Part 3. Modeling of data

1. Introduction to Least Squares Fitting. Much of the material in this section comes from *Numerical Recipes,* chap. 15, SGN, 6th ed., chapter XXII, and Bevington and Robinson. Consult these for more complete coverage of this topic.

We have a data set $\{y_i, x_i\}$. The y_i are values of a "dependent" variable which we measured at N different values of the "independent" variable x_i for i=1 to N We also have (usually) a physical model of the system, implying a functional relationship between y and x:

$$y(x) = f(x;\alpha_0,\alpha_1,\ldots,\alpha_{m-1}) \tag{44}.$$

The independent (measured or controlled) variables are represented by x; the parameters of the model are the α_j for j=0 to m-1. An example is the integrated Clausius-Clapeyron equation describing the variation of the vapor pressure of a pure liquid with T:

$$\ln\left(\frac{p}{p_0}\right) = C - \frac{\Delta H_{vap}}{RT} \tag{45}.$$

Here x might correspond to 1/T, y corresponds to $\ln(p/p_0)$, and the model parameters are C (an uninteresting constant) and ΔH_{vap}.

2. Concept of Least-Squares Fitting. In a modeling problem we want to find: 1) the "best" values of the model parameters $\alpha_0 \ldots \alpha_{m-1}$, 2) uncertainties (confidence limits) of those parameters, and 3) an indication of whether the model actually "fits" the data at all. We can't calculate from a data set the probability that any particular parameter set is correct. The approach is to calculate the probability of obtaining our actual data given any set of parameters, and choose the parameter set which gives the greatest probability of finding that data set. That corresponds again to the principle of maximum likelihood: we proceed by assuming that our data set is the *most probable* one. If the error distributions of the y_i are normal, there is no or negligible error in the values of the x_i, and we know that the model is correct, the probability for obtaining a single one of the y_i is

$$P(y_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{\left\{-\frac{[y_i - f(x_i;\alpha_j)]^2}{2\sigma_i^2}\right\}} \tag{46}.$$

Since the different y_i measurements are independent, calculating the probability density of the entire data set given the parameters is straightforward: it's just the product of the probability for obtaining each y_i independently.

$$P(\{y_i\}) = \prod_{i=1}^{N} P(y_i) = \left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i}\right) e^{\left\{-\frac{1}{2}\sum_{i=1}^{N}\frac{[y_i - f(x_i;\alpha_j)]^2}{\sigma_i}\right\}} \tag{47}.$$

We want to maximize this probability by varying the α_j fitting parameters. Since the part outside the exponential is independent of the values of the α_j, we can just maximize the exponential, which is the same as minimizing the value of χ^2:

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i;\alpha_j)}{\sigma_i} \right)^2 \qquad (48).$$

This is called least-squares fitting, or χ^2 fitting. Its algorithm is to adjust the parameters to minimize the sum of the squared deviations of the data from the model. Some things to be noted: 1)If your data are not really described by a normal error distribution, a least-squares fit will always pay too much attention to the "outlier" points. This can be a real problem. 2)If the errors are not normally distributed but you know the error distribution, you can still use the principle of maximum likelihood. This will give much better results for non-Gaussian systems (for instance, "counting" experiments with small numbers of counts.) Chapter 10 in Bevington and Robinson discusses the technique.

To minimize χ^2, take its derivative with respect to each of the α_j and set it equal to 0. If you can solve the resulting m equations to get $\alpha_1 ... \alpha_m$, you've done the first part of the problem, namely finding the best values of the model parameters. The second part of this problem is deciding how well the α_j parameters are determined.

3. Linear Regression. The model is

$$y(x) = f(x;\alpha_0,\alpha_1) = \alpha_0 + \alpha_1 x \qquad (49),$$

where we have a set of data $\{x_i,y_i\}$, and we want to find values for the y-intercept α_0 and the slope α_1. The formula for χ^2 is therefore

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - \alpha_0 - \alpha_1 x_i}{\sigma_i} \right)^2 \qquad (50).$$

The derivatives of χ^2 are needed to minimize the function

$$\frac{\partial \chi^2}{\partial \alpha_0} = -2 \sum_{i=1}^{N} \frac{(y_i - \alpha_0 - \alpha_1 x_i)}{\sigma_i^2} \quad \text{and} \quad \frac{\partial \chi^2}{\partial \alpha_1} = -2 \sum_{i=1}^{N} \frac{(y_i - \alpha_0 - \alpha_1 x_i) x_i}{\sigma_i^2} \qquad (51).$$

If wet set both derivatives equal to zero, we have two equations with two unknowns. After some simple manipulations, we have

$$\begin{cases} 0 = \sum_i \frac{y_i}{\sigma_i^2} - \alpha_0 \sum_i \frac{1}{\sigma_i^2} - \alpha_1 \sum_i \frac{x_i}{\sigma_i^2} \\ 0 = \sum_i \frac{x_i y_i}{\sigma_i^2} - \alpha_0 \sum_i \frac{x_i}{\sigma_i^2} - \alpha_1 \sum_i \frac{x_i^2}{\sigma_i^2} \end{cases} \quad (52).$$

Solving for α_0 (the intercept) and α_1 (the slope) gives:

$$\alpha_0 = \frac{\sum_i \frac{y_i}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \sum_i \frac{x_i}{\sigma_i^2} \sum_i \frac{x_i y_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \left(\sum_i \frac{x_i}{\sigma_i^2} \right)^2} \quad \text{and} \quad \alpha_1 = \frac{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i y_i}{\sigma_i^2} - \sum_i \frac{x_i}{\sigma_i^2} \sum_i \frac{y_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \left(\sum_i \frac{x_i}{\sigma_i^2} \right)^2} \quad (53).$$

To find the uncertainties in α_0 and α_1, we use propagation of error. B&R do the problem explicitly on pages 108 and 109. The formulas for α_0 and α_1 are differentiated with respect to each of the y_i and the propagation of error formula is used to evaluate the overall uncertainty. We simply quote the result:

$$\sigma_{\alpha_0}^2 = \frac{\sum_i \frac{x_i^2}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \left(\sum_i \frac{x_i}{\sigma_i^2} \right)^2} \quad \text{and} \quad \sigma_{\alpha_1}^2 = \frac{\sum_i \frac{1}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \left(\sum_i \frac{x_i}{\sigma_i^2} \right)^2} \quad (54).$$

The covariance of α_0 and α_1, which you might need if you plan to do further propagation of error with the values of α_0 and α_1 is

$$\sigma_{\alpha_0 \alpha_1}^2 = -\frac{\sum_i \frac{x_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2} \sum_i \frac{x_i^2}{\sigma_i^2} - \left(\sum_i \frac{x_i}{\sigma_i^2} \right)^2} \quad (55).$$

4. Goodness of Fit. Finally, we need a goodness-of-fit measure, in order to have some indication of whether the model (in this case, a straight line) describes the data in a reasonable way at all. It is possible to apply the formulas above to any set of $\{x_i, y_i\}$ data, whether or not they resemble a line! We obtain the goodness of fit by determining the probability that we would get a χ^2 as bad as the one we have, even though the model was correct.

To use the test, calculate the number of degrees of freedom $\nu = n\text{-}m$ from the number of points less the number of parameters. Table IV in Young (p. 163 and reconstructed at the end of this section) and Table C.4 in B&R give the probability P that

a value of χ^2 as bad as the one you obtained could occur by chance if the model is correct. Find your value of χ^2 in the row corresponding to ν, and look at the top of the table for P.

Critical Values of χ^2 from Young

degrees of freedom	Probability								
ν	0.99	0.98	0.95	0.90	0.10	0.05	0.02	0.01	0.001
1	0.0016	0.00628	0.0039	0.0158	2.706	3.841	5.412	6.635	10.827
2	0.0201	0.0404	0.103	0.211	4.605	5.991	7.824	9.210	13.815
3	0.115	0.185	0.352	0.584	6.251	7.815	9.837	11.341	16.268
4	0.297	0.429	0.711	1.064	7.779	9.488	11.668	13.277	18.465
5	0.554	0.752	1.145	1.610	9.236	11.070	13.388	15.086	20.517
6	0.872	1.134	1.635	2.204	10.645	12.592	15.033	16.812	22.457
7	1.239	1.564	2.167	2.833	12.017	14.067	16.622	18.475	24.322
8	1.646	2.032	2.733	3.490	13.362	15.507	18.168	20.090	26.125
9	2.088	2.532	3.325	4.168	14.684	16.919	19.679	21.666	27.877
10	2.558	3.059	3.940	4.865	15.987	18.307	21.161	23.209	29.588
11	3.053	3.609	4.575	5.578	17.275	19.675	22.618	24.725	31.264
12	3.571	4.178	5.226	6.304	18.549	21.026	24.054	26.217	32.909
13	4.107	4.765	5.892	7.042	19.812	22.362	25.472	27.688	34.528
14	4.660	5.368	6.571	7.790	21.064	23.685	26.873	29.141	36.123
15	5.229	5.985	7.261	8.547	22.307	24.996	28.259	30.578	37.697
16	5.812	6.614	7.962	9.312	23.542	26.296	29.633	32.000	39.252
17	6.408	7.255	8.672	10.085	24.769	27.587	30.995	33.409	40.790
18	7.015	7.906	9.390	10.865	25.989	28.869	32.346	34.805	42.312
19	7.633	8.567	10.117	11.651	27.204	30.144	33.687	36.191	43.820
20	8.260	9.237	10.851	12.443	28.412	31.410	35.020	37.566	45.315
21	8.897	9.915	11.591	13.240	29.615	32.671	36.343	38.932	46.797
22	9.542	10.600	12.338	14.041	30.813	33.924	37.659	40.289	48.268
23	10.196	11.293	13.091	14.848	32.007	35.172	38.968	41.638	49.728
24	10.856	11.992	13.848	15.659	33.196	36.415	40.270	42.980	51.179
25	11.524	12.697	14.611	16.473	34.382	37.652	41.566	44.314	52.620

If the errors are normally distributed, the model is correct, and your estimates of the measurement errors σ_i are good, you should get $P \geq 0.1$ or so. If you have underestimated the errors, or your experiment does not really have normally distributed errors, correct models can sometimes give values of P as low as perhaps 10^{-3}. Genuinely wrong models often give $P << 10^{-3}$. This method of checking goodness-of-fit works for all linear least squares problems, and works decently for nonlinear problems as well. *Numerical Recipes* describes how to calculate this P (they call it Q) so you needn't rely on the tables.

The reduced χ^2 of a fit is defined as

$$\chi^2_\upsilon = \frac{\chi^2}{\upsilon} = \frac{\sum_i \left(\frac{y_i - f(x_i;\alpha_j)}{\sigma_i^2} \right)^2}{N - m} \qquad (56).$$

This quantity is most useful for comparing the quality of fitted data sets taken under different or variable conditions (as is often the case when work is done by different investigators).

5. Example of Linear Regression with Individual Uncertainties. In the following an example is given for linear regression with uncertainties. This result may be profitably compared to a later example using the exact same set of $\{x_i,y_i\}$ but without uncertainties. Different final outcomes result from differences in the nature of the errors, not the $\{x_i,y_i\}$ set.

Linear Regression with Individual Uncertainties

$i := 0 .. 4$

input

$x_i :=$ | $y_i :=$ | $\sigma_i :=$

x_i	y_i	σ_i
1	2.133	0.1
2	2.987	0.2
3	4.134	0.4
4	4.992	0.8
5	6.087	1.6

weight

$$w_i := \frac{1}{(\sigma_i)^2}$$

$$\alpha_0 := \frac{\left(\sum_i w_i \cdot y_i\right)\cdot\left[\sum_i w_i\cdot(x_i)^2\right] - \left(\sum_i w_i\cdot x_i\right)\cdot\left(\sum_i w_i\cdot x_i\cdot y_i\right)}{\left(\sum_i w_i\right)\cdot\left[\sum_i w_i\cdot(x_i)^2\right] - \left(\sum_i w_i\cdot x_i\right)^2}$$

intercept

$\alpha_0 = 1.177$

$$\alpha_1 := \frac{\left(\sum_i w_i\right)\cdot\left(\sum_i w_i\cdot x_i\cdot y_i\right) - \left(\sum_i w_i\cdot x_i\right)\cdot\left(\sum_i w_i\cdot y_i\right)}{\left(\sum_i w_i\right)\cdot\left[\sum_i w_i\cdot(x_i)^2\right] - \left(\sum_i w_i\cdot x_i\right)^2}$$

slope

$\alpha_1 = 0.945$

$$\Delta\alpha_0 := \sqrt{\frac{\left(\sum_i w_i\cdot x_i\right)}{\left(\sum_i w_i\right)\cdot\left[\sum_i w_i\cdot(x_i)^2\right] - \left(\sum_i w_i\cdot x_i\right)^2}}$$

e.s.d. of intercept

$\Delta\alpha_0 = 0.154$

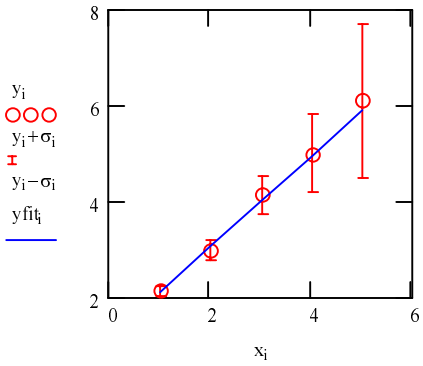
$$\Delta\alpha_1 := \sqrt{\frac{\left(\sum_i w_i\right)}{\left(\sum_i w_i\right)\cdot\left[\sum_i w_i\cdot(x_i)^2\right] - \left(\sum_i w_i\cdot x_i\right)^2}}$$

e.s.d. of slope

$\Delta\alpha_1 = 0.134$

$yfit_i := \alpha_0 + \alpha_1\cdot x_i$ $\quad \chi 2 := \sum_i \left(\frac{y_i - yfit_i}{\sigma_i}\right)^2$ $\quad \chi 2 = 0.28$ $\quad \chi 2reduced := \frac{\chi 2}{5-2}$ $\quad \chi 2reduced = 0.093$

i =	x_i =	y_i =	$yfit_i$ =	$y_i - yfit_i$ =
0	1	2.133	2.122	0.011
1	2	2.987	3.067	-0.08
2	3	4.134	4.011	0.123
3	4	4.992	4.956	0.036
4	5	6.087	5.901	0.186



6. Linear Regression Example without Individual Uncertainties. It often happens that one does not know the individual uncertainties σ_i , however it is still possible to obtain a least squares fit with uncertainties for the slope and intercept. When you do not know the individual errors σ_i, then it is **not** possible to obtain an estimate of goodness of fit, and you have no statistical arguments available to help you claim that your model function is a realistic one. One way, and perhaps the most commonly needed approach, is to assume that all the measurements have an equal uncertainty, ie. $\sigma_i=\sigma$. Under this assumption, a factor of $1/\sigma^2$ can be factored out of each summation and canceled entirely from the previous expressions for α_0 and α_1 giving

$$\alpha_0 = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i x_i \sum_i x_i y_i}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \quad \text{and} \quad \alpha_1 = \frac{N\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \qquad (57).$$

The uncertainties in the slope and intercept become

$$\sigma_{\alpha_0}^2 = \frac{\sigma^2 \sum_i x_i^2}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \quad \text{and} \quad \sigma_{\alpha_1}^2 = \frac{\sigma^2 N}{N\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \qquad (58),$$

but σ has not been determined yet. Under the assumption that $\sigma_i=\sigma$, the reduced χ^2 becomes related to S_y, the standard deviation in y:

$$\chi_\upsilon^2 = \frac{\chi^2}{\upsilon} = \frac{1}{\sigma^2}\frac{\sum_i \left(y_i - f(x_i;\alpha_j)\right)^2}{(N-m)} = \frac{S_y^2}{\sigma^2} \qquad (59),$$

where S_y^2 is the experimental estimate of the variance in y. At this point one typically assumes that the χ_υ^2 is equal to one, i.e. that we are using the perfect model, so

$$\sigma^2 \approx S_y^2 = \frac{\sum_i (y_i - \alpha_0 - \alpha_1 x_i)^2}{N-2} \qquad (60),$$

which may be substituted into the expressions above for the uncertainties in the slope and intercept. In a sense, we are putting in the weights after performing the fit. To summarize, under the $\sigma_i=\sigma$ assumption, you calculate the intercept and slope, α_0 and α_1, from summations involving only x_i , y_i, and N. Then calculate the S_y, the standard deviation in y, from summations involving x_i, y_i, N, and the newly determined α_0, and α_1. Finally, calculate the uncertainties in the intercept and slope from summations involving $S_y\approx\sigma$, x_i, and N. Remember, when you use these equations, you are giving up any information about the goodness of the fit.

6.1 Mathcad Example of Linear Regression without Uncertainties.

Linear Regression without Individual Uncertainties

$i := 0 .. 4$

input

$x_i :=$ $y_i :=$

x_i	y_i
1	2.133
2	2.987
3	4.134
4	4.992
5	6.087

$$\alpha_0 := \frac{\left(\sum_i y_i\right)\cdot\left[\sum_i (x_i)^2\right] - \left(\sum_i x_i\right)\cdot\left(\sum_i x_i\cdot y_i\right)}{\left(\sum_i 1\right)\cdot\left[\sum_i (x_i)^2\right] - \left(\sum_i x_i\right)^2}$$

intercept

$\alpha_0 = 1.093$

$$\alpha_1 := \frac{\left(\sum_i 1\right)\cdot\left(\sum_i x_i\cdot y_i\right) - \left(\sum_i x_i\right)\cdot\left(\sum_i y_i\right)}{\left(\sum_i 1\right)\cdot\left[\sum_i (x_i)^2\right] - \left(\sum_i x_i\right)^2}$$

slope

$\alpha_1 = 0.991$

$yfit_i := \alpha_0 + \alpha_1 \cdot x_i$

$$Sy := \sqrt{\frac{\sum_i (y_i - yfit_i)^2}{5 - 2}}$$

$Sy = 0.083$

$$\Delta\alpha_0 := Sy\cdot\sqrt{\frac{\left(\sum_i x_i\right)}{\left(\sum_i 1\right)\cdot\left[\sum_i (x_i)^2\right] - \left(\sum_i x_i\right)^2}}$$

e.s.d. of intercept

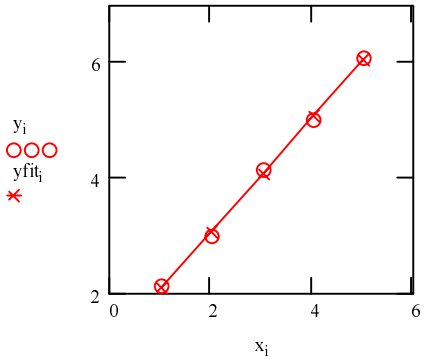
$\Delta\alpha_0 = 0.045$

$$\Delta\alpha_1 := Sy\cdot\sqrt{\frac{\left(\sum_i 1\right)}{\left(\sum_i 1\right)\cdot\left[\sum_i (x_i)^2\right] - \left(\sum_i x_i\right)^2}}$$

e.s.d. of slope

$\Delta\alpha_1 = 0.026$

i =	x_i =	y_i =	$yfit_i$ =	$y_i - yfit_i$ =
0	1	2.133	2.084	0.049
1	2	2.987	3.075	-0.088
2	3	4.134	4.067	0.067
3	4	4.992	5.058	-0.066
4	5	6.087	6.049	0.038



7. General Mathcad Program, linfit.mcd, that Covers both Cases (With and Without Individual Uncertainites). The following mathcad template is made available of the course web site to aid students in fitting data. It is imperative that students distinguish whether there are or are not individual uncertainties available, as the weighting is different and the parameters characterizing the quality of the fit are different in each case. When you have individual uncertainties, they are used to determine individual weights, and one reports χ^2 or reduced χ^2 and the confidence in the whole fit (by virtue of the probabilitiy of exceeding χ^2 by virtue of the table in section 4, Critical Values of χ^2 from Young. When you do not have individual uncertainties, you generally assume all points have the same weight and set all weights to 1. You characterize the quality of the fit by reporting the standard deviation in y (as given by eq. 60). Do not report χ^2 or reduced χ^2 in this case.

On the following page is a Mathcad template that chooses the case of individual uncertainties by whether you type individual uncertainties or not into an input table at the beginning of the template. The data is typed into a table (matrix) called data that is pictured below. The first column is the independent variable (x), the second column is the dependent variable (y), while the third column is for the uncertainty or error in y. In the example pictured, there are no values entered into the third column, so the program sets all of the weights equal to 1. In this case the Mathcad template parameter, S1, is the standard deviation in y. It should be reported to characterized the quality of the fit. If there were individual uncertainties placed in third column of "data", then S1 would be reduced χ^2 and it would be reported along with the confidence in the whole fit. The data is fit to the equation

data :=

1	2	
2	4.6	
3	7.2	
4	9	
5	11.3	
6	12.2	
7	14.9	

$$y(x, p_0, p_1, ..) = p_0 \cdot F(x)_0 + p_1 \cdot F(x)_1 + ..$$

where $F(x)_0$ and $F(x)_1$ are determined by

$$F(x) := \begin{pmatrix} 1 \\ x \end{pmatrix}$$

In this case, we are fitting $y(x, p_0, p_1) = p_0 + p_1 x$. In either uncertainty case, the results are reported as the fit parameters, their estimated standard deviations, and the Variance/Covariance matrix (diagonal elements are the variances of the parameters, i.e. the uncertainties squared, off diagonal elements are the covariances, as per Part 1, Section 4.2, eq. 14]

3. Results

npts = 7

Parameters	Uncertainties	Covariance Matrix
$p = \begin{pmatrix} 0.4571 \\ 2.0714 \end{pmatrix}$	$\Delta p = \begin{pmatrix} 0.437 \\ 0.098 \end{pmatrix}$	$Covar = \begin{pmatrix} 0.191 & -0.038 \\ -0.038 & 9.531 \times 10^{-3} \end{pmatrix}$

Finally, you can change this template in order to fit other functional forms. For example, if you change F(x) to

$$F(x) := \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}$$

you fit $y(x, p_0, p_1) = p_0 + p_1 x + p_2 x^2$.

Linear Least Squares

1. Input (x, y) pairs and standard deviations, if known.

data :=

1	2	
2	4.6	
3	7.2	
4	9	
5	11.3	

Double-click table to edit, or right-click and choose Import... to read from a data file. First column is x, second column is y, third column (optional) is the standard deviation for each y value.

2. Enter vector of fitting functions.

$F(x) := \begin{pmatrix} 1 \\ x \end{pmatrix}$ Use insert matrix from insert menu to modify.

Fitting function has the form,

$y(x, p_0, p_1, ..) = p_0 \cdot F(x)_0 + p_1 \cdot F(x)_1 + ..$

Solve for best fit parameters using the least squares method.

$x := data^{\langle 0 \rangle}$ $y := data^{\langle 1 \rangle}$ $npts := length(y)$ number of data pairs

$i := 0 .. npts - 1$ $npar := length(F(1))$ number of parameters in model function

$\Delta y_i := if(cols(data) < 3, 1, data_{i,2})$ Set standard deviations to 1, if no values were specified in column 3. In this case the uncertainties in the parameters will be estimated from the quality of the fit. *This analysis assumes that the model is correct.*

$j := 0 .. npar - 1$

$k := 0 .. npar - 1$

$w_i := \frac{1}{(\Delta y_i)^2}$ weights

$A_{j,k} := \sum_i w_i \cdot F(x_i)_j \cdot F(x_i)_k$ $h_k := \sum_i w_i \cdot y_i \cdot F(x_i)_k$

$B := A^{-1}$ $p := B \cdot h$

$yfit(x) := \sum_k p_k \cdot F(x)_k$ This defines the best-fit function. Evaluate at any x to determine the calculated value of the best-fit function at that x.

$chisq := \sum_i (y_i - yfit(x_i))^2 \cdot w_i$ $S1 := \sqrt{\frac{chisq}{npts - npar}}$ Estimated standard deviation of a point of unit weight, should be reasonably close to 1.

$S1 = 0.517$

$Covar := if(cols(data) > 2, B, B \cdot S1^2)$ Scale covariance matrix correctly, if y errors are not given.

$\Delta p_j := \sqrt{Covar_{j,j}}$

$Q := 1 - pchisq(chisq, npts - npar)$ Goodness of fit. Should be > 0.001, meaningless without y errors.

$Q = 0.931$

3. Results

$npts = 7$

Parameters

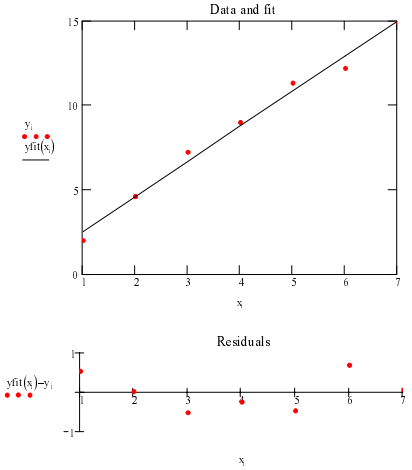
$p = \begin{pmatrix} 0.4571 \\ 2.0714 \end{pmatrix}$

Uncertainties

$\Delta p = \begin{pmatrix} 0.437 \\ 0.098 \end{pmatrix}$

Covariance Matrix

$Covar = \begin{pmatrix} 0.191 & -0.038 \\ -0.038 & 9.531 \times 10^{-3} \end{pmatrix}$



8. General Linear Least Squares Approach. There is no reason to limit ourselves to fitting only straight lines. The general linear least squares problem can be very simply stated in a matrix formalism. If you can master this method, then a means of inverting square matrices is all that stands between you and the ability to fit any model form which is linear in its fitting parameters, that is

$$f(x;\alpha_0,\alpha_1 \ldots \alpha_{m-1}) = \alpha_0 f_0(x) + \alpha_1 f_1(x) + \ldots + \alpha_{m-1} f_{m-1}(x) \qquad (61).$$

The f_j can be nonlinear functions (exponentials, cosines, etc) as long as they contain no adjustable parameters. If we are fitting a set of data $\{x_i,y_i\}$ for i=1-N to the form $y(x)=f(x;\alpha_j)$ for m parameters (with j=0 to m-1) in which $f(x;\alpha_j)$ is linear in the coefficients α_j, then the fitting parameters are given by

$$\boldsymbol{\alpha} = \mathbf{A}^{-1}\mathbf{H} \qquad (62),$$

where $\boldsymbol{\alpha}$ is a column vector of the fitting parameters, and **A** and **H** are defined below,

$$A_{jk} = \sum_i \frac{1}{\sigma_i^2}\left(\frac{\partial f(x_i;\alpha_0 \ldots \alpha_{m-1})}{\partial \alpha_j}\right)\left(\frac{\partial f(x_i;\alpha_0 \ldots \alpha_{m-1})}{\partial \alpha_k}\right) \qquad (63), \text{ and}$$

$$H_j = \sum_i \frac{1}{\sigma_i^2}\left(\frac{\partial f(x_i;\alpha_0 \ldots \alpha_{m-1})}{\partial \alpha_j}\right) y_i \qquad (64),$$

where k (like j) is an index over the number of fitting parameters. The variances of the fitting parameters can be found on the diagonal of the inverted **A** matrix and the covariances are off diagonal, so $\mathbf{A}^{-1}$ is called the variance-covariance matrix. The fitting parameter errors are

$$S_{\alpha_j}^2 = A_{jj}^{-1} \qquad (65).$$

For example, the linear least squares problems that we have addressed earlier are a special case where m=2, $f_0(x)=1$, and $f_1(x)=x$, to obtain $y=\alpha_0+\alpha_1 x$. So the matrix equation is

$$\begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix} = \begin{bmatrix}A_{00} & A_{01}\\ A_{10} & A_{11}\end{bmatrix}^{-1}\begin{bmatrix}H_0\\ H_1\end{bmatrix} = = \frac{1}{A_{00}A_{11}-A_{01}A_{10}}\begin{bmatrix}A_{11} & -A_{01}\\ -A_{10} & A_{00}\end{bmatrix}\begin{bmatrix}H_0\\ H_1\end{bmatrix}$$

$$= \begin{bmatrix}\dfrac{A_{11}H_0 - A_{01}H_1}{A_{00}A_{11}-A_{01}A_{10}}\\ \dfrac{-A_{10}H_0 + A_{00}H_1}{A_{00}A_{11}-A_{01}A_{10}}\end{bmatrix} \qquad (66).$$

and variances (e s.d.)^2 are

$$\begin{bmatrix} S^2_{\alpha_0} \\ S^2_{\alpha_1} \end{bmatrix} = \begin{bmatrix} \dfrac{A_{11}}{A_{00}A_{11} - A_{01}A_{10}} \\ \dfrac{A_{00}}{A_{00}A_{11} - A_{01}A_{10}} \end{bmatrix} \tag{67}.$$

To use the above expressions for A_{jk}, and H_j, you need to be able to take single partial derivatives of the f_j with respect to α_j (simply the rest of the term without α_j). In this case

$$A_{00} = \sum_i \frac{1}{\sigma_i^2} \quad A_{01} = A_{10} = \sum_i \frac{x_i}{\sigma_i^2} \quad A_{11} = \sum_i \frac{x_i^2}{\sigma_i^2} \quad H_0 = \sum_i \frac{y_i}{\sigma_i^2} \quad H_1 = \sum_i \frac{x_i y_i}{\sigma_i^2} \tag{68},$$

giving the same expressions as determined previously, but this set of equations is easily adapted to any other functional forms with linear fitting coefficients.

9. Linearizable Models. Often the physical model function will vary nonlinearly with one or more parameters; one common example is the exponential model,

$$y(x) = \alpha_0 e^{-\alpha_1 x} \tag{69},$$

where α_0 and α_1 are the parameters to be obtained by fitting a set of $\{x_i,y_i\}$ data. Differentiating χ^2 with respect to α_0 and α_1 and setting the derivatives equal to 0 gives a set of nonlinear equations, which cannot be solved in a simple way. The algebraic methods useful for fitting linear models are therefore not applicable.

In the example above, it is possible to transform the data in a way that gives a linear model:

$$\ln[y(x)] = \ln(\alpha_0) - \alpha_1 x \tag{70},$$

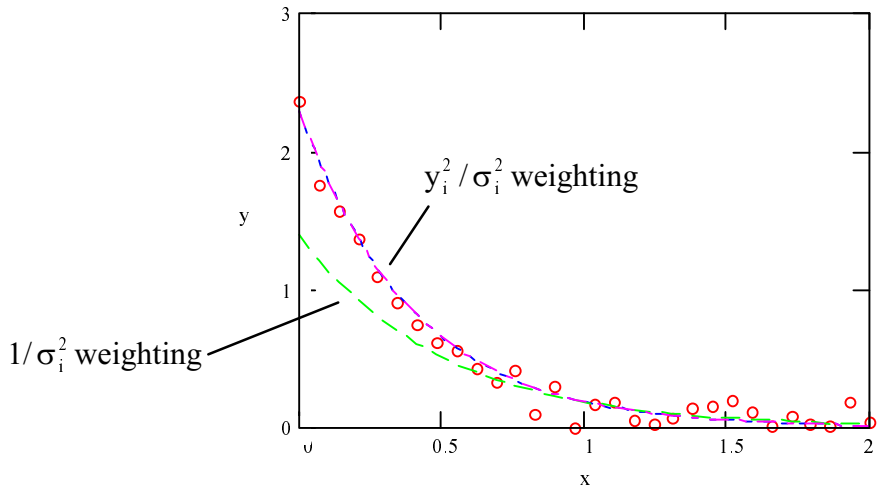
so that plotting ln(y) vs. x should give a straight line with intercept $\ln(\alpha_0)$ and intercept $-\alpha_1$. Linear least squares may be used to fit the transformed data. This is a completely legitimate strategy, and is often useful for simple models. However, to use it correctly, you *must* correctly weight the transformed data, i.e. correctly transform the errors. This is an important step which is often overlooked.

The weights in a least-squares fit are the numbers which multiply the squares of the deviations of each data point from the model function in the calculation of χ^2. For linear fits of raw $\{x_i,y_i\}$ data, the weights are simply $1/\sigma_i^2$ as can be seen in sections 3 and 5. If you make a linear fit to transformed data, you must do a simple propagation of error calculation to find the appropriate uncertainties in the transformed data.

For example, if the y_i values have a constant error of σ_y in the exponential model given above, then that error will not be constant in the ln(y). Since

$$\frac{\partial \ln(y)}{\partial y} = \frac{1}{y} \qquad \text{then} \qquad \sigma^2_{\ln y} = \frac{1}{y^2}\sigma_y^2 \tag{71}.$$

The correct weights in a least-squares fit of ln(y) vs. x with uncertainties in y (σ_i) are therefore y_i^2/σ_i^2, rather than $1/\sigma_i^2$. The fig. below shows a simulated set of data created from the first equation in this section, with $\alpha_0 = 2.3$, $\alpha_1 = 2.5$, and normally distributed noise with standard deviation 0.1 added to the y values. Also shown are the results of a least-squares fit to eq. 70 with uniform weighting($1/\sigma_i^2$ weighting) and a fit with correct weighting (y_i^2/σ_i^2 weighting).



Note that the uniformly weighted linear fit diverges from the data set at the early part of the decay. That's a crucial error, since those early points carry the most information about the α_1 parameter (which might be a rate constant and is usually the one you're most interested in). Since the correctly weighted fit pays most attention to those, it does a much better job. In fact on this plot it is not distinguishable from the original analytical function before noise was added. Note that the parent error distribution of the transformed data is not exactly normal, even though the parent distribution of the raw data was normal. The technique of χ^2 fitting is therefore not strictly applicable to the data once it has been transformed. Usually this subtlety is not particularly important, since the transformed parent distribution is still close to Gaussian. It does, however, produce small errors in the parameters.

10. Nonlinear Least Squares. If there is no simple transformation of the model which produces a linear fitting problem (or even if there is but you want to avoid the problems already mentioned), then you must find a way to do the χ^2 minimization without the benefit of linear least squares. Generally you must resort to iterative minimization techniques.

Nonlinear functions can be treated if the nonlinear model function can be *linearized* by expanding the function in a Taylor series about an initial guess of its fitting parameters and keeping only the first order terms. This linearized approximation to the function can be solved with the linear least squares methods already discussed to give a better approximation to the fitting constants than the initial guess. The process is then iterated using each better approximation to the fitting constants as a new initial guess

until the approximations stop improving as judged by χ^2. The formalism is a little different than the linear case. The basic equation for each iteration is

$$\boldsymbol{\Delta\alpha = A^{-1}H} \qquad (72),$$

where $\boldsymbol{\Delta\alpha}$ is a column vector of change in the fitting parameters from an initial guess of the parameters, α_j^o. The new approximation is $\alpha_j = \Delta\alpha_j + \alpha_j^o$. For the nonlinear case **A** and **H** are defined as,

$$A_{jk} = \sum_i \frac{1}{\sigma_i^2}\left(\frac{\partial f(x_i;\alpha_0^o \ldots \alpha_{m-1}^o)}{\partial \alpha_j^o}\right)\left(\frac{\partial f(x_i;\alpha_0^o \ldots \alpha_{m-1}^o)}{\partial \alpha_k^o}\right) \qquad (73),$$

$$H_j = \sum_i \frac{1}{\sigma_i^2}\left(\frac{\partial f(x_i;\alpha_0^o \ldots \alpha_{m-1}^o)}{\partial \alpha_j^o}\right)(y_i - y_i^o) \qquad (74),$$

where the y_i^o are calculated from the guessed parameters α_j^o and the derivatives are determined about the guessed parameters.

Each iteration determines a value of χ^2 which can be used to characterize χ^2 's hyperspace which depends on the α_j fitting parameters. Imagine a "terrain" where the distances north-south and east-west correspond to the values of your parameters (the image only works for two parameters, but the math works for any number). The altitude corresponds to the value of χ^2 . Your task is to find the lowest point in this terrain. To make the metaphor more realistic, you cannot "survey" the terrain by looking all around; you must be satisfied with evaluating the altitude at specific points. Several methods are available to you. They are outlined in Chapter 8 of B&R.

10.1 Grid Search. This is the most obvious (and usually one of the slowest) techniques. You hold all but one of your parameters fixed, and you vary that one, looking for the value which produces the minimum value of χ^2 . You then fix that one at the optimum value you just found, choose another parameter, and vary it. When you have worked your way through all the parameters once, you usually find that the value of the first one is no longer optimum, so you start over. This method corresponds to limiting yourself to moving only north-south or east-west along your unknown terrain; you never permit yourself to move northeast, say. When you find that all your parameters seem to be optimum, you quit.

10.2 Gradient Search, or Steepest Descent Search. In this technique you start at some position and evaluate the *gradient* (the vector of partial derivatives of χ^2 with respect to the parameters). The gradient shows you the direction which is steepest downhill. You move in that direction until you find yourself moving uphill again. You then reevaluate the gradient, and repeat. When you can find no downhill directions (the gradient is zero), you stop. This method is almost always faster than the grid search, but it can still be

pretty slow, especially near the true minimum. It has the advantage that it always finds *some* minimum (assuming the surface has one.)

10.3 Expansion methods: Here you assume that you are already pretty close to the minimum, and therefore a low-order Taylor expansion of the χ^2 function with respect to the parameters is likely to be pretty good. By evaluating several derivatives at your current position, you can obtain enough information to model the function as a paraboloid or ellipsoid. You then move to the position you calculate to be the minimum of that surface, and start over. When you are in fact close to the minimum, this works great and is very fast. When you are far away, though, it is unstable and can carry you on awful wild goose chases.

10.4 Marquardt method: The Marquardt method is a way of going smoothly from the gradient search method to the expansion method as you get closer to the minimum. It works very well, and is the industry standard. It's a little complicated to program (though not actually difficult), but there are lots of available programs which already include it.

Recommendations: When you have it available, a Marquardt fit is almost always fastest and is easy to perform. Often the program will calculate the required derivatives $\partial\chi^2 / \partial\alpha_j$ numerically for you, leaving you to specify only the model function, the list of parameters to vary, and the data. Good implementations will let you specify individual weights for the data points if you wish. If you have no Marquardt program available, grid-search (easy but slow) and gradient-search methods work pretty well. The expansion method is dangerous unless you are quite sure that you are starting with very good guesses for the parameter values (that is, you are already quite close to the minimum).

Okay, you have now performed a Marquardt fit and your program has kicked out "optimum" values of the parameters and their standard deviations. What do you do now? Don't believe the program's answer yet. It may have converged to a false minimum; your model may not actually fit the data; the stated standard deviations may be garbage because of correlations between different parameters; or you may have started out in a flat region of χ^2 hyperspace to far from the minimum for the "smart" fitting routine. Things you can do:

- First, make a plot of the *residuals*, the differences between the data points and your model: $r_i = y_i - f(x_i, \alpha_1, \alpha_2...)$. The residuals should bounce up and down randomly about zero. If they have a big hump in them somewhere, then either your model does not describe the data well or you have not found the true minimum in χ^2.
- Second, if the residuals look okay, do a goodness-of-fit test as described in section **3.3**. (You must have used true standard deviations in your weighting to have a useful absolute magnitude of χ^2. You should come up with a reasonably large value of P, certainly at least 0.001.
- Finally, you need to evaluate the uncertainties in the values of your fitted parameters. (If the first two tests fail, the uncertainties (and probably the fitted parameters too) will be meaningless.)

There are several options for dealing with the uncertainties in fitted parameters. Good fitting programs will provide you with estimated standard deviations for individual parameters, and should also provide you with covariances giving their correlations.

- You can use the standard deviations for the parameters which your nonlinear fitting program provided. Those will be meaningful only if the errors in your experiment were truly normal, your model function is an accurate description of the data, and the goodness of fit is okay. Then the calculated standard deviations are good for *individual* parameters, that is, are useful if you care about only one of the parameter values. To get confidence intervals on any one of the values, you can use the t-table as we did with simple averages.

- You can use a projection method, described in SGN 6th ed. on page 726. That will give you more accurate confidence limits when the parameters are strongly correlated or the errors are slightly nonnormal.

- You can do a Monte Carlo simulation, generating many synthetic data sets and subjecting each of them to your nonlinear fit. The resulting lists of parameter values can then be treated separately (as discussed before in section **1.3.2** to obtain individual confidence intervals, or plotted together to get joint intervals. The latter treatment allows you to say "I am 95% confident that a lies between 1.1 and 1.5 units *and* that b lies between 3.2 and 4.1 units."

- You can do a full-blown multidimensional analysis, giving the boundaries of the *error ellipsoid* for your data. This, too, will permit you to make statements about more than one of the values. See *Numerical Recipes* section 15.6 (second edition in C) for details.

-

10.5 Nonlinear Least Squares Example with Mathcad. A nonlinear least squares Mathcad routine follows that fits singlepoint calculations of the electronic energy of the H_2 molecule as a function of bond distance to the Morse potential function. You can replace the Morse potential function definition with any function of interest, being careful

$$F(r, DE, a, re, con) := DE\cdot\left[1 - e^{-a\cdot(r-re)}\right]^2 + con$$

to redefine the new variable within the worksheet. The error analysis is performed using numerical differentiation (as discussed in Part 1, Section 4.1, p.5). Please use this template as a starting point for any nonlinear fitting problems that you might encounter in the future. finally, remember that nonlinear least squares fitting is much fussier than linear least squares fitting. Use the plot to make sure that the routine is focussing-in on the answer. There are many ways for nonlinear routines to get into flat portions of the χ^2 hyperspace, at which point they stop. Try many different initial conditions, to see if there are different local minima, and make sure that the fitted line mimics the data.

Fit to a Morse Potential

input the data (distance,energy)

$A := READPRN("h2.txt")$ $npts := rows(A)$ $i := 0..npts - 1$ $r_i := A_{i,0}$ $E_i := A_{i,1}$

fit to a Morse potential, eq. 14.2-36 in Mortimer
DE=equilibrium bond dissociation energy
a=curvature parameter=(k/(2*DE))**.5 where k is Hooke's Law constant
re=equilibrium bond distance
con=constant to make dissociation limit zero in energy, i.e. the energy reference

$$F(r, DE, a, re, con) := DE\left[1 - e^{-a\cdot(r-re)}\right]^2 + con$$

we want to minimize the sum of the squares of the deviations by varying the parameters

$$SSE(DE, a, re, con) := \sum_i \left(E_i - F(r_i, DE, a, re, con)\right)^2$$

Minerr fitting procedure initial guesses $DE := .4$ $a := 2$ $re := .7$ $con := -1.0$

Given $DE > 0$ $re > 0$ $a > 0$ $SSE(DE, a, re, con) = 0$

(you need four constraints for four unknowns)

$$\begin{pmatrix} DE \\ a \\ re \\ con \end{pmatrix} := Minerr(DE, a, re, con)$$

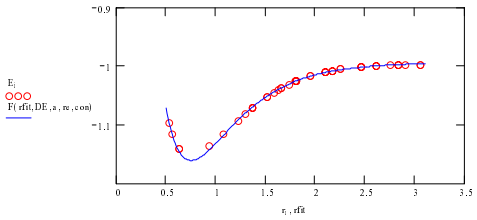
results: the fitted parameters

$DE = 0.167$ Hartree $a = 2.194$ 1/angstrom $re = 0.75$ ang. $con = -1.161$ Hartree
$DE\,27.21 = 4.538$ eV

vib. freq. $\nu := \frac{a\cdot 10^{10}}{2\cdot\pi}\cdot\sqrt{\frac{2\cdot DE\,27.21\,1.602\,10^{-19}}{\left(\frac{1\cdot 1}{1+1}\right)\cdot 1.6606\,10^{-27}}}$ $\nu = 1.461\times 10^{14}$ Hz $\frac{\nu}{2.997910^{10}} = 4.874\times 10^{3}$ cm^{-1}

plot results

$rfit := .5, .51.. 3.1$



Error Analysis

switch fitting parameters into a matrix

$j := 0..3$ $k := 0..3$ $\alpha_0 := DE$ $\alpha_1 := a$ $\alpha_2 := re$ $\alpha_3 := con$

standard deviation of fit $\sigma := \sqrt{\frac{\sum_i \left(E_i - F(r_i, DE, a, re, con)\right)^2}{npts - 4}}$ $\sigma = 2.103\times 10^{-3}$

Calculate derivatives numerically $d := 1000$

$$F1(j,r) := F\left(r, \alpha_0 + \frac{\alpha_0}{d}\cdot\delta(0,j), \alpha_1 + \frac{\alpha_1}{d}\cdot\delta(1,j), \alpha_2 + \frac{\alpha_2}{d}\cdot\delta(2,j), \alpha_3 + \frac{\alpha_3}{d}\cdot\delta(3,j)\right)$$

$$F2(j,r) := F\left(r, \alpha_0 - \frac{\alpha_0}{d}\cdot\delta(0,j), \alpha_1 - \frac{\alpha_1}{d}\cdot\delta(1,j), \alpha_2 - \frac{\alpha_2}{d}\cdot\delta(2,j), \alpha_3 - \frac{\alpha_3}{d}\cdot\delta(3,j)\right)$$

$$F3(k,r) := F\left(r, \alpha_0 + \frac{\alpha_0}{d}\cdot\delta(0,k), \alpha_1 + \frac{\alpha_1}{d}\cdot\delta(1,k), \alpha_2 + \frac{\alpha_2}{d}\cdot\delta(2,k), \alpha_3 + \frac{\alpha_3}{d}\cdot\delta(3,k)\right)$$

$$F4(k,r) := F\left(r, \alpha_0 - \frac{\alpha_0}{d}\cdot\delta(0,k), \alpha_1 - \frac{\alpha_1}{d}\cdot\delta(1,k), \alpha_2 - \frac{\alpha_2}{d}\cdot\delta(2,k), \alpha_3 - \frac{\alpha_3}{d}\cdot\delta(3,k)\right)$$

$$a_{j,k} := \sum_i \frac{F1(j,r_i) - F2(j,r_i)}{\left(\frac{2\cdot\alpha_j}{d}\right)}\cdot\frac{F3(k,r_i) - F4(k,r_i)}{\left(\frac{2\cdot\alpha_k}{d}\right)}$$

$ainv := a^{-1}$ $\Delta\alpha_j := \sigma\cdot\sqrt{ainv_{j,j}}$

the parameters and their uncertainties

$\alpha_j =$	$\Delta\alpha_j =$
0.167	$1.385\cdot10^{-3}$
2.194	0.025
0.75	$1.96\cdot10^{-3}$
-1.161	$1.429\cdot10^{-3}$

Variance Covariance Matrix

$$ainv = \begin{pmatrix} 0.433 & 3.001 & 0.119 & -0.393 \\ 3.001 & 144.163 & -2.367 & -5.896 \\ 0.119 & -2.367 & 0.868 & -0.015 \\ -0.393 & -5.896 & -0.015 & 0.461 \end{pmatrix}$$