HCS program for simulating harvest control rules. Program description and instructions for users

Version HCS 17_1
Manual revised October 2017.

Dankert W. Skagen
www.dwsk.net

1. Overview

The ***hcs*** program ('harvest control simulation') is a general purpose program for stochastic simulation of management decision rules (harvest control rules). Several versions have been distributed over time, this manual is for the version ***hcs17_1.***

The ***hcs*** program simulates rules for a single age disaggregated stock exploited by a single fleet. It can simulate a quite wide range of rules, and has numerous options to parametrise the rules. The underlying population model (operating model) has a wide range of options for recruitment, including regime shifts, and for variable and density dependent weights and maturities. Deriving data for decisions from the operating model is done with an algorithm that imitates the effect of some sources of errors in analytic stock assessments, but there is no full assessment in the loop. The program is run as a bootstrap with randomly drawn stochastic terms in each realization, and the distributions of results are derived as frequency distributions in the set of realizations.

hcs17_1 can scan automatically over ranges of harvest rule options and parameters, as well as some stochastic distribution parameters. The results are partly presented as annual means and percentiles, partly as detailed data for generating distributions, but also as assembly tables allowing performance measures to be compared across options. Since there is no full assessment in the loop, the program is fast - each scenario takes 10-30 seconds with 1000 iterations. This should make it a versatile tool for searching for optimal design of rules and for sensitivity testing.

1.1. Some version history

The ***hcs*** program originally was developed as an experimental tool, to study generic properties of harvest control rules, but has been adapted to assist in the design and evaluation of harvest control rules for real stocks. Therefore, it has options for generating input data either according to basic biological dynamics or by using data observed for real stocks, and it can generate initial stock numbers by 'priming' the stock with a fixed fishing mortality as an alternative to taking such data as input. It also has the option to take in stock-recruit data and initial stock numbers from externally generated bootstrapped assessments. The core of the program was developed for a generic study of fixed quota regimes in 2006[1]. Since then, it has been continuously extended and adapted, and has gradually developed into a general simulation tool. Since 2010, versions are numbered by year and version within the year. Hence version 17_1 is the first revision published in 2017. The versions are to some extent backwards compatible, but not fully.

---

1) Skagen , D.W. (2007)
Management strategies for reducing variation in annual yield: when can they work?
ICES Journal of Marine Science. 64: 698-701

Version 15_1 was a major revision. Large parts of the code was rewritten and better structured. The input was reorganized, the harvest rules that can be simulated were revised, some were removed and some new added. However, the results should be close to those with older versions under the same conditions and options, although not necessarily identical, because of some minor bug fixes. As far as this has been tested, the differences are minor.

Version 17_1 has some amendments and some bugs fixed:

New model options

- Autocorrelation can now be included in the noise terms in recruitment and in the year factor in the observation noise model, as one-step autoregressive functions.
- Recruitment variation can also be modelled as a longer autoregressive process.
- Depensatory stock recruitment function is no longer available.
- A decision basis can now also be calculated as a median over some previous years
- A new rule is introduced (rule 4) which is similar to Rule 1, but with a target F that is a linear function of a basis. The old rule 4 (smooth transitions at breakpoints) is no longer available.

New printout options

- A printout file has been made showing the distribution of realized autocorrelations in true SSB, observed SSB and recruitment, as well as the CV of the SSB in year 0, which often is considered when calibrating the assessment noise.
- A new performance measure ProcMSY is printed: The catch in percent of the yield per recruit at Fmax times mean recruitment.
- A separate output file is made with only the first overview table in the results file.
- Assembly tables can now be made for time intervals stated by the user, and for as many intervals as wished.

Bug fixes

- A bug that could create problems when using SSB in the last assessment year (or earlier) as decision basis was fixed.
- A bug that could create problems when combining several stabilizing rules was fixed
- Some minor bugs in output formats and headers were fixed.

Please note: HCS has evolved gradually over many years. New features have been added, and very few have been discarded. Therefore, some routines have not been tested for quite a number of years, and although they have not been touched, there is no guarantee that subsequent extensions of the program has not led to unintended side-effects. In particular, that goes for using list of bootstrap results for initial numbers and for sr-function parameters, and for the p-option for biological properties. Therefore, if these routines shall be used, be prepared that some de-bugging may be needed. In the manual, there is warnings when these routines are described.

1.2. Modifying the program

For each new stock or management plan, some modification of the program is almost unavoidable. Some parameters that are hard coded (number of iterations, number of years, maximum number of ages) can be changed very easily in the top of the include file *hcscom17_1.i* that is part of the program code. Just change the number in question there and compile.

Other modifications may be more challenging. The program is quite modular, and some modifications (but not all) should be relatively easy to write, but beware that changes may have side-effects that can be difficult to trace. As the program has evolved, some modularity has been lost, in particular with respect to exchange of data between parts of the program. Hence, when modifying the program, checking that variables get values is strongly recommended. A scratch file 'kladd' is routinely opened, and convenient for dumping intermediate values for control. In the standard version of the program, the file is opened, but not written to. Introducing new rules can be relatively simple, depending on what the changes are. Modifying the program requires insight in Fortran77, and that a compiler is available.

1.3. Input-output

All input and output is through ascii-files, with hard coded file names. There are two standard input files called ***bio.inn*** and ***opt.inn***, and three optional input files (***canum.inn, srspec and nin.xxx***- the actual suffix indicates the format - see the input section. There is a wide range of output options. Which files to produce is specified in the ***opt.inn*** file.

Most input files were changed in the ***15_1*** revision. Both in the input file ***bio.inn*** and the options file ***opt.inn*** the lay-out were new. The ***canum.inn*** file is unchanged. The ***nin.xxx*** files are unchanged, but one optional lay out (nin.num) has been removed and another (nin.lst) added. The optional file with stock-recruit parameters has a completely new lay-out. The changes in version 17_1 are minor. The opt.inn and bio.inn files are almost backwards compatible, the changes are outlined in Section 3.2. The .nin files and canum.inn and srspec file formats are unchanged.

The format of the input files is described in detail in Section 3.2.

1.4. Distribution

There is no special installation procedure needed. ***hcs17_1*** is distributed as a zip-file, which contains the code (all in a single file ***hcs17_1.f*** + an include file ***hcscom17_1.i***), this manual, the ***hcs17_1 .exe*** file for running under Windows and a set of input files that can serve as a template. The zip-file can be downloaded from my home page www.dwsk.net, or by contacting me on dankert@dwsk.net. For running under Linux or UNIX, the program has to be compiled which can be done with any Fortran77 compiler. I have used the ***gfortran*** compiler from GNU for both Linux and Windows, which can be downloaded free on the internet (see e.g. http://gcc.gnu.org/wiki/GFortran).

Two words of caution, though. First: Common variables are used extensively. They are assembled in the ***hcscom17_1.i*** - file which is included in most subroutines. The way the common variables allocate memory may vary between compilers, and sometimes unexpected results appear if the sequence of small and large common variables is not what the compiles expects. I have encountered such problems with ***hcs*** previously, but the way it is set up now seems to work well. Second: Slightly different results have been observed with executables generated with different compilers. I have not tried to trace the origin of these differences, the suspicion goes in the direction of the random number generator and the representation of double precision numbers.

No special files or libraries are normally needed to compile or run the program, but the include file ***hcscomm17_1.i*** has to be present along with the code file. Running under some variants of Windows requires the presence of a file *libgcc_s_dw2-1.dll* which is included in the distribution of the program.

The program is distributed according to the open source principles. Hence it is distributed free of charge, the code is distributed with the program and it is free to use for everyone. It has been tested quite extensively, but there is no guarantee that it is completely free of bugs, and the author cannot take any responsibility for consequences of program bugs, misunderstandings when using the program or any other errors. If results are published, reference to the source would be appreciated, and if modifications are made by future users, that should be clearly stated, the modified program should be re-named and there should be reference to the original.

1.5. Running the program

Before running the program, input files must be prepared. Then the program can be run as further described in Section 3.1.2. All communications is trough the input and output files. Both have hard-coded names and old output files are over-written without warning.

1.6. Organizing of the manual

Section 2 is a complete documentation of all algorithms and models. Section 3 on input-output describes how to specify both the options to be used and actual values for parameters, as formal descriptions of file formats and contents. There are some cross references between Sections 2 and 3. Section 4 contains some practical advise. The

manual is not intended as a textbook in management simulation - it is assumed that the user is familiar with that field.

2. Models and options.

The program is designed the standard way for harvest rule simulation programs. It consists of a population (operating) model that generates yearly true stock numbers at age, an observation ('assessment') model that transfers the stock numbers into noisy, 'observed' numbers, decision rules by which a TAC (Total Allowable Catch) is derived according to the observed stock (projected forward if relevant) and an implementation model that translates the TAC into actual removals. These removals are then input to the population model for the next time step. The outline is shown in Figure 2.1 and the components are further described below.

Please note the terminology here. The term 'population model' is the forward projection of the true stock, and corresponds roughly to some usages of the term 'operating model'. The term 'observation model' here is used for the whole process that leads from the true stock numbers to the apparent stock numbers that are visible for decision makers, including an imitation of an assessment and a projection. The decision rules include harvest rules and the data that go into the harvest rule (called 'basis' here). The final outcome of the decision process is always a TAC. The term 'implementation model' is used for the translation of the decided TAC to actual removals in numbers at age, which may include noise and bias.

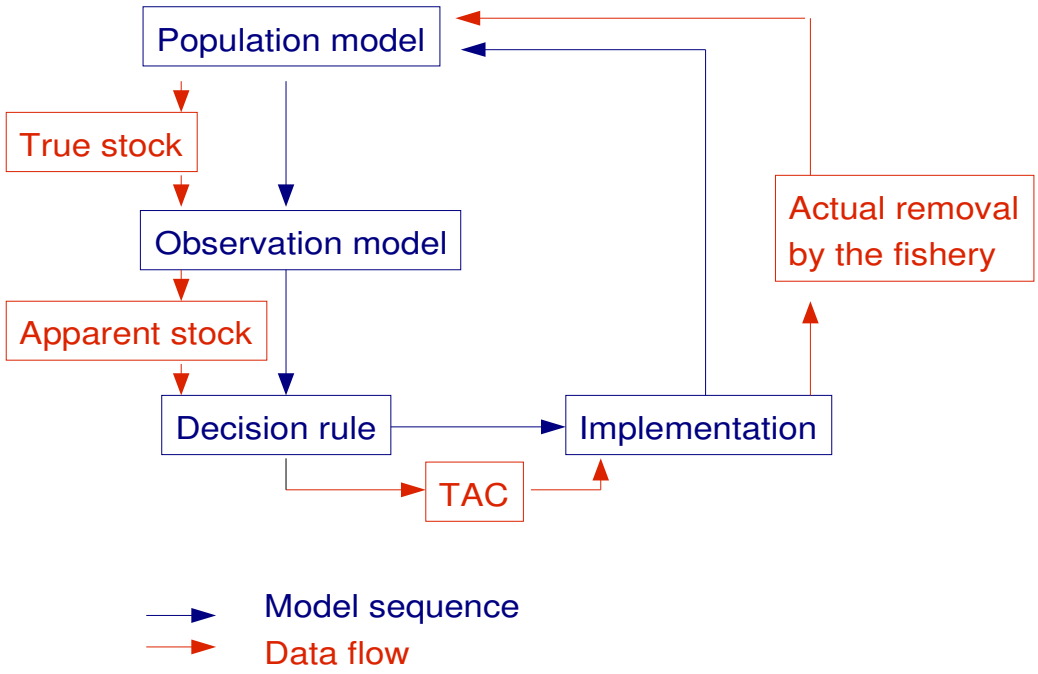

Figure 2.1. *Outline of the simulation loop in the hcs programs.*

Note on timing.
The loop outlined above is a year loop. A new year *y* starts with true numbers at the start of the year, which was the last thing that was generated in the previous year. Once the true numbers are in place, the observed numbers for year *y* can be generated. Then a TAC for year *y* is decided, the stock is projected forward through year *y* with the implemented removals and is ready for the next year.

When observation numbers are derived, historical stock numbers are recalculated, as it would be done in an ordinary assessment and the assessed numbers are kept for later use. The assessment backwards in time is imitated by doing a VPA calculation with noisy catch data and a model for generating retrospective error. The observation data are represented as data for a year *y1* as seen in a year *y2; y1 ≤ y2*. The observation model data are available in the loop once the true stock numbers are established. However, the decision process can use data from the past as relevant, and it is not bound to the standard assessment year - intermediate year - TAC year scheme.

In a normal procedure, the TAC for year *y* is set according to an assessment done some previous year, called *y-lastassm*, which normally will be *y-1*. The TAC is derived by applying a measure of exploitation (for example a fishing mortality) to the stock numbers expected to be present at the start of the TAC year. If that does not coincide with the end of the assessment year (but for example after an intermediate year) these stock numbers are obtained by projecting from the observation model numbers for year *y-lastassm*, as seen in the year. *y-lastassm*.

The measure of exploitation is decided according to some indicator (called basis here) of the state of the stock at some time or time period. The time of the basis is independent of the timing of the stock numbers used for calculating the TAC. For example, the basis may be the average SSB over the years *y-3* to *y-1* relative to the TAC year *y*. An F is decided according to that basis, and applied to stock numbers projected forward from assessed numbers representing year *y-1*, as seen in year *y-1*.

2.1 Population model:

2.1.1.Conceptual alternatives.

The population model represents the stock that we want to manage. It is an artificial stock, that we may examine as such on generic terms or take as a best possible representation of a particular stock. The program is run as a bootstrap, where each iteration is one example of what the stock might be and how it might develop over time.

It is common to think of the modelled stock as one stock with specified, but uncertain properties. We specify the stock by its properties and uncertainties and admit variability over time. The model translates the uncertain properties into differences between trajectories that gives us uncertainties and risks in the outcome.

An alternative would be to regard the population model as representing a collection of stocks. The principle underlying the specification of uncertainties should then be that the collection should represent a plausible range of properties. This alternative could be relevant if the stock that we want to examine is poorly known, and we like to ensure that our harvest rule works for all stocks with properties within the plausible range. When the stochastic range is taken to represent the plausible range of properties for a stock, the risks that are calculated get a slightly different meaning. Rather than representing the probability that something unwanted will happen, it will be the fraction of the plausible range that does not behave as wanted.

When we specify stock properties, they will typically be derived from known or estimated data related to the stock, like weights and maturities as observed, recruitments according to an analytic assessment etc. For more generic studies, it could be more logical to create artificial stock properties, rather than taking one example stock as the archetype representing a whole category of stocks. HCS has facilities for that, where life history parameters are derived from a small number of parameters: Growth rate and Linf, parameters in a stock-recruit relation and a logistic selection in the fishery.

The population model in HCS17_1 is primarily directed towards the single stock way of thinking. The 'standard' option is to specify parameters in population-submodels on the input file. The alternative would be to specify such parameters on separate files, with one set of parameters for each iteration. For the recruitment, which typically is the most important source of diversity in this context, HCS has an option to use a collection of stock recruit parameters (both function, function parameters and error distribution parameters) read from a file (*srspec* - see Section 3.2.5) rather than drawing recruitments around one specified function. The idea stems from a paper by Simmonds & al[2] There is also an option to read individual initial numbers from a file, see Section 3.2.3.

2 Simmonds, E. J, Campbell, A, Skagen, D, Roel, B.A and Kelly, C.J. (2011).
Development of a stock–recruit model for simulating stock dynamics for uncertain situations: the example of Northeast Atlantic mackerel (Scomber scombrus)
ICES Journal of Marine Science 68(5):848-859.

2.1.2. Model overview.

The population model projects a vector *N(a,y)* of stock numbers at age *a* forwards in yearly time steps *y* with a given fishing mortality *F(a,y)* and natural mortality *M(a,y)*:

$N(a+1,y+1)=N(a,y)*exp(-M(a,y)-F(a,y))$.

The oldest age *A* is a plus-group, modelled as a dynamic pool:

$N(A,y+1)=N(A,y)*exp(-M(A,y)-F(A,y))+N(A-1,y)*exp(-M(A-1,y)-F(A-1,y))$.

The fishing mortality is separable:

$F(a,y) = Fy(y)*Sel(a)$.

where *Fy(y)* is a year factor and *Sel(a)* is the selection at age.

Natural mortality at age is at present entered as constant over time, i.e. as *M(a).* A plan to represent it as disaggregated by age and year (*M(a,y)),* still has to be implemented.

The fishing mortality is derived by exposing the stock to the decided catches (as implemented with error). Accordingly, the realized fishing mortality may have a selection component that differs from that used in the decision process, as described in Section 2.2.4..

The initial *N*-values at the start of year 0 are stochastic, to represent what is regarded as a plausible range of realities. They are obtained either by priming the population with a fixed *F* and stochastic recruitments and weights, or by taking numbers from a file. When taking numbers from a file, there are several options for making the numbers stochastically variable, see Section 2.2.5

Each year, a new year class is entered at the youngest age, according to a stock-recruit model. There are several options for stock-recruit models, as described in Section 2.2.3. If the youngest (recruiting) age is greater than 0, recruits are still entered at age 0, but are projected forward with zero mortality and weight until they reach recruiting age.

The numbers are converted to biomasses by taking the sum pf products over ages of stock numbers, the relevant weights and if relevant, proportions mature valid for the year in question.

Weights and maturities at age can be entered (see section 3.2.1) as input numbers (***i-option***), or derived according to a von Bertalanffy growth model and a logistic maturity ogive (***p-option)***. Basically, they are fixed numbers, but there are options to let them vary from year to year. This variation includes dependency on the total biomass (TSB) in the year before, on year class strength, periodic fluctuations and stochastic variation with normal distributions, as described in Sections 2.2.1 and 2.2.2. below.

2.1.3. Note on total biomasses:

HCS operates with two kinds of total biomass:

- **TSB** from a lowest age as specified in the ***bio.inn*** file and using stock weights. This is the general TSB, that is reported in the results, and used for density dependence and other biological relations.

- **B+,** from a lowest age specified in the ***opt.inn*** file, using catch weights. This is used in harvest rules where a decided harvest rate is converted to a TAC. Likewise, B+ is imitated when a biomass survey is used as decision basis in a harvest rule.

2.2. Details of the population model

2.2.1 Variable growth, including density dependence.

Weight of individuals in the catch (catch weight) and in the stock (stock weight) are handled separately, but are partly linked, as described below. The catch weights are used to convert catches in numbers to catches in weight and vice versa, while stock weights are used to convert stock numbers to biomasses. The growth rates are not modelled explicitly, only the weights themselves.

The default is weights and maturities at age being constant over time. However, the following options are available for the annual weights and maturities:

- Dependence on the total biomass
- Dependence on year class strength
- Periodic trends
- Random variations

All the parameters in these functions, including the reference (standard) weights, are specified in the ***bio.inn*** file.

The rules are similar for catch weights and stock weights, here commonly termed *w*. The starting point is the deterministic reference weights at age *w0*. The final weights are then calculated as

$w(a,y) = w0(a)*TSB\_factor(y)*Ycl\_factor(y-a)*Trend\_factor(y)*Random\_factor(a,y))$

Each of the factors is 1 if no modification takes place.

The ***TSB_factor*** modifies the weight according to the current stock abundance, expressed as TSB. It applies in a year to all ages. It is calculated as a linear function of the total biomass *TSB* in the previous year. Following Kovalev and Bogstad[3] (2005), it is expressed by a slope α_{TSB} and a reference TSB termed *B0*, at which the factor is 1.

$TSB\_factor = 1+\alpha_{TSB}*(B0-TSB)/B0$

The α_{TSB} is defined for each age separately, but applies in all years.

Considering *w0(a)*TSB_factor(y)* as a model for density dependence has the drawback that the biomass in one year becomes inversely related to the biomass the previous year. Suppose the biomass was high in one year. That will lead to low weights the next year and a correspondingly low biomass. That low biomass leads to high weights, and a high biomass the following year. This switching between low and high biomass is not what we see in the real world, and is probably caused by using the TSB as a proxy for the depletion of the food resources for the stock.

An alternative formulation might be to model the growth increment rather than weights themselves. With von Beralanffy growth, the yearly increment is $\Delta w = (W_\infty - w)*(1-exp(-k))$ which leads to Δw as a linear function of *w*.

$w(a,y) = w(a-1,y-1)*exp(-k) + W_\infty*(1-exp(-k))$

We may let *k* be a truncated linear function of TSB and enter W_∞ as a parameter. The slope and intercept for *k* may be entered as age-dependent parameters. This option has been coded, but is commented out at present. Including it would require changes in the bio.inn file.

The year class factor (Ycl_factor) modifies the weight according to the *strength* of the year class. The *Ycl_factor* follows the year class at all ages. It depends on the recruitment value *R* of the year class, and the reference is *R0*

$Ycl\_factor\_temp = 1+\alpha_{Ycl}*(R0-R)/R0$

It is truncated at a lower and upper bound for the factor.

The ***Trend factor*** is a cosine function that applies to all ages in a year. It is specified by an amplitude, a period and a phase:

---

3). Y. Kovalev & B. Bogstad 2005:
Evaluation of maximum long-term yield for North-East Arctic cod. Proceedings of the 11th Russian-Norwegian Symposium , Murmansk, 15-17 August 2005, PINRO Press. Murmansk 2005 .

*Trend factor(year) =1+ ampl*cos(2π*(year-phase) /period)*

The amplitude should be < 1, to avoid negative weights.

The ***Random_factor(a,y)*** is the product of two normally distributed random terms:

1. A random year factor, common to all ages.
2. An random age factor, specific for each individual age and separate for each of the weights and the maturities.

Both the year factor and age factor components of the random factor are normally distributed with mean 1 and a sigma and are updated in each simulation year, independently of the previous values. The sigma for the year factor is common to all years. The sigma for the age factor is defined for each age separately, but valid in all years.

Truncation. The final weights are truncated by a lower and an upper bound for each age. In addition, the year class factor is truncated, as noted above.

Overview of parameters: The algorithm described above applies to both weights in the catch and weights in the stock. The parameters

- *w0*,
- Upper and lower bound for weights at age
- α*TSB*,
- sigma for the random age factor

are specified by age and separately for stock weights and catch weights.

All other parameters are common to all ages and to stock and catch weights. These are:

- B0: TSB corresponding to reference values
- R0: Recruitment of reference year class
- Lower and Upper values and Alfa for year class factor.
- Sigma for random year factor noise
- Amplitude, period and phase for periodic trends.

The *Ycl_factor*, the *Trend_factor* and the year component of the *Random factor* are common to both catch weights and stock weights in a year, and to all ages that year. This is to ensure that catch weights and stock weights change in some synchrony.

2.2.2 Variable maturity, including density dependence

The algorithm for weights is not suited for maturity (fraction mature at age). All fish should mature sooner or later, so functions for density dependence that cannot reach one at very high age are not feasible. Instead, we assume that the effect of a large TSB is to delay maturation. A direct modelling of maturation, involving the probability that a fish that is still not mature will mature in the coming year, can become rather complex. A more simplistic approach is taken here, where the fraction mature as a function of age is just moved along the x-axis. For example, if TSB is large and the maturation is delayed, the proportion mature at age *a* becomes the proportion that would be normal at a lower age. The move is expressed as a delay factor, such that the proportion mature at age *a* is taken as

Reference proportion mature at the age (a / Delay_factor.).

The delay factor is the product of a *TSB-sensitive term*, a year class *Ycl_factor* and *Random* multipliers as described for the weights above. The sensitivity of the age delay to TSB is expressed by a sensitivity factor α_{delay}: Then

Delay_factor =
$(1+\alpha_{delay}$(TSB-TSBref)/TSBref)*Ycl_factor(y-a)*Trend_factor(y)*Random_factor(y,a))*

The reference maturity ogive is input or derived at age, and connected with straight lines between the ages.

The α_{delay} and the age component in the *Random_factor(y,a)* are specific for the maturation, the other factors that go into the calculation of the delay factor are shared with the weights calculation. Hence, maturity and weights are to some extent altered synchronously. The usage of the delay factor implies that maturation age is inversely related to growth, i.e. rapid growth leads to early maturation. Further, there is a minimum and maximum proportion mature at each age, specified in the ***bio.inn*** file.

Updating of weights and maturity takes place when projecting the true stock one year ahead. The weights and maturities used in the decision process are always the most recently derived true values, which also are assumed valid for the whole projection period if the observed stock is projected forwards as part of the decision process.

The weights and maturities are specified on the ***bio.inn*** file. With the p-option for specification of biology, the age specific information is generated internally. The density dependence is still specified for the year class factor. Specifications for alpha of the TSB factor and for upper and lower bounds are common to all ages.

It is strongly recommended to check that the specification of the weights and maturities actually leads to the expected results. A printout option is available that presents the annual weights and maturities at age for each of the first 10 iterations in the simulation (print option 25, file Wema).

2.2.3. Recruitment models.

The recruitments in HCS are generated by applying stochastic multipliers to deterministic stock-recruit functions. Hence, the specification of recruitment models includes parameters in deterministic relations (stock-recruit functions) and parameters for stochastic terms (distribution types, sigmas, autocorrelations, autoregressive models and truncation rules). HCS allows for multiple recruitment regimes, with various rules and for shifting between them at fixed or random times. On top of all this, HCS also allows for exceptional year classes ('spasmodic recruitments') that are implemented as a multiplier to the recruitment determined by the models so far.

The specifications of recruitment models can be entered on the bio.inn input file, of read from a file prepared in advance. This file will specify a full set of parameters for each regime for each iteration. In the program, a list of regime shifts is set up for each iteration according to the common specification of regime shifts. Then, for each iteration, a list of parameters is prepared for each year according to the decided regime and the parameters belonging to that regime. When drawing future recruitment in the forward projection, the necessary parameters are taken from this list.

2.2.3.1. Regimes and regime shifts.

In ***hcs***, recruitment does not have to have stationary statistical properties, but can change between 'regimes'. This is a pragmatic approach to the problem of managing stocks with changing recruitment, without entering into debates about the nature of regime shifts. In ***hcs***, a recruitment regime is defined as a set of recruitment function specifications, and it is possible to switch between such sets, either randomly or in a fixed schedule.

There are two options for deciding on timing of regime shifts:

- A fixed schedule of years and regimes: The years where shifts take place are specified as input. The regimes will apply in the order they are specified on the input file. The number of shifts must be consistent with the number of regimes, i.e. number of shifts = number of regimes -1. This option is suitable for examining the robustness of a rule to shifts in the recruitment regime in a controlled way.
- Random years and regimes: The years are drawn randomly, according to the geometric distribution of 'time x to next success' in a Bernoulli trial:
  *Prob(x years) = 1/mean * (1-1/mean)* ^{x}, where the mean is specified.
  At each shift, the regime to be applied is drawn randomly amongst those available. This option is suitable for examining the effect of uncertainty with regard to future recruitment regimes.

In both cases, the simulation will always start with the first regime that is specified. The set of recruitment parameters in this regime should be consistent with the initial stock numbers.

In ***hcs***, recruitment is always calculated at age 0. If recruiting age is higher, the year classes are just carried forward with no mortality or weight until they reach maturation age. This implies that when a regime shift takes place in one year, it will not materialize in the recruitment until those born after the shift have reached recruitment age, which can be several years later.

To use the same regime throughout, set the number of regimes to 1 and specify that regime.

2.2.3.2. Recruitment function specifications for each regime.

Each recruitment regime is characterized by a set of specifications. These specifications define the recruitment as a product of several multipliers:

- A deterministic stock recruitment function *R0(SSB)*, which can be Hockey stick (in which constant recruitment is a special case), Beverton-Holt or Ricker.
- A periodic component multiplier *period_factor*, as a cosine function with amplitude, period and phase.
- Occasional strong year classes (sometimes termed spasmodic recruitment), as a stochastic multiplier *spasm_factor* in selected years

The deterministic (mean) recruitment is the product

*Rmean(y)=R0(SSB(y))*period_factor(y)*spasm_factor(y)*

Random noise $\xi noise$ is generated as described below.

Each years recruitment is the product:
$R(y) = Rmean(y)*\xi noise(y)$

Each of the components are described in detail below:

The ***Deterministic S-R functions*** are (with parameters *a_par* and *b_par*)

1. Hockey - stick: *S>b_par: R = a_par*
   *S<b_par: R = a-par*SSB/b_par*
2. Beverton-Holt: *R = a_par*SSB/(SSB+b_par)*
3. Ricker: *R = a_par*SSB/b_par*exp(1-SSB/b_par)*

Note the form of the Ricker function, the *a* parameter is the maximum recruitment and the *b* parameter is the biomass at the maximum recruitment.

Normally, the parameters themselves and the dispersion parameters of the stochastic elements have fixed values for each regime. Alternatively, individual parameters for each iteration can be read from a file (see Section 3.2.5).

Which SSB to use in the S-R function is sometimes a dilemma. The convention in ***hcs*** is that recruitment is recorded at the start of the year at age 0, even though the fish is actually born later in the year. If spawning takes place later in the year, the SSB, and accordingly the recruitment, may depend on the exploitation in the year. Therefore, recruitment in year *y+1* cannot be determined before the TAC for year *y+1* has been implemented, and that TAC may include fish at age 0. In previous versions of ***hcs***, the solution was to use the SSB from year *y* in the S-R function when deciding the recruitment in year *y+1*. Since version ***hcs15_1*** this has been changed. A delay between spawning and birth at age 0 is introduced in the ***bio.inn*** file. Normally, that delay will be 0, but for some autumn spawners, it will be 1 (age 0 in year y is generated by the SSB in year y-1). When the delay is 0, a provisional SSB for year *y+1* is calculated once the true stock numbers for year *y+1* have been calculated. Obviously, the SSB can be calculated without the contribution from the 0-group since fish does not spawn before it is born. If spawning takes place later than 1. January, the numbers in year *y+1* are reduced assuming the same fishing mortality as in year *y*. If there is a delay, i.e. the spawning delay is >0, the true SSB value in year (*y+1-delay*) are used for calculating the recruitment in year *y+1*.

Periodic fluctuations in the recruitment
Periodic fluctuations is included through a multiplier $\xi period$, specified by an amplitude, period and phase.

$$period\_factor(Year) = 1.0 + Amplitude * \cos(2\pi * (Year - Phase)/Period)$$

The amplitude should be in the interval $0.0 \leq Amplitude \leq 1.0$ to avoid negative values of *period_factor,* and represents the amplitude in relative terms. The period is in years. The phase is a year (relative to the starting year of the run) where the periodic function has its maximum. The option can be used both to include periodic fluctuations in the recruitment, and to generate a trend. To make a downward trend, the period should be about the double of the time span for the simulations, and the phase about -1/4 of the period. The recruitment will then start at the standard level specified in the recruitment function, and be reduced to a fraction *1-Amplitude* of the standard level at the end of the simulation period.

To exclude periodicities, set the amplitude to 0. The period should not be 0.

Spasmodic recruitment
This is an option to include occasional strong year classes, that come with more or less regular intervals. If asked for, the program sets up - for each iteration - a list of years where the recruitment is multiplied with a specified fixed magnitude factor *spasm_factor*. This is done by drawing random intervals between years with high recruitment, specified by a *mean* and optionally a *sigma*. The first interval starts in the first 'spasmodic' year, which may be a year in the past (stated relative to the starting year). Note that although the multiplier *spasm_factor* has a fixed value, it applies to recruitments that are stochastic, so the spasmodic year classes will not all be equal.

There are three options for deriving random intervals:

1. According to a log-normal distribution with specified *mean* and *sigma*.
2. According to the geometric distribution of 'time x to next success' in a Bernoulli trial: $Prob(x\ years) = 1/mean * (1-1/mean)^{x}$
3. According to a flat (boxcar) distribution, with specified *mean* and relative *range*, centered at mean and covering *(mean-range*mean, mean+range*mean)*

The first and third options are best suited to the situation where the intervals are supposed to be more or less regular, in particular if the purpose is to investigate how dependent management is on the regular occurrence of big year classes. The second is the right formulation when examining the effect of the uncertainty caused by sporadic year classes occurring as random events with a probability of *1/mean.*

To exclude spasmodic recruitment, set the magnitude factor to 1.0.

Random noise and autocorrelation
Noise around the deterministic stock recruit function ($\xi noise$) can be normal or log-normal, and can include autocorrelation generated by a one-step autoregressive function with coefficient ρ, as well as an autoregressive process with several terms.

The $\xi noise$ term is generated in a stepwise process: First, a random number x with standard normal distribution is drawn and a normally distributed random number $\xi temp1 = x * ps$ is derived. Here, *ps* is a dispersion parameter, which is used as a CV with the normal distribution and a sigma with the log-normal distribution.

Autocorrelation can be imitated as a one step autoregressive process with coefficient ρ applied to the random term $\xi temp1$ as described by Wiedenmann & al (2015)[4] , leading to a new random number $\xi temp2$:

$$\xi temp2(year) = \xi temp2(year-1) * \rho + \xi temp1(year) * (1 - \rho^2)^{1/2}$$

Autocorrelation is ignored by setting $\rho = 0$.

4 Wiedenmann, J, Wilberg, M.J., Sylvia, A nd Miller, T.J. (2015). Autocorrelated error in stock assessment estimates. Implications for management strategy evaluation. Fisheries Research 172: 325 - 334.

Autoregressive variation
Another way of introducing structured variations around the stock-recruit function is to regard it as an autoregressive (AR) process. An AR process can be designed in several ways. Here (as in the observation model) it is specified by the number *n* of coefficients and their values. Another way, which is not implemented in ***hcs*** at present, would be to design it to have desired spectral properties by utilizing that there is a close link between coefficients of an autoregressive process and the frequency spectrum of the time series.

Denoting the coefficients $\alpha_0,\ldots,\alpha_n$, the model is

$$\xi temp3(year) = \alpha_1 * \xi temp3(year-1) + \alpha_2 * \xi temp3(year-2) + \ldots\ldots\ldots + \alpha_n * \xi temp3(year-n) + \xi temp2(year)$$

To ignore the AR component, set the number of coefficients to 0.

Then the final noise multiplier is derived as:

1. Normal distribution: $\xi noise = (1+\xi temp3)$
2. Lognormal distribution: $\xi noise = exp(\xi temp3 - ps^2/2.0)$

Truncation
The random noise multiplier $\xi noise$ can be truncated, to avoid randomly drawn recruitments outside what may be considered as a realistic range. There are two options for how the truncation is done, either on the primary standard normal number *x* or on the final number $\xi noise$. For each option there is a lower and upper bound.

The rule is to draw a new random number if:
Option 1: $x > trunc\_upper$ or $x < trunc\_lower$
Option 2: $\xi noise > trunc\_upper$ or $\xi noise < trunc\_lower$

Note that applying asymmetric constraints will alter the mean recruitment and that both this and the introduction of all the variants of random noise means that the distribution of $\xi noise$ is no longer normal (or lognormal),. ***Therefore, it is strongly recommended to inspect the distribution of actual recruitments (which can be obtained with printout option 20) and compare it with the recruitment distribution you want to imitate.*** Another check can be obtained by comparing the output from the Y/R calculations with the original stock-recruit relation (see Section 3.4)

All the options described above can be combined, for example a trend with sporadic strong year classes, and autoregressive noise on top of that. The full set of options is specified separately for each regime, hence it is for example fully possible to have one regime with spasmodic recruitments and another without. However, combining the autoregressive and the autocorrelation mechanisms may be somewhat disputable from a statistical point of view. Also, the autocorrelation generated by the autocorrelation generating process is a stochastic variable, and it will often be far away from the intended value, in particular if the time series is short. A printout routine (printout option 13) provides the sample autocorrelation of the recruitments for each iteration together with some other model performance parameters.

2.2.4. Selection at age.

A standard selection at age is entered as relative F at age. The selection is normalized so that the *Fy* is scaled as the average *F* over a standard range of ages.

When implementing a TAC, the resulting noisy catch numbers at age will lead to a different (realized) selection, which is what the operating model will use to reduce true stock numbers according to mortality. The original standard selection is used everywhere else, including in the decision model and there is no options to change it over time.

2.2.5 Natural mortality

Natural mortality is at present assumed to be constant over time but variable with age. There are some plans for options for letting natural mortality vary over time, where a number of regimes can be specified, with shifts in specified or random years, linked to shifts in growth-maturity regimes. The purpose is mostly to allow examining the robustness of a rule to changes in natural mortality.

2.2.6 Initial numbers

These are the stock numbers at age at the start of year 0, which is the start of the simulation. When working with a specific stock, these numbers will typically be taken from the last assessment of the stock, and be similar to those used for the intermediate year in a short term prediction. Alternatively, HCS can provide initial numbers by running the stock to a stochastic equilibrium with a fixed fishing mortality. This is referred to as ***priming*** with fixed F.

When numbers from an assessment are used, there may be uncertainty to these numbers. This uncertainty should be regarded as representing the plausible range of values for the true present state, which conceptually is different from observation and implementation error. However, many analysts prefer to let the range of plausible initial numbers be represented by the uncertainty in the present assessment. A fair assumption about the plausible range of initial numbers may be that the uncertainty in initial numbers according to the last assessment, and the uncertainty in future assessments is the same. This is option 3 below.

Altogether, HCS has 4 options for specifying initial numbers:

1. Priming. Build a population by running the population model with a fixed F and deterministic recruitment at the value of the stock-recruit function when SSB equals the b-parameter, and then with a stochastic recruitment for a number of years that is twice the number of ages. The fixed F is implemented without error, and the replicas vary due to variation in recruitment, growth and maturity.
2. Numbers at age are entered together with a matrix with variances and correlations. In the matrix, the diagonal terms are covariances on the log scale, and the other entries are correlation coefficients. The vector of initial numbers in the model are drawn assuming a lognormal distribution where the expectation is the entered numbers and with the specified variance-covariance matrix.
3. Only numbers at age are entered, and the observation model is used to draw random numbers for each bootstrap replica. In this case, any bias in the observation model is reversed when using the obs. model for this purpose. For example, if we think the stock is underestimated, the initial numbers should be higher than the ones coming from the assessment. Later on, when the obs. model is used in the decision process, the perceived stock numbers should be lower than the true ones.
4. Sets of initial numbers at age are entered for each bootstrap replica. These sets must be generated externally. To link them to recruitment parameter sets generated a similar way, just note that both files will be read sequentially, one line at each replica. Please note that this option has not been properly controlled in recent versions.

The option to be used is linked to the format of the nin.xxx file, where the actual suffix indicates the format (see Section 3.2.3.

2.3. Observation (assessment) model:

This model transforms the vector of true *N*-values at age to observed *N*-values at age. Note that the term observations here refers to the stock numbers at age that typically are estimated in an assessment. The usage of the term is often different in other contexts. The observations model does not perform a full analytic stock assessment, but is a short-cut that may be regarded as a means to mimic the behaviour of a full assessment. Rather than generating noisy catches and survey data from the true population (which in itself is a challenge) and perform a full analytic assessment, the assessment is short cut by generating its presumed output directly, which drastically reduces computation time. The procedure in ***hcs*** attempts to some extent to reproduce the effect of structured noise in assessment input data. Hence, it goes far beyond just applying random noise to the true values, and even further beyond just putting some noise on the SSB values. In particular, the effect of noise in the tuning data (surveys or CPUE), that will fade away over a number of years, is mimicked, and if the history is needed in the decision process, it will come from a VPA which is updated each year. There also is an option to include

autocorrelation in the observation noise, which some users prefer.

The starting point is true stock numbers at the start of a reference year (the 'assessment year', typically the year before the TAC year). These numbers are modified by stochastic multipliers that are derived as an autoregressive process along the year classes, described below, to mimic the influence of noisy input data in an assessment. If 'assessed' stock numbers further back in time are needed, such numbers are derived through a VPA, using observed catches and starting with the stock numbers in the reference year. Accordingly, the noise in the assessed stock numbers is generated by mimicking the effect of noise in the input data to the assessment, but without fitting a model to the data. More specifically, it is the effect of the noisy data on the terminal stock numbers that is mimicked. Then the history is back-calculated from there.

2.3.1. Generating noise in stock numbers at age

This process generates an *Nobs* matrix with dimension (*age, y1, y2*) that contains perceived stock numbers at *age* for year *y1* as estimated in year *y2*. The process has the following steps:

Noise matrix
The annual noise in the terminal stock numbers is assembled in a matrix (age,years) of noise terms *x(a,y)*. Each year *y*, a new year column is added to that table. Basically, the noise is analogous to a separable model, i.e. the product of a year factor and an age factor. Here, both the year factor and the age factors are redrawn every year, but the year factor is common to all ages. The year factor can have an autocorrelation effect similar to that of recruitment. The terms are derived as followed for the year *y*..

- Year factor: Draw a random number *x* from a lognormal distribution with a year factor sigma as input. Include an autoregressive effect with a coefficient ρ to get the final year factor number $x_{1y}(y) = x_{1y}(y\text{-}1) *\rho + x*(1-\rho^2)^{1/2}$
- For each age *a*: Draw a random age factor number x_{2a} from a lognormal distribution with age-specific sigmas.
- Take the product $x(a,y) = x_{1y}*x_{2a}$ as entries in the noise matrix for the year *y*.

Autoregressive noise multipliers
When imitating an assessment in year *y*, observed terminal stock numbers *Nobs(a,y)* at age *a* in the year *y* are derived as follows:

- Take *N(a,y)* from the true stock
- Derive for each age *a* a number $\xi(a) = \alpha_0*x(a,y)+\alpha_1* x(a\text{-}1,y\text{-}1) + \alpha_2* x(a\text{-}2,y\text{-}2) + ... \alpha_i* x(a\text{-}i,y\text{-}i)$, for as many years backwards as can be assumed to be influenced by a noisy survey observation, or to the first year the year class appears, as outlined below.
- The observed stock numbers in year *y* are then taken as $Nobs(a,y) = N(a,y)* \xi(a)$

Hence, the influence of previous observation noise follows the year classes, as they would if fitting an assessment model with fixed terminal stock numbers. The coefficients α_i are input. As a guideline, one may argue that the influence is related to the fraction of the *N* back in time that is needed to account for the current survivors. That will decline backwards in time according to *exp(-Cum(Z-M)):* the $N=S*exp(CumZ)$ and the amount needed for the survivors is $S*exp(CumM)$. The α_i are scaled internally so that the sum of the applied α_i is 1.0. If this algorithm is not to be used, and the drawn $x(a,y)$ is used directly, i.e. $\xi(a) = x(a,y)$, the number of α_i -terms should be set to 1 and the α_0 to 1.0. Alternatively, since this is a place where errors are likely because intuitively, ignoring this model should mean 0 coefficients, then if the number is set to 0, the model will assume 1 and set α_0 to 1.0. This also makes it easier to switch the AR model on and off.

This autoregressive model is independent of the autocorrelation of the year factor described earlier, and comes on top of that.

VPA backwards in time
If observation model values for *N* backwards in time, i.e. $Nobs(age, y1, y2)$ for $y1<y2$ are needed in the decision

process, the following VPA-algorithm applies:

- Build a catch matrix *Cobs*
  - For years prior to the start of the simulation, observed catch numbers at age are input.
  - For each new year *y*, derive observed catches from the true catches at age *C(y,a)* in year *y* from the implementation model.
    - Draw a random ζ_1 number from a lognormal distribution with an input CV.
    - For each age *a*: Draw a random ζ_{2a} number from a lognormal distribution, and take the product $\zeta(a,y) = \zeta_1 * \zeta_a$
    - Take *Cobs(a,y) = C(a,y)*ζ(a,y)*
- Calculate stock numbers backwards with Popes equation, using the present and the previous *Cobs* values: *Nobs(a-i,y-i,y) = Nobs(a-i+1,y-i+1,y)*exp(M(a-1)) + C(a-i,y-i)*exp(M(a-1)/2).* The still existing year classes are started with the terminal *N*-values in the observation model. Older year classes are started with *N*-values derived from the catches in *Cobs* at the oldest true age and mortalities equal to those at the penultimate true age.

The *Nobs* matrix is revised each year. The older versions are kept. Accordingly, *Nobs*(*age*, *y1*, *y2*) contains perceived stock numbers at age for year *y1* as estimated in year *y2*.

This mechanism is evoked automatically if assessment results back in time are needed for some reason. For example, the rule may be such that the F is set according to the average SSB in the previous 3 years, as estimated in the last year.

2.3.2. Projection in the decision process.

When setting a **TAC for year *y,*** we start with the most recent perceived stock numbers that can be obtained from the observation model. These are stock numbers by 1. Jan in the year referred to as the *last assessment year* - the last year with estimates of *N* at the start of the year from an assessment. The typical case when setting a TAC for year *y* is that the assessment is done with data up to and including year *y-2*, which gives stock numbers at the start of year *y-1* (the 'intermediate' year). If needed, and that is the typical case, the stock numbers are projected forwards, starting with the stock numbers in the last assessment year, and the projection is carried on as far as needed.

This projection is deterministic. The removals are according to the TACs set for the years in the projection period. Weights and maturities are the true values from the first year of the projection. The standard input selection at age is assumed, as is the natural mortality. Recruitments in the projection is discussed below.

Assumed recruitments in the projection. Year classes that have not yet been recruited when the projection starts are not 'assessed' by the observation model. In the model, new year classes are always born at age 0, and if recruitment age is higher, they are just kept unchanged until they reach recruitment age. The following options are available:

- Year classes that are already born by the time of the last assessment year, but not recruited yet, because recruitment age is greater than 0. Prior to the projection, each of these stock numbers are revised by making them either:
  - Corrupted with a random multiplier, that is lognormal with a specified sigma. This can imitate the practise of deriving incoming year classes according to survey information.
    or
  - Substituted with a fixed assumed recruitment, which is the geometric mean perceived recruitment over a specified range of reference years.
- Future year classes are either:
  - derived according to the deterministic stock-recruit function, with basic parameters valid for the a reference year i.e. *R0(SSB)* as described in Section 2.2.3.2, but with the perceived SSB in the spawning year. No additional effects (periodic fluctuations, exceptional year classes (spikes) or noise terms) are included. If there is a delay due to autumn spawning, that can be taken into account.
    or
  - a fixed assumed recruitment, which is the same geometric mean perceived recruitment over specified reference years as in the point above.

  The reference years here are all specified relative to the year for which the TAC is to be decided.

2.4 Implementation model:

This model transforms a TAC coming from the decision model to an actual catch, both in tonnes and in numbers at age, and derives the true fishing mortality accordingly.

- The TAC is translated into catch numbers at age by a searching routine that finds the overall F leading to the TAC when applied to the true population, assuming the standard selection at age and the currently valid weights at age.
- Then noise is added to these catch numbers at age as age dependent log-normally distributed random noise, with CVs specified individually for each age. This gives preliminary catch numbers at age.
- The *realized total catch* is the decided TAC multiplied by a log-normally distributed year factor, biased if requested.
- The *final catch numbers at age* are obtained by adjusting the primary catch numbers at age with a common factor to give a sum of products of the catches and catch weights equal to the realized total catch.
- Realized fishing mortalities are derived from the final catch numbers at age and the true stock numbers.

If in this procedure, some true stock numbers at age are too small to allow the realized catches, new catches are derived assuming a fishing mortality specified as *Fmax*. The TAC remains for reference, but the true stock numbers are reduced according to *Fmax*. This procedure is only included to prevent the program from crashing, and is not intended as a part of the harvest rule. Accordingly, *Fmax* should be set at an unrealistically high value. A flag is set that indicates that the stock has collapsed in this iteration.

2.5 General description of the decision model:

2.5.1 General outline

The decision process leads to a *decided TAC for a year y.* The process is structured in several steps which are outlined here and described in detail below. The underlying idea is to find a rational and strictly structured way to design harvest rules. The decision process is sequential: First a **primary TAC** is set according to a rule, then this primary TAC is **modified** through additional rules.

First step (primary TAC):

- A **basis** derived from the perceived present and past state of the stock, projected forwards in time if needed.
- A **rule** derives a **measure of exploitation** from the basis.
- If needed, this measure is **translated** into a TAC.
- If the basis depends on the TAC (e.g. the basis includes a biomass after the TAC has been taken), the process is repeated in an **iteration** loop until convergence

This gives a **primary TAC.**

Second step (modifying the TAC): The primary TAC can be modified through a sequence of constraint rules to give the final TAC. Some of the modifying rules have a basis and a rule, as outlined above

Multi-annual TAC. Optionally a TAC can be set for several years ahead.

2.5.2. Primary TAC

2.5.2.1 Decision basis

The basis is the information about the stock that goes into a decision. Most often this will be the state of the stock expressed as SSB at some time, but in principle, any relevant information can be taken into account. Furthermore, the basis does not have to be a single number, it may well be a vector, where each component has some role in the decision process. For example, the decision can depend on a survey measuring the total biomass, supplemented

with the trend in a recruitment survey. ***Hcs*** has a number of options for deriving bases, but not everything thinkable. The first limitation is that it must be derived from the simple population model that is used here. Hence, length distributions, area distributions or climatic variables cannot be included. Nevertheless, by combining the present opportunities, a quite wide range of decision bases can be explored.

The specification of a basis has the following elements:

- A measure of abundance
- A time frame
- A calculation rule

For example, the basis can be the average *SSB* over the years *y1* to *y2*, or the linear trend in a survey over the years *y1* to *y2*.

The measure of abundance always represents perceived abundance according to the observation model, projected forwards as needed.
The calculation rule can be to take the mean (which includes a single value) , maximum, minimum, median or a linear trend.

2.5.2.2 Decision rule

The rule itself generates a measure *v* of exploitation as a parametric function of the decision basis vector ***Basis***, with a parameter vector $\boldsymbol{\theta}$:
$v = f(\boldsymbol{Basis}, \boldsymbol{\theta})$

The measure *v* can be a fishing mortality, the TAC itself or some other measure. Later on, this measure may have to be translated into a TAC (see below)

The parameters include standard levels of for example F, and trigger or reference points related to the basis, which can lead to different decisions depending on the basis. Each rule function is coded as a separate subroutine. The repertoire of rule functions in ***hcs*** has evolved gradually as various people have launched new ideas. Each rule has a number, and requires a certain dimension of the basis and a certain set of parameters. The meaning of the output *v* is not part of the rule, the rule just generates a number.

2.5.2.3. Translation

Unless the outcome of the rule is regarded as a TAC, it has to be **translated** into a TAC. The translation depends on the interpretation of the measure *v*, see section 2.6.1.3. The general rule is that when *v* is applied to some measure of abundance, it is always observed abundance (*Nobs* from the observation model) that is used. If more recent, or future numbers are required, the stock numbers are projected forwards (see Section 2.3.2)

2.5.2.4. Iteration

If the basis depends on the TAC (e.g. the basis includes a biomass after the TAC has been taken), the process of finding the **primary** TAC (for all years if multi-annual) is repeated in an iteration loop until convergence. The projections assume that the decided or temporary TACs are taken. If the projection period goes beyond the last TAC year, the same F as in the last TAC year is assumed.

In some cases, this searching routine may lead to an eternal loop: One value of *v* leads to another value of *v* that leads to the first value again. Typically, this can happen near a breakpoint, where the function in the rule is discontinuous. If this happens, the search is broken and a value midway between the two *v*'s is attempted. If that is unstable, it is still used but a warning is issued. In such cases, the rule should probably be regarded as equivocal and be revised.

2.5.3. Modifying the primary TAC

When the primary TAC has been calculated, it may be modified by additional rules. This is a very common feature in management plans, and the purpose is most often to reduce inter-annual variation in the TACs. ***Hcs*** has a sequence of modifying rules:

1. A filter rule:
2. A percentage rule
3. Minimum and maximum TAC

Each rule applies under a set of criteria. The general framework is similar to that outlined above. The criteria relate to a basis, which is constructed according to the rules above. The rule modifies the TAC conditional on the basis. The outcome is a revised TAC. Hence, the rule has the form *NewTAC = f(OldTAC,**Basis**,θ). OldTAC* here is the TAC derived so far in the sequential process. The 'parameters' θ may include the TAC the year before.

2.6 Available options in the decision process

This is an overview of the rule options that currently are available. The timing of various values in the decision process are always stated as years ***relative to the first TAC year, which here is referred to as year 0.***

2.6.1. Options for primary TAC

2.6.1.1 Available specifications of the basis.

The basis is a vector, where each entry is some information that should be taken into account when deciding the future exploitation. Each entry is derived by defining *a type*, *a time frame* and *a calculation method*, which are stated as input in the options file. For the time being, the following elements are available.

- Type:
  1. SSB according to observation model
  2. B+ (biomass above some age) according to observation model, derived using the catch weights.
  3. Perceived past recruitment according to observation model
  4. A 'survey' recruitment indicator (e.g. a noisy recruitment survey, derived by adding noise to the true recruitment)
  5. Result of a spawning biomass survey, derived by adding noise to the true SSB.
  6. Result of a total biomass survey, derived by adding noise to the true B+.
- Time frame:
  First and last year (relative to the first TAC year)
- Calculation:
  1. Average in the period
  2. Smallest in the period
  3. Largest in the period
  4. Relative slope in a linear regression over the period. The slope is relative to the mean, i.e. if the regression equation is $y = b*x + a$, the relative here is *b/ymean.*
  5. Median in the period

2.6.1.2. Available rules:

The functions are numbered. To each function, there is associated a dimension of the basis vector and of the parameter vector, and the function form is coded.

The parameters are input, and the dimension of the input parameter vector must be in accordance with the requirement of the selected function. The parameters can be included in the scanning loop.

1. ***'Alpha-rule':***

This rule, with alpha2=0, is one of the most common formulations at present.

Basis: 2 dimensions
7 rule parameters: *Btrig1,Btrig2, vstd, alpha1, alpha2, vmin,vmax*

Function:

- If Basis(1) < Btrig1: *v = vstd*(1.0-alpha1*(btrig1-Basis(1))/btrig1)*. If that leads to *v<0*, set *v = 0*
- If Basis(1) > Btrig1 and Basis(2) < Btrig2: *v = vstd*
- If Basis(1) > Btrig1 and Basis(2) > Btrig2: *v=vstd*(1.0+alpha2*(Basis(2)-btrig2)/btrig2).*
- *If v<vmin so far, set v=vmin*
- *If v>vmax so far, set v=vmax*

Note that the Basis(2) is used to modify the *v* relative to the *vstd*, and only when *Basis(1)* is above the *Btrig1* and *Basis(2)* is above the *Btrig2*. Note also that *Basis(1)* and *Basis(2)* are derived independently of each other. For example *Basis(1)* can be an SSB and *Basis(2)* a measure of recruitment. Also note that *alfa2 = 0* gives a flat exploitation above *Btrig1*.

2. 'Alpha rule with small v at small stock':

At presxent, this is the standard rule in the ICES MSY framework. The difference from rule 1 is that there is a third trigger, called *B0*, below which *v = Level1*, which represents the 'lowest possible exploitation'. Typically, *B0* will be *Blim*, but it is independent here. Between *B0* and *Btrig1*, the v is linearly changing with *Basis(1)*. Above Btrig1, the rule is similar to rule 1. Note that the *B0* ref. point refers to *Basis(1).* Also note that *alfa2 = 0* gives a flat exploitation above *Btrig1*.

Basis: 2 dimensions
7 rule parameters: *B0, Btrig1,Btrig2, vstd, Level1, alpha2. vmax*
Function:

- If Basis(1) < B0: *v = Level1*
- If B0 < Basis(1) < Btrig1: *v = vstd*(1.0+Level1*(Basis(1)-btrig1)/(btrig1-B0))*
- If Basis(1) > Btrig1 and Basis(2) < Btrig2: *v = vstd*
- If Basis(1) > Btrig1 and Basis(2) > Btrig2: *v=vstd*(1.0+alpha2*(Basis(2)-btrig2)/btrig2)*
- *If v>vmax so far, set v=vmax*

The rules 1 and 2 are illustrated in the graph below. In this example, the x-axis could be SSB or B+ . The *v* (for example F) is on the y-axis. The rules and their respective alpha1 have different colours.

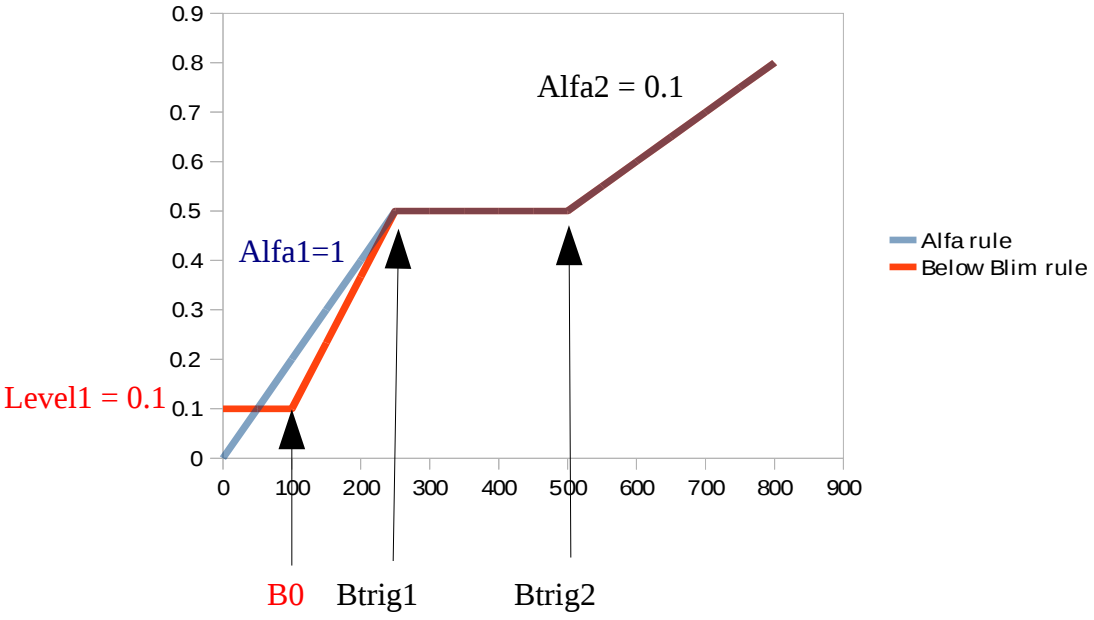

Figure. Rules 1 and 2.

Rule 3. Smooth transitions: Upper part of logistic function below B1 and above B2
Similar to rule 1, but with a smooth response and an upper limit to *v*

Basis: 2 dimensions
6 rule parameters: *Btrig1,Btrig2, vstd, slope1, slope2, vmax.*
Function:
In the formulas below, *vact* and *v0 are i*ntermediate variables.

- If Basis1<Btrig1) *vact=1.0/(1.0+exp(-slope1*(Basis1/Btrig1)))*
  v0=1.0/(1.0+exp(-slope1))-0.5
  v=vstd(vact-0.5)/v0*
- If (Basis1>Btrig1 and Basis2 >Btrig2) *vact=1.0/(1.0+exp(-slope2*(Basis2-Btrig2)/Btrig2))*
  v=vstd+(vact-0.5)(vmax-vstd)*
- Else *v=vstd*

Figure: Rule 3 with B1 at 3500 and B2 at 7000

4. 'Alpha-rule with variable level':
Similar to rule 1, but with a standard *v* called *vact* here that depends on a Basis(3). Typically Basis(3) will be a slope.

Basis: 3 dimensions
8 rule parameters: *Btrig1,Btrig2, v0, alpha1, alpha2, gain,vmin,vmax*
Function:

vact is an intermediate variable, that represents the 'standard' level of *v* as a function of the third dimension *Basis(3)* of the basis. This function is a straight line with parameters *v0* and *gain.*

- *vact=v0+gain*Basis(3)*
- If *Basis(1) < Btrig1*: *v = vact*(1.0-alpha1*(btrig1-Basis(1))/btrig1)*. If that leads to *v<0*, set *v = 0*
- If *Basis(1) > Btrig1 and Basis(2) < Btrig2: v = vact*
- If *Basis(1) > Btrig1 and Basis(2) > Btrig2: v=vact*(1.0+alpha2*(Basis(2)-btrig2)/btrig2)*
- If *v<vmin so far, set v=vmin*
- If *v>vmax so far, set v=vmax*

Note that a negative *v0* combined with a positive *vmin* gives a rule close to rule 2, but with a variable F-level.

5. 'Alpha-rule' with two levels of vstd:
Similar to rule 1, but with two levels of the standard v, depending on a Basis(3)
Basis: 3 dimensions
8 rule parameters: *Btrig1,Btrig2, RefR, vstdlow, vstdhigh, alpha1, alpha2, vmax*
Function:
In the formulas, *vstd* is an intermediate variable.

- If Basis(3) < RefR *vstd=vstdlow*
- If Basis(3) > RefR *vstd=vstdlhigh*
- If Basis(1) < Btrig1: *v = vstd*(1.0-alpha1*(btrig1-Basis(1))/btrig1)*. If that leads to *v<0*, set *v = 0*
- If Basis(1) > Btrig1 and Basis(2) < Btrig2: *v = vstd*
- If Basis(1) > Btrig1 and Basis(2) > Btrig2: *v=vstd*(1.0+alpha2*(Basis(2)-btrig2)/btrig2). If that leads to v<0, set v = 0*
- *If v>vmax so far, set v=vmax*

6. 'Small v below Blim' with two levels of vstd:
Similar to rule 2, but with two levels of the standard v, depending on a Basis(3)
Basis: 2 dimensions
9 rule parameters: *B0, Btrig1,Btrig2, RefR, Level1, vstdlow, vstdhigh, alpha2, vmax*
Function:

- If Basis(3) < RefR *vstd=vstdlow*
- If Basis(3) > RefR *vstd=vstdhigh*
- If Basis(1) < B0: *v = Level1*
- If B0 < Basis(1) < Btrig1: *v = vstd*(1.0+Level1*(Basis(1)-btrig1)/(btrig1-B0))*
- If Basis(1) > Btrig1 and Basis(2) < Btrig2: *v = vstd*
- If Basis(1) > Btrig1 and Basis(2) > Btrig2: *v=vstd*(1.0+alpha2*(Basis(2)-btrig2)/btrig2)*
- *If v>vmax so far, set v=vmax*

7. Escapement rule
Escapement strategy:
Basis: One dimension
2 parameters: *B1, B2*
Function:

- If Basis(1) < B1 + B2: *v=0*
- else *v=Basis(1)-B1*

v should be interpreted as the TAC. *B1* is the spawning biomass that should be left behind. B1+B2 is a lower limit for opening the fishery. Most often, the gap B2 will be 0, but does not need to be. Opening the fishery when there is less fish than the candidate catch does not make sense, so B2 should not be negative.

8. Data poor
Designed for data poor situations, where the TAC is maintained unless there are strong indications that the stock is changing.

Basis: Two dimensions
5 parameters: *B1, B2, vlow, vstd, vhigh*
Function:

- If Basis(1) < B1: *v=vlow*
- If Basis(1) > B1 and Basis(2) > B2: *v=vhigh*
- Else: *v=vstd*

v will typically be interpreted as a relative change in TAC (exploitation measure 6) and Basis will often be a trend in a survey biomass or CPUE. Accordingly, *vlow < 1, vstd =1* and *vhigh>1* will be the normal case.

Note on rules
The rules 1 and 2 are those commonly used, but formulated here with some additional options. It has the option to increase the exploitation *v* when the stock is large. The *vmax* in these (and some other) rules is a protection that the *v* then does not go sky high. For example, one would not like to see a harvest rate above 1. In some contexts, it may be relevant to set it corresponding to FMSY. If no upper bound on *v* is needed, set *vmax* to a high value, for example corresponding to *fmax*.

Rule 3 allows for a smoother transition around the breakpoints. I have not seen that used, but it could be worth considering. Here the *vmax* is part of the general rule.

Rule 4 is similar to rule 1, but with a standard exploitation that depends on something exåressed as basis (3). It was made to make the fishing mortality linearly dependent of recent recruitment.

Rules 5 and 6 are extensions of rules 1 and 2, with two levels of the standard v.

Rule 7 is an escapement rule. Typically, the v should be the TAC, and it is equal to the measure of the stock minus the amoult left behind. It is triggered by another basis, which can be the same, but not necessarily so.

Rule 8 is a simple implementation of the idea: Keep the exploitation constant unless there is a good reason to change it.

2.6.1.3. Options for translating the rule outcome (v) to TAC

Following the calculation of *v*, it is translated into a TAC, depending on the meaning given to it. The following meanings of the exploitation measure can be translated at present:

1. *v* is an F-value.
2. *v* is a Harvest rate, defined as TAC = HR*B+
3. *v* is the TAC itself (no translation needed)
4. *v* is a Harvest rate, defined as TAC = HR* SurveySSB (biomass survey)
5. *v* is a Harvest rate, defined as TAC = HR* Survey B+ (biomass survey)
6. *v* is the relative change in TAC from last year.

The survey results used are those for the year before the TAC year. Changes to this timing has to be hard-coded, subroutines *decideprimtac* and *getvfromv.*

2.6.2 Options for modifying the TAC by constraints:

The constraint rules basically use the TAC derived so far for year *y* together with previous TACs. Rules 1 and 2 apply under criteria that relate to a basis The basis is constructed according to the rules above (Section 2.6.1.1.) The basis is always of one dimension in the constraint rules.

The rule modifies the TAC conditional on the basis. The outcome is a revised TAC. Let *OldTAC* be the TAC set so far in the sequential process, and *NewTAC* the TAC according to the constraint. *PrevTAC* is the TAC the year before.

2.6.2.1. Filter rule:

Parameters:
- Trigger level, corresponding to basis
- gain.

If Basis satisfies the criteria: *NewTAC = gain*PrevTAC + (1-gain)*OldTAC*
else *NewTAC=OldTAC*

Criteria: If the basis is a trend, the filter only applies when the trend is lower than a trigger level, if it is a biomass

it only applies if it is higher than the trigger level.

Setting the gain to 0 disables the filter. Gain = 1 gives a constant TAC. To ensure that the filter always applies, use SSB as basis and set the trigger level to 0.

2.6.2.2. Percentage rule:

Parameters:

- Trigger level, corresponding to basis
- Percentage.

Then the rule is:

If Basis satisfies the criteria:	If *OldTAC > PrevTAC*(1+p)*	*NewTAC = PrevTAC*(1+p)*
	else	
	If OldTAC < PrevTAC(1-p)*	*NewTAC = PrevTAC*(1-p)*
	else	*NewTAC=oldTAC*
else (criteria not satisfied)	*NewTAC=oldTAC*	

or in simple terms: The TAC shall not deviate by more than p*100 percent from last years TAC, if B satisfies the conditions.

If the previous TAC was 0, the percentage rule does not apply.

Criteria: The basis for the percentage rule is calculated assuming the constrained TAC. Accordingly, the constraint applies only if it leads to satisfactory conditions.
If the basis is a trend, the percentage rule only applies when the trend is lower than a trigger level, if it is a biomass the rule only applies if it is higher than the trigger level.

Setting the percentage to a negative number disables the percentage constraint. To ensure that the percentage constraint always applies, use SSB as basis and set the trigger level to 0.

2.6.2.3. Minimum and maximum TAC:

Parameters:

- Minimum TAC
- Maximum TAC

The TAC shall not exceed a lower and an upper bound. To disable this step, set the lower bound to 0 and the upper bound to an incredible high number.

2.6.3. Additional management options.

Multi-annual TACs
If the TAC is to be valid for more than one year, there are two options for how to change the TAC.

1. Abrupt: The new TAC is set for all years in the whole period
2. Gradual: The TAC is changed linearly until the new value is reached in the last year of the period.

Transition rule towards a target in the initial phase.
When a harvest rule is introduced, it may imply a quite substantial change from the current exploitation. Managers sometimes would like to make that change gradual. ***hcs11_2 and later*** versions have the option to change the rule target gradually from the current value (i.e. that observed for year 0) to the final target. This is specified as a 'number of years to reach target'. If this number is greater than 1, the target management measure value *v* to be applied in each year is derived according to a 'Target reduction rule'. When the target has been reached, the

ordinary rules as described above take over.

The rule that is implemented is that the value of the target *vstd* is reduced linearly from the initial value to the final one over the specified number of years. This is done by setting up a table for future target values when the rule is introduced. Hence, the change is not relative to the currently perceived rule parameter, only to the initial one.

Note that this arrangement is different from some rules that have been proposed by EU in connection with the introduction of FMSY, where the reduction is relative to each years estimate of the previous years fishing mortality.

Banking and borrowing.
An option for simulating a regime with banking and borrowing (transferring parts of one years TAC to the following year (banking) or previous year (borrowing)) is included. This transfer is done every year. After a TAC for a year has been decided, it is modified by a factor *bb* which is input. The final TAC is set as *(1-bb)*tac(year) +bb*tac(year-1).* A positive factor is banking, a negative is borrowing. Effectively, it just means moving the transition between quota years a bit, and the effect is usually small.

Some comments to the constraint rules:
The rules provide great flexibility, way beyond what is relevant in most cases. In practice, one will have either the option to set the TAC as a weighted average of the present temporary TAC and the previous TAC (referred to as the filter rule), or apply a constraint on the percentage change. Combining these two will imply a sequence of two constraints, if the filter is not enough, then there is a percentage rule on top of it. For most practical purposes, this is over-kill. Also, the option to change the TAC gradually when a multi-annual TAC is used, is rather similar to applying the filter rule. Combining the two means that the change will be linear but the target will be determined by the filter rule. Again, this is probably over-kill.

The *'only'* option in the exception rule for the percentage constraint covers the situation where the constraint applies when and only when the constrained catch leads to Basis above the trigger level. This is a common feature in proposed harvest rules, and may be necessary to ensure sufficiently swift action if the stock abundance goes down, but leads to the paradox that the TAC can get trapped at a low level when rebuilding the stock. This is a quite well known weakness of the percentage type of constraints. When this situation has appeared, managers have sometimes chosen to deviate from the rule, if that still leads to a satisfactory biomass. This tactics has not been implemented here, mostly because it is hard to foresee how managers will solve such problems.

2.6.4. The concept of risk in HCS

Risk is defined here as the probability (percentage of trajectories) of a measure of stock abundance being below a reference value. The cost aspect is not taken into account.

The risk that is presented in HCS is of 3 kinds:

1. Risk to Blim: Probability that the true SSB is below the stated value for reference Blim. Alternatively, Blim can be stated as a distribution, and the risk refers to the compound distribution of Blim and SSB.
2. Prob. trigger: The probability that the Basis(1) is below the Btrigger1 value, i.e. the perceived state of the stock is such that the exploitation measure is reduced below the standard exploitation.
3. Prob crash: The probability that either
    - The decided catch cannot be taken because there is not enough fish. The test is that the catch corresponding to fmax is smaller than the decided catch.
      or:
    - The true SSB is less than 10% of the reference Blim.

The probability is calculated over a time period.

- In the results table, the probabilities are calculated for each year. The Probcrash is the ***cumulated*** probability of crash up to and including that year.

- In the collecting tables (Tables 12, 13, 14), the risks are expressed as the highest annual risk in the period, except Probcrash which is the cumulated probability in the last year in the time period. Previous versions of HCS (before version 13_1) had different definitions.

2.6.5. Decision models vs. standard practice

The standard practice when setting a TAC varies a good deal, and the procedures in ***hcs*** may differ from these practices. In particular, the following points may matter:

Weights and maturities:

The weights and maturities used in the decision process are always the most recently derived true values, which also are assumed valid for the whole projection period. There is no averaging of weights and maturities that go into the decision model, and no assumptions are made about future change in weight. In particular, trends or year class effects are not continued.

Selection at age
HCS uses a standard selection at age, which is input, in the decision process. The 'true' selection at age in the operating model may deviate from this, because it is derived from the actual catches at age coming from the implementation model. The common practise to derive a selection for projections and decisions from an assessment is not implemented. Neither is there any opportunity to change selection over time so far.

3. Instructions for use

3.1 Setting up a simulation.

3.1.1 Preparation and running hcs

To run a set of simulations, you will need to prepare the files:

- *bio.inn*
- *opt.inn*

You may also have to prepare the optional files:

- *nin.xxx* (the suffix depends on the format) for initial numbers, if you want to specify the initial numbers yourself. The alternative is priming with a fixed F.
- *canum.inn* if you need stock numbers in previous years in the decision process. This will be the case if the observation model includes a VPA, i.e. if perceived stock numbers in earlier years are needed for the decision process.
- *srspec* if you want individual sets of recruitment parameters for each iteration. Presumably, such sets will be drawn stochastically. This will have to be done outside ***hcs***, for example by running bootstrap assessments in which a set of recruitment parameters is produced for each bootstrap replica.

The format and contents of these files is described in detail in Section 3.2 below. The example files that are provided with the program can be used as templates to get all necessary lines in place, but they do not represent any kind of standard set up - there is no such thing!

The files are all ascii-files, and should be prepared with a 'clean' editor with no hidden tabulators etc.

The opt.inn file has a list of outputs, described in Section 3.3 . Pick the ones you will need there. Note that on output files 12, you will also have to specify years. The upper year should not be above the maxyr (hard coded in the hcscom17_1.i file).

It is usually easiest to use an old file as a template, but resist the temptation to assume that entries that you don't understand don't matter. To run the program, the executable and the input files have to be in the same folder The output files also go there. The output file names are hard coded, and old files are over-written without warning. ***Therefore, a good advice is to create a new folder for each set of runs.*** Copy the input files and program over there, modify them as needed and run.

- In Windows, the program is best started in a console window (dos-window). The command is `hcs17_1`. You may also start the program by double-clicking the *hcs17_1.exe* file in Explorer, but if the program stops for some reason, the console window closes and you cannot see error messages.
- In Unix, open a terminal window, go to the folder where you have the run, compile unless you have done that already (the command with gfortran will be: `gfortran hcs17_1.f`) and run (`./a.out`).

The program runs without any interaction. Under way, a listing of the current parameter values and a few main results are printed to the screen, just to see what happens.

Some errors on input files trigger a warning message and sometimes a stop in the program. Other errors, in particular missing numbers or missing names in input files only create an error message pointing to a line in the program code. Such errors may be hard to see on the input files, and it may be useful to look at the code to at least find out which file to look at. The *opt.inn* file is read by the *subroutine readoptinn* and the *bio.inn* file by the *subroutine readpopinn*. There are some comments in the code that may give you further indication where to look. See also the introduction in Section 3.2 for more on reading problems.

3.1.2 Modifying ranges.

The actual age range is input, on the biological data file. Note that the oldest age is always regarded as a plus age.

There is a maximum age that is hard coded as the parameter *maxage* on the ***hcscom17_1.i*** file, any oldest age not exceeding that is valid.
The year range (number of years) and the number of iterations are hard coded on the top of the ***hcscom17_1.i*** file as *maxyr* and *niter* parameters respectively. To change the year range or number of iterations, these parameters have to be changed and the program has to be recompiled. No changes are required elsewhere. However, the *maxyr* should not be set higher than 98, because some printout formats are adapted to 2-digit years.

3.2 Input.

3.2.1 General on input

There are two compulsory input files: *bio.inn*, *opt.inn* and optionally 3 more input files: a *nin*.xxx file, a *canum.inn* file and a *srspec* file. All file names are standard and hard-coded. On the files, numbers must be space separated. On the *bio.inn* and *opt.inn* files, comments and explanatory extra lines are permitted. They are skipped when reading the file:

3.2.1.1. Permitted comments on bio.inn and opt.inn, and how the input files are read:

When the file is read, information that the program needs is copied to an internal file, the rest is skipped. When reading the internal file, the line numbers tell what the line means. Therefore, lines must come in the right sequence, and missing lines cause the program to crash.

When reading into the temporary file:

- Blank lines are skipped
- Lines where the first non-blank character is not a number or a - (minus) are skipped. Thus, you may write a comment without any special marker, if it does not start with a number or a -. See how it is done in the example files.
- Text after the valid numbers is ignored (the program stops reading the line when it has got the numbers it expects). In the examples provided with the program, this is used extensively, to tell what the coding in each line means.

If you want to comment out lines, for example to have several alternatives ready, just put some character in a position before the numbers.

On the other files, comments are not permitted except after the valid numbers.

3.2.1.2. Tracing errors and redundant information.

Tracing reading errors sometimes is a bit of detective work. If lines are missing, the program will take what it finds literally. Most often, that will sooner or later lead to a format error or missing values, which makes the program stop. The program line number where the program stops is reported, and looking at what is done there in the code can sometimes be helpful, but often the mistake can be earlier in the file. One common error is that some ages are missing in a list of age-dependent entries, typically when a file with a different age range is used as a template.

Sometimes, different options require different additional information. The relevant information for the option must be on the file, but if there are additional redundant numbers, they will not be read. That makes it easier to swap between options, but there is some risk that input is used when you think it is ignored. For example, the first line specifying recruitment regimes starts with the number of regimes, followed by the specification of the shift between regimes and if that is the code 1 (shifts in fixed years), the years are specified.

```
3 1 12 20
```

If you want only one regime, it is sufficient to just change the number from 3 to 1:

```
1 1 12 20
```

The rest of the line is now redundant and is not read, so you do not have to delete anything if you should want to change back later on.

There are cases, however, where redundant data have to be on the line. For example, the line specifying spasmodic recruitments has 4 entries, e.g.: 3 6.2 0.6 1.0 -3 where the third number is the magnitude of the spikes. The spasmodic recruitment option is ignored by setting the third entry (relative magnitude of the speikes) to 1.0, as in the example, but for the program to recognize that number, some numbers have to be in the other posistions as well.

In the descriptions of the files below, line numbers refer to lines that are not comment lines. Entries are in this font.

3.2.1.3. Backwards compatibility

The format is generally not backwards compatible with older versions of HCS. In ***hcs17_1*** the input format for the **opt.inn** file has been completely revised. The **bio.inn** file has an additional line (the 6th non-comment line), and the new conventions for comments have been introduced. The previous comments are still understood.

3.2.2. Biological data (File bio.inn).

The input file for biological data is called ***bio.inn.*** There are two alternatives for the biological data, indicated by the code 1 or 2 in the appropriate line (see below). The first part of the file (before the code) contains specifications of basic bounds, of recruitments, and of general specifications for weights and maturities. The lines below the code specifies biology by age, i.e. natural mortality, selection, weights and maturities, that can either be specified as such (i-option = option 1) or derived from a growth model (p-option = option 2). Option 1 is typically used with real stocks, option 2 may be more useful for generic studies.

Both options

Lines 1-6: Basic settings

- Line 1: Year 0 : Initial year for reference in tables etc. The TAC for year 0 is known, year 1 is the first year where a TAC is set according to the rules. The initial numbers (nin.xxx file) are at the start of year 0.
- Line 2: Youngest_age Oldest_age: The oldest age is always a plus-age. If the youngest age is > 0, the program will draw recruits at age 0 in the year they are born, but project them forwards with zero weights and maturities. You do not need to worry about the time lag.
- Line 3: Lower_age Upper_age: Reference age range for fishing mortality. Fishing mortality is reported as the average over this age range.
- Line 4: Youngest_age for calculating TSB for use in models for density dependent weight and maturity. This age is independent of the age for calculating the B+ in harvest rules (opt.inn file).
- Line 5: Prop_F Prop_M Proportion of F and M before spawning
- Line 6: Delay from spawning to recruitment
  Options:
  - 0: 0-group recruits in the spawning year; this is the normal.
  - 1: Recruitment as 0-group the year after spawning, as in North Sea autumn spawning herring.

Lines 7 onwards: Specification of recruitment. One line specifying regime shifts, and a block of 4 lines for each regime.

- **Line 7: Recruitment regime shifts.** Number of regimes (code for shift, additional information)
  The rest of the line depends on the number of regime shifts.
  - If there are no regime shifts, set the number of regimes to 1, no further entries are needed:
    Number of regimes = 1
  - If there are more than on regime, there are two options, signalled by the code for shift.

    - If shift code = 1: Shift in fixed specified years. Listed after the shift code.
      Number of regimes, Code for shift =1 Years where shift takes place (relative to year 0)
      The number of shifts must be number of regimes - 1. The program will always start with regime 1.
    - If shift code = 2: Shift at random intervals, drawn according to a geometric distribution, with a

mean interval length and a starting year.
Number of regimes, Code for shift = 2 Mean interval Year where current regime started (relative to year 0, may be negative)

- **Line 8:** Code for where to get recruitment regime specification.
  Number Number .....Number
  One number for each regime:
  1: Read from the present file
  2: Read from a *srspec* file.
  The *srspec* file is opened and read if at least one of the regimes have code 2. Once the file is opened, the entries for all regimes that are on the file are read, but will only be used as needed. Please note that this option has
  not been properly controlled for the more recent versions of hcs.
- **Line 9 onwards:** Blocks of 5 lines with recruitment options, one for each regime. Only regimes where specifications are read from the present file (code = 1 in line 8) are specified. The number of blocks has to match the number of regimes with code = 1.

The 5 lines in each recruitment block are:

1. **First recruitment line:** Keys for recruitment options:
   S-R-function Distribution Truncation_type
   - Code for the S-R function: 1= Hockey stick, 2 = Beverton- Holt, 3=Ricker
   - Code for distribution type: 1=Normal, 2=Lognormal
   - Option for truncation: 1 or 2, see second recruitment line

Details of the S-R functions

The S-R functions are (with parameters *a_par* and *b_par* set in line 2 in the recruitment block)

1. Hockey - stick: $S>b\_par: \ R = a\_par$
   $S<b\_par: \ R = a\_par*SSB/b\_par$
2. Beverton-Holt: $R = a\_par*SSB/(SSB+b\_par)$
3. Ricker: $R = a\_par*SSB/b\_par*exp(1-SSB/b\_par)$

Note the form of the Ricker function, the *a* parameter is the maximum recruitment and the *b* parameter is the biomass at the maximum recruitment.

2. **Second recruitment line:** Parameters in recruitment function:
   a-parameter, b-parameter, ps, trunc_upper, trunc_lower, autocorrelation

ps - parameter.
Noise around the deterministic stock recruit function can be normal or log-normal. In both cases it is implemented through a multiplier $\xi noise$ .derived from a random number x with standard normal distribution. The ps-parameter is a dispersion parameter which is a CV with the normal distribution and a sigma with the lognormal distribution. The noise can be autocorrelated with a coefficient autocorrelation, and an autoregressive process with coefficients specified in recruitment line 5. The derivation of $\xi noise$. is described in detail in Section 2.2.3.2

Truncation (lower and upper bound)
The interpretation depends on the truncation option (on the first recruitment linee). x and $\xi noise$ are as defined above and in Section 2.2.3.2.

Rule: Draw a new random number if:

Option 1: $x > trunc\_upper$ or $x < trunc\_lower$
Option 2: $\xi noise > trunc\_upper$ or $\xi noise < trunc\_lower$

3. **Third recruitment line:** Periodic recruitment variation of the *a*-parameter:
Amplitude period phase

Use 0≤ Amplitude ≤ 1

The *a*-parameter is periodic according to
a_param(year) =standard_a_param(1.0+Amplitude* cos(2π*(Year-Phase)/Period))*

The phase is the actual year (relative to year 0) where the cosine function has its maximum, and is not linked to the start of the recruitment regime

If no periodic fluctuations: Set Amplitude = 0. The other parameters become dummies, but have to be there.

4. **Fourth recruitment line:** Spasmodic recruitments:
Distribution code Mean_interval Sigma _interval, Magnitude _factor, First_spasmodic_year

The first spasmodic year is relative to the year of regime change and can be negative.
Distribution code:
1: Lognormal intervals,
2: Binomial intervals (the sigma is dummy), *Prob(x years) = 1/mean * (1-1/mean)* ^{x}
3: Boxcar distribution of intervals – the sigma parameter is used for the relative range: The interval is drawn randomly in the range
*(mean-sigma*mean, mean+sigma*mean)*, rounded to the nearest integer

If no spasmodic recruitments: Set Magnitude _factor =1. The other entries become dummies, but have to be there

5. **Fifth recruitment line:** Autoregressive process (see Section 2.2.3.2)
Number_of_terms Coefficient _1 ..... Coefficient_ n

Surplus coefficients on the line are just ignored.
If no autoregression: Set Number_of_terms = 0.

- **General Weight and maturity properties**

The following 5 lines must be in the sequence below, the line number depends on the number of recruitment blocks:

1. B0: Reference biomass for density dependent annual effects on weights and maturities
2. R0: Reference recruitment for density dependent year class effects on weights and maturities
3. Sigma for normally distributed yearly noise multiplier on weights and maturities.
4. Amplitude (≤ 1) period, phase for periodic trends in weights and maturities
5. Lower_bound, Upper_bound, alpha for year class effects on weights and maturities (see section 2.2.1)

- ***Age dependent Weights and maturity properties***
Specification of option - input or parametric model
1 or 2: Code for biological data option: 1: Input data; or 2: Parametric model

The rest of the file depends on the option here.

Code = 1: Input data option:

There shall be **5 blocks of lines** (one block for each bullet point below, within each block there is one line per age. The age range must be in accordance with the age range specified in line 3. A wrong number of lines is the most

common cause of program crash when taking over the file from a different stock, The entries go into the models described in Sections 2.2.1 and 2.2.2:

- Weight at age in the catch: Age Standard_value Lower_bound Upper_bound Slope (alpha-value) Sigma for age-specific noise
- Weight at age in the stock: Age Standard_value Lower_bound Upper_bound Slope (alpha-value) Sigma for age-specific noise
- Maturity at age: Age Standard_value Lower_bound Upper_bound Slope (alpha-value) Sigma for age-specific noise

  The Lower bound, upper bound and slope refer to the density dependence of weights and maturities. With the i-option these are specified for each age and category separately.

- Natural mortality: Age Value
- Selection at age: Age Value

  Note that the selection is normalized internally in the program, so the scaling does not matter.

Code = 2: Parametric option:

One line for each bullet point:

- Natural mortality (assumed equal for all ages)
- Length at infinity (meters)
- Length at entrance (at youngest age)
- von Bertalaffy k
- Foulton condition factor ($W = cond*L^3$)
- Slope of maturity at age logistic function (Length with 50% maturity is set automatically at 2/3 of Linf)
- sel50:Length at 50% selection in the fishery
- slopesel *:* Slope of selection at age

  The selection is computed according to the length at age as a fraction of the selection at Linf:
- *Selection(age)=(1.0+exp(-slopesel*(Linf-sel50)))/ (1.0+exp(-slopesel*(Length(age)-sel50)))*
- Lower bound, Upper bound , alpha (slope): Density dependence parameters for weights and maturities (common to all ages). The bounds are relative to the standard value.

Please note that this option has not been properly controlled for the more recent versions of hcs.

Example of a bio.inn file with the i-option:

Could have been kind of Blue whiting
Note: the # is used to mark lines that are not used, but just kept for convenience. Any non-numeric letter can be used for this.

```
Basic framework:
2016                    Year 0
1 10                    First and last age
3 7                     Age range for average F
1                       First TSB-age in TSB dependent biology
0 0                     Fraction of F and M before spawning
0                       Delay from spawning to recruitment (0: 0-group recruits in the spawning year; 1: NS autumn spawners)

Recruitment regimes specification
3 1 5 30          3 regimes, fixed shift (code 1) in years 5 and 30
Code for how to read recruitment model specification
1 1  1              1= read from this file; 2= read from bootstrap file. One number for each regime
#  2 There is an SR_param file in the collection. If this option is used, the 5 regime-lines for that regime must be removed or
commented out.

#Regime 1 : Hockey stick,l ognorm, breakpoint at Blim, High R
1   1  2                    Recruitment model (1=HS),  distribution (1=lognorm) truncation_type 2
35000.0 1500.0 0.45 0.0  10.0 0.75  A-par, B-par, CV, trunc_lower, trunc_upper,  Autocorrelation
0.0 54 5                    Recruitment periodicity: Amplitude (0 is neutral), period, phase
3 6.2 0.6 1.0 -3              Spasmodic recruitment: Dist. type (3=boxcar), interval, range interval, Factor (1=neutral), first year).
0  Terms in autoregressive model. 0 when not used
#5 1.10 -0.33 -0.05 0.08 -0.10 Autoregressive model with 5 terms. Not used at present:
```

#Regime 2 : Hockey stick,lognorm, breakpoint at Blim Very low R
1 1 2
6000.0 1500.0 0.45 0.0 10.0 0.75 A-par, B-par, CV, trunc_lower, trunc_upper,
0.0 54 5 Recruitment periodicity: Amplitude (0 is neutral), period, phase
3 6.2 0.6 1.0 -3 Spasmodic recruitment: Dist. type (3=boxcar), interval, range interval, Factor (1=neutral), first year).
0

#Regime 3 : Hockey stick,lognorm, breakpoint at Blim Historic R
1 1 2 Recruitment model (1=HS), distribution (1=lognorm) truncation_type 2
9000.0 1500.0 0.45 0.0 10.0 0.75 A-par, B-par, CV, trunc_lower, trunc_upper, Autocorrelation
0.0 54 5 Recruitment periodicity: Amplitude (0 is neutral), period, phase
3 6.2 0.6 1.0 -3 Spasmodic recruitment: Dist. type (3=boxcar), interval, range interval, Factor (1=neutral), first year).
0

Weights and maturities: Parameters that are common for all ages and types. Age/type specific parameters are further down.
This setting if for no density dependence
6000.0 B0: TSB corresponding to reference values (density dep)
7000.0 R0: Recruitment of reference year class (year class factor)
0.0 Sigma for random year factor noise
0.0 50.0 -10.0 Amplitude, period and phase for periodic trends. Neutral; Ampl=0
0.5 2.0 0.0 Lower and Upper values and Alfa for year class factor. Neutral: Alfa=0

Code for biological data source
1 1=input ('i-option'), 2=parametric model ('p-option')

Weight at age in the catch (average 2011-2015)
1 0.042 0.5 2.5 0.0 0.1
2 0.070 0.5 2.5 0.0 0.1
3 0.090 0.5 2.5 0.0 0.1
4 0.111 0.5 2.5 0.0 0.1
5 0.129 0.5 2.5 0.0 0.1
6 0.150 0.5 2.5 0.0 0.1
7 0.167 0.5 2.5 0.0 0.1
8 0.181 0.5 2.5 0.0 0.1
9 0.199 0.5 2.5 0.0 0.1
10 0.212 0.5 2.5 0.0 0.1

Weight at age in the stock
Equal to catch weight
1 0.042 0.5 2.5 0.0 0.1
2 0.070 0.5 2.5 0.0 0.1
3 0.090 0.5 2.5 0.0 0.1
4 0.111 0.5 2.5 0.0 0.1
5 0.129 0.5 2.5 0.0 0.1
6 0.150 0.5 2.5 0.0 0.1
7 0.167 0.5 2.5 0.0 0.1
8 0.181 0.5 2.5 0.0 0.1
9 0.199 0.5 2.5 0.0 0.1
10 0.212 0.5 2.5 0.0 0.1

Proportion mature age
No density dependent effects
1 0.11 0.0 10.0 0.0 0.0
2 0.40 0.0 10.0 0.0 0.0
3 0.82 0.0 10.0 0.0 0.0
4 0.86 0.0 10.0 0.0 0.0
5 0.91 0.0 10.0 0.0 0.0
6 0.94 0.0 10.0 0.0 0.0
7 1.000 0.0 10.0 0.0 0.0
8 1.000 0.0 10.0 0.0 0.0
9 1.000 0.0 10.0 0.0 0.0
10 1.000 0.0 10.0 0.0 0.0

Natural mortality at age
1 0.2
2 0.2
3 0.2
4 0.2
5 0.2
6 0.2
7 0.2
8 0.2
9 0.2
10 0.2

Selection at age
1 0.094
2 0.178
3 0.463
4 0.736
5 1.038
6 1.192
7 1.5
8 1.5
9 1.5
10 1.5

Example of lines specific for the parametric (p-) option

2 Option 2 means parametric model

0.2 Natural mortality (all ages)

Growth and maturity
1.0 Length at infinity (meters)
0.05 Length at entrance
0.25 von Bertalaffy k
8.0 Foulton condition factor
5.0 Slope of maturity at age function (50% is at 2/3 of Linf)
Selection at age
4.0 Age at 50% selection
5.0 Slope of selection at age

Parameters for **TSB factor** of density dependent weights and maturities Lower bound Upper bound alpha factor. These parameters apply to all ages
0.5 1.2 0.5

3.2.2. Run options (File opt.inn):

This file contains all specifications for the harvest rule, some other options and choice of printout.

Scanning.
Many of the parameters are stated with a lower, upper value and step value. The program will scan over these values. Setting the upper equal to the lower means only this value is used. The interval should not be zero.

1. Delay between last assessment year and first TAC year. Normally this is -1, which means that there is an intermediate year between assessment and TAC year.
2. Interval between decisions. Normally 1, but can be longer.
3. Reference Blim. This is not used in harvest rules, but risk to Blim is evaluated for this value. The Blim is a distribution, with the stated value as expectation, and a sigma for a log-normal distribution. If sigma=0, the stated value is used as it is.
4. Maximum F-factor. When running the program, this F is assumed if there is insufficient fish in the population to take a decided catch. This is simply to prevent program crashes, and not an instrument to use as part of the rule design or evaluation. If you need an upper bound for the exploitation measure, the *vmax* which appears in most of the rules, is a better way. The *fmax* should be a value that hardly is met, 3.0 can be a good value if you don't have other ideas.

General specifications

Line 1: Code for initial stock numbers at age.
Options
1. Priming with a fixed F
2. Reading stock numbers at age and variance-covariances from a file (expects a *nin.inn*-file). Note that the entries in the diagonal in this matrix are sigmas in a log-normal distribution, other

elements are correlations between log numbers at age.

3. Reading stock numbers from a file (expects a *nin.obs* - file), and apply the observation model to these stock numbers.
4. Read a set of initial numbers corresponding to each iteration, expects a *nin.lst* file

Line 2: Priming value for F (if initial numbers option is 1) or value of TAC in year 0 (if initial numbers option is 2, 3 or 4)
Line 3: Maximum possible F (to protect the run if there is not enough fish for the TAC)
Line 4: Last assessment year: Last year with N-estimate in assessment relative to TAC year (normally 0 or a negative number, -1 for one intermediate year)
Line 5: Interval between decision years (1 means annual decisions)

Distribution of Blim; Blim is used for reporting risks (probabilities that SSB < Blim), but not used directly in harvest rules. Blim is regarded as a stochastic variable, with a reference value, a distribution type and a dispersion parameter. These are specified in the lines 6-8:

Line 6: Reference Blim
Line 7: Distribution type for stochastic Blim: (0=Fixed, 1=Normal, 2= Lognormal)
Line 8: Dispersion parameter for stochastic Blim (Normal dist: CV; Lognormal dist: sigma; 0 = Fixed Blim)
Line 9: Youngest age in B+: B+ is the total biomass from the youngest age onwards for use ***in harvest rules***. It is independent of the age used for calculating TSB related to density dependent growth on the bio.inn file and reporting in the Yield per recruit routine. See Section 2.1.3.
Line 10: Code number for multi-annual TAC How a multiannual TAC is implemented: (1: Abrupt, 2: Gradual) Dummy if the interval in line 5 is 1.
Line 11: Bank-borrow factor: Fraction of TAC to be transferred between years. (>0: bank; <0: borrow; 0=no bank-borrow)

6 lines specifying options for handling of recruitments in projections: Some of these numbers are dummies, depending on the options. However, all numbers have to be on the file.

Line 12: Option init. year
Option numbers for the initial year:
1. Geometric mean values in the period stated by the reference years (CV is dummy)
2. Noise on true values

Line 13: Option for further years.
Option numbers for further years:
1. Geometric mean values in the period stated by the reference years
2. Use stock - recruitment function with SSB in the year specified in line 17. The recruitment regime (if there are several) will be that for the last year in the reference range (line 15)

Line 14: First reference. year,
Line 15: Last reference. year,
Line 16: CV for noise (option 2 in initial year)
Line 17: Reference year for SSB (option 2 for later years).

When projecting the stock forward to decide on the TAC, something has to be assumed about the youngest ages. Different assumptions are allowed for the youngest age in the initial year of the projection, and recruitments in subsequent years.

- The first and last reference year are relative to the last assessment year (i.e. the last year for which there are estimates of stock numbers at 1st. January) and should normally be negative. Reference year =0 will be the year after the last year with assessment data, for which there is no recruitment data yet.
- With option 1 in lines 12 and/or 13, the geometric mean is that of the perceived stock numbers coming from the VPA in the observation model.
- If option 2 is stated for the initial year (Line 12) , the CV in line 16 applies. Else, this CV is a dummy
- If option 2 for further years is stated (line 13) the stock recruit function with the recruitment regime in the first reference year is assumed, the second reference year is a dummy. The SSB in this stock recruit

function refers to the year stated on line 17. This year is dummy in all other cases.

Definition of basis for decisions on primary TAC

Each Harvest rule (see below) has a number of bases (entries in the basis vector). This is different for the different rules.

Line 18: Number of bases. The necessary number depends on the harvest rule. Additional bases are calculated, but not used.

Necessary numbers:

Rule 1:	2 bases
Rule 2:	2 bases
Rule 3:	2 bases
Rule 4	3 bases
Rule 5:	3 bases
Rule 6:	2 bases
Rule 7:	1 base
Rule 8:	2 bases

Each base has a block of 4 lines.

Line 1 in base block: Type
Line 2 in base block: First year (relative to TAC year)
Line 3 in base block: Second year (relative to TAC year)
Line 4 in base block: Calculation code

Codes in base specification:

Type:

1. SSB according to observation model
2. B+ according to observation model, derived using the catch weights.
3. Perceived past recruitment according to observation model
4. A 'survey' recruitment indicator (e.g. a noisy recruitment survey, derived by adding noise to the true recruitment)
5. Result of a spawning biomass survey, derived by adding noise to the true SSB.
6. Result of a total biomass survey, derived by adding noise to the true B+.

Calculation code:

1. Average in the period
2. Smallest in the period
3. Largest in the period
4. Relative slope in a linear regression over the period. The slope is relative to the mean, i.e. if the regression equation is $y = b*x + a$, the relative here is $b/ymean$.
5. Median in the period

Block stating the harvest rule.

The number of lines depend on the rule.

First harvest rule line: Rule number (listed below together with the parameters.)
Second harvest rule line: Exploitation measure

Code for Exploitation measures:

1. F
2. HR based on assessed B+
3. TAC directly
4. HR based on SSB survey
5. HR based on B+ survey
6. Relative change in TAC (TAC(y)/TAC(y-1))

Following lines: Parameters lines - one line for each parameter The number of parameters depends on the harvest rule number. Please consult Section 2.6.1.2 for precise definitions of the rules and parameters.
Format:
lower, upper step: Values for the parameter, scanned from lower to upper in steps

Parameter lines by rule:

- Rule 1 (Alpha-rule):
  - Btrig1
  - Btrig2
  - Vstd
  - Alpha1
  - Alpha2
  - vmin
  - vmax
- Rule 2 (Alpha rule flat below B0):
  - B0
  - Btrig1
  - Btrig2
  - Vstd
  - Level1
  - Alpha2
  - vmax
- Rule 3 (Alpha rule with smooth transitions):
  - Btrig1
  - Btrig2
  - Slope 1
  - Slope 2
  - vstd
  - vmax
- Rule 4 (Alpha rule with variable vstd)
  - Btrig1
  - Btrig2
  - v0
  - Alpha 1
  - Alpha2
  - Gain
  - Vmin
  - Vmax
- Rule 5(Alpha rule with two levels of vstd)
  - Btrig1
  - Btrig2
  - Refr
  - vlow
  - vhigh
  - Alpha 1
  - Alpha 2
  - vmax
- Rule 6 (Alpha rule with two levels of vstd, flat below B0)
  - B0
  - Btrig1
  - Btrig2
  - Refr
  - vlevel 1
  - Vlow
  - Vhigh
  - Alpha2
  - vmax
- Rule 7 (Escapement rule)
  - Btrig1
  - Distance between Btrig2 and Btrig1
- Rule 8 (Data poor)

- Btrig1
- Btrig2
- vlow
- vstd
- vhigh

Definition of constraint rules.
There are 3 kinds of constraints. All must be specified in sequence on the file. If they do not apply, use parameters that disable them. For each constraint, the following are specified:

- One line with the
  constraint number. This is a reference when reading the file, and should not be touched.
- Then, lines with parameters:
  - For constraint number = 1 (filter):
    Basis for derogation (4 lines, entries as for bases for harvest rules above. Only one basis is permitted):
    Type
    First_year
    Last_year
    Calculation
    Trigger for derogation (lower, upper, step) The constraint applies if the basis is above the trigger
    Filter gain (lower, upper, step)

  - For constraint number = 2 (percentage):
    Basis for derogation (4 lines, entries as for bases for harvest rules above. Only one basis is permitted):
    Type
    First_year
    Last_year
    Calculation
    Trigger for derogation (lower, upper, step) The constraint applies if the basis is above the trigger
    Percentage (lower, upper, step) Max permitted change in percent. Negative number: No constraint
  - For constraint number = 3 (min-max). 2 lines:
    Minimum TAC (lower, upper, step)
    Maximum TAC (lower, upper, step)

Uncertainties block

- ***Bias obs model*** (0=not used)
  lower, upper step:
- ***CV for year factor in obs. model***
  lower, upper step:
- ***Age factors for obs. model uncertainty*** (one line for each age)
  Age CV
  .
  .

- ***Coefficients for autoregressive error*** in survivors in obs. model (see Section 2.3.1)
  Number_of_coeffiicients Coeff 0, Coeff 1 .....Coeff n-1 (number=0: not applied)

CV for observed catches (for use in VPA in obs. model)
lower, upper step:
Age factors for obs. catches uncertainty (one line for each age)
Age CV
.

.

Observed surveys

These survey data are used in decision bases. The kind of survey is determined by the kind of basis. Specifications here for biomass survey covers both surveys for SSB and for B+. The lines have to be in the file, but unless such surveys are actually used, the numbers are dummies.

Catchability for recruitment survey
lower, upper step
CV for recruitment survey
lower, upper step
Catchability for biomass survey
lower, upper step
CV for biomass survey
lower, upper step

Implementation noise

Implementation bias (0 = no bias)
lower, upper step:
CV year factor implementation noise (0 = not used)
lower, upper step
Age factors for implementation noise
Age CV
.
.

Prinout options:

Keys for printout options (one line for each kin of file - the file names are hard-coded and appear heret as reminders)

General format

First number: Internal code (don't touch it!); Second number p: p=1: print, p=0: do not print.
11 p 'Results': Results file (+ Main results: Extract of Results)

Special format: Collected results files from year y1 to year y2

12 n Don't touch the 12. The number n is the number of collected files (max 10) One line for each file stating the years
y1 y2
y1 y2
.
.
y1 y2

13 p 'Autoc_CVS0': Autocorrelations and SD for SSB in year 0
15 p 'traj': Individual trajectories
16 p 'Recovery': Years to recover from below Blim
17 p 'Yieldrecr': Deterministic Yield & SSB per recruit
18 p 'Catch_dist': Equil. age distr. in catch
19 p 'SSB_dist': Equil. age distr. in SSB
20 p 'srpairs': All stock-recruit pairs
21 p 's0dist': All SSBs in year null
22 p 'Fdist: All F-values (iteration and year)
23 p 'IAVdist': All IAV values (by iteration and year)
24 p 'compbas': True and assumed basis, but only for the TAC year and the first 100 iterations
25 p 'Wema': Weights and maturities at age

26 p 'Comprecr': Compares assumed and real recruitments

The output files as explained further in the output section (3.3) below.

Example of Options input file

```
Options input file

Template with example numbers filled in.
Substitute numbers at the start of each non-comment line with your own specifications.
Consult the manual for detailed interpretation of each entry.

Lines where numbers must be filled in are marked with one * for each number needed
Options and some parameters have 1 number, which wil be used throughout
Some parameters are scanned automatically over the specified range, these
parameters are entered with 3 numbers: Lowest, highest,interval.
CVs at age have two numbers, where the first number is the age.
Space is expected as separator everywhere.

The * are just reminders, they have no meaning in the program.
Important: Avoid tabulators - plain ascii text please!

Permitted comments:
When the file is read,
     Blank lines are skipped
     Lines where the first non-blank character is not a number or a - are skipped when reading the file
     Text after the valid numbers is skipped.
If you want to comment out lines, for example to have several alternatives ready, just put some character in position 1.

General conditioning block:
3      *Code for initial numbers: (1=prime,2=numbers and vcv, 3=numbers and obs model, 4. Read list from file)
1150  *Exploitation in initial year (year 0): (If prime, priming F; if numbers, TAC in year 0)
3.0    *Maximum F-factor
-1     *Delay between obs. year and first TAC year: -1= One intermediate year
1      *Interval between decision years: 1=annual TAC decisions
1500  *Reference Blim amd alternative Blim (not often used)
0      *Distribution type for Blim (0: fixed, 1: Normal, 2. Lognormal
0.0    *Sigma for stochastic reference Blim: 0=fixed Blim
1      *Youngest age for TSB in harvest rules
1      *How a multiannual TAC is implemented: (1: Abrupt, 2: Gradual) Dummy if interval=1.
0.0    *Bank-borrow: >0: bank; <0: borrow 0=no bank-borrow

Recruitment assumption in projections in decision process (see Section 2.3.2. in the manual!)
Not all entries are used for all options, but all entries have to be present on the file.
1      * Option for ages 0-lowest_age in initial year: (1: Geomean over time interval. 2.True recruitment with noise)
2      * Option for recruitment in later years (1: Geomean over time interval. 2: Deterministic SR-function with SSB in ref. year))
-5     * Start time interval (relative to TAC year)
-2     * End time interval  (relative to TAC year)
0.0    * CV for noise (option 2 for initial year).
-1     * Reference year for SSB (option 2 for later years).

Harvest rule block:

Rules for the primary TAC

Basis for decisions.
There must be at least as many bases as required by the harvest rule declared below.
Redundant bases are calculated, but not used
Each basis is defined by 4 numbers.
The time frame is stated relative to the year for which the TAC is to be decided.

2    *Number of bases (there shall be exactly 4 valid entries for each basis)
First basis
1      *Type basis (1=SSB, 2=TSB, 3=assessed recruitment, 4=recruitment survey, 5 SSB survey,6=B+survey)
0      *Time frame - first year
0      *Time frame last year
1      *Calculation of basis: 1=mean over time frame, 2=minimum, 3=maximum, 4=linear trend

 Second basis
```

1 *Type basis (1=SSB, 2=TSB, 3=assessed recruitment, 4=recruitment survey, 5 SSB survey,6=B+survey)
0 *Time frame - first year
0 *Time frame last year
1 *Calculation of basis: 1=mean over time frame, 2=minimum, 3=maximum, 4=linear trend

Selected harvest rule.
2 *Rule number (see list in the manual)
1 *Exploitation measure
(1=F, 2=HR based on assessed B+, 3: TAC rule, 4: HR on SSB survey, 5: HR on SSB survey,6: Relative change in TAC)
Surveys (measure 4 and 5) are for the year before the TAC year

Rule parameters.
The number of parameters and their interpretation depends on the rule.
See manual Section 2.6.1.2.for list of rules and parameters needed.
All parameters are scanned

Each line has 3 numbers: Lowest, highest and interval in the scan
The number of valid lines must be exacly that required by the rule in question.
1500 1500 1500 *** Scan B0
2250 2250 750 *** Scan Btrig1
100000 100000 100000 *** Scan Btrig2
0.18 0.32 0.14 *** Scan v standard
0.05 0.05 0.015 *** Scan Level 1
0.0 0.0 1.0 *** Scan Alfa2
1.0 1.0 0.08 *** Scan vmax

Constraint rules:
All rules 1-3 must be on the file.

1 *Constraint rule 1 (filter) Don't touch this number!
Basis for filter derogation (4 lines)
1 *Type basis (1=SSB, 2=TSB, 3=assessed recruitment, 4=recruitment survey, 5=biomass survey)
0 *Time frame - first year
0 *Time frame last year
1 *Calculation of basis: 1=mean over time frame, 2=minimum, 3=maximum, 4=linear trend
0 0 2360 *** Scan Trigger for filter derogation (Does not matter with no filter)
0.0 0.0 0.5 *** Scan Filter gain 0 <= gain <=1: 0.0 = No filter; 1.0: Locked TAC

2 *Constraint rule 2: Percentage Don't touch this number!
Basis for derogation from percentage rule (4 lines)
1 *Type basis (1=SSB, 2=TSB, 3=assessed recruitment, 4=recruitment survey, 5=biomass survey)
1 *Time frame - first year
1 *Time frame last year
1 *Calculation of basis: 1=mean over time frame, 2=minimum, 3=maximum, 4=linear trend
2250 2250 500 *** Scan Trigger for percentage derogation
-5 -5 25 *** Scan Percentage: Negative = ignored

Constraint rule 3: Minimum and maximum TAC
3 *Constraint rule 3: Don't touch this number!
0 0 100 *** Scan Minimum TAC (0: No minimum)
10000 10000 500 *** Scan Maximum TAC (10000: presumably never reached)

Uncertainties block

Obs. model ('assessment') uncertainty
0.0 0.0 0.2 *** Scan Bias obs model (fixed bias - 0=not used)
0.25 0.25 0.05 *** Scan CV year factor obs model
0.60 0.60 0.3 *** Scan autocorrelation in year factor obs model

Age factors for obs. model uncertainty: Age and CV. from Table 2c in assessment output pacage
Rounded to nearest 0.05
1 0.45 **
2 0.30 **
3 0.30 **
4 0.25 **
5 0.25 **
6 0.25 **
7 0.25 **
8 0.25 **
9 0.25 **
10 0.45 **

Coefficients for autoregressive error in survivors
0 0.3 0.25 0.20 0.15 0.1 *(****) Number of coefficients and their values (number=0, or number = 1 and first coeff=1: not applied)

Observed catches (for use in VPA in obs. model)
0.2 0.2 0.2 *** Scan year factor CV for observed catches (same for all ages)
Age factors for noise on observed catches: Age and CV.
1 0.10 **
2 0.10 **
3 0.10 **
4 0.10 **
5 0.10 **
6 0.10 **
7 0.10 **
8 0.10 **
9 0.10 **
10 0.10 **

Observed surveys
1.0 1.0 1.0 *** Scan Catchability for recruitment survey 21
0.2 0.2 0.2 *** Scan CV for recruitment survey 22
1.0 1.0 1.0 *** Scan Catchability for biomass survey 23
0.6 0.6 0.1 *** Scan CV for biomass survey 24

Implementation noise
0.0 0.0 0.2 *** Scan Implementation bias (0 = no bias) 25
0.1 0.1 0.2 *** Scan CV year factor implementation noise (0 = not used) 26

Age factors for implementation noise: Age and CV.
1 0.10 **
2 0.10 **
3 0.10 **
4 0.10 **
5 0.10 **
6 0.10 **
7 0.10 **
8 0.10 **
9 0.10 **
10 0.10 **

Printout list

First number: Reference file number, don't touch it!
Second number: 0=skip, 1=print
The text after the numbers is the file names and some explanation, just as remindder.
11 1 'results': Results file (+ Main results: Extract of Results)

Collected results files from year x to year y
12 3 Dont touch the 12. The next number is the number of collected files (max 10)
10 14
20 24
41 50 Years_y1_y2': Averages from year x to year y. Years specified

13 1 Autocorrelations and SD for SSB year 0
Additional results
15 1 'traj': Individual trajectories
16 1 'Recovery': Years to recover from below Blim
Deterministic equilibirium results
17 1 'Yieldrecr': Deterministic Yield & SSB per recruit
18 0 'Catch_dist': Equil. age distr. in catch
19 0 'SSB_dist': Equil. age distr. in SSB
Distributions for control
20 0 'srpairs': All stock-recruit pairs
21 0 's0dist': All SSBs in year null
22 0 'Fdist': All F-values (iteration and year)
23 0 'IAVdist': All IAV values (by iteration and year)
24 0 'compbas': True and assumed basis, but only for the TAC year and the first 100 iterations
25 0 'Wema': Weights and maturities at age
26 0 'Comprecr': Compares assumed and real recruitments

3.2.3. Files with initial numbers (nin-files - optional)

There are several options for input of initial numbers. The input file depends on the option stated in Line 1 of the ***opt.inn*** file. The program expects a file with the right name and contents. If the option is 1, no file of this kind is needed.

- The first number on each line is the age. The second number is the stock number at age. The rest of the line depends on the options below.
- If the youngest age is higher than 0, and the initial data are taken from a nin-file, SSBs for the years prior to the initial years are stated at the end of the file, starting with the earliest year. This is to have a basis for drawing recruitments for the year classes that were born, but not recruited at the start of the simulation.

Note: The previous nin.num format is no longer valid. It can be substituted by the nin.inn format with only non-zero entries on the diagonal of the matrix.

nin.inn - corresponding to option 2
Initial stock numbers at age, and a matrix with sigmas and correlations. The numbers at age are plain numbers, which are converted to logs internally by the program. The diagonal elements on the matrix are sigmas in a log-normal distribution, the other elements are correlation coefficients (see example below). If you only have estimates of the sigmas, set all non-diagonal elements to 0.0

Example - nin.inn

```
 3 1645.200   0.017 0.005  0.007 0.007 0.007 0.007 0.008 0.008 0.008 0.000
 4 3923.600   0.005 0.013  0.007 0.007 0.007 0.008 0.008 0.008 0.009 0.000
 5 1431.000   0.007 0.007 0.014 0.009 0.009 0.010 0.010  0.010 0.011 0.000
 6  245.570   0.007 0.007 0.009 0.014 0.009 0.010 0.010  0.011 0.012 0.000
 7  343.940   0.007 0.007 0.009 0.009 0.014 0.011 0.011 0.012  0.012 0.000
 8  174.690   0.007 0.008 0.010 0.010 0.011 0.015 0.012 0.013  0.013 0.000
 9  106.100   0.008 0.008 0.010 0.010 0.011 0.012 0.016 0.014  0.014 0.000
10   74.954   0.008 0.008 0.010 0.011 0.012 0.013 0.014 0.018  0.016 0.000
11   44.756   0.008 0.009 0.011 0.012 0.012 0.013 0.014 0. 016  0.020 0.000
12   58.783   0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000  0.000 0.020
500
500
500
```

nin.obs - corresponding to option 3
Initial stock numbers at age only. To get random numbers later on, the observation model is used.

Format:

```
Age Number
.
.
SSBs for previous years, starting with the most distant year
```

Example of a nin.obs file. Here, since this is an example with youngest age >0, there are lines at the end with previous SSBs.

```
3     6000
4     3058
5    16441
6     6963
7    16602
8      700
9      730
10    2905
11    2922
12     377
13      80
500
```

500
500

nin.lst - corresponding to option 4: Initial numbers at age read from a list, one set for each iteration.

Format:

Iteration_ number Number lowest age ..... Number oldest age

3.2.4. File with historical catch numbers at age (canum.inn) - optional

Contains catch numbers at age for the years prior to the year defined as year 0. This file is needed by the observation model if years prior to year 0 are used in the decision process. It is assumed that the numbers represent ages starting at the age stated as the youngest age and ending with the age stated as the oldest age in the bio.inn file. The year sequence does not matter, and redundant years are ignored. The year is specified by the calendar years.

Format:

Year Number lowest age ..... Number oldest age
.
.

Example of a canum.inn file. In this case, the lowest ages stated in the bio.inn file is at 0, so the numbers start there. The oldest age is 12 in this example, so there are 13 columns.

2010 23453 78605 137101 303928 739221 611729 284788 143039 102072 45841 21222 6255 8523
2011 30429 62708 115346 322725 469953 654395 488713 244210 113012 53363 25046 12311 10775
2012 23803 66164 200064 214251 416037 454147 510469 323103 142390 69454 30573 11648 11741
2013 11325 46977 226179 430081 342280 437248 421220 338388 192725 118727 46253 18932 17856

3.2.5. File with individual Stock-Recruit parameters (srspec - optional)

This file is read if parameters for any of the recruitment regimes shall be read as individual iterations from a file (read option = 2 on line 8 of the bio.inn file).
There should be a number of lines equal to the number of iterations, for each regime where this option is used. Each line has 19 entries, to be read space separated. All numbers can have format as reals – those that are used as integers will be converted by the program. The definition of each entry is given in the description of the bio.inn file (Section 3.2.1.)

1. Regime number
2. Iteration number
3. jfunc: The code for stock recruit function (see below)
4. jdist: The code for distribution of noise around the S-R function
5. jtrunc: The code for truncation type
6. ioptspasm: Code for type of spasmodic function
7. lastspasm: Year where counting of spasmodic recruitments start
8. asrparam: a-parameter in SR function
9. bsrparam: b-parameter in SR function
10. sigmar: dispersion parameter for noise around SR function
11 trunc1: lower level for truncation
12. trunc2: upper level for truncation
13. depens: Depensation parameter
14. amplitude: amplitude in periodic function
15. period: period in periodic function
16. phase: phase in periodic function
17. interval: Interval between spasmodic years
18. intvcv: Dispersion parameter for spasmodic intervals

19. spasmfact: multiplier in spasmodic years

Example. Data for regime 2, iterations 1 to 1000

```
2      1      1      1      2      3     -2  4272.   1840.  0.287  0.50   1.50   1.0    0.0    30.0   -15.0   6.21   0.60  1.0
       .
       .
2   1000      1      1      2      3     -2  4272.   1840.  0.287  0.50   1.50   1.0    0.0    30.0   -15.0   6.21   0.60  1.0
```

3.3.Output.

The output is to files. All the file names are hard coded. The user specifies which of these files to produce in the opt.inn file (see above). In addition, two more files are generated. One is a scratch file called 'kladd'. It is used mostly for dumping intermediate results during debugging, and the routine versions of the program leave it empty. The other is called inputlist and is a printout of the valid lines in the bio.inn and opt.inn files with line numbers. It can be useful for tracing errors in input files.

Some output files have one section for each run option (set of of rule parameters). Each section is headed with a listing of all options for that run. Summary files (***Years y1-y2***) have one line for each option, with all option parameters and some selected summary statistics for that option). These files are best read by importing them into a spreadsheet.

A third group of files come from the analysis of yield per recruit.

3.3.1. Interpretation of outputs:

Some headers on the files are somewhat cryptic and are explained here.

11: Results-file:
This is the printout of the main conditioning options and results (catch, F, SSB, recruitment, IAV and risks). It hopefully is mostly self-explanatory. The percentage inter-annual variation IAV is defined as *[TAC(y)-TAC(y-1)]/ [(TAC(y-1)+TAC(y)) /2]*100, i.e.* the change from one year to the next relative to the average in those years. The mean IAV in the top table is the mean absolute IAV, while the table with fractiles is the signed IAV. The probability of crashing the stock is the cumulated probability that either the stock is below 1/10 of Blim or that the decided catch would require a fishing mortality higher than the possible maximum.
There is one set of tables for each run option, and the conditioning is listed at the top.

A shorter version (only the first table for each scenario) is copied on a separate file: ***MainResults.***

12: Years y1-y2 files
These files have one line for each scenario, with the conditioning for the scenario and mean values over the indicated period, taken over all bootstrap replicas.
You can have as many of these files as you wish, indicate the number of files on the first line and the year intervals (relative to the starting year) on the following lines. The sets of years and the number of files indicated must match. If years are beyond the maximum number of years, the maximum number is used automatically. The file(s) are renamed accordingly.

The first columns are the rule parameters, the actual parameters depend on the rule.

The results:

Cmean C10 C50 C90	Mean and 10, 50 and 90 percentiles for realized catch over all

	the years in the time period and all the bootstrap replicas.
Fmean F10 F50 F90	Mean and 10, 50 and 90 percentiles for realized fishing mortality over all the years in the time period and all the bootstrap replicas.
Smean S10 S50 S90	Mean and 10, 50 and 90 percentiles for SSB over the years in the time period of all the bootstrap replicas.
IAV:	Mean (over all years in the time frame and all bootstrap replicas) of the absolute inter-annual variation in percent: *abs{[TAC(y)-TAC(y-1)]/[(TAC(y-1)+TAC(y)) /2]*100}*.
Risklim:	See note above
Probtrig:	The probability that the Basis(1) is below the first breakpoint in the harvest rule (Btrigger 1). Calculated as Blim.
Crash:	Probability that the agreed TAC could not be taken due to shortage of fish, or that the SSB was below 1/10 of Blim at any time before the end of the time frame. Note that this is a cumulated probability while the Risklim is the maximum annual risk. The risk of crash may be larger late in the time course.

13: Autoc_CVS0:

*Distributions (mean and 10-50-90 percentiles) of the autocorrelation*s in the iterations over the time course for :

ST: True SSB
SOD: Difference true - observed SSB
R: Recruitment

Distribution of observed SSB (S0): Mean and 10-50-90 percentiles

15: Traj-file:

Time trajectories of essential results. On this file some results are printed for all years for each bootstrap replica. Only the first 20 replicas are printed for each scenario. The variables are:

- TAC as decided
- Basis values as seen by the decision model (Basis 1 if there are several). It is the measure (SSB, TSB or Recruitment) that is presented for the actual year
- Percentage change in TAC from the previous year. Note that this is different from the IAV used elsewhere.
- True basis values, corresponding to the decision basis values above.
- True F
- Number of recruits
- Annual surplus production: TSB(y+1)-TSB(y) + TAC(y)

16: recovery: Prints the probability of recovery within N years, N = 1 - 5, when SSB drops below Blim. The number of incidents of drops below Blim is counted over all replicas and over all years where the stock can still be followed for another N years. A drop is defined a true SSB < Blim in one year, but not in the year before. For each of the N time intervals, the number of 'drops' where recovery was in place in the year N after the drop, was counted, and expressed as percent of the drops. There is one line with the number of relevant drops, and one with percentage recovery.

The files ***'17: Yieldrecr', '18: Catch_dist' and '19: SSB_dist'*** are generated by the yield per recruit routine, as described below.

20: SRpairs - ***file***: Stock-recruit pairs
Iteration number, year True SSB and the recruitment generated by that SSB

21: S0distr - ***file***: Initial SSB-values
This file lists all SSB values in year 0. It may be useful for looking at the distribution of initial conditions. The main characteristics are printed on the Autoc_CVS0-file (13)

22: Fdistr - ***file***: Assumed and true fishing mortalities.
For every iteration and year, it prints iteration number, year, decision F and true F. The decided F is the F

corresponding to the decided TAC and the observation model N-values.

23: IAVdistr - ***file***: Distribution of IAV
Iteration number, year, IAV-value in percent, defined as *[TAC(y)-TAC(y-1)]/[(TAC(y-1)+TAC(y)) /2]*100*.

24: Compbas: Comparing all the bases used for decision with the values that would have been obtained using true stock numbers. Only results for the first 100 iterations are presented.

25: wema: Prints weights and maturities by age and year

26: Comprecr. All true and assumed recruitments. The assumed recruitments are the numbers at youngest age that appear in the projection in the TAC year. They depend on the assumption made in the projection, and should be used for checking how that works.

Warning: Some of the files, in particular those in the lower part of the list, may become very large when screening over a large number of scenarios. They are intended to control distributions and ensure that the model is properly conditioned, and it is recommended not to use them in routine screening.

3.4. Yield per recruit

Yield per recruit is calculated routinely with the data on the bio.inn file, for a range of fixed *F*-values from 0 up to *Fcrash* (or to 1.0 if *Fcrash* > 1.0). The yield per recruit routine is run before any of the simulations. If only the yield per recruit is of interest, the run can be broken (CtrlC) when the first option run starts. If multiple recruitment regimes are specified, the calculations are repeated for each regime.

The results that are presented also take into account the stock-recruit relation and the dependence of weights and maturities on TSB. Hence, for each level of *F*, the equilibrium SSB and catch is presented, scaled to the equilibrium recruitment (according to the deterministic stock-recruit function) and weights and maturities. If a pure Yield per recruit calculation is wanted, make a run where the b_parameter in the stock-recruit function is set to zero and there is no density dependence of weights and maturities.

Periodic or autoregressive fluctuations in recruitment are not taken into account, the *a* and *b* parameters in the stock-recruit function are used as they stand. However, if spasmodic recruitment is stated, the scaling is to the long term average recruitment. Density dependent weights and maturities are accounted for - the equilibrium weights and maturities are according to the equilibrium TSB, but year class effects and time trends are not taken into account.

If individual SR-parameters for each iteration are read from a file instead of taken from the bio.inn file, the yield per recruit is calculated with the parameters stated for the first iteration.

The a-parameter in the stock-recruit relation is adjusted to take asymmetry in the recruitment distribution induced by truncation into account. This is done by integrating the probability density function of recruitments over the interval specified by the truncation, and take the mean as sum of products of probability and stochastic variable. As a diagnostic on truncation, a stock-recruit curve can be plotted with the SSB and recruitment are that are printed for each F-level, and compared with the stock-recruit relation originally stated.

In addition to yields, SSB and TSB, the mean age of the catch and spawning stock are calculated as the sum of age_times_biomass of fish at age in the catch or spawning, divided by the total catch or SSB.

A search is made to find *F0.1* where the slope of the yield curve is 1/10 of the slope at the origin. An approximate *Fmax* is at the F where the catch is largest. Since the scanning is stopped at *Fcrash*, the highest catch may be just there. If *F0.1* is above *Fcrash*, -1 is returned. Note that F0.1 and Fmax refer to the production curve, taking the stock-recruitment and the density dependence into account.

The output is on 3 files if asked for:

17 'Yieldrecr': Yield & SSB per recruit, adjusted for the stock-recruit relation:
On top, the input data are listed.
Then for each F:

- F-factor;
- Yield
- SSB;
- %of virgin biomass;
- Y/SSB;
- Mean age of catch;
- Mean age of the spawning stock;
- Recruits

In addition, F0.1 and Fmax are printed on this file, with values of F, Yield and SSB.
Please note that if Fmax is not reached at F=1.0, F=1.0 is reported as Fmax.

18 'Catch_dist': Equilibrium age distribution in the catch
19 'SSB_dist' : Equilibrium age distribution in the spawning stock

3.5. Screen output.

When the program is started, F0.1 and Fmax, with corresponding catch and SSB are displayed, separately for each recruitment scenario.
For each option run, all option parameters are listed (sequence as in the input) when each option is started. If any collecting tables over time frames is asked for (Files 12), some average results for the first specified year span are displayed when the run is done. This is mostly to allow the user to monitor the progress.

4. Some practical advice:

1. It sometimes happens that the program crashes when reading the input files. The most common error is additional or missing lines at the end of the file, for example missing historic SSBs at the end of the nin.xxx when the recruiting age is higher than 0. Other common causes are missing lines, missing spaces or too few entries on some line. If you change the number of ages, make sure that all ages (and no more) are represented on all the input files. In particular, the number of lines specifying observation noise at age in the opt.inn file must be adjusted. Fortran is extremely sensitive to such things, it takes everything literally and never understands what we really mean to say. And if you get frustrated because the program will not read your files, you are not the first one with that experience.

2. Some error messages refer to the line in the program where the error occurs. It may sometimes be useful to look up that line in the program code, just to see what goes on there. When running under Windows, it is useful to start the program in a dos-widow, else the error messages disappear immediately.

3. It is very easy to make mistakes when setting up the run options, weight and maturity models and stock recruit functions. **Never trust that options on an old file are** OK **if you don't quite understand them. Read the definitions, use the manual, check every line carefully, and use the output opportunities to control that you get what you want.**

4. Note that the file formats very often has changed with new versions.

5. Be careful with the distribution and truncation of recruitments. Asymmetric limits give a skewed distribution. A useful procedure is to take recruitments from the file 'srpairs' (number 20) into a spreadsheet or R, make a cumulated distribution and compare that with the cumulated distribution of historical recruitments that you want to reproduce, Often, some trial and error is needed to get it right.

6. Resist the temptation to scan over everything. The best advice is to be problem-oriented and systematic.

With numerous options, the number of runs becomes very large, and even if the program is fast (each option will normally take 10 - 30 seconds on a modern PC) hundreds of options do take time. Afterwards, you will experience that presenting this vast amount of information in an understandable way is a major challenge.

7. All output file names are hard coded, and all files are over-written without warning. Orderly book-keeping saves much work, waiting for re-runs and frustration.
8. If you need to change the year range or the age range, you will need a Fortran compiler that handle F77. I am used to gfortran that is part of the GNU gcc package. This is free software. Many alternatives exist, some are more expensive than others. Go into the include file hcscom17_1.i. On the very top you will find something that looks like this:

```
c hcscom13_5
c include file for hcs
      parameter(maxage=10,maxyr=40,maxyrp=maxyr+maxage,niter=1000,
     &mbasis=5,nopt=26,nityr=niter*maxyr)
```

Change the maxyr and/or the maxage to the numbers you want, save and compile hcs again: With gfortran the command is gfortran hcs17_1.f in the folder where you have the code. The maxyr should not be more than 98 - higher numbers create trouble in some printouts.