1 Lectures 5 and 6

1.1 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 1.1. 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 1.2. 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 1.3. 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 1.4. 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 1.5. $k(\gamma A) = k(A)$. 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 1.6. 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.

1.2 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 1.7. $\|A\| = 14$. 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,

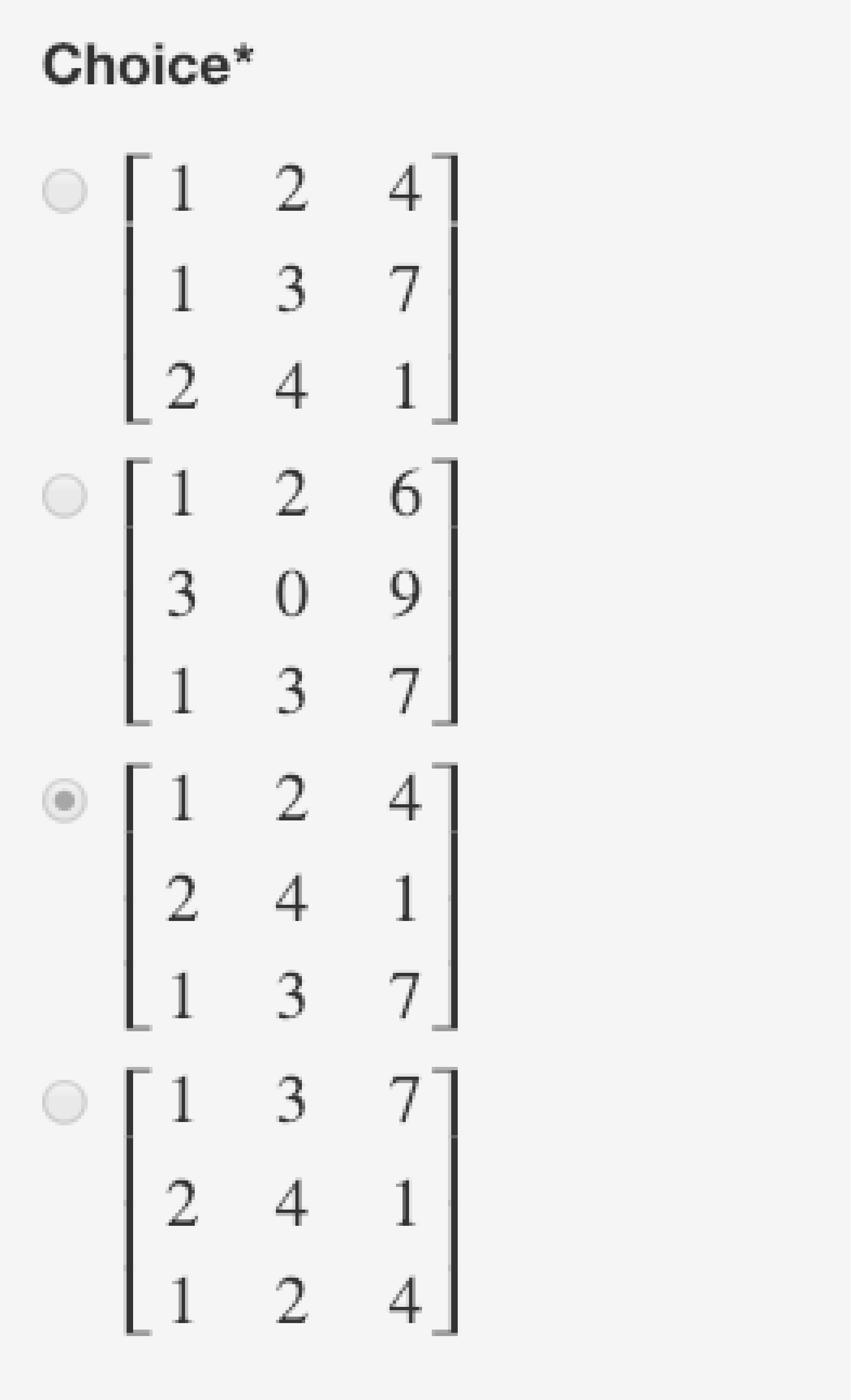
$$A = \begin{bmatrix} 2 & 1 & 3 \\ 2 & 4 & 8 \\ 4 & -7 & 4 \end{bmatrix}$$

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 1.8. I have



Choice*

○ $\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 7 \\ 2 & 4 & 1 \end{bmatrix}$

○ $\begin{bmatrix} 1 & 2 & 6 \\ 3 & 0 & 9 \\ 1 & 3 & 7 \end{bmatrix}$

● $\begin{bmatrix} 1 & 2 & 4 \\ 2 & 4 & 1 \\ 1 & 3 & 7 \end{bmatrix}$

○ $\begin{bmatrix} 1 & 3 & 7 \\ 2 & 4 & 1 \\ 1 & 2 & 4 \end{bmatrix}$

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

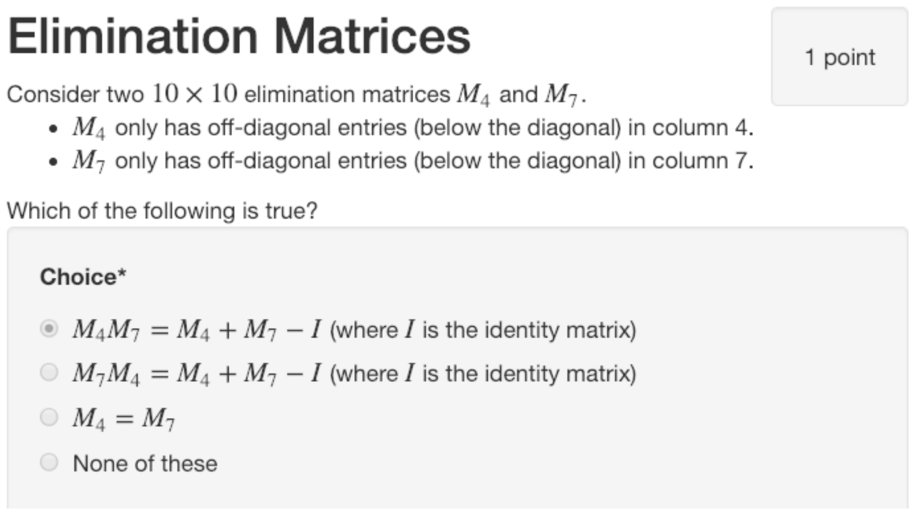
LU factorization.

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.

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

Remark 1.9. 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:

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

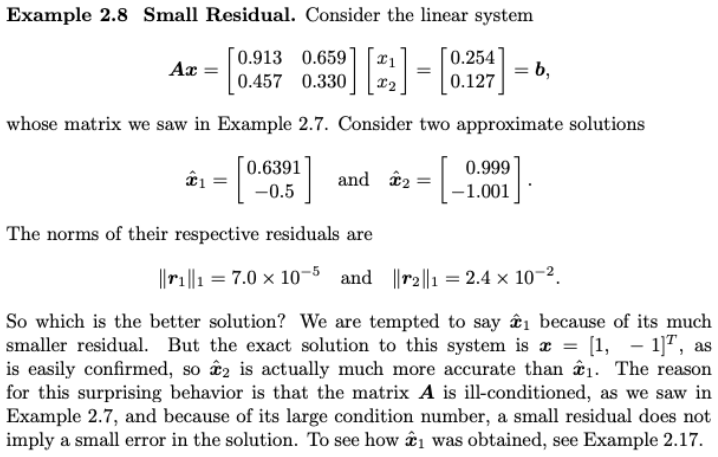
We can obtain a bound on the relative forward error required to solve $Ax = b$ in terms of r.

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

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

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

Remark 1.10. 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

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

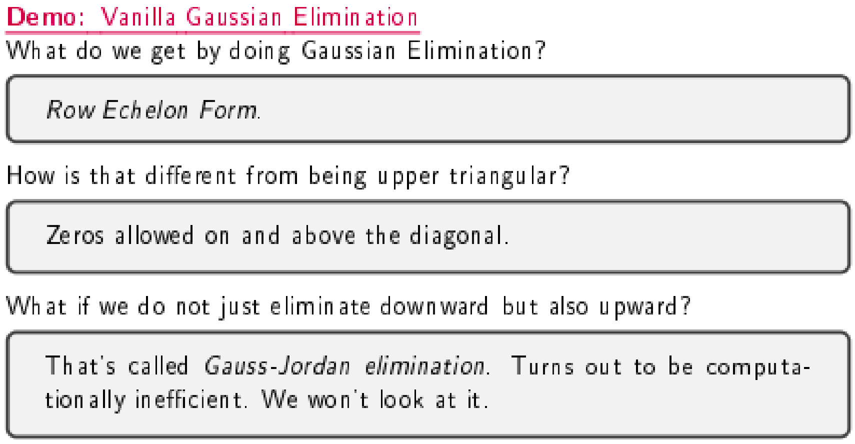
whose matrix we saw in Example 2.7. Consider two approximate solutions

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

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 $\boldsymbol{x} = [1, \quad -1]^T$, 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.

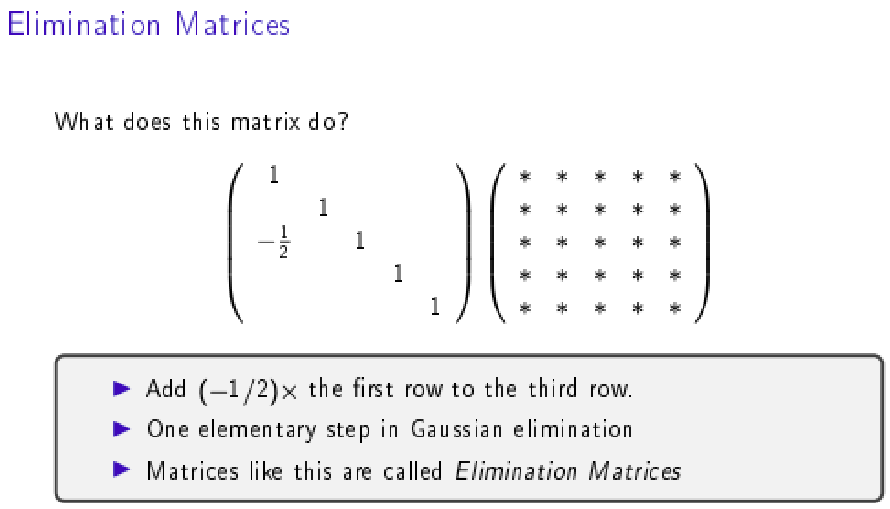
How is that different from being upper triangular?

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 1.11. 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.



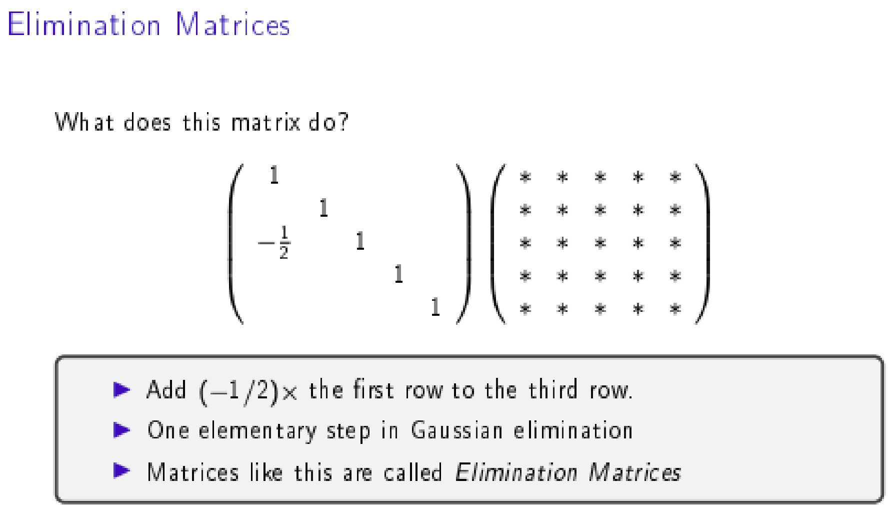
Elimination Matrices

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 1.12. 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?

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

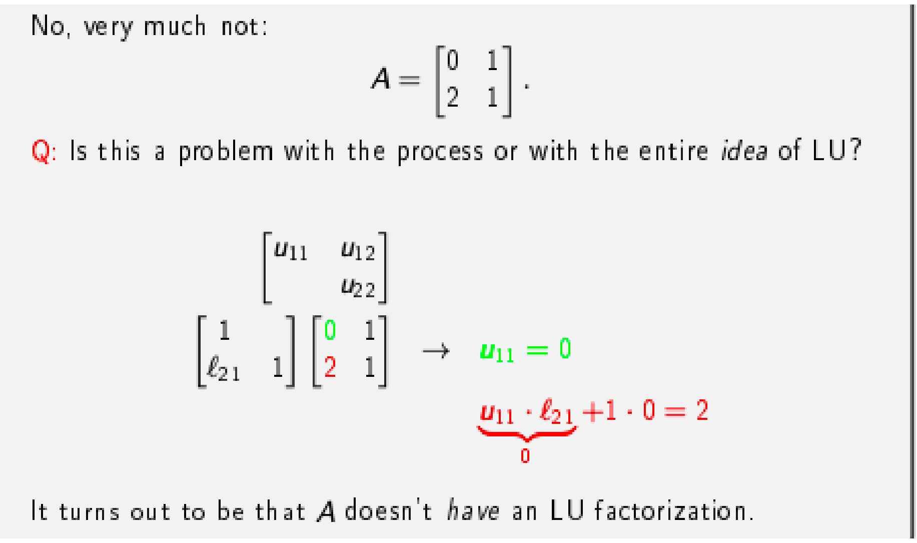
- Add $(-1/2)\times$ the first row to the third row.
- One elementary step in Gaussian elimination
- Matrices like this are called *Elimination Matrices*

Remark 1.13. 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[1]

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?

$$\begin{bmatrix} u_{11} & u_{12} \\ & u_{22} \end{bmatrix}$$

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

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

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

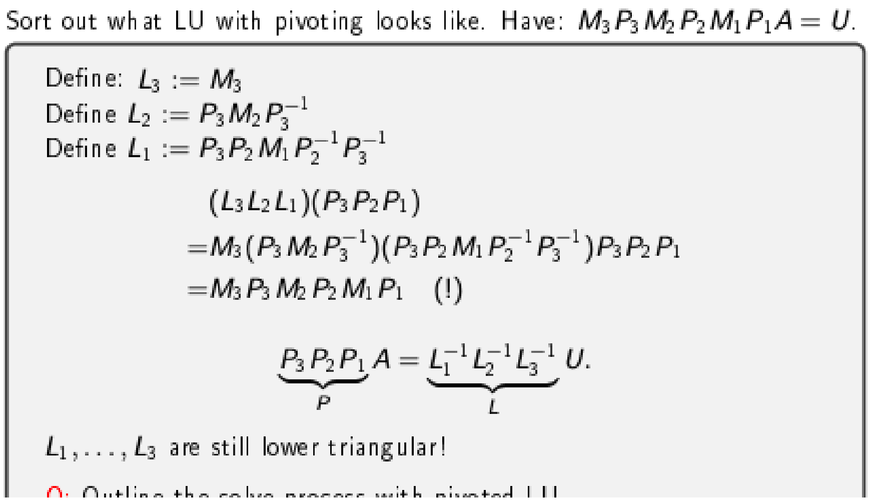
The solution is to repeatedly apply permutations to A (in the form of permutation matrices) so that the pivot is the largest element in terms of absolute value in its column.

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.

[1] 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: $M_3P_3M_2P_2M_1P_1A = U$.

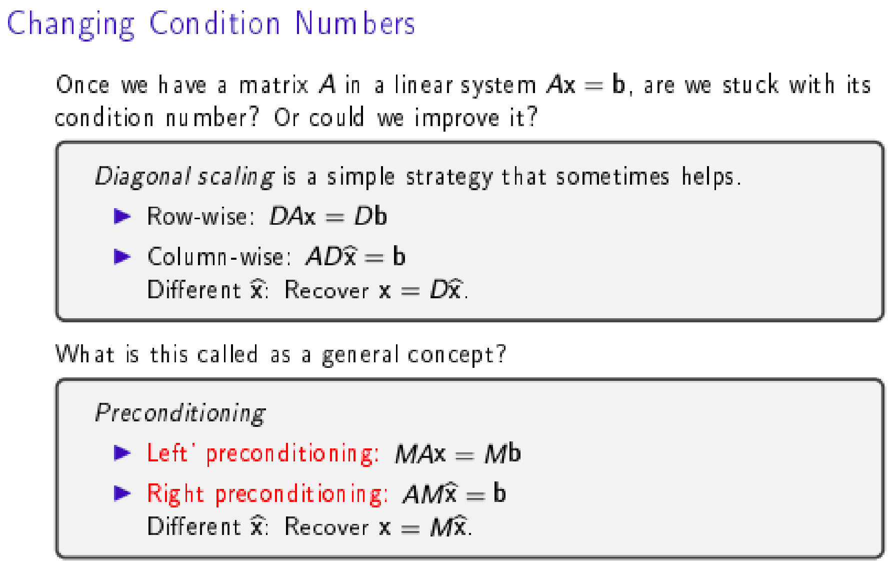
Define: $L_3 := M_3$
Define $L_2 := P_3M_2P_3^{-1}$
Define $L_1 := P_3P_2M_1P_2^{-1}P_3^{-1}$

$$\begin{aligned}&(L_3L_2L_1)(P_3P_2P_1)\\ =&M_3(P_3M_2P_3^{-1})(P_3P_2M_1P_2^{-1}P_3^{-1})P_3P_2P_1\\ =&M_3P_3M_2P_2M_1P_1 \quad (!)\end{aligned}$$

$$\underbrace{P_3P_2P_1}_{P}A = \underbrace{L_1^{-1}L_2^{-1}L_3^{-1}}_{L}U.$$

$L_1, \dots, L_3$ 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: $AD\widehat{\mathbf{x}} = \mathbf{b}$
  Different $\widehat{\mathbf{x}}$: Recover $\mathbf{x} = D\widehat{\mathbf{x}}$.

What is this called as a general concept?

Preconditioning

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

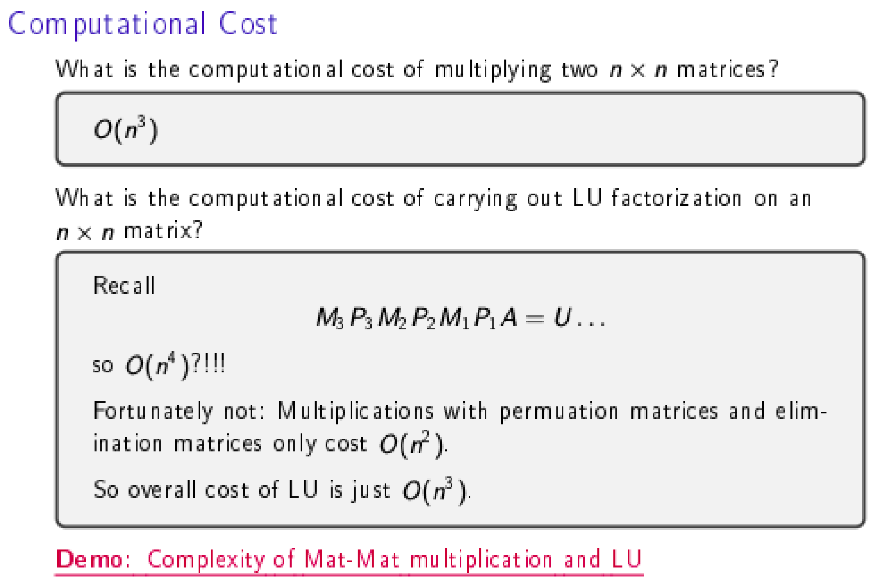
Remark 1.14. Suppose that D above satisfies $k(D) \approx 1$. Then

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

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 1.15.



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

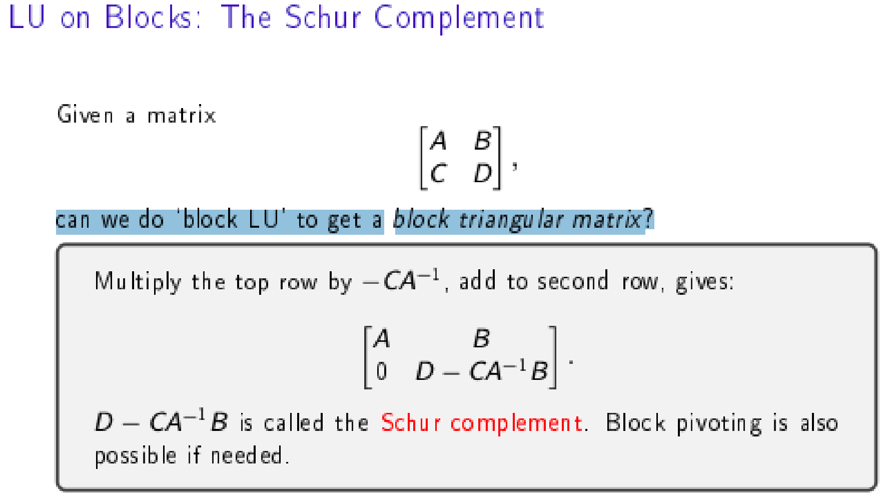
Fortunately not: Multiplications with permuation matrices and elimination matrices only cost $O(n^2)$.

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 1.16.



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*?

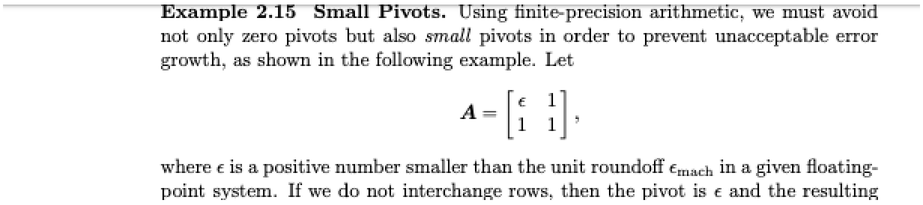
Multiply the top row by $-CA^{-1}$, add to second row, gives:

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

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

Not sure why this is significant.

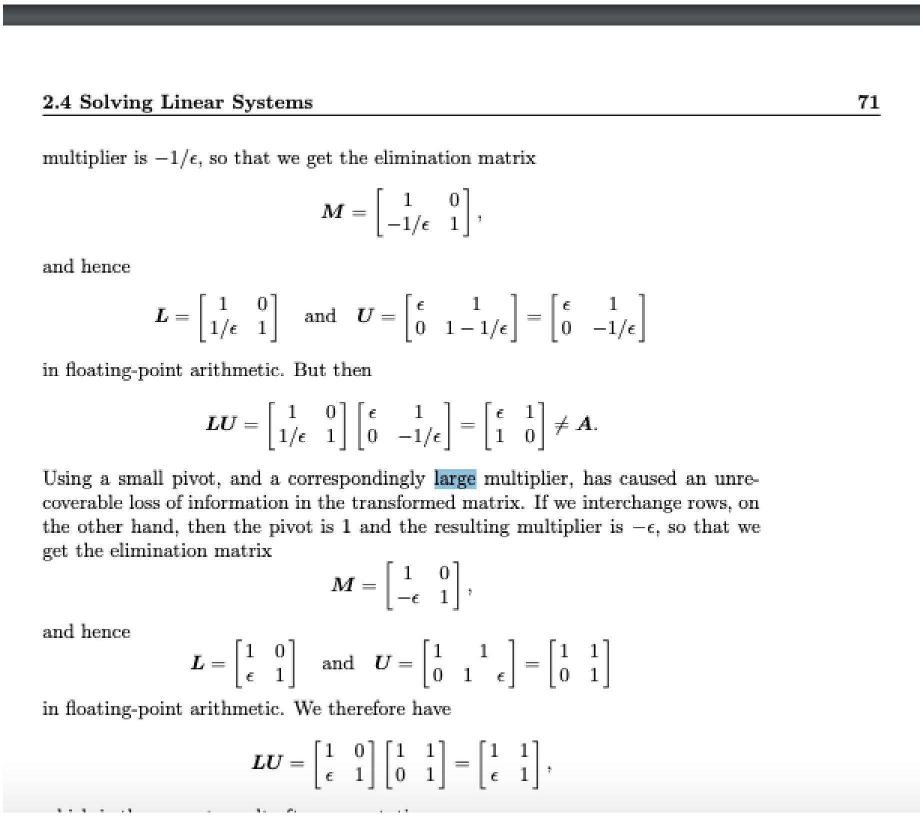
Remark 1.17. Unresolved



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



2.4 Solving Linear Systems 71

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

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

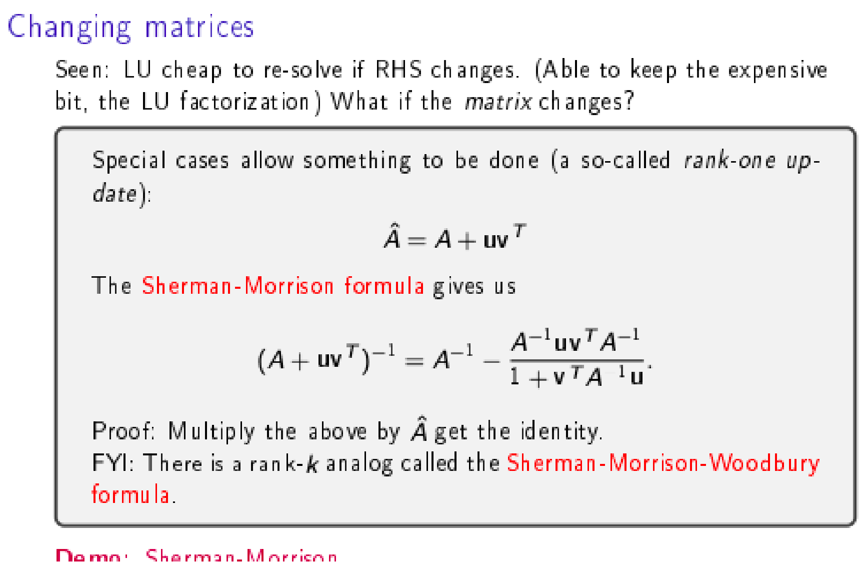
and hence

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

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 1.18. 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.

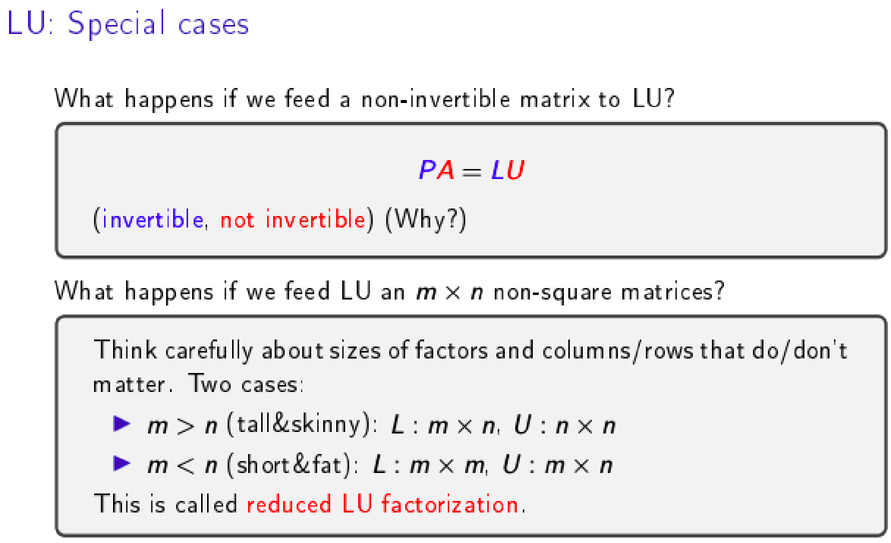
Demo: Sherman-Morrison

For

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

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 1.19.



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.

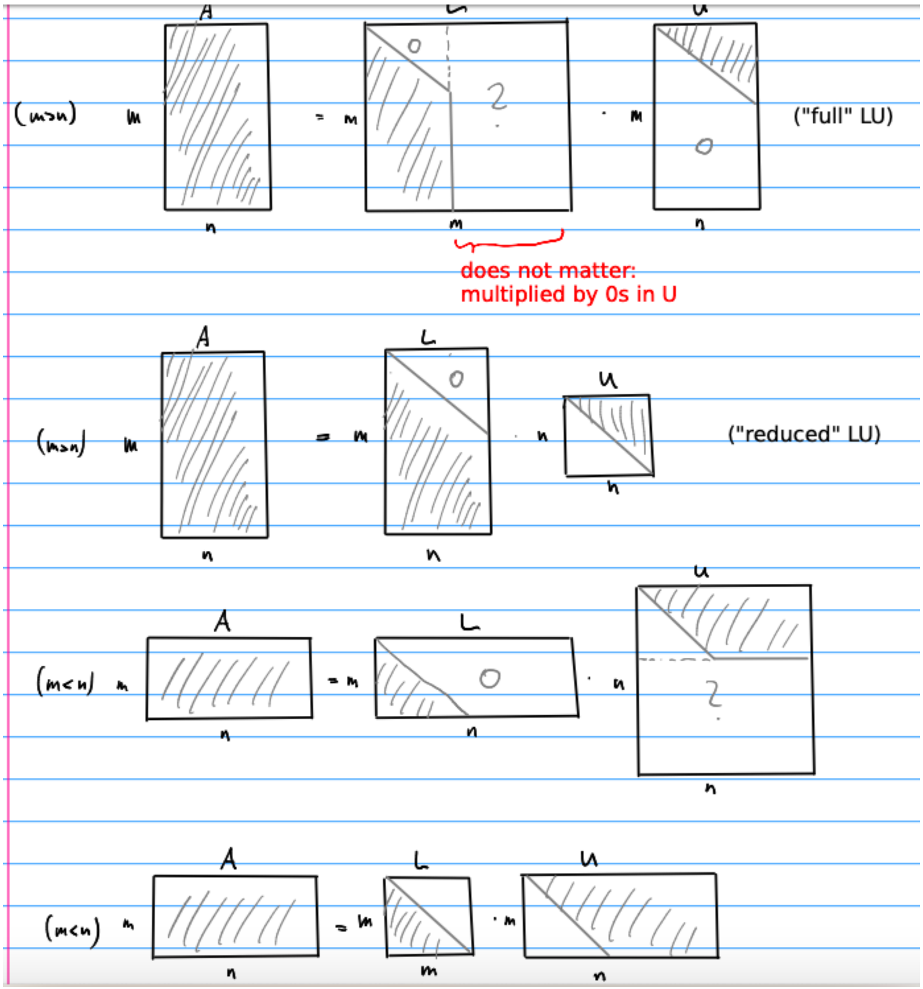
A matrix A always admits an LU factorization, even if A is singular. First, observe that every column of A must contain at least one non-zero number – or else, why would the column be part of A. Thus, if a pivot entry does not contain a non-zero value, we can rotate rows so that the pivot entry does 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 1.20. 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 $\{n+1 \ldots m\}$ 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 $n > m$.

2 Lecture 7

Least Squares

2.1 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 2.1. 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 2.2. 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 2.3. 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 2.4.

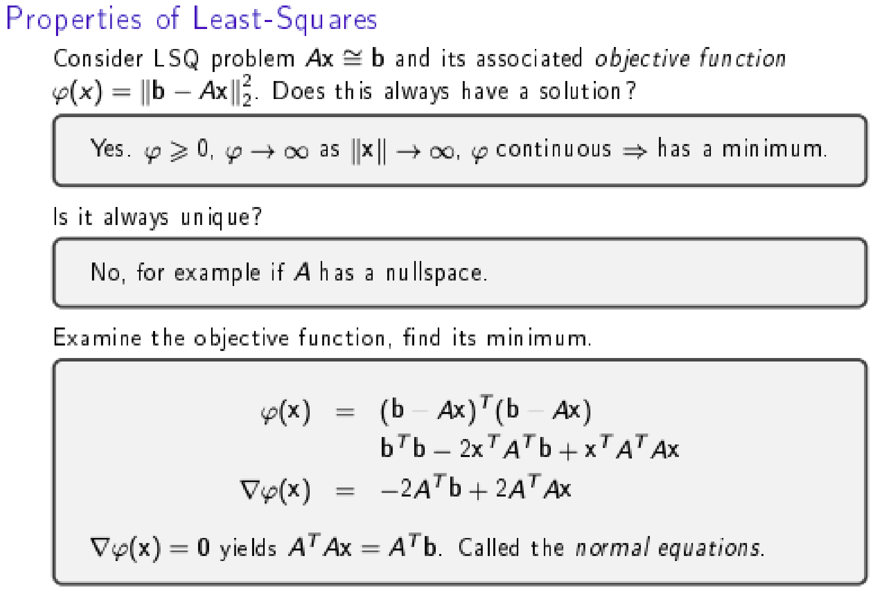
$$(9 \times 9) \times (9 \times 24)$$

Problem 6.

Which problem requires the largest amount of work:
kkkkkk

Remark 2.5. We assume that we work with tall, skinny matrices that have full column rank.

Remark 2.6.



Properties of Least-Squares

Consider LSQ problem $A\mathbf{x} \cong \mathbf{b}$ and its associated *objective function* $\varphi(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2$. 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.

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

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

The textbook proves that there is always a unique vector $y \in \mathrm{Span}(A)$ 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 2.7. 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 2.8. 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 2.9. Given an orthogonal projection P, any vector x can be expressed as $x = Px + P_\perp x$.

Proposition 2.10. 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 2.11. The x aforementioned is $(A^T A)^{-1} A^T b$

Proof.

$$\begin{aligned} Ax &= Pb \\ \iff A^T Ax &= A^T Pb \\ \iff A^T Ax &= (PA)^T b \\ \iff A^T Ax &= (A)^T b \\ \iff x &= (A^T A)^{-1} (A)^T b \end{aligned}$$

□

Proposition 2.12. $P = (A^T A)^{-1} A^T$ 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 2.13. 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 2.14. 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.

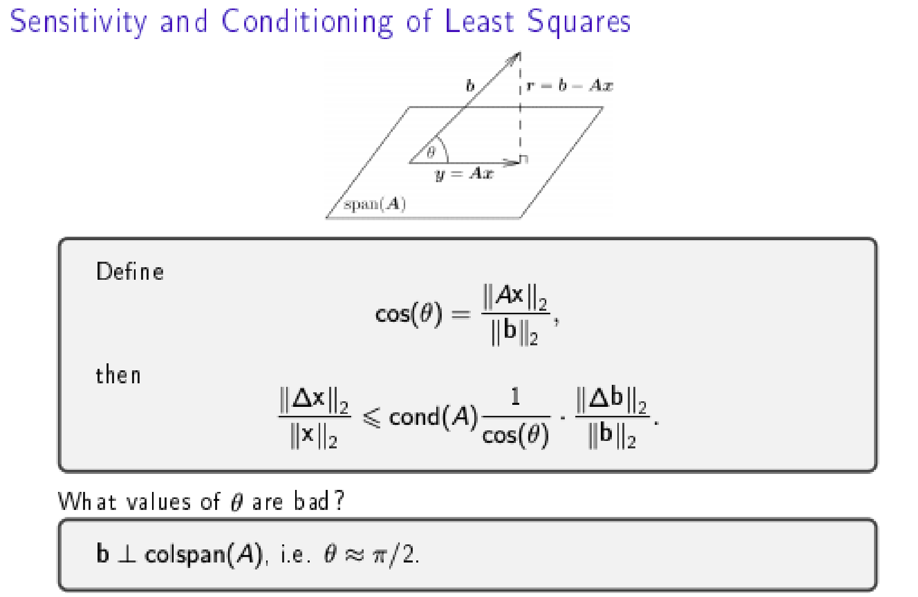
Proof. $(QQ^T)(QQ^T) = QQ^T$. Thus, this matrix is a projection; is it also clearly symmetric; finally, note that its span is precisely the span of A. □

Corollary 2.15. 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 2.16. I take the following as definitions. No time to look into their proofs:


b
r = b − Ax
θ
y = Ax
span(A)

Sensitivity and Conditioning of Least Squares

Define

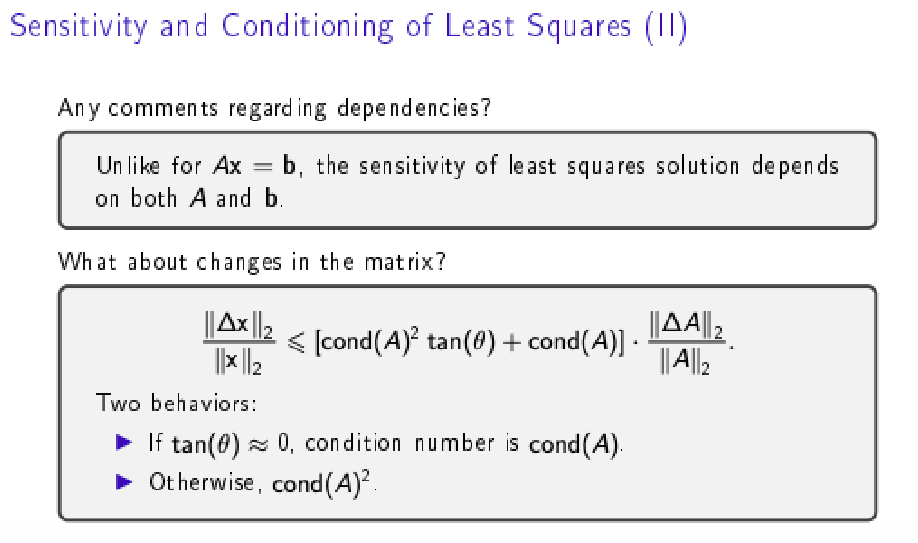
$$\cos(\theta) = \frac{\|Ax\|_2}{\|\mathbf{b}\|_2},$$

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, $\text{cond}(A)^2$.

3 Lecture 8

Problem Transformations

3.1 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 3.1. The solution is $(A^TA)^{-1}A^Tb$. Remember that the inverse of a general matrix is x^2.

Proposition 3.2. $k(A^TA) = k(A)^2$.

Proof.

- Recall that $\|A\| = \max_\sigma(A)$.
- $(A^TA)^T(A^TA) = (A^TA)^2$.
  - It follows that $\|A^TA\| = \|A\|^2$.
- 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 3.3. 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 3.4. If Q is orthogonal, then $\|v\| = \|Qv\|$.

3.2 Householder Transformations

Remark 3.5. 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 3.6. Suppose that we obtain a system of the form:

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

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 3.7. 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 3.8. 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 $Q_1 R$ is unique if we force the diagonal entries of R to be positive

$Q_1 R$ 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 3.9. Observe that many methods to compute a QR factorization rely on forming Q multiplying successive orthogonal matrices like

$$Q_n Q_{n-1} \cdots Q_1$$

To obtain Q, we need to use all of the left factor. By this, I mean that if we compute $Q_2 Q_1$, 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.

<++>

4 Lecture 9

Problem Transformations II – February 5th

Definition 4.1. Given a matrix v, a projection matrix onto span v is $\frac{vv^T}{v^T v}$. This is unique, I think. In any case, this projection matrix is symmetric and satisfies $P^2 = P$, making it an orthogonal transformation.

Proposition 4.2. 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

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

Proof.

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

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 $\left\{x | v^T x = 0\right\}$, 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

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

□

Proposition 4.3. 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 4.4. 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 4.5. 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 4.6. 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 $A_{a,a} = M_{0,0}$, $A_{a,b} = M_{0,1}$, $A_{b,a} = M_{1,0}$, $A_{b,b} = M_{1,1}$.

5 Lecture 10

SVD

Definition 5.1. 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 5.2. If $\|A\| = \|\Sigma\| = \sigma_1$ where σ_1 is the largest diagonal entry in Σ.

Remark 5.3. If a singular value appearing in Σ is negative, can it be made positive?

Solution 5.4. Yes, flip an appropriate singular vector in sign. UNRESOLVED.

Proposition 5.5. 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 5.6. 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 5.7. 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 5.8. 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 5.9. 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 5.10. 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 5.11. 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 5.12. 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 5.13. Using a k rank approximation is not unambiguously good, for computation of the SVD requires $O(n^3)$ time.

Unresolved 5.14. 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 5.15.

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 $\widehat{y}$ that solve $\Sigma \widehat{y} \cong U^T b$. 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 5.16. 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 5.17. Remember the cost of householder for nonsquare and perhaps the cost for all $n \times n$ matrix.

6 Lecture 11

Eigenvalue Problems

Definition 6.1. 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 6.2. Similar matrices share the same eigenvalues.

Theorem 6.3. If a matrix is defective, then it cannot have an eigenvector basis.

Proposition 6.4. A matrix is diagonalizable iff it has an eigenvector basis.

Proposition 6.5. 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 6.6. 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 $k(A)\|E\|$.

7 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 7.1. 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 7.2. 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 7.3. 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 7.4. $X^{-1}\hat{A}X = X^{-1}AX + X^{-1}\delta AX = D + X^{-1}\delta AX$

Definition 7.5. The eigenvector corresponding to the largest eigenvalue will hence forth be called the maximal eigenvector.

Remark 7.6. 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 7.7. The rayleight quotient is the quantity

$$\frac{x^TAx}{x^Tx}$$

Remark 7.8. Let $e_k = \|v_1^k - x_1\|$ 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 7.9. 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 7.10. 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 7.11. 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 7.12. 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 7.13. 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 7.14. 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 7.15. 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 7.16. 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 7.17. 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 $\hat{Q}_k := \hat{Q}_{k-1}Q_k$, and appeal to the uniquess of QR, then we are done.

Definition 7.18. 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 7.19. 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 7.20. 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 7.21. 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. □

8 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 8.1. 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 8.2. 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 8.3.

- 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 8.4. 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 8.5. 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 8.6. 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 8.7. In a Krylov subspace method, we restrict our focus to finding a few eigenvect

<++>

Definition 8.8.

- 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

$$AK_n = [Ax_0 \quad A^2x_0 \quad ... \quad A^nx_0] = K_n \underbrace{[e_2 \quad e_3 \quad ... \quad e_n \quad K_n^{-1}A^nx_0]}_{H}$$

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

Remark 8.9. 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 8.10.

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

17

where C is some Hesenberg matrix in addition to H.

Proposition 8.11. 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 $Q_n =$

$$[q_1 \quad q_2 \dots q_n] \implies Aq_j = \sum_{i=1}^{j+1} q_iC_{i,j}$$

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

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

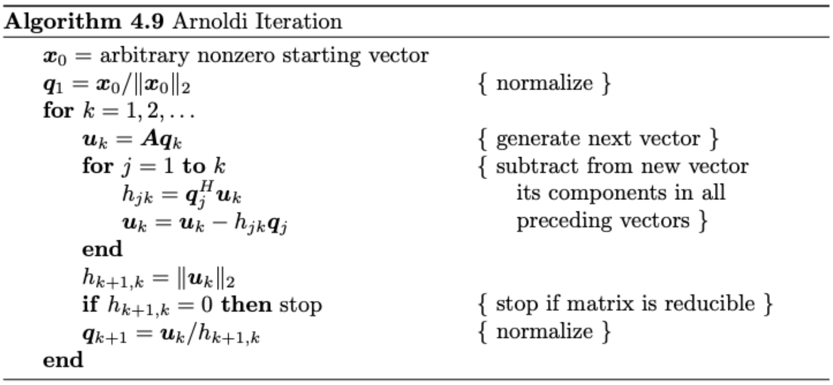
Note that Aq_1 is the “next” vector in a Krylov subspace that one would compute with knowledge of q_0.

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 8.12. 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 $k = 1, 2, \ldots$
    $u_k = Aq_k$                                { 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 8.13. 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 8.14. 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

$$Q_k = [q_1 \quad ... \, q_k]$$

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

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

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

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

□

9 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 9.1. 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 9.2. 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 9.3. See the notes previously that explain that if we gather the fist k Arnoldi vectors into a matrix Q then $Q^T AQ$ 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 9.4. 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 9.5. 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 9.6. Why do we favor symmetric problems in terms of condition number?

Remark 9.7. 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 9.8. 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.

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

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 9.9. 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 9.10. 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 9.11.** 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 9.12. The condition number of solving a root problem is the same as the condition number of evaluating the inverse function at 0.

Proof.

Lemma 9.13. 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 9.14. A function has a root of multiplicity m if it holds that

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

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

Remark 9.15. 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 9.16. 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 9.17. 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.

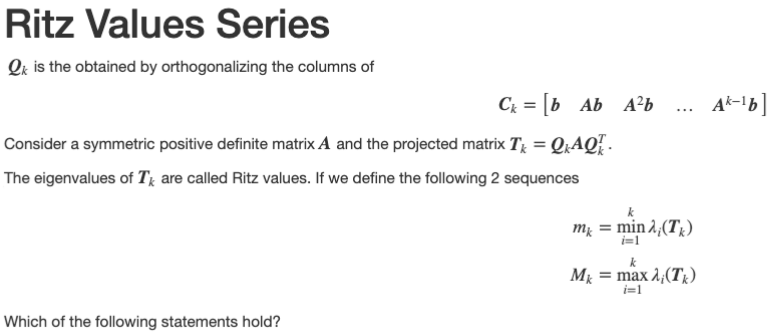
Definition 9.18. One of the following criteria for convergence if often employed, although all criterion have their flaws:

- $$|f(x)| < \epsilon$$
  - 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.

10 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 10.1. 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 10.2. 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 10.3. 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 $[x_0 A x_0 \dots A^{k-1} x_0]$.

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 10.4. 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 10.5. 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 10.6. 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 10.7. 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 10.8. 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 10.9. 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 10.10. 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.