Week 10

More Issues in Non-linear Fitting
Load \( \{x_i, y_i, \sigma_i\}_N \)

Select initial values for \( \{a_j\}_n \)
Set \( \chi^2_{\text{old}} = \text{huge} \)

Evaluate \( \chi^2_{\text{new}} \)

\[ \Delta \chi^2 < 0.01 \] ?

yes

Declare Victory! Compute \( \sigma_{a_j} \)’s

no

Set \( \chi^2_{\text{old}} = \chi^2_{\text{new}} \)

Compute \( \delta a_j \)’s

Update \( a_j \)’s using \( \delta a_j \)’s

By hand or by Monte Carlo

Grid Search
Gradient Search
Expansion Method
Marquardt’s Method
Gradient Method pay-off:

**FIGURE 8.4**
Contour plot of \( \chi^2 \) as a function of two highly correlated variables. The zigzag line represents the search path approach to a local minimum by the grid-search method.
Gradient Search  (Bevington 8.4; pp.153)

• Find the “steepest” descent down the valley of $\chi^2$ by computing the normalized gradient
  - Gradient always computable numerically
  - Sometimes can be computed algebraically

• Pro’s and Con’s:
  - Fast approach to the minimum from far away
  - Very slow near the minimum where the gradient is near zero
Summary from last time

- Linear vs. non-linear fitting functions
  - Linear functions can be fitted in one algebraic iteration
  - Non-linear function require iterated solutions when fitting to data
- Grid search is simple but tends to be slow
- Gradient search is fast when far away from minimum but slow near the minimum
- We need a method that is fast near the minimum...next time
Expansion Method (cf. Bevington 8.5; pp156)

Exploit the quadratic nature of $\chi^2$ near its minimum. Expand to 2nd order around whatever point in parameter space we are sitting, where $\chi^2 = \chi_0^2$:

$$\chi^2 \approx \chi_0^2 + \sum_{j=1}^{m} \left\{ \frac{\partial \chi_0^2}{\partial a_j} \delta a_j \right\} + \frac{1}{2} \sum_{k=1}^{m} \sum_{j=1}^{m} \left\{ \frac{\partial^2 \chi_0^2}{\partial a_j \partial a_k} \delta a_j \delta a_k \right\}$$ (Bev. Eq. 8.22)

Minimize $\chi^2$ with respect to the parameter increments $\delta a_j$, not the parameters themselves. Why? Because this linearizes the problem and it can be solved exactly.

$$\frac{\partial \chi^2}{\partial (\delta a_k)} = \frac{\partial \chi_0^2}{\partial a_k} + \sum_{j=1}^{m} \left\{ \frac{\partial^2 \chi_0^2}{\partial a_k \partial a_j} \delta a_j \right\} = 0 \quad k = 1, m$$

This is a set of $m$ linear equations in $m$ unknowns!

$$\beta_k \equiv -\frac{1}{2} \frac{\partial \chi_0^2}{\partial a_k}$$

$$\alpha_{jk} \equiv \frac{1}{2} \frac{\partial^2 \chi_0^2}{\partial a_j \partial a_k}$$

“Curvature Matrix” $\alpha$ measures curvature of $\chi^2$ hypersurface
Expansion Method

Matrix equation to solve:

\[ \vec{\beta} = \delta \vec{a} \vec{\alpha} \]

Invert the matrix \( \alpha \) to get the matrix \( \varepsilon \), the so-called “error matrix” \( \varepsilon = \alpha^{-1} \):

\[ \delta \vec{a} = \vec{\beta} \varepsilon \]

\[ \delta a_k = \sum_{j=1}^{m} \varepsilon_{kj} \beta_j \]

How to compute the matrix \( \alpha \) and the vector \( \beta \)?

\[ \beta_k \equiv \sum \frac{1}{\sigma_i^2} \left[ y_i - y_{fit}(x_i) \right] \frac{\partial y_{fit}(x_i)}{\partial a_k} = -\frac{1}{2} \frac{\partial \chi_0^2}{\partial a_k} \]

\[ \alpha_{jk} \equiv \sum \frac{1}{\sigma_i^2} \left\{ \frac{\partial y_{fit}(x_i)}{\partial a_j} \frac{\partial y_{fit}(x_i)}{\partial a_k} - \left[ y_i - y_{fit}(x_i) \right] \frac{\partial^2 y_{fit}(x_i)}{\partial a_j \partial a_k} \right\} = \frac{1}{2} \frac{\partial^2 \chi_0^2}{\partial a_j \partial a_k} \]

- Small compared to first term; vanishes for linear functions; neglect this term
- Product of first derivatives
The “error matrix”

\[ \alpha^{-1} = \varepsilon = \begin{pmatrix}
\sigma^2_{11} & \sigma^2_{12} & \sigma^2_{13} & \cdots \\
\sigma^2_{21} & \sigma^2_{22} & \cdots \\
\sigma^2_{31} & \cdots & \sigma^2_{33} \\
\vdots & \ddots & \ddots
\end{pmatrix} \]

\[ \sigma_{jj} = "\text{diagonal errors" } \equiv \sigma_j \]
\[ \sigma_{jk} = "\text{correlated errors" } ; \sigma^2_{jk} = "\text{covariances" } \]

Example...
Marquardt (Levenberg) Method
(cf. Bevington 8.6; pp161)

\[ \beta = \delta a \alpha' \quad \text{with} \quad \alpha'_{jk} = \begin{cases} \alpha_{jk}(1 + \lambda) & \text{for} \ j = k \\ \alpha_{jk} & \text{for} \ j \neq k \end{cases} \]

- For small \( \lambda \) it behaves like the expansion method.
- For large \( \lambda \), fit behaves like the gradient method with small steps, since off-diagonals are negligible.
- Make \( \lambda \) just large enough such that \( \chi^2 \) decreases.

The "recipe" for fast fit convergence:

1. Compute \( \chi^2(a) \).
2. Start initially with \( \lambda = 0.001 \).
3. Compute \( \delta a \) and \( \chi^2(a + \delta a) \) with this choice of \( \lambda \).
4. If \( \chi^2(a + \delta a) > \chi^2(a) \), increase \( \lambda \) by a factor of 10 and repeat step 3.
5. If \( \chi^2(a + \delta a) < \chi^2(a) \), decrease \( \lambda \) by a factor of 10, consider \( a' = a + \delta a \) to be the new starting point, and return to step 3, substituting \( a' \) for \( a \).
Marquardt Method Fit

red = “true” parent
blue = initial guess
Numbers...

$$N(t) = N_0 e^{-t/\tau} + B$$

<table>
<thead>
<tr>
<th>Trial</th>
<th>$\chi^2$/d.o.f.</th>
<th>$N_0$</th>
<th>$\tau$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.364</td>
<td>150.0</td>
<td>20.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>8.030</td>
<td>69.9</td>
<td>6.4</td>
<td>47.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.4</td>
<td>1.8</td>
<td>4.3</td>
</tr>
<tr>
<td>2</td>
<td>0.841</td>
<td>100.6</td>
<td>8.1</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.0</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>0.796</td>
<td>103.4</td>
<td>7.2</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.6</td>
<td>0.8</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>0.795</td>
<td>102.9</td>
<td>7.4</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.2</td>
<td>0.8</td>
<td>1.2</td>
</tr>
</tbody>
</table>
Marquardt Method Fit

Iteration 1:
- **red** = “true” parent
- **blue** = fit result

Time (arbitrary units)

Counts
**Numbers...**

\[
N(t) = N_0 e^{-t/\tau} + B
\]

<table>
<thead>
<tr>
<th>Trial</th>
<th>(\chi^2/d.o.f.)</th>
<th>(N_0) a_1</th>
<th>(\tau) a_2</th>
<th>(B) a_3</th>
<th>Value</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.364</td>
<td>150.0</td>
<td>20.0</td>
<td>0.0</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8.030</td>
<td>69.9</td>
<td>6.4</td>
<td>47.5</td>
<td>Value</td>
<td>4.3</td>
</tr>
<tr>
<td>2</td>
<td>0.841</td>
<td>100.6</td>
<td>8.1</td>
<td>29.1</td>
<td>Value</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>0.796</td>
<td>103.4</td>
<td>7.2</td>
<td>29.3</td>
<td>Value</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>0.795</td>
<td>102.9</td>
<td>7.4</td>
<td>29.1</td>
<td>Value</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Carnegie Mellon University Physics
Marquardt Method Fit

red = “true” parent
blue = fit result

Time (arbitrary units)
Numbers...

\[ N(t) = N_0 e^{-t/\tau} + B \]

<table>
<thead>
<tr>
<th>Trial</th>
<th>(\chi^2/d.o.f)</th>
<th>(N_0)</th>
<th>(\tau)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.364</td>
<td>150.0</td>
<td>20.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Value</td>
</tr>
<tr>
<td>1</td>
<td>8.030</td>
<td>69.9</td>
<td>6.4</td>
<td>47.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.4</td>
<td>1.8</td>
<td>4.3</td>
</tr>
<tr>
<td>2</td>
<td>0.841</td>
<td>100.6</td>
<td>8.1</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.0</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>0.796</td>
<td>103.4</td>
<td>7.2</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.6</td>
<td>0.8</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>0.795</td>
<td>102.9</td>
<td>7.4</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.2</td>
<td>0.8</td>
<td>1.2</td>
</tr>
</tbody>
</table>
Marquardt Method Fit

Iteration 3

red = “true” parent
blue = fit result
Numbers...

\[ N(t) = N_0 e^{-t/\tau} + B \]

<table>
<thead>
<tr>
<th>Trial</th>
<th>( \chi^2/d.o.f. )</th>
<th>( N_0 )</th>
<th>( \tau )</th>
<th>( B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.364</td>
<td>150.0</td>
<td>20.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8.030</td>
<td>69.9</td>
<td>6.4</td>
<td>47.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.4</td>
<td>1.8</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td>Uncertainty</td>
</tr>
<tr>
<td>2</td>
<td>0.841</td>
<td>100.6</td>
<td>8.1</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.0</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td>Uncertainty</td>
</tr>
<tr>
<td>3</td>
<td>0.796</td>
<td>103.4</td>
<td>7.2</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.6</td>
<td>0.8</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td>Uncertainty</td>
</tr>
<tr>
<td>4</td>
<td>0.795</td>
<td>102.9</td>
<td>7.4</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.2</td>
<td>0.8</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td></td>
<td>Uncertainty</td>
</tr>
</tbody>
</table>
Marquardt Method Fit

Iteration 4

red = “true” parent
blue = fit result

Counts vs. Time (arbitrary units)
## Numbers...

\[ N(t) = N_0 e^{-t/\tau} + B \]

<table>
<thead>
<tr>
<th>Trial</th>
<th>( \chi^2/d.o.f. )</th>
<th>( N_0 )</th>
<th>( \tau )</th>
<th>( B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.364</td>
<td>150.0</td>
<td>20.0</td>
<td>0.0 \ Value</td>
</tr>
<tr>
<td>1</td>
<td>8.030</td>
<td>69.9</td>
<td>6.4</td>
<td>47.5 \ Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.4</td>
<td>1.8</td>
</tr>
<tr>
<td>2</td>
<td>0.841</td>
<td>100.6</td>
<td>8.1</td>
<td>29.1 \ Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.796</td>
<td>103.4</td>
<td>7.2</td>
<td>29.3 \ Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.6</td>
<td>0.8</td>
</tr>
<tr>
<td>4</td>
<td>0.795</td>
<td>102.9</td>
<td>7.4</td>
<td>29.1 \ Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

\[ 95 \quad 8 \quad 30 \quad \leftarrow \text{The “real” parent numbers} \]
Covariance Among Parameters

**Covariance Matrix**

\[
\begin{pmatrix}
N_0 & \tau & B \\
a_1 & a_2 & a_3 \\
--- & --- & --- \\
N_0 & 67.64 & -4.35 & 1.84 \\
\tau & -4.35 & 0.58 & -0.59 \\
B & 1.84 & -0.59 & 1.55
\end{pmatrix}
\]

\[
\alpha^{-1} = \varepsilon = \begin{pmatrix}
\sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 & \ldots \\
\sigma_{21}^2 & \sigma_{22}^2 & \ldots \\
\sigma_{31}^2 & \ldots & \sigma_{33}^2 \\
\vdots & \ddots & \ddots
\end{pmatrix}
\]

\[
\sigma_{N_0} = \sqrt{67.64} = 8.2, \text{ etc...}
\]

The off-diagonal terms specify the covariances or “correlations” among fit parameters.
Covariance Among Parameters

"Reduced" Covariance Matrix

<table>
<thead>
<tr>
<th></th>
<th>$N_0$</th>
<th>$\tau$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$-0.112$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$0.018$</td>
<td>$-0.664$</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\mathcal{E}_{ij}' &= \frac{\mathcal{E}_{ij}}{\mathcal{E}_{ii} \mathcal{E}_{jj}}, \quad i \neq j \\
\end{align*}
\]

We see that the lifetime and background rate are strongly correlated.
Summary

- Expansion method uses quadratic approx. of the $\chi^2$ function near the minimum to zero in on the minimum.
- Marquardt method combines the best features of the gradient and expansion methods to fit a function to data quickly.
- Covariance matrix of parameter errors is one outcome of the expansion method algebra.
Backup Slides…