6.2 Grid Search of Chi-Square Space

• example data from a Gaussian-shaped peak are given and plotted
• initial coefficient guesses are made
• the basic grid search strategy is outlined
• an actual manual search is performed
• chi-square space is examined around the minimum
• an automatic Mathcad program is used
• one difficulty with a grid search is described
Non-Linear Equation

This is the same example used in Section 6.1. Find the least-squares coefficients that minimize chi-square for the Gaussian equation.

\[ y = \frac{1}{a_0 \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-a_1)^2}{a_0^2}} \]

The following data were collected across the peak:

(45,0.001) (46,0.010) (47,0.037) (48,0.075) (49,0.120) (50,0.178) (51,0.184) (52,0.160) (53,0.126) (54,0.064) (55,0.034)
I used familiarity with a Gaussian for the initial guesses, $g_1^0$ and $g_1^1$. I set $g_1^1$ equal to the mode, 51, and $g_1^0$ equal to one fourth the peak baseline width, $11/4 = 2.75$. The fit is shown at the left.

It seems that $a_0$ is too large, so I tried 2.00. The right graph shows this change. This is good enough to start, so $g_1^0 = 2.00$ and $g_1^1 = 51.0$. 
Start with the first guesses, $g_1^0$ and $g_1^1$. Compute chi-square.
- Change $g_1^0$ by an amount equal to the desired precision, say in this case 0.01, and compute a new chi-square. If you don't know which direction to go, try both $g_1^0+0.01$ and $g_1^0-0.01$.
- Continue in the same direction along $a_0$ until the value of chi-square no longer decreases.
- The value producing the lowest chi-square is your second guess, $g_2^0$.

Now switch to the other coefficient.
- Change $g_1^1$ by an amount equal to the desired precision, say in this case 0.1. Again, you might try both $g_1^1+0.1$ and $g_1^1-0.1$.
- Continue in the same direction until the value of chi-square no longer decreases.
- The value producing the lowest chi-square is your second guess, $g_2^1$.

Repeat the process with $g_2^0$ and $g_2^1$, etc. Iterate until chi-square no longer changes.
Varying $g_0$ while $g_1 = 51.0$

<table>
<thead>
<tr>
<th>$g_0$</th>
<th>$g_1$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
<td>51.0</td>
<td>0.000784</td>
</tr>
<tr>
<td>2.10</td>
<td>51.0</td>
<td>0.000333</td>
</tr>
<tr>
<td>2.20</td>
<td>51.0</td>
<td>0.000350</td>
</tr>
<tr>
<td>2.15</td>
<td>51.0</td>
<td>0.000290</td>
</tr>
<tr>
<td>2.16</td>
<td>51.0</td>
<td>0.000294</td>
</tr>
<tr>
<td>2.14</td>
<td>51.0</td>
<td>0.000290</td>
</tr>
</tbody>
</table>

Looking at the fit on the last slide, I decided to try increasing $a_0$. Increasing to 2.10 lowered chi-square. Moving in the same direction, 2.20 gave a higher chi-square so the minimum is in between the two values. I split the difference using 2.15. Further investigation showed that either 2.14 or 2.15 could be used.
Varying $g_1$ while $g_0 = 2.14$

<table>
<thead>
<tr>
<th>$g_0$</th>
<th>$g_1$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.14</td>
<td>51.0</td>
<td>0.000290</td>
</tr>
<tr>
<td>2.14</td>
<td>52.0</td>
<td>0.015281</td>
</tr>
<tr>
<td>2.14</td>
<td>50.0</td>
<td>0.012648</td>
</tr>
<tr>
<td>2.14</td>
<td>50.5</td>
<td>0.003017</td>
</tr>
<tr>
<td>2.14</td>
<td>50.6</td>
<td>0.001906</td>
</tr>
<tr>
<td>2.14</td>
<td>50.7</td>
<td>0.001077</td>
</tr>
<tr>
<td>2.14</td>
<td>50.8</td>
<td>0.000530</td>
</tr>
<tr>
<td>2.14</td>
<td>50.9</td>
<td>0.000268</td>
</tr>
</tbody>
</table>

Moving along the $a_1$ axis proved much more laborious, because the second guess was too far from the first. Continuation of the grid search along $g_0$, showed that 2.15 and 2.13 gave higher $\chi^2$ when $g_1 = 50.9$. 
A contour graph was constructed showing both the starting point and the region around the minimum.

The contour graph also shows the route taken while using a manual grid search to locate the minimum.

Chi-square space is smooth with no apparent false minima.

The minimum $\chi^2$ at the resolution used for graph ($\Delta a_0 = 0.001$, $\Delta a_1 = 0.01$) occurs at $a_0 = 2.143$ and $a_1 = 50.94$).

$\chi_{\text{min}}^2 = 0.000243$
A program that automatically computes the minimum in chi-square space is shown in the Mathcad worksheet, "6.2 Grid Search Program.mcd".

It uses five functions with \( x, y \) and \( a \) as vector inputs, and \( res \) as a scalar input:

- \( f(x,a) \), inputs are the \( x \)-data and the initial coefficient guesses, \( a \); output is the corresponding \( y \)-value as a scalar.
- \( chisqr(y,x,a) \), inputs are the \( x \)- and \( y \)-data, and the \( a \)-coefficients; output is chi-square at the location given by \( a \).
- \( dir(y,x,a,res) \), inputs are the data, coefficients, and resolution along all coefficient axes, \( res \); output is a vector containing the coefficient and direction yielding the largest drop in chi-square.
- \( move(y,x,a,res) \), moves along a coefficient axis until a minimum in chi-square is found; output is the \( a \)-vector at the minimum.
- \( grid(y,x,a,res) \), performs a grid search using the initial guesses given in the \( a \)-vector at the specified resolution; output is a vector containing the coefficient values at the minimum.
Start with the initial guesses and a resolution about ten times finer than the resolution of the guesses.

\[
\text{grid} \left[ y, x, \begin{pmatrix} 2 \\ 51 \end{pmatrix}, 0.01 \right] = \begin{pmatrix} 2.14 \\ 50.94 \end{pmatrix} \quad \text{chisqr} \left[ y, x, \begin{pmatrix} 2.14 \\ 50.94 \end{pmatrix} \right] = 0.0002460
\]

This result matches the manual search. If higher resolution is desired, use the above output as the initial guesses.

\[
\text{grid} \left[ y, x, \begin{pmatrix} 2.14 \\ 50.94 \end{pmatrix}, 0.001 \right] = \begin{pmatrix} 2.14100 \\ 50.93800 \end{pmatrix} \quad \text{chisqr} \left[ y, x, \begin{pmatrix} 2.14100 \\ 50.93800 \end{pmatrix} \right] = 0.0002459
\]

Quit when the number of significant figures in chi-square exceed the number of significant figures in the maximum amplitude. To estimate the standard deviation of the fit, divide \( \chi^2 \) by \( N-1 \) and take the square root, e.g. 0.005. Thus, 0.184 has two significant figures, so stop when \( \chi^2 \) is stable at three significant figures.
Verifying the Grid Search Result

When using any method of searching chi-square space it is important that you attempt to verify that your solution is valid.

- examine a contour plot of chi-square space to see if you are at the true minimum
- compare the regression line to the data
- relocate the minimum using different starting guesses for the coefficients

To obtain different starting guesses, use the first set of guesses and the final coefficients to obtain three more sets of guesses

\[ \Delta a_0 = 2.00 \rightarrow 2.14 = +0.14 \]
\[ g_0 = 2.14 + 0.14 = 2.28 \]
\[ \Delta a_1 = 51.00 \rightarrow 50.94 = -0.06 \]
\[ g_1 = 50.94 - 0.06 = 50.88 \]

\[
\begin{align*}
g_0 &= \begin{bmatrix} 2 \ 50.88 \end{bmatrix}, 0.001 \\
g_1 &= \begin{bmatrix} 2.28 \\ 50.88 \end{bmatrix}, 0.001 \\
g_2 &= \begin{bmatrix} 2.28 \\ 51 \end{bmatrix}, 0.001 \\
g_3 &= \begin{bmatrix} 2.14100 \\ 50.93700 \end{bmatrix}
\end{align*}
\]
When two equation coefficients have a covariance, chi-square space becomes tilted at an angle to the grids.

The grid search algorithm has trouble with angled chi-square space because it oscillates back and forth between coefficient axes.

The straight line example shown took 23 changes in axis to locate the minimum. Doing this by hand is tiring!