Name: ____________________

Date: ____________________

Question:	1	2	3	4	Total
Points:	15	20	20	15	70
Score:					

Catastrophic cancellation example

1. As you know the following expression,

$$\frac{1}{\sqrt{10^7 - x}} - \frac{1}{\sqrt{10^7 + x}},$$

if used with finite-precision floating point arithmetic and for small values of x, $|x| \ll 1$, leads to loss of significance (known as *catastrophic cancellation*).

(a) (5 points) Write a matlab function, `original(x)`, that returns the values of the expression above using its "original" form.

(b) (5 points) Rewrite the expression above to fix the catastrophic cancellation problem. Write a matlab function, `modified(x)`, that returns the values of the expression above using your "modified" form.

(c) (5 points) Write a matlab script (call it e.g. **hw05p1.m**) that on the same graph plots the functions `original(x)` and `modified(x)` for $10^{-10} \le x \le 10^{-8}$. Use at least 100 points for your graphs. You can initialize the vector of x values for your functions as following:

```
x = 1.e-10*(1:100);
```

Solving linear systems

2. Students of linear algebra learn that the solution to

$$Ax = b$$

can be written

$$x = A^{-1}b,$$

where A^{-1} is the inverse of matrix A. However, in the vast majority of practical computational problems, it is unnecessary and inadvisable to actually compute A^{-1}.

It is of course *possible* to compute A^{-1}. Here is one of many methods: the inverse of a matrix A can be defined as the matrix X whose columns x_j solve the equations

$$Ax_j = e_j,$$

where e_j is the jth column of the identity matrix.

(a) (15 points) Starting with the code we develop in class (functions `ludecomp`, `forward`, and `backsubs`), write a Matlab function `myinverse(A)` that computes the inverse of A. Your function should call `ludecomp` only once and should not use the built-in Matlab backslash operator or `inv` function.

(b) (5 points) Write a matlab script (call it e.g. **hw05p2.m**) that tests your function by comparing the inverses it computes with the inverses obtained from the built-in `inv(A)` on two matrices which are generated by **truss.m** and `resistors.m` scripts that we developed earlier in this course.

Hint: one way to compare two matrices, say A and B, in matlab is to calculate `max(abs(A(:)  - B(:)))`

3. The matrix factorization

$$LU = PA$$

can be used to compute the determinant of A. We have

$$\det(L)\det(U) = \det(P)\det(A).$$

Because L is triangular with ones on the diagonal, $\det(L) = 1$. Because U is triangular,

$$\det(U) = u_{11}\, u_{22} \cdots u_{nn}.$$

Because P is a permutation, $\det(P) = +1$ if the number of interchanges is even and 1 if it is odd. So

$$\det(A) = \pm u_{11}\, u_{22} \cdots u_{nn}.$$

Hint: in Matlab, the product $u_{11}\, u_{22} \cdots u_{nn}$ can be computed using the expression `prod(diag(U))`.

(a) (15 points) Starting with the code we developed in class (functions `ludecomp`, `forward`, and `backsubs`), write a Matlab function `mydet(A)` that computes the determinant of A. Your function should should not use the built-in `det` function. How you determine if the number interchanges of rows is even or odd from the vector p returned by `ludecomp`?

(b) (5 points) Write a matlab script (call it e.g. **hw05p3.m**) that tests your function by comparing the determinants it computes with the determinants obtained from the built-in `det(A)` on two matrices which are generated by **truss.m** and `resistors.m` scripts that we developed earlier in this course.

Gitlab

4. Create a gitlab project called **hw05** (named it exactly as shown)

   (a) (10 points) create README file that describes your **actual results** for each of the problems in this homework

   (b) (5 points) upload your matlab code; share the project with the instructor.