SCIENTIFIC COMMITTEE
EIGHTEENTH REGULAR SESSION

ELECTRONIC MEETING
10-18 August 2022

DEVELOPMENTS IN THE MULTIFAN-CL SOFTWARE 2021-22

WCPFC-SC18-2022/SA-IP-03
22 July 2022

Nick Davies[3], David Fournier[2], Fabrice Bouyé[1], and John Hampton[1]

Oceanic Fisheries Programme
The Pacific Community, Noumea, New Caledonia

[1] Oceanic Fisheries Programme, The Pacific Community, Noumea, New Caledonia

[2] Otter Research Ltd, North Saanich, British Columbia, Canada

[3] Te Takina Ltd, Whangarei, New Zealand

Table of Contents

EXECUTIVE SUMMARY

This paper summarises the developments made within the MULTIFAN-CL software project as carried out by the team at the Oceanic Fisheries Programme (OFP, The Pacific Community, Noumea, New Caledonia) from August 2021 to July 2022, and updates the report of Davies et al. (2021).

The progress made in implementing new features related largely to improving its capability for stock assessments and to SC's discussions on reducing model complexity. Development of the catch-conditioned method for estimating fishing mortality has been in progress since early April 2020 (Davies et al. 2020), and has been the main focus since then. Highlights during 2021-22 that relate to this development include:

- ensuring robust minimisations and avoiding population crashes during the Newton-Raphson solution for fishing mortality estimation,
- improving the regression for the fishing mortality:observed effort relationship (fml_effort_rltnshp), and
- extending its operation into the post-minimisation procedures, including undertaking zero-fishing mortality model evaluations for fishing impact analyses, and deterministic projections.

The method appears to be stable, and with the most important components needed for production stock assessments that employ it operating satisfactorily.

A new feature was completed that implements the CPUE likelihood, being a separate and discrete term for fitting relative abundance indices. This was a high priority for allowing the comprehensive implementation of the catch-conditioned method plus survey fisheries CPUE indices into a production stock assessment; as implemented within the 2022 WCPO skipjack stock assessment.

The new feature added in 2021-22 for long-term tag loss has direct utility for using MULTIFAN-CL in a spatial modelling simulation analysis (currently in progress in a separate study), where the operating model assumes a constant rate. Completing this feature has addressed one of the last two remaining recommendations from the independent review of MULTIFAN-CL by Ianelli et al. (2012).

A significant enhancement made was to add new dependent variables to the variance and correlation reports, including the WCPFC stock depletion reference points, thereby addressing a feature identified during the 2021 SW Pacific swordfish assessment. This has direct relevance for the skipjack 2022 assessment, but also in general for examining parameter uncertainty and the correlations of depletion variables with respect to other dependent variables. A wide range of other enhancements and corrections were also made, with those relating to the initial population conditions having high priority for the 2022 skipjack assessment.

As for 2020-21, in 2021-22 no on-site training workshops at Nouméa were held for stock assessment analysts due to travel restrictions associated with the COVID-19 pandemic. However, considerably more resources were directed towards training and user support in 2021-22 compared to previous years; with a total of 21 on-line sessions having been provided to OFP scientists. These included an in-progress review and support for the catch-conditioned model development for the skipjack 2022 stock assessment.

A significant milestone in the history of the project was the retirement of the lead developer and original architect of MULTIFAN-CL, Dr David Fournier, in December 2021. The loss of his creative force in developing innovative applied mathematical and statistical methods is reflected in the proposed workplan for 2022-23 that excludes large-scale new developments, but rather seeks to consolidate those aspects of recent new features that are incomplete, undertake enhancements of existing features, and complete documentation. Key elements of the plan in order of priority include:

- Consolidate the catch-conditioned method for:
    - the implementation of stochastic recruitment in projections,
    - fishery-specific penalties on the fml_effort_rltnshp,
    - update the likelihood components report, and
    - testing of the control phases and for applying the method to multi-species.

- CPUE log-normal likelihood
    - Review the concentrated formulation,
    - Allow fishery-specific penalties for grouped survey fisheries.
- Update the User's Guide for the catch-conditioned method, and other recently added features not yet documented.
- Tidy the code, complete a backlog of bug fixes, and any "small-scale" requests in the tasks list.
- Provide support for MSE requirements and improvements.
- Training and stock assessment support for OFP.

Some of the proposed items from the 2021-22 plan will be retained in the MULTIFAN-CL development list, and carried over into the 2022-23 workplan.

1 INTRODUCTION

MULTIFAN-CL is a statistical, age-structured, length-based model routinely used for stock assessments of tuna and other pelagic species. The model was originally developed by Dr David Fournier (Otter Research Ltd) and Dr John Hampton (The Pacific Community) for its initial application to South Pacific albacore tuna (Fournier et al. 1998). It has since provided the basis for undertaking stock assessments in the Western and Central Pacific Ocean.

The MULTIFAN-CL model is described in detail in the User's Guide (Kleiber et al. 2018). It is typically fitted to total catch, catch rate, size-frequency and tagging data stratified by fishery, region and time period. For example, recent tuna and billfish assessments (e.g., Castillo-Jordan et al. 2022) encompass long time periods, e.g., 1952 to 2019 in quarterly time steps, and model multiple separate fisheries occurring in up to 9 spatial regions. The main parameters estimated by the model include: initial numbers-at-age in each region (usually constrained by an equilibrium age-structure assumption), the number in age class 1 for each quarter in each region (the recruitment), growth parameters, natural mortality-at-age (if estimated), movement, selectivity-at-age by fishery (constrained by smoothing penalties or splines), catch (unless using the catch-conditioned catch equation), effort deviations (random variations in the effort-fishing mortality relationship) for each fishery, initial catchability, and catchability deviations (cumulative changes in catchability with time) for each fishery (if estimated). Parameters are estimated by fitting to a composite (integrated) likelihood comprised of the fits to the various data types, and penalized likelihood distributions for various parameters.

Each year the MULTIFAN-CL development team works to improve the model to accommodate changes in our understanding of the fishery, to fix software errors, and to improve model features and usability. This document records changes made since August 2021 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.8.0), and the current unreleased development version, and updates the report for the previous period, 2020-21 (Davies et al. 2021).

2 DEVELOPMENT OVERVIEW

2.1 Team

The senior developer of MULTIFAN-CL during part of 2021-22 was Dr David Fournier, of Otter Research Ltd, (Canada), and he has retired since December 2021. Assisting with programming is Nick Davies. Other tasks include testing and debugging (ND, John Hampton, and Fabrice Bouyé (SPC)); documentation (ND); and planning and coordination (DF, ND, and JH). Related project software are developed or managed by FB (MULTIFAN-CL Viewer, Condor, GitHub, Jenkins), ND, and Robert Scott (R4MFCL, FLR4MFCL).

2.2 Calendar

August – November: Testing, planning and ongoing code development, Developer's workshop

December – February: Testing and ongoing code development

March-April: Training and Developer's workshops

May-July: Testing, ongoing code development and support for stock assessments

2.3 Collaboration and versioning

The repository and overall development are coordinated via the GitHub website on GitHub.com at https://github.com/PacificCommunity/ofp-sam-mfcl which is administered by Fabrice Bouye (fabriceb@spc.int) (section 2.4.7).

Problems with MULTIFAN-CL operation or compilation have been reported to the project management website so as to maintain a list of desired enhancements, and to allocate tasks among the project team. A "master" branch exists for the MULTIFAN-CL source code from which release versions are posted, and development branches ("ongoing-dev", "mac-dev") have been created for holding development versions of the source undergoing development and testing. A formal testing procedure has been designed before source code is merged from the branch to the trunk, and a manual for the testing of new compilations, standardizing the source code compilation procedure, and posting of executables is maintained.

2.4 Compilation framework and Source code repository

2.4.1 Compilation framework

A continuous integration facility allows for automatic nightly compilations of the MULTIFAN-CL source on the GitHub repository "master" branch. This automation is done using the software called Jenkins (https://jenkins-ci.org/), an Open Source continuous integration tool that comes bundled with a web server used for administration. This software is now installed on a Linux Virtual Machine (VM) that is dedicated to MULTIFAN-CL development, and administers the compilations over the OFP network.

In this tool, we've added a custom scheduled task that automatically retrieves the MULTIFAN-CL source code out of the GitHub code repository (master branch); it also retrieves required libraries for the compilation. When done, our task compiles both debug and optimized versions of the software. We've also configured this task to produce code documentation out of the source code and to run some C++ code quality checking.

Doing a nightly compilation allows us to find out more quickly whether issues have been included in the source code repository without being solved by the developer. It also helps us identify issues in the makefile configurations that may prevent the compilation of MULTIFAN-CL on some more neutral environment (i.e.: on a machine that is different from the one of the developer's).

During 2018-19 this facility was extended to support automated builds of the Windows (Visual Studio 2019) and the macOS release executables. The Windows10 VM used for undertaking the benchmark testing framework (see section 2.6) and the Mac Mini provided the platforms for undertaking the routine compilation administered by Jenkins (see section 2.4.4). These automated builds were maintained throughout 2021-22.

It is also intended to add to the Jenkins tool the running of automated tests using example fish model data, and, in the future, unit tests for the software.

A directory structure on the dedicated VM was used that is mirrored on all the developer's platforms in respect of source code **Projects/**, associated libraries **libs/**, and **Testing/** directories. This ensures portability of source and makefiles among the developers and the automated build software.

2.4.2 Compilation of dependent libraries

For compilation of the dependent **OpenBLAS** library, the "dynamic architecture feature" is included to the routine compilations that builds several kernels for various processor types, and allows selection of the appropriate kernel at run-time. This may avoid the case where a MULTIFAN-CL executable that was compiled with OpenBLAS on a platform having a very recent processor, fails upon execution because function calls to the OpenBLAS library are attempted on platforms having relatively older processors. This compilation method results in a substantial increase (22 MB) in the executable size. However, it was noted that OpenBLAS libraries are important for the calculations used for the eigenvalues and eigenvectors of the Hessian and also aspects of the self-scaling size composition likelihood. This trade-off is therefore considered acceptable for the increased utility achieved.

In order for the MULTIFAN-CL project to be completely portable, three shell scripts automate the compilation of all the dependent libraries, before compiling MULTIFAN-CL. These scripts apply different options for OpenBLAS, QD and compilation flags for MULTIFAN-CL. The script “build_openblas4mfcl.sh” builds 3 options of this library: "default", "generic", and "dynamic", where the "dynamic architecture feature" builds several kernels for various processor types, and allows them to be selected at run-time. Similarly, the script “build_qd4mfcl.sh” builds 4 options of the QD library: "default", "O3", "O3fma", and "native". Given the various combinations of compilation options among the dependent libraries, ADMB and MULTIFAN-CL, compilations of 25 different executables may be produced. For a single option, it compiles in total: 49 minutes 5 seconds. This facilitates the portability of the entire MULTIFAN-CL compilation project including the dependent libraries, such that the complete project may be constructed and compiled with one step.

It is now possible to include in the automated compilation administered by Jenkins, compilation of the dependent libraries QD and OpenBLAS. With the exception of a couple of manual steps required to configure particular options, the integrated compilation of the entire project is now undertaken within the Jenkins routine compilation procedure.

2.4.3 Compilation of Linux executable

The Linux version currently used for compilations is Ubuntu 18.04, with the gcc compiler version 7.3.0. The version of glibc installed with ubuntu18.04 is glibc 2.27, and provides better (faster) Math functions than the earlier versions. It is intended to upgrade to Ubuntu 20.04 and gcc 9.3 or 9.4 during 2022-23.

2.4.4 Compilation of Mac OS executable

During 2018-19, routine macOS compilations of the “master” and “development” branches were added to the compilation framework of the MULTIFAN-CL project.

The version 2.0.8.6 compilation was done on the MULTIFAN-CL Testing PC (Mac Mini) that has the macOS Mojave installed (“macOS 10.14.6 Mojave”). The PC has two compilation directories:

- Local compilation - stand-alone directory for testing development versions
- Jenkins compilation - for routine automated Jenkins compilations of the master branch (checked out from the repository, see section 2.4.1)

A Software ID certificate was assigned to the macOS compilation using an Apple Developer ID certificate for SPC. The macOS version is signed “Developer ID Application: The Pacific Community” issued by Apple. No differences were detected in the computations or performance among the signed and un-signed compilations.

Apple has now officially announced that they are ceasing production with Intel’s CPUs in favour of their own RISC ARM-based CPUs. The macOS compilation of MULTIFAN-CL will ultimately need to accommodate this change. A draft strategy for changing to the ARM-mac compilation for MULTIFAN-CL may entail:

- Following the 2021 Apple conference decide on a machine purchase to be made during 2022
- Explore the potential for compiling on the existing Intel-mac with output target set to the new ARM-CPU using flags in the most up-to-date or the next version of XCode (the Apple dev tools/compiler)
- Maintain a careful watch of the capability of the Rosetta 2 emulator for running the MULTIFAN-CL executable (compiled for the Intel CPU) on an ARM-mac; this could offer a "breathing space" for our switch to the ARM-mac compilation
- Consider the merits of upgrading the mac Mini from Mojave to Catalina or Big Sur
- Potentially consider the lead developers purchasing an ARM-mac (external of SPC) for developing the compilation
- Aim for making the switch to the ARM-mac compilation for MULTIFAN-CL in mid-late 2022

2.4.5 Visual Studio 2019 Windows compilation

Few difficulties were encountered with the VS2019 compilations during 2021-22, and generally the MULTIFAN-CL code was compatible among the three compilation platforms. The VS2019 compiler treats differently the casting of constant variables passed as formal arguments, and this was readily addressed using the tool: ADUNCONST(dvar_len_fish_stock_history,fsh).

A Software ID certificate was assigned to the VS2019 compilation using a Windows Developer ID certificate for SPC. The Windows version is signed: "The Secretariat of the Pacific Community" issued by DigiCert. No differences were detected in the computations or performance among the signed and un-signed compilations.

2.4.6 Development version

Upon completing benchmark testing of a development version, the source code in the repository development branch is merged to the master branch and tagged with a release version number. At this point the development branch is created afresh for implementing any subsequent code developments. Another point where a new development version is made is immediately preceding each developer's workshop at which large scale code changes are rapidly made. These are then added to the development branch after the workshop and following preliminary testing. Points during 2021-22 where development versions were created included:

- Following the benchmark testing of version 2.0.8.5
- Preceding the ongoing developer's workshop

The main developments currently implemented in the development version since version 2.0.8.0 that have not yet been benchmark tested include:

- Enhancements and corrections to the catch-conditioned fishing mortality calculation
- Implementation for a CPUE likelihood term
- Additional dependent variables in the *.var
- Predicted tag recaptures – added constant for robustness
- Maturity-at-length conversion to age calculations - corrections
- Parameter scaling - revisions
- Optional penalty for movement coefficients
- Various corrections (section 5.10)

2.4.7 Source code repository

The MFCL project is hosted on GitHub.com at:

- https://github.com/PacificCommunity/ofp-sam-mfcl

This site is only accessible to registered members of the OFP-SAM team. In order to better coordinate developments within components of the project, separate repositories were created for the:

- User's Guide: https://github.com/PacificCommunity/ofp-sam-mfcl-manual
- ADMB dependent library: https://github.com/PacificCommunity/ofp-sam-admb

The branches of the repository are managed such that following benchmark testing, the development version that has tested positive and held in either of the "mac-dev" or "ongoing-dev" branches, is then merged to the "master" branch. This creates a clear node in the "master" branch tagged as being the next release version. At that point a new development version is created in one of the "mac-dev" or "ongoing-dev" branches for undertaking the next phase of developments. This approach was followed for each of the versions during 2021-22, with the current development version being maintained in the "ongoing-dev" branch (version 2.0.8.6).

Between 9 Aug. 2021 and 16 July 2022, a total of 41 source code commits were made to the master and development branches (67 source files have been modified, 5 source code files added, see section 12), including merges to the master branch in 31 Jan. 2022 preceding the release of version 2.0.8.4.

2.5 Developer's workshops

The format of on-going workshops among the primary developer Dr David Fournier and Mr. Nick Davies in an on-line format was continued during 2021-22. This was due to the COVID-19 pandemic (since early 2020) under which travel was not possible. Rather than staging two separate discrete workshops, the collaborative developments were performed as an ongoing continuation of the method used for the April-May 2020 workshop, i.e. using an "on-line" workshop. This entails a Linux **Nomachine** remote desktop protocol server on a mainframe, that enables both developers to share, with equal participation, a Linux session simultaneously, thus allowing collaborative real-time work on the same desktop. With a concurrent Skype session for verbal and visual communication, this facilitates a “virtual” workshop with considerable success. This facility was employed for on average 4 days per week for between 2-5 hours per workshop, until January 2022. As such, many of the new features and enhancements described in Sections 4 and 5 were achieved using this collaboration method. Since January 2022, Mr. Nick Davies has continued the consolidation of new features.

2.6 Benchmark testing during 2021-22

The benchmark testing framework is described in section 2.9.2, and two sets of benchmark tests were undertaken in 2021-22. A brief description of the tests, and the features tested, is provided in this section.

2.6.1 Version 2.0.8.2

In January 2022, and following the on-going on-line developer’s workshops to that time, a benchmark test was done of the development version (to become version 2.0.8.2) versus the release version 2.0.8.0. Substantial developments and bug fixes had been made since the previous testing in February 2021. Details of the new and enhanced features in the development version are described in section 4, with the highlights including:

- **Catch-conditioned method –** improved the regression options for the fishing mortality:observed effort relationship (fml_effort_rltnshp); added capability for survey fisheries (non-extraction) with a likelihood term for the associated CPUE indices; refined the initial survival rate calculations, and added the option for the initial population calculations to be based only upon natural mortality; consolidated the partial fisheries impact analysis for the catch-conditioned case such that all those fisheries with assumed zero-fishing mortalities are calculated via the Newton-Raphson with observed catches set to small values; to achieve minimisation stability, robustness was applied to avoid excessive fishing mortalities (Zmax_fish) by adding a penalty for instances of fishing mortalities exceeding a fraction of Zmax_fish; added the feature for zero-fishing compensation of BH-SRR predicted recruitments in projections.
- **Variance calculation of dependent variables** – increased calculation efficiency; added the dependent variable SB_latest/mean(SB_f=0)_nyr-9 to the *.var report.
- **PDH diagnostics** - additional reports for the positivised Hessian diagonals, covariance and correlation matrices of the independent variables.
- **Enhancements** – including: spline selectivity estimation, orthogonal polynomial recruitment parameterisation, estimation of movement coefficients, input checks on flags, percent maturity and age-length input data, pointers for fisheries grouping.
- **Corrections** – made to many aspects of the code, including the catch-conditioned feature.

Note: all the benchmark testing was done using the compilation of the development version for the “standard” 64-bit precision, as this ensures comparability with the benchmark version 2.0.8.0. A summary of the testing is:

Single evaluation tests for single-species and multi-species data, with or without gradient calculations and a minimisation step – tagging likelihood percentage differences in the order of 1.e-06 to 1.e-03. Attributed to tag recapture predictions being made robust to ensure minimisation stability (pred_tag_robust).

Stochastic projection final year absolute biomass – percentage differences of generally 5%, but as large as 27%. This appeared due to refinements made to the adjustment of fishing mortalities exceeding the Zmax_fish values (adj_Zmax).

Doitall fits of single species data – negligible differences (<1%) in key model quantities, viz. SBcurr/SBcurrF0, for three of the test data sets (SKJ2016, YFT2017, SWO2017) with minor differences (<5%) for the other sets. Due to pred_tag_robust, adj_Zmax, the revised scalar on the tag_fish_rep parameters (tag_rep_scalar), and the lower bound on totpop_coff changed from 1 to 8 to avoid population crashes in early phase iterations (totpop_bnd).

Doitall fits of multi species and multi-sex data – minor differences (<6%) for species 1, with moderate differences for species 2, in particular in respect of fishing impact dependent variables, e.g. 24% for

SBcurr/SBcurrF0; and minor differences (<3%) for either sex. Due to pred_tag_robust, tag_rep_scalar, totpop_bnd, adj_Zmax, and pmat_multspp.

While these code changes produce essentially identical function evaluations for the same set of independent variables, they result in slight differences in the gradient calculations between iterations, causing different minimisation paths being taken that affect the solution obtained by the development version compared to that of vsn.2.0.8.0. For reasonably well-determined examples, this results in only minor differences in the final converged solution, however differences become most apparent for the poorly-determined examples that appear to have multiple local minima.

Tests of the development version concluded that the results were consistent with respect to the benchmark version 2.0.8.0 as all existing features remain intact, therefore the development version may be advanced to the new MULTIFAN-CL release version, 2.0.8.2.

2.6.2 Version 2.0.8.5

During May 2022, comparative testing was done between the MULTIFAN-CL development version, and the benchmark release version 2.0.8.2 previously tested in February 2022. An abbreviated set of tests were undertaken, simply to examine the effect of a single significant correction made to the source code relating to the initial equilibrium population as obtained using the catch-conditioned method for fishing mortality estimation, and under the zero-fishing mortality model evaluation (see 5.10.1).

Single evaluation tests for single-species and multi-species data, without gradient calculations and no minimisation step taken – were identical.

Doitall fits of single species data – negligible differences (<1%) in key model quantities.

The tests concluded that the development version results were consistent with respect to the benchmark version 2.0.8.2 as all existing features remain intact, therefore the development version may be advanced to the new MULTIFAN-CL release version, 2.0.8.5.

2.7 Postings to website

There have been no postings of the MULTIFAN-CL release versions to the website since July 2020.

2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment

An outcome of an independent peer review of the 2011 bigeye tuna stock assessment (Ianelli et al. 2012) was a set of recommendations for improvements and developments to the MULTIFAN-CL software. These aim not only to improve the software's application in the context of the bigeye assessment specifically, but also its stock assessment application more generally. These recommendations have been the basis of MULTIFAN-CL developments since the review, and an outline of the status in fulfilling these recommendations is provided.

At the beginning of 2021-22, of the thirteen recommendations, 11 had been implemented and tested, and 2 remained yet to be developed:

- Non-uniform size bins (recommendation "b")
- Long-term and initial tag loss (recommendation "c")

Progress was made on one of the recommendations ("c") during 2021-22, and which has been completed. A description is provided in section 4.2.

2.9 Tool development

2.9.1 R4MFCL

The R scripts for working with MULTIFAN-CL, developed by OFP are maintained on a GitHub repository and have been partially updated to adapt to the recent MULTIFAN-CL release version file formats. These scripts are used to manipulate the input files, so that submitting model runs can be automated from R. Other scripts can be used to read in the output files, analyze the results, and generate plots and tables. Only limited and minor improvements were made to this aspect of the project during 2021-22.

2.9.2 Testing framework

The testing framework for MULTIFAN-CL compilations first developed in 2011-12, was applied during 2021-22 for the benchmark testing of versions 2.0.8.2 and 2.0.8.5 (section 2.6). This framework ensures the repeatability and traceability of testing by streamlining the process for new source code developments through a system of model testing procedures and directories. The testing criterion is based upon pair-wise comparisons of model run results obtained using an existing MULTIFAN-CL compilation (usually the current release version) versus those from a development version compilation. Tests are undertaken over multiple processor platforms (64-bit architecture only), with application to multiple input testing data sets, and with various options for the MULTIFAN-CL operation, viz. single or multiple model evaluations, or full doitall model fits to convergence. This ensures a thorough integrity-check of model quantities and components of the objective function prior to the distribution of new versions.

Since March 2013, the MULTIFAN-CL source code has undergone substantial developments, and those have been described in earlier reports (e.g. Davies et al. 2020), and the recent developments in 2021-22 are described in Sections 4 and 5.

Following the addition of these new features to the development version, regular testing of this versus the release version aims to ensure the integrity of existing operations. Known as “benchmark tests”, those undertaken in 2021-22 are described in section 2.6. The development version was last tested in May 2022 versus the release version 2.0.8.2. The positive result then defines the development version as being the **benchmark** source code, then to be posted as version 2.0.8.5. Subsequent development versions will then be tested relative to the benchmark to establish their integrity, after which they may be defined as the new benchmark development version. The testing framework entails two levels of tests.

1. Establish the accepted development version

The first level of testing ensures the integrity of existing model features by undertaking tests using a range of single-species data including: ALB2012, ALB2015, BET2011, BET2014, BET2017, YFT2011, SKJ2011, STM2012, SWO2013, SWO2017, YFT2014, YFT2017, SKJ2014 and SKJ2016; to conclude that single model evaluations and the fitted solutions are sufficiently close to regard the development version estimates as being essentially similar to the benchmark version. This indicates integrity of the development version for undertaking single-species model evaluations. Results are compared among the versions and operating systems, to confirm that the development and release versions produced identical solutions. When differences are found, which can be attributable to improvements in the development version, these are accepted.

Tests using multi-species data disaggregated among species are done which entails comparing the fitted solutions of the development version code versus those solutions obtained using the corresponding data for each species fitted individually. These tests concluded that the operations applying to each population in the disaggregated model have integrity and effectively emulate the solutions obtained when each population is modelled individually. Note that species-specific fisheries data were supplied to the models in the test data examples used. Testing was not conducted using test data for which all fisheries data were aggregated among species (or sexes).

Similarly, tests are done for deterministic and stochastic projections with the pair-wise comparisons among versions and operating systems being made.

A positive test result is when the benchmark tests conclude that the development version conserves the existing features, and so can either be advanced as the new release version, or accepted for the new benchmark development version.

2. Establishing integrity of new features

This second level of testing entails a detailed examination of new features. The inputs and model configuration are customized for the new features and the operation of the new algorithms are evaluated in respect of the original formulations. During 2021-22 this level of testing was done for the new features (see section 4), to ensure the correct calculations and the expected results.

Review of Testing Framework

In January 2016 the testing framework was reviewed by project members with the following agreed tasks for improvements:

a) Tidy up the testing framework functions and utilities so as to be as automated as possible and more user-friendly with a view to including other team members in running the tests.
b) Upgrade testing framework functions and utilities for applicability to both single-sex and multi-sex file formats, with portability over condor.
c) Integrate the testing framework functions and utilities into the R4MFCL package and ensure compatibility with all assessment modelling applications.
d) Create a GitHub repository for the testing framework functions, utilities, and testing data.
e) Consolidate the R4MFCL GitHub repository with Rob Scott as the lead developer, and add access levels to Nick Davies as a support developer.
f) Construct a suite of routine tests for the R4MFCL package to be run following each revision to the repository, and load the updated R4MFCL package to the testing framework.
g) Construct a single routine MULTIFAN-CL test operation (e.g. single-evaluation of a fitted test model solution) to be conducted daily and directly from the Jenkins compilation utility that returns an exit status value, with an email report sent to the project developers.

Little action has been taken on these tasks and is also unlikely in the remaining part of 2022 It has been identified as a concern, and that they be included in the 2022-23 work plan for the MULTIFAN-CL project.

With the routine compilation and development of a macOS executable now firmly implemented in the project, the testing framework includes the MacMini host, and the macOS executable within tests among platforms and versions. The framework therefore conducts testing upon all 3 platforms simultaneously over the Condor network. The test analyses perform pair-wise comparisons among versions and over three platforms: Linux, Windows, and macOS.

2.9.3 Viewer

The main notable changes during 2021-22 included:

- renewing the digital certificate that is used when signing the Microsoft Windows version of the software for distribution
- MFCL Viewer has been ported to Java 18 and JavaFX 18 but, due to issues during packaging, it is distributed with Java 18 and JavaFX 17 instead.
- A new series of tabs and plots related to Observed and Predicted CPUE have been added.

2.9.4 Condor parallel processing facility

The Condor (www.condor.wisc.edu) facility has been used routinely for managing multiple MULTIFAN-CL model runs on a grid currently numbering more than 40 computers; being windows or Linux platforms. Support for 32-bit architecture has been discontinued since MULTIFAN-CL executables in this architecture have not been produced since version 1.1.5.9. This grid enables intensive model runs for: benchmark testing MULTIFAN-CL development versions; undertaking stock assessments that entail multiple model runs (e.g. sensitivity analyses and structural uncertainty analyses), and for management strategy evaluation. During 2021-22, additional Linux Virtual Machines were added to the grid to increase the number of model runs possible using the Linux development version executable.

2.10 User’s guide

A revision to the MULTIFAN-CL User’s Guide (Kleiber et al. 2018) has not yet been completed to include the developments since version 2.0.5.1. Proposed future revisions include: incorporating the suggestions arising from the earlier Training workshops; and the recent features added to up to and including version 2.0.8.6. The revised version will be posted on the website http://www.multifan-cl.org/.

3 TRAINING WORKSHOP

Preceding April 2020, training workshops for users of MULTIFAN-CL were scheduled for April of each year and held on-site at the Oceanic Fisheries Programme (OFP, Pacific Community) in Nouméa, New Caledonia.

Due to the COVID-19 pandemic and the related travel restrictions, this format for the workshop is no longer possible. During February-July 2021, on-line training was held for the OFP stock assessment scientists. This format of personal training proved to be very effective, and was therefore repeated and expanded during 2021-22, as follows:

- 29 Sep. - 6 Oct.2021 – a five-day workshop, 2-3 hours per session
- 26 November 2021 – session on Hessian diagnostics
- 16 December 2021 – session on the catch-conditioned method
- 10 February to 2 June – 14 sessions on a wide range of topics, including development of the SKJ2022 stock assessment

4 NEW FEATURES

All new features implemented into MULTIFAN-CL source code have firstly been added to the development version. Once these features have been tested for their integrity, with no impacts on existing features, then the development version is merged into the release version of the code. The current release version is 2.0.8.5. All of the developments described below are currently implemented in this version, while others (such as enhancements) have been made to the development version, (since May 2022 and as listed in section 2.4.6), and will be merged to the next release version upon the completion of the benchmark testing. An overview of these new features was provided to the pre-assessment workshop (Hamer, 2022), but are described in more detail here.

4.1 Survey Fishery CPUE likelihood

Davies et al. (2021) described the initial development stages of a log-likelihood term for CPUE indices derived from survey fisheries. The feature was completed and an outline is provided here with an example presented of its implementation.

4.1.1 Rationale

The existing method for estimating fishing mortalities in MULTIFAN-CL is called the “catch-errors” approach, and while it is effective, and especially in cases where effort is better observed than total catch, an alternative formulation with fewer variables could be an improvement, both in terms of convergence characteristics and ultimately run time. The development of an approach that calculates fishing mortality conditioned upon the observed catches was formulated and implemented in MULTIFAN-CL during 2020-21, and improvements and testing have been made during 2021-22. Part of this improvement is the implementation of a new feature being a likelihood term for CPUE indices from survey fisheries.

4.1.2 Methods and Testing

Typically, certain fisheries (called survey fisheries) provide indices of relative abundance, usually expressed as catch-per-unit-effort (CPUE) and in many fisheries modelling software it is fitted via a dedicated likelihood term. The survey fishery data are input as a "normal" fishery, but are controlled in how the data are treated in the model based on an activation flag. The configuration of a survey fishery in respect of how the catch and effort data are employed is as a **non-extraction fishery**. Catch data are for a "true" survey fishery in that they are negligible, e.g., 1 fish or 1 kg. As such, the fishery is excluded from the Newton-Raphson procedure for estimation of fishing mortalities or within the regression for the fishing mortality levels and observed effort relationship (**fm_level_rltnshp**). The sole purpose of the fishery is to provide CPUE observations and size compositions (if available), and the model is fitted to the indices in a dedicated likelihood function. The selectivity at age of each survey fisheries is estimated either by fitting the fishery-specific size composition data, or if there is no size composition data available for the survey fishery, its selectivity can be shared with the estimated selectivity from an extraction fishery.

A general form for the negative log-normal likelihood for the survey index for a specific fishery, $\boldsymbol{k}$, is:

$$0.5\sum_{i}\log(\gamma_i\hat{\sigma}^2)+0.5\sum_{i}\frac{(P_i-O_i)^2}{\gamma_i\hat{\sigma}^2}$$

Where P_i and O_i are the normalised predictions and observations, respectively, on the log-scale. The predicted index is:

$$\hat{I}_{ik} = \sum_j N_{rij}^{mid} S_{kj} \overline{w}_j$$

Where N_{rij}^{mid} is the mid-period population numbers at age in the i^{th} period in the region r that corresponds to that of the survey fishery k observation, S_{kj} is the selectivity at age of survey fishery k, and $\overline{w}_j$ is the mean weight of fish at age j. Where observed indices are in respect of population numbers, the $\overline{w}_j$ is omitted. The normalised prediction is:

$$P_{ik} = \log(\hat{I}_{ik}) - \text{mean}(\log(\hat{I}_{ik}))$$

Simplifying the negative log-normal likelihood slightly gives:

$$0.5 \sum_i \log(\gamma_i) + n\log(\hat{\sigma}) + 0.5 \sum_i \frac{(P_i - O_i)^2}{\gamma_i \hat{\sigma}^2}$$

Where n is the number of observations (i.e. time periods), γ_i is the relative precision of the ith observation. Taking the derivative with respect to $\hat{\sigma}$ and setting it equal to zero

$$\hat{\sigma}^2 = \sum_i \frac{(P_i - O_i)^2}{n\gamma_i}$$

So, the concentrated likelihood is:

$$0.5 \sum_i \log(\lambda_i) + n \log\left(\sqrt{\sum_i \frac{(P_i - O_i)^2}{n\lambda_i}}\right) + 0.5 \sum_i \frac{(P_i - O_i)^2}{\lambda_i \sigma^2}$$

The observed and predicted CPUE indices for the survey fisheries were added to the plot-*.rep report to facilitate the ready inspection of model fits within the Viewer (see section 2.9.3). The CPUE likelihood term as estimated was added to the likelihood components report (test_plot_output).

4.1.2.1 Grouped catchabilities

The likelihood term can be applied for the case of more than one survey fishery having the same catchability, e.g., where an identical fishery (either commercial or research) operates in more than one region. The assumption of stationary catchability in such cases is reasonable. The above likelihood readily accommodates this instance such that the fisheries are "grouped" into a single fishery, and the normalised predictions P_{ik} are calculated for all k within the group. A simple 3-region albacore model was used to demonstrate the implementation of grouped survey fisheries, and is presented in the next section.

4.1.2.2 Concentrated likelihood

The simple 3-region albacore model was also used to demonstrate the implementation of the concentrated likelihood where CVs for each observation were available. The model was first fitted to the simple likelihood that excludes the λ_i, and then to the concentrated term. A comparison is presented in the next section.

The 3-area example was used to explore the effect of altering the λ_i in the concentrated CPUE likelihood. The base values of the λ_i are illustrated in Figure 2, as 2 standard deviations of each observation. A range of alternative values was explored for fishery 20:

- 0.5 * λ_i (**cv_sclr_0.5**)
- 1 * λ_i (**cv_sclr_1**)
- 1.5 * λ_i (**cv_sclr_1.5**)
- 2.5 * λ_i (**cv_sclr_2.5**)

Fishery 20 was selected because of the contrast in relative abundance and a wide range of λ_i values throughout the time series. The other two survey fisheries were held at their original λ_i values.

Each model in the range, aside from **cv_sclr_1**, was run to a reasonable convergence (an additional 3000 evaluations) starting from the **cv_sclr_1** solution. The fit to the CPUE indices were compared and also several key dependent variables.

4.1.2.3 Testing

Example grouped survey fisheries

The effects of assuming stationary catchability are immediately apparent in the relative difference in the absolute index among the fisheries, and hence, among the model regions (Figure 3). Therefore, the observed region-specific catch rates are reflected in the fit of the grouped survey fishery predictions as derived from region-specific absolute abundances. This method provides a means of inferring region-specific abundances from a single fishery supplying observations from a range of regions.

Example concentrated likelihood

Figure 3 illustrates the effect of applying the concentrated likelihood where the model was fitted both excluding and including the λ_i for each observation. The effect is most evident in Fishery 20, where the predicted indices are higher for the observations in the early time period of the series.

Altering λ_i does impact on the relative influence of the CPUE time-series within the integrated model fit by: changing its absolute CPUE likelihood contribution; and, the estimated relative trend in relative abundance (Figure 4). For this example, lower precision results in higher relative abundance in the early time periods of the series. This alters the regional absolute biomass estimates, due to differences in the fit with respect to the other survey fisheries time series. Altering λ_i may therefore offer a means for making assumptions as to the reliability of each of the observed CPUE time series.

Employing in production assessments

The preliminary testing indicated the concentrated likelihood demonstrated certain expected effects on a simple example. A small correction was made to the formulation that did not substantially alter the likelihood values. However, we remained unsatisfied that it is being implemented as intended, and it is unclear what interaction occurs between the λ_i and a fishery-specific penalty weight. It was concluded that this requires further testing.

The non-concentrated form of the likelihood, that assumes temporally stable CVs for the survey fisheries CPUE indices, was considered to be correct and able to be employed for production assessments, with group-specific penalty weights being applied.

4.1.2.4 Further work required

The concentrated likelihood requires further work to evalulate the relative effects of the λ_i and a fishery-specific penalty weight. Alternative formulations will also be explored, such as that employed in the generic software CASAL (Bull et al. 2012).

4.2 Long-term tag loss

4.2.1 Rationale

The impetus of this feature was the 2012 Independent Peer Review of the BET2011 model that included the development of this feature in the set of recommendations arising from the workshop.

Recommendation “c”. Long-term tag loss

“Allow for long-term and initial tag-loss. Currently initial tag-loss is implemented by reducing the number of animals tagged when inputting data to the model and no account can be taken of long-term tag-loss.”

4.2.2 Method

It is required to implement the probability $\boldsymbol{Q_t}$ of the tag being retained at time $\boldsymbol{t}$ after release, i.e.

$$Q_t = e^{-L_t}$$

Where $\boldsymbol{L}$ is continuous (often called type 2) tag shedding. It is, in effect, an addition to natural mortality, $\boldsymbol{M}$, for the tagged population.

$\boldsymbol{L}$ may be assumed to be constant over all $\boldsymbol{t}$, but other more complicated models are possible, e.g. where $\boldsymbol{L}$ changes over time. The simple time-invariant method was implemented in MULTIFAN-CL as a prototype, with $\boldsymbol{L}$ being a parameter value to be provided, i.e. assumed, (usually estimated from stand-alone analysis of double tagging data) when tagging data are included in the integrated model fit.

The number of tagged fish in a release cohort $\boldsymbol{c}$ in a time interval $\boldsymbol{t}$ is therefore a function of total mortality, $\boldsymbol{Z}$ (comprised of fishing and natural mortality) and $\boldsymbol{L}$:

$$N^T_{ct} = N^T_{ct-1} e^{-(Z_{ct-1} + L_{ct-1})}$$

where

$$Z^T_{ct} = \sum_f F^T_{ctf} + M_{a(ct)}$$

and where

$$a(ct) = a^{rel}_c + t - t^{rel}_c$$

that relates historical time to the age of a particular cohort in respect of its release event time. The same implementation is applied for cohorts that have graduated to the tag-pooled cohort.

An adjustment is applied to the fishing mortality to account for $\boldsymbol{L}$ when deriving the predicted catch of tagged fish (tag recaptures) from a cohort:

$$\acute{F}^T_{ctf} = F^T_{ctf} + (\log(1 - S^T_{ct}) - L_t) - (\log(Z^T_{ct}) + L_t)$$

This adjustment ensures compatibility with the tagged population dynamics equation above, and the second and third terms simplify to zero when $\boldsymbol{L}$=0.

Note that the current implementation of the tag loss variable is:

- a fixed, assumed value over time (time-invariant);
- constant among tagging events; and,
- may not be estimated as an independent variable.

The implementation in MULTIFAN-CL accommodates the cases of:

- Catch-errors and catch-conditioned methods for estimating fishing mortality
- Mixing and non-mixing periods
- Grouped and ungrouped tagging events
- Pooled tagged group
- Annual and quarterly recruitments

4.2.3 Testing

The design for evaluating deterministically the relative effects of long-term tag loss on the tagged population and predicted recaptures entailed a grid of the 4 scenarios:

- Unpooled and pooled tagged cohorts
- Annual or quarterly recruitments, both models with quarterly time-steps. The example for annual recruitments was ALB2021 (having 24 tag release events), and for quarterly recruitments SKJ2019 (having 269 tag release events).

For each scenario, a deterministic run using an existing converged solution (without tag loss, i.e. preceding the feature) was performed: with, and without tag loss, assumed level = 0.05. The models employed the catch-conditioned feature. The following table illustrates the design.

	af57=4 SKJ2019	af57=1 ALB2021
pooled af96>0	• no tag loss • tag loss=0.05	• no tag loss • tag loss=0.05
unpooled af96==0	• no tag loss • tag loss=0.05	• no tag loss • tag loss=0.05

A comparison of the predicted catches of tagged fish illustrates the effect of deterministic long-term tag loss on the tagged population numbers, resulting in a consistent relative decrease in predicted recaptures over time (Figure 5). The decrease is consistent with the assumed exponential loss rate = 0.05, and for each of the four scenarios. The reduction in the predicted recaptures will obviously depend upon the assumed loss rate, and for the value assumed in these tests (0.05), the effect on the tagging data likelihood term for the deterministic runs was marked.

A comparison was made of the converged solutions of the SKJ2019 catch-conditioned example with pooled tag groups, fitted with, and without, long-term tag loss = 0.05. As might be expected, the predicted recaptures of both solutions are very similar as they are fitted to the same observed recapture data, and the solution including tag loss produced proportionally lower population abundance estimates due to the smaller tagged population resulting from the assumption of tag loss, and consequently higher fishing mortality (Figure 6).

Testing the relative effects of long-term tag loss on the tagged population and predicted recaptures using a catch-errors example with a pooled tag group produced very similar results to that of the catch-conditioned example, and are not presented here.

The testing indicated the feature is performing as expected, it is operational for almost all model scenarios undertaken in MULTIFAN-CL, and is ready for implementation in production stock assessment models.

5 ENHANCEMENTS AND BUG FIXES

5.1 Catch-conditioned method

Substantial enhancements were made to the catch-conditioned feature during 2021-22 to prepare this feature developed in 2020-21 for implementation in production stock assessment models. These included ensuring the method is robust during minimisations by avoiding stock collapses when deriving the Newton-Raphson solutions for fishing mortality, improving the regression for the fishing mortality vs observed effort relationship, undertaking zero-fishing mortality model evaluations for fishing impact analyses, and undertaking deterministic model projections.

5.1.1 Fishing mortality vs observed effort relationship (fml_effort_rltnshp)

Background is provided here of this relationship. The Newton-Raphson (N-R) solution for the fishing mortality levels applies in the cases where an observed catch is available. The exceptions are the cases where observed catch is unavailable. An orthogonal-polynomial regression model using a Gram-Schmidt (G-S) design matrix is formulated, and a robustified fit using a mixture distribution, to estimate fishery-specific relationships between the: **fishing mortality levels** (estimated from the N-R) for those incidents where catch is available, and the **observed effort**. The observed (O) and predicted (P) factors in this relationship are:

- O = F_pred – log(E_obs) for the fishery where F_pred is that prediction by the N-R for that fishing incident associated with the catch for the associated effort, E_obs. (log-scale); and,
- P = poly_coffs * G-S where G-S are the corresponding rows of the Gram-Schmidt design matrix, and poly_coffs are the estimated polynomial coefficients.

In effect the observations are the ratio: the estimated fishing mortality/observed effort; and is analogous to a catchability. The polynomial * effort therefore provides a predicted fishing mortality for that given level of effort, and therefore the polynomial is itself analogous to a predicted catchability. The predicted fishing mortality is then used, as normal, for calculating the predicted catches.

- F^{pred} = poly_coffs * G-S * E_{obs}
- $C^{pred} = F^{pred} S_{age} N_{age}$

This estimated relationship is denoted here as the **fm_level_rltnshp**. The degrees for the estimated polynomial for each fishery are specified, and may include seasonality in the catchabilities, that define the dimensions of the G-S, and the parameters are estimated within the integrated model fit. Then for those incidents having missing catch, the fishing mortality level prediction from the relationship using the observed effort, is applied to predict the catch. The calculation is also extended to cope with the case of grouped fisheries.

5.1.1.1 Robust student-t

The regression between observed (O) and predicted (P) was enhanced by including a revised formulation for the student-t distribution with the estimation of degrees of freedom (dof). Robustness of the student-t regression was ensured by adding a very small value to the variance to prevent implausibly low dof values and subsequent minimisation instability.

5.1.1.2 Iterative Least-Squares Regression

A method was attempted that precludes the estimation of parameters by using an iterative least-squares solution of the fm_level_rltnshp. However, it cannot be performed within a model iteration because it is non-differentiable (i.e., the derivatives of the previous iteration are not available). The key point is the implementation of the fml_effort_rltnshp within the model evaluation to predict catches where they are not known. In the case where the relationship coefficients are independent variables, these predictions are related to these variables for the current derivative calculation. In the case where the coefficients are solved from an iterative least-squares solution, the missing catch predictions within the model evaluation are made using the coefficients from the previous derivative calculation (i.e., the previous model evaluation). Consequently, this latter case is not feasible when using the ADMB minimisation method.

5.1.2 Robust Newton-Raphson

Robustness in the N-R calculation was ensured to deal with low population sizes - within an iteration, the population can crash completely before the end of the estimation period and this results in minimisation instability. In most examples trialled, a maximum fishing mortality penalty is sufficient to prevent this from happening, but obviously that is not always true when the crash occurs within an iteration. A “soft” penalty was therefore added to prevent this from happening among model iterations, that imposes a penalty gradually as a fraction of a specified maximum level and applies the adjusted total mortality that ensures a viable population size.

5.1.3 Zero-fishing mortality model evaluations (F0eval) of the catch-conditioned method

Zero-fishing mortality model evaluations are undertaken to investigate the relative impacts of fishing. For the catch-conditioned method, this was firstly implemented by setting log(fishing mortality) = -70.0, and completely excluding all fisheries from the N-R procedure. However, this generated several problems with partial fishery closures and mixed catch or effort incidents. Rather it was altered to use the N-R to act upon near-zero catches = 1.e-03 for those fisheries having zero fishing mortality. This was a more stable method and maintains consistency in the algorithm used for the normal and F0eval model evaluations.

5.1.3.1 F0eval model evaluation with dynamic recruitment compensation

Dynamic recruitment compensation for the F0eval entails scaling the estimated recruitments by the ratio of the BH-SRR predictions at the levels of the fished and unfished adult biomass corresponding to each time period. For the catch-errors model, an iterative method is applied with fixed fishing mortalities, such that the F0eval evaluations converge upon an unfished spawning biomass given the compensated estimated recruitments. This feature had not yet been implemented or tested for the catch-conditioned method issued in version 2.0.8.2 and was subsequently developed.

There are two aspects of this feature unique to the catch-conditioned method. For the F0eval, only the unfished population (N_q0) for the first time period (quarter) is available from the initial population calculations; and, the fishing mortalities are not fixed but are estimated based upon the catch and the vulnerable biomass. For both aspects, when the annualized BH-SRR is employed, this requires a method using a prior model evaluation without recruitment compensation, before applying the method.

The algorithm entails two model iterations to be performed, with the recruitment compensation multiplier calculation being done within the second iteration. This avoids the re-initialisation of N_q0 outside of this routine. This must occur following the iterations needed for convergence to obtain the initial population (N_{init}) conditions, i.e., after the 5th N_{init} iteration. The total iterations number only 7, and because this is undertaken only during the reporting phase of model execution, it does not add to the total processing time for obtaining a model solution.

The SKJ2021 catch-conditioned model was used for testing the feature in two configurations, with the two options for recruitment compensation (activated by age_flags(171) =1) applied to each:

- Quarterly BH-SRR – age_flags(171) = 0 or 1
- Annual BH-SRR – age_flags(171) = 0 or 1

A comparison of the unfished adult biomass ($SB_{f=0}$) among the two options for recruitment compensation illustrates the effect of the feature, where the recruitment compensation increases the unfished adult biomass compared to the F0eval lacking compensation (Figure 7).

5.1.4 <u>Catch-conditioned projections</u>

The aim of this development was to implement and test the projection capability of the catch-conditioned method with fisheries conditioned upon either catch or effort during the projection time periods.

5.1.4.1 Method

There are two aspects of implementing this feature for the catch-conditioned method. For the projection time periods, the fishing mortality calculations for those fisheries conditioned in respect of catch are managed as is done normally for the estimation model time periods. No additional development was required for this aspect of the projections. However, for those fisheries conditioned in respect of effort, an algorithm was developed consistent with that used for fishing incidents during the estimation time periods having missing catches. This entailed developing structures taken from the Gram-Schmidt design matrix applied to the estimated polynomial relationship between effort and estimated fishing mortality levels (fml_effort_rltnshp), to be applied for those projection fishing incidents having only effort data. To predicted catches for effort-conditioned fisheries in the projection time periods the algorithm entails:

- Copying the Gram-Schmidt design matrix rows for the fish times in the final calendar year of the estimation period which are stored specific to each effort-conditioned fishery; and,
- In each projection fishing incident, the fishery- and month-specific design vectors are applied to the fml_effort_rltnshp polynomial function and the “observed” future effort to predict the catch for that incident.

Effectively, this is analogous to the algorithm applied in the catch-errors model, where the catchabilities in the terminal calendar year of the estimation period are assumed for each projection year.

In addition to this, it was necessary to upgrade the population numbers at age routine to:

- apply projection recruitments as predicted from the annualized BH-SRR, with the application of both regional and seasonal distributions for apportioning the predictions; and,
- Ensure that both catch and effort were set to zero for the F0eval evaluations applied in fishing impact analyses.

5.1.4.2 Testing

A SKJ2022 catch-conditioned projection model was used for testing the feature in two configurations for future recruitments:

- **Proj_avg_recr** – predictions from BH-SRR distributed in respect of the average regional distribution
- **Proj_seas_recr** – predictions from BH-SRR distributed in respect of both the average regional and seasonal distributions

Comparisons of various model variables among the estimation and projection time periods are presented to illustrate the implementation of the feature, and also among the two projection recruitment configurations. The total biomass during the estimation periods of both projection models were identical to that of the original estimation model from which the projection models were derived, as expected (Figure 8). Similarly, total recruitments during the estimation periods of both projection models were identical to that of the original estimation model, with the application of the seasonal distribution having a marked effect on the Proj_seas_recr projection recruitments (Figure 9). All penalties and likelihoods as calculated over the estimation time periods were identical among the three models compared. This test ensured the function evaluation for the estimation model time periods were unaffected by the projection model configuration.

Of the 39 fisheries, those conditioned in respect of effort for the projection time periods were: 2, 5, 8, 12, 14, 15, 19, 20, 25, 26, 29, and 30. Status-quo effort, i.e., as taken from the final calendar year of the estimation time periods, was assumed for these fisheries in the projection time periods. While future catches will depend upon the recruitments in each projection time period, a useful comparison can be made between observed catches in the latter estimation time periods and those predicted based upon the assumed future effort. Projection catches were very consistent with those of the latter estimation time periods for both projection models (Figure 10 and Figure 11). This is expected given that the main model variables are held constant or relatively similar during the projection periods: catchability, effort, and recruitment.

The zero-fishing model evaluation undertaken for fishing impact analysis entailed setting catches and effort to zero for the catch- and effort-conditioned fisheries, respectively. A comparison of the recruitments with that of the normal fishing model evaluation illustrated: identical recruitments during the estimation time periods (as expected); and, around 5-6% higher recruitments on average for the zero-fishing model evaluation during the projection time periods due to the higher adult biomass (Figure 12 and Figure 13). Given that depletion is a ratio estimate, the levels during the projection time periods were very similar among the two projection model options (Figure 14).

5.2 F0eval with the initial equilibrium population a function of x*M

Testing was done of the functionality of the F0eval model evaluation under the assumption that the initial population (N_{init}) is a function of $x*M$ at equilibrium, where x is a defined multiplier value on natural mortality, M. This option is denoted: **Ninit_M**.

Given the method used for the Ninit_M option, there is no difference in N_{init} between the normal and F0eval model evaluations, because:

- For both the normal and F0eval evaluations, the actual dynamic model total mortality Z over n time periods is not applied in deriving N_{init}; and,
- N_{init} is only a function of M and the multiplier x.

As such, the definition of this option for N_{init} is not a population in exploited equilibrium, but rather an "adjusted" unexploited equilibrium population. The notion was explored where x is considered as being this adjustment from the unexploited equilibrium (virgin) state, and produces an un-natural condition based upon an apriori assumption that the population was not in unexploited equilibrium. For the F0eval, one might then expect that this adjustment would not be valid, so it could be excluded entirely, and the model commenced from the virgin state. Note that this adjustment applies to initial population conditions generally, and is functional irrespective of the fishing mortality model used (catch-conditioned or catch-errors methods).

A comparison of the adult biomass among the normal fishing model evaluation and the two options for the F0eval evaluation - including and excluding the multiplier - illustrate the effect of removing the multiplier for the F0eval such that N_{init} is restored to the virgin level (Figure 15).

5.3 Dependent variables in the *.var and *.cor reports

Several new dependent variables were added to the variance (*.var) and correlation (*.cor) reports, and all the ratio dependent variables from the normal and F0eval evaluations were added to the *.cor report.

5.3.1 $SB/SB_{F=0,10yr.avg}$ new dependent variables

New dependent variables (dep_vars) were added to the *.var and *.cor reports for fished:unfished adult biomass ratio reference points. The unfished reference point variable is the unfished adult biomass (SB), being an average over 10 calendar years up to the year preceding the latest calendar year, (as defined by SC9):

$SB_{f=0\_10yr.avg}$ = mean(SB_{f0}) for the years (ny-10) to (ny-1)

where ny are the number of calendar years in the model.

The two reference points in respect of this unfished reference point are:

- SB_{latest}/ $SB_{f=0\_10yr.avg}$ – "latest" is the last calendar year; and,
- $SB_{recent}/SB_{f=0\_10yr.avg}$ – where "recent" relates to the average fished adult biomass over the last 4 calendar years including the latest, i.e., mean(SB) for the years (ny -3) to (ny)

In order to "future-proof" the $SB_{recent}/SB_{f=0\_10yr.avg}$ dependent variable from adjustments to the period that defines SB_{recent} , "recent" may be defined as:

- the average over the last N years including the latest (where N are calendar years) i.e., mean(SB) for the years: (ny - (N -1)) to (ny)

where for the default case N =4, but its value may be assigned using parest_flags(59).

5.3.2 Correlation reports – new log(MSY) dep_vars

Analyses between the correlations for dependent variables expressed on either the log-scale and normal-scale were not possible, as they need to be expressed on the same scale. Correlations between ratio depletion variables and MSY-related dep_vars may be of interest in terms of their effect on management quantities of interest. The MSY-related dep_vars are reported on the normal scale, while the ratio depletion variables are reported on the log-scale. Therefore, a set of MSY-related dep_vars (SB_{latest}/SB_{MSY}, B_{latest}/B_{MSY}, and F_{latest}/F_{MSY}) were expressed on the log-scale and added to the *.var and *.cor reports.

5.3.3 Correlation report includes ratios of normal and F0eval dep_vars

The depletion ratio dep_vars, e.g., $SB/SB_{F=0}$ have been reported in the *.var report but have not been reported in the *.cor report. Code was added to the covariance and correlation calculations to include these variables such that two cases may occur:

- When the dep_var gradients for the F0eval evaluation are absent, the existing code operates with the correlation report produced for only the normal fishing evaluation dep_vars; and,

- When the dep_var gradients for the F0eval evaluation are present (file: *.dp2), the new code operates with the correlation report produced for both the normal and F0eval evaluation dep_vars.

The row labels of the correlation report have also been corrected to identify both dep_vars comprising the ratio.

5.4 Initial survival rates for N_{init}

The options for deriving N_{init} frequently contribute to minimization instability, and it is convenient to eliminate this potential cause of convergence problems. Therefore, a developer's option was added for assuming a fixed initial survival rate specified by a flag setting that sets it equal to a fixed value = 0.8.

In addition, this area of the code was improved by consolidating the initial survival rate assignments depending upon the catch-conditioned method and the selected N_{init} population option. Local survival rate variables were created within the various routines to avoid conflict with the corresponding global variables.

5.5 Selectivity coefficients penalty

It is possible that during minimization implausibly low selectivity coefficients may be trialled that result in spline node coefficients approaching the bounds, resulting in an ill-determined model. This condition was remedied with an option for applying a penalty of the form:

```
tmp=spen*square(fsh.bs_selcoff(i,j,k,l)+15.0);
```

where spen may be specified by a flag setting, and 15.0 is a nominal value consistent with the range of the selectivity coefficient bounds.

5.6 Bounds on totpop_coff

Testing undertaken in 2021-22 among a range of examples revealed instances where extremely small populations may be estimated. It was necessary for the lower bound on the total population scaling parameter (totpop_coff) to be reduced to 10 to allow for such small populations.

5.7 Hessian diagnostics - additional reports

In addition to the Hessian diagonals reported as required for the method "empirical evidence for relative influence of negative eigenvalues" (Davies et al. 2020), a number of additional reports have been added for undertaking diagnostics of the Hessian solution itself. These are:

- *_pos_sqrt_hess_diag2 – the diagonal of the positivised Hessian square-root inverse
- *_pos_hess_inv_diag2 – the diagonal of the positivised Hessian inverse
- *_pos_hess_cor – the pseudo-covariance matrix taken from the positivised Hessian inverse
- *_hess_inv_diag – the inverse normal (non-positivised) Hessian diagonal
- *_pos_hess_cov – the covariances of the positivized Hessian

5.8 Fish_flags sanity checks

Several routines exist that review the settings applied to fish_flags for their validity in respect of the assumed grouping among fisheries as specified for a range of features. Screen error messages are displayed and the execution is stopped. During 2021-22 these were enhanced to include a number of flags critical to the catch-conditioned method and other model aspects, for example:

- age_flags(177) and parest_flags(155) are not both zero – indicating an attempt for the mean+deviate recruitment estimation, while the assignment to totpop_coff is disabled
- N_{init} option as a function of M and the catch-conditioned option for initial survival as a spline function are both active
- Various grouping flags as used in catch-conditioned method
- The data_fish_flags specifying the units of catch and the grouping flags for the CPUE likelihood calculation

- The number of recruitments per calendar year (age_flags(57)) and the number of time periods for deriving average total mortality that specifyies initial survival rate (age_flags(95)), such that age_flags(95) must be > age_flags(57)

These routines assist the user in avoiding spurious flag settings that will cause model execution failures.

5.9 Independent variables - labelling

Ready identification of the confounded parameters contributing to negative Hessian eigenvalues requires viewing the parameter indices in respect of the various stratifications or structures assumed in the model, i.e., spatial, species/stock/sex, temporal and fisheries. The listing of independent variables (estimated parameters) is provided in the "xinit.rpt" displaying the variable number from 1 to nvar, and the name, e.g., effort_dev_coffs. The independent variable structures are often multi-dimensional arrays, and it is difficult to identify to which model structure they relate, e.g., a fishery, region or time period. What is needed is for the "xinit.rpt" to include the indices within the variable's data structure to which each individual variable relates.

The report listing the independent variables was therefore improved to include details of the indices of those being provided in arrays and matrices.

5.10 Bug fixes

5.10.1 Catch-conditioned method – F0eval initial population calculations

For users of MULTIFAN-CL, the most commonly used assumption for initial population conditions is that of exploited equilibrium (section 5.8), with the initial fishing mortality being the average over the first *n* specified time periods. While this assumption operates satisfactorily with the catch-conditioned model for the “normal” fishing mortality (Fnorm) model evaluation, it was incorrect for the F0eval evaluation, such that N_{init} was essentially identical for both the Fnorm and F0eval evaluations. This is counter-intuitive given that the F0eval excludes all the initial fishing mortality (F_{init}) included in the initial survival rate (S_{init}) used for calculating N_{init}.

For the catch-conditioned model, F_{init} is unknown in the first model time period, and while this may be approximated for the F0eval case, it will only be an accurate approximation if all fisheries have zero fishing mortality. This is not always the case during fishing impact analyses with partial fishery de-activation. It is therefore necessary to use an iterative method for converging upon a S_{init} for the F0eval. The algorithm for this is as follows:

1. Fnorm evaluation complete – produces the model actual S_{init} over the ***n*** time periods
2. F0eval Iteration 1:
    a. N_{init} is calculated using model actual S_{init} from the Fnorm evaluation in 1.
    b. Undertake the F0eval with fishing mortalities calculated for observed catches of effectively zero (0.001) in all fishing incidents
    c. Calculate the model actual S_{init} obtained from this evaluation
3. F0eval Iterations 2 to 5:
    a. N_{init} is calculated using model actual S_{init} from the F0eval from iteration ***n***-1
    b. Undertake the F0eval with fishing mortalities calculated for observed catches of effectively zero (0.001) in all fishing incidents
    c. Calculate the model actual S_{init} obtained from the ***n***th iteration’s evaluation over the ***n*** time periods
4. Final evaluation
    a. N_{init} is calculated using model actual S_{init} from the F0eval from the 5th iteration
    b. Undertake the F0eval with fishing mortalities calculated for observed catches of effectively zero (0.001) in all fishing incidents
    c. Report the F0eval dependent variables

With all fisheries having zero catches, the algorithm converges after 2 iterations, but may take more with partial fishery de-activation (only selected fisheries having zero fishing mortality).

Given the corrected, and higher, S_{init} obtained from the algorithm, the F0eval estimates of the model dependent quantities for N_{init} and for some time periods immediately following, are now clearly different to that of the Fnorm evaluation. This is illustrated in a comparison of total biomass for the two SKJ2019 model evaluations (Figure 16).

A correction was also made to the assignment of the population numbers at age during the F0eval during fishing impact analysis with partial fishery de-activation. These assignments were required for the N-R calculations in this special case.

5.10.2 Catch-conditioned method – correction to G-S matrix

The construction of the G-S design matrix made robust to the instances of fisheries having no effort data available, and therefore the matrix variable is unallocated. The call to the routine that builds the fishery-specific design matrices was made conditional upon the number of rows being > 0. This condition had already been implemented for the estimation of the fml_effort_rltnshp among grouped fisheries, but was required for the ungrouped case.

5.10.3 Catch-conditioned method – consolidated flag conditions

For the catch-conditioned method, there are numerous flag settings that control the activation, estimation and penalty calculations of parameters for the spline coefficient of S_{init} (kludged_equilib_coffs), polynomial coefficients of the fml_effort_rltnshp (implicit_fm_level_regression_pars), and the fml_effort_rltnshp regression likelihood calculation. The operations conditional upon these flags were reviewed and consolidated so as to avoid potential conflicts and errors during run time.

5.10.4 Catch-conditioned method – infinite loop on calculating the initial q

For efficiency, a rapid algorithm is used to obtain an initial estimate for fishing mortality prior to undertaking the N-R estimation that employs high precision. An improvement was made to this algorithm that avoids an infinite loop occurring (that is possible in certain extreme circumstances) by setting a maximum number of iterations.

5.10.5 Catch-conditioned method – identical Fnorm and F=0 depvars in the .var report

In the case of the catch-conditioned method, the adult biomass dep_vars calculated from the normal and F0eval model evaluations as required for the variance calculations were identical, resulting in the log-depletion ratio equal to 0. It was diagnosed that the call to the routine zero_out_catches() for the F0eval evaluation was only being made from the minimizing_routine(), and not during the dep_var gradient calculation routine. Adding the required call to this routine for the F0eval resolved the error.

5.10.6 Multi-spp percent maturity in the F0eval adult biomass calculation

The percent maturity used for calculating the unfished adult biomass from the F0eval in the case of multi-species was not being correctly assigned when reporting to the plotq0-*.rep report. This was corrected with the species-specific pmature variable being assigned in this instance.

5.10.7 F0eval fishing mortalities reported in plotq0-*.rep

A correction was made to the F0eval fishing mortalities being reported to plotq0-*.rep that required the assignment of the unfished population numbers at age to the relevant routine.

5.10.8 Negative correlations with respect to depletion dependent variables

A correction was made to the calculation of the gradients of the ratio depletion variables, $SB/SB_{F=0}$ which were inversed, causing the covariances to be negative. The ratio dep_var gradients were revised to be consistent with the form of the ratio estimates themselves. While this error (applying the inverse) did not impact the variance calculations, it was a problem for calculating the correlations.

5.10.9 Likelihood components report - total catch and Dirichlet Multinomial

Individual components of the function evaluation total negative log-likelihood are reported in the "test_plot_output" report. For the case of fishery-specific weightings being applied in the total catch likelihood calculation, no assignments to the class members were being made, and were therefore absent from the report. The necessary assignments to ppstf->tot_catch_like_by_realization(ir,ip,fi) were therefore added.

For the Dirichlet_Multinomial size composition likelihood, it was identified that while the fishery-specific components were correct, the fishery_realisation-specific components were not. An error was found in the assignment to the class member lencontrib_by_realization(ir,ip,fi) = value(ttmp0), and corrected.

5.10.10 Pre-minimisation orthogonal polynomial fit

When converting from the mean+deviates recruitment parameterization to the orthogonal polynomial parameterization, a pre-minimisation fit using the existing solution provides excellent initial values for the polynomial coefficients at the start of the minimization. It was necessary to make this pre-minimisation fit robust to: the output of the -makepar operation, where the 00.par has by default parest_flags(155)=0; and, if in phase 1 the flag is set > 0.

6 APPLICATION OF NEW FEATURES

6.1 Stock assessments for SC18

The catch-conditioned feature has been implemented, and is fundamental to, the 2022 stock assessment for skipjack tuna in the Western and Central Pacific Ocean, and is described by Castillo-Jordan et al. (2022). The enhancements made to this feature during 2021-22 were also essential to the development of this stock assessment.

6.2 Spatial simulation modelling study

The new feature for long-term tag loss has been implemented in a MULTIFAN-CL estimation model that has been developed as a candidate within a simulation study in which the operating model (Dunn et al. 2020) assumes a constant rate. The simulation study is a collaborative workshop between NOAA and NIWA (https://github.com/aaronmberger-nwfsc/Spatial-Assessment-Modeling-Workshop).

7 FUTURE WORK

A listing of the status of the work originally proposed for, that is pending, in progress, or completed during 2021-22 is provided in Table 1. The proposed future work plan for the development of MULTIFAN-CL in 2022-23 is presented in Table 2, which is somewhat different to that provided by Davies et al. (2021) for 2021-22. This is because of the change in the composition of the MULTIFAN-CL development team (retirement of the lead developer, see section 2.1) and the evolving needs of the OFP.

Compared to previous work plans, no substantial new developments are proposed that require innovative mathematical elements. Rather, resources are directed to: consolidating those aspects that are incomplete for recent new features; enhancements of existing features; and, documentation. Specifically, it includes:

- Testing the implementation of examples that employ all the new features and refine the I/O and diagnostic reports.
- The code for existing features will be reviewed and refined; a backlog of bug fixes will be completed; the single long outstanding task from the bigeye tuna independent review panel recommendations will be addressed; and any "small-scale" requests in the tasks list.
- Provide training and support for OFP stock assessment scientists (including participation in an independent review of the yellowfin tuna stock assessment model).
- Providing support for MSE requirements and improvements.
- Catching up on the remaining documentation required for updating the MULTIFAN-CL User’s Guide.

Some of the items in Table 2 (Tasks rolled over from 2021-22) have been retained from 2021-22, but will be fit within the context of the above workplan for 2022-23.

8 DISCUSSION

The main area of work during 2021-22 was the feature for applying the catch-conditioned method for estimating fishing mortality. The method offers the most efficient means to reduce model complexity characteristic of the catch-errors method, with examples demonstrating up to an 83% reduction in the number of independent variables. While this feature was largely developed and implemented since early 2020, completing its development in preparation for use in a production stock assessment was a high priority in 2021-22. The skipjack tuna stock assessment for 2022 is the first such implementation of the method using MULTIFAN-CL for presentation to the WCPFC Scientific Committee (Castillo-Jordan et al. 2022). Completing its development entailed a range of enhancements and corrections for: ensuring robust minimisations; improving the regression for the fishing mortality:observed effort relationship; extending its operation into the post-minimisation procedures, including fishing impact analyses and deterministic projections; and, refining the reports generated from the method. It appears to be stable, and with the most important components needed for a production assessment operating satisfactorily. Aspects of it still require development, and are tasks identified for 2022-23, with that having highest priority being the implementation for stochastic projections.

The new feature for the CPUE likelihood was a high priority for allowing the comprehensive implementation of the catch-conditioned method in a production stock assessment; it being a separate and discrete term for fitting relative abundance indices. This was completed, with a task being identified for 2022-23 to further explore concentrated forms of the likelihood that account for index-specific precision within the regression.

The new feature for long-term tag loss currently has direct utility for using MULTIFAN-CL in a spatial modelling simulation analysis, where the operating model assumes a constant rate. Completing this feature has addressed one of the last two remaining recommendations from the independent review of MULTIFAN-CL (Ianelli et al. 2012).

A significant enhancement was to add new variables to the variance and correlation reports, including the WCPFC stock depletion reference points. This has direct relevance for the skipjack 2022 assessment, but also in general for examining parameter uncertainty and the correlations of depletion variables with respect to other dependent variables.

Considerably more resources were directed towards training and user support in 2021-22 compared to previous years, with a total of 21 on-line sessions having been provided to OFP stock assessment scientists. These included an in-progress review and support for the catch-conditioned model development for the skipjack 2022 stock assessment.

A very significant milestone in 2021-22, and indeed in the history of the project, was the retirement of the lead developer, and original architect of MULTIFAN-CL, Dr David Fournier. This is a substantial loss to the project given his creative force in developing innovative applied mathematical and statistical methods that he has implemented in MULTIFAN-CL since its inception nearly 30 years ago. In the 6 months since his retirement, efforts have been largely focussed on completing those tasks outstanding on the final new development that Dr Fournier implemented - the catch-conditioned method.

The future work plan for the project consequently reflects this change to the MULTIFAN-CL team, with no substantial innovative developments proposed. Rather, the focus is on consolidating those aspects that are incomplete for recent new features, enhancements of existing features, and documentation. In the past 3-4 years, new developments have progressed at a rapid pace, and therefore, the User's Guide for MULTIFAN-CL requires substantial updates.

The limited improvements to the support structures of the MULTIFAN-CL project made during 2020-21, and these related to adapting the Viewer to aspects of the reports linked to the catch-conditioned method and the CPUE likelihood, and nominal improvements to R4MFCL. Lower priority was assigned to this area of the project because of the importance of preparing the catch-conditioned and CPUE features for production assessments. A consequence of this, as has also occurred in previous years, is that: the User's Guide lacks documentation for many of the new features; versions since (and including) 2.0.7.0 have not yet been posted on the website; and, few updates to R4MFCL have been made. Also, testing of some of the new features (e.g., new movement parameterization) is incomplete, which makes their utility relative to existing methods unclear.

The proposed workplan presented to the 2022 pre-assessment workshop (Hamer 2022), and listed in section 7, seeks to address this situation, and will be tractable now that no substantial new developments are proposed.

9 REFERENCES

Bull, B., Francis, R.I.C.C., Dunn, A., McKenzie, A., Gilbert, D.J., Smith, M.H., Bian, R., and Fu, D. 2012. CASAL (C++ algorithmic stock assessment laboratory): CASAL User Manual v2.30-2012/03/21. NIWA Technical Report 135. 280 p.

Castillo-Jordan, C., et al. 2022. Stock assessment of skipjack tuna in the western and central Pacific Ocean - 2022. WCPFC-SC18-2022/SA-WP-01, Electronic Meeting, 10–18 August 2022.

Davies, N., Fournier, D., Bouyé, F., and Hampton, J. 2020. Developments in the MULTIFAN-CL software 2019-20. WCPFC-SC16-2020/SA-IP-01

Davies, N., Fournier, D., Bouyé, F., and Hampton, J. 2021. Developments in the MULTIFAN-CL software 2020-21. WCPFC-SC17-2021/SA-IP-01

Dunn, A., Hoyle, S., and Datta, S. 2020. Development of spatially explicit operating models for yellowfin tuna populations in the Indian Ocean. IOTC-2020-WPT22(AS)-19.

Fournier, D.A., Hampton, J., and Sibert, J.R. 1998. MULTIFAN-CL: a length-based, age-structured model for fisheries stock assessment, with application to South Pacific albacore, *Thunnus alalunga*. *Can. J. Fish. Aquat. Sci.* **55**:2105-2116

Hamer, P. 2022. Report from the SPC pre-assessment Workshop – March 2022. WCPFC-SC18-2022/SA-IP-02, Online, 10–18 August 2022

Ianelli, J., Maunder, M., and Punt, A. 2012. Independent review of 2011 WCPO bigeye tuna assessment. WCPFC-SC8-SA-WP-01

Kleiber, P., Fournier, D., Hampton, J., Davies, N., Bouyé, F., and Hoyle, S. 2018. MULTIFAN-CL User's Guide. http://www.multifan-cl.org/

10 TABLES

Table 1. New features added to MULTIFAN-CL with respect to their state of completion as of July 2022.

Peer review recommendations		
Task	Description	Status of completion
b. Non-uniform size bins	Allow the length bins to be of different widths. One might, for example, want many narrow length bins for the smaller lengths, but fewer but wider length bins for the larger lengths.	Development 0%
c. Long-term tag loss	Allow for long-term and initial tag-loss. Currently initial tag-loss is implemented by reducing the number of animals tagged when inputting data to the model and no account can be taken of long-term tag-loss.	Complete 100%; Testing 100%
d. Tags inform movement	Include an option which allows the tagging data to inform movement only rather than movement and mortality.	Development 100%; Testing 90%
Other developments		
Task	Description	Status of completion
Catch-conditioned	Catch-conditioned fishing mortality estimation with minimisation stability.	Development 95%; Testing 95%
	F0eval with compensated predicted recruitments in respect of the BH-SRR prediction on SBf=0	Development 100%; Testing 100%
	Testing of catch-conditioned projections, both catch- and effort-conditioned fisheries	Development 90%; Testing 90%
CPUE likelihood	Survey fisheries CPUE indices likelihood, options for grouped (stationary) catchability and concentrated formulation.	Development 100%; Testing 100%
Initial population	Confirm operation of age_flags(95) in relation to the option for initial population as a function of natural mortality	Development 100%; Testing 100%
Dependent variables	Add to the *.var and *.cor reports for the fished:unfished reference point dependent variables	Development 100%; Testing 100%
Recruitment random effects	Estimate the autocorrelation and variance of recruitment deviates as random effects rather than as independent parameters. Attempt to extend this functionality also to effort deviates.	Development 60%; Feasibility has been assessed
Constrain deviates	Apply constraints on the recruitment deviates such that the $\bar{x}$ = 0. Apply also to the effort deviates for fisheries with missing effort data for the complete time series.	Development 10%; Testing 0%
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	Development 0%; Testing 0%
EM fit only projection pars	For the "assessment" estimation model (EM) embedded in a management procedure, only fit parameters relating to the "new" data provided for the projection time periods, i.e. for effort devs, catchability devs, recruitment devs, while holding all other parameters fixed at the initial values.	Development 80%; Testing 0%
OM size comps	Generate a report of the OM size compositions for projection period without error at the end	Development 0%; Testing 0%

	of the projection period as required for deriving economics-based indicators.	
Turing test	Ensure the quality of pseudo-observations to be made more realistic by: • Including the sel_dev_coffs and eff_devs estimates in applying process error in projection size compositions and effort • Including over-dispersion error in tagging data	Development 0%; Testing 0%
Stochastic projection functionality	• Implement process error in future recruitments with application of the derived autocorrelation coefficient in historical recruitment estimates • Fix a bug in generating inputs for stochasticity in $N_{terminal}$ (more stable method is to use terminal year less 5 as the period for obtaining variance) and eff_devs	Development 0%; Testing 0%

Table 2. Proposed workplan for MULTIFAN-CL in 2022-23 and subsequent years, and those for which implementation and testing is to be completed.

Peer review recommendations		
Task	Description	Implementation
b. Non-uniform size bins	Allow the length bins to be of different widths. One might, for example, want many narrow length bins for the smaller lengths, but fewer but wider length bins for the larger lengths.	2022-23
d. Tags inform movement	Testing for the case of multi-species/stocks/sexes.	2022-23; Development 100%; Testing 90%
Other developments		
Task	Description	Implementation
Catch-conditioned method	Add the fml_effort_rltnshp penalty term to the members of the class used for creating test_plot_output.	2022-23
	Allow the fml_effort_rltnshp penalty calculation to be conditional on a fish_flags(fi) that facilitates it being disabled, or specified, only for particular fisheries.	2022-23
	Testing of catch-conditioned projections – implementation for stochastic projections.	2022-23
	Review the operation of existing control phase routines undertaken in Phase 1 in respect of their suitability for a catch-conditioned model, and draft a new control phase specific to the catch-conditioned model as needed.	2022-23
	Review the fish_flags(92 and 99) sanity check that issues an error message when the penalty weights are different within a group	2022-23
	Fishing mortality estimation for multi-species	2022-23
CPUE likelihood	Review the formulation of the concentrated log-normal likelihood formulation as used in CASAL for its suiability for implementation in MULTIFAN-CL.	2022-23
	Allow fishery-specific penalties in the CPUE likelihood for grouped survey fisheries	2022-23
Hessian operations	Debug the running of the Hessian calculations that interact with the test_plot_output report; replaces 0s in for all of the length comp likelihood entries	2022-23
User's Guide	Documentation of the catch-conditioned method and flags	2022-23
Tasks rolled over from 2021-22		

Recruitment random effects	Report on the feasibility of its implementation in MULTIFAN-CL.	post-2022-23
Constrain deviates	Apply constraints on the recruitment deviates such that $\bar{x}$ = 0. Apply also to the effort deviates for fisheries with missing effort data for the complete time series.	post-2022-23
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	post-2022-23
Self-scaling multinomial with random effects	Complete draft paper for peer review.	post-2022-23
Recruitment correlates	Region-specific environmental recruitment correlates estimated within the orthogonal polynomial parameterisation for recruitments	post-2022-23
Movement correlates	Add a time-series structure (e.g. random walk, time blocks or using environmental correlates) to movement coefficients	post-2022-23
Recruitment deviate penalties	Allow for time-variant penalties on recruitment deviate estimates	post-2022-23
Tagging multi-sex	Account for instances of differences between size composition the tag releases and the sex-specific populations	post-2022-23
Region specific SRR	Allow that region-specific spawning biomass is responsible for recruitment within that region. This is consistent with the assumption that stocks may not be truly panmictic. This would estimate region-specific SRRs.	post-2022-23
Outstanding testing of existing features	- Reinstate the original estimation of length-based selectivities as activated by fish_flags(i,26)=3 that integrates the distribution of lengths-at-age in calculating the selectivity-at-age. (single species only) - Multi-sex model projections	Complete development post-2022-23 post-2022-23
Report comments	Add comment descriptions of the selectivity parameter configurations in the output .par and .rep reports	post-2022-23
EM fits only projection parameters	For the “assessment” estimation model (EM) embedded in a management procedure, only fit parameters relating to the “new” data provided for the projection time periods, i.e. for effort devs, catchability devs, recruitment devs, while holding all other parameters fixed at the initial values.	post-2022-23
OM size comps	Generate a report of the OM size compositions for projection period without error at the end of the projection period as required for deriving economics-based indicators.	post-2022-23
Stochastic projection functionality	- implement process error in future recruitments with application of the derived autocorrelation coefficient in historical recruitment estimates - consolidate the generation of stochasticity in effort_dev_coffs	post-2022-23
Turing test	Ensure the quality of pseudo-observations to be made more realistic by: - Including the sel_dev_coffs and effort_dev_coffs estimates in applying process error in projection size compositions and effort - Including over-dispersion error in tagging data	post-2022-23

11 FIGURES

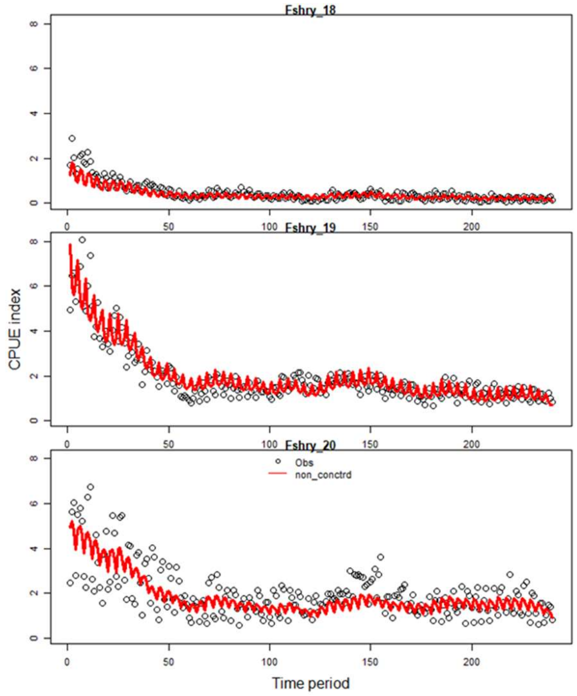


Figure 1. CPUE indices fitted with grouped catchabilities for the model employing the catch_cond feature (ccond).

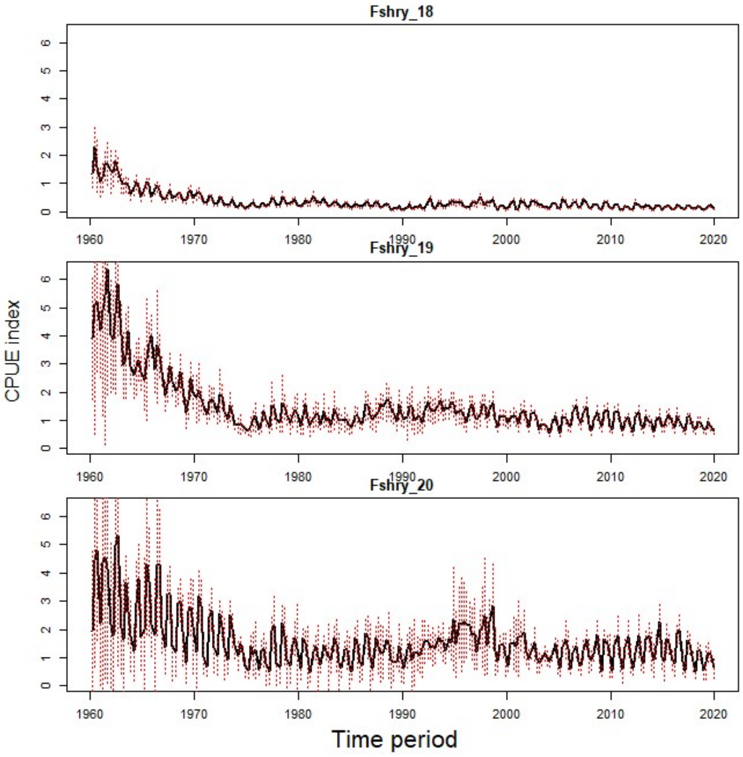


Figure 2. Observed CPUE indices (error bars are 2 standard deviations) for the simple 3-region model.

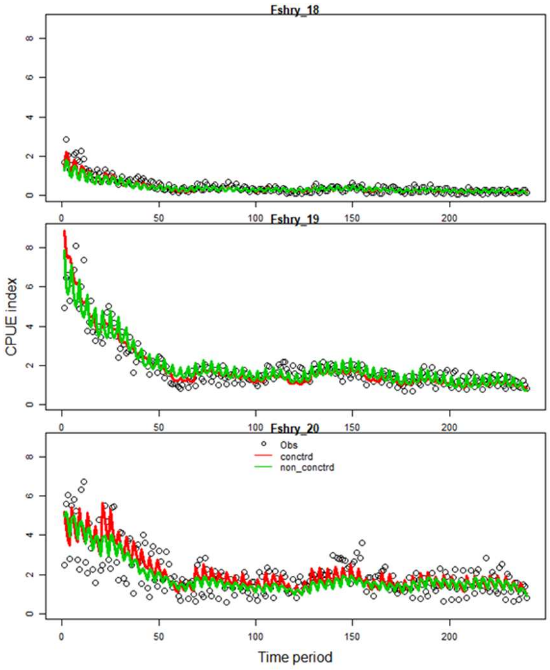


Figure 3. CPUE indices fitted with grouped survey fisheries having stationary catchability, with comparison among models fitted with (conctrd) or without (non_conctrd) the concentrated likelihood formulation.

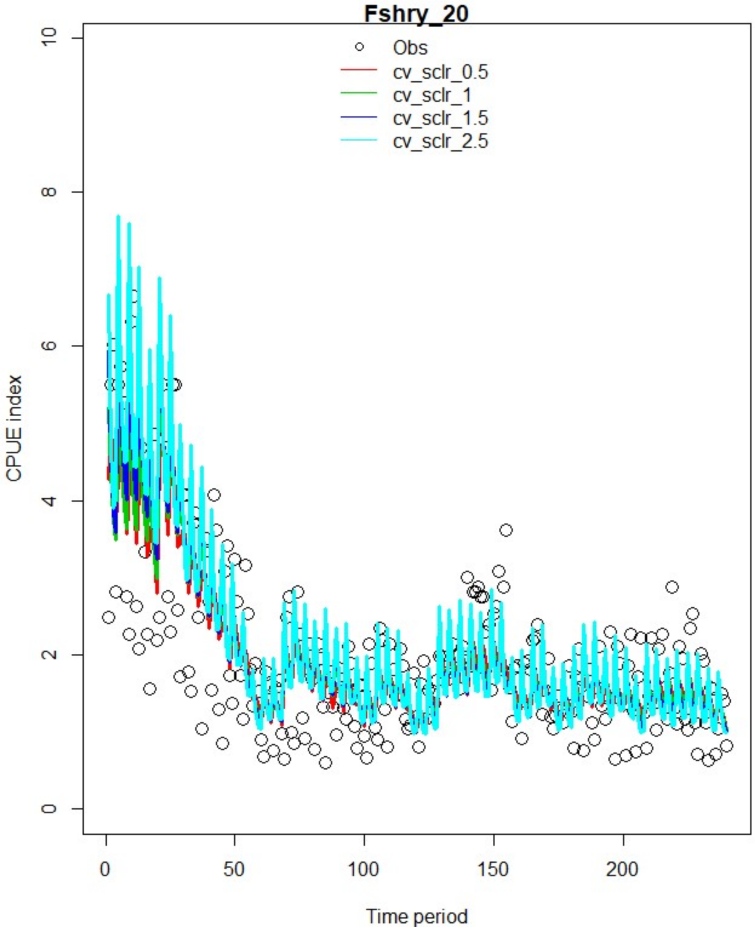


Figure 4. CPUE indices fitted with observation-specific precision scaled over a range 0.5 to 2.5, illustrating the effect upon the concentrated likelihood.

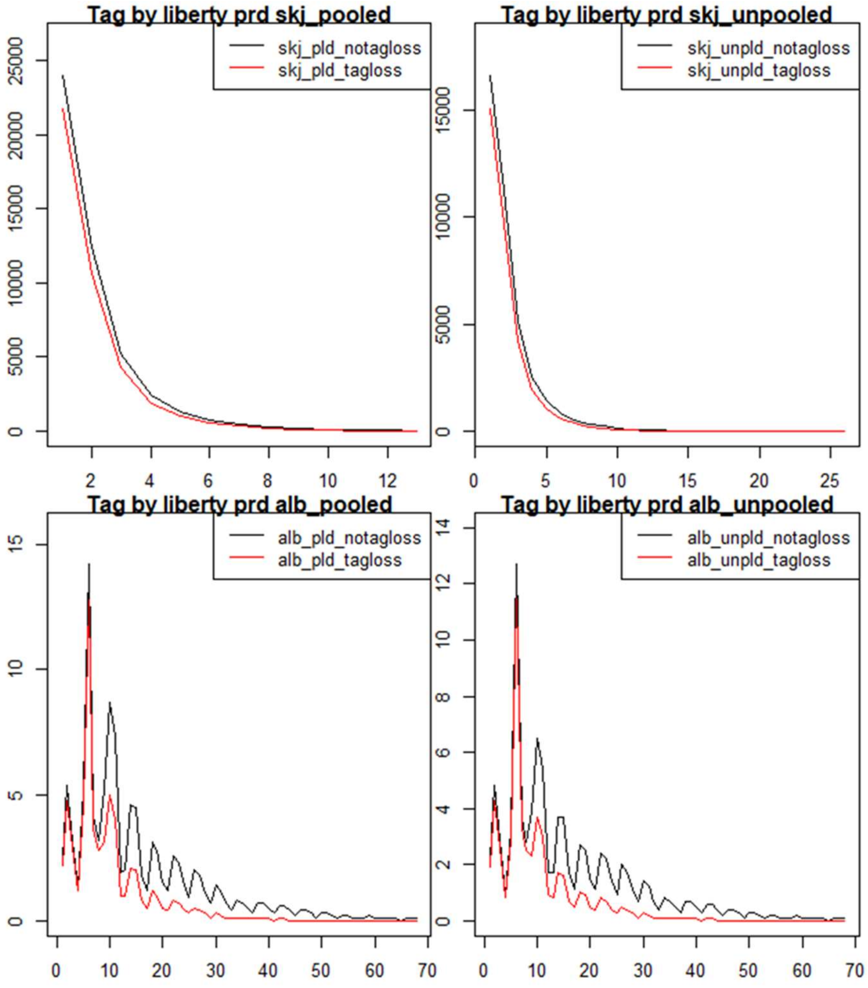


Figure 5. Comparisons of deterministic evaluations with options for long-term tag loss included (*_tagloss) and excluded (*_notagloss), for models employing the catch conditioned feature.

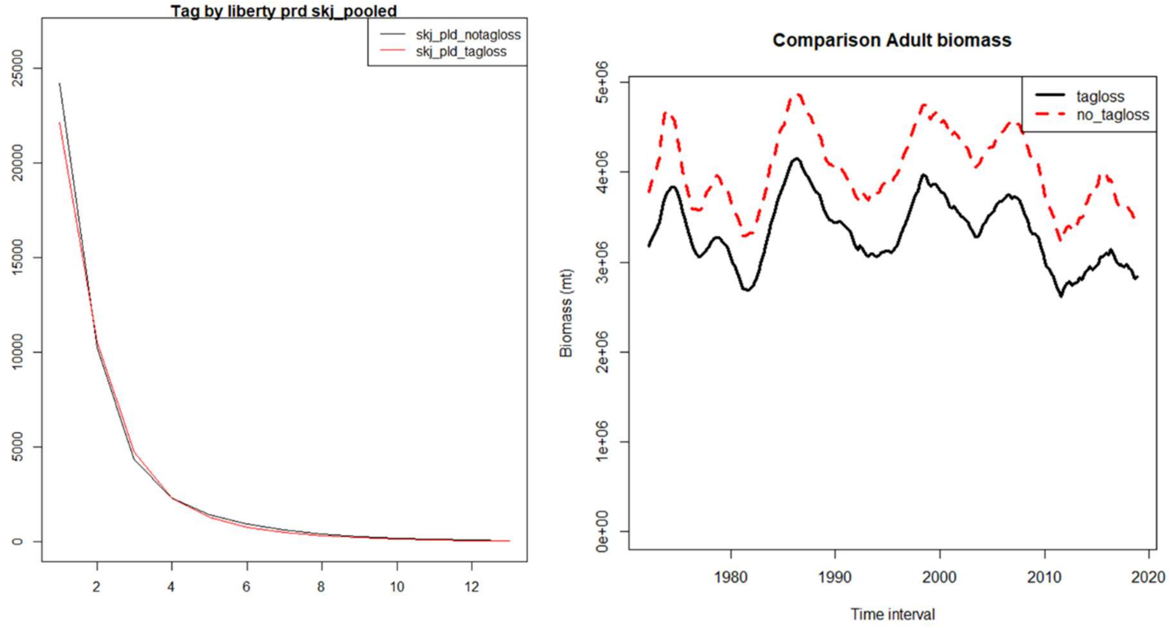
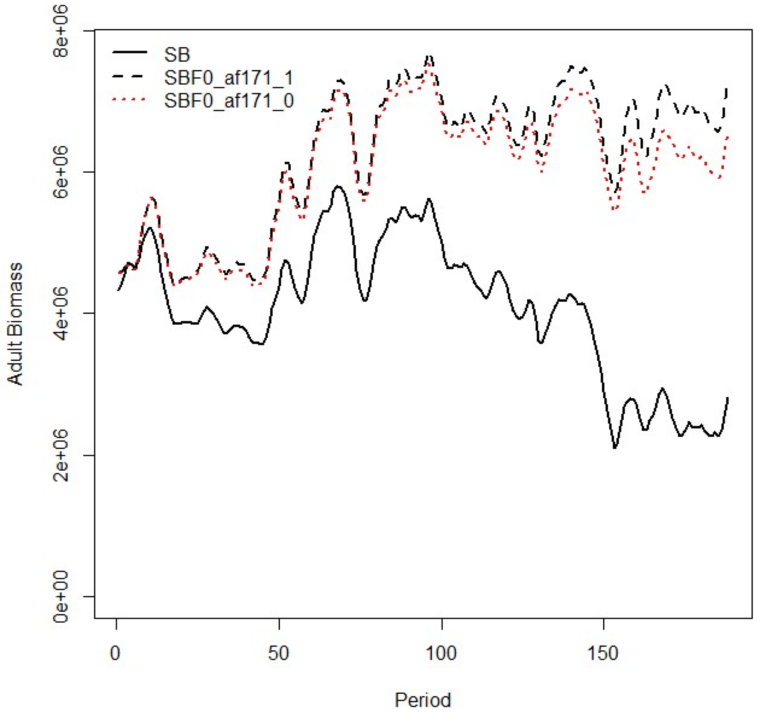


Figure 6. Comparisons of model solutions with (*_tagloss) or without (*_notagloss) long-term tag loss, for the SKJ2019 model employing the catch conditioned feature, showing the estimated tag returns by period at liberty (left panel), and estimated adult biomass (right panel).



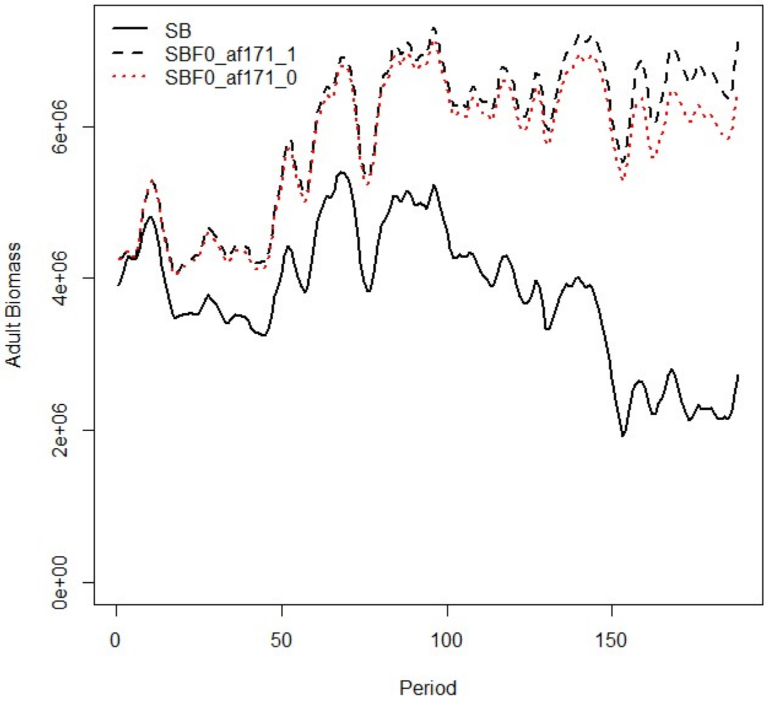


Figure 7. Pair-wise comparison of the unfished adult biomass (SBF0) for the two options: with, or without, recruitment compensation, (SBF0_af171_1 and SBF0_af171_0, respectively), and for the models having quarterly or annual BH-SRR predictions (top and bottom panels, respectively).

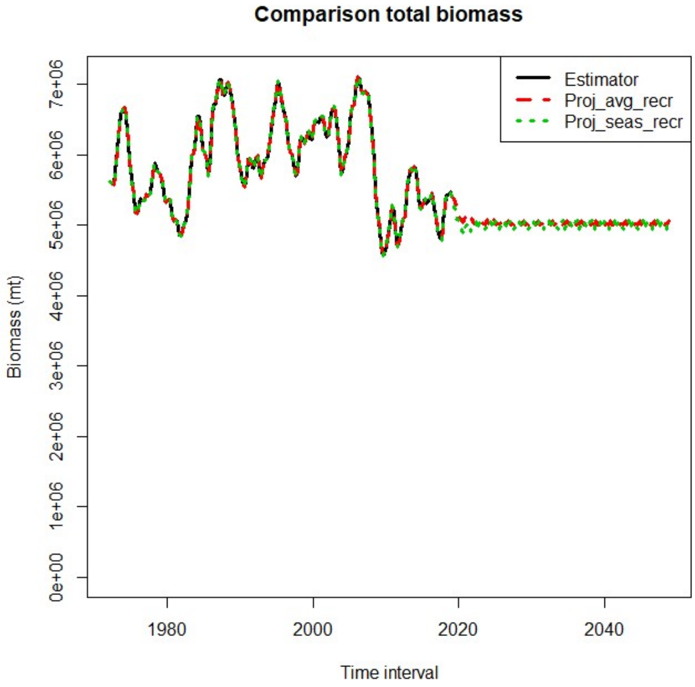


Figure 8. Comparison of the total biomass for the two projection model options (Proj_avg_recr, Proj_seas_recr), and with respect to the original estimation model (Estimator).

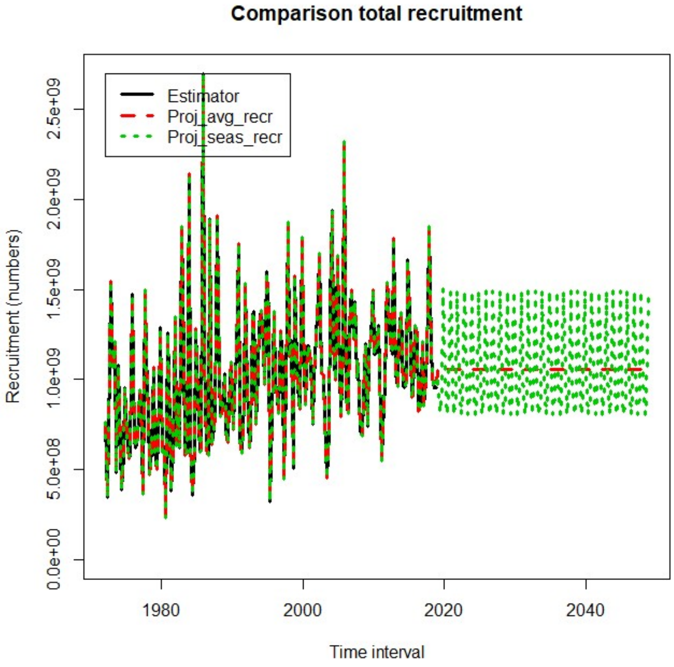


Figure 9. Comparison of the total recruitments for the two projection model options (Proj_avg_recr, Proj_seas_recr), and with respect to the original estimation model (Estimator).

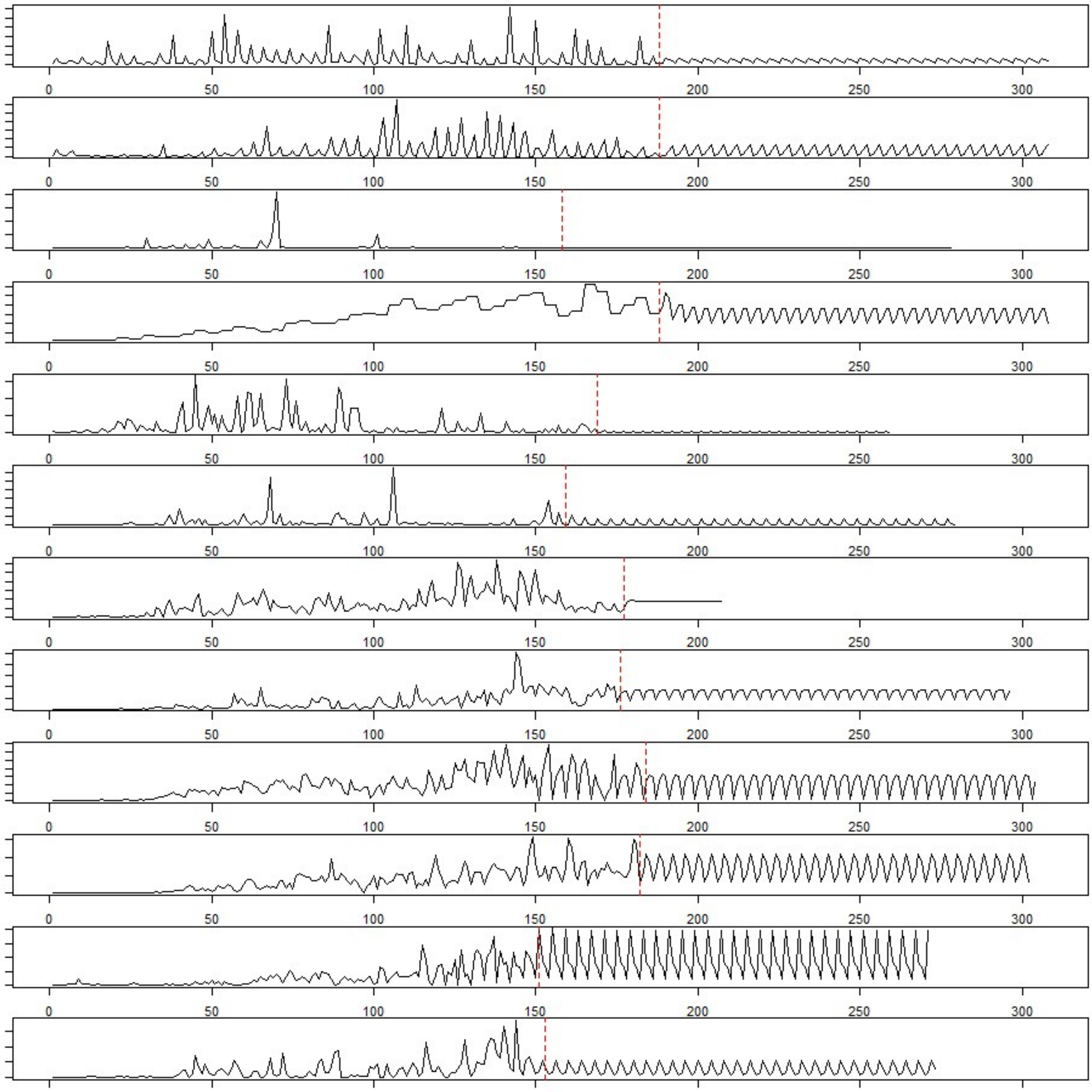


Figure 10. Catches of the projection model option, Proj_avg_recr, for the fisheries conditioned on effort (in descending order from top panel) for the projection time periods; with the vertical dashed red line indicating the end of the estimation time periods and the beginning of the projection time periods.

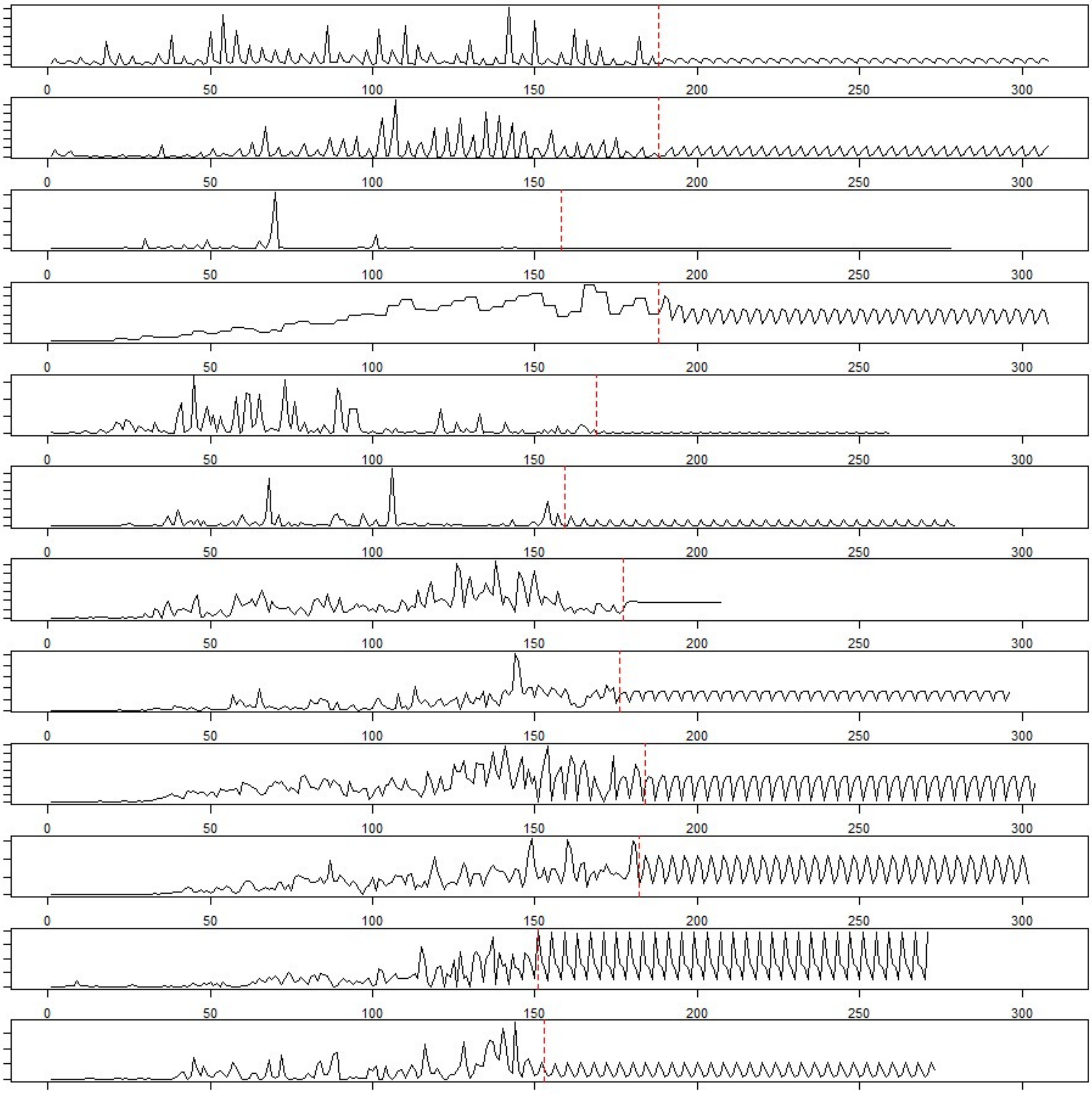


Figure 11. Catches of the projection model option, Proj_seas_recr, for the fisheries conditioned on effort (in descending order from top panel) for the projection time periods; with the vertical dashed red line indicating the end of the estimation time periods and the beginning of the projection time periods.

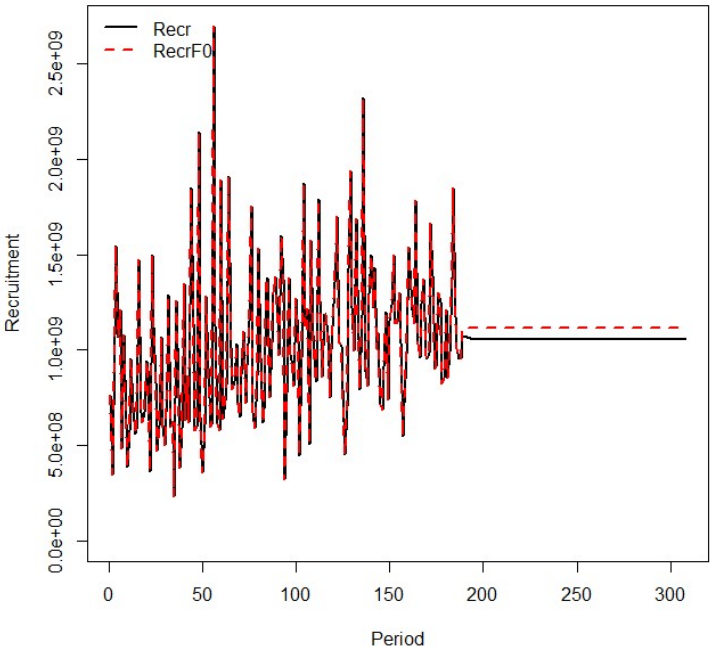


Figure 12. Comparison of the total recruitments for the projection model options Proj_avg_recr, under the normal fishing (Recr) and zero-fishing model (RecrF0) evaluations.

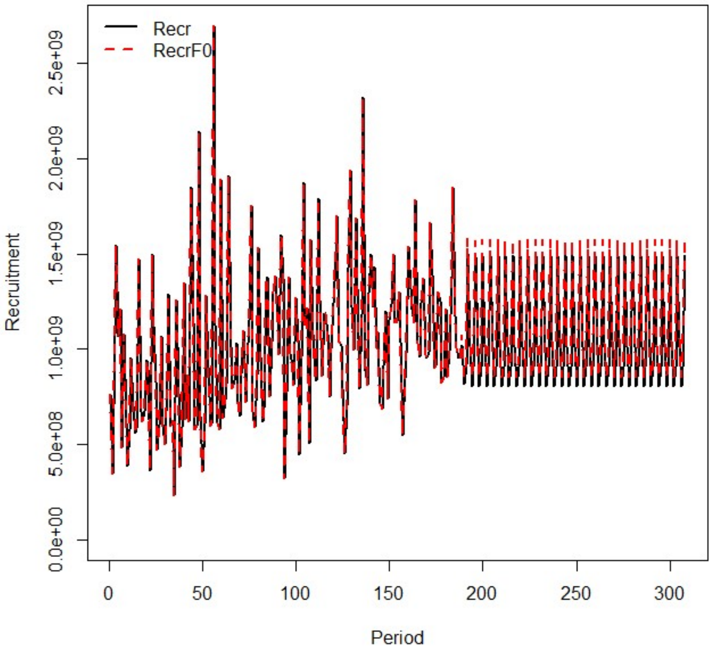


Figure 13. Comparison of the total recruitments for the projection model options Proj_seas_recr, under the normal fishing (Recr) and zero-fishing model (RecrF0) evaluations.

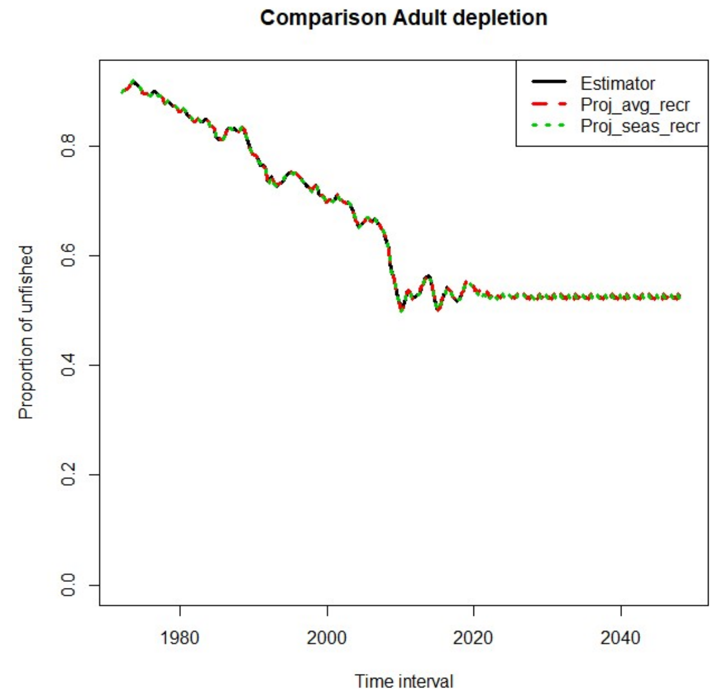


Figure 14. Comparison of the depletion of adult biomass relative to the unfished state for the projection model options Proj_avg_recr and Proj_seas_recr.

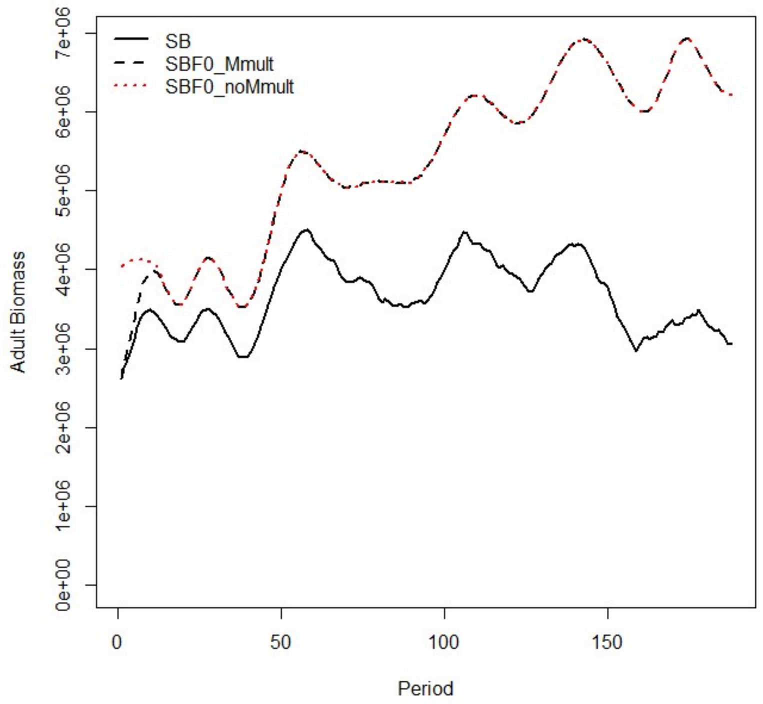


Figure 15. Comparison of adult biomass for the normal fishing model evaluation (SB) that applies the initial population condition as a function of natural mortality with a multiplier, and two options for the zero-fishing model evaluation: including the multiplier (SBF0_Mmult); and, excluding the multiplier (SBF0_noMmult).

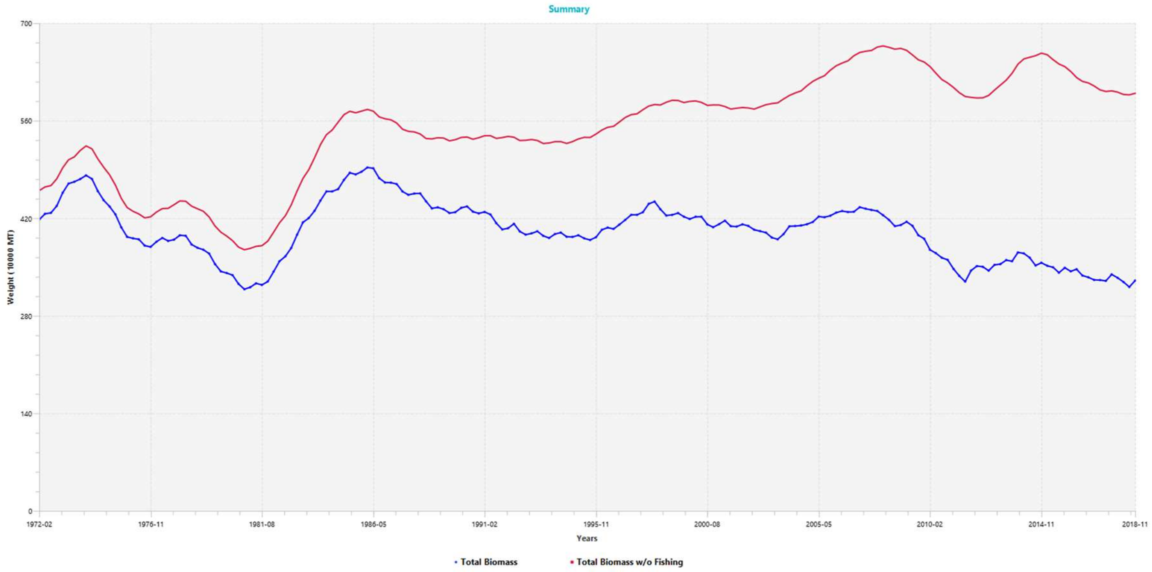


Figure 16. Comparison of total biomass for the normal fishing model evaluation (blue line) and the fishing mortality=0 model evaluation (red line) for the catch-conditioned model following the correction, illustrating the clearly different initial population conditions, and for all subsequent time periods.

12 ANNEX 1

Modified source code files (67) during 2021-22.

agelength.cpp
all.hpp
avcatfit.cpp
biodynam.cpp
callpen.cpp
do_all_for_empirical_autocorrelated_bh.cpp
eq2.cpp
equilib.cpp
ests.cpp
get_orth_poly_info.cpp
get_orth_weights.cpp
getinp2.cpp
goodpen.cpp
gradrout.cpp
grpcatch.cpp
kludged_equil_surv.cpp
len_dm_nore.cpp
lesmatrix.cpp
minvmult.cpp
mspcode.cpp
mymeld
mysort.cpp
newgradc.cpp
newgradc_noeff.cpp
newl2.cpp
newl5.cpp
newl9.cpp
newm_io3.cpp
newmaturity.cpp
newmau5a.cpp
newmaux5.cpp
newmult.cpp
newmult.hpp
newrshimp.cpp
newrshimp_experiment.cpp
nmsort.cpp
nnewlan.cpp
nrcatch3.cpp
nrcatch4.cpp
optalloc.cpp
plot.cpp
plotN0.cpp
plotstuff.cpp
rationalize_movement.cpp
readtag.cpp
real_ddnrc3.cpp
recinpop_orth.cpp
rnaux2.cpp
rshort1.cpp
rshort3.cpp
safold64.mak
setcomm.cpp
setcomm1.cpp
svd.h
tag3.cpp
testeq1.cpp
testeq2.cpp
testnewl3.cpp
testsvdinv_noeff.cpp
variable.hpp
version.h
wght_dm_nore.cpp
win2opt_nick.mak
xequilib.cpp
xwww2.cpp
yield_bh.cpp
yieldbhp.cpp

New source code files added (5) during 2021-22.

ils_qr.cpp
ils_qr.hpp
lapack_inv.cpp
natmort.cpp
print_survey.cpp