Introduction to Computational Physics

Matthias Jamin

(Additional discussion on Friday, 27.6. and Monday, 30.6.)

EXERCISE 7.1: **Gauss-Legendre Integration**
My solution to the exercise in Fortran is given in `Ex7.1.f`. As one observes, the convergence is not overwhelming, but nevertheless, for 1024 Gauss points, a deviation of around $1.5 \cdot 10^{-9}$ from the exact result π is achieved, much better than the Romberg algorithm for the direct integration. The deviation of the corresponding numerical integral in Mathematica from π is two orders of magnitude smaller and found to be $1.5 \cdot 10^{-11}$.

EXERCISE 7.2: **Gauss-Laguerre Integration**
My solution to the exercise in Mathematica is given in `Ex7.2.m` and for Fortran in `Ex7.2.f`. Maybe you could discuss the implementation of the weight function in the Numerical Recipes algorithm which employs $\Gamma(z) = \exp(\ln(\Gamma(z)))$ in order to avoid large numbers at intermediate steps. As far as the result is concerned, one observes that in this case it is actually more precise at lower orders, since the weight function contains the desired result explicitly. If one uses higher orders, than the result is broken up into more terms and rounding errors creep in which make the result less accurate. The deterioration of the accuracy in Mathematica is even worse, probably due larger errors in the weight function when inserting the roots into the analytical result for the polynomial, but maybe you could investigate this further in the tutorials.

EXERCISE 7.3: **Black Body Radiation**
My solution to the exercise in Fortran is given in `Ex7.3.f`. In this example, one can play with different choices of putting factors x^α into the Gauss-Laguerre integral weight or keeping them in the integrand. One actually finds that the highest accuracy is achieved, when keeping the factor x^3 in the integrand and employing the Gauss-Laguerre algorithm with $\alpha = 0$.

EXERCISE 7.4: **Integration of Data**
My solution to the exercise in Fortran is given in `Ex7.4.f`. In the output, I present different ways of arriving at the desired integral: first case trapezoidal rule with half the data points; second case trapezoidal rule with all data points; third case result of the first Romberg step using case one and two; finally the results of Gauss integration with 2^k points for $k = 4 \dots 12$ and cubic spline interpolation the get the function values at the Gauss points. Performing different comparisons of the results, my conclusion would be that a realistic estimate of the uncertainty should be $1 \cdot 10^{-4}$ and the final result: 2.4342(1).