MATH 315, Fall 2020
Homework 3

James Nagy

Homework Team *(team number)*: *(team name)*

This assignment is associated with Chapter 3, Solution of Linear Systems. It illustrates numerical methods for solving linear systems of equations, i.e. $Ax = b$, and also some complexities introduced when solving this problem on a computer. It should also reinforce some of the theory of vector norms.

1. Systems of linear equations have a unique solution when the matrix A is nonsingular, i.e. when $\det(A) \neq 0$. Consider the formula for the determinant of a 2×2 matrix.

   (a) What are two numerical issues introduced in Chapter 2 that may arise when computing this value?

   (b) Develop two matrices that suffer from these issues. Assume a floating point system that represents the mantissa with 3 digits.

2. (a) Write a MATLAB code that will implement row-oriented forward substitution (see Fig. 3.5). Your code should be written as a function m-file, with input A (an $n \times n$ matrix) and b (an $n \times 1$ vector), and output the solution x (the $n \times 1$ vector that solves $Ax = b$).
   More specifically, the beginning of your code should contain the following:

```
function x = RowSolve(A,b)
% This function computes the solution of the linear system Ax=b using
% forward substitution, using a row-oriented approach
%
%   Input:  A - nxn matrix
%           b - nx1 vector
%
%   Output: x - solution of the linear system
%

n = length(b);
x = zeros(n,1);

for ...
```

(b) Write a MATLAB code that will implement column-oriented forward substitution (see Fig. 3.5). Your code should be written as a function m-file, with input A (an $n \times n$ matrix) and b (an $n \times 1$ vector), and output the solution x (the $n \times 1$ vector that solves $Ax = b$). More specifically, the beginning of your code should contain the following:

```
function x = ColSolve(A,b)
% This function computes the solution of the linear system Ax=b using
% forward substitution, using a column-oriented approach
%
%   Input:  A - nxn matrix
%           b - nx1 vector
%
%   Output: x - solution of the linear system
%

n = length(b);
x = zeros(n,1);

for ...
```

(c) The following Matlab statements can be used to explore the computational cost of each function and compare and assess both solutions:

```
n = 10;
A = tril(rand(n));
b = rand(n,1);

disp('row-oriented:')
tic
xR = RowSolve(A,b);
toc

disp('column-oriented:')
tic
xC = ColSolve(A,b);
toc

disp('infinity norm of difference:')
norm((xC - xR), Inf)

disp('infinity norm of residual:')
norm((b - A*xR), Inf)
```

Create a Matlab script file, called containing the above Matlab statements. Run your script multiple times for various n, e.g. $n = 10, 100, 1000, \ldots$. Does one approach produce a solution faster than the other? Does n influence this? Do the two approaches produce the same results? Is the residual zero? Discuss your findings and rationale for them.

(d) What is a line of code that serves the same function as `norm((xC - xR), Inf)`?

3. Show that the ∞-norm satisfies the properties of a vector norm.