



Met Office

# ENDGame, system progress...

## UM development

Glenn Greed Oct 2012



## Contents:

- UM and multiple 'dynamical cores'
- Global, LAMs and DA.
- Multiple execs or a single one?
- Testing space



# ENDGame

- Alternative dynamical core; [grid and prognostics]
- **GA5.0 based upon ENDGame**
  - target operational date post Easter 2013.

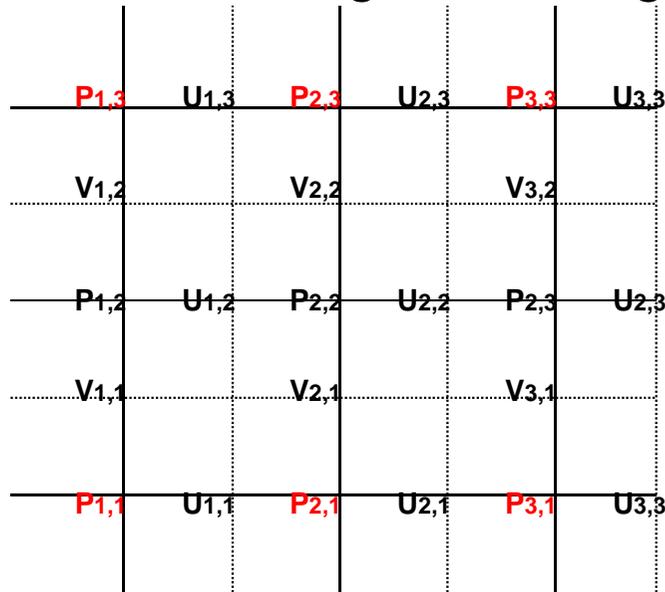
## UM impacts:

- duplication of code within the UM control structure
  - atm\_step, um\_model
- dynamical core is selected at compile time. (CPP flags; IF DEFs)
- Headache for maintenance and development.
- Testing overheads
- “divergence of code risk” as we develop/support 2 cores.



# UM step change or multiple support.

Comparison of grids: *need to support alternate array bounds when switching between grids.*



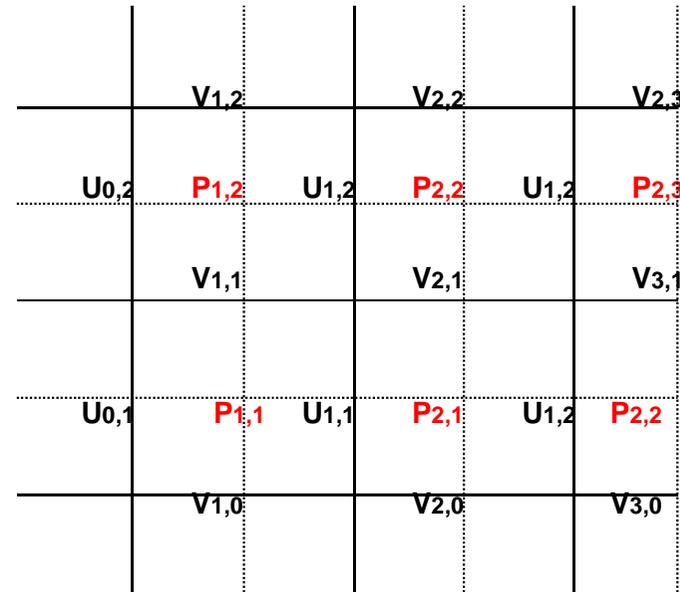
Simple 3x3 C grid u at the poles (N/S boundaries)

Scalar (row length, rows) = (3,3)

u wind (row\_length, rows) = (3,3)

v wind (row\_length, n\_rows) = (3,2)

**n\_rows=rows-1**



Simple 3x3 C grid v at the poles (N/S boundaries)

Scalar (row length, rows) = (3,2)

u wind (row\_length, rows) = (3,2)

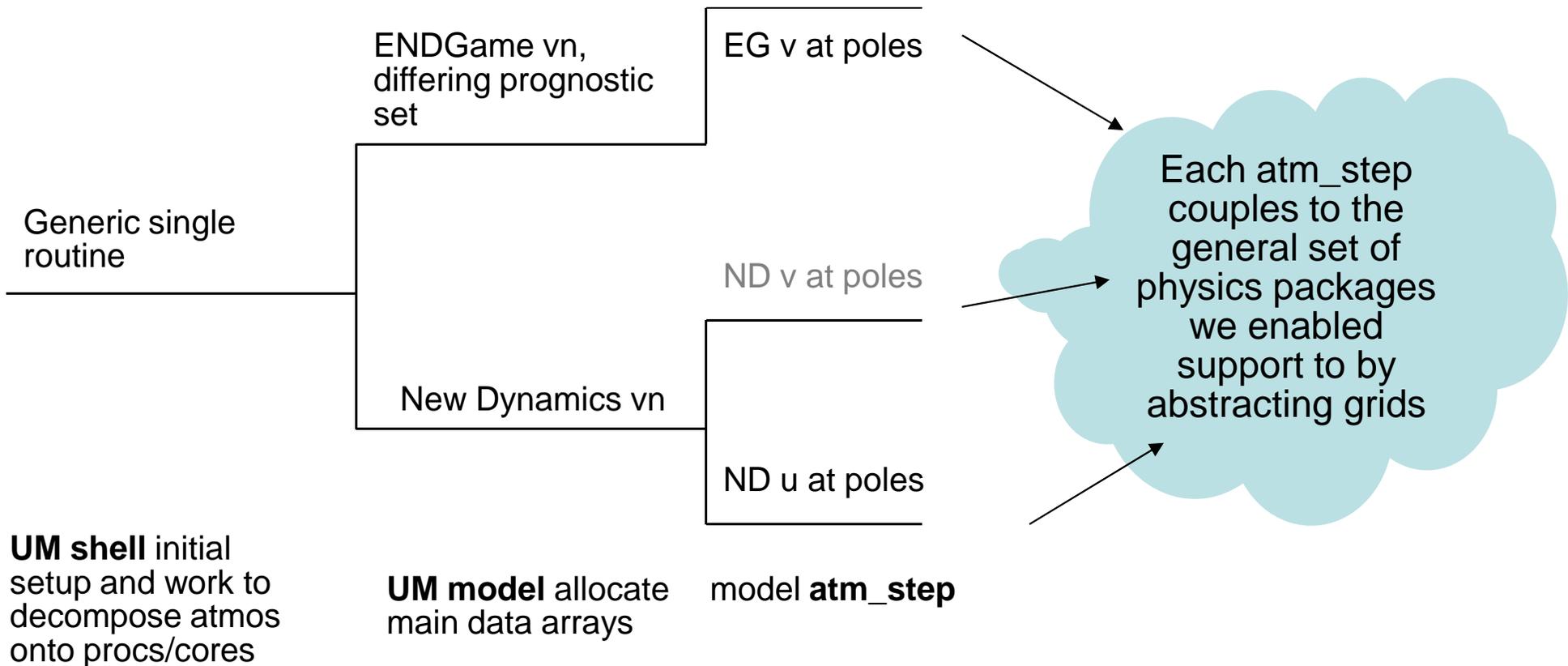
v wind (row\_length, n\_rows) = (3,3)

**n\_rows=rows+1**



# UM step change or multiple support.

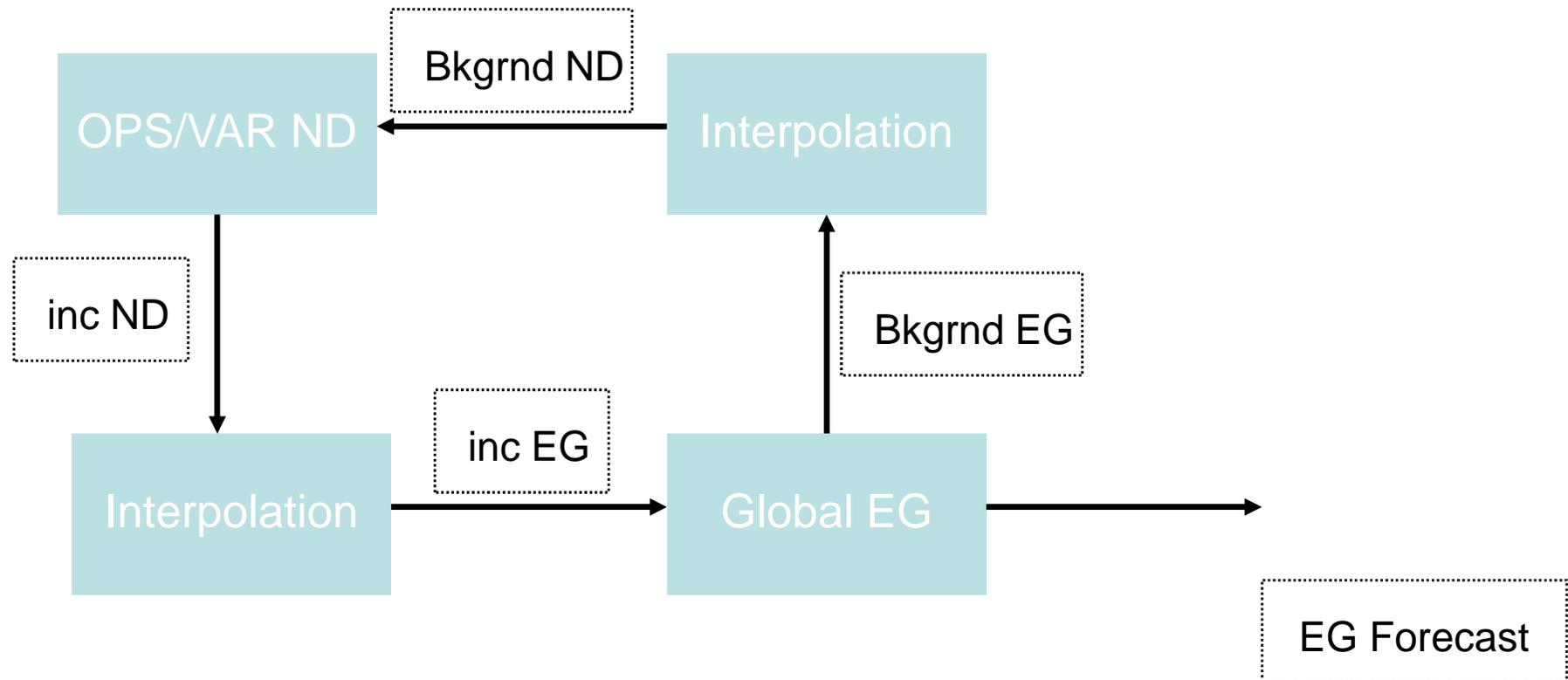
UM control routines define how the UM timesteps through the dynamics and physics coupling, selected by CPP flags.





# Adding DA support for EG to enable trials

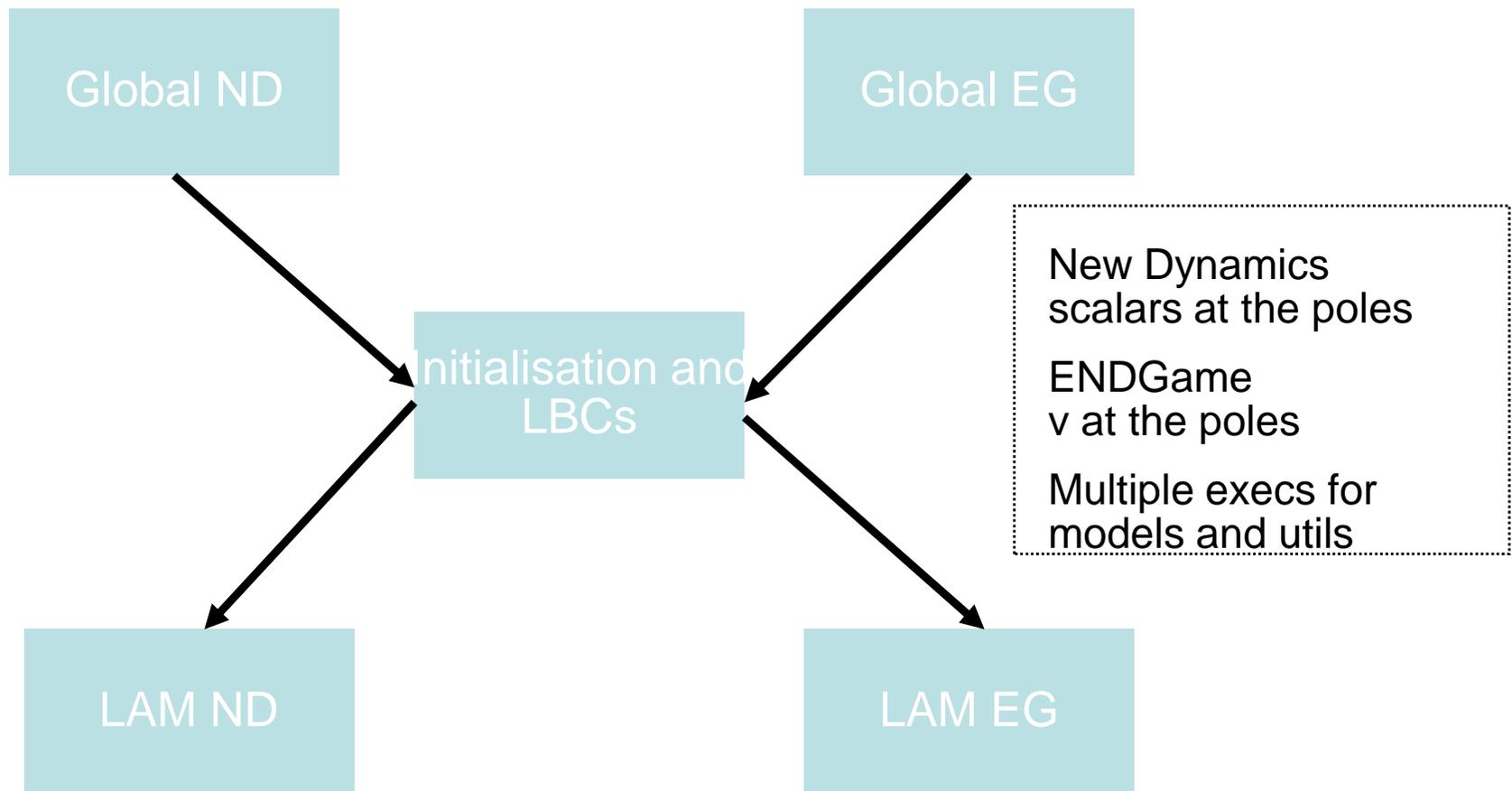
While the UM forecasts future is ENDGame with v at the poles, the data assimilation code shall continue to use the underlying New Dynamics grid with scalars at the poles.





## Adding LAM support for both ND and EG

It is unlikely that all UM configurations will move to EG on the same day. Global initially then followed by the LAMs. Our partners may choose to update on a different timescale.



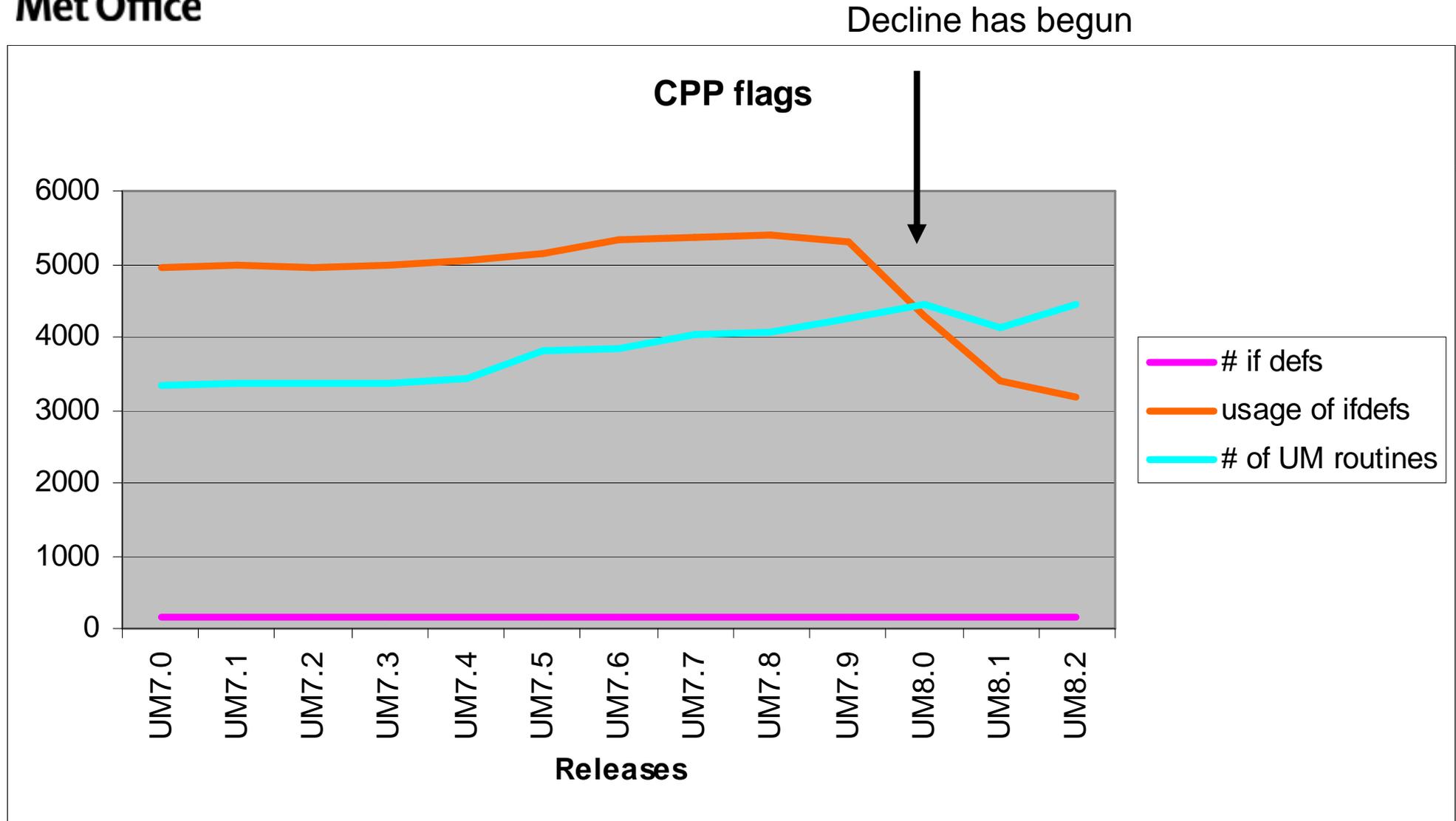


## Compile time flags:

- Can we move the UM away from its reliance upon IFDEFs?
- Any show stoppers?



# IF DEFs CPP flags



***Need to check dev do not reintroduce retired IFDEFs,  
UTF tests for this now.***



## Move away from IFDEF reliance (BoM)

Protoype inserting runtime switching to replace IFDEFs

```
IF ( .NOT. L_ENDGame ) THEN
    CALL u_model (...)
ELSE
    CALL u_model_EG (...)
END IF
```

Code changes throughout UM have been made to enable the running of both dynamical cores from a single executable and simple logical switching

	Compile	MaxMemory/node	Run t+9
Control ND	805secs	30.174GB	995secs
Test code ND	1660secs	30.183GB	998secs



## Single exec for all?

Protoype could be extended for the whole UM which currently uses IFDEFs CPP flags to select code section versions, ie physics params, dynamics and service routines.

In the past new development code sections have routinely been delivered protected by a compile time flag. Runtime flags were used for options within the sections.

Thus this could be a large change to push through the whole UM.

---	Advantages	Disadvantages
Compile flags	Protect dev code Protect platform specifics	Exec is section specific Maintain multiple prebuilds
Runtime flags	<b>Single platform/opt prebuild</b>	Long initial compile



# Testing update

- UM UTF update
- New test headaches



# UTF User Test Framework

- The UTF enables developers to easily perform a selection of tests on their code branch prior to submission for review. Comparing results with known good output for given configurations.
- Command line tool
  - `utf run fcm:um_br/dev/<....>`
  - Making the review process faster and more robust
- The UTF is extensible, in terms of both jobs and tests performed on them; not just UM.
- Used for nightly and weekly extended tests of the head of UM trunk.



# UTF test configurations

- Global forecast (IBM and Linux desktop)
  - Current oper science, Next agreed science package GA4
- HadGEM3 climate model
  - atmos only and coupled
- LAMs, UKV, UK4, MOGREPS-R
- Idealised, dynamics only, SCM, etc
- Now need to mirror the above tests with a similar set using the ENDGame dynamical core.



## UTF a test too far?

- Code Styling and standards test.
- All code changes now undergo a UTF test to check code style and standards automatically.
- This test has proven troublesome to implement with many false alarms when initially released.
- It has taken numerous iterations to get to a point where the code standards test is now stable and ready for wider release to the UM community.
- Aim has been for code developers to identify and resolve code standard issues prior to submitting code for review.



**Met Office**

- Thank you for listening
- Questions?