

# Applied Math HW 5

Colin Williams

November 6, 2021

## Question 1

Write in pseudocode the algorithms for the Gaussian elimination method, the LU factorization method, and the Cholesky factorization method.

- Gaussian Elimination:

(For solving  $Ax = b$ , let  $A$  represent the augmented matrix  $[A|b]$ )

```
for j = 1 to n
  if A[j,j] = 0
    Swap row j with next row that has non-zero j-th entry.
    If no such non-zero entry exists, return error.
  for i = j + 1 to n
    Take  $(-A[i,j]/A[j,j])$  times row j and add it to row i.

x[n] = b[n]/A[n,n]
for i = n-1 to 1
  x[i] = (b[i] - (sum(j=i+1 to n) of A[i,j]*x[j]))/A[i,i]
```

- LU factorization

$L$  =  $n \times n$  identity matrix

$U$  =  $n \times n$  matrix with first row equal to first row of  $A$ , and the rest zeroes

```
for j = 1 to n
  for i = 2 to j
    U[i,j] = A[i,j] - (sum(k=1 to i-1) of L[i,k]*U[k,j])
  if U[j,j] = 0, return error
  for i = j+1 to n
    L[i,j] = (A[i,j] - (sum(k=1 to j-1) of L[i,k]*U[k,j]))/U[j,j]
```

- Cholesky factorization

```
for j = 1 to n
  B[j,j] = sqrt(A[j,j] - (sum(k=1 to j-1) of B[j,k]^2))
  if B[j,j] is 0 or imaginary, return error
  for i = j+1 to n
    B[i,j] = (A[i,j] - (sum(k=1 to j-1) of B[j,k]*B[i,k]))/B[j,j]
```

## Question 2

Compare the pros and cons of these direct methods for solving linear systems.

**Answer.**

The pros of Gaussian Elimination are primarily the fact that it is the easiest computation to do by hand among these methods. If we needed to solve the system  $Ax = b$  only one time, this would be the most straightforward method. The computational complexity of solving this system is  $\frac{n^3}{3} + O(n^2)$ . One con of Gaussian Elimination is that if we wanted to solve  $Ax = \hat{b}$  for  $\hat{b} \neq b$ , then we would need to repeat the entire process which makes the complexity  $\frac{n^3}{3} + O(n^2)$  again.

LU factorization has the same initial overhead as Gaussian Elimination. Namely, it takes approximately  $\frac{n^3}{3} + O(n^2)$  operations to calculate the matrices  $L$  and  $U$ . However, one pro of LU factorization is that once we know what  $L$  and  $U$  are, we can solve  $Ax = \hat{b}$  for  $\hat{b} \neq b$  without having to recompute  $L$  and  $U$ . In fact, all we need to do is use back/forward-substitution twice to solve  $Ly = \hat{b}$  and then  $Ux = y$  which has complexity of  $O(n^2)$  since  $L$  and  $U$  are triangular. A con

of LU factorization is that we can only guarantee the factorization will work given that each diagonal submatrix,  $\Delta^{(k)}$ , of  $A$  is invertible. While this condition holds for most applied instances, it still means this method is not as universal as Gaussian Elimination.

Cholesky factorization is the most computationally efficient of the three methods. It only takes  $\frac{n^3}{6} + O(n^2)$  operations to compute the matrix  $B$ . In other words, approximately half as many computations as Gaussian Elimination or LU factorization. In addition, the same property of being able to re-use the factorization to solve systems with new data is true here. However, the con of Cholesky factorization is that we can only guarantee that the factorization works when our matrix  $A$  is symmetric and positive-definite. This is even more restrictive than using LU factorization, making it even less universal, but still rather frequent in many applications.

## Question 3

Write a Matlab code implementing one of the three direct methods we studied for solving a system of 5 equations and five unknowns (submit a Matlab file for the code together with your homework pdf file).

*\*\*See attached file for Cholesky Factorization\*\**

## Question 4

For solving  $Ax = b$  with  $n \times n$  invertible matrix  $A$ , show that the number of multiplications and divisions needed in the Gaussian elimination method and the LU factorization method is  $n^3/3 + O(n^2)$ .

**Answer.**

I will be referring to the pseudo-code I wrote in question 1 in order to obtain my operation count. I will be ignoring my stopping conditions in the code though since  $A$  is invertible, we will not run into any errors.

- Gaussian Elimination

- In my most nested loop, I say **Take  $(-A[i,j]/A[j,j])$  times row  $j$  and add it to row  $i$** . Calculating  $(-A[i,j]/A[j,j])$  takes one operation. At this point in the process, row  $j$  should have zeroes in all of its first  $j - 1$  columns. Thus, multiplying this scalar times each non-zero entry of row  $j$  is  $n - (j - 1) = n - j + 1$  operations. This gives a total of  $n - j + 2$  operations at this step.
- I am then repeating this process for all  $i$  starting at  $j + 1$  and ending at  $n$ . Thus, the number of operations becomes

$$\sum_{i=j+1}^n (n - j + 2) = (n - j)(n - j + 2) = n^2 - 2nj + j^2 + 2n - 2j$$

- Lastly, I am repeating this process for all  $j$  starting at 1 and ending at  $n$ . Therefore, using some elementary summation formulas, the total operation count becomes

$$\begin{aligned} \sum_{j=1}^n n^2 - 2nj + j^2 + 2n - 2j &= \sum_{j=1}^n n^2 - 2n \sum_{j=1}^n j + \sum_{j=1}^n j^2 + \sum_{j=1}^n 2n - 2 \sum_{j=1}^n j \\ &= n^3 - 2n \frac{n(n+1)}{2} + \frac{n(n+1)(2n+1)}{6} + 2n^2 - 2 \frac{n(n+1)}{2} \\ &= n^3 - n^3 - n^2 + \frac{2n^3 + 3n^2 + n}{6} + 2n^2 - n^2 - n \\ &= \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} \\ &= \frac{n^3}{3} + O(n^2) \end{aligned}$$

- Note, I did not count the operations involved in the backwards substitution at the end of the code since that only has complexity of  $O(n^2)$  and will not affect this previous count.

- LU factorization

- In one of my inside loops, I use the phrase **sum(k=1 to i-1) of  $L[i,k]*U[k,j]$** . This summation involves one multiplication for each term from  $k = 1$  to  $i - 1$ . Thus, we have a total of  $i - 1$  operations for this summation.

- We do this previous summation one time for each  $i$  from 2 to  $j$ . Thus, we get a total number of operations for this part is equal to

$$\begin{aligned}
 \sum_{i=2}^j i - 1 &= \sum_{i=2}^j i - \sum_{i=2}^j 1 \\
 &= \sum_{i=1}^j i - 1 - (j - 1) \\
 &= \frac{j(j+1)}{2} - j \\
 &= \frac{j(j-1)}{2}
 \end{aligned}$$

- In the next interior loop, I use the phrase `sum(k=1 to j-1) of L[i,k]*U[k,j]`. This summation also involves only one multiplication per term from  $k = 1$  to  $j - 1$ . Thus, we have a total of  $j - 1$  operations for this summation. Additionally, after doing the summation, we do subtraction and then divide an expression by  $U[j,j]$  which adds one more operation for a total of  $j$  operations.
- We do these previous operations one time for each  $i$  from  $j + 1$  to  $n$ . Thus, we get a total number of operations for this part equal to

$$\sum_{i=j+1}^n j = (n - j)j$$

- Thus, our total number of operations in our first-level loop is equal to

$$\frac{j(j-1)}{2} + nj - j^2 = nj - \frac{j(j+1)}{2}$$

- Lastly, we do these number of operations one for each  $j$  ranging from 1 to  $n$ . In total by using some elementary summation formulas, this gives our operation count as

$$\begin{aligned}
 \sum_{j=1}^n nj - \frac{j(j+1)}{2} &= n \sum_{j=1}^n j - \frac{1}{2} \sum_{j=1}^n j^2 - \frac{1}{2} \sum_{j=1}^n j \\
 &= n \frac{n(n+1)}{2} - \frac{1}{2} \frac{n(n+1)(2n+1)}{6} - \frac{1}{2} \frac{n(n+1)}{2} \\
 &= \frac{n^3 + n^2}{2} - \frac{2n^3 + 3n^2 + n}{12} - \frac{n^2 + n}{4} \\
 &= \frac{6n^3 + 6n^2 - 2n^3 - 3n^2 - n - 3n^2 - 3n}{12} \\
 &= \frac{4n^3 - 4n}{12} \\
 &= \frac{n^3}{3} + O(n^2)
 \end{aligned}$$

- Note, it may have been more accurate to write this using  $O(n)$  instead of  $O(n^2)$ , but if something has complexity  $O(n)$  then it trivially also has complexity  $O(n^2)$  since that is a weaker statement. Additionally, once we finish the LU factorization, we need to do back substitution to find our solution which *does* have  $O(n^2)$  complexity, so our simplification is fine.

## Question 5

Prove the uniqueness of the LU factorization we studied in the lecture (note that we already proved the existence of the LU factorization you just need to prove the uniqueness part.)

*Proof.*

Assume that  $A = LU$  and  $A = MV$  are two lower-upper factorizations. This means  $L, M$  are both lower triangular with  $l_{ii} = m_{ii} = 1$  for all  $i = 1, 2, \dots, n$  and that  $U$  and  $V$  are both upper triangular. In the theorem on the existence of an LU factorization, we required that each diagonal submatrix of  $A$  was invertible. In fact, since  $A$  itself is a diagonal submatrix of  $A$ , we must have that  $A$  is invertible. Since  $L$  and  $M$  are lower triangular matrices with all non-zero diagonal entries,  $L$  and  $M$  are invertible. Therefore, we also have that  $U = L^{-1}A$  and  $V = M^{-1}A$  must also be invertible as they are the product of invertible matrices. With this in place, we have

$$\begin{aligned}
 LU &= MV \\
 \implies M^{-1}L &= VU^{-1}
 \end{aligned}$$

Note that the product of two lower (upper) triangular matrices are lower (upper) triangular. Furthermore, the inverse of a lower (upper) triangular matrix is lower (upper) triangular. Thus,  $M^{-1}L$  is lower triangular and  $VU^{-1}$  is upper triangular. Since these products are equal, we must have  $M^{-1}L$  and  $VU^{-1}$  are both diagonal matrices (since diagonal matrices are the only lower AND upper triangular matrices).

Furthermore,  $L$  has all diagonal components equal to 1 and the diagonal entries of the inverse of a triangular matrix is the reciprocal of the corresponding diagonal entries of the original matrix. Thus, since the diagonal entries of  $M$  were equal to 1, then the diagonal entries of  $M^{-1}$  are also equal to 1. Then, since the diagonal entries of the product of triangular matrices is simply the product of their diagonal entries, we get that each diagonal entry of  $M^{-1}L$  is equal to one. In other words

$$M^{-1}L = I \implies L = M$$

This also gives us

$$VU^{-1} = I \implies V = U$$

Therefore, any lower-upper factorization of the matrix  $A$  must be written as  $L$  and  $U$  from the Theorem, which proves uniqueness.  $\square$