Palmer Edholm
Dr. Joe Koebbe
MATH 4200
November 19, 2021

Tasksheet 7

Task 1 The following code initializes the matrix as it is piecewise defined.

```python
import numpy as np


def upr_trng(n):
    # Initialize n x n matrix of zeros
    A = np.zeros((n, n))
    # Traverse the matrix to change the values of the upper
    # triangular portion to satisfy the piecewise definition
    for i in range(n):
        for j in range(i, n):
            A[i, j] = (i + 1) + (j + 1) - 1
    # Return the matrix
    return A
```

See my software manual entry to see my routine for back substitution performed on this matrix, the 4×4 case of which is

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 3 & 4 & 5 \\ 0 & 0 & 5 & 6 \\ 0 & 0 & 0 & 7 \end{bmatrix}.$$

Task 2 The following code transposes a matrix.

```python
def transpose(A):
    # Transpose the given matrix A
    return [[A[j, i] for j in range(len(A))] for i in range(len(A[0]))]
```

Transposing A, we get

$$A^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 4 & 5 & 6 & 7 \end{bmatrix}.$$

See my software manual entry to see my routine for forward substitution performed on A^T.

Task 3 See my software manual entry to see my routines for generating random square, upper triangular, lower triangular, and diagonal matrices.

Task 4 Using my routine to generate a random, diagonal, 4×4 matrix with random values in the interval $[-10, 10]$, I get

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix}.$$

For simplicity, I'll use the b vector $b_i = 1$. The following code solves the system in a very efficient manner.

```python
def diagsub(A, b):
    # Initialize vector of unknowns and variable for loops
    x = []
    n = len(b)
    # compute values of unknowns
    for i in range(n):
        x.append(b[i] / A[i, i])
    # Return the solution of the system
    return x
```

Now, I can run the code

```python
A = np.asarray(A)
b = np.ones(4)
print(diagsub(A, b))
```

and I get

```
[0.5, -1.0, 0.1111111111111111, 0.125].
```

Task 5 See my software manual entry to see my routine for Gaussian elimination on a random, nonsingular, 5×5 matrix.

Task 6 In [1], the author talks about how a small residual norm of an estimate $\vec{z}$ to the exact solution $\vec{x}$ to the system of equations can be misleading because it does not imply that $\vec{z}$ is close to $\vec{x}$. If we look at the residual bound, we can see why approximations with a small norm can still be bad approximations. Let $A \in \mathbb{C}^{n \times n}$ be nonsingular, $A\vec{x} = \vec{b}$, and $\vec{b} \neq 0$. If $\vec{r} = A\vec{z} - \vec{b}$, then

$$\frac{\|\vec{z} - \vec{x}\|_p}{\|\vec{x}\|_p} \leq \kappa_p(A) \frac{\|\vec{r}\|_p}{\|A\|_p \|\vec{x}\|_p}$$

Where $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$. The p subscript denotes the perturbation. The bound implies that the linear system is well-conditioned if $\kappa_p(A)$ is small.

References

[1] https://ipsen.math.ncsu.edu/ps/OT113_Ipsen.pdf