Deploying and extending the Emulator

Andrew Parnell and Philip Cardiff
andrew.parnell@mu.ie

https://github.com/andrewcparnell/intro_emulators

Introduction

- The final step in our process is to use the emulator
- This might seem simple, but there are many choices and coding difficulties
- Much depends on how the emulator will be used
- Lastly we will talk about how to extend the emulator to more advanced problems

How will you use your emulator?

- If the emulator will just will be used by you only then it can stay as R/Python code (but please share it somewhere to make your work reproducible)
- If the emulator is required to be used by others then you should try to deploy it in a user-friendly manner
- There are two R packages that might be helpful here:
  - The Shiny package that builds interactive dashboards using standard R commands
  - The plumber package which allows R functions to be deployed as APIs
- The former allows you to package up your emulator in a neat visual interface with plots; the latter will allow others to run your emulator and incorporate it into their own code

Building robust predict functions

- When deploying your emulator make sure it is robust to mis-specification of the input values and does not perform unrealistic extrapolation
- Use base R functions such as `stopifnot` or the `checkmate` package to ensure that all inputs are validated

```
f <- function(x1, x2) {
  stopifnot(x1 >= 0 & x1 <= 1) # Range of x1 is (0,1)
  stopifnot(x2 >= 0 & x2 <= 1) # Same for x2
  return(10 * sin(pi * x1 * x2)) # Simulator function
}
```

- (Obvious alert) Comment Your Code and follow a style guide

Extending emulators

Some real world simulators have:

- ... multiple outputs. Some of the packages listed back in in part 1 can deal with multiple outputs
- ... time series outputs. You can use the same multiple output methods but you might need a bespoke solution if there's a strong time series dependence
- ... stochastic behaviour. In which case you might use a Gaussian Process Regression approach or another machine learning model
- ... rapid turn-around time. In which case you might repeatedly run the simulator and optimise using adaptive design techniques

If your simulator has any of these properties then please contact me to discuss further

Course summary

- Remember the steps: design, build, check, (calibrate), deploy
- Designing the emulator involves building an experimental design using a Latin hypercube or similar
- Building the emulator requires fitting a Gaussian Process
- Check the emulator using marginal plots and cross validation
- Deploy the emulator in a manner that maximises its use for the purpose it was created

Enjoy building your emulators!