



ADMB Foundation Newsletter

Volume 2, Issue 1

June 2010

Inside this issue:

<i>ADMB Foundation and Project Milestones</i>	2
<i>Converting WinBUGS into ADMB</i>	3
<i>AD Model Builder and Fisheries Library in R</i>	5
<i>GLMM's on large spatial grids</i>	6
<i>Recent ADMB based publications</i>	7

Economics application of ADMB

One of the Foundations goals is to attract users outside of fisheries. Hans Skaug and his co-authors Jun Yu and David Fournier presented the paper "Automated likelihood based inference for stochastic volatility models using AD model builder" at the Computational and Financial Econometrics conference (<http://www.dcs.bbk.ac.uk/cfe09/>)

Abstract: The Laplace approximation is commonly used to fit stochastic volatility (SV) models. We argue that the use of a numerical technique known as automatic differentiation in combination with the Laplace approximation greatly simplifies the practical fitting of SV models. Automatic differentiation, not to be confused with symbolic differentiation, evaluates first and higher order derivatives of computer programs (such as the likelihood of an SV model) accurately and efficiently. The approach is implemented in the open source software package ADMB (<http://admb-project.org/>). By exploiting the state-space structure of SV models we can implement the basic SV model in ADMB using less than 40 lines of C++ code. We also consider different variations over the SV model and techniques for improving the accuracy of the Laplace approximation.

ADMB Project General Meeting

The ADMB project held a general meeting on March 29-31, 2010, at the UCSD Scripps Institution of Oceanography campus in La Jolla, CA. The primary goals of the meeting were to (1) recruit active participants to the work of the ADMB Project with a discussion of ongoing activities, (2) identify priority activities for 2010-2012, and (3) create a sustainable organization to maintain and develop AD Model Builder. The participants ranged from the ADMB Foundation members and NCEAS staff to academics and agency staff. A wide range of topics were discussed and a summary is provided below. Further details can be found in the meeting report (<http://admb-project.org/community/admb-meeting-march-29-31>). People wishing to contribute to the project should send an email to developers@admb-project.org

Ongoing activities discussed at the meeting

Documentation

Documentation of ADMB using the Doxygen system has been initiated (<http://admb-project.org/documentation/draft-api-documentation/>), but there is a lot of work needed to fully document the code. Doxygen is a style of in-code comments that allows the automatic generation of documentation. It was decided that any new code should require Doxygen comments. The LaTeX macro used for the updated ADMB manual has been completed (<http://admb-project.org/documentation>).

Training

An "ADMB Getting Started Guide", written by Kiersten

Major-Henderson, based on the ADMB manual and the Foundation's ADMB course material has been developed and is currently being reviewed by users. This will be a valuable tool for teaching new users. The admb-project.com website now has contact information for instructors available for teaching ADMB courses (<http://admb-project.org/courses/contact-information-for-admb-instructors>)

Integration with other software

There are many projects that link ADMB with R (<http://admb-project.org/community/related-software/r>). It was decided that the ADMB project would focus on developing ADMB and not try to coordinate the projects linking ADMB with R.

ADMB promotion and Outreach

The group noted that there is a need to draw people from outside of fisheries so that the

benefits of integrated modeling approaches can be realized. Currently there is the ADMB Foundation newsletter, disbursement of prizes for recognizing excellence in modeling, presentations at selected conferences, and courses. In order to attract new users, presentations at non-fish related conferences and workshops with broader non-fisheries scope are considered important.

"Generic" code replacement

"Generic" code replacement is continuing with the hire of a full-time programmer (Derek Seiple) at the University of Hawaii in February 2010.

Maintaining source code on multi-platform testing

The source code is maintained on a secure Subversion repository. Authorized users can check out working

— continued on page 2 —



Meeting participants. From left to right: Johnnoel Ancheta, Allan Hicks, Rick Methot, Ray Conser, Anders Nielsen, Hans Skaug, Arni Magnusson, Jim Benice, Mark Maunder, Steve Martell, John Sibert, and Jim Regetz. Participants Not pictured: Jon Brodziak, Christina Show, Alex Aires-da-Silva, Brian Linton, Jim Ianelli. Photo by J. Ianelli.

Continued from page 1: ADMB Project General Meeting

copies of the code and commit modifications (<http://code.google.com/p/admb-project/>). An automated process, Buildbot, detects changes to the repository and supervises compilation and testing of the libraries for a variety of compilers, operating system and hardware targets.

The Buildbot enables continuous integration and testing, but more tests are desirable to help ensure that all parts of the ADMB code are checked. Developers who introduce new capabilities to ADMB should also supply example template files for testing. The repository and Buildbot are currently maintained by John- Noel Ancheta at the University of Hawaii.

Website

It was generally recognized that despite the improvements to the websites appearance, the structure of the website requires work. There is also a need for adding new content and updating old content.

Priorities for technical development.

The four highest priorities identified were:

1. Parallelization. Both automatic and user defined.
2. Hybrid MCMC. Fully implement and document.
3. Matrix library improvement. Linking external (large) matrix library to improve random effects models.
4. MCMC. Conversion to constant objects to increase speed.

Additionally, lower priority activities were identified including, but not limited to: a WinBUGS to ADMB parser, adding capabilities for creating design matrices, holding a developers workshop to train more people on how to write and contribute code to ADMB.

Assignment of major responsibilities

Training coordinator - Jim Bence

Duties: Develop calendar, coordinate requests with instructors, maintain training section of website

Fundraising coordinators - Mark Maunder, Steve Martell, Rick Methot

Duties: Solicit funding from appropriate donors

Outreach coordinator - Mark Maunder

Duties: Broaden applications to other fields, scholarship/prize coordination, courses and conference presentations, track applications of ADMB, ADMB Foundation Newsletter, update the community web page

Development coordinator - John Sibert

Duties: implement priorities, oversee code replacement project

Website

Due to the large nature of the website, the responsibility of maintaining each section of the website was assigned to different people. Allan Hicks and Jim Regetz were assigned the task of redesigning the structure of the whole site.

Google summer of Code and ADMB

John Nash, retired Full Professor in the Telfer School of Management of the University of Ottawa, has developed a Google Summer of Code (<http://code.google.com/soc/>) project to integrate ADMB and R. (see <http://rwiki.sciviews.org/doku.php?id=developers:projects:gsoc2010:admr>).

Google Summer of Code is a global program that offers student developers stipends to write code for various open source software projects. Since its inception in 2005, the program has brought together nearly 2500 successful student participants and 2500 mentors from 98 countries worldwide, all for the love of code. Through Google Summer of Code, accepted student applicants are paired with a mentor or mentors from the participating projects, thus gaining exposure to real-world software development scenarios and the opportunity for employment in areas related to their academic pursuits. In turn, the participating projects are able to more easily identify and bring in new developers. Best of all, more source code is created and released for the use and benefit of all.

The availability of AD tools in R would greatly assist users requiring derivatives in solving optimization, differential equation, and similar problems. Linking R to Automatic Differentiation Tools such as ADMB, ADOL-C or cppAD provides one possibility for such capabilities. A full or partial implementation in R is another.

Several ad hoc methods have been developed for linking ADMB to R but it is difficult to transparently exploit ADMB capabilities from within R. In particular, interfaces to R optimization packages need to be better established and documented. However, ADMB is already fairly well understood by some R users and may be well-suited for a Google-project to interface to R. There is currently considerable support from ADMB users to improve the interaction between ADMB and R.

ADMB Foundation and Project Milestones

The ADMB Foundation has been in existence for nearly three years and in combination with the ADMB-project, substantial progress has been made towards the Foundation's goals. Below is a list of milestones that were presented at the recent ADMB-project General Meeting.

ADMB Foundation Milestones

- August 2007 – Incorporated in State of Hawaii
- December 2008 – 501(c)(3) Status from IRS
- May 2008 – Receives start-up funding from several fisheries organizations (IATTC, IPHC, AZTI)
- June 2009 – David Fournier receives AFS Ricker award
- May, June 2009 – Sponsors student awards

at Tuna Conference and AFS

- August 2009 – Sponsors introductory training workshops in Spain and Denmark

ADMB-project Milestones

- December 2007 – Grant from Moore Foundation to NCEAS
- October 2008 – University of California purchase rights to ADMB from Otter Research
- December 2008 – ADMB binaries on line for public download on code.google
- April 2009 – Over 2000 downloads of binaries
- May 2009 – Preliminary source code release
- December 2009 – Complete source code available
- March 2009 – ADMB supported on all major computing platforms



Jordan Watson from the University of Washington was awarded the 2010 ADMB Foundation Tuna Conference award

ADMB Foundation Tuna Conference award

The 2010 ADMB Foundation award for the 61st Tuna Conference was presented to Jordan Watson from the University of Washington for his paper "Trade-offs in the design of fishery closures: Silky shark bycatch management in the eastern Pacific Ocean tuna purse seine fishery." The full paper is published in Conservation Biology 23(2):626.

Converting WinBUGS into ADMB

By Mark N. Maunder

WinBUGS (and other versions of BUGS) is one of the most popular environments for statistical inference using hierarchical models. As with all statistical tools, WinBUGS has some deficiencies. In particular, it has trouble with some large problems and inference is based on the Bayesian framework. ADMB may be beneficial in these situations since inference can be implemented in both Bayesian and Frequentist frameworks and ADMB can deal with some large models that WinBUGS can't. Here I describe how to translate the first WinBUGS example "Rats: Normal hierarchical model" (from section 6 of Gelfand et al (1990)) into an AD Model Builder (ADMB) model.

First, there are several things to remember when translating a WinBUGS program to ADMB

(1) ADMB is a procedural language. Variables have to be assigned in order so that any assignment is based on variables that have already been assigned their values. In general, this means that the ADMB code will be in the reverse order of the WinBUGS code.

(2) ADMB is based on a minimizer so the objective function uses negative log likelihoods and the negative log of probability distributions (e.g. the negative log of the prior). Don't worry, ADMB does the correct transformations for Bayesian MCMC analyses.

(3) Section headings have to be on the left margin and statements have to be indented.

(4) To create an ADMB model you generally need to have a *.tpl file that includes the code representing the model a *.dat file that includes the data to use in the model and a *.pin file that contains the initial values for the parameters. These files relate to the WinBUGS code (page 4), data list (page 5), and initial parameter value list (page 5) in a WinBUGS model.

Create the DATA_SECTION based on the WinBUGS data list (page 5). First define the dimensioning variables N and T as integers that are read in from the data file. This is done by using the prefix "init_" in front of the integer definition "int". The dimensioning variables can then be used to dimension the variables that are used to hold the data. Read in xbar using the number variable definition since it is only a single value. Read in y as a vector and dimensioned from 1 to N. Read in y as a matrix with the first dimension 1 to N and the second dimension 1 to T.

DATA_SECTION

```
init_int N
init_int T
init_number xbar
init_vector x(1,T)
init_matrix y(1,N,1,T)
```

PARAMETER_SECTION

```
init_bounded_number tauc(0.00001,100,3)
init_number alphac
init_bounded_number alphatau(0.00001,100,2)
init_number betac
init_bounded_number betatau(0.00001,100,2)
random_effects_vector alpha(1,N)
random_effects_vector beta(1,N)
matrix mu(1,N,1,T)
objective_function_value f
```

Create the first part of the PARAMETER_SECTION based on the WinBUGS initial parameter list (page 5). Remember that the random effect parameters have to be defined last. ADMB cannot use "." in a variable name so delete these. Use the "init_" prefix to indicate that the variable is a parameter to estimate. Create the variables for tauc, alphac, alphatau, betac, and betatau. These are all single values so use the number variable with the "init_" prefix. It is a good idea to estimate the precision parameters in later phases to improve convergence. It is also a good idea to put bounds on parameters to keep them away from unrealistic parameter space. To include bounds you have to add the additional prefix "bounded_". In particular, you can use bounds to keep the precision parameters from going negative. Add the bounds and phases to these variables by including them in brackets after the variable definition. The format is ParameterName(LowerBound,UpperBound,Phase). Remember that if you run ADMB in the Bayesian MCMC mode this is putting a uniform prior between the bounds on the model parameter. Create the random effect variables for alpha and beta. These are vectors of dimension 1 to N and use the prefix "random_effects_" to define them as random effect parameters. Other variables that are used for intermediate calculations in the winbugs code need to be defined in the PARAMETER_SECTION. The matrix mu is used in the WinBUGS code. It is used inside loops that go from 1 to N and 1 to T. So use those dimensions to define the variable mu. mu is used for intermediate calculations so it does not need a prefix. Finally you need to define the objective_function_value that is the variable that ADMB minimizes. f is commonly used for this variable.

The PROCEDURE_SECTION is where the calculations take place. These calculations define the objective function value as a function of the model parameters defined in the PARAMETER_SECTION and the data read in the DATA_SECTION. Other variables defined in the PARAMETER_SECTION may be used to facilitate the calculations. All statements must end with a semicolon ";". ADMB is a procedural language so that assignment statements must be based on variables that have already had their values assigned. In general, this means following the WinBUGS code in reverse order. The WinBUGS code can be divided into priors, random effects, and likelihood components. Since we are considering a non-Bayesian random effects model we will ignore the priors, although they could be implemented (page 6) and ADMB's MCMC algorithm used to implement a Bayesian analysis. The assignment statements for sigma and alpha0 are not necessary as they are not used in the calculations, but could be implemented if their values are of interest. Note that ADMB uses "=" as the assignment statement rather than "<=". First, implement the two loops. ADMB uses C++ so the loop syntax follows C. The basic loop statement is "for(initialization;conditional expression;increment) statement;". The curly brackets work the same as in WinBUGS to group a block of statements. The conditional expression determines whether or not

— Continued on page 4 —

Continued from page 3: Converting WinBUGS into ADMB

the loop will repeat. The "int" statement just creates an integer variable (in this case i or j, respectively). The "+" signs after the i or j just means increment the variable by 1. Because of the procedural nature, mu has to be assigned a value before it can be used in the objective function. Array indexing is the same in ADMB as it is in WinBUGS except round brackets are used instead of square brackets. The WinBUGS code "Y[i , j] ~ dnorm(mu[i , j],tau.c)" defines the likelihood function. In ADMB you need to add the negative log-likelihood to the objective function. The negative log-likelihood is minimized (or if used in Bayesian MCMC mode it is converted into a posterior). The "+" sign after the "=" sign is just short hand for adding to itself (i.e. "f=f+..."). The negative log-likelihood is written here without constants, which could be added if desired. Since the precision is used the form of the negative log-likelihood ignoring constants is $-0.5 \log(\tau) + 0.5 \tau \text{square}(y - \text{mean})$. Finally, add the contribution of the random effects distribution to the objective function value. Remember that these should be based on the negative log of the probability distribution to be consistent with the negative log-likelihood used in ADMB. The function "norm2" does the sum of squares of a vector (or other array) and can be used to implement the negative log likelihood. Since the precision is used the form of the negative log-probability ignoring constants is $-0.5 \cdot n \cdot \log(\tau) + 0.5 \cdot \tau \cdot \text{norm2}(y - \text{mean})$ where n is the number of realizations of the random effect and y is a vector containing the random effect.

— Continued on page 5 —

PROCEDURE_SECTION

```

for(int i=1;i<=N;i++)
{
    for(int j=1;j<=T;j++)
    {
        mu(i,j)=alpha(i) + beta(i) * (x(j) - xbar);
        f+=-0.5*log(tauc)+0.5*tauc*square((y(i,j)-mu(i,j)));
    }
}

f+=-N*0.5*log(alphatau)+0.5*alphatau*norm2(alpha-alphac);
f+=-N*0.5*log(betatau)+0.5*betatau*norm2(beta-betac);

```

WinBUGS Model

```

model
{
    for( i in 1 : N ) {
        for( j in 1 : T ) {
            Y[i , j] ~ dnorm(mu[i , j],tau.c)
            mu[i , j] <- alpha[i] + beta[i] * (x[j] - xbar)
        }
        alpha[i] ~ dnorm(alpha.c,alpha.tau)
        beta[i] ~ dnorm(beta.c,beta.tau)
    }
    tau.c ~ dgamma(0.001,0.001)
    sigma <- 1 / sqrt(tau.c)
    alpha.c ~ dnorm(0.0,1.0E-6)
    alpha.tau ~ dgamma(0.001,0.001)
    beta.c ~ dnorm(0.0,1.0E-6)
    beta.tau ~ dgamma(0.001,0.001)
    alpha0 <- alpha.c - xbar * beta.c
}

```

Continued from page 4: Converting WinBUGS into ADMB

The ADMB program can be created by using the command line command “admb -r name” where “-r” indicates that it includes random effects and name is the root of the *.tpl file name. Data and initial value files also need to be created before the model is run (see below). Once the model is compiled, the command line option “-mno 4000” needs to be used to allocate the appropriate amount of memory for this model. Run the model with the command line command “name -mno 4000”. The parameter estimates will be in the *.par file, the standard deviations in the *.std file, and the correlation matrix in the *.cor file.

Creating the data “*.dat” and “*.pin” files. The data and initial values have to be put in the order that they are defined in the DATA_SECTION and PARAMETER_SECTION sections. The best way to create the files is to copy the DATA_SECTION from the *.tpl file and paste it in a new file and copy the init_ and random_effects_ variable definitions in the PARAMETER_SECTION from the *.tpl file and paste it in a new file. Then comment out the variable definition statements using the “#” symbol. This makes it easy to see the placement and dimensions of the data and initial values.

— continued on page 6 —

```
winbugs data
Data list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), xbar = 22, N = 30, T = 5,
         Y = structure(
           .Data = c(151, 199, 246, 283, 320,
             ...
           .Dim = c(30,5)))

//Winbugs init file
list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250,
              250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250,
              250, 250, 250, 250),
      beta = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
      alpha.c = 150, beta.c = 10,
      tau.c = 1, alpha.tau = 1, beta.tau = 1)
```

AD Model Builder and Fisheries Library in R

Fisheries Library in R (FLR) is a collection of tools to support models of fisheries and ecological systems in the statistical environment R (<http://www.rproject.org>). The core group of the FLR project met in Lisbon, 9-10 December 2009 to discuss ‘The future of FLR’. The AD Model Builder foundation was invited to share knowledge between these two open source projects regarding technical and organizational issues, and to explore ways collaborate in the future. Anders Nielsen participated and combined it with a public introduction to AD Model Builder at IPIMAR. The discussions focused mainly on how to call an AD Model Builder program from an R package, which would allow the computational demanding parts of for instance a stock assessment model to be run in AD Model Builder and the data handling and graphics to be controlled from R. Both the low-tech option (via ASCII files) and the more advanced option of using shared object (DLL on windows) was presented in a small example. Many AD Model Builder users work with fisheries related data analysis, so FLR and AD Model Builder potentially share many users, but currently no FLR

tool uses AD Model Builder. In a few places FLR uses ADOL-C (<http://www.coin-or.org/projects/ADOL-C.xml>), which is another open source automatic differentiation package. The main reason for not using AD Model Builder was that at the time of development AD Model Builder was not open source. The source for AD Model Builder is now available, so that will not be an obstacle in the future.



Anders Nielsen presenting ADMB to members of the FLR project in Lisbon.

New ADMB Foundation members

The ADMB welcomes four new members. These are the first members who have joined the Foundation since its inception in 2007. They were chosen as members due to their substantial contributions to ADMB.

Johnoel Ancheta of the University of Hawaii maintains the ADMB source code

Jim Ianelli of NMFS Alaska Science Center was intimately involved with ADMB from its inception and was one of the uses that Dave Fournier relied upon for feedback while creating ADMB

Arni Magnusson of the Icelandic Marine Research Institute created the ADMB IDE

Hans Skaug of the University of Bergen is the co-developer of ADMB-re

GLMM's on large spatial grids

by Kasper Kristensen, Hans J. Skaug and David Fournier

It has for a long time been possible to fit GLMMs (Generalized Linear Mixed Models) in ADMB-RE. A typical example is correlated count data with Poisson distribution. However, when the observation are located on a spatial grid the number of latent variables (random effects in the ADMB-RE terminology) grows quadratically in the number of grid points in each geographical direction. The large number of random effects causes a computational challenge.

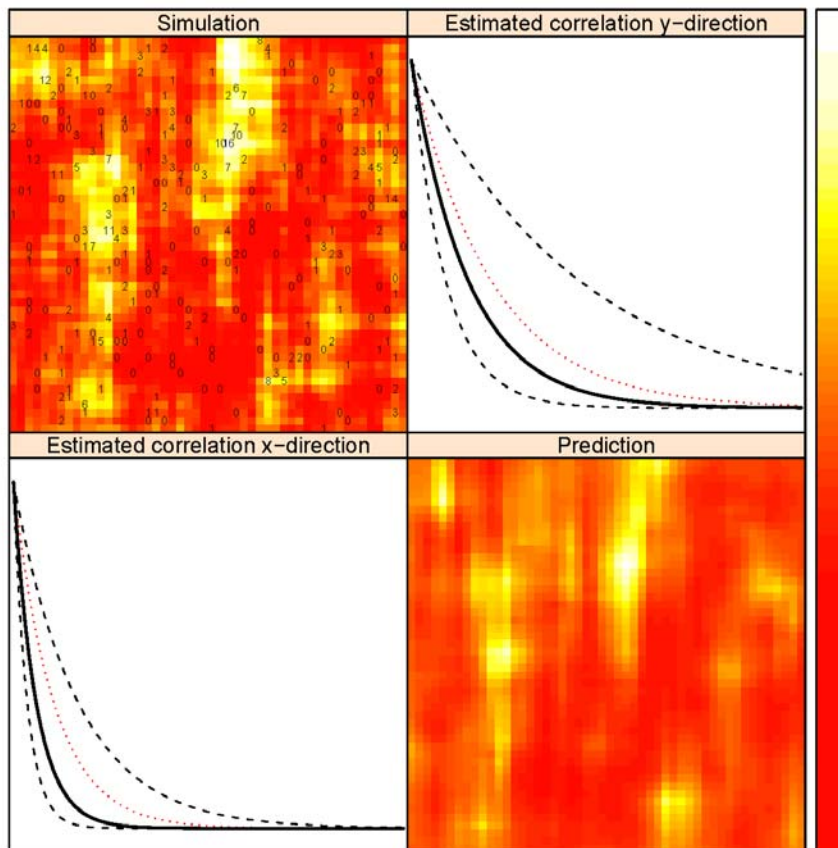
It has recently been shown (see link below) that a particular class of spatial correlation functions gives rise to a particularly efficient implementation in ADMB. The correlation between two points in the latent Gaussian random field, separated by a longitudinal distance Δx and a latitudinal distance Δy , has the separable form, $cor(\Delta x, \Delta y) = cor_1(\Delta x)cor_2(\Delta y)$ where cor_1 and cor_2 are longitudinal and latitudinal correlation functions, respectively. Now, if an autoregressive model is used to model the dependency in each direction, i.e. $cor_1(\Delta x) = \exp(-\alpha_1 \Delta x)$ and $cor_2(\Delta y) = \exp(-\alpha_2 \Delta y)$, it can be shown that the inverse correlation matrix (precision matrix) over all points in the grid is a sparse matrix. Hence the sparse matrix feature of ADMB-RE can be exploited to speed up calculations. An example with 10,000 grid points can easily be fit on a computer with 8Gb of memory. Current research on the ADMB source code has shown that relatively small modifications will make it feasible to fit a 200x200 grid on a 1.5 Gb machine. These improvements will find their way into the official ADMB source soon.

Example: Poisson counts with spatial dependence

Consider a situation where we want to model the spatial variation in animal density. The coloring in the upper left corner of the Figure below shows the $\log(\text{density})$ of animals. However, the density surface is unknown to us. We only observe the animal density at certain locations. The number of animals present in a small region (not necessarily the same as the grid points below) around the locations is counted. The locations, and the number of animals present, are also indicated in the plot (the numbers). Our task is to estimate the underlying density field from these observations.

We start by constructing a fine grid (50 times 50) such that each grid cell contains at most one observation. Note that most grid cells are empty (no observation is made in that cell). The animal density is assumed to be generated by a Gaussian spatial field with the correlation structure described earlier. Conditionally on the value of this field (the coloring in the figure) the observation follows a Poisson distribution with expected value proportional to the density field. The Figure shows both estimated correlation functions and the estimated (predicted) $\log(\text{density})$ field. More details and ADMB files can be found at:

<http://admb-project.org/examples/spatial-models/glmm2019s-on-large-spatial-grids>



Results from running the model on grid with simulated data. Image of simulated $\log(\text{density})$ in each lattice point with observed Poisson counts corresponding to missing data (top-left). Estimated exponential correlation functions with confidence intervals (black lines) and true correlation functions (red dotted line) (top-right and bottom-left). Prediction of un-observed $\log(\text{density})$ (bottom-right).

Continued from page 5: Converting WinBUGS into ADMB

The priors (in this case) can be added anywhere in the PROCEDURE_SECTION because they are not dependent on any assignment statements. Therefore, it is convenient to add them to the end of the PROCEDURE_SECTION. ADMB has a built in log gamma probability function and this can be used to implement the gamma priors. The normal priors are implemented ignoring the constants, but they can be added if desired. Add the priors to the objective function at the end of the PROCEDURE_SECTION. (note that when compiling this model you will get a warning about the `log_gamm_density` because the function has not been defined for all possible arguments. The arguments below will be coerced into usable variables and will work. This will be corrected in later versions of ADMB)

```
f+=-log_gamma_density(tauc,0.001,0.001);
f+=0.5*1.0E-6*square(alpha-0.0);
f+=-log_gamma_density(alpha,0.001,0.001);
f+=0.5*1.0E-6*square(beta-0.0);
f+=-log_gamma_density(beta,0.001,0.001);
```

Compiler benchmarks

Arni Magnusson has updated the compiler benchmarks (<http://admb-project.org/community/benchmarks>). In the 'catage' benchmark, linux64 runs around 64% faster than win32. Previous benchmarks have shown that linux32 is only marginally slower than linux64, implying that the 64% difference is mainly a Linux vs. Windows effect, rather than 32-bit vs. 64 bit. In Windows, GCC and Visual C++ show similar performance, around 59% faster than the current Borland compiler.



ADMB Foundation Newsletter

Newsletter Editor:
Mark Maunder

Inter-American Tropical Tuna Commission

Send comments and contributions to
newsletter@admb-foundation.org

Signup for the newsletter at
ADMB Foundation website or send an e-
mail to newsletter@admb-foundation.org

The objectives of the ADMB Foundation are to (a) advance the ADMB project for improving data analysis and non-linear statistical modeling; (b) act as an official voice for the ADMB project, providing means of communication with the press, commercial and noncommercial organizations interested in the ADMB software; (c) coordinate development and promote use of ADMB.

The ADMB Foundation is a non-profit organization working in the public interest for charitable, educational and scientific purposes under section 501(c)(3) of the United States Internal Revenue Code.

President: John Sibert

Treasurer: Mark Maunder

Secretary: Anders Nielsen

**Were on the
web! admb-
foundation.org**

Recent ADMB based publications

- A'mar, Z.T., Punt, A.E., and Dorn, M.W. (2009) The impact of regime shifts on the performance of management strategies for the Gulf of Alaska walleye pollock (*Theragra chalcogramma*) fishery. *Can. J. Fish. Aquat. Sci.* 66(12): 2222–2242.
- Branch, T. A. and Hilborn, R. (2010) A general model for reconstructing salmon runs. *Canadian Journal of Fisheries and Aquatic Sciences*, 67(5): 886–904.
- Broms, K., Skalski, J.R., Millsbaugh, J.J., Hagen, C.A., and Schulz, J.H. (2010) Using Statistical Population Reconstruction to Estimate Demographic Trends in Small Game Populations. *Journal of Wildlife Management* 74(2):310-317.
- Conn, P. B., Williams, E. H., and Shertzer, K. W. (2010) When can we reliably estimate the productivity of fish stocks? *Canadian Journal of Fisheries and Aquatic Sciences*, 67(3): 511–523.
- Edeline, E., Le Rouzic, A., Winfield, I.J., Fletcher, J.M., James, J.B., Stenseth, N.C., and Vøllestad, L.A. (2009) Harvest-induced disruptive selection increases variance in fitness-related traits. *Proc. R. Soc. B* 276: 4163–4171.
- Flores, L., Ernst, B., and Parma, A.M. (2010) Growth pattern of the sea urchin, *Loxechinus albus* (Molina, 1782) in southern Chile: evaluation of growth models. *Mar Biol* 157: 967–977.
- Frisk, M. G., Martell, S. J.D., Miller, T. J., and Sosebee, K. (2010) Exploring the population dynamics of winter skate (*Leucoraja ocellata*) in the Georges Bank region using a statistical catch-at-age model incorporating length, migration, and recruitment process errors. *Canadian Journal of Fisheries and Aquatic Sciences*, 67(5): 774–792.
- Haist, V., Breen, P.A., and Starr, P.J. (2009) A multi-stock, length-based assessment model for New Zealand rock lobster (*Jasus Edwardsii*). *New Zealand Journal of Marine and Freshwater Research*, 43(1): 355 – 371.
- Holtgrieve, G.W., Schindler, D.E., Branch, T.A., and A'mara, Z.T. (2010) Simultaneous quantification of aquatic ecosystem metabolism and reaeration using a Bayesian statistical model of oxygen dynamics. *Limnol. Oceanogr.*, 55(3): 1047–1063.
- Jensen, O.P., Gilroy, D.J., Hogan, Z., Allen, B.C., Hrabik, T.R., Weidel, B.C., Chandra, S., and Vander Zanden, M.J. (2009) Evaluating recreational fisheries for an endangered species: a case study of taimen, *Hucho taimen*, in Mongolia. *Can. J. Fish. Aquat. Sci.* 66: 1707–1718.
- Kocovsky, P.M. and Stapanian, M.A. (2010) Night sampling improves indices used for management of yellow perch in Lake Erie. *Fisheries Management and Ecology* 17: 10–18.
- Krag, L. A., Holst, R., and Madsen, N. (2009) The vertical separation of fish in the aft end of a demersal trawl. *ICES Journal of Marine Science*, 66: 772–777.
- Le Rouzic, A., Skaug, H.J., and Hansen, T.F. (2010) Estimating genetic architectures from artificial-selection responses: A random-effect framework. *Theoretical Population Biology* 77(2): 119–130.
- Liow, L. H., Skaug, H.J., Ergon, T. and Schweder, T. (2010) Global occurrence trajectories of microfossils: environmental volatility and the rise and fall of individual species. *Paleobiology* 36(2): 224–252.
- Marty, G.D., Hulson, P.-J. F., Miller, S.E., Quinn II, T.J., Moffitt, S.D., and Merizon, R.A. (2010) Failure of population recovery in relation to disease in Pacific herring. *Diseases of Aquatic Organisms* 90: 1–14.
- Maunder, M.N. and Deriso, R.B. (2009) Dealing with missing covariate data in fishery stock assessment models. *Fisheries Research*, 101: 80–86.
- Schweder, T., Sadykova, D., Rugh, D., and Koski, W. (2010) Population estimates from aerial photographic surveys of naturally and variably marked bowhead whales. *Journal of Agricultural, Biological, and Environmental Statistics*, 15(1): 1–19.
- Sveda'ng, H., Bartolino, V., Gedamke, T., Hjelm, J. Bo'rjesson, P., and Nore'n, F. (2010) Fishing through time: population dynamics of plaice (*Pleuronectes platessa*) in the Kattegat–Skagerrak over a century. *Population Ecology* 52:251–262.