Designing tools to evaluate fishery management strategies: can the scientific community deliver?

Jon T. Schnute, Mark N. Maunder, and James N. Ianelli

Schnute, J. T., Maunder, M. N., and Ianelli, J. N. 2007. Designing tools to evaluate fishery management strategies: can the scientific community deliver? – ICES Journal of Marine Science, 64: 1077–1084.

Techniques for quantitative fishery management have evolved rapidly during a period when computers, programming languages, and computational algorithms have also changed dramatically. Despite these advances, many stock assessment methods remain untested. A process of management strategy evaluation (MSE) could potentially rectify this problem, but it would require a framework in which to conduct systematic tests. We survey the tools currently used for stock assessments and discuss the development of new standards for testing management procedures. A successful project would depend on human skills scattered among various nations, organizations, and academic disciplines. Analogies from civil engineering illustrate the discipline and collaboration required for an effective outcome. If the world community of fishery scientists could design, build, and support such a project, it would revolutionize the theory, teaching, and practice of scientific fishery management.

Keywords: AD Model Builder, ADOL-C, Fisheries Library in R, management strategy evaluation, next generation, stock assessment, R, software.

Received 17 May 2007; accepted 15 June 2007; advance access publication 18 July 2007.

J. T. Schnute: Fisheries and Oceans Canada, Pacific Biological Station, 3190 Hammond Bay Road, Nanaimo, British Columbia, Canada, V9T 6N7. M. N. Maunder: Inter-American Tropical Tuna Commission, 8604 La Jolla Shores Drive, La Jolla, CA 92037-1508, USA. J. N. Ianelli: Alaska Fisheries Science Center, REFM Division, NMFS/NOAA, Bldg 4, 7600 Sand Point Way NE, Seattle, WA 98115, USA. Correspondence to J. T. Schnute: tel: +1 250 7567000; fax: +1 250 7567053; e-mail: schnutej@pac.dfo-mpo.gc.ca.

Introduction

Fisheries stock assessment methodology has progressed rapidly during the past three decades. Most of this progress would have been impossible without corresponding improvements in the tools available for computation. For example, the modern technique of management strategy evaluation (MSE; de la Mare, 1998; Butterworth and Punt, 1999; Smith *et al.*, 1999; Sainsbury *et al.*, 2000; Butterworth and Rademeyer, 2005; Schnute and Haigh, 2006) requires extensive simulations to test the effectiveness of proposed management strategies. Obviously, this depends on an ability to perform simulations rapidly and efficiently.

As stock assessment science matures, the computational requirements will continue to increase, but fast hardware and software alone do not guarantee progress. To be really effective, the world's research community needs to adopt standards that facilitate *cumulative* knowledge, in which clear guidelines emerge for the conduct of stock assessments and their practical application to management. We discuss the current situation and offer suggestions for the future. We also speculate on the prospects for a computational framework that would meet the needs of stock assessment scientists worldwide. In particular, we examine the implications of this idea for designing the next generation of stock assessment software.

Software environments

Always eager to try new approaches, fishery scientists have tested many computing environments. Maunder *et al.* (in press) document some of the history, including the use of classical programming languages, spreadsheets, statistical packages, and customized or configurable stock assessment software. Readers may recognize some or all of the names in a (very incomplete) list: BASIC, FORTRAN, C, C++, Pascal, APL, Python, VisiCalc, Lotus 123®, Microsoft® Excel™, Visual Basic™, SYSTAT®, Minitab®, SAS®, GAUSS™, S-PLUS®, R, ACON, AUTODIF©, ADMB©, WinBUGS, OpenBUGS, Stock Synthesis 2 (SS2), CASAL, and Coleraine. At various times, the authors of this paper have used most of these, as well as many others. Indeed, we become a bit nostalgic as we contemplate the roles that these frameworks have played in our careers. The Appendix gives historical details and websites associated with every software package mentioned in this report. It highlights the remarkably checkered history of software used to investigate fishery population dynamics.

Schnute *et al.* (1998) compared this multilingual situation with the Babel tower myth:

> After the people of Babel sought to build a tower to heaven, the Lord God devised a plan (Genesis 11: 4-7). "Behold the people is one; and they all have one language; and this they begin to do; and *now nothing will be restrained from them, which they have imagined to do* . . . . Let us go down, and there confound their language, that they may not understand one another's speech." Italics highlight the prospects for accomplishment with a common language, if the scientific community could ever agree on one.

The cosmic plan for confounding software languages seems to be working remarkably well among the community of quantitative fishery scientists!

Perhaps fishery analysts worldwide can never adopt a common software environment. The suggestion might even be counter-productive, because new computing technology will doubtless make new analyses possible in the future. Nevertheless, as the Babel story illustrates, something is lost if a scientific community attempts to make progress while using very different standards of analysis and software. Science requires repeatability, with a transparent link between assumptions and conclusions. Without a common framework at some level, results from disparate computer programs simply are not comparable. Object-oriented programming (OOP) techniques might help to mitigate this problem by providing rigorous definitions of data structures and associated methods.

Here, we focus on two streams of development that currently play important roles in our field. The first became prominent as non-linear fishery models grew to include large numbers of unknown parameters, hyperparameters, and state variables. Search algorithms to estimate all these unknown quantities benefit greatly from a computing technique known as automatic differentiation or algorithmic differentiation (AD). Griewank (2000, p. 1) traces the concept back to a 1976 PhD thesis by Johannes Zoos. In fishery science, this idea first found expression in two generations of software (Otter Research Ltd., 1994, 2000) called AUTODIF© and AD Model Builder (ADMB©). These C++ environments not only provide extremely fast modal estimates for parameter-rich non-linear models, but they also have evolved to generate Bayesian posterior samples and support other advanced statistical concepts. ADMB© has enabled analysts to use the model, data, and methods that they deem appropriate, without simplifications that make it feasible to use customized methods of parameter estimation. Consequently, a wider range of data can be included in stock assessments, and increasingly complex models can introduce greater realism into the population dynamics and associated sources of uncertainty (Maunder, 2007).

Despite the advantages of stock assessments programmed in ADMB©, relatively few analysts have the technical skill required to use it effectively. This limitation has motivated the development of general stock assessment models programmed in ADMB©, starting with Coleraine and soon followed by SS2, which is part of the US NOAA Fisheries Toolbox (NFT). Both these tools strive to be quite general, and the latter still receives continuous development and support. A somewhat different model, MULTIFAN-CL (MFCL), uses the AUTODIF© library that underlies ADMB©.

AUTODIF© and ADMB© were developed in the private sector by Otter Research (http://otter-rsch.com/). A license fee supports project development, and the source code remains proprietary. Griewank (2000) produced a public domain C++ package ADOL-C (Walther *et al.*, 2005) with source code that implements AD routines similar to those in AUTODIF©. Griewank's library was used to produce CASAL, a semi-commercial product similar to SS2. A state-owned company in New Zealand uses this package to conduct stock assessments for profit. Permission is generally given to use it for non-commercial purposes, although the source code remains proprietary. The core of CASAL essentially provides a wrapper to the ADOL-C library that gives it functionality similar to AUTODIF©.

In making these comparisons, we should point out that AUTODIF© and ADMB© offer many features not available in the rudimentary AD library ADOL-C. For example, ADMB© includes extra libraries for minimizing functions and performing other important statistical calculations. It also supplies specialized C++ object classes, such as ragged arrays with arbitrarily high dimensions. These features can greatly facilitate code design and operation, particularly since the software itself uses efficient implementation techniques, such as intelligent memory allocation.

We focus next on a second stream of development associated with the R language and environment for statistical computing (R Development Core Team, 2007). This remarkable open source project includes a large number of useful algorithms for statistical analysis and modelling, and the high level R language makes coding relatively easy. A few lines of R code might accomplish what would otherwise require dozens or even hundred of lines in C/C++, if indeed the required C/C++ libraries were readily available. R also has extensive integrated graphing capabilities that can be nicely coupled with computational products like ADMB©.

Venables and Ripley (2000) document the history of R, which has roots in the language S designed principally by John Chambers at Bell Labs (http://cm.bell-labs.com/cm/ms/departments/sia/jmc/). S has evolved through four versions, and R currently has features inherited from version 3 (S3; Chambers and Hastie, 1992) and version 4 (S4; Chambers, 1998). In particular, R supports the concept of "class" in two ways, with old-style S3 classes taken from S version 3 and new-style S4 classes taken from version 4 (Chambers, 1998, section 2.8 and chapter 7; Venables and Ripley, 2000, chapter 5). The new S4 classes particularly facilitate object-oriented programming, somewhat like that available in C++. However, because R runs interpretively, an algorithm written in R normally takes much more computing time than equivalent code written and compiled in C++.

R has some compelling benefits. It solves many important problems with a fairly easy, well-designed language for calling the requisite algorithms. It comes entirely without cost as open source software that is carefully maintained and easy to install. It has an associated Comprehensive R Archive Network (CRAN, http://cran.r-project.org/), with contributed libraries that substantially enhance the R environment. Users regularly add packages that implement new algorithms at the forefront of statistical research. A disciplined system of contribution to CRAN assures reasonable quality, even when the library includes code in C or FORTRAN. All packages, as well as R itself, are automatically compiled for a variety of computing platforms and operating systems.

Not surprisingly, this critical mass of ideas and support has caught the attention of much of the world's scientific community, including those in fishery science. Even scientists committed to ADMB© or one of its counterparts are likely to use R for representing model data and results graphically. One of us (JTS) has produced two such libraries that have direct relevance to fishery data analysis. PBS Mapping (Schnute *et al.*, 2004) enables users to plot data on maps and conduct other spatial analyses in the context of polygons that represent the world's coastlines. PBS Modelling (Schnute *et al.*, 2006) makes it easy to construct graphical user interfaces (GUIs) that facilitate model development and testing.

A group of largely European scientists has worked actively in recent years to build a Fisheries Library in R (FLR, http://flr-project.org/) that supports algorithms needed for MSE. In describing their approach, Kell *et al.* (2007) focus on interdisciplinary collaboration among biologists, ecologists, statisticians, mathematicians, economists, and sociologists. They emphasize the need for open source software that promotes transparency and technology transfer among disciplines. Specifically, their framework links "a variety of fishery, biological, and economic software packages so that alternative management strategies and procedures can be evaluated for their robustness to uncertainty before implementation". FLR takes advantage of R's S4 classes to achieve an object-orientated design. This feature gives FLR something in common with ADOL-C and AUTODIF©. All three packages use classes in the relevant language (R or C++) to define the data structures and algorithms (or methods) required for the intended functionality.

Designing comprehensive assessment software

As discussed by Kell *et al.* (2007) and illustrated metaphorically in the Babel story, fisheries stock assessment could progress more rapidly in the context of a comprehensive software and computing environment. Currently, qualified stock assessment scientists are in short supply. For example, the US National Research Council considered the problem important enough to hold a workshop in 2000 on the topic "Recruiting fishery scientists: workshop on stock assessment and social science careers" (http://www.nap.edu/books/0309073081/html/).

As more stocks are actively managed and management becomes more detailed, the need grows for new stock assessments and related analyses. Furthermore, these demands come with added pressure to use all available data and to take advantage of new technologies, such as those that provide fine-scale spatial resolution. Our science has a supply–demand problem. A greater demand for analyses cannot be met by a limited supply of analysts without an increase in efficiency. A comprehensive assessment framework could seriously address this issue, if the players worldwide agreed to support and use it. Currently, the available expertise is spread thinly throughout national and international agencies, universities, private laboratories, and consulting companies. It is also spread among diverse fields, including wildlife biology, fisheries, computing science, statistics, and mathematics.

A comprehensive software environment would also help standardize approaches and promote a wider understanding of stock assessments. For example, review panels now play an increasing role in the process of ensuring assessment quality. Standardized software, with agreed relevance and validity, can greatly facilitate the work required for reviews. This has become evident in the Stock Assessment Review (STAR) Panels for US west coast groundfish fisheries, which have consistently used SS2 during the last few years.

Any software project begins with a set of problems to be solved. If successful, it produces a language (or other interface) to generate solutions. For example, when R and its historical precursor S were developed, techniques associated with linear regression and ANOVA already existed in the statistical literature, and R functions (such as `lm`, used to fit linear models) were designed to carry out a specified set of calculations. A large number of independent textbooks already described the mathematics, and public domain algorithms existed for their implementation. The language designers had to find a convenient framework in which to make these algorithms easily available to users. The old-style S3 class structure of R played an important role by linking the numeric output (`list` objects in R) with standard methods for displaying summaries and rendering graphs.

Free software (http://www.fsf.org/), the open source concept (http://www.opensource.org/), and powerful microcomputers have brought R to the critical mass of expertise that it enjoys today. It grew under the guidance of experts who ensured the quality control necessary for a relatively bug-free platform on an international network. Furthermore, extensive user testing and feedback has helped ensure that the code actually works. Raymond (2001) calls this phenomenon Linus's Law (in reference to Linus Torvalds, who developed the kernel for Linux): "Given enough eyeballs, all bugs are shallow". At this stage, it would be astonishing if a function such as `lm` contained computational errors, given that the concepts are well understood and a large number of benchmarks must have been tested. With an extremely high level of probability, someone would have reported a bug in `lm`, if it existed.

Documentation also plays a key role in the utility of software, and by now many books give detailed descriptions of the R language and its applications. A (very incomplete) list includes Venables and Ripley (1999, 2000), Dalgaard (2002), Maindonald and Braun (2003), Murrell (2006), and Wood (2006). Most of these books have supporting R libraries, freely available from the CRAN site. The number of such books continues to grow rapidly. Not only do they document software, but also they serve as teaching aids for concepts in statistics, mathematics, and other scientific fields.

Unfortunately, the problem of fishery management probably can never be as well defined as linear regression. In our field, assessment and management techniques evolve with the software, and any comprehensive framework needs to be somewhat dynamic. Nevertheless, no software project can begin without at least some core design concepts. Like the authors of FLR, we consider MSE to be a dominant issue in the future of fisheries stock assessment. This technique requires evaluating all aspects of a management strategy, including the data to be collected, methods of analysis, management actions that follow from the analyses, and uncertain consequences of the these actions (often called implementation error). MSE helps identify management strategies that are robust to uncertainties in the stock assessment. Owing to the intensive nature of these calculations, in which the stock assessment needs to be simulated thousands of times, MSE has generally been limited to relatively simple stock assessment methods.

Figure 1 illustrates the concept; it includes two central components: a management procedure (MP, or equivalently a management strategy) and an operating model (OM). The MP includes classical "Assessment" based on available "Data", as well as a method of translating assessment results into "Management" decisions. Part of the MP includes a decision about the "Data" to be collected and used. Many current stock assessment processes end there, but the MSE paradigm insists that any useful MP should be tested by simulation before it is applied in practice.

The OM provides a means of conducting such tests with a model designed to reflect realism and uncertainty in both the population dynamics and the response by harvesters to the "Implementation" of "Management" decisions (For example, harvesters might not always respond as anticipated). Management influences only the behaviour of harvesters, with an indirect effect on the fish population. Fishery managers regulate people, not fish. The two-headed arrow connecting "Harvest Activity" and "Population Biology" reflects an interaction: harvesters impact the fish population, but also alter their own behaviour in

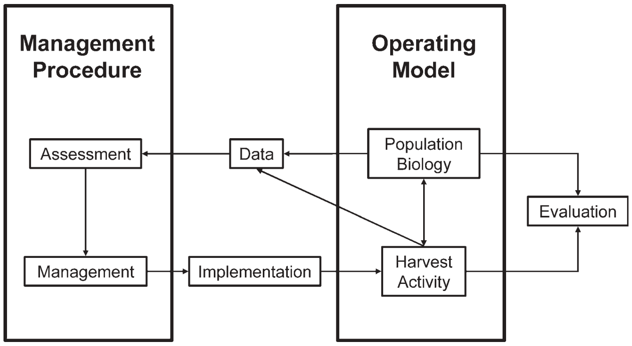

Figure 1. A framework for MSE. An OM tests a MP to produce an Evaluation. The MP depends on Data collected from harvesters and the fish population. Small rectangles highlight seven basic components of the process, as discussed in text.

response to fish biology. A process of collecting "Data" from the "Harvest Activity" and the fish "Population" gives the input to the next round of management. The "Evaluation" box on the right side of Figure 1 allows stakeholders to judge whether or not the MP has produced a good or bad outcome. This calculation can include internal data from the OM, beyond that available as "Data" collected for the MP.

Although the concepts behind Figure 1 are widely known to MSE advocates, we include the figure here to emphasize that a modern stock assessment framework would need to begin with a few core ideas. The debate shifts from a discussion of the merits of ADMB©, R, SS2, CASAL, or Microsoft® Excel™ to something more fundamental. How can the ideas in Figure 1 best be implemented in software so that a worldwide community has access to them? We recognize that the concepts in our figure may be debatable or incomplete, but any software project would need to start with something like this. Has anyone addressed this problem systematically? Yes! We encourage our readers to obtain the recent paper by Kell *et al.* (2007) about FLR. Our figure abstracts a few central ideas from their Figure 1, where they also contemplate the design of R libraries to support the requirements for each component of MSE. Their paper points the way to a conceptual framework from which standardized software for MSE might emerge. (See also ICES, 2004.)

An analogy from civil engineering illustrates why we consider MSE important. The properties of steel used in construction have been thoroughly investigated with specialized testing equipment. Engineers take advantage of these known properties when they design buildings to hold specified loads. They ignore the realities of physics at their peril, and buildings sometime collapse when the design has been done improperly. If we use a classical assessment framework to manage a fishery, it typically has not been tested. Although the design might look reasonable, given available knowledge of the stock biology, how robust is it to realistic biological complexity? We cannot test living fish populations as engineers test steel, but we can at least use simulations to investigate how our MPs might ultimately affect harvesters and fish stocks. The OM in Figure 1 plays the role of our "specialized testing equipment" for a particular MP. Just as engineers choose construction materials to suit the job, fishery scientists need to choose MPs appropriate for a particular context. This choice becomes scientific only when tested guidelines are available, along with a means of testing proposed new MPs.

Make it so!

Anyone who has watched episodes from the TV series "Star Trek: The Next Generation" will probably remember Captain Jean-Luc Picard of the Federation Starship Enterprise telling his crew to "Make it so!" After someone had suggested an ingenious course of action, the Captain would deliberate over the options and issue his memorable command. Because an episode typically ran for only one hour, things usually became "so" fairly quickly. No doubt it helped to have the talented Shakespearean actor Patrick Stewart play the role of Jean-Luc. He also helped Bill Gates announce Windows 2000 in a high-energy performance that featured musician Carlos Santana and his band (Microsoft.com, 2000).

Having described the task portrayed in Figure 1, we next ask if the community of fishery scientists could adopt such a plan and "make it so". Perhaps the TV series was a bit prophetic in suggesting that *the next generation* had the capacity to make ingenious things happen. Similarly, Bill Gates dubbed Windows 2000 *the next generation* of PC software. Such a generational shift has not yet taken place in fishery science, which still lacks tested standards such as those available for construction materials. Furthermore, it is clear that no one person or institution can act alone to develop all the required standards. It will have to be a collective effort that spans organizations and disciplines before it will be recognized as adequate. Results will need to be replicated at different times and places, as must have occurred when material standards were formalized. Potentially, the tests need to be replicated easily by anyone who thinks the existing standards might contain errors.

We neither believe nor recommend that everyone who currently uses ADMB©, Excel™, SS2, Coleraine, CASAL, or any other platform should lightly shift to a new framework. It would, however, greatly enhance the progress and general acceptance of a comprehensive framework if representatives of diverse approaches contributed knowledge based on their own experiences. Although we consider it essential that participants agree to a plan similar to that in Figure 1, all components of this testing scheme will have many variations. People currently use a great number of "Assessment" methods in the MP phase, and an unlimited number of variations could be introduced into the OM test phase. Methods of "Evaluation" require subjective decisions about the characteristics of harvesting and population dynamics that really matter.

Returning to the construction analogy, we still do not really know what materials (MPs) we want to test, we are not sure about the contexts (OMs) in which they might be applied, and we lack definitions of good and bad outcomes. These vague goals imply an important characteristic of the final product. The computational engine should have a friendly interface that allows users to explore results and participate in selecting options for management methods and evaluation criteria. Genuine success ultimately depends on good communication.

Initially, a realistic implementation of Figure 1 would require careful decisions to limit the numbers of MPs, OMs, and "Evaluation" criteria to small enough numbers for a systematic investigation to be conducted. A good management evaluation framework needs the same standards of experimental design associated with a good field programme. Ideally, the choice of software would depend on the design, but participants must start somewhere. We anticipate frequent iterations between the design and implementation phases. The FLR authors have sensibly chosen R as a suitable initial platform, partly because its new-style S4 classes can be used to formalize the design process. Aspects of the software will probably require compiled C/C++ code that runs much faster than the interpretive R engine. Preliminary tests with ADOL-C look promising, and FLR may provide C++ header files that would enable developers to invoke the corresponding AD algorithms (Iago Mosqueira and Ernesto Jardim, pers. comm.)

The R project itself illustrates a development model that could apply to a comprehensive framework for testing management strategies. It grew from relatively modest origins to a project that ultimately garnered the support of a worldwide community. It now plays an important role in many university courses and scientific projects. At the very least, it offers a powerful tool for teaching fundamental modelling concepts. Similarly, the framework here could foster a much better understanding of fishery science. With a proper development plan, users worldwide could contribute features (perhaps as R libraries) that would enrich the software's capabilities and make it indispensable for teaching fishery scientists. As with R, books could be written about the software and its results. Known strengths and limitations of various MPs could be documented systematically, similar to the measured properties of construction materials. The list of established results could be checked by students and systematically extended, perhaps by Internet users who donate computer time to test new possibilities. The Great Internet Mersenne Prime Search (GIMPS) illustrates the possibilities for such an exercise. With GIMPS, Internet users can download and run software to help search for the next "Mersenne Prime", a prime number of the form $2^n - 1$, where n is an integer. (See the Appendix for further details.)

We cannot avoid addressing one key issue as we contemplate this project. Who will pay, and who will be paid? Open source software may or may not be free of charge, but it is not produced without cost. Various organizations normally pay the salaries of those who develop it. That certainly applies to R, and it would very likely apply to any serious MSE framework. The private sector could also make important contributions, based on contracts with experts who could supply segments of the code. However, this particular project will almost certainly require open source code to garner the support of the scientific community and to receive the development required from many sources. Kell *et al.* (2007) point out that open source "facilitates better collaboration and the transfer of knowledge within and between disciplines". Unfortunately, this limitation discriminates against vendors and other contributors who do not wish to provide source code. We see no easy way past this problem, even though it may eliminate contributions from some highly qualified experts. As with R, the project may grow more rapidly as its advantages become obvious to a broader community and experts find stronger motivations to contribute.

Having made this point, we recognize that debates continue about the precise meanings of "open source" and "free". Most fishery scientists use commercial operating systems and office applications, with significant costs for the licenses. For example, Haddon (2001) provides Excel™ source code for examples in his book. Many users would probably consider this open source, despite the fact that the source code for Excel™ itself is unavailable and legally protected. Similarly, code written for ADMB© and a large number of commercial packages can be made available without charge, although such code has value only to users who possess the software required to implement it. According to the philosophy of the GNU project (http://www.gnu.org/philosophy/free-sw.html), "Free software is a matter of liberty, not price. To understand the concept, one should think of *free* as in *free speech*, not as in *free beer*". Just as good beer comes from a blend of fine ingredients (such as malt, hops, yeast, and specialty grains), comprehensive MSE software will require a delicate mix of theory, bright ideas, documentation, and code in various languages. If enough skilled people contribute to this software brewery, perhaps fishery scientists will one day be able to conduct MSE from tools available without charge on the Internet—just like free beer!

Schnute and Richards (2001) mention a problem that faces anyone currently engaged in stock assessments. An elegant model can become alluring to the analyst who invented it. After a while, the model's output starts to appear as if it gives a reasonable picture of the real world. "Like the mythical sculptor Pygmalion, the creator can fall in love with his creation and become blind to other realities". MSE acts as a healthy antidote to the Pygmalion effect. Tests might reveal surprising limitations to a long-standing MP. It would greatly advance our field if the worldwide community of fishery scientists agreed to the design a comprehensive testing system and found a way to "Make it so!".

Acknowledgements

We thank Rowan Haigh, Rick Methot, Laura Richards, and Alan Sinclair for useful contributions to the ideas in this paper. Ernesto Jardim and Iago Mosqueira helped us understand the goals of the FLR project, and also provided thoughtful reviews leading to an improved final draft.

References

Butterworth, D. S., and Punt, A. E. 1999. Experiences in the evaluation and implementation of management procedures. ICES Journal of Marine Science, 56: 985–998.

Butterworth, D. S., and Rademeyer, R. A. 2005. Sustainable management initiatives for the southern African hake fisheries over recent years. Bulletin of Marine Science, 76: 287–319.

Chambers, J. M. 1998. Programming with Data: a Guide to the S Language. Springer, New York. 469 pp.

Chambers, J. M., and Hastie, T. J. (Eds.). 1992. Statistical Models in S. Chapman and Hall, New York. 608 pp.

Dalgaard, P. 2002. Introductory Statistics with R. Springer, New York. 267 pp. (software available in the R package *ISwR*).

de la Mare, W. K. 1998. Tidier fisheries management requires a new MOP (management-oriented paradigm). Reviews in Fish Biology and Fisheries, 8: 349–356.

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*. Canadian Journal of Fisheries and Aquatic Sciences, 55: 2105–2116.

Griewank, A. 2000. Evaluating derivatives: principles and techniques of algorithmic differentiation. Frontiers in Applied Mathematics 19. Society for Industrial and Applied Mathematics. 369 pp.

Haddon, M. 2001. Modelling and Quantitative Methods in Fisheries. Chapman & Hall/CRC Press, Boca Raton, FL. 406 pp. (software available at http://www.utas.edu.au/tafi/TAFI_Download.htm).

Hampton, J., and Fournier, D. A. 2001. A spatially disaggregated, length-based, age-structured population model of yellowfin tuna (*Thunnus albacares*) in the western and central Pacific Ocean. Marine and Freshwater Research, 52: 937–963.

ICES. 2004. Report of the Working Group on Methods of Fish Stock Assessments, 11–18 February 2004, Lisbon, Portugal. ICES CM 2004/D: 03 Ref. G, ACFM. 232 pp. http://www.ices.dk/reports/RMC/2004/WGMG/wgmg04.pdf

Kell, L. T., Mosqueira, I., Grosjean, P., Fromentin, J-M., Garcia, D., Hillary, R., Jardim, E., *et al.* 2007. FLR: an open-source framework for the evaluation and development of management strategies. ICES Journal of Marine Science, 64: 640–646.

Maindonald, J. M., and Braun, J. 2003. Data Analysis and Graphics using R: An Example-Based Approach. Cambridge University Press, Cambridge, UK. 362 pp. (software available in the R package *UsingR*).

Maunder, M. N. 2007. A reference list of papers that use AD Model Builder and its precursor AUTODIF. http://iattc.org/PDFFiles2/ADMBreferences.pdf.

Maunder, M. N., Schnute, J. T., and Ianelli, J. N. Computers in fisheries population dynamics. *In* Computers in Fisheries Research, 2nd edn. Ed. by B. A. Megrey, and E. Moksness. Springer, New York, in press.

Microsoft.com. 2000. Gates ushers in next generation of PC computing with launch of Windows 2000, San Francisco, 17 February 2000. URL: http://www.microsoft.com/Presspass/press/2000/feb00/w2kgatespr.mspx.

Murrell, P. 2006. R Graphics. Chapman and Hall/CRC Press, Boca Raton, FL. 301 pp. (software available in the R package *RGraphics*).

Otter Research Ltd. 1994. AUTODIF: a C++ array language extension with automatic differentiation for use in nonlinear modeling and statistics. 123 pp. (available from http://otter-rsch.com/admodel.htm).

Otter Research Ltd. 2000. An introduction to AD Model Builder (version 4) for use in nonlinear modeling and statistics. 127 pp. (available from http://otter-rsch.com/admodel.htm).

Raymond, E. S. 2001. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly, Cambridge, MA. 256 pp. (see also http://www.catb.org/~esr/writings/cathedral-bazaar/ and http://en.wikipedia.org/wiki/Linus's_Law).

R Development Core Team. 2007. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org.

Sainsbury, K. J., Punt, A. E., and Smith, A. D. M. 2000. Design of operational management strategies for achieving fishery ecosystem objectives. ICES Journal of Marine Science, 57: 731–741.

Schnute, J. T., Boers, N. M., and Haigh, R. 2004. PBS Mapping 2: user's guide. Canadian Technical Report of Fisheries and Aquatic Sciences, 2549. 126 pp. (software available in the R package *PBSmapping*, which includes this report as *PBSmapping- UG.pdf*).

Schnute, J. T., Couture-Beil, A., and Haigh, R. 2006. PBS Modelling user's guide. Canadian Technical Report of Fisheries and Aquatic Sciences, 2674. 114 pp. (software available in the R package *PBSmodelling*, which includes this report as *PBSmodelling-UG.pdf*).

Schnute, J. T., and Haigh, R. 2006. Reference points and management strategies: lessons from quantum mechanics. ICES Journal of Marine Science, 63: 4–11.

Schnute, J. T., and Richards, L. J. 2001. Use and abuse of fishery models. Canadian Journal of Fisheries and Aquatic Sciences, 58: 10–17.

Schnute, J. T., Richards, L. J., and Olsen, N. 1998. Statistics, software, and fish stock assessment. *In* Fishery Stock Assessment Models, pp. 171–184. Ed. by F. Funk, T. J. Quinn, J. Heifetz, J. N. Ianelli, J. E. Powers, J. F. Schweigert, P. J. E. Sullivan, *et al.* Alaska Sea Grant College Program Report No. AK-SG-98-01. University of Alaska, Fairbanks.

Smith, A. D. M., Sainsbury, K. J., and Stevens, R. A. 1999. Implementing effective fisheries-management systems–management strategy evaluation and the Australian partnership approach. ICES Journal of Marine Science, 56: 967–979.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. 2004. WinBUGS User Manual, version 2.0. Available at http://mathstat.helsinki.fi/openbugs/.

Thomas, N. 2004. BRugs User Manual (the R interface to BUGS), version 1.0. Available at http://mathstat.helsinki.fi/openbugs/.

Venables, W. N., and Ripley, B. D. 1999. Modern Applied Statistics with S-PLUS, 3rd edn. Springer, New York. 501 pp. (software available in the R package *MASS*).

Venables, W. N., and Ripley, B. D. 2000. S Programming. Springer, New York. 264 pp. (software available at http://www.stats.ox.ac.uk/pub/MASS3/Sprog/).

Walther, A., Kowarz, A., and Griewank, A. 2005. ADOL-C: a package for the automatic differentiation of algorithms written in C/C++, version 1.10.0. Report and software available from: http://www.math.tu-dresden.de/~adol-c/.

Wood, S. 2006. Generalized Additive Models: An Introduction with R. Texts in Statistical Science, 67. Chapman and Hall/CRC Press, Boca Raton, FL. 391 pp. (software and data available in the R packages *mgcv* and *gamair*).

Appendix

Software packages

The list below provides websites and additional background for every software package mentioned herein. We encourage readers to scan this list because it illustrates the astonishing developments that have taken place over the past three decades. Software history helps give perspective to a new project, like that proposed here. Readers with a historical bent might particularly enjoy the interview with Amir Aczel and Lee Wilkinson, cited in the context of SYSTAT®. If you have Pygmalion tendencies (as do the authors of this paper), drop them long enough to contemplate some alternative sculptors and sculptures.

ACON (http://www.mar.dfo-mpo.gc.ca/science/acon/). "A CONtouring application" for data visualization, developed in the Atlantic region of Canada. It includes an embedded programming language and a number of intrinsic functions supporting fisheries assessment. Some Canadian scientists have written impressive fishery applications with ACON.

ADMB© (http://otter-rsch.com/admodel.htm). "AD Model Builder", a commercial tool for the rapid development and implementation of non-linear statistical models. It achieves remarkably fast function minimization through the use of algorithms for reverse AD, and it has been applied extensively in fishery stock assessments.

ADOL-C (http://www.math.tu-dresden.de/~adol-c/). A public domain package to facilitate the evaluation of first and higher derivatives of vector functions that are defined by computer programs written in C or C++.

APL [http://en.wikipedia.org/wiki/APL_(programming_language)]. "A Programming Language" based on a notation invented in 1957 by Kenneth E. Iverson. Many fishery scientists used it as one of the earliest tools available for array programming. It still continues to be supported and used (e.g. http://www.apl2000.com/).

AUTODIF© (http://otter-rsch.com/admodel.htm). A C++ class library for automatic differentiation and function minimization. It serves as the core of ADMB©, and programs written for ADMB© can inherit class structures from AUTODIF©.

BASIC (http://en.wikipedia.org/wiki/BASIC_programming_language). "Beginner's All-purpose Symbolic Instruction Code", originally designed in 1963 (by John George Kemeny and Thomas Eugene Kurtz at Dartmouth College) to give non-science students access to computers. Fishery scientists particularly used a dialect of BASIC that appeared as "Applesoft" on Apple II microcomputers. See also "Visual Basic".

C (http://en.wikipedia.org/wiki/C_programming_language). A widely used programming language originally developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories. R supports functions written in C to speed computation time.

C++ (http://en.wikipedia.org/wiki/C++ or http://en.wikipedia.org/wiki/C%2B%2B). An enhancement to C originally developed by Bjarne Stroustrup in 1983 at Bell Telephone Laboratories. It includes class structures, operator overloading, and other advanced features. AUTODIF© and ADOL-C use C++ classes to define the structures and methods needed for automatic differentiation. R supports C++, but only via a C wrapper function.

CASAL (http://www.niwascience.co.nz/ncfa/tools/casal). "C++ Algorithmic Stock Assessment Laboratory", an advanced software package for fish stock assessment developed at New Zealand's National Institute of Water and Atmospheric Research (NIWA). The CASAL software, documentation, example files, and S-PLUS®/R utility files are available on request. CASAL is freely distributed under a restricted license.

Coleraine (http://www.fish.washington.edu/research/coleraine/). A user-friendly, general age-structured model for fisheries stock assessment. It combines a familiar Excel™ environment with a powerful ADMB© application. The name comes from a New Zealand winery. For further inspiration, see http://www.temata.co.nz/TemataColeraine.asp.

CRAN (http://cran.r-project.org/). "Comprehensive R Archive Network", used to facilitate the distribution of R and its user-contributed libraries.

Excel™ (http://office.microsoft.com/en-us/excel/default.aspx). A popular spreadsheet program by Microsoft®. It comes as a component of the Microsoft® Office suite. Haddon (2001) uses Excel™ in a course on quantitative methods in fisheries. See also the entry for Coleraine above. Fishery scientists have particularly used Solver©, Excel's numerical function optimizer, for model fitting and parameter estimation. In our experience, Solver© has proved somewhat problematic; for example, it lacks the robustness of methods that use automatic differentiation (such as ADMB©).

FLR (http://flr-project.org/). "Fisheries Library in R", a development effort directed towards the evaluation of management strategies. Some packages can be directly downloaded from CRAN; others are available from the FLR site. See Kell *et al.* (2007).

FORTRAN (http://en.wikipedia.org/wiki/Fortran). A programming language developed by IBM in the 1950s for scientific and engineering applications. The name is derived from "The IBM Mathematical FORmula TRANslating System". R supports FORTRAN code and uses classical FORTRAN algorithms in its core. For example, `lm` depends on a FORTRAN function `dqrls`.

GAUSS™ (http://www.aptech.com/). A commercial language for mathematical and statistical programming, particularly known for fast execution.

GIMPS (http://www.mersenne.org/). "The Great Internet Mersenne Prime Search" for prime numbers of the form 2^n-1, where n is an integer. Users can download software that performs calculations to assist the search for the next Mersenne prime. On 4 September 2006, Curtis Cooper and Steven Boone's team broke their previous world record by discovering the 44th known Mersenne prime, $2^{32\ 582\ 657}-1$. This new prime, with 9 808 358 digits, falls short of the 10 million digits required for GIMPS to claim the Electronic Frontier Foundation $100 000 award.

Lotus 123® (http://en.wikipedia.org/wiki/Lotus_1-2-3). A spreadsheet program similar to its predecessor VisiCalc. Just as VisiCalc turned the Apple II into a business computer, Lotus 123® made a serious business computer out of the original IBM PC, running under DOS. See also http://www.lotus.com

MULTIFAN-CL (http://www.multifan-cl.org/). A computer program that implements a statistical, length-based, age-structured model for use in fisheries stock assessment (© 2003, Otter Research Ltd., http://otter-rsch.com/, and the Secretariat of the Pacific Community, http://www.spc.int/). This software, associated programs, and documentation are made available to the scientific community gratis, subject to a license that places minor restrictions on use and distribution. See Fournier *et al.* (1998) and Hampton and Fournier (2001).

Minitab® (http://www.minitab.com/). A computer program to perform statistical calculations, developed in 1972 by instructors at Pennsylvania State University. Some fishery scientists used this as an early programming language for statistical analyses. See also http://en.wikipedia.org/wiki/Minitab.

NFT (http://nft.nefsc.noaa.gov/). NOAA Fisheries Toolbox, currently Version 2.11, 2007. A set of assessment tools widely used by the US National Marine Fisheries Service (NMFS). The website is password-protected. See also SS2 below.

OpenBUGS (http://mathstat.helsinki.fi/openbugs/). "Open" source software to perform "Bayesian inference Using Gibbs Sampling" (Spiegelhalter *et al.*, 2004). The package uses a specialized language based on a current set of supported statistical distributions. The R library BRugs (Thomas, 2004) supports OpenBUGS, and the executable files come with the Windows binary distribution. PBSmodelling includes examples that illustrate the use of BRugs to invoke OpenBUGS.

Pascal [http://en.wikipedia.org/wiki/Pascal_(programming_language)]. A highly structured programming language, developed in 1970 by Niklaus Wirth. After the introduction of microcomputers, many fishery scientists used Turbo Pascal and other versions of the language produced by Borland (http://en.wikipedia.org/wiki/Turbo_Pascal). The source code compiled very rapidly and produced fast executable code. Turbo Pascal compilers (versions 1.0, 3.02, and 5.5) now appear in the public domain, although Free Pascal (http://www.freepascal.org/) offers a better option. OpenBUGS is written in Component Pascal, a successor to the original Pascal language.

PBSmapping (http://cran.r-project.org/). An R library that proves two-dimensional plotting features similar to those available in a Geographic Information System (GIS). Embedded C code speeds algorithms from computational geometry, such as finding polygons that contain specified point events or performing Boolean operations on polygons (Schnute *et al.*, 2004).

PBSmodelling (http://cran.r-project.org/). An R library that facilitates design, testing, and operation of computer models. It focuses particularly on tools for easy development and modification of customized graphical user interfaces (GUIs). Although the software depends heavily on the R interface to the Tcl/Tk package, a user does not need to know Tcl/Tk. The package contains examples that illustrate models built with other R packages, including PBSmapping, odesolve, and BRugs (Schnute *et al.*, 2006).

Python [http://en.wikipedia.org/wiki/Python_(programming_language)]. A high-level programming language first released by Guido van Rossum in 1991. It can be used as a free scripting language for the operating system, with support for mathematical computations and GUI design via Tcl/Tk. Python might play a useful role in linking the MSE components in Figure 1.

R/S (http://www.r-project.org/). The R framework for statistical computing described in this paper. R comes originally from S, a programming language developed primarily by John Chambers and (in earlier versions) by Rick Becker and Allan Wilks of Bell Laboratories. Chambers (1998, Preface) describes S as "a programming language for all kinds of computing involving data. It has a simple goal: **To turn ideas into software, quickly and faithfully**".

S-PLUS® (http://www.insightful.com/products/splus/). A commercial version of S, sold by Insightful®.

SAS® (http://www.sas.com/). "Statistical Analysis System", originally conceived by Anthony J. Barr in 1966. He first created an analysis of variance modeling language inspired by the notation of statistician Maurice Kendall, followed by a multiple regression program that generated machine code for performing algebraic transformations of the raw data (http://en.wikipedia.org/wiki/SAS_System).

SS2 (http://nft.nefsc.noaa.gov/). "Stock Synthesis 2" (currently version 2.00c) in the NOAA Fisheries Toolbox (version 2.11, 2007). SS2 provides a statistical framework for calibrating a population dynamics model with a diversity of fishery and survey data. The program can accommodate both age and size structure, with multiple stock subareas. The website is password-protected. For further information and access, contact Richard Methot (Richard.Methot@noaa.gov), the lead author of SS2.

SYSTAT® (http://www.systat.com/). A statistics and graphics software package, developed in the late 1970s by Lee Wilkinson. He and his wife incorporated SYSTAT® in 1983 and sold it to SPSS in 1995. In turn, SPSS sold it in 2002 to Cranes Software International, located in Bangalore, India (http://www.spss.com/research/wilkinson/SYSTAT/systat.html). Some fishery scientists have used SYSTAT® for years as their main programming language. The noted mathematician and historian Amir Aczel once interviewed Wilkinson and obtained an amazing story that could inspire developers of a new MSE framework. See http://www.mhhe.com/business/opsci/bstat/wilkinson.mhtml.

VisiCalc (http://en.wikipedia.org/wiki/VisiCalc). The first spreadsheet program available for personal computers, generally considered the application that turned the microcomputer into a serious business tool. Fishery scientists used it on the Apple II to build rudimentary population dynamics models.

Visual Basic™ (http://en.wikipedia.org/wiki/Visual_Basic). A variant of BASIC, produced by Microsoft® in part to support the development of applications with GUIs. Major fishery programs have been written with Visual Basic™, such as the popular Ecopath with Ecosim (EwE, http://www.ecopath.org/). Fishery scientists wishing to switch from Visual Basic™ to R would do well to look carefully at PBSmodelling as a library for building GUI-based applications.

WinBUGS (http://www.mrc-bsu.cam.ac.uk/bugs/). A stage of the BUGS project, prior to OpenBUGS (mentioned above). The website states that "there are now a number of versions of BUGS, which can be confusing". Although we remain somewhat confused, we recommend OpenBUGS with the BRugs library for R.

doi:10.1093/icesjms/fsm109