Time-Varying Parameter Values for SS3.30

The approach to allowing parameters to have time-varying values has been completely overhauled in the transition from SS3.24 to SS3.30. Fortunately, the sstrans.exe will do the conversion for you, but you should review the new control file closely before simply running with it, especially for time-varying catchability parameters.

Motivation: In SS3.24, the group of biology parameters (termed MGparm) and the selectivity parameters used the same long parameter line approach, but it was implemented with entirely different code, and hence was inefficient. The spawner-recruitment parameters used short parameter lines and a different approach for linkage to an environmental variable and the R1 offset provided a limited type of block. The catchability parameters also used short parameter lines and had its own approach to doing environmental linkage and random deviations, but not blocks. The tagging parameters had long parameter lines, but there was no code to interpret any time-varying info in those lines. The situation was begging for a more modular approach.

New Code Flow: In SS3.30, MG, Sel, SRR, Q, and Tag base parameters all will use long parameter lines and will invoke blocks, trends, environmental linkages, and random deviations using identical syntax. As SS3.30 executes the SS\_readcontrol code, it calls a function in SS\_global called “create\_timevary” whenever a base parameter has any one of the 4 types of time-varying options. In fact, block/trend, env and devs all can be applied to the same base parameter. Only blocks and trends are mutually exclusive. “Create\_timevary” creates all needed information to describe and index a list of time varying parameter specifications. Then as SS gets into iterative parameter updating it starts by calling a function in SS\_timevaryparm that processes each time varying parameter specification (each of which can contain any combination of block/trend, env and dev specs) and creates a time series of the parameter value that are used as SS subsequently loops through the years.

New Parameter Order: The order of parameters has changed and the re-ordering is handled by the sstrans.exe. Previously, for each of MG and Sel parameters all env link parameters were listed first, then block/trend parameters and then dev parameters. In SS3.30, these parameters are re-organized such that all parameters that affect a base parameter are clustered together with block/trend first, then env, then dev. So, if MG base parameters 3 and 7 had time varying changes, the order would look like:

|  |  |
| --- | --- |
| MG base parm 3 | Block parm 3-1 |
|  | Block parm 3-2 |
|  | Env link parm 3-1 |
|  | Dev se parm 3 |
|  | Dev rho parm 3 |
| MG base parm 7 | Block parm 7-1 |
|  | Dev se parm 7 |
|  | Dev rho parm 7 |

Link Functions: The functional form by which a time vary parameter, Q, changes a base parameter, P, is a link function: P’(y)=f(P,Q). Typically this is additive or multiplicative function, but the parameter mirroring feature is essentially a link that takes no parameter. Another type of link in SS is between a model state variable, such as available biomass, and the expected value for a survey. Typically, this is a simple proportional link taking one parameter, q. But the q power feature is essentially a 2 parameter link function. So, a parameter link function can change q over time, and a survey link function then uses the annual value of q to link the annual value of a state variable to the expected value for a survey. In SS 3.24, various usages of positive and negative codes and other conventions were used to invoke additive vs multiplication links and other options. But as 3.30 builds capability to allow an environment index to be a “survey” of a parameter deviation, we need a larger family of link functions such as logistic and even dome-shaped.

The table below shows the link specifications in 3.24 and the corresponding link specification in 3.30. Take special note of the env linkage specification where two bits of information are coded into one number. An env link specification of 204 is parsed by SS to use link type 2 using env variable 4.

In version 3.24, the linkage options were:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Environ | Annual Devs | | | | Blocks | |
| NAME: | Env Var | Use dev | Dev minyr | Dev maxyr | Dev  stddev | Block | Block Fxn |
| ELEMENT: | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| OPTIONS: | >0: mult | 1: mult | 1973 | 1985 | 0.4 | >0:block index | 0: mult |
|  | <0: additive | 2: additive |  |  |  |  | 1: additive |
|  | abs(value): env index | 3: additive  randwalk |  |  |  |  | 2: replace |
|  |  | 4: mean reverting randwalk |  |  |  |  | 3: randwalk |
|  |  |  |  |  |  | -1: trend with final as offset from baseparm, and offset value is in log space\* | NA |
|  |  |  |  |  |  | -2: trend with final as standalone value | NA |
|  |  |  |  |  |  | <=-3: cycle with one parameter per season | NA |

The available options for time-varying parameters in SS 3.30 are described in the table below.

\*\*Fractional approach inflection year for trend. If the trend parameter upper bound is set <=1.1, then the parameter is treated as being the fraction of the time between start year and end year, else the parameter is treated as having units of year.

\*Logistic approach to trend as offset from baseparm:

temp=log((MGparm\_1(j,2)-MGparm\_1(j,1)+0.0000002)/(MGparm(j)-MGparm\_1(j,1)+0.0000001)-1.)/(-2.); // transform the base parameter

temp+=MGparm(k+1); // add the offset Note that offset value is in the transform space

temp1=MGparm\_1(j,1)+(MGparm\_1(j,2)-MGparm\_1(j,1))/(1.+mfexp(-2.\*temp)); // backtransform

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Environ | Annual Devs | | | | Blocks | |
| NAME: | Env Link & Var | Dev Link | Dev minyr | Dev maxyr | Dev  Phase\* | Block | Block Fxn |
| ELEMENT: | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| PARSING: | Link = int(value/100.)  Env\_var = value-Link\*100. |  |  |  |  |  |  |
| OPTIONS: | Link=1: P’=P\*exp(L1\*env) | 1: mult | 1973 | 1985 | <integer> | >0:block index | 0: mult; \*exp(X) |
|  | Link=2: P’=P+L1\*env | 2: additive |  |  |  |  | 1: additive, +X |
|  | Link=3:  Reserve for density dependence | Random walk options are now implemented by using rho in the objective function |  |  |  |  | 2: replace; =X |
|  | Link 4: logistic  P’=P\*2.0/  (1+exp(-L1\*(env-L2)) |  |  |  |  | 3: additive randwalk across blocks |
|  |  |  |  |  |  | -1: trend with final as offset from baseparm, and offset value is in log space\*\*; also inflection year is in log space and an offset from log(0.5) | NA |
|  |  |  |  |  |  | -2: trend with final as standalone value | NA |
|  |  |  |  |  |  | -3: end value is a fraction of baseparm max-min; inflection year is as fraction of endyr-styr | NA |
|  |  |  |  |  |  | <=-4: cycle with one parameter per season |  |

\*\*Logistic approach to trend as offset from baseparm:

temp=log((MGparm\_1(j,2)-MGparm\_1(j,1)+0.0000002)/(MGparm(j)-MGparm\_1(j,1)+0.0000001)-1.)/(-2.); // transform the base parameter

temp+=MGparm(k+1); // add the offset Note that offset value is in the transform space

temp1=MGparm\_1(j,1)+(MGparm\_1(j,2)-MGparm\_1(j,1))/(1.+mfexp(-2.\*temp)); // backtransform

\*Dev\_se is now a parameter and so is dev\_rho. This available element in the long parameter line is now a dev\_vector specific phase control.