Hans Guthrie
3/5/2014
ECE 540
Dr. Day

Home Work 3: Matrix Operations

This assignment consisted of two parts and involved matrices and different techniques to solve them.

Question 1:
The first question was to print out the LU factorization of the following matrix using the LU Factorization code in the Matrix.hpp file.

```
2/3 x  +      y  + 1/2 z  = 3
1/2 x  + 2/3 y  +      z  = 2
     x  + 1/2 y  + 2/3 z  = 1
```

I created a matrix named `m1` to hold all the coefficients for x, y, and z, and a matrix `x` to hold the solutions.

$$m1 = \begin{bmatrix} 2/3 & 1 & 1/2 \\ 1/2 & 2/3 & 1 \\ 1 & 1/2 & 2/3 \end{bmatrix}$$

$$x = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Then I copied `m1` to the matrix `lu_m1` and performed the LU factorization on the `lu_m1` matrix. After this was computed, I printed the output to the screen as well as the pivots for the factorization, and the solution to the system of equations for a given matrix. I have included the output to my program at the end of the lab.

Question 2:
For the second question we had to use code in the Matrix.hpp file to calculate the solutions to the matrix from question 1 using an iterative approach. We were given the following equation to use to estimate the values.

$$\underline{x_{i+1}} = (I - \mu A^t * A) * \underline{x_i} + \mu A^t * \underline{b}$$

I calculated the iteration matrix ($I - \mu\, A^t * A$) once at the beginning and then used it and the vector value ($\mu\, A^t * b$) throughout the loop. I performed 100 iterations and used $\mu = 0.3$.

Since there were 100 iterations and it printed the contents of the X matrix and the error I have only included the final results in my write-up, and these can be found at end of this lab.

Output for Question 1:

```
Input m1

 Columns 1 to 3
|     0.666667,     1.000000,     0.500000,  |
|     0.500000,     0.666667,     1.000000,  |
|     1.000000,     0.500000,     0.666667,  |


 LU Factorization of m1

 Columns 1 to 3
|     1.000000,     0.500000,     0.666667,  |
|     0.666667,     0.666667,     0.055556,  |
|     0.500000,     0.625000,     0.631944,  |


Permutation Matrix

2, 0, 1

Vector b =

 Columns 1 to 1
|     3.000000,  |
|     2.000000,  |
|     1.000000,  |

Solution Vector x =

 Columns 1 to 1
|    -0.791209,  |
|     3.494505,  |
|     0.065934,  |
```

Output for Question 2:

```
Columns 1 to 1
|    -0.786743,  |
|     3.487807,  |
|     0.068167,  |


Error =

 Columns 1 to 1
|     0.000014,  |
```

Code Appendix:

```
#include "matrix.hpp"
#include "MatrixOutputs.hpp"
#include <stdio.h>

int main()
{
	//Problem 1. LU Factorization
	{
		//Create two matrix objects. One for the input matrix,
		//the other for the output of the LU Optimization
		matrix m1, lu_m1, x(3), b;

		int permutationvector[ 3 ];

		m1 = matrix( 3, 3 );
		//row 1
		m1( 0, 0 ) = 2.0 / 3.0;
		m1( 0, 1 ) = 1.0;
		m1( 0, 2 ) = 1.0 / 2.0;
		//row 2
		m1( 1, 0 ) = 1.0 / 2.0;
		m1( 1, 1 ) = 2.0 / 3.0;
		m1( 1, 2 ) = 1.0;
		//row 3
		m1( 2, 0 ) = 1.0;
		m1( 2, 1 ) = 1.0 / 2.0;
		m1( 2, 2 ) = 2.0 / 3.0;

		x( 0 ) = 3;
		x( 1 ) = 2;
		x( 2 ) = 1;

		lu_m1 = m1; //Copy the input matrix to the LU Optimization matrix
		LU( lu_m1, permutationvector ); //Perform LU Optimization on the lu_m1 object

		printf( "\nInput m1\n" );
		PrintMatrix( m1 );
		printf( "\n LU Factorization of m1\n" );
		PrintMatrix( lu_m1 );

		printf( "\nPermutation Matrix \n\n" );
		printf( "%d, %d, %d\n\n", permutationvector[ 0 ], permutationvector[ 1 ],
	permutationvector[ 2 ] );
		b = x;

		//Display b vector
		printf( "Vector b = \n" );
		PrintMatrix( b );

		// Solve for "x" and display solution.
		x = usolve( lu_m1, lsolve( lu_m1, permutate( b, permutationvector ) ) );
		printf( "Solution Vector x = \n" );
		PrintMatrix( x, 5 );

		printf( "Press enter to continue with Question 2" );
		getchar();

	}
```

```
	//Problem 2.
	//Use code in the Matrix.hpp file to calculate the solutions to the matrix from
	//question 1 using an iterative approach.
	{
		//Declare matrix objects needed
		matrix A, B, Identity_Mat, X, Iter, bIter, Y, e;
		double mu; //μ variable used during calculations
		int    k; //loop counter
		mu = 0.3; //We will use an μ value of 0.3

		A = matrix( 3, 3 ); //Initialize the A matrix and then load it with the following values
		//row 1
		A( 0, 0 ) = 2.0 / 3.0;
		A( 0, 1 ) = 1.0;
		A( 0, 2 ) = 1.0 / 2.0;
		//row2
		A( 1, 0 ) = 1.0 / 2.0;
		A( 1, 1 ) = 2.0 / 3.0;
		A( 1, 2 ) = 1.0;
		//row3
		A( 2, 0 ) = 1.0;
		A( 2, 1 ) = 1.0 / 2.0;
		A( 2, 2 ) = 2.0 / 3.0;

		B = matrix( 3 ); //Initialize the B matrix and load it with the following
	values
		B( 0 ) = 3;
		B( 1 ) = 2;
		B( 2 ) = 1;

		Identity_Mat = eye( 3, 3 ); //Initialize the Identity_Mat object with the
	Matrix.hpp's eye function

		X = matrix( 3 );
		X( 0 ) = 0;
		X( 1 ) = 0;
		X( 2 ) = 0;

		//Do the actual computation of ( I - mu * A' * A )
		Iter = ( Identity_Mat - A.transpose() * A * mu);
		bIter = ( A.transpose() * B * mu );

		//Loop that performs the iteration and calculates the iterations
		for ( k = 0; k < 100; k++ )
		{
			//Performs the iteration
			Y = Iter * X + bIter;
			e = A * X - B;
			e = e.transpose() * e; //The error will be in location e(0)
			//Print X and the error
			printf( "X=\n" );
			PrintMatrix( X );
			printf( "\nError =\n" );
			PrintMatrix( e );
			X = Y; //Replace x with y so the iteration can continue
		}
	}

	printf( "Press enter to exit" );
	getchar();
}
```

Home Work 3

0) Coding	
Commenting	2/2
Error Checking	0/1
Variable Naming	1/1
Structure	1/1
1) LU Solution	
LU factorization written out	2/2
Pivots written out	1/1
Correct Solution	2/2
2) Iterative Solution	
Iter matrix correct	0/3
Iter Vector correct	0/3
Stopping Criteria	1/1
Correct answer	3/3
Total	13/14

You need to check and make sure each matrix was allocated correctly
-1
You also need to print out Iter matrix and vector
-6