

ADMB Developers Workshop Report

March 13-16, 2012
Imin Conference Center, Kamehameha Room (2nd floor)
1777 East-West Road
University of Hawaii at Manoa Campus
8:30 AM - 4:30PM



Introduction

The meeting convened and adopted the Agenda attached below. Jim Ianelli was appointed rapporteur. Several new participants were welcomed to the workshop. In particular, Matthew Supernaw, a recent hire by NMFS to guide NOAAs involvement in ADMB developments. The list of participants is also attached.

During the week the ADMB foundation held their annual meeting and nominated Dr. Teresa Amar and Casper Berg, as foundation members.

The following sections summarize the activities of the workshop and is followed by a task list.

Developments and issues

Contributed Libraries

Steve Martell presented an update from the previous developers workshop. A new “Contrib” directory was created where developer could submit their own custom routines and functions for wider use among the ADMB community. At this developers workshop, modifications to the compile scripts and make files

were made such that in the very near future users will be able to use the contributed libraries by adding an `#include<libname>` statement to the `GLOBALS_SECTION` of their ADMB template files. There are now a large number of statistical routines and ecological routines that have been incorporated into the contributed libraries of the source code, and documentation for these routines are available through the doxygen API on the ADMB-project website. There is also a short presentation on how to include user contributed functions in your ADMB programs available at: <http://admb-project.org/developers/admb-developers-workshop-march-13-16-2012/user-contributed-libraries/view>.

GLMMADMB issues

As part of the ADMB initiative to broaden applications into different fields (other than fisheries) the mixed model package developed for the R environment has received a fair amount of activity and interest. Key points identified were:

- The group recommended that there is a need to collect example issues and differences from LME
- Some R users have found the value in learning ADMB directly
- Documentation on how to extend the `.tpl` file/application outside of R as an example is worth pursuing.

ADMB running parallel processors

Derek Seiple made presentation on the status of parallel processing with ADMB. Open-MPI approach was selected and send and return functions done for constants. This provided the ability to test with some time-intensive calculations (e.g., Hessian matrix). Since open MPI recommends wrapper compilers a Configure script has been developed:

```
make ./configure --enable-mpi
```

So far the constants is clearly working for Hessians and separable functions. The group inquired about converting a model to constants and running for MCMC purposes. This could be implemented as a possibility but wouldn't be required to test the efficacy of parallelization (i.e., is it worth the trouble?).

Derek Seiple gave a presentation on the current efforts and progress of parallel processing using Open MPI. He led a break out session to demonstrate how to:

- Install Open MPI,
- Get and compile the parallel code, and
- run sample programs with parallel features.

Testing was performed to see how the new parallel code improved speed. For the standard Hessian calculation a speedup of more than double was observed. For separable functions a speedup of about 30 percent was observed. Potential bottle-necks in the source code were discussed along with how other Open MPI features may be used to improve computation speed.

For the Hessian problem Yukio had a savings of more than 50% for about 5 cores. It may be slower as the cache fills up.

An example was also provided by Ian Taylor was run on the HPC at Univ of FL <http://www.hpc.ufl.edu/> and was allocated 2000MB for each model run. The model was shared by master and slaves. Results indicated (for the yelloweye rockfish SS3 application) had improvements 10% with two cores and changed less with more slaves (Figure 2). The workshop participants noted that for comparison of run-

times for Hessian computations one would need to subtract the computation time involved in inverting the matrix (since this part of the code isn't parallelized).

Info on using OpenMPI on this cluster is here <http://wiki.hpc.ufl.edu/index.php/OpenMPI>

Version	ABI	Compiler	Location	Notes
1.2.7	64 bit	Intel v10	/usr/mpi/intel/openmpi-1.2.7-1	For Torque/Maui

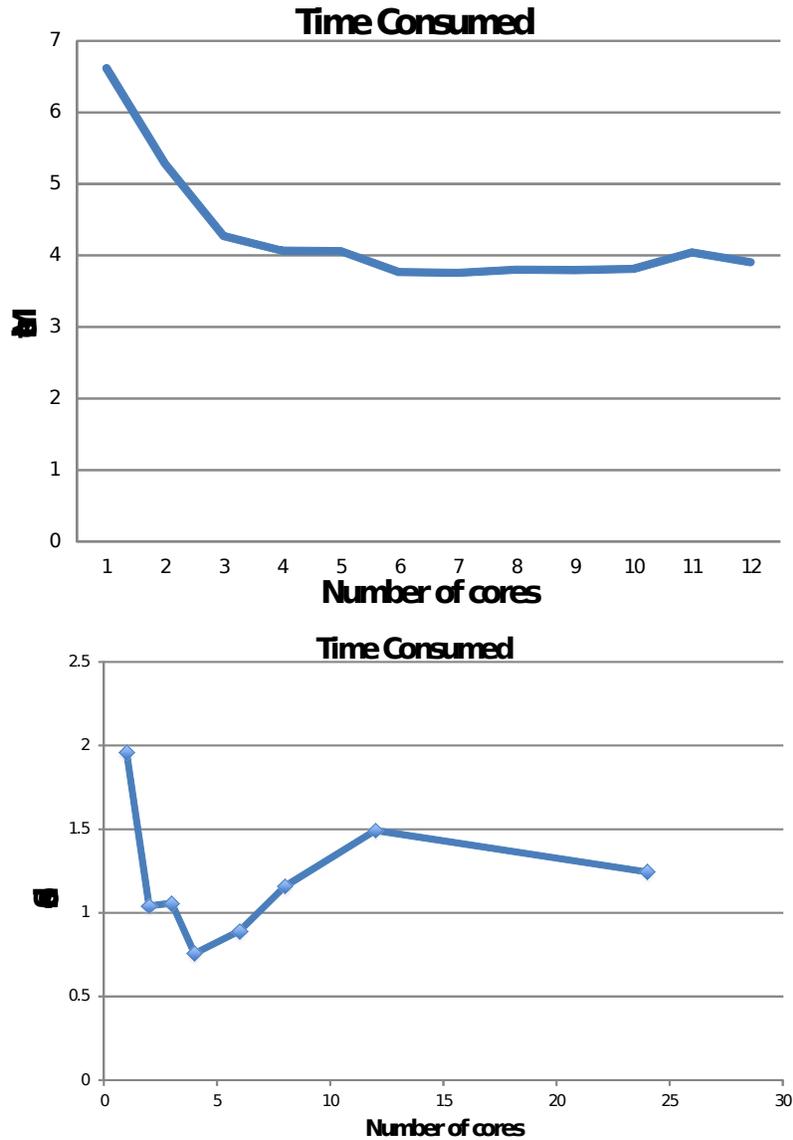


Figure 1. Tests of parallelization performed by Yukio Takehashii using SS3 and the Hessian calculation.

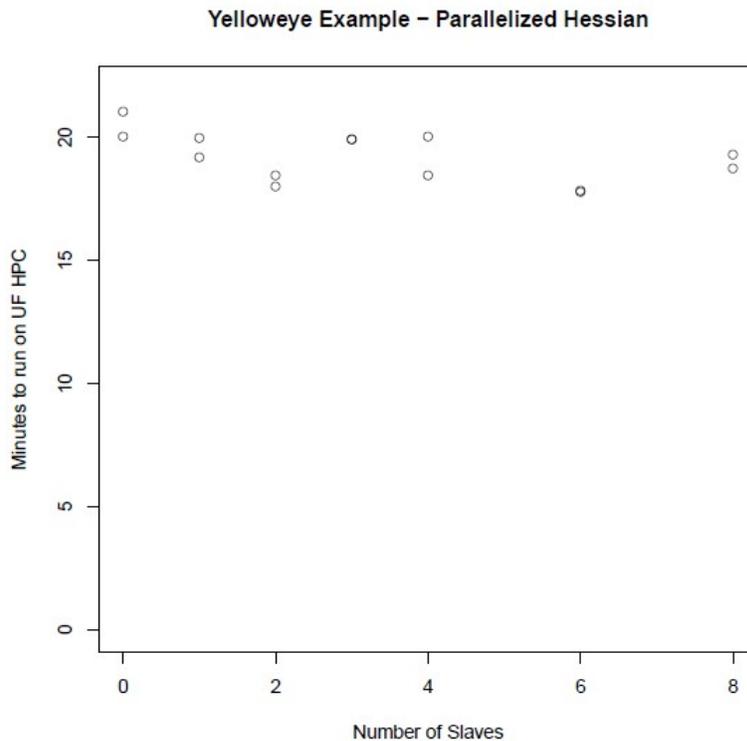


Figure 2. Example performance change with number of slave processors for a yelloweye rockfish application of SS3. Running it with `-sdonly` took 17.3 minutes which implies that the savings (percentage-wise) is greater than indicated here.

For the two tests that were conducted, gains generally appeared to diminish after 3 or 4 nodes (CPUs).

Factors ADMB:

Anders Nielsen presented his work on developing a new class of variable in ADMB. These are factors (categorical variables) which can be useful when prescribing separate parameters to a finite number of predefined categories (e.g. gender specific effects). Such facility is common in most statistical software. In ADMB this process has always been fairly manual following general steps:

- 1) Read in the data and convert the relevant vector to a integer vector.
- 2) Ensure that it is numbered sequentially.
- 3) Count the unique levels.
- 4) Declare a parameter vector with length equal to the number of unique levels. Then finally
- 5) when predicting the i 'th observation a sort of double indexing can be used, as in: $\text{theta}(\text{iv}(i))$, where theta is the parameter, iv is the categorizing vector.

Step 1)-5) is doable and second nature to most AD Model Builder users, but it can appear cumbersome to new users of ADMB. A first step to introduce factors in AD Model was taken at this meeting. An internal factor class was added and appropriate changes to the `tpl2cpp` parser (`lex-file`) was made. These changes allow the user to declare a factor in the parameter section, as in: `init_factor theta(gender)`. That statement takes care of 2)-4) above, and when predicting i 'th observation step 5), the user can avoid the double index and only have to write `theta(i)`. The changes will appear in the next release of AD Model Builder.

(More info see the presentation on factors at: (<http://admb-project.org/community/admb-developers-workshop-march-13-16-2012>))

Spatial modelling in ADMB

Fitting GLM-type of models with latent Gaussian random field was outlined. The latent random field is implemented via the random effects functionality of ADMB. There are three available approaches described under Examples on the website:

- 1) The geostatistical approach, in which one specify the spatial covariance matrix.
- 2) Gaussian Markov random field (GMRF) and CAR (conditional autoregressive) where a neighbor structure is defined and
- 3) hybrid (between 1 and 2) approach. The second and third approach utilizes the sparse matrix feature of ADMB, and is hence computationally efficient.

A way to improve the geostatistical approach 1) has been identified, and will be attempted implemented during the workshop.

A datafile reading issue was identified and Casper was to put on issues list

Speedup of general spatial models:

Following Hans Skaug's presentation it was decided that a change was required in the cpp2rem parser (lex-parser), such that it could take advantage of some improved c++-code that Dave has supplied to Hans earlier. The changes are affecting how the so called 'NORMAL_PRIOR_FUNCTION' is processed. Dave's improved c++-code is more advanced in how it updates the model parameters, as it does not update (the few) parameters in the covariance structure at all iterations, which leads to a great reduction in the time it takes to compute the likelihood. In test cases this leads to a 10 times speedup of the entire optimization problem. The required changes to the parser code has now been completed and tested. A minor task of merging Dave's changed C++-files with the current versions in the repository - which may have changed in the meantime - is remaining before this change is released. (More info see the presentation on spatial models at:

(<http://admb-project.org/community/admb-developers-workshop-march-13-16-2012>))

The workshop participants were interested in learning more Kasper (at Danish Institute) efforts using an optimized matrix library for working with high resolution spatial models. There is interest in implementing this in ADMB, need to review examples online.

CLANG

Clang was presented by Matthew Supernaw (programmer hired by NMFS presently working out of St. Petersburg FL). During the week this compiler was used to compile ADMB from source and some limited tests were made. Also the compiler was built on a windows machine for eventual testing and further development as an alternative in that environment since it is touted as being platform independent. A utility called "Valgrind" which can automatically detect many memory management and threading bugs, and perform detailed profiling to help speed up your programs.

PRIORS_SECTION

Weihai Liu presented an update and example of application using PRIORS_SECTION. There remains some lex issues (admb2cpp etc) that was worked on and fixed during the week and will form part of the June 2012 release.

Templates for functions

Mollie Brooks presented examples of how templates for writing functions can be conducted. It was noted that the forthcoming Newsletter should include a how-to for including such functionality.

Underflow issue

Some functions that need to invert a matrix first check if there are any zeros along the determinant. They used to do this by taking the product of the elements of the determinant. This was susceptible to underflow. I experienced this problem with a variance-covariance matrix of size 350 with 0.007 on the diagonal. Derek showed me how to grep for 'det *=' and 'det*='. We changed all occurrences of this so that now we keep track of the log of the determinant (e.g. `ln_det += log(A(j,j))`) and check if `exp(ln_det) == 0.0`. This might not be a very good check (Dave's old version of the solve function only checked if the entire matrix was full of zeros) but at least this isn't as susceptible to underflow. We ran my code and got the same answers as when I run it using Dave's old solve function. We also did make verify and the benchmarks were identical.

ADMB manual and other documentation

It was noted that there is very little activity in correcting errors in manual and documentation. John Sibert reported that the SVN works well for updating and that corrections are automatically propagated to the online version.

Build bot

Johnoel Ancheta provided status updated on the build bot and where things are working. A new version (10.2) was recommended for release immediately after workshop with a version 11.0 scheduled for June.

The group recommended that revisions on buildbot/snapshots should be named 10.2-r## where ## is the SVN version number. Issues that are required to be in 10.2: upgraded solve function and resolution of underflow issue.

Debugging

The utility of using compiler functions to catch floating point errors was demonstrated by Dave Fournier using `Fenv.h` and `Feenable.h`. These work to easily trap the line (using a simple debugger) where the floating point error occurred. However, tests on windows platform failed because the files haven't been adopted for the windows gnu compiler standard installation.

Building from scratch on windows platforms

The group discussed aspects of issues required to build ADMB using public domain software. Workable approaches were identified/demonstrated:

- 1) Using mingw (msys)

Pros: Configure scripts under linux appear to work the same

- Cons: Adds a layer of tools/requirements (not really very minimal)
- 2) Using vanilla windows tools
 - Pros: really minimal
 - Cons: may be unstable in future versions
- 3) Clang
 - Pros: portable over all OS's
 - Cons: Binaries for windows unavailable (can be compiled)

All three of these methods were worked on fsin breakout sessions.

Arni's IDE and other work

The following points were raised during or prior to the workshop:

1. Enable safe compilation in user scripts (adcomp, adlink, admb) as default. The new scripts will introduce -o as a new option for optimized compilation and ignore the -s option since that will already be the default behavior. These are expected to be implemented prior to next release.
2. Dave Fournier reported success with DLL compilation on the Linux-GCC platform. With renewed enthusiasm, "task forces" have been formed for different platforms, coordinated by Arni. Homepage, <http://admb-project.org/developers/dll>.
3. **MAC ADMB-IDE** was working on Steve Martell's MAC OSX after modifying a few lines in the .emacs settings. The required changes included commenting out max screen size and rebinding some keys. Still, many keybindings were not working, due to Mac-specific keybindings that are related to multimedia. Would need more work before thinking about ADMB-IDE installer for MacOS. Steve indicated that some of his students might be interested in working on this.
4. Now that the development team is modifying the tpl2* Flex code, it would be very beneficial to embed Doxygen comments to describe the changes and to improve our overall understanding about the syntax and structure of the Flex files. Someone at the meeting mentioned that Doxygen comments could probably not be embedded into the Flex files This should be explored further
5. Increase default memory allocation settings (ARRAY_MEMBLOCK_SIZE, CMPDIF_BUFFER_SIZE, GRADSTACK_BUFFER_SIZE, MAX_DLINKS, MAX_NVAR_OFFSET, NUM_DEPENDENT_VARIABLES, NUM_RETURN_ARRAYS, RETURN_ARRAYS_SIZE, etc.). The current settings were sensible in the 1990s, but today users need to increase the limits for typical modelling applications. Which current memory settings have users found most limiting? This is being covered in recent emails, thanks so much for you contributions here!

Task List

List of tasks discussed at the Workshop with names of people who agreed to take responsibility for each.

Installation

Ensure that the installation for all supported platforms work seamlessly

IDE Arni Magnuson

Mac/Unix Matthew Supernaw

Windows Chris Grandin

Documentation Ian Taylor

Ongoing

Building from source by platform/environment

IDE Arni Magnuson
Mac/Unix Matthew Supernaw
Windows Chris Grandin
Documentation Ian Taylor, each of above

Documentation

Developers to insert dox in new code and and updated files including modifications to LaTeX documents as changes are made to the code and as errors are noticed.

Overall coordination and advice: John Sibert
Doxygen documentation John Sibert
Develop "style" guide for documentation: John Sibert
Manuals All

Technical tutorials

Using SVN, building source, making changes, committing changes
using debugger (e.g., in IDE or DDD) Ian/Johnoel
Coding standards/conventions Johnoel/Matthew

Redmine issue list maintenance

Redmined people to contribute/complete tasks Mark Maunder

Education/outreach

Coordinate advertising of workshops Jims Ianelli and Bence
Facilitate ADMB developers' workshops John Sibert, Dodie Lau

MPI on multiple cores

Derek/Teresa/Anders/Yukio

Priors section

Maintain tutorials and post examples Weihai

Spatial-temporal modeling

Implement examples Hans/Anders

Solve function

Underflow done
Old version Mollie

Website

Mollie/Allan

Contributed libraries

How to write contributed functions

Basics Steve Martell

Templates Mollie Brooks

How to use contributed functions Mollie/Steve

Catalog of contributed functions Mark Maunder

Including contributed functions with build Chris Grandin

Put functions in common libraries Mark Maunder

Robustifying the random effects module

Noinit ??

-ilmn gradient Mollie Brooks

Efficient robust minimizer that uses gradient and hessian in inner minimization Dave Fournier

Improving solve function, use old algorithm Derek Seiple

Protocol for providing errors to Dave (ADMB-GLMM) Ben Bolker

Debugging

Clear instructions that work Arni Magnuson

Should have a debug and no optimization version available for every compiler version Group

Make functions to print out objects in debugging Completed (?) needs posting

Underflow issue

Need to maintain pivoting stuff Apparently done?

Summing log of diagonal or using Numerical recipes version,

Website organization and examples

Reconsider website organization John Sibert

Amalgamation of tpl2cpp and tpl2rem

Combine this functionality Dave Fournier

ADMB-GLMM R

List of issues discovered Ben

Catalog of examples Ben

Show example how to take ADMBGLMM and change it Course

Test 64 bit code

Is it really working? (make long to longlong, Sibert fixed it?)

Memory allocation

Decide which memory allocation settings should have larger defaults Ian

Profile likelihood

Revisit the profile likelihood code to:

- 1) extend it to RE models,
- 2) optionally change the method to augmented Lagrangian, and
- 3) more flexible point selection.

Report

Everyone who presents/discusses/breakout groups provide summary Jim Ianelli

Valgrind tutorial

Provide a simple example (from a .tpl file) of how to apply Matthew Supernaw

Agenda

Working hours 8:30AM - 4:30PM

Breaks: 10:00AM-10:30AM; 12:00PM-1:00PM; 2:30PM-3:00PM

First day will be a plenary meeting; subsequent days will be *ad hoc* break out sessions to work on specific topics.

At some point, there will be an ADMB Foundation official meeting.

Tuesday, March 13

Opening and welcome

Discussion of agenda

Continuation of issues from June 2011 workshop

Parallel processing - Derek

Priors section - Mark, Weihai

Redmine

Website organization

Amalgamation of `tpl2cpp` and `tpl2rem`

64 bit code - is it really working?

Update on repository and buildbot - Johnnoel

Documentation: User manuals and API

Courses

Debug

Potential topics for breakout sessions

Useful Functions for stock assessment models

Examples - Hans

Funding sources - Mark

Spatio-temporal random effect models

Robustifying the random effects module

Factors in ADMB - Anders

Including `valgrind` in buildbot

Decide which memory allocation settings should have larger defaults

Revisit the profile likelihood code to 1) extend it to RE models, 2) optionally change the method to augmented Lagrangian, and 3) more flexible point selection.

Debugger tutorial.

Wednesday, March 14

3:00PM ADMB Foundation General Meeting

Thursday, March 15

6:00PM Workshop dinner, Sibert's house. Plenty food. BYOB.

Friday, March 16

Wrap up

Participants

ADMB Foundation

Mark Maunder <mmaunder@iattc.org> La Jolla, USA
Jim Ianelli <Jim.Ianelli@noaa.gov> Seattle, USA
Anders Nielsen <anders@niensensweb.org> Copenhagen Denmark

Core Developers

Johnnoel Ancheta <johnnoel@hawaii.edu> Hawaii USA
Arni Magnusson <arnima@hafro.is> - Iceland
Hans Skaug <Hans.Skaug@math.uib.no> Norway
Steve Martell <s.martell@fisheries.ubc.ca>- Canada
Mollie Brooks <mbrooks@ufl.edu> - Florida USA
John Sibert <sibert@hawaii.edu>- Hawaii USA
Casper Berg <cbe@aqua.dtu.dk> - Denmark
Weihei Liu <liuweih@msu.edu> - Michigan US
Chris Grandin <Chris.Grandin@dfo-mpo.gc.ca> Canada

Consultants

David Fournier <davef@otter-rsch.com> - Canada
Derek Seiple <dseiple84@gmail.com> Florida US

Other Developers

Pierre Kleiber <Pierre.Kleiber@noaa.gov> Hawaii US
Yukio Takeuchi <yukiote@fra.affrc.go.jp> Japan

NOAA

Jon Brodziak <Jon.Brodziak@noaa.gov>
Chris Legault <chris.legault@noaa.gov>
Matthew Supernaw <matthew.supernaw@noaa.gov>
Ian Taylor <ian.taylor@noaa.gov>
Teresa Amar <teresa.amar@noaa.gov>
Ray Conser <ray.conser@noaa.gov>
Jim Ianelli <Jim.Ianelli@noaa.gov>