

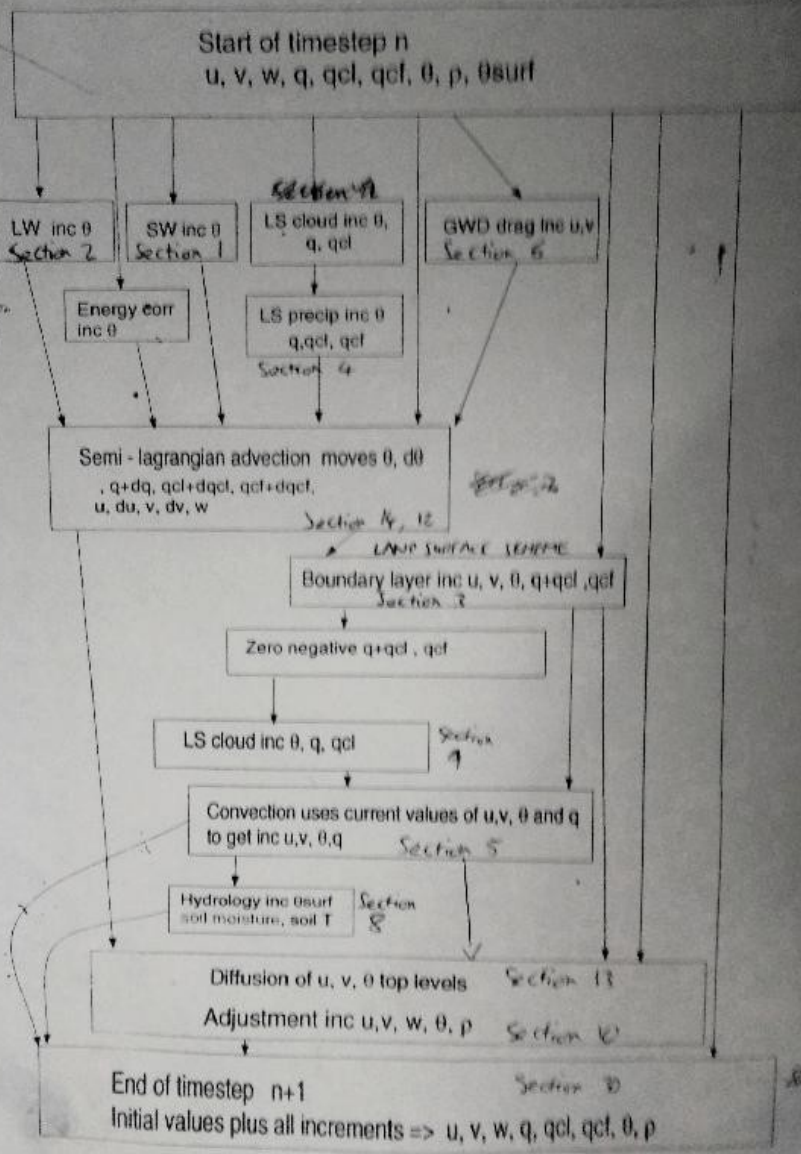
What does the algorithm layer look like?

Or, it's more than dynamics

Simon Wilson NCAS-CMS

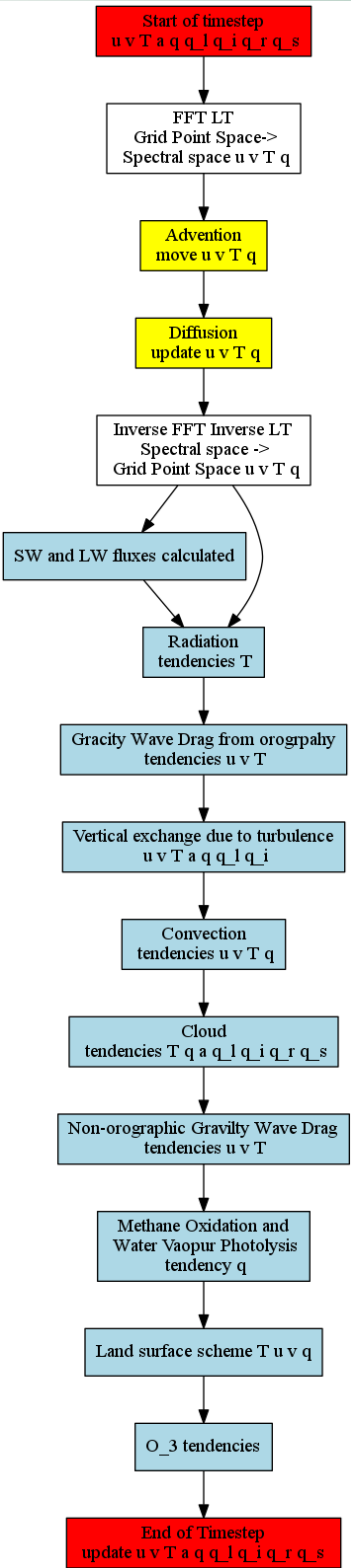
- What does a model timestep look like?
- Part of a study as part of IS-ENES2 WP3 on developing convergent model codes, investigated relationships between model components and prognostics, focusing on dependencies and flow.
- A comparison of the atmosphere timestep and code structure of three different atmosphere models
 - Unified Model from the Met Office in a HadGAM3 configuration
 - HiRAM from GFDL
 - OpenIFS from ECMWF

- Fundamentally most (all?) models consist of a number of prognostics (state variables) such as wind, temperature, moisture, pressure/density,... moved and/or altered by a number of model components split into fluid motions and thermodynamics (dynamics), and physical parametrisations (physics).
- Need a way to represent the structure and relationships.
- Initially something like this:

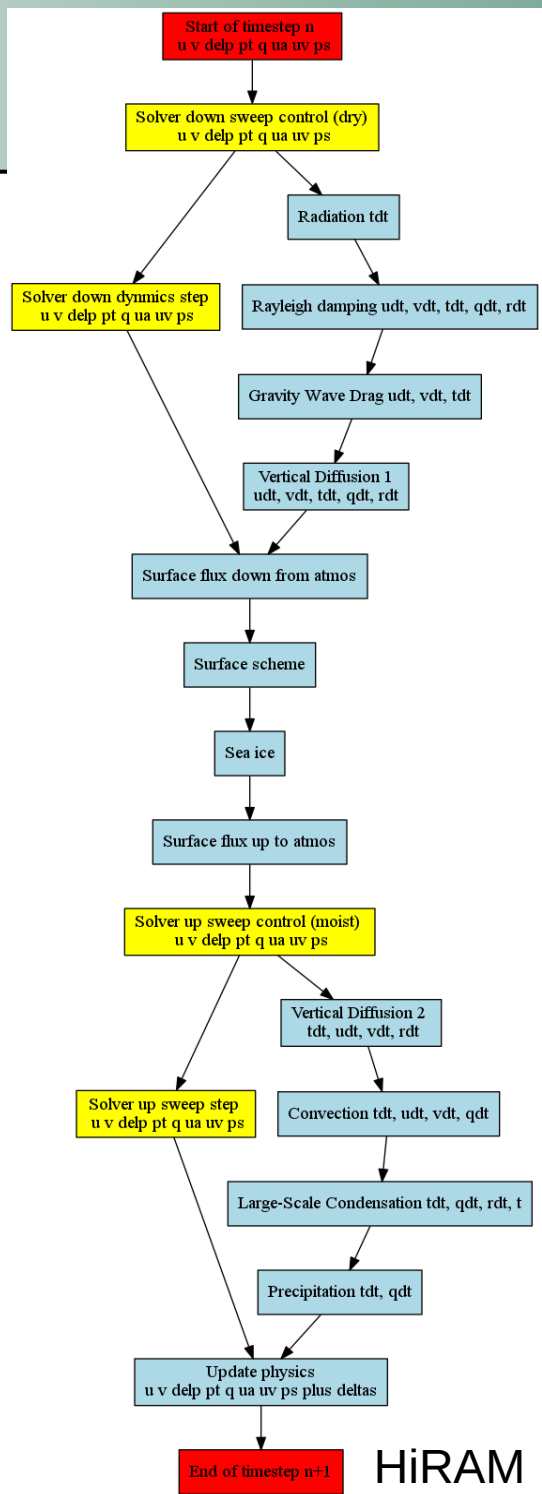


My Office Wall.
 The anatomy of
 a timestep in the
 UM.
 Original version
 long lost.

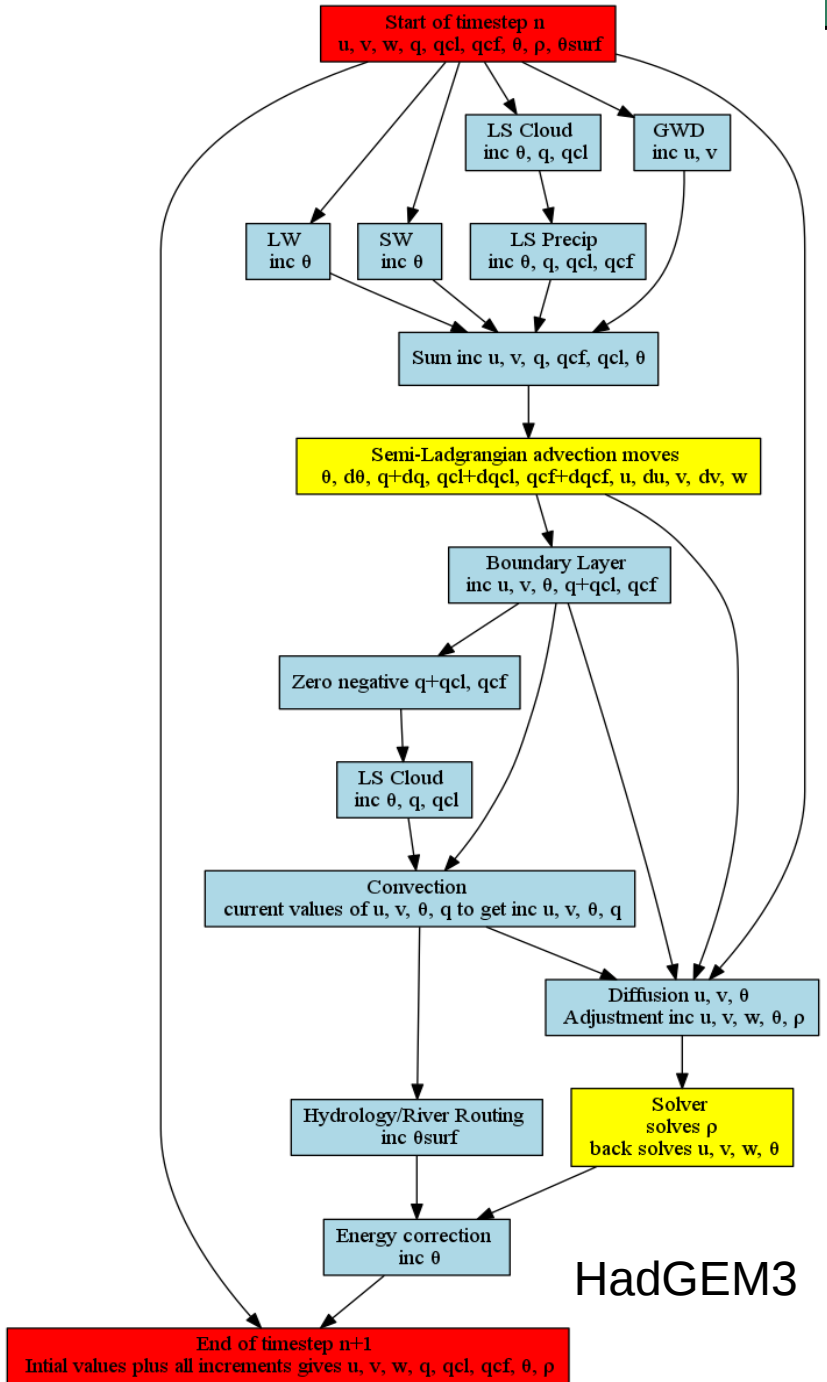
- By inspection, from model documentation papers (which can number in the 10s), code comments and documentation, code analysis, speaking to model gurus (and some guesswork), generated these...



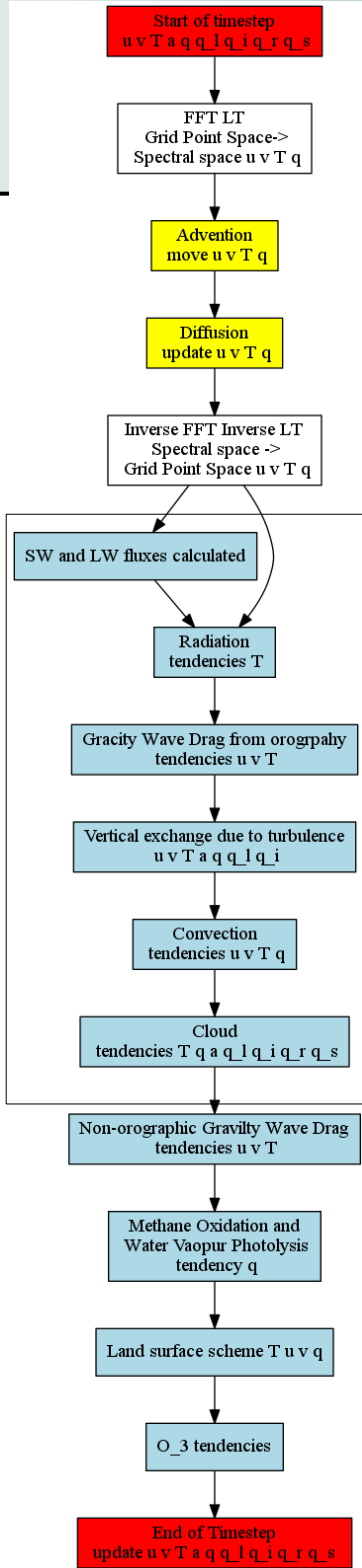
IFS



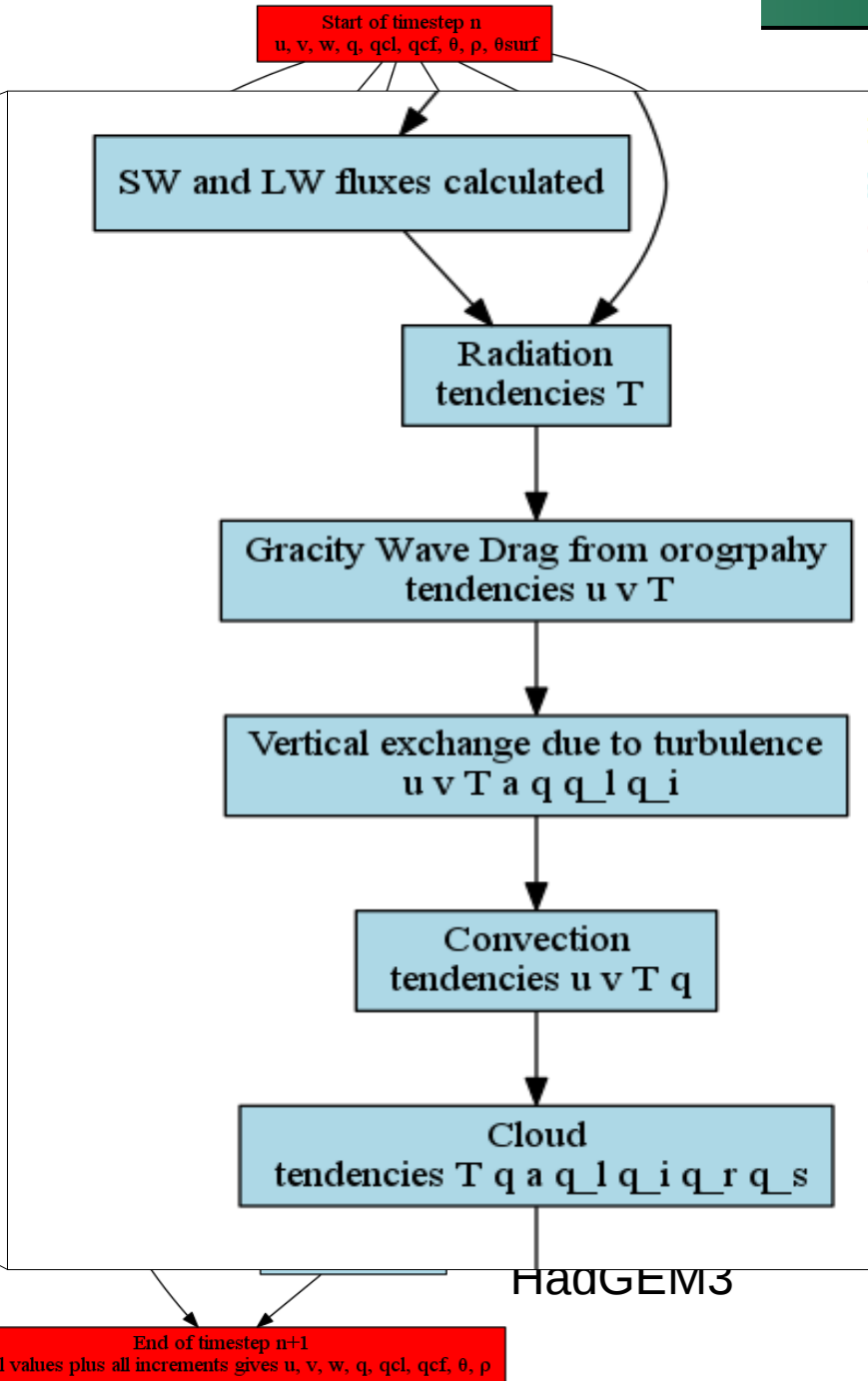
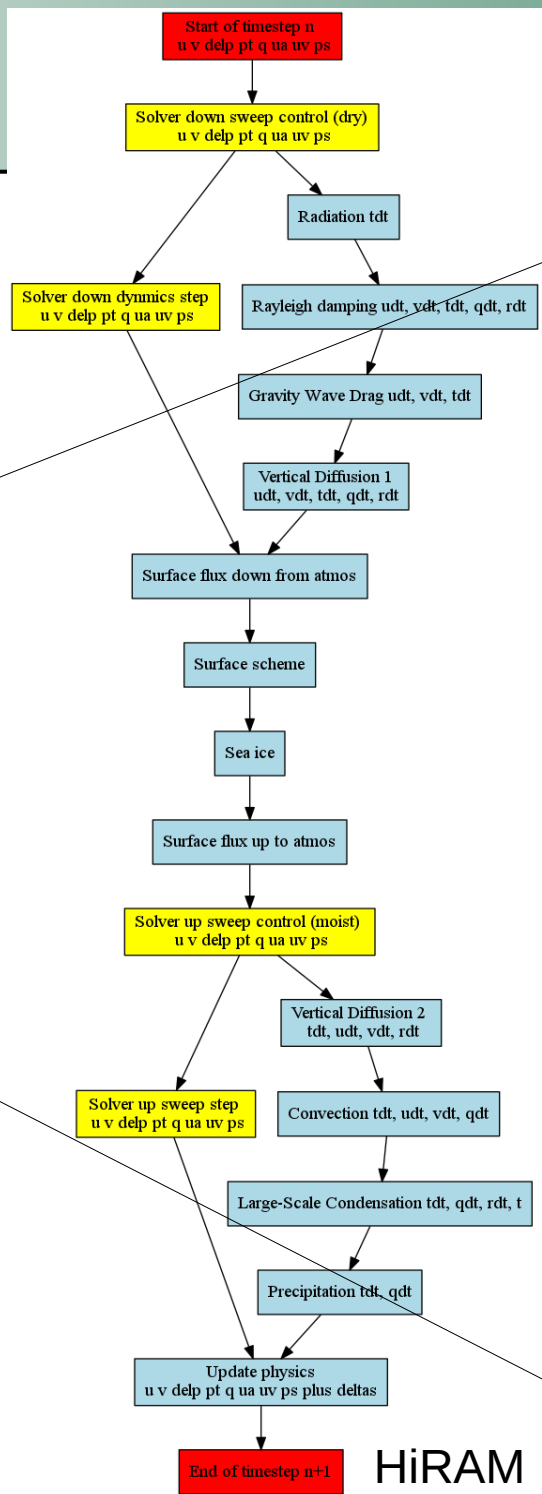
HiRAM



HadGEM3



IFS



ynamics
ics
/End
ice

- Not just the dynamical core, various physics APIs coupled with prognostics variables within models.
- Optimisation possibilities available through model structure and parallelisation of the the physics.
- Syntax required to describe these relationships and dependencies of model components.

- For model comparison and convergence, consistent documentation for model structure and variables needed.
- Eg: Model variables. Many types and descriptions of a single variable:
 - u , u' , advected u , du , moved u , u_T , u_{T-1} , u_{T+1} , u tendency, u increment, udt
- This is where the CIM (Common Information Model) is useful. It has “Variable” and “SoftwareComponent” constructs which allows model structure to be formalised, and can be used to automatically produce charts as previously shown.



Questions/Discussion

CIM construct descriptions

```
software.SoftwareComponent: (  
  'composition',  
  'connection_points',  
  'coupling_framework',  
  'dependencies',  
  'depends_on',  
  'grid',  
  'language',  
  'license',  
  'sub_components',  
  'description',  
  'development_history',  
  'documentation',  
  'long_name',  
  'name',  
  'release_date',  
  'repository',  
  'version',  
)
```

This corresponds to the prognostic variables

The input
SoftwareComponent
dependencies

```
software.Variable: (  
  'description',  
  'name',  
  'prognostic',  
)
```