PACIFIC SALMON COMMISSION
JOINT CHINOOK TECHNICAL COMMITTEE

MONTE CARLO SIMULATION FOR CTC MODEL IMPROVEMENT

Prepared by

Li Ding

Vitech Innovative Research and Consulting

September, 2018

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The PSC Chinook Technical Committee (CTC) has employed deterministic, assumption-based analytical methods and models since the early 1980s. In large part, these methods and models are based on analysis of Coded Wire Tag recovery data and do not consider the effects of uncertainties involved in assumed fish behavior, hypothetical analytical biological and fishing structures, and parameter values. The Data Generation Model (DGM) was developed to produce datasets of TRUE mortalities which can be used to assess the performance of methods and models employed by the CTC.

The model consists of two components: Data Generation Module (DGM) that simulates catch and mortality of individual fish at specific fisheries, and Sampling Module (SM) that simulates recovery of coded-wire-tags (CWTs) from databases generated by the data generation module, given user-specified sampling rates for fisheries and spawning escapements. AABM, ISBM, and Terminal fisheries and spawning escapements are simulated.

The DGM model was implemented as a standalone software system on the Microsoft Visual Studio platform using the Visual Basic programming language. The system also includes a database component for management of input and output files. Microsoft Access was used to develop the database management component. With this system, the user is able to simulate a wide variety of scenarios by providing a set of input data files which specify the simulation configuration (number of stocks, ages, time periods, and fisheries, migration mechanism, sizes of marked and unmarked releases), scenario (time-specific fishery harvest rates and retention restrictions), and the number of times the scenario is to be simulated. For each specified scenario, the simulation module generates a database of catches and escapements by stock, age, and fishery, and the SM module simulates CWT recovery on a given DGM data set. Each set of simulated CWT recoveries is saved independently in its own database for analysis. Management of these datasets is supported by the database component.

This report summarizes the algorithms, procedures, and input/output data structures used in the simulator, while a separate user manual will be released with source code.

.

1 INTRODUCTION

The PSC Chinook Technical Committee (CTC) has employed deterministic, assumption-based analytical methods and models since the early 1980s. In large part, these methods and models are based on analysis of Coded Wire Tag recovery data and do not consider the effects of uncertainties involved in assumed fish behavior, hypothetical analytical biological and fishing structures, and parameter values. The Data Generation Model (DGM) was developed to produce datasets of TRUE mortalities which can be used to assess the performance of methods and model employed by the CTC. The DGM consists of two components:

(1) **Monte Carlo Data Generation Module (DGM):** This module simulates, given a stock distribution, selective and non-selective fisheries encompassing appropriate types of release mortalities in the presence of uncertainty and variability. It simulates encounters experienced by individual fish in fishing processes that simultaneously harvest fish found within a given area. Detailed mortalities (landed catches, incidental) are recorded for marked and unmarked components of individual stocks. Mortality statistics include stock-age specific mortality by fishery and spawning escapements during different time periods, for four components, namely, unmarked and untagged (wild), marked and untagged (Mass marked), unmarked and tagged (unmarked DIT), and marked and tagged (marked DIT or single index tagged). AABM, ISBM, and Terminal fisheries and spawning escapements are simulated. The user is able to simulate a wide variety of scenarios by providing a set of input data files which specify the simulation configuration (number of stocks, ages, time periods, and fisheries, migration mechanism, sizes of marked and unmarked releases), scenario (time-specific fishery harvest rates and retention restrictions), and number of times the scenario is to be simulated. For each specified scenario, the simulation module generates a database of catches and escapements by stock, age, and fishery.

(2) **CWT Sampling Module (SM):** The SM simulates recovery of coded-wire-tags (CWTs) from databases generated by the simulation module, given user-specified sampling rates for fisheries and spawning escapements under visual vs. electronic tag detection methods. The user can specify the number of times the sampling process is to be simulated on a given DGM data set. Each set of simulated CWT recoveries is saved independently in its own database for analysis.

This system was developed on the Microsoft Visual Studio platform using the Visual Basic programming language. The system also includes a database component for management of input and output files. Microsoft Access was used to develop the database management component.

2 OVERVIEW AND MODEL SETUP

2.1 Overview

The DGM system was built on a technical memo provided by Gary Morishima (2009). Figure 1 shows an overview of DGM, which consists of two major blocks: pre-terminal fishing and terminal fishing. Pre-terminal fishing is simulated for the number of regions and periods specified by the user for each year. At the end of each period, fish are redistributed across regions or may mature (maturation period is specified by stock). When maturation occurs, a portion of the population at each age matures and is subject to terminal fishing. The immature population that remains is aged, natural mortality is applied, and the fish are subjected to preterminal fishing in the next time period. At the end of terminal fishing, escapements are calculated, and the production of the next generation is calculated and added to the Age 1 (or 2) population. The entire loop is repeated by selecting a different random seed. Statistics on cohort sizes, mortalities, landed catch and mortalities are saved to text files for further analysis. Each block in the diagram is described in Section 3.

The simulated data (pre-terminal catch, terminal catch, and escapement) output from the DGM module are sampled, without replacement, to simulate the CWT recovery. The sampling process requires a set of sampling rates specified for each period. Sampling is repeated until a desired number of sampled fish is reached. Using these data sets, the effects of visual or electronic sampling can be evaluated by the user at the time of analysis.

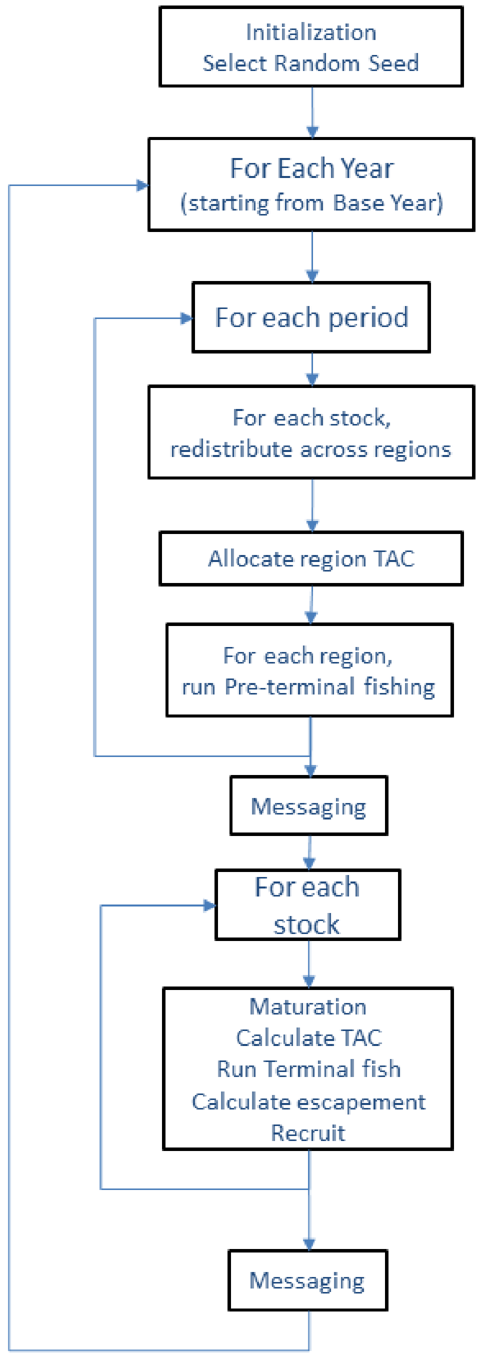


Figure 1: DGM flow chart

2.2 Software System

The current version software was built on Microsoft Visual Studio Community Edition 2015. Here we provide an overview of the system. For software installation and operation, refer to a separate user manual (Vitech 2018). The system is deployed as a standalone program on MS Windows 7, 8 and 10. As shown in Fig. 2, the system consists of the following modules:

1. **Data Generation Module**: As described above, this module generates catch and mortality data in pre-terminal fisheries, terminal fisheries, and escapement.
2. **Sampling Module**: It includes three functions: pre-terminal sampling, terminal sampling, and escapement sampling.
3. **Database Module:** The database is a container for the latest DGM input data provided or modified by the user and model configuration data.
4. **File Reader/Writer:** the system also includes a file reader and a file writer that read and write input/output files according to file formats in Appendix A.

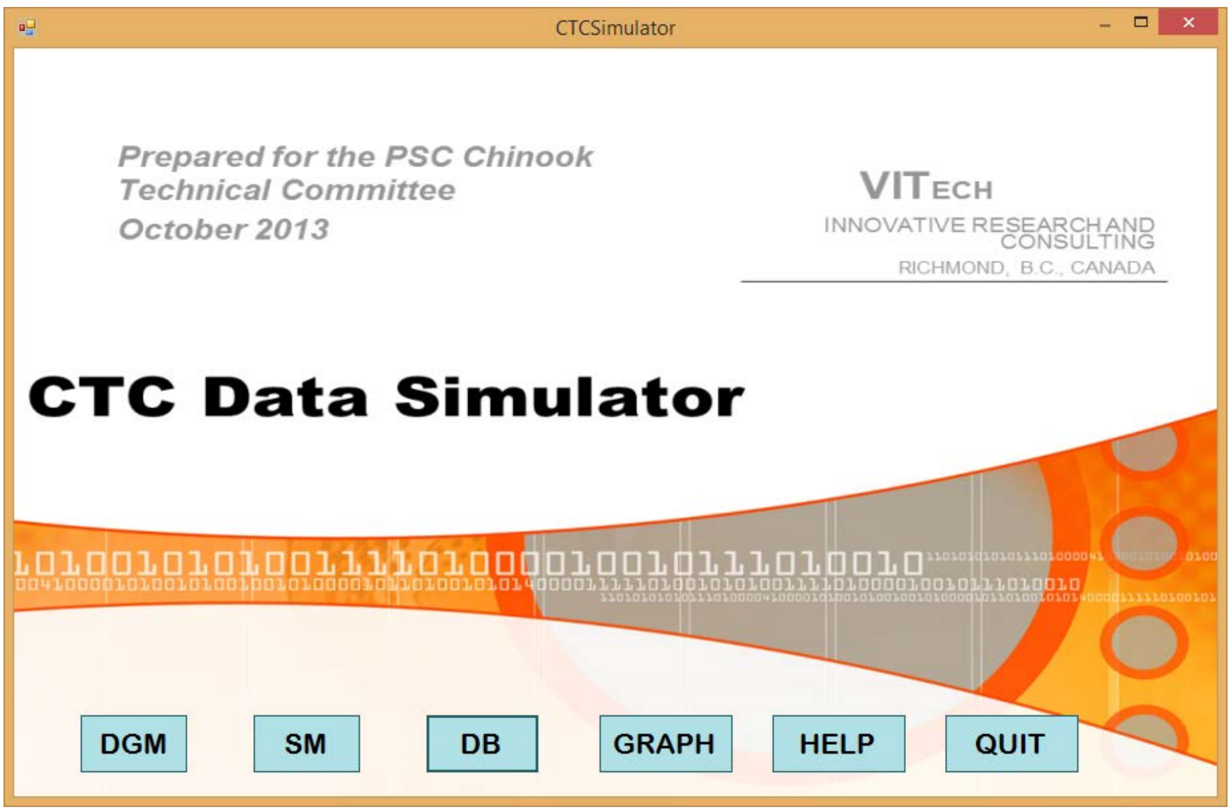


Figure 2: Startup Interface

2.3 Running DGM Model

We start from the data generation module, whose interface is shown in Fig. 3. Below are the procedure steps:

1. When the user starts a simulation run, the first step is to provide input data to the system. There are three options as shown in Fig. 3:
    a. Use Data from a control file. Go to File→Load DGM Control File. This loads a control file which points to a number of file names that contain necessary input data for the model to run (see Section 2.4 for more details).
    b. Use Data from a Previous Simulation. If this option is selected, the user chooses a simulation ID, and the program will use the input data associated with the selected simulation ID.
    c. Use Current Data in DB. The third option is to use the data currently in the database. For more details, see the user manual (Vitech 2018).

2. Selects a data store folder. There is a default path in the 'Data Store Folder', but the user can go to File→Select Data Store Folder to change to a target folder.

3. Enter a simulation ID, which will be used to create a subfolder under the data store where the system will save output data and a copy of the input data. The subfolder path and other run parameters are also saved to the database.

4. Select run parameters including the number of repeated simulations to run, and the number of accounting years (Currently base year is fixed at 1979, and the number of periods per year fixed at 3). For each run, simulation is performed for the periods in a year and for all the accounting years.

5. For each run, the random number seeds used to generate random numbers are automatically generated. If the user checks 'Enter an integer as RND seed', then this number will be used in the current run (only one simulation run will be generated).

6. Select a method (single or multi pool) for stock distribution across regions.

7. If the user wants to provide a specific lookup table for the conversion of abundance index (AI) to total allowable catch (TAC) in AABM fisheries (see Section 3 below), go to File→Load AABM TAC Lookup Table.

When DMG starts, messages will be printed onto the Progress Status box. After DGM is completed, the user can proceed to run the SM module. Click on the 'SM' tab, the interface will be as shown in Fig. 4. The user has to select a DGM ID (i.e. use the DGM output data associated with the ID), and enter an SM ID and the number of SM runs. Then the user loads a sampling rate data file from the File menu, and starts an SM run from the Run menu, on preterminal catches, terminal catch, or escapements.

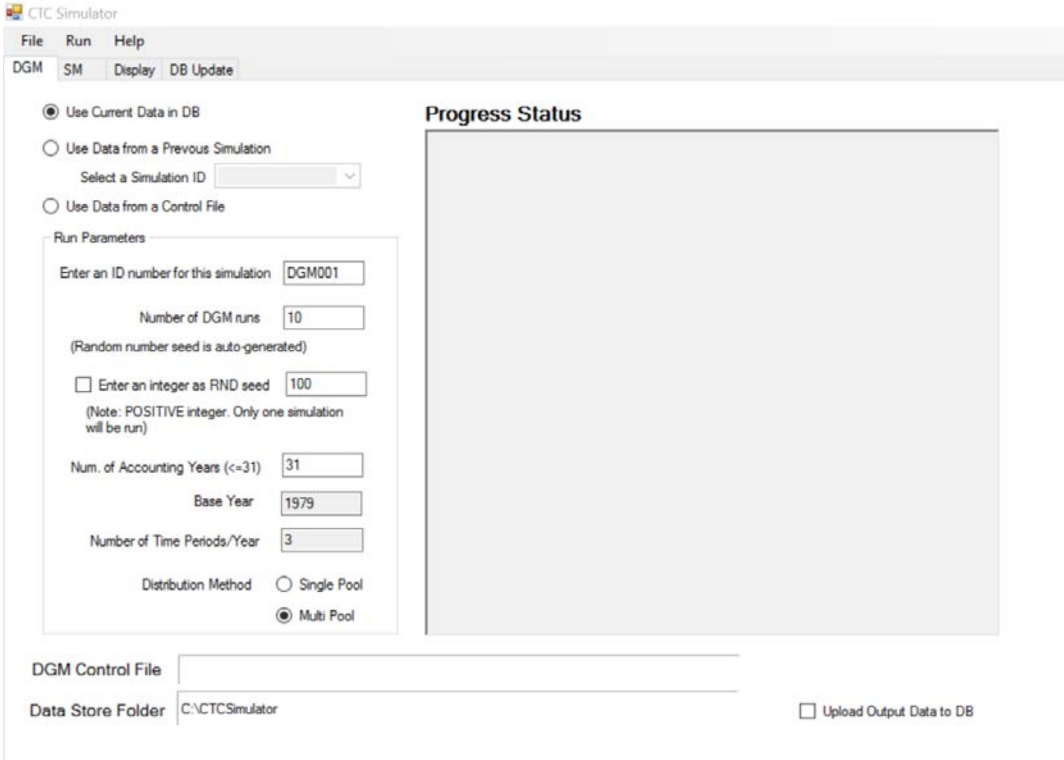


Figure 3: DGM Module Interface

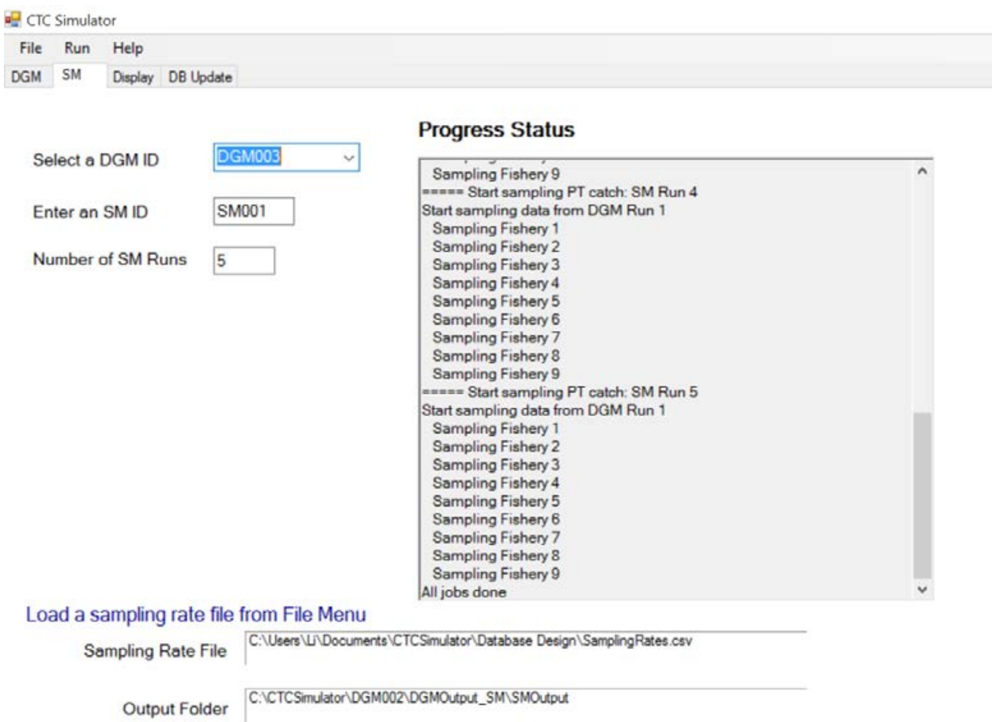


Figure 4: User input interface for Sampling Module

2.4 Input Data Structures

Model configurations and simulation scenarios are defined in a set of input files as listed below. To load these files into the system, the basic option is to create a control file, which points to the paths of the files. The program reads the control file first, and then reads the corresponding input files provided by the control file.

The input files are essentially text-based data tables with formats and column fields defined in Appendix A. The user can use a text editor to enter data based on the format. We have also prepared a set of input files as example in the release package. Here we provide a description of these files:

1. CTCStock.csv: defines basic properties for stock, such as type (natural or hatchery), residence (ocean or stream), maturation period, and Ricker coefficients. This is the master file for stock definitions. A number of other properties linked to the stock IDs in this file are provided in separate files. The stock IDs in this file must be unique.

2. CohortSize_Age.csv: provides the cohort size of a stock at different ages in the base year. There is a column for total abundance, with partitions into unmarked, unmarked with CWT, marked, marked with CWT. The sum of the partitions should be equal to the total.

3. Stock_Age_Size.csv: provides fish length (mean and SD) of a stock at different ages and in different time periods.

4. Stock_MatRate.csv: provides year specific maturation rates (mean and SD) of stocks at different ages.

5. Stock_PrespawnMort.csv: provides year-specific pre-spawn mortality of stocks. If the rate is not specified here for a stock, it is considered to be zero.

6. Stock_TermMortRetention.csv: provides year-specific data in terminal fisheries, including mortalities (release and drop-off) and marked retention data. There is a flag in the table, 'IsMSF', which indicates whether the fishery is a mark selective fishery. There is also a field called 'Max_Unmarked_Fish_Allowed'. If this value is zero and IsMSF is set, then the fishery is a straight MSF.

7. Stock_TermHR.csv: provides age-specific terminal harvest rates of stocks and their size limits.

8. Stock_TermHRScalar.csv: provides year-specific stock terminal harvest rate scalars, for different sectors.

9. GoupSurvival.csv: provides group-based survival rates (mean and SD) at different ages and in different years. The group a stock is associated with can be looked up in the 'CTCStock' table.

10. CohortRegionDist_SinglePool.csv: provides regional distribution coefficients of a stock at different ages and years, from a single pool.

11. CohortRegionDist_MultiPool.csv: provides regional distribution coefficients of a stock at different ages and years, from multi-pools. The coefficients are specified for a source region and a target region.

12. CTCFisheries.csv: defines basic properties for fishery, such as type (AABM, or ISBM, or Terminal). For AABM, set IsAIDriver to True if the fishery is an AI driver. For the rest, set to False. This is the master file for fishery definitions. A number of properties linked to the fishery ID are provided in other files. The fishery ID must be unique.

13. Fishery_SizeLimit.csv: provides fish size limit at a fishery and in different years.

14. Fishery_HrScale.csv: provides fishery specific harvest rate scales in different years.

15. Fishery_Mortality.csv: provides preterminal mortality rates (mean and SD) at a fishery in a particular year. Mortalities include release (legal/sub-legal) and drop off mortalities.

16. Fishery_MarkedRetention.csv: provides year-specific marked retention data in a fishery. There is a flag in the table, ‘IsMSF’, which indicates whether the fishery is a mark selective fishery. There is also a field called ‘Max_Unmarked_Fish_Allowed’. If this value is zero and IsMSF is set, then the fishery is a straight MSF.

17. Fishery_Dist_Period.csv: provides allocation of allowed catch at an AABM fishery across different periods in a specific year.

18. Region_Sector_Alloc.csv: this is a table for regional sector allocation. It is used to allocate total allowed catch in a region to different sectors in a year.

19. BasePeriod_ER.csv: provides base period exploitation rates at different ages of a stock at a fishery and in a certain period.

2.5 Output File Structures

When DGM is completed, the output data are saved to the same folder where the input data are located (DGMOutput_SM in Fig. 5). The output data include catch and mortality data in pre-terminal and terminal fisheries, and escapement data (see Appendix B for more details). If the checkbox, ‘Upload output data to DB’, in Fig. 3 is checked, the system will upload the DGM output data from the current simulation run to the database, in the same structure as in the output files.

When the user starts an SM run, the system will retrieve the project folder path from the database using the DGM ID, and loads the DGM output files (catch data) in the project folder *as the input data of SM*. So the DGM output folder is also the SM run folder, and the sampling rate data file is copied to this folder, with the SM Run ID as the prefix of the file name. The SM output data will also be stored in a subfolder **SMOutput**, as shown in Fig. 5. See also

Appendix B for detailed explanation. Below is a description of DGM outputs.

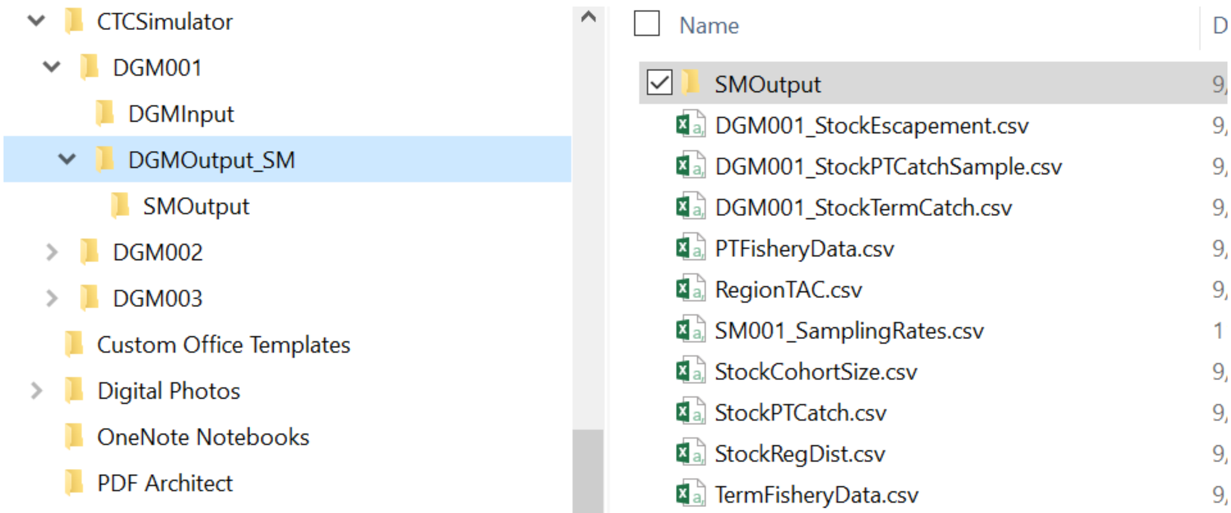


Figure 5: File organization structure of input/output data files for DGM/SM

xxx_StockPTCatch: This file contains preterminal catch data, which is counts of fish caught in each region during a given time period, in a preterminal fishery. It also includes dropoff mortality count and release mortality count.

xxx_StockTermCatch: This is for terminal catch and mortality data, which is counts of fish caught in a terminal fishery in a year.

xxx_Escapement: This contains escapement data, which is counts of fish of a stock that escape during a given period.

The above three files are used in the sampling module. There are additional output files:

PTFisheryData: This records some preterminal fishery data, which monitors AI and TAC (total allowed catch) in a fishery and in a given period.

RegionTAC: This file tracks the regional TAC of a stock in a region.

TermFisheryData: This records terminal fishery data, including terminal run and TAC in a terminal fishery.

StockCohortSize: this file tracks stock cohort size at ages over years.

StockRegDist: this file tracks the regional distribution of stock population over years.

2.6 Database Update

Click the ‘DB’ and the main interface for input data entry and update will be shown (Fig. 6). Note that there are two options for data input: direct input from a control file (which points to individual data files), and manual entry by the user. As can be seen, the left side is for direct update and the right side for manual entry. See the user guide for more details (Vitech 2018). The first option allows the user to avoid having to enter data manually. This is particularly useful for loading a large amount of data such as single/multi pool distribution coefficients, which are too labor intensive for manual entry. The second option allows the user to fine-tune specific data conveniently.

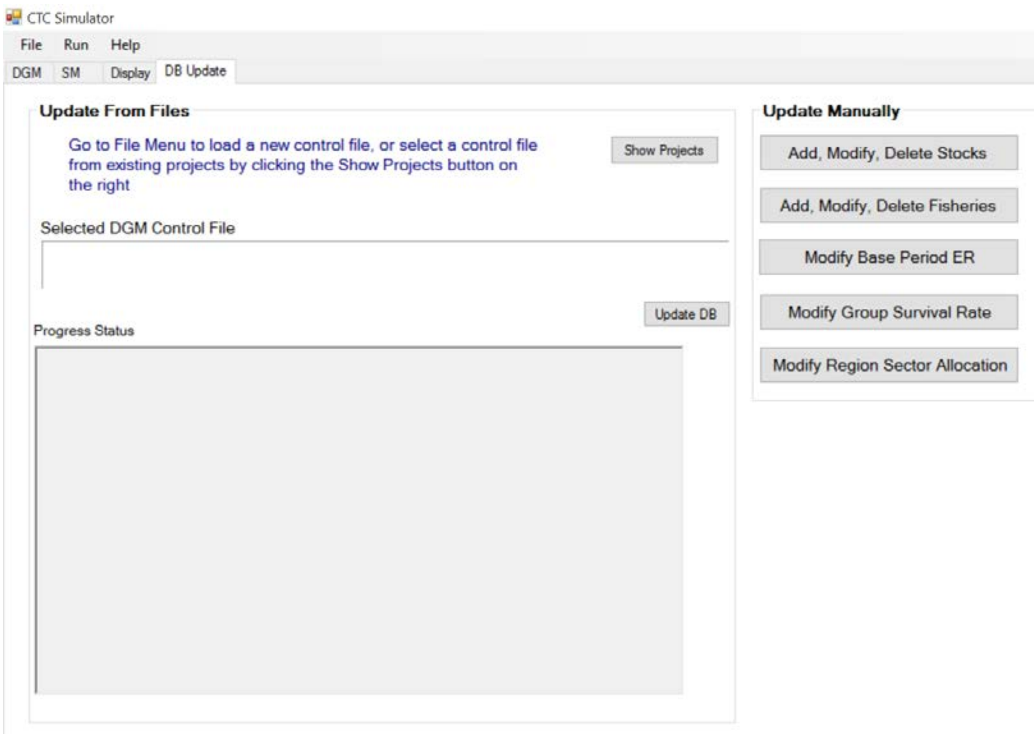


Figure 6: Database Update Module

3 IMPLEMENTATION DETAILS

3.1 Initiate Random Number Seed

The user has the capability to randomize the initial seed or use a specified value to start the sequence of random numbers generated by the DGM. See Appendix D for more details of random number generation.

3.2 Randomization of Input Parameters

A number of input parameters are given in mean and standard deviation. In a simulation run, they are randomized with a Gaussian random number with the corresponding mean and standard deviation. These parameters include:

1. Single pool and multi-pool distribution coefficients (per period)
2. Region sector allocation ratio (per year)
3. Fishery distribution ratio across periods (per year)
4. Mortality rates in pre-terminal fisheries (per year)
5. Mortality rates in terminal fisheries (per year)
6. Stock maturation rates (per year)
7. Stock survival rate (per year)
8. Richer Coefficients (per run)

These parameters are randomized independently except for stock survival rate. The survival rates in Group A (ocean residence) are randomized with a correlation between stocks in the group specified in the stock input file. They are randomized each year with the same correlation (but no cross-year correlation). See Appendix E for generation of correlated Gaussian random number.

3.3 Stock Redistribution across Regions

We have implemented two methods of redistribution: single pool and multi-pool. First, coefficients of distribution, which depend on fish age, region, period in a year, and year, are provided (these two methods have different sets of distribution coefficients). They are randomized using a truncated Gaussian distribution with the value in [0, 1], subject to the constraint that the sum over regions must be equal to one.

In the single-pool method, fish distributions across regions at a time period are obtained by multiplying the total remaining abundance from all regions at that time by the single-pool coefficients. Therefore, these regional distributions are age and time (period and year) dependent. The fish available in each region (r) at the beginning of the period ($RDist_{r,}$) is:

$$RDist_r = \sum_s \sum_a N_{s,a} \bullet d_{s,a,r} \qquad \textbf{(1)}$$

where $N_{s,a,}$ is the total abundance in all regions at the beginning of the time period by stock (s) and age (a) and $d_{s,a,r}$ is the single pool distribution coefficient (see Figure below).

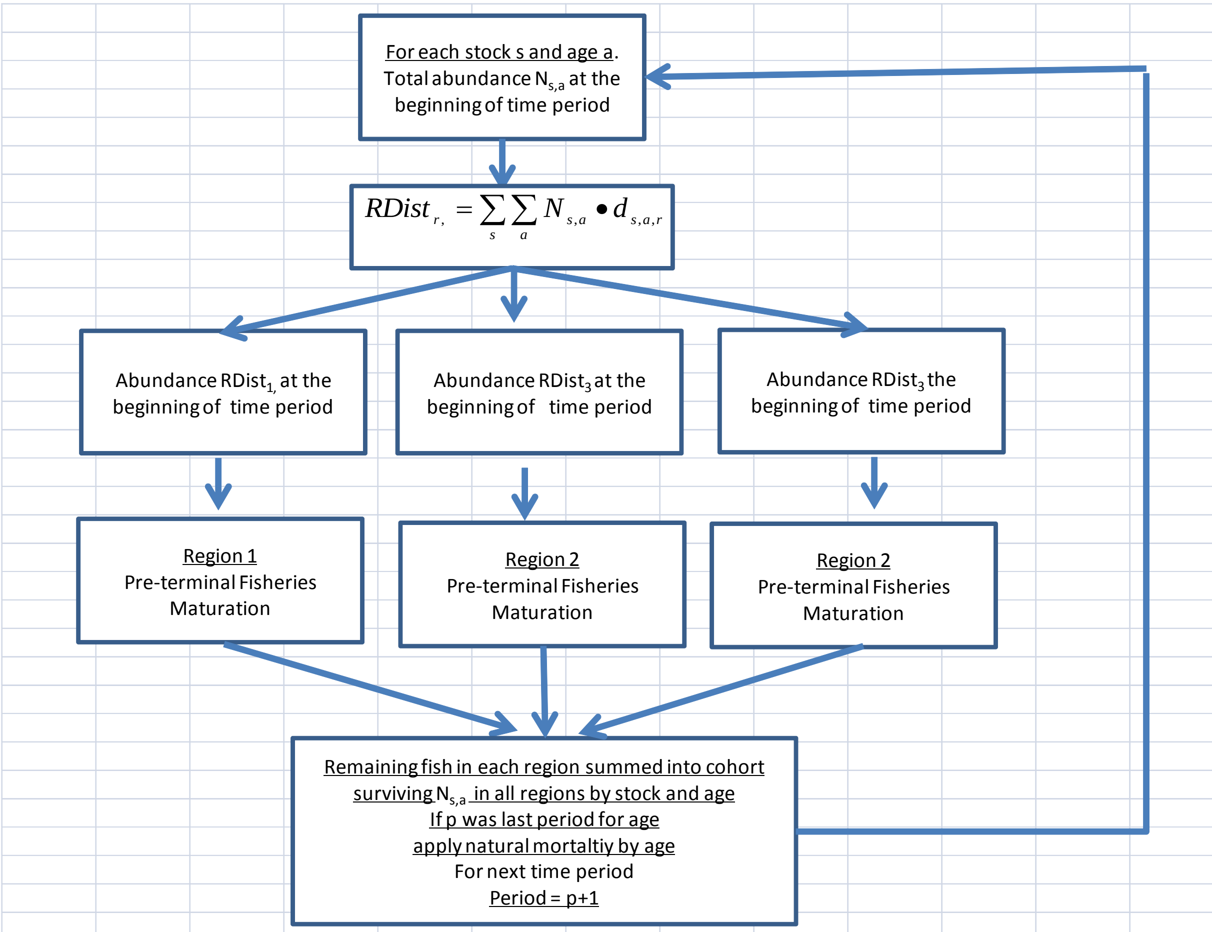


Fig. 7: Single pool distribution

In the multi-pool method, the fish abundance in a region at a time period is the remaining (from the previous period) fish abundance in *each region* multiplied by multi-pool (from region to region) redistribution coefficients. Specifically, this method is (see Fig. 3 below):

- Initial distributions. If time is the first period of the first simulation year, distribute initial cohorts of all ages using the single pool method (because there is no previous distribution). In addition, for the initial ocean age (age 2) in the first time period when a stock recruits to pre-terminal fisheries, use the single pool method to distribute the initial age ocean abundance among regions.
- For years and periods after initial distributions, the multi pool distribution coefficients are used. At the beginning of the period, the abundance in each region ($N_{s,a,rf}$) for each stock *s* and age *a* is redistributed among regions and summed to get the new regional abundance

$$RDist_{r,p+1} = \sum_s \sum_a \sum_{rf} N_{s,a,rf} \bullet d_{s,a,rf,r} \qquad \textbf{(2)}$$

where $RDist_{r,p+1}$ is the abundance in region r for the next period (p+1), $N_{s,a,rf}$ is the abundance remaining in region rf after preterminal fisheries in period p and maturation, and $d_{s,a,rf,r}$ is the redistribution coefficient which gives the proportion of fish remaining in region *rf* that will move to region r. At the end of the last period, the multi pool redistribution coefficients are used to distribute fish to the first period of the next year (with the exception of first ocean age recruits).

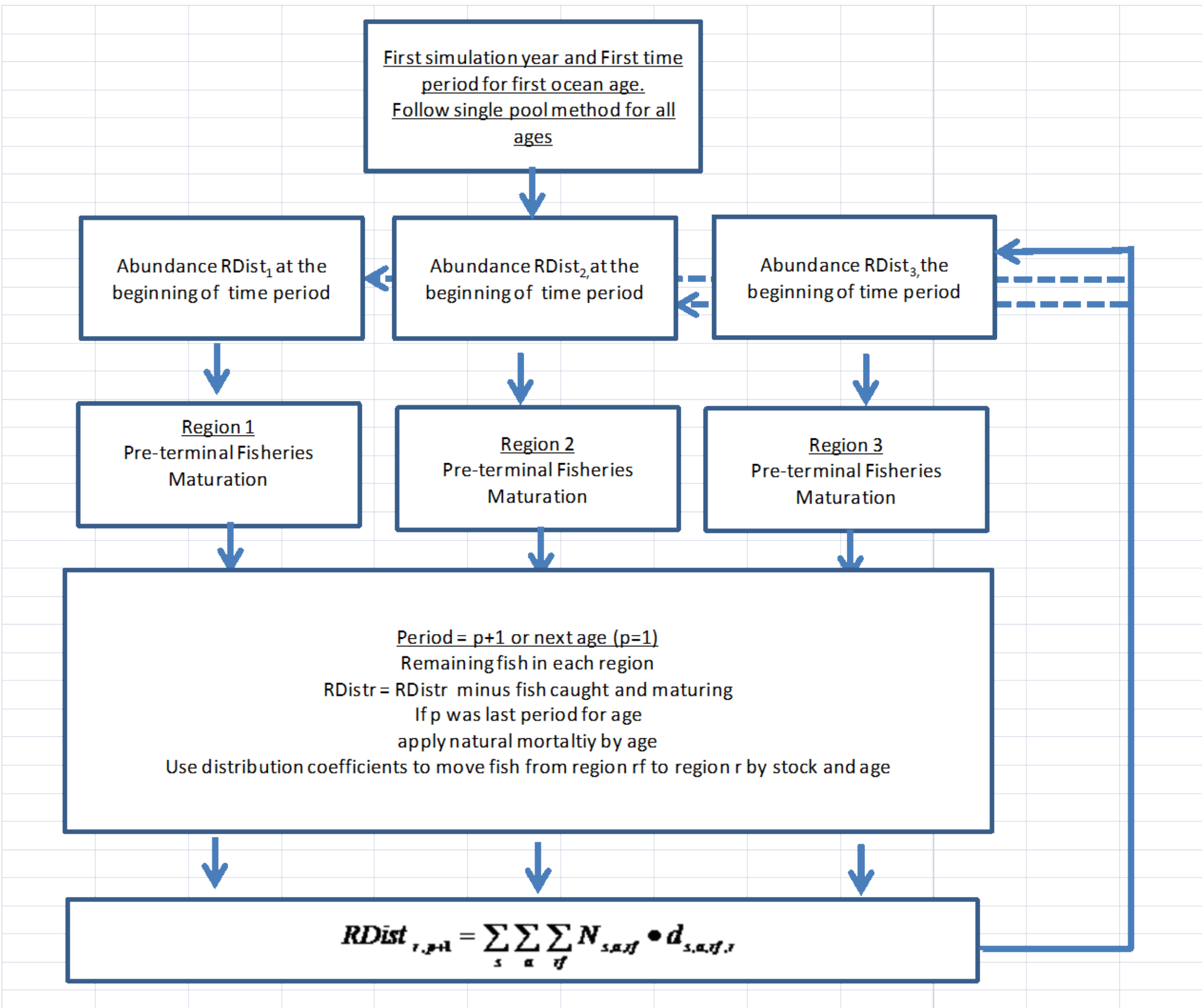


Figure 8: Multi-pool distribution

3.4 Calculation of Preterminal Fishery TAC

Regional total allowed catch (TAC) is calculated separately, depending on the type of fisheries (AABM or ISBM). For AABM fisheries, the TAC is determined by a region specific parameter: Abundance Index (AI), which depends on the catch of the driver fishery in a region (there is only one driver fishery in each region). On the other hand, the TAC of fisheries in an ISBM region is calculated directly as described below.

3.4.1 AABM

For pre terminal AABM (Aggregate Abundance Based Management) fisheries:

1. For each region, compute the catch for the driver fishery (the fishery that is used to establish the allowable catch) using.

$$Catch_{r,f=driver} = \sum_s \sum_a (N_{s,a,r} \bullet d_{s,a,r}) \bullet PV_{s,a,f=driver} \bullet BPER_{s,a,p,f=driver} \quad \textbf{(3)}$$

   where $BPER_{s,a,p,f}$ is the base period exploitation rate and $PV_{s,a,f}$ is the proportion of stock age cohort vulnerable to fishery.

2. Compute the region specific Abundance Index (AI_r), defined as the ratio of the catch of the AI driver fishery in the current period, to the catch in the base period (1979, period 1)

3. Use the regional AI to derive the regional TAC. Each region uses a different set of equations (see Appendix C).

4. The TAC of each individual fishery in an AABM region is then determined by distributing the total regional TAC to individual fisheries, according to the parameters specified in files Fishery_Dist_Period.csv and Region_Sector_Alloc.csv.

3.4.2 ISBM

For ISBM (Individual Stock-Based Management) preterminal fisheries (Net fisheries in Region 2 and 3), the TAC is calculated as,

$$TAC_f = \sum_s \sum_a (N_{s,a} \bullet d_{s,a,r}) \bullet PV_{s,a,f} \bullet BPER_{s,a,p,f} \bullet HRScalar_f \quad \textbf{(4)}$$

where $HRScalar_f$ is the harvest rate scalar input for that fishery.

Note that in Eq. (3) and (4), $PV_{s,a,f}$ is given by

$$PV = 1 - PNV = 1 - \frac{1}{1 + e^{-1.7(\frac{SL-\mu}{\sigma})}}$$

where SL is the lower size limit of the fishery, μ and σ are mean and standard deviation of the fish length at the specified age. PNV is Proportion not vulnerable given in TCCHINOOK (2009).

3.5 Pre-terminal Fishing

Pre-terminal fishing data is generated in the following steps:

1. Select a pre-terminal fishing region.
2. Using the abundance of each individual stock and age in this region, organize a cumulative index for the entire population of fish available in the region, i.e., each portion of the index would represent the number of fish in each stock-age cohort in the region.
3. Randomly generate an integer [1, total number of fish in the region] to simulate the random selection of an encountered fish

4. Determine the stock and age of the selected fish by determining if the integer is less than the cumulative abundance at a stock-age index.
5. Determine the mark type (U, U+, M, M+) of the selected fish:
    a. Find the total abundance of the stock at the selected age.
    b. Calculate the cumulative abundance over the mark types for this stock and age, and normalize it by the total abundance (i.e., compute the proportion of the stock-age that is comprised of U, U+, M and M+ fish).
    c. Generate a uniform random number in [0,1], and if it is less than the cumulative abundance of a mark type, choose this type.
6. Identify the sector in the selected fish is encountered:
    a. Identify all fishery sectors in this region that operate on the same population (generally, troll and sport for preterminal fisheries), and find their remaining TACs.
    b. Compute the total remaining TAC of these sectors by addition and normalize the TACs for individual sectors by the total remaining TAC.
    c. Generate a random number from a uniform [0, 1] distribution.
    d. If this random number is less than the cumulative TAC for a sector, record this fishery ID and its corresponding sector (based on input data files).
7. Simulate drop off mortality:
    a. Randomly generate a random number in [0, 1] to represent the simulated drop-off rate from a normal distribution using mean and SD for the sector in the region. This is done only once each year for each simulation run.
    b. Randomly generate a uniform [0, 1] number. If this number is less than or equal to the simulated drop-off rate, the fish is considered dead and removed from both the cohort and region population in the region. Record mortality loss. Then go to Step 2. Otherwise, proceed as below.
8. Determine whether the fish is above legal size:
    a. Determine the size of the encountered fish by generating a positive random variable from a normal distribution with mean and SD (specified by input data) for the stock, age, and period.
    b. Compare this number with the size limits for the sector. If it is within the limits, proceed to the next step. Otherwise, simulate release mortality.
        i. Generate a uniform [0,1] random number
        ii. Generate a Gaussian random number in [0, 1] for release mortality based on mean and SD of sublegal mortality rate. This is generated only once each year for each simulation run.
        iii. If the uniform RN is less than the Gaussian RN, reduce the regional cohort size and population by 1. Record the mortality. Otherwise proceed to the next step.
9. If the sector is operating as a mark-selective fishery, go to the MSF procedure (see the Section on MSF below). Otherwise, proceed to the next step.
10. Add the fish to retained catch, remove it from the regional population and cohort in the region, and decrease TAC by 1. Go to Step 2 until regional catches are complete.
11. Do the same for all regions.

3.6 Mark Selective Fishery

If the selected fishery is Mark Selective, follow the steps below

1. Calculate the TAC for mark selection, which is defined as the abundance of unmarked fish available to fishery times a parameter called the Proportion of unmarked fish removed. MSF continues while the unmarked mortalities are less than the Mark Selective TAC or the fishery TAC, whichever is less. The fishery can be pre-terminal or terminal.
2. If the fish is M (Marked) or M+ (Marked with CWT), then apply MRE (Marked Retention Error) as follows:
    a. Generate a normal random number (constrained to [0, 1]) with mean MRE and standard deviation MRE_SD (specified by input data) .
    b. Generate a uniform random number [0, 1]. If this is larger than the simulated MRE, the fish is identified as correctly marked and eligible to be retained. Therefore, update the catch and population data as in Step 10 of Pre-Terminal Fishing (increment catch and decrement population). Then check whether the fishery is straight MSF
        i. If it is straight MSF, return.
        ii. Otherwise, it is the case of MIXED BAG. Add the marked fish to bag and reset if the bag limit reached. Return.
    c. If the uniform random number in Step b is less than the simulated MRE, then the fish is incorrectly identified as being unmarked and is released. Simulate release mortality:
        i. Generate a uniform [0,1] random number
        ii. Generate a Gaussian random number in [0, 1] for release mortality based on mean and SD of release mortality rate. This is generated only once each year for each simulation run.
        iii. If the uniform RN is less than the Gaussian RN (dead fish), reduce the regional cohort size and (U, U+, M, M+) population by 1. Record the mortality.
3. If the fish is U (Unmarked) or U+ (Unmarked with CWT), then apply URE (Unmarked Recognition Error) as below:
    a. Generate a normal random number (constrained to [0, 1]) with mean URE and standard deviation URE_SD (specified by input data).
    b. Generate a uniform [0, 1] random number. If this number is less than the simulated URE, the unmarked fish is erroneously kept. Update the sector catch and reduce the (U, U+) population by 1.
        i. If the fishery is straight MSF, return.
        ii. Otherwise, add the fish to bag as if it is marked, and reset bag if the bag limit is reached.
    c. If the uniform random number in Step b is larger than the simulated URE, then the fish is recognized as an unmarked fish.
        i. If the fishery is straight MSF, then the fish is released (retention of unmarked fish is not allowed). Simulate release mortality as described above.
        ii. Otherwise, it is the case of mixed bag and we may keep the unmarked

fish and add it to bag (the same bagging algorithm as above). However, if the maximum number of unmarked fish allowed in bag is reached, release this one and simulate mortality as above.

3.7 Stock Maturation

Fish surviving pre-terminal fishing mature to become terminal population at a rate which is simulated as a Gaussian random number in [0 1]. The mean and standard deviation are specified in the input files. Each stock may mature in a different period of a fishing year. Those fish that survive pre-terminal fishing and do not mature will become the population for the next period, adjusted for the survival rate. This survival rate is group-based (depending on which group the stock belongs to), and provided in the input files.

3.8 Terminal TAC

The TAC for a terminal fishery is sector based: for each sector (Sport or Net), calculate TAC by multiplying terminal abundance of the stock at specific ages by terminal harvest rate and harvest scalar, and summing the result over ages.

3.9 Terminal Fishing

Terminal fishing is generated on a per stock basis (*each terminal fishery contains only one stock*), in the following steps:

1. For a stock, select a fish from sectors Sport or Net, based on terminal TAC.
2. Determine the age of the selected fish:
    a. Calculate a cumulative index for terminal stock abundance by age Generate a uniform number [1, total terminal abundance of the stock].
    b. Randomly generate an integer and compare with the cumulative abundance index. If it is less than the cumulative abundance at an age index, select the age.
3. Simulate the mark type (U, U+, M, M+) of the selected fish (similar to the corresponding step in Pre-terminal fishing).
4. Simulate terminal drop off mortality (see the corresponding step in Pre-terminal fishing). If the fish is considered dead, update terminal population, and then go to Step 1.
5. If the fish is not killed as drop off, determine if the fish is of legal size (see corresponding step in Pre-terminal fishing).
6. If it is legal size, proceed to the next step. Otherwise, simulate release mortality (see the corresponding step in Pre-terminal fishing). If release mortality is true, then update terminal population and record the mortality. Go to Step 1.
7. If the sector is operating as a mark-selective fishery, go to the MSF procedure (see above). Otherwise, proceed to the next step.
8. Add the fish to retained catch, remove it from the regional population and cohort in the region, and decrease TAC by 1. Go to Step 1 until the terminal TAC is depleted for this stock.
9. Do the same for all stocks.

3.10 Escapement

For each individual stock, escapement is the terminal abundance remaining after fishing, adjusted for the pre-spawn mortality of the stock in the year as specified by user input data.

3.11 Recruitment

Recruitment is based on escapement from previous years (one year or two, depending on whether the stock type is ocean or river). If the stock is a river type, then recruit from escapement of two years ago. If the stock is ocean type, then recruit from escapement one year ago.

Production of a cohort from natural spawning escapement is determined by the Ricker formula with coefficients specified in the input files for each stock. Given an escapement, the Ricker formula yields an AEQ production:

$$\text{AdultReturn} = ESC \bullet \text{Exp}\left(A \bullet (1 - \frac{ESC}{B}) \right) \quad \textbf{(5)}$$

where escapement (ESC) is capped at Richer coefficient *B*.

Since production is expressed in terms of adult equivalents, it has to be converted to the Age 1 population based on survival rate and maturation rate. One algorithm is provided as below:

```
For each stock,
At Age 1, set spawnEQ = survivalRate(Age 1)
    Age1toAdultSurvivalRate = 0.0
For age = 2 to NumAges
     spawnerEQ = spawnerEQ * survivalRate(age)
     Age1toAdultSurvivalRate(stock) = Age1toAdultSurvivalRate
     +spawnerEQ*maturationRate(age)
     spawnerEQ = spawnerEQ * (1 – maturationRate(age))
Next age
Age1FishPopulation = TotalAdultReturnFromRicker / Age1toAdultSurvivalRate
```

For hatchery stocks, the Age 1 population is simply given by the escapement multiplied by the exponent of Richer A, where Richer A is the value from the natural stock with which the hatchery stock is associated.

3.12 CWT Recovery

The simulated data (pre-terminal catch, terminal catch, and escapement) output from the DGM module are sampled, without replacement, to simulate the CWT recovery under user-specified sampling rates. Sampling requires a set of sampling rates specified for each period (only one period for terminal fisheries), year, and fishery (escapement is treated as a fishery). See Appendix A for the sampling rate file format. The sampling procedure is as follows:

- A cumulative catch index of the total population to be sampled for CWTs is created from the simulated DGM catch data for all ages, mark types, and stocks.

- Then a uniform random integer in the range of [1, total population to be sampled] is generated. If the generated random integer is less than the cumulative value corresponding to a certain array index, then a fish is considered to be sampled from the stock-age-mark type.
- Once a fish is selected for sampling, it is removed from the index. Sampling is repeated until a desired number of sampled fish (determined by the overall sampling rate specified for the fishery) is reached.

The DGM data are handled separately as follows:

- Preterminal Catch: Sampling is performed on fish from all types (U, U+, M, M+) and ages of all stocks in each pre-terminal fishery, given a sampling rate in each specified period in the fishery.
- Terminal Catch: Sampling is performed on fish from all types (U, U+, M, M+) and ages for the stock harvested by a terminal fishery, at a rate specified for a given year (all terminal fisheries for a given stock are assumed to operate in only a single time period).
- Escapement: Sampling is performed on escapement for each stock from all ages and from all types (U, U+, M, M) at a rate specified for each year.

Using these data sets, the effects of visual or electronic sampling can be evaluated by the user at the time of analysis. Since the same DGM sample data sets are available, the effect of visual or electronic detection can be examined. If visual sampling is specified, only M+ fish will be employed in the analysis. If electronic sampling is specified, both U+ and M+ fish will be employed in the analysis. Simulation of multiple sampling is intended to provide datasets to evaluate variability in CWT recoveries.

4 DATA MANAGEMENT

The management of input/output data consists of two components: text file management and database management. Specifically, the input/output data management contains the following components:

1. **DGM Input Control file**: This is a file which lists input file names for the DGM.
2. **Input Files:** These are a set of transparent text files defining stocks and fisheries. See Table ?? for the complete list of required input files.
3. **Output Files:** These are also text files generated during a simulation and stored in a separate folder identified by a simulation ID.
4. **Database:** The database is a container for the latest DGM input data provided or modified by the user. It was developed on Microsoft Access, and is supported by a visual user interface for data entry and display.

4.1 Text File Management

When the user starts a simulation run, one option is to provide the program with a control file, which points to a number of file names that contain necessary input data for the model to run. The program reads the control file first, and then reads corresponding input files according to the control file. Input files can also be loaded into the database by choosing a control file. The program also maintains a record of these input files for each simulation, in a folder identified by a simulation ID, where it also stores all output files from the simulation. The project file structure is shown in Fig. 5. For example, the folder, **DGM001**, is the project folder, and **DGMInput** is the input data folder for the project. The DGM output data are stored in **DGMOutput_SM,** under the same project folder.

When the user starts an SM run, the system will retrieve the project folder path from the database, and loads the DGM output files (those file names with the DGM run ID, **DGM001**, as their prefix) in the project folder *as the input data of SM*. So the DGM output folder is also the SM run folder, and the sampling rate data file is copied to this folder, with the SM Run ID as the prefix of the file name. The SM output data will also be stored in a subfolder **SMOutput**.

CTCStock.csv	Definition of stocks
CohortSize_Age.csv	Cohort size at different ages
CohortRegionDist_MultiPool.csv	Cohort size distribution over regions:multi pool
CohortRegionDist_SinglePool.csv	Cohort size distribution over regions:single pool
GroupSurvival.csv	Survival rates of stocks as a group
Stock_Age_Size.csv	Stock size (length) at different ages
Stock_MatRate.csv	Stock year specific maturation rates
Stock_PrespawnMort.csv	Stock prespawn mortality
CTCFisheries.csv	Definition of fisheries
Fishery_Dist_Period.csv	Fish allocation over time periods
Fishery_HrScalar.csv	Fishery specific harvest rate scale
Fishery_MarkedRetention.csv	Fishery specific marked retention data
Fishery_Mortality.csv	Fishery specif mortality rates
Fishery_SizeLimit.csv	Fishery specific size limit
Region_Sector_Alloc.csv	Regional sector allocation
BasePeriod_ER.csv	Base period exploitation rates
Stock_TermHR.csv	Stock terminal harvest rates
Stock_TermHRScalar.csv	Stock terminal harvest rate scalars
Stock_TermMortRetention.csv	Stock terminal mortalities and retention data

Table 1: List of required input file

DGM generates three output data files that are used as input files for SM:

StockPTCatchSample.csv:	Preterminal catch data for SM (output)
StockTermCatch.csv:	Terminal catch data for SM (output)
StockEscapement.csv:	Escapement data for SM (output)

There are other output files for diagnosis purpose. When SM starts, it will look for these files in a target project folder. Correspondingly, SM will generate three output files:

SampledPTCatch.csv:	CWT recoveries for preterminal catch
SampledTermCatch.csv;	CWT recoveries for terminal catch
SampledEscapement.csv: :	CWT recoveries for escapements

4.2 Database

While the above input management approach works well if input data are already organized correctly in the files, it would not be convenient for a user to modify data inside the files, especially with addition or deletion of stocks and fisheries. Therefore, we have organized input data in a relational database based on MS Access, which contains all data in input files specified in the control file used in the previous program, and can be expanded as needed. A schematic view of the database tables is shown as below:

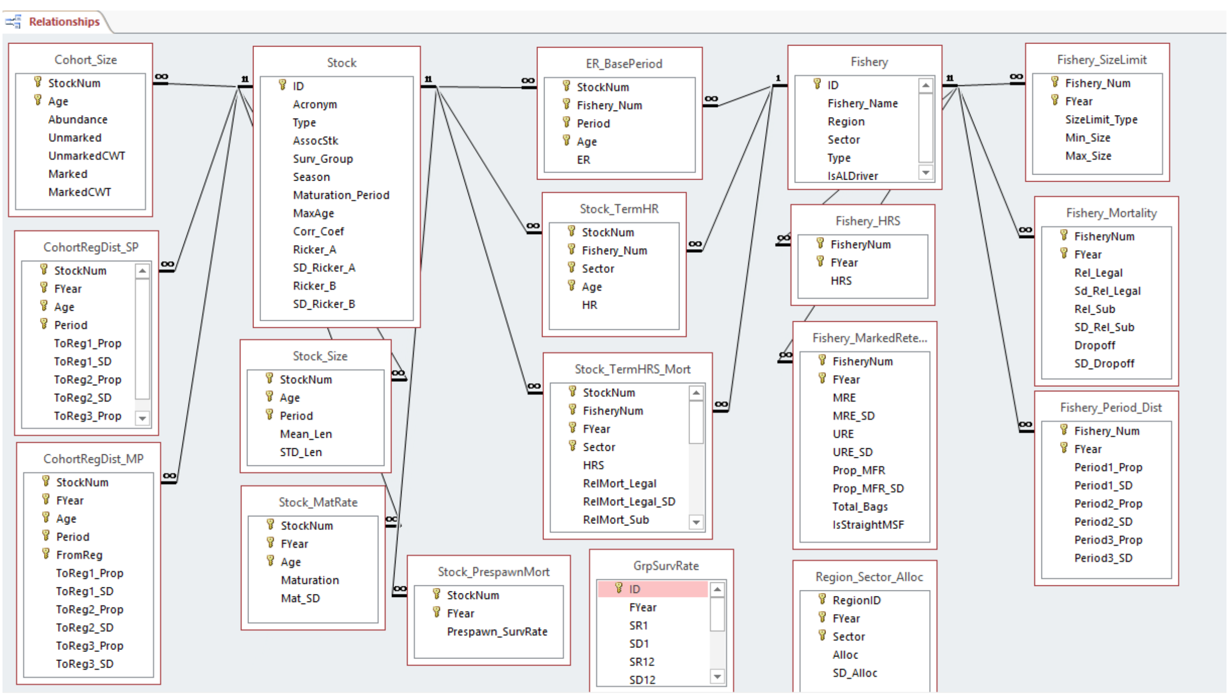


Figure 9: Schematic view of database

The database can be populated with input files, as well as via an interface described below.

Once input files are loaded into the database, the input data can be further modified prior to running a simulation. Note that the database keeps the latest copy of input data provided by the user. When the user starts a simulation run using the input data in the database, the system will copy the input data from the database, to a project folder identified by the simulation ID, as described above in the text file management section.

4.3 Visual Database Interface

The main interface for database entry and update is shown in Fig. 6 in Section 2, where it can be seen that there are two data entry cases: direct input from a control file (which points to individual data files), and manual entry by the user. As can be seen, the left side is for direct update and the right side for manual entry.

In the first case, the user loads a control file, and then clicks ‘Update DB’. The update will automatically start and progress messages will be shown in the box below. The user will be able to edit the control file to decide which file contains data that need to be added (or updated) to the database. This function allows the user to avoid having to enter data manually. This is particularly useful for loading a large amount of data such as single/multi pool distribution coefficients, which are too labor intensive for manual entry.

In the second case, the user can edit data via the interface. Specifically, the user can add, delete or modify a stock (or fishery) and its related data, without having to modify the data files listed in the control file. For example, if the user clicks ‘Add, Modify, Delete Stocks’, the following window shows up (Fig. 10). If another Stock ID is selected, its data will be shown accordingly. Click ‘Add a new stock’ or ‘Delete a stock’ for addition or deletion of stocks.

Similarly, click on ‘Add, Modify, Delete Fisheries’, and the following interface (Fig. 10) will show up (Fig. 11). See the user guide for detailed operations (Vitech, 2018).

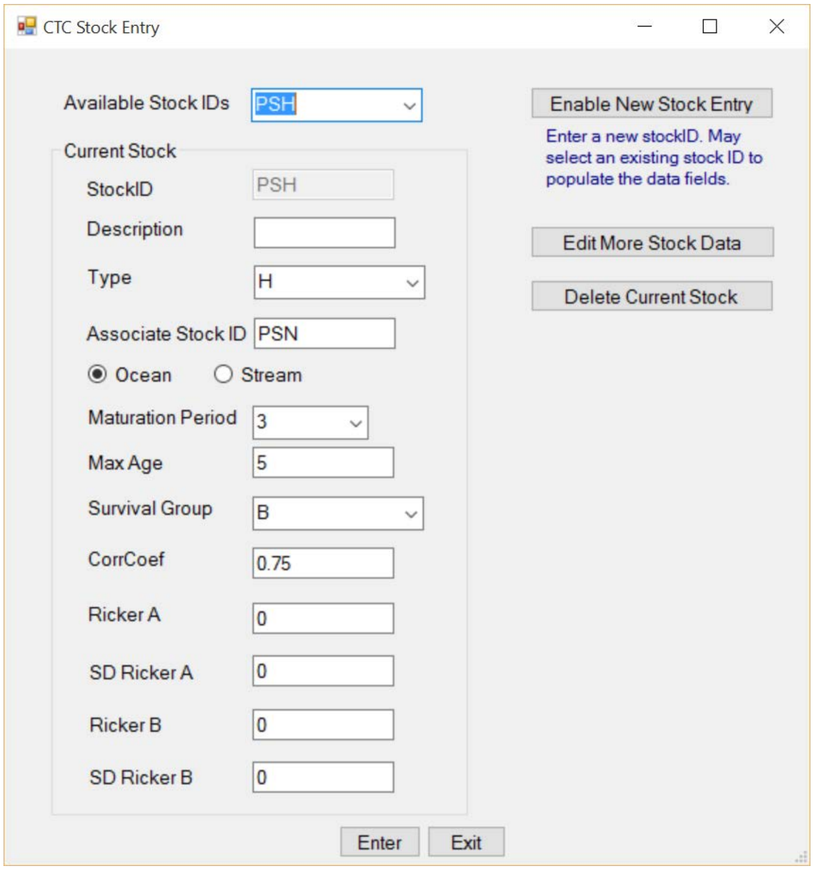


Figure 10: Stock Entry Interface

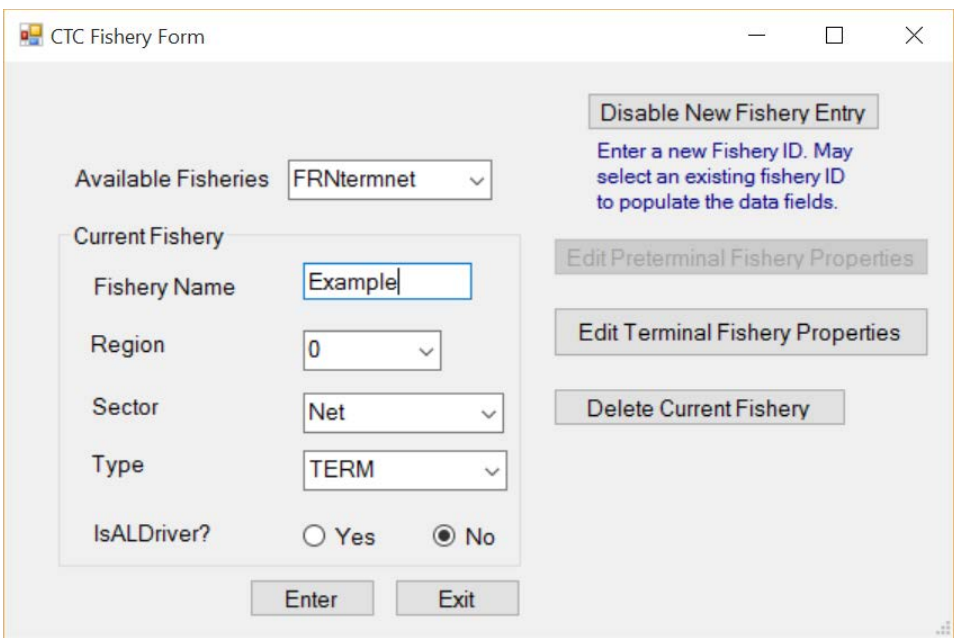


Figure 11: Fishery Entry Interface

5 ACKNOWLEDGEMENTS

Marianna Alexandersdottir of NWIFC and Gary Morishima of Mori-ko LLC made significant contributions and editing to this report. Brooke Davis of the Department of Fisheries and Oceans reviewed the VB code and made some corrections. These contributions are gratefully acknowledged.

6 APPENDIX

A. Input Data Files and Column Fields

The following input data files are used to allow users to specify the structure of the DGM and simulation/CWT sampling. These input data files are similar to tables in a database to allow convenient creation of. Note that all the stock IDs and Fishery IDs are uniquely defined in the Fisheries and Stocks files. Below is a list of the input files with column field names.

1. CTCStock.csv: defines basic properties for stock.
   Stock ID, Description, Type, Assoc Stock, Ocean/Stream, Maturation Period, Survival Group ID, Corr Coeff, Ricker A, SD Ricker A, Ricker B, SD Ricker B

2. CohortSize_Age.csv: provides cohort size of a stock at different ages.
   Stock ID, Age, Abundance, Unmarked(U), Unmarked+CWT (U+), Marked(M), Marked+CWT(M+)

3. Stock_Age_Size.csv: provides fish length (mean and SD) of a stock at different ages and in different time periods.
   Stock ID, Age, Period, Mean, SD

4. Stock_MatRate.csv: provides year specific maturation rates of stocks
   Stock ID, Year, Age, MatRate, SD

5. Stock_PrespawnMort.csv: provides year-specific pre-spawn mortality of stocks
   Stock ID, Year, Mortality

6. Stock_TermMortRetention.csv: provides a number of year-specific data in terminals, including mortalities, and retention data.
   StockID,Year,FisheryID,Sector,RelMort_legal,SD_RelMort_legal,RelMort_subl,SD_RelMort_subl,Dropoff_Mort,SD_Dropoff_Mort,MRE,SD_MRE,URE,SE_URE,Prop_Unmarked_Fish_Removed,SD_PMR,Total bags, Max_Unmarked Fish Allowed, IsMSF
   (Note: in this table, if the field Max_Unmarked_Fish_Allowed is zero, and IsMSF is set, then the fishery is a straight MSF.)

7. Stock_TermHR.csv: provides age-specific terminal harvest rates and size limits of stocks
   StockID, FisheryID, Sector, HR_Age2, HR_Age3, HR_Age4, HR_Age5, Size_Lower, Size_Upper

8. Stock_TermHRScalar.csv: provides stock terminal harvest rate scalars
   StockID, Year, FisheryID, Sector, HRS

9. GoupSurvival.csv: provides group-based survival rates and SDs for a stock at different ages and in different years.
   GroupID, Year, SR_Age1, SD1, SR_Age2, SD2, SR_Age3, SD3, SR_Age4, SD4, SR_Age5, SD5

10. CohortRegionDist_SinglePool.csv: provides regional distributions of a stock at different ages from a single pool.
    Stock ID, Year, Age, Period, P_r1, SD, P_r2, SD, P_r3, SD

11. CohortRegionDist_MultiPool.csv: provides regional distributions of a stock at different ages from multi-pools.
    Stock ID, Year, Age, FromReg, Period, P_r1, SD, P_r2, SD, P_r3, SD

12. CTCFisheries.csv: defines basic properties for fishery.
    Fishery ID, Name, Region, Sector, Type, IsAIDriver

13. Fishery_SizeLimit.csv: provides fish size limit at a fishery and in different years.
    Fishery ID, Type, Year, Min, Max

14. Fishery_HrScale.csv: provides fishery specific harvest rate scales.
    Fishery ID, Year, hrScalar

15. Fishery_Mortality.csv: provides various mortality rates at different ages at a fishery in a particular year.
    Fishery ID, Year, Rel_MortRate_Legal, SD_Rel_MortRate_Legal, Rel_MortRate_Sub, SD_Rel_MortRate_sub, Dropoff_MortRate, SD_Dropoff_MortRate

16. Fishery_MarkedRetention.csv, Fishery specific marked retention data
    Fishery ID, Name, Year, MRE, SD_MRE, URE, SD_URE, Prop Unmarked fish removed, SD_PMR, Total Bags, Max_Unmarked Fish Allowed, IsMSF
    (Note: in this table, if the field Max_Unmarked_Fish_Allowed is zero, and IsMSF is set, then the fishery is a straight MSF.)

17. Fishery_Dist_Period.csv: provides fish allocation at a fishery across different periods in a specific year
    Fishery ID, Year, Prop Period 1, SD Period 1, Prop Period 2, SD Period 2, Prop Period3, SD Period 3

18. Region_Sector_Alloc.csv: this is a table for regional sector allocation
    Attributes:
    Region, Year, Sector 1, Alloc, SD_Alloc, Sector 2, Alloc, SD_Alloc, Sector 3, Alloc, SD_Alloc

19. BasePeriod_ER.csv: provides base period exploitation rates at different ages of a stock at a fishery and in a certain period
    Stock ID, Fishery ID, Period, ER_Age2, ER_Age3, ER_Age4, ER_Age5

20. CTCSimInputControl.ctrl: this is a control file that lists all input file names that will be loaded into the simulation model.

In addition, there are also a number of constants the user can provide via the user graphical interface:

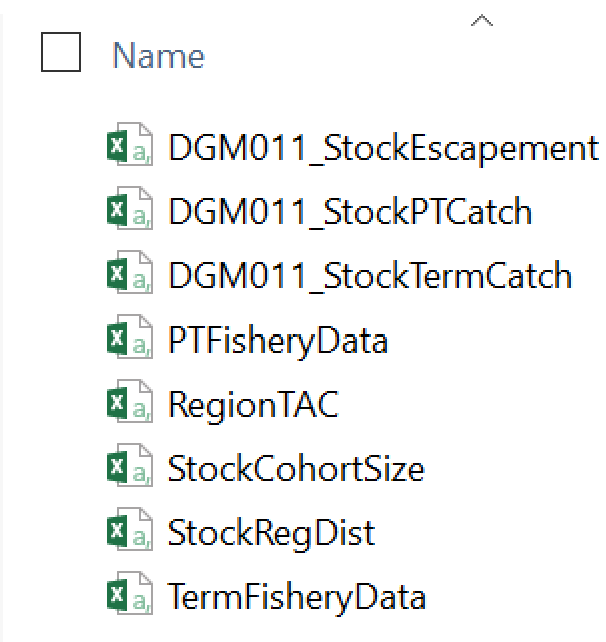
- Number of Regions
- Number of Periods in a year
- Number of Years
- Number of Ages

The input data are not provided for all years. So in a year where data are missing, we use those of the previous year.

For sampling, it requires a sampling rate data file, which has the following fields:
Fishery ID, Year, Period, SamplingRate

B. Output Data File Formats

DGM generates the following data files:



Catch and escapement data are partitioned into mark types of U (unmarked), U+ (unmarked + CWT), M (marked), M+ (mark + CWT). Specifically, they are in the following formats:

xxx_StockPTCatch: This file contains preterminal catch data, which is counts of fish caught in each region during a given time period, in a preterminal fishery. It also includes dropoff mortality count and release mortality count. It has the following fields:
RunID, StockID, Year, Period, Fishery, Age, C_U, C_U+, C_M, C_M+, DM_U, DM_U+,

DM_M, DM_M+, RM_U, RM_U+, RM_M, RM_M+

xxx_StockTermCatch: This is for terminal catch data, which is counts of fish caught during a given time period, in a terminal fishery. It has the following fields:
RunID, StockID, Year, Age, C_U, C_U+, C_M, C_M+, DM_U, DM_U+, DM_M, DM_M+, RM_U, RM_U+, RM_M, RM_M+

xxx_Escapement: This contains escapement data, which is counts of fish of a stock that escape during a given period.
RunID, StockID, Year, Age, U, U+, M, M+

The above three files are used in the sampling module. The remaining five files are for analysis and validation.

PTFisheryData: preterminal fishery data, which monitors TAC (total allowed catch) in a fishery.
RunID, StockID, Year, Period, Fishery, AI, HRI, TAC

RegionTAC: this file tracks regional TAC in a region.
RunID, StockID, Year, Period, Region, TAC

TermFisheryData: terminal fishery data, which tracks a few parameters in a terminal fishery.
RunID, StockID, Year, Period, Fishery, MSF, TermRun, HRS, TAC_Sport, TAC_Net

StockCohortSize: this file tracks stock cohort size over years.
RunID, StockID, Year, Age, U, U+, M, M+

StockRegDist: this file tracks the regional distribution of stock population over years.
RunID, StockID, Year, Region, Age, U, U+, M, M+

The sampling module generates output files as shown here:

where

xxx_SampledEscapement.csv
SMRunID, DGMRunID, Year, StockID, Age, SampledU, TotalU, SR_U, SampledU+, TotalU+, SR_U+, SampledM, TotalM, SR_M, SampledM+, TotalM+, SR_M+

xxx_SampledPTCatch.csv
SMRunID, DGMRunID, Year,Period, FisheryID, StockID, Age, SampledU, TotalU, SR_U, SampledU+, TotalU+, SR_U+, SampledM, TotalM, SR_M, SampledM+, TotalM+, SR_M+

xxx_SampledTermCatch.csv

SMRunID, DGMRunID, Year, FisheryID, StockID, Age, SampledU, TotalU, SR_U, SampledU+, TotalU+, SR_U+, SampledM, TotalM, SR_M, SampledM+, TotalM+, SR_M+

C. TAC for AABM Fisheries

For AABM fisheries, the TAC is determined from a regional aggregation index (AI). The system implements two options:

C1. Model based

First, calculate the TAC for troll

$$\mathrm{TrollTAC} = \mathrm{Exp}\left(PC + \ln(HRI * AI)\right) \quad \textbf{(6)}$$

based on AI values in a region, where HRI and PC values can be looked up in the table below (PC is the proportionality constant used to adjust catch to account for stocks not in model. For the DGM, PC=1).

Region	AI	HRI	Calculate range	PC
1	0 to 1.005	0.371		1
1	1.005 to 1.2	0.371 to 0.445	HRI=0.3795AI - 0.0104	1
1	1.2 to 1.5	0.51		1
1	>1.5	0.5525		1
2	0 to 1.205	0.757		1
2	1.205 to1.5	0.757 to 0.85	HRI = 0.3153 AI + 0.3771	1
2	>1.5	0.85		1
3	0 to 0.5	0.21		1
3	0.5 to 1	0.245		1
3	> 1	0.28		1

Where

Region 1: Southeast Alaska All Gear
Region 2: North BC Troll & QC1 Sport
Region 3: WCVI Troll & Outside Sport

Then the total catch is calculated as:

Region 1: Allocate 25% of Troll TAC to Sport Troll TAC and add 17000 for Net TAC). Then the total is the sum of these three TACs

Total TAC = 17000 (Net TAC) + 1.25*Troll TAC

Region 2&3: There is no Net TAC.

Total TAC = 1.25*TrollTAC

C2. User specified

The system has an option to allow users to load a lookup table file formatted as below in the CSV format:

Year	Region	AI	HRI	Troll	Sport	Net	Total
1979	1	0	0	0	0	0	0
1979	1	0.01	0.371	884	221	17,000	18,105
1979	1	0.02	0.371	1,768	442	17,000	19,210
1979	1	0.03	0.371	2,652	663	17,000	20,315
1979	1	0.04	0.371	3,536	884	17,000	21,420

Note that the rows are ordered in increasing AI values. The system looks up in the table for values of HRI and TAC for Troll, sport, net, given an AI value. For AI values falling between cells in the AI column, linear interpolation is applied. If AI is greater than the largest value in the table, use the data corresponding to the largest AI.

In either option, the total AABM TAC is then allocated to fisheries (sectors) and periods using inputs from files Fishery_Dist_Period.csv and Region_Sector_Alloc.csv.

D. Random Number Generation

We first started with the implementation of a random generator using the Marsenne Twister (MT) algorithm (Matsumoto and Nishimura, 1998). Random number generation can be computationally intensive so efficiency is a major consideration. We chose an implementation of the MT algorithm for random numbers of uniform distributions provided by Agner Frog. The code generates very good randomness with long cycle length and sufficient performances. This library also includes non-uniform random number generators including normal, Poisson, Binomial, Hypergeometric, Fisher's noncentral hypergeometric and Wallenius' noncentral hypergeometric distributions, as well as multivariate versions of these distributions.

D1. Kumaraswamy Distribution

We have also supplemented the library with an implementation of the Kumaraswamy distribution (KD) (Kumaraswamy, 1980) using the Inversion Method. The probability density function of the KD distribution is defined by two parameters as

$$f(x;a,b) = abx^{a-1}(1-x^a)^{b-1} \qquad \textbf{(7)}$$

and the cumulative distribution function is therefore

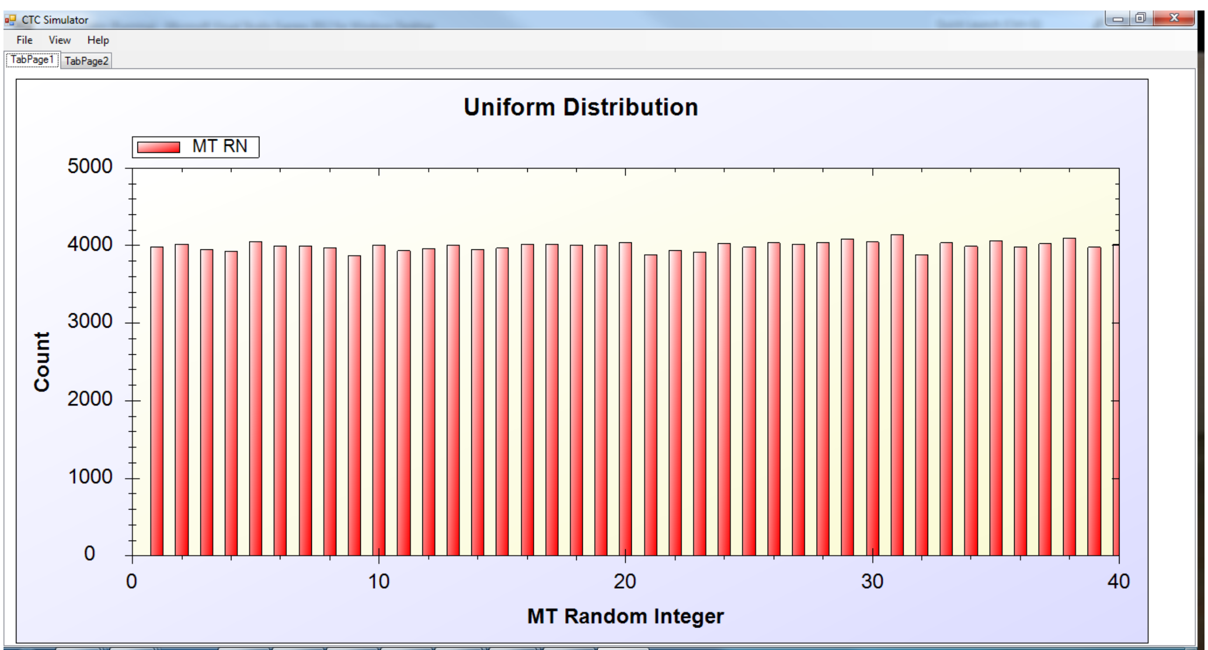
$$F(x;a,b) = \int_0^x f(t;a,b)dt = 1-(1-x^a)^b \tag{8}$$

If u is a random variable of uniform distribution within [0, 1], then a variable defined by

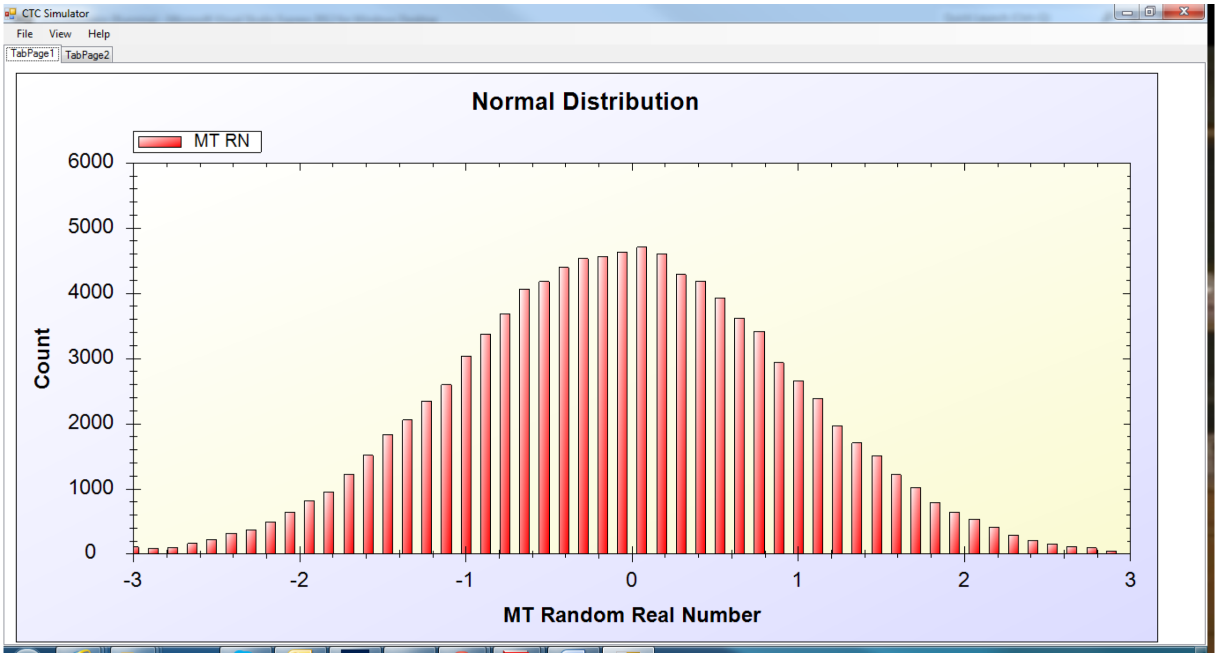
$$x = F^{-1}(u;a,b) = (1-(1-u)^{1/b})^{1/a} \tag{9}$$

has the KD distribution.

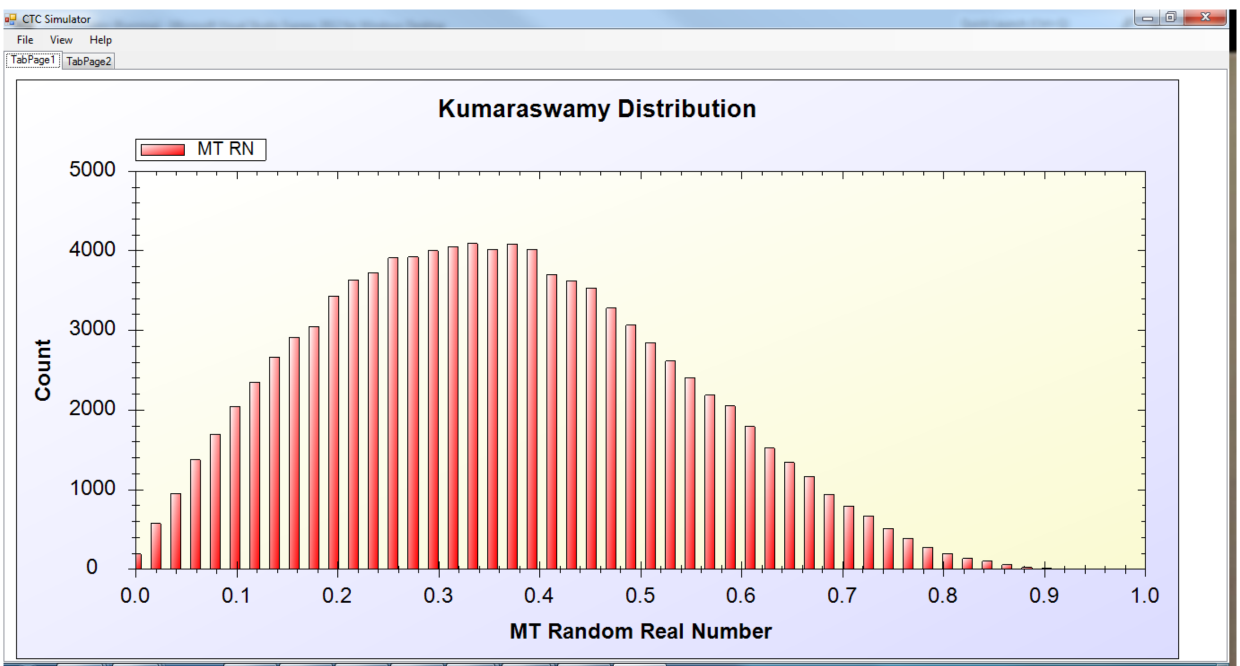
We have integrated the random number generation library into our system under the .net framework. Histogram plots of three cases from the random number generator are shown as below. All of them are based on the MT algorithm for generating uniform random integers. The first plot is for the uniform distribution of integers (only partially shown), and the second one is for the normal distribution with zero mean and standard deviation of 1. The third one is for the KD distribution with a = 2, b = 5. The results appear consistent with theoretical models of these distributions. In addition, this library is very fast and efficient: it can generate a set of 800,000 uniformly distributed integers with a few seconds on a typical laptop computer nowadays.



Uniform distribution of integers (shown only up to 40)



Normal distribution with zero mean and standard deviation of 1



KD distribution with $a = 2$, $b = 5$

We originally planned to use the Kumaraswamy distribution, but only the mean and standard deviation of the coefficients are given. While it is possible to transform mean and standard deviation to the KD parameters, it is not a trivial task so we left this issue as a future work item.

D2. Generation of Correlated Gaussian RN

Correlated Gaussian random numbers can be generated from independent random numbers with a transformation so that the result satisfies a specific covariance matrix. Let **x** be a vector of independent random numbers drawn from a multivariate normal distribution with mean = 0 and variance = 1. That is

$$\mathbf{x} \sim N(0, \mathrm{I})$$

where I is the identity matrix of size *n* x *n*, where n is the length of **x**. Now we define another vector **y** as

$$\mathbf{y} = \mathbf{L} \cdot \mathbf{x} + \mathbf{m}$$

such that

$$\mathbf{y} \sim N(\mathbf{m}, \mathrm{R})$$

where **L** is the transformation matrix and R is the covariance matrix

$$\mathbf{R} = E\{(\mathbf{y} - \mathbf{m})(\mathbf{y} - \mathbf{m})^T\} = \mathbf{L} \cdot \mathbf{L}^T$$

Note that the diagonal elements in **R** are the variances of elements in **y**, and the other elements in **R** are the covariance between the elements in **y**.

The transformation matrix can be found by performing Cholesky decomposition of the covariance matrix (Press et al, 1992), as long as the matrix is positive definite. Using the resulting transformation matrix, we can generate a vector of Gaussian random numbers with the specified mean vector **m**, and covariance matrix **R**. Note that in our application, **y** usually is usually bounded by [0, 1]. In this case, we can solve the linear transformation for **x** given the y bounds to find the bounds for **x**.

7 REFERENCES

Fog, Agner (2013): Pseudo random number generators: uniform and non-uniform distributions, www.agner.org/random

Kumaraswamy, P. (1980): A generalized probability density function for double-bounded random processes, Journal of Hydrology 46: 79–88

Matsumoto, M. and T. Nishimura (1998): Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Trans. Model. Comput. Simul. 8, 3.

Morishima, Gary (2009): REVISED Data Generation for CTC Model Improvements Project v 1.1.

Press, W.H., Teukolsy, S.A., Vetterling, W.T. and Flannery, B.P., 1992. Numerical recipes 2nd edition: The art of scientific computing. Cambridge University Press.

TCCHINOOK (2009): Special Report of Chinook Technical Committee HRI Workgroup on the Evaluation of Harvest Rate Indices for use in Monitoring Harvest Rate Changes in Chinook

AABM Fisheries. Report of Chinook Technical Committee to PSC (09)-2

Vitech Innovative Research and Consulting (2018): CTC Data Generation Model (DGM): Installation and Operating Procedures.