ADMB Developers' Workshop

June 20 - 23, 2011
National Center for Ecological Analysis and Synthesis
Santa Barbara, California

Organization of the workshop

The workshop was structured into a plenary secession on June 20 and subsequent breakout sessions starting on the afternoon of June 20 and continuing through June 23.

Plenary Session

Johnoel Ancheta presented an overview of using the subversion version control system and the BuildBot used for tracking and testing changes in the ADMB code. The ADMB project is currently using NCEAS facilities and Google code to house various components of the project. Google code is used to distribute binaries, keeps track of downloads, and issue tracking. NCEAS uses Redmine for issue tracking and it can be connected to subversion version control. Track for subversion and bugzilla are also options. It was suggested that Redmine should be considered as a better alternative. Redmine also other features such as a wiki. Jim Regetz gave a demonstration of the Redmine software. The Buildbot could be used to build and test glmmADMB and user supplied libraries that are not part of the source code.

John Sibert presented an overview of the ADMB code organization and documentation. The doccumention is generated automatically from comments in the code. Many of the available functions have yet to be fully documented. Several other issues were identified with the code organization and documentation. For example the names of the directories that contain the source files are not very informative. Doxygen allows the definition of “groups” and “subgroups” that can be used to organize the functions into related groups. A set of appropriate groups needs to be identified. The documentation needs to be easier to find on the website and search capabilities need to be added to the website.

John also presented an over view of how C++ functions are constructed and how arguments passed to and from functions. If you are returning a variable from a function you should always include the RETURN_ARRAYS_INCREMENT(); and RETURN_ARRAYS_DECREMENT(); function calls. Several conventions were identified including: arguments should be passed by reference (using &) for speed and to save stack memory. The const keyword should be used, when possible, so the value argument can’t be inadvertently changed. If only a single value is returned it should not be included as a argument passed to the function unless necessary.

Prevariable should be used instead of dvariable to allow the passing of sub-components of arrays.

Anders Nielsen demonstrated how a function is added to the source code using the matrix exponential as an example. Anders also demonstrated how to modify the LEX code in the TPL2CPP translator using the development of an R-like read.table() type functionality in the DATA_SECTION as an example.

Steve Martell demonstrated how user defined libraries can be created and used in ADMB code. Steve's extensive library of functions will be eventually incorporated in system of "contributed" libraries and header files. The plan will be to distribute the library as a new folder in the src directory such that the new code will contain the libraries. The documentation will appear in the API or Reference Guide Documentation (perhaps as a separate page) as the libraries contain the necessary doxygen code for documentation. Users wishing to use the functions defined in this library will have to explicitly include the library (using #include <statslib.h>) in the GLOBALS_SECTION of the ADMB template file.

Topics for breakout sessions:

Several breakout sessions were identified before and during the workshop. The breakout sessions were discussed and prioritized based on their importance and interest of the participants. The highest priority breakout sessions were then assigned to groups of participants. The potential breakout sessions included:

Improving API documentation with Doxygen*
Adding the "big picture" to API documentation
Writing adjoint code and interfacing to ADMB
Modifications to flex scripts*
Merge tpl2cpp and tpl2rem
Create PRIORS_SECTION*
Intuitive function names
glimm.ADMB*
Source code organization
Parallelization*
Issue tracking*
Automatic inclusion of RETURN_ARRAYS
Fixing the gamma function*

*indicates that the topic was taken up in a breakout session.

Source code documentation

The doxygen source code documentation system was discussed extensively. Several changes were made to the documentation of functions and classes.

The link to on the ADMB project website documentation page (http://admb-project.org/documentation/api) was changed from "API" to the more intuitive "Reference Guide"

A "search" tool was added.

A link to the up-to-date version of the ADMB README file was added provide information on obtaining and installing ADMB .

The list of files in Doxyfile was modified so that all source files in the source code directories are included in the documentation whether or not doxygen comments are present in the files.

New documentation was added to several source code files, and more information on the RETURN_ARRAYS_INCREMENT(); and RETURN_ARRAYS_DECREMENT(); functions was included.

Parallel processing

Chris Grandin demonstrated the use of parallel processing using graphics cards. Standard graphics cards on PCs have numerous processors that can be accessed from programs. The parallel processing was implemented in the TOP_OF_MAIN_SECTION of ADMB and therefore did not involve the calculation of derivatives.

The potential of using pthreads with the thread pool as implemented by Derek Seiple was discussed. It became clear by that the pthreads approach has a problem with the fact that there are static global variables, which are not thread safe. The pthreads approach will require either a major rewrite or some way of giving each thread its own copy of the static global variables. The possibility of using OpenMP was explored. OpenMP apperas to take care of the static global problem, but it has some other subtleties for which it is unclear how to handle.

One possibility of a simple implementation of parallel processing is to call an external program from within ADMB that does a computationally intensive calculation using parallel processing and returns the appropriate values (e.g. value and gradient) to ADMB. This is similar to the example presented by Chris Grandin.

Redmine page

A Redmine page (http://admb-project.org/redmine/projects/admb-test) was created to track issues (e.g. bugs) in the ADMB project. A list of issues and improvements were compiled for the new redmine page and will be added as time permits.

Gamma function error on the MacOS

During the compliation of the source code on the MacOS there was a warning about a duplicate declarations of the lgam function. This warning did not appear to cause any noticeable problems, except on the MacOS platforms using any function that required the lgam() function (e.g., gammln(), factln(), log_comb(), to name a few). The correct answer was obtained for data type objects, but an incorrect value was obtained for variable type objects. Multiple instances of gammln( ) were discovered in the source code. The duplicate symbols produced different results on different computing platforms (e.g. MacOS and linux). Solution of this issue was complex and provided a good opportunity to learn the use the “gdb” debugger. A temporary solution to the problem was identified during the course of the developers workshop, and once a permanent solution has been verified the changes will be committed to the SVN repository.

PRIORS_SECTION

The PRIORS_SECTION was created to allow users to easy put priors on model parameters using sytax like.

```
PRIORS_SECTION
a~dnorm(a_mean,a_variance)
```

Implementation of the priors section required modifying the LEX code for the TPL2CPP translator. The experienced gained during the development of the PRIORS_SECTION will facilitate future additions that require modification of the TPL2CPP translator.

Conclusions

Participants generally agreed that the first-ever ADMB developers workshop was very a productive and useful activity. Participants learned to download, modify, and upload the ADMB code from the subversion repository. Eleven new developers were authorized to commit code to the repository and over 25 revisions to the source code were committed. Several improvements were made to the code, test versions of the code, and infrastructure. These improvements included documentation, PRIORS_SECTION, and a Redmine website. The source code documentation was expanded to be more inclusive and other improvements were added to make the information more accessible to beginning users.

A second developers workshop was tentatively scheduled for February 2012 in Honolulu.

Specific Code Changes

Resolve some R-compatibility issues

Replace 'std dev' with 'std.dev' in manuals and examples

Write out 'std.dev' instead of 'std dev' in *.cor and *.std files

Simply some function names to make them more intuitive: define 'sumsq' as alias of 'norm2'

Matrix Exponential

Added matrix exponential for df1b2matrix

Added dmatrix version of matrix exponential expm

Dox for matrix exp

Added test case expm

Changes to class and function documentation (dox);

Documented negative binomial density functions

Simply dox build process with changes to Doxyfile and dox Makefile

Dox for dvar sum and *

Dox for RETURN_ARRAYS...

Dox for min max og dvar_vector

Add 'make docs' to configured makefile.

Define PI (and M_PI) in fvar.hpp, replace 3.14159* with PI everywhere, and fix PI compile error in cgamdev.cpp

Modifications to flex code

Fix warnings in tpl2cpp.lex and tpl2rem.lex

Fix the flex builds

Modications to admb scripts

Fix scripts to pass CXXFLAGS and LDFLAGS to adcomp and adlink.

Simplify and comment CXXFLAGS and LDFLAGS in adcomp, adlink, and admb

Remove old scripts.

Remove unused file.

Add missing installation step in README.txt and suggestions for MacOS users

Don't force build hs_sparse.cpp

List of Participants

Brooks, Mollie McMaster University mbrooks@ufl.edu	Fournier, Dave Otter Research Ltd. 250-655-3364 davef@otter-rsch.com
Grandin, Chris Fisheries and Oceans Canada (250) 756-7170 chris.grandin@dfo-mpo.gc.ca	Liu, Weihai Michigan State University (517) 355-0126 liuweih@msu.edu
Magnusson, Arni Marine Research Institute +354 575-2000 arnima@hafro.is	Martell, Steve J. University of British Columbia (604) 822-0484 s.martell@fisheries.ubc.ca
Maunder, Mark N. InterAmerican Tropical Tuna Commission (858) 546-7027 mmaunder@iattc.org	Nielsen, Anders Technical University of Denmark anders@nielsensweb.org
Regetz, Jim University of California, Santa Barbara (805) 892-2533 regetz@nceas.ucsb.edu	Seiple, Derek University of Hawaii dseiple84@gmail.com
Sibert, John R. University of Hawaii, Mānoa (808) 956-4109 sibert@hawaii.edu	Sippel, Tim University of Hawaii, Mānoa (808)956-6319 tsippel@gmail.com
Skaug, Hans J. University of Bergen +47 55 58 48 61 hans.skaug@math.uib.no	Willestofte Berg, Casper Technical University of Denmark +45 35883433
A’mar, Teresa NOAA Alaska Fisheries Science Center (206) 526-4068 teresa.amar@noaa.gov	