Gadget course

Day 1

Bjarki Þór Elvarsson and Guðmundur Þórðarsson

University of Iceland, Marine Research Institute

19. september 2014

Today:

1. How does Gadget work?
   - Basic building blocks
   - Order of calculations
   - Initialization
   - Migration
   - Predation and Fleet Operations
   - Growth and other processes
2. Gadget Input/Output files – The nitty gritty of it....
   - The main file
   - Time file
   - Area file
   - Other input files
     - Aggregation files
     - TimeVariable files
   - Stock file
   - Fleet Files
   - Parameter File

What is Gadget?

- shorthand for: **"Globally applicable Area Disaggregated General Ecosystem Toolbox"**
- is an age-length structured forward-simulation model,
- has an extensive set of data comparison and optimization routines.
- Processes are generally modeled as dependent on length, but age is tracked in the models.
- data can be compared on either a length and/or age scale.

The model is designed as a:

- multi-area,
- multi-fleet model,
- capable of including predation and mixed fisheries issues.
- can also be used on a single species basis.

History

`Gadget` is based on previous ideas by several authors

- Conceptionally it extends earler multispecis frameworks such ar `MSVPA` and `MULTISPEC` into a more generic statistical framework
- `Gadget` is a conceptional extension of the `Stock Synthesis` statistical assessment single-species framework into a multi species setting.

`Gadget` immediate predecessor were the `BORMICON` and `Fleksibest` models.

- The initial BORMICON code was developed as part of the multi species program by MRI in 1992 with Ólafur K. Pálsson as project coordinator and Gunnar Stefánsson coordinating the modelling work.
- The BORMICON code became the basis for Fleksibest at the IMR in Bergen, Norway.
- Further development work in 1999-2003 was partly funded by EU grant QLK5-CT1999-01609.

People behing Gadget

The Gadget code originates from:

- the original BORMICON code of Halldór Narfi Stefánsson, Höskuldur Björnsson and Hersir Sigurgeirsson.
- Subsequent contributions include program additions from Morten Nygaard and Kristin Froysa in the Fleksibest model.
- Parallel computing was implemented by Auðbjörg Jakobsdóttir and mathematical additions and changes to allow parallel computing were implemented by Þórdís Linda Þórarinsdóttir and Kristjana Ýr Jónsdóttir.
- Inclusion of tagging data was implemented by Sigurdur Hannsesson
- Work on multivariate distributions, iterative reweighting and BFGS was implemented by Bjarki Þór Elvarsson.
- Last but not least, James Begley has been responsible for the code and its maintainance.

Uses of Gadget in stock assessment

Gadget has been used as a single stock assessment models for various stocks in the North Atlantic.
The reasons include:

- lack of age data
- Incomplete data set (Gaps in age structured data etc)
- Flexibility of the Gadget framework to use various different types of data

Examples of stocks

Iceland:

- Shrimp (Multi species)
- Wolffish
- Plaice (been a while)

ICES

- Golden Redfish (NWWG, AFWG)
- Tusk Va and XIV (WGDEEP)
- Ling Va and XIV (WGDEEP) - exploratory
- Southern Hake (WGHMM)

Where are we now...

1. How does Gadget work?
   - Basic building blocks
   - Order of calculations
   - Initialization
   - Migration
   - Predation and Fleet Operations
   - Growth and other processes
2. Gadget Input/Output files – The nitty gritty of it....
   - The main file
   - Time file
   - Area file
   - Other input files
     - Aggregation files
     - TimeVariable files
   - Stock file
   - Fleet Files
   - Parameter File

Basic building blocks

- Gadget contains a highly congurable ecosystem simulator that emulates the interaction between **substocks**, representing either parts of the same species or separate species, living in one or more **areas**.
- Gadget is a forward simulator where the simulation takes place in a set number of years.
- It allows for the simulation of (irregular sized) timesteps within those years to allow for the emulation of various cyclical process such as spawning or stock migrations.
- Any process can be parametrised by the user, both for use estimation and/or scenario analysis.

- The fundamental block in Gadget is the **substock**, an entity that is homogeneous with respect to various processes such as growth and migration. In this way the immature and mature population within a species can be modelled as separate substocks with, as an example, different migration patterns or growth process.
- The substocks live in an **area**, or areas, where they optionally migrate to and from. In this setup processes such as fleet harvest or recruitment can be restricted to take place only in certain areas.
- Harvesting of the substocks is defined through **fleets** that fish according to effort (or quota) and use selection functions (either length or age based but more on that later). And as with other processes fleet operations can be defined on some or all areas.

- The state of the ecosystem that is simulated in Gadget is observed through the substock abundance, consumption and harvest.
- The abundances of the substocks in the simulation are observed through a number of individuals N_{sralt} of that particular substock s, area r, age group a, length group l and time t[1].
- Similarly the consumption and harvesting is observed through the number of individuals consumed E_{psralt} and caught C_{fralt} by predator p, fleet f with r, a, l and t as before.
- Based on these measurements a variety of indexes or data of interest can be extrapolated (such as *age-length-key* by fleet) that are nescessary for comparison with real data.

[1] A note on the notation, here a subscript will be omitted when a process applies to all elements of the subscript.

Order of calculations

In the following slides it will be useful to keep in mind the order of which Gadget performs its calculations. For each timestep of the simulation the order of calculations is:

- **Migration:** Move the stocks around between areas.
- **Consumption:** Calculate the consumption of predators and fleet harvest.
- **Natural mortality:** Reduce the population by the fraction due to natural mortality.
- **Growth:** Calculate the increase in length and weight.
- **Spawning:** Calculate the effects of spawing on the mature fish and place any spawned stock into temporary storage.

Order of calculations (cont.)

- **Maturation:** Add newly matured fish into the model.
- **Recruitment:** Add recruitment i.e. spawned fish, if modeled or direct recruitments.
- **Straying:** Move fish between stocks.
- **Likelihood comparison:** Calculate the likelihood score based on available data and current state of the simulation.
- **Printing:** output the state of the simulator (if required).
- **Ageing:** Age the fish.

Initial state

Gadget is a forward simulator and as such it requires a starting point for its simulation. The simulator starts by generating 10.000 individuals for each age and user is given a choice between three options to assign lengths and weights to them:

- **Normal condition distribution:** Lengths at age are normally distribution and weights are relative to a reference weight and weight condition factor.
- **Normal parametric distribution:** Lengths at age are normally distribution and weights are determined by $W = \alpha L^{\beta}$.
- **Numerical distribution:** Initial conditions are given in the numbers of fish by area, age, length and weight.

Migration

- If more than one area is specified migration can be defined between them. The migration procedure used in Gadget is similar to the one used in MULTSPEC.
- Migration is defined via a set migration matricies. The proportion that migrates from area i to j can be different between different time steps (seasons) and years.
- Migration at time step t is defined by the user as an $n \times n$ transition matrix $P_t := [p_{ij}]$ where p_{ij} is the proportion moving from area j to area i, ignoring other processes.
- For P to be a transition matrix $\sum_i p_{ij} = 1$, for each j. The vector of abundance for all areas at time t is therefore:

$$\mathbf{N_t} = P_t \mathbf{N_{t-1}} \tag{1}$$

In a two area example this would mean that if $N_{1,t-1}$ is a matrix containing the abundance in numbers in area 1 at time step t before migration and $N_{2,t-1}$ is the same number for area 2, the numbers after migration will be

$$\begin{aligned} N_{1,t} &= p_{11} \cdot N_{1,t-1} + p_{12} \cdot N_{2,t-1} \\ N_{2,t} &= p_{21} \cdot N_{2,t-1} + p_{22} \cdot N_{2,t-1} \end{aligned} \tag{2}$$

NOTE: Migration needs to be defined with some care, especially if the number of areas is large.

Predation

- In Gadget a number of predator-prey relationships can be defined. They include the obvious case where one substock predates the other.
- Cannibalistic relationships can be defined if two stocks are modelled as an immature and a mature part of the same species.
- Fleets (survey and/or commercial) can be integrated into the model by being considered as one of the predators.

Consumption

In general, Gadget will conceptually assign the following to its predators:

- **Prey preference**, d_p, how likely the predator is willing to pursue
- **Maximum consumption** is the maximum possible consumption by the predator. It based on the predator length and temperature as follows:

$$M_L = m_0 e^{(m_1 T - m_2 T^3)} L^{m_3}$$

- **Half feeding value** H is the density (biomass per area unit) of available food at which the predator can consume half maximum consumption. It determines how quickly the predator reaches maximum consumption.

- **Availability** of a certain prey in comparison with other food sources

$$\phi_p(l,L) = \frac{F_p(l,L)}{\sum_p \sum_{l'} F_p(l',L) + OA}$$

where $F_p(l,L) = (S_p(l,L)E_pN_lW_l)^{d_p}$, S_p is the prey **suitability**, E_p energy content, O is the amount of other specified food sources (not modeled) per unit area and A is the area.

- **Feeding level**, i.e., fraction of available food the predator wants to consume.

$$\psi_L = M_L \frac{\sum_p F_p(l,L) + OA}{HA + \sum_p F_p(l,L) + OA}$$

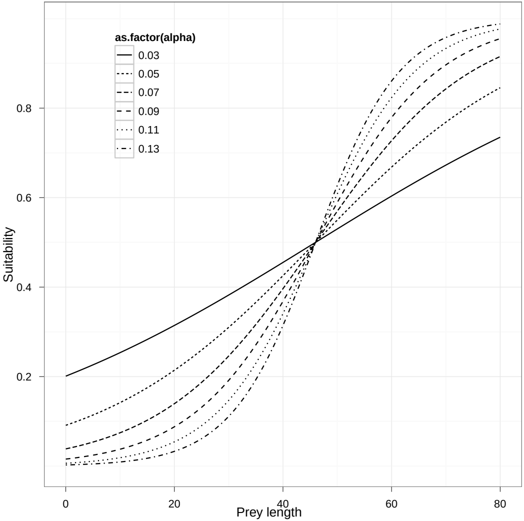
Combining these factors $C_p(l,L) = N_L\phi_p(l,L)\psi_L$, i.e. the number of predators, food availability and feeding level the amount that the predator wants to consume is calculated. If the total biomass that the predators want to consume is greater than the total biomass the consumption is adjusted accordingly.

Suitability functions

Food availability is directly determined from the total biomass and the suitability functions that the predator uses. Suitability functions can be interpreted as the proportion of prey of length l predators of length L are willing to consume. In Gadget the user can define a number of suitability functions.

Constant	$S(L,l) = \alpha$
StraightLine	$S(L,l) = \alpha + \beta l$
Exponential	$S(L,l) = \frac{\delta}{1+e^{-\alpha-\beta l-\gamma L}}$
ExponentialL50	$S(L,l) = \frac{\delta}{1+e^{-\alpha(l-l_{50})}}$
Richards	$S(L,l) = \left(\frac{\delta}{1+e^{-\alpha-\beta l-\gamma L}}\right)^{\eta}$

ExponentialL50

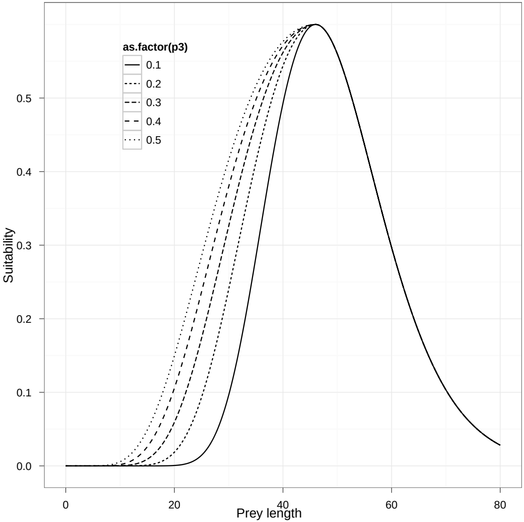


Suitability functions (continued)

Andersen $$S(L,l) = \begin{cases} p_0 + p_2 e^{-\frac{(\log\frac{L}{l} - p_1)^2}{p_4}} & \log\frac{L}{l} \leq p_1 \\ p_0 + p_2 e^{-\frac{(\log\frac{L}{l} - p_1)^2}{p_3}} & \log\frac{L}{l} > p_1 \end{cases}$$

Gamma $$S(L,l) = \left(\frac{l}{(\alpha-1)\beta\gamma}\right)^{(\alpha-1)} e^{\alpha-1-\frac{l}{\beta\gamma}}$$

Andersen


as.factor(p3)
0.1
0.2
0.3
0.4
0.5
Suitability
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
Prey length

Fleet operations

As noted earlier, fleets operations are considered to similar to predator consumption. There are, however, some differences in implementation as fleets do not have a maximum consumption or half feeding value.
When defining the fleet consumption the user can choose between 5 different types of fleet operations:

- The first two are **TotalFleet** and **NumberFleet** that harvest according to landed biomass and landed numbers using the formula:

$$C_s(l) = \frac{ES_s(l)N_{sl}W_{sl}}{\sum_{stocks}\sum_{lengths} S_s(l)N_{sl}W_{sl}}$$

  where E is the biomass/numbers.

Fleet operations (continued)

- The next two are **LinearFleet** and its multi stock extension **EffortFleet** that harvest according to fleet effort:

$$C_s(L) = E\Delta t q_s S_s(l) N_{sl} W_{sl}$$

  where E is the effort and q_s is the stock catch ability (i.e. preference). These fleet types are often used in future predictions.
- **QuotaFleet** is simple extension to LinearFleet that allows for the use of simple harvest control rules based on total biomass in future predictions.

Growth

Lengthgroup growth is implemented using a two step approach:

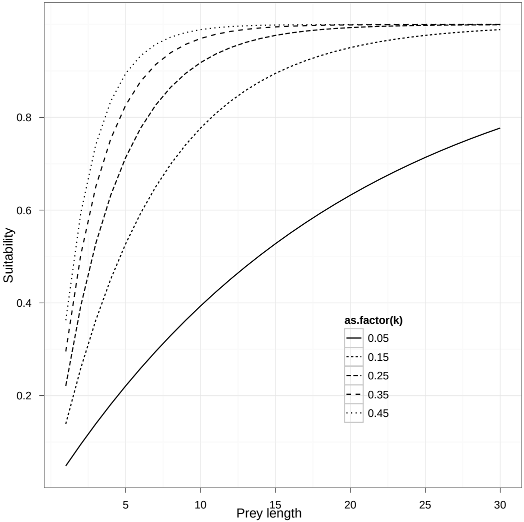
- Average lengthgroup growth is calculated according to a growth function.
- The number in the lengthgroup prior to the length update is dispersed around the average growth according the growth implementation function.

A number of growthfunctions are implemented in Gadget, most of which are variants of the Von Bertalanffy curve:

1. MULTSPEC growth function
2. WeightVB growth function
3. WeightJones growth function
4. WeightVBExpanded growth function
5. LengthVB growth function
6. LengthPower growth function
7. LengthVBSimple growth function

$$\Delta L_i = (q_0 - L_i)(1 - e^{-q_1 \Delta t})$$

Von Bertanlanffy growth update



The length distribution is updated using a beta-binomial distribution, i.e., the probability of growing λ length groups, given maximum length group growth n, is

$$P[\Delta l = \lambda] = \frac{\Gamma(n+1)}{\Gamma(n-\lambda+1)\Gamma(\lambda+1)} \frac{\Gamma(\alpha+\mathfrak{b})}{\Gamma(n+\alpha+\mathfrak{b})} \frac{\Gamma(n-\lambda+\mathfrak{b})}{\Gamma(\mathfrak{b})} \frac{\Gamma(\lambda+a)}{\Gamma(\alpha)} \quad (3)$$

with $\alpha = \frac{\mathfrak{b}\Delta L}{n-\Delta L}$ to preserve the mean length growth according to the growth update function and $\mathfrak{b}$ to control the spread of the distribution. Note that the expected value of ΔL should be taken into consideration when fixing n.

Other processes

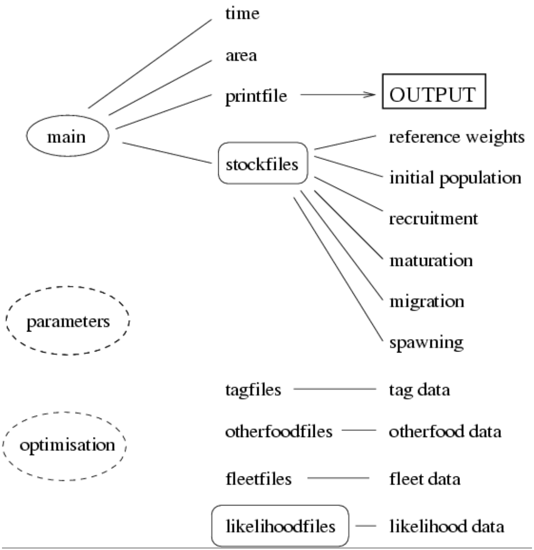
- **Natural mortality:** $N_{a+1} = N_a e^{-m_a \Delta t}$
- **Maturation:** based on length, age and weight according to:

$$P(l,a) = \frac{1}{1 + e^{-4\alpha(l-l_{50}) - 4\beta(a-a_{50}) - 4\gamma(k-k_{50})}}$$

- **Recruitment:** Works in the same way as the initial data.
- **Spawning:** renewals can also be formulated using one of four spawning functions:
  - Fecundity
  - SimpleSSB
  - Ricker
  - Beverton-Holt
- **Movement and straying:** Parts of one stock can be moved/transferred to another stock. Potential uses include maturity stages or stock migrations.
- **Tagging:** allows the use of tagging data to estimate stock movement.

Where are we now...

Gadget file structure



Gadget main-file

Main purpose to guide Gadget to all the info it needs:

```
; Main file for the TUSK model
timefile    TIME
areafile    AREA
printfile   PRINTFILE
[stock]
stockfiles       TUSK
[tagging]
[otherfood]
[fleet]
fleetfiles       FLEET
[likelihood]
likelihoodfiles  LIKELIHOOD
```

Gadget time-file

Gives the timeframe for `Gadget` and division of years into time-steps:

```
; Time file for the TUSK model
;
firstyear       1980
firststep       1
lastyear        2011
laststep        4
notimesteps     4 3 3 3 3
```

`notimesteps` refers to number of steps (4) and the length in months of each time step (3).

Gadget area-file

Specifies the area Gadget works on (Not important in our case):

```
; Area file for the tusk example
areas  1
size   200000
temperature
;
; year  step  area  temperature
1980 1 1 5
1980 2 1 5
1980 3 1 5
1980 4 1 5
1981 1 1 5
1981 2 1 5
.... to 2011s4
```

temperature does not matter for us at this stage.

Other input files

These are 3 different kinds of files:

- Aggregation files: Information on the scale calculations are performed.
- TimeVariable files: Define variables that can change over time (Selection, growth etc)
- ActionAtTime is a list of timesteps within a data file when a specified action takes place. For example printing. We will not go into this any further.

Aggregation files

Aggregation files define the scale of calculations:

```
; TUSK
; Length aggregation file - 1cm length groups
len-7cm   6.5   7.5
len-8cm   7.5   8.5
...
len-109cm 108.5 109.5
len-110cm 109.5 110.5

Another example
; TUSK
; Length aggregation file
len7-110cm   6.5   110.5
```

Same for age

TimeVariable files

Come in handy when we have reason to belive that something has changed during the time we are modelling.

Below is an example of estimating differrent selectivity in survey:

```
SurveySelection
nrofcoeff 0
data
; year  step  value
1970    1     (* 10 #IceSur70L50)
1990    1     (* 10 #IceSur90L50)
2000    1     (* 10 #IceSur00L50)
```

This is advanced stuff and will not be covered in any detail here.

Gadget stock-file

This is the main file for Gadget, and the most complicated one:

- Info on the stock, max/min age and length.
- Weight-length relationship
- Growth.
- Natural mortality
- Consumption of the stock and if the stock is a prey.
- Initial conditions
- Recruitment

Gadget stock-file (Part I)

Info on the stock, max/min age and length:

```
; TUSK stock file
;
stockname        tusk
livesonareas     1
minage           2
maxage           20
minlength        6.5
maxlength        110.5
dl               1
```

`dl` refers to the step size for the length groups.

Gadget stock-file (Part II)

Growth and Weight-length relationship:

```
refweightfile   Data/tuskrefw.dat
growthandeatlengths   AggFiles/len.agg
doesgrow              1
growthfunction  lengthvbsimple
; growthparameters     linf      k   a b
growthparameters     #Linf  ( * 0.001 #k) 0.00000659    3.01721
beta                  (* 50 #bbeta)
maxlengthgroupgrowth  15
```

beta is the 'spread' parameter that affects the transistion of growth in time. High values produce narrow distribution.
`maxlengthgroupgrowth` is the maximum number of length catagories a fish is permitted to grow in a single timestep in the model. Should be set several lenngth groups higher than can be expected to produce flexibility for the beta-binomial distribution.

refweightfile

This file as the name implies gives the model reference weights for a given length interval:

```
1.5 2.90516142097719e-05
2.5 0.000137589975722335
3.5 0.00038324128370853
4.5 0.000823685921604867
5.5 0.00151736328130283
6.5 0.00252330312113541
7.5 0.00390101992743393
8.5 0.00571044014214957
```

Note that this is in kg!!!

growthandeatlengths

Gives the length intervals the stock eats and growths at

```
; TUSK
; Length aggregation file - 1cm length groups
len-7cm   6.5   7.5
len-8cm   7.5   8.5
len-9cm   8.5   9.5
...
len-107cm 106.5 107.5
len-108cm 107.5 108.5
len-109cm 108.5 109.5
len-110cm 109.5 110.5
```

Sould be a fairly fine scale.

Gadget stock-file (Part III)

Natural mortality:

```
naturalmortality 0.15 0.15 0.15 0.15 0.15
0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15
0.15 0.15 0.15 0.15 0.15 0.15
```

A vector of the same length as the age-groups.

Gadget stock-file (Part IV)

Consumption of the stock and if the stock is a prey:

```
iseaten          1
preylengths      AggFiles/len.agg
doeseat          0
```

Tusk is eaten (by the fleet) but it only drinks water!

Gadget stock-file (Part V)

Initial conditions:

```
initialconditions
numbers
minage          3
maxage          20
minlength       6.5
maxlength       110.5
dl              1
initstockfile   InitFiles/tusk.init
doesmigrate     0
doesmature      0
doesmove        0
```

Normally the same as at the start of the stock file. Gadget may complain if this is different.

initstockfile

Initial conditions:

```
; initial stock file for the TUSK example
;age  area  agemult  areamult  mle    sd   cond
   3  1  (* 1000 #ic03)    1   25.3   3.656 1
   4  1  (* 1000 #ic04)    1   31.26  4.917 1
   5  1  (* 1000 #ic05)    1   36.68  5.400 1
   6  1  (* 1000 #ic06)    1   41.64  5.400 1
   7  1  (* 1000 #ic07)    1   46.17  5.400 1
   8  1  (* 1000 #ic08)    1   50.29  5.400 1
   9  1  (* 10 #ic09)      1   54.07  5.400 1
  10  1 (* 0.1 #ic09)      1   57.52  5.400 1
  11  1 (* 0.07 #ic09)     1   60.69  5.400 1
  12  1 (* 0.055 #ic09)    1   63.54  5.400 1
  13  1 (* 0.04 #ic09)     1   66.17  5.400 1
  14  1 (* 0.025 #ic09)    1   68.57  5.400 1
  15  1 (* 0.001 #ic09)    1   70.76  5.400 1
  16  1 (* 0.0007 #ic09)   1   72.76  5.400 1
  17  1 (* 0.0005 #ic09)   1   74.59  5.400 1
  18  1 (* 0.0001 #ic09)   1   76.26  5.400 1
  19  1 (* 0.00005 #ic09)  1   77.78  5.400 1
  20  1 (* 0.00005 #ic09)  1   79.18  5.400 1
```

Here Gadget will generate 10 million age 3 tusk multiplied with the ic03 parameter

Gadget stock-file (Part VI)

Recruitment:

```
; Recruitment
doesrenew          1
minlength          6.5
maxlength          35.5
normalparamfile    InitFiles/tusk.rec
doesspawn          0
```

normalparamfile

Recruitment:

```
; Recruitment data for the tusk example
; year  step  area  age  number                mle          stddev        alpha    beta
1980    1     1     2    (* 1000 #rec1980)   22#le.rec    3.8#sdev.rec  0.94e-5  3.078
1981    1     1     2    (* 1000 #rec1981)   22#le.rec    3.8#sdev.rec  0.94e-5  3.078
1982    1     1     2    (* 1000 #rec1982)   22#le.rec    3.8#sdev.rec  0.94e-5  3.078
1983    1     1     2    (* 1000 #rec1983)   22#le.rec    3.8#sdev.rec  0.94e-5  3.078
1984    1     1     2    (* 1000 #rec1984)   22#le.rec    3.8#sdev.rec 0.94e-5  3.078
...
2006    1     1     2    (* 1000 #rec2006)    22#le.rec    3.8#sdev.rec  0.94e-5  3.078
2007    1     1     2    (* 1000 #rec2007)    22#le.rec    3.8#sdev.rec  0.94e-5  3.078
2008    1     1     2    (* 1000 #rec2008)    22#le.rec    3.8#sdev.rec  0.94e-5  3.078
2009    1     1     2    (* 1000 #rec2009)    22#le.rec    3.8#sdev.rec  0.94e-5  3.078
2010    1     1     2    (* 500 #rec2007)    22#le.rec     3.8#sdev.rec  0.94e-5  3.078
```

Fleet Files

Defines the fleets in our model. In the simulation model we have only one called `comm` but tomorrow we will add another fleet called `survey`

```
[fleetcomponent]
linearfleet        comm
livesonareas      1
multiplicative    1.0
suitability
tusk      function ExpsuitfuncL50     #alphacomm #L50comm
amount            Data/fleet.data
```

In this case we have a `linearfleet` which catches according to F as opposed to tonnes. The fleet has the selection curve `ExpsuitfuncL50` which means it has a S-shaped selection

Parameter file (Part I)

Normally this is the result from a `Gadget` optimisation. But here we interact with this file in order to simulate various stuff. The file includes all the parameters we estimate in the model (well not really).
The first part is the growth (`Linf`, `k`, `bbeta`) and initial conditions (`ic03`, `ic04....ic09`).

```
switch value lower upper optimise
Linf         120        50       200         0
k       75.989866       0.1      1000        1
bbeta    0.24152015     0.001       15         1
ic03     1.1788271     0.001       15         1
ic04     1.1042464     0.001       15         1
ic05    0.63687283     0.001       25         1
ic06    0.10000007     0.001       15         0
ic07     1.7301384     1e-05       15         0
ic08    0.27719361     1e-05       10         0
ic09    0.08066132     1e-06       20         0
```

Note that not all the parameters are estimated!!!

Parameter file (Part II)

The second part is the estimates of recruitment.
Length and sd at recruitment and number of recruits each year.

```
rec1980     1.6242899    0.001       10        1
le.rec     13.814487         7       40        1
sdev.rec    4.3065463     0.01       15        1
rec1981     1.8567933    0.001       10        1
rec1982     1.4241382    0.001       10        1
rec1983     1.1947081    0.004       10        1
rec1984    0.76408298    0.004       10        1
...
rec2007     1.4690311    0.004       10        1
rec2008     1.2031934    0.004       10        1
rec2009    0.76175458    0.004       10        1
```

Parameter file (Part III)

The last part is the selection of the predators (fleet).

```
alphacomm   0.072713053   0.03   10   1
L50comm     46.721171     20     50   1
```

Likelihood File

The `likelihood`-file is the bridge between the model and the data.
We will look at the `likelihood`-file in much greater detail tomorrow.

Gadget print-file

Specifies the output Gadget generates:

```
; Print file for the TUSK model
;
[component]
type            stockstdprinter
stockname       tusk
printfile       out/tusk.std
yearsandsteps   all all
;
[component]
type           predatorpreyprinter
predatornames     comm
preynames         tusk
areaaggfile       AggFiles/allarea.agg
ageaggfile        AggFiles/age.agg
lenaggfile        AggFiles/alllen.agg
printfile         out/Catch-Longline
printatstart   0
yearsandsteps    all all
```

We will look at this in much greater detail tomorrow when we try to fit to data.

Running Gadget

Like stated above `Gadget` is run in two steps. Optimisation and simulation. Today we have covered the simulation. To make `Gadget` do a simulation one simply types at the command line:

```
gadget -s -i params.in
```

And `Gadget` will produce the output requested in the `printfile`.