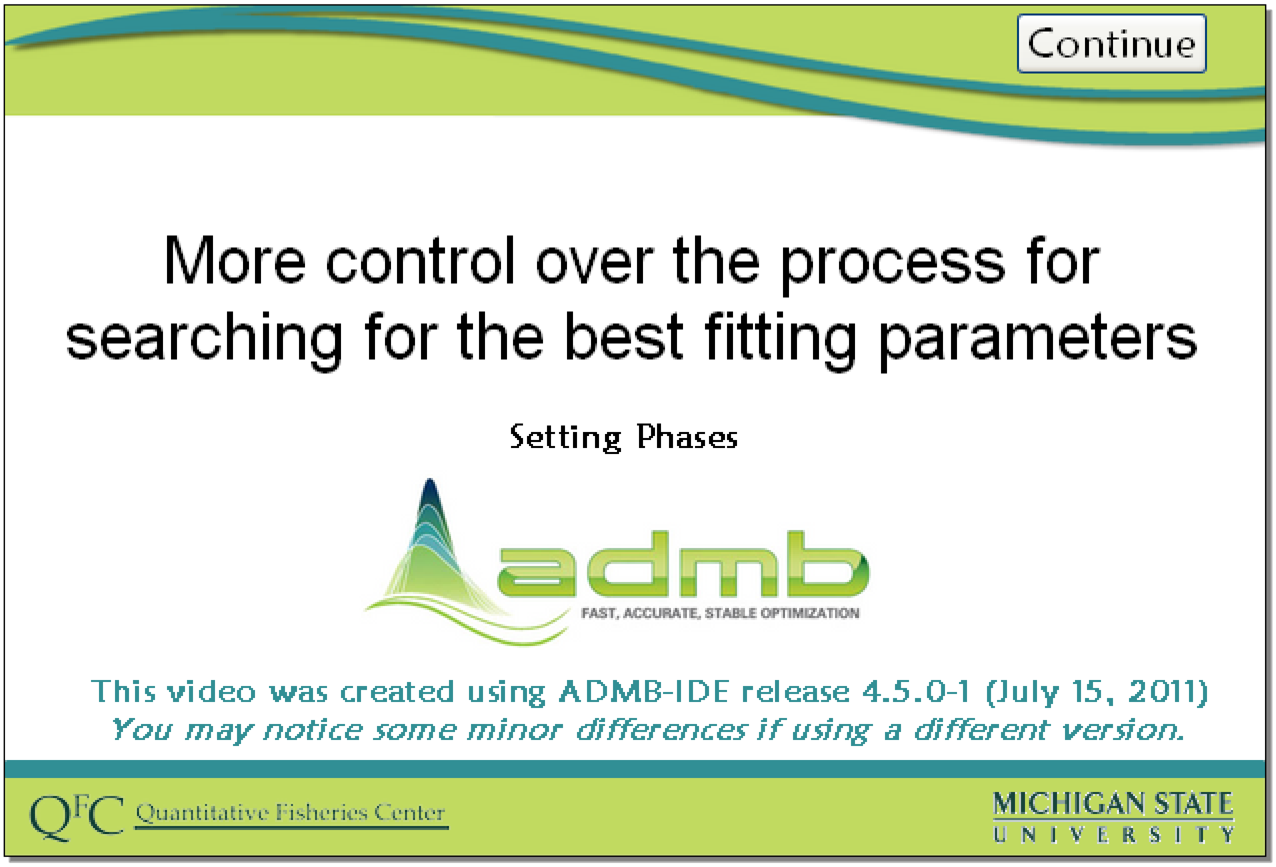
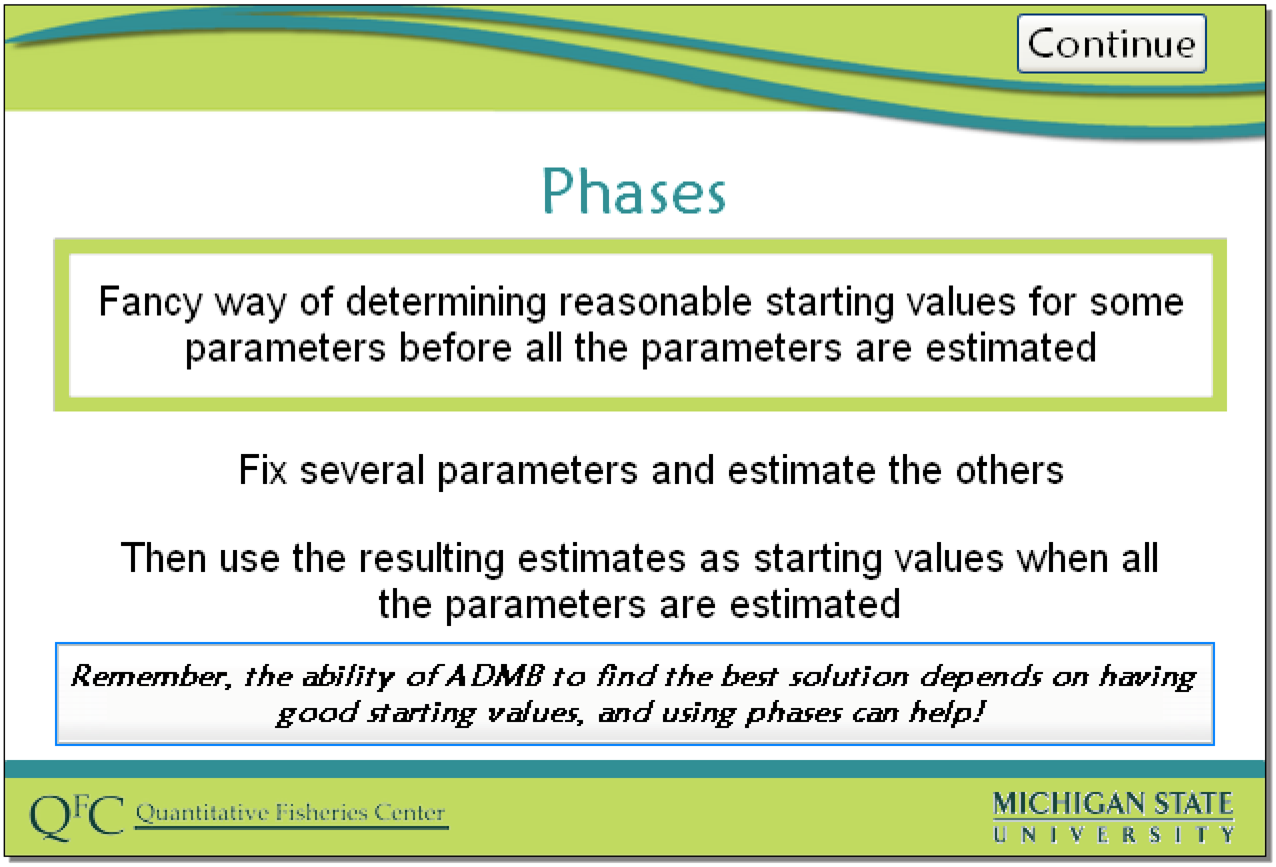

Continue
admb
FAST, ACCURATE, STABLE OPTIMIZATION
QFC Quantitative Fisheries Center
MICHIGAN STATE UNIVERSITY

More control over the process for searching for the best fitting parameters

Setting Phases

This video was created using ADMB-IDE release 4.5.0-1 (July 15, 2011)
You may notice some minor differences if using a different version.

In this video you will continue to learn some ways to better control the process of searching for the best fitting parameters used by AD Model Builder. In this video you will learn about estimation in phases. When you estimate parameters in phases, at first some parameters are fixed at their starting while others are adjusted, and sequentially in latter phases more and more of the parameters are estimated. This approach can sometimes help when estimation of a parameter is sensitive to the values of other parameters, so the search for them works better if the other parameters are closer to where they should be.



Phases

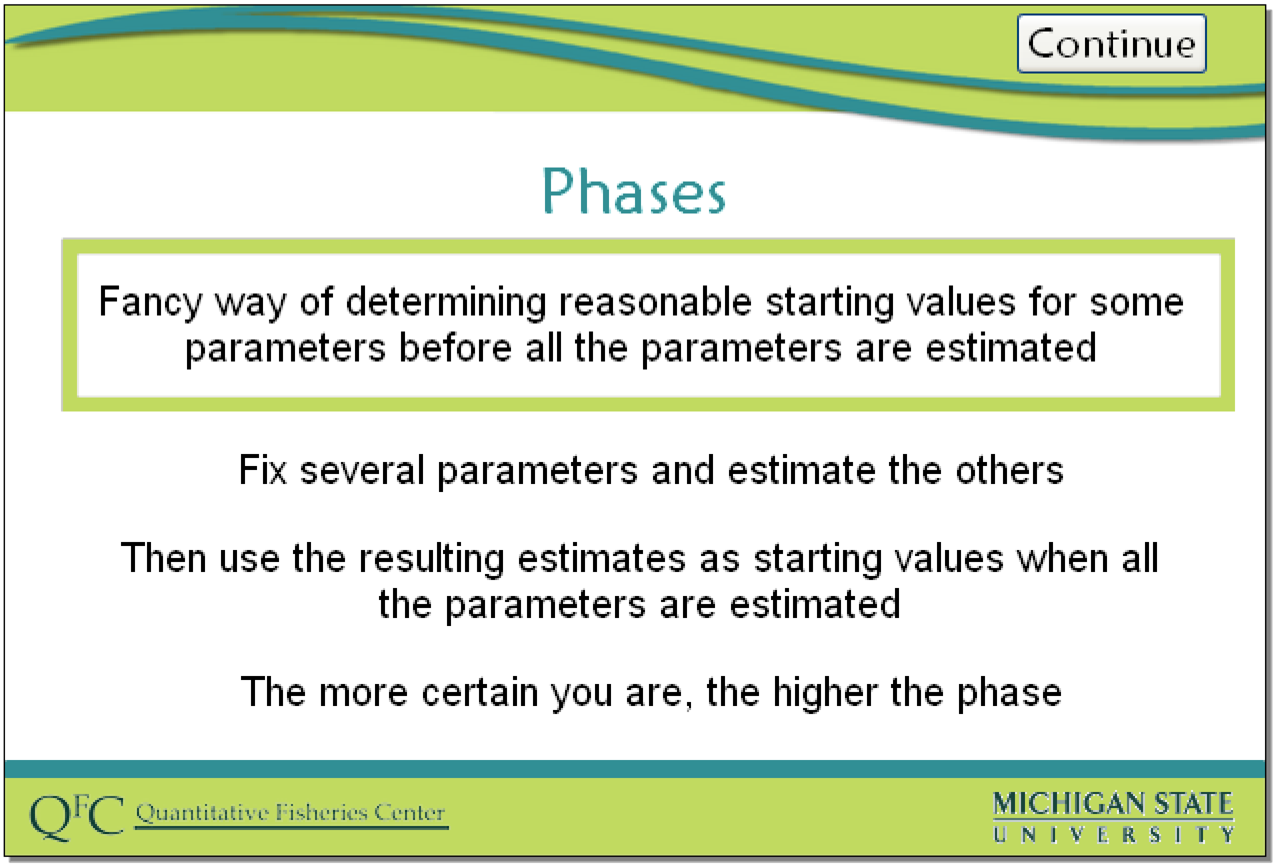
Fancy way of determining reasonable starting values for some parameters before all the parameters are estimated

Fix several parameters and estimate the others

Then use the resulting estimates as starting values when all the parameters are estimated

Remember, the ability of ADMB to find the best solution depends on having good starting values, and using phases can help!

Estimating parameters in phases is really a fancy way of getting the starting parameters for the final phase when all the parameters are estimated simultaneously. There can be cases when one or more parameter is especially hard to estimate and it's hard to get starting values for them that are appropriate. One way to proceed in these cases is to fix their values, estimate all the other parameters when they are fixed, and then start from those values of the other parameters as you obtain estimates for all the parameters. This can be made fancier in that you can have more than two phases.



Phases

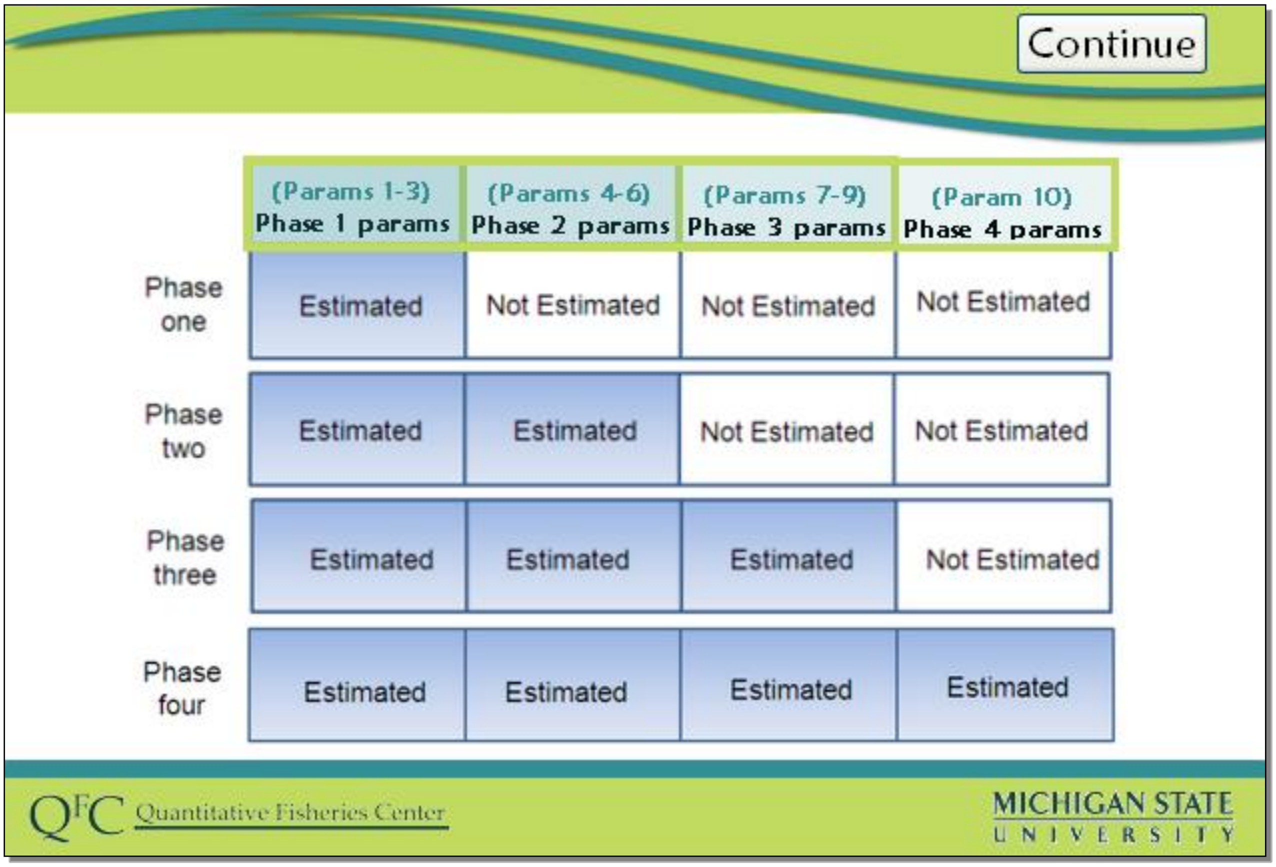
Fancy way of determining reasonable starting values for some parameters before all the parameters are estimated

Fix several parameters and estimate the others

Then use the resulting estimates as starting values when all the parameters are estimated

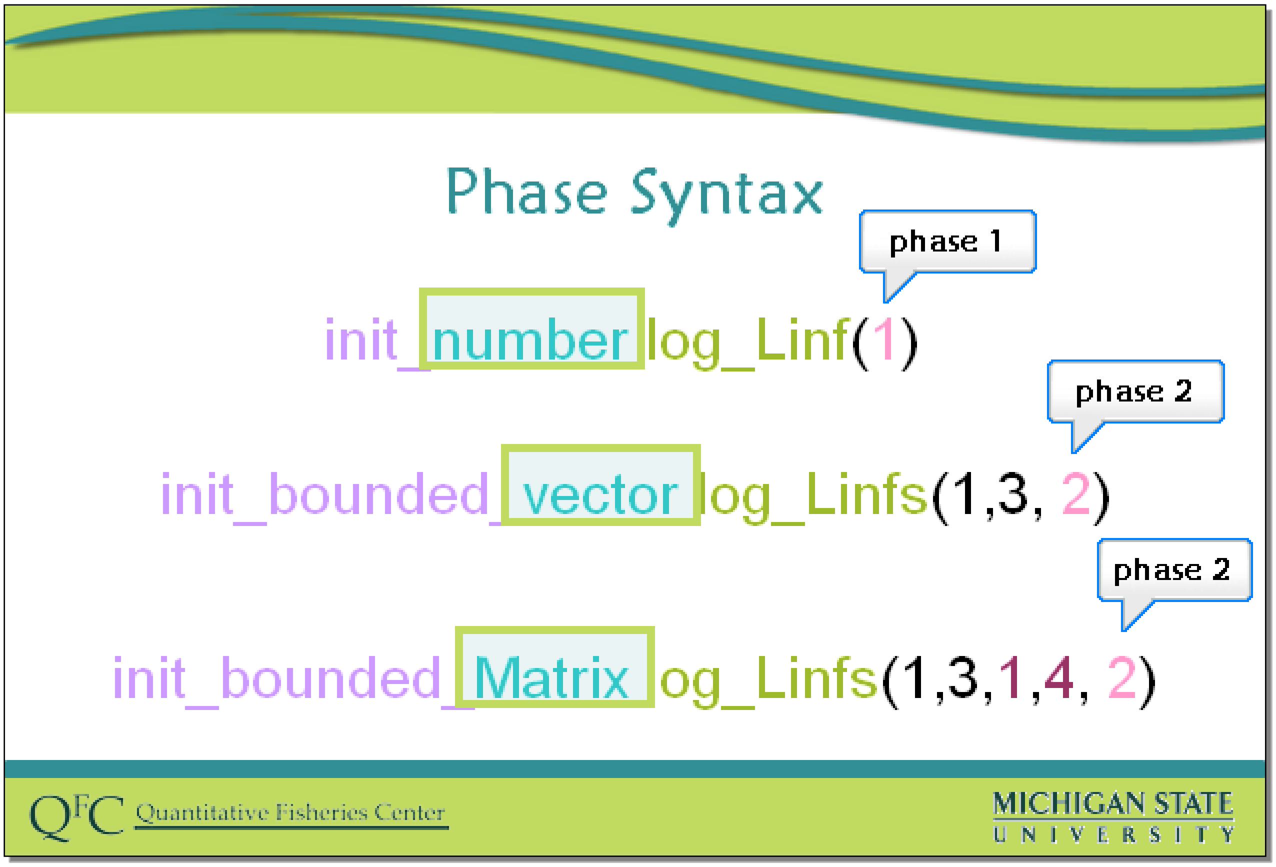
The more certain you are, the higher the phase

In other words, the more certain you are of a starting value, the higher the phase. For example, the asymptotic length for our model could be guessed by looking at the data, so that would be phase 2, Others, which are more uncertain, would be phase 1.



	(Params 1-3) Phase 1 params	(Params 4-6) Phase 2 params	(Params 7-9) Phase 3 params	(Param 10) Phase 4 params
Phase one	Estimated	Not Estimated	Not Estimated	Not Estimated
Phase two	Estimated	Estimated	Not Estimated	Not Estimated
Phase three	Estimated	Estimated	Estimated	Not Estimated
Phase four	Estimated	Estimated	Estimated	Estimated

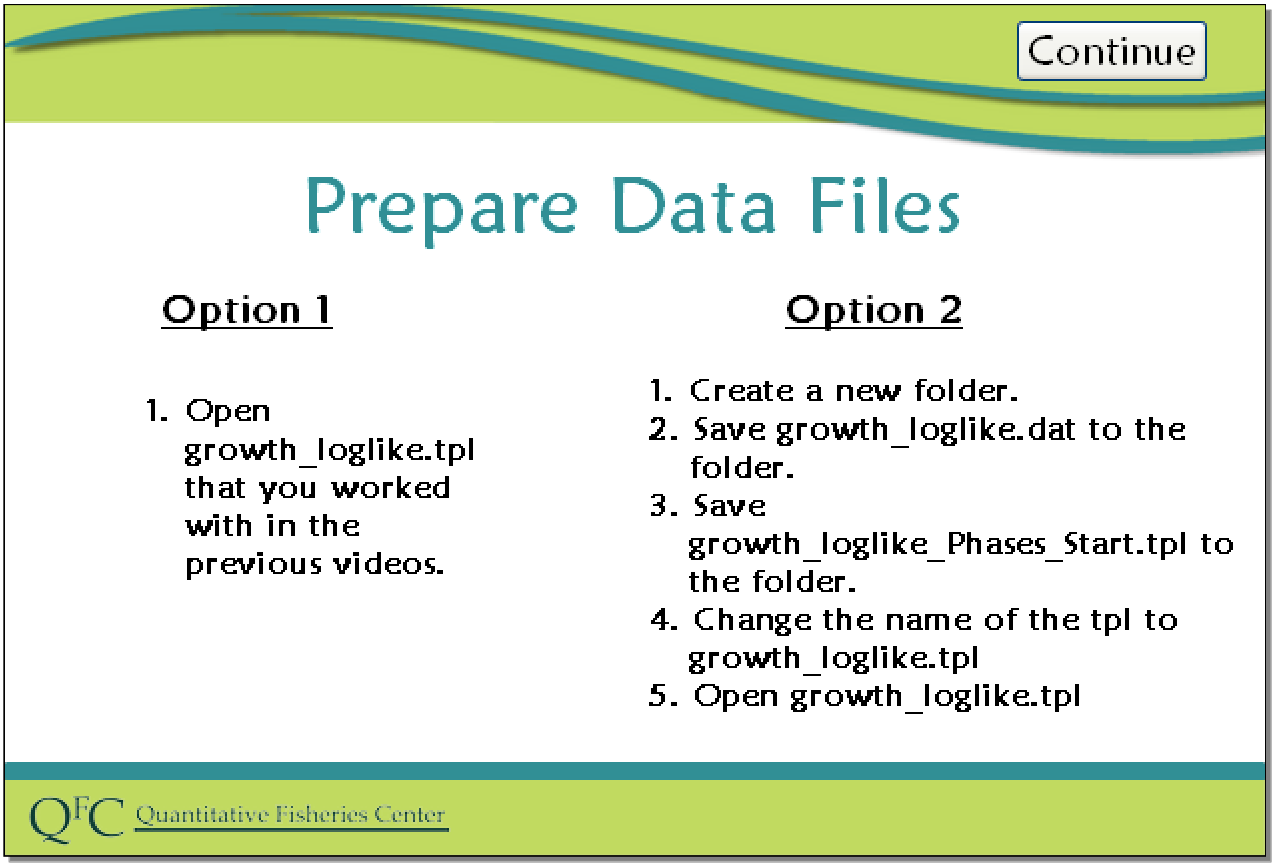
Consider a hypothetical case of a model with 10 parameters and four phases, with parameters 1 through 3 being phase 1 parameters, parameters 4 through 6 being phase 2 parameters, and parameters 7 through 9 being a phase 3 parameters, and the 10th parameter being a phase 4 parameter. An ADMB application would proceed by first doing a numerical search with only the three phase 1 parameters being adjusted and once convergence is achieved, then the six phase 1 and phase 2 parameters would be adjusted until convergence. This phase 2 step would use the estimates obtained for the phase 1 parameters during phase 1 as their starting values for phase 2. During phase 2 the phase 3 and phase 4 parameter are still left at their initial starting values. In phase 3 the nine phase 1 through 3 parameters are estimated while the one phase 4 parameter is still left at its initial starting value. The phase 1 and 2 parameters start where they ended after phase 2, whereas the phase 3 parameters start at the initial starting values you specified. Finally all ten parameters would be adjusted simultaneously until a converged solution is obtained. As for phases 2 and 3, the parameters that were actively estimated in phase 3 start where they ended for phase 3 as phase 4 begins. You could of course do this all manually without phases but it would be a lot of work.



Phase Syntax

```
init_number log_Linf(1)
init_bounded_vector log_Linfs(1,3, 2)
init_bounded_Matrix log_Linfs(1,3,1,4, 2)
```

We can add phases to single parameters, vectors of parameters, or matrices of parameters. For a single ordinary parameter we add the parentheses and put the phase inside. For example this line of code has the parameter estimated in phase is 1. If no phase is indicated the default phase is 1, so by specifying the phase as phase 1 we really have not changed anything. Next is an example of a vector of parameters with an index from 1-3 and a phase of 2. Last is a matrix of parameters with the row index of 1-3, the column index of 1-4 and the phase of 2. Also, notice that the phase is always the last thing that is specified in parenthesis after the array name.



Prepare Data Files

Option 1

1. Open growth_loglike.tpl that you worked with in the previous videos.

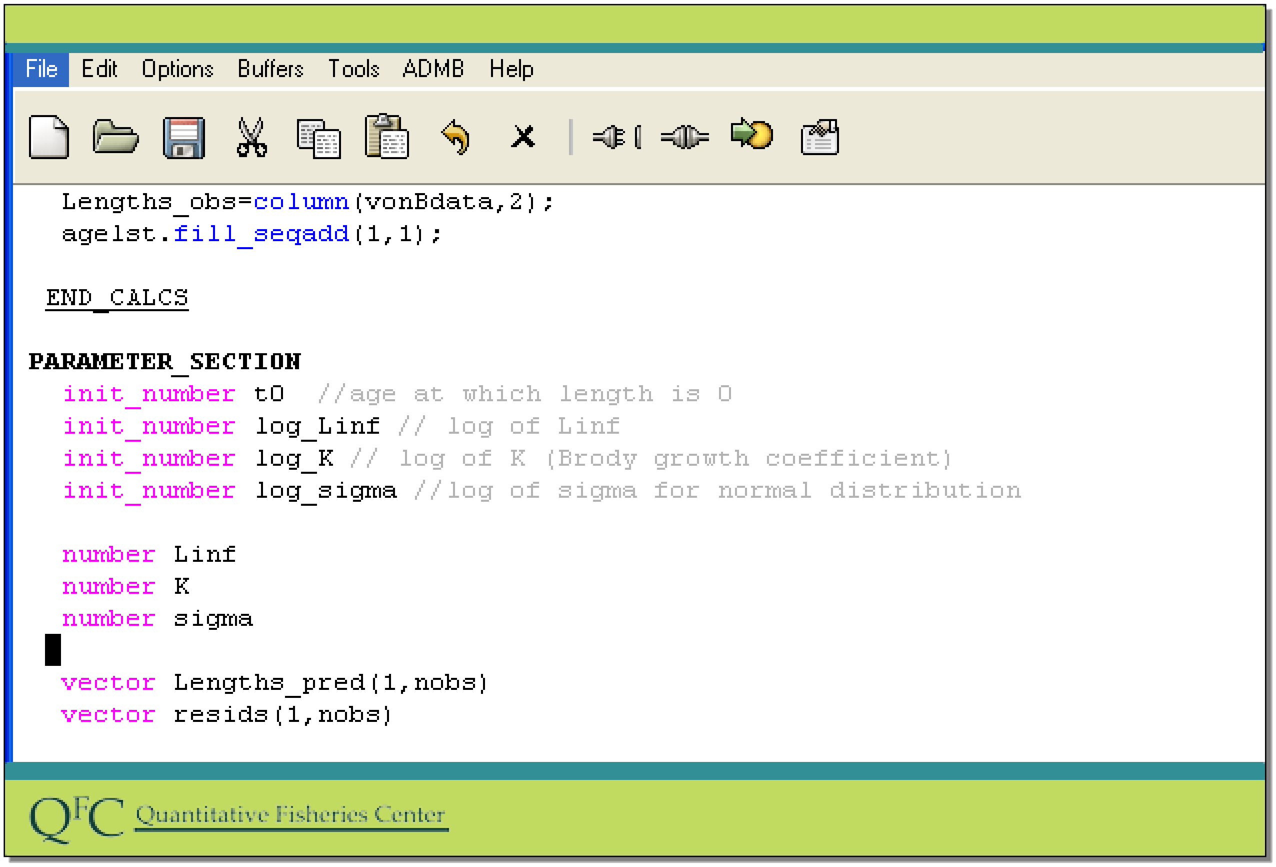
Option 2

1. Create a new folder.
2. Save growth_loglike.dat to the folder.
3. Save growth_loglike_Phases_Start.tpl to the folder.
4. Change the name of the tpl to growth_loglike.tpl
5. Open growth_loglike.tpl

Prepare and open your files. Click continue when you are ready.

Either open growth_loglike.tpl that you worked with in the previous videos or

1. create a new folder.
2. Save growth_loglike.dat to the folder.
3. Save growth_loglike_Phases_Start.tpl to the folder.
4. Change the name of the tpl to growth_loglike.tpl
5. Open growth_loglike.tpl

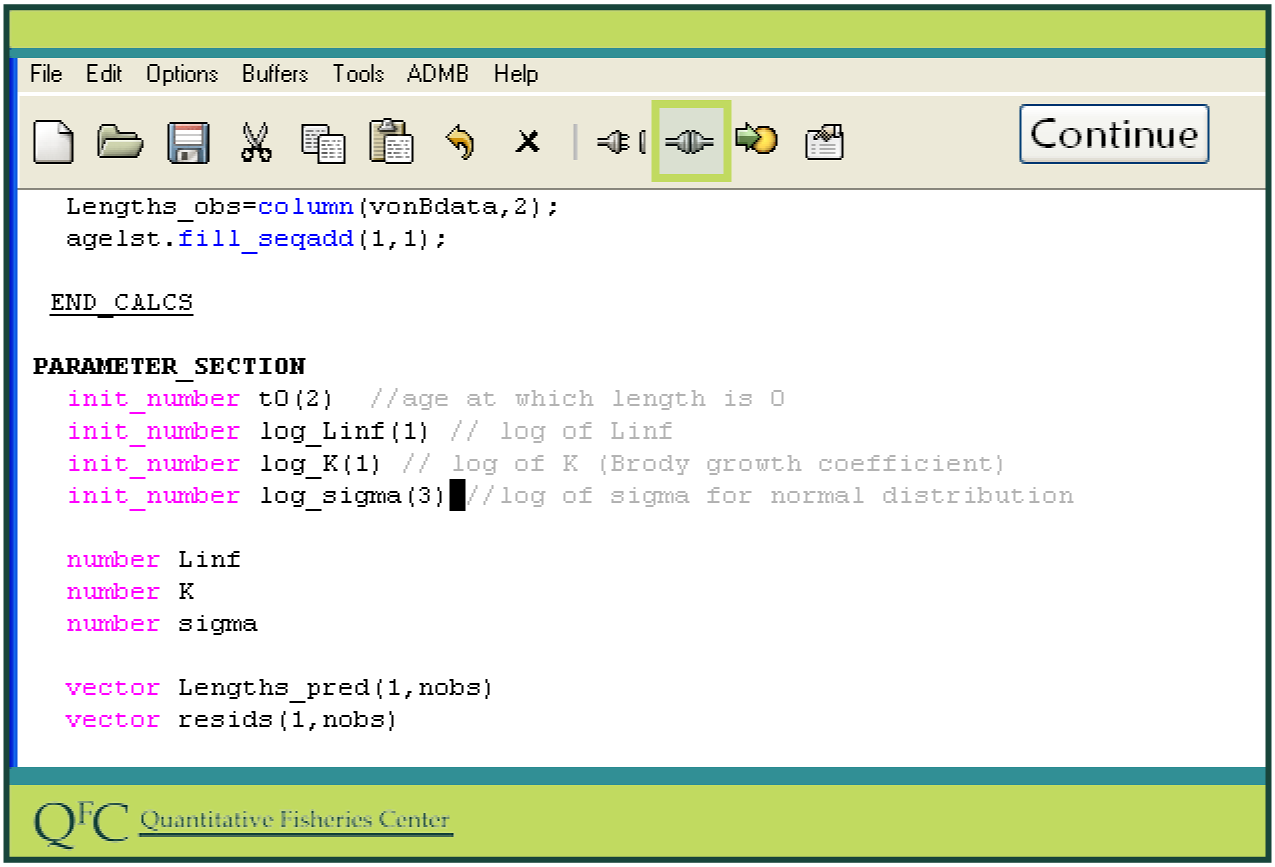


```
  Lengths_obs=column(vonBdata,2);
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0  //age at which length is 0
  init_number log_Linf // log of Linf
  init_number log_K // log of K (Brody growth coefficient)
  init_number log_sigma //log of sigma for normal distribution

  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)
```

To demonstrate the use of phases, we will use the same growth model we previously worked with. Follow along as we alter our code. First we need to return the log_Linf paramter back to a number without bounds.

Slide Code:

init_number log_Linf **(instead of init_bounded_number log_Linf (6.2, 7.3)**


File Edit Options Buffers Tools ADMB Help
Continue
QFC Quantitative Fisheries Center

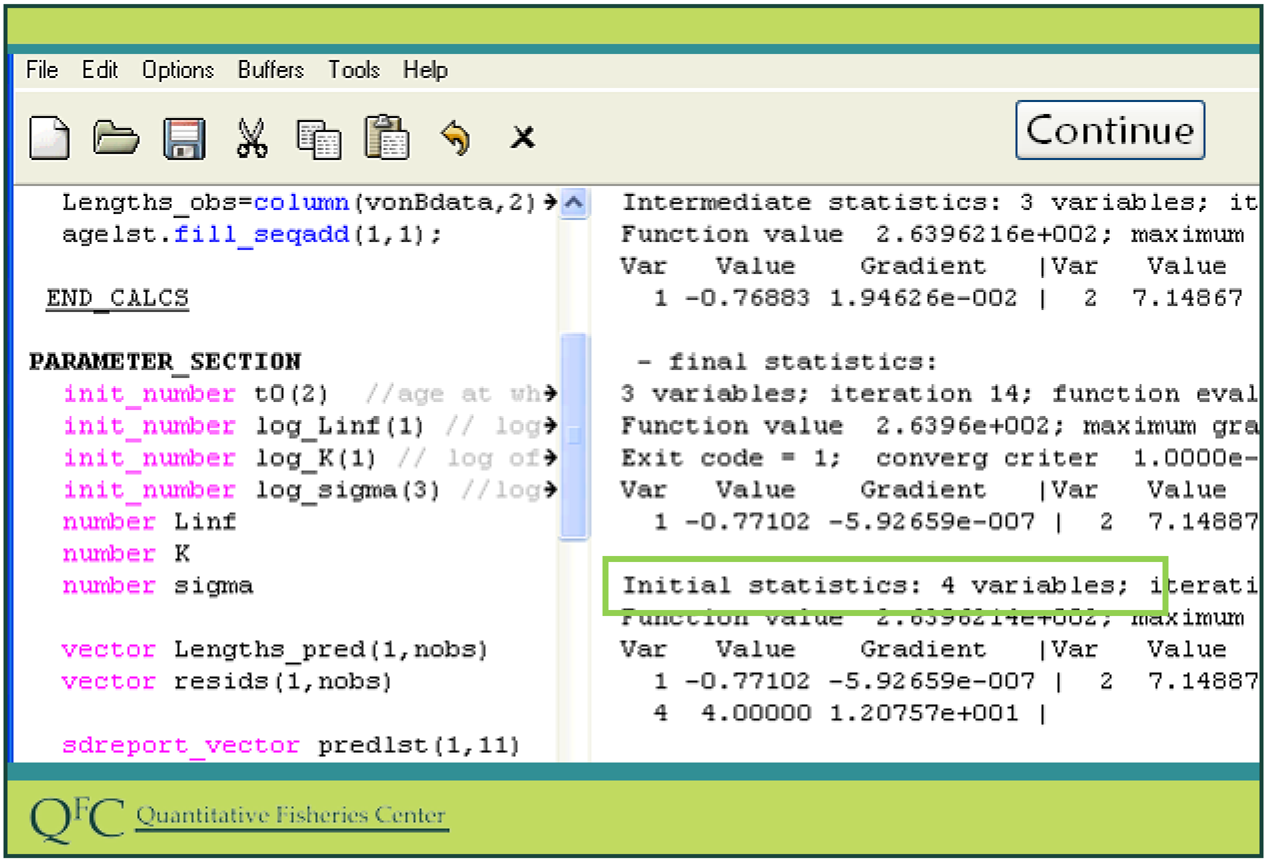
```
  Lengths_obs=column(vonBdata,2);
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0(2)  //age at which length is 0
  init_number log_Linf(1) // log of Linf
  init_number log_K(1) // log of K (Brody growth coefficient)
  init_number log_sigma(3)//log of sigma for normal distribution

  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)
```

We will set L-infinity and K so they are estimated during phase 1. T naught during phase 2 and the error variance in phase three.

Slide Code:

```
init_number t0 (2)
init_number log_Linf (1)
init_number log_K (2)
init_number log_sigma (3)
```

Slide 15 - Slide 15

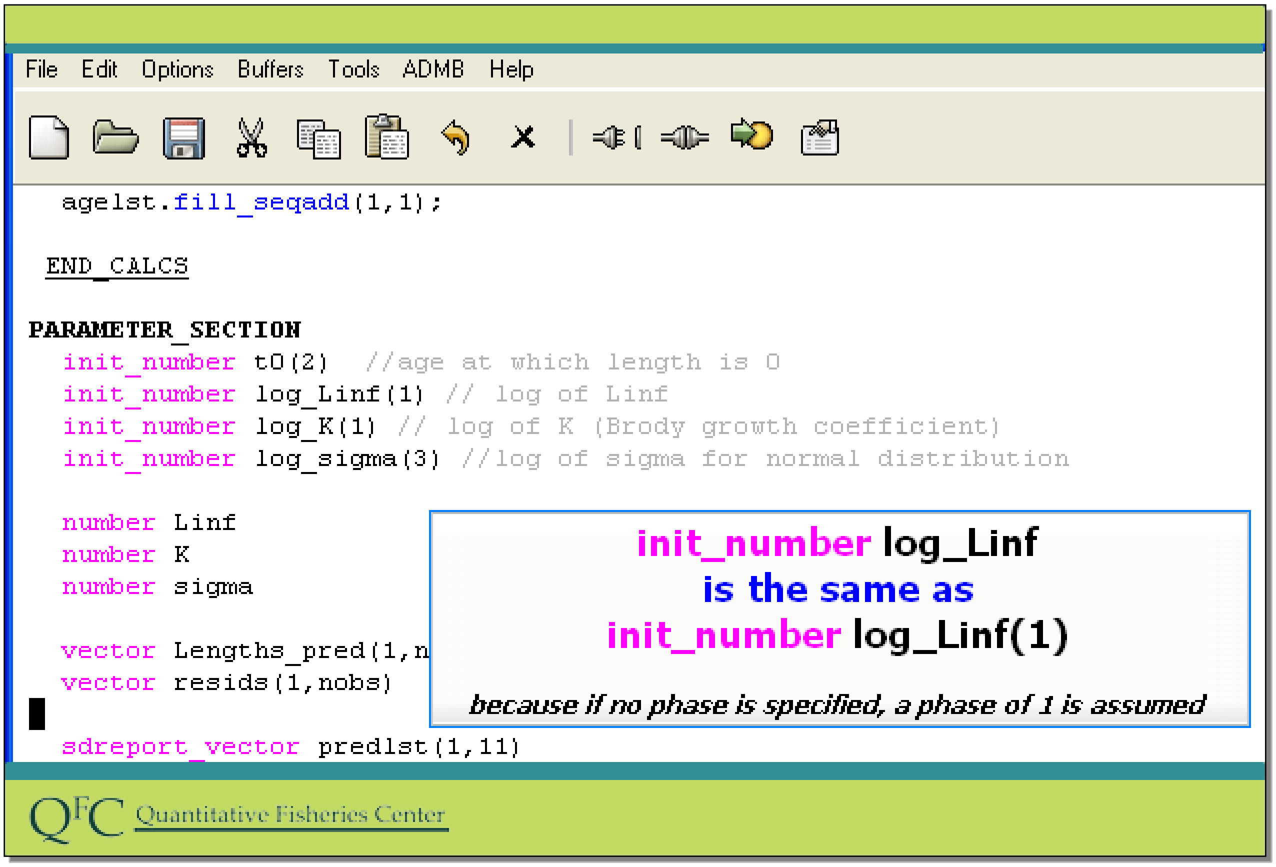

File Edit Options Buffers Tools Help
Continue
QFC Quantitative Fisheries Center

```
  Lengths_obs=column(vonBdata,2)
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0(2)  //age at wh
  init_number log_Linf(1) // log
  init_number log_K(1) // log of
  init_number log_sigma(3) //log
  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)
```

```
Intermediate statistics: 3 variables; it
Function value  2.6396216e+002; maximum
Var   Value    Gradient   |Var   Value
  1 -0.76883 1.94626e-002 |  2  7.14867

 - final statistics:
3 variables; iteration 14; function eval
Function value  2.6396e+002; maximum gra
Exit code = 1;  converg criter  1.0000e-
Var   Value    Gradient   |Var   Value
  1 -0.77102 -5.92659e-007 |  2  7.14887

Initial statistics: 4 variables; iterati
Function value  2.6396214e+002; maximum
Var   Value    Gradient   |Var   Value
  1 -0.77102 -5.92659e-007 |  2  7.14887
  4  4.00000 1.20757e+001 |
```

We now build and run the model. With this simple a model and a small data set things go by so fast it is hard to see the model was actually fit three times.

We can scan through the output generated while the model was fit to see that first a message is generated indicated that we are starting with two estimated parameters, called here variables. They eventually adjusted sufficiently so the model is considered converged. Then three parameters are adjusted, then all four are adjusted, and convergence is achieved.

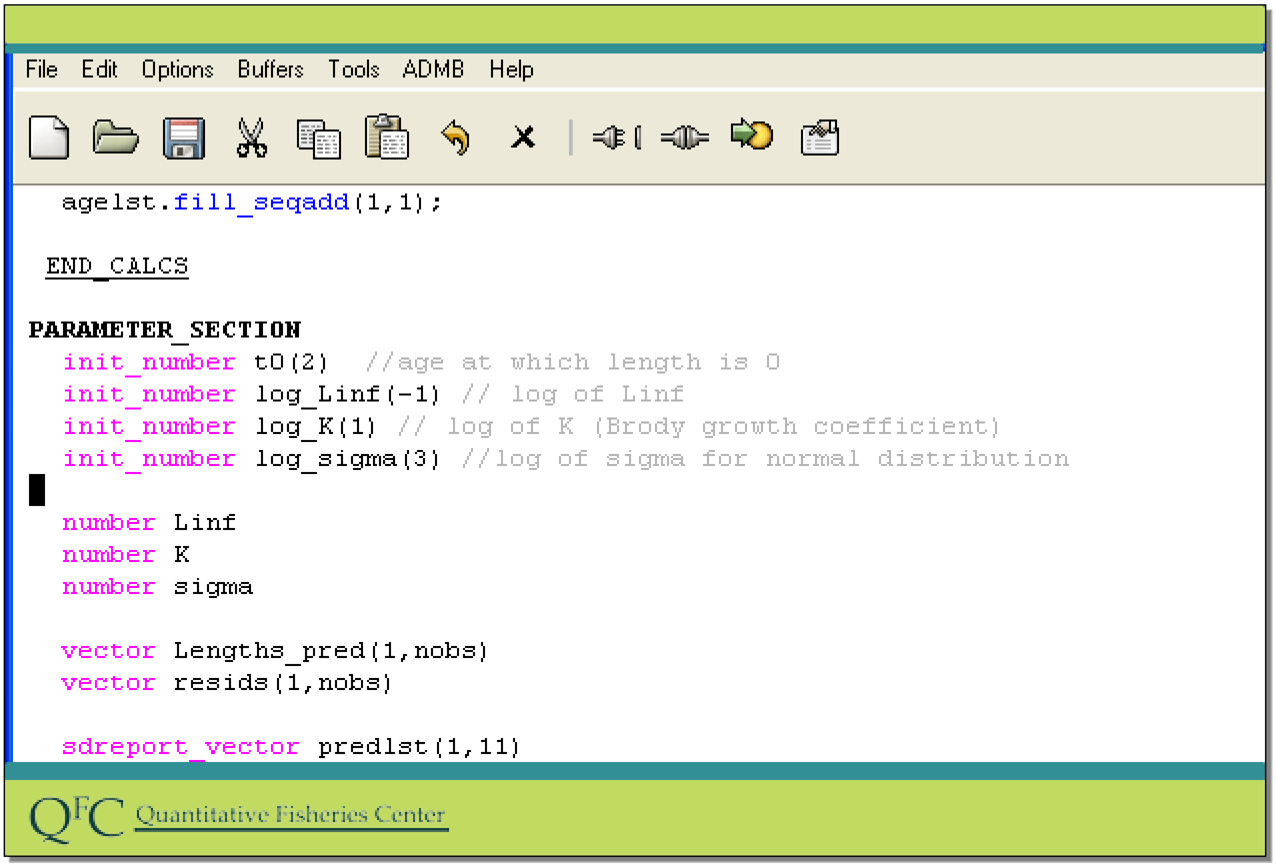

File Edit Options Buffers Tools ADMB Help
QFC Quantitative Fisheries Center

```
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0(2)  //age at which length is 0
  init_number log_Linf(1) // log of Linf
  init_number log_K(1) // log of K (Brody growth coefficient)
  init_number log_sigma(3) //log of sigma for normal distribution

  number Linf
  number K
  number sigma

  vector Lengths_pred(1,n
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)
```

init_number log_Linf
is the same as
init_number log_Linf(1)

because if no phase is specified, a phase of 1 is assumed

Here are a few additional comments regarding specifying phases and bounds. As we indicated before, If no phase is specified, a phase of 1 is assumed. So in this example we could have left out the phase 1 indications for log_Linf and log_K and the same thing would have happened.


File Edit Options Buffers Tools ADMB Help

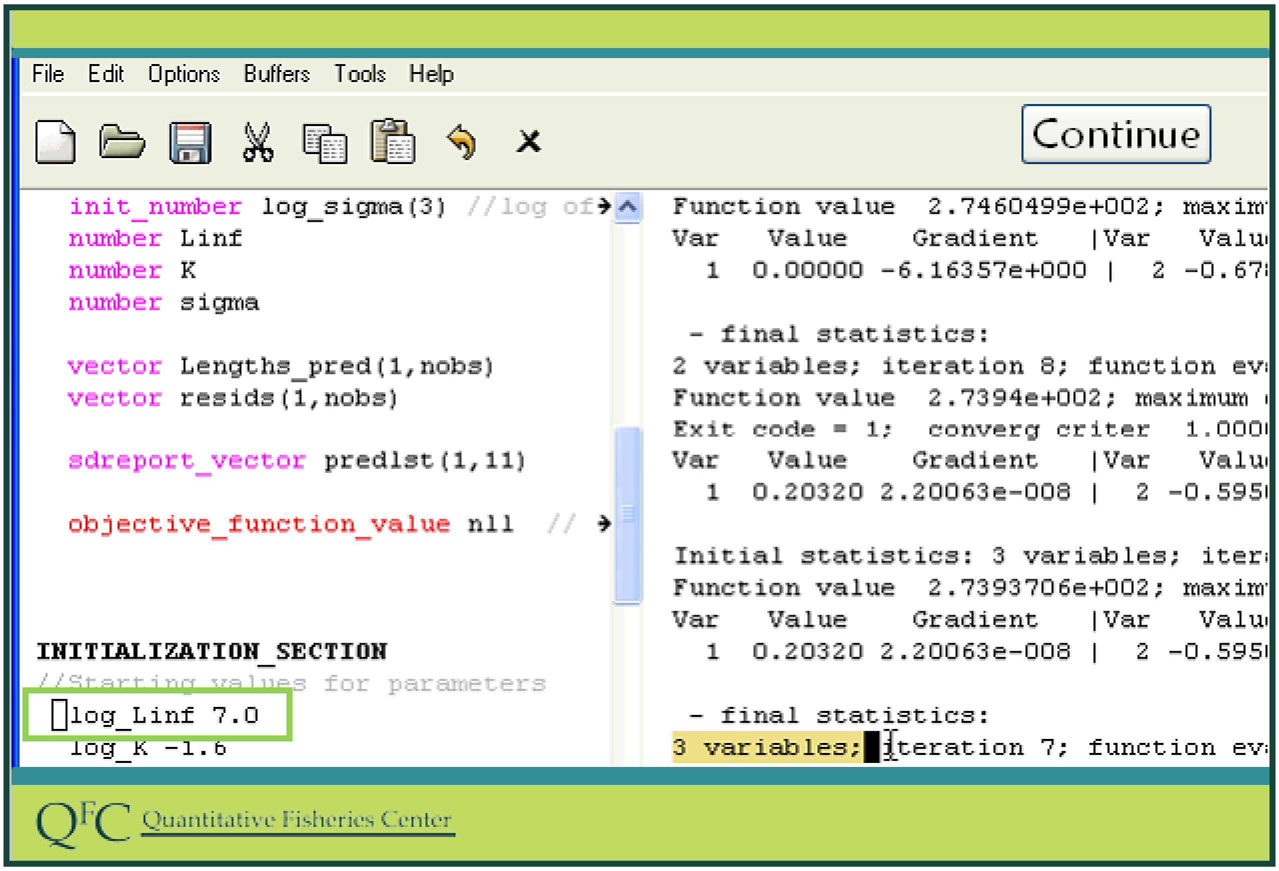
```
   agelst.fill_seqadd(1,1);

  END_CALCS

PARAMETER_SECTION
   init_number t0(2)   //age at which length is 0
   init_number log_Linf(-1) // log of Linf
   init_number log_K(1) // log of K (Brody growth coefficient)
   init_number log_sigma(3) //log of sigma for normal distribution

   number Linf
   number K
   number sigma

   vector Lengths_pred(1,nobs)
   vector resids(1,nobs)

   sdreport_vector predlst(1,11)
```

QFC Quantitative Fisheries Center

If a phase is specified as negative, then the parameter is not estimated at all but instead is left fixed at its starting value. For example let's change the phase for Linf from 1 to -1.

Slide Code:

init_number log_Linf (-1)

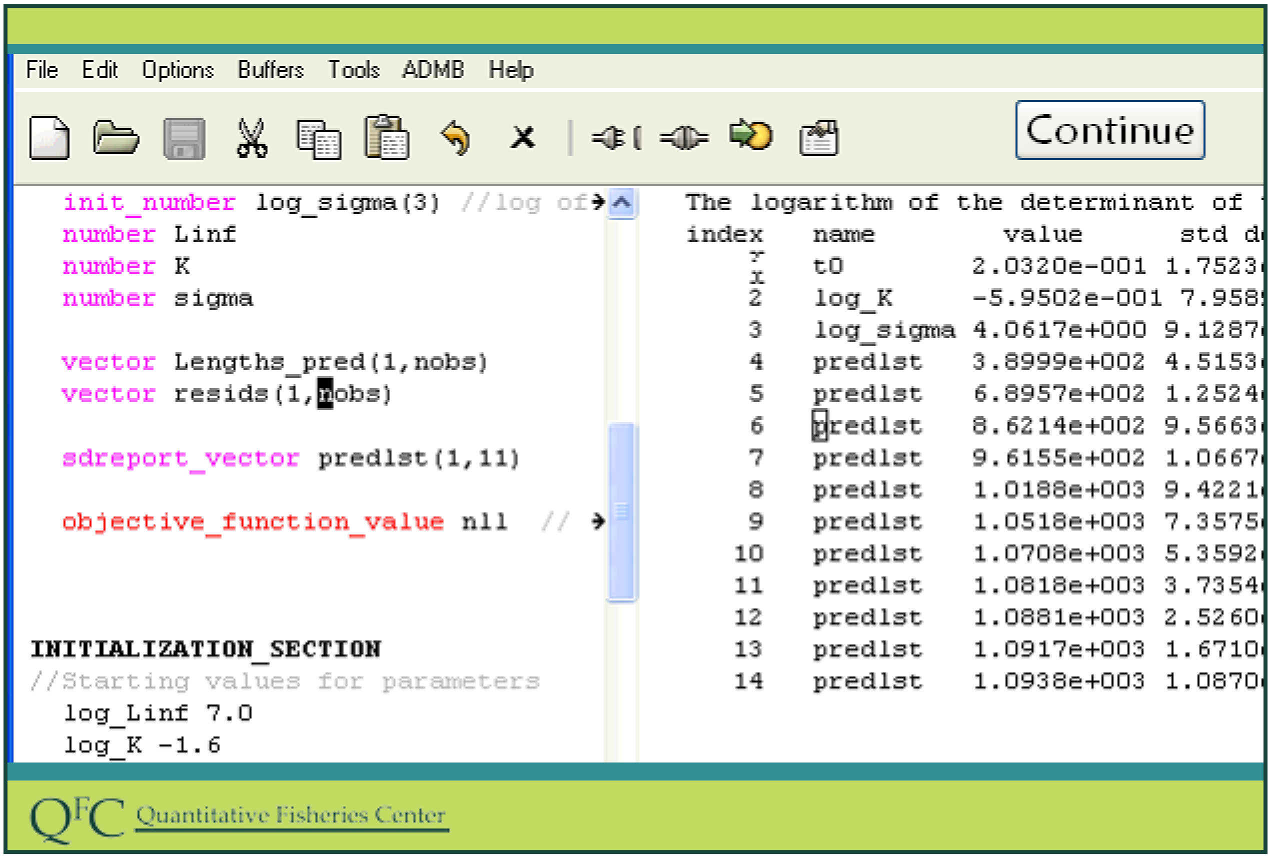

File Edit Options Buffers Tools Help
Continue
QFC Quantitative Fisheries Center

```
  init_number log_sigma(3) //log of
  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)

  objective_function_value nll  //


INITIALIZATION_SECTION
//Starting values for parameters
  log_Linf 7.0
  log_K -1.6
```

```
Function value   2.7460499e+002; maxim
Var    Value     Gradient    |Var   Valu
  1  0.00000 -6.16357e+000 |   2 -0.67

 - final statistics:
2 variables; iteration 8; function ev
Function value   2.7394e+002; maximum
Exit code = 1;  converg criter   1.000
Var    Value     Gradient    |Var   Valu
  1  0.20320 2.20063e-008 |   2 -0.595

Initial statistics: 3 variables; iter
Function value   2.7393706e+002; maxim
Var    Value     Gradient    |Var   Valu
  1  0.20320 2.20063e-008 |   2 -0.595

 - final statistics:
3 variables; iteration 7; function ev
```

Now we rebuild and refit

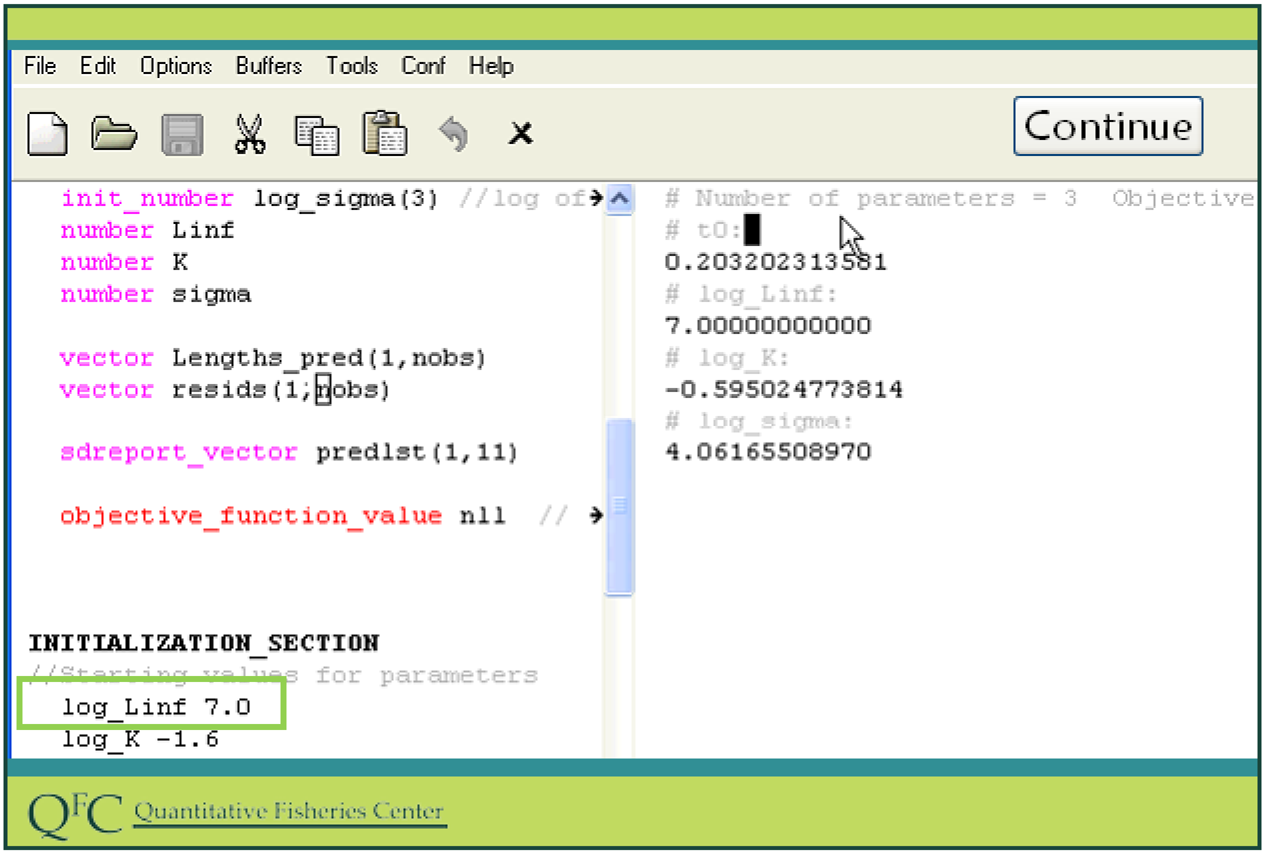
What we have done is search for the best set of parameters for K, t naught and the variance, with Linf assumed known to a value corresponding the starting value we specified. Notice that the output message indicates that only one parameter was adjusted in the first phase and only three in the final third stage.



```
  init_number log_sigma(3) //log of
  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)

  objective_function_value nll  //


INITIALIZATION_SECTION
//Starting values for parameters
  log_Linf 7.0
  log_K -1.6
```

```
The logarithm of the determinant of
index    name        value       std d
   1     t0         2.0320e-001 1.7523
   2     log_K     -5.9502e-001 7.958
   3     log_sigma 4.0617e+000 9.1287
   4     predlst    3.8999e+002 4.5153
   5     predlst    6.8957e+002 1.2524
   6     predlst    8.6214e+002 9.5663
   7     predlst    9.6155e+002 1.0667
   8     predlst    1.0188e+003 9.4221
   9     predlst    1.0518e+003 7.3575
  10     predlst    1.0708e+003 5.3592
  11     predlst    1.0818e+003 3.7354
  12     predlst    1.0881e+003 2.5260
  13     predlst    1.0917e+003 1.6710
  14     predlst    1.0938e+003 1.0870
```

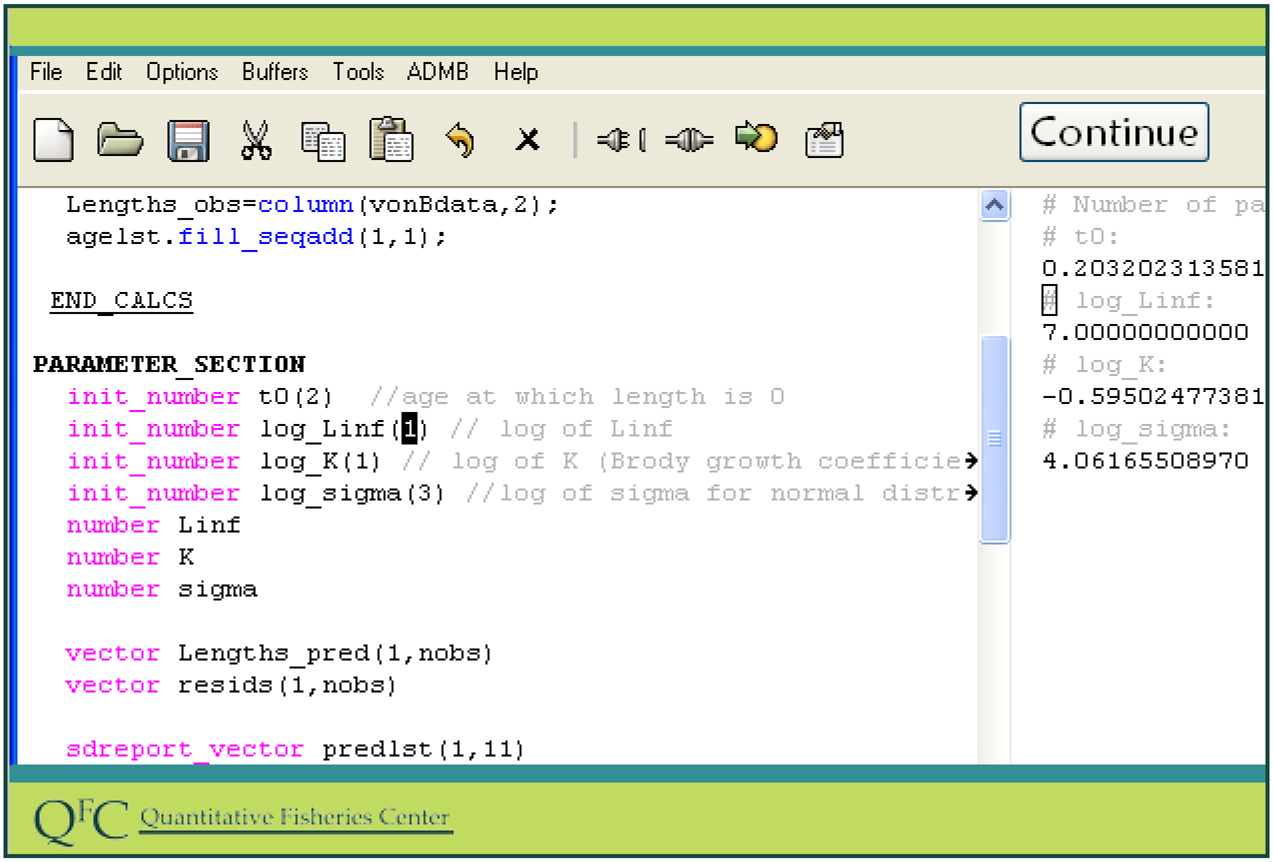
Notice also that if we go look at cor file (ADMB menu to View Estimates) that log_Linf is not included as an estimated quantity.



```
  init_number log_sigma(3) //log of
  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)

  objective_function_value nll   //


INITIALIZATION_SECTION
//Starting values for parameters
  log_Linf 7.0
  log_K -1.6
```

```
# Number of parameters = 3  Objective
# t0:
0.203202313581
# log_Linf:
7.00000000000
# log_K:
-0.595024773814
# log_sigma:
4.06165508970
```

If we go to the par file, (ADMB menu to View Point Estimates) log_Linf is listed but it has its initial value of 7.0 which was our starting valule in the initialization section.


File Edit Options Buffers Tools ADMB Help
Continue
QFC Quantitative Fisheries Center

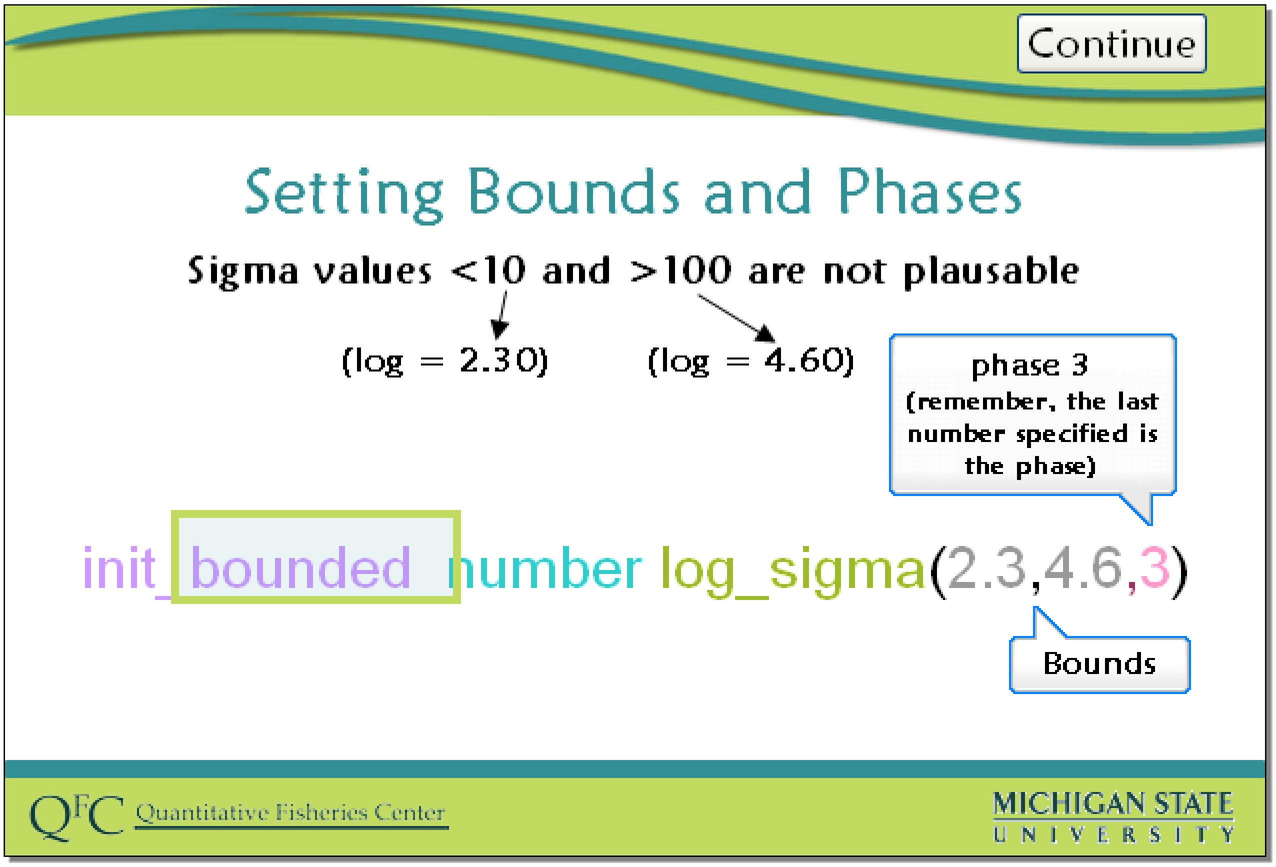
```
  Lengths_obs=column(vonBdata,2);
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0(2)  //age at which length is 0
  init_number log_Linf(1) // log of Linf
  init_number log_K(1) // log of K (Brody growth coefficie
  init_number log_sigma(3) //log of sigma for normal distr
  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)

  sdreport_vector predlst(1,11)
```

```
# Number of pa
# t0:
0.203202313581
# log_Linf:
7.00000000000
# log_K:
-0.59502477381
# log_sigma:
4.06165508970
```

We will now remove the negative sign in front of the Log_Linf phase to make it active again.

Slide Code:

init_number log_Linf (1)



Setting Bounds and Phases

Sigma values <10 and >100 are not plausable

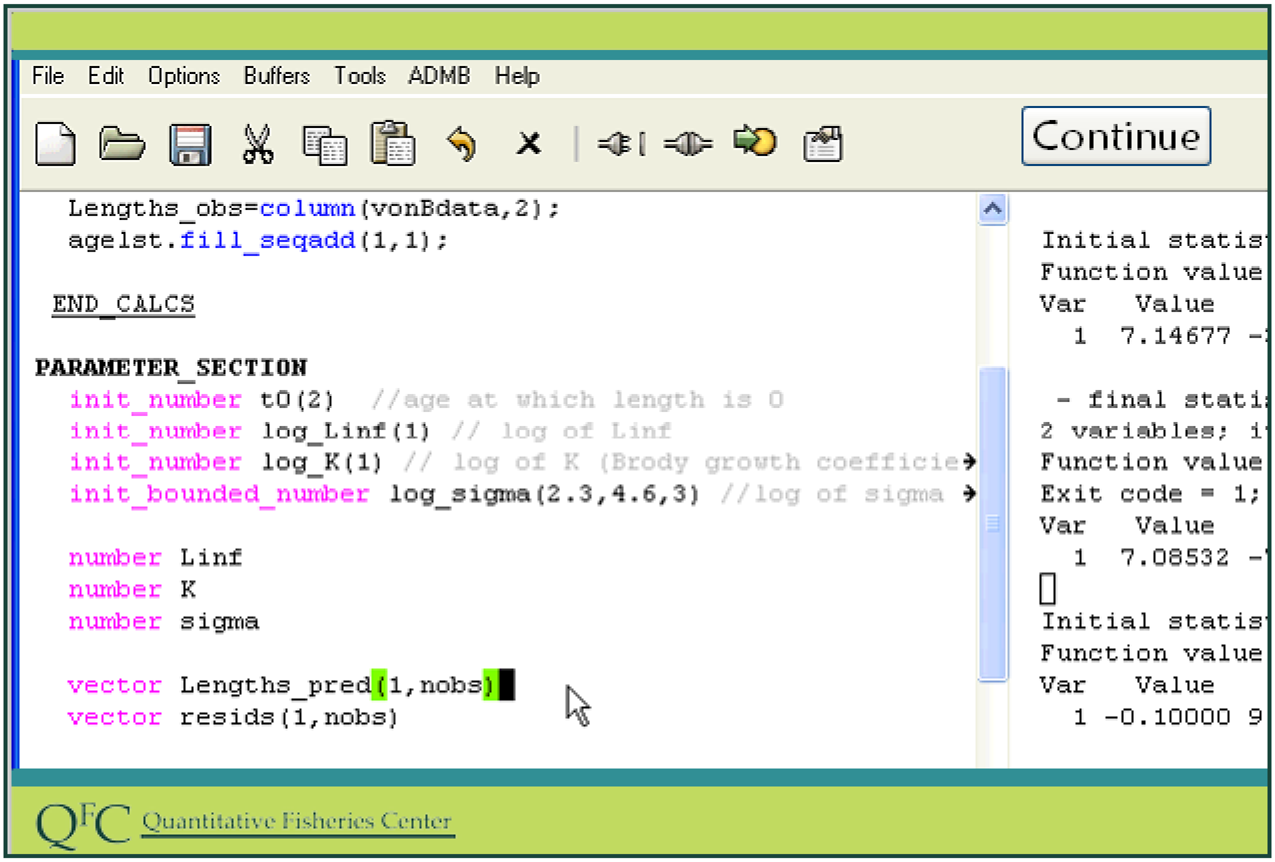
(log = 2.30) (log = 4.60)

phase 3 (remember, the last number specified is the phase)

init_bounded_number log_sigma(2.3,4.6,3)

Bounds

You can use phases and bounds together. For example let us assume that we know that sigma values less than 10 or greater than 100 are not plausible so we set bounds of log_sigma of 2.3 and 4.6. We would add these bounds just as we did earlier, by adding bounded and adding the lower and upper bounds in parenthesis. We also added the phase after the upper bound. Let's try it with our template.



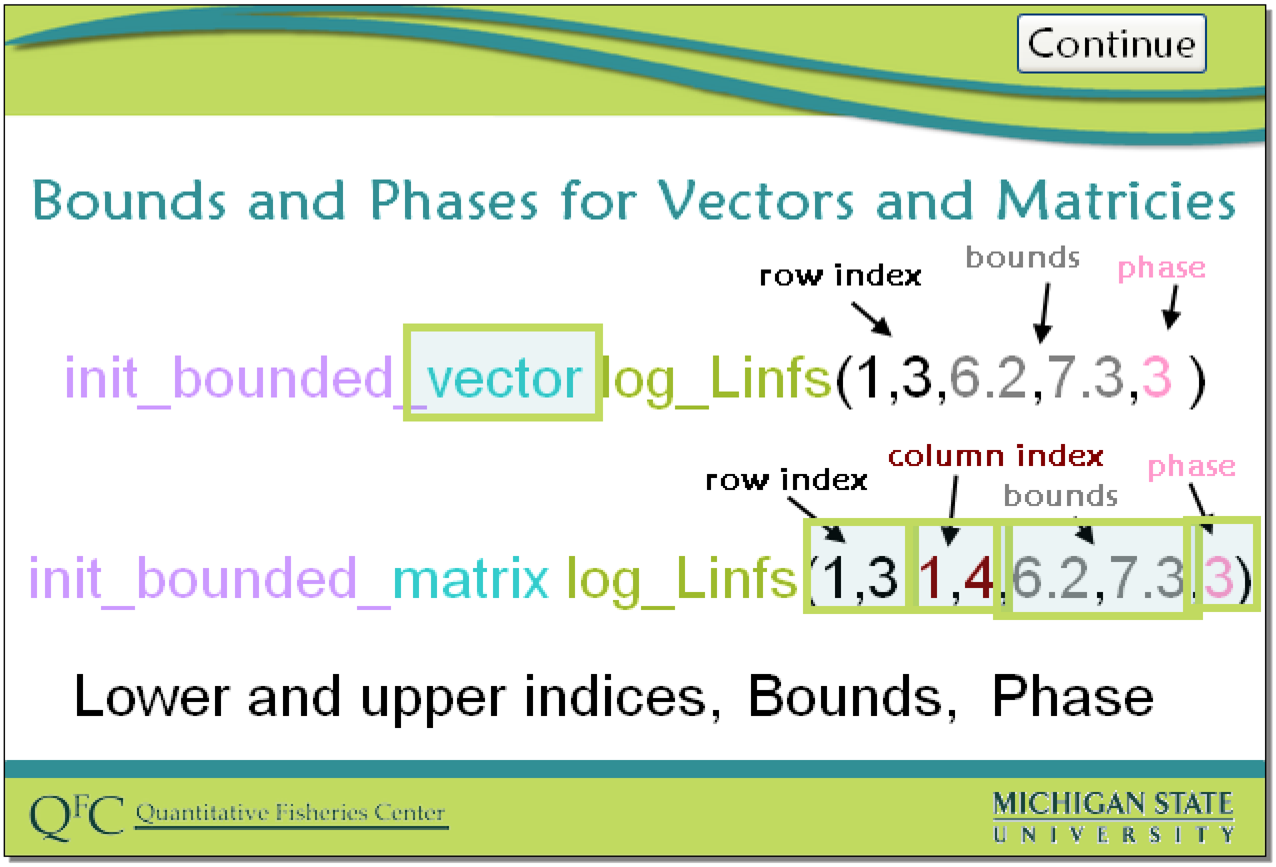
```
  Lengths_obs=column(vonBdata,2);
  agelst.fill_seqadd(1,1);

 END_CALCS

PARAMETER_SECTION
  init_number t0(2)  //age at which length is 0
  init_number log_Linf(1) // log of Linf
  init_number log_K(1) // log of K (Brody growth coefficie➔
  init_bounded_number log_sigma(2.3,4.6,3) //log of sigma ➔

  number Linf
  number K
  number sigma

  vector Lengths_pred(1,nobs)
  vector resids(1,nobs)
```

```
 Initial statis
Function value
Var    Value
  1   7.14677 -

  - final stati
2 variables; i
Function value
Exit code = 1;
Var    Value
  1   7.08532 -
▯
 Initial statis
Function value
Var    Value
  1 -0.10000 9
```

Here we adjust log_sigma to include the bounds in addition to keeping it at phase 3.

Slide Code:

init_bounded_number log_sigma (2.3, 4.6, 3)

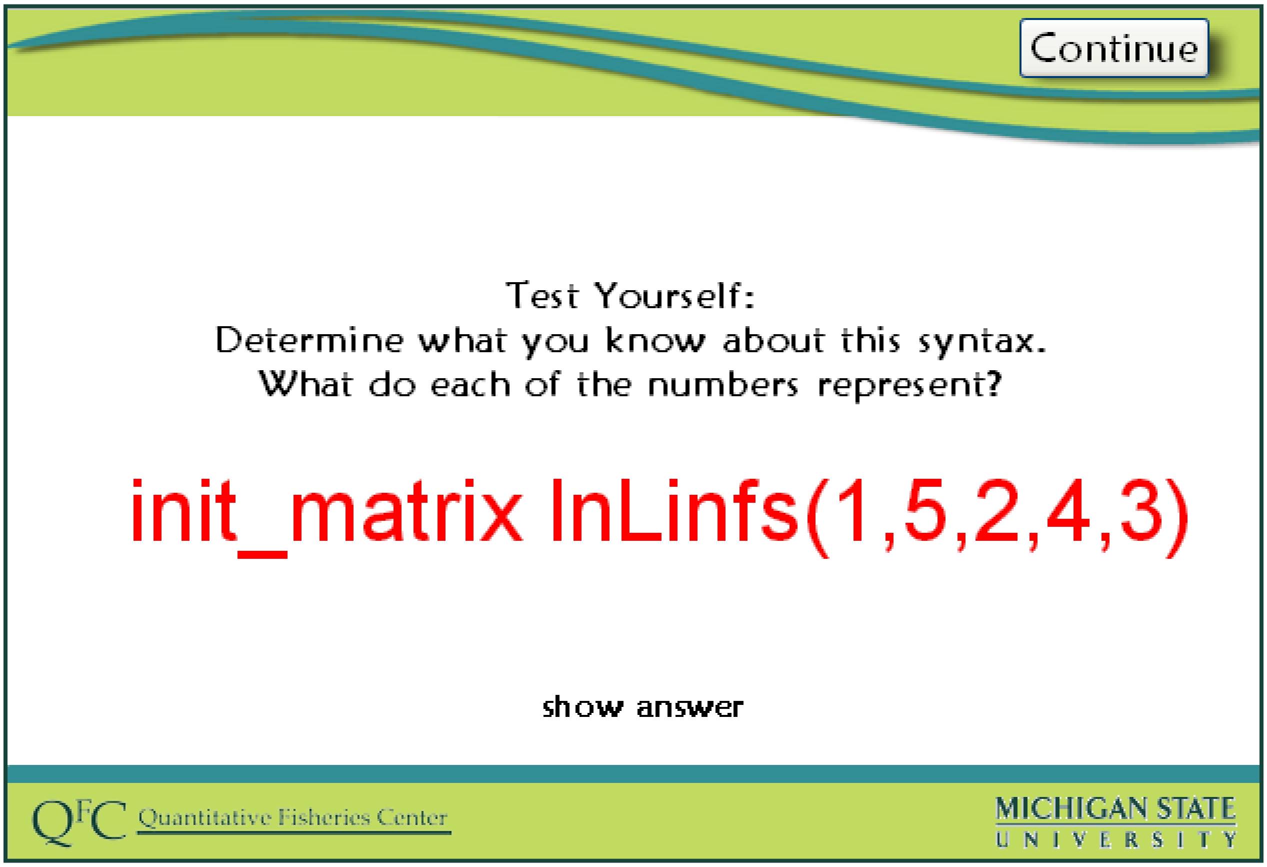


Bounds and Phases for Vectors and Matricies

```
init_bounded_vector log_Linfs(1,3,6.2,7.3,3 )
init_bounded_matrix log_Linfs(1,3,1,4,6.2,7.3,3)
```

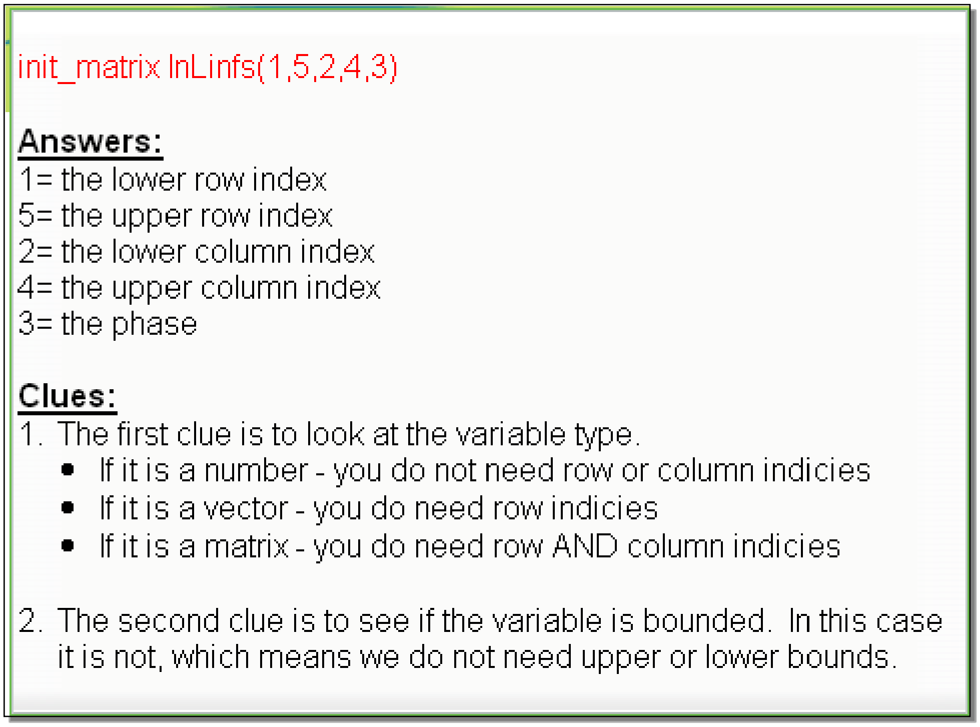
Lower and upper indices, Bounds, Phase

We sometimes define vectors or even matrices of parameters, together with bounds, and a phase. In such situations we end up with a bunch of numbers within parenthesis separated by commas. For a vector, the first 2 numbers are the row indicies, followed by the lower and upper bounds, and last by the phase. Matricies have more numbers. First are the row indicies, followed by the column indicies, then the lower and upper bounds and last by the phase. Keep in mind that the order things go in are: lower and upper limits for row then column indices, followed by bounds, then followed by the phase which is always last. AD Model builder knows whether you need to specify index limits and bounds based on the kind of variable you are creating.



Test Yourself:
Determine what you know about this syntax.
What do each of the numbers represent?

init_matrix lnLinfs(1,5,2,4,3)



init_matrix lnLinfs(1,5,2,4,3)

Answers:

1= the lower row index
5= the upper row index
2= the lower column index
4= the upper column index
3= the phase

Clues:

1. The first clue is to look at the variable type.
   - If it is a number - you do not need row or column indicies
   - If it is a vector - you do need row indicies
   - If it is a matrix - you do need row AND column indicies

2. The second clue is to see if the variable is bounded. In this case it is not, which means we do not need upper or lower bounds.

The End

admb

FAST, ACCURATE, STABLE OPTIMIZATION

QFC Quantitative Fisheries Center

MICHIGAN STATE UNIVERSITY

You have now completed the video on phases which also showed you how to combine phases with bounds.