ADMB

Automatic Differentiation Model Builder

ADMB Foundation newsletter

Volume 1, Issue 3

July 2009

ADMB-IDE: Easy and efficient user interface

For over a decade, ADMB has been known for excellent performance, but somewhat tricky installation and usage. Users have been required to install and configure a C++ compiler, set environment variables, and in many cases make minor adjustments to user compilation scripts. Even after ADMB is properly set up, the workflow loop (write, compile, test) has involved switching repeatedly between the text editor and a shell. Some users have enjoyed syntax highlighting of some kind, while others have edited their code in black and white, making it harder to navigate and debug large models. Many ADMB users have found the ADMB user interface hard and inefficient.

In January 2009, Arni Magnusson released admb-mode for Emacs that he had been using for some years. It provides syntax highlighting, IDE compilation and file manipulation, outline code navigation, templates, and smaller tools. This contribution was only helpful for ADMB users who were already familiar with Emacs, or willing to invest time to learn and configure this complex and powerful editor. One could describe Emacs in admb-mode as a hard but efficient user interface.

May 2009 saw the release of ADMB-IDE (integrated development environment), an easy and efficient ADMB interface. It uses Emacs admb-mode as the engine, but instead of the default Emacs keybindings, it uses familiar ones like C-x to cut, C-c to copy, C-v to paste, f8 to build, f9 to run, C-q to quit, etc. Hence, the ADMB-IDE tagline "Emacs admb-mode without the Emacs".

The intended user group is large, as ADMB-IDE can be used efficiently by ADMB beginners and experts, and it is available for Windows, Linux, and Mac OS X. ADMB-IDE is not intended for seasoned Emacs users (who are better served by plain admb-mode), and some ADMB users may feel that their current working environment is better for their needs. Longtime users of the Vi editor should try ADMB-IDE in viper-mode, which emulates Vi in Emacs.

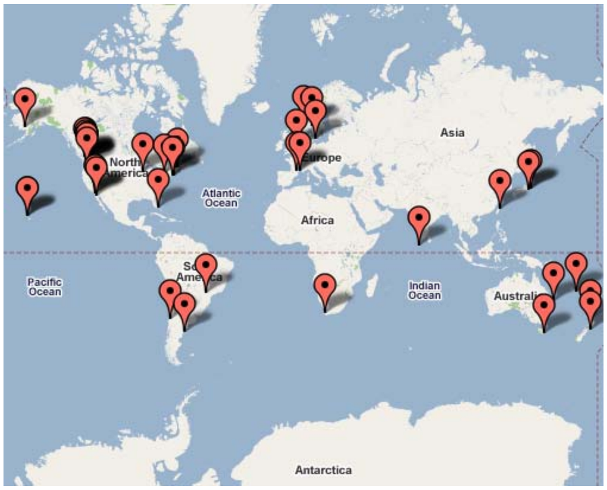
Windows users can choose between two bundled ADMB-IDE formats: (1) Installer that sets up ADMB, GCC, GDB, Emacs, admb-mode, and takes care of the necessary configurations; (2) Zip archive that contains exactly the same components. One is a true click'n'go solution, the other a ready-to-assemble kit for manual installation. These two bundles are available from the ADMB Google Code website (see URL below).

Linux and Mac OS X users should first install ADMB, Emacs, admb-mode, GCC, and optionally GDB, and then copy the ADMB-IDE configuration file (.emacs) to the user home directory. The configuration file is available from the ADMB-IDE homepage (see URL below).

continued on page 2

ADMB: a new pandemic!

ADMB is spreading faster than swine flue. Well, not quite, but swine flue can't estimate parameters of highly parameterized nonlinear models. As can be seen on the Google Maps picture, ADMB is used across the globe, from New Zealand and Chile in the south to Alaska and Norway in the north. An updated interactive map can be found on the admb-project.org website. If you use ADMB and your institution is not on the map, let us know so we can update it.



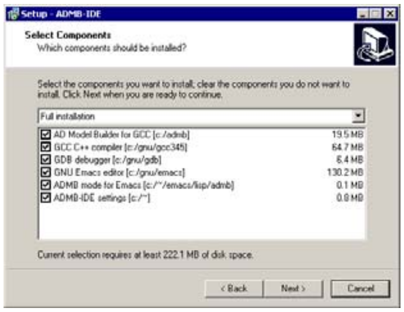
Inside this issue:

ADMB Course

The Quantitative Fisheries Center at Michigan State University will be teaching two short courses on the use of AD Model Builder in East Lansing, Michigan in August. The first course "AD Model Builder Basics" will be held August 18 and 19, 2009 and is intended for those who have never used the software or those needing a refresher in software basics. The second course "AD Model Builder Advanced Fishery Applications" will be held August 20 and 21, 2009 and will cover more advanced applications will cover more advanced topics (e.g., assessing uncertainty in model results; instruction on writing functions; alternative model parameterization. Instructors for the short courses are Drs. Jim Bence, Brian Irwin, and Travis Brenden. For additional information, including enrollment cost, please contact Travis Brenden at brenden@msu.edu

Continued from page 1: ADMB-IDE: Easy and efficient user interface

Positive and critical user feedback from various corners of the world has improved ADMB-IDE. Johnoel Ancheta (Manoa, Hawaii) demonstrated the Inno Windows installer, Hans Skaug (Bergen, Norway) made helpful comments on an early version of the Windows bundle, Shige Song (Los Angeles) submitted Linux bug reports, Kray Van Kirk (Juneau, Alaska) suggested improved shell interaction, and Jim Ianelli (Seattle, Washington) reported that viper-mode can be useful for longtime Vi users.



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

☑ AD Model Builder for GCC [c:/admb]	19.5 MB
☑ GCC C++ compiler [c:/gnu/gcc345]	64.7 MB
☑ GDB debugger [c:/gnu/gdb]	6.4 MB
☑ GNU Emacs editor [c:/gnu/emacs]	130.2 MB
☑ ADMB mode for Emacs [c:/~/emacs/lisp/admb]	0.1 MB
☑ ADMB-IDE settings [c:/~]	0.0 MB

Current selection requires at least 222.1 MB of disk space.

ADMB-IDE installation screen

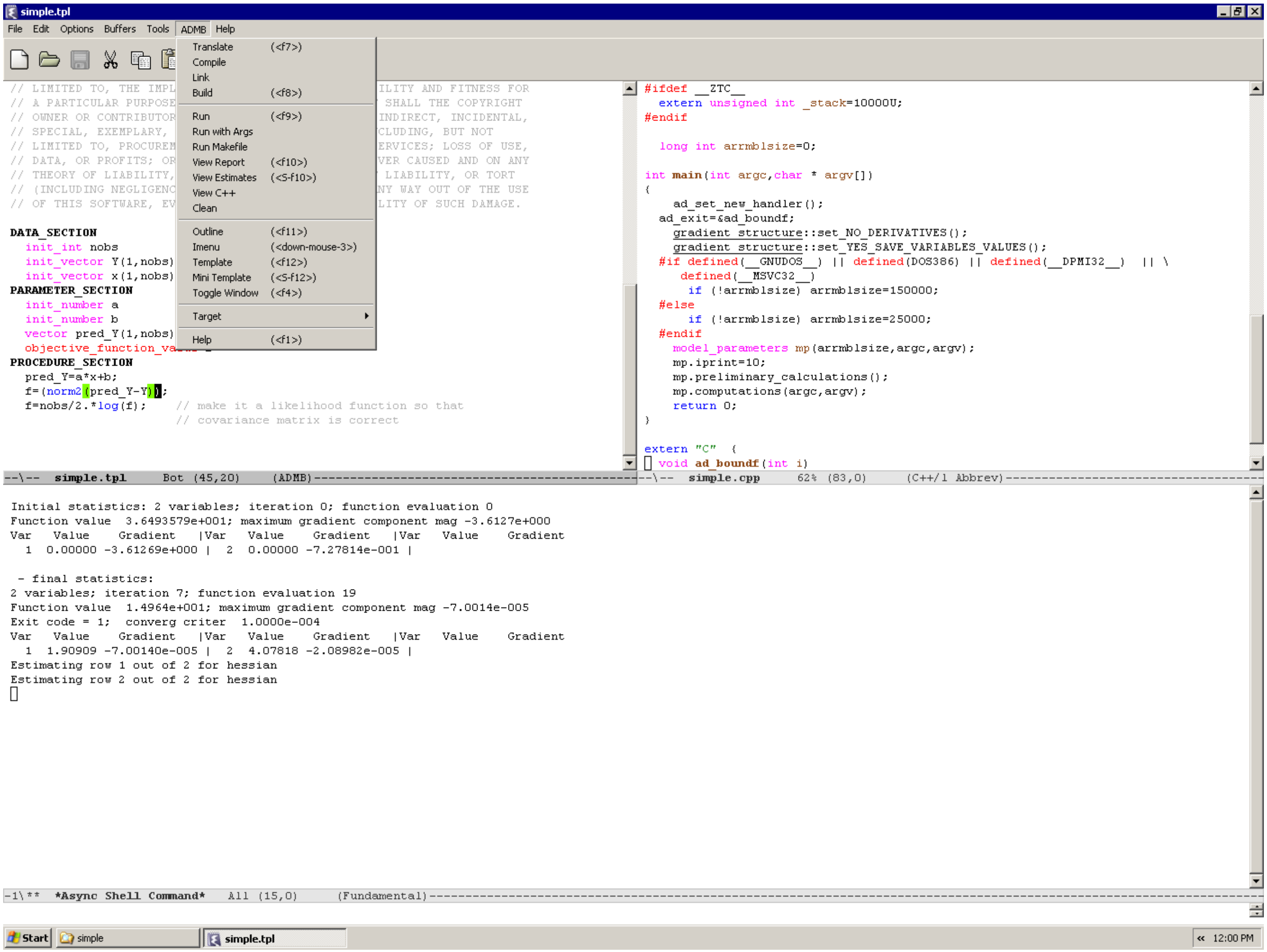
ADMB-IDE is an efficient working environment for existing ADMB users, and also a friendly interface (colorful with buttons) that will attract many new users.

URLs

ADMB Google Code (http://code.google.com/p/admb-project/)

provides fast download mirrors for the large ADMB-IDE Windows bundles (~60 MB). ADMB-IDE homepage

(http://admb-project.org/community/editing-tools/admb-ide) provides individual components of ADMB-IDE, such as the configuration file and user manual.



```
// LIMITED TO, THE IMPL                          ILITY AND FITNESS FOR
// A PARTICULAR PURPOSE                           SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTOR                          INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY,                           CLUDING, BUT NOT
// LIMITED TO, PROCUREM                          ERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR                          VER CAUSED AND ON ANY
// THEORY OF LIABILITY,                           LIABILITY, OR TORT
// (INCLUDING NEGLIGENC                          NY WAY OUT OF THE USE
// OF THIS SOFTWARE, EV                          LITY OF SUCH DAMAGE.

DATA_SECTION
  init_int nobs
  init_vector Y(1,nobs)
  init_vector x(1,nobs)
PARAMETER_SECTION
  init_number a
  init_number b
  vector pred_Y(1,nobs)
  objective_function_va
PROCEDURE_SECTION
  pred_Y=a*x+b;
  f=(norm2(pred_Y-Y));
  f=nobs/2.*log(f);    // make it a likelihood function so that
                       // covariance matrix is correct
```

```
--\--  simple.tpl     Bot (45,20)     (ADMB)----------------------------------------------
```

```
#ifdef __ZTC__
  extern unsigned int _stack=10000U;
#endif

  long int arrmblsize=0;

int main(int argc,char * argv[])
{
    ad_set_new_handler();
  ad_exit=&ad_boundf;
    gradient_structure::set_NO_DERIVATIVES();
    gradient_structure::set_YES_SAVE_VARIABLES_VALUES();
  #if defined(__GNUDOS__) || defined(DOS386) || defined(__DPMI32__)  || \
     defined(__MSVC32__)
      if (!arrmblsize) arrmblsize=150000;
  #else
      if (!arrmblsize) arrmblsize=25000;
  #endif
    model_parameters mp(arrmblsize,argc,argv);
    mp.iprint=10;
    mp.preliminary_calculations();
    mp.computations(argc,argv);
    return 0;
}

extern "C"  {
  void ad_boundf(int i)
```

```
--\--  simple.cpp      62% (83,0)      (C++/l Abbrev)-----------------------------------
```

```
Initial statistics: 2 variables; iteration 0; function evaluation 0
Function value  3.6493579e+001; maximum gradient component mag -3.6127e+000
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1  0.00000 -3.61269e+000 |  2  0.00000 -7.27814e-001 |

 - final statistics:
2 variables; iteration 7; function evaluation 19
Function value  1.4964e+001; maximum gradient component mag -7.0014e-005
Exit code = 1;  converg criter  1.0000e-004
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1  1.90909 -7.00140e-005 |  2  4.07818 -2.08982e-005 |
Estimating row 1 out of 2 for hessian
Estimating row 2 out of 2 for hessian
```

```
-1\**  *Async Shell Command*   All (15,0)     (Fundamental)---------------------------------
```

ADMB-IDE session, showing the ADMB menu entries, TPL code (upper right), C++ code (upper left), and optimization report (lower panel)

ADMB course in Spain

The ADMB Foundation offered an introductory course on AD Model Builder 6-7 July 2009 in Donostia-San Sebastián, Spain . Fourteen students from fisheries research agencies in Spain and France attended the two-day course. Anders Nielsen, Danish Technical University, and John Sibert, University of Hawaii, were the instructors. Installation, automatic differentiation, specification of model parameters, data input, reporting results, likelihood-based inference, random effects models and uncertainty were among the topics discussed in the course. Participants provided feedback on the course which will be used to improve **the next course scheduled for 17-18 August in Copenhagen**. The Foundation thanks to AZTI-Tecnalia for hosting the course at their Pasaia headquarters and Haritz Arrizabalaga and Paul de Bruyn for making local arrangements.

Bayesian analysis in ADMB

It is easy to produce Bayesian posterior distributions from your ADMB model. Simply define the values of interest as sdreport_ (or likeprof_) parameters and use the –mcmc N command line option (where N is the number of samples taken) when you run the model. A summary of the posterior distributions will be output in *.hst file. If posterior distributions are desired for multiple quantities of interest or the quantities from each sample are desired, a custom file can be created. The function "mceval_phase()" is used to write the data to file at each sample from the posterior. First the command line options –mcmc N and –mcsave S (where S represents how often to save the samples and if it is greater than one is used for thinning) are used to save the samples and then the command line option –mceval is used to evaluate the samples. Priors can be included by simply adding the negative log of the prior probability distribution to the objective function. The tpl file "simple.tpl" shows MCMC implementation for a simple linear regression with a prior [N(mean = 1,sd = 0.2)] on the slope (see sidebar).

```
simple -mcmc 10000 -mcsave 10

simple -mceval
```

Simple.tpl

```
DATA_SECTION
  init_int Nobs
  init_vector x(1,Nobs)
  init_vector y(1,Nobs)

PARAMETER_SECTION
  init_number a
  init_number b
  init_bounded_number sd(0.01,1)
  sdreport_number b_posterior
  objective_function_value f

PROCEDURE_SECTION
  f=Nobs*log(sd)+0.5*norm2((y-(a+b*x))/sd);
  f+=0.5*square((b-1.0)/0.2);
  b_posterior=b;
  if (mceval_phase())
  {
  ofstream out("posterior.dat",ios::app);
  out<<a<<" "<<b<<" "<< a+b*x<<endl;
  out.close();
  }
```

Simple.dat

```
# init_int Nobs
10
# init_vector x(1,Nobs)
1 2 3 4 5 6 7 8 9 10
# init_vector y(1,Nobs)
4.08     4.89     6.7      8.31     9.48
         10.89    10.53    12.51    14.24
         14.57
```

Reading ADMB custom output into R

Steve Martell has written a useful R function that reads the contents of a report file (or any output file) and stores the contents in R in the form of a list object. This function is capable of reading single variables, vectors, and 2-D arrays (including ragged arrays). The R-code was inspired by some earlier code developed by George Watters.

The format of *.tpl code requires the object name be printed first then the value(s), e.g.:

```
report<<"Biomass"<<endl<<bt<<endl;
```

Once the above line has been read into R the bt vector will be available as A$Biomass, where A is the list object, and Biomass is the name of the bt vector. To use this function, copy and past the R-code below and save this file as "reptoRlist.R". Then source this file in R (or put it at the top of your R-script). The function requires a file name argument (e.g., fn="MyModel.rep"). To read the contents of a report file, simply use:

```
A=reptoRlist(fn)
```

Then all of the objects in your report file will be stored in the list object A.

```
reptoRlist = function(fn)
{
ifile=scan(fn,what="character",flush=T,blank.lines.skip=F,quiet=T)
idx=sapply(as.double(ifile),is.na)
vnam=ifile[idx] #list names
nv=length(vnam) #number of objects
A=list()
ir=0
for(i in 1:nv)
{
ir=match(vnam[i],ifile)
if(i!=nv) irr=match(vnam[i+1],ifile) else irr=length(ifile)+1 #next row
dum=NA
if(irr-ir==2) dum=as.double(scan(fn,skip=ir,nlines=1,quiet=T,what=""))
if(irr-ir>2) dum=as.matrix(read.table(fn,skip=ir,nrow=irr-ir-1,fill=T))

if(is.numeric(dum))#Logical test to ensure dealing with numbers
{
A[[ vnam[i ]]]=dum
}
}
return(A)
}
```

Good News for Apple Users

Bill Clark and Johnoel Ancheta recently completed a port of AD Model Builder to MacOS. ADMB versions for GCC 4.0 and GCC

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

Were on the web! 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

Recent ADMB based publications

Branch, T.A. & Mikhalev, Y.A. (2008). Regional differences in length at sexual maturity for female blue whales based on recovered Soviet whaling data. Marine Mammal Science. 24: 690-703.

Branch, T.A., Mikhalev, Y.A. & Kato, H. (2009). Separating pygmy and Antarctic blue whales using long-forgotten ovarian data. Marine Mammal Science.

Dawson, H.A., Jones, M.L., Scribner, K.T., and Gilmore, S.A. (2009) An Assessment of Age Determination Methods for Great Lakes Larval Sea Lampreys. North American Journal of Fisheries Management 29: 914-927.

Gibson, A.J.F, Jones, R.A., and Bowlby, H.D. (2009) Equilibrium Analyses of a Population's Response to Recovery Activities: A Case Study with Atlantic Salmon. North American Journal of Fisheries Management 29: 958-974

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 (4):772-777.

Lehodey, P. Senina, I. and Murtugudde, R. (2008) A spatial ecosystem and populations dynamics model (SEAPODYM) – Modeling of tuna and tuna-like populations. Progress in Oceanography 78: 304–318.

Mesnil, B. Cotter, J. Fryer, R.J., Needle, C.L. and Trenkel, V.M. (2009) A review of fishery-independent assessment models, and initial evaluation based on simulated data. Aquat. Living Resour, 22: 207–216.

Senina, I. Sibert, J. Lehodey, P. (2008) Parameter estimation for basin-scale ecosystem-linked population models of large pelagic predators: Application to skipjack tuna. Progress in Oceanography 78: 319–335.

Wang, S-.P., Maunder, M.N., Aires-da-Silva, A. And Bayliff, W.H. (2009) Evaluating fishery impacts: application to bigeye tuna (Thunnus obesus) in the eastern Pacific Ocean. Fisheries Research 99: 106-111.