Week 9

Methods for Non-linear Fitting
Modern Physics Lab Grading Standard

Each experiment gets 3 scores, weighted as follows:

1) Doing the experiment & learning the physics  50%
2) Scientific Journal Article  25%
3) Notebook  25%

For items (1) and (2) the scale is:

10  - everything done exceptionally well, carrying all aspects of the work beyond typical expectations, no errors of any significance
9  - everything done very well and at least one thing done beyond typical expectations; no major errors
8  - everything done well; a few small errors but no major errors
7  - everything done well but one thing done not so well (but not terribly)
6  - everything done but one thing done not so well; some other small errors allowed
5  - everything done with some number of errors
4  - some major element missing or incorrect
3  - two or three major elements missing or incorrect
For the Notebook the scale is:

10 - everything done exceptionally well, no errors of any kind
9  - everything done very well; no significant errors
8  - everything done; a few small errors but no major errors
7  - everything done but one thing done not so well (but not terribly)
6  - one major element missing or incorrect
5  - two major elements missing or incorrect
≤4 - three or more major elements missing or incorrect

“Errors” above do not refer to the standardized deductions from the check-off list you will use. These standardized deductions (0.5 per instance) come off your score before the grading above is applied.

Final letter grades will be assigned using the following nominal cut-offs:

9, 10 → “A”
7,  8 → “B”
5,  6 → “C”
3,  4 → “D”
GRADES

- For the whole course
  - 20% 1st experiment
  - 25% 2nd experiment
  - 30% 3rd experiment
  - 15% final oral presentation
  - 10% recitation participation: quizzes, short talks

- Late penalties for missed deadlines on article hand-ins **
  - 0.2 / weekday for 1st 5 days (on a scale of 10)
  - 0.5 / weekday for 2nd 5 days
  - >10 weekdays averaged as R in final grade
  ** Ask us early for extensions and you are likely to get a sympathetic hearing

- Cannot pass the course without:
  - being there for 3 full experiments
  - 3 lab books handed in
  - 3 articles completed
  - final “long” talk given

Carnegie Mellon University Physics
Why are my scores not the same as those of my partner?

- We treat you as an individual; scores developed mostly separately
  - Article: you may be a better or worse writer, your feedback is customized to your effort
  - Notebook: each of you must have a complete record of the work done
  - Experiment: it’s a team effort, but we are aware of who’s doing the most / best work
Week 9

Methods for Non-linear Fitting
Linear vs. Non-linear Fitting
(“Parent”) Functions

Suppose you have a set of data points \(\{x_i, y_i, \sigma_i\}\). The fitting functions contain fit parameters \(\{a_j\}\). Some are linear in the fit parameters and some are \textit{non-linear} in the parameters.

- \(y_{\text{fit}} = a_0 + a_1 x + a_2 x^2 + a_3 x^3\) \hspace{1cm} \text{Linear}
- \(y_{\text{fit}} = a_1 (x^3 - x) + a_2 \sin x + a_3 e^x\) \hspace{1cm} \text{Linear}
- \(y_{\text{fit}} = a_1 \sin(a_2 x + a_3) + a_4\) \hspace{1cm} \text{Non-Linear}
- \(y_{\text{fit}} = a_1 (x - a_2) + a_3 (x^2 - a_4)^3\) \hspace{1cm} \text{Non-Linear}
- \(y_{\text{fit}} = a_0 P_0(x) + a_1 P_1(x) + a_2 P_2(x) + a_3 P_3(x)\) \hspace{1cm} \text{Linear}

where the \(P_j\)'s are the Legendre polynomials.

\textbf{Linear} \rightarrow \text{solve for} \(\{a_i\}\) in \textit{one iteration} using least-squares fitting \(\text{à la Bevington Chapter 6}\)

\textbf{Non-linear} \rightarrow \text{solve for} \(\{\delta a_i\}\) using least-squares fitting \(\text{à la Bevington Chapter 8 and} \text{ iterate as needed}\)
mpl_DATAFIT for non-linear fits

Modern Physics Lab Data Fitting

Date Now: Time Now: Log Book Page: ________

<table>
<thead>
<tr>
<th>Value</th>
<th>Initial Step Size</th>
<th>Final Step Size</th>
<th>Estimated Uncertainty</th>
<th>Refined Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit Parameter 1=</td>
<td>-2.04</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit Parameter 2=</td>
<td>0.30</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit Parameter 3=</td>
<td>2.85</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit Parameter 4=</td>
<td>2.46</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit Parameter 5=</td>
<td>16.50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \chi^2 = 16.50 \]

\[ \chi^2 / \text{d.o.f.} = 1.03 \]

<table>
<thead>
<tr>
<th>Entries</th>
<th>Constraints</th>
<th>d.o.f.</th>
<th>c(\chi^2/dof)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>4</td>
<td>16</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Substitute your data (up to 1090 lines) below. Program "FIT Theory" column with your fitting function.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>DY</th>
<th>FIT</th>
<th>Deviation</th>
<th>Dev.^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Theory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
<td>2.20</td>
<td>0.20</td>
<td>2.48</td>
<td>1.40</td>
</tr>
<tr>
<td>2</td>
<td>1.50</td>
<td>3.00</td>
<td>0.20</td>
<td>2.78</td>
<td>-1.08</td>
</tr>
<tr>
<td>3</td>
<td>2.00</td>
<td>3.20</td>
<td>0.20</td>
<td>3.08</td>
<td>-0.50</td>
</tr>
<tr>
<td>4</td>
<td>3.00</td>
<td>3.50</td>
<td>0.20</td>
<td>3.62</td>
<td>0.62</td>
</tr>
<tr>
<td>5</td>
<td>4.00</td>
<td>4.20</td>
<td>0.20</td>
<td>4.07</td>
<td>-0.67</td>
</tr>
<tr>
<td>6</td>
<td>5.00</td>
<td>4.50</td>
<td>0.20</td>
<td>4.36</td>
<td>-0.68</td>
</tr>
<tr>
<td>7</td>
<td>6.00</td>
<td>4.20</td>
<td>0.20</td>
<td>4.49</td>
<td>1.46</td>
</tr>
<tr>
<td>8</td>
<td>7.00</td>
<td>4.60</td>
<td>0.20</td>
<td>4.44</td>
<td>-0.80</td>
</tr>
<tr>
<td>9</td>
<td>8.00</td>
<td>4.20</td>
<td>0.20</td>
<td>4.21</td>
<td>0.06</td>
</tr>
<tr>
<td>10</td>
<td>9.00</td>
<td>3.60</td>
<td>0.20</td>
<td>3.83</td>
<td>1.15</td>
</tr>
<tr>
<td>11</td>
<td>10.00</td>
<td>3.50</td>
<td>0.20</td>
<td>3.32</td>
<td>-0.88</td>
</tr>
<tr>
<td>12</td>
<td>11.00</td>
<td>2.90</td>
<td>0.20</td>
<td>2.74</td>
<td>-0.79</td>
</tr>
<tr>
<td>13</td>
<td>12.00</td>
<td>2.00</td>
<td>0.20</td>
<td>2.13</td>
<td>0.67</td>
</tr>
<tr>
<td>14</td>
<td>13.00</td>
<td>1.50</td>
<td>0.20</td>
<td>1.56</td>
<td>0.28</td>
</tr>
<tr>
<td>15</td>
<td>14.00</td>
<td>1.00</td>
<td>0.20</td>
<td>1.06</td>
<td>0.30</td>
</tr>
<tr>
<td>16</td>
<td>15.00</td>
<td>0.50</td>
<td>0.20</td>
<td>0.69</td>
<td>0.93</td>
</tr>
<tr>
<td>17</td>
<td>16.00</td>
<td>0.50</td>
<td>0.20</td>
<td>0.47</td>
<td>-1.15</td>
</tr>
<tr>
<td>18</td>
<td>17.00</td>
<td>0.50</td>
<td>0.20</td>
<td>0.43</td>
<td>-0.34</td>
</tr>
<tr>
<td>19</td>
<td>18.00</td>
<td>0.80</td>
<td>0.20</td>
<td>0.57</td>
<td>-1.13</td>
</tr>
<tr>
<td>20</td>
<td>19.00</td>
<td>0.60</td>
<td>0.20</td>
<td>0.88</td>
<td>1.41</td>
</tr>
</tbody>
</table>

Any function here...
Generic Fitting Algorithm

Load \( \{x_i, y_i, \sigma_i\}_N \)

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

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

\( \Delta \chi^2 < 0.01 \) ?

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

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

Grid Search
Gradient Search
Expansion Method
Marquardt's Method
Grid Search (Bevington 8.3; pp.151)

**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.
Grid Search (Bevington 8.3; pp.151)

• Treat each $a_j$ in turn
• Let $a_j \rightarrow a_j \pm \Delta a_j$

Must pre-select step size
Pick sign such that $\chi^2$ decreases

• Take repeated steps until $\chi^2$ increases... then...
• Reverse direction, cut step size in half, etc...
• Pro’s and Con’s:
  - Simple to program (mpl_DATAFIT)
  - Can be slow if parameters are highly correlated
  - User must pick starting step sizes intelligently
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
Gradient Search  (Bevington 8.4; pp.153)

\[
\nabla \chi^2 = \sum_{j=1}^{n} \left[ \frac{\partial \chi^2}{\partial a_j} \hat{a}_j \right]
\]

Estimate the partial derivatives numerically:

\[
\left( \nabla \chi^2 \right)_j = \frac{\partial \chi^2}{\partial a_j} \approx \frac{\chi^2(a_j + f \Delta a_j) - \chi^2(a_j)}{f \Delta a_j}
\]

(Bev.8.15)

Concoct a \text{dimensionless} gradient using:

\[
b_j = \frac{a_j}{\Delta a_j}
\]

(Bev.8.18)

\[
\frac{\partial \chi^2}{\partial b_j} \approx \frac{\chi^2(a_j + f \Delta a_j) - \chi^2(a_j)}{f \Delta a_j}
\]

\[
\Delta a_j = \frac{\chi^2(a_j + f \Delta a_j) - \chi^2(a_j)}{f}
\]

The gradient vector: \[
\gamma_j = \frac{\partial \chi^2 / \partial b_j}{\sqrt{\sum_{j=1}^{m} (\partial \chi^2 / \partial b_j)^2}}
\]

The parameter increment:

\[
\delta a_j = \gamma_j \Delta a_j
\]
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

- 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