

Non-linear least squares

The sum of squares of residuals

Consider a set of m data points, $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ and a curve (model function) $y = f(x, \beta)$ that in addition to the variable x also depends on n parameters, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ with $m > n$. It is desired to find the vector β of parameters such that the curve fits best the given data in the least squares sense, that is, the sum of squares

$$S = \sum_{i=1}^m r_i^2$$

is minimized, where the residuals (errors) r_i are given by

$$r_i = y_i - f(x_i, \beta)$$

for $i = 1, 2, \dots, m$.

The minimization criterion

The minimum value of S occurs when the gradient is zero. Since the model contains n parameters there are n gradient equations:

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0 \quad (j = 1, \dots, n)$$

In a non-linear system, the derivatives are functions of both the independent variable and the parameters, so these gradient equations do not have a closed solution. Instead, initial values must be chosen for the parameters. Then, the parameters are refined iteratively, that is, the values are obtained by successive approximation,

$$\beta_j \approx \beta_j^{k+1} = \beta_j^k + \Delta\beta_j$$

Calculation of the residuals

Here, k is an iteration number and the vector of increments, $\Delta\beta$ is known as the shift vector. At each iteration the model is linearized by approximation to a first-order Taylor's series expansion about β^k

$$f(x_i, \beta) \approx f(x_i, \beta^k) + \sum_j \frac{\partial f(x_i, \beta^k)}{\partial \beta_j} (\beta_j - \beta_j^k) \approx f(x_i, \beta^k) + \sum_j J_{ij} \Delta\beta_j$$

The Jacobian, \mathbf{J} , is a function of constants, the independent variable *and* the parameters, so it changes from one iteration to the next. Thus, in terms of the linearized model,

$$\frac{\partial r_i}{\partial \beta_i} = -J_{ij}$$

and the residuals are given by

$$r_i = \Delta y_i - \sum_{s=1} J_{is} \Delta\beta_s; \Delta y_i = y_i - f(x_i, \beta^k)$$

The normal equations in matrix form

Substituting these expressions into the gradient equations, they become

$$-2 \sum_{i=1}^m J_{ij} \left(\Delta y_i - \sum_{s=1}^n J_{is} \Delta \beta_s \right) = 0$$

which, on rearrangement, become n simultaneous linear equations, the **normal equations**

$$\sum_{i=1}^m \sum_{s=1}^n J_{ij} J_{is} \Delta \beta_s = \sum_{i=1}^m J_{ij} \Delta y_i$$

for $j = 1, \dots, n$.

The normal equations are written in matrix notation as

$$(J^T J) \Delta \boldsymbol{\beta} = J^T \Delta \mathbf{y}$$

Weighted sum of squares

When the observations are not equally reliable, a weighted sum of squares may be minimized,

$$S = \sum_i W_{ii} r_i^2$$

Each element of the diagonal weight matrix \mathbf{W} should, ideally, be equal to the reciprocal of the error or variance of the measurement. The normal equations are then

$$(\mathbf{J}^T \mathbf{W} \mathbf{J}) \Delta \boldsymbol{\beta} = \mathbf{J}^T \mathbf{W} \Delta \mathbf{y}$$

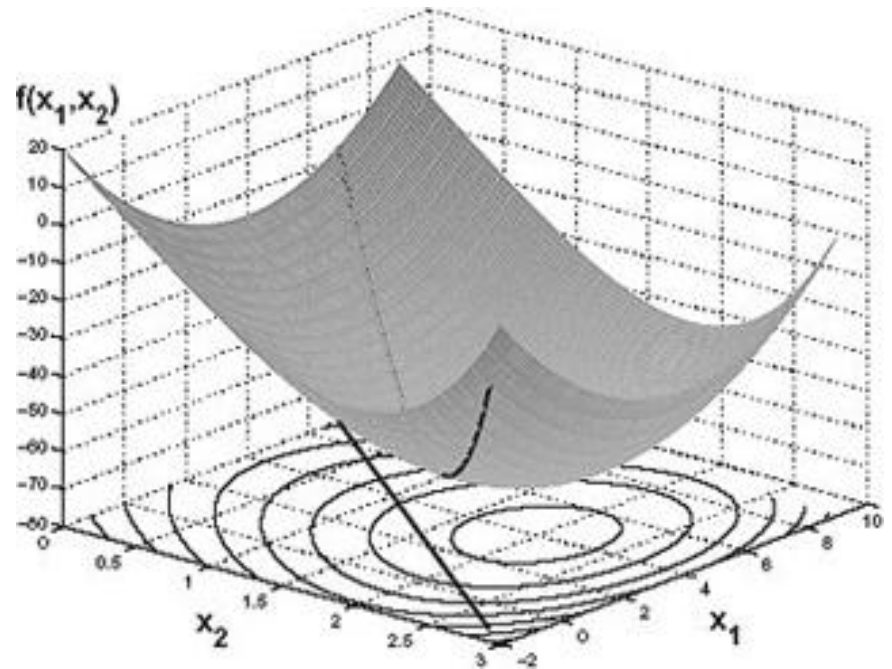
These equations form the basis for the Gauss-Newton algorithm for a non-linear least squares problem.

The parameter surface

In linear least squares the objective function, S , is a quadratic function of the parameters.

$$S = \sum_i W_{ii} \left(y_i - \sum_j X_{ij} \beta_j \right)^2$$

The minimum parameter values are to be found at the minimum of a surface in parameter space. With two or more parameters the contours of S with respect to any pair of parameters will be concentric ellipses.

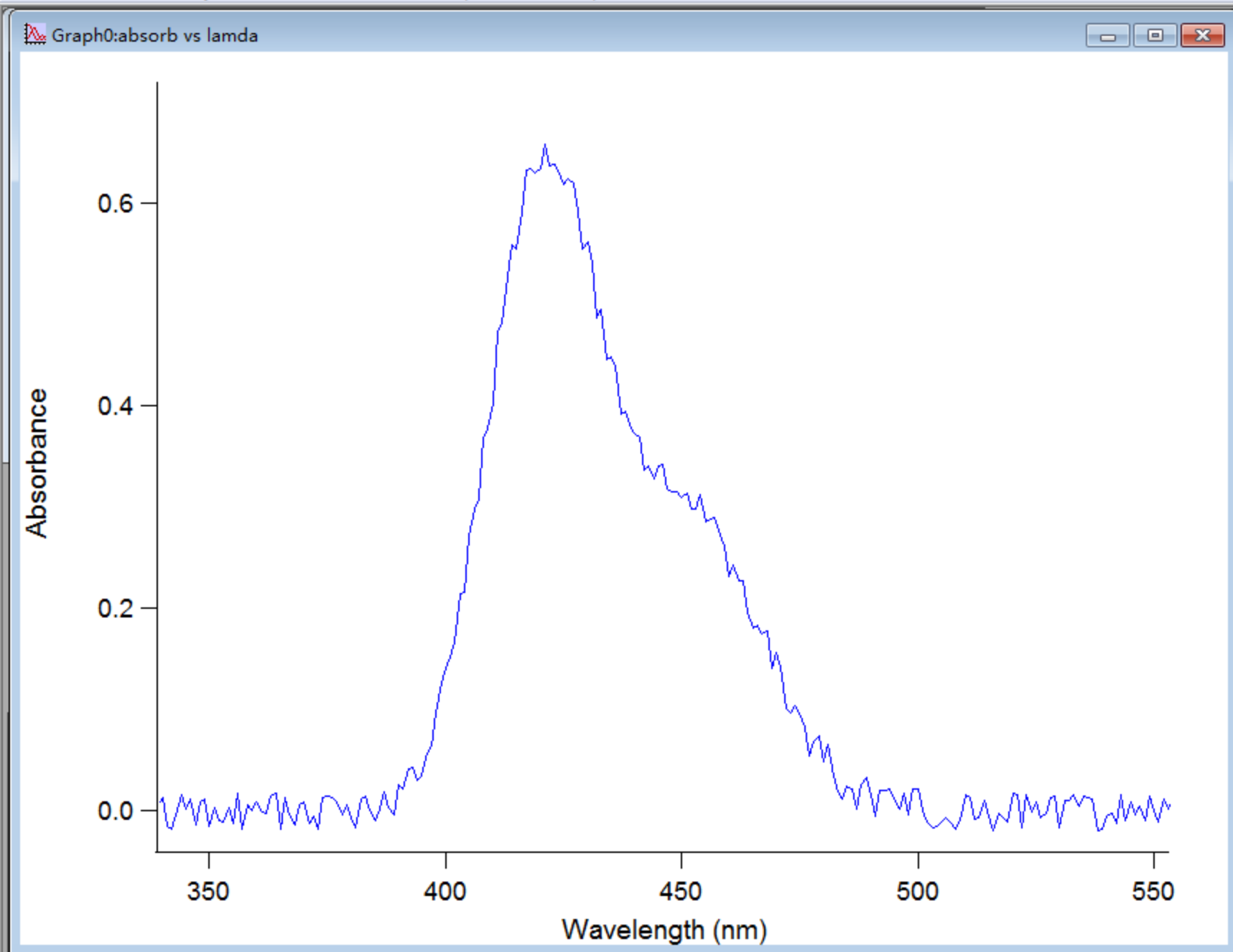


Approximating the surface as a quadratic

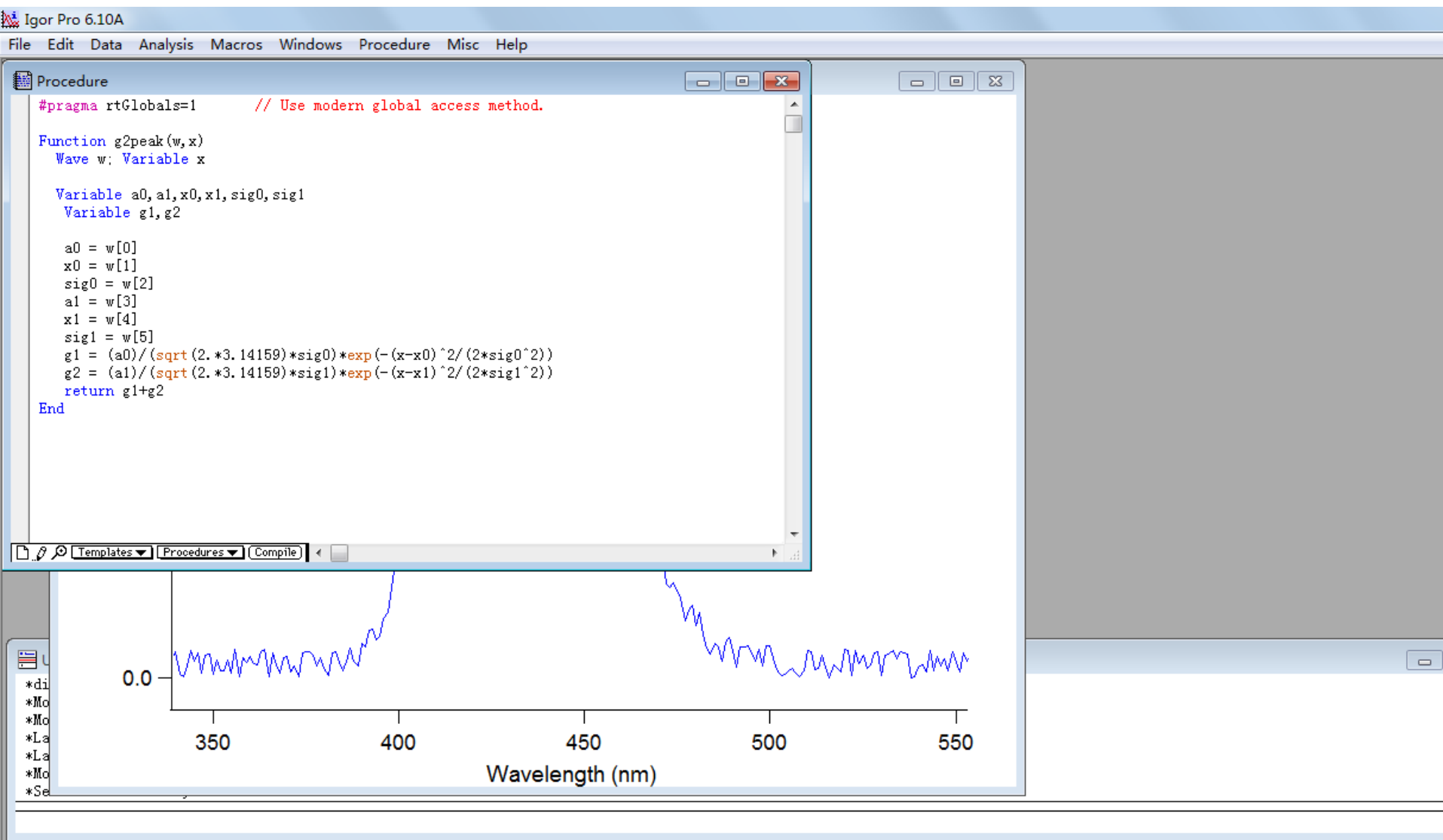
The objective function is quadratic with respect to the parameters only in a region close to its minimum value, where the truncated Taylor series is a good approximation to the model.

$$S \approx \sum_i W_{ii} \left(y_i - \sum_j J_{ij} \beta_j \right)^2$$

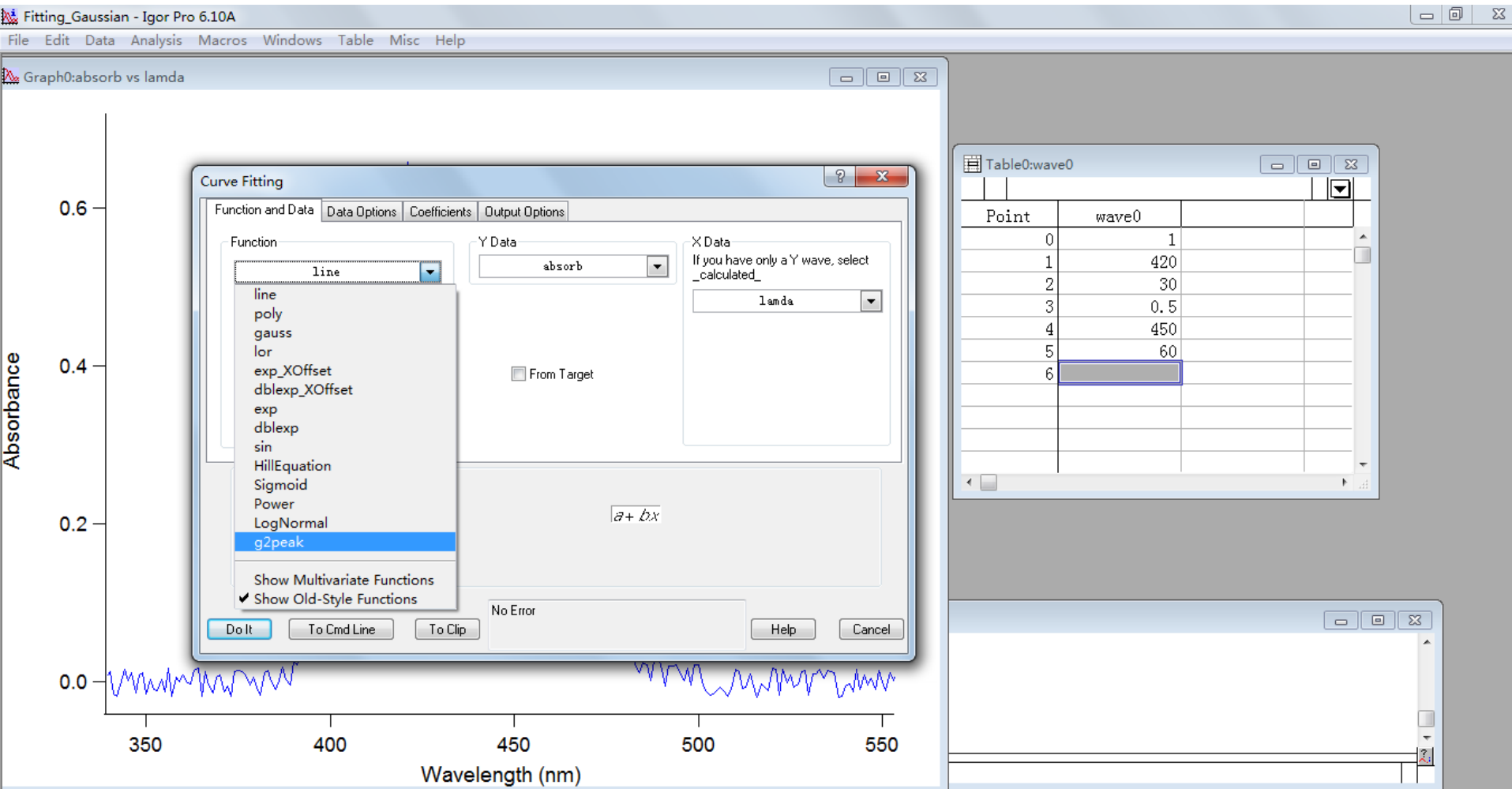
The more the parameter values differ from their optimal values, the more the contours deviate from elliptical shape. A consequence of this is that initial parameter estimates should be as close as practicable to their (unknown!) optimal values. It also explains how divergence can come about as the Gauss–Newton algorithm is convergent only when the objective function is approximately quadratic in the parameters.



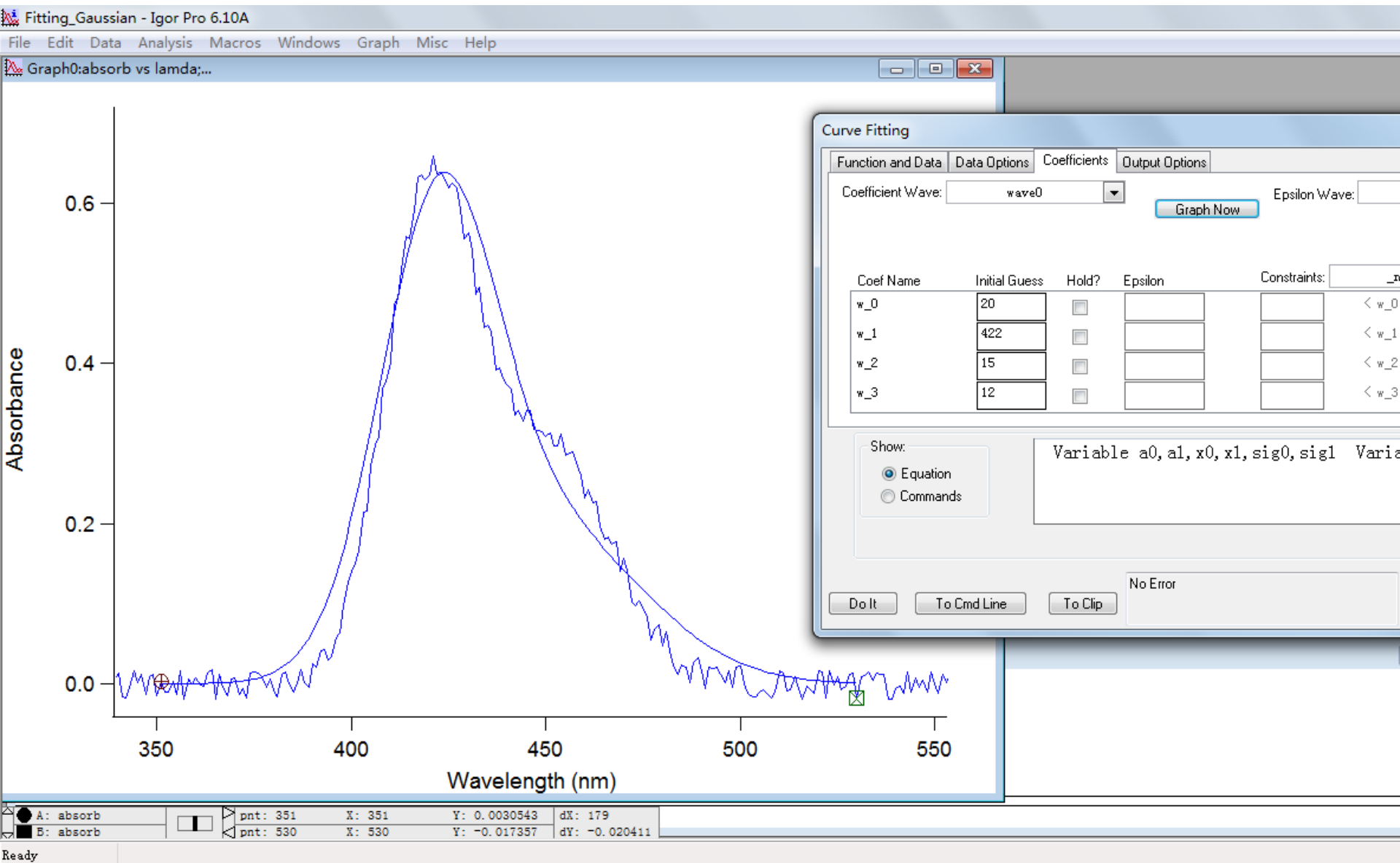
Read in the macro for two Gaussians



Select the macro in the Curve Fitting menu



Make an initial guess of fitting parameters



Do it

