UKCA–GungHo! Coupling: Possible Strategies

Luke Abraham & Fiona O'Connor 2015-04-07

1. **UKCA code is distributed throughout the appropriate physics routines, e.g. wet removal is done in convection, emissions are done in the boundary-layer scheme etc.**

Pros:

- No need to pass large numbers of variables around the model.
- UKCA processes occur concurrently with the relevant physical process.
- Testing and code management is supported by the LFRic team.

Cons:

- UKCA code is spread between many different sections – ownership of code is not the main UKCA developers.
- Difficult to maintain, easy to break.

2. **UKCA code is self contained within the LFRic source-tree, similar to its location within the MetUM code (although not necessarily at the end of the timestep).**

Pros:

- Familiar set-up and similar to current structure.
- Testing and code management is supported by the LFRic team.
- Coupling to LFRic straight-forward (i.e. modules, subroutine calls etc.).

Cons:

- Tied to LFRic release cycles.
- Load balancing may be harder to achieve due to existing PE decomposition.
- May be harder to run with a different resolution to LFRic, if this is required.

3. **UKCA is held within a separate code repository, e.g. similar to JULES.**

Pros:

- Allows for easier creation of stand-alone model, e.g. box model, trajectory following, CTM etc.
- Possibility to re-merge with TOMCAT/SLIMCAT/GLOMAP model.
- Testing and code management will be supported by the LFRic team if we continue to adhere to UM working practices, UM deadlines etc.. as we

currently do. Allows for possibility of more flexible release cycle (only if UKCA code repository is managed solely by UKCA).
- Allows for the possibility of running the same version of LFRic with different UKCA versions.
- Allows for different PE decomposition to LFRic (depending on how the coupling is done).
- Allows for different resolution to LFRic (depending on how the coupling is done).

Cons:

- Testing, code management, and lodging would need to be supported by the UKCA team, if we wanted certain freedoms from the UM such as our own code freeze deadlines, our own working practices etc.. Extra resources would be needed, but may not be available.
- Extra work and extra resources will be required to maintain capability of running in multiple models.
- How would the diagnostics be output? As an example, JULES uses STASH when running coupled to the UM but uses its own output routines when running in standalone mode.
- How would the model restart system work? Would this be managed through LFRic, or be UKCA specific? As an example, JULES adds prognostics to the atmosphere dump when running coupled to the UM but writes its own restart dump when running standalone.
- Coupling would not be straight-forward. Will it be a separate binary or built as part of LFRic? It is likely that a large number of 3D fields will need to be passed. Would OASIS cope?
- Would tracer transport be done in UKCA? How about processes like convection and boundary-layer mixing etc.?

Notes:

- It may be possible to use this UM+UKCA only code that is being developed for UKESM1 here – this would also cover the output (STASH) and restart (dumps) system.
- Just because UKCA is in a separate repository it doesn't mean that it can't be built in the same executable as LFRic.
- It also doesn't mean that we must use a coupler such as OASIS, we could couple with subroutine calls or modules etc. However, using a coupler would allow for more flexibility in terms of resolution and PE configuration.