AD Model Builder IDE

Emacs `admb-mode` without the Emacs
Version 3.4.5-1

Arni Magnusson

Table of Contents

1 Introduction

1.1 Emacs admb-mode

The process of creating statistical models with AD Model Builder (ADMB) involves writing, compiling, and testing. An integrated development environment (IDE) allows the user to perform these tasks more efficiently than with a basic editor and a shell.

GNU Emacs is a complex and powerful editor that comes with particularly good support for R, LaTeX, backup/revision control, and other useful tools for statistical computing. Its `admb-mode` provides syntax highlighting, compilation, file manipulation, outline code navigation, templates, and smaller tools for creating ADMB models. Emacs users can fetch `admb-mode` from http://admb-project.org/community/editing-tools/emacs/admb.el/view and start using it right away, after reading the commentary at the top of the file.

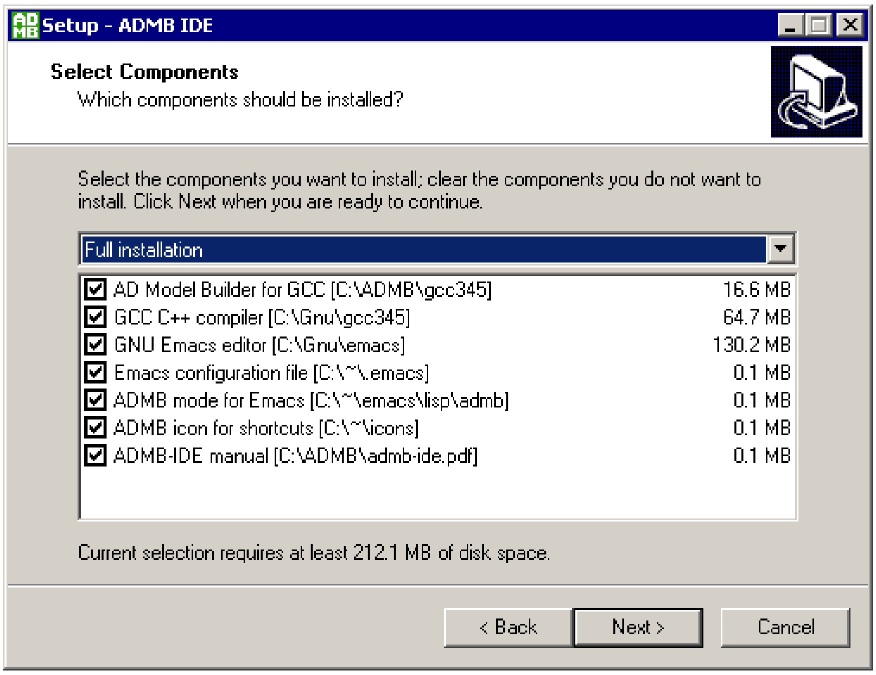
The problem with Emacs is that it requires considerable time to learn and configure, although for heavy-duty statistical computing this can be a rewarding investment. As the programmer Larry Wall once said: “If ease of use was the highest goal, we’d all be driving golf carts.” The http://admb-project.org/community/editing-tools/emacs page contains some pointers for setting up and learning Emacs. There are, however, good reasons why many users may not feel like adopting Emacs as their main editor, but would still appreciate a simple IDE for ADMB.

The rest of this tutorial demonstrates how Emacs with `admb-mode` can be configured as a user-friendly ADMB-IDE, without learning the details of Emacs. This is achieved with an unusual ‘`.emacs`’ configuration file that emulates common keybindings of basic editors, while disabling some of the most used Emacs keybindings. This ‘`.emacs`’ file is therefore not intended for experienced Emacs users, although they may find it an interesting read.

1.2 ADMB-IDE for Windows

Somebody set up us the bomb.

There are two ways to install ADMB-IDE for Windows. The easiest way is to download ‘`admb-ide-345-1.exe`’ from http://admb-project.org and install all components as shown in this screenshot:



Select Components

Which components should be installed?

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\gcc345]	16.6 MB
☑ GCC C++ compiler [C:\Gnu\gcc345]	64.7 MB
☑ GNU Emacs editor [C:\Gnu\emacs]	130.2 MB
☑ Emacs configuration file [C:\~\.emacs]	0.1 MB
☑ ADMB mode for Emacs [C:\~\emacs\lisp\admb]	0.1 MB
☑ ADMB icon for shortcuts [C:\~\icons]	0.1 MB
☑ ADMB-IDE manual [C:\ADMB\admb-ide.pdf]	0.1 MB

Current selection requires at least 212.1 MB of disk space.

The components are arranged in three directories:

`c:/~` Home directory for configuration files

`c:/admb` AD Model Builder

`c:/gnu` Emacs and GCC

This is a practical setup for other free software as well. Take for example the R statistical software. By separating the main program (`c:/gnu/r`) from the user settings (`c:/~/.Rprofile`, `c:/~/Rconsole`) and user libraries (`c:/~/r/library`), the main program can be removed and upgraded without affecting the user setup.

The ADMB-IDE installer offers the user to unselect any of the components, but this is not recommended unless that component is already residing in the specified directory. Users that are not happy with the default directory structure can set up and configure individual components by hand. The following guidelines may be useful for that:

http://admb-project.org/documentation

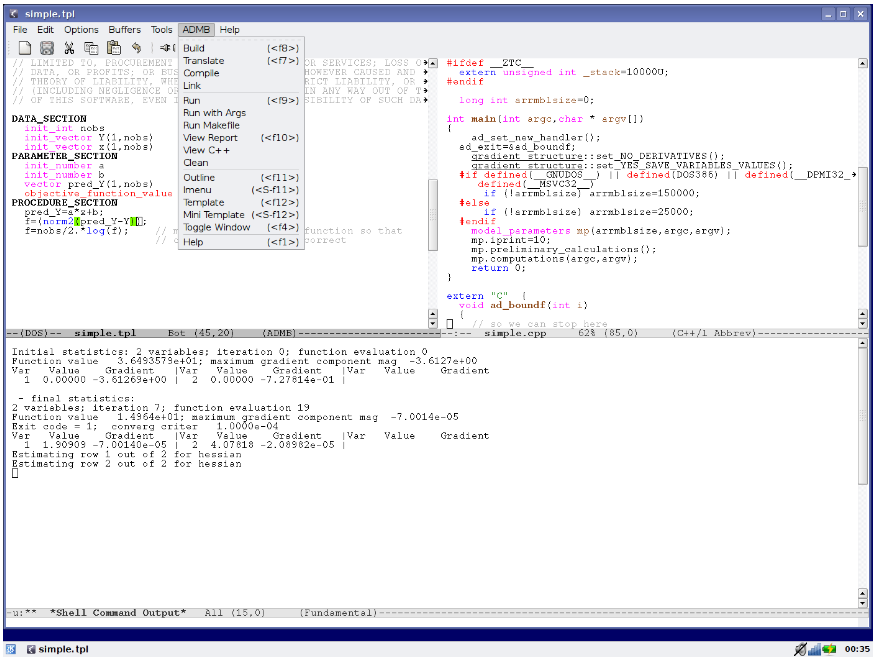
http://admb-project.org/community/editing-tools/emacs/install

http://mingw.org/wiki/HOWTO_Install_the_MinGW_GCC_Compiler_Suite

1.3 ADMB-IDE for Linux

Behold, the blueprints of a golf cart.

All Linux distributions include GCC, it's easy to install GNU Emacs using a package manager, and ADMB binaries can be downloaded from http://admb-project.org. With these programs in place, `admb-mode` can be used with or without the "non-Emacs" keybindings, as explained in Section 1.1 [Emacs `admb-mode`], page 1.



```
// LIMITED TO, PROCUREMENT                OR SERVICES; LOSS O
// DATA, OR PROFITS; OR BUS               HOWEVER CAUSED AND
// THEORY OF LIABILITY, WHE               RICT LIABILITY, OR
// (INCLUDING NEGLIGENCE OR               IN ANY WAY OUT OF T
// OF THIS SOFTWARE, EVEN I               SIBILITY OF SUCH DA

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_value
PROCEDURE_SECTION
  pred_Y=a*x+b;
  f=(norm2(pred_Y-Y));
  f=nobs/2.*log(f);      // m                 function so that
                         // c                 correct
```

```
#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)
  {
    // so we can stop here
```

```
Initial statistics: 2 variables; iteration 0; function evaluation 0
Function value   3.6493579e+01; maximum gradient component mag  -3.6127e+00
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1  0.00000 -3.61269e+00 |  2  0.00000 -7.27814e-01 |

 - final statistics:
2 variables; iteration 7; function evaluation 19
Function value   1.4964e+01; maximum gradient component mag  -7.0014e-05
Exit code = 1;  converg criter   1.0000e-04
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1  1.90909 -7.00140e-05 |  2  4.07818 -2.08982e-05 |
Estimating row 1 out of 2 for hessian
Estimating row 2 out of 2 for hessian
```

2 Tutorial

2.1 Create working copy of `simple`

Open Windows Explorer and create a folder called `c:/simple`. Then navigate to `c:/admb/gcc/examples/admb/simple`, ignore the ‘`Makefile`’, and just copy the model and data files, creating:

```
c:/simple/simple.dat
c:/simple/simple.tpl
```

Now double-click ‘`simple.tpl`’. It should open in Emacs and the code should be in different colors, depending on the syntax.

2.2 Compile, run, and view the results

2.3 Other tasks

3 Graphical interface

3.1 Menu

Menu label	Emacs command	Purpose
Build	`admb-make`	Build executable from TPL
Translate	`admb-tpl2cpp`	Translate TPL to C++
Compile	`admb-compile`	Compile C++ to object code
Link	`admb-link`	Link object code to exe
Run	`admb-run`	Run executable
Run with Args	`admb-run-args`	Run executable with args
Run Makefile	`admb-run-makefile`	Run Makefile in current dir
View Estimates	`admb-cor`	Open .cor file
View Report	`admb-rep`	Open .rep file in browser
View C++	`admb-cpp`	Open C++ file
Clean	`admb-clean`	Remove temporary files
Outline	`admb-outline`	Navigate with outline
Imenu	`imenu`	Navigate with imenu
Template	`admb-template`	Insert template
Mini Template	`admb-template-mini`	Insert minimal template
Toggle Window	`admb-toggle-window`	Toggle secondary window
Help	`admb-help`	Show help page

3.2 Toolbar

Translate, Build, Run, View report

4 Shortcut keys

4.1 Function keys

`f1`	Help
`f2`	ADMB mode
`f3`	Data mode
`f4`	Toggle secondary window
`C-f4`	Close
`M-f4`	Quit
`f5`	Reload
`f6`	Other window
`f7`	Translate TPL → C++
`f8`	Build executable from TPL
`f9`	Run executable
`f10`	Open .rep file in browser
`f11`	Navigate with outline
`S-f11`	Navigate with imenu
`f12`	Insert template
`S-f12`	Insert minimal template

4.2 Rest of keyboard

`escape`	Cancel dialog, close other windows
`C-return`	Rectangle functions
`C-space`	Open recent files
`C-a`	Select all
`C-c`	Copy
`C-f`	Find
`C-g`	Goto line
`C-l`	Recenter
`C-n`	New
`C-o`	Open
`C-q`	Quit
`C-r`	Replace
`C-s`	Save
`C-S`	Save as
`C-v`	Paste

`C-w`	Close
`C-x`	Cut
`C-x 2`	Split window above/below
`C-x 3`	Split window left/right
`C-z`	Undo

4.3 Mouse

`C-mouse-1`	Switch buffers
`mouse-3`	Navigate with imenu