Initial report on the application of derivative methods developed on toy models to C-GOLDSTEIN

Gemma Stephenson

March 17, 2009

Abstract

It is possible to emulate the derivatives of model outputs. Results from experiments on toy models are encouraging and here we extend the investigation by attempting to emulate the derivatives of the intermediate complexity climate model, C-GOLDSTEIN.

Contents

1 Introduction

1.1 Purpose of this document

Running a model's adjoint to obtain derivatives, while more efficient and accurate than other methods, is a computationally expensive task. In addition, the effort taken initially to write the adjoint is considerable and the task, time consuming. Therefore, if we can emulate the derivatives of a simulator this would decrease the demand for writing and running adjoints. The aim of this study is to perform an initial investigation into the suitability of an emulator to predict derivatives of the C-GOLDSTEIN model.

1.2 Design of experiment

The C-GOLDSTEIN model produces many outputs and we choose global mean air temperature to study here. The adjoint model has been adapted to generate the derivatives of global mean air temperature with respect to 12 of the input parameters and was run to provide validation data.

We choose to vary just one of the 12 input parameters, scf, which is the wind stress scale. While thought to have some effect on global mean air temperature this input parameter is chosen primarily due to the performance of the adjoint. All finite difference (FD) experiments undertaken thus far in an attempt to validate the adjoint have shown consistently good agreement between the adjoint and FD for this parameter. In an initial investigation, therefore this parameter is thought to be suitable.

All emulators in these experiments are built with function output alone and with a linear form for the prior mean. We choose the following covariance function:

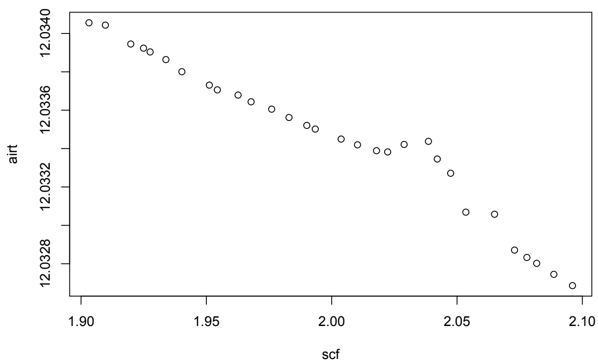
$$\mathrm{c}(\mathbf{x}, \mathbf{x}') = \exp\left\{-b_i(x_i - x_i')^2\right\}.$$

For details of the intermediate compexity climate model, C-GOLDSTEIN and the adjoint see MUCM internal reports 3.2b.3 and 3.2b.5. For details of the methods applied here and results from toy model investigations, see MUCM internal report 3.2b.2.

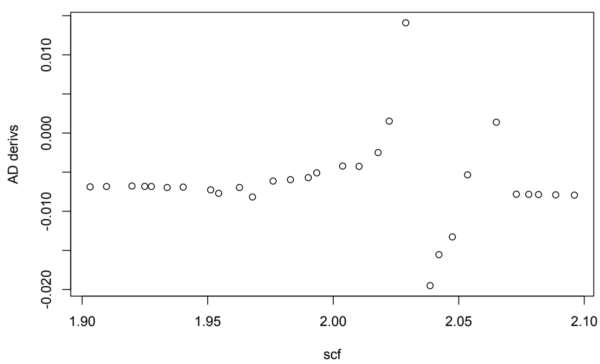
2 Emulation of C-GOLDSTEIN derivatives

2.1 Initial runs

We choose to run the C-GOLDSTEIN adjoint at 10 points to provide training data for the emulator and a further 20 points to provide validation data. We only include the function output in the training data so this provides 30 validation derivatives. Figures 1a and 1b show the how the global mean air temperature and the partial derivatives of the temperature with respect to wind stress scale vary for all 30 runs.



(a) Global mean air temperature



(b) Partial derivatives

Figure 1: C-GOLDSTEIN adjoint output

We begin by emulating the air temperature at the validation points. Figure 2 shows the performance of the emulator (in red) and the the points included in the training data are in bold. We can see that the emulator performs well across most of the design space.

We proceed by emulating the derivatives at all 30 points and Figure 3 compares the emulator with the adjoint output. The emulator performs quite well up to approximately scf = 2.02 but after this value there is much conflict between the emulator and C-GOLDSTEIN adjoint.

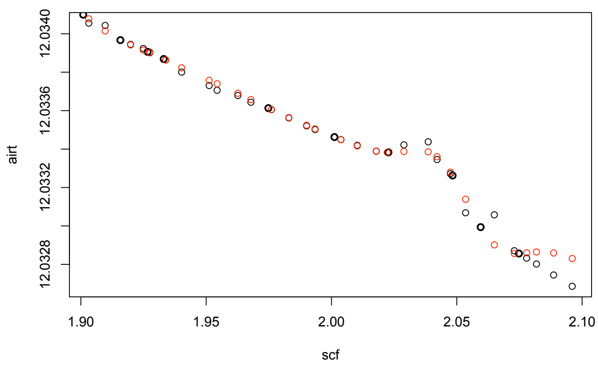


Figure 2: Emulating function output

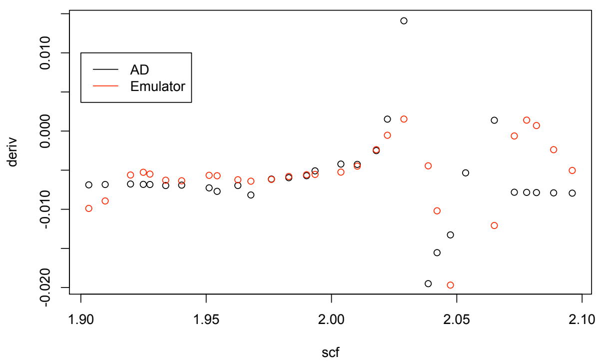
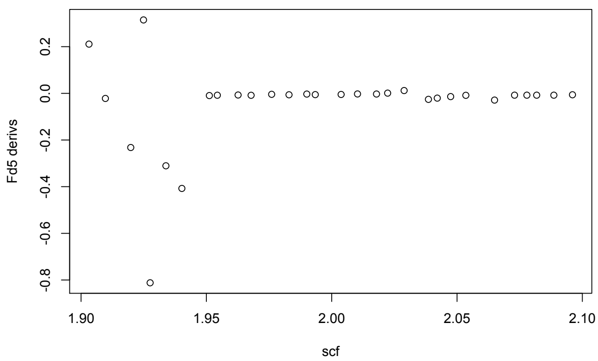


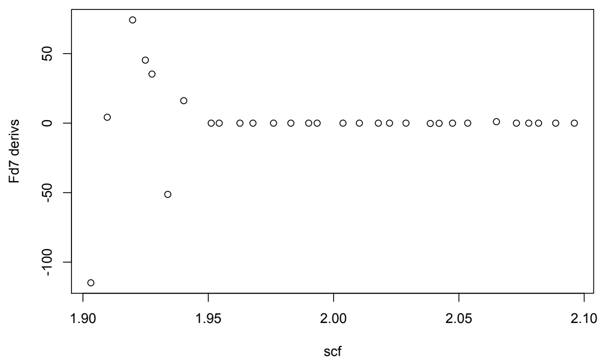
Figure 3: Emulating derivatives

2.2 Further validation of adjoint

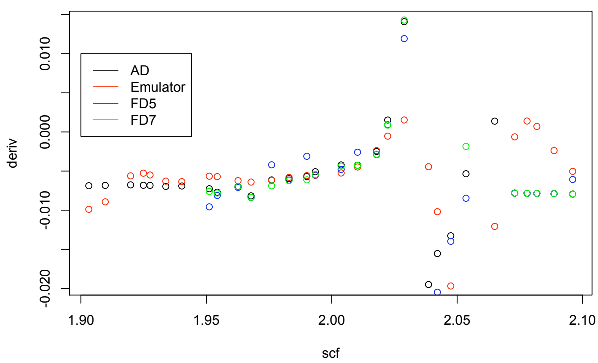
For completeness FD experiments are conducted with $\epsilon = 1e-5$, and $1e-7$. In addition to this the NAG fortran library routine, D04AAF, is called to provide a variation to the aforementioned FD experiments. The routine required 21 runs of the function and returns an estimate of the derivatives and a corresponding estimate of the error in that derivative. Here, we used a step length of $1e-5$. For more detail about the routine see Numerical Algorithms Group (2006). Figure 4 show comparisons of all the methods of obtaining derivatives used here. Figures 4a and 4b both show the FD method is unreliable between



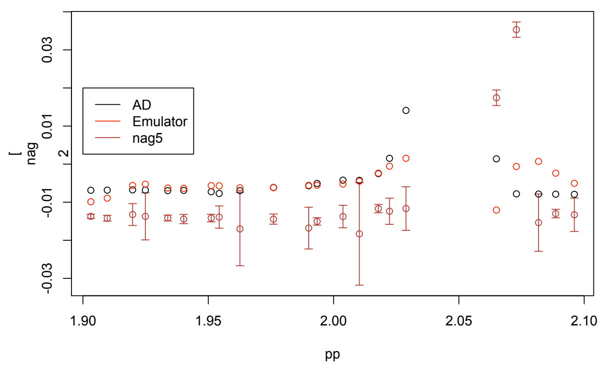
(a) FD derivatives with $\epsilon = 1e-5$



(b) FD derivatives with $\epsilon = 1e-7$



(c) Comparison of adjoint and emulator with FD



(d) Comparison of adjoint and emulator with NAG routine ± error

Figure 4: Comparison of derivatives

scf = 1.9 and 1.95, the output for the remaining input space is misleading due to the scale of the y-axis. Figure 4c shows how FD compares to the adjoint and the emulator and here we can see that the FD method generally agrees with the adjoint for scf $\in \{1.95, 2.10\}$. Figure 4d shows that the NAG estimates of the derivatives are consistently lower than the adjoint and emulator. Even allowing for the error estimate the range in which the derivatives falls is only occasionally in agreement with the adjoint. In summary though, Figure 4 shows the adjoint can be trusted to provide reliable validation data for this parameter, across this input space.

2.3 Further runs and narrowing the input space

As seen in Section 2.1 the derivative emulator performs badly for approximately scf $\in \{2.02, 2.10\}$. We attempt to improve on this by performing an additional 50 runs of C-GOLDSTEIN across the whole input space, in order to gain a better understanding of the function and provide more validation data for the standard emulator. We then restrict the training data to 11 points between 2.02 and 2.10 and build a second emulator for function output and derivatives. Figure 5 compares the air temperature of the emulator with the validation data. The black line is C-GOLDSTEIN output and the red line is the posterior mean evaluated at the validation points. The output is shown by a solid line but consists of 80 points. With this training data we now see the emulator is performing very well across this narrow input region. No further runs of the adjoint are performed

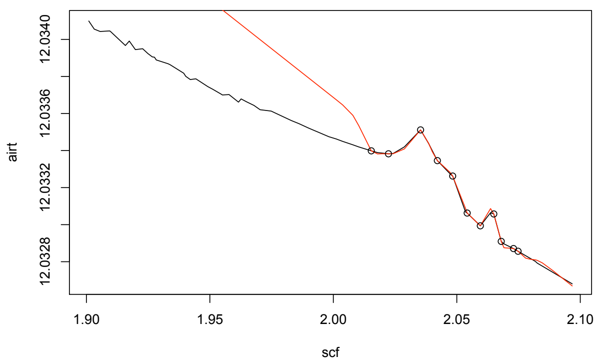
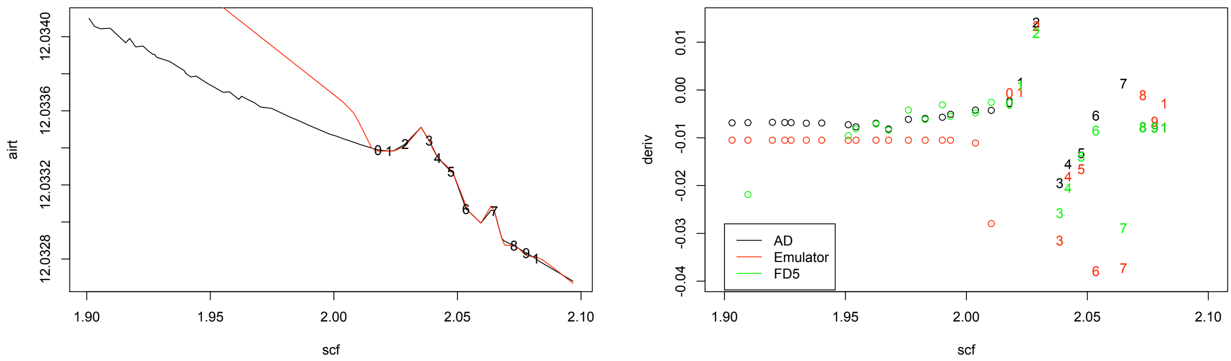


Figure 5: Emulating air temperature in the narrowed input space

due to the computational expense. Figure 6a shows where the validation data for the derivatives is located in the input space and Figure 6b shows the performance of the emulator here. The derivatives up to the point scf = 2.01 are included for completeness. Figure 6b shows much less conflict between the emulator and the adjoint than before (in



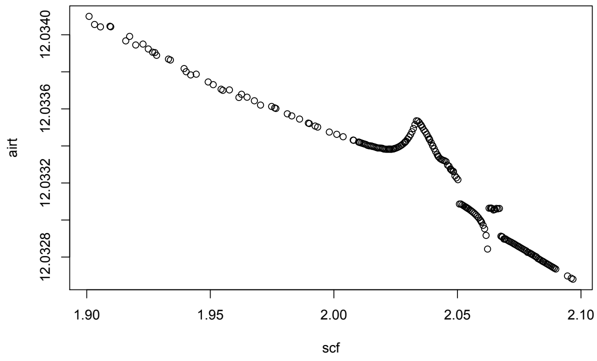
(a) Points in the narrowed space where the derivatives are predicted

(b) Emulating the derivatives

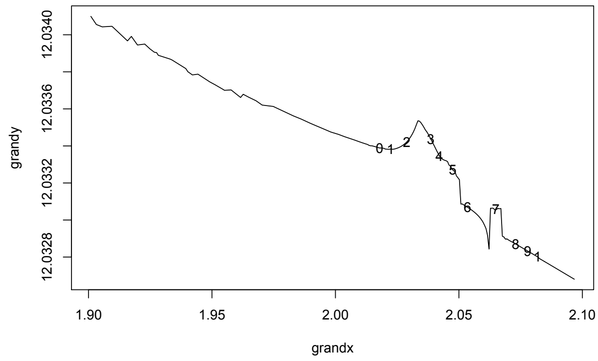
Figure 6: Emulator in the narrowed input space

Section 2.1), the emulator even matches the adjoint at the high peak at derivative point 2. The emulator however, still performs very badly at derivative points 6 and 7. There is not a clear explanation for this as Figure 6a shows we can emulate the function output well at these points.

We now choose to run C-GOLDSTEIN intensively in this narrowed region. The resulting data set has a further 134 points in the narrowed region and 214 points altogether. The augmented data set, shown in Figure 7a, is quite alarming as we had previously believed C-GOLDSTEIN to be a smooth, continuous function. Figure 7b shows the points where the derivative validation data is located and it is now clear why in Figure 6b, the emulator behaved poorly at point 7 and to a lesser extent at point 6.



(a) C-GOLDSTEIN air temperature



(b) C-GOLDSTEIN air temperature with points where derivative validation data is located

Figure 7: Augmented data set

3 Conclusions

An initial investigation into emulating the partial derivatives of C-GOLDSTEIN has had mixed success. Firstly, perhaps it should be noted that the function of the derivatives, as shown in Figure 1b, is not a function we would confidently claim to be able to accurately emulate. Given enough training data, however, we can emulate the derivatives of this model quite well, even in a rough patch. If so many simulator runs are required though, it is unlikely that emulation will be an efficient solution to an adjoint model. Of course, an adjoint to the required model might not exist, so the question becomes about whether emulation is more efficient than the many runs a finite differences experiment would entail. Worthy of consideration also is the accuracy of the FD method; Figures 4a and 4b show how here the FD method produced disastrous results at the beginning of the input region, where the emulator performed well.

In summary, this investigation has shown that the C-GOLDSTEIN model does not behave as we would expect. Consultation is underway with one of the adjoint authors, Neil Edwards, on this matter. Further work on validating the derivative emulators by adopting the diagnostics of Bastos and O'Hagan (2008) is also in progress.

References

Bastos, L. S. and O'Hagan, A. (2008). Diagnostics for gaussian process emulators. *Submitted to Technometrics*.

Numerical Algorithms Group (2006). *NAG Fortran Library Manual Mark 21*. The Numerical Algorithms Group Ltd, Oxford, UK. 2006.