Practical 2

Series for Functions and the LU Decomposition

In this practical, we will look at two different topics. Firstly, we will use the series approximation to evaluate two functions. Secondly, we will implement the LU decomposition of a matrix and use it to solve a set of linear equations.

Part A

The sin function

In Lecture 8, we looked at the evaluation of functions using a series approximation (the Maclaurin and Taylor series specifically). In part A, you must use the Maclaurin series to evaluate $\sin(x)$ and compare it to the built-in function.

Implement the function `mysin(x,n)` which returns the approximate value of $\sin(x)$ using n terms of the Maclaurin series:

$$\sin(x) \simeq x - \frac{x^3}{3!} + \frac{x^5}{5!} \dots$$

Call `partA`; the result should be plot which compares the different order series to $\sin(x)$ from the built-in function.

The blue dashed line represents the true value of the function and the red line is the approximation. You should see that the sequence of plots (upper-left to lower-right) gets closer to the true function.

Part B

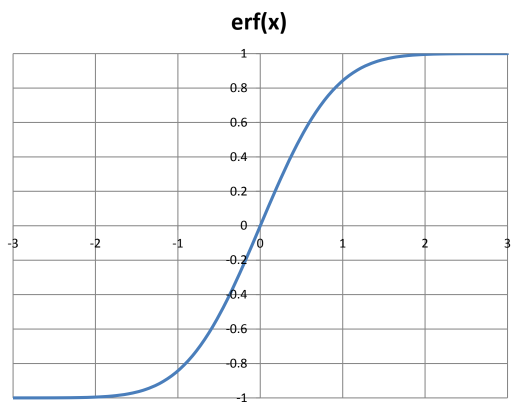
The erf function

The erf function will probably be less familiar to most of you. The definition is

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$$

Find a Maclaurin series for erf(x) and use it to implement the function `myerf(x,n)`.

Call `partB`; the result should be plot which shows the different order series for erf(x) from the built-in function. The correct plot of the error function is below:



Part C

LU Decomposition

In Lecture 7, we developed a matrix method of solving systems of linear equations, which led us to the idea of the LU decomposition. The algorithm is as follows:

We begin with the original (matrix) equation

$$\mathbf{Ax} = \mathbf{c}$$

Let $\mathbf{U}^{(0)} = \mathbf{A}$ and $\mathbf{M}^{(0)} = \mathbf{I}$ (i.e. the identity matrix). The superscript $^{(k)}$ refers here to the number of iterations we have done. We then iteratively zero elements in the lower triangle of $\mathbf{U}$ using the element-zeroing matrix $\mathbf{M}_{ij}$

$$\mathbf{M}_{ij} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ & & \ddots & & \\ \vdots & & -\frac{U_{ij}}{U_{j,j}} & & \\ 0 & & & & 1 \end{pmatrix}$$

The single non-zero off-diagonal element appears in element (i, j) and zeros element (i, j). We multiply each size of the equation to get

$$\mathbf{U}^{(k+1)} = \mathbf{M}_{ij}\mathbf{U}^{(k)}$$

$$\mathbf{M}^{(k+1)} = \mathbf{M}_{ij}\mathbf{M}^{(k)}$$

This continues until all of the lower triangle elements are eliminated from $\mathbf{U}$. They should be zeroed in the order of the first column, then second column etc.

After this process, the equation is reduced to the form

$$\mathbf{U}\mathbf{x} = \mathbf{M}\mathbf{c}$$

Now we can write $\mathbf{L} = \mathbf{M}^{-1}$ and the LU decomposition is

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

Implement this algorithm in the function `LUDecomposition(A)`.

Backsubstitution

We easily solve the equations now using the backsubsitition algorithm. The equation is

$$\mathbf{U}\mathbf{x} = \mathbf{M}\mathbf{c} = \mathbf{c}'$$

Starting at the bottom row (row n), we should get

$$U_{nn}x_n = c_n'$$

$$x_n = \frac{c_n'}{U_{nn}}$$

which gives us x_n. The row next up is

$$U_{n-1,n-1}x_{n-1} + U_{n-1,n}x_n = c'_{n-1}$$

$$x_{n-1} = \frac{1}{U_{n-1,n-1}}(c'_{n-1} - U_{n-1,n}x_n)$$

which we can evaluate as we know x_n, and so on up the rows.

Implement the backsubstitution algorithm in the method `backSubstitution(U,c)`.

Call `partC`; it will test your LU decomposition, and you should find the solution to the system of equations as $(2, -4, 3)$.