SCIENTIFIC COMMITTEE
TWENTIETH REGULAR SESSION

Manila, Philippines
14 – 21 August 2024

Developments in the MULTIFAN-CL Software 2023-24

WCPFC-SC20-2024/SA-IP-02
01 August 2024

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

[1] Te Takina Ltd, Whangarei, New Zealand

[2] Oceanic Fisheries Programme, The Pacific Community, Nouméa, New Caledonia

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 2023 to July 2024, and updates the report of Davies et al. (2023).

The current production version 2.2.5.0 has been benchmark-tested (Nov. 2023) and is available for wide distribution. The development version 2.2.7.0 has been tested for the specific developments to existing features, has undergone frequent abbreviated benchmark tests, and has been employed within the OFP, Pacific Community, for undertaking the 2024 tuna and billfish stock assessments.

During 2023-24, no significant mathematical innovations were implemented into the MULTIFAN-CL source code, and this was not intended in the workplan. Rather, the aim was for the recent features added since 2019 (e.g., catch-conditioned method for estimating fishing mortality) to be consolidated, enhanced, and extended for their implementation in producing stock assessment models, and undertaking population projection analyses.

An important enhancement of an existing feature that was essential for providing stock assessment advice was that to extend the catch-conditioned feature to perform stochastic projections; having future fisheries conditioned in respect of either catch or effort. Briefly, this included:

- Allowing alternatives for the terminal catchabilities (either implicit or predicted) to be assumed for the projection periods, that are applied to effort-conditioned projection fisheries to derive projection fishing mortalities;
- Developing a method using the terminal catchabilities for generating pseudo-observations of effort from catch-conditioned projection fisheries; and,
- Generating simulation CPUE pseudo-observations.

These three developments substantially improved the catch-conditioned operating model (OM) for its application in management strategy evaluations (MSE) and target reference point (TRP) simulation studies; which are essential for providing stock assessment advice. In particular, the feature allowing detailed evaluation of the assumed terminal catchabilities in respect of those in the recent periods, say the past 3 to 5 years, is very useful.

The second-most important area of development during 2023-24 was a thorough consolidation of the CPUE likelihood, both in its formulation and implementation in MULTIFAN-CL. This development was significant because of the important role of relative abundance index data in the WCPO tuna stock assessment models. Key areas were:

- re-formulation of the non-concentrated likelihood to include the constant term;
- re-formulation of the concentrated CPUE likelihood to express time-variant precision in the form of a normalised deviate on the assumed error, $\boldsymbol{\sigma}$;
- enabling the stationary catchability assumption among grouped fisheries to be made, but while allowing for differences in the relative precision among the fisheries from each region.

Other important enhancements made offer assistance to the stock assessment analyst when undertaking model development in: reviewing parameter configurations; imposing prior assumptions on the estimated regional distribution of recruitments; and, obtaining estimates of parameter uncertainty for the key quantities of management interest. These aim to make the use of MULTIFAN-CL easier for the analyst.

A number of changes to the development version have been made since version 2.2.5.0 (April 2023), and most of these were accompanied by abbreviated benchmark testing of the implications on the function evaluation and dependent variable estimates. These are described in sections 2.6.2 to 2.6.6. While not fully benchmark tested, the detailed testing of the effects specific to each development, and the abbreviated testing, confirm no negative impacts on other existing features employed in the 2024 stock assessment models produced using the MULTIFAN-CL development version. These tests also provide reference points when future stock assessments are undertaken using the current, or future, versions of MULTIFAN-CL. Typically, the starting point of a stock assessment is to replicate the previous assessment model solution using the most recent MULTIFAN-CL version. Interpreting the differences in dependent variables and management quantities with respect to the previous solution, is well assisted by making reference to the tests performed in this, and previous, update reports.

An operational success during 2023-24 was to restore the production of the macOS executable. The Mini Mac PC used for this compilation had been made unavailable to the MULTIFAN-CL project since 2022 due to cybersecurity concerns in the Pacific Community computing network. During 2023-24, its access was reinstated, allowing compilations and testing of new versions for the macOS platform to resume.

Substantially fewer project resources (due to Dr Fournier's retirement in Dec. 2021), and lower assigned priority to maintaining the project's support structures (because of the importance of preparing features for production assessments), means this area continues to be neglected. A particular area of concern is the lack of documentation in the Manual for implementing the catch-conditioned method in stock assessments and simulation projections. This lack of a reference document for this feature presents a substantial obstacle to analysts, and a potential source of errors. It is therefore a high priority for 2024-25. Generally, strategic planning is urgently needed to address this shortage of resources in the project for the medium term. It also worth noting that Nick Davies, the current (and only) developer of MFCL, is also contributing to the WCPFC Project 123 on exploring the options for successor software to MFCL. Nick has the most in depth knowledge of MFCL, and his expertise is essential to supporting project 123, but this does add extra burden on his time.

The focus of the 2024-25 workplan, is a continuation of that for the past 3 years, i.e., to: consolidate recent new features; enhance existing features; improve processes and reporting; make corrections; and update User support documentation. No substantial new developments to the project are proposed.

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., Day et al. 2023) 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 2023 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.8.7), and the current unreleased development version, and updates the report for the previous period, 2022-23 (Davies et al. 2023).

2 DEVELOPMENT OVERVIEW

2.1 Team

The senior developer of MULTIFAN-CL until December 2021 was Dr David Fournier, of Otter Research Ltd, (Canada), who has since retired. Development and testing are now undertaken by Mr Nick Davies. Other tasks include testing and debugging (ND, John Hampton, and Fabrice Bouyé (SPC)); documentation (ND); and planning and coordination (ND, Paul Hamer and JH). Related support project software is developed or managed by FB (MULTIFAN-CL Viewer, Condor, GitHub, Jenkins), Arni Magnusson, ND, and Robert Scott (R4MFCL, FLR4MFCL).

2.2 Calendar

In the absence of more than one developer, developer's workshops are no longer held, and the calendar year is less structured. However, broad periods may be identified with those for which more support is given to OFP stock assessment modelling.

August – November: Consolidating recent developments, benchmark testing, developments required for risk analyses

December – February: Code development, and testing

March – April: Training, stock assessment support, code development, and testing

May – July: Code development, and testing

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 that 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 provides the platform for undertaking the routine compilation administered by Jenkins (see section 2.4.4). These automated builds were maintained throughout 2023-24, apart from a problem encountered with accessing the Mac Mini since October 2022 (see section 2.4.4).

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

In August 2023 the Linux version used for compilations was upgraded to Ubuntu 20.04.2, with the gcc compiler version 9.4.0. No changes were required to the source code for the new compiler.

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. Compilations are done on the MULTIFAN-CL Testing PC (Mac Mini) that has the macOS Mojave installed ("macOS 10.14.6 Mojave").

In October 2022, a major setback occurred in respect of the macOS compilations. The MULTIFAN-CL Testing PC (Mac Mini) was removed from the Pacific Community computing network for security reasons. During 2023-24, a secure arrangement suitable for the network's standards to reinstate this PC was made, such that the Mac Mini is now again accessible to the MULTIFAN-CL project. The opportunity was taken to: complete updates of the macOS from Mojave (2018) to Sonoma (2023); apply all recent security updates; update Xcode, Xcode command line tools, and Homebrew distribution and packages, to the most recent release versions. Subsequently on 20 April 2024, a successful compilation for the macOS executable was produced of the current release version 2.2.5.0, with a successful test using a tuna model example with respect to the corresponding Linux executable.

The Mac Mini 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 (no machine has been purchased)
- 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 2024

2.4.5 Visual Studio 2019 Windows compilation

Compiling the Windows executable is done using Visual Studio 2019, (VS2019) and all compilations were successfully completed during 2023-24 on the developer’s workstation. No issues were encountered with the compilations as a result of updates to the VS2019 compiler.

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, and a new compilation directory created. Other points where a new development version number is assigned is immediately following changes that may impact upon a minimised model solution, or alterations to the format of output files. These are then added to the development branch following preliminary testing, and tagged with the new version number. During 2023-24 a new development version was created following the benchmark testing of version 2.2.5.0.

A number of changes to the development version have been made since version 2.2.5.0 (April 2023), and most of these were accompanied by abbreviated benchmark testing of the implications on the function evaluation and dependent variable estimates. These are described in sections 2.6.2 to 2.6.6. While not fully benchmark tested, the detailed testing of the effects specific to each development, and the abbreviated testing, confirm no negative impacts on other existing features employed in the 2024 stock assessments models produced using the development version.

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 2023-24, with the current development version being maintained in the "ongoing-dev" branch (version 2.2.7.0).

Between 2 Aug. 2023 and 22 Jul. 2023, a total of 42 source code commits were made to the master and development branches (49 source files have been modified, and 1 new source file produced, see section 12), including a merge to the master branch on 13 Nov. 2023 for the distribution of version 2.2.5.0 to the Pacific Community. The current version in the "ongoing-dev" development branch is version 2.2.7.0.

2.5 Developer's workshops

In the absence more than one developer, no workshops were held during 2023-24. Since January 2022, Mr. Nick Davies has continued alone with the consolidation, enhancement and corrections to the existing features.

2.6 Benchmark testing during 2023-24

The benchmark testing framework is described in section 2.9.2, and one set of benchmark tests, and numerous abbreviated tests, were undertaken in 2023-24. When relatively few changes have been made, the abbreviated tests explore their specific effects on example solutions obtained with the previous version. This is a precursor for undertaking the comprehensive benchmark test at a later date. A brief description of the tests, and the features tested, is provided in this section.

2.6.1 Version 2.2.5.0

In October-November 2023, comprehensive benchmark testing was done between the MULTIFAN-CL development version, and the benchmark release version 2.1.0.0 previously tested in April 2023. A complete set of tests were undertaken, to examine the cumulative effect of a number of enhancements and corrections made to the development version source code since version 2.1.0.0., as described in section 5 and by Davies et al. (2023), primarily including:

- **Catch-conditioned method –** allow for grouped survey fisheries that share the same stationary catchability, to have the capability for non-shared penalty weights; the concentrated form of the negative log-likelihood was implemented where time-variant index-specific precision is available, and this formulation also included the capability for non-shared penalty weights, the non-concentrated likelihood formulation was revised to include the constant term of the variance summation.
- **von Bertalanffy and Richards growth variance calculations** – the method for calculating the variance of mean length-at-age derived from the von Bertalanffy and Richards growth functions was corrected.
- **Dirichlet-Multinomial size composition likelihoods in test_plot_output components report** – corrections were made that prevents modification of the Dirichlet-Multinomial no-random-effects likelihood (DM-like) component in the report.
- **Simulated CPUE pseudo-observations** – a feature was developed that generates simulation CPUE pseudo-observations for survey index fisheries of a catch-conditioned model for both the estimation and projection time periods.
- **Abbreviated derivatives calculation of dependent variables** – the parest_flags(37)=1 was assigned that "leap-frogs" through parts of the default list of dependent variables (dep_vars) to only calculate the derivatives for an abbreviated, essential set required for reports.

- **Various enhancements** – reversed the action of parest_flags(387); implemented a new minimisation scaling for the Lorenzen natural mortality parameter and allowed for starting the minimisation from initial assumed values; implemented a new lower bound on the fml_implicit_regression_parameters.

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 2.1.0.0. The range of testing data sets was the same as for the previous benchmark test, comprising: 7 single-species sets; a multi-species set; a multi-sex set; a single-species deterministic projection set; and, a single-species stochastic projection set.

Issues identified during testing

Code changes made in the development version were identified that caused notable differences in the operation or results obtained versus those of **vsn2.1.0.0**.

- Different flag settings were required to ensure backward compatibility with **vsn2.1.0.0** because the operations of the following flags were altered in the development version: parest_flags(387) and parest_flags(34) (a setting of 1 is required).
  Therefore, the settings of these flags must match that of the previous benchmark tests:
    - parest_flags(387) was set = 0, therefore, devvsn16 requires a setting of 1
    - parest_flags(34) was set = 0, therefore, devvsn16 requires a setting of 1
- Both **vsn2.1.0.0** and the development version require the input values of the maturity-at-length ogive in the .ini or .par files to be greater than 0 and less than 1
- The non-concentrated CPUE survey fishery CPUE likelihood formulation in the development version includes the added constant term of the variance summation. This can substantially change the total magnitude of the likelihood, and may affect the solution derived from an integrated likelihood minimisation.
- The code implementation made to the variable **global_vars** for the von Bertalanffy stdev(mean length-at-age) were made only for the single species/sex cases. This caused errors for the multi-species/sex cases in respect of species > 1. The code implementation was therefore replicated in the multi-species/sex class **pmsd**, that rectified the error.
- A correction was made in the development version that ensured consistency in the plot.rep report values of the von Bertalanffy stdev(mean length-at-age) with those of the global variables. This was an error in vsn2.1.0.0.

A single correction was required to the development version during the testing

Results

Single evaluation tests for single-species, multi-species, multi-sex data, and deterministic single species projection, with or without gradient calculations and a minimisation step – produced identical model quantities among versions; except for the SKJ2022 example due to the change to the CPUE likelihood.

Doitall fits of single species data – produced identical model quantities among versions; except for the SKJ2022 example due to the change to the CPUE likelihood.

Doitall fits of multi species data, deterministic and stochastic projections data – produced identical model quantities among versions.

Tests of the development version concluded that the results were consistent with respect to the benchmark version 2.1.0.0 as all existing features remain intact, therefore the development version was advanced to the new MULTIFAN-CL release version, **2.2.5.0**.

2.6.2 Abbreviated test – version 2.2.5.1

On 19 January 2024, an abbreviated test was undertaken, to examine the effect of two enhancements made to the development version source code described in section 5:

- Consolidation of the test_plot_output likelihood components report;
- Penalty on estimated regional distribution of orthogonal-polynomial recruitments;

A doitall fit of the BET2023 single species example (catch-conditioned), indicated identical results relative to the benchmark version 2.2.5.0.

2.6.3 Abbreviated test – version 2.2.5.1

On 20 February 2024, an abbreviated test was undertaken, to examine the effect of two enhancements made to the development version source code described in section 5:

- Terminal predicted catchabilities applied in simulation projections of effort-conditioned fisheries
- Comprehensive report of the estimated independent variables

Doitall fits of two single species examples (catch-conditioned) were undertaken (BET2023, SKJ2022). The recent developments in vsn.2.2.5.1 have made as slight difference to the gradient calculation due to the changes to the dvar_variables related to the fml_effort_rltnshp regression. This made almost negligible differences to the test using the SKJ2022 example, and only slight differences to the BET2023 example (some dependent variables altered by <2%). The general degree of the effect of these changes in producing different minimisation paths will be case-specific, mostly depending upon how well-determined is the solution. For the BET2023 example, this produced visible differences in the dependent variables.

2.6.4 Abbreviated test – version 2.2.5.1

On 20 April 2024, an abbreviated test was undertaken, to examine the effect of a small correction made to the development version source code described in section 5:

- Improve the calculation algorithm for the subset period specified for the BH-SRR regression

A doitall fit of the BET2023 single species example (catch-conditioned), indicated a small change in the total integrated log-likelihood value at the second decimal place, due entirely to the change in the BH-SRR regression value. This changed the minimisation solution, with very slight differences to the dependent variables.

2.6.5 Abbreviated test – version 2.2.6.0

On 17 June 2024, abbreviated tests were undertaken, to examine the effects of enhancements made to the development version source code described in section 5:

- Terminal implicit or predicted catchabilities applied in simulation projections of effort- and catch-conditioned fisheries

These tests included:

- Doitall fits of catch-conditioned models (ALB2024, BET2023)
- Single function evaluations with a single iteration step, catch-conditioned models (ALB2024, BET2023), and catch-errors model (YFT2014)
- Single function evaluations of a catch-conditioned deterministic projection model, with and without fml_effort_rltnshp predicted catchabilities in the projection periods (SKJ2022).

The removal of several unnecessary assignments made to redundant dvar_vectors relevant only to the catch-errors model catchabilities, altered the order of the derivative calculations that produced extremely small differences to the gradient calculation only. The function evaluation and dependent variables were identical among the versions. This slight change in the gradient produced slight changes to the minimisation path taken during the doitall fit, especially for complex examples, e.g., BET2023. For both doitall fit examples tested, the objective function was improved, and for the BET2023 example, the gradient was also improved. The estimated depletion dependent variables differed by <2% and 3%, for the ALB2024 and BET2023 examples, respectively.

All model operations involving a single evaluation, including projections, were immune to the effects of the code changes, i.e., identical results were obtained from both versions.

The code changes made in version 2.2.6.0 represented an improvement, and provide the functionality needed for the MSE and TRP work. It was recommended that the new development version (2.2.6.0) replace that being used for the 2024 stock assessments (2.2.5.3).

2.6.6 Abbreviated test – version 2.2.7.0

On 5 July 2024, abbreviated tests were undertaken, to examine the effect of enhancements made to the development version source code described in section 5:

- Input of sigma for the non-concentrated CPUE likelihood

This test using the ALB2024 example, examined the effect of the altering the input method for the assumed error of the non-concentrated CPUE likelihood, and is fully described in section 5.2.5.

The slight difference possible in converting from the input penalty weight (in earlier versions) to a $\boldsymbol{\sigma}$ value could alter the converged solution of some complex models, but this is unlikely to be significant in terms of the management quantities of interest.

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 2023-24, of the thirteen recommendations, 12 had been implemented and tested, and 1 remained yet to be developed:

- Non-uniform size bins (recommendation "b")

No further progress was made on recommendation ("b") during 2023-24, and remains as an incomplete task on the work plan.

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 4 commits were made to the repository during 2023-24, indicating the low use or maintenance of this package.

2.9.2 Testing framework

The testing framework for MULTIFAN-CL compilations first developed in 2011-12, was applied during 2023-24 for the benchmark testing of version 2.2.5.0 (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. 2022), and the recent developments in 2023-24 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 2023-24 are described in section 2.6. The development version was last tested in July 2024 versus the version 2.2.6.0 in abbreviated tests. The positive result of a comprehensive benchmark test of this development version will then define it as being the **benchmark** source code, and then posted as the release version. 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, SKJ2016, YFT2020 and SKJ2022; 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, enhancements, and corrections

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 2022-23 this level of testing was done for the enhancements and corrections (see section 5), to ensure the correct calculations and the expected results.

<u>Review of Testing Framework</u>

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 2024. It has been identified as a concern, and that they be included in the 2024-25 work plan for the MULTIFAN-CL project.

The routine compilation and development of a macOS executable is fundamental to the project, and the testing framework includes the MacMini host, and the macOS executable within tests among platforms and versions. The framework therefore has capacity for conducting tests 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 MULTIFAN-CL Viewer provides a ready means of examining independent and dependent variables of model solutions by illustrations and plots. A number of corrections were made from 5 Dec. 2003 to 7 Mar. 2024.

- 2 issues with missing dialog boxes on startup
- A reported issue concerning multi-sex input files
- A reported issue regarding scaled mean lengths-at-age in growth plots
- An issue with the file loading dialog box (due to a change of library).

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 Linux, Windows or macOS platforms. This grid enables parallel 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 evaluations. During 2023-24, 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 made since version 2.0.5.1. Proposed future revisions include: incorporating the suggestions arising from the earlier Training workshops; and the recent features and enhancements added up to and including version 2.2.7.0. The revised version will be posted on the website http://www.multifan-cl.org/.

3 TRAINING WORKSHOP

No formal training tutorials were required during 2023-24, although regular training support and Q and A support was provided to SPC analysts, and Nick Davies attended most weekly meetings of the stock assessmemt team.

4 NEW FEATURES

No significantly innovative or new features were implemented into the MULTIFAN-CL source code during 2023-24. Rather, the features added since 2019 (e.g., catch-conditioned method for estimating fishing mortality) have been consolidated, enhanced, and extended for their implementation in population projections. The current development version is **2.2.7.0** which holds all the developments described in section 5 that relate to enhancements and corrections made to the existing features since July 2023. These will be merged to the next release version upon the completion of the forthcoming benchmark testing.

5 ENHANCEMENTS AND BUG FIXES

An overview of the enhancements and corrections made to existing features in MULTIFAN-CL during 2023-24 (up to March 2024) was provided to the pre-assessment workshop (Hamer, 2024). Those, and other developments made subsequent to that meeting, are described in more detail in this section.

5.1 Simulation projections

Substantial enhancements have been made to the catch-conditioned feature in respect of undertaking population projections in the previous two years, and this was continued in 2023-24. These related primarily to the assumed catchabilities applied to effort-conditioned fisheries, and to catch-conditioned fisheries for which pseudo-observations of effort were required. Also, the capability was added to generate pseudo-observations of CPUE indices from stochastic projections.

5.1.1 Terminal catchabilities for effort-conditioned projections

Undertaking projections entailing effort-conditioned fisheries, requires an assumption for constant catchability for the projection time periods, and typically this is equal to the catchabilities estimated for the terminal estimation time periods. The two approaches for deriving catchability for the estimation time periods are:

- estimating a regression relationship between the observed effort and catchability; and,
- taking the “observed” catchabilities based upon the Newton-Raphson solution for fishing mortality and the observed effort.

Both approaches require that observed effort data is supplied for the extraction fisheries.

For the first approach, as part of the catch-conditioned method for fishing mortality estimation, MULTIFAN-CL has an existing feature for estimating a relationship between fishing mortality and observed effort (**fml_effort_rltnshp**) that provides catchability predictions. For fishing incidents where observed catch is unavailable, but effort is available, the predictions can be used for deriving fishing mortalities, and hence catch. During 2023-24, an enhancement was made to formally apply this relationship in model projections that include effort-conditioned fisheries. Typically, only those catchability predictions from the terminal year of the estimation model are employed for the projection fishing incidents.

For the second approach, an enhancement was made to derive the model “observed” catchabilities, i.e., the empirical values derived from the Newton-Raphson solution of fishing mortality, together with the observed effort. The advantage is that it avoids estimating the fml_effort_rltnshp regression when conditioning an Operating Model (OM) to be used for projections, which adds a component term to the integrated likelihood. Consequently, the stock assessment models (MLE) can be used directly as OMs without modifications to the MLE estimates. However, the empirical catchabilities for the terminal time periods have variability due observation error (in effort), that would otherwise be “smoothed” by the regression derived in the first approach. This creates potential for the variability in the terminal catchabilities having influence on the projections.

5.1.1.1 Method

The relationship between fishing mortality F_y, effort E_y, and catchability q_y is:

$$F_y = q_y E_y \qquad \text{Eq. 1}$$

and the predicted catch C_y in year y is,

$$C_y = (1 - e^{F_y})N_y \qquad \text{Eq. 2}$$

where N_y are the population numbers.

In order to derive one of the terms in Eq. 1, it is necessary for the other two terms to be known. Therefore, for an effort conditioned fishery, and to derive the catch term, it is necessary for the for E_y and q_y to be known. The availability of these terms depends upon the method of estimating F_y in any one model time period, and in projections, whether fisheries are either catch- or effort-conditioned. The catch-conditioned method for estimating fishing mortality in projection time periods employs two methods:

- Where catch is unknown, effort is known (**effort-conditioned fishery**) – fishing mortality is derived using: the observed effort, and the assumed catchability from the estimation model terminal time periods, $\hat{q}_{term}$
- Catch is known (**catch-conditioned fishery**) – fishing mortality is solved via the Newton-Raphson (N-R) procedure given the observed catch

For effort-conditioned fisheries, it is assumed the catchability estimated for the terminal year of the estimation periods, $\hat{q}_{term}$, is applied in each of the projection periods. This catchability is obtained using one of the two approaches.

1. Predictions of the fml_effort_rltnshp

   The fml_effort_rltnshp predicted catchability is:

$$\hat{q}_{term} = \hat{P}_i * G_{term}$$

where $\hat{P}_i$ is the estimated polynomial function for the i coefficients, and G_{term} is the Gram-Schmidt design vector for the terminal year of the estimation model time periods. The regression is fitted to the "observed" empirical catchabilities derived from the N-R solution for $\hat{F}$ and the observed effort in the terminal fishing incidents.

2. Model "observed" (empirical) catchability

   The "observed" empirical catchability, is derived from the N-R solution for $\hat{F}$ and the observed effort:

$$q_{term} = \frac{\hat{F}_y}{E_y}$$

Projected catches of effort-conditioned fisheries are obtained using equations 1 and 2 with substitution for q_y with either $\hat{q}_{term}$ or q_{term} depending upon the approach being used. Using the assumed catchability for the terminal year, in each of the projection model time periods the predicted fishing mortality F_y for a projection time period is derived using the observed effort:

$$\hat{F}_y = \hat{q}_{term} * E_y$$

Consequently, the model "predicted" effort is the same as that observed, and can be obtained by the simple reversal of the equation, with the application of the normalization factor to express the predicted effort in the true units:

$$\acute{E}_y = \left(\frac{\hat{F}_y}{\hat{q}_{term}}\right) \times \bar{E}$$

Whereas the predicted catch will depend upon which catchability is employed, either $\hat{q}_{term}$ or q_{term}.

5.1.1.2 Testing example

A suitable projection model (skipjack tuna 2022 assessment model) that employs the catch-conditioned method for fishing mortality estimation was selected, that included both effort- and catch-conditioned projection fisheries, with the estimation of fml_effort_rltnshps for a selection of each. It was configured to undertake projections over 30 calendar years.

A total of 39 fisheries are defined, of which 8 (fisheries 32 – 39) are CPUE survey index fisheries. Of the remaining fisheries, 20 fisheries were included in the fml_effort_rltnshp regressions. The example had established these 20 fisheries as being either effort- or catch-conditioned for the projection periods as follows.

Effort-conditioned (eff_proj_fshry) 2, 5, 8, 12, 14, 15, 19, 20, 25, 26, 29, 30

Catch-conditioned (catch_proj_fshry) 1, 4, 7, 13, 18, 22, 24, 28

The estimated fml_effort_rltnshp regressions for these fisheries are presented in Figure 1, with catchability expressed on the log-scale. The number of polynomial degrees for each relationship was 8, of which two coefficients determined the seasonal pattern, and five were higher order coefficients for the year effect.

In the terminal calendar year of the estimation periods, the observed effort was not available for all the time periods. As such, for some fisheries, the design vectors for the terminal year were incomplete. Table 1 shows the number of incidents (with a maximum of 4, quarters) in the terminal calendar year having effort available for each of the projection fisheries. The fml_effort_rltnshp regressions for these fisheries are presented in respect of only the terminal calendar year fishing incidents in Figure 2, illustrating the number of incidents is 4 or less, and the sign and magnitude of the deviates from the fitted fml_effort_rltnshp predicted catchabilities. Note that the y-axis scale is unique to each fishery's catchability range.

Consequently, for each projection year, the effort data (projection fishery is effort-conditioned) was replicated exactly as was available in the terminal year, i.e., only for those quarters having data available for deriving the fml_effort_rltnshp predictions. As such, the projections were essentially the "status quo" fishing strategy as occurred in the terminal year of the estimation periods.

There were 47 estimation model calendar years (188 time periods), and 30 projection model calendar years (120 time periods). However, as indicated above, for not all projection fisheries was effort available for all periods of the calendar year, and consequently the vector lengths of predicted effort differ among the fisheries.

5.1.1.3 Testing design

The example was used to undertake deterministic and stochastic projections that employed the alternative two approaches for the terminal year's catchabilities, and the two models were denoted:

- **Proj_fml** –fml_effort_rltnshp predictions
- **Proj_qterm** – "observed" (empirical) catchabilities

Using a deterministic projection, pair-wise comparisons among the models were made in respect of: catchabilities in the estimation and projection time periods; and, catch and spawning biomass predictions for the projection time periods.

Three stochastic projection simulations were performed, with random recruitments, and no observation error assigned to predicted catches. Comparisons were made within each simulation among the two models.

5.1.1.4 Results

Deterministic projection

The projection model was run for a single deterministic simulation under predicted recruitment (Beverton-Holt stock-recruitment relationship predictions) without error.

The catchability time-series of both models are illustrated in Figures 3 and 4, for both the estimation and projection periods. This serves to confirm the methodology is implemented correctly, and assesses how reasonable is each approach in respect of the terminal year catchabilities being applied for the projection periods.

The Proj_fml model catchabilities show the correct match between the terminal year catchabilities and those applied in the projection periods for all fisheries, i.e., the method is being correctly applied, (Figure 3). In general, and for most fisheries, the assumption was reasonable that the terminal year catchabilities as predicted from the fml_effort_rltnshp were consistent with those of the "observed" (empirical) catchabilities in recent time periods (previous 3-5 years), however for some fisheries, e.g., fisheries 12, 20, 28 and 29, the fml_effort_rltnshp predictions in the terminal year values were anomalies, due to the large deviates in the terminal year (Figure 2).

The Proj_qterm model catchabilities, similarly show the correct match between the terminal year catchabilities and those applied in the projection periods for all fisheries, i.e., the method is being correctly applied (Figure 4). In general, and for most fisheries, the assumption was reasonable that the terminal year catchabilities were consistent with those of the recent time periods (previous 3-5 years), however for some fisheries, e.g., fisheries 12, 13, 14, 19 and 28, the terminal year values were anomalies.

Predicted catches of the effort-conditioned fisheries from the Proj_qterm model are lower than that of the Proj_fml model for fisheries 5, 12, 19, 29 and 30 (Figure 5). For these fisheries, negative deviates were estimated in the fml_effort_rltnshp regressions for the terminal year (Figure 2), and for quarters having high effort. The corresponding opposite occurred for fisheries 2, 8, 14, and 15 for which positive deviates were estimated, however, these fisheries account for lower fishing mortalities. Consequently, the overall effect of using the Proj_qterm approach was to reduce total removals during projections, resulting in a 3.6% on average higher biomass than that of the Proj_fml model (Figure 6).

<u>Stochastic projections</u>

The projection model was run for a small set of 3 simulations with catch predicted for the effort-conditioned fisheries; from each simulation and without pseudo-observation error.

Given that the underlying model parameters were identical for the deterministic and stochastic projections, the relative differences between the Proj_fml and Proj_qterm predictions were similar. Also, the relative effect would be constant among the simulations, and therefore the results of only the first simulation are presented for making comparisons. As for the deterministic projections, those fisheries for which negative deviates of the fml_effort_rltnshp regression were estimated (5, 12, 19, 29 and 30) the Proj_qterm model predicts lower catches, and fisheries for which positive deviates were estimated (2, 8, 14, and 15) the Proj_qterm model predicts higher catches, compared to those of the Proj_fml model (Figure 7).

As for the deterministic projections, the overall effect of Proj_qterm model's approach was to reduce total removals during projections, resulting in a 3.6% on average higher biomass in all three simulations than that of the Proj_fml model (Figure 8).

5.1.1.5 Conclusions

Differences in the projection dependent variables caused by using either of the two approaches for assuming the terminal catchabilities is dependent upon the magnitude and sign of the fml_effort_rltnshp regression deviates in the terminal year. For effort-conditioned fisheries, negative fml_effort_rltnshp regression deviates result in lower predicted catches for the Proj_qterm model compared to the Proj_fml model, and vice versa. Estimating the regression relationship between the "observed" and polynomial predictions (fml_effort_rltnshp) of catchability effectively "smooths" the variability in the observed values, and provides the predicted mean catchabilities. The decision as to which approach to employ is case-specific, and most likely depends upon the level of observation error in the observed effort in the terminal calendar year of the estimation model. This will determine the magnitude and sign of the fml_effort_rltnshp regression deviates in the terminal year, and hence the level of difference among the approaches. For the testing example used, the mean percentage difference in projection spawning biomass was around 3.6%, since most of the terminal year deviates were negative that were applied to the larger effort-conditioned projection fisheries, resulting in lower total removals for the Proj_qterm model.

The feature to implement either of the two approaches for assumed terminal catchabilities in the effort-conditioned fisheries in projection model time periods has been implemented in MULTIFAN-CL, with testing using a relevant example producing the expected results.

5.1.2 Simulation pseudo-observations of effort for catch-conditioned projection fisheries

For projections entailing catch-conditioned fisheries, it may be desirable to obtain predictions of the effort associated with the assumed projected catches. This effort prediction has utility for Management Strategy Evaluations (MSE), where the catch-conditioned operating model (OM) is used to generate pseudo-observations of effort subsequently input to separate OMs for a related species. A method for deriving this projection quantity (i.e., effort pseudo-observations for a catch-conditioned projection fishery) has been developed. This method uses the assumed catchability for the terminal calendar year derived using either of the two approaches described in section 5.1.1, and its implementation is presented using the same example.

5.1.2.1 Method

For the catch-conditioned fisheries, fishing mortality is solved from the Newton-Raphson (N-R) procedure given the observed catch, C_y

$$\hat{F}_y = f(N - R\ solution, C_y)$$

Using the assumed catchability based upon either: the fml_effort_rltnshp prediction; or, the "observed" values, i.e., either $\hat{q}_{term}$ or q_{term} for the terminal year; in each of the projection model time periods the predicted effort is:

$$\hat{E}_y = \frac{\hat{F}_y}{\hat{q}_{term}}$$

Therefore, the predicted effort will depend upon the approach used for the assumed terminal catchability.

In MULTIFAN-CL, the catch equation calculations are undertaken using fishing effort normalized over the estimation model time periods. To express the predicted effort $\hat{E}_y$ in the same units as the observed effort input for the fishery, the average observed effort input the estimation period $\bar{E}$ (in true units) is applied:

$$\acute{E}_y = \left(\frac{F_y}{\hat{q}_{term}}\right) \times \bar{E}$$

where $\bar{E}$ is taken from the calculations for normalising fishing effort input to MULTIFAN-CL. Note that the normalisation is done only for the model "estimation period", i.e. for the periods within that applied in the minimization, and do not include the projection periods. This ensures the catchabilities are estimated for normalised effort over the estimation period only.

5.1.2.2 Testing example

Testing of the method employed the same example as that in section 5.1.1.2, for which certain projection fisheries were catch-conditioned, and for which fml_effort_rltnshps were estimated. As such the example was used to undertake stochastic projections that employed the alternative two approaches for the terminal year's catchabilities, and the two models were denoted:

- **Proj_fml** –fml_effort_rltnshp predictions
- **Proj_qterm** – "observed" (empirical) catchabilities

Comparisons among the two approaches for assumed terminal catchabilities were made in respect of catch-conditioned fisheries for which pseudo-observed effort is predicted. Using stochastic projections, the projection model was run for a small set of 3 simulations with status-quo catches in the projection periods equal to that of the terminal calendar year of the estimation model time periods, and effort predicted for the catch-conditioned fisheries from each simulation both with, and without, pseudo-observation error. The observation error CV was set at 0.3 (age_flags(186) = 30). Pair-wise comparisons of predicted effort were made among the two models from projections with and without error.

5.1.2.3 Results

Fishing incident-specific observed catch and effort data for the catch-conditioned fisheries over the estimation periods, and the predictions from three simulation projections (without error), is presented in Figure

9. Clearly, the predicted effort in the projections correspond well with the observed effort of the terminal estimation model time periods. This may be expected given the assumption of constant catchability in the projections equal to the fml_effort_rltnshp predictions for the terminal estimation model time periods, and a generally stable population abundance. One exception was that for quarter 1 of fishery 1, where the predicted effort was substantially lower. This is attributable to the large deviate of the fml_effort_rltnshp prediction for that fishing incident (Figure 2), such that the difference in the observed and predicted catchabilities (in normal space) is 57%. The higher predicted catchability produced lower predicted effort in the projections. This identifies the possible impact of the constant catchability assumption upon the projection predictions of effort.

This result is also illustrated in a relative comparison of the effort time series over the estimation and projection model time periods (Figure 10), such that the projection period effort predictions, both with and without error, are within the range of magnitude of that observed in the terminal estimation model time periods. The predictions without error remain reasonably consistent over time, maintaining the estimated seasonal catchability patterns, and exhibit less variability than those including pseudo-observation error, as expected.

A comparison is made between the predicted effort obtained using assumed constant catchability $\hat{q}_{term}$ from either the estimated the fml_effort_rltnshp regression (**Proj_fml**) or the "observed" empirical values (**Proj_qterm**), (Figure 11). For those catch-conditioned fisheries having negative deviates of the fml_effort_rltnshp regression (1, 4, 13, and 18, see Figure 2), the Proj_qterm model predicts higher effort compared to the Proj_fml model (Figure 11). Fishery 28 is an exception, where higher effort was predicted despite having a positive deviate, and this is due to the low precision possible in the Newton-Rapshon calculations for the extremely small levels of catch and effort. The corresponding opposite pattern is found for the fishery for which a positive deviate of the fml_effort_rltnshp regression was estimated (7, second quarter), such that the Proj_qterm model predicts lower effort compared to the Proj_fml model (Figure 11).

5.1.2.4 Conclusions

To derive the effort term employed in the model catch equation, it is necessary for fishing mortality, F_y, and catchability, q_y, to be known. For a catch-conditioned fishery in any projection time period, it is therefore necessary to assume a catchability based upon that estimated for the terminal year $\hat{q}_{term}$, this being the constant catchability assumption for projections. The method presented here obtains $\hat{q}_{term}$ either from the estimated the fml_effort_rltnshp regression, or the "observed" empirical values.

The scatterplot (Figure 9) confirms that the constant catchability assumption is being implemented correctly for the projection time periods, with predicted effort from the catch-conditioned fisheries being consistent with that observed, and to which the fml_effort_rltnshp was fitted in the terminal estimation model time periods.

The general effects of differences among the two assumed terminal catchability approaches upon predictions of effort have no impact upon model dependent variables, such as biomass; since the method is simply producing pseudo-observations of projection effort, and has no effect on fishing mortality. However, the $\hat{q}_{term}$ assumed (either from the estimated the fml_effort_rltnshp regression, or the "observed" empirical values) does impact on the effort predictions. Negative fml_effort_rltnshp regression deviates result in higher predicted effort for the Proj_qterm model compared to the Proj_fml model, and vice versa.

5.1.3 Simulated CPUE pseudo-observations for catch-conditioned model

The feature for a "simulation mode" in MULTIFAN-CL enables the generation of pseudo-observations that can be used, for example, in stochastic projections for MSE. This entails using an OM conditioned from the fit to actual observations, and then run in simulation mode for generating pseudo-observations, i.e., to simulate pseudo-observations from the operating model predictions. This feature was developed preceding the catch-conditioned model (CCond) feature in MULTIFAN-CL, such that the catch-errors model variables for catch and effort were simulated. For the CCond model fitted to observed standardised catch-per-unit-effort indices (CPUE), the model predictions do not relate to catch and effort, but rather to relative indices of vulnerable abundance. As such, the feature required development for the case of the CPUE likelihood.

5.1.3.1 Method

The negative log-normal likelihood for the survey index for a specific fishery, k, is:

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

where P_i and O_i are the normalised predictions and observations in each time interval *i*, respectively, on the log-scale. The predicted index is:

$$\hat{I}_{ki} = \sum\nolimits_j N_{rij}^{mid}\, S_{kj}\, \overline{w}_j$$

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*. As such, the prediction is a function of the vulnerable population abundance, rather than a relative catch rate.

Ignoring the variance terms, the simple form of the normalised predictions and observations entering into the numerator term of the likelihood is:

$$\left(\log(I_{ki}) - mean(\log(I_{ki}))\right) - \left(\log(\hat{I}_{ki}) - mean(\log(\hat{I}_{ki}))\right)$$

To produce pseudo-observations derived from the model predictions $\hat{I}_{ki}$ on the same scale of magnitude as the observed indices as input on the normal scale, one can assume from the likelihood that the approximation is:

$$\left(\log(I_{ki}) - mean(\log(I_{ki}))\right) = \left(\log(\hat{I}_{ki}) - mean(\log(\hat{I}_{ki}))\right)$$

And therefore, the pseudo-observation on the normal-scale is:

$$P_{ki} = e^{\log(\hat{I}_{ki}) - mean(\log(\hat{I}_{ki})) + mean(\log(I_{ki}))}$$

that essentially re-scales the normalised predictions in log-space to that of the mean of the normalised observations, and then exponentiates the result to the normal-space.

Note that the intervals *i* for the $mean(\log(I_{ki}))$ and $mean(\log(\hat{I}_{ki}))$ relate only to the estimation model time periods. Whereas, the $\log(\hat{I}_{ki})$ can include predictions generated for the projection model time periods. This ensures the magnitude of the pseudo-observations relate to the scale of the observed CPUE input for the estimation model time periods only.

<u>Grouped fisheries</u>

Special attention is paid to the mean observed and predicted indices for the case of fisheries assumed to have stationary catchability, i.e., the fisheries are grouped. In this case, the means are derived among all indices *i* for all fisheries *k* making up fisheries grouping *g*. Therefore, the pseudo-observation for a grouped fishery *k* within a grouping *g* on the normal-scale is:

$$P_{ki} = e^{\log(\hat{I}_{ki}) - mean(\log(\hat{I}_{gi})) + mean(\log(I_{gi}))}$$

<u>Pseudo-observation error</u>

Random log-normal error is applied to the re-scaled predictions given an assumed standard error *se*:

$$\varepsilon = (se * \varphi) - \frac{se^2}{2}$$

Where φ is a random number, and the randomised predictions on the magnitude scale of the observed CPUE for the estimation model time periods in normal space are:

$$P_{ki}^{*} = P_{ki} * e^{\varepsilon}$$

5.1.3.2 Testing example and design

The SKJ2022 assessment model was used as the example with which to develop this feature. Eight survey fisheries are defined, each within one of the eight model regions. Clear differences in the index magnitude exist among these fisheries (Figure 12), reflecting the “relative weighting” principle assumed in the model, i.e., the indices are used to infer differences in absolute abundance among regions. There are 188 quarterly time intervals in the model estimation period. The example was configured for undertaking projections over 120 quarterly time intervals, within in each of three simulations with randomised recruitments. While this is a low number of simulations, it is considered sufficient to demonstrate the feature is working correctly. A CV = 0.3 was assumed for the pseudo-observed CPUE observation error.

Firstly, to ensure the pseudo-observations produced without error were of the equivalent magnitude of scale as the observed CPUE, the “true” CPUE observations taken from the likelihood, were substituted into the routine that implements the equation for P_{ki} for the term $\log(\hat{I}_{ki})$. Identical values to those input in the fisheries data were obtained, confirming the method for scaling the indices relative to the observed indices was being calculated correctly within the routine.

Three scenarios were investigated for the generation of pseudo-observations:

sim_cpue_noerr – generated with standard error = 0, i.e., model predictions without error
sim_cpue_projerr – generated with error on the predictions for the projection time periods
sim_cpue_allerr – generated with error on the predictions for both the estimation and projection time periods

The assumed CV for CPUE error was: 0.3 (age_flags(26) = 30). For each scenario, the pseudo-observations are compared relative to the input observations over the estimation model time periods in normal space.

5.1.3.3 Results

Comparisons between the observed indices and the pseudo-observations of the sim_cpue_noerr scenario, confirms that the pseudo-observations are generated on the correct magnitude scale in normal space for both the estimation and projection time periods (Figure 13). As expected, variability among the simulations is limited to only that due to the random recruitments. It is noted that the projection variability due to the random recruitments is variable among the regions, being higher in the regions associated with fisheries 35 to 37.

Comparisons among the sim_cpue_noerr and sim_cpue_projerr scenarios confirm the implementation of observation error for the projection time periods only, and consequently substantially increases the variability in the pseudo-observations for the sim_cpue_projerr scenario (Figure 13).

Comparisons among the sim_cpue_projerr and sim_cpue_allerr scenarios confirm the implementation of observation error for both the estimation and projection time periods (Figure 13). Note that while the same input seed was applied to both scenarios, it applies to different vector lengths; one excluding the estimation periods, and the other including. Therefore, the random numbers used for the random deviates will differ, and so the random variability in the projection periods are different among the scenarios.

5.1.3.4 Conclusions

Implementation of the enhancement to the simulation mode feature in MULTIFAN-CL to generate pseudo-observations of CPUE indices from model projections has been completed. These when produced without error were of the equivalent magnitude of scale as the observed CPUE. The method takes account of the survey fisheries being grouped in respect of having a shared “stationary” catchability.

5.1.4 Summary

The enhancements made to the simulation model feature of MULTIFAN-CL that employs the catch-conditioned model to undertake population projections have significantly improved the utility of this feature. In particular, the capability for specifying which approach for assuming constant catchability in the projection periods (either predicted or empirical) has been well-defined, tested, and easy to implement. This assumption takes effect in projections of effort-conditioned fisheries, in determining future fishing mortalities, and also in catch-conditioned fisheries for which pseudo-observations of effort are generated. Together with the added capability to generate pseudo-observations of CPUE indices from stochastic projections, the simulation model feature is well-equipped for undertaking MSE projects.

5.2 Survey Fishery CPUE likelihood

5.2.1 Rationale

The feature for fitting to CPUE indices of relative abundance when employing the catch-conditioned model was first developed in 2020 (Davies et. al 2021), with enhancements and corrections made since then. During 2023-24, further enhancements were made to: instate the constant term in the non-concentrated likelihood form; allow variable likelihood weighting for grouped fisheries; correctly define time-variant precision of the observed indices in the concentrated likelihood formulation; and, ensure consistency in the implementations of these enhancements for grouped and un-grouped fisheries, and in the flag settings used. This consolidates this feature to be consistent among the two likelihood forms, and ensures their seamless implementation respectively during model runs that explore the two forms.

5.2.2 Constant term for non-concentrated CPUE likelihood

MULTIFAN-CL versions preceding version 2.2.5.0 have employed the non-concentrated CPUE likelihood that excludes the constant term from the negative log-normal formulation. While having no effect on the goodness of fit, it is: not formally correct; is inconsistent with the concentrated CPUE likelihood formulation; and, excluding the term alters the absolute magnitude of likelihood value. The effect of this exclusion will influence the relative influence of the CPUE term when fitted in an integrated regression that includes other data types, and may affect the converged solution. The formulation of the concentrated likelihood includes the constant term. Consistency among the two forms is preferrable for validating comparisons of alternative and explorative models that employ either of the two formulations. Therefore, the non-concentrated formulation was corrected to include the constant term.

5.2.2.1 Method

When no index-specific precision (λ_i) is available, the general likelihood form simplifies to include only the fishery-specific penalty weights σ_k, and is called the **non-concentrated likelihood** form:

$$0.5\sum_{k}(n_k * \log(\sigma_k^2)) + 0.5\sum_{k}\sum_{i}\frac{(P_{ik}-O_{ik})^2}{\sigma_k^2}$$

where n_k is the number of observed indices for fishery k. The above formulation as previously implemented, excluded the constant term:

$$0.5\sum_{k}(n_k * \log(\sigma_k^2))$$

This term was instated in the non-concentrated CPUE likelihood formulation.

5.2.2.2 Testing examples and design

The stock assessment diagnostic case models for BET2023, YFT2023, and SKJ2022 were used as examples to explore the implications of the CPUE likelihood constant term in the integrated model fit.

Deterministic evaluation

To clearly demonstrate the constant term, a deterministic comparison using the BET2023 example was made of single model evaluations using the previous and corrected formulations, i.e., excluding and including the constant term, respectively. The two alternative evaluations were denoted in respect of the MULTIFAN-CL version number associated with the implementation of the two formulations:

vsn2220 – excludes constant term (version 2.2.2.0)
vsn2250 – includes constant term (version 2.2.5.0)

Minimisation evaluations

To explore the effect on a converged solution caused by the change in the CPUE likelihood term, a comparison was made between the solutions obtained using the two versions (using the Linux platform executable):

nonconc_pre – excludes constant term (version 2.2.2.0)
nonconc_post – includes constant term (version 2.2.5.0)

In each case of three example models tested (BET2023, YFT2023, and SKJ2022), both solutions were obtained using the identical doitall script file, with the only difference relating to the CPUE likelihood. Detailed pairwise comparisons were made among the two solutions obtained in respect of the likelihoods and selected dependent variables.

5.2.2.3 Results

Deterministic evaluation

The difference among the CPUE likelihoods, and therefore the integrated total likelihoods, was attributable only to the constant term included in the vsn2250 model evaluation, equal to -3452.4 points, or 77% of the absolute magnitude of the CPUE likelihood term (Table 4).

Minimisation evaluations

The solutions of both versions for all three examples tested converged from stable minimisations, with both solutions requiring a similar number of function evaluations in the final phase, and achieving similar maximum gradients. No visible differences were evident in the quality of fit to the observed CPUE indices among the two solutions for most of the survey fisheries, which may be expected given that the sums of squares term in the formulation is identical.

For the BET2023 example, with respect to the **nonconc_pre** solution, the **nonconc_post** CPUE likelihood is around 3400 points lower (Table 5); a similar difference to that of the deterministic comparison made above. The terms for most other data types were similar, although the weight frequency term improved by 170 points, while others worsened by between 12 and 95 points, (tagging and length frequency data, respectively). The **nonconc_post** solution has affected the fit among the data types, indicating the relative influence of the respective data types in the integrated likelihood had altered because of the change in the CPUE likelihood term's absolute magnitude. Both solutions converged with positive definite Hessian estimates.

However, given the change in the integrated total likelihood, differences in the dependent variables of the **nonconc_post** solution were found. Absolute adult abundance was on average 7.5% lower (Table 5) which was consistent over all time periods because the relative trends were almost identical; reflecting consistency in the CPUE among the solutions. Equilibrium yield quantities were around 3% higher, and this is attributable to a slightly higher estimated natural mortality, moderately higher estimated absolute recruitments, and a minor increase in mean length-at-age for the older age classes; with equilibrium biomass quantities being around 6-10% lower (Table 5). No differences in estimated selectivity-at-age were visible. Despite these relatively minor changes, only a 0.3% difference in a key management quantity of interest, the depletion level of adult biomass ($SB_{recent}/SB_{f=0}$, SB_{latest} /$SB_{f=0}$), resulted from the re-formulation of the CPUE likelihood for the BET2023 model.

For the YFT2023 example, with respect to the **nonconc_pre** solution, the **nonconc_post** the CPUE likelihood is around 1888 points lower. Negligible differences in the terms for the other data types are evident, at the 2nd or 3rd decimal place. There appears to be negligible effect on the **nonconc_post** solution upon the relative quality of fit among the component data types. Consequently, negligible differences of 0.01% or less were obtained for: absolute adult abundance, estimated natural mortality, estimated absolute recruitments, and growth (Table 6). Similarly, the key management quantity of interest, the depletion level of adult biomass ($SB_{recent}/SB_{f=0}$, $SB_{latest}/SB_{f=0}$), was essentially identical, indicating that no effect resulted from the re-formulation of the CPUE likelihood for the YFT2023 model. This result is consistent with the likelihood profile for the **nonconc_pre** solution, that indicated little conflict between the CPUE data and other types in the total integrated likelihood, as illustrated by the likelihood profile (Magnusson et al. 2023).

For the SKJ2022 example, with respect to the **nonconc_pre** solution, the **nonconc_post** CPUE likelihood is around 1598 points lower (Table 7). The terms for the other two data types have altered: the length frequency term worsened by 594 points; and, and the tagging data term improved by 17 points; indicating an effect of the reformulated CPUE likelihood. Castillo Jordan et al. (2022) illustrated data conflict among the CPUE and length-frequency data types for the SKJ2022 assessment model, and this could explain the change in the relative fit among these two data types for the **nonconc_post** solution.

Despite this difference in the fit among the data types, only relatively minor differences were evident in the dependent variables. With respect to the **nonconc_pre** solution, for the **nonconc_post** solution, absolute adult abundance is on average only 0.6% lower (Table 7) which was consistent over all time periods because the relative trends were almost identical; reflecting the similar fit to the CPUE time series among the solutions. Equilibrium yield quantities are around 0.3% lower, and this is attributable to a slightly lower estimated natural mortality, and slightly lower estimated absolute recruitments. Estimated growth was almost identical among the solutions (Table 7). No differences in estimated selectivity-at-age were visible. Consequently, only a 0.07% difference in a key management quantity of interest, the depletion level of adult biomass ($SB_{recent}/SB_{f=0}$, $SB_{latest}/SB_{f=0}$), resulted from the re-formulation of the CPUE likelihood for the SKJ2022 model.

5.2.2.4 Conclusions

The re-formulation of the non-concentrated CPUE likelihood to include the constant term has no effect on the sums-of-squares term, and therefore no direct effect on the goodness of fit to the CPUE was visible in all three examples tested.

For complex or poorly-determined solutions, the change in absolute magnitude of the CPUE term (that was relatively large) may alter the relative influence of this term within the integrated likelihood, and result in differences in the solution's dependent variables. However, for the complex example tested (BET2023), this effect on the key management quantity of interest (the adult biomass depletion level) was only slight (0.3%); while for the other two examples the effect was very slight or negligible.

5.2.3 Variable likelihood weighting for grouped fisheries

Grouping of the survey fisheries is possible such that fisheries within a group are assumed to share the same stationary catchability. Consequently, the absolute CPUE index values among the fisheries impact upon the model predictions for each component fishery within the group. This enables the assumption that the fishery-specific indices may reflect relative abundance among the population in the areas where the fisheries operate; often termed as "regional weighting".

In the general form for the negative log-normal likelihood, grouping is implemented, where P_i and O_i are the normalised predictions and observations, respectively, on the log-scale, and i are the observations of all fisheries within the group; such that the normalised predictions can be apportioned to fishery-specific sub-vectors within the group, k:

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

yet this is derived for fishery k in respect of the $\text{mean}(\log(\hat{I}_i))$ of the indices for all fisheries within the group. This maintains the assumption of stationary catchability in each sub-vector.

While the indices are grouped by sub-vector for deriving the normalised predictions and observations, P_i and O_i, respectively, fishery-specific assumed error, σ_k are assigned to the corresponding sub-vectors within the likelihood formulation in respect of fishery k, i.e., they are assigned via the distributive property. When no index-specific precision, λ_i, is available the general likelihood form will then simplify and include the fishery-specific assumed σ_k, and is called the **non-concentrated likelihood** form:

$$0.5 \sum_k (n_k * \log(\sigma_k^2)) + 0.5 \sum_k \sum_i \frac{(P_{ik} - O_{ik})^2}{\sigma_k^2}$$

where n_k is the number of observed indices for fishery k.

In the instance of grouping the survey fisheries, all the indices are grouped into single vectors for deriving the normalised predictions and observations, P_i and O_i, respectively. However, the assumption can be made that the **term σ is constant among all fisheries within the group**. In other words, a single assumed error term is assigned to the likelihood term. Preceding this development, this was the status as coded in MULTIFAN-CL.

The purpose of this development to MULTIFAN-CL, was to group the survey fisheries, so as to share the same stationary catchability, but **to have the capability for non-shared assumed error**. This will allow flexibility in the assumption, and that the σ is not constant among all fisheries within the group. This capability was also implemented for the concentrated form of the negative log-likelihood where **index-specific precision λ_i** is available (see section 5.2.4).

5.2.3.1 Testing example

The testing of the feature used the example tuna model ALB2021-3region for which one survey fishery occurred in each of the three regions: fishery 18 (region 1), fishery 19 (region 2) and fishery 20 (region 3).

5.2.3.2 Testing design

The implementation of this feature in the grouped CPUE likelihood entailed pair-wise comparisons among the minimised model solutions assuming equal versus variable assumed error terms, for both the concentrated and non-concentrated likelihood forms:

- Minimisation with variable σ_k
    - λ_i = input CVs from the external standardisation, and a range for σ_k equivalent to that explored with the non-concentrated form – the difference in the solutions illustrates the effect of the concentrated likelihood versus the non-concentrated form in respect of varying σ_k.

For this minimisation test, a small range of scenarios was explored to illustrate the effect of relaxing the assumption of a single error term within a fishery grouping, such that the relative weights may vary among the fisheries within the grouping. Fishery 19 was selected for applying a variable weighting because of the higher absolute CPUE indices compared to the other fisheries, and it may therefore exhibit greater effects of this feature.

For convenience, the variable error term was expressed in terms of a penalty weight corresponding to the value for σ_k based upon a normal prior:

$$w = \frac{1}{2\sigma_k^2}$$

The following table specifies the respective values for the penalty weights assigned for fisheries k = 18, 19, and 20, respectively for the **non-concentrated** likelihood form.

	pen_eq	**pen_7**	**pen_2**	**pen_50**
Penalty weight	10:10:10	10:7:10	10:2:10	10:50:10

- **pen_eq** – maintains the assumption of equal penalty weights
- **pen_7** – applies minimal variation in penalty weights (lower)
- **pen_2** – applies moderate variation in penalty weights (much lower)

- **pen_50** – applies large variation in penalty weights (substantially higher)

For the models employing the **concentrated** likelihood, the corresponding σ_k values associated with the assumed range of penalty weights are in the following table.

Penalty weight	10	7	2	50
σ	0.22	0.27	0.50	0.10

The following table specifies the respective values for the σ_k for fisheries *k* = 18, 19, and 20, respectively within the range.

	conc_sig_eq	**conc_sig_27**	**conc_sig_50**	**conc_sig_10**
Assumed σ_k	0.22 : 0.22 : 0.22	0.22 : 0.27 : 0.22	0.22 : 0.50 : 0.22	0.22 : 0.10 : 0.22

This range of equivalent relative weighting as applied to both likelihood forms, allows pairwise comparisons among the likelihood formulations, from constant to high variability in the assumed error.

5.2.3.3 Results

Pair-wise comparisons among the minimised solutions obtained using the two likelihood forms illustrates the effect of the index-specific precision λ_i in the concentrated likelihood in respect of the varying assumed error, σ_k.

The quality of fit to the fishery-specific CPUE indices changed visibly with altered penalty weights among the fisheries (Figure 18). Reducing the weight for fishery 19, (pen_2), worsened the fit to that fishery but improved the fit for fishery 20. Conversely, increasing the weight for fishery 19, pen_50, improved the fit, but worsened the fit for fishery 20. This pattern was evident for both likelihood forms. A difference among the non-concentrated and concentrated likelihood models was most evident for the extreme case for increased weight for fishery 19, (pen_50 and conc_sig_10), where the fit for fishery 20, was slightly worse for the concentrated model (Figure 18). This difference is most evident in the early time periods where the index-specific precision was lowest (Figure 14). Over these periods, the conc_sig_10 fit for fishery 20 was visibly worse, where the fit is compromised due to the very high weight assigned to Fishery 19, and the lower precision for fishery 20, enabling a poorer fit relative for the non-concentrated model.

For all models the assumption of stationary catchability resulted in the non-uniform distribution of absolute biomass among the regions, with the region of fishery 19 having highest biomass, and the region of fishery 18 the lowest (Figure 19). The effect of substantially increasing the penalty weight for fishery 19 (pen_50) is evident in a small change in the biomass for that region, but also a marked increase for the region of fishery 20 (Figure 19), for which the fit to the observed CPUE was worse (Figure 18). This effect was slightly more exaggerated for the concentrated likelihood model (conc_sig_10), probably due to the generally low index-specific precision, and hence worse fit, for fishery 20. The indirect effects of variable penalty weights may therefore influence not only the quality of fitting the observed CPUE, but also to some extent the stationary catchability assumption's effect in regional biomass distributions.

Only for the case of high variability in penalty weight (pen_50, conc_sig_10), having a 55% increase in the precision (weight) of fishery 19, was a marked difference produced in total absolute biomass (Figure 20), being generally higher. This being due to the higher estimated biomass in region 3. Similarly, only for the pen_50 and conc_sig_10 models, did the estimates of adult biomass depletion differ substantially from those of the other models (Figure 21). These results were almost identical among the non-concentrated and concentrated likelihood formulations.

5.2.3.4 Conclusion

In summary, varying the penalty weights in the grouped CPUE likelihood alters the relative goodness of fit to the observed indices according to the assumed weights, as might be expected. In the cases of high contrast in, or generally low, index-specific precision, this effect may be larger for the concentrated form. However, it may also influence the stationary catchability assumption's effect on the distribution of regional biomass. This may have a moderate effect on estimates of total biomass and depletion levels.

5.2.4 Concentrated CPUE likelihood with normalised lambda

The general form for the negative log-normal likelihood for CPUE indices is

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

where P_i and O_i are the normalised predictions and observations, respectively, on the log-scale, and *i* are the observations for a given fishery. This is the form where index-specific precision λ_i is available, and is called the **concentrated likelihood** form, where both the $\hat{\lambda}_i$ and σ terms are operational, and the $\hat{\lambda}_i$ terms are **normalised** over all *i* so as to retain the specified mean level of error defined by σ, and yet with the observed time-variance. When λ_i is unavailable, essentially equal to 1, i.e., time-invariant precision, the likelihood simplifies to the non-concentrated form.

5.2.4.1 Testing example

The testing of the feature used the example tuna model ALB2021-3region for which one survey fishery occurred in each of the three regions: fishery 18 (region 1), fishery 19 (region 2) and fishery 20 (region 3).

The observed CPUE trends were somewhat different among the fisheries (Figure 14), as were the catch rate magnitudes (highest in fishery 19 and lowest in fishery 18), and the time-variant precision of the observed indices also differed, with those for fishery 18 having highest precision, medium precision for fishery 19, and lowest precision for fishery 20. For all fisheries, precision is lowest for the initial period of about 10 years.

5.2.4.2 Testing design

The testing of the concentrated likelihood with normalised $\hat{\lambda}_i$ was undertaken in two parts in respect of comparisons with the simpler non-concentrated likelihood:

- Deterministic evaluations
  - single evaluation with fixed parameters – to confirm mean($\sigma_k * \lambda_i$) = σ_k. This ensures the normalised λ_i is intact. A fixed σ_k = 0.25 was applied.
  - single evaluation with fixed parameters, λ_i = 1 for all *i* and fixed σ_k = 0.25. To confirm the likelihood term for both the concentrated and non-concentrated forms are identical.
- Minimisation evaluations
  - λ_i = 1 for all *i* and fixed σ_k = 0.25 – determines concentrated and non-concentrated likelihoods produce very similar solutions
  - λ_i = input CVs from the external CPUE standardisation for time-variant precision, and fixed σ_k = 0.25 – the difference in the solutions illustrates the effect of the concentrated likelihood versus the non-concentrated form

5.2.4.3 Results

<u>Deterministic evaluations</u>

The first of the deterministic tests entailed a single model evaluation with fixed parameters to confirm the mean($\sigma_k * \lambda_i$) = σ_k, to ensure the normalised λ_i values are intact in the concentrated likelihood calculation. This is illustrated for each of the three survey fisheries, such that the mean value was equal to the fixed σ_k = 0.25 for each fishery k (Figure 15). Survey fishery 19 exhibits higher contrast in the time-variant precision of the indices, with those of the earlier period having substantially lower precision versus the latter 100 periods; whereas fishery 20 has generally lower precision over more of the time periods.

The second of the deterministic tests entailed single model evaluations with fixed parameters and applying both: the concentrated form with setting λ_i = 1 for all i; and, the non-concentrated form; assuming σ_k = 0.25 for all survey fisheries k. The component terms of the non-concentrated and concentrated likelihoods are presented in Table 2. As expected, the normalised λ_i condition with setting all λ_i set equal to 1, reduces the concentrated likelihood to an equivalent value to that of the non-concentrated likelihood form, and the survey and total likelihoods are identical (Table 2), for both the grouped and ungrouped survey fisheries cases. This deterministic test confirms the integrity of the likelihood formulations in the code.

Minimisation evaluations

The first minimisation test was to take the results of the second deterministic test (above) where: for the concentrated form, λ_i = 1 for all i, and σ_k = 0.25 for both the likelihood forms; and to run both model evaluations to convergence. The minimisation test was performed for the case of grouped survey fisheries only. The converged solutions of the two models were essentially identical, as may be expected given the equivalence in the σ_k and the normalised λ_i = 1 (Table 3).

The second minimisation test implemented the index-specific estimates of precision, λ_i, as available for the example (Figure 14), with fixed σ_k = 0.25 and to run the model evaluations for the concentrated likelihood form to convergence (conc_sig_0.25). The corresponding model solution is obtained for the non-concentrated form assuming the same value of σ_k, (nonconc_pen_8). This test illustrates the effect of the index-specific estimates of precision in the concentrated likelihood on the fit to the CPUE (Figure 16), and the biomass dependent variable (Figure 17). Differences in quality of the CPUE fit among the non-concentrated and concentrated likelihood models are consistent with the expected effects of the time-variant precision in the observations being accounted for within the concentrated likelihood. These effects were slight, most likely because of the relatively low contrast and generally high index-specific precision of the example used.

5.2.5 Sigma for non-concentrated CPUE likelihood

Input of the sigma term of the two CPUE likelihood formulations is implemented via a fishery-specific flag setting. The operation on this setting differed between the forms such that:

- non-concentrated - derives the penalty weight
- concentrated - derives the σ

This operational difference is error-prone for analysts to use, and cumbersome for comparing alternative and explorative models that employ either of the two formulations. Also, exact translations in the assigned error are not possible for certain values of the penalty weights and σ. Therefore, the flag operation for the non-concentrated formulation was made consistent with that of concentrated likelihood method, i.e., derives σ. This is converted internally to assign to the likelihood penalty weight. This achieves consistency among the two formulations.

5.2.5.1 Testing

Evaluation of the effect of this change upon existing model solutions employing the non-concentrated formulation focussed upon the input settings for the penalty weights. For example, a penalty weight value that enables a direct conversion to a σ value to within 2 decimal places might be:

- penwt = 12.50
- σ = 0.20000

This setting allows for an exact replicate function evaluation with the revised flag operation. However, a penalty weight value that prevents a direct conversion to a σ value to within 2 decimal places might be:

- penwt = 18.59
- σ = 0.164001

This setting does not allow for an exact replicate function evaluation with the revised flag operation because a direct conversion to a σ value to within 2 decimal places is not possible. Only a close approximation can be achieved, i.e., σ = 0.16.

The effect of this approximation was evaluated using a tuna example (ALB2024) with a deterministic evaluation using a solution having input of the penalty weights, and converting these to the closest approximation to two decimal places for an input of σ.

5.2.5.2 Results

Testing indicates that among 6 of the 7 survey fisheries, an exact conversion from the input penalty weight to a σ value was not possible, altering the assumed error of the non-concentrated likelihood, and hence its calculated value (Table 8). For this example, the necessary approximations to the nearest σ values, altered the total CPUE likelihood slightly by 0.06 points.

5.2.5.3 Conclusion

Having the input of the assumed σ value being consistent among both CPUE likelihood formulations is an improvement in terms of avoiding input errors, and enabling seamless transitions among the two formulations during model exploration.

This slight difference possible in converting from the input penalty weight (in earlier versions) to a σ value could alter the converged solution of some complex models, but this is unlikely to be significant in terms of the management quantities of interest.

5.3 Orthogonal-polynomial recruitments – regional distribution constraint

5.3.1 Rationale

While employing the orthogonal polynomial recruitment parameterization, it may be desirable to constrain the mean distribution of recruitments among regions to prior assumed values. However, the region level coefficients of the orthogonal polynomial parameterisation are applied to the formulation via the Gram-Schmidt design matrix, and it is not clear how these can be specified to set values for producing a desired prediction of the regional distribution.

A solution has been developed that assigns a normal prior penalty on the predicted regional recruitment distribution to constrain the estimated polynomial region level coefficients to a range which assigns the recruitment proportions as defined in the prior. Other priors have been applied directly upon the polynomial coefficients, and therefore this parameterisation is amenable to this method of constraint.

5.3.2 Method

A suitable vectorized normal prior penalty was implemented, where the elements for the mean proportions of total recruitments in each region *ir* were:

$$X_{ir} = \frac{\bar{R}_{ir}}{\sum_{ir} \bar{R}_{ir}}$$

where $\bar{R}_{ir}$ is the mean of the absolute recruitments over all time periods in region *ir*. The typical normal prior penalty function is:

$$p = w * (X_{ir} - P_{ir})^2$$

where P_{ir} is the target prior proportion in region *ir*, and w is the multiplicative penalty weighting factor assumed. The penalty value p is added to the integrated total negative log-likelihood calculated in the model function evaluation.

The P_{ir} are supplied as a vector entered into the input *.ini file in the section supplying the initial proportion of total recruitment occurring in each region. Upon undertaking the `−makepar` operation option, this input populates the `region_pars(1)` which are applied in the penalty when it is activated.

The constraint on the mean distribution of recruitments among regions entailing the normal prior penalty is denoted: **orthp_reg_recrs_pen**

5.3.3 Testing example and design

The design makes a demonstration of the implementation of this constraint, and compares it to a model that excludes the prior term from the total likelihood.

The example model used was a simple two-region model, employing the orthogonal-polynomial recruitment parameterisation, and fixed movement rates among the regions. It was based upon the albacore 2021 stock assessment data. Recruitments are defined to be quarterly, and occur in both regions. The model options explored were:

- **no_pen** – excludes orthp_reg_recrs_pen from total likelihood
- **orthp_pen** – includes orthp_reg_recrs_pen from total likelihood

The **orthp_pen** solution was obtained starting from the **no_pen** .par file, with the only difference being to include the orthp_reg_recrs_pen in the total negative log-likelihood, and to run multiple evaluations to convergence.

The mean regional distribution of recruitments estimated using the no_pen model was around 10 and 90% in regions 1 and 2, respectively. The target prior mean distribution specified for the orthp_pen model was: **0.741015** and **0.258985**; for regions 1 and 2, respectively.

Using **orthp_pen**, the sensitivity of the estimated mean regional distribution of recruitments among regions was tested relative to the assumed multiplicative weighting value (w in the normal prior penalty). The range tested for w was: 1, 10, 100, 1000, 10000.

5.3.4 Results

Applying the orthp_reg_recrs_pen with w = 10,000 altered the predicted recruitment distribution among the regions from that of the no_pen model, to being almost identical to the assumed prior target values (Table 9). The effect of this is illustrated in a comparison among the no_pen and orthp_pen models of the time series of absolute recruitments in each region (Figure 22), with an absolute shift in the recruitments over all time periods among the regions. However, this change produced only very slight differences (2% or less) in the model dependent variables, with the adult biomass in each region being relatively similar (Figure 23), although being slightly displaced among the regions. However, the total model biomass trajectory for the no_pen and orthp_pen models was almost identical (Figure 24). Overall, the effect of the imposed constraint was very slight on the model dependent variables and, particularly for the quantities of management interest ($SB_{current}/SB_{current\_F=0}$) that differed by <1%. This degree of effect is of course case-specific to this example used for testing the constraint.

Increasing the multiplicative weighting of the orthp_reg_recrs_pen produced a “staged” shift in the recruitment distribution among the regions towards that of the specified prior target values (Table 10). A weighting factor of 10 increased the penalty term value by the same order, with only a moderate change in the regional distribution (Table 3). However, a factor of 100 or more reduces the penalty term value substantially as a consequence of the predicted regional distribution being close to that of the prior assumed values.

Given that the penalty term value is relatively small (<0.6) in achieving predictions close to the prior assumed values, and with relatively slight changes to the model dependent variables and the negative log-likelihood; applying this penalty to the orthogonal-polynomial recruitment parameterisation appears to be a reasonable approach. For the example tested, a multiplicative weighting of 100 to 1000 appears adequate for constraining the regional recruitment distribution close to prior assumed values. However, this weighting value will be case-specific.

5.4 Independent variables report

5.4.1 Rationale

An aid for analysts during exploratory model development is a detailed and tabulated list of the independent variables being estimated relative to the prior bounds specified and the estimated gradient. This is useful for identifying poorly-determined parameters, and estimates at, or close to, the prior bounds. The parameter configurations of the model may then be assessed accordingly.

5.4.2 Method

During the final function evaluation for generating output reports, the placements of the independent variables into the vector used for the minimization is copied with assignments to the report variables, along with the index numbers, independent variable labels, respective bound values, and gradients. It was ensured these conditional assignments do not affect: the parameters vector used for the minimization; the reporting of the independent variables to the solution *.par file; nor, the capability of a model re-start from an existing solution *.par. As such, the minimization over multiple phases is robust to the generation of the independent variables report. If the total number of independent variables in the minimization vector does not match that assigned for generating the independent variables report, a sanity check throws an error message and the model evaluation will terminate.

If this feature is activated by the designated flag, and output file called “indepvar.rpt”, with an example of the format (using the SKJ2022 model example) is presented in Table 11.

5.5 Diagnostic model intact with fml_effort_rltnshp regression

5.5.1 Rationale

Implementing the feature for estimating a relationship between fishing mortality and observed effort (**fml_effort_rltnshp**) that provides catchability predictions, adds a term to the integrated total likelihood that in some cases may be large, say around 1.e+03. For complex models, this additional term may affect the minimisation solution depending upon the relative influences of the component data terms. If the relationship is essential for accounting for missing catch data, this might be acceptable. However, if it is simply required for using the model solution as a projection operating model (OM) with effort-conditioned fisheries data in the future, the effect on the solution may be undesirable, especially if it is substantial.

A feature was added that retains as intact a model solution, i.e., to keep all the independent variables fixed at their estimated values, and to include only the fml_effort_rltnshp regression to the minimisation. This additional regression then alters none of the other model dependent variables, but produces the fitted fml_effort_rltnshp function, from which catchabilities may be predicted for projections.

5.5.2 Method

The placements of the independent variables, besides the fml_effort_rltnshp regression coefficients, into the vector used for the minimization was made conditional upon a single flag setting. This removes these independent variables from the scope of the vector assignments, and retains only the fml_effort_rltnshp coefficients.

Operationally, given a converged model solution obtained without estimating the fml_effort_rltnshp regression, a further minimization is undertaken while holding fixed all the solution’s independent variable values, but with estimating those for fml_effort_rltnshp regressions for specified fisheries having observations of effort. This entails activating the flag settings for the fml_effort_rltnshp regression, and a single additional flag setting (parest_flags(392)=1) to hold fixed all other independent variables.

The method is denoted: **diagcs_intct_fml**

5.5.3 Testing example and design

The ALB2024 candidate diagnostic case model was used for testing the new feature, with three models developed:

- **diagcs_clipd** - diagnostic case model with no fml_effort_rltnshp estimated
- **seas_poly_fml_3degs_term** – start from the diagcs_clipd model, run for an additional phase minimisation with the terminal 3 calendar years having available effort data, and the fml_effort_rltnshp estimated with seasonality and the polynomial having 3 degrees

- **intct_fml_term** – start from the diagcs_clipd model, run for an additional phase minimisation with the terminal 3 calendar years having available effort data, fixing the diagcs_clipd independent variables; and the fml_effort_rltnshp estimated with seasonality and the polynomial having 3 degrees

The model **intct_fml_term** implements the **diagcs_intct_fml** method, and entailed the estimation of 91 independent variables for the fml_effort_rltnshp for 13 fisheries having effort data in the terminal 3 years.

5.5.4 Results

The **diagcs_intct_fml** method has been shown to function as intended, with the dependent variables of the **intct_fml_term** model being identical to that of the **diagcs_clipd** model (Table 12).

With respect to the of the fml_effort_rltnshp regressions, the diagcs_intct_fml method produced a moderately worse fit for almost all fisheries compared to the normal regression (the **seas_poly_fml_3degs_term** model), with the estimated $\boldsymbol{\sigma}$ values being larger for the **intct_fml_term** model (Figure 25). This is because the "observed" empirical catchabilities fitted in the regression are derived from estimated fishing mortalities and observed effort, that in the the normal regression are affected by the entire set of model independent variable estimates, rather than just the fml_effort_rltnshp coefficients. This is evident in the minor differences in the dependent variables and growth parameters of the **seas_poly_fml_3degs_term** model relative to the **diagcs_clipd** model (Table 12). For the **intct_fml_term** model the "observed" empirical catchabilities are fixed at the **diagcs_clipd** model values, and are not optimised during the minimisation. Hence, the fml_effort_rltnshp likelihood term for the **intct_fml_term** model is larger than that of the **seas_poly_fml_3degs_term** model (Table 12). However, the general patterns of the estimated polynomial functions were visibly similar (Figure 25).

5.6 MSY-related recent ratio reference points in variance report

An enhancement was made to the report of the estimated variances of the dependent variables in respect of adding the following ratio reference points:

$F_{recent}/F_{\mathrm{MSY}}$ - the average fishing mortality for a "recent" period i.e., over the last 4 calendar years including the latest; relative to that at the maximum sustainable yield (MSY) level

$SB_{recent}/SB_{F=0}$ - the average spawning biomass for a "recent" period (defined as above); relative to the average spawning biomass predicted to occur in the absence of fishing over a 10-year period preceding the most recent year less 1

$SB_{latest}/SB_{F=0}$ - the spawning biomass for the most recent year; relative to the average spawning biomass predicted to occur in the absence of fishing over a 10-year period preceding the most recent year less 1

$SB_{recent}/SB_{\mathrm{MSY}}$ - the average spawning biomass for a "recent" period (defined as above); relative to the spawning biomass at the MSY level

5.6.1 Method and results

The above dependent variables were added to the derivatives and standard deviation calculations in respect of the periods defined. Account was taken of the number of recruitments defined within the model calendar year, so as to ensure the calculation over the periods of ratio reference point were correct. These ratio estimates and their standard deviations were added to end of the output *.var report following the dependent variables obtained from the zero-fishing model evaluation.

It is noted that currently no estimated variance can be calculated for the MSY dependent variables. This is because the optimisation for MSY is done as part of the final evaluation for the report, and not during the gradient calculations of the model independent variables, and therefore, no gradient is calculated for MSY. Obtaining the gradients would require a customised model function evaluation to obtain all the dvariables needed for calculating the derivatives of the MSY-related variables.

5.7 Bug fixes

5.7.1 Frecent/Fmsy dependent variable derivative and variance calculation

In calculating the F_{recent} dependent variable, this period, as implemented, was defined over the model years, rather than the calendar years. For models configured to have more than one recruitment event per calendar year, i.e., a calendar year is comprised of multiple model years, the recent period would be incorrect. The correction was made to take due account of the defined recruitment frequency, such that the total number of years for the average was that defined for "recent" multiplied by the number of recruitments per year.

5.7.2 Likelihood components report: consolidated all terms

The values presented in the likelihood components report ("test_plot_ouput"), were consolidated and reconciled with that of the total integrated negative log-likelihood used in the minimisation.

In 2014, a feature was added to MULTIFAN-CL that produced an output report ("test_plot_ouput") to contain the component terms of the total integrated negative log-likelihood used in the minimisation procedure, and that was reported to the screen output. This report included the major components for each observation data type (e.g., size compositions, standardised CPUE, tagging data, etc.), but was yet to include the range of assumed priors and penalties on both independent and dependent variables. As such, the aggregate sum of the components included in the output report was not equal to that of the total integrated negative log-likelihood. The differences were attributable to the missing assumed priors and penalty terms applied for a given model, but also some terms for particular data types were missing or incorrect.

5.7.2.1 Method

The primary terms of the total integrated negative log-likelihood relate to the observation data types, and their values in the output report were checked for their integrity. For the tagging data, a correction was required to the pooled tag group term, due to the lack of an assignment to its class member.

All assumed priors and discretionary penalty terms applied to independent and dependent variables were traced during the calculation of objective function, as they contributed to the total integrated negative log-likelihood. In addition, certain penalties applied during the fishing mortality calculations were identified. Class members were defined for each, and then included in the output report.

The aggregate sum of the component terms, for the data types, priors and penalties as reported, was compared with that of the total integrated negative log-likelihood reported to the screen following the minimisation iteration.

Testing of the feature was carried out using the examples employing: catch-errors method (YFT2014); catch-conditioned method (BET2023, YFT2023, and SKJ2022); self-scaling multinomial size-composition likelihood with no random effects estimation (BET2017_ssmult, a catch-errors model); and, multi-sex structure (SWO2021_2sex, a catch-errors model).

Tests were also made that ensured the report was robust to: multiple evaluations; post-convergence fishing impact analyses; and, the Hessian calculation; such that the likelihood components in the report relate to the function evaluation of the converged solution only.

5.7.2.2 Results

Comparisons of the function evaluation total integrated negative log-likelihood values as reported to the screen output, versus the sum of the component terms reported to the "test_plot_output" report for the six model examples are presented in Table 13. The absolute difference between the total of the components in the report, and the screen report total is less than a maximum of 0.001 for the examples tested. This difference is likely due to the floating point of the C++ output to the "test_plot_output" report, and that of the R script during input and output. This slight difference in the total may be considered negligible for the purposes of using the components for detailed likelihood profile analyses, or for making relative comparisons among model examples.

5.7.3 Non-decreasing penalty function for time-variant selectivities

The existing feature for estimating time-variant selectivity was found not to account for a separate feature that imposes a non-decreasing penalty upon the estimated functions at age. Consequently, the two features could not be implemented in unison.

Code was implemented for when both features are activated, that indexes to which selectivity stratum the constraint is to be applied, i.e., in respect of: fishery; season; and, time-block. This is achieved by the input of two files prepared by the user containing the indices, called: “selseas_ff16.dat” and “selblocks_ff16.dat”. The format is: rows are the fisheries; and, columns are a vector on each row with the number of elements being the relevant seasons or time-blocks specified for each fishery (as defined by the analyst for the time-variant selectivity). The row elements are values either 0 or 1, where a non-zero value of 1 activates the non-decreasing penalty constraint for that fishery selectivity.

A sanity check is made among the settings supplied for the non-decreasing penalty versus the indices in the two input files, to ensure consistency in the temporal strata specified for the application of the constraint. This ensures that for an identified fishery, at least one time-block and one season is indexed for application of the constraint.

The correction was tested using the ALB2024 assessment model, where the non-decreasing penalty constraint was activated for either single, or multiple seasons within the specified time blocks (Figure 26).

5.7.4 von Bertalanffy stdev(length-at-age) correction for multi-spp/sex cases

During 2022-23, a substantial correction was made to implementing the linear relationship between length and the variance of the von Bertalanffy predictions of mean length-at-age (Davies et al. 2023). During 2023-24, this correction was extended for the cases of multiple sexes or species. The global growth variance variables for these cases were corrected to be implemented identical to that for the single species case, and tested with examples for both cases (species and sexes).

5.7.5 Resolve conflict in flag operations

Two flags were identified to have conflicting functionality, i.e., a single flag having more than one role in configuring the model.

The parest_flags(173) is defined in the Manual to specify the number of age classes for which independent mean length-at-age parameters are to be estimated, i.e., offsets from the von Bertalanffy growth curve. A second operation for this flag was identified, to activate a penalty function applied to one of the seasonal growth parameters; active in an experimental feature for age-specific diffusion that was never consolidated and was rather replaced with the current form for age-specific movement governed by: age_flags(88, 89, 90, 91, 28, 29). The second of the duplicate operations was therefore removed.

An existing feature optimises simulation projection calculations to ignore the gradient calculations during the function evaluation to substantially improve performance. The parest_flags(353) activates this feature. A second operation of this flag was identified that activates the estimation of a proxy independent variable having no functional role in the model evaluation. This is a useful tool for debugging derivative errors. This duplication was resolved by assigning the second operation to parest_flags(361).

5.7.6 Maximum index of tagged fish in pooled group

A default value is specified for the maximum number of tag recapture events per time period for the tagged fish making up the pooled group. While the number was very large and reasonable for most cases, when run in simulation mode and to improve performance, some analysts specify a very low number of periods at liberty before adding to the pooled group (say, 2 periods) which vastly increases the pooled group size. It was necessary to double the specified maximum number of tag recapture events to accommodate this special case.

5.7.7 xinit.rpt indexing due to grouped selectivity parameters

In the case where identical selectivities are assumed among specified fisheries, i.e., the fisheries are grouped and share a single set of selectivity parameters, the problem occurred where the indices in the report of the independent variables (xinit.rpt) took no account of the grouping, but rather reported those indices for each fishery. This created discontinuity among the variable's indices for the selectivity and the other variables subsequently written to the report.

A new routine was drafted, `xinit_message()`, having the grouping pointer (an i3array fishery_group_ptr) as a formal argument, and allows looping over only the first fishery within each group when incrementing the independent variable indices. The check of the xinit.rpt following the change indicated that: the index numbers and row numbers were identical; and, for the selectivity independent variables, each row has the group number labelled followed by the indices for time-blocks, seasons, and number of parameters, to assist the analyst in identifying the fisheries to which the independent variables relate.

5.7.8 SSMULT likelihood - corruption of observed sample sizes

The feature for a self-scaling multinomial likelihood (SSMULT) for size composition data was implemented in MULTIFAN-CL in 2015-16 (Davies et al. 2016). During model exploration using this feature in 2023-24, it was noticed that upon entry to the routines implementing the likelihood, the observed sample sizes were correct, but upon successive calls to the routine, the values were modified incorrectly. This was traced to a copy constructor from the observed sample sizes that created a deep copy variable, to which subsequent operations were applied, and which modified both variables. Corrections were made to remove the copy constructor, and this was extended to all SSMULT routines; with and without random effects estimation.

5.7.9 Input of simulation tagging data

For simulation projections including the generation of simulation tagging data, there may be the special case where: no grouping of tagged fish into a pooled group is specified; and, there is only a single tag release event specified during the projection time periods. In this special case, only the first tagging event holds the initial year in which the first projection tagging event occurred. The code was made robust for this instance.

5.7.10 Terminal catchabilities for eff_cond projections

Assignment of the implicit catchabilities estimated for the terminal calendar year of the estimation model is required for undertaking model projections with effort-conditioned fisheries. In the case of multiple recruitments in a calendar year, the model years do not correspond with the calendar years. To ensure the assignment is made with the correct temporal configuration, it was made robust for both cases of single (annual) or multiple recruitments in a single calendar year.

5.7.11 Generic correction to BH-SRR subset interval of temporal deviates

There is a feature allowing specification of the temporal range of the estimated recruitments to be included in the Beaverton-Holt stock-recruitment relationship (BH-SRR) regression. The analyst specifies the start and end years of the range using flag settings. The algorithm for assigning the recruitments from this range must take account of: excluding the first model year (since the recruitments in that year relate only to the assumed initial population assumptions); and, whether a lag is specified for the period between spawning and recruitment.

In the rare case where: recruitment occurs part-way through the calendar year; incomplete fisheries data is supplied for the final year; a lag>0 is specified; and, the final year is included in the BH-SRR recruitment range, the algorithm failed. A generic correction was made that adjusts the algorithm in respect of the lag and the month in which recruitment occurs.

5.7.12 Implicit catchabilities – an incidental derivative calculation

In the case of a catch-conditioned model function evaluation, it was detected that a dvar_vector for the catchabilities only implemented for the catch-errors modelling approach, was incidentally being assigned zero values. While this assignment has absolutely no impact upon the catch-conditioned fishing mortality estimation, or the function evaluation, subsequent operations using this variable would have impacted upon the order of the gradient calculations. Testing using an extremely complex and rather poorly-determined example indicated this effect altered the minimization path taken, and produced a small change in the converged solution.

The assignment to this variable was made conditional upon the method for fishing mortality estimation being only for the catch-errors method.

5.7.13 Lorenzen natural mortality function – include growth offsets

The existing derivation of the Lorenzen natural mortality function employs an “un-dimensionalised” von Bertalanffy growth formulation for the mean lengths-at-age.

$$\mu_a = \frac{L_1}{L_A} - \left(\frac{L_A}{L_A} - \frac{L_1}{L_A}\right) * \left[\frac{1-\rho^{(a-1)}}{1-\rho^{(A-1)}}\right]$$

It was found that the derivation as implemented, took no account of the case where the mean lengths-at-age of specified age classes were estimated independent of the growth curve, i.e., as growth offsets. The “un-dimensionalised” mean length-at-age in this case would be:

$$\mu_a = \frac{L_1}{L_A} - \left(\frac{L_A}{L_A} - \frac{L_1}{L_A}\right) * \left[\frac{1-\rho^{(a-1)}}{1-\rho^{(A-1)}}\right] + \frac{\delta_a}{L_A}$$

where δ_a is the growth offset for age class a. This alternative to the Lorenzen derivation was added and tested using the ALB2024 stock assessment model as an example. The effect of including the estimated offsets upon the mean length-at-age were substantial for this case (Figure 27).

6 APPLICATION OF FEATURES

6.1 Stock assessments for SC20

The 2024 stock assessment models (Hampton et al. 2024, Castillo-Jordan et al, 2024) and the stochastic projection OM models for the MSE analyses (Scott et al. 2024) were undertaken with implementation of the above-mentioned enhancements in the updated development version 2.2.7.0 of MULTIFAN-CL.

7 FUTURE WORK

A listing of the status of the work: originally proposed, that is pending, in progress, or completed during 2023-24 is provided in Table 14. While a large proportion of what was proposed, as well as unforeseen tasks, were completed, a number of tasks were not completed.

The proposed future work plan for the development of MULTIFAN-CL in 2024-25 is presented in Table 15. Those tasks not completed in 2023-24 have been carried over into this workplan. As was the case for the 2023-24 workplan, no new developments that require substantive mathematical innovation are proposed for the 2024-25 workplan. Rather, resources are directed to: consolidating those aspects that are incomplete for recent new features; enhancements of existing features; and, documentation. The general approach for the future workplan 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; outstanding tasks from the bigeye and yellowfin 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
- Provide support for WCPFC Project 123 to scope the next generation tuna model and the succession beyond MFCL.
- 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 15 (tasks rolled over from 2023-24) have been retained, but will be fit within the context of the 2024-25 workplan, and others have been set aside for the years that follow.

8 DISCUSSION

No significantly innovative or new features were implemented into the MULTIFAN-CL source code during 2023-24. Rather, the substantive features added since 2019 (e.g., catch-conditioned method for estimating fishing mortality) have been consolidated, enhanced, and extended for their implementation in population projections for MSE and TRP analyses.

The greatest achievement made in updating MULTIFAN-CL for 2023-24 was that enabling simulation projections of an operating model (OM) developed using the catch-conditioned method for estimating fishing mortality. This was a significant enhancement of the simulation mode feature, and briefly this included:

- Allowing alternatives for the terminal catchabilities (either implicit or predicted) to be assumed for the projection periods, that are applied to effort-conditioned projection fisheries to derive projection fishing mortalities;
- Developing a method using the terminal catchabilities for generating pseudo-observations of effort from catch-conditioned projection fisheries; and,
- Generating simulation CPUE pseudo-observations.

These three developments substantially improved the catch-conditioned OM for its application in MSE and target reference point (TRP) simulation studies; and are essential for providing stock assessment advice. In particular, the feature allowing detailed evaluation of the assumed terminal catchabilities in respect of those in the recent periods, say the past 3 to 5 years, is very useful. This may avoid making a spurious assumption based upon anomalous values in the terminal calendar year, and which may be avoided by using predicted values from the fitted polynomial (fml_effort_rltnshp). In this regard, the related enhancement that allows this relationship to be fitted while retaining completely intact the stock assessment diagnostic case solution, ensures no changes are made in conditioning the OM for its use in projection analyses. These tools have been well-defined, tested, and are easy to implement, now making the simulation mode feature well-equipped for undertaking MSE and TRP projects.

The second-most important area of development during 2023-24 was a thorough consolidation of the CPUE likelihood, both in its formulation and implementation in MULTIFAN-CL. This development was significant because of the important role of relative abundance index data in the WCPO tuna stock assessment models. The re-formulation of the non-concentrated likelihood to include the constant term ensures this likelihood is now formally correct, and is consistent, in respect of both its formulation and absolute magnitude, with those of the other data types in the integrated likelihood, (that all include a constant term). Of similar importance was the re-formulation of the concentrated CPUE likelihood to express time-variant precision in the form of a normalised deviate on the assumed error, $\boldsymbol{\sigma}$, such that the mean error over all indices making up the time series is equal to the assumed value. Transparency is therefore accorded when transitioning among the two likelihood formulations that illustrates the effect of the index-specific estimates of precision in the model fit. The third re-formulation of the likelihood related to the stationary catchability assumption, that affects the estimated distribution of regional biomass. This formulation enables this assumption to be made, but while allowing for differences in the relative precision among the fisheries from each region. During 2023-24, this re-formulation was extended to the concentrated likelihood. Finally, some small changes to the practical implementation of the CPUE likelihood have reduced the potential for input error, and make transitioning between alternative formulations easier. Cumulatively, these developments serve to consolidate the CPUE likelihood in MULTIFAN-CL.

The third set of enhancements offer assistance to the stock assessment analyst when undertaking model development. The independent variables may now be readily reviewed during model exploration for the most appropriate parameter configuration. Those contributing most to the high gradients, or being estimated at their bounds, can be easily identified. Imposing prior assumptions on the estimated regional distribution of recruitments is now possible when employing the orthogonal-polynomial recruitment parameterisation. Obtaining estimates of parameter uncertainty for the key quantities of management interest is now a relatively trivial task compared to earlier versions of MULTIFAN-CL. These represent firm steps towards making the use of MULTIFAN-CL easier for the analyst.

Numerous corrections were made during 2023-24, some of which may be considered enhancements of existing features that lacked certain aspects (e.g., likelihood components report – consolidated all terms, Lorenzen natural mortality function – include growth offsets, and non-decreasing penalty function for time-variant selectivities), while others corrected errors that positively impacted upon the minimization solutions obtained from models developed using earlier versions (e.g., implicit catchabilities – an incidental derivative calculation, SSMULT likelihood – corruption of observed sample sizes). All these, and the other corrections, are an on-going task in refining and improving the software, that seeks to ensure the integrity and credibility of the stock assessment models developed from it.

All the enhancements and corrections were accompanied by detailed testing with tuna stock assessment model examples, with the outcomes and implications illustrated and described. This confirms the developments to have been correctly implemented, also they were accompanied by abbreviated benchmark testing to assess their implications on the function evaluation and dependent variable estimates. These are described in sections 2.6.2 to 2.6.6. While not fully benchmark tested, these tests of the effects specific to each development, confirm it has not negatively impacted on other existing features employed in tuna and billfish models developed for the 2024 stock assessments. These tests also provide reference points when future stock assessments are undertaken using the current, or future, versions of MULTIFAN-CL. Typically, the starting point of a stock assessment is to replicate the previous assessment model solution using the most recent MULTIFAN-CL version. Interpreting the differences with respect to the previous solution is well assisted by making reference to the tests performed in this, and previous, update reports.

A number of the tasks set out in the 2023-24 workplan were replaced by higher priority tasks introduced during the year as were required or considered important for the 2024 stock assessments. This flexibility in implementing the workplan is a necessary feature in order to ensure the production version, i.e., the version being applied currently to produce stock assessment models for delivering advice to the WCPFC, has all the functionality required to implement the necessary assumptions, parameter configurations, and to obtain well-determined solutions. Consequently, those lower priority tasks for 2023-24 are carried over to the 2024-25 workplan.

The recommendations of an independent peer review panel convened in 2022, (Punt et al., 2023) were added to the workplan for 2023-24, and carried over to the 2024-25 plan. Some of the recommended tasks will draw heavily upon the project's resources for development, implementation and testing. An important first step in the coming year, will be to prioritise the list of tasks to identify those of immediate priority for the 2025 stock assessments.

As mentioned, the developments in 2023-24 were restricted to consolidating recent features (e.g., catch-conditioned projections), enhancing existing features, improvements to processes and reporting, and making corrections. The workplan for 2024-25 continues with this theme, but strongly aims to address the much-needed task of documentation and project administration. In the past 5 years, new developments have progressed at a rapid pace, and therefore, the User's Guide for MULTIFAN-CL requires substantial updates. This is considered critically important as many new flag settings vital to stock assessment model development, remain undocumented, and so the risk of errors in model inputs and flag settings is increasingly high.

The continuation of limited improvements to the support structures of the MULTIFAN-CL project in the past 3-5 years, is an ongoing concern. Substantially fewer of the project's resources are available (due to Dr Fournier's retirement), and lower priority continues to be assigned to this area, usually because of the importance of preparing features for production assessments. The support structures are now very neglected. An obvious consequence of this, is that versions since (and including) 2.0.7.0 have not yet been posted on the website, and documentation of significant new features, viz. the catch-conditioned method, are not included in the Manual. The proposed 2024-25 workplan (section 7), seeks to address this situation, but this will be subject to the priority-setting for the immediate future; and accounting for the requirements for the 2025 stock assessments.

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., Day, J., Teears, T., Davies, N., Hampton, J., Magnusson, A., Vidal, T., Williams, P. and Hamer, P. 2024. Stock Assessment of Striped Marlin in the Southwest Pacific Ocean: 2024. WCPFC-SC20/SA-WP-03. Manila, Philippines, 14-21 August 2024

Davies, N., Fournier, D., Bouyé, F., and Hampton, J. 2016. Developments in the MULTIFAN-CL software 2015-16. WCPFC-SC12-2016/SA-IP-10. Bali, Indonesia, 3 – 11 August 2016.

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

Davies, N., Fournier, D., Bouyé, F., and Hampton, J. 2022. Developments in the MULTIFAN-CL software 2021-22. WCPFC-SC18-2022/SA-IP-03. Electronic Meeting.

Davies, N., Fournier, D., Bouyé, F., Hampton, J. and Magnusson, A. 2023. Developments in the MULTIFAN-CL software 2022-23. WCPFC-SC19-2023/SA-IP-02. Koror, Palau, 16 – 24 August 2023.

Day, J., Magnusson, A., Teears, T., Hampton, J., Davies, N., Castillo Jordan, C., Peatman, T., Scott, R., Scutt Phillips, J., MacKenzie, S., Scott, F., Yao, N., Pilling, G., Williams, P., and Hamer, P. 2023. Stock assessment of bigeye tuna in the Western and Central Pacific Ocean: 2023. WCPFC-SC19-2023/SA-WP-05. Koror, Palau, 16 – 24 August 2023

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. 2023. Report from the SPC pre-assessment Workshop – April 2023. WCPFC-SC19-2023/SA-IP-01. Koror, Palau, 16 – 24 August 2023

Teears, T., Hampton, J., Davies, N., Castillo-Jordan, C., Day, J., Magnusson, A., Peatman, T., Xu, H., Vidal, T., Williams, P., Pilling, G., and Hamer, P. Stock assessment of South Pacific albacore: 2024. WCPFC-SC20-SA-WP-02.

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/

Magnusson, A., Day, J., Teears, T., Hampton, J., Davies, N., Castillo Jordan, C., Peatman, T., Scott, R., Scutt Phillips, J., MacKenzie, S., Scott, F., Yao, N., Pilling, G., Williams, P., and Hamer, P. 2023. Stock assessment of yellowfin tuna in the Western and Central Pacific Ocean: 2023. WCPFC-SC19-2023/SA-WP-04. Koror, Palau, 16 – 24 August 2023

Pacific Community. 2022. Evaluations to support decisions on the WCPO skipjack tuna target reference point based upon the 2022 stock assessment. WCPFC19-2022-10

Punt, A.E., Maunder, M.N., and Ianelli, J.N. 2023. Independent review of recent WCPO yellowfin tuna assessment. WCPFC-SC19-2023/SA-WP-01. Koror, Palau, 16 – 24 August 2023

Scott, R., Scott, F., Yao, N., Hamer, P. and Pilling, G. 2024. Selecting and conditioning operating models for south Pacific albacore. WCPFC-SC20/MI-WP-04, Manila, Philippines, 14-21 August 2024

10 TABLES

Table 1. Number of fishing incidents (in time periods) occurring within the terminal calendar of the SKJ2022 estimation model, for those fisheries being either effort- or catch-conditioned in the projection periods.

Effort-conditioned fishery	2	5	8	12	14	15	19	20	25	26	29	30
Terminal year no. incidents	4	4	4	4	3	4	1	4	4	4	4	4
Catch-conditioned fishery	1	4	7	13	18	22	24	28				
Terminal year no. incidents	2	1	2	4	4	2	2	2				

Table 2. Component terms of the non-concentrated and concentrated CPUE likelihoods from single model evaluations with fixed parameters and equivalence in the assumed σ_k. For the concentrated form λ_i were all set to 1. Comparisons are made for the cases: survey fisheries are grouped; and, ungrouped (compares fishery 18 only).

	lambda	SSQ	vhat_wtd	Constant	Total Survey Like	Total likelihood all data types
Grouped survey fisheries						
non-concentrated	-	113.5217	908.1737	-998.132	-89.9581932	-187693.2681639860
concentrated	1	113.5217	908.1737	-998.132	-89.9581932	-187693.2681639860
Ungrouped survey fisheries – fishery 18 values shown						
non-concentrated	-	35.95829	287.6663	-332.711	-45.0444	-187699.5265310980
concentrated	1	575.3326	287.6663	-332.711	-45.0444	-187699.5265310980

Table 3. Dependent variables and likelihood terms of models employing the non-concentrated (nonconc_pen_eq) and concentrated (conc_eq_lambda_1) CPUE likelihoods from the first minimisation test that applies equivalence in the assumed σ_k. For the model employing the concentrated CPUE likelihood form λ_i were all set to 1.

Model quantity	nonconc_pen_eq	conc_eq_lambda_1	%diff
MSY	86350	86590	0.28
Fmsy	0.181	0.181	0.00
B0	1115000	1118000	0.27
Bmsy	477100	478400	0.27
Bcurr	689022	689025	0.00
SBmsy	147100	147500	0.27
SBcurr	354886	354886	0.00
Bcurr.Bmsy	1.444	1.440	-0.27
SBcurr.SBmsy	2.413	2.406	-0.27
SBcurr.SBcurrF0	0.592	0.592	0.00
SBlatest.SBlatestF0	0.528	0.528	0.00
obj_lencomp	-192317.523	-192317.523	0.00
obj_cpue	57.887	57.887	0.00
Obj	187751.330	187751.330	0.00
gradient	0.0000573	0.0000552	-3.60

Table 4. Components of the CPUE non-concentrate likelihood, and the total integrated log-likelihood, obtained from deterministic evaluations of the BET2023 model with (vsn2250) or without (vsn2220) the constant term in the CPUE likelihood formulation.

	vsn2220	**vsn2250**
Sums of squares term	1047.83821793	1047.83821793
Constant term		-3452.39900119
CPUE term total	1047.83821793	-2404.56078326
Total integrated likelihood	-1027530.20448548	-1030982.603486671695

Table 5. Model quantities of interest, likelihood components and estimated growth parameters for the converged BET2023 solutions obtained excluding (nonconc_pre) and including (nonconc_post) the constant term in the CPUE non-concentrated likelihood; with the percentage difference (%diff) among the solutions.

Model quantity	**nonconc_pre**	**nonconc_post**	**%diff**
MSY	36200	37150	2.62
Fmsy	0.044	0.048	9.03
Bmsy	829600	780900	-5.87
Bcurr	1301127	1237749	-4.87
SBmsy	505900	456300	-9.80
SBcurr	849811	785422	-7.58
SBcurr.SBcurrF0	0.348	0.349	0.26
SBlatest.SBlatestF0	0.346	0.347	0.35
obj_bhsteep	0.428	0.357	-16.53
obj_lencomp	-74470.824	-74375.539	-0.13
obj_wtcomp	-962237.027	-962407.418	0.02
obj_tagdata	6324.599	6336.886	0.19
obj_agelngdata	1572.449	1601.329	1.84
obj_cpue	1051.070	-2406.220	-328.93
Obj	1027463.608	1030962.951	0.34
Lmin	22.999	23.723	3.15
Lmax	146.409	148.991	1.76
K	0.117	0.112	-4.54

Table 6. Model quantities of interest, likelihood components and estimated growth parameters for the converged solutions of YFT2023 obtained excluding (nonconc_pre) and including (nonconc_post) the constant term in the CPUE non-concentrated likelihood; with the percentage difference (%diff) among the solutions.

Model quantity	nonconc_pre	nonconc_post	%diff
MSY	169500	169500	0.00
Fmsy	0.073	0.073	0.00
Bmsy	2327000	2327000	0.00
Bcurr	4623833	4624061	0.00
SBmsy	1064000	1064000	0.00
SBcurr	2385485	2385656	0.01
SBcurr.SBcurrF0	0.454	0.454	0.00
SBlatest.SBlatestF0	0.429	0.429	0.00
obj_bhsteep	0.314	0.314	0.00
obj_lencomp	-154972.922	-154972.916	0.00
obj_wtcomp	-610341.254	-610341.250	0.00
obj_tagdata	13215.327	13215.317	0.00
obj_agelngdata	2479.579	2479.577	0.00
obj_cpue	732.908	-1155.370	-257.64
Obj	748630.016	750518.297	0.25
Lmin	19.800	19.800	0.00
Lmax	141.959	141.959	0.00
K	0.132	0.132	0.00

Table 7. Model quantities of interest, likelihood components and estimated growth parameters for the converged solutions of SKJ2022 obtained excluding (nonconc_pre) and including (nonconc_post) the constant term in the CPUE non-concentrated likelihood; with the percentage difference (%diff) among the solutions.

Model quantity	nonconc_pre	nonconc_post	%diff
MSY	608900	606900	-0.33
Fmsy	0.244	0.244	0.04
Bmsy	2492000	2483000	-0.36
Bcurr	5102615	5081202	-0.42
SBmsy	1098000	1094000	-0.36
SBcurr	3198075	3180303	-0.56
SBcurr.SBcurrF0	0.519	0.519	0.07
SBlatest.SBlatestF0	0.498	0.499	0.06
obj_bhsteep	0.131	0.131	-0.45
obj_lencomp	400559.4432	401153.228	0.15
obj_tagdata	31300.762	31284.125	-0.05
obj_cpue	1629.860	31.697	-98.06
Obj	-440125.141	-438528.412	-0.36
Lmin	23.176	23.178	0.01
Lmax	85.555	85.370	-0.22
K	0.203	0.203	0.44

Table 8. Assumed error values for the non-concentrated CPUE likelihood term expressed as penalty weights (penwt), with the conversion to the best approximation of σ to within two decimal places, and showing the effect on the likelihoods for each survey fishery of the ALB2024 example model.

Fishery	penwt*100	derived σ	CPUE likelihood		Input σ	CPUE likelihood
18	1859	0.164001	-27.6232		0.16	-27.4845
19	1837	0.164980	-21.1822		0.16	-21.1058
20	1978	0.158991	10.5853		0.16	9.965038
21	1250	0.200000	-6.88347		0.20	-6.88347
22	833	0.244998	-23.1501		0.24	-23.2371
23	1043	0.218949	-19.2928		0.22	-19.2736
24	2164	0.152004	-20.6851		0.15	-20.1567
Total			-108.23156859759			-108.17607790707

Table 9. Estimated mean distribution of recruitments among regions for the models: excluding (no_pen) and including (orthp_pen) the orthp_reg_recrs_pen penalty term in the integrated negative log-likelihood (penalty weight = 10,000), relative to the prior assumed values (prior distribution).

	Proportion region 1	Proportion region 2
no_pen	0.112	0.880
prior distribution	0.741	0.259
orthp_pen	0.740	0.260

Table 10. Sensitivity of the predicted mean distribution of recruitments among regions of the orthp_pen model over a range of multiplicative weighting values (w), relative to the specified prior values of 0.741015 and 0.258985 for regions 1 and 2, respectively. The value of the orthp_reg_recrs_pen penalty term is shown.

w	Penalty term	Proportion in Region_1 : Region_2
1	0.795	0.110 : 0.890
10	5.322	0.225 : 0.775
100	0.592	0.687 : 0.313
1000	0.075	0.735 : 0.265
10000	0.001	0.740 : 0.260

Table 11. Example of the detailed independent variables report: indepvar.rpt.

```
Index  Var_name   Estimate    L_bound  U_bound  gradient
1 diff_coffs                              1.060563      1e-06   1000   6.0386655e-07
2 diff_coffs                              0.33677164    1e-06   1000   -5.0429655e-07
3 diff_coffs                              0.062614188   1e-06   1000   -7.7439031e-07
...
2241 implicit_fm_level_regression_pars(30,7)   0.23161927   -500   500   -1.3252495e-07
2242 implicit_fm_level_regression_pars(30,8)   0.012382473  -500   500   1.6661377e-06
2243 sv(21)                               0.14476858    1e-06   1000   5.3416969e-06
2244 age_pars(5)                          -0.70912659   -20   2   -1.2837603e-05
2245 age_pars(5)                          -1.1268347    -20   2   -9.2155344e-06
2246 age_pars(5)                          -0.86757428   -20   2   -8.2239655e-06
2247 age_pars(5)                          -0.84007807   -20   2   -5.6710682e-06
2248 age_pars(5)                          -1.244723     -20   2   -6.7582681e-08
2249 vb_coff(1)                                  22.76083     10    30    7.8976279e-06
2250 vb_coff(2)                                  83.86536     70    130   -2.0026656e-06
2251 vb_coff(3)                                  0.21499091   0.05  0.4   4.750589e-06
2252 var_coff(1)                                 7.5627276    1     12    1.2928132e-06
2253 var_coff(2)                                 0.59949421   0     3     -1.2709221e-06
```

Table 12. Model quantities of interest, likelihood components and estimated growth parameters for the converged solutions of the three ALB2024 models that explore the diagcs_intct_fml feature: diagcs_clipd; seas_poly_fml_3degs_term; and, intct_fml_term; with the percentage difference in respect of the diagcs_clipd model for: the seas_poly_fml_3degs_term model (%diff_poly3); and, the intct_fml_term model (%diff_intct).

Model quantity	diagcs_clipd	seas_poly_fml _3degs_term	intct_fml_term	%diff _poly3	%diff _intct
MSY	88530	87490	88530	-1.17	0.00
Fmsy	0.159	0.157	0.159	-1.13	0.00
Bmsy	557400	557400	557400	0.00	0.00
Bcurr	800555	801505	800555	0.12	0.00
SBmsy	150900	146900	150900	-2.65	0.00
SBcurr	268551	257153	268551	-4.24	0.00
SBcurr.SBcurrF0	0.408	0.397	0.408	-2.69	0.00
SBlatest.SBlatestF0	0.450	0.438	0.450	-2.51	0.00
obj_bhsteep	8.233	8.739	8.233	6.14	0.00
obj_lencomp	-45907.997	-45909.214	-45907.997	0.00	0.00
obj_agelngdata	2030.970	2031.716	2030.970	0.04	0.00
obj_cpue	-110.992	-101.149	-110.992	-8.87	0.00
obj_impfml	0.000	125.073	181.742	-	-
Obj	-43972.290	-43837.416	-43790.548	-0.31	-0.41
No. parameters	253	344	91	35.97	-64.03
gradient	0.0000644	0.0002859	0.0000006	344.22	-99.04
Lmin	45.538	45.538	45.538	0.00	0.00
Lmax	100.953	101.171	100.953	0.22	0.00
K	0.361	0.353	0.361	-2.36	0.00

Table 13. Comparisons of the function evaluation total integrated negative log-likelihood (Integr_tot_like) as reported to the screen output, versus the sum of the component terms reported to the "test_plot_output" report (Sum_compnts_report) in respect of the absolute differences (Abs_diff) and the percentage differences (%diff) for the example models: YFT2014, BET2023, YFT2023, SKJ2022, BET2017_ssmult, and SWO2021_2sex.

Model example	Integr_tot_like	Sum_compnts_report	Abs_diff	%abs_diff
YFT2014	-1193780.43908	-1193780.43821	0.0008688	0.00000007%
BET2023	-1030852.37671	-1030852.37662	0.0000858	0.00000001%
YFT2023	-750516.46035	-750516.46069	0.0003372	0.00000004%
SKJ2022	438528.41183	438528.41161	0.0002192	0.00000005%
BET2017_ssmult	-454091.705379619	-454091.705379623	0.000000004	8.84e-13%
SWO2021_2sex	-54102.0511180384	-54102.0511180445	0.000000006	1.13e-11%

Table 14. Modifications to MULTIFAN-CL with respect to their state of completion as of July 2024.

2011 Bigeye Tuna 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%
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 (* those added to the workplan after July 2023)		
Task	Description	Status of completion
Catch-conditioned model projection	Generation of simulation stochastic CPUE data	Development 100%; Testing 100%
	Obtain predictions of the effort associated with catch-conditioned projection fisheries	Development 100%; Testing 100%
	Consolidate methods for the catch-conditioned calculation of fishing mortalities and catchabilities, for their utility in projections	Development 100%; Testing 100%
	OM conditioned maintained intact but with estimation of the fml_effort_rltnshp regression*	Development 100%; Testing 100%
CPUE likelihood consolidation	Add constant term to non-concentrated formulation; revise concentrated formulation with normalised λ_i; variable assumed error for fisheries within grouping for stationary q; make consistent the assumed error input to both formulations*	Development 100%; Testing 100%
Likelihood components report	Establish consistency in total likelihood components report with screen output values; ensure production is robust to model operations	Development 100%; Testing 100%
Orthogonal-polynomial penalty*	Allow a constraint on the estimated regional recruitment distribution	Development 100%; Testing 100%
Independent variables diagnostics	Generate a report of the parameter estimates; indices; character labels; estimation bounds; and gradients	Development 100%; Testing 100%
Natural mortality*	Lorenzen natural mortality formulation to take account of independent growth offsets	Development 100%; Testing 100%
Variance report*	Add MSY- and depletion-related quantities to the variance report of dependent variables	Development 100%; Testing 100%
Viewer	Ensure is compatibility with multi-sex input and report file formats	Development 100%; Testing 100%

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%
MSE Team support*	Ongoing support for the MSE Team in transitioning to the catch-conditioned OM	Complete 100%
macOS compilation	Develop an arrangement for the Mac Mini PC to be connected to the Pacific Community computing network, to facilitate compilations for macOS and testing of the executable versions.	Development 100%; Testing 100%
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 of the projection period as required for deriving economics-based indicators.	Development 0%; Testing 0%
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 15. Proposed workplan for MULTIFAN-CL in 2024-25 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.	2023-24
d. Tags inform movement	Testing for the case of multi-species/stocks/sexes.	2023-24; Development 100%; Testing 90%
Developments carried over from 2023-24		
Task	Description	Implementation
Catch-conditioned method	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.	2024-25
	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.	2024-25
	Document the catch-conditioned method, and flags for Manual	2024-25
	Fishing mortality estimation for multi-species	2024-25
CPUE likelihood	Review the formulation of the concentrated log-normal likelihood formulation as used in CASAL (Bull et al. 2012) for its suitability for implementation in MULTIFAN-CL.	2024-25
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	2024-25
User's Guide	Documentation of the catch-conditioned method and flags	2024-25
Outstanding testing of existing features	Multi-sex model projections	2024-25
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-2024-25
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-2024-25
Stochastic projection functionality	- implement process error in future recruitments with application of the derived autocorrelation coefficient in historical recruitment estimates - stochastic variability in terminal numbers at age - consolidate the generation of stochasticity in effort_dev_coffs	post-2024-25
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-2024-25
Recruitment random effects	Report on the feasibility of its implementation in MULTIFAN-CL.	post-2024-25
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-2024-25
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	post-2024-25
Length-based selectivity	Resolve the anomalies that produce undesirable discrepancies in predicted size compositions.	post-2024-25

Self-scaling multinomial with random effects	Complete draft paper for peer review.	post-2024-25
Recruitment correlates	Region-specific environmental recruitment correlates estimated within the orthogonal polynomial parameterisation for recruitments	post-2024-25
Movement correlates	Add a time-series structure (e.g., random walk, time blocks or using environmental correlates) to movement coefficients	post-2024-25
Recruitment deviate penalties	Allow for time-variant penalties on recruitment deviate estimates	post-2024-25
Tagging multi-sex	Account for instances of differences between size composition the tag releases and the sex-specific populations	post-2024-25
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-2024-25
Report comments	Add comment descriptions of the selectivity parameter configurations in the output .par and .rep reports	post-2024-25
Developments for 2024-25		
Task	Description	Implementation
Catch-conditioned method	Enable the estimation of selectivity deviate coefficients sel_dev_coffs	2024-25
Growth estimation	Implement the linear relationship for standard deviation of mean length-at-age and length for the Richards parameterisation and for multi-species/sex instances	2024-25
BH-SRR	Ensure robustness when steepness approaches a value = 1	2024-25
Terminal constant recruitment	Add the option for specifying the range of years for calculating the mean value to which the terminal recruitments are constrained	2024-25
Variance estimation	Restrict the variance estimation to a defined subset of time series dependent variables	2024-25
Tasks from the Yellowfin Tuna Independent Peer Review Panel 2022		
Length-weight variance	Extend MULTIFAN-CL so that variability in weight-at-length can be taken into account	2024-25
Selectivity splines	Extend MULTIFAN-CL so that it is possible to specify the number of spline knots when defining selectivity and where they are located with respect to age (length) as the current approach means that the selectivity for some knots is constrained to zero.	2024-25
Age-reading error	Extend MULTIFAN-CL so that account can be taken of age-reading error when fitting to conditional age-at-length data.	2024-25
CPUE overdispersion	Add the ability to specify overdispersion in CPUE as an additive rather than multiplicative factor.	2024-25
Natural mortality at age	Integrate the calculation of M-at-age from the sex-ratio data into MULTIFAN-CL unless a sex-specific assessment is used.	2024-25
Tasks newly added		
PDH diagnostics	Refine and describe the algorithm of positive definite Hessian (PDH) diagnostics to be done for an assessment; add a unique identifier to the Hessian file, therefore its parallelised components, and possibly to the *.par, to ensure continuity among them when stitching parallel *.hes files.	post-2024-25
Evaluations report	Report the number of function evaluations completed during the minimisation in an output file.	post-2024-25
Simulation pseudo-observations	Correction to the generation of simulation pseudo-observations of tagging data for the catch-conditioned model	post-2024-25

Variance report	Resolve the discrepancy between F_{recent}/F_{msy} in the dependent variables variance report and the Fmult in the plot.rep report	post-2024-25
Tagging multi-sex	Account for instances of differences between size composition the tag releases and the sex-specific populations	post-2024-25
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-2024-25
Report comments	Add comment descriptions of the selectivity parameter configurations in the output .par and .rep reports	post-2024-25

11 FIGURES

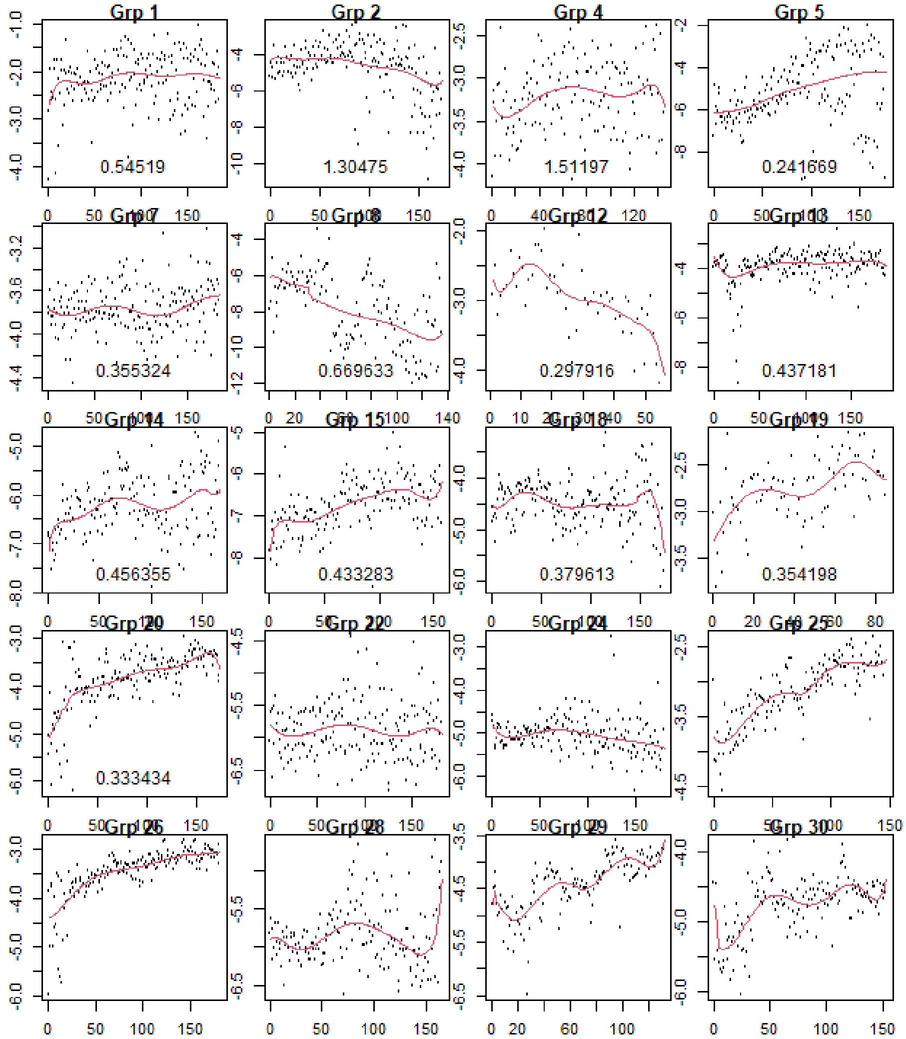

Figure 1. The fitted fml_effort_rltnshp regressions over all time periods for the 20 fisheries of the SKJ2022 example model.

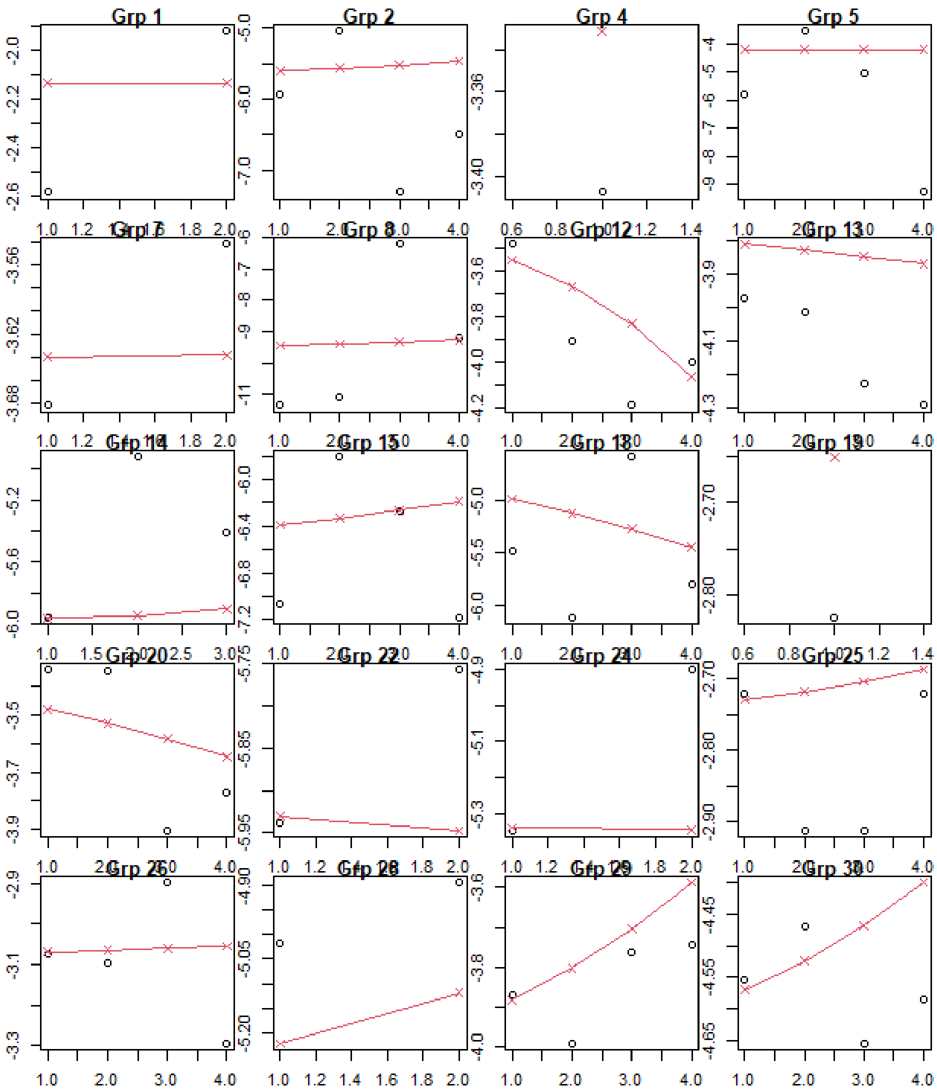

Figure 2. The fitted fml_effort_rltnshp regressions for the 20 fisheries of the SKJ2022 example for the estimation model time periods in the terminal calendar year only; with the regression predictions – red line and crosses; and, "observed" model values – black circles.

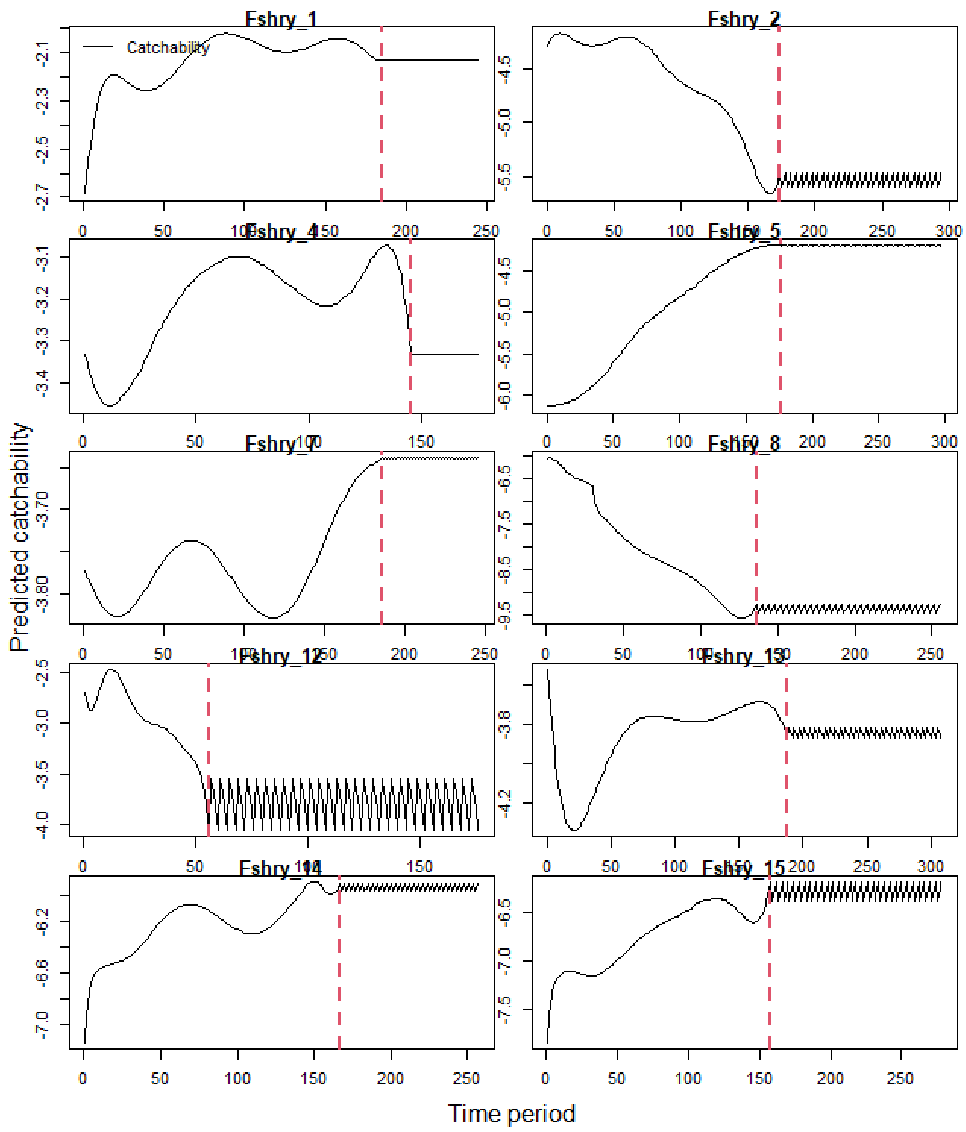

Figure 3. Fishery-specific time series of the predicted catchabilities from the fml_effort_rltnshp regression as estimated during the estimation model time periods, and those predicted for the terminal year and assumed for the constant catchabilities during the projection periods (the Proj_fml model). Time period of the first projection year indicated by the vertical dashed red line.

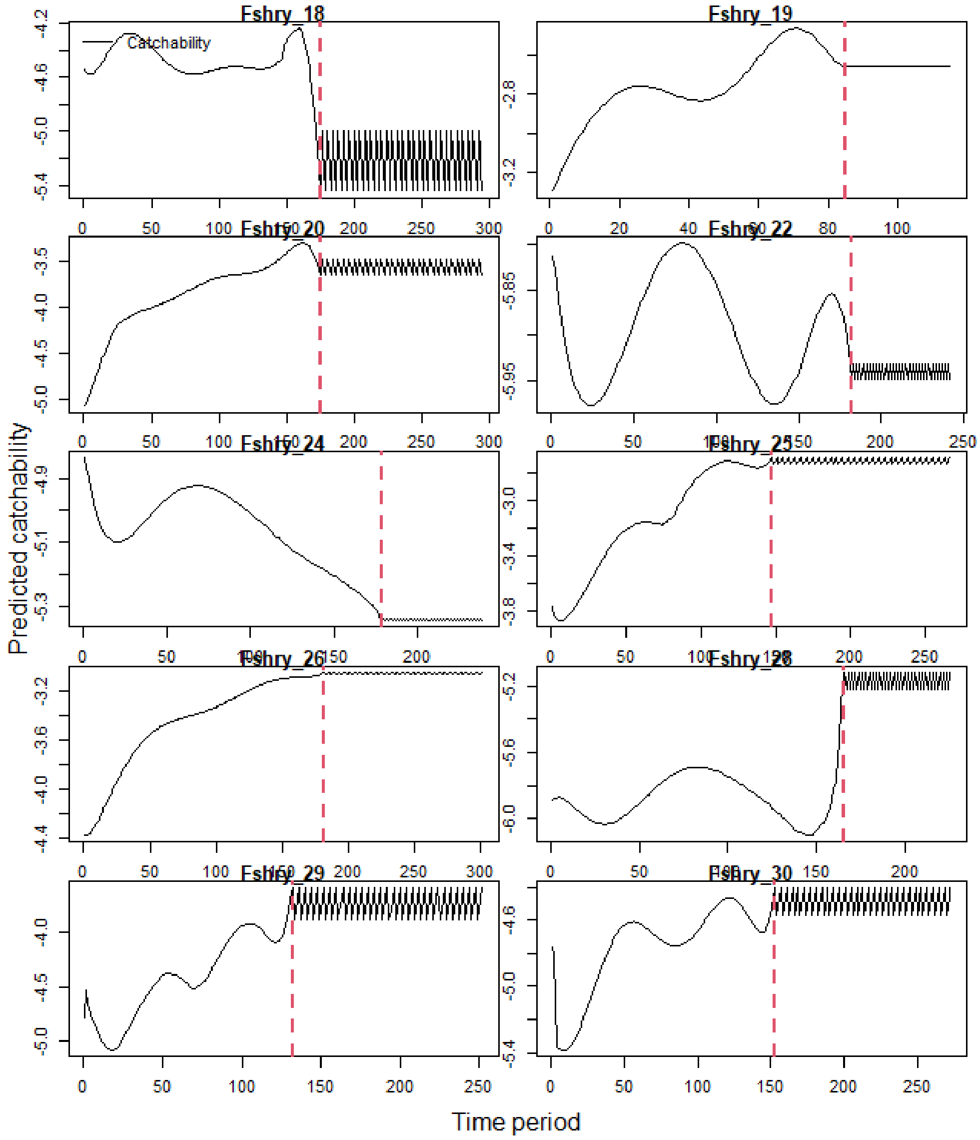

Figure 3. cont.

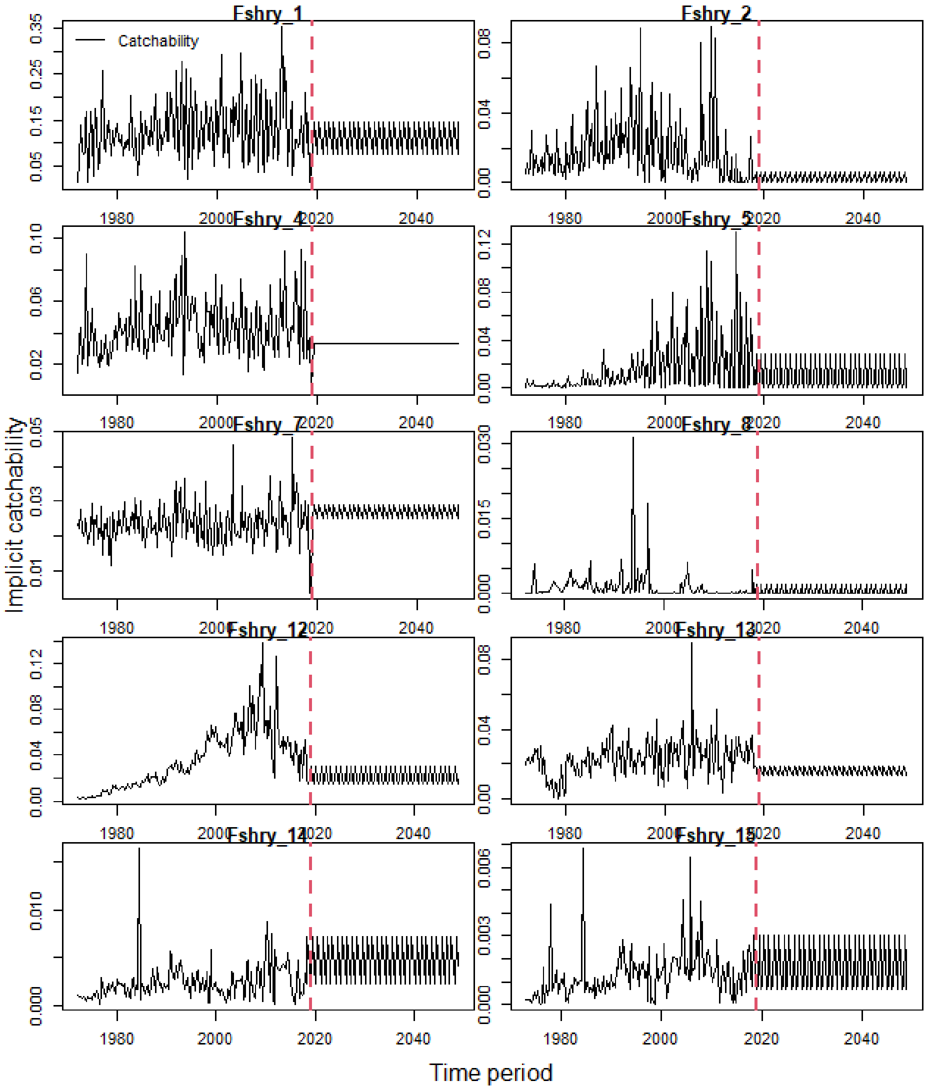

Figure 4. Fishery-specific time series of "observed" (empirical) catchabilities as estimated during the estimation model time periods, and those of the terminal year assumed for the constant catchabilities during the projection periods. Time period of the first projection year indicated by the vertical dashed red line.

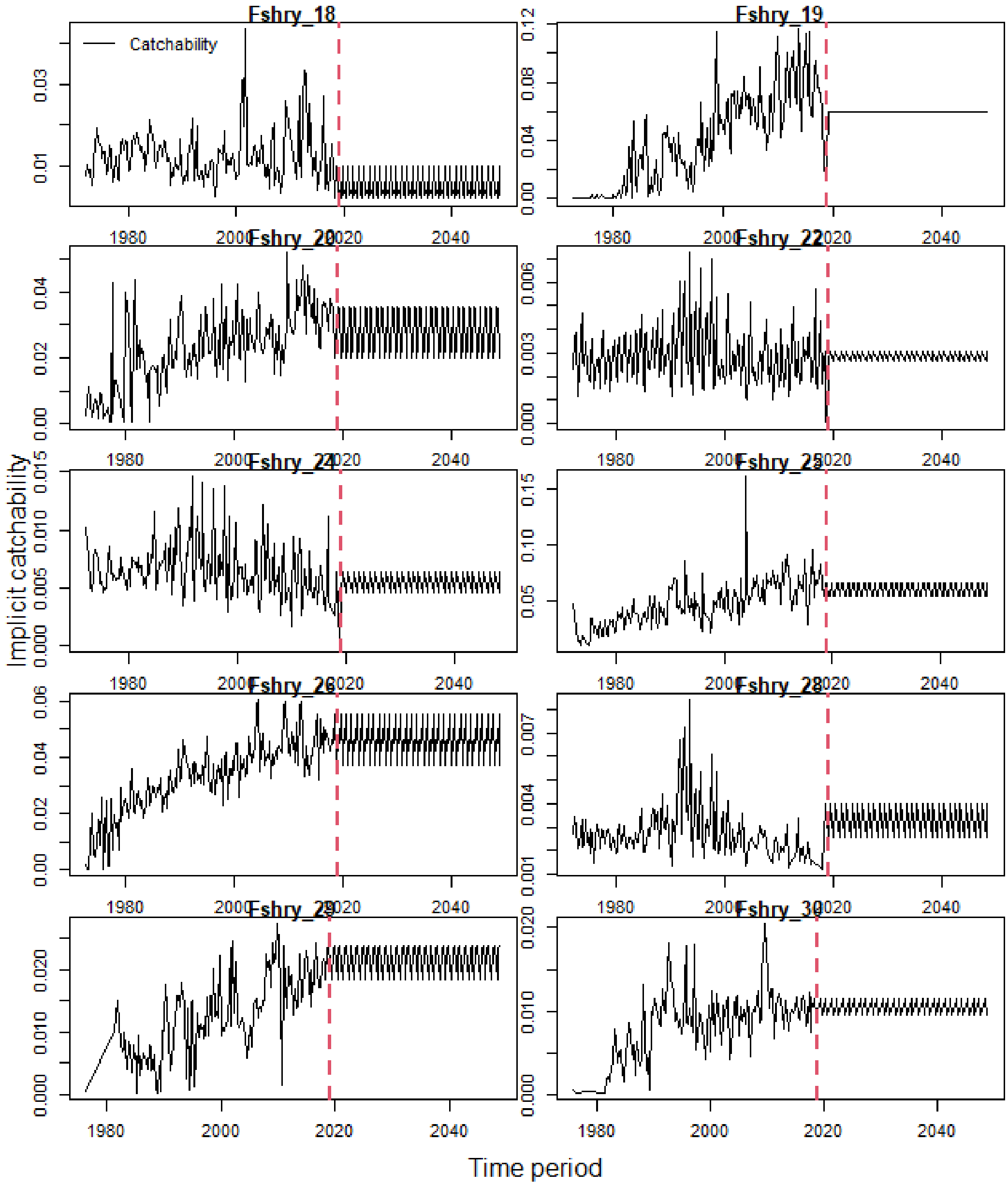

Figure 4. cont.

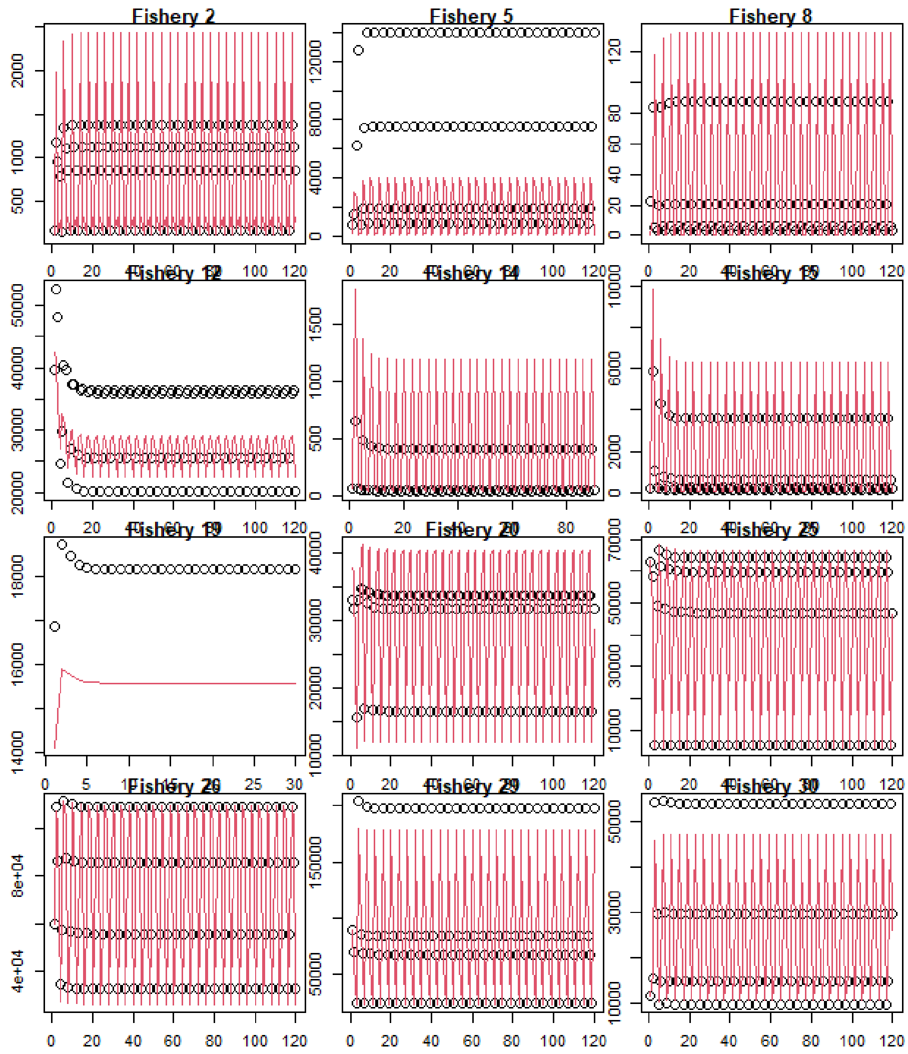

Figure 5. Comparison of the deterministic projection catches for effort-conditioned fisheries as predicted for: the Proj_fml model (employs fml_effort_rltnshp catchability predictions, black circles); and, the Proj_qterm model (employs "observed" model terminal year catchabilities, red lines).

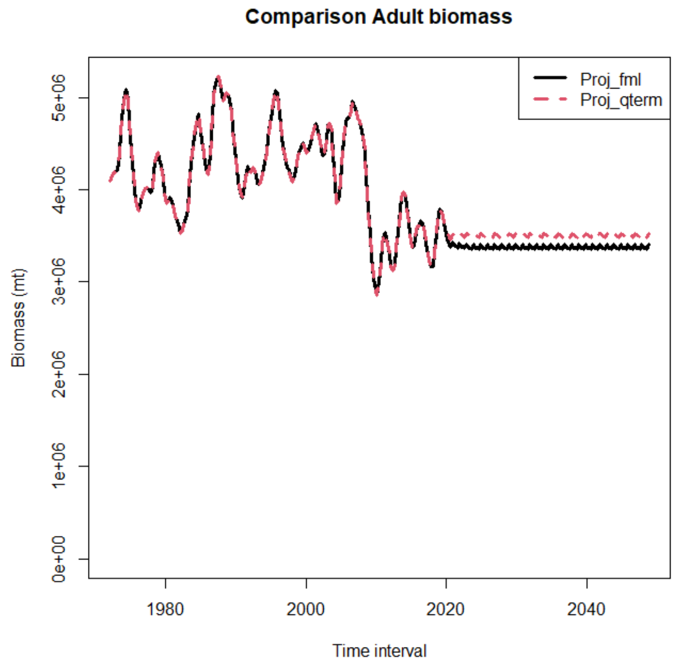

Figure 6. Spawning biomass as predicted by deterministic projection for: the Proj_fml model (employs fml_effort_rltnshp catchability predictions, black line); and, the Proj_qterm model (employs "observed" model terminal year catchabilities, dashed red lines). First projection year is 2019.

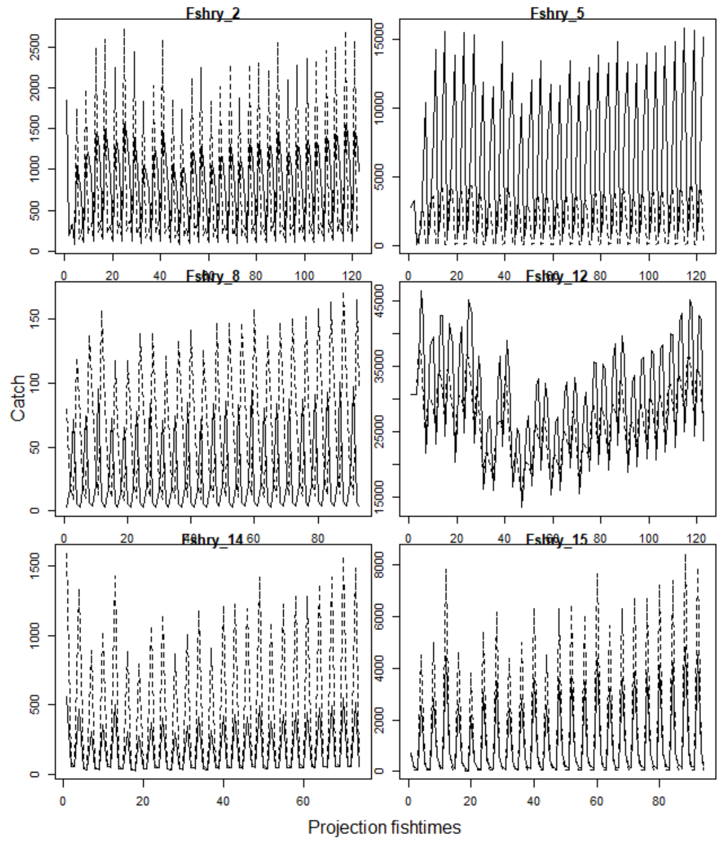

Figure 7. Predicted catches (without pseudo-observation error) by stochastic projection of the effort-conditioned fisheries for: the Proj_fml model (employs fml_effort_rltnshp catchability predictions, solid line); and, the Proj_qterm model (employs "observed" model terminal year catchabilities, dashed line) for the first of three simulations.

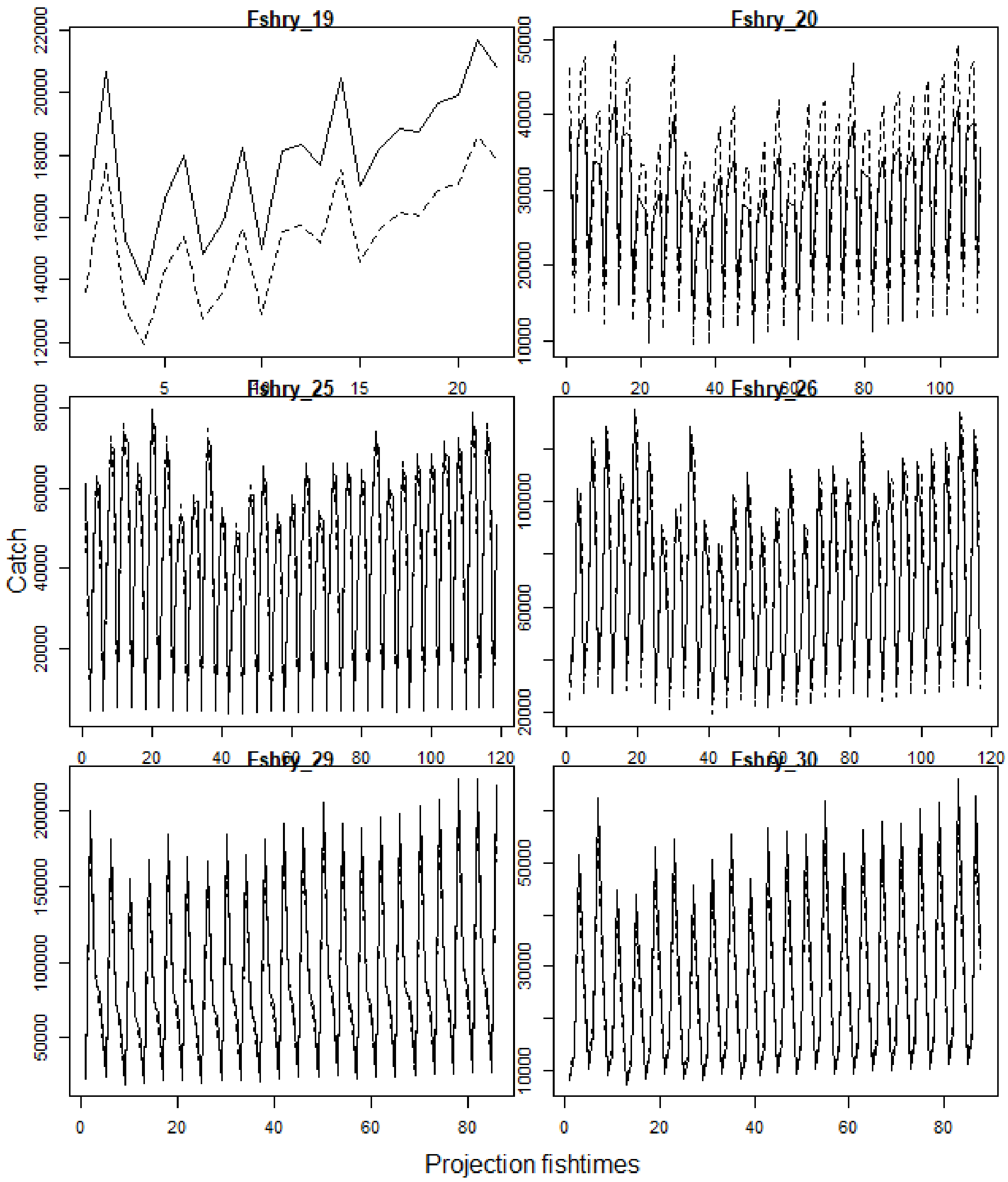

Figure 7. cont.

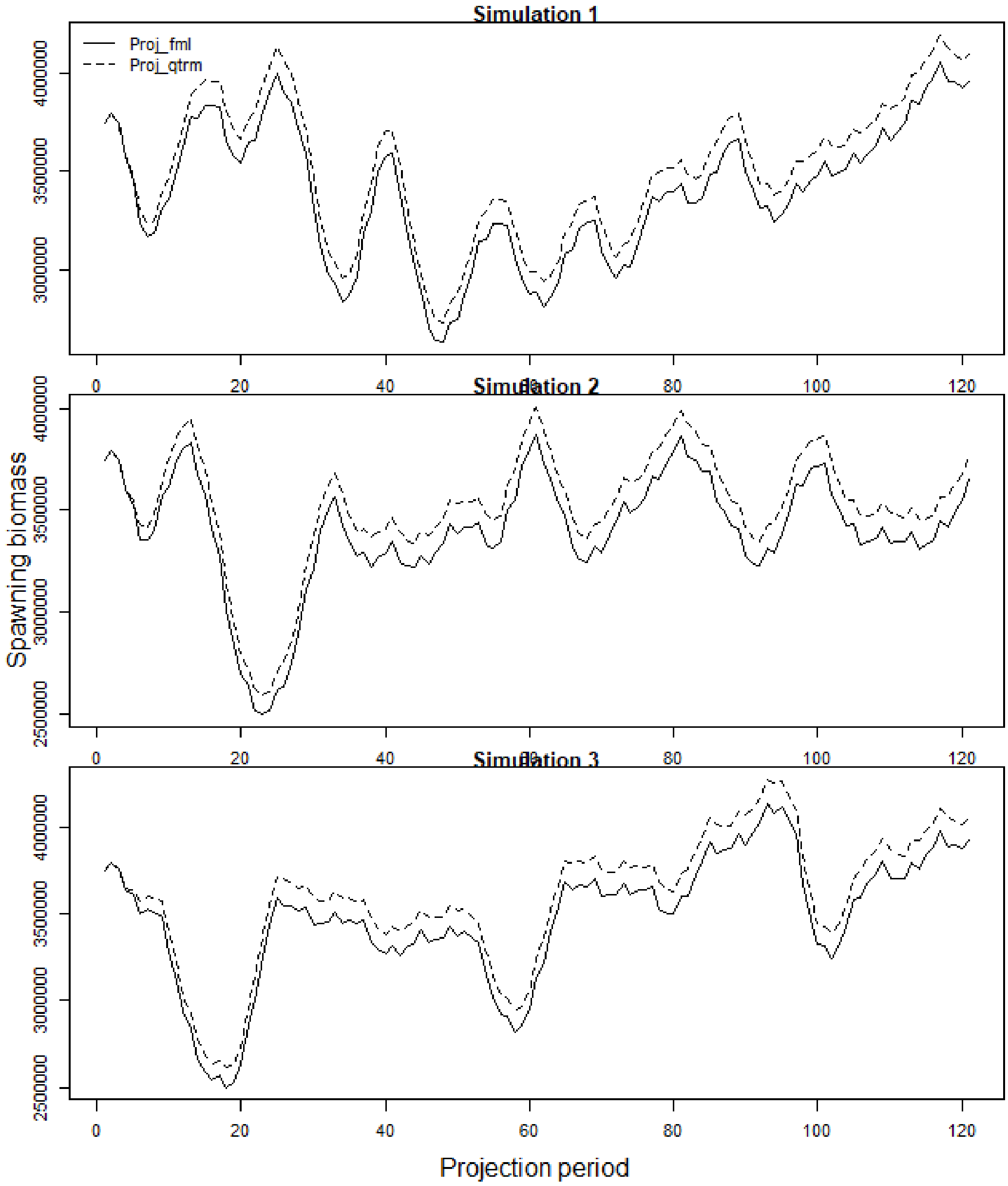

Figure 8. Predicted spawning biomass by stochastic projection for: the Proj_fml model (employs fml_effort_rltnshp catchability predictions, solid line); and, the Proj_qterm model (employs "observed" model terminal year catchabilities, dashed line) for three simulations.

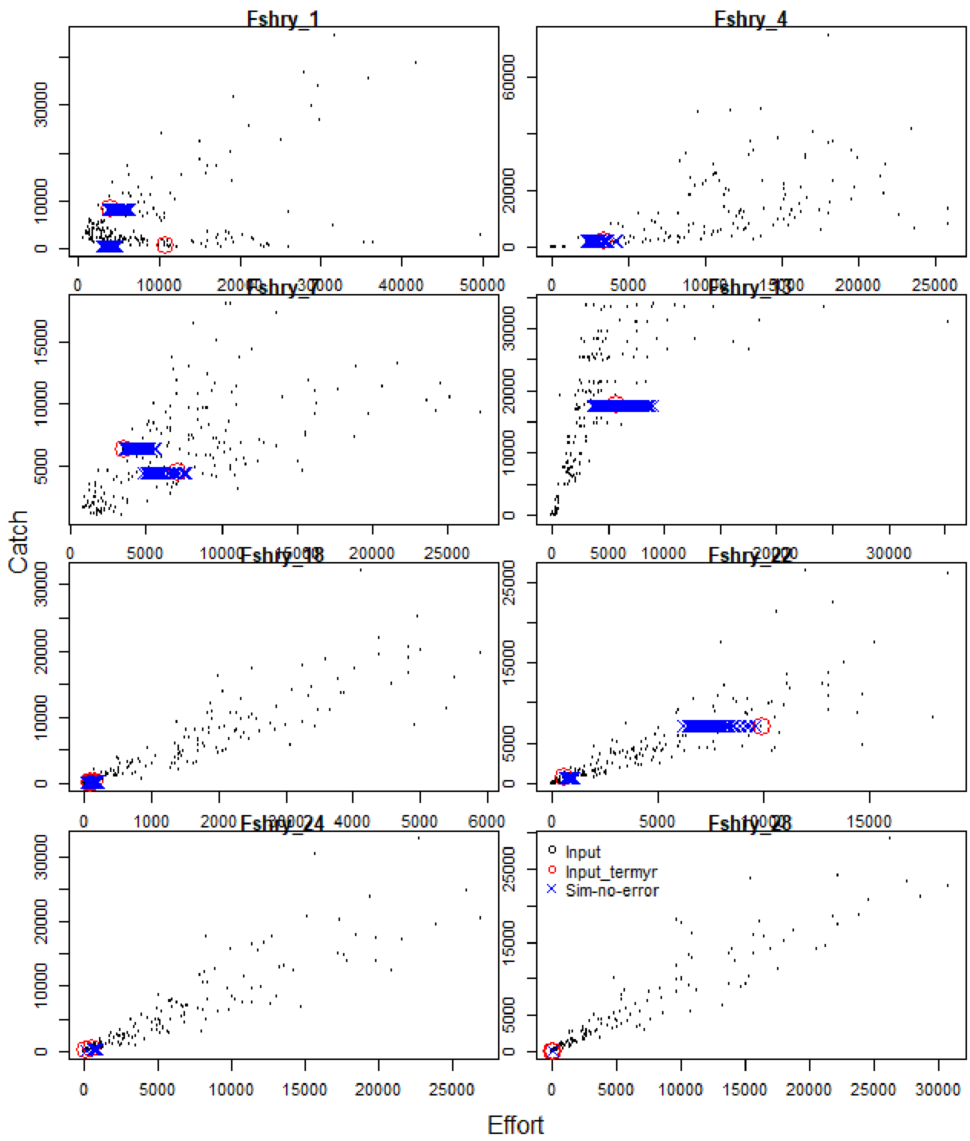

Figure 9. Catch and effort of the catch-conditioned projection fisheries for: the estimation model time periods (Input); those for the terminal calendar year (Input_termyr); and, the predictions without pseudo-observation error for the projection time periods (Sim-no-error).

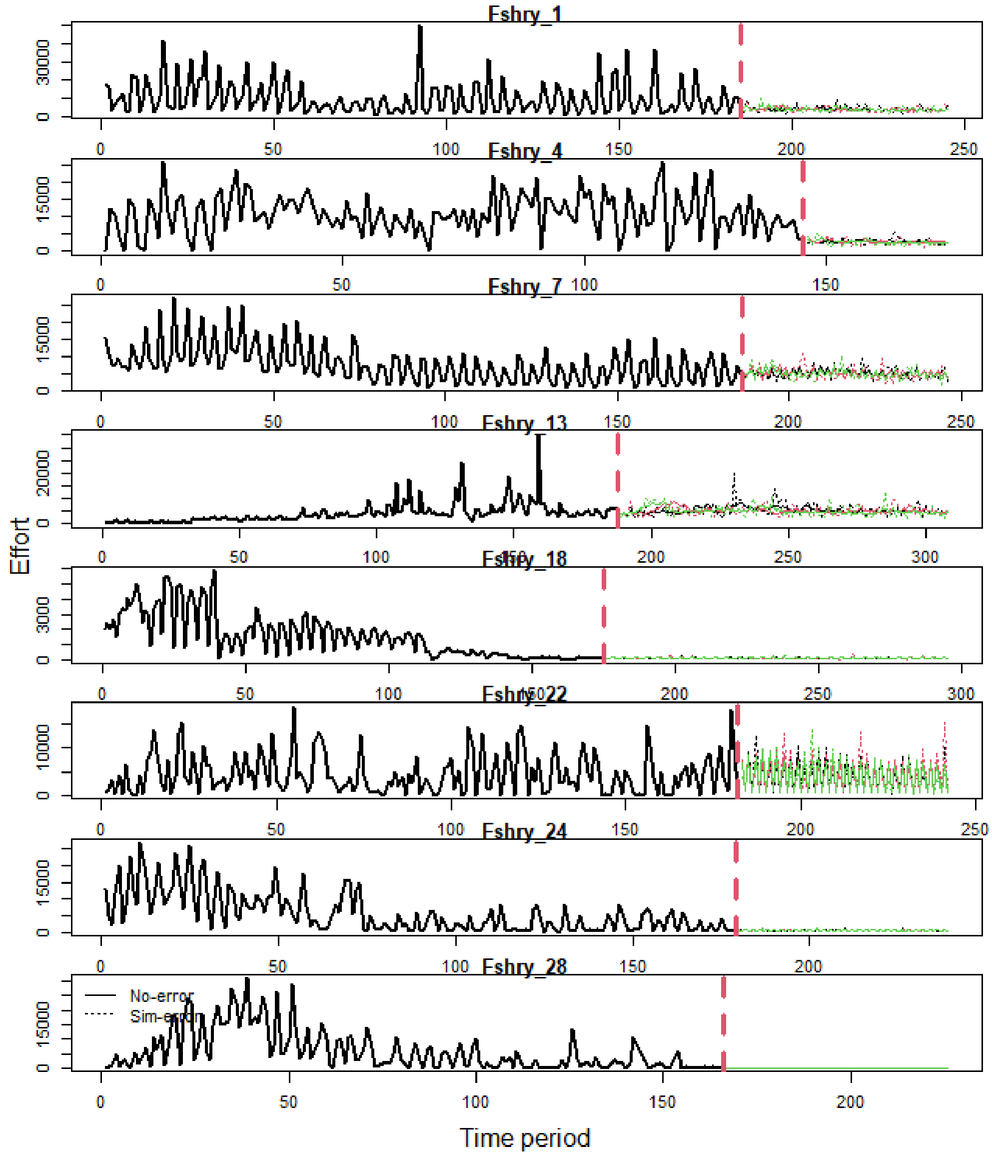

Figure 10. Observed effort for the estimation periods, and predicted effort from three simulations (with and without error) for the catch-conditioned fisheries during the projection period. The terminal estimation time period is indicated by the vertical dashed red line. Simulation pseudo-observations of projection effort are indicated by the solid and dashed black, red and green lines.

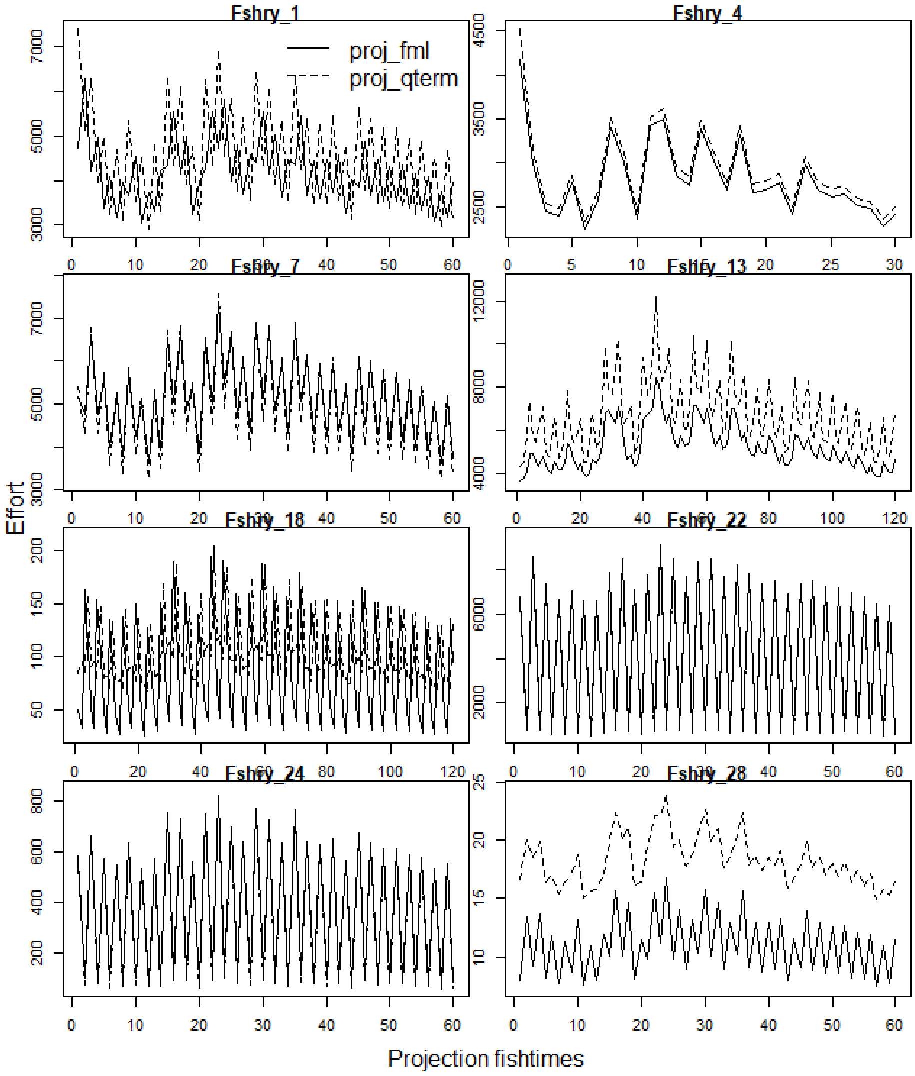

Figure 11. Predicted effort (without pseudo-observation error) by stochastic projection of the catch-conditioned fisheries for: the Proj_fml model (employs fml_effort_rltnshp catchability predictions, solid line); and, the Proj_qterm model (employs "observed" model terminal year catchabilities, dashed line) for the first of three simulations.

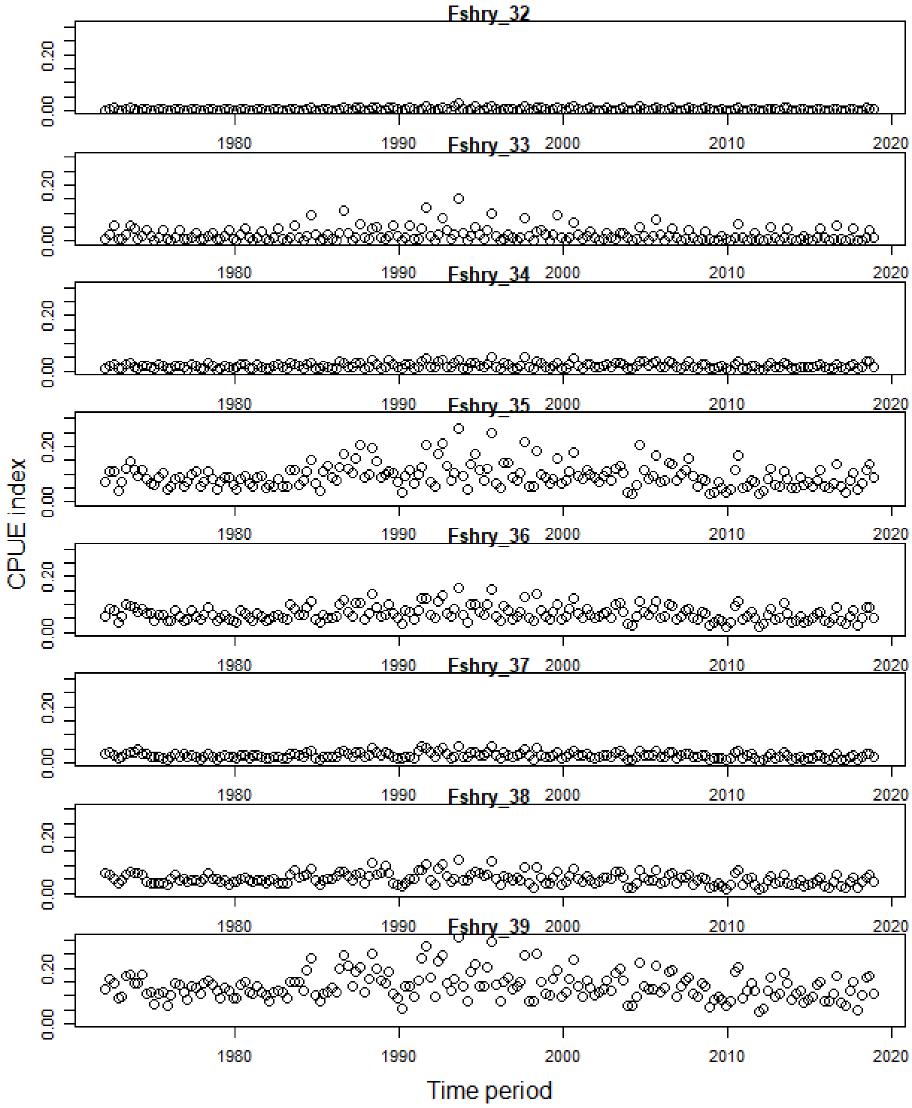

Figure 12. Observed CPUE indices for each of the eight survey fisheries in the SKJ2022 example.

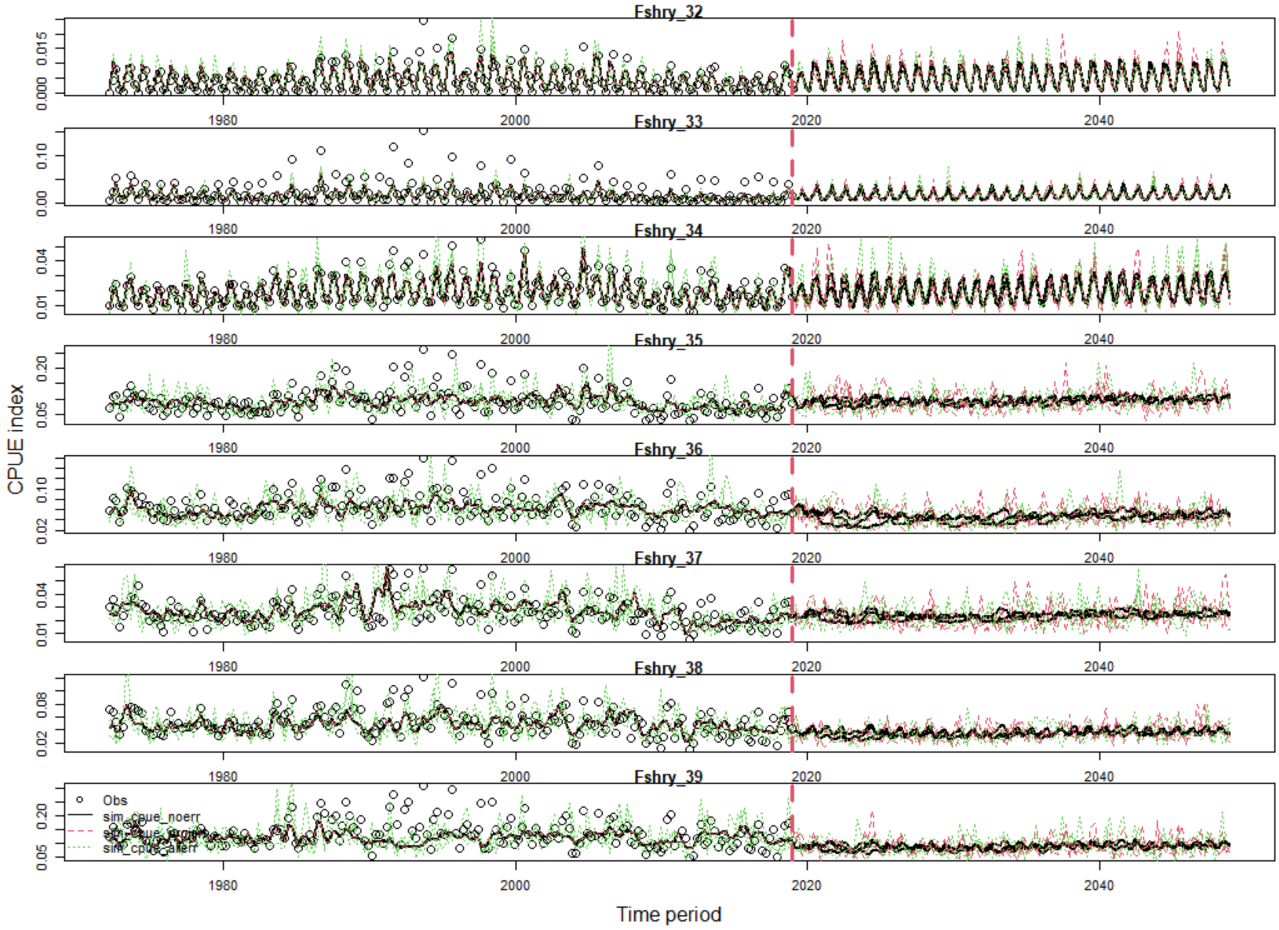

Figure 13. Comparison of the three pseudo-observation error scenarios over three simulations, relative to the input CPUE observations (Obs).

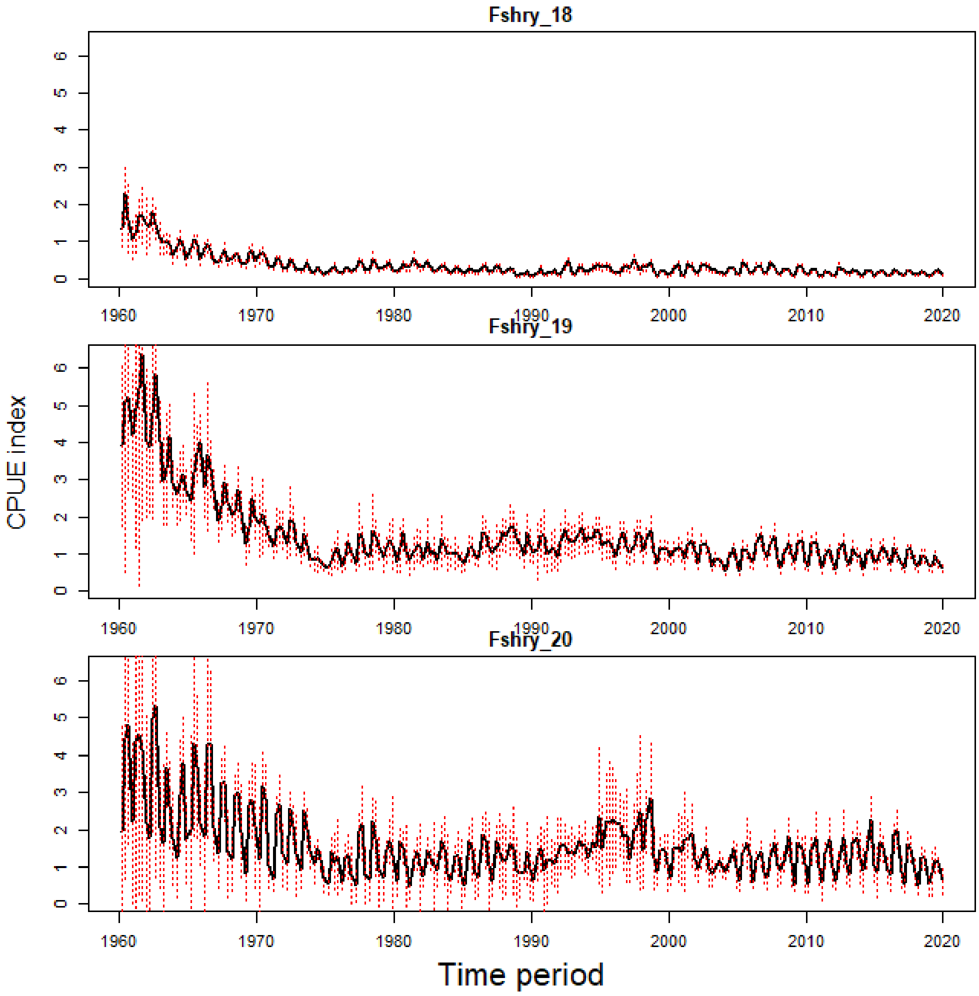

Figure 14. Observed CPUE indices for the ALB2021 3-region model for survey fisheries 18 to 20 in the three respective regions.

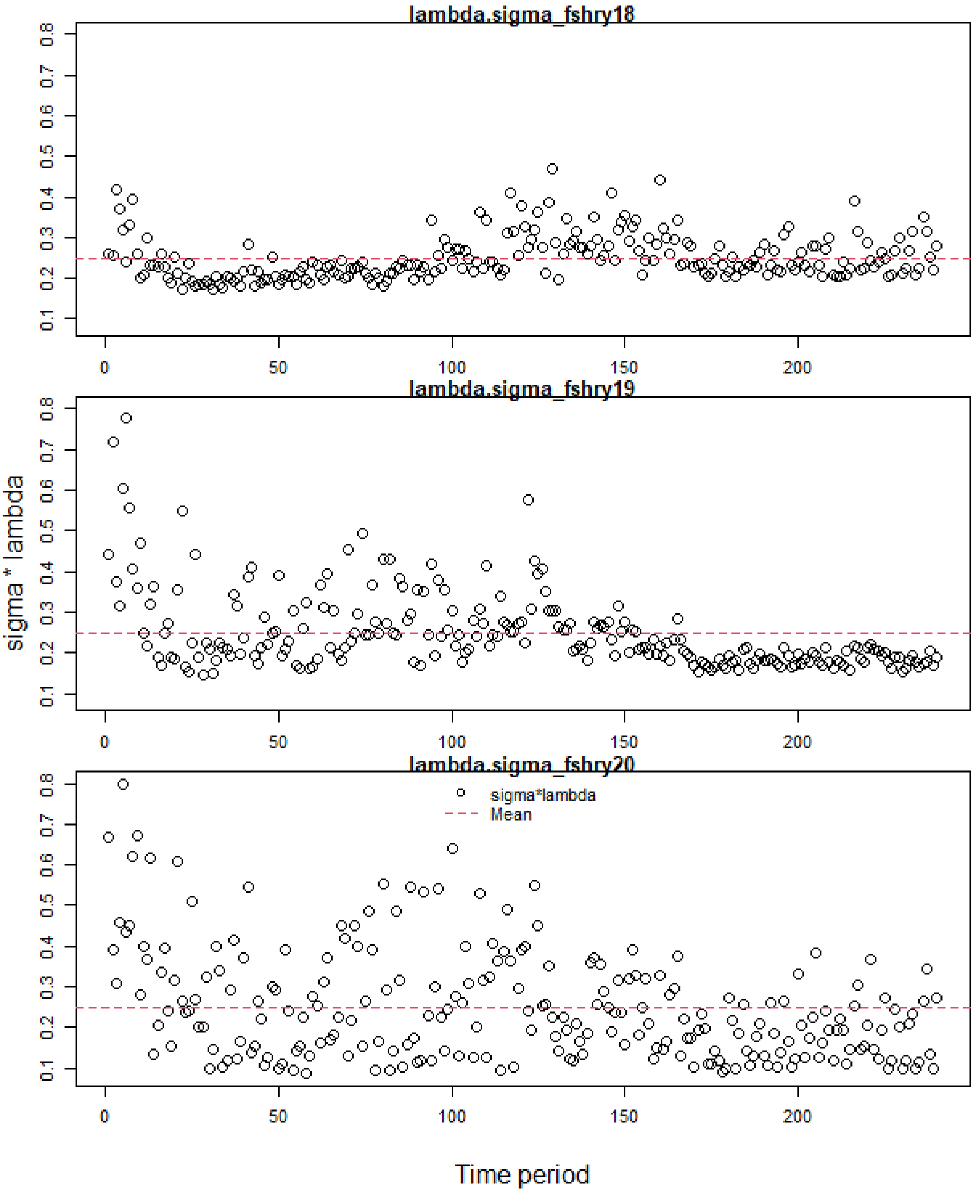

Figure 15. The product of the assumed σ_k = 0.25 and λ_i (sigma*lambda) with the mean of the product (Mean) for each of the survey fisheries; illustrating the consistency among the means and the assumed σ_k.

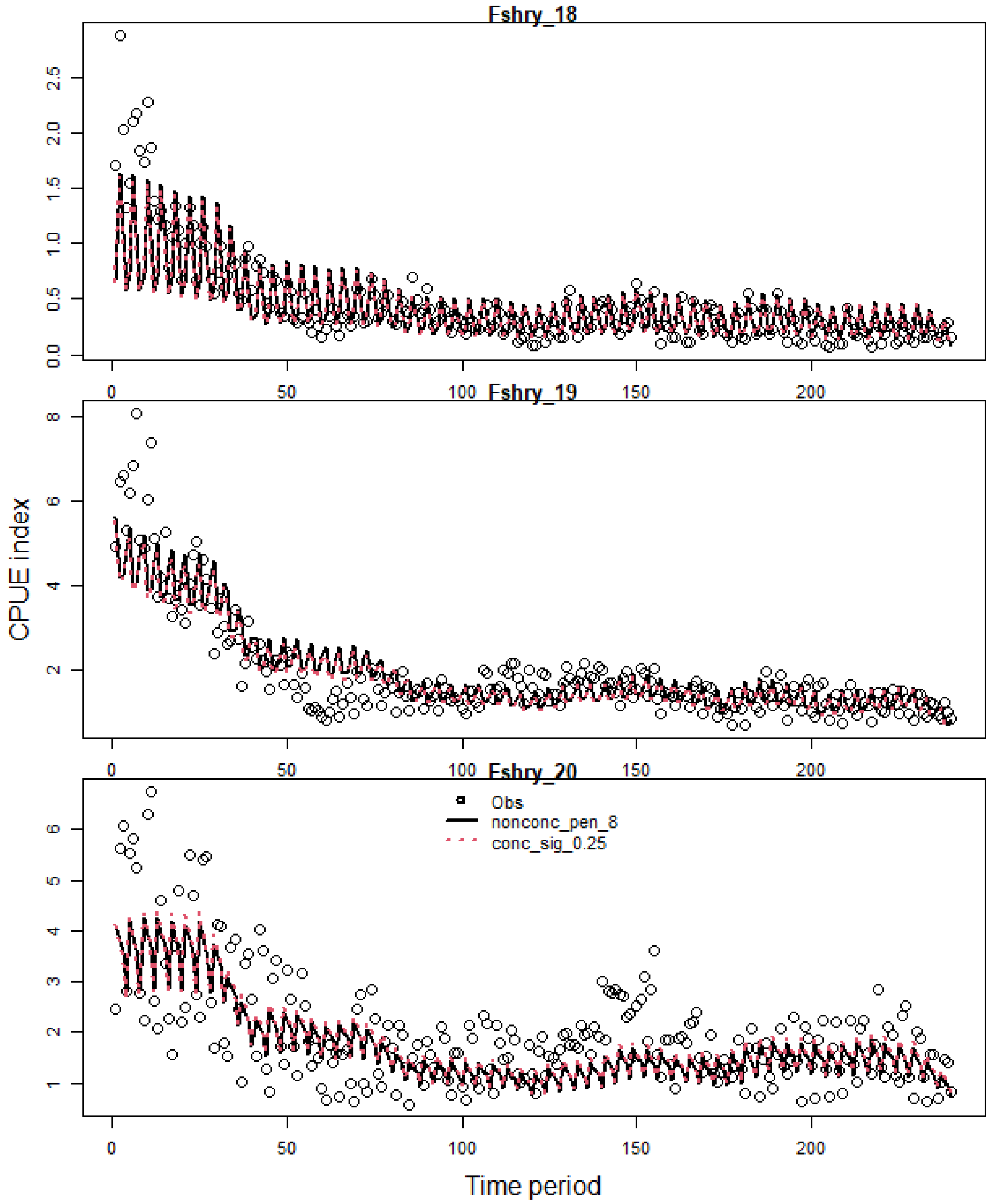

Figure 16. Fits to the observed CPUE time series of the non-concentrated (nonconc_pen_8) and concentrated (conc_sig_0.25) likelihood models with equivalent σ terms.

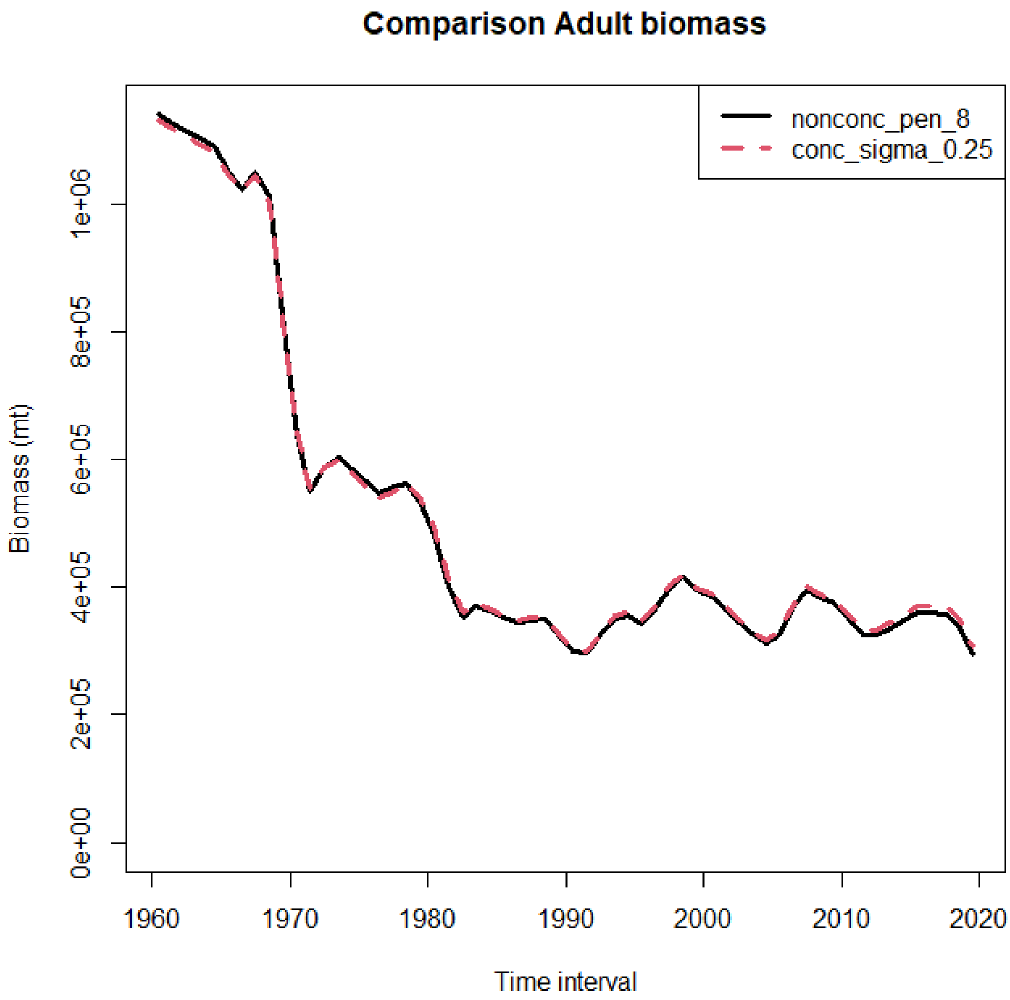

Figure 17. Derived variable: total adult biomass among all regions as estimated by the non-concentrated (nonconc_pen_8) and concentrated (conc_sigma_0.25) likelihood models with equivalent σ terms.

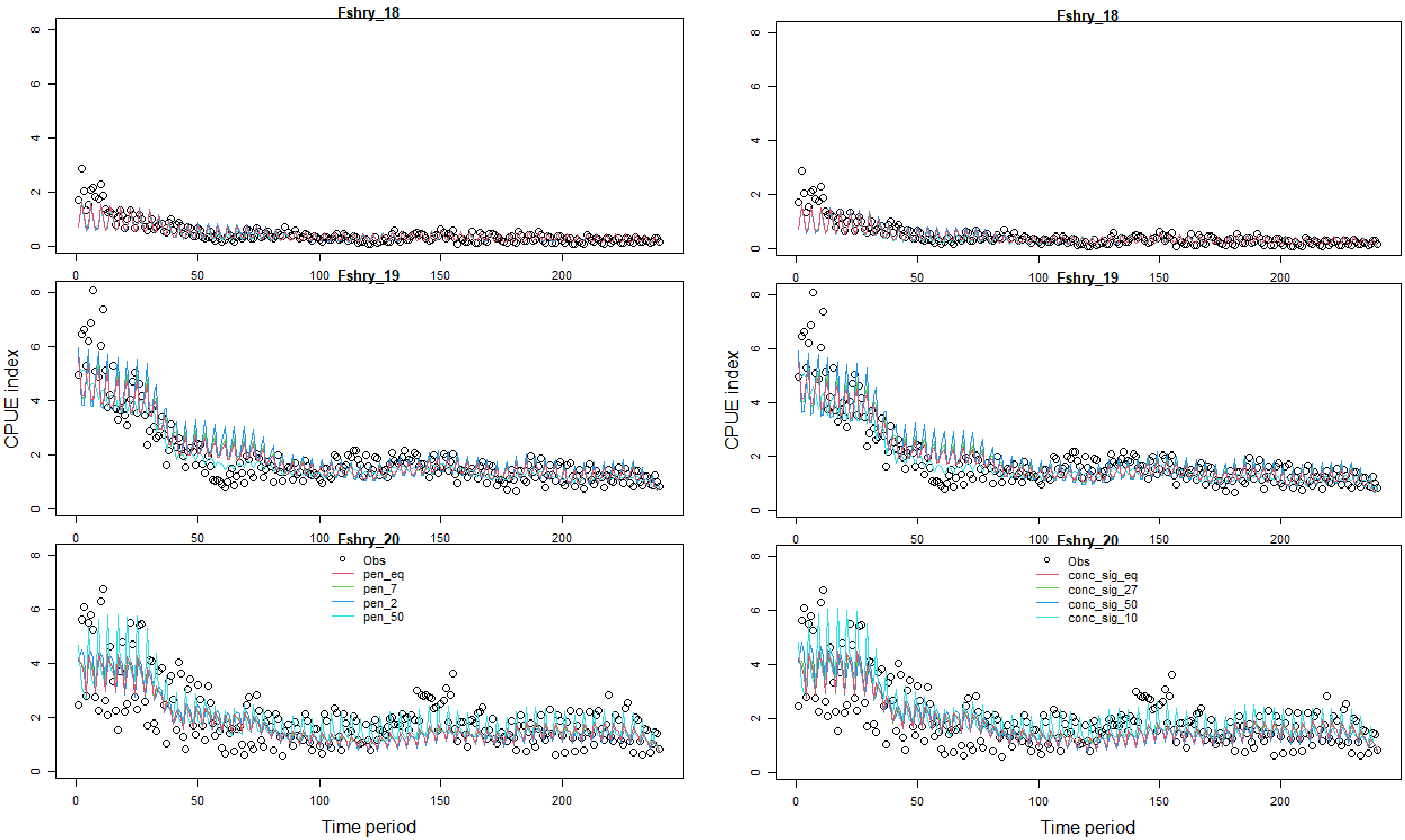

Figure 18. Fits of the non-concentrated likelihood models (left panel) and concentrated likelihood models (right panel) to the observed CPUE time series, and the equivalent range of assumed error.

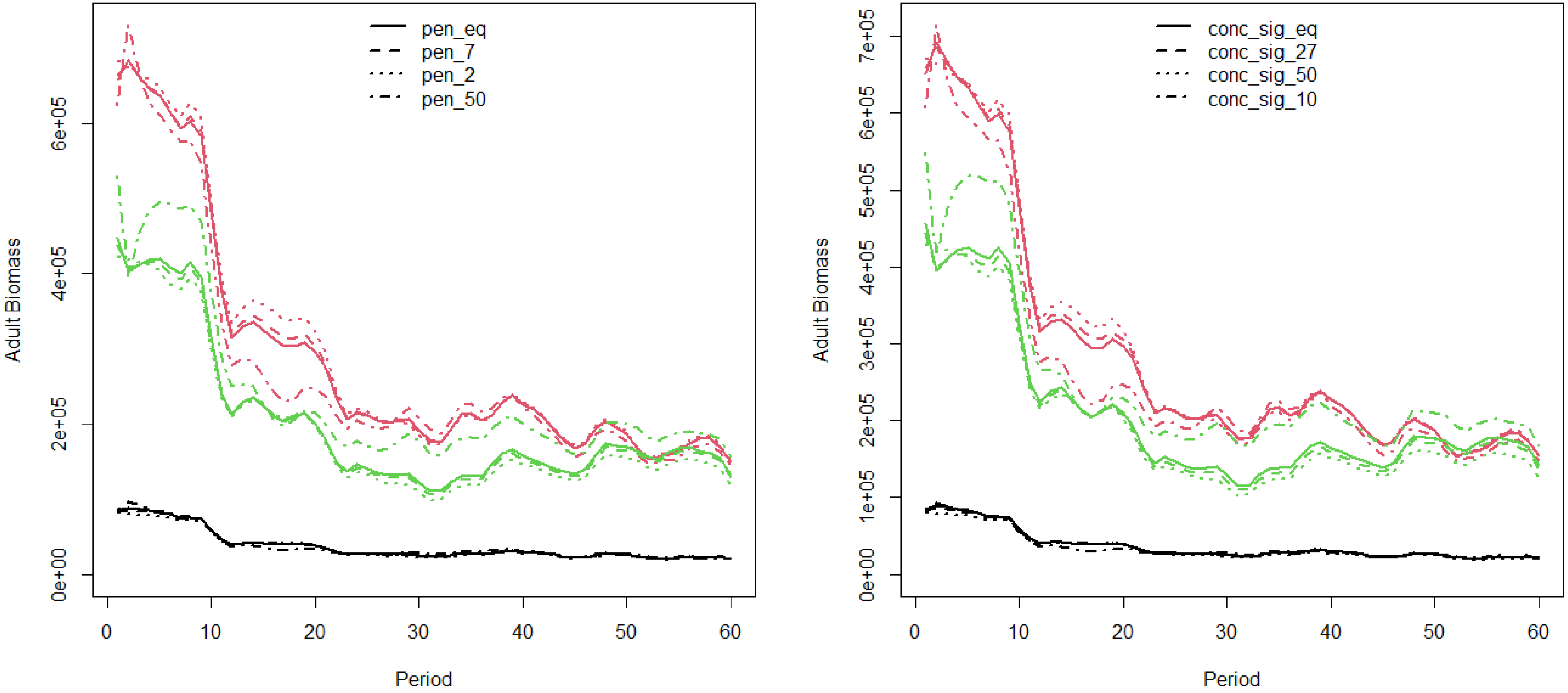

Figure 19. Model derived variable: adult biomass in each region, for the non-concentrated likelihood models (left panel) and concentrated likelihood models (right panel) with plotted lines corresponding to: fishery 18 (black lines), fishery 19 (red lines), and fishery 20 (green lines).

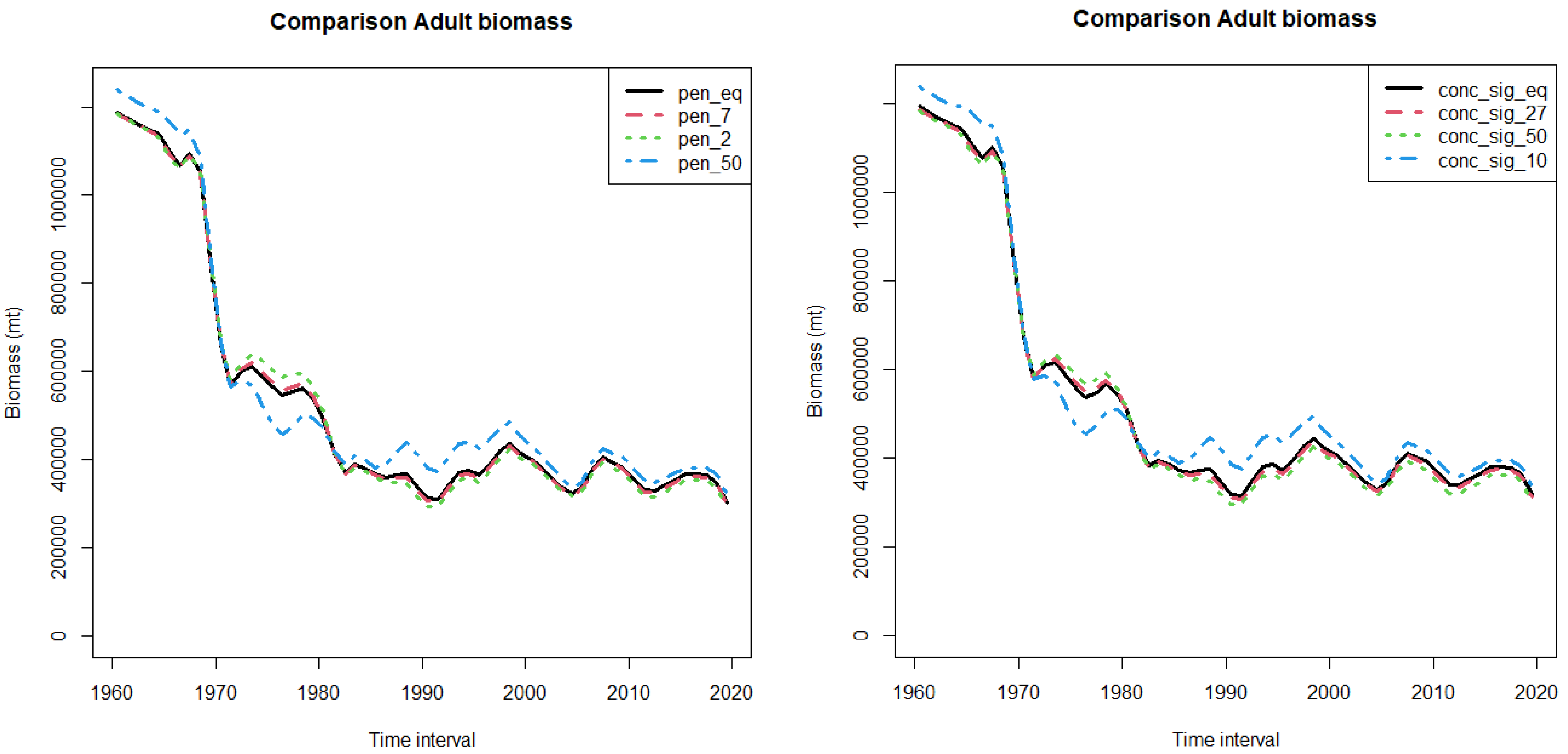

Figure 20. Derived variable: total adult biomass among all regions, for the non-concentrated likelihood models (left panel) and concentrated likelihood models (right panel), as estimated for the equivalent range of assumed error, σ.

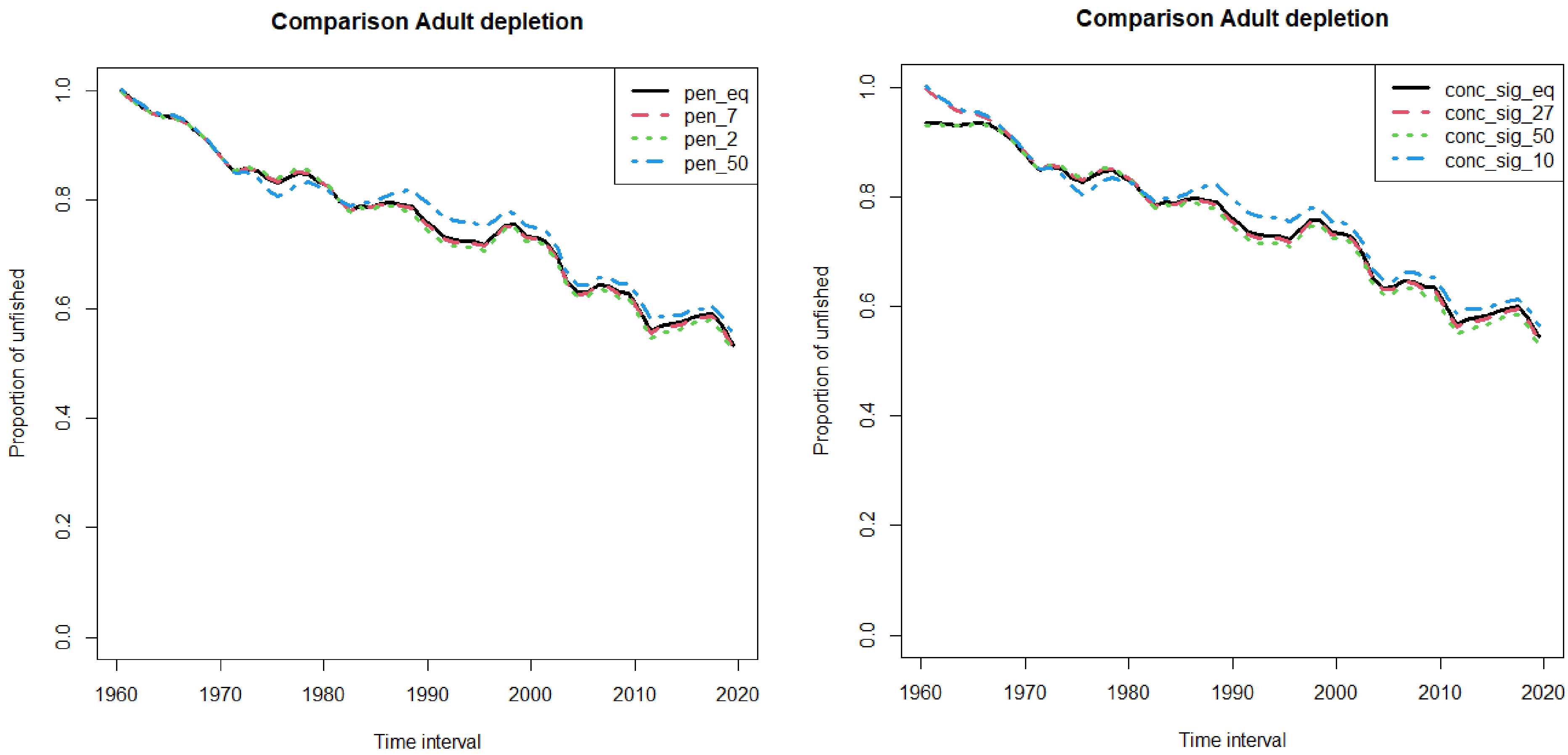

Figure 21. Derived variable: depletion level of adult biomass (proportion of unfished biomass) among all regions, for the non-concentrated likelihood models (left panel) and concentrated likelihood models (right panel), as estimated for the equivalent range of assumed error, σ.

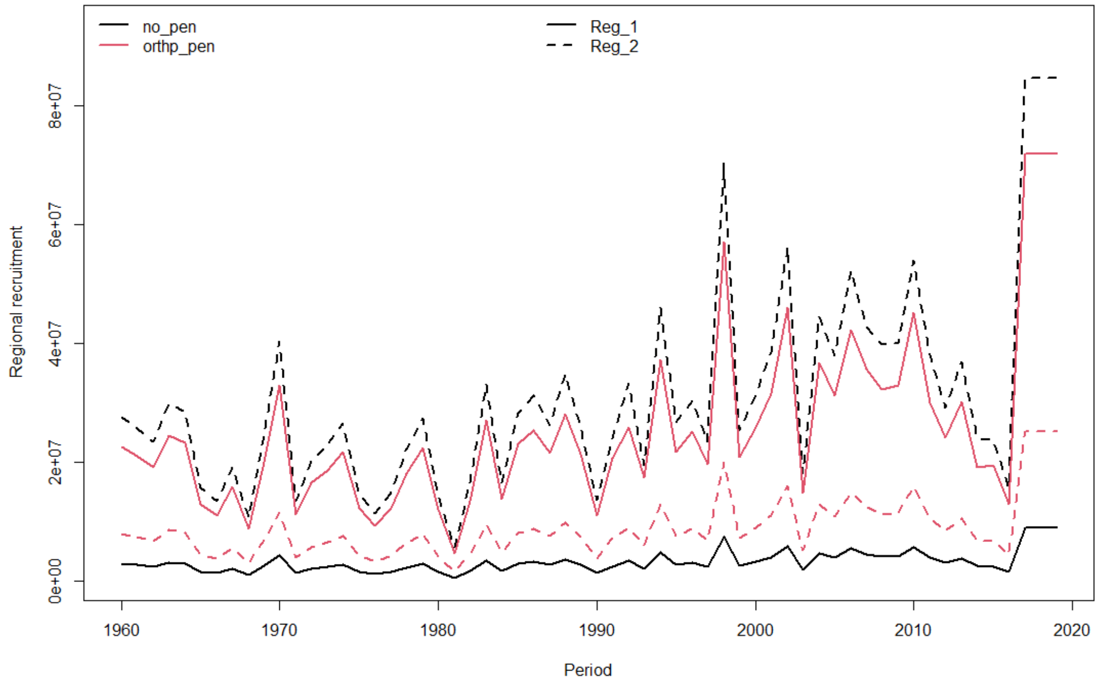

Figure 22. Absolute regional recruitments for the converged ALB2021 solutions obtained excluding (no_pen) and including (orthp_pen) the normal prior penalty (orthp_reg_recrs_pen) on the mean distribution of recruitments among regions; for regions 1 and 2, solid and dashed lines, respectively.

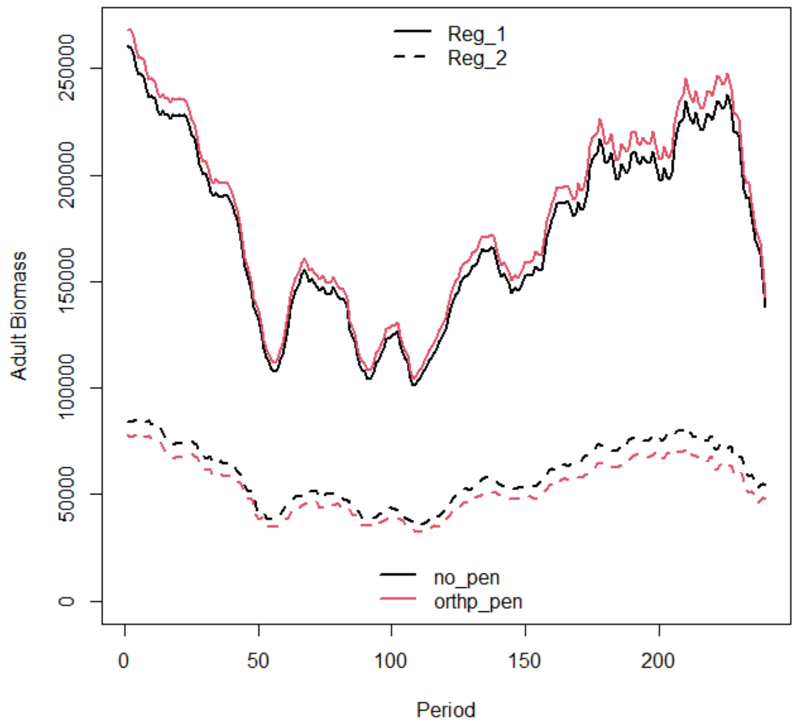

Figure 23. Absolute regional adult biomass for the converged ALB2021 solutions obtained excluding (no_pen) and including (orthp_pen) the normal prior penalty (orthp_reg_recrs_pen) on the mean distribution of recruitments among regions; for regions 1 and 2, solid and dashed lines, respectively.

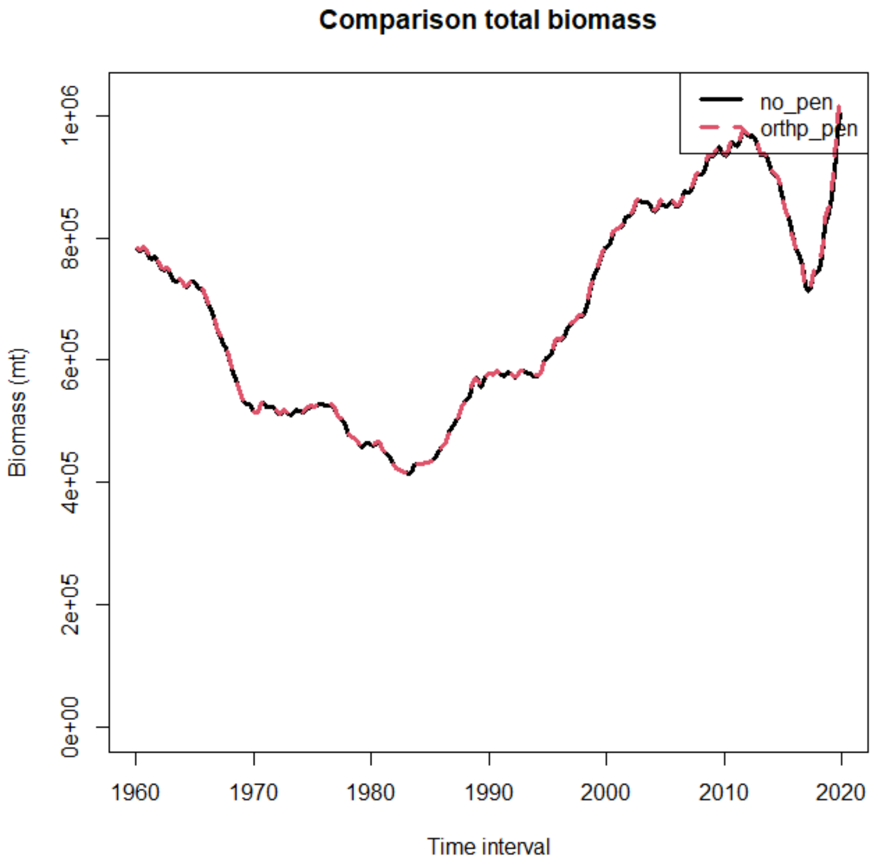

Figure 24. Absolute total biomass for the converged ALB2021 solutions obtained excluding (no_pen) and including (orthp_pen) the normal prior penalty (orthp_reg_recrs_pen) on the mean distribution of recruitments among regions.

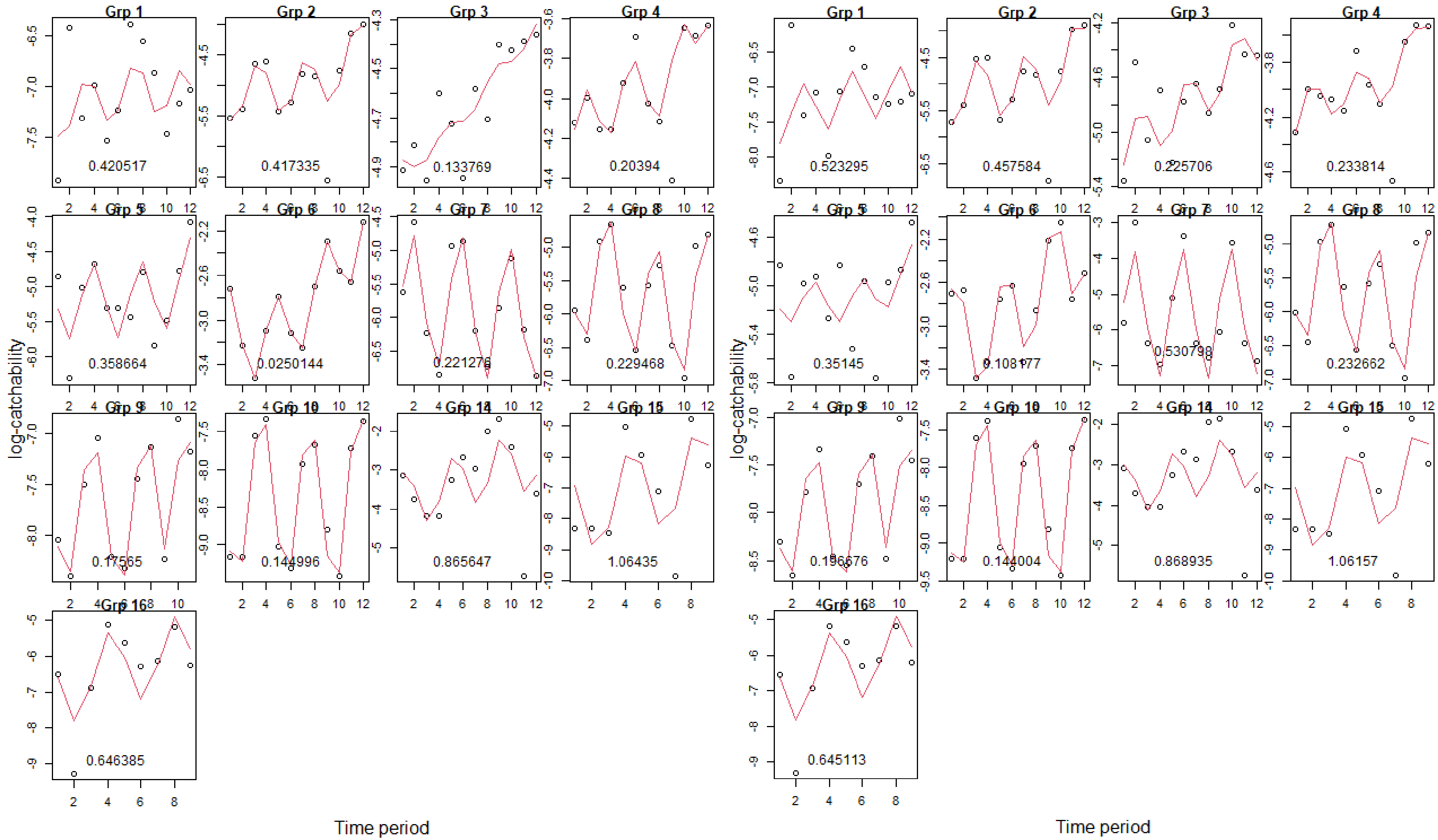

Figure 25. The estimated fml_effort_rltnshps for the: seas_poly_fml_3degs_term model that includes estimating the regression with all independent variables estimated (left panels); and, intct_fml_term model that implements the diagcs_intct_fml feature that fixes all independent variables except for the fml_effort_rltnshp polynomial coefficients (right panels).

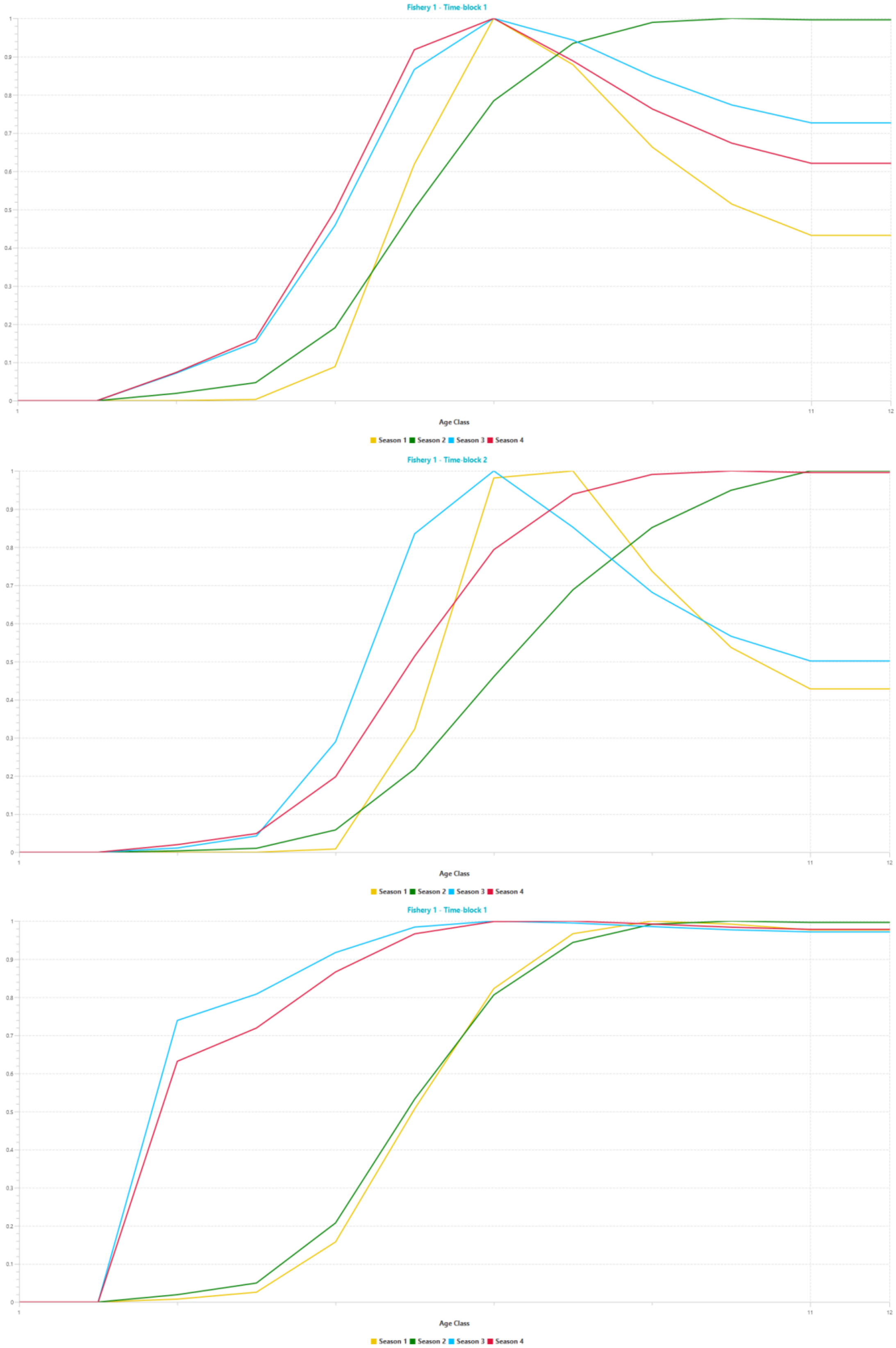

Figure 26. Time-variant selectivity including the constraint for a non-decreasing function activated for a single season in time-block 1 (top panel), or multiple seasons (middle and bottom panels).

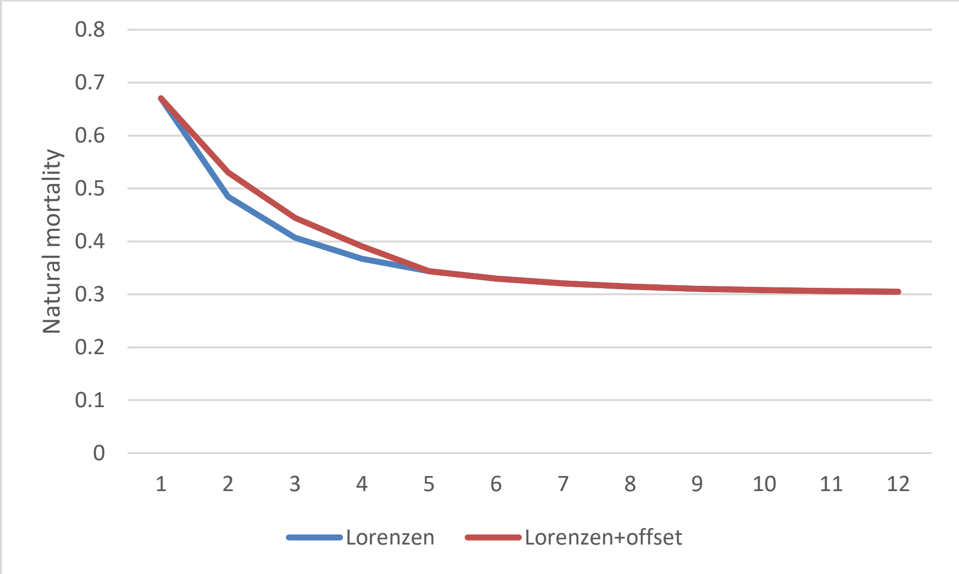

Figure 27. Natural mortality at age for the derived Lorenzen function without (Lorenzen), and with the estimated independent growth function offsets (Lorenzen+offset).

12 ANNEX 1

Modified source code files (49) during 2023-24:

alldevpn.cpp avcatfit.cpp callpen.cpp do_all_for_empirical_autocorrelated_bh.cpp getinp2.cpp goodpen.cpp learner_code.cpp lmul_io2.cpp lmult.cpp lwsim.cpp mfcl_thrd_linux64_debug.mak mfcl_thrd_linux64_opt.mak new-len-self-scaling-multiomial.cpp new-wght-self-scaling-multiomial.cpp new_incident_calc.cpp newgradc.cpp newgradc_noeff.cpp newl2.cpp newl4.cpp newl5.cpp newmau5a.cpp newmaux5.cpp newmprot.hpp newmult.cpp newmult.hpp	newrshimp.cpp newrshimp_experiment.cpp nnewlan.cpp nopenalties.cpp nrcatch4.cpp onevar.cpp plot.cpp plotstuff.cpp ptagfit.cpp recrpen.cpp rsh3imp.cpp selbreaks.cpp setcomm.cpp short2.cpp simulation_mode.cpp size.cpp test_msy.cpp testnewl3.cpp variable.hpp VERSION version.h version3_len_self_scaling_multinomial_re_multi_rho_multi_var.cpp version3_wght_self_scaling_multinomial_re_multi_rho_multi_var.cpp yield_bh.cpp

One new source code file was added during 2023-24:

indepvars.cpp