P_3 P_2 M_1 P_2^{-1} P_3^{-1}) P_3 P_2 P_1\\ =&M_3 P_3 M_2 P_2 M_1 P_1 \quad (!)\end{aligned}$$

$$\underbrace{P_3 P_2 P_1}_{P} A = \underbrace{L_1^{-1} L_2^{-1} L_3^{-1}}_{L} U.$$

$L_1, \ldots, L_3$ are still lower triangular!

Q: Outline the solve process with pivoted LU

Changing Condition Numbers

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

Diagonal scaling is a simple strategy that sometimes helps.

- Row-wise: $DA\mathbf{x} = D\mathbf{b}$
- Column-wise: $AD\hat{\mathbf{x}} = \mathbf{b}$
  Different $\hat{\mathbf{x}}$: Recover $\mathbf{x} = D\hat{\mathbf{x}}$.

What is this called as a general concept?

Preconditioning

- Left preconditioning: $MA\mathbf{x} = M\mathbf{b}$
- Right preconditioning: $AM\hat{\mathbf{x}} = \mathbf{b}$
  Different $\hat{\mathbf{x}}$: Recover $\mathbf{x} = M\hat{\mathbf{x}}$.

Remark 6.14. Suppose that D above satisfies $k(D) \approx 1$. Then

$$k(DA) = \|DA\|\|(DA)^{-1}\| \leq \|D\|\|A\|\|A^{-1}\|\|D^{-1}\| \leq k(A)$$

so that the condition number of $K(DA)$ is no greater than the condition number of A.

Assuming that D is invertible, then the set of x satisfying $Ax = b$ is precisely the set of x satisfying $Ax = b$. Left multiplication by D of A is called, understandably, left preconditioning and scales A in a row-wise manner; right multiplication by D of A is called right preconditioning.

Remark 6.15.

Computational Cost

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

$O(n^3)$

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

Recall

$$M_3 P_3 M_2 P_2 M_1 P_1 A = U \ldots$$

so $O(n^4)$?!!!

Fortunately not: Multiplications with permuation matrices and elimination matrices only cost $O(n^2)$.

So overall cost of LU is just $O(n^3)$.

Demo: Complexity of Mat-Mat multiplication and LU

Multiplication by a permutation matrix is only an n operation, since it involves switching rows. Multiplication by an elimination matrix simply involves scaling one row and mulitplying it by another, and this process is done at most n times for any one elimination matrix (making it $O(n^2)$ as well). Since these transformations are applied at most n times, the process of getting a matrix into LU form is only $O(n^3)$.

Remark 6.16.

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*?

Multiply the top row by $-CA^{-1}$, add to second row, gives:

$$\begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix}.$$

$D - CA^{-1}B$ is called the Schur complement. Block pivoting is also possible if needed.

Not sure why this is significant.

Remark 6.17. Unresolved

Example 2.15 Small Pivots. Using finite-precision arithmetic, we must avoid not only zero pivots but also *small* pivots in order to prevent unacceptable error growth, as shown in the following example. Let

$$\boldsymbol{A} = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix},$$

where ϵ is a positive number smaller than the unit roundoff ϵ_{mach} in a given floating-point system. If we do not interchange rows, then the pivot is ϵ and the resulting

multiplier is $-1/\epsilon$, so that we get the elimination matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 \\ -1/\epsilon & 1 \end{bmatrix},$$

and hence

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{U} = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - 1/\epsilon \end{bmatrix} = \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}$$

in floating-point arithmetic. But then

$$\boldsymbol{LU} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix} \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix} = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix} \neq \boldsymbol{A}.$$

Using a small pivot, and a correspondingly large multiplier, has caused an unrecoverable loss of information in the transformed matrix. If we interchange rows, on the other hand, then the pivot is 1 and the resulting multiplier is $-\epsilon$, so that we get the elimination matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 \\ -\epsilon & 1 \end{bmatrix},$$

and hence

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{U} = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

in floating-point arithmetic. We therefore have

$$\boldsymbol{LU} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix},$$

Remark 6.18. Notice that if we already have an LU factorization, then computing a rank 1 update is just an $O(n^2)$ operation.

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?

Special cases allow something to be done (a so-called *rank-one update*):

$$\hat{A} = A + \mathbf{u}\mathbf{v}^T$$

The Sherman-Morrison formula gives us

$$(A + \mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{u}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{u}}.$$

Proof: Multiply the above by $\hat{A}$ get the identity.
FYI: There is a rank-k analog called the Sherman-Morrison-Woodbury formula.

Demo: Sherman-Morrison

For

$$\left(A + uv^T\right)^{-1} b = A^{-1}b - \frac{\left(A^{-1}u\right) v^T A^{-1} b}{1 + v^T A^{-1} u}$$

And $A^{-1}x$ for any x is an $O(n^2)$ operation. The only other operation in this formula is to compute a dot product.

Remark 6.19.

LU: Special cases

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

$$PA = LU$$

(invertible, not invertible) (Why?)

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

Think carefully about sizes of factors and columns/rows that do/don't matter. Two cases:

- $m > n$ (tall&skinny): $L : m \times n$, $U : n \times n$
- $m < n$ (short&fat): $L : m \times m$, $U : m \times n$

This is called reduced LU factorization.

A matrix A always admits an LU factorization, even if A is singular. First, observe that every column of A must contain at least one non-zero number – or else, why would the column be part of A. Thus, if a pivot entry does not contain a non-zero value, we can rotate rows so that the pivot entry does have

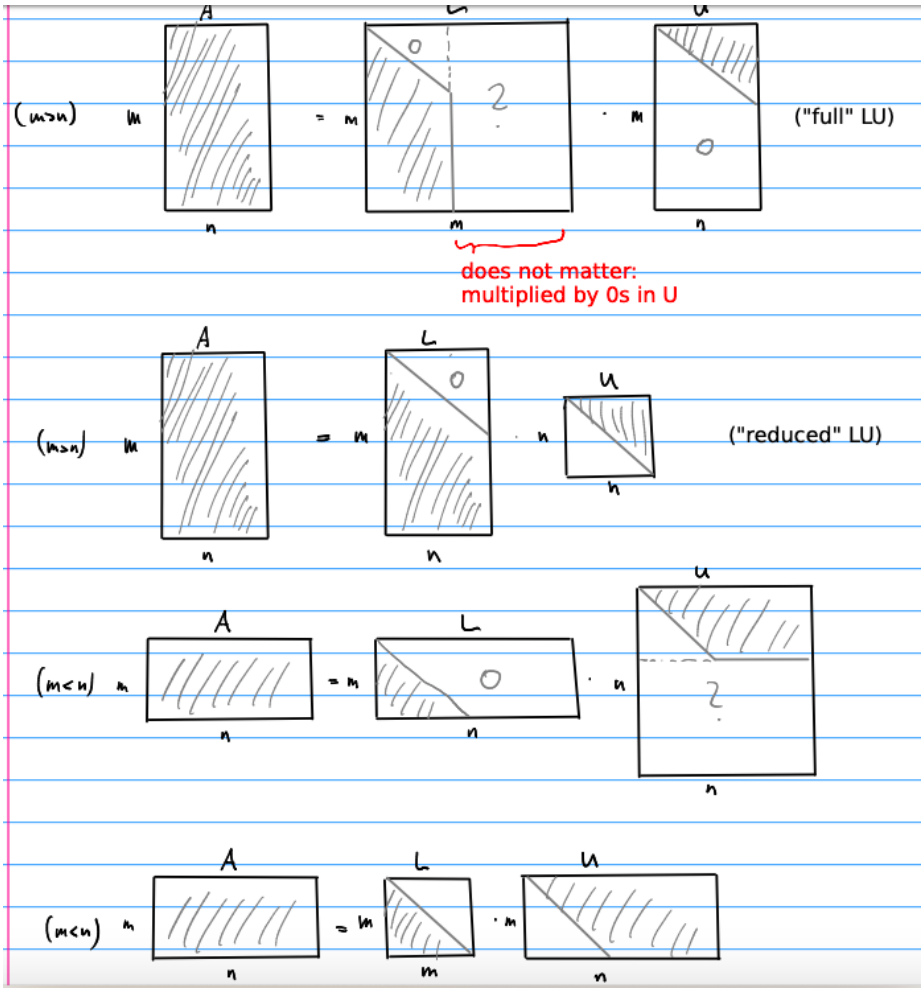
a non-zero value. We then can apply elimination matrices, as needed, until all row values in the pivot's column are 0.

The foregoing tells us that there still exists a sequence of permutations and elimination matrices that bring A into upper triangular form. Since these matrices are each invertible, it follows that L is still invertible. Since $|PA| = 0$, we must have, therefore, $|U| = 0$, which means that 0 must occupy some diagonal entry of U.

Why? A matrix fails to be invertible iff 0 is an eigenvalue. 0 is an eigenvalue iff some entry on the diagonal is 0, because the eigenvalues of a triangular matrix are precisely its diagonal entries.

Why are permutation matrices invertible? Group theory promises us that some power of a permutation brings it back to the identity. That is $P^k = I$ for some I. Thus $P^{-1} = P^{k-1}$.

Remark 6.20. Why can we take an LU decomposition and then reduce it, as shown below?



If $m > n$, applying elimination matrices that use row $n + 1$ is a no-op, since the use of row n has (along with prior use of rows $1 \ldots n - 1$) already made

row $n+1$ be entirely 0. or if $n > m$. As a consequence, in the unreduced LU factorization (the first and third pictures above), we see that every column in $\{n+1 \dots m\}$ is really just 0 except at a diagonal entry (where it is 1). As the picture points out, however, determinining what exists in the unreduced L is not useful, since $A = LU$ where $L = [QB]$ and $U = \begin{bmatrix} Q' \\ B' \end{bmatrix}$ where $Q = m \times n$, $B = m \times m - n$, $Q' = n \times n$ and $B' = m - n \times n$. Since $BB' = \mathbb{0}$, there is no need to store B or B'.

Using similar reasoning, we can understand the case that $n > m$.

7 Lecture 7

Least Squares

7.1 Quiz

Problem 1.

Gaussian Elimination with Partial Pivoting Under what conditions will Gaussian elimination with partial pivoting succeed in computing the LU factorization of an n×n matrix A?

Solution 7.1. Always.

Problem 2.

Basic Linear Algebra Subprograms Which of the following is true?

Select all that apply: Most BLAS level 3 functions can be composed out of multiple lower level functions BLAS level 1 functions do not involve matrices BLAS level 2 functions generally do more arithmetic operations per matrix/vector entry than level 3 functions BLAS level 2 functions do not involve vectors

Solution 7.2. Recall that level 1 is vector-vector operations; level 2 is matrix vector operations and level 3 is matrix matrix operations.

Problem 4.

Rank One Change If an n×n linear system Ax=b has already been solved by LU factorization, and then the matrix is changed by adding a matrix of rank 1, how much work is required to solve the new linear system with the same right-hand side?

Solution 7.3. The Sherman Morrison Formula tells us that this cost is $O(n^2)$.

Problem 5.

If we reduce a 9×24 matrix, then the shapes work out to be:

Solution 7.4.

$$(9 \times 9) \times (9 \times 24)$$

Problem 6.

Which problem requires the largest amount of work:
kkkkkk

Remark 7.5. We assume that we work with tall, skinny matrices that have full column rank.

Remark 7.6.

Properties of Least-Squares

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

Yes. $\varphi \geqslant 0$, $\varphi \to \infty$ as $\|\mathbf{x}\| \to \infty$, φ continuous $\Rightarrow$ has a minimum.

Is it always unique?

No, for example if A has a nullspace.

Examine the objective function, find its minimum.

$$\begin{aligned} \varphi(\mathbf{x}) &= (\mathbf{b} - A\mathbf{x})^T(\mathbf{b} - A\mathbf{x}) \\ &\quad \mathbf{b}^T\mathbf{b} - 2\mathbf{x}^T A^T\mathbf{b} + \mathbf{x}^T A^T A\mathbf{x} \\ \nabla\varphi(\mathbf{x}) &= -2A^T\mathbf{b} + 2A^T A\mathbf{x} \end{aligned}$$

$\nabla\varphi(\mathbf{x}) = \mathbf{0}$ yields $A^T A\mathbf{x} = A^T\mathbf{b}$. Called the *normal equations*.

The textbook proves that there is always a unique vector $y \in \text{Span}(A)$ such that $\phi(y) = \|b - y\|^2$ is minimal; as a consequence, there is at least one vector $x \in \mathbb{R}^m$ where A is $\mathbb{R}^{n\times m}$ that minimizes $Ax \approx b$. This vector x is unique iff A is full rank.

Definition 7.7. A matrix P is a projection if $P^2 = P$. A matrix is an orthogonal projection if $P^2 = P$ and $P^T = P$.

Proposition 7.8. If P is an orthogonal projection, then the span of $P_\perp = (I - P)$ is orthogonal to the span of P.

Proof. Given $x, y \in \mathbb{R}^n$, we see that

$$\begin{aligned} &\langle Px, (I-P)y\rangle \\ &= \langle Px, y - Py\rangle \\ &= \langle Px, y\rangle - \langle Px, Py\rangle \\ &= \langle Px, y\rangle - \langle x, Py\rangle \\ &= 0 \end{aligned}$$

□

Corollary 7.9. Given an orthogonal projection P, any vector x can be expressed as $x = Px + P_\perp x$.

Proposition 7.10. The vector x satisfying $\min_{x\in\mathbb{R}^n} \|Ax - b\|_2$ is precisely the x such that $Ax = Pb$ where P is a projection onto A.

Proof. Note that in what follows, all norms refer to the 2 norm.

$$\|Ax - b\| = \|P(Ax - b) + P_\perp(Ax - b)\|$$

Since P and $P_\perp$ map to orthogonal subspaces, we can apply the Pythagorean theorem

$$\begin{aligned} &= \|P(Ax - b)\| + \|P_\perp(Ax - b)\| \\ &= \|P(Ax - b)\| + \|-P_\perp b\| \\ &= \|P(Ax - b)\| + \|P_\perp b\| \\ &= \|(Ax - Pb)\| + \|P_\perp b\| \end{aligned}$$

The RHS is fixed, so we can only minimize the LHS

□

Corollary 7.11. The x aforementioned is $(A^TA)^{-1}A^Tb$

Proof.

$$\begin{aligned} Ax &= Pb \\ \iff A^TAx &= A^TPb \\ \iff A^TAx &= (PA)^Tb \\ \iff A^TAx &= (A)^Tb \\ \iff x &= (A^TA)^{-1}(A)^Tb \end{aligned}$$

□

Proposition 7.12. $P = (A^TA)^{-1}A^T$ is an orthogonal projection, assuming that A has full column rank.

Proof. Verify to yourself that it is symmetric and $P^2 = P$. Also verify that $\text{span}(P) = \text{span}(A)$. □

Corollary 7.13. The x aformentioned is orthogonal to $b - Ax$, the residual.

Proof.

$$b = Pb + P_\perp b$$

Substitute the definition of x and P found above

$$b = Ax + (b - Ax)$$

□

Proposition 7.14. Suppose we know that the columns of $Q \in \mathbb{R}^{m \times n}$ form an orthonormal basis for span(A). Then QQ^T is an orthogonal projector for A.

Proof. $(QQ^T)(QQ^T) = QQ^T$. Thus, this matrix is a projection; is it also clearly symmetric; finally, note that its span is precisely the span of A. □

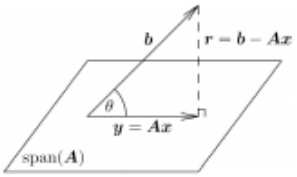
Corollary 7.15. With Q as above and $P = QQ^T$, then the optimal x satisfying $Ax \approx b$ is $Ax = Pb$. Leftmulitply both sides by Q^T to obtain

$$Q^T Ax = Q^T b$$

If we do this, we can avoid the hassle of using the normal equations.

Definition 7.16. I take the following as definitions. No time to look into their proofs:

Sensitivity and Conditioning of Least Squares



Define

$$\cos(\theta) = \frac{\|A\mathbf{x}\|_2}{\|\mathbf{b}\|_2},$$

then

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leqslant \text{cond}(A)\frac{1}{\cos(\theta)} \cdot \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}.$$

What values of θ are bad?

$\mathbf{b} \perp \text{colspan}(A)$, i.e. $\theta \approx \pi/2$.

Sensitivity and Conditioning of Least Squares (II)

Any comments regarding dependencies?

Unlike for $A\mathbf{x} = \mathbf{b}$, the sensitivity of least squares solution depends on both A and $\mathbf{b}$.

What about changes in the matrix?

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leqslant [\text{cond}(A)^2 \tan(\theta) + \text{cond}(A)] \cdot \frac{\|\Delta A\|_2}{\|A\|_2}.$$

Two behaviors:

- If $\tan(\theta) \approx 0$, condition number is $\text{cond}(A)$.
- Otherwise, $\text{cond}(A)^2$.

8 Lecture 8

Problem Transformations

8.1 Quiz

Problem 1. Polynomial Data Fitting If a first-degree polynomial x1+x2t is fit to the three data points (1,1), (2,1), (3,2), by linear least squares, what are the resulting values of the parameters x1 and x2?

Choice* x1=1, x2=0 x1=1/2, x2=1/2 x1=1/3, x2=1/2 x1=−1, x2=1

Solution 8.1. The solution is $(A^TA)^{-1}A^Tb$. Remember that the inverse of a general matrix is x^2.

Proposition 8.2. $k(A^TA) = k(A)^2$.

Proof.

- Recall that $\|A\| = \max_\sigma(A)$.
- $(A^TA)^T(A^TA) = (A^TA)^2$.
  - It follows that $\left\|A^TA\right\| = \|A\|^2$.
- Let $\Sigma(A)$ be the set of eigenvalues of A^TA. Strangely, it is difficult to argue that if $\lambda \in \Sigma(A)$, then $\frac{1}{\lambda} \in \Sigma(A^{-1})$. It can be argued, however, that $\lambda \in \Sigma(A^TA) \iff 1/\lambda \in \Sigma(A^TA)^{-1}$. Thus the smallest singular value of A^TA is, when reciprocated, the largest singular value of A^TA^{-1}.
  Not sure why this is significant. Crap

□

Remark 8.3. It seems that using A^TA is also disadvantageous insofar as if matrix looks like say

$$\begin{bmatrix} 1 & 0 \\ \epsilon & 0 \\ 1 & \epsilon \end{bmatrix}$$

and we compute A^TA then we will get a term that may be $1 + \epsilon^2$. Suppose that $\epsilon < \sqrt{\epsilon_{mach}}$. Then the term will end up being 1.

Remark 8.4. If Q is orthogonal, then $\|v\| = \|Qv\|$.

8.2 Householder Transformations

Remark 8.5. Recall that a matrix is orthogonal iff its transpose is its inverse. This means that both the rows and columns of the matrix, say Q, constitute an orthonormal set: the vectors are pairwise orthonormal, and they each have unit length.

Remark 8.6. Suppose that we obtain a system of the form:

$$\begin{bmatrix} R \\ 0 \end{bmatrix} x = Q^Tb \coloneqq \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Then we can solve for $Rx = c_1$ if R is invertible but not for c_2; as a consequence, the minimal norm that we can obtain when we ssolve this system is c_2^2.

Definition 8.7. A QR factorization expresses an $m \times n$ matrix A where $m \geq n$ as QR where Q is $m \times m$ and R is $m \times n$. This often better expressed as

$$\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where Q_1 is $m \times n$, Q_2 is $m \times (m-n)$; R is $n \times n$ and 0 represents a $(m-n) \times n$ block. The reduced QR factorizaton is defined to be Q_1R.

Remark 8.8. A natural question raised is: what is the purpose of Q_2? Q_2 is the orthogonal complement of Q_1 – it is sometimes helpful to remember this. Since there are many ways to identify the orthogonal complement of Q_1, Q_2 is not unique. It can be shown, however, that Q_1R is unique if we force the diagonal entries of R to be positive

Q_1R is also unique up to multiplication of the diagonal entries of R by -1 and corresponding multiplication of a column in Q_1 by -1 (in particular, if $R_{i,j}$ is multiplied by -1, then $R_{i,j'}$ for $j' \geq j$ must be multiplied by -1 and column i of Q_1 must be multiplied by -1.

Remark 8.9. Observe that many methods to compute a QR factorization rely on forming Q multiplying successive orthogonal matrices like

$$Q_n Q_{n-1} ... Q_1$$

To obtain Q, we need to use all of the left factor. By this, I mean that if we compute Q_2Q_1, then while we can drop columns of Q_1 past the nth column, we need to use all of Q_2 when performing the multiplication.

<++>

9 Lecture 9

Problem Transformations II – February 5th

Definition 9.1. Given a matrix v, a projection matrix onto span v is $\frac{vv^T}{v^Tv}$. This is unique, I think. In any case, this projection matrix is symmetric and satisfies $P^2 = P$, making it an orthogonal transformation.

Proposition 9.2. A householder transformation finds the normal vector v such that if α is reflected across the plane whose normal is given by v, then the resulting vector is nullified in all but the first k components. The projection matrix is given by

$$I - 2\frac{vv^T}{v^Tv}$$

Proof.

$$P = \frac{vv^T}{v^Tv}$$

is the projection matrix that projects onto span(v). We established that P is an orthogonal transformation; therefore, $(I - P)x$ will project onto the set $\{x|v^Tx = 0\}$, which is the plane whose normal is v. Multiplication by $I - P$ amounts to travelling from x and then onto this plane; we now need to travel once more to attain a reflection. Thus, the projection that will allow us to attain the form is

$$I - 2\frac{vv^T}{v^Tv}$$

□

Proposition 9.3. It can be shown that the vector v which allows us to zero the last $m-k$ components of an m vector

$$a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

where a_1 is $k-1$, a_2 is $m-k+1$ and $1 \leq k < m$ is

$$\begin{bmatrix} 0 \\ a_2 \end{bmatrix} - \alpha e_k$$

where $\alpha = -sign(a_k)\|a_2\|_2$.
The choice in sign for α reflects our desire to avoid cancellation.

Definition 9.4. The matrix that rotates a vector θ degrees counter clockwise in the $\mathbb{R}^2$ plane is

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Note that since sin is an odd function ($f(-x) = -f(x)$) and cos and even function ($f(-x) = f(x)$), it follows that the matrix rotating an angle θ degrees clockwise is

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Remark 9.5. We are interested in finding the angle θ and hence the value of $\cos(\theta)$ and $\sin(\theta)$ that solves the problem:

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sqrt{a_1^2 + a_2^2} \\ 0 \end{bmatrix}$$

We will call the resulting matrix M that performs the transformation.
Note that the answer to this question is

$$c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}$$

If we had instead solved (note the difference in the matrix M).

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sqrt{a_1^2 + a_2^2} \\ 0 \end{bmatrix}$$

the resulting values of c and s would change, but that is okay – so long as we redefine M to be this matrix with the solved values for c and s.

Remark 9.6. Suppose that we wish to rotate the b entry of some column onto the a entry of the same column. Suppose that we have solved for the matrix M that accomplishes this transformation where the b entry is a_2 and the a entry is a_1. Then we can construct a givens transformation:

A givens rotation imbeds M into a matrix A with ones (initially) along the diagonal so that $A_{a,a} = M_{0,0}$, $A_{a,b} = M_{0,1}$, $A_{b,a} = M_{1,0}$, $A_{b,b} = M_{1,1}$.

10 Lecture 10

SVD

Definition 10.1. Every matrix admits a decomposition:

$$A = U\Sigma V^T$$

The reduced SVD is the same decompositon but Σ is reduced to be the small as possible.

Entries of U are called left singular vectors, entries in Σ are singular values. Entries in V^T can

Theorem 10.2. If $\|A\| = \|\Sigma\| = \sigma_1$ where σ_1 is the largest diagonal entry in Σ.

Remark 10.3. If a singular value appearing in Σ is negative, can it be made positive?

Solution 10.4. Yes, flip an appropriate singular vector in sign. UNRESOLVED.

Proposition 10.5. Take it for granted that $\|A\|_2 = \sigma_1$. Given this, $k(A) = \frac{\sigma_1}{\sigma_{\min m,n}}$

Proof. Recall that we have redefined $k(A)$ to now be

$$\|A\|\|A^+\|$$

We have also found that $A^+ = V\Sigma^+U^T$, from which it follows that $\|A^+\|$ is the greatest diagonal entry of Σ^+ which is the reciprocal of the smallest diagonal entry in Σ. □

Proposition 10.6. The null space of V^T is given by the rows of V^T corresponding to the singular values of A (ie the values in Σ) that are 0.

Proof. Let these rows be collected in the set V. We argue that $V \subseteq \mathcal{N}(A)$. Just hit A by each $v_i \in V$.

Recall that $A = U\Sigma V^T$ which is really (assuming that A is an $m \times n$ matrix)

$$[u_1 \dots u_n] \begin{bmatrix} \sigma_1 & \dots & \\ & \sigma_2 \dots & \\ & & \sigma_3 \dots \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$$

from which it follows that

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

Recall that V_i form an orthogonal basis for $\mathbb{R}^n$. Observe that Av_i where $\sigma_i = 0$ results in 0. On the other hand, Av_i where $\sigma_i \neq 0$ is non-zero. Thus, we have determined which of a basis' vectors result in annihilation. This has determined the null space. □

Proposition 10.7. The rank of A is given by the number of singular values that are not zero.

Proof.

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

tells us that to obtain any vector in the column space of A we only need to have access to the u_i such that $\sigma_i \neq 0$. However many u_i exist is the rank of the matrix. □

Remark 10.8. Rank is not robust to rounding error. Suppose we have a rank one matrix; then introduce rounding error (each entry in the matrix has ϵ_{mach} added or subtracted from it. By doing this, it is conceivable that the rank of the matrix is changed significantly. Far better, as a consequence, is to compute the numerical rank, which asks how many singular values fall above a tolerance. That is, for $\sigma \in \Sigma$, determine whether $|\sigma| > \epsilon$.

Theorem 10.9. Eckhart Young Mirsky The best k rank approximation to A is given by

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

Remark 10.10. Convince yourself that the summation above is the matrix that one obtains by, alternatively, zeroing all but the first k diagonal entries of Σ and then multiplying out $U\Sigma V^T$.

Proposition 10.11. In the circumstance that A^TA is invertible, the inverse $(A^TA)^{-1}A^{-1}$ agrees with the inverse obtained from the SVD $V\Sigma^+U^T$.

Proof.

$$\begin{aligned} & A = U\Sigma V^T \\ \implies (A^TA)^{-1}A^{-1} &= (V\Sigma U^T U\Sigma V^T)^{-1} V\Sigma^+ U \\ &= V\Sigma^+ U \end{aligned}$$

□

Example 10.12. To compute the pseudoinverse of a matrix like

$$\begin{bmatrix} a & & \\ & b & \\ & & c \\ 0 & 0 & 0 \end{bmatrix}$$

the foregoing establishes that we need not compute the SVD and then obtain the pseudoinverse by re-arranging the resultant factors. We can compute the SVD by also computing $({}^{-1}A^TA)A^T$ which, in this case, gives us

$$\begin{bmatrix} 1/a & & \\ & 1/b & \\ & & 1/c \\ 0 & 0 & 0 \end{bmatrix}$$

Remark 10.13. Using a k rank approximation is not unambiguously good, for computation of the SVD requires $O(n^3)$ time.

Unresolved 10.14. If A_k is the best k rank approximation to A, is it the case that

$$\|A_k - A\|$$

is the norm of the matrix obtained by zeroing out the first k singular values of A.

Theorem 10.15.

Proof.

$$\begin{aligned} U\Sigma V^T x &\cong b \\ \Sigma(V^T x) &\cong U^T b \\ \Sigma y &\cong U^T b \end{aligned}$$

Solve for y

$$\begin{aligned} y_i = (U^T b)_i/\sigma_i \text{ for } i \in [k] \\ y_i = 0 \text{ ow} \end{aligned}$$

Then solve for x where $V^T x = y$. Note that y is the optimal of all vectors $\hat{y}$ that solve $\Sigma\hat{y} \cong U^T b$. It follows that x is the minimal of all vectors that solve $Ax \cong b$ since $x = Vy$, meaning that $\|x\|_2 = \|y\|_2$. □

Definition 10.16. The solution to the total least squares problem
is $V\Sigma^+ U^T b$ where Σ^+ is computed by reciprocating singular values in their spots (save those singular values that are 0.

Remark 10.17. Remember the cost of householder for nonsquare and perhaps the cost for all $n \times n$ matrix.

11 Lecture 11

Eigenvalue Problems

Definition 11.1. ALgebraic multiplicity of an eigenvalue counts the number of times the eigenvalue occurs as a root of the characteristic equation. Geometric multiplicity counts how many linearly independent eigenvectors correspond to an eigenvalue. It is a proveable fact that algebraic multiplicity is at least as much as the geometric multiplicity. If algebraic multiplicity is greater than geometric multiplicity, we say that the matrix is defective.

Theorem 11.2. Similar matrices share the same eigenvalues.

Theorem 11.3. If a matrix is defective, then it cannot have an eigenvector basis.

Proposition 11.4. A matrix is diagonalizable iff it has an eigenvector basis.

Proposition 11.5. The following matrix transformations change eigenvalues and eigenvectors as follows:

- $A \to (A - \sigma I)$ causes $\lambda \to \lambda - \sigma$.
- $A \to A^{-1}$ causes $\lambda \to \frac{1}{\lambda}>$
- $A \to A^k$ causes $\lambda \to \lambda^k$.
- If $A = PXP^{-1}$, then X has the same eigenvalues but every eigenvector v now becomes $P^{-1}v$.

Proposition 11.6. Suppose that we perturb a diagonal matrix A with some matrix E. Then the distance between any eigenvalue u of $A+E$ to an eigenvalue of A λ_k closest to u is bounded by $k(A)\|E\|$.

12 Lecture 12

Problem C. haracteristic Polynomial For which of the following reasons is the characteristic polynomial of a matrix NOT useful, in general, for computing the eigenvalues of the matrix?

Select all that apply: Its coefficients may not be well determined numerically. Its roots may be difficult to compute. Its roots may be sensitive to perturbations in the coefficients. None of these

Solution 12.1. Yes, the roots change if the coefficients change – hence any perturbation will affect the roots; determining the coefficients numerically is also problematic, simply because numerical comptutation requires rounding error and truncation error. The second is a given.

Problem P. roblem Transformations and Spectral Radius Which of the following transformations preserve the spectral radius of a matrix A?

Select all that apply: Powers None of these Shift Polynomial Inversion

Solution 12.2. Obviously, none of them, since they all change the eigenvalues.

Problem D. iagonalizability Of the classes of n×n matrices listed below, which is the smallest class of matrices that are not necessarily diagonalizable by a similarity transformation?

Choice* normal matrices all matrices real symmetric matrices matrices with n distinct eigenvalues

Solution 12.3. A spectral theorem asserts that normal matrices are unitarily diagonalizable; a theorem asserts that real symmetric matrices have an orthogonal basis – hence they are diagonalizable; in general, any matrix with an eigenbasis is diagonalizable.

Problem L. et $A = XDX^{-1}$. Suppose $\hat{A} = A + \delta A = \hat{X}\hat{D}(\hat{X})^{-1}$. Which matrix is $\hat{A}$ similar to?

Solution 12.4. $X^{-1}\hat{A}X = X^{-1}AX + X^{-1}\delta AX = D + X^{-1}\delta AX$

Definition 12.5. The eigenvector corresponding to the largest eigenvalue will hence forth be called the maximal eigenvector.

Remark 12.6. Power iteration can fail because of certain reasons; certain of these can be remedied:

- No component alongside the dominant eigenvector.
  - Rounding error usually introduces some component – and a random vector will usually include the component.
- Overflow of entries in the vector being iterated upon:
  - Normalize the vector
- There is no one dominant eigenvector, because two distinct eigenvalues of equal, maximal magnitude exist.

Definition 12.7. The rayleight quotient is the quantity

$$\frac{x^T A x}{x^T x}$$

Remark 12.8. Let $e_k = \left\| v_1^k - x_1 \right\|$ where v_1^k is the estimate of x_1 at the kth iteration.

It can be shown that error $\approx c\left|\frac{\lambda_2}{\lambda_1}\right|^k$, which implies that

$$\frac{\|e_{k+1}\|}{\|e_k\|} = \left|\frac{\lambda_2}{\lambda_1}\right|$$

This is called linear convergence, because the error in the next iteration is a linear (think $y = ax$) scaling of the previous error.

Power iteration obviously costs $O(n^2)$ at each iteration, because we're repeatedly multiplying a matrix by a vector.

Definition 12.9. Inverse power iteration is the algorithm that repeatedly hits x with A^{-1}. Its statistics:

- The error rate is

$$\frac{|1/\lambda_{n-1}|}{|1/\lambda_n|} = \frac{|\lambda_n|}{|\lambda_{n-1}|}$$

- The cost is initially $O(n^3)$ since we solve for y in $Ay = x$. Thereafter, the cost is $O(n^2)$.
- The largest eigenvalue is $\left|\frac{1}{\lambda_n}\right|$

Definition 12.10. Shifted inverse power iteration iterates on x using $(A - \sigma I)^{-1}$.

The dominant eigenvector corresponds to the eigenvalue $\frac{1}{\lambda' - \sigma}$ where λ' is the closest eigenvalue of A to σ. If λ'' is the second closest eigenvalue to A then the error rate is given by

$$\frac{|\lambda' - \sigma|}{|\lambda'' - \sigma|}$$

The cost is initially $O(n^3)$ but then $O(n^2)$ thereafter.

Definition 12.11. Shifted Rayleigh iteration leverages both power iteration and the Rayleight quotient method in order to determine the maximal eigenvector:

- Compute the rayleigh quotient to obtain σ_k.
  - Multiply $(A - \sigma_k I)^{-1}$ by x_k.
    - In reality, by multiplication, we mean compute the LU factorizaton of $A - \sigma_k I$ and then use it
- This will presumably give us an eigenvector that corresponds to the eigenvector closest to σ.
  - Note that σ was not chosen with any prior intent however – it just happens to be our estimate of the eigenvalue that corresponds to some initial x, and that initial x is our estimate of some eigenvector. to compute the multiplication.
- This is good, because it can be shown to converge at a quadratic rate – whereas ordinary power iteration convergence is linear.
  - This is bad, because we have to factor $A - \sigma_k I$ every iteration, which can be expensive.
  - This may not also converge to an eigenvector we want to converge to apriori.

Definition 12.12. Simultaneous iteration seeks to determine the maximal p eigenvectors; or rather a basis for them.

Initialize some X_0, an $n \times p$ matrix.
$X_{k+1} = AX_k$
⟨⟨ You can perform orthonormalization if needbe ⟩⟩

Remark 12.13. This algorithm is problematic, because the column vectors in X_k may tend to the dominant eigenvector. As a result, X_k becomes increasingly more and more ill conditioned. This is bad, in and of itself, because we typically desire that $X_k \to X$, which we define to be a basis for the span of the dominant p eigenvectors.

Definition 12.14. One remedy to this problem is to orthonormalize the iterate of the basis. The resulting algorithm is called orthonormal iteration.

In what follows, assume that we use the full unreduced QR factorization.

Initialize some Q_0, corresponding to our guess of a basis for the dominant p eigenvectors.
Q_0 is an orthogonal matrix that is $n \times n - p$
While $\|Q_{k+1} - Q_k\| >$ some tolerance
  $X_{k+1} := AQ_k$
  $Q_{k+1}R_{k+1} := X_{k+1}$

Notice that at each iteration, we have $A = Q_{k+1}R_{k+1}Q_k^T$. If $Q_k \to Q$ then $A = QRQ^T$ (R also converges as a consequence), so that we have found some (allegedy dominant) p eigenvectors in Q and their eigenvalues in the $p \times p$ upper triangular R.

Remark 12.15. Assume now that we use the reduced QR factorization so that Q is $n \times r$ and R is $r \times r$.

The span of Q_k is the span of X_k at any point. Thus $X_{k+1} = AQ_k$ gives us a set of column vectors whose span is the same as the span of AX_k. Thus, we can use Q_k as a proxy to X_k.

Why is span(Q_k) =span(X_k) – since otherwise, the QR factorization of X_k (which we assume to be of rank r – or else – we are not determining distinct eigenvectors) will reduce the rank of X_k.

Definition 12.16. In QR iteration, we decompose A_k as Q_kR_k and then obtain $A_{k+1} = R_kQ_k$. Notice that $A_{k+1} = Q_k^TA_kQ_k$ so that if Q_k converges then A_{k+1} is now (this is not yet underestood) really an upper triangular matrix with eigenvalues along its diagonal.

Remark 12.17. It is instructive to look at the iterates of simultaneous iteration and QR iteration:

$$\begin{aligned}
\hat{Q}_1R_1 &= X_0 \\
X_1 &= A\hat{Q}_1 \\
\hat{Q}_2R_2 &= X_1 \\
X_2 &= A\hat{Q}_2 \\
\hat{Q}_3R_3 = X_2 \implies \hat{Q}_3R_3\hat{Q}_2^{\ H} = A &\implies \hat{Q_{k+1}}R_{k+1}\hat{Q}_k^{\ H} = A
\end{aligned}$$

QR iteration:

$$\begin{aligned}
Q_1R_1 = A_0 &= A \\
A_1 &= R_1Q_1 \\
Q_2R_2 &= A_1 \\
A_2 &= R_2Q_2 \\
\implies A_2 &= Q_2^HA_1Q_2 \\
\implies A_{k+1} &= Q_{k+1}^HA_kQ_{k+1}
\end{aligned}$$

There is an equivalence between the two forms of iteration that can be expressed as follows:

Suppose that we have $\hat{Q_{k-1}}$ and that we wish to form $X_{k-1} = A\hat{Q_{k-1}}$ and then factorize the resultant product as $\hat{Q}_kR_k$. Assume further that have already factorized A_{k-1} as Q_kR_k. Then we can compute X_{k-1} without any work that we would normally do for simultaneous iteration:

$$X_{k-1} = A\hat{Q_{k-1}} = \hat{Q_{k-1}}\hat{Q_{k-1}}^{\ H}A\hat{Q_{k-1}} = \hat{Q_{k-1}}A_{k-1} = Q_{k-1}\hat{Q}_kR_k$$

If we set $\widehat{Q_k} \coloneqq \widehat{Q_{k-1}}Q_k$, and appeal to the uniquess of QR, then we are done.

Definition 12.18. In QR iteration with shifting, we factor $A_k - \sigma_k I$ as $Q_k R_k$ and then compute

$$A_{k+1} = R_k Q_k + \sigma I$$

Note that since $R_k = Q_k^T(A_k - \sigma_k I)$, it follows that

$$A_{k+1} = Q_k^T(A_k - \sigma_k I)Q_k + \sigma I = Q_k^T A_k Q_k$$

Remark 12.19. Once again, it is instructive to see what the iterates of QR iteration with shifting look like:

$$\begin{aligned} Q_1 R_1 &= (A_0 - \sigma_0 I) \\ A_1 &= R_1 Q_1 + \sigma_0 I \\ \implies A_{k+1} &= Q_{k+1}^H (A_k - \sigma_k I) Q_{k+1} + \sigma_0 I \\ &= Q_{k+1}^H A_k Q_{k+1} \end{aligned}$$

Definition 12.20. The Schur decomposition of a matrix $A = QUQ^H$ expresses A as similar to an upper triangular matrix U. A proof of this theorem is given in `http://people.inf.ethz.ch/arbenz/ewp/Lnotes/chapter2.pdf` at page 6. We now list some properties pertaining to the matrix:

- Since $AQ = QU$, we can conclude that $AQ_1 = Q_1(U_{1,1})$. From this, it follows that the first schur vector Q_1 is an eigenvector.
- If Q and U are real, then QU is a QR decomposition – obviously.
- If A is real symmetric, then it must be true that U is diagonal, in which case it contains all the eigenvalues of A.
  - Moreover, since $U = Q^H AQ$, one can shown that $U^* = U$, implying that all the eigenvalues are real. A proof near the proof listed above additionally explains that eigenvectors corresponding to distinct eigenvalues are orthonormal.

Proposition 12.21. With the schur form of a matrix, we can also use the triangular matrix U to construct eigenvectors.

Proof. Subtract λI from U. Then we get

$$\begin{bmatrix} U_{1,1} & u & U_{1,3} \\ & 0 & v^T \\ & & U_{3,1} \end{bmatrix}$$

Note that $U_{1,1}$ is also triangular, u is a column vector, $U_{1,3}$ a rectangular block, v a column vector and $U_{3,1}$ another triangular matrix. Then

$$\begin{bmatrix} -U_{1,1}^{-1}u \\ 1 \end{bmatrix}$$

is an eigenvector. □

13 Lecture 13

Problem S. uppose we are given the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 3 & 0.1 \end{bmatrix}$$

Then how many power iterations do we need in order to reduce the error by 10^{-10}.

Solution 13.1. The convergence rate is given by $\left|\frac{\lambda_2}{\lambda_1}\right|$; hence the rate is 10^{-1} and we need at least 10 iterations. Note that we additionally assume that the starting vector has components in both directions corresponding to the eigenvalue.

Problem A. ssume that we are given an LU factorization from a previous computation. What is the cost of performing inverse power iteration for k iterations on an $n \times n$ matrix.

Solution 13.2. The cost is $O(kn^2)$. Recall that LU factorizations, once calculated, allow us to solve inverse problems in $O(n^2)$ time.

Problem E. stimate the rayleight quotient of a given matrix for a given eigenvector. Trivial; whence the solution is ommitted.

Problem P. roperties of the Schur Form Consider the Schur decomposition of a matrix A=QUQH where U is upper-triangular and Q is unitary. The Schur vectors are the columns of Q. While of the following are true?

Select all that apply: If A is real, Q is orthogonal and U is real The first Schur vector is an eigenvector of A The Schur form of a matrix is unique so long as its eigenvalues are not all the same If A is real symmetric, the Schur decomposition is the eigenvalue decomposition of A If Q and U are real, QU is a QR decomposition of AQ

Solution 13.3.

- If A is real, Q is orthogonal and U is real
  - A coudl be real but Q and U still complex.
- The first Schur vector is an eigenvector of A
  - True – see above.
- The Schur form of a matrix is unique so long as its eigenvalues are not all the same
  - In the proof of the schur decomposition, the eigenvalue (which is entry occupying the first diagonal entry of Λ) was chosen without any specificity – so that any eigenvalue can be chosen there.
- If A is real symmetric, the Schur decomposition is the eigenvalue decomposition of A
  - Yes, see above.
  - If Q and U are real, QU is a QR decomposition of AQ.
  - Yes, can be observed from the form of the schur decomposition itself.

Problem Q. R Iteration for Special Matrices Which of the following statements about doing one iteration of QR decomposition on a matrix An×n is true?

Select all that apply: For a Hessenberg matrix A, it takes O(n2) time. For a Hessenberg matrix A, it takes O(n3) time. For a tridiagonal matrix A, it takes O(n) time. For a tridiagonal matrix A, it takes O(n2) time.

Solution 13.4. Imagine that we use Givens' rotations. Take the kth column as an example. After finding the rotation that takes the $k+1$th row to the kth row, we need to apply this same rotation to the remaining $n-k$ columns to the right of the kth column. Thus, even if the givens' rotation takes $O(n)$ time to both compute and apply its effects to the $k$the column, the cost of its application to the remaining columns results in $O(n^2)$. By contrast, in a tridiagonal matrix, after finding the rotation that "settles" the kth column, the rotation only need be applied to the $k+1$th column, since the rotation will change the kth entry of that column; no other column to the right will be affected. Thus, this results in an $O(n)$ cost.

Remark 13.5. To arrive at the schur form of a matrix, we need to perform some unspecified number of QR factorizations. Each factorization takes $O(n^3)$. Thus the cost can often near $O(n^4)$.

Remark 13.6. A better technique is to use a trick involving Householder Transformations:

- Having computed a householder vector h_i that nullifies everything beneath and including row $i+2$, and having formed H_i which is the transformation that achieves this, carry out $H_i A (H_i)^T$. What will this do? It can be shown that this will zero out all entries past and including the $i+2$ row. Doing this for each column, we will attain Hessenberg form. Having attained Hessenberg form. That is $A = \prod_{i=1}^{n} (H_i)^T H (\prod_{i=1}^{n} H_i)$,which is a similarity transform to A, meaning that the eigenvalues for A are contained in H. We can now perform QR iteration on H. It costs $O(n^2)$ to perform the necessary number of Givens rotations to QR factorize the matrix. Suppose that we obtain at the very first iteration the factors $Q_1 R_1 = H$. Observe that $A_2 = R_1 Q_1 = R_1 H R_1^{-1}$. Further reacall that post or pre multiplying a Hessenberg matrix by an upper triangular matrix preserves the form of the Hesenberg matrix. It therefore follows that A_2 is Hessenberg as well; thus every QR iteration now requires $O(n^2)$ as opposed to $O(n^3)$.

Remark 13.7. In a Krylov subspace method, we restrict our focus to finding a few eigenvect

<++>

Definition 13.8.

- Suppose that we begin with an initial vector x_0. Let

$$K_n = [x_0 \quad Ax_0 \quad ... \quad A^{n-1}x_0]$$

Observe that

$$AK_n = \begin{bmatrix} Ax_0 & A^2x_0 & \dots & A^nx_0 \end{bmatrix} = K_n \underbrace{\begin{bmatrix} e_2 & e_3 & \dots & e_n & K_n^{-1}A^nx_0 \end{bmatrix}}_{H}$$

That is we found that $K_n^{-1}AK_n = H$, where H is upper Hesenberg.

Remark 13.9. So what? Why is this important? We wish to work with H to determine its eigenvalues. The foregoing reveals that we will need to compute K_n if we wish to compute H. The problem is that K_n will tend to multiples of the dominant eigenvector of A, since K_n is obtained by performing power iteration a few times on x_0. This makes K_n highly ill conditioned, so that the Hesenberg form that we would obtain by mat-mat multiplication is likely not useful. What we will therefore do is that we will find a Q_nR_n factorization for K_n.

Remark 13.10.

$$Q_n^HAQ_n = (K_nR_n)^{-1}A(K_nR_n) = (R_n)^{-1}K_n^{-1}K_nK_nR_n = (R_n)^{-1}HR_n \cong C$$

17

where C is some Hesenberg matrix in addition to H.

Proposition 13.11. The vectors of Q_n can each be computed alone.

Proof.

$$AQ_n = Q_nC$$

For notational convenience, let $Q = Q_n$

Let us focus on the jth column. Suppose that $Q_n =$

$$\begin{bmatrix} q_1 & q_2 \dots q_n \end{bmatrix} \implies Aq_j = \sum_{i=1}^{j+1} q_iC_{i,j}$$

Now solve for q_2 assuming that we know q_1 and that we have orthonormalized q_1. This gives us:

$$\begin{aligned} &\implies Aq_1 = \sum_{i=1}^{2} q_iC_{i,1} \\ &\implies Aq_1 = q_1C_{1,1} + q_2C_{2,1} \\ &\implies Aq_1 - q_1C_{1,1} = q_2C_{2,1} \end{aligned}$$

Note that Aq_1 is the "next" vector in a Krylov subspace that one would compute with knowledge of q_0.

Now in general $C_{i,j} = q_i^H A q_j$. In particular $C_{1,1} = q_1^H A q_1$. If we let $u_1 = (Aq_1)$, then we see that $C_{1,1} = q_1^H u_1$ and that

$$Aq_1 - q_1 C_{1,1} = u_1 - q_1 C_{1,1}$$

This is tantamount to orthogonalizign u_1 by removing all of the projections of u_1 onto previous q_k (in this case, only q_1). Thus if we set $q_2 = \frac{u_1}{\|u_1\|}$, then we will have effectivel found the vector q_2 that is part of the orthonormal set of Krylov space vectors. □

Remark 13.12. This can be formalized as part of an algorithm:

Algorithm 4.9 Arnoldi Iteration

```
x_0 = arbitrary nonzero starting vector
q_1 = x_0/||x_0||_2                    { normalize }
for k = 1, 2, ...
    u_k = A q_k                        { generate next vector }
    for j = 1 to k                     { subtract from new vector
        h_jk = q_j^H u_k                   its components in all
        u_k = u_k - h_jk q_j               preceding vectors }
    end
    h_{k+1,k} = ||u_k||_2
    if h_{k+1,k} = 0 then stop         { stop if matrix is reducible }
    q_{k+1} = u_k/h_{k+1,k}            { normalize }
end
```

Remark 13.13. Note that this algorithm provides us with a means of obtaining the Hesenberg matrix that the original matrix A is orthogonally similar to. The vectors q are not so important save that we need them in order to determine the values in the matrix C (which stands for the Hesenberg matrix).

This is done as follows: the crux of this algorithm is the equation

$$Aq_k = \sum_{i=1}^{k+1} C_{ik} q_i$$

Assuming that we already know q_j for $j \leq k$, it is our goal to determine C_{ik} for all $i \leq k+1$. To do this, we observe that $q_i^H A q_k = C_{ik}$. Thus, we know all values of C_{ik} where $i \leq k$. To determine $C_{k+1,k}$, observe that

$$\underbrace{Aq_k - \sum_{i=1}^{k} C_{ik} q_i}_{\gamma} = C_{k+1,k} q_{k+1} = \underbrace{(Aq_k) - \sum_{i=1}^{k} q_i^H (Aq_k) q_i}_{\gamma} = C_{k+1,k} q_{k+1}$$

Since q_{k+1} is destined to be part of an orthogonal matrix, we know that $C_{k+1,k}$ is $\|\gamma\|$ and that q_{k+1} is $\gamma/\|\gamma\|$.

Observe that the algorithm above unfolds like Gram Schmidt in that we take the next vector Aq_k and then subtract away the projection of this vector onto previous q_i where $i \leq k-1$.

Proposition 13.14. We can compute the eigenvectors corresponding to the k greatest eigenvalues in modulus.

Proof. We found that $Q_n^h A Q_n \cong H$, an upper Hesenberg matrix. Define

$$Q_k = [q_1 \quad \dots q_k]$$

to be the $n \times k$ matrix that consists of the first k Arnoldi vectors. Define

$$U_k = [q_{k+1} \quad \dots q_n]$$

to be the matrix consisting of the remaining k uncomputed vectors. It follows that

$$Q_n^H A Q_n = \begin{bmatrix} Q_k^h \\ U_k^h \end{bmatrix} A \begin{bmatrix} Q_k & U_k \end{bmatrix}$$

□

14 Lecture 14

Nonlinear equations

Problem Q. R Iteration Convergence For which of the following classes of matrices will QR iteration always converge and produce all the eigenvalues?

Select all that apply: Symmetric matrix Diagonal matrix Orthogonal matrix Upper Triangular matrix General matrix with complex eigenvalues

Solution 14.1. Stupid trick question. The answer is a diagonal matrix and an upper triangular matrix. For we don't need to invoke the QR algorithm on these matrices; we already know their eigenvalues (just read the diagonal).

QR is never guaranteed to always converge (ie in a finite number of steps), because there is proveably no algorithm that can give us the eigenvalues for matrices with length at least 5.

Problem K. rylov Subspace Conditioning Given a matrix A with condition number =100 and an initial vector x0, how many additional Krylov vectors can be calculated while ensuring the relative error of each vector remains less than or equal to 1×10−6 assuming IEEE double precision.

Solution 14.2. The answer is 5. The relative error is bounded by the condition number multiplied by the relative input error. At the kth iteration the relative input error is the relative output error of the $k-1$th iteration where the relative input input error of the 0th iteration is given by machine epsilon, ie $\approx 2 \times 10^{-16}$. Therefore, relative output error at iteration k is $100^k 2 \times 10^{-16}$, meaning that we can let k be at most 5.

Problem S. uppose A is a general n×n matrix and Q=[q1q2…qk] is an orthogonal matrix with qjKj(A,b), the Krylov subspace associated with A. How many nonzero elements are in the matrix QTAQ?

Solution 14.3. See the notes previously that explain that if we gather the fist k Arnoldi vectors into a matrix Q then $Q^T A Q$ is upper Hesenberg and $k \times k$, meaning it has k^2 non zero entries.

Problem Q. k is the matrix obtained by orthogonalizing the columns of Ck=[bAbA2b…Ak−1b]. Which of the following is equal to QTkb?

Solution 14.4. Q_k will contain column vectors such that only the first vector q_1 is orthonormal with b. In fact that $\langle q_1, b\rangle = \langle q_1, \frac{b}{\|b\|}\|b\|\rangle = \|b\|\langle q_1, \frac{b}{\|b\|}\rangle = \|b\|e_1$.

Problem C. oding question:

Solution 14.5. Here the solution is to recall that once we factorize $K_n = Q_n R_n$ then the upper Hesenberg matrix $K_n^T A K_n$ can be alternatively be expressed as

$$Q_n^T A Q$$

since this simplifies to

$$R_n K_n^{-1} A K R_n$$

which is still upper Hesenberg.

Question 14.6. Why do we favor symmetric problems in terms of condition number?

Remark 14.7. Arnoldi is a way to get eigenvalues especially if you cannot store the entire matrix. Why? Since if we compute $Q_k^T A Q_k$, we get a matrix consisting of the first k ritz eigenvalues, which approximate the k eigenvalues greatest in modulus.

Proposition 14.8. We can compute the SVD of a square matrix:

Proof. The following is a naive way of computing the SVD:

Suppose that we have already obtained the eigenvalues and eigenvectors of $A^T A$ as part of a matrix V. Note that the eigenvectors constitute an eigenbasis, since $A^T A$ is symmetric and thus has an eigenvector basis. It follows that $A^T A$ is diagonalizable. That is we have:

$$A^T A = V \Sigma^2 V^T$$

We know that the diagonal matrix can be represented as the square of a matrix, because $A^T A$ is positive semidefinite and, hence, has non-negative eigenvalues.

$$A = U \Sigma V^T \implies U = A V \Sigma^{-1}$$

Indeed, we will find that U is orthogonal, since

$$U^T U = \Sigma^{-1} V^T A^T A V \Sigma^{-1} = I$$

Note that we seem to have assumed that $A^T A$ is full rank.

□

Definition 14.9. A non linear equation $f : \mathbb{R}^n \to \mathbb{R}$ is solved when we find the tuple x such that $f(x) = 0$. If we are given a function $g : \mathbb{R}^n \to \mathbb{R}$ and we want to find the x such that $g(x) = c$, that we can alternatively find the x such that $f(x) = g(x) - c = 0$.

Remark 14.10. We have three tools in order to assert the existence of a solution.

- The intermediate value theorem tells us that if $f(a) < 0$ and $f(b) > 0$ and f is continuous, then there is a c such that $a < c < b$ such that $f(c) = 0$.
- The inverse function theorem tells us that if $f : \mathbb{R}^n \to \mathbb{R}$ and the Jacobian of f is non-singular at a point x, then there is a neighborhood B of $f(x)$ such that for every $y \in B$, there is an x such that $f(x) = y$.
- **Definition 14.11.** A function $g : \mathbb{R}^n \to \mathbb{R}^n$ is a contraction if it holds that for some $0 \leq \alpha < 1$, we have $\|g(x) - g(y)\| \leq \alpha\|x - y\|$.
  - The contraction mapping theorem tells us that if S is a closed set and g a contraction such that $g(S) \subseteq S$, then there is a unique fixed point in S.
    * For if there were two fixed points, x and y, then we would not have $\|g(x) - g(y)\| \leq \alpha\|x - y\|$.
    * We often reduce root finding problems of the form $f(x) = 0$ to fixed point problems $g(x) = f(x) + x$.

Proposition 14.12. The condition number of solving a root problem is the same as the condition number of evaluating the inverse function at 0.

Proof.

Lemma 14.13. The condition number of evaluating a function f at x is $|f'(x)|$.

Proof. Suppose that we perturb the input x by h; then the relative difference in the output is $f(x + h) - f(x) = hf'(x)$ using the Taylor approximation. It follows that

$$\left|\frac{f(x+h) - f(x)}{h}\right| = |f'(x)|$$

□

Using the preceding lemma, it follows that the function that we wish to find the absolute condition number of is $x \mapsto f^{-1}(x)$. Note that this function has derivative given by $\frac{1}{f'(f^{-1}(x))}$. Whence the condition number is given by

$$\frac{1}{f'(x^*)}$$

where x^* is the root such that $f(x^*) = 0$.

For this reason, we submitted the proposition above that the condition number is the same as the condition number of evaluating the inverse. □

Definition 14.14. A function has a root of multiplicity m if it holds that

$$f^j(x) = 0$$

where $0 \leq j \leq m-1$.

Remark 14.15. If a function has a root of high multiplicity then evaluating this root has a high condition number. For the function f will be very flat near the root x (since its derivatives at x are all 0), meaning that f^{-1} will be steep near 0, which implies that small input perturbations will have a high output perturbation.

Definition 14.16. Suppose that a function f outputs an estimate μ_k of some quantity μ at every iteration k. We say that f converges at rate r if

$$\lim_{k \to \infty} \|e_{k+1}\| / \|e_k\|^r \leq C$$

where $e_k = \mu_k - \mu$ and $0 \leq C < \infty$.

Remark 14.17. Suppose that $e_0 < 1$ is a starting error for two processes, one quadratically converging and the other linearly converging. In this case, quadratic convergence will converge faster (in fewer iterations) than linear convergence. For linear iteration will decrease the magnitude of the error vector by a constant factor every, say, k iterations whereas quadratic convergence will decrease the magnitude of the error vector by a factor of 2. Quadratic convergence will converge, to finite precision, in no more than 5 iterations, assuming that the starting error vector is less in magnitude than 10^0, because as we discussed just now, floating point precision will only allow for accuracy for numbers up to 10^{-16}.

In light of this quick convergence near the solution, it is not meaningful to have operations with convergence rate greater than 2.

Definition 14.18. One of the following criteria for convergence if often employed, although all criterion have their flaws:

- $$|f(x)| < \epsilon$$
  - A very flat function (one with, for example, a root of high multiplicity) may become small at points that are not roots.
- $$\|x_k - x_{k+1}\| < \epsilon$$
  - An algorithm may fail to make progress, accounting for incremental differences.
- $$\frac{\|x_k - x_{k+1}\|}{\|x_k\|} < \epsilon$$
  - The same as above.

15 Lecture 15

Bisection Method

Problem 1.

Ritz Values Series

Q_k is the obtained by orthogonalizing the columns of

$$C_k = \begin{bmatrix} b & Ab & A^2b & \dots & A^{k-1}b \end{bmatrix}$$

Consider a symmetric positive definite matrix A and the projected matrix $T_k = Q_k A Q_k^T$.

The eigenvalues of T_k are called Ritz values. If we define the following 2 sequences

$$m_k = \min_{i=1}^{k} \lambda_i(T_k)$$

$$M_k = \max_{i=1}^{k} \lambda_i(T_k)$$

Which of the following statements hold?

Which of the following hold?

Solution 15.1. The estimates of the biggest and smallest can only get better and better as we increase k. So we find that M_k is a non-decreasing sequence and m_k a non-increasing sequence.

Problem 2.

Equal Eigenvalues Assume you are given a matrix whose eigenvalues are all the same. You wish to use Arnoldi iteration to compute the spectrum. What can you say about using this method? Choice* Having equal eigenvalues has no effect in the convergence of Arnoldi iteration Having equal eigenvalues guarrantees the matrix is diagonalizable and therefore Arnoldi always converges The k-th Ritz value converges to the k-th eigenvalue at iteration k Arnoldi reduces the matrix to tridiagonal form since having equal eigenvalues means the matrix is symmetric

Solution 15.2. Having equal eigenvalues gives no informatin about the matrix.

Problem 3.

Lanczos Iteration Lanczos iteration modifies Arnoldi iteration when the matrix is symmetric or Hermitian so that recurrence then has only three terms.

Which of the following are properties of Lanczos?

Select all that apply: Lanczos requires less work than Arnoldi Lanczos iteration generates an upper Hessenberg reduced Krylov matrix Lanczos produces a different result than if Arnoldi was applied to the same symmetric matrix Lanczos iteration generates a tridiagonal reduced Krylov matrix

Solution 15.3. Lanczos require less work – since the resulting hesenberg matrix that we fill in the arnoldi vector computation is really tridiagonal then. A tridiagonal matrix is also upper hesenberg, so this works there; lanczos produces no different result – it just takes less time sinc eit exploits the tridiagonal form; yes the last is a definition.

Note that the krylov matrix is just the matrix that A is similar to via orthogonalization by an orthogoanl basis of the krylov subspace $[x_0 Ax_0 \dots A^{k-1}x_0]$.

Problem 4.

2-Norm of a Matrix Suppose you are given an arbitrary matrix, A, of which you want to compute the 2-norm What method would you use to accomplish this in the fastest and most accurate way?

Choice* Randomly select a set of vectors, x, evenly distributed and apply the definition Compute the SVD of A directly and read of the largest singular value Use power iteration to compute the largest eigenvalue which corresponds to the largest singular value Apply a few iterations of Lanczos to obtain singular value estimate from Krylov subspace of AA

Solution 15.4. The two norm of a matrix is just the largest singular value; computing the svd takes $O(n^3)$ time – hitting several random unit vectors by A may not be accurate; the largest eigenvalue does not correspond to the largest singular value unless the matrix is symmetric (note that this is a sufficient condition but not a necessary one – ie it could be the case that a matrix exists with largest eigenvalue corresponding to the largest singular value, but yet the matrix is not symmetric). Arnoldi iteration takes $O(k^3+nk^2) = O(\max k^3, nk^2)$ in general – since we need $O(nk)$ to orthogonalize the kth arnoldi vector against the previous $k-1$ vectors and we need to do this $nk + n(k-1) + nk \approx nk^2$ times. Then when we apply eigenvalue methods to the resulting reduced Arnoldi matrix (the ritz matrix), the costs there is $O(k^3)$.

Problem 5.

Just know that the condition number of solving the root problem is the same as teh condition n umber of evaluating the inverse, which is $\frac{1}{f'(x^*)}$ where x^* is a root.

Definition 15.5. The bisection method operates on an interval $[a, b]$ where we assume that $sign\,(f(a)) = -sign\,(f(b))$, from which it follows that there exists a zero in $[a, b]$. We then do the following: Letting $m = a+b/2$, if $f(m)f(a) \geq 0$, we recurse on $[m, b]$ or else on $[a, m]$.

Remark 15.6. At any iteration the distance from either a or b to the root is at most $b - a$. We define $b - a$ to be the error at this iteration. A subsequent iteration will halve the interval, from which we conclude that the error rate is given by

$$e_k \leq e_{k+1}\frac{1}{2}$$

Remark 15.7. Since the magnitude in the error drops by 2^{-1} on every iteration, bisection will terminate in at most 52 iterations, since the error after 52 iterations is $2^{-52} = \epsilon_{mach}$, and we cannot represent any number thereafter. UNRESOLVED – well, shouldn't the relative error have to be 2^{-52} for us to make this conclusion.

Definition 15.8. Assume that there is a fixed point x^* of a smooth function g (continuously differentiable). Assume further that $g'(x^*) < 1$, which implies that $g' < 1$ on a neighborhood of x^*. Then it follows that

$$g(x_k) - g(x^*) = g'(\theta)(x_k - x^*)$$

where $g(x_k) = x_{k+1}$ and θ is some point in between $[x_k, x^*]$. If θ falls into the neigborhood where $g' < 1$, then the equation above is a contraction (here, we use the stronger definition) and it converges linearly.

Remark 15.9. Suppose that $g'(x^*) = 0$. Then using a second order taylor expansion, it holds that $g'(\theta) = g'(x^*) + g''(x^*)(\theta - x^*) \leq g''(x^*)(e_k)$. Here we assume that $\theta > x^*$, meaning that the fixed point is the point closer to ∞. Then it follows that

$$\underbrace{g(x_k) - g(x^*)}_{e_{k+1}} \leq g''(\theta)e_k^2$$

Remark 15.10. The fact that $g'(\delta) = g'(x^\star) + g''(x^\star)(x^\star - \delta)$ implies that as δ approaches x^*, the magnitude of the first derivative drops. In summary if this derivative drops enough that it is 0 at x^*, then convergence is quadratic and if, at least, the derivative is less than 1, then convergence is linear.

16 Lecture 16

Newton

16.1 Quiz

Problem 1.

Rate of Convergence

Suppose you applied an iterative numerical method for finding the roots of a scalar function, $f(x) = 0$, starting from an initial guess of x_0. The error in the approximate solution at the kth step, $e_k = |x_k - x^*|$, is given by the following sequence,

k	1	2	3	4
e_k	1	1/2	1/4	1/8

what is the rate of convergence?

Choice* (●) Linear ( ) Quadratic ( ) Superlinear (but less than quadratic) ( ) Cubic

Solution 16.1. The error is increasing by a constant multiplicative factor on every iteration; hence the error must be linear.

Problem 2.

Which of the following is true:

lect all that apply:

A small residual $\|\mathbf{f}(\mathbf{x})\|$ guarantees a solution of a system of nonlinear equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.

For a given fixed level of accuracy, a superlinearly convergent iterative method always requires fewer iterations than a linearly convergent method to find a solution to that level.

If an iterative method for solving a nonlinear equation gains more than one bit of accuracy per iteration, then it is said to have a superlinear convergence rate.

It is preferable to use the absolute condition number to assess the sensitivity of a solving a nonlinear equation.

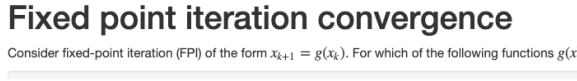
Solution 16.2. A small residual may just indicate that the values of $\|f\|$ are generally quite small. The second: if the convergence constant is above 1, then in fact, the method will not converge – so it really depends on the convergence constant (if the convergence constant is less than 1 and we start off with the same initial error, the n the answer is yes). Superlinear methods need to have an increasing number of bits that they gain. In general, if the initial error is e and the the convergence is rate r with some constant less than 1 as C then the error at stage m can be seen just by repeatedly substituting Ce^r into the cheapo version of this equation and observing how the resulting value changes.

Problem 3.

Which value of d allows us to achieve optimal convergence.

Solution 16.3. Note that this question is a bit misleading in that we can only determine if a variant of a fixed point function is better than another variant on the basis of its behavior near a fixed point. Away from the fixed point, all bets are off. Thus, we really should be comparing, for each value of d, what the consequent derivative of the fixed point function $g(x) = x - f(x)/d$ is at the fixed point of the function f. If, however, we assume that the derivative of g as 2 sufficiently approximates the derivative of g at $\sqrt{5}$ which is the fixed point that g intends to navigate to, then we find that $g' = 1 - 4/d$ at $x = 2$ making $d = 4$ the best choice.

Problem 4.



Which value of d allows us to achieve optimal convergence.

$$g(x) = \cos(x).$$

$$g(x) = 100\cos(x).$$

$$g(x) = \cos(100x).$$

$$g(x) = 5.$$

$$g(x) = \frac{2}{3}x + \frac{1}{3}\frac{A}{x^2}, A > 0.$$

$$g(x) = \frac{A}{x}, A > 0.$$

Solution 16.4. The strategy is to either realize that some functions will always have a derivative at any point less than 1 – meaning that if the function does have a fixed point, then it will converge to the fixed point sufficiently closed to it. These functions are $g(x) = \cos(x)$ and $g(x) = 5$. Otherwise, for the remaining functions solve for $g(x^*) = x^*$ and see if $g'(x^*) < 1$. The only function that satisfies this for all values of A is $g(x) = \frac{2}{3}x + \frac{1}{3}\frac{A}{x^2}$; the other $g(x, A)$ function does not; finally, I am unsure how we invalidate the functions $g(x) = 100\cos(x)$ and $g(x) = \cos(100x)$.

Problem 5.

Which value of d allows us to achieve optimal convergence.

Interval Bisection

Which of the following properties is necessarily true of the interval bisection method for finding a

It converges.

Its convergence rate is linear.

For a polynomial P, there exists a bracket, $[a, b]$ such that $\text{sign}(P(a)) \neq \text{sign}(P(b))$.

It gains one bit of accuracy per iteration.

The number of iterations required to attain a given accuracy depends on the particular function.

Solution 16.5. Note that if we have a starting interval $[a, b]$ and we wish to reduce the error to tol, then we need to solve for the k such that

$$\frac{(b-a)}{2^k} = tol$$

which turns out to be $\log(\frac{b-a}{tol})$. This is true for any function.

Note that given an interval where we've identified a root to fall into, this method always converges; its convergence is linear, because we halve the interval each time; the third remark is non-sequitur and useless; the fourth remark is true because linearly convergent methods gain a constant number of bits per iteration and here in particular, since the error goes down by a half each iteration, we gain one bit in accuracy (assuming that our interval $b - a$ is intially less than 1).

Definition 16.6. We want to find the h such that $f(x+h) = f(x)+hf'(x) = 0$. This sets $h = \frac{-f(x)}{f'(x)}$, which implies that given an initial x_k,

$$x_{k+1} = x_k + h = x_p + \frac{-f(x_p)}{f'(x_p)}$$

Set $g(x)$ to be

$$x + \frac{-f(x)}{f'(x)}$$

Observe that if $f'(x) = 0$ (ie if f has a double root at a fixed point x^*, then $g(x^*)$ is indeterminate and Newton fails.

Observe that if Newton's method hits a point y such that $f(y) = 0$, then y is fixed point, since all successive iterates will still remain at y.

Also observe that $g'(x) = \frac{f(x)f''(x)}{f'(x)^2}$ so that if $f(x) = 0$ but $f'(x) \neq 0$ (ie x is a simple root), then fixed point iteration converges quadratically if we are close enough to the root – then newton's method converges quadratically (assuming that there is no double root). Even in the event that a double root exists, it can be shown that the convergence is then linear with constant $C = 1 - 1/m$ where m is the multiplicit of the root.

Remark 16.7. There are two downsides to using Newton's method:

- We need the derivative.
- If we want other roots, we need to hope that some choice of another initial vector will converge to root different than the one we first found.

Remark 16.8. It is claimed that even if newton is not quadratically convergent, then it is at least linearly convergent since the derivative of the fixed point function $g'(x) = \frac{f(x)f''(x)}{f'(x)^2}$ is such that $f(x)f''(x) \to 0$ faster than $f'(x) \to 0$. Why this is true is unresolved. Indeed, Heath on page 251 explains that Newton's method is convergent with linear rate $1-1/m$ where m is the multiplicity of the root for $m \geq 2$. It is claimed that $f(x) \to 0$. Newton breaks down in the event that the derivative function does not approximate a function very well or if the derivative function change sign often, taking us "left" and "right" of the roots repeatedly.

Definition 16.9. The secant method works by replacing $f'(x)$ in Newton's method with $s = f(x_k) - f(x_{k-1})$. We can also use a quadratic interpolant. What does this mean? Recall that we approximated $f(x+h)$ by $f(x)+hf'(x)$. We could have also approximated $f(x+h)$ by $f(x)+hf'(x)+h^2f''(x)/2$. Then we need to approximate $f'(x)$ and $f''(x)$ using three points (can you guess what we might do?). Alternatively, we find an inverse quadratic interpolant that takes three points $(x, f(x)), (y, f(y)), (z, f(z))$ and then finds an inverse interpolant q such that $q(f(x_i)) = x_i$. Having found this interpolant, we then let the next point be $q(0)$, since we expect that $f(q(0)) = 0$.

Remark 16.10. Both the use of newton's method through a quadratic interpolant and inverse quadratic iteration converge with superlinear rate $r \approx 1.81$.

Remark 16.11. Safeguarded methods combine several numerical methods together; for example:

Suppose that we apply newton's method within a bracket (that is, we are given an input that falls in some set S). If it happens that $x_{k+1} \notin S$, then we apply bisection and return the midpoint of the half that bisection concludes that we should use. If it does converge, then use Newton's iterate as the solution.

17 Lecture 17

Remark 17.1. Newton's method performs poorly if there are multiple roots.

Definition 17.2. In n dimensions, fixed point iteration generalizes as follows:

- Suppose that $g : \mathbb{R}^n \to \mathbb{R}^n$. Then if $p(J(g(x^*))) < 1$, g converges to its fixed point x^* linearly. Note that p is the spectral radius.
- If $J(g(x^*)) = 0$, then g converges to x^* quadratically.

Remark 17.3. Bisection is hard to generalize for even if we could recursively find good bounds for a function f_1 where $f = [f_1 \dots f_n]$, bounds particular to f_1 are likely unrelated to the bounds needed for f_2.

Definition 17.4. We can approximate $f : \mathbb{R}^n \to \mathbb{R}^n$ as

$$f(x+h) = f(x) + J(f(x))h + O(h^2)$$

solving for h we find

$$h = -J(f(x))^{-1}f(x)$$

meaning that the new iteration scheme is:

$$x_{k+1} = x_k - J(f(x))^{-1}f(x) \tag{}$$

Remark 17.5. This has a couple of downsides; namely, computing the Jacobian may be expensive; moreover, computing the inverse may be expensive; finally this is only locally convergent (as before).

Remark 17.6. UNRESOLVED: when is this method quadratically convergent?

Remark 17.7. The secant method is hard to generalize, because it is not clear how we can obtain an approximation to the Jacobian $J(f(x))$ using only $f(x_1), f(x_2), x_1, x_2$ although obtaining an approximation to $f'(x)$ given $f(x_i)$ and x_i is certainly plausible. Broyden's method starts with a guess for $J(f(x))$, and then with some update akin to (α)it then finds the next Jacobian.

17.1 Optimization

Lemma 17.8. If a function is continuous on a closed and bounded set $S \subseteq \mathbb{R}^n$, then it attains its minimum and maximum. See `https://en.wikipedia.org/wiki/Extreme_value_theorem#Proof_of_the_extreme_value_theorem` for the proof.

Definition 17.9. A function is coercive if $\lim_{\|x\|\to\infty} f(x) = \infty$. A coercive function attains a global minimum, although the minimum may not be unique. Think of a cubic function with two dips.

Definition 17.10. A set S is convex if for $x, y \in S$ it holds that $x(\alpha)+y(1-\alpha) \in S$ for all $\alpha \in [0, 1]$. A function f is convex if it holds that $f(x\alpha + y(1-\alpha)) \leq \alpha f(x) + (1-\alpha)f(y)$.

Proposition 17.11. If a function f is continuous and convex, then it attains a minimum; if it is instead strictly convex and continuous, then its minimum is unique.

Lemma 17.12. The following are sufficient and necessary conditions for minimality in 1 dimension and in $\mathbb{R}^n$.

- $\mathbb{R}^1$:
  - $f'(x) = 0$ is a necessary condition.
  - In conjunciton with $f''(x) > 0$, the foregoing is a sufficient condition.
  - $\nabla f(x) = 0$ is a necessary condition.
  - In conjunction with $H(f(x)) > 0$ – ie the Hessian being positive definite – we get a sufficient condition.

Definition 17.13. A matrix admits a Cholesky Decomposition, a factorization of the form LL^* where L is lower triangular, iff it is positive definite hermitian. In particular, if A is positive definite, then the factorization is unique; but if A is merely positive semidefinite, then the factorization is not unique; if A is not event positive semidefinite, then the cholesky decomposition will have a negative entry somewhere along the diagonal (this is not true otherwise).

Remark 17.14. We can find the minimum of a function f by solving for $\nabla f = 0$ using any of the methods that we've developed for multidimensional root finding. To then assert that the found root actually corresponds to a minimum, we need to check whether the Hessian of the matrix is positive definite. Note that since H is symmetric, if we attempt to decompose H into a Cholesky Decomposition and find that it has a negative entry on one of its diagonals, then the matrix is in fact not positive definite.

Proposition 17.15. The error in a solution to an optimization problem is at best 10^{-8}.

Proof. Suppose that x^* is a true minimizer but that we estimate $\hat{x}$ to be the minimizer instead. Let $\hat{x} = x^* + h$. Then observe that

$$f(x^*+h) \approx f(x^*)+f'(x^*)h+f''(x^*)h^2/2 = f(x^*)+f''(x^*)h^2/2 \implies h^2 = \frac{2(f(x^*+h)-f(x^*))}{f''(x^*)}$$

What this implies is that if we take $\hat{x}$ to be a solution and we have $|f(\hat{x}-f(x^*)| < \epsilon$, then our error in $\hat{x}$, which is h, is at best $\sqrt{\frac{2\epsilon}{f''(x^*)}}$. This is about 10^{-8}, implying 8 accurate digits, if $\epsilon = \epsilon_{mach}$. We upper bounded the error in x^*, which suggests that we should revise our original statement to be that we can at least get a tolerance of 10^{-8}. The fact that during the course of a problem, we can conceivably find $\hat{x}$ such that $|\hat{x} - x^*| \approx \epsilon_{mach}$ and declare this value our minimum makes this a best bound however. □

Remark 17.16. The follwoing facts about convexity should be known:

- If a convex function has a local minimum, then the local minimum is, in fact, a global minimum.
- If a strictly convex function has a local minimum, then the local minimum is a unique global minimum.
- If a continuous function is restricted to a closed and bounded set $S \subseteq \mathbb{R}^n$, then $f(S) \subseteq \mathbb{R}$ is compact and, hence, obtains a minimum, so that there is a minimum $x \in S$ of f.
- More generally, if f is a continuous function restricted to a closed but unbounded set and f is coercive, then f attains a minimum. This can be proven. Note that a convex function on a convex set is necessarily continuous at interior points of its domain. This has never been proven.

18 Lecture 18

18.1 Quiz

Problem 1.

Comparison of Newton and Broyden Methods Which of the following is NOT an advantage of Broyden's method over Newton's method for solving a system of nonlinear equations? Choice* Matrix factorization can be updated each iteration rather than having to be recomputed No derivatives required Less work per iteration Fewer iterations required for given accuracy

Solution 18.1. Matrix factorization has to be updated in Newton's method everytime – meaning that we have to compute the Jacobian every time. The Jacobian method also requires a Jacobian which is not required by Broyden's method. The update in Broyden's method also resembles a Sherman-Morrison update, making it less costly per iteration to compute the next Jacobian (approximation to the Jacobian) than in Newton's method (this is true in spite of the fact that Broyden's method also computes a difference of the form $f(x_{k+1})-f(x_k)$.

Algorithm 5.5 Broyden's Method

$\boldsymbol{x}_0$ = initial guess
$\boldsymbol{B}_0$ = initial Jacobian approximation
for $k = 0, 1, 2, \ldots$
  Solve $\boldsymbol{B}_k \boldsymbol{s}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$ for $\boldsymbol{s}_k$ { compute Newton-like step }
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ { update solution }
  $\boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k)$
  $\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + ((\boldsymbol{y}_k - \boldsymbol{B}_k \boldsymbol{s}_k)\boldsymbol{s}_k^T)/(\boldsymbol{s}_k^T \boldsymbol{s}_k)$ { update approx Jacobian }
end

Problem 2.

Easy to tell what a convex set looks like.

Problem 3.

Have to review convexity.

Problem 4.

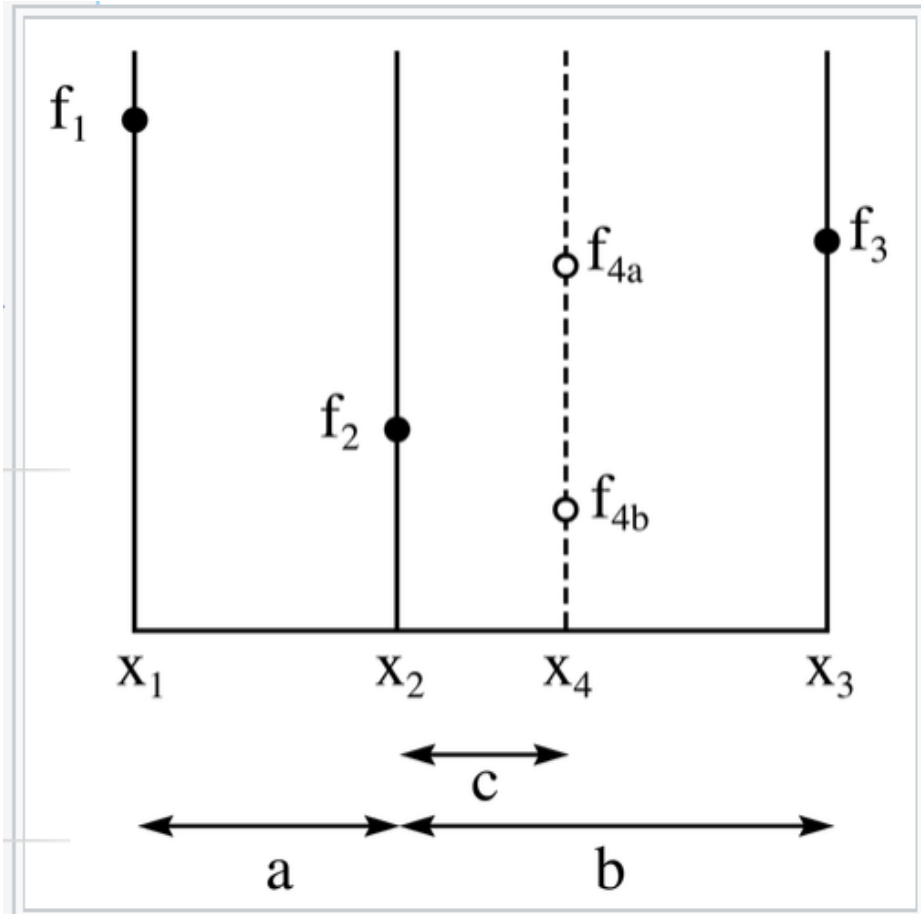
Simply remember that the Hessian is the transpose of the Jacobian of the gradient. ALso remember that if a hessian is neither positive semidefinite nor negative semidefinite at a point where the gradient is zero, then the critical point is a saddle point.

Problem 5.

Optimization Accuracy Suppose that f can be evaluated to within a relative error of 10−10. If x∗ is an extreme point of f, with $|f''(x)| = 1$, then about how many decimal digits of x∗ can reliably be computed as a solution to an optimization problem?

Solution 18.2. Apply the lemma. from the previous lecture.

Definition 18.3.



In golden section search, given an interval $[x_1, x_3]$ (or in our case the interval is often denoted $[a, b]$) we choose points inside the interval x_2, x_4 (in our case, we define these points to be $a + (b - a)(1 - \tau)$ and $a + (b - a)\tau$ such that

$$\frac{|x_3 - x_2|}{|x_2 - x_1|} = \frac{|x_2 - x_1|}{|x_4 - x_2|}$$

In the diagram above, we see that at any point in the initiation of a golden search we have three points that are known, and depending on the function evaluation of the fourth point, we will choose the subinterval $[x_1, x_4]$ or $[x_2, x_3]$.Thus the stipulation that $c/a = a/b$ represents our desire that when we recurse into a new subinterval that the three given points have been some subset of the givens and function evaluations in the interval "one frame up." Note that this inequality assumes that we progress into the left subinterval; the analysis for the right subinterval is the same, however, since $a + c = b$ and $b - c = a$ (that is $|x_3 - x_4| = |x_2 - x_1|$

It can be shown that solving this proportion requires that τ be the golden ratio.

Definition 18.4. Newton's method for optimization is to use newton's method for root finding in order to solve $f'(x) = 0$. We explore the 1-D case, because this easily generalizes to the n-D case:

$$f(x+h) = hf'(x) + h^2 f''(x)/2 + O(h^3)$$

Differentiate this with respect to h

$$f'(x+h) = f'(x) + hf''(x) + O(h^3)$$

:

Set this to 0 and solve for h

$$h = \frac{-f'(x)}{f''(x)}$$

So the update becomes

$$x \leftarrow x - \frac{f'(x)}{f''(x)}$$

In n-D, this will generalize to:

$$\mathbb{x} \leftarrow \mathbb{x} - (H_f(x))^{-1} \nabla f(x) \qquad (\beta)$$

This quadratic approximation of f via a taylor series expansion is only accurate within a region. If we define what that region us as a ball centered at $x$with radius r and we see that $r < \left|(H_f(x))^{-1} \nabla f(x)\right|$ then we can bind the decrement to be precisely $r * (H_f(x))^{-1} \nabla f(x) / \left|(H_f(x))^{-1} \nabla f(x)\right|$.

For reasons that are not clear, this approach (called using a trust region), may also change the direction of the search. Note that we can also employ a so called line search to find the scalar factor of $(H_f(x))^{-1} \nabla f(x) / \left|(H_f(x))^{-1} \nabla f(x)\right|$ that best minimizes (β).

Definition 18.5. Gradient descent repeatedly runs the following function

$$\mathbb{x}_{\mathbb{k}+\mathbb{1}} \leftarrow \mathbb{x}_{\mathbb{k}} - \alpha_k \nabla f(x_k)$$

where α_k is chosen using some other method. For example. One may try to optimize the function $g(\alpha_k) = \mathbb{x}_{\mathbb{k}} - \alpha_k \nabla f(x_k)$.

Steepest Descent: Convergence

Consider quadratic model problem:

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

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

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

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

$\rightarrow$ confirms linear convergence.

Convergence constant related to conditioning:

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

Hacking Steepest Descent for Better Convergence

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

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k) + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1})$$

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

$$||\mathbf{e}_{k+1}||_A = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} ||\mathbf{e}_k||_A$$

Conjugate gradient method:

$$(\alpha_k, \beta_k) = \mathbf{argmin}_{\alpha_k, \beta_k} \left[f\left(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k) + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1})\right)\right]$$

- 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)

19 Lecture 19

19.1 Quiz

Problem 1.

Unimodal Function Suppose the real-valued function f(x) is unimodal on the interval [a,b]. Let x1 and x2 be two points in the interval, with a<x1<x2<b. If f(x1)=1.232 and f(x2)=3.576, then which of the following statements is valid on the subinterval [a,b].

Solution 19.1. Recall that unimodality tells us that if we have two points x and y such that $a \leq x < y \leq b$ and $f(x) < f(y)$ then the minimum will be found going "down the slope", meaning that it will be contained in the interval $[a, y]$. Then tells us that the solution is that the minimum must lie in the interval $[a, x_2]$. There is an option that attempts to dupe us: The subinterval of the minimum of f is indeterminate without knowing the values of $f(a)$ and $f(b)$. This is not true, however – if we are granted unimodality on $[a, b]$ even without knowing the values of a and b we can be assured of the statements in this solution.

Problem 2.

Golden Section Search Suppose we decided to use golden section search for finding the minimum of a unimodal function, but, in the inner loop of the algorithm, instead of picking evaluation points x1,x2 using the formulas

x1=a+(1−)(b−a) x2=a+(b−a) at each iteration (where a and b are the endpoints of the interval, and =12(5√−1)), we chose

x1=a+13(b−a) x2=a+23(b−a). The remainder of the algorithm is the same. This modified strategy is called ternary search.

What is a legitimate disadvantage of using ternary search compared to golden section search?

Solution 19.2. These were the choices:

On average per iteration, ternary search necessarily requires evaluating the function at more points. There are no substantial disadvantages compared to golden section search. The convergence rate of ternary search may be sublinear, while that of golden section search is always linear. Ternary search may fail to converge for some unimodal functions for which golden section search converges.

If we use ternary search, then we don't preserve the proportion that we listed above, requiring us to have on more function evaluation. Note that with ternary search, the interval length shortens by 2/3 every time. Note that because unimodality is always preserved and the interval decreases, ternary search must converge. Thus, the answer is the first choice.

Problem 3.

Robust Newton Methods For unconstrained optimization of a function f:n→, Newton's method is often unreliable when started far from the solution. How do line searches or trust regions improve its reliability?

Select all that apply: A trust region modifies the length of the Newton step. A line search modifies the length of the Newton step. A line search modifies the direction of the Newton step. A trust region modifies the direction of the Newton step.

Solution 19.3. See the logic above. The following are true: A trust region modifies the length of the Newton step. A line search modifies the length of the Newton step. A trust region modifies the direction of the Newton step.

Problem 4.

Convergence Rates Recall the definition of convergence rates

limk→∞‖ek+1‖‖ek‖r=C We are trying to minimize

$f(x) = \frac{1}{2}x^T \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix} x$ If we are using steepest descent to minimize f, what is r and C? Give 4 significant figures.

Solution 19.4. See the page on convergence rates in the your notes from the previous lecture.

Problem 5.

Newton's Method Consider minimization of the function $\phi(x, y) = (x-2)^2 + (y-1)^2$ Starting with an initial guess [x,y]0:=[0,0], what are the values of x and y after one round of Newton's Method

Solution 19.5. Compute the gradient; compute the hessian $A = H_f(0, 0)$
Then solve for

$$A(s_k) = -\nabla f(0, 0)$$

Definition 19.6. Newton's method in N dimensions does the following:

$$f(x + h) = f(x) + \nabla f(x)h + \frac{1}{2}h^T H_f(x)h$$

Now differentiate with respect to h and solve for h

$$0 = \nabla f(x) + H_f(x)h$$

Then set $x_k \leftarrow x_k + h$

Remark 19.7. This method has a few problems:

- It depends on the validity of a Taylor expansion and, hence, is only locally convergent.
- It requires that we compute second derivatives.
- UNRESOLVED (works poorly when H_f is nearly indefinite).

Definition 19.8.

Nelder-Mead Method

Idea:

> Form a *n*-point polytope in *n*-dimensional space and adjust worst point (highest function value) by moving it along a line passing through the centroid of the remaining points.

Definition 19.9. Let $y - a(x) = r(x)$ and suppose that we wish to minimize

$$\phi(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^{n} r_i(x)^2$$

$$\frac{\partial}{\partial x_i} \phi(x) = \sum_{j=1}^{n} \frac{\partial r_j(x)}{\partial x_i} (r_j(x)) \implies \nabla \phi(x) = J^T(x) r(x)$$

This then gives:

$$\frac{\partial^2}{\partial x_k \partial x_i} \phi(x) = \sum_{j=1}^{n} \frac{\partial^2 r_j(x)}{\partial x_k \partial x_i} r_j(x) + \frac{\partial r_j(x)}{\partial x_i} \frac{\partial r_j(x)}{\partial x_k} \implies H_\phi(x) = J^T J + \sum_{j=1}^{n} r_j H_{r_j}(x)$$

We assume that the terms $r_j H_{r_j}(x) \approx 0$ so that $H_\phi(x) = J^T J$. Then the step size in newton's method, $-H_\phi(x)^{-1} \nabla \phi(x)$ becomes $-(J^T J)^{-1} J^T r(x)$.

That is, Gauss Newton computes the step size by computing s where $J(x)s \approx r(x)$

20 Lecture 20

20.1 Quiz

Problem 1.

We use the approximation

$$x_k \leftarrow x_{k-1} + \alpha (H_f(x))^{-1} (-\nabla_f(x))$$

This works out to be

$$x_k \leftarrow [2,1]^T + 1 \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \end{bmatrix} (- \begin{bmatrix} \cos(x_1) \\ -\sin(x_2) \end{bmatrix})$$

Let us approximate $\cos(x_1) = \cos(2)$ by -0.41614 and let us approximate $-\sin(x_2) = -\sin(1)$ by -0.84147. This gives us

$$[2,1]^T + \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \end{bmatrix}^{-1} (- \begin{bmatrix} -0.41614 \\ -0.84147 \end{bmatrix})$$

See the code below

Problem 2.

Secant Updating Methods For unconstrained minimization of a function f:n→, why is it not a good idea to find a critical point by using Broyden's method to solve the nonlinear system f(x)=0? Select all that apply: It would require a matrix factorization at each iteration. It would not preserve symmetry of the approximate Hessian matrix. It would require evaluation of the Hessian matrix. It would not converge superlinearly.

Solution 20.1. First recall what Broyden's method is. Brodyen's method computes the solution to $\nabla_f(x) = 0$ using a series of approximations. We start with an initial approximation of the Jacobian B_k. Then we compute a step size that will update the current value of x_{k+1}. This gives us an updated value of the diference $f(x_{k+1}) - f(x_k) = y$. We then use y and x_{k+1}, x_k to update B_k with some update resembling a Sherman Morrison update. Note taht this update is not guaranteed to preserve symmetry of the Hessian matrix. Realize that here B_k corresponds to the Hessian. Secant methods have been shown to converge superlinearly, so the convergence would still be superlinear; at no point do we ever explicitly evalute the object that B mocks so there is no evaluation of a Hessian or anything of that sort.

Problem 3.

Gauss-Newton vs. Newton Which of the following statements about Gauss-Newton in comparison with 'regular' Newton applied to the 2-norm of the residual norm

$$\phi(x) := \frac{1}{2}\|f(x) - y\|_2^2$$

Gauss-Newton requires fewer known derivatives.
The approximation to H used by Gauss-Newton may be inaccurate if the residual is large.
Gauss-Newton has a lower cost per iteration.
The two are equivalent.
Gauss-Newton uses a more accurate approximation to f in computing the next iterate.
Both converge globally

Solution 20.2. Recall that Gauss newton approximates the Hessian as J^TJ and ignores the terms $r_iH_{r_i}(x)$;if the residual terms actually happen to be big, then our approximation is off. Note that since Gauss Newton does not use the Hessian explicitly, the costs of finding derviatives that are part of the Hessian are non-existent in Gauss-Newton (and so Guass newton has a lower costs per iteration).

Problem 5.

Constrained Optimization Consider the constrained optimization problem
minf(x),g(x)=0. At a minimal point, f(x) must...

Solution 20.3.

$$\begin{aligned} f(x) - \lambda^T g(x) &= 0 \\ \implies J(f(x)) - J(\lambda^T g(x)) &= 0 \\ = J(f(x)) - \lambda^T J(g(x)) &= 0 \\ \implies \nabla_f(x) &= J(g(x))^T \lambda \end{aligned}$$

hence only the first choice is correct.

Definition 20.4. A constrained optimization problem is of the form:

$$\min f(x) \text{ such that } g(x) = 0$$

here $f : \mathbb{R}^n \to \mathbb{R}^1$ and $g : \mathbb{R}^n \to \mathbb{R}^m$.

Remark 20.5. Let the feasible set (ie those x such that $g(x) = 0$) be S. Given a point $x \in S$, we say that a direction s is feasible if for some $\alpha \in [0, r)$ where $r > 0$ that $x + \alpha * s \in S$ as well. A first order condition is that at the boundary of the feasible set, if s is a feasible direction then,

$$\nabla f(x^*)^T s \geq 0$$

or else, we would proceed in the direction of s and hence minimize f.

If we are at an interior point, then both s and $-s$ are feasible directions in which csae the optimality condition implies that $\nabla f(x^*) = 0$, which coincides with the first order optimality condition for unconstrained optimization. Indeed, constrained optimization is only interesting at the boundary of the feasible set, because constrained optmization problems behave like unconstrained optmization problems at the interior of the feasible set.

Remark 20.6. A second order optimality condition is that for any point x with a feasible direction s it holds that

$$s^T H_f(x^*) s \geq 0$$

It is easy to see why this is a necessary condition for the unconstrained case, for an approximation to $f(x^* + s)$ is given by

$$\nabla f(x^*) \cdot s + s^T H_f(x^*) s$$

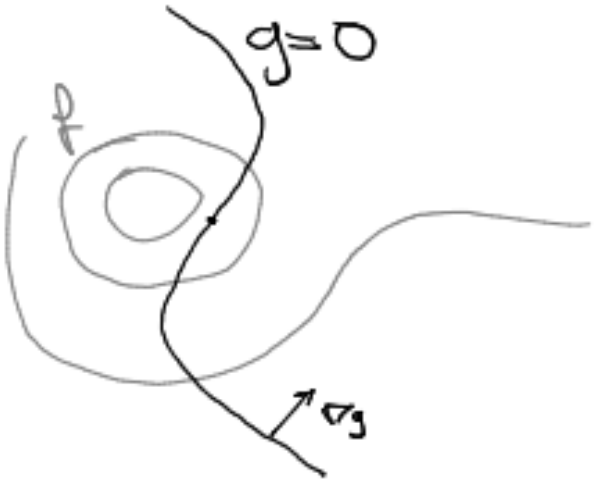
so that if $s^T H_f(x^*) s < 0$, then $f(x^* + s) < f(x^*)$, contradicting that $f(x^*)$ is optimal. It is not clear, however, why this is a necessary condition for a boundary point, for it may just take place that in the approximation above of $f(x + a)$, if $\nabla f(x^*) \cdot s > \left| s^T H_f(x^*) s \right|$, then $f(x^* + s) > f(x^*)$.

Proposition 20.7. At an optimal point x^*, it holds that

$$-\nabla f(x^*) = J_g^T \lambda^*$$

for some $\lambda^* \in \mathbb{R}^m$.

Proof. This is easiest to imagine:



Suppose that the dotted point is x^*. Now if we try to proceed in the direction of greatest descent, which is $-\nabla f(x^*)$, then we find that we must move in the direction of greatest ascent for g. That is, any further attempt to minimize f necessarily forces us to violate our constraint on g and, hence, fall off of the level set $g = 0$.

It is known that $\nabla f(x)^T$ is the direction of greatest ascent (recall that there is some calculus proof that explains that not only does $\nabla f(x) \cdot s$ where s is some n vector give us the instantaneous increase in the direction s but that were s in the direction of $\nabla f(x)$ itself, we would increase most rapidly (indeed use the cosine identity involving the dot product). One can similarly generalize (though, I am not sure how) and conclude that $(\nabla g)^T = J_g^T$ gives us the direction of greatest ascent in g.

We hit J_g^T by some m vector to give us an n vector. □

Definition 20.8. The lagrange function is defined to be

$$L(x, \lambda) f(x) + \lambda^T g(x)$$

since then $\nabla L = 0 \implies \nabla f(x) + J_g(x)^T \lambda = 0$ and $g(x) = 0$. That is

$$\nabla L = \begin{bmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{bmatrix} = \begin{bmatrix} \nabla f(x) + \lambda^T g(x) \\ g(x) \end{bmatrix}$$

which are necessary conditions that we previously discussed.

Remark 20.9. If we wish to compute the hessian of this function, then simply do the following, take the jacobian with respect to x of the first row of ∇L and with respect to λ of first row; the resulting entries become the first row of the Hessian; likewise, obtain the second row of the Hessian. This gives us

$$H_L(x,\lambda) = \begin{bmatrix} B(x,\lambda) & J_g^T x \\ J_g(x) & 0 \end{bmatrix}$$

where $B(x,\lambda) = H_f(x) + \sum_{i=1}^m \lambda_i H_{g_i}(x)$.

Note that since a matrix is positive definite if and only if its eigenvalues are all positive, and the product of the diagonals of a matrix is the product of its eigenvalues, H_L is not positive definite (since it has 0 as an eigenvalue). This matrix is, however, symmetric (check this yourself).

Remark 20.10. As a consequence, Heath asserts that any critical point of the Lagrangian is necessarily a saddle point. That is, it must hold that at x^*, there exists s_1 such that $s_1^T H_f(x^*) s_1 > 0$ and s_2 such that $s_2^T H_f(x^*) s_2 < 0$. UNRESOLVED (Why is this true?) At present, the only lead I have is that if, suppose, that x^* corresponded to a minimum, it would have to hold that for all feasible directions s, there is some $\alpha \in [0,1]$ such that $f(x^* + \alpha s) \geq f(x)$.

This woudl imply using the Taylor remainder theorem that

$$f(x^* + s) = f(x^*) + \underbrace{\nabla f(x^*) \cdot s}_{0} + s^T H_f(x^* + \alpha' s) s \geq 0 \implies s^T H_f(x^* + \alpha' s) s \geq 0$$

where $\alpha' \in [0,1]$

It is not clear, however, that

$$s^T H_f(x^* + \alpha' s) s \geq 0 \implies s^T H_f(x^*) s > 0$$

for all s

Remark 20.11. UNRESOLVED: if f is convex, then any critical point must be a global minimum (a statement made in Heath). This implies that convex functions have either minimums or maximums but not both?

Definition 20.12. When we introduce inequality constrains, ie functions $h_i : \mathbb{R}^n \to \mathbb{R}^m$ such that $h_i(x) \leq 0$, then we redefine the Lagragian to be

$$f(x) + \lambda_1^T (g(x)) + \lambda_2^T (h(x))$$

Remark 20.13. We must have $\lambda_{2,i} h_i(x) = 0$ for all i that correspond to indices of the inequality functions h_i. It is not known why this must hold.

We say that h_i is active at x^* if $h_i(x^*) = 0$. If h_i is active, then it can happen that in some direction s, for some small α we have $h_i(x^* + \alpha s) > 0$. This cannot happen, by contrast (assuming that h_i is sufficiently smooth, if $h_i(x^*) < 0$.

Assuming $J_{\mathbf{g}}$ and $J_{\mathbf{h},\text{active}}$ have full rank, this set of conditions is *necessary*:

$$\begin{aligned} (*) \quad \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda_1^*, \lambda_2^*) &= \mathbf{0} \\ (*) \quad \mathbf{g}(\mathbf{x}^*) &= \mathbf{0} \\ \mathbf{h}(\mathbf{x}^*) &\leqslant \mathbf{0} \\ \lambda_2 &\geqslant \mathbf{0} \\ (*) \quad \mathbf{h}(\mathbf{x}^*) \cdot \lambda_2 &= 0 \end{aligned}$$

These are called the Karush-Kuhn-Tucker ('KKT') conditions.

Computational approach: Solve $(*)$ equations by Newton.

Assemble the problem's inequalities as well as the complentarity condition for h_i and the fact that $\lambda_2 \geq 0$ to obtain the KKT conditions listed above.

21 Lecture 21

21.1 Quiz

Problem 1.

Constrained Optimization In a constrained optimization problem, when is the Hessian matrix of the Lagrangian function positive definite?

Hint: All diagonal entries of a SPD matrix must be positive.

Solution 21.1. Recall that the hessian contains a zero on a diagonal (in fact a block of zeros), so that it cannot be positive definite.

Problem 2.

Lagrange Multiplier

1 point

To minimize $f(x, y) = x^2 + 2y^2$ subject to $g(x, y) = x + 4y - 6 = 0$, we can look for critical points of the Lagrangian function $\mathcal{L}_f(x, y, \lambda) = f(x, y) + \lambda g(x, y)$. Provide the value of λ at the minimum by solving the the system of equations given by $\nabla \mathcal{L}(x, y, \lambda) = \mathbf{0}$.

Solution 21.2. Recall what the lagrangian looks like rather than combine the terms and then differentiate.

Your answer is correct.

A correct answer is: '-1.3333333333333333'.

We obtain the system of equations:

$$\mathbf{0} = \nabla \mathcal{L}(x, y, \lambda) = \begin{bmatrix} 2x + \lambda \\ 4y + 4\lambda \\ x + 4y - 6 \end{bmatrix}.$$

The first two equations imply that $x = -\lambda/2, y = -\lambda$, and inserting these in to the last, we obtain $0 = x + 4y - 6 = -\frac{9\lambda}{2} - 6$, so $\lambda = -\frac{4}{3}$.

Problem 3.

Constrained Optimization

Which of the following statements are true for finding the solution $\boldsymbol{x}^*$ of the constrained optimization problem,

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

$\nabla f(x^*) = 0$

x^* is a critical point of the Lagrangian function.

Sequential Quadratic Programming may be unreliable unless it starts close to the solution.

The inequality constraint $h(x)$ is only active on the interior of the feasible set.

Solution 21.3. We know that a first order optimality condition is simply that $\nabla f(x^*)^T s \geq 0$, so the first answer cannot be correct(actually, the fist answer cannot be correct because KKT tells us that $\nabla_x(L, \lambda_1, \lambda_2) = 0$ and not $\nabla f(x^*) = 0$; we know that x^* is a critical point of the Lagrangian by definition; SQP may be unreliable unless it starts close to the solution, precisely because SQP is Newton, which may fail if we don't apply it close to the solution.

The last choice needs to be examined more closely.

In general, for h to be a non trivial constraint there has to be a point (and hence a neighborhood) where $h(x) > 0$ (this means that some h_i gives $h_i(x) > 0$). It follows that by continuity there is a sequence of points $\{x_n\}$ such that $h(x_n) \to 0$, meaning that $x_n \to$ some point $z \in h^{-1}(0)$. It follows that z must be some point on the boundary. UNRESOLVED

We can design a function h so taht $h(x) = 0$ and there exists a neighborhood at x called C such that there exist c_1 and c_2 such that $h(c_1) > 0$ and $h(c_2) < 0$.

Just take $h : \mathbb{R} \to \mathbb{R}$ where $h(x) = x$ and consider $x = 0$.

Problem 4.

Inequality Constrained Minimum
Find the optimal parameters and λ inside the feasible set for the minimization function $f(x, y) = 5x^2 + 3y$ and $x + y \geq 5$.

Solution 21.4. Form the lagrange function, which is

$$5x^2 + 3y + \lambda(-x - y + 5)$$

Then solve for x, y, λ such that $\nabla_L = \mathbf{0}$.

After you solve for the parameters, tweak them so that $x + y \geq 5$.

Note that, in general, one would apply Newton's method (or a similar optimization scheme) to try and determine the minimum of ∇_L.

Problem 5.

Sequential Quadratic Programming Which of the following are true about sequential quadratic programming (SQP)?

Select all that apply: A single iteration of conjugate gradient suffices for each step of SQP Each iteration of SQP requires a single iteration of Newton's method SQP is more effective at handling inequality constraints than equality constraints When dealing only with equality constraints, SQP is Newton's method for optimization

Solution 21.5.

- Conjugate gradient is not used in SQP – Newton's method is.
- Yes, this is true – by definition.
- yes, for Heath explains that when we use inequality constraints, at a given iteration, we consider those constraints that are active and include them in the Lagrangian to which we then apply Newton's method.
  - For reasons that are not explained, this complication of having varying lagrangians that we minimize is less effective than simply using one, non-varying Lagrangian throughout the course of the problem.
- Yes, indeed – when we don't have inequality constraints, we are simply evertime computing $H_L(x)s = -\nabla_L(x)$ for s

Definition 21.6. An interpolatio problem asks us to find a function f such that given a set of points $\{(x_i, y_i)\}$ it holds that $f(x_i) = y_i$ for all i.

Definition 21.7. Suppose that we define $f = \sum_{i=1}^{n} \phi_j(x)\alpha_j$. Then$f(x_i) = y_i = \sum_{k=1}^{n} \phi_k(x_i)\alpha_k$.

This can be framed as solve for α in

$$Y = V\alpha$$

where $V_{i,k} = \phi_k(x_i)$ and Y has all the solutions.

If we require V to be square, and hence there to be precisely as many basis functions as there are points n – then the system above admits a unique solution in the event that V is full rank.

Definition 21.8. When we use a monomial basis we ask to solve for $(x_1 \dots x_n)$ such that

$$p(t) = \sum_{j=0}^{n-1} x_{j+1} t^j = y_{j+1}$$

where $((t_1, y_1) \dots (t_n, y_n))$ are the function points.
To solve for the x, we can solve the following system

$$\begin{bmatrix} 1 & t_1 & t_1^2 \dots (t_1)^{n-1} \\ 1 & t_2 & t_2^2 \dots (t_2)^{n-1} \\ \vdots & & \\ 1 & t_n & t_n^2 \dots (t_n)^{n-1} \end{bmatrix} t = y$$

Suppose that $Vz = 0$ where z is some vector. This implies that a polynomial of degree $n-1$ has n roots (ie $t_1 \dots t_n$), which can only take place if the resulting polynomial is the zero polynomial, which implies that $z = 0$. This tells us that under this basis, there necessarily exists an interpolant.

Definition 21.9. Lagrange interpolation seeks to make a basis V that is the identity matrix. To that end we define

$$\phi_j = \frac{\prod_{i \neq j}(x - x_i}{\prod_{i \neq j}(x_i - x_j)}$$

Then observe that $\phi_j(x_j) = 1$ and that $\phi_j(x_i) = 0$ for any $i \neq j$.

Definition 21.10. Newton polynomials are obtained by using the basis function

$$\phi_j(x) = \prod_{k=1}^{j-1} (x - x_k)$$

where $\phi_1 = 1$. As a consequence, this gives us a matrix that is lower triangular.

Finding the α vector then is an $O(n^2)$ process.

Remark 21.11. The conditioning of determining an interpolant via a basis matrix is approximately the conditioning of the underlying linear system. We need to also pay attention, however, to the behavior of the interpolant at points other than the nodes, for it should not be the case that an interpolant swings wildly at other points. This is foramlzied by saying that:

$$\max_{x \in [a,b]} |f(x)| \leq \Lambda_i \|y\|_\infty$$

where Λ is a constant that depends on the points chosen for the interpolant, the basis of functions, and (we're working with polynomials of degree at most i), then the degree i.

In general, for polynomials, since all bases give all possible polynomials, the constant solely depends on the points and the maximal degree i.

Definition 21.12. Polynomials are orthogona iff over an interval (for example $[-1, 1]$), given some weight function $w(x)$, we have

$$\int_{-1}^{1} f(x)g(x)w(x) = 0$$

This can be shown to induce an inner product and, hence, we can invoke Gram Schmidt on a collection of n polynomials to obtain a basis for $\mathbb{P}_{n-1}$, the vector space of all polynomials of degree $n-1$.

Remark 21.13. This is useful, because if our interpolants constitute a basis, then we can be assured that we are representing as many functions as possible; it also turns out to be true that when we solve the normal equations $Va = y$ where V is the vandermonde matrix, the matrix V^TV is diagonal (UNRESOLVED: there is some matrix that – according to Heath – ends up being diagonal, but it is not known whether this is V^TV). Orthogonal polynomials also satisfy a three term recurrence, making computation of successive polynomials tractable.

Definition 21.14. In one special case, if we let the basis functions be

$$\phi_j(x) = \cos(j\cos^{-1}(x))$$

and we use equispaced points as our nodes, ie we use

$$x_i = \cos\left(\frac{i}{k}\pi\right)$$

then the entries of the vandermonde matrix, say $V_{i,j}$ end up being

$$\cos\left(j(i/k)\pi\right)$$

It can be shown that the mat-vec of the resultant matrix by any other vector is an $n\log n$ process. This is somehow related to the fourier transform. There are other useful properties of the chebyshev polynomials that Heath discusses.

22 Lecture 23

22.1 Quiz

Problem 2.

see the code below that is commented and hence won't appear on th erendered page

Problem 3.

Let hte interpolant be $a_0 + a_1x$.

$$\implies a_0 + a_1(41.08) = 1.5945575554$$

$$\implies a_0 + a_1(41.09) = 1.5944984844$$

Solve for a_0 and a_1
will give us an ouput that is less than $f(41.08)$.
Suppose that we estimate a function using linear interpolation. The error is then bounded by:

$$\frac{f''(\zeta)}{2!}h^n$$

Nah use the bound h^nCM and take C and M to be 1 and h is by definition the distance between endpoints of the interval, which in this case is 0.04.

Problem 5.

Assume that the error bound is given by

$$h^nCM$$

Remark 22.1. We jsut learned that the error of any interpolant which is a polynomial to a function f where p is at most of degree $n-1$ satisfies the result that the error $p_{n-1}(x) - f(x)$

$$= \frac{f^n(\zeta)}{n!}\prod_{i=1}^{n}(x - x_i)$$

for some $\zeta \in I$, which is a closed intervable on which f is n times continuously differentiable.

We can use this to obtain a generic error bound on our interpolant:
Assume that $x_i \in [x_1, x_n] = h$. Then

$$\max|p_{n-1}(x) - f(x)| \leq h^n \frac{f^n(\zeta)}{n!} = h^n \frac{MC}{1}$$

where we assume that $f^n(x) \leq M$ for all $x \in I$ and C is some constant independent of h.

Remark 22.2. How to perform piecewise polynomial interpolation of degree n.

- Assume that we're given two points (x_1, y_1) and (x_2, y_2) for one interval and that there are N intervals. Then form the interpolant:

  $$a_0 + a_1x + a_2x^2 + ... a_nx^n = f(x)$$

  and solve for a_i using $f(x_1) = y_1$ and $f(x_2) = y_2$.

  - In general, observe that there will be $(n+1)(N)$ unknowns.
    - For cubic splines, there are $4N$ unknowns
  - This gives us two equations for one interval.
    - In total, there fore we have $2N$ equations here.

- Add some smoothness conditions; ie the derivative of the polynomial in one interval must agree with the derivative in the next interval and we get more equations. Make this hold with respect to second derivatives as well.

  - This gives us $2(N-1)$ equations.

    - Finally require that the second derivatives be zero at each of the endpoints.
        - This gives us 2 equations.
- At this point, we have $4N$ equations. If we're working with cubic splines, then we have as many unknowns as equations and we can proceed to solve the system.
    - This construction above is called a cubic spline

23 Lecture 24

Composite Quadrature and introduction to differentiation

Definition 23.1. In a composite quadrature rule, we take an interval with endpoints a, b we subdivide it into k intervals; then we interpolate over these k intervals, each interval having its own interpolant. Using these interpolants, we then construct integrals.

Remark 23.2. Suppose that inside an interval, we use n points and that the interval is open – then we will need kn interpolation points. On the other hand, if we interpolate over closed intervals, then we will need $(n-1)k+1$ points. Why? For suppose that the closed interval is first really an open interval, meaning that it requires kn points. Now correct our mistaken assumption, by realizing that we have overcounted some points. In fact, we have over counted those points on the boundaries of intervals, each point exactly once – there are $k-1$ such points so the total number of points needed is really $kn-(k-1) = (n-1)k+1$.

Example 23.3. Suppose that we divide an interval into k parts, each part solely containing two points and that the first point is a, the last point b, and points in between are given by $x_j = a+(b-a)/kj$ from $j = 0$ up to $j = k$. Then we apply the composite midpoint rule we find that the approximation is

$$\sum_{j=0}^{k-1} f(\frac{x_j + x_{j+1}}{2})(x_{j+1} - x_j)$$

$$= h \sum_{j=0}^{k-1} f(\frac{x_j + x_{j+1}}{2})$$

We can similarly derive the composite trapezoid rule.

The following is not proven (although shown in Heath 8.3.5):

Proposition 23.4. In the limit – as we use more and more subintervals of equal length – provided that the underlying quadrature rule is stable, composite quadrature gives us an approximation to the integral.

Unresolved 23.5. Do all interpolation formulas give the same interpolant?

<++>

24 Lecture 25

Definition 24.1. An ordinary differential equation is a differential equation where the state vector depends only on one variable, usually time. A partial differential equation has the state vector depend on several variables.

Definition 24.2. A partial differential equation is in implicit form, if for each derivative of the state vector y that we denote by y^k, we have

$$f(t, y, y^2, \dots y^k) = 0$$

By contrast, an ODE is said to be in explicit form is we can write y^k as

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

Definition 24.3. The order of an ODE is the degree of the highest derivative that occurs in the implicit form representation of an ODE.

Remark 24.4. We will restrict our attention to first order ODEs, for suppose we are given an explicit ODE, then we can define $u_k = y^k = f(t, y, y'', \dots y^{k-1})$ and we will find that

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_1' \\ u_2' \\ u_3' \end{bmatrix}$$

which is a system of kn first order equations (here, realize that u_k is a vector in $\mathbb{R}^n$).

Remark 24.5. We can numerically solve for an ordinary differential equation $y' = f(t, y)$ by first making the following approximation:

$$\begin{aligned} &\text{Let} \\ y_0 = &\text{ The initial position} \\ t_0 = &\text{ The initial time} \\ h_0 = &\text{ The distance between } t_0 \text{ and } t_1 \end{aligned}$$

In some sense, the slope at time t_0 is given by

$$\begin{aligned} &f(t_0, y_0) \\ &\text{Approximate} \\ y_1 &= y_0 + h_0 f(t_0, y_0) \end{aligned}$$

Note that this approximation can also be understood as a difference quotient:

$$\frac{y_{k+1} - y_k}{h_k} = f(t_k, y_k)$$

Remark 24.6. The equation $y' = f(y,t)$ by itself doe snot completely specify the solution y. For the ODE merely specifies the slope of a solution and, thus, there are in principle an infinite number of solutions to it. We can often constrain the class of solutions to a unique solution, however, if we additionally maintain the solution must satisfy an initial condition (t_0, y_0). Note, however, that this is merely a necessary condition and not a sufficient one for the solution to an ODE to be unique..

<++>

Definition 24.7. Suppose that we are presented with the ODE $y' = f(t,y)$ with intiial conditions (t_0, y_0). SUppose that $\widehat{y}(t)$ is a solution to the ODE. Then we say that $\widehat{y}(t)$ is a stable solution if it holds that for every ϵ there exists some δ such that whenever $|\widehat{y}(t_0) - \widetilde{y}(t_0)| \leq \delta$ where $\widetilde{y}$ is some other solution, then $|\widetilde{y}(t) - \widehat{y}(t)| \leq \epsilon$ for all $t \geq t_0$. We say that $\widetilde{y}$ is asymptotically stable if $|\widetilde{y}(t) - \widehat{y}(t)| \leq 0$ as $t \to \infty$.

<++>

25 Lecture 26

Definition 25.1. Let t_k be some time at iteration k. Let y_k be the computed solution at t_k. Let $y(t_k)$ be the true solution of hte ODE (given initial conditions y_0, t_0). Then

global error is the error that we've made in an unforgiving sense: it $e_k \coloneqq y_k - y(t_k)$.

By contrast, local error is the error that we make at iteration k assuming that our guess at iteration $k-1$ was accurate. That is, we let $l_k \coloneqq y_k - u_{k-t}(t_k)$ where u_k is the solution to the ODE with initial conditions given by (t_{k-1}, y_{k-1}).

26 Lecture 28

Definition 26.1. A boundary value problem is of the form

$$u'(x) = f(u(x))$$

and $g(u(a), u(b)) = 0$ is the boundary condition.
where $u : [a,b] \to \mathbb{R}^n$

Remark 26.2. A boundary value problem has a solution (it is hinted in lecture) if the IVP

$$u'(x) = f(u(x))$$

with initial condition (t_0, e_i)
is solveable for all unit vectors.

Definition 26.3. The shooting method does the following:
Suppose that we're handed

$$u''(x) = f(u(x)) \tag{1}$$
$$u(a) = \text{ some vector} \tag{2}$$
$$u(b) = \text{ some vector} \tag{3}$$
$$\tag{4}$$

Then we will set $u'(a)$ to a random value and check to see whether the resulting IVP gives a solution that matches the value of $u(b)$ above.