Project 1
SF2568 Program construction in C++ for Scientific Computing

September 28, 2015

In this project you will implement some simple numerical problems in C++ in order to become comfortable with the basic C++ syntax and the development environment.

Task 1 The Taylor series of the sine function $\sin(x)$ and the cosine function $\cos(x)$ are given by

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!},$$
$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}.$$

Write functions `sinTaylor(N,x)` and `cosTaylor(N,x)` that calculate the sum of the first `N` terms in the series. Compare these results with the sine and cosine functions included in the C standard library (`#include <cmath>`): Verify that the errors

$$|\sin(x) - \texttt{sinTaylor}(N,x)| \quad \text{and} \quad |\cos(x) - \texttt{cosTaylor}(N,x)|$$

are bounded by the $(N+1)$-st term in the corresponding Taylor series. Show this for $x = -1, 1, 2, 3, 5, 10$ and a number of selected values for N!

Hint: For larger values of $|x|$ you may observe overflow conditions during the evaluation of the terms. Therefore, you should implement the polynomial evaluation using Horner's scheme! The use of `pow` is herewith explicitly forbidden!

Task 2 Adaptive Integration.

Consider the computation of the definite integral

$$I = \int_a^b f(x)dx$$

for a smooth function $f : [a,b] \to \mathbb{R}$. The task consists of computing an approximation to the integral with a prescribed tolerance ε. We will use adaptive Simpson quadrature. You have learned about it in the basic course in Numerical Analysis. Let

$$I(\alpha,\beta) = \frac{\beta-\alpha}{6}(f(\alpha)+4f((\alpha+\beta)/2)+f(\beta))$$

be the Simpson rule applied to evaluating the integral $\int_\alpha^\beta f(x)dx$. Then, it holds for the error

$$I(\alpha,\beta) - \int_\alpha^\beta f(x)dx = \frac{(\beta-\alpha)^5}{2880} f^{(4)}(\xi_1).$$

If we introduce the midpoint $\gamma = \frac{1}{2}(\alpha+\beta)$ and $I_2(\alpha,\beta) := I(\alpha,\gamma)+I(\gamma,\beta)$, this equation leads to

$$I_2(\alpha,\beta) - \int_\alpha^\beta f(x)dx = \frac{(\beta-\alpha)^5}{46080} f^{(4)}(\xi_2).$$

These two equations can be used in order to estimate the error of the Simpson rule applied to f on $[\alpha,\beta]$: If we assume that $f^{(4)}(\xi_1) \approx f^{(4)}(\xi_2)$, then

$$I_2(\alpha,\beta) - \int_\alpha^\beta f(x)dx \approx \frac{1}{15}(I_2(\alpha,\beta) - I(\alpha,\beta)).$$

Thus, the error of the numerical approximation $I_2(\alpha,\beta)$ can be estimated by the right-hand term,

$$\texttt{errest} = \frac{1}{15}(I_2(\alpha,\beta) - I(\alpha,\beta)).$$

So we can design the following algorithm: (ASI = Adaptive Simpson Intergration)

```
I = ASI(f,a,b,tol)
    I1 = I(a,b);
    I2 = I_2(a,b);
    errest = abs(I1-I2);
    if errest < 15*tol return I2;
    return ASI(f,a,(a+b)/2,tol/2) + ASI(f,(a+b)/2,b,tol/2);
```

Implement the algorithm! Try to avoid the recursive call of ASI. Apply your method to approximate the integral

$$\int_{-1}^{1}(1+\sin e^{3x})dx$$

with a tolerance of $10^{-2}, 10^{-3}, 10^{-4}$! Compare your results with the value provided by matlab (with a tolerance of 10^{-8}.

Note: This problem is a prerequisite for Project 3.

The programming exercises should be done individually, or in groups of two. Hand in a report containing:

- Comments and explanations that you think are necessary for understanding your program.
- The output of your program acording to the tasks.
- Program listing.
- E-mail the source code to `hanke@nada.kth.se`.