Software overview

TMB and stock assessment

Arni Magnusson
Anders Nielsen

ICES, 2–6 Nov 2015

Outline

Compiler

required to build model

Outline

Outline

Compiler

required to build model

TMB

history, components, usage

Install

basic installation, Virtual TMB, configuration

Outline

Compiler

required to build model

TMB

history, components, usage

Install

basic installation, Virtual TMB, configuration

Test

verify that TMB is ready to use

C++ compilers

GCC

default compiler in Linux and Windows (with Rtools)

free software, Linux/Mac/Windows

C++ compilers

GCC

default compiler in Linux and Windows (with Rtools)
free software, Linux/Mac/Windows

Clang

default compiler in Mac (with Xcode)
free software, Linux/Mac/Windows

History of TMB

2009

Kasper Kristensen begins developing TMB

History of TMB

2009

Kasper Kristensen begins developing TMB

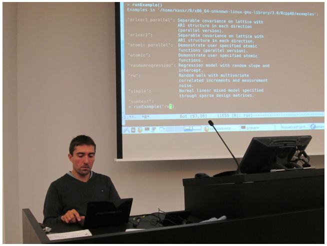
2013

uploaded to GitHub

demonstrated at ADMB workshop in Reykjavik

History of TMB

Flashbacks from Reykjavik 2013



```
> runExample()
Examples in '/home/kaskr/R/x86_64-unknown-linux-gnu-library/3.0/RcppAD/examples':

"ar1xar1_parallel": Separable covariance on lattice with
                    AR1 structure in each direction
                    (parallel version).
"ar1xar1":          Separable covariance on lattice with
                    AR1 structure in each direction.
"atomic_parallel":  Demonstrate user specified atomic
                    functions (parallel version).
"atomic":           Demonstrate user specified atomic
                    functions.
"randomregression": Regression model with random slope and
                    intercept.
"rw":               Random walk with multivariate
                    correlated increments and measurement
                    noise.
"simple":           Normal linear mixed model specified
                    through sparse design matrices.

"sumtest":
> runExample("rw")
-1:**-  *R*       Bot (97,16)    (iESS [R]: run)------------------
```

History of TMB

2009

Kasper Kristensen begins developing TMB

2013

uploaded to GitHub

demonstrated at ADMB workshop in Reykjavik

2014

workshops in different parts of the world

History of TMB

2009

Kasper Kristensen begins developing TMB

2013

uploaded to GitHub

demonstrated at ADMB workshop in Reykjavik

2014

workshops in different parts of the world

2015

paper available on arXiv

package available on CRAN

TMB components

TMB

CRAN package: R functions, TMB.so/TMB.dll

Laplace approximation: integrate out random effects

TMB components

TMB

CRAN package: R functions, TMB.so/TMB.dll

Laplace approximation: integrate out random effects

C++

CppAD: automatic differentiation

BLAS and EIGEN: linear algebra in C++

CHOLMOD: sparse matrix routines

OpenMP: parallel computations

TMB components

TMB

CRAN package: R functions, TMB.so/TMB.dll

Laplace approximation: integrate out random effects

C++

CppAD: automatic differentiation

BLAS and EIGEN: linear algebra in C++

CHOLMOD: sparse matrix routines

OpenMP: parallel computations

R

Matrix: linear algebra in R

nlminb: fit model

Using TMB

Basic steps

Prepare data

- Write R and C++ code
- Compile
- Run model

View results

Using TMB

Basic steps

Prepare data

- Write R and C++ code
- Compile
- Run model

View results

User environment

- Editor and R (separate)
- IDE (Rgui, RStudio, Emacs, other)

Using TMB

Short **TMB-IDE** demo

Basic installation

Linux

- Install TMB from CRAN

Basic installation

Linux

- Install TMB from CRAN

Mac

- Install Xcode
- Install TMB from CRAN

Basic installation

Linux

- Install TMB from CRAN

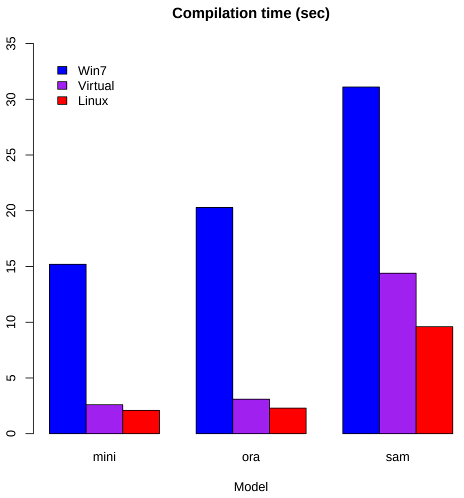
Mac

- Install Xcode
- Install TMB from CRAN

Windows

- Install Rtools, set PATH
- Install TMB from CRAN

Compilation takes a long time in Windows . . .



Virtual TMB

Linux

- Install TMB from CRAN

Mac

Install Xcode

Install TMB from CRAN

. . . or

- Install VirtualBox
- Install Virtual TMB

Faster? Convenient?

Windows

Install Rtools, set PATH

Install TMB from CRAN

- Install VirtualBox
- Install Virtual TMB

$6 \times$ faster

Configure

Linux and Mac

- Precompile
- Decide whether to use GCC or Clang
- Define environment variable `R_MAKEVARS_USER` and select compiler settings `CXX` and `CXXFLAGS`

Configure

Linux and Mac

- Precompile
- Decide whether to use GCC or Clang
- Define environment variable `R_MAKEVARS_USER` and select compiler settings `CXX` and `CXXFLAGS`

Windows

- Set the `PATH` environment variable so it points to the Rtools `make` and `gcc`, and no other instances of `make` and `gcc`

Configure

Linux and Mac

- Precompile
- Decide whether to use GCC or Clang
- Define environment variable `R_MAKEVARS_USER` and select compiler settings `CXX` and `CXXFLAGS`

Windows

- Set the `PATH` environment variable so it points to the Rtools `make` and `gcc`, and no other instances of `make` and `gcc`

Virtual TMB

Fully configured, ready to use

Test installation

Open **R** and type:

```
library(TMB)
runExample("simple", clean=TRUE)
example(sdreport)
```

Linear regression

1. Create working folder like `c:/workshop/linreg`
2. Copy `linreg.cpp`, `linreg.dat`, and `linreg.R` into working folder
3. Build and run

Exercise

1. Copy `c:/workshop/linreg` to `c:/workshop/mu`
2. Rename `linreg.*` files to `mu.*`
3. Modify R and C++ code so the model estimates μ