Response to the reviews and short comments for the manuscript “sympl (v. 0.3.2) and climt (v. 0.11.0) – Towards a flexible framework for building model hierarchies in Python”

We thank both reviewers for their fairly extensive and insightful comments. The main issues raised by both reviewers has lead to an extensive re-organisation of the manuscript. We hope that the new version will be better organised, more readable and engaging.

1 Common issues raised by reviewers

1.1 Lack of examples

We have addressed this issue by replacing the current section 2, which presents an abstract discussion about modelling frameworks with a new section which presents a set of examples which try to accomplish the following –

- Show the user how to build physically useful models
- Show how easy it is to change a single column model to a GCM
- Show the steps required to write a `sympl` component

Spencer mentioned in his review that it might be useful to build a model using only `sympl` without `climt`. However, given that `sympl` does not aim to provide scientific components, we felt that such an example would be too simplistic to be of real interest to readers. We prefer instead to present the layout of a `sympl` component and emphasise how easy it is to write such a component.

1.2 Mixing conceptual issues with API description

The new organisation first presents a quick user guide to build models. We then discuss the conceptual issues (what is currently most of Section 2) in a separate section. This allows to the reader to first see how easily scientifically useful models can be built, and then read about the reasons why such ease is possible with `sympl` and `climt`.

To address the issue that the conceptual and manual-like parts of the paper do not complement each other (raised by the anonymous reviewer), we have rewritten the conceptual parts of the paper to refer frequently to the examples presented in the new Section 2. We hope this will help make the design decisions clearer since every such description of a design decision will now refer to actual line/lines of code.

1.3 Current scope of `climt`

As it currently stands, `climt` is simply a collection of `sympl` compatible components (a toolkit). The framework-like features of earlier versions of climt, encapsulated in the `Federation` object, have not yet been implemented in the current version. This has also been noted in the previous Figure 1 of the manuscript. We are still mulling over the design of `Federation` given the new structure of `climt`, and thank the anonymous reviewer for pointing us to relevant references.

1.4 Figure 1 and 4 unclear

1.4.1 Figure 1

- We have fixed the balloon corresponding to Computing resource configuration to reflect the terminology in Section 3.

- We have renamed "Behavioural Configuration" to "Interfacial Configuration" to emphasise the fact that we only modify quantities at the interface (input/output) of a component. This change has been made in the text as well.

- We have lengthened the curly bracket to an appropriate size.

- From the very outset, we had tried to reduce the number of intersections of the lines for exactly the same reasons that Spencer mentioned. This was the best we could do.

- The caption has been modified to highlight the fact that the arrows correspond to control or responsibility, i.e, what file or module is typically responsible for a particular kind of configuration.

1.4.2 Figure 4

- We have redrawn the figure to make it clearer

- We have expanded the captions substantially explaining what each panel means.

1.5 Code availability

The model scripts to produce the results in the paper will be added as supplementary material during this revision. We have also generated DOIs for both packages, which will be incorporated into the manuscript. These changes also address similar issues raised by Lutz Gross in his short comment.

2 Issues raised by Anonymous Reviewer

2.1 Link between features/design decisions and objective (traversing model hierarchies) unclear

We thank the reviewer for pointing this out. We intend on addressing this issue by presenting two examples, a radiative-convective equilibrium model and a moist GCM in Section 2 and referring back to these examples frequently in subsequent sections where we describe features and design decisions of both packages.

We hope this strategy will achieve two things – illustrate how easy it is to move from a single column model to a 3D model, and what design decisions make it easy to do so.

3 Issues raised by Spencer Hill

We thank Spencer for his detailed and constructive comments. His suggestions for restructuring the paper were definitely very useful in making this manuscript as accessible as `sympl` and `climt` hope to be!

3.1 Platform Independence

Though we have not tested our packages on AWS explicitly, we do not forsee any obvious difficulties, since wheels are available for both packages. In fact, we have successfully been able to use `climt` on the web-based Google Colaboratory notebooks. We will make this clear in the revised manuscript.

3.2 Other comments

Most of the line-by-line comments have been addressed either by making appropriate changes or deleting sentences which could potentially confuse the reader.

4 Changes made to `sympl` and `climt` since the time of submission

Since the time manuscript was submitted, we have made some changes to both packages, which are reflected in the manuscript. The major changes are:

- Renaming the basic classes in `sympl`. Feedback from some of our users led to renaming the basic classes to clarify what they actually are. Also, we realised that `TimeStepper` was simply a kind of `Implicit`. Thus, the following changes were made:
  - `Prognostic` → `TendencyComponent`
  - `Diagnostic` → `DiagnosticComponent`
  - `Implicit` → `Stepper`
  - `TimeStepper` → `TendencyStepper`
  - `ImplicitPrognostic` → `ImplicitTendencyComponent`
- `sympl` classes have been restructured to make them more than just a basic description. In particular, the state dictionary is no longer passed onto the component code directly. Instead, a set of consistency checks are performed, the numpy arrays are extracted with the correct units, and these raw numpy arrays are passed onto the scientific code. This eliminates the requirement for developers to write boilerplate code for consistency checks and array extraction. This change has also been made in the corresponding section in the manuscript.
- `climt` has also been refactored to reflect these changes in `sympl`, resulting in a much cleaner API. However, there is very little if any change to how the user interacts with `climt`.
- These changes have resulted in new releases of both packages. Hence, the version numbers in the title will also be changed.