Math 218D Problem Session: Week 2

September 9, 2022

1. Reduced Row Echelon Form.

For each of the following augmented matrices:

$$\textbf{a)} \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 \\ 3 & 0 & 0 & 1 \end{array}\right) \quad \textbf{b)} \left(\begin{array}{ccc|c} 1 & 0 & 3 & 2 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 1 & 2 \end{array}\right) \quad \textbf{c)} \left(\begin{array}{cc|c} 2 & 0 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{array}\right) \quad \textbf{d)} \left(\begin{array}{c|c} 1 & -1 \\ 0 & 1 \\ 0 & 0 \end{array}\right)$$

answer the following questions:

(1) Is the matrix in REF?

(2) Is the matrix in RREF?

(3) If the matrix is not in RREF, use the Gauss–Jordan algorithm to convert it into RREF.

(4) Circle the pivot positions of the augmented matrices and determine the number of solutions to the corresponding system of linear equations.

2. Elementary matrices

a) Write down the 3×3 elementary matrix which E_1 which performs the row operation $R_2 \mathrel{+}= 3R_1$.

b) Write down the 3×3 elementary matrix E_2 which performs the row operation $R_1 \longleftrightarrow R_3$.

c) What (non-elementary) matrix first performs the row operation $R_2 \mathrel{+}= 3R_1$ and *then* does $R_1 \longleftrightarrow R_3$?

3. True or false?

If true, give an explanation; if false, give a counterexample.

a) All elementary matrices are invertible.

b) A 3×2 matrix can have 3 pivots.

c) If a 3×4 augmented matrix has 3 pivots, then the corresponding system of equations has a unique solution.

4. Solving $Ax = b$ using $A = LU$.

When you know the LU factorization of a matrix A, you can use it to solve the matrix equation $Ax = b$. In this problem we will go through this process in an example.

Solve the matrix equation

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 2 & 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix},$$

using the $A = LU$ decomposition

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 2 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}.$$

a) Identify A, b, L, and U.

b) Convert $Ly = b$ into 3 linear equations, and solve for $y = (y_1, y_2, y_3)$ using forward-substitution.

c) Convert $Ux = y$ into 3 linear equations, and solve for x using back-substitution.

d) Check your answer, by multiplying $A \cdot x$ and confirming that it equals b.

Why does this work? If $Ly = b$ and $y = Ux$, then $L(Ux) = b$. Since $A = LU$, this means that $Ax = b$.

5. **Finding $A = LU$ and A^{-1} using elementary matrices**

Consider the matrix

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 2 & -1 & 4 \\ 1 & 4 & 6 \end{pmatrix}.$$

a) Explain how to reduce A to a matrix U in REF (not RREF) using three row replacements.

b) Let E_1, E_2, E_3 be the elementary matrices for these row operations, in order. Fill in the blank with a product involving the E_i:

$$U = \underline{\hspace{3em}} A.$$

c) Fill in the blank with a product involving the E_i^{-1}:

$$A = \underline{\hspace{3em}} U$$

d) Evaluate that product to produce a lower-unitriangular matrix L such that $A = LU$.

e) Explain how to reduce U to the 3×3 identity matrix using three more elementary matrices E_4, E_5, E_6 (scaling, followed by row replacements).

f) Fill in the blank with a product involving the E_i:

$$A^{-1} = \underline{\hspace{5em}}.$$

g) Compute A^{-1} by row reducing $\left(A \mid I_n \right)$. This is exactly the same as evaluating the product above!

6. **Finding $PA = LU$**

a) Find the $PA = LU$ decomposition of $A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 1 & 2 & 3 \end{pmatrix}$ using the 3-column method.

b) Find the $A = LU$ decomposition of $A = \begin{pmatrix} 1 & 1 & 1 \\ -10 & -20 & -30 \\ 5 & 15 & 10 \end{pmatrix}$ using the 2-column method.

c) Find the $PA = LU$ decomposition of $A = \begin{pmatrix} 1 & 1 & 1 \\ -10 & -20 & -30 \\ 5 & 15 & 10 \end{pmatrix}$, this time using *maximal partial pivoting*.

7. Using $PA = LU$

In this problem, we'll solve the matrix equation

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 1 & 2 & 3 \end{pmatrix} x = \begin{pmatrix} 1 \\ 5 \\ 2 \end{pmatrix}$$

using the $PA = LU$ decomposition of problem 6a).

a) Identify A, b, P, L, and U.

b) Compute Pb. What does P do to b?

c) Convert $Ly = Pb$ into 3 linear equations, and solve for $y = (y_1, y_2, y_3)$ using forward-substitution.

d) Convert $Ux = y$ into 3 linear equations, and solve for x using back-substitution.

e) Check your answer, by multiplying $A \cdot x$ and confirming that it equals b.

8. Maximal Partial Pivoting

Consider the linear system

$$x_2 = 1$$
$$x_1 + x_2 = 2.$$

Clearly the solution is $x_1 = 1$ and $x_2 = 1$. Let's modify the system just a little bit:

$$10^{-17}x_1 + x_2 = 1$$
$$x_1 + x_2 = 2.$$

Presumably the solution (x_1, x_2) will be very close to $(1, 1)$.

a) Perform Gauss–Jordan elimination on the augmented matrix

$$\left(\begin{array}{cc|c} 10^{-17} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right)$$

to solve the modified system. You should obtain

$$x_1 = \frac{1}{1-10^{-17}} \qquad x_2 = 1 + \frac{1}{1-10^{17}},$$

which are indeed very close to 1.

Now let's see if a computer can do the same. Load up the Sage cell on the course homepage. Create the augmented matrix as follows:

```
from sympy import *
# 1e-17 is 10^(-17)
A = Matrix([[1e-17, 1.0, 1.0], [1.0, 1.0, 2.0]])
pprint(A)
```

b) Now let's perform Gauss–Jordan elimination:

```
# This does R2 -= 10^(17) R1
# (force sympy to use the smaller pivot)
A.row_op(1, lambda v, j: v - 1e17*A[0,j])
pprint(A)
# Now do Jordan substitution
pprint(A.rref(pivots=False))
```

c) Verify that the last matrix has the form

$$\left(\begin{array}{cc|c} 1 & 0 & (?) \\ 0 & 1 & (?) \end{array}\right).$$

What does the computer think x_1 and x_2 are? What went wrong?

d) Python uses 64-bit floating point numbers. This means that they have about 16 decimal digits of precision. Try evaluating `1+1e17`. What did you get?

The problem was that you produced enormous numbers by dividing by the tiny number 10^{-17}. When you're doing math on a computer, *you never want to divide by tiny numbers.*

e) Now try performing Gauss–Jordan elimination again, after selecting the maximal pivot in the first column:

```
A = Matrix([[1e-17, 1.0, 1.0], [1.0, 1.0, 2.0]])
A.row_swap(0,1)
pprint(A.rref(pivots=False))
```

Did that work? What does the computer think x_1 and x_2 are now?

9. $PA = LU$ **on a computer**

The purpose of this problem is to convince you that computing a $PA = LU$ decomposition really is faster for solving $Ax = b$ for many values of b. Load up the Sage cell on the course homepage.

a) Let's create a 10×10 invertible matrix and a column vector of length 10:

```
from sympy import *
from time import time
A = (eye(10) + ones(10)) * 1.0
b = ones(10, 1) * 1.0
```

Here `eye(n)` is the $n \times n$ identity matrix, and `ones(m,n=m)` is the $m \times n$ matrix of 1's. We multiply by `1.0` in order to force Sympy to do floating-point (as opposed to symbolic) arithmetic. Now

$$A = \begin{pmatrix} 2 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 2 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 2 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}.$$

b) Let's solve $Ax = b$ using a $PA = LU$ decomposition.

```
start = time()
L, U, _ = A.LUdecomposition()
for _ in range(1000):
    U.upper_triangular_solve(
        L.lower_triangular_solve(b))
end = time()
print(end - start)
```

This solves $Ax = (1, 1, \ldots, 1)$ 1000 times, using the $PA = LU$ decomposition.

c) Now let's solve $Ax = b$ without using $PA = LU$.

```
start = time()
for _ in range(1000):
    A.solve(b)
end = time()
print(end - start)
```

What do you notice?