Assignment V : Data Fitting

Due 10/03/2016

1. Polynomial Interpolation

Consider the data set for a neutron cross section given in *crossX2.dat*.

(a) Use the Lagrange interpolation formula of Eq. (5.3) (R.H. Landau, 1st edition or subsection 9.1.2 2nd edition) to fit the entire spectrum with one polynomial. Then use this fit to plot the cross section in steps of 5 MeV. Include the data in your plot (with error bars).

(b) Use your graph (as well as the calculated data) to deduce the energy of the peak, E_γ as well as the resonance width γ (defined as the width at half the maximum). Compare your results with those given in the book as prediction.

(c) Your fit from **(a)** contains spurious peaks at both ends of the spectrum. The reason is that the degree of the polynomial chosen to interpolate the data is too high. A more realistic use of Lagrange interpolation is using a local interpolation with a small number of points, e.g. three. Interpolate the same data in 5 MeV steps using three-point Lagrange interpolation. Modify your code so that you interpolate the data in smaller intervals. Add this interpolation to the plot made for **(a)**.

(d) Use the interpolation routine *polint.f90* from Numerical Recipes and repeat (a) and (c). This routine gives in addition to the interpolated function values $y = f(x)$ and estimated error dy. Include this error into your plot and attempt an interpretation.

(e) Use the spline interpolation of xmgrace to interpolate your data and discuss your result.

2. Spline Interpolation

Continue with the data set from Problem 1. For cubic splines cubic polynomials are fit the function in each interval with the additional constraint that the first and second derivatives of the polynomials must be continuous from one interval to the next. Instead of writing your own spline programs, you should test three different splines:

1. Cubic Spline Interpolation from *Numerical Recipes*. The description is given in the notes under *Splines-NR*. The two codes are contained in *spline.f90*. The driver program is *splineNR.f90*.

2. Cubic Spline Interpolation based on the codes *icsccu.f90* and *icsevu.f90*, where the first routine prepares the spline coefficients, and the second is used to perform the actual interpolation.

3. The cubic Hermite splines as given in Appendix B of D. Hüber *et al.*, Few-Body Systems **22**, 107 (1997), given in the notes under *CubHerm-Spline*. These cubic Hermite splines are implemented for one dimensional interpolation in the code *cubhermdh.f90* with the driver program *splinecbh.f90*.

(a) Use both spline routines with the data set from Problem 1. Plot your interpolation.

(b) A standard procedure to assess the quality and properties of an interpolation routine is a test with a known function. The data set *crossX2.dat* does not have enough points to allow you to investigate how the quality of the spline interpolations depend on the number of points given. So you need to create such a function. Use the Breit-Wigner function of Eq. (5.1) (1st edition) or Eq. (9.1) (2nd edition) with the values of E_r and γ obtained in Problem 1, and determine σ_0 such that your 'theory curve' matches the data in the peak. Write a code that is able to create your own 'experimental' data.

(c) Obviously, if you have a denser grid of points in your given function, the spline interpolation should get better. Create e.g. a set of 'data' every 10 MeV from 0 to 200 MeV. Then delete every other point from your data set, and calculate the deleted point via spline interpolation. Since you know the exact value (the point you deleted), you can determine the absolute error of your spline interpolation with respect to the 'correct' answer. Study the quality of the two different spline packages as function of the interval between points. Make a sequence of three different interval lengths and plot the error as function of this length. Discuss your findings.

(d) Integrate the spectrum using only the given values in the table and your spline function. You want to use here the Cubic-Hermite splines. Use Gauss-Legendre integration and reason to which value you want to set E_{max}. Then carry out the same integration using the analytic function of Eq. (5.1) (or (9.1)) and obtain a converged result (5 significant figures). Then discuss the different types of errors when you integrate the splined function, and how you dealt with minimizing those errors.

(e) Could you think of an algorithm to directly use the Cubic-Hermite splines to integrate, and thus arrive at a semi-analytical result for your integration?