User’s Manual for ASPIC: A Stock-Production Model Incorporating Covariates (ver. 5)

And Auxiliary Programs

Michael H. Prager Population Dynamics Team

Center for Coastal Fisheries and Habitat Research National Oceanic and Atmospheric Administration 101 Pivers Island Road

Beaufort, North Carolina 28516 USA

National Marine Fisheries Service Beaufort Laboratory Document BL–2004–01

February, 2004

Last revised May 18, 2004

# Preface

This user’s manual describes Version 5.0 of ASPIC,a computer program to estimate parameters of a non- equilibrium surplus-production model from ﬁsher- ies data. Several utility programs (ASPICP, FTEST, AGRAPH) are also described. Their purposes are making projections, comparing models, and quickly making graphs from ASPIC and ASPICP output ﬁles. The programs together are referred to here as the ASPIC Suite.

The major change from previous versions of ASPIC is the ability to ﬁt the Pella–Tomlinson (generalized) production model, with the Fox exponential yield model included as a special case. The Schaefer (lo- gistic) production model, the main component of earlier versions, is still part of ASPIC, and because many of its computations can be done analytically rather than numerically, it will be found quicker, and its solutions may be more stable.

The ASPIC Suite is not commercial software, and the programs are not warranted in any way, either by the author or by the U.S. government. The software was developed for use in the author’s research, and it is used regularly. Distribution to fellow scientists in made in a cooperative spirit. *The software is in- tended as a set of research tools, and those who use them do so at their own risk.* ASPIC has been used on thousands of real and simulated data sets, and all supplied programs are believed to be substantially correct. The author appreciates receiving advice of suspected ﬂaws, and he attempts to correct errors promptly.

By no means is ASPIC the ﬁnal word in production modeling. It is intended as a reasonably ﬂexible pro- gram that can serve as a basis for further innovation.

Formal description of the theory behind ASPIC is given in [Prager](#_bookmark64) ([1994](#_bookmark64)). Further references are given in the bibliography. The author requests that this manual and [Prager](#_bookmark64) ([1994](#_bookmark64)) be cited in any report or published article that uses ASPIC.

Those who have used version 3.x of ASPIC and who now are presented with version 5.x might ask what happened to version 4.x. The answer is simple: 4.x were test versions. It seemed more logical to release the new version as 5.0, rather than some number in the middle of the 4.x series.

Many colleagues have given valuable technical sug- gestions or assistance while ASPIC was being writ- ten and as it has been revised through the years. I

thank S. Cadrin, R. Deriso, K. Hiramatsu, J. Hoenig,

R. Methot, C. Porch, J. Powers, A. Punt, V. Restrepo,

G. Scott, K. Shertzer, P. Tomlinson, D. Vaughan, E. Williams, and the many ﬁshery scientists who have sent data sets to illustrate their applications and questions. Development of ASPIC and related re- search is supported by the Southeast Fisheries Sci- ence Center of the U.S. National Marine Fisheries Ser- vice.

This software is distributed to interested scientists free of charge. It is the property of the United States government. No individual or group is authorized to charge for it or distribute it as part of any com- mercial product.

## Typographical conventions

In this manual, user commands, ﬁle names, and items in input ﬁles are displayed in a

monospaced font. Some important sections •

are marked by a symbol in the margin, as here; attention to such material is especially important to obtaining good results from ASPIC. Material new in

this version of the program is marked by a diﬀerent \*

marginal symbol, as here.

Michael H. Prager Beaufort, North Carolina January, 2004

# Contents

[Preface](#_bookmark0) 2

[Typographical conventions](#_bookmark1) 2

[Contents](#_bookmark2) 3

[List of Tables](#_bookmark3) 3

1. [Introduction](#_bookmark4) 5
2. [New in **ASPIC** 5.0](#_bookmark5) 5
   1. [Major changes](#_bookmark6) 5
   2. [Using new features](#_bookmark7) 6
      1. [Execution speed](#_bookmark8) 6
      2. [Developmental features](#_bookmark9) 7
3. [Installation and Interaction](#_bookmark10) 7
4. [Auxiliary Programs](#_bookmark36) 12
   1. [Overview of ASPICP](#_bookmark37) 12
   2. [Overview of FTEST](#_bookmark41) 12
   3. [Overview of AGRAPH](#_bookmark35) 12
5. [**ASPIC** Input File Speciﬁcation](#_bookmark38) 12
   1. [Generating a sample input ﬁle](#_bookmark39) 12
   2. [General format guidelines](#_bookmark40) 12
   3. [The ASPIC input ﬁle, line by line](#_bookmark43) 13
   4. [Questions about data series](#_bookmark48) 17
      1. [Missing values and zeroes](#_bookmark49) 17
      2. [Allocating yield](#_bookmark51) 19
      3. [Caution](#_bookmark52) 19
6. [Advanced Options for **ASPIC**](#_bookmark53) 19
7. [**ASPICP** Input File Speciﬁcation](#_bookmark54) 19
   1. [Line by line](#_bookmark55) 20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [3.1](#_bookmark11) | [Compatibility](#_bookmark11) . . . . . . . . . . . . . . . . | 7 |  | | |
| [3.2](#_bookmark9) | [Installation](#_bookmark9) . . . . . . . . . . . . . . . . . | 7 |  | [8.2 Sample ASPICP input ﬁle](#_bookmark56) . . . . . . . . . | 20 |
| [3.3](#_bookmark12) | [Interface](#_bookmark12) . . . . . . . . . . . . . . . . . . . | 7 | [**9**](#_bookmark57) | [**Interpretation of ASPIC Results**](#_bookmark57) | **21** |

* + 1. [ASPIC and ASPICP](#_bookmark13) 7
    2. [Auxiliary programs](#_bookmark15) 8
    3. [NMFS Toolbox](#_bookmark17) 8

1. [**Overview of ASPIC**](#_bookmark19) **8**
   1. [Data requirements](#_bookmark20) 8
   2. [Program limits](#_bookmark16) 8
   3. [Program modes](#_bookmark18) 8
   4. [Several data series](#_bookmark22) 9
   5. [Precision of parameter estimates](#_bookmark58) 21
   6. [Using several data series](#_bookmark59) 21

[9.2.1 Catchability over time](#_bookmark60) 21

* 1. [Estimation diﬃculties](#_bookmark61) 22

## [Program change history](#_bookmark62) 23

## [Source Code](#_bookmark63) 24

## [References](#_bookmark65) 25

* 1. [Objective function](#_bookmark26) . . . . . . . . . . . . . 9
     1. [Penalty for initial biomass](#_bookmark23) 9
     2. [Conditioning on yield](#_bookmark24) 9
  2. [Bootstrapping](#_bookmark25) 9
  3. [Input and output ﬁles](#_bookmark29) 10
     1. [Input ﬁle](#_bookmark31) 10
     2. [Editing input ﬁles](#_bookmark28) 10
     3. [Output ﬁles](#_bookmark30) 10
  4. [Starting ASPIC](#_bookmark33) 11

# List of Tables

1. [Data-series types and codes](#_bookmark27) 10
2. [Files used by the ASPIC suite](#_bookmark32) 11
3. [Missing values and zeroes](#_bookmark50) 18

# 1 Introduction

This user’s manual describes Version 5.0 of ASPIC,a computer program to estimate parameters of a non- equilibrium surplus-production model from ﬁsher- ies data. Several utility programs (ASPICP, FTEST, AGRAPH) are also described. Their purposes include making projections, comparing models, and quickly making graphs from ASPIC and ASPICP output ﬁles. The programs together are referred to here as the ASPIC Suite.

The surplus-production model has a long history in ﬁshery science and has repeatedly proven useful in management of ﬁsh stocks. The appeal of produc- tion models is in large part due to their conceptual and computational simplicity. Despite that simplic- ity, production models incorporate an implicit re- cruitment function, and thus can be used for studies of sustainability. Production models have also been found especially useful in stock assessments when the age-structure of the catch cannot be estimated.

Many early treatments of surplus-production mod- els assumed that the yield taken each year could be considered the equilibrium yield (e. g., [Fox](#_bookmark71) [1975](#_bookmark71)). However using that “equilibrium assumption” tends to overestimate MSY when used to assess a declining stock, and it has been found problematic by several studies ([Mohn](#_bookmark82) [1980](#_bookmark82); [Williams and Prager](#_bookmark90) [2002](#_bookmark90)). The assumption was a computational convenience that is no longer needed, and ASPIC does not use it.

* Earlier versions of ASPIC could ﬁt only the logis- tic production model ([Schaefer](#_bookmark80) [1954](#_bookmark80), [1957](#_bookmark83); [Pella](#_bookmark85) [1967](#_bookmark85)), in which the production curve (curve of sur- plus production vs. biomass) is symmetrical around MSY. Version 5.x also ﬁts the generalized model of [Pella and Tomlinson](#_bookmark87) ([1969](#_bookmark87)) in the revised parame- terization of [Fletcher](#_bookmark66) ([1978](#_bookmark66)).

ASPIC incorporates several extensions to classical stock-production models. One extension is that ASPIC can ﬁt data from up to 10 data series. These may be catch–eﬀort series (from diﬀerent gears or diﬀerent periods of time), catch–abundance-index series, biomass indices, or biomass estimates made independently of the production model. This fea- ture is described in §[4.4](#_bookmark22). A second major extension is the use of bootstrapping for bias correction and construction of approximate nonparametric conﬁ- dence intervals. A third extension is that ASPIC can ﬁt a model under the assumption that yield in each year is known more precisely than ﬁshing eﬀort or

relative abundance; in other words, ﬁtting can be sta- tistically conditioned on yield, rather than on ﬁshing eﬀort or relative abundance.

The theory behind ASPIC and several worked ex- amples were ﬁrst presented in working documents of the International Commission for the Conserva- tion of Atlantic Tunas (ICCAT) by [Prager](#_bookmark88) ([1992a](#_bookmark88),[b](#_bookmark91)). Those reference have been superseded by the more formal and complete treatment of [Prager](#_bookmark64) ([1994](#_bookmark64)). The model and its extensions are also described in [Quinn and Deriso](#_bookmark78) ([1999](#_bookmark78)) and [Haddon](#_bookmark76) ([2001](#_bookmark76)). The basic theory of production models is of course also described in many other texts, including [Hilborn and Walters](#_bookmark79) ([1992](#_bookmark79)), and is the subject of a recent FAO publication ([Punt and Hilborn](#_bookmark77) [1996](#_bookmark77)).

The ASPIC computer program as described here has been used by several assessment groups and in many studies, including [Prager et al.](#_bookmark72) ([1996](#_bookmark72)), [Prager and Goodyear](#_bookmark69) ([2001](#_bookmark69)), [Prager](#_bookmark67) ([2002](#_bookmark67)), [Shertzer and Prager](#_bookmark86) ([2002](#_bookmark86)), and [Williams and Prager](#_bookmark90) ([2002](#_bookmark90)). In the course of those studies, the program has been exercised on over 100,000 sets of simu- lated data. The resulting experience has been used to improve the program’s reliability.

1. **New in ASPIC 5.0** •

This section gives an overview of changes intro- duced between ASPIC 3.x and ASPIC 5.x. Although this section will be of most interest to users of previ- ous versions, new users should also review it brieﬂy for information on running ASPIC 5.0 productively.

## Major changes \*

*Generalized production model.* Earlier versions of ASPIC could ﬁt only the logistic form of the pro- duction model ([Graham](#_bookmark73) [1935](#_bookmark73); [Schaefer](#_bookmark80) [1954](#_bookmark80), [1957](#_bookmark83); [Pella](#_bookmark85) [1967](#_bookmark85); [Prager](#_bookmark64) [1994](#_bookmark64)). As well as that form, ASPIC 5.0 can ﬁt the generalized production model ([Pella and Tomlinson](#_bookmark87) [1969](#_bookmark87); [Fletcher](#_bookmark66) [1978](#_bookmark66)) in one of three ways: by direct optimization, by a grid of ﬁts on the model shape, or with ﬁxed model shape to implement the [Fox](#_bookmark70) ([1970](#_bookmark70)) model or other pre- determined shape.

*Parameterization change.* The generalized model requires parameterization in terms of MSY and car- rying capacity *K* rather than MSY and intrinsic rate of increase *r* . This occurs because when the exponent

*n* in the generalized model is in the region *n ≤* 1,

then *r = ∞*. As a result, ASPIC 5.0 requires a start-

ing guess for *K*, not for *r* .

*Starting biomass parameterization.* A parameter es- timated by ASPIC is the biomass in the ﬁrst year of the analysis. In previous versions, this was ex- pressed (both in the input ﬁle and in ASPIC reports) as a ratio to the biomass providing MSY; i. e., as *B*1*/B*MSY. In version 5.0, it is expressed as a ratio to the carrying capacity, i. e., as *B*1*/K*. This change is required because in the generalized model, *B*MSY is no longer a ﬁxed proportion of *K*. The change re- duces correlation between estimates of the starting biomass ratio and of *B*MSY.

*Conditioning options.* Option names for condition- ing on yield or eﬀort have been revised to indicate conditioning, rather than residuals. The new speci- ﬁcations are given in §[6.3](#_bookmark42).

*Fitting criteria.* An additional objective function, least absolute values (LAV), is available. It is recom- mended that this robust objective function be used only in conjunction with a regular least squares ﬁt, because the optimizer has a more diﬃcult task in ﬁnding the best minimum of LAV ﬁts. Nonetheless, LAV can be valuable where one or more data observa- tions are markedly disjoint from the rest. For guide- lines on appropriate use of LAV, please consult the statistical literature.

*Bounds on catchability coeﬃcient.* To improve con- vergence, estimates of *q* (catchability) are now bounded to a geometric range around the user’s starting guess. The bounds are determined inter- nally by ASPIC and are not under the user’s control. If the starting guess for *q* is severely wrong, the es- timate may hit a bound, and the ASPIC report will indicate whether the starting guess was too low or too high. If that happens, the user should revise the starting guess of *q* accordingly and rerun the analy- sis.

*Restarts during optimization.* Previous versions of ASPIC required the optimizer to return to the same solution 3 times in a row to indicate convergence. That worked well on most data sets, but was not al- ways suﬃcient. The number of identical returns is now speciﬁed in the input ﬁle. The recommended default is 6. This can improve stability of the ﬁt on some poor data sets. This is speciﬁed on line 8 of the input ﬁle; see §[6.3](#_bookmark45) on page [15](#_bookmark45).

*Time steps.* The generalized model is implemented by numerical integration that approximates a continuous–time solution. The number of time steps per year is speciﬁed on line 9 of the input ﬁle; see

§[6.3](#_bookmark46) on page [15](#_bookmark46).

*Setting advanced options.* A new INI ﬁle in the working directory can contain values for some ad- vanced options, described in §[7](#_bookmark53) on page [19](#_bookmark53).

*Updated ASPICP.* An updated version of the projec- tion program ASPICP is compatible with analyses from ASPIC 5.0. It is also backwardly compatible with ASPIC 3.x and 4.x.

*Windows installer.* This release is distributed as a self-installing binary ﬁle for Windows. Versions for other operating systems may be available on request.

*Drag-and-drop versions.* ASPIC has always been a non-interactive program that reads from and writes to ASCII ﬁles. This release includes alternative ver- sions that accept drag-and-drop of input ﬁles and that display their output in scrolling windows. The original command-line versions are also supplied.

## Using new features •

When using the new features of ASPIC 5.0, please consider the following—

* + 1. Execution speed

Fitting the generalized model is done with a numeri- cal solution of the catch equation and is thus slower than ﬁtting the logistic model (which has an analyti- cal solution). Execution will be especially slow in any the following cases:

* + - 1. Poor agreement between model and data
      2. Analysis of more than one data series
      3. Bootstrapping, especially in combination with

#1 or #2

* + - 1. Extensive Monte Carlo trials in ﬁtting

Program speed can be improved by reducing the number of time steps per year in the input ﬁle (see

§[6.3](#_bookmark46) on page [15](#_bookmark46)). For the closest approximation to a continuous-time model, set this value to a large number; e.g., 80. For a fairly close approximation, use a number in the range 12–24 (recommended). For fastest operation set this to 2 steps per year.

* 2.2.2 Developmental features

The following are believed to work correctly but have not been tested extensively:

* + Generalized estimation (in all forms) with more than one data series.
  + Objective functions other than SSE.
  + Projections of generalized bootstraps with

ASPICP.

* + The values of the AIC (Akaike Information Cri- terion) printed when ﬁtting the generalized model.
  + The *F* –statistic printed for comparing the lo- gistic and generalized models may be incor- rect when more than one data series is ana-

lyzed. More importantly, simulations suggest that such tests are of little value ([Prager](#_bookmark67) [2002](#_bookmark67)).

* It is wise to repeat estimation with several diﬀerent random-number seeds. If results cannot be dupli- cated within a few percent (usually less), a ﬁtting failure is indicated, and such results should not be considered valid estimates.
* The author will appreciate receiving reports of suc- cessful or unsuccessful use of the features itemized above. He will attempt to ﬁx all bugs promptly.

# Installation and Interaction

## Compatibility

The ASPIC suite is compatible with personal comput- ers running Microsoft Windows 9x (including Win- dows 95, 98 and Me) or Windows NT (including Win- dows NT 4.0, 2000, and XP).[1](#_bookmark14)

ASPIC is written in standard Fortran 95 and is portable to other operating systems. Please consult the author if you would like to use ASPIC under op- erating systems other than Windows.

1Use of tradenames does not imply endorsement by NMFS, NOAA, or the author.

## Installation

This version of ASPIC is available as a self-installing executable ﬁle for Windows. The installer performs the following tasks:

* + Installs binary ﬁles for ASPIC, ASPICP, FTEST and

AGRAPH to a location speciﬁed by the user

* + Installs this User’s manual and a Quick Refer- ence Card to the doc subdirectory of the instal- lation location
  + Installs sample input and output ﬁles to the samples subdirectory of the installation loca- tion
  + Adds the installation location to the user’s PATH speciﬁcation so that ASPIC and related pro- grams can be executed from a command win-

dow open to any directory

* + Adds a GINO environment variable pointing to the installation location, as required by a graph- ics support library used in AGRAPH. If a GINO

environment variable already exists on the sys- tem, it is not modiﬁed.

* + Adds shortcuts to the Windows Start menu and Desktop, including a command window opening in a user-speciﬁed working directory
  + Adds an uninstaller to the installation location and adds ASPIC to the system’s “Remove Pro- grams” list

The ASPIC uninstaller removes all of the above. How- ever, any ﬁles added by the user are not removed.

This User’s Manual is supplied with all distributions of ASPIC as an Adobe PDF ﬁle named ASPICMAN.PDF. It may be distributed freely.

## Interface •

* + 1. Interface of ASPIC and ASPICP

The standard versions of ASPIC and ASPICP do not include graphical user interfaces. Instead, the pro- grams are console-mode (character) programs that read all input from and write all output to ASCII (text) ﬁles. The screen is used only for status mes- sages.

For added ease of use, the ASPIC 5.0 instal- lation includes versions of ASPIC and ASPICP that support drag and drop. (The executable

ﬁles of the command-line versions are aspic.exe and aspicp.exe; of the drag-and-drop versions, aspicw.exe and aspicpw.exe.

* + 1. Interface of auxiliary programs

The auxiliary program FTEST has a text-mode user interface, is interactive, and does not require an in- put ﬁle. Output is written to the screen.

The graphics program AGRAPH incorporates a stan- dard Windows GUI, with output available to any Win- dows printer or to a graphics ﬁle (WMF or EPS). It can be executed from the command line, by drag and drop, or by starting the program from its shortcut.

* + 1. NMFS Toolbox
* The US National Marine Fisheries Service (NMFS) has developed a “toolbox” of computer programs for stock assessment. The toolbox currently includes a graphical editor speciﬁcally for ASPIC input ﬁles and graphics that work with ASPIC output. It also includes many other stock–assessment and projec- tion tools. For further information, contact Dr. Paul Rago at the NMFS laboratory in Woods Hole, Mas- sachusetts.

1. **Overview of ASPIC**

## 4.1 Data requirements

Data needed by ASPIC are a series of observations on yield (catch in biomass) and one or more correspond- ing series of relative abundance. Data on ﬁshing ef- fort rate can be used instead of relative abundance, and when used are assumed to represent eﬀective (standardized) eﬀort. ASPIC assumes that the sup- plied abundance index is an unbiased index of the stock’s abundance in biomass. If data on ﬁshing ef- fort are provided, ASPIC assumes that eﬀort divided by yield forms an unbiased index of the stock’s abun- dance.

In this User’s Guide, the terms “catch” and “yield” are used interchangeably to mean total removals in biomass. Similarly, CPUE is used to mean relative abundance. The presumption is that the CPUE has been standardized before being used for modeling.

* In addition to data, ASPIC requires starting guesses of its estimated parameters. Parameters directly es- timated are *K*, the stock’s maximum biomass or car- rying capacity; MSY, the maximum sustainable yield; *B*1*/K*, the ratio of the biomass at the beginning of

the ﬁrst year to *K*; and for each data series *i*, *qi*, the catchability coeﬃcient for that series. Description of the input ﬁle format, given in §[6](#_bookmark38), includes sug- gestions for starting guesses.

## Program limits

The array limits of ASPIC are as follows:

* + - Number of years of data: 90
    - Number of data series: 10
    - Number of bootstrap trials: 1,000

Any user with larger requirements is invited to con- tact the author.

## Program modes

ASPIC has three modes of operation, here called “program modes.”

* + - In FIT program mode, ASPIC ﬁts the model and computes estimates of parameters and other quantities of management interest, including

time trajectories of ﬁshing intensity and stock biomass. Execution time is relatively short.

* + - In BOT program mode, ASPIC ﬁts the model and computes bootstrapped conﬁdence intervals on estimated quantities. Because computations are

extensive, execution time in BOT mode is consid- erably longer than in FIT mode. For example, a bootstrap with 500 trials might take 200–500 times as long as a single ﬁt.

* + - In IRF program mode, ASPIC conducts an iter- atively reweighted ﬁt when two or more data series are analyzed. Iterative reweighting of

the data series (inverse-variance weighting) pro- vides, under many circumstances, a maximum- likelihood solution.

The modes BOT and IRF cannot be combined. In other words, ASPIC cannot run a bootstrap on an it- eratively reweighted ﬁt.

A typical analysis might begin with FIT mode, in- cluding several runs to explore diﬀerent model structures. If questions about series weighting are to be addressed, IRF mode might be used, instead, in the initial analysis. After model and data struc- ture have been decided, BOT mode can be used to es- timate the uncertainty in assessment results. ASPIC

bootstrap runs do not incorporate iterative reweight-

*I N* I

*Yij* \

ing, and this will cause underestimation of variabil- ity when IRF mode has been used to develop a model structure.

Ω *=* ) ) *wi* ln

*i=*1 *j=*1 *ij*

*Y*ˆ

2

(1)

IRF mode was developed in response to requests during assessment workshops, but it has not been much used by the author, and thus is less thoroughly tested than the other modes. Experience has shown that while series weights estimated in IRF mode may be statistically unbiased, they can be of high vari- ance. (This is a characteristic of such weights gen- erally, and is not speciﬁc to ASPIC.) For that rea- son, sensitivity to series weights should be examined whenever IRF program mode is used.

## Fitting more than one data series

ASPIC can ﬁt data on up to 10 simultaneous or serial ﬁsheries (or biomass estimate series or biomass in- dex series). Data series may be of several types (Ta- ble [1](#_bookmark27)), but at least one series must be type CE (eﬀort and yield) or type CC (CPUE and yield). When more than one series is analyzed, common estimates of *B*1*/K*, MSY, and *K* are made, along with an estimate of *qi* for each series. The interpretation of *qi* de- pends on the type of data series to which it pertains.

A statistical weight *wi* for each ﬁshery is speciﬁed by the user in the input ﬁle. In summing the ob- jective function, each squared residual from ﬁshery *i* is multiplied by *wi*. If the series have equal er- ror variances, using weights of unity for each series provides a maximum-likelihood solution under the lognormal error structure assumed by ASPIC.

In FIT program mode, the program normalizes the user’s *wi* so that they sum to unity. In IRF mode, the program adjusts the weights interatively to provide nearly equal estimated variances. Weights are also adjusted so they sum to unity.

The computer time needed to obtain estimates gen- erally increases as more data series are added. The increase is due both to addition of data and in- creased diﬃculty of optimization.

## Objective function and penalty term

Parameters are estimated under the assumption that the errors in yield or eﬀort are multiplicative with

for residuals accumulated in yield (EFT optimization

mode, §[6.3](#_bookmark42)), or a similar expression for residuals in eﬀort (YLD optimization mode).

In equation ([1](#_bookmark21)), *i* indexes the data series, *j* the year, *w* is the series’ statistical weight, *Yij* is the observed yield (or biomass index or estimates) from series *i* in year *j*, and *Y*ˆ*ij* is the corresponding predicted value.

* + 1. Penalty for initial biomass

A penalty term can be added to the objective func- tion to discourage estimates in which the ﬁrst year’s biomass *B*1 is greater than the carrying capacity *K*. This penalty can aﬀect the estimates of other param- eters, so when this term is used, the results should be compared to those obtained by setting the term to zero. The penalty term is described in more de- tail in [Prager](#_bookmark64) ([1994](#_bookmark64)), and its use is described in the section describing the input ﬁle format.

* + 1. Conditioning on yield

ASPIC can consider yield known exactly and accu- mulate residuals in eﬀort. Yield is usually observed more precisely than eﬀort or the abundance index, and it is usually preferable on statistical grounds to compute residuals in the more imprecise quan- tity. Thus, conditioning on yield is recommended for most analyses. An additional advantage is that estimation of missing eﬀort values is quite simple (and is included in this version of ASPIC). When con- ditioning on yield, an iterative solution of the catch equation is used, and computation is slower than when conditioning on eﬀort.

## Bootstrapped conﬁdence intervals

In BOT mode, ASPIC uses bootstrapping to estimate bias-corrected conﬁdence intervals on many quan- tities of interest. In doing this, estimated yields (if conditioning on eﬀort; estimated eﬀorts, if condi- tioning on yield) and residuals from the original ﬁt are saved. The residuals are then increased by an adjustment factor ([Stine](#_bookmark89) [1990](#_bookmark89), p. 338), which is re- ported in the output ﬁle.

Bootstrapped data sets are then constructed by

constant standard deviation. Thus the residuals are

*ij*

combining each saved predicted yield

*Y*ˆ*ij* with a

accumulated in logarithmic transform. The objec- tive function Ω minimized, then, is

randomly-chosen adjusted residual to arrive at a pseudo-yield value *Y*ˆ*∗*. (This procedure assumes

*Table 1. Codes for the eight types of data series allowed in ASPIC.*

|  |  |  |
| --- | --- | --- |
| Code | Data type | When measured |
| CE | Fishing eﬀort rate, catch (weight) | Eﬀort rate: annual average Catch: annual total |
| CC | CPUE (weight-based), catch (weight) | CPUE: annual average Catch: annual total |
| B0 | Estimate of biomass | Start of year |
| B1 | Estimate of biomass | Annual average |
| B2 | Estimate of biomass | End of year |
| I0 | Index of biomass | Start of year |
| I1 | Index of biomass | Annual average |
| I2 | Index of biomass | End of year |

that the statistical series weights *wi* are correct.) The model is then reﬁt, using the pseudo-yields in place of the original observed yields. The process is repeated (always using the original predicted values) up to 1,000 times.

From the bootstrap results, bias-corrected (BC) con- ﬁdence intervals can be computed by standard meth- ods ([Efron and Gong](#_bookmark68) [1983](#_bookmark68)). The statistical literature recommends 1,000 bootstrap trials when comput- ing 95% conﬁdence intervals. ASPIC computes 80% conﬁdence intervals, and should require fewer trials. The author recommends using at least 500 trials for bootstrap runs.

## Input and output ﬁles

As noted, all ASPIC input and output ﬁles (Table [2](#_bookmark32) on p. [11](#_bookmark32)) are in plain ASCII format. Sample ﬁles are provided with the ASPIC distribution.

* + 1. Input ﬁle

An ASPIC input ﬁle contains all data and settings re- quired for a single ASPIC run. It is recommended that when series of runs is made, that each input ﬁle be given a distinct name. This will ensure that the resulting output ﬁle names also are distinct.

The input ﬁle format is described in detail in §[6](#_bookmark38) on page [12](#_bookmark38). The simplest way to generate an ASPIC in- put ﬁle is to run the command

aspic -help

from the command line, to generate the ﬁle sample.inp. That ﬁle can then be renamed and edited to the user’s speciﬁcations. It may be use- ful to save an extra copy of the resulting ﬁle for use as a template.

* + 1. Editing input ﬁles

To create and edit ASPIC input ﬁles and ASPICP control ﬁles, a text editor is best used. Windows Notepad is a simple example of such a program. (Text editors are sometimes known as programmer’s editors or ASCII editors.) Many high-quality text ed- itors are available as freeware, shareware, or com- mercial software.

An especially useful feature in an editor used for ASPIC input ﬁles is the ability to cut and paste rectan- gular blocks of text. A relatively simple editor having that feature is ConTEXT, which as of January, 2004, was available without charge and could be located through Web search engines such as Google. Other well known editors, such as xemacs, are also suited to this task.

* + 1. Output ﬁles

Output from ASPIC includes parameter estimates; measures of goodness of ﬁt; and estimates of popu- lation benchmarks, biomass levels, and exploitation levels. Simple character plots are also provided. In addition, the output from bootstrap runs includes bias-corrected conﬁdence intervals on parameters and on other quantities of management interest.

The name used by ASPIC for its main output ﬁle de- pends on the mode of operation (Table [2](#_bookmark32)). Suppose the data (input) ﬁle is named sword.inp. Then in FIT and IRF modes, the main output would be writ- ten to a ﬁle named sword.fit. In BOT mode, the main output would be written to ﬁle sword.bot, and detailed intermediate results of the bootstrap would be written to sword.det and sword.bio. The spe- cial output ﬁle, if written (see next paragraph) would be written to sword.prn.

*Table 2. Files read (R) or written (W) by ASPIC and related programs.*

File Used

type Action by File contents and description

INP R ASPIC Input ﬁle with data, starting guesses, and run settings.

INI R ASPIC Optional ﬁle to set certain advanced options.

FIT W ASPIC Output ﬁle with estimates and graphs; written in FIT

and IRF program modes.

BOT W ASPIC Output ﬁle with estimates and graphs; written in BOT

program mode.

BIO W, R ASPIC, ASPICP

Stores estimated *B* and *F* trajectory for each bootstrap trial; used by ASPICP to generate conﬁdence intervals.

DET W ASPIC Stores estimates from each bootstrap trial.

SUM W ASPIC Stores summary estimates from all runs made in a di-

rectory. Set verbosity to 10 or above in INP ﬁle to write to a SUM ﬁle.

PRN W ASPIC Has estimated trajectories in ASCII format; easily read

for graphing by S–Plus, R, SAS, or spreadsheet.

GEN W ASPIC S-compatible ASCII ﬁle with summary results from GEN-

GRID mode.

GRD W ASPIC S-compatible ASCII ﬁle with summary results from LOG-

GRID mode.

Although the .FIT and .BOT ﬁles can be read into a spreadsheet or statistical package, they contain many headings that can be confusing. Thus, a spe- cial ﬁle designed to be compatible with computer programs including S-Plus and R, can be written if desired. This .PRN ﬁle contains input and estimated time series from an ASPIC BOT or FIT run. For infor- mation on enabling this ﬁle, see §[6.3](#_bookmark44) on p [14](#_bookmark44) or §[7](#_bookmark53) on p. [19](#_bookmark53).

To aid in simulation studies, a summary ﬁle (.SUM ﬁle) may also be written in the current directory. This is described in more detail in §[6.3](#_bookmark44). The .SUM ﬁle can be written by S-Plus or R with S code like

read.table("aspic.sum",header=T)

The .BIO ﬁle is used by ASPICP (described below) for its computations after a bootstrap run. The .DET ﬁle provides information on the individual bootstrap tri- als. It is not used directly by any supplied program, but is provided in case needed for the user’s conve- nience.

The main .FIT and .BOT output ﬁles, and all other ASPIC related ﬁles intended to be read by the user, and written with a maximum line length of 120 char- acters. The .BIO and .DET output ﬁles, which are in- tended to be read by computer programs, may have much longer lines.

**4.8 Starting ASPIC**

First prepare an input ﬁle in the correct format (de- scribed in §[6](#_bookmark38) on page [12](#_bookmark38)). It’s easiest to copy one of the sample input ﬁles provided to use as a template.

Then start the program, giving the input ﬁle name on the command line.[2](#_bookmark34) For example, the command

aspic sword.inp

or just

aspic sword

will cause the program to read an input ﬁle named sword.inp and produce corresponding estimates and output ﬁles. If only the command

aspic

is given, the program looks for the default input ﬁle,

ASPIC.INP.

If the .SUM ﬁle as been enabled, summary output from each run in the directory will be written to it. The default name is aspic.sum. To use a diﬀerent name for the .SUM ﬁle, give the name on the com- mand line. For example, the command

2Most operations are also possible by dragging and dropping icons. The ASPIC Quick Reference, available from the ASPIC shortcuts folder installed on the user’s Windows Desktop, in- cludes more information on such use.

aspic sword mysum

will read the ﬁle sword.inp and create (or write to) the summary ﬁle mysum.sum, along with the usual ASPIC output ﬁle(s).

Most errors detected while ASPIC is reading the in- put ﬁle will cause the program to print a descriptive message and stop. If the message is not clear, com- paring the input ﬁle to the samples provided may reveal format errors.

# Overview of Auxiliary Programs

* 1. **Overview of ASPICP**

The auxiliary program ASPICP can be used following an ASPIC bootstrap run. It provides estimated time trajectories of population biomass and ﬁshing mor- tality rate with bias-corrected conﬁdence intervals. ASPICP is also used for making population projec- tions beyond the observed data set. When making projections, the user can specify future harvests or eﬀort levels, and the program projects biomass and ﬁshing-mortality trajectories for up to 15 years past the original data. Printer plots of the trajectories are also provided.

ASPICP reads information recorded in the BIO ﬁle of the corresponding bootstrap run. The user con- trols the program with a simple control ﬁle, default ASPICP.CTL. Thus, the ﬁrst step in using ASPICP is to create a proper control ﬁle with a text editor. De- tails of the ﬁle contents are given in §[8](#_bookmark54), and sample ASPICP input ﬁles are included in the ASPIC distribu- tion.

When starting ASPICP, the control ﬁle name is given on the command line; for example the command,

aspicp sword

or equivalently

aspicp sword.ctl

starts ASPICP as described in control ﬁle sword.ctl. All output from ASPICP is written to a ﬁle whose name is supplied by the user within the .CTL ﬁle.

* 1. **Overview of FTEST**

A small program named FTEST is provided to per- form signiﬁcance tests when comparing diﬀerent ASPIC models of a stock. The program is designed for comparing pairs of models that diﬀer only in complexity (number of parameters). The TEST pro- gram has a text user interface. To run, type

ftest

at a Windows command prompt and answer the pro- gram’s prompts.

* 1. **Overview of AGRAPH**

The Windows program AGRAPH is intended to pro- vide quick, good-quality graphics of ASPIC and ASPICP results. Preformatted time-series plots of rel- ative benchmarks and of observed and ﬁtted abun- dance indices are provided. Plots can be viewed on the screen, sent to a Windows printer, or saved as graphics ﬁles in several formats.

The AGRAPH program was not meant to meet all graphics needs of ASPIC users. Instead, it allows one to examine results quickly and to have graphics suit- able for assessment reports. Operation of AGRAPH is similar to that of any Windows program. It can also be started from the command line. For exam- ple, to make graphs from results in ﬁle sword.fit, use the command

agraph sword.fit

1. **ASPIC Input File Speciﬁcation**

ASPIC reads its input from a single ﬁle containing control parameters and data. The format of that ﬁle is described here.

## Generating a sample input ﬁle

A new feature of ASPIC 5.0 is that a sample input ﬁle can be generated with the command

aspic -help

The sample ﬁle is useful as a template for making new input ﬁles. It could also be helpful to have it available when reading this section.

## General format guidelines

The representation of values in the input ﬁle must follow certain rules, which follow from the use of Fortran list-directed read statements to read the data.

* + - The exact position of values on a line is not im- portant. However, if a line contains more than one value, they must be in the correct order.
    - When a line contains more than one value, they

must be separated by spaces (blanks). *Using* tab *characters to separate values is not recom- mended.*

The remaining rules depend upon the type of the data item (integer, real, or character).

* + - Each *real number* should contain a decimal point, an exponent (marked by the letter d or e), or a decimal point and an exponent. Examples:

1.0, 2e3, 1.3d6. (Note that the notation 2e3

means 2 *×* 103.) An integer can be used in place

of a whole real number.

* + - *Integers* must not contain decimal points or ex- ponents. Examples: 0, 2, 94541.
    - *Character strings* may be delimited by matched apostrophes or quotation marks. However, this is necessary only if the string contains embed-

ded blanks or other special characters. Exam- ples (each on a separate line):

IRF

'This is a valid string' "Another valid string"

* + - Each line must have the speciﬁed number of val- ues, separated by spaces. Values may not be otherwise arranged among lines.
    - After the speciﬁed number of values have been read from a line, the program does not read it further. Thus, the rest of the line may be used

to contain comments. Comments are included in the sample input ﬁles, preceded by pound signs, ##. The pound signs are used to make the comments stand out to the eye and do not themselves denote comments to ASPIC.

* + - After all data have been read from the ﬁle, as

Line 2: Title of analysis

This is a character string of length 110 characters or less. The title is written to the main output ﬁle to identify the particular analysis. The title will also ap- pear on graphs made with AGRAPH and projections made with ASPICP.

Since the title almost always contains spaces, it should be surrounded by quotation marks.

If the ﬁrst character in the title is an asterisk (\*), the main output ﬁle will contain control codes to acti- vate the “lineprinter” font on many laser printers. This provides a simple method of neatly printing ASPIC output ﬁles, which are 120 characters wide.

Example: "\*Run 4 for Redfin Tuna, 1994"

However, printer control codes so generated can be a nuisance when the ﬁles are *not* being printed in this way.

Line 3: Model shape and optimization control Note: In an eﬀort to make ASPIC 5.0 as compatible \* as possible with earlier versions, the input ﬁle has

the same general arrangement. However, additional control values are needed. Many of them appear on line 3, which make this section rather long.

Line 3 has a varying number of items. •

I@' The *ﬁrst value* on line 3 is a character string specifying the model shape (program shape mode).

Value Meaning

LOGISTIC Fit the logistic (Schaefer) model.

GENGRID Fit the generalized model at grid of values *or at one speciﬁed value.*

FOX Fit the Fox model (a special case of

GENFIT, below).

GENFIT Fit the generalized model and esti- mate its exponent directly.

determined by the number of years of data and

number of data series, any further contents of the ﬁle are ignored by ASPIC. Thus, additional comments may be appended to the ﬁle.

## The **ASPIC** input ﬁle, line by line

Line 1: Program mode

This is a character string of length 3, with possible values FIT (ﬁtting mode), BOT (bootstrap mode), or IRF (iteratively reweighted ﬁt mode). Further expla- nation of program modes is found in §[4.3](#_bookmark18).

I@' The *second value* on line 3 is a character string specifying the conditioning mode for the ﬁt. For more information, see §[4.5.2](#_bookmark24) on page [9](#_bookmark24).

Value Meaning

YLD Condition ﬁtting on yield (recommended for most analyses).

EFT Condition ﬁtting on ﬁshing-eﬀort rate.

I@' The *third value* on line 3 is a character string specifying the objective function.

Value Objective function

SSE Sum of squared errors (recommended de- fault).

LAV Least absolute values (robust objective function).

Before setting values 4, 5, and 6 on line 3, deﬁne

*φ* as the decimal fraction deﬁning model shape,

*φ ≡ B*MSY*/K* (thus 0 *< φ <* 1). Then deﬁne Φ *=*

nint*(*100*φ)*, where “nint” is the nearest integer func- tion, and thus 0 *<* Φ *<* 100).

For example, in the logistic model, *B*MSY *= K/*2, so *φ =* 0*.*5 and Φ *=* 50. For the Fox model, *φ =* exp*(−*1*) ≈* 0*.*3679 and Φ *=* 37.

I@' The *fourth value* on line 3 is an integer, the lowest Φ to consider in GENGRID and GENFIT shape

modes. A reasonable default might be 25. If present, this value is ignored in LOGISTIC and FOX shape modes.

I@' The *ﬁfth value* on line 3 is an integer, the highest

Φ to consider in GENGRID and GENFIT shape modes.

a reasonable default might be 75. If present, this value is ignored in LOGISTIC and FOX shape modes.

I@' The *sixth value* on line 3 is an integer whose in- terpretation depends on the shape mode chosen:

Shape mode Meaning of ﬁfth value

GENGRID Step size for grid of shape param-

eters examined.

GENFIT Starting value for shape parame-

ter.

LOGISTIC Ignored.

FOX Ignored.

In GENFIT shape mode, a reasonable default is often to use the logistic, i. e., to use 50.

I@' The *seventh value* on line 3 is a real number that sets bounds to constrain the generalized ﬁt near the logistic. For example, a value of 8*.*0 means that MSY

for the generalized ﬁt must lie between 1*/*8*×* and

8*×* the MSY estimated in the logistic ﬁt. This ad hoc

method is used to increase stability in ﬁtting. In sim- ulation studies, the value 8.0 has proven a reason- able default. The parameter *K* is also constrained, but on much wider bounds.

*To use a speciﬁed model shape, use* GENGRID *shape mode, set the fourth and ﬁfth values equal to the spec- iﬁed shape, and set the sixth value (the step size) to zero.*

Examples of line 3

**Example 1.** Specify a grid-search for the shape pa- rameter between Φ *=* 40 and Φ *=* 60 (a moderate range around the logistic, Φ *=* 50). Use a step size in Φ of 5. Set bounds on MSY of 1*/*8*×* to 8*×* the logistic

estimates. Use SSE (least squares) objective function (in log space), conditioned on (matching) the eﬀort in the input ﬁle. Note that EFT here is the equivalent to the optimization mode CAT in ASPIC 3.x.

GENGRID EFT SSE 40 60 5 8.0

**Example 2.** Fit the generalized model conditioned on eﬀort, with bounds of Φ *= [*40*,* 70*]* and starting value Φ *=* 50. Constrain the generalized estimate

closer to the logistic than in Example 1.

GENFIT EFT SSE 40 70 50 5.0

**Example 3.** Use a ﬁtting procedure that could be accomplished with ASPIC 3.x. Fit the logistic model, conditioned on yield. When ﬁtting the lo- gistic model, only the ﬁrst three values on line 3 are required.

LOGISTIC YLD SSE

Line 4: Verbosity & output ﬁle control

This is a single integer value that controls the amount of output printed to the screen during ex- ecution (the “verbosity”) and whether the optional

\*.SUM and \*.PRN ﬁles (§[4.7](#_bookmark29)) are generated. To gen- erate those ﬁles, set this value in the range 10–14; to suppress the ﬁles, set the value in the range 0–4.

Note that the optional ﬁles can also be controlled by options in the .INI ﬁle (§[7](#_bookmark53) on page [19](#_bookmark53)). If the ﬁles are turned on *in either way,* they will be generated.

To control the amount of screen output, set the value within the ranges 0–4 or 10–14. In terms of screen output, 0 is equivalent to 10, 1 is equivalent to 11, and so on. A setting of 0 or 10 gives very little screen output; 4 or 14 are intended for debugging, and give too much for any practical purpose. The recom- mended value is 2 or 12 (moderate screen output).

Line 5: Number of bootstrap trials

An integer 0 *≤ n ≤* 1000. A reasonable default is 500. Although this is used only in BOT program mode, in other program modes it still must be set to

a valid integer, which will be ignored.

Line 6: Monte Carlo searching

This line contains two integers to control the op- tional Monte Carlo (MC) search during ﬁtting.

I@' The *ﬁrst value* on line 6 may be 0 to disable the Monte Carlo search during ﬁtting; 1 to enable MC searching; or 2 for repeated searching. Turning MC on can help when a repeatable solution is otherwise diﬃcult to ﬁnd. Unfortunately, when strong local minima are present, the MC search can cause more problems than it solves. The author recommends leaving it oﬀ unless it is deﬁnitely needed.

I@' The *second value* on line 6 sets the initial num- ber of Monte Carlo trials. when repeated searches are enabled, this number is reduced by the program in searches after the ﬁrst. Even if the ﬁrst number on this line is 0, the second number is needed as a placeholder.

As stated above, the recommended procedure is to leave searching oﬀ unless it is needed. If turned on, suggested parameters are: 1 50000. Monte Carlo searching, and particularly repeated searching, in- creases execution time considerably.

Line 7: Convergence criterion for optimizer

This convergence criterion is a real number denoted *ε*1. After each adjustment of the simplex, the ob- jective function is computed for each vertex of the simplex. Convergence is deﬁned to occur when the following condition is met:

2 *|L*1 *− L*0*| < ε .*

1

Line 8: Restart control

Randomized restarts are used by the ASPIC opti- mizer to avoid local minima. The two values (real, integer) on line 8 control this mechanism.

I@' The *ﬁrst value* on line 8 is the tolerance *ε*2 for ending restarts. When objective function values from *k* restarts in a row agree to within this toler-

ance, the solution is accepted. The recommended value is *ε*2 *=* 3*×*10*−*8, which is written as 3d-8 in the

input ﬁle. Changing this value is not recommended.

I@' The *second value* on line 8 sets *k*, the minimum \*

number of restarts required. The recommended de- fault for this integer value is 6. Larger numbers can be used if needed to obtain a solution not overly sen- sitive to starting values. (The value used by ASPIC 3.x

was *k =* 3, increased in version 3.89 to *k =* 6.)

Line 9: Control of iterative computations

Iterative computations are used by ASPIC in several places. The two values (real, integer) on line 9 con- trol two important sets of iterative computations.

I@' The *ﬁrst value* on line 9 is the tolerance *ε*3 for computing the annual ﬁshing mortality rate (*F* ). When conditioning on yield, an iterative method must be used to estimate *F* ; it continues until suc- cessive estimates are within *ε*3. The recommended

value is *ε*3 *=* 1*×*10*−*4, which is written as 1d-4 in the

input ﬁle. Changing this value is not recommended.

In EFT optimization mode, *ε*3 must be present in the input ﬁle, but it is ignored.

I@' The *second value* on line 9 is the number of time \*

steps used per year for the generalized model, range 2–100. A reasonable default is between 12 and 24 steps. The choice aﬀects execution speed (§[2.2.1](#_bookmark8)).

When ﬁtting the logistic (Schaefer) model, this num- ber is not required. If present, it is ignored.

*L*1 *+ L*0

where *L*1 is the highest objective-function value in the simplex and *L*0 is the lowest.

The recommended value is *ε*1 *=* 1 *×* 10*−*8 , which is written as 1d-8 in the input ﬁle. Using a diﬀerent value is not recommended.

Line 10: Maximum estimated *F*

This line contains a real number specifying the max- imum allowable estimate of *F* . This maximum (used when conditioning on yield) serves to aid the op- timizer. The recommended default is 8d0, which works well in most cases.

Line 11: Statistical weight for *B*1 penalty in ob- jective function

This line contains a real number that controls the inﬂuence of the penalty term on *B*1 *> K* (see §[4.5.1](#_bookmark23)). To omit the penalty term, set this to 0d0. To use the penalty term, enter a positive real number (usu- ally 1d0). The penalty is useful in analyses showing a sharp decline in relative abundance in the initial years; such data sets can otherwise result in an ex- tremely high estimate of *B*1.

The recommended default is no penalty. If the re- sulting estimate of *B*1*/K* is too high, the analyst can try either the penalty term or ﬁxing *B*1*/K* rather than estimating it. Either approach can aﬀect estimates of management quantities; sensitivity analyses are use- ful to examine this. The penalty term is described in [Prager](#_bookmark64) ([1994](#_bookmark64)); ﬁxing *B*1, in [Punt](#_bookmark74) ([1990](#_bookmark74)).

Line 12: Number of data series

This line has a single integer from 1 to 10 that indi- cates how many data series are to be analyzed. The types of allowable data series are summarized in Ta- ble [1](#_bookmark27) on page [10](#_bookmark27).

Line 13: Series-speciﬁc statistical weights

The program reads as many real numbers from this line as series were speciﬁed on the preceding line. The statistical weight *wi* for series *i* is multiplied by each squared residual for that series when the objective function is computed. When IRF program mode is used to analyze more than one data series, the *wi* are adjusted to implement inverse-variance weighting. They can all be set to unity, best written 1d0 in the input ﬁle, unless there is reason to set them otherwise.

Line 14: Starting guess for *B*1*/K*

This line contains a single real number between zero and one. Set this value based on your belief about the stock’s condition at the start of the data set. In the absence of other information, a reasonable de- fault is 0.5.

Line 15: Starting guess for MSY

In the absence of other information, half the largest yield can be used as a starting guess. This should be entered as a real number.

Line 16: Starting guess for *K*

In the absence of other information, a reasonable guess is 2 to 20 times the largest recorded yield. This should be entered as a real number.

Line 17: Starting guess(es) for *q*

The program reads as many real numbers from this line as there are data series speciﬁed on line 12. The meaning of *q* depends on the data type that it refers to. When it refers to an eﬀort–yield data series (code CE in Table [1](#_bookmark27)), *q* is the catchability coeﬃcient. When it refers to a biomass index data series (codes I0, I1, or I2, Table [1](#_bookmark27)), *q* is the constant relating the index data to the internal ASPIC estimates of biomass; e.g.,

if *q =* 2*.*0, the index data are divided by 2.0 before

being compared to the estimated biomass. When it refers to a biomass estimate series (codes B0, B1, or B2, Table [1](#_bookmark27), the user’s value of *q* is ignored, but a number must be present as a placeholder.

For technical reasons, optimization is more diﬃcult • when *q* is large. Thus, a successful result is most often obtained when the catch and index data are scaled so that all *qi <* 0*.*01. This, of course, does

not apply to Bn data series, for which by deﬁnition

*qi =* 1.

Line 18: Flags to estimate (or ﬁx) individual parameters

If line 12 speciﬁes *I* data series, the program reads *I +* 3 integer values (“ﬂags”) from this line. The ﬂags refer, in order, to *B*1*/K*, MSY, *K*, and *qi,i =*

*{*1*,* 2*,...,I}*. Set the ﬂag to 1 to estimate the cor-

responding parameter, or 0 to keep the parameter constant at the starting guess. No ﬂag should be set to any value other than 0 or 1. Although *qi* is not es-

timated for some series types, *I +* 3 ﬂags are always

required.

Line 19: Bounds on MSY

This line contains maximum and minimum bounds on the estimate of MSY. These two real numbers are used to limit the solution to reasonable values. The user deﬁnes what is reasonable by setting these values. If ﬁnal estimates are at either constraint, an error message is printed on screen and in the output ﬁle. Bootstrap trials falling outside these bounds are discarded.

Line 20: Bounds on *K*

This line contains maximum and minimum bounds on the estimate of *K*. They are used in the same way as the bounds on MSY.

Line 21: Random number seed

Use a large (7-digit) positive integer. Diﬀerent num- bers result in diﬀerent random number sequences. Using the same seed allows duplication of a previous run.

Using a diﬀerent seed should result in the same an- swer (within expected computation errors); if results are substantially diﬀerent, at least one of the solu- tions was a local minimum. The user can attempt to remove sensitivity to random number seed by increasing the number of restarts required (second number on line 8).

Line 22: Number of years in data set

(c2) *Second number —* a real number whose meaning depends on the series type. For type CE, it is the ﬁshing-eﬀort rate *f* for the year. For type CC, it is the average rel- ative abundance (usually based on CPUE). For types B0, B1, or B2, it is a stock-biomass estimate. For types I0, I1, or I2, it is a rel- ative abundance value.

(c3) *Third number —* a real number, required for CE or CC series only, giving the total yield (catch in biomass) from the ﬁshery for that year. For other types of series (B*n* or I*n*), the third number is not needed and if present, it is ignored.

Although yield–eﬀort data series are designated type • CE, eﬀort is entered before yield on these lines. Sim- ilarly, in series type CC, the relative abundance ap- pears before the yield.

As noted in §[6.3](#_bookmark47), it is recommended (although not absolutely necessary) to scale the catch and index

The total number *y* of years described by the input

ﬁle, including any years with missing values. Within

data so that all *qi*

*<* 0*.*01. This does not apply to Bn

the ﬁle, each data series must be of length *y* and describe the same speciﬁc years. Nonoverlapping series can be accommodated by padding each series with missing values or zeroes as appropriate.

Following lines: Individual data series

There must be one data block (group of lines) for each data series. Each block should include data for all *y* years; thus, thus each data block must be the same length *y* . The composition of each block is as follows —

1. On the ﬁrst line of the block, a series title (char- acter string, length *≤* 40, in quotes). Example:

''Spring survey & total landings''

1. On the second line of the block, a character string of length 2 with the type code for the se- ries. Type codes are listed in Table [1](#_bookmark27) (p. [10](#_bookmark27)).
2. Starting on the third line of the block, one data line for each year, with the following data on each line, separated by blanks —

(c1) *First number —* the year or other ID num- ber. These are consecutive integers and must be identical from block (data series) to block. Numbers greater than 9999 will not print correctly.

data series, for which by deﬁnition *qi =* 1.

## Common questions about data series

* + 1. Missing values and zeroes

Missing or zero data values are allowed in an ASPIC input ﬁle in some cases, depending on the condition- ing mode and type of data series. All possible cases are described in Table [3](#_bookmark50), along with the action taken by ASPIC. A data line with a missing value or with

*f =* 0 does not contribute to the objective function;

however, the information present on the line is used in the analysis and does inﬂuence the estimates.

Any negative data item in the input ﬁle is considered a missing value by ASPIC. Thus a value can be set missing by inserting a minus sign in front of it, and the value can be restored in a later analysis by remov- ing the minus sign. When a missing value appears in the ASPIC input ﬁle, an estimate of the underlying value appears in the output ﬁle.

Missing values are always distinct from true zero val- ues. Zero should never be used to indicate a missing value, and a negative number should never be used for an observed zero.

Zero values of the abundance measure (CPUE) are never permitted, because it is assumed that the re- source is not extinct during the analysis period. If an abundance index calculated prior to using ASPIC is

*Table 3. Actions taken by ASPIC when data series include data record(s) with missing value(s) or zero(es). Dash (—) indicates normal data (neither missing nor zero).* M *indicates a record with missing datum;* Z, *with zero datum. Series type* Index *includes* I0*,* I1*,* I2*,* B0*,* B1*, and* B2 *series.*

Cond. Series CPUE or

mode type eﬀort Yield Action by ASPIC

YLD CC M — Fit; estimate missing CPUE.

YLD CC — M Stop: missing yield not allowed when conditioning on yield.

YLD CC M M Stop: missing yield not allowed when conditioning on yield.

YLD CC M Z Fit with *F = Y =* 0 (no ﬁshing).

YLD CC Z M Stop: missing yield not allowed when conditioning on yield.

YLD CC Z — Stop: zero CPUE never allowed.

YLD CC — Z Fit with *F = Y =* 0 (no ﬁshing).

YLD CC Z Z Stop: zero CPUE never allowed.

YLD CE M — Fit; estimate missing CPUE.

YLD CE — M Stop: missing yield not allowed when conditioning on yield.

YLD CE M M Stop: missing yield not allowed when conditioning on yield.

YLD CE M Z Fit with *F = Y =* 0 (no ﬁshing).

YLD CE Z M Stop: missing yield not allowed when conditioning on yield.

YLD CE Z — Stop for error: when *F =* 0, *Y* must be 0.

YLD CE — Z Stop: zero CPUE never allowed.

YLD CE Z Z Fit with *F = Y =* 0 (no ﬁshing).

YLD Index M — Fit; estimate missing CPUE.

YLD Index Z — Stop: zero CPUE never allowed.

EFT CC M — Stop: missing CPUE not allowed when conditioning on eﬀort.

EFT CC — M Stop: missing yield not estimable in this case.

EFT CC M M Stop: missing eﬀort not allowed when conditioning on eﬀort. EFT CC M Z Stop: missing eﬀort not allowed when conditioning on eﬀort. EFT CC Z M Stop: zero CPUE never allowed.

EFT CC Z — Stop: zero CPUE never allowed.

EFT CC — Z Fit with *F = Y =* 0 (no ﬁshing).

EFT CC Z Z Stop: zero CPUE never allowed.

EFT CE M — Stop: missing eﬀort not allowed when conditioning on eﬀort.

EFT CE — M Fit; estimate missing catch

EFT CE M M Stop: missing eﬀort not allowed when conditioning on eﬀort.

EFT CE M Z Stop: missing eﬀort not allowed when conditioning on eﬀort.

EFT CE Z M Estimate with *F = Y =* 0 (no ﬁshing).

EFT CE Z — Stop; if *F =* 0, *Y* must be 0.

EFT CE — Z Stop: zero CPUE never allowed.

EFT CE Z Z Estimate with *F = Y =* 0 (no ﬁshing).

EFT Index M — Fit; estimate missing eﬀort.

EFT Index Z — Stop: zero CPUE never allowed.

zero in a given year, one could try using a small num- ber (e. g., 20% to 50% of the lowest nonzero value) in its place. Use of an *extremely* small number (e. g., 1% of the lowest nonzero value) will usually result in a large residual during the ASPIC analysis; such

a residual can inﬂuence the results strongly. Thus, converting zeroes to very small numbers is not rec- ommended.

* + 1. Allocation of yield among series

When analyzing more than one data series, it is not always possible — or desirable — to associate a yield with each measure of ﬁshing eﬀort rate or relative abundance. A common example is having several abundance indices for a stock, but only the total an- nual yield. This section aims to describe how ASPIC assumes yield is allocated among data series.

Yield is entered in both CE and CC series. Because ASPIC derives an abundance index from each CE se- ries, it is important that the yield in a CE series cor- respond to the ﬁshing-eﬀort rate in the same series.

In contrast, it is not necessary for the abundance in- dex in a CC series to correspond to the yield in the same series. For example, a valid CC series might have an abundance index computed from one ﬁsh- ery on a stock, paired with the total catch from all ﬁsheries on that stock.

Despite the above, it is important that yield, summed across series, represent a constant proportion (usu- ally assumed 1*.*0) of total removals. Changes in that proportion, whether due to reporting changes or changes in discarding practices, violate a funda- mental assumption of ASPIC (and of most other as- sessment models). The consequences of that viola- tion will depend on its severity.

* 6.4.3 Caution on zeros, missing values

The author has attempted to ensure that results of computations including missing and zero values are correct under all combinations of data series type, conditioning mode, and model shape. To that end, a simple test has been done of every combination shown in Table [3](#_bookmark50). Still, some cases occur infre- quently in real data and so have not been tested re- peatedly. Users are urged to examine results criti- cally when missing and zero values are used and to advise the author if any problems should arise.

# Advanced Options for **ASPIC**

Several advanced options for ASPIC are available if the ﬁle ASPIC.INI is found in the directory where the program is run (usually the directory contain- ing the data ﬁles). ASPIC.INI is a simple ASCII ﬁle.

It should have on each line an option name, an *=*

sign, and a value. Depending on the option, the value should be a number or a binary indicator. If binary, the values 1 or T turn the option on; values 0 or F turn it oﬀ. Comments may be included in

ASPIC.INI as lines beginning with the hash charac- ter #.

The options are listed here with their meanings—

* + sfile—If this is turned on, ASPIC writes an ex- tra output ﬁle (extension .PRN) in an ASCII for- mat readable by S-Plus or R with a command like

read.table("sword.prn",header=T)

The ﬁle can also be imported by spreadsheet programs, as it has ﬁxed column widths de- limited by spaces. After importing the ﬁle, it may be necessary to use search-and-replace to change the text ’NA’ (the missing-value indicator in the .PRN ﬁle) to the spreadsheet’s missing- value indicator.

The .PRN ﬁle has the observed and estimated CPUE series and estimated series of *F/F*MSY and *B/B*MSY. It is intended mainly to facilitate mak- ing time-series plots.

* + sumfile—If turned on, this option causes ASPIC to write in the current directory a summary ﬁle (ﬁle extension .SUM) with summary parameter

estimates for each run made in that directory. The SUM ﬁle also can be read easily by S-language packages. It is intended mainly for use in simu- lation studies.

* + ci—This option, if set to an integer *n*, 50 *< n <* 100, adjusts the intervals calculated and printed in bootstrap mode. ASPIC prints two sets of con-

ﬁdence intervals. The defaults are 80% and 50% intervals. With this option the 50% intervals can be replaced by another value.

* + ljcode—If turned on, this causes the main out- put ﬁle to be begun with an escape code that turns on the “lineprinter” font of most laser

printers. This is a small font that allows ASPIC output ﬁles (which are 120 characters wide) to be printed without line breaks. (The same eﬀect can be achieved by beginning the run title in the input ﬁle with an asterisk.)

Of the preceding options, only the option ci cannot be controlled by other means.

1. **ASPICP Input File Speciﬁcation**

The control ﬁle for ASPICP is relatively short; it should have the ﬁle extension .CTL. For the cor- rect way to represent diﬀerent data types in the ﬁle,

see §[6.2](#_bookmark40). A sample ﬁle is provided with the ASPIC

distribution.

## Line by line

Line 1: Projection title

This is a character string, length *≤* 70. The title usu- ally contains blanks and should be delimited by quo- tation marks. The ASPICP output ﬁle will also include

the title of the original ASPIC run.

Line 2: Name of BIO ﬁle

A character string specifying the name of a BIO ﬁle from an ASPIC bootstrap run. Results from ASPICP will be based on the data in that ﬁle.

Line 3: Name of output ﬁle

A character string with the name of the ASPICP out- put ﬁle. If the ﬁle already exists, it will be overwrit- ten. For compatibility with AGRAPH, use the ﬁle ex- tension .PRJ.

Line 4: Any real number

A real number, not used at present, must be put here as a placeholder. It is best to use zero (represented 0d0), in case this option is implemented later.

Line 5: Number of years to skip at start of plots

An integer with recommended values 0 to 3. The ﬁrst few years of biomass and mortality estimates are especially imprecise. Also, analysis of certain data sets can give in very high estimated biomasses in the ﬁrst few years. Thus, omitting the ﬁrst few years from the plots can be useful.

Line 6: Number of years of projections

Integer between 0 and 15. The longer projections ex- tend, the more speculative they are. For that reason, early versions of ASPICP limited projections to only 10 years, but some users found that overly restric- tive. Projections are theoretical constructs and are most useful when comparing management strate- gies, rather than as forecasts of the future.

Following lines: management regime to project

Each following line has data for one projection year; the number of lines should equal the number of years speciﬁed on line 6. On each line, enter a real number followed by a single character. The real number represents the yield or relative ﬁshing mor- tality rate to be applied that year, and the character tells which type of value the number is. For example, line 7 of the .CTL ﬁle might read

1.456d3 Y

to indicate that in the ﬁrst projection year, a yield of 1,456 units will be taken. Thus, lines ending in Y are used for making projections conditioned on quota (TAC) management measures.

As another example, line 8 of the .CTL ﬁle might read

0.85d0 F

to indicate that in the second projection year, the ﬁshing eﬀort rate will be 85% of the rate in the ﬁ- nal year of the original data. Thus, lines ending in F are used for making projections based on propor- tional reductions in ﬁshing mortality rate. This use of relative values allows *F* -based projections to be made with reasonable conﬁdence even when the es- timated ﬁshing mortality in absolute terms is quite imprecise.

F lines and Y lines can be mixed in the .CTL ﬁle. That might be done, e. g., when yield in the ﬁrst pro- jection year is already known, and management in subsequent years is to be by control of ﬁshing ef- fort.

* 1. **Sample ASPICP input ﬁle**

"Case with Y02=Y01; F03 to F07 = F(MSY)"

test.bio test.prj 0d0

0

6

1200 Y

0.55 F

0.55 F

0.55 F

0.55 F

0.55 F

# Interpretation of **ASPIC** Results

* This section explains some features of ASPIC esti- mates, and reviews considerations important when using ASPIC. [Prager](#_bookmark64) ([1994](#_bookmark64)) and [Prager et al.](#_bookmark72) ([1996](#_bookmark72)) contain additional discussion.

## Precision of parameter estimates

Production models tend to estimate some quantities considerably more precisely than others. Among the quantities more precisely estimated are maximum sustainable yield (MSY), optimum eﬀort (*f*MSY), and relative levels of stock biomass and ﬁshing mortal- ity rate. Here, relative levels means the biomass level relative to the level at which MSY is attained or the level of ﬁshing mortality relative to that at which MSY is attained.

To provide more precise estimates, then, it is often useful to divide the stock-size estimates provided by ASPIC by the corresponding estimate of stock size at MSY (*B*MSY). Similarly, the estimates of ﬁshing mor- tality rate *F* are divided by *F*MSY to obtain relative estimates. In its output ﬁles, ASPIC provides such relative estimates. The relative estimates present a more precise picture of the condition of the stock, because in normalization, the estimate of *q* — which is usually imprecise — cancels out.

In contrast, absolute levels of stock biomass (and related quantities), which include uncertainty in the estimate of *q*, are usually estimated much less pre- cisely. One cannot place nearly as much credence in the absolute estimates of stock size, *F* , or any quan- tities that depend upon them. Absolute estimates of *Bt* and *Ft* from ASPIC are provided for the mod- eler’s information and are not intended for use as management guidelines.

When two or more data series are analyzed, esti- mated ratios of catchabilities are typically estimated more precise than estimates of each *q*. Also, *K* may be estimated imprecisely or inaccurately even when MSY and *f*MSY are estimated well. Again, this reﬂects the diﬃculty of translating relative biomass changes to an absolute scale.

The starting biomass, estimated as *B*1*/K*, may be considered a nuisance parameter, and its estimate is often imprecise. [Punt](#_bookmark74) ([1990](#_bookmark74)) recommended ﬁxing

*B*1*/K =* 1*.*0 (rather than estimating it) for the Cape

hake stock oﬀ southern Africa, but it is not clear that that approach is appropriate for every stock. A

similar approach is taken in using the penalty term described in §[4.5.1](#_bookmark23).

To stabilize estimates from a particular data set, it can be useful to ﬁt the model with *B*1*/K* ﬁxed at a range of values. Although the resulting estimates of the biomass trajectory will of course diverge at the beginning, they may provide suﬃciently consistent estimates of present stock status for management purposes.

## Estimating several catchability coeﬃcients

ASPIC can use more than one data series in esti- mation. The analyst should be aware that the un- derlying assumption is that each abundance mea- sure reﬂects the entire stock, except for random er- ror. Thus, using this feature is similar to deriving an abundance index from each series and averaging them together.

It is not recommended to use abundance indices that are uncorrelated or negatively correlated with one another, unless their overlap is short. When abun- dance indices present diﬀerent pictures, CPUE might instead be standardized with a model to remove ef- fects of vessel type, area, gear, season, etc., before ﬁtting an assessment model. The resulting index of yearly abundance can then be used as a ‘CC’ se- ries with the total catch. This provides quicker and more reliable estimation from ASPIC, but more im- portantly, it removes explainable variation from the data, which would otherwise become noise.

9.2.1 Catchability over time

The user can estimate separate catchability coeﬃ- cients for diﬀerent periods of time. This is accom- plished in practice by putting the periods of time in separate data series, each padded with zeroes or missing values as appropriate. This procedure can be used to examine hypotheses about changing catchability with time, perhaps as a result of chang- ing ﬁshing gear or changing environmental condi- tions. In interpreting such models, there are several considerations.

One concern is estimating whether the improvement in ﬁt obtained from a more complex model is statis- tically signiﬁcant. An ASPIC model with time-varying catchability can be tested against the base model (i.e., the simpler model with constant catchability) with an *F* –ratio test. Here *F* is the *F* distribution of statistics, not ﬁshing mortality rate. The test statis- tic is

*F∗ =*

*(*SSE*s −* SSE*c)/ν*1

SSE*c/ν*2

*,* (2)

## 9.3 Estimation diﬃculties

The information in this section is central to obtain- •

ing correct results. Please read it thoroughly.

where SSE*s* and SSE*c* are is the error sums of squares of the simple and complex models, respectively; *ν*1 is the diﬀerence in number of estimated parameters between the two models; and *ν*2 is the number of data points less the total number of estimated pa-

rameters. The signiﬁcance probability of *F∗* can be

obtained from standard tables of the *F* -distribution with *ν*1 and *ν*2 degrees of freedom.

A small program called FTEST is supplied with ASPIC to facilitate making certain such tests. This program assumes that the same data are used for both mod- els, but are divided into diﬀerent periods with dif- ferent estimates of *q*. The weighting for the penalty term (line 11 in the ASPIC input ﬁle) should be set to zero for this *F* -ratio test to be theoretically correct. The FTEST program is interactive, and takes all input from the screen.

Three often-repeated caveats apply when using the *F* -ratio test for this purpose. First, hypothesis tests are invalid when suggested by examination of the data. Instead, the test should be suggested by ex- ternal information, such as changes in gear. Sec- ond, the signiﬁcance of a series of tests is less than that of a single test. For information on this point, consult a reference on multiple comparisons (e. g., [Klockars and Sax](#_bookmark81) [1986](#_bookmark81)). Third, hypothesis tests gen- erally assume correct speciﬁcation of the model. Thus, signiﬁcance probabilities of tests on assess- ment models are always approximate. This caveat is especially important when there is evidence that the model does not ﬁt well.

A nonparametric test of the null hypothesis *q*1 *= q*2

can be conducted from the ﬁtting results. This test is constructed by examining the bootstrap estimates of the ratio of the two catchability coeﬃcients. As an example, assume that the alternative hypothesis

is that *q*1 */= q*2. Then the null would be rejected

at *P <* 0*.*05 if a bias-corrected 95% conﬁdence in-

terval on *q*1*/q*2 did not include the value 1.0. Like the *F* test, this test is approximate because of the possibility of speciﬁcation error. In addition, boot- strapping residuals may underestimate the true vari-

ability present in a time series ([Freedman and Peters](#_bookmark75) [1984](#_bookmark75)). This has been addressed to some degree in the current version of ASPIC by the adjustment made to the residuals before bootstrapping is begun.

The optimization method used in ASPIC ([Nelder and Mead](#_bookmark84) [1965](#_bookmark84)) is quite robust, but in unmodiﬁed form frequently stops at local minima (these represent sub-optimal solutions). This has been addressed in ASPIC with a restarting algorithm that requires the same solution to be found several times in a row before it is accepted. In the author’s experience, the resulting optimizer is reasonably eﬀective at avoiding local minima.

Nonetheless, ASPIC, like other programs that at- tempt complex nonlinear optimization, occasionally ﬁnds local, rather than global, minima. Two features of the program—beyond the restarting algorithm al- ready mentioned—are available to detect and rem- edy this problem. First, solutions obtained at local minima are often not reasonable, and this will often cause one of the parameters to be estimated at either its minimum or maximum bound. In such a case, a warning message is printed, both on screen and in the output ﬁle.

A second feature that can help avoid local minima is an optional Monte Carlo phase of estimation. When enabled, this tries to improve the initial ﬁt by ran- domly searching for a better one in the neighbor- hood of the initial ﬁt. If multiple searches are en- abled, a shorter Monte Carlo search takes place peri- odically during ﬁtting. Although such searches con- siderably increase the time required to ﬁnd a solu- tion, they can be helpful in avoiding local minima. If a solution is diﬃcult to ﬁnd, it can be helpful to enable the Monte Carlo searches.

When ﬁtting diﬃcult data sets, it can be useful to make several runs with diﬀerent random number seeds. Agreement among a number of runs suggests that the solution is stable.

Occasionally ASPIC fails to converge to a minimum at all. This often indicates that the data do not ﬁt the model very well, which can sometimes be ver- iﬁed by examining the results with AGRAPH. When there is no ﬁt, the input ﬁle should be checked for er- rors (e. g., reversed catch and eﬀort values). Rarely, changing the maximum value of *F* allowed (line 10 of the ASPIC input ﬁle) can improve convergence, if the problem occurs in EFF optimization mode. If the ob- jective function appears (from the screen output) to have been near convergence, simply trying a second ASPIC run that uses the ﬁrst run’s results as starting

guesses can sometimes provide a good solution. If the model includes several data sets (ﬁsheries), it can be useful to eliminate one or more of them, at least temporarily, to see if convergence can be achieved.

If none of these suggestions is successful, estimates can often be made with the following strategy. Set one parameter (usually *B*1*/K*) to a ﬁxed value by set- ting the corresponding estimation ﬂag (line 18 of the ASPIC input ﬁle) to zero. A solution might be possi- ble conditional upon that value of *B*1*/K*. If this tech- nique leads to a solution, a range of ﬁxed values of *B*1*/K* can be tried and the solutions examined. Simi- lar values of the objective function among solutions indicate that the solutions are nearly equivalent in terms of ﬁt. Although the solutions will diﬀer some- what, they still may be useful, especially as conﬁr- matory information or if little other information is available for management.

Although ASPIC has been tested on thousands of simulated and real data sets and is believed to oper- ate correctly, errors can exist in any computer pro- grams. Any user experiencing bugs or suspected bugs is asked to send the author copies of the input and output ﬁles by email: [Mike.Prager@noaa.gov.](mailto:Mike.Prager@noaa.gov) The author attempts to correct all errors promptly.

# Program change history

## Changes in Version 3.33

Between version 2.8 and version 3.33, major changes were as follows:

* Addition of EFF mode (conditioning on yield) and estimation of missing eﬀort
* Bias corrections of conﬁdence intervals on pa- rameters and population projections (using bootstrap)
* ASPIC-P program for computing bias corrected trajectories and projections with approximate nonparametric conﬁdence intervals
* Character plots added to the program output
* The starting guess and estimate of *B*1 speciﬁed as ratios to *B*MSY
* Optional Monte Carlo phase to increase resis- tance to local minima
* Added iterative reweighting (IRF mode) when analyzing several series
* User-speciﬁed limits on MSY and *K*
* Management benchmarks *f*0*.*1 and *Y*0*.*1 com- puted
* User-speciﬁed random number seed
* Added detailed messages for errors in the input ﬁle

## Changes in Version 3.55

Between version 3.33 and version 3.55, major changes were as follows:

* Added more statistics on stock status in ﬁnal year
* Replaced *K* with MSY in parameterization
* Added CC series type to avoid manually convert- ing CPUE to eﬀort.
* Added CPUE plots to output
* Added correlation matrix among indices to out- put
* Revised IRF mode so that the sum of weights remains equal to the number of data points
* Added residual adjustment factor for boot- strapping
* Added "coverage" and "nearness" statistics
* Improved Monte Carlo search algorithm

## Changes in Version 3.82

Between version 3.55 and version 3.82, major changes were as follows:

* Several improvements to Monte Carlo search routine
* Fixed bug in plotting index (I0, I1, I2) series and improved plot layouts
* Increased maximum number of years in data from 60 to 90
* Fixed a bug that didn’t replace bad bootstrap trials
* Added LaserJet code option for output ﬁles
* Changed output for CE data series from ob- served and estimated eﬀort to observed and es- timated CPUE
* Fixed a crash when the number of bootstraps

was set to 1

* Added printout of Monte Carlo setting to output ﬁle
* Allowed user to specify input ﬁle name on com- mand line

## Changes in Version 5.00

Version numbers in the 4.xx series were used for test releases of what is now designated version 5.00. Be- tween version 3.82 and 5.00, the following additions and corrections were made:

* Added generalized model conditioned on eﬀort and on catch.
* Added user-speciﬁed number of restarts for convergence.
* Renamed conditioning modes to YLD and EFT for conditioning on yield and eﬀort, respec- tively.
* Added an optional basic, S-compatible output ﬁle (.PRN ﬁle).
* Added .SUM output ﬁle for simulation studies. File name can be passed as second command- line argument.
* Fixed a bug in which residuals r *>* 4 (in log space) weren’t plotted.
* Made changes to the .BIO output ﬁle for com- patibility of the generalized model with the new version of ASPICP.
* Changed negative correlation action from pro- gram stop to issuing a warning.
* Add FOX model shape as a special case of the generalized model.
* Revised format of the .DET output ﬁle for better compatibility with S-Plus and R.
* Made changes to unambiguously handle zero CPUE in a CC series as a missing value.
* Version 4.12: Added estimate of *Y .(F*MSY*)* to output ﬁle at suggestion of A. D. MacCall.
* Corrected handling of the rare case of a year

(1) conditioning on yield, (2) abundance index present, (3) yield is zero. A residual is now com- puted based on estimated and observed abun- dance indices.

* Added a penalty when MSY *> K* during ﬁtting.
* Printed AIC on GENFIT and GENGRID output.
* Added internal constraints on *q*.
* A sample input ﬁle (sample.inp) can be gener- ated by aspic -help on the command line.

# Source Code

The Fortran source code for this software uses cer- tain proprietary routines from the book *Numerical Recipes* by Press et al., and for that reason can not be freely distributed. Numerical Recipes Software has kindly granted their permission (ID number V95038) for the author to supply the source code to users upon speciﬁc request. However, any source code so supplied must not then be redistributed to others.

The author also wishes to be aware of all distribution of the source code, so that any useful modiﬁcations or error corrections can be made in the master copy of the software to beneﬁt all users.

If you require a copy of the Fortran source code for this software, please request it from the author. In your letter or email, please include the following:

1. Your true name, institutional aﬃliation, phys- ical address, and email address or telephone number
2. Your agreement that you will not redistribute the source code to others.
3. Your agreement that, if you modify the source code, you will not distribute any resulting pro- gram (or programs), nor the modiﬁed source code, beyond your immediate working group at your own location.
4. Your agreement that, if you modify the source code, you will ensure that your users do not redistribute either the modiﬁed source code or any resulting program or programs.
5. Your agreement that if you identify errors in the software, you will contact the author promptly so that the errors can be ﬁxed for all users.

# References

Efron, B. E., and G. Gong. 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. American Statistician 47: 36–48.

Fletcher, R. I. 1978. On the restructuring of the Pella– Tomlinson system. Fishery Bulletin 76: 515–521.

Fox, W. W. Jr. 1970. An exponential yield model for optimizing exploited ﬁsh populations. Trans- actions of the American Fisheries Society 99: 80– 88.

Fox, W. W., Jr. 1975. Fitting the generalized stock production model by least-squares and equilib- rium approximation. Fishery Bulletin 73: 23–37.

Graham, M. 1935. Modern theory of exploiting a ﬁsh- ery, and application to North Sea trawling. Journal du Conseil International pour l’Exploration de la Mer 10: 264–274.

Freedman, D. A., and S. C. Peters. 1984. Bootstrap- ping a regression equation: some empirical re- sults. Journal of the American Statistical Associ- ation 79: 97–106.

Haddon, M. 2001. Modeling and quantitative meth- ods in ﬁsheries. Chapman & Hall, Boca Raton, Florida. 406 p.

Hilborn, R., and Walters, C. J. 1992. Quantitative ﬁsh- eries stock assessment. Chapman and Hall, New York, NY. 570 p.

Klockars, A. J., and G. Sax. 1986. Multiple compar- isons. Sage University, Quantitative Applications in the Social Sciences, Paper No. 07–61.

Mohn, R. K. 1980. Bias and error propagation in logis- tic production models. Canadian Journal of Fish- eries and Aquatic Science 37: 1276–1283.

Nelder, J. A., and R. Mead. 1965. A simplex method for function minimization. Computer Journal 7: 308–313.

Pella, J. J. 1967. A study of methods to estimate the Schaefer model parameters with special reference to the yellowﬁn tuna ﬁshery in the eastern tropical Paciﬁc ocean. Dissertation, University of Washing- ton, Seattle.

Pella, J. J., and Tomlinson, P. K. 1969. A general- ized stock production model. Bulletin of the Inter- American Tropical Tuna Commission 13(3): 419–

496.

Prager, M. H. 1992a. ASPIC: A Surplus-Production Model Incorporating Covariates. Collected Volume of Scientiﬁc Papers, International Commission for the Conservation of Atlantic Tunas (ICCAT) 28: 218–229.

Prager, M. H. 1992b. Recent developments in extend-

ing the ASPIC production model. ICCAT Working Document SCRS/92/127. 10 p.

Prager, M. H. 1994. A suite of extensions to a nonequilibrium surplus–production model. Fish- ery Bulletin 92: 374–389.

Prager, M. H. 2002. Comparison of logistic and generalized surplus-production models applied to swordﬁsh, *Xiphias gladius*, in the north Atlantic Ocean. Fisheries Research 58: 41–57.

Prager, M. H., and C. P. Goodyear. 2001. Eﬀects of mixed-metric data on production model estima- tion: simulation study of a blue-marlin-like stock. Transactions of the American Fisheries Society 130: 927–939.

Prager, M. H., C. P. Goodyear, and G. P. Scott. 1996. Application of a surplus production model to a swordﬁsh-like simulated stock with time-changing gear selectivity. Transactions of the American Fisheries Society 125: 729–740.

Punt, A. E. 1990. Is *B*1 *= K* an appropriate as-

sumption when applying an observation error production-model estimator to catch–eﬀort data?

S. African Journal of Marine Science 9: 249–259.

Punt, A. E., and R. Hilborn. 1996. Biomass dynamic models. User’s manual. FAO Computerized Infor- mation Series (Fisheries), No. 10. Food and Agri- culture Organization of the U.N., Rome. 63 p.

Quinn, T. J., and R. B. Deriso. 1999. Quantitative Fish Dynamics. Oxford University Press, New York. 542 p.

Schaefer, M. B. 1954. Some aspects of the dynam- ics of populations important to the management of the commercial marine ﬁsheries. Bulletin of the Inter-American Tropical Tuna Commission 1(2): 27–56.

Schaefer, M. B. 1957. A study of the dynamics of the ﬁshery for yellowﬁn tuna in the eastern tropical Paciﬁc Ocean. Bulletin of the Inter-American Trop- ical Tuna Commission 2: 247–268.

Shertzer, K. W., and M. H. Prager. 2002. Least median of squares: a suitable objective function for stock assessment models? Canadian Journal of Fisher- ies and Aquatic Science 59: 1474–1481.

Stine, R. 1990. An introduction to bootstrap meth- ods: examples and ideas. Pages 325–373 *in* J. Fox and J. S. Long (eds.), Modern methods of data anal- ysis. Sage Publications, Newbury Park, California. 446 p.

Williams, E. H., and M. H. Prager. 2002. Comparison of two estimators for the generalized production model. Canadian Journal of Fisheries and Aquatic Science 59: 1533–1552.