How Do I Find the Minimum (or Maximum) Of A Function?

For a function of one variable you normally:
- Take the first derivative of a function,
- Set that equal to zero, and
- Solve for the variable!

Then
- Find the second derivative
- Evaluate that derivative (positive for a min, negative for a max!)

MATLAB can’t do these things!
Max/Min of a 3-D Function $f(x, y)$.

To test for a 3-D function, you

- Find $f_x(x, y) = 0$.
- Find $f_y(x, y) = 0$.
- Then check for the positivity of

$$D(x, y) = f_{xx}(x, y)f_{yy}(x, y) - [f_{xy}(x, y)]^2.$$ 

We don’t really even talk about higher than three dimensions.
If you type `ver`, you’ll see a list of the installed toolboxes for MATLAB. One of those should be “Optimization”.

- “Optimization” is another word for finding the max or min of a function.
- We want to find the “optimal” value of the function.
- MATLAB includes multiple functions for both unconstrained and constrained minimization!
Looks to minimize a function!

How does it do this?

2-D Nelder Meade

Basically, it looks around like you would!
Example: \( f(x) = x^2 \).

```matlab
function out=fcn1(x)
  out=(x-100)^2;
end
```

A help `fminsearch` tells us that

```matlab
[xf]=fminsearch(@fcn1,x0);
```

- `xf`-final best iterate
- `@fcn1`-function handle for the desired function
- `x0`-initial iterate

Set \( x0=10 \);
Notice (in help fminsearch) the input for fminsearch labeled “options”.

\[ X = \text{FMINSEARCH}(\text{FUN}, X0, \text{OPTIONS}) \] minimizes with the default optimization parameters replaced by values in the structure OPTIONS, created with the OPTIMSET function. See OPTIMSET for details. FMINSEARCH uses these options: Display, TolX, TolFun, MaxFunEvals, MaxIter, FunValCheck, PlotFcns, and OutputFcn.

Now we do a help on “optimset”.
Doing a `help optimset` reveals

```
OPTIONS = OPTIMSET('PARAM1',VALUE1,'PARAM2',VALUE2,
                    'PARAM3',VALUE3,...)
```

where ‘PARAM1’ is the tag of the value we’d like to change and VALUE1 is the value to which we’re setting that parameter, etc..

The three most useful types of parameters to alter are the ‘MAXs’, the ’Display’, and the ’Tol’ tags.

For instance,

```
options=optimset('MaxIter',1000,'TolX',1e-10,'Display','iter');
```

sets the value of maximum number of iterations to be 1000, the tolerance on the parameter we’re finding to be $1 \times 10^{-10}$, and to display the iteration information.

This has to be passed as

```
xf=fminsearch(@fcn1,x0,options).
```
Now, what about functions like

\[ f(x, y) = x^2 + y^2? \]

An Expansion to the idea is simple!

3-D Nelder_Meade2

Anything higher than 3-D is hard to visualize, but the algorithmic expansion is quite clear.
Code up \( f(x, y) = x^2 + y^2 \).

```matlab
function output=fcn2(x)
    %x is a 2x1 vector
    output=x(1)^2+x(2)^2;
```

Consequently we make the following changes to our script:

```matlab
x0=[20,50];
options=optimset('MaxIter',1000,...
    'TolX',1e-10,'Display','Iter');
xf=fminsearch(@fcn2,x0,options);
```

Alter the functional to be

\[
 f(x, y) = (x - 100)^2 + (y + 3.76)^2
\]

What are your results?
Code up \( f(x, y) = x^2 + y^2 + w^2 + z^2 \).

\[
\text{function output=fcn3(x)} \\
\text{\%x is a 4x1 vector} \\
\text{output=x(1)^2+x(2)^2+x(3)^2+x(4)^2;}
\]

Consequently we make the following changes to our script:

\[
\text{x0=[20,50,3000,103];} \\
\text{options=optimset('MaxIter',1000,...} \\
\text{\quad 'TolX',1e-10,'Display','Iter');} \\
\text{xf=fminsearch(@fcn3,x0,options);} \\
\]

Alter the functional to be

\[
f(x, y) = (x - 100)^2 + (y + 3.76)^2 + (w - \pi)^2 + (z + e^1)^2
\]

What are your results?
• \textit{fminunc}-Newton-step based. This is infinitely better than \textit{fminsearch} for many applications.

• Code up \( f(x, y) = x^2 + y^2 + w^2 + z^2 \).

\begin{verbatim}
function output=fcn3(x)
    %x is a 4x1 vector
    output=x(1)^2+x(2)^2+x(3)^2+x(4)^2;
\end{verbatim}

• Consequently we make the following changes to our script:

\begin{verbatim}
x0=[20,50,3000,103];
options=optimset('MaxIter',1000,...
        'TolX',1e-10,'Display','Iter');
xf=fminsearch(@fcn3,x0,options);
\end{verbatim}
Can I Fit A Model To Data This Way?

- **Short Answer:** Yes (well, probably)!
- **How?** You have to compare the results in a manner which makes the optimization routine see that the model is getting closer to the data.
- **We call this** parameter estimation. This is a huge and very modern topic.
- **We estimate the parameter** $\mathbf{q}$ (multi-dimensional containing all the values to be estimated) in the form

  $$\min_{\mathbf{q}} \| f(\mathbf{q}) - data \|^2$$
Download the following files

- `weather_data.mat`
- `fminunc_ex2.m`
- `obj_fcn1.m`
Something of form

\[ f(t) = A \sin(w(t - \phi)) + B \]

where

- \( A \) is amplitude
- \( w \) is frequency
- \( \phi \) is phase
- \( B \) is vertical shift.
We must choose our initial estimate:

\[ q_0 = [A_0, w_0, \phi_0, B_0] \]

How might I choose some of these values?

Objective functional:

```matlab
function output = obj_fcn1(q)

a = q(1); b = q(2);
c = q(3); d = q(4);

load weather_data;
time = x;
y = a * sin(b * (time - c)) + d;
output = 0.5 * norm(y - five_day_temp)^2;
```
Results

- Ran with \texttt{fminunc} and got back optimized values of
  \[ A_f = 14.7332 \]
  \[ w_f = 6.2818 \]
  \[ \phi_f = 2.4981 \]
  \[ B_f = 84.8282 \]

- Data was created by adding random Gaussian noise to function output of
  \[ f(x) = 15 \sin \left( 2\pi \left( x - \frac{1}{2} \right) \right) + 85 \]
Results, cont.

Model
Data

Temperature (°F)

Time t (in days)

Jon M. Ernstberger
MATLAB for the Sciences
Thoughts

- What errors might creep in?
- Are these results perfect?
- How does this compare to `fminsearch`?
- Why did `fminunc` move the $\phi$ parameter to approximately 2.5?
- Can I do more complex things inside the objective functional?