



Rcpp demonstration

Use R!

Dirk Eddelbuettel

Seamless R
and C++
Integration
with Rcpp

Finlay Scott
Maritime Affairs Unit - IPSC
European Commission
Joint Research Center

Why mix R and C++

- Speed (run-time, *not* time to code)
- Take advantage of existing C++ libraries

The old way

The core of R is a program written in C

- Use *.Call()* function and **SEXP** types
- Compile DLL / SO or a package
- Dealing with **SEXP** types is fiddly
- Very limited documentation (the source code)
- Manual garbage collection
- Fails badly

.Call() example

Rcpp

- Started life in 2005
- Used in >100 packages on CRAN - RcppArmadillo
- **RObject** - thin wrapper around **SEXP**
- Basic R types directly translated to C++ types (including S4 classes)
- Vectors and Matrices use templating
- 'sugar' functions
- No manual garbage collection
- Fails nicely

More Rcpp

- **Attributes** - expose functions if ip / op have *as()* / *wrap()*
- Can write your own *as()* / *wrap()* for your own classes - very powerful
- Easy package building
- *sourceCpp()* compile C++ code from inside R session
- RInside - embed R in C++
- Expose classes (modules)

Example Rcpp

- Attributes
- Custom as / wrap

Autodif + R: example with RcppAutodif

<https://github.com/drfinlayscott/RcppAutodif>

Two examples:

- Calculate gradients
- Use the minimiser

ADMB + Rcpp?

- Modify existing C++ code (from tpl) by hand so that it integrates with R
- Include R integration in *.tpl file?
- Write C++ directly using AD, use Rcpp to link directly to R

/note{ Could write simple C++ AD function in R-land, source()
it, then get gradients / minimise? }