CS601: Software Development for Scientific Computing

Bonus Assignment - Erf Function
Due: 6/10/2021

The objective of this assignment is to improve the accuracy of the `erf` function.

1 Problem

Implement a C++ program that generates the value of error function[1], `erf` accurately. The error function is defined by the equation: $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$

1.1 What you need to submit

Create a branch called ERF in your PA1 repository. Your branch should contain only those files that implement the `erf` function. Place all of the necessary code that *you* wrote for implementing `erf` function in this branch. Create a shell script (this must be written in bash) called `runme` at the top level (outside your `inc`, `src` folders if any). The `runme` script compiles and runs your C++ program. This script should take in two arguments: first, the parameter `x` that needs to be passed to your `myerf` function. Second, `accuracy`, another parameter passed to your `myerf` function. E.g. given `x`=-3.000000 and `accuracy`=1^e-6, your program should output *blankspace separated real numbers (3 of them, showing results up to 6 decimal digits) (Do not print anything else. Not even the column header shown below)*:

Your implementation must use the least number of terms for achieving the desired/given accuracy. Compare with the built-in function `erf` for *validating* your results (i.e. checking if you are 'building the right thing" / checking correctness).

You must tag your source code and submit as done for previous assignments. The tag name to be used is: `cs601erfsubmission`*. All tag names are case-sensitive..*

x	myerf	erf
-3.000000		

2 Hints

[1] see erf for information the error function

Hints to PA1

One of the applications of the Erf function is to use it while computing the probability that a standard normal variate X assumes a value in the interval [0,x] given the *standard normal distribution curve*.

$$Erf(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!\,(2n+1)}$$

$$= \frac{2}{\sqrt{\pi}} (\mathrm{x} - \frac{1}{3}\mathrm{x}^3 + \frac{1}{10}\mathrm{x}^5 - \frac{1}{42}\mathrm{x}^7 + \frac{1}{216}\mathrm{x}^9 + \ldots$$

The above is called McLaurin series.

- Note the sum is to infinity. So we need to decide on the *number of terms* to include in the numerical computation to meet the desired accuracy.

- For $|x| < 1$ the series is well behaved in that as n goes to infinity, x^{2n+1} will go to zero (i.e., the terms get smaller and smaller). However, for $|x| > 1$, x^{2n+1} will go to infinity as n goes to infinity.

- For increasing values of n, when $|x| > 1$, the value of the term is reduced by the n! in the denominator. $n!$ grows rapidly and hence, the magnitude of $n!$ could cause problems: may lead to overflow. E.g., for x=-3.0, n should be 32 to obtain 10^6 accuracy in the calculation. 32! is $2.6\mathrm{x}10^{35}$, which is 25 orders of magnitude larger than the maximum integer value possible to store in 4 bytes. Hence, although factorial is an integer, it will need to be saved in a `double` in this calculation.

- If large arguments (x) are to be used, an alternative to the Maclaurin series should be considered. The Wolfram site gives some alternative expressions for large arguments.