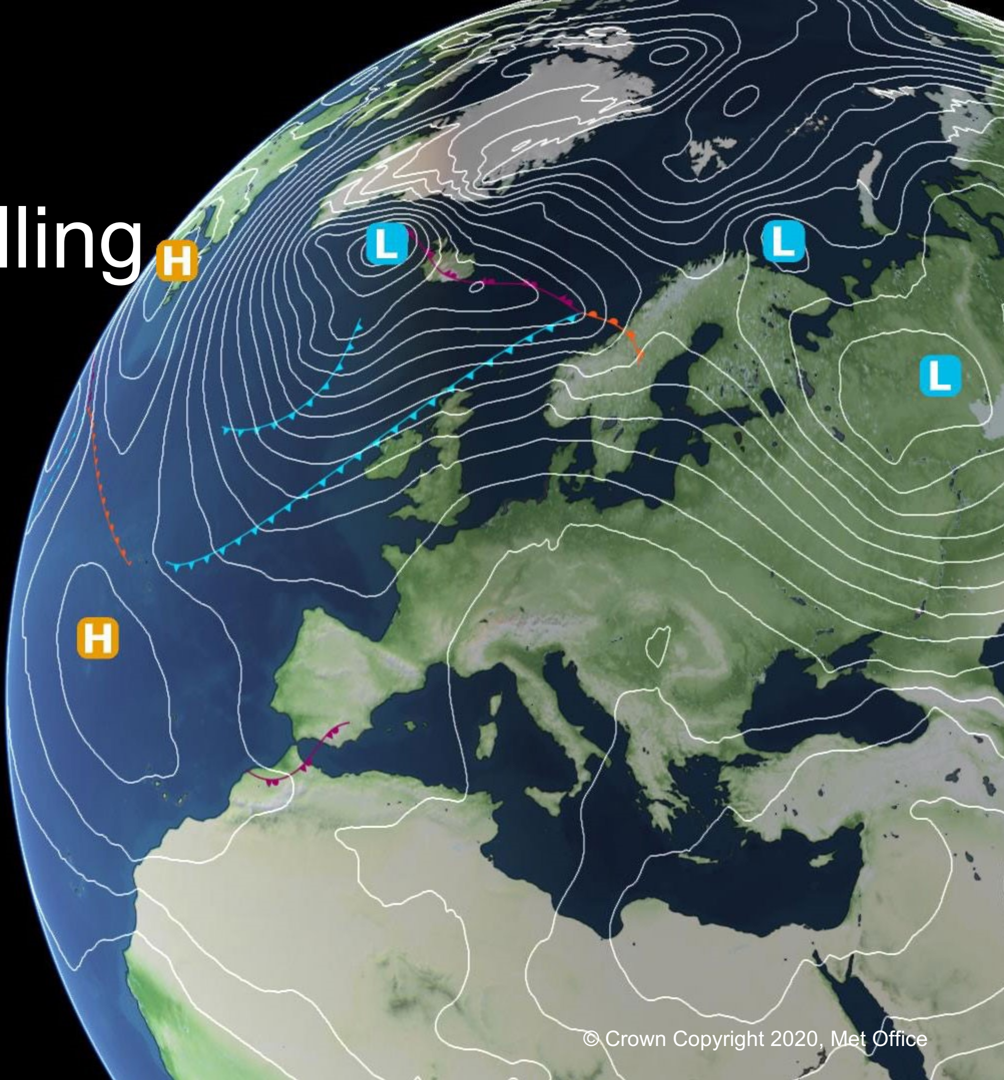
Met Office

Next generation modelling with LFRic

UKCA Chemistry and Aerosol Tutorials

7 December 2022

Ben Shipway

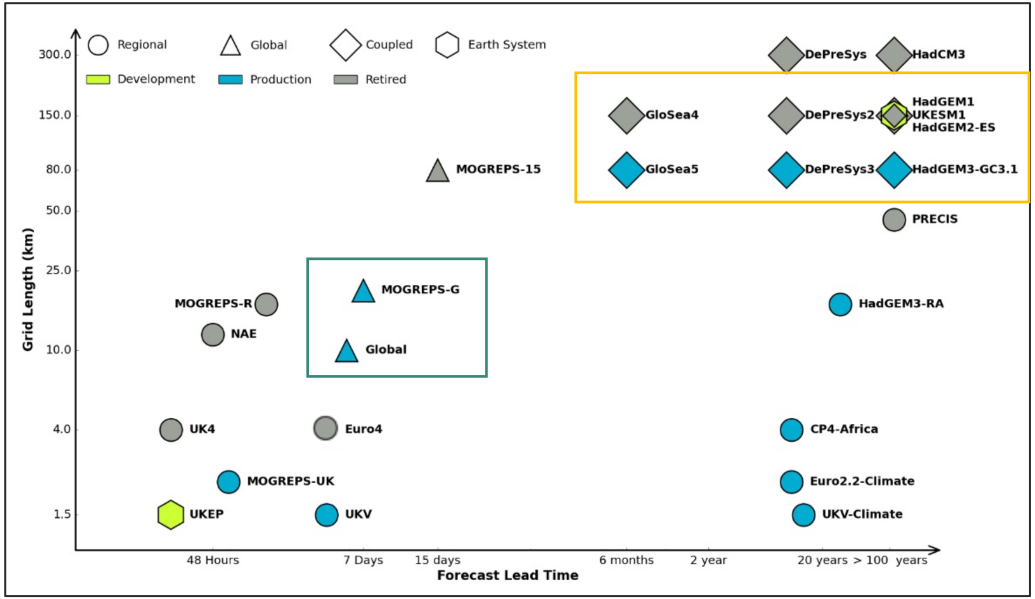


Outline

- Current Generation Modelling
  - UM atmosphere and ENDGame dynamical core
- Motivation
  - Why do we need a ‘Next Generation’
- Next Generation Modelling
  - LFRic atmosphere and GungHo dynamical core
  - Look a bit at the code
- Current status and plans
  - When with LFRic be ready?

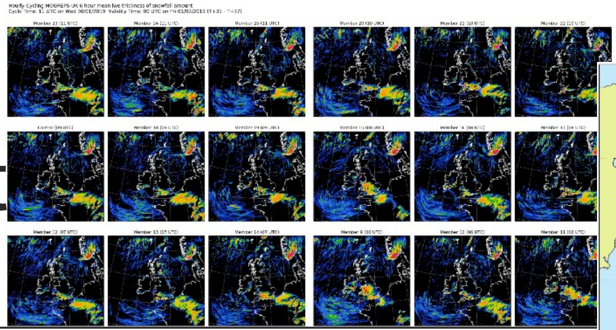
Seamless development

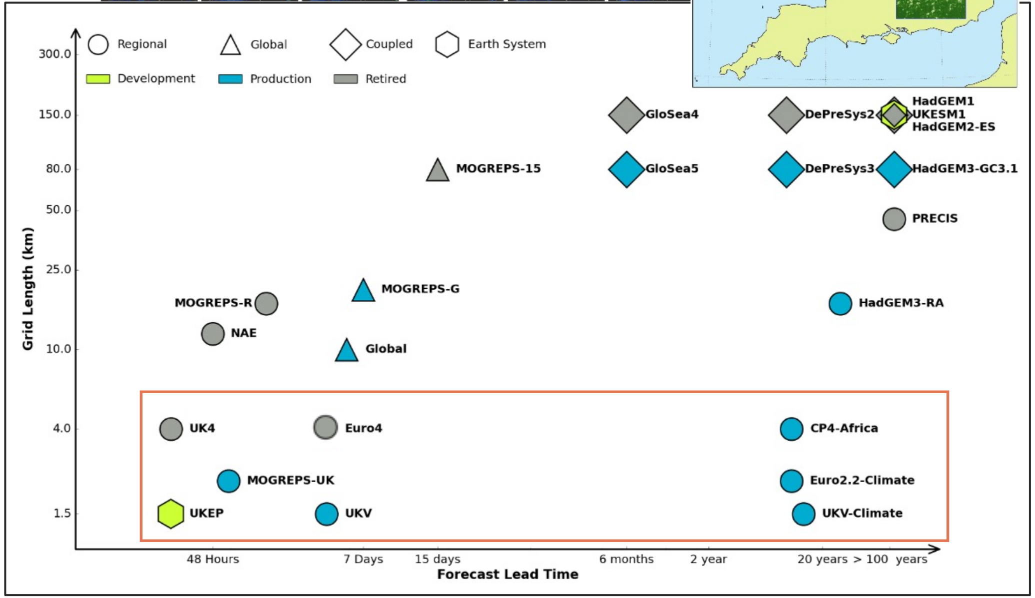
Global configurations	Horizontal resolution	Vertical levels
Deterministic NWP	0.14° x 0.09° (~10 km at mid-latitudes)	70 levels to 80 km
Ensemble NWP	0.28° x 0.18° (~20 km at mid-latitudes)	70 levels to 80 km
Seasonal/ Decadal climate	0.83° x 0.55° (~50 km at mid-latitudes)	85 levels to 85 km
Centennial/ ESM climate	1.875° x 1.25° (~140 km at mid-latitudes)	38 levels to 40 km



Seamless development

Regional configurations	Horizontal resolution	Vertical levels
UKV	1.5x1.5 km 950 x 1025 grid points	70 levels to 40 km
Ensemble	2.2x2.2 km 740 x 752 grid points	70 levels to 40 km
London Model	300mx300m 380x420	90 levels to 40 km
Relocatable LAM	various	various



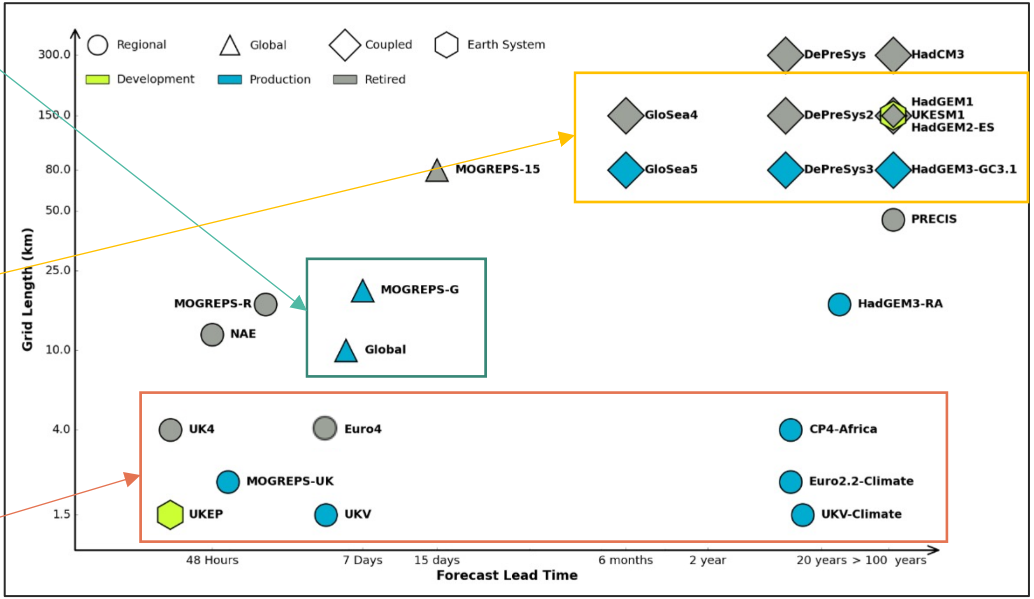


Seamless development

Global hi-resolution (NWP):
- Reaching limit of strong scaling
- Primary motivation for GungHo

Global low-resolution (climate/ESM):
- Dynamical core a smaller proportion of overall cost
- Greater cost from chemistry and diagnostics
- Motivates equations that are accurate on long timescales (e.g., conservation, no systematic bias)

Regional hi-resolution (NWP):
- Spatial domain scales well
- Motivates non-hydrostatic equation set



Met Office

Constraints for GungHo design

Algorithmic drivers

- Minimize total number of (expensive) operations required to get to solution. E.g.,
  - # of timesteps
  - # of iterations
  - # of stages
  - # of substeps
- Flexible design to mitigate against ‘dead ends’
- Flexible design for different science configurations

Computational drivers

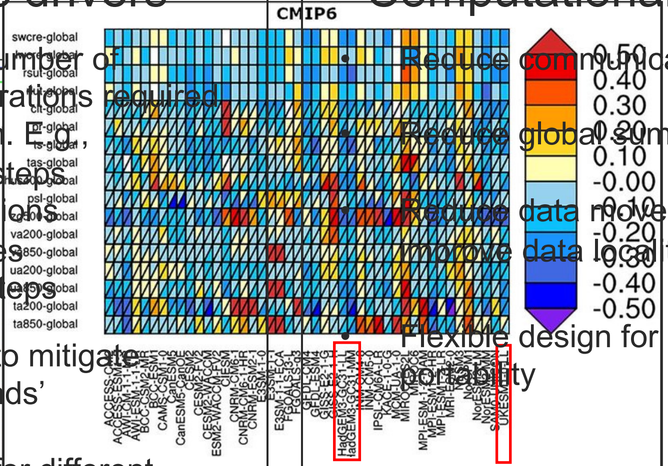
- Reduce communication
- Reduce global sums
- Reduce data movement and improve data locality
- Flexible design for performance portability

Science drivers

- Retain characteristics of ENDGame numerics:
  - Accurate representation of physical equations
  - Accurate dispersion
  - No computational modes (grid staggering)
  - Semi-implicit timestep (physics coupling)
  - Retain physical parametrizations
- Additionally want to improve inherent conservation properties

We have good NWP skill

Climate skill



Bock et al. 2020

Current generation modelling

Unified Model and ENDGame dynamical core

Met Office

Governing Equations

Momentum Equation

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\Omega}\times\boldsymbol{u} + \nabla\Phi + \frac{c_p\theta}{1+\Sigma m_x}\nabla\Pi = \boldsymbol{S}_{\boldsymbol{u}}$$

Thermodynamic Equation

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u}\cdot\nabla\theta = S_\theta$$

Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) = 0$$

Moisture Transport Equation

$$\frac{\partial m_x\rho}{\partial t} + \nabla\cdot(m_x\rho\boldsymbol{u}) = S_x$$

Equation of State

$$\Pi^{\frac{1-\kappa}{\kappa}} + \frac{R}{p_0}\rho\theta = 0$$

Moisture

Wave motion

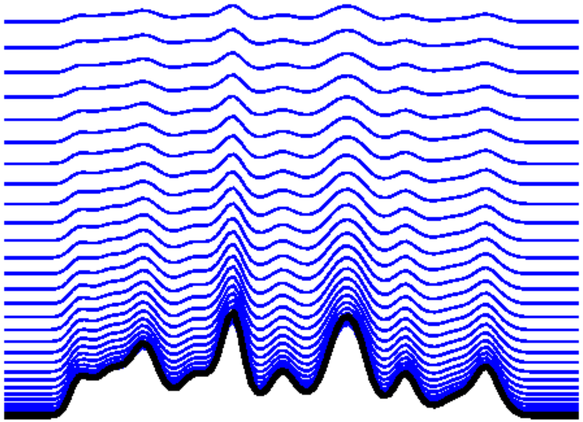
Transport terms

Subgrid forcings

Structure of the Grid

Horizontal: latitude-longitude

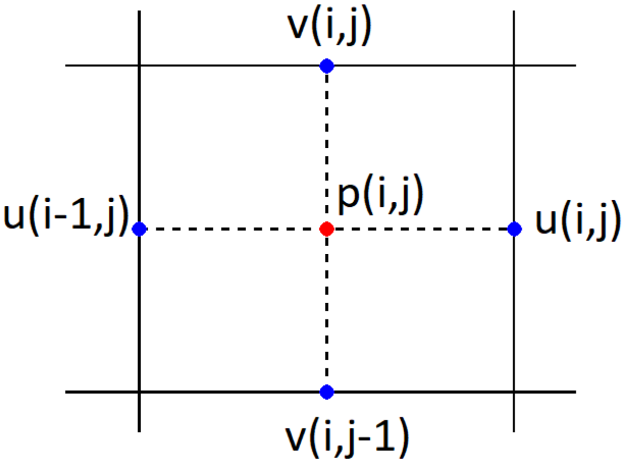
Vertical: terrain following



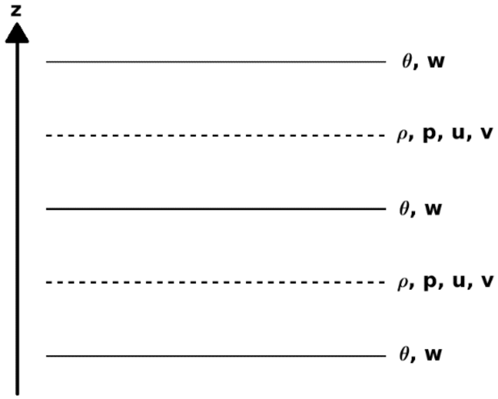
Spatial Discretisation: Staggered Finite Differences

Approximate spatial derivatives by finite differences: $$\frac{\partial X}{\partial z} \rightarrow \frac{X\left(t, z + {}^{\Delta z}/_{2}\right) - X\left(t, z - {}^{\Delta z}/_{2}\right)}{\Delta z}$$

Horizontal:
C-grid



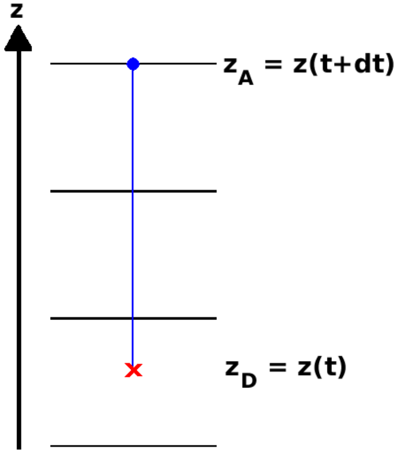
Vertical:
Charney-Phillips grid



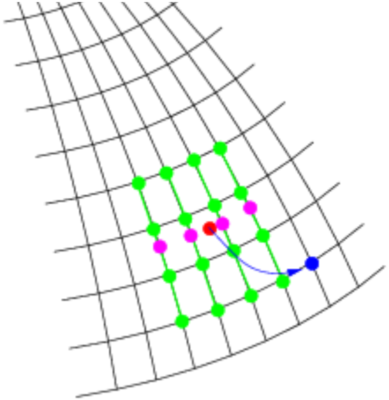
Transport: Semi-Lagrangian

Finite difference along a fluid parcel trajectory:

$$\frac{DX}{Dt} \rightarrow \frac{X\left(t+\Delta t, z(t+\Delta t)\right) - X\left(t, z(t)\right)}{\Delta t}$$



Interpolate *X* at time *t* to the departure point

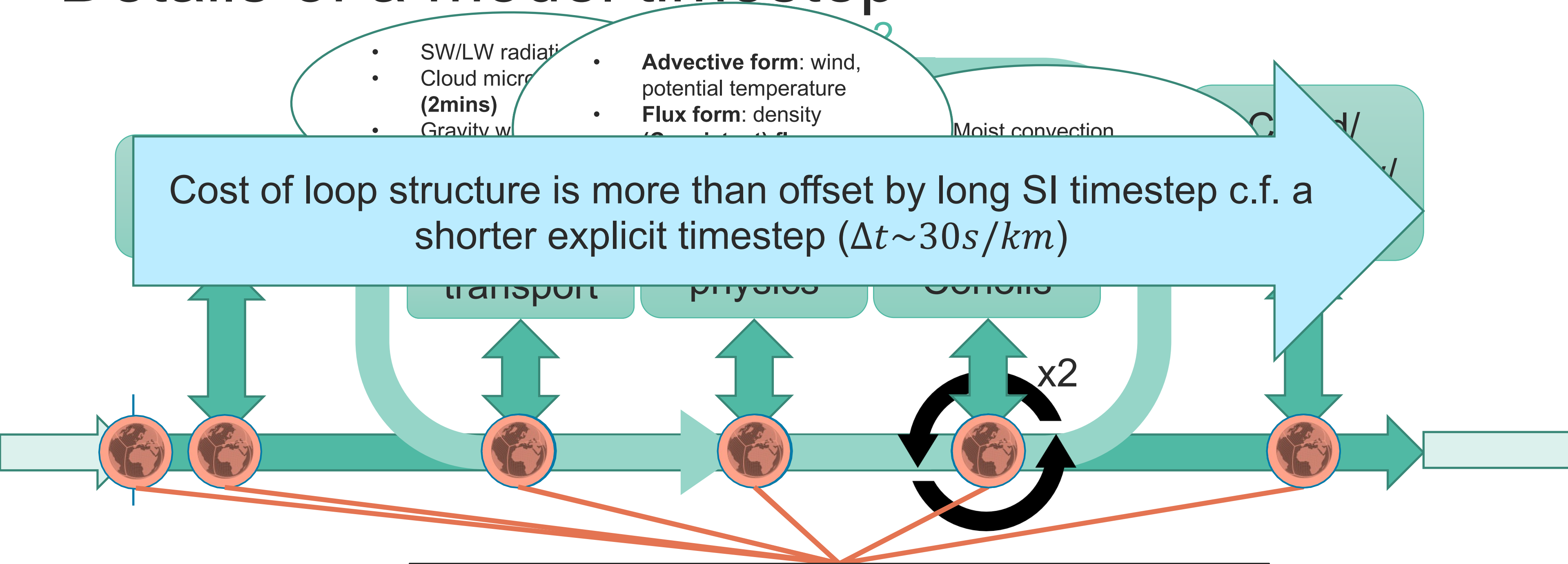


Met Office

Semi-implicit timestep

- Continuous: $\frac{\partial x}{\partial t} = f\left(x(t)\right)$
- Explicit: $\frac{x^{n+1} - x^n}{\Delta t} = f(x^n)$
- Implicit: $\frac{x^{n+1} - x^n}{\Delta t} = f(x^{n+1})$
- Semi-implicit: $\frac{x^{n+1} - x^n}{\Delta t} = \alpha f(x^{n+1}) + (1-\alpha) f(x^n)$
- Picard iteration: $x^{k+1} = x^n + \Delta t(\alpha f\left(x^k\right) + (1-\alpha) f(x^n))$, take $x^{n+1} = x^{k+1}$after k iterations

Details of a model timestep



Slow physics = 1 call
Core transport = 2 calls
Tracer transport = 1 call

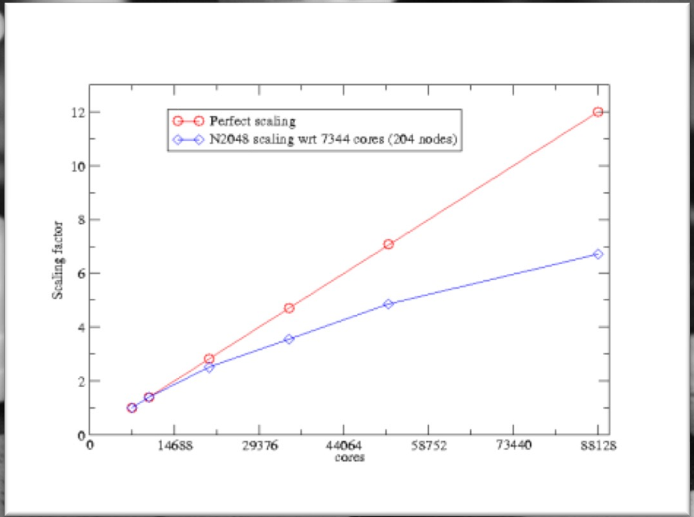
Fast physics = 2 calls
Linear solver = 4 calls
Chem/aerosol = 1 call

Why the next generation?

LFRic and GungHo dynamical core

Story begins at the ENDGame

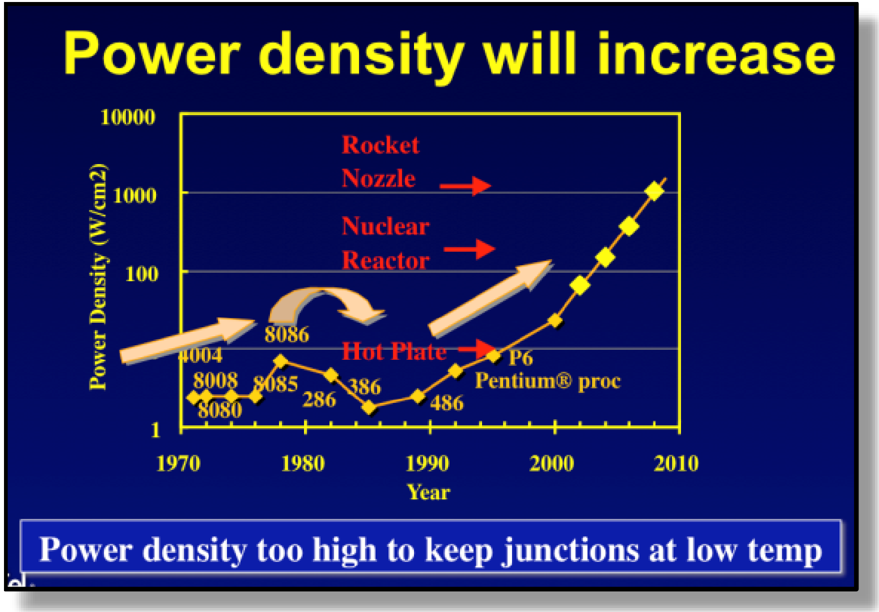
- ENDGame operational 2014
- Greatly improved scalability…
- …but not enough for Exascale



Andy Malcolm, Paul Selwood

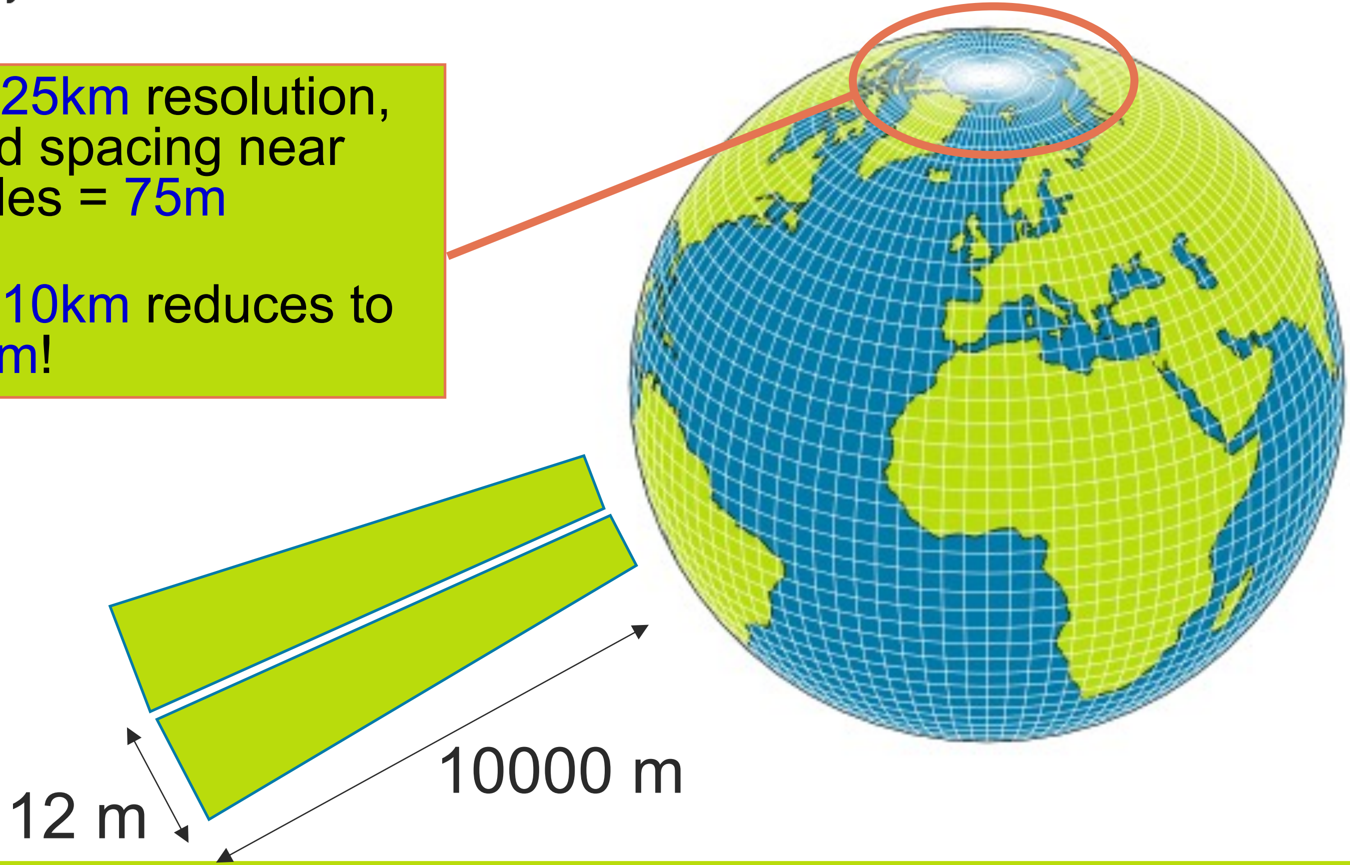
Met Office Hadley Centre

The problem is hardware

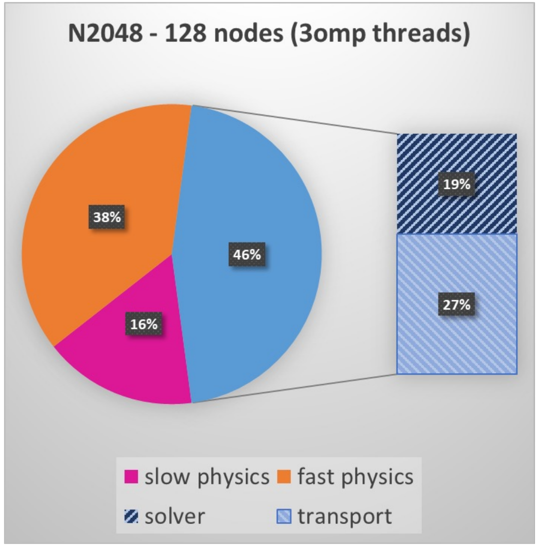


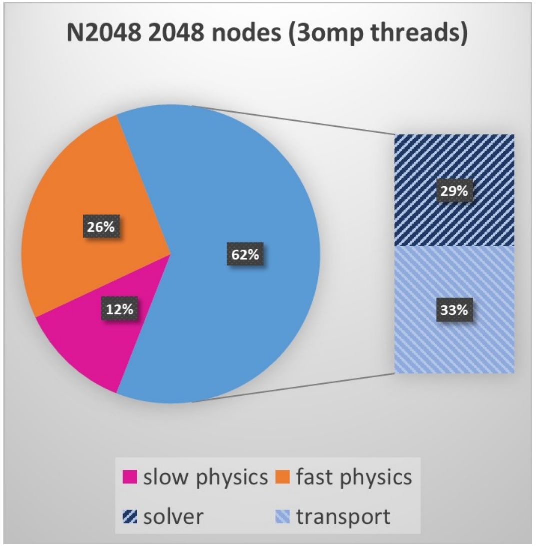
Intel, 2000

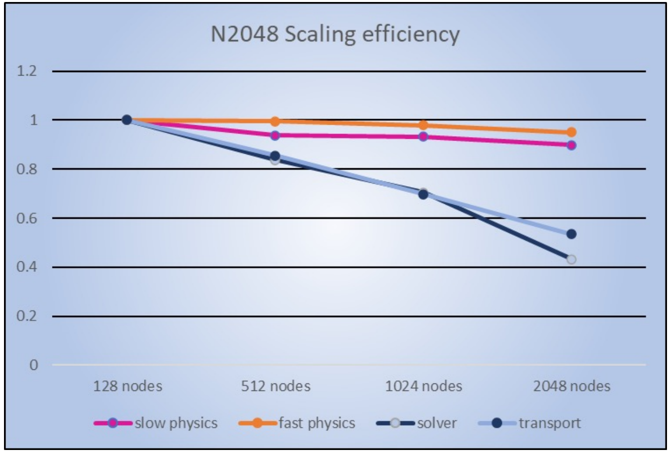
- At 25km resolution, grid spacing near poles = 75m
- At 10km reduces to 12m!



UM strong scaling at N2048 (~6km)

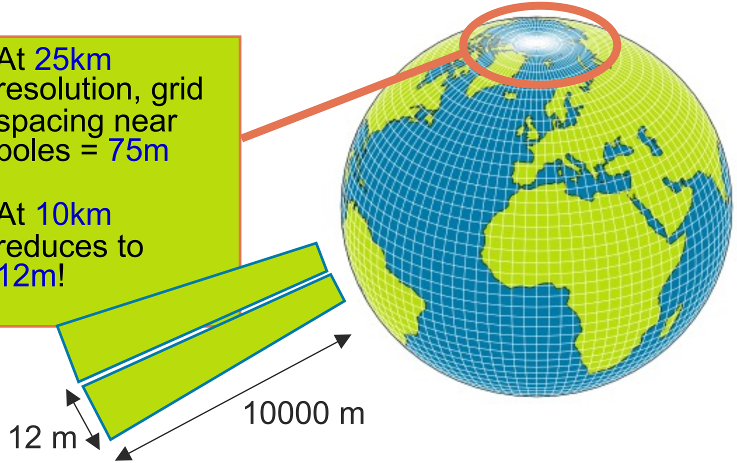






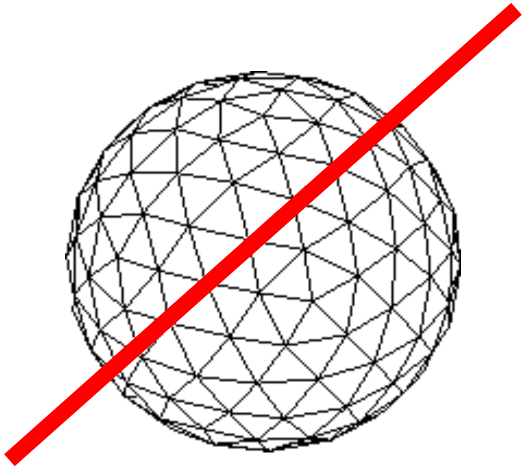
Performance Motivation

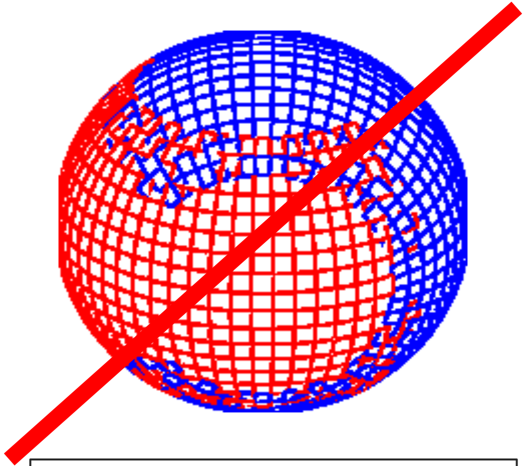
- At 25km resolution, grid spacing near poles = 75m
- At 10km reduces to 12m!



- Scalability for Lat-Lon mesh is limited by polar singularities
- Assumption of regular, orthogonal, finite difference scheme is deeply embedded in UM code
- No clear separation between computational and science code
- Choice to develop new infrastructure alongside the dynamical core

Choice of mesh (cubed-sphere)





- Triangles potentially difficult for coupling to physics and other system components.
- Not ruled out by GungHo and could be explored in future.

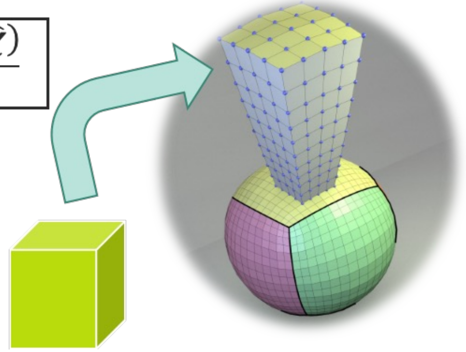
- Primary candidate and following assessment of scientific performance, is now becoming ‘baked into’ NGMS plans.

- Could retain ENDGame numerics.
- Not good for conservation/physics in the overlap region.

Met Office

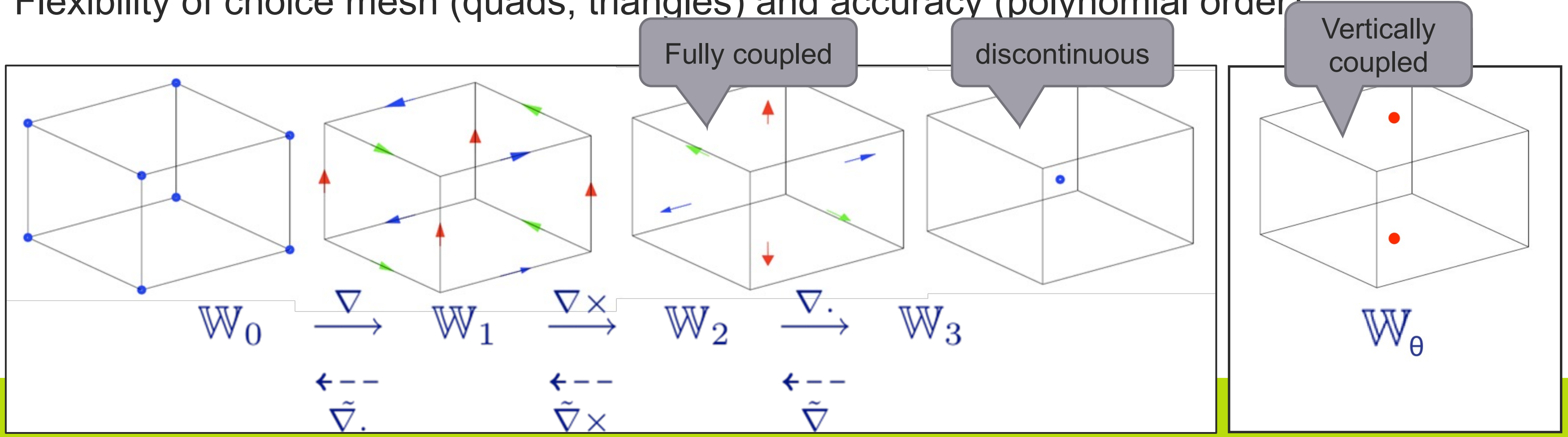
Mixed Finite Elements

$$\mathbf{J} \equiv \frac{\partial \boldsymbol{\phi}(\hat{\chi})}{\partial \hat{\chi}}$$



Mixed Finite Element method gives

- Compatibility: $\nabla\times\nabla\varphi = 0, \nabla\cdot\nabla\times\boldsymbol{v} = 0$
- Accurate balance and adjustment properties
- No orthogonality constraints on the mesh
- Flexibility of choice mesh (quads, triangles) and accuracy (polynomial order)



Met Office

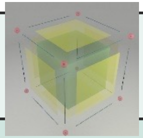
Mixed Finite Element Method

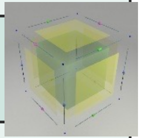
- Retains layout and properties of UM
- No requirement for orthogonality

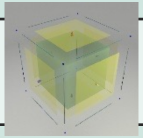
$$\mathbb{W}_0 \xrightarrow{\nabla} \mathbb{W}_1 \xrightarrow{\nabla\times} \mathbb{W}_2 \xrightarrow{\nabla\cdot} \mathbb{W}_3.$$

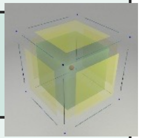
$$\mathbb{W}_\theta = \mathbf{k}\cdot\mathbb{W}_2$$

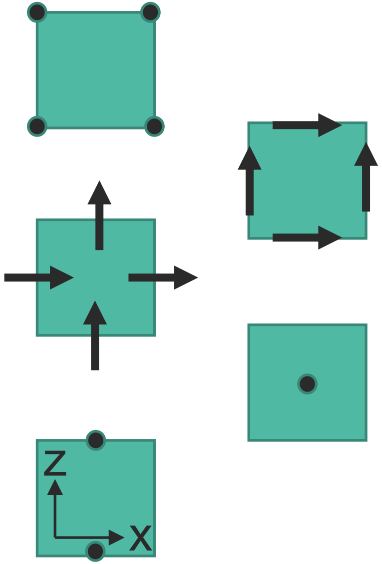
$\mathbb{W}_0$	Pointwise scalars	
$\mathbb{W}_1$	Circulation Vectors	
$\mathbb{W}_2$	Flux Vectors	Velocity
$\mathbb{W}_3$	Volume integrated Scalars	Pressure, Density
$\mathbb{W}_\theta$	Pointwise scalars	Potential Temperature, moisture



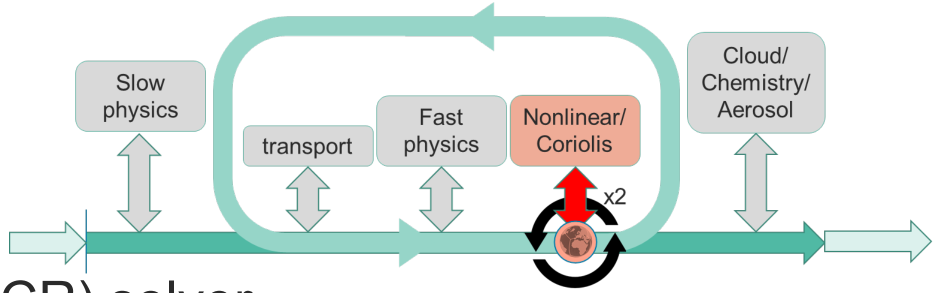








Met Office



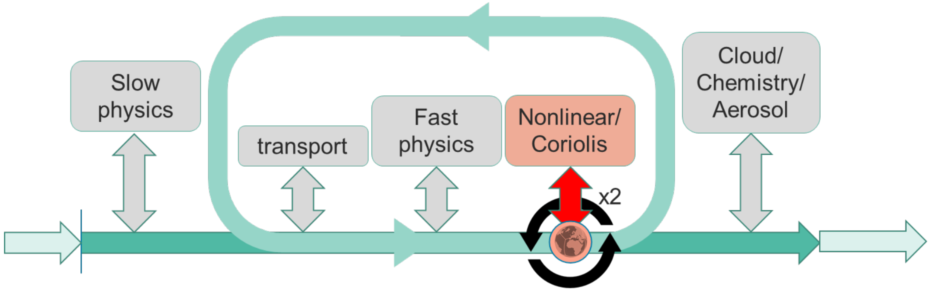
Linear Solver

- Solver Outer system with Iterative (GCR) solver

$$
\begin{pmatrix}
M_2^{\mu,C} & & -P_{2\theta}^{\Pi^*} & -G^{\theta^*} \\
D^{\rho^*} & M_3 & & \\
P_{\theta 2}^{\theta^*} & & M_\theta & \\
 & -M_3^{\rho^*} & -P_{3\theta}^{*} & M_3^{\Pi^*}
\end{pmatrix}
\begin{pmatrix}
\widetilde{u}' \\ \widetilde{\rho}' \\ \widetilde{\theta}' \\ \widetilde{\Pi}'
\end{pmatrix}
=
\begin{pmatrix}
-\mathcal{R}_u \\ -\mathcal{R}_\rho \\ -\mathcal{R}_\theta \\ -\mathcal{R}_\Pi
\end{pmatrix}
$$

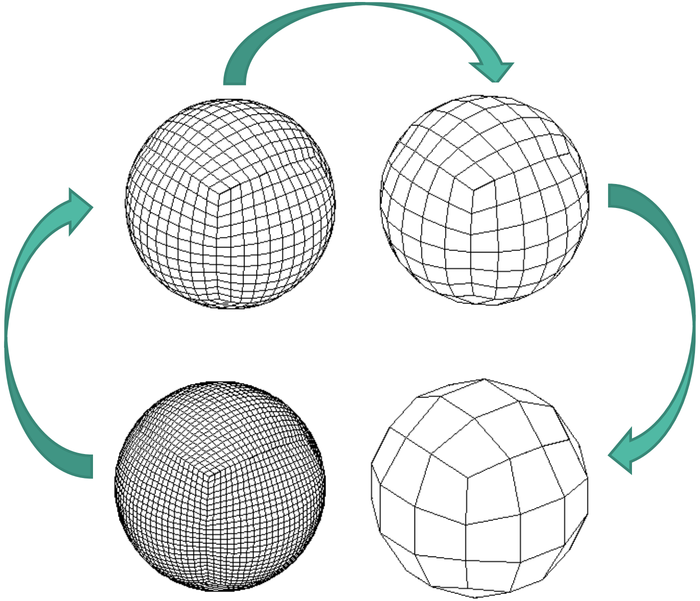
- Contains all couplings
- Velocity mass matrices couple in all directions and potential temperature mass matrices couple in the vertical
- Efficient solution requires a good preconditioner…

Multigrid preconditioner



- Helmholtz system $H\Pi' = R$ solved using a single Geometric-Multi-Grid V-cycle with block-Jacobi smoother

$$H = M_3^{\Pi^*} + \left(P_{3\theta}^{*}\mathring{M}_\theta^{-1}P_{\theta 2}^{\theta^*,z} + M_3^{\rho^*}M_3^{-1}D^{\rho^*}\right)\left(\mathring{M}_2^{\mu,C}\right)^{-1}G^{\theta^*}.$$



- Coupled mass matrices are now lumped
- Preconditioner system resembles single ENDGame solve

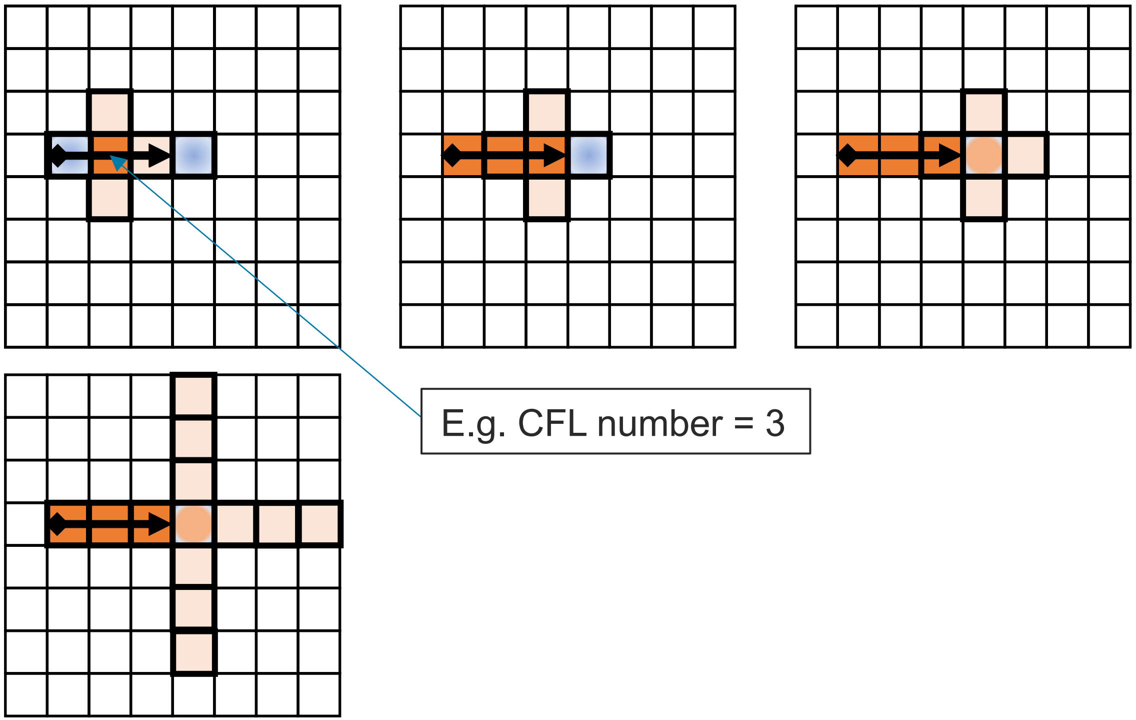
Met Office

Transport – horizontal options

Stage/substep 1 Stage/substep 2 Stage/substep 3

Eulerian scheme

Flux-form Semi-Lagrangian scheme



- Small mpi halo exchange
- Multiple exchanges
- Cheap computation
- Multiple calls

- Large mpi halo exchange
- Only 1 exchanges
- More expensive computation
- Only 1 call

Met Office

UM-LFRic differences/similarities

	ENDGame/UM	GungHo/LFRic
Grid	Lat-Long	Cubed-Sphere
Equation set	Deep, non-hydrostatic	Deep, non-hydrostatic
Prognostic Variables	ρ, θ, Π, u, v, w	ρ, θ, Π, **u**
Moist Variables	**Specific humidities & mixing ratios**	**mixing ratios !**
Spatial Discretisation	**2^{nd} Order FD**	**Mixed FEM**
Temporal Discretisation	**Iterative Semi-Implicit**	**Iterative Incremental Semi-Implicit**
Advection	Semi-Lagrangian	Dimensionally split, Eulerian or FFSL
Physics parametrizations	**UM,SOCRATES,JULES**	**UM,SOCRATES,JULES**

Algorithms, kernels and Psyclone

- Algorithms work on full domain fields:

```
!!!! Compute pressure for CFL and qsat removal
call exner_wth%copy_field_properties(pressure)
inv_kappa = 1.0_r_def/kappa
call invoke(setval_X(pressure,exner_wth),         &
            inc_X_powreal_a(pressure,inv_kappa),  &
            inc_a_times_X(p_zero,pressure))
```

- No need to worry about loops, mesh structure, parallelism – this is all taken care of in the mysterious “invoke”

- Kernels work on individual grid points:

```
do k = 0, nlayers-1

  ! Pressure on layer boundaries (note, top layer is set to zero below)
  p_theta_levels(1,1,k) = real(p_zero*(exner_in_wth(map_wth(1) + k)) &
                                **(1.0_r_um/kappa), r_um)

end do   ! k
```

- Direct looping over k dimension
- Indirect looping over horizontal mesh using “map_wth”

Algorithms, kernels and Psyclone

- Psyclone is auto-generated code which sits between algorithms and kernels
- Deals with data parallelism – halo exchanges, openMP
- Does looping over horizontal mesh if required
  - Option to pass the entire horizontal domain (of MPI rank) to a kernel, to allow i-first looping if this is more performant
- Metadata is added to kernels to configure Psyclone:

```
!> Kernel metadata for Psyclone
type, public, extends(kernel_type) :: methox_kernel_type
  private
  type(arg_type) :: meta_args(3) = (/                          &
       arg_type(GH_FIELD,  GH_REAL, GH_WRITE, WTHETA), &
       arg_type(GH_FIELD,  GH_REAL, GH_READ,  WTHETA), &
       arg_type(GH_SCALAR, GH_REAL, GH_READ          )  &
       /)
  integer :: operates_on = CELL_COLUMN
contains
  procedure, nopass :: methox_code
end type methox_kernel_type
```

Met Office

Psyclone auto-generated code:

```
! Compute rhs = u + dt*grad(p)
call rhs%initialise( u%get_function_space() )
call grad%initialise( u%get_function_space() )
call invoke( setval_c(grad, 0.0_r_def),                 &
             gradient_kernel_type(grad, pressure, qr), &
             aX_plus_Y(rhs, dt, grad, u) )
```

PSy Layer

New algorithm

```
CALL rhs % initialise(u % get_function_space())
CALL grad % initialise(u % get_function_space())
CALL invoke_0(grad, pressure, rhs, dt, u, qr)
```

```
SUBROUTINE invoke_0(grad, pressure, rhs, dt, u, qr)

....

!$omp end parallel
```

MPI

```
IF (pressure_proxy%is_dirty(depth=1)) THEN
  CALL pressure_proxy%halo_exchange(depth=1)
END IF
!
```

Horizontal Looping

```
DO colour=1,ncolour
  !$omp parallel default(shared), private(cell)
  !$omp do schedule(static)
  DO cell=1,mesh%get_last_halo_cell_per_colour(colour,1)
    !
```

Kernel Call

```
    CALL gradient_code(nlayers,                                       &
                       grad_proxy%data, pressure_proxy%data,          &
                       ndf_w2, undf_w2, map_w2(:,cmap(colour, cell)), &
                       diff_basis_w2,                                 &
                       ndf_w3, undf_w3, map_w3(:,cmap(colour, cell)), &
                       basis_w3, np_xy, np_z, weights_xy, weights_z)
  END DO
  !$omp end do
  !$omp end parallel
END DO

! Set halos dirty/clean for fields modified in the above loop
CALL grad_proxy%set_dirty()
```

OpenMP

```
!$omp parallel default(shared), private(df)
!$omp do schedule(static)
```

Built-in

```
DO df=1,rhs_proxy%vspace%get_last_dof_annexed()
  rhs_proxy%data(df) = dt*grad_proxy%data(df) + u_proxy%data(df)
END DO
!$omp end do

! Set halos dirty/clean for fields modified in the above loop
```

MPI

```
!$omp master
CALL rhs_proxy%set_dirty()
!$omp end master
!
!$omp end parallel
END SUBROUTINE invoke_0
```

Met Office

Coupling in the physics

- Map LFRic fields onto UM arrays
  - Example here is k-first data, but can add i-loop:

```
j = 1
do k = 1, model_levels
  do i = 1, seg_len
    ! height of levels from centre of planet
    r_rho_levels(i,j,k)    = height_w3(map_w3(1,i) + k-1) + planet_radius
```

- Call the UM scheme
  - All spatially varying information must be passed in through argument list – no use of modules

- Map UM outputs back to LFRic fields

```
!-----------------------------------------------------------------------
! For the initial implementation we pass each individual column
! of data to an array sized (1,1,k) to match the UMs (i,j,k) data
! layout.
! assuming map_wth(1) points to level 0
! and map_w3(1) points to level 1
!-----------------------------------------------------------------------
do k = 1, nlayers
  ! wet density on theta and rho levels
  rho_wet_tq(1,1,k) = wetrho_in_wth(map_wth(1) + k)
  rho_wet(1,1,k) = wetrho_in_w3(map_w3(1) + k-1)
  ! dry density on theta and rho levels
  rho_dry_theta(1,1,k) = rho_in_wth(map_wth(1) + k)
  rho_dry(1,1,k) = rho_in_w3(map_w3(1) + k-1)
```

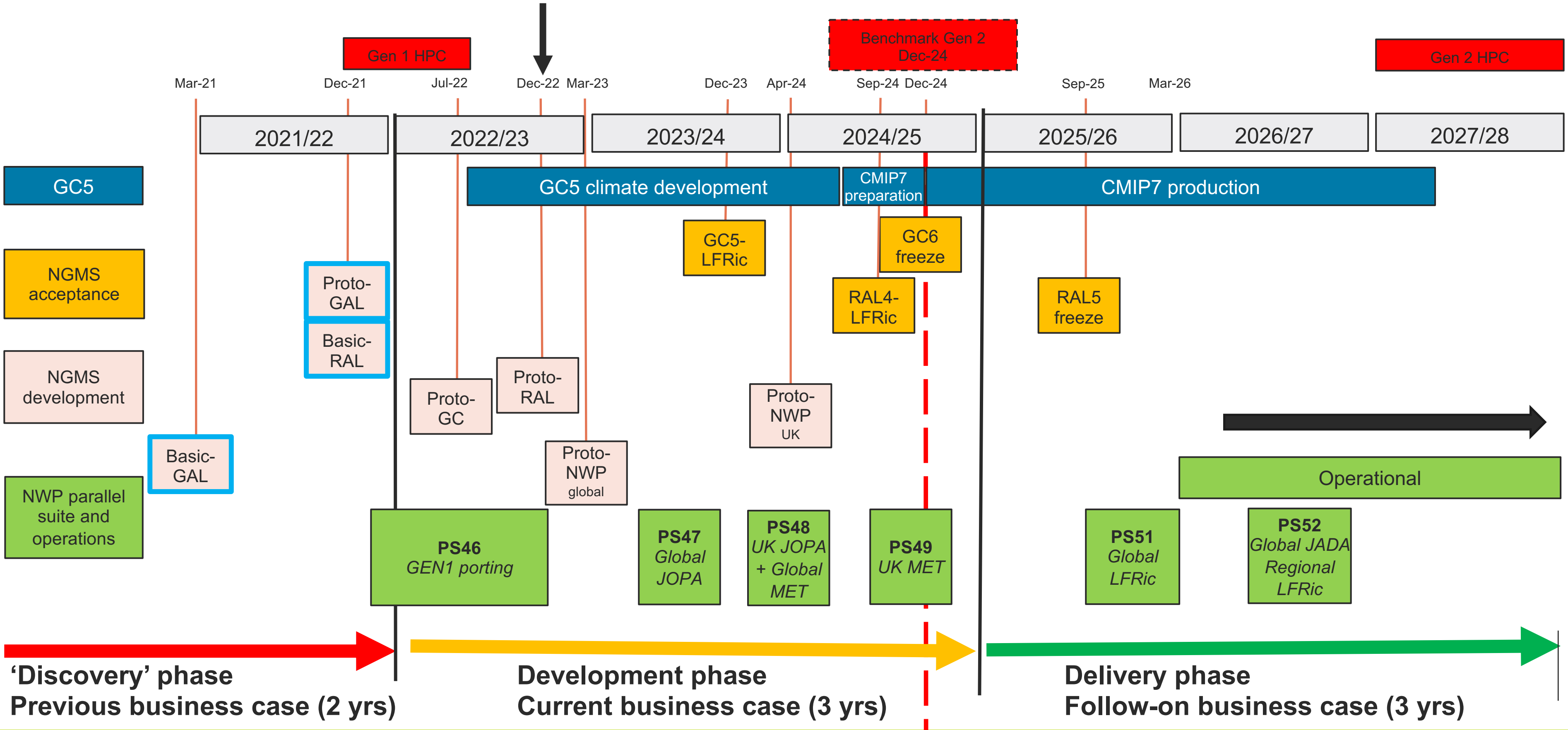
```
call glue_conv_6a                                                          &
  ( rows*row_length, segments, n_conv_levels, bl_levels, call_number, &
   seg_num, theta_conv, q_conv, qcl_conv, qcf_conv                         &
  , qrain_conv, qgraup_conv, qcf2_conv                                     &
  , cf_liquid_conv, cf_frozen_conv, bulk_cf_conv                           &
  , p_star, land_sea_mask                                                  &
  , u_conv, v_conv, w(1,1,1)                                               &
  , tot_tracer, dthbydt, dqbydt,   dqclbydt, dqcfbydt                      &
  , dcflbydt, dcffbydt, dbcfbydt, dubydt_p, dvbydt_p                       &
```

```
if (l_mom) then
  do k = 1, n_conv_levels
    u_conv(1,1,k) = u_conv(1,1,k)    + dubydt_p(1,1,k) * timestep_conv
    v_conv(1,1,k) = v_conv(1,1,k)    + dvbydt_p(1,1,k) * timestep_conv
    ! total increments
    du_conv(map_w3(1) + k -1) = du_conv(map_w3(1) + k -1) + dubydt_p(1,1,k) * timestep_conv
    dv_conv(map_w3(1) + k -1) = dv_conv(map_w3(1) + k -1) + dvbydt_p(1,1,k) * timestep_conv
  end do
end if    !l_mom
```

Current status

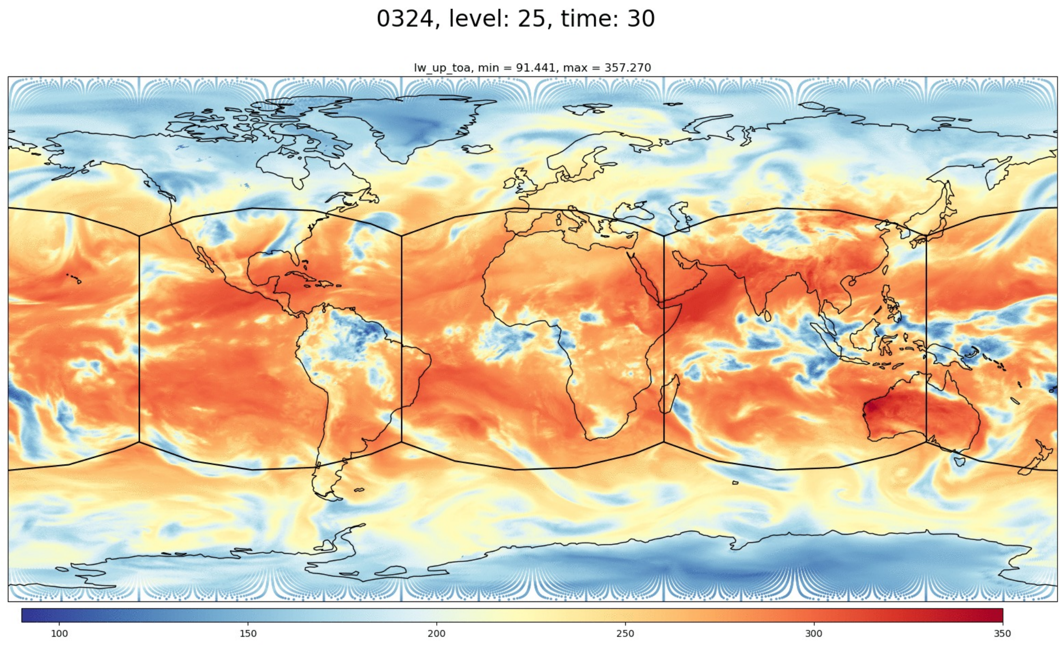
Met Office

NGMS timeline



Met Office

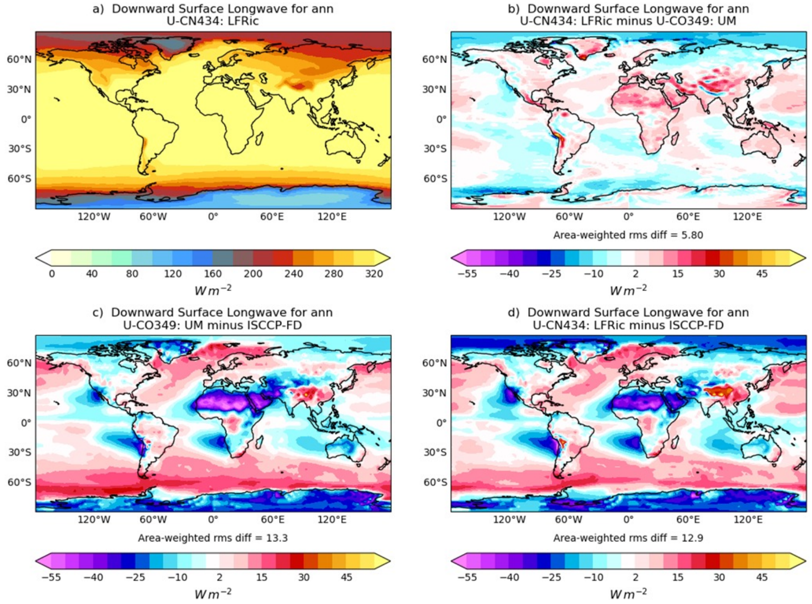
Global status

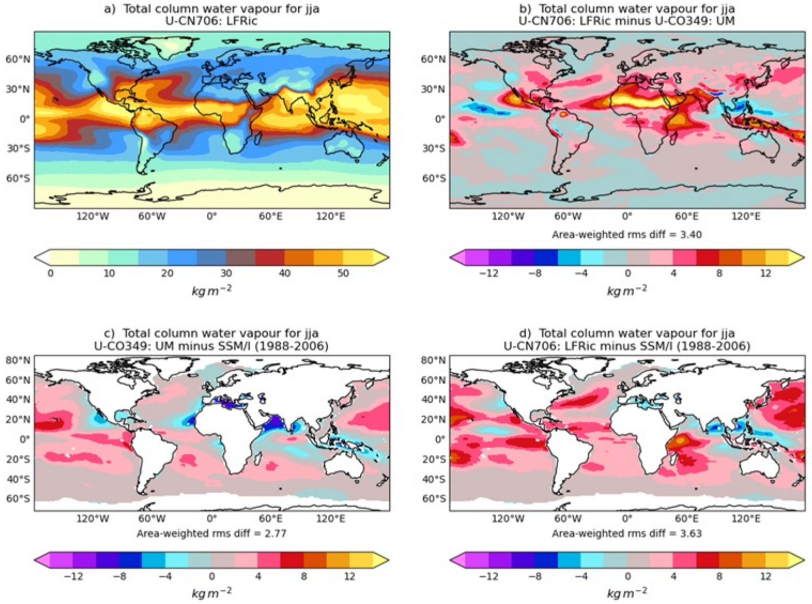


- C448 (20km) simulations running
- Currently attempting C896 (10km)…

Climate assessment

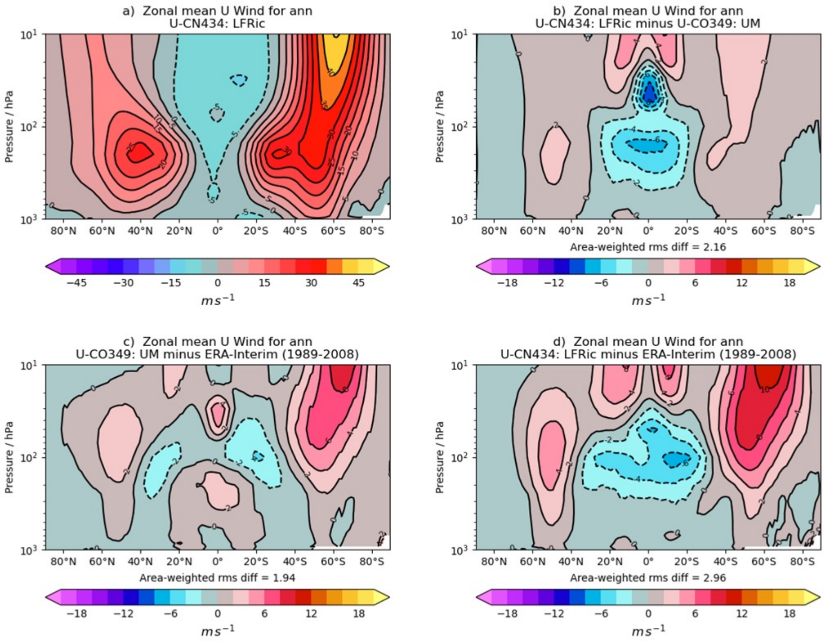
- Increased humidity over land – broadly seems like a good thing





Met Office

Climate assessment



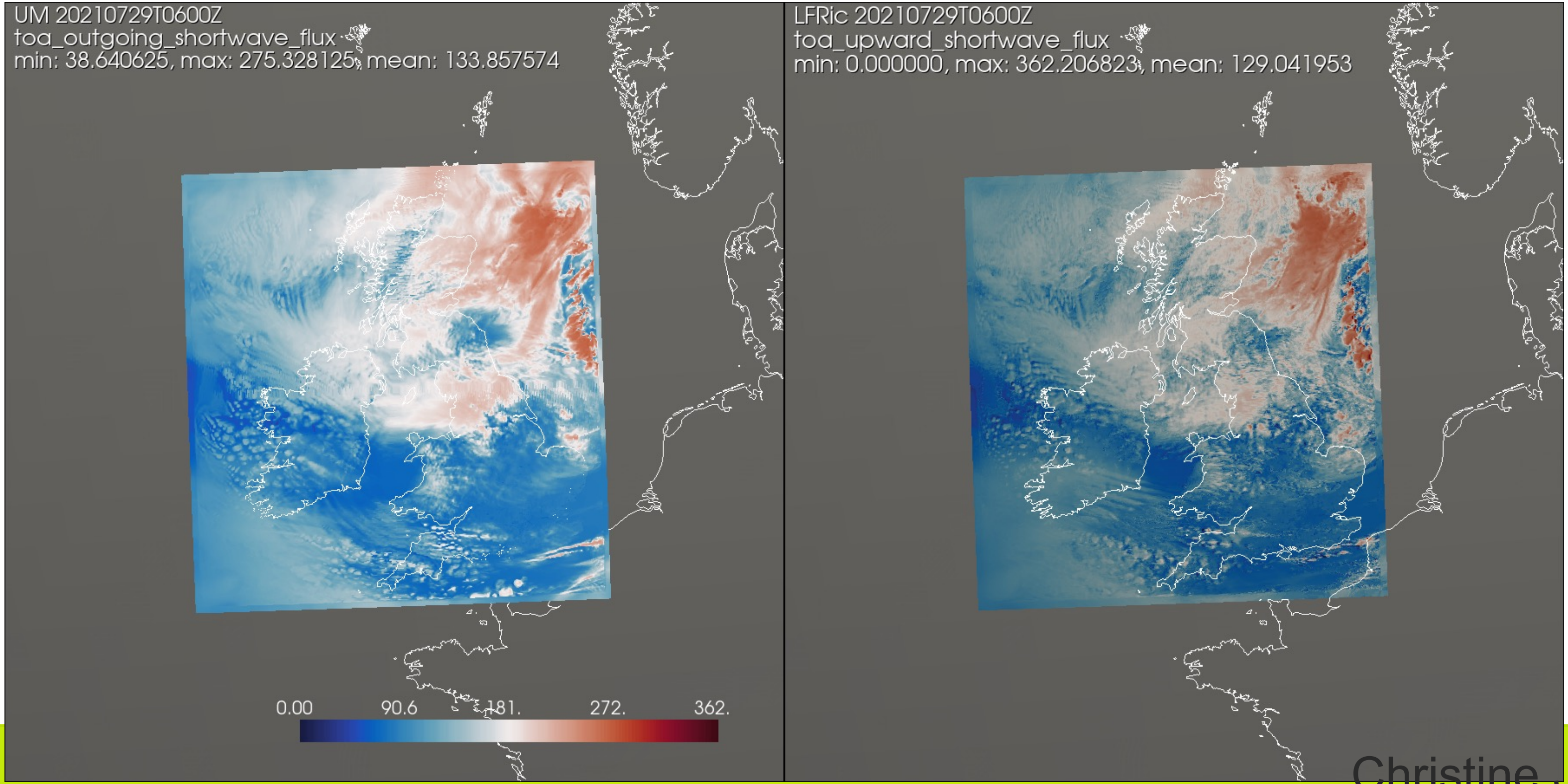
- Large scale circulation is a bit different
- Jets moved polewards

- In gross terms, the mean climate looks broadly similar however

Met Office

Regional status

- 1.5km UK simulations looking reasonable

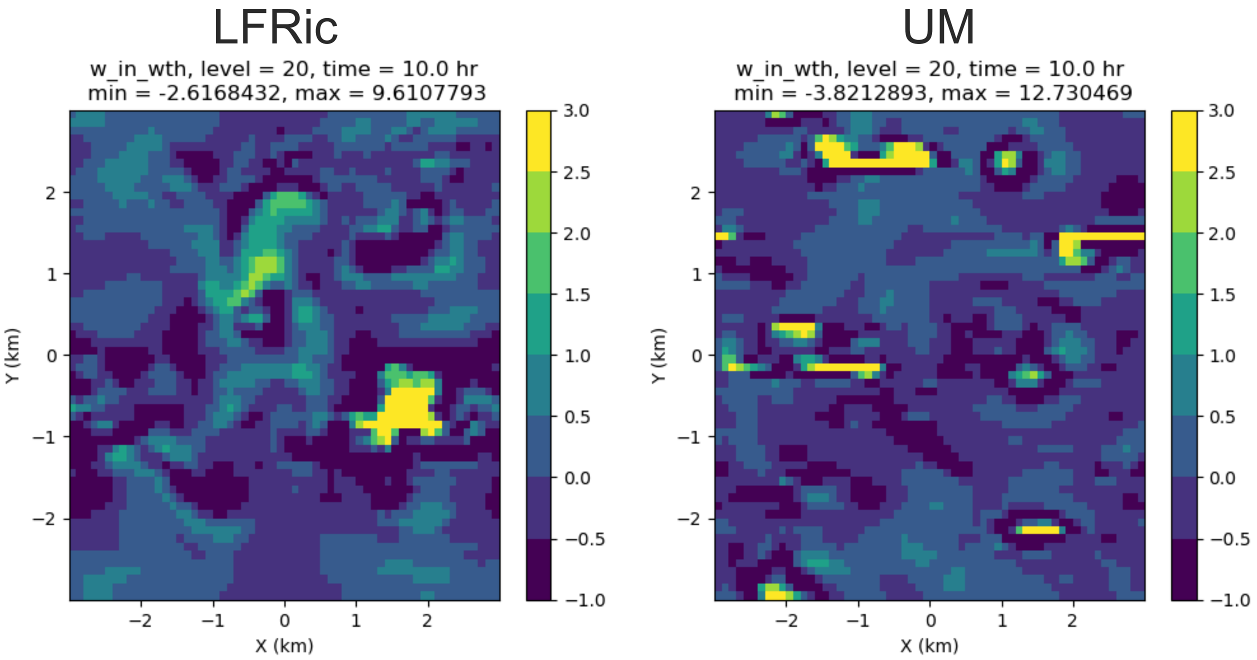


Christine Johnson

Met Office

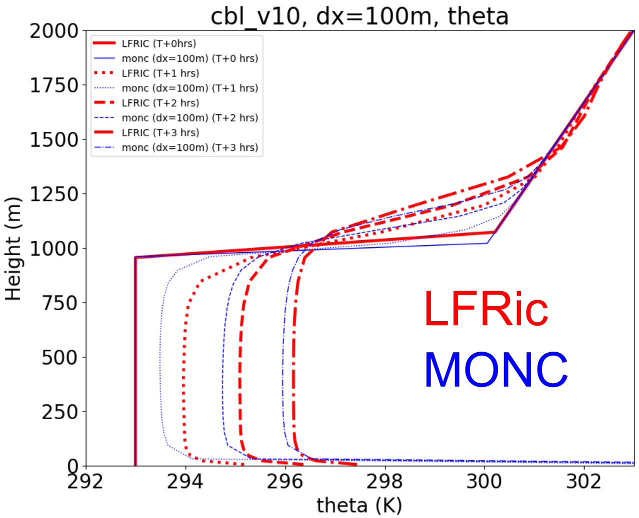
Very high resolution

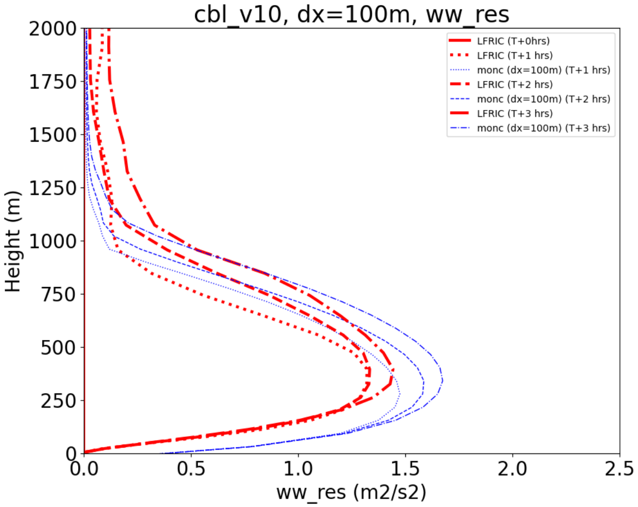
- Idealised "radiative-convective equilibrium" at 100m grid-length also shows good comparison
- Structures appear more coherent and better resolved in LFRic



Very high resolution

- Idealised dry convective boundary layer at 100m grid-length
- Comparison to large-eddy model (MONC) is pretty good
- Promising signs that LFRic could behave reasonable as a large-eddy model, and certainly be suitable for ~100m NWP



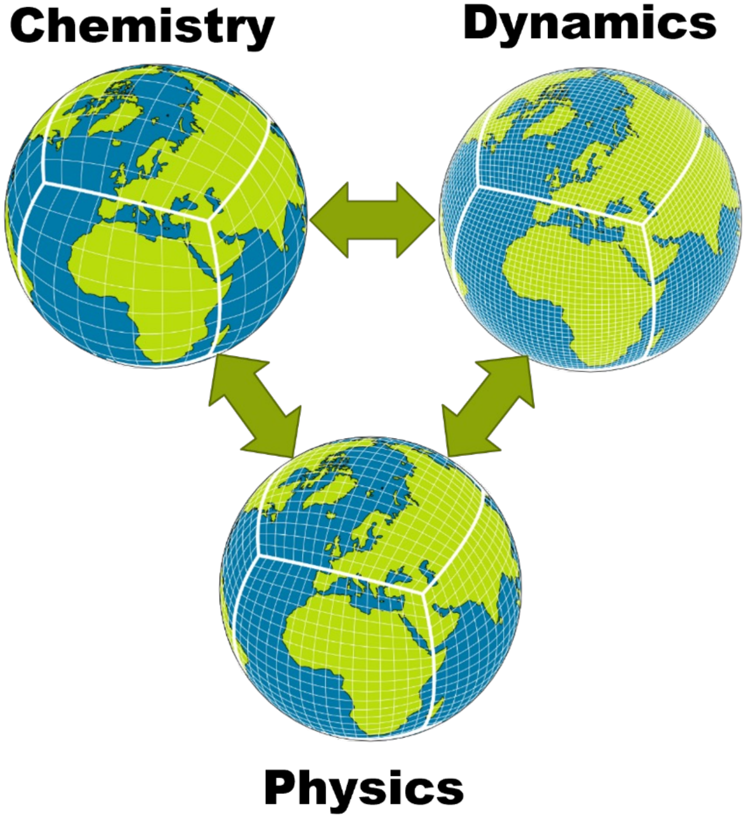


Met Office

Future plans

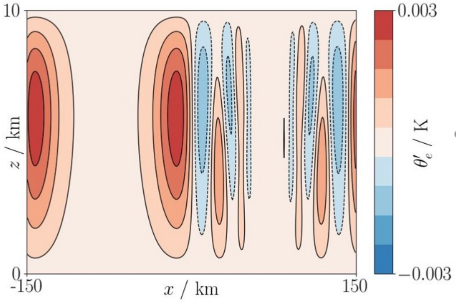
Multi-resolution coupling

- Currently the UM runs a hybrid "junior-senior" Earth system model, coupling an N216 physical model to N96 chemistry & ES components
- Aim to design this from the bottom up with LFRic, allowing coupling between different model components (dynamics, physics, chemistry) on different meshes
- Infrastructure can be equally applicable to limited-area models

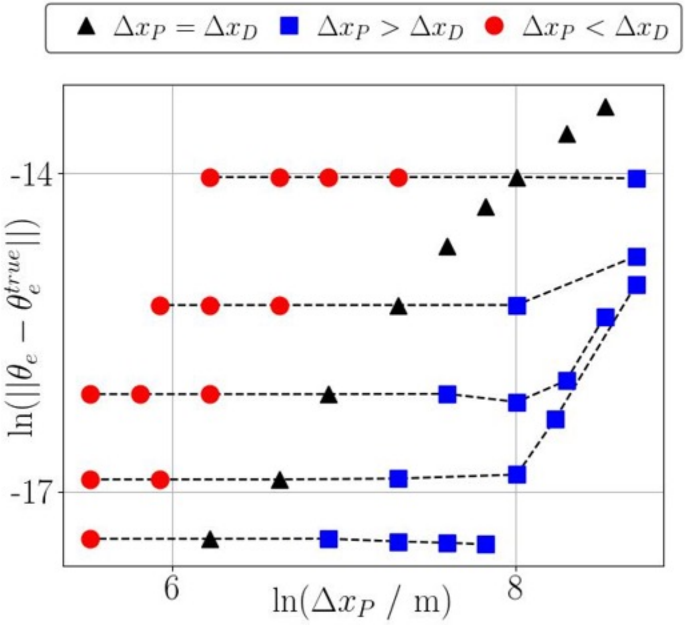


Met Office

Idealised testing

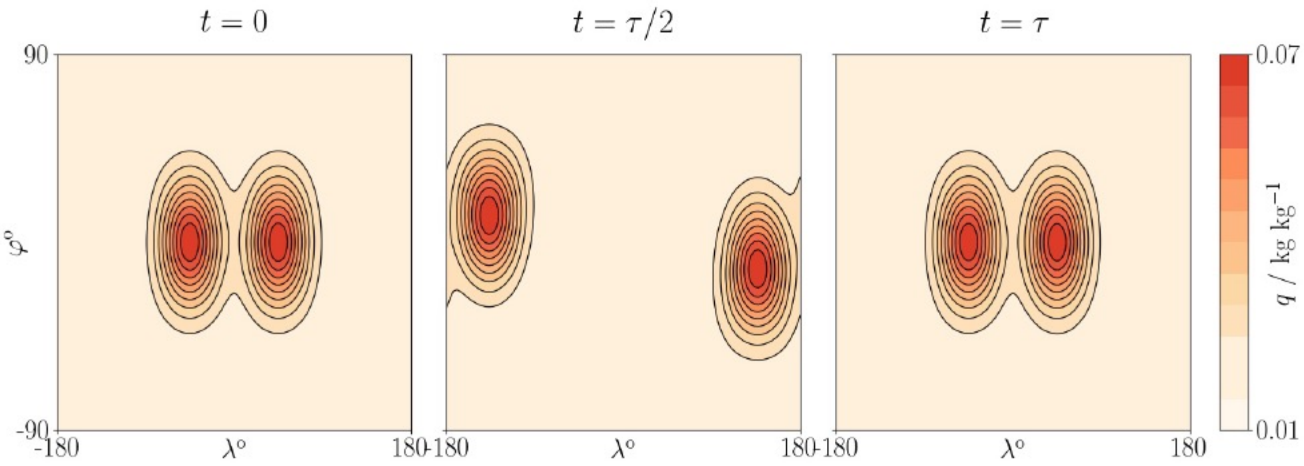


- Moist gravity wave with condensation/evaporation
- Black shows convergence of results as dynamics and physics resolution is increased
- Red shows increasing physics resolution beyond dynamics does not improve errors
- Blue shows degrading physics resolution initially has no effect, but then leads to strong degradation



Transport on a different mesh

- Tracer bubble exists at coarser resolution than model dynamics
- Dynamical winds driver tracer transport on it's native mesh
- Works, and is much cheaper than transport on dynamics mesh

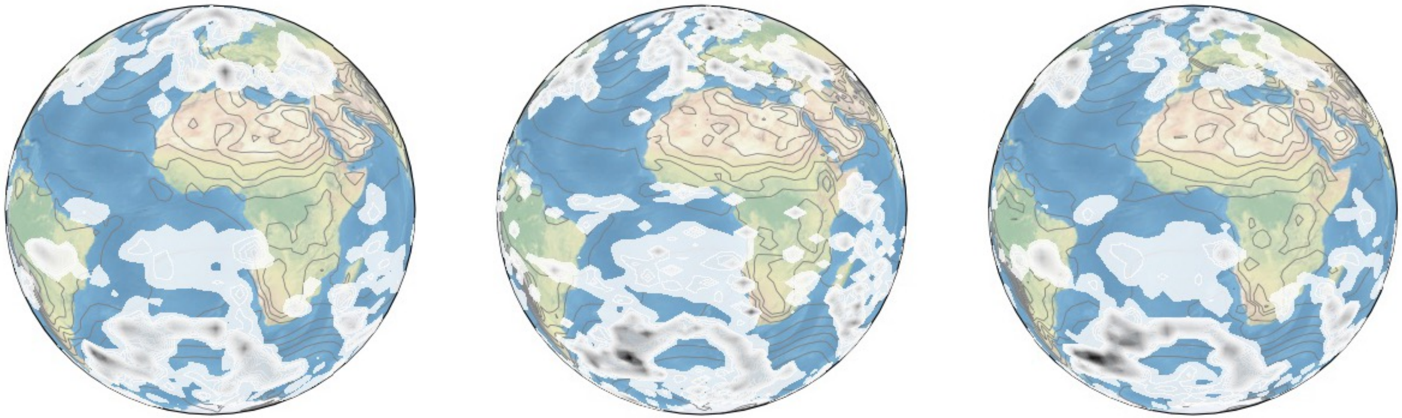


Global NWP simulations

- Initial tests demonstrate that it's possible to run full model with physics at different resolution to dynamics
- Results look plausible in all cases

t = -1.5 hr

$\Delta x_{phys} = \frac{1}{2}\Delta x_{dyn}$ $\quad$ $\Delta x_{phys} = \Delta x_{dyn}$ $\quad$ $\Delta x_{phys} = 2\Delta x_{dyn}$



Met Office

What’s changing

- Infrastructure
- Code layout (algorithms & kernels)
- Data layout (k-first ordering)
- Input/output & file format (netcdf)
- Transport (substepped method of lines or flux-form semi-lagrangian – both locally conservative)
- Solver (mixed system with multigrid preconditioner)

What’s not changing (much)

- Physical parametrizations
  - UM, Jules, Socrates, Casim, UKCA built directly from their current repositories
- Horizontal and vertical grids (C-grid horizontal, Charney-Phillips vertical)
- Timestep structure (slow/fast physics, iterative outer/inner loop dynamics)
- Use of Rose, fcm etc