PH2150A Scientific Computing Skills Team Project
Least squares fit of a polynomial

1 Introduction

In this project you will be fit a polynomial through a set of measured points with errors using the method of least squares. That is, you will determine the coefficients of a polynomial such that the sum of squares of measured values minus the corresponding polynomial values is a minimum. This will involve investigating techniques for solving systems of linear equations.

The data points you will use are shown in Fig. 1. The values can be obtained from the file `ls_data.dat` from the course web site.

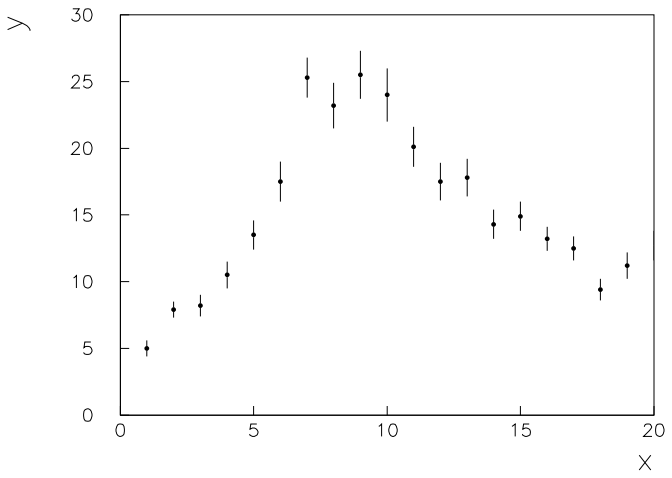

Figure 1: The measured values y (shown with error bars as $y \pm \sigma$) versus x for use in the least squares fit.

2 The assignment

Your assignment includes the following:

- Fit polynomials of different orders using the method of least squares to the data shown in Fig. 1. Find out as much as you can about different methods for finding the solution.
- Investigate how the minimized value of the χ^2 varies as the order of the polynomial increases.
- Investigate what happens if the errors for all of the data points are set to unity.
- Examine what happens to the polynomial outside of the range of measured data points.

In order to minimize the χ^2, you can implement, for example, the technique of *Gauss–Jordan elimination with pivoting.*

3 The method of least squares

Once the order of the polynomial has been specified, we need to determine the values of the coefficients such that we achieve the best agreement with the data. This can be done with the method of least squares. Consider a polynomial of order m,

$$f(x) = \sum_{j=0}^{m} c_j x^j \ . \tag{1}$$

This has $m+1$ coefficients which we can write in vector notation as $\mathbf{c} = (c_0, \ldots, c_m)$. The idea is to construct a quantity which reflects the level of agreement between the data and the polynomial. For this we can use the 'chi-square' variable, defined by

$$\chi^2(\mathbf{c}) = \sum_{k=1}^{N} \frac{(y_k - f(x_k; \mathbf{c}))^2}{\sigma_i^2} \ , \tag{2}$$

where y_i is the ith measured value, $f(x_i; \mathbf{c})$ is the value of the polynomial at x_i, and σ_i is the 'uncertainty' of the ith measurement. (More precisely, the measured values are treated as random variables, and σ_i is the standard deviation of the ith measurement.) In the method least squares, you choose the polynomial coefficients such that the χ^2 is a minimum.

The value of χ^2 at its minimum is a measure of the 'goodness-of-fit'. If the hypothesized functional form is correct, then you expect a contribution to the minimized χ^2 of about 1 per data point, since the deviation between the data and the prediction should be on the order of the uncertainty σ. In fact one can show the expected value of the minimized χ^2 is equal to $N - n_{\rm p}$ (called the *number of degrees of freedom*), where N is the number of data points and $n_{\rm p}$ is the number of fitted parameters. For the polynomial of order m we have $n_{\rm p} = m + 1$. Note that if the number of coefficients is equal to the number of data points, then we can arrange for the polynomial to pass exactly through every measured point. (You should try this, perhaps using a subset of the data points in Fig. 1.)

4 Minimizing the χ^2

For a general fitting function, finding the parameter values which minimize $\chi^2(\mathbf{c})$ can be a very challenging task. For the current problem, however, we are dealing with a function $f(x; \mathbf{c})$ which is linear in the parameters $\mathbf{c}$. In this case, the solution can be found by solving a system of linear equations. While straightforward in principle, linear systems can also present programming challenges when the number of variables (here, the number of polynomial coefficients) is large. The equations are determined by setting the derivatives of χ^2 with respect to the coefficients equal to zero. This gives

$$\frac{\partial \chi^2}{\partial c_i} = -2 \sum_{k=1}^{N} \frac{(y_k - \sum_{j=0}^{m} c_j x_k^j)}{\sigma_k^2} x_k^i = 0 \ , \tag{3}$$

which can be rewritten as

$$\sum_{j=0}^{m} \sum_{k=1}^{N} \frac{x_k^{i+j}}{\sigma_k^2} c_j = \sum_{k=1}^{N} \frac{x_k^i y_k}{\sigma_k^2} \ , \quad , i = 0, \ldots, m \ . \tag{4}$$

This is a set of $m+1$ linear equations for the unknown coefficients c_i, called the *normal equations*. They can be expressed in matrix form by defining

$$A_{ij} = \sum_{k=1}^{N} \frac{x_k^{i+j}}{\sigma_k^2} \ , \quad b_i = \sum_{k=1}^{N} \frac{x_k^i y_k}{\sigma_k^2} \ , \quad i, j = 0, \ldots, m \ . \tag{5}$$

The normal equations are then simply

$$A\mathbf{c} = \mathbf{b} \ , \tag{6}$$

where in (6), $\mathbf{b}$ and $\mathbf{c}$ should be understood as column vectors.

Finding the coefficients $\mathbf{c}$ thus boils down to solving equation (6). Many algorithms exist for doing this, and they are widely available in standard programming packages. Rather than (or in addition to) using a canned routine, however, you are encouraged to implement the solution yourselves on the computer. A simple but robust technique is called *Gauss–Jordan elimination with pivoting*. Information on this method is provided on a separate handout.

5 References

More information on Gauss–Jordan elimination and on the method of least squares can be found in:

W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes*, 2nd edition, Cambridge University Press, Cambridge (1992);

S. Brandt, *Statistical and Computational Methods in Data Analysis*, Springer, New York (1997).

G. Cowan, November 1998