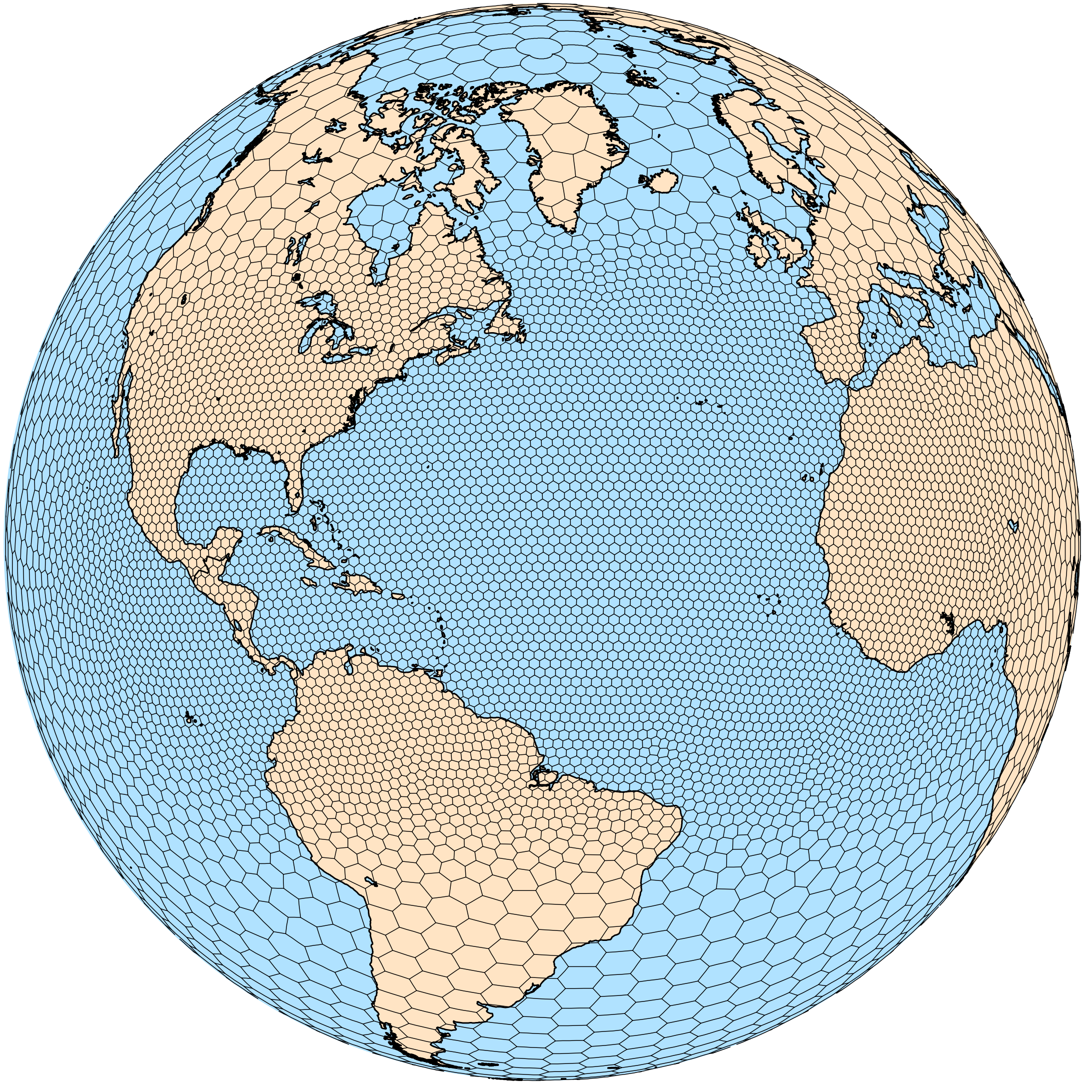
Dynamical Core

- Time integration
  - Algorithms
  - Timesteps
  - Namelist parameters
  - References
- Spatial Discretization for the dynamics



MPAS
Model for Prediction Across Scales

MPAS Nonhydrostatic Atmospheric Solver

Equations

- Prognostic equations for coupled variables.
- Generalized height coordinate.
- Horizontally vector-invariant equation set.
- Continuity equation for dry air mass.
- Thermodynamic equation for coupled potential temperature.

Variables: $(U, V, \Omega, \Theta, Q_j) = \tilde{\rho}_d\,(u, v, \omega, \theta, q_j)$ $\qquad \tilde{\rho}_d = \rho_d/\zeta_z$

Vertical coordinate: $z = \zeta + A(\zeta)h_s(x,y,\zeta)$

Prognostic equations:

$$\frac{\partial \mathbf{V}_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial \mathbf{z}_H p}{\partial \zeta}\right] - \eta\,\mathbf{k}\times\mathbf{V}_H - \mathbf{v}_H\nabla_\zeta\cdot\mathbf{V} - \frac{\partial \Omega \mathbf{v}_H}{\partial \zeta} - \rho_d\nabla_\zeta K + \mathbf{F}_{V_H}$$

$$\frac{\partial W}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\frac{\partial p}{\partial \zeta} + g\tilde{\rho}_m\right] - (\nabla\cdot\mathbf{v}\,W)_\zeta + F_W$$

$$\frac{\partial \Theta_m}{\partial t} = -(\nabla\cdot\mathbf{V}\,\theta_m)_\zeta + F_{\Theta_m}$$

$$\frac{\partial \tilde{\rho}_d}{\partial t} = -(\nabla\cdot\mathbf{V})_\zeta$$

$$\frac{\partial Q_j}{\partial t} = -(\nabla\cdot\mathbf{V}\,q_j)_\zeta + F_{Q_j}$$

Diagnostics and definitions:

$$\frac{\rho_m}{\rho_d} = 1 + q_v + q_c + q_r + \ldots$$

$$p = p_0\left(\frac{R_d\zeta_z\Theta_m}{p_0}\right)^\gamma \qquad \theta_m = \theta[1 + (R_v/R_d)q_v]$$

Time Integration

3rd Order Runge-Kutta time integration

Advance one time step $\phi^t \rightarrow \phi^{t+\Delta t}$

$$\frac{\partial U}{\partial t} = RHS_u$$

$$\frac{\partial W}{\partial t} = RHS_w$$

•
•
•

$$\phi^* = \phi^t + \frac{\Delta t}{3} RHS(\phi^t)$$

$$\phi^{**} = \phi^t + \frac{\Delta t}{2} RHS(\phi^*)$$

$$\phi^{t+\Delta t} = \phi^t + \Delta t \, RHS(\phi^{**})$$

$$\phi_t = ik\phi; \quad \phi^{n+1} = A\phi^n; \quad |A| = 1 - \frac{(k\Delta t)^4}{24}$$ + H.O.T

Time Integration

2nd-order RK variant – default in MPAS

$$\frac{\partial U}{\partial t} = RHS_u$$

$$\frac{\partial W}{\partial t} = RHS_w$$

•
•
•

Advance one time step $\phi^t \rightarrow \phi^{t+\Delta t}$

$$\phi^* = \phi^t + \frac{\Delta t}{2} RHS(\phi^t)$$

$$\phi^{**} = \phi^t + \frac{\Delta t}{2} RHS(\phi^*)$$

$$\phi^{t+\Delta t} = \phi^t + \Delta t \, RHS(\phi^{**})$$

$$\phi_t = ik\phi; \quad \phi^{n+1} = A\phi^n; \quad |A| = 1 - \frac{(k\Delta t)^3}{12}$$ + H.O.T

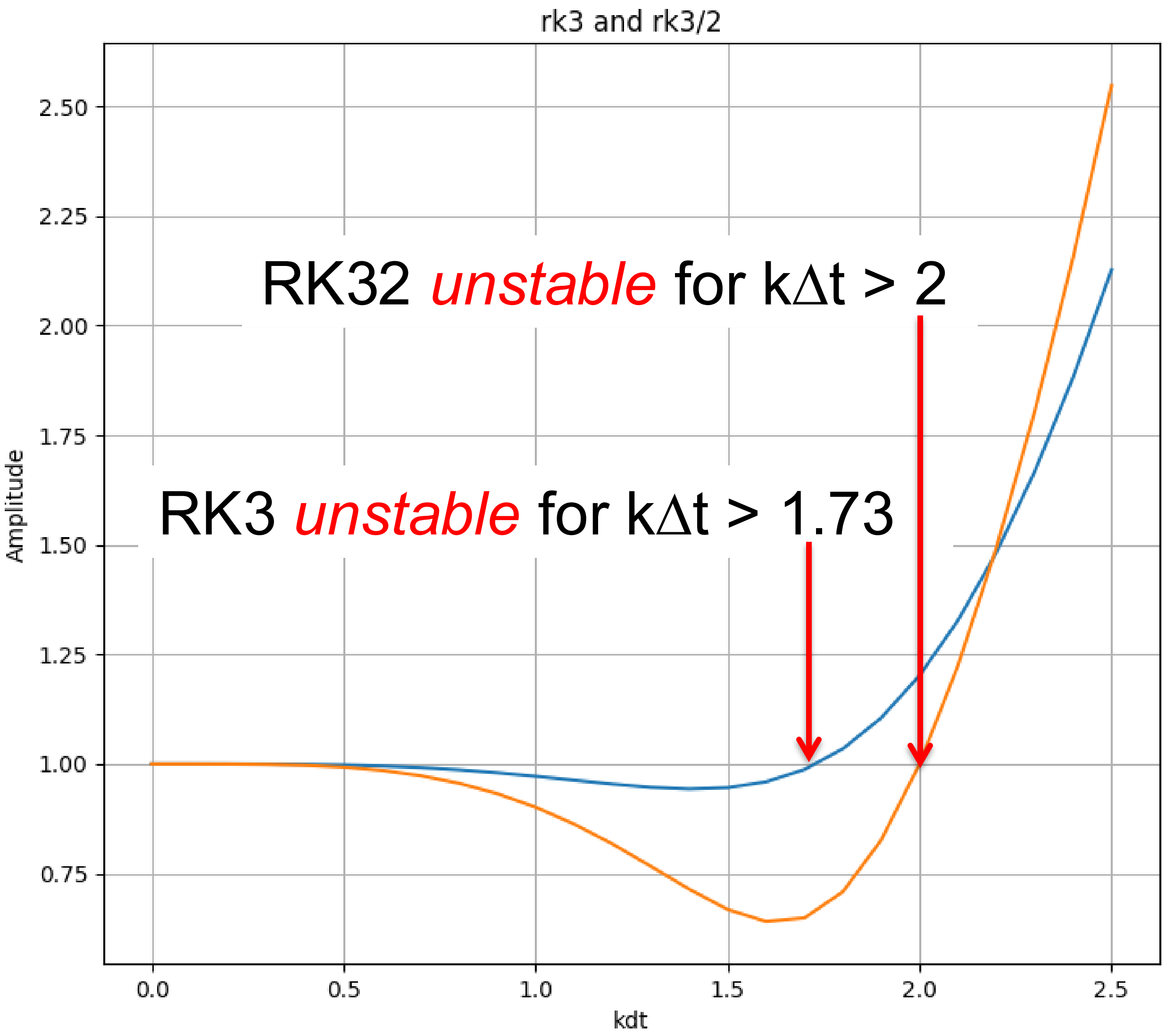
MPAS
Model for Prediction Across Scales

Time Integration

$$\phi_t = i k \phi; \quad \phi^{n+1} = A \phi^n$$

Exact: $|A| = 1$
RK3 and RK32 →

In applications we see little difference in MPAS solutions using RK3 compared to those using RK32



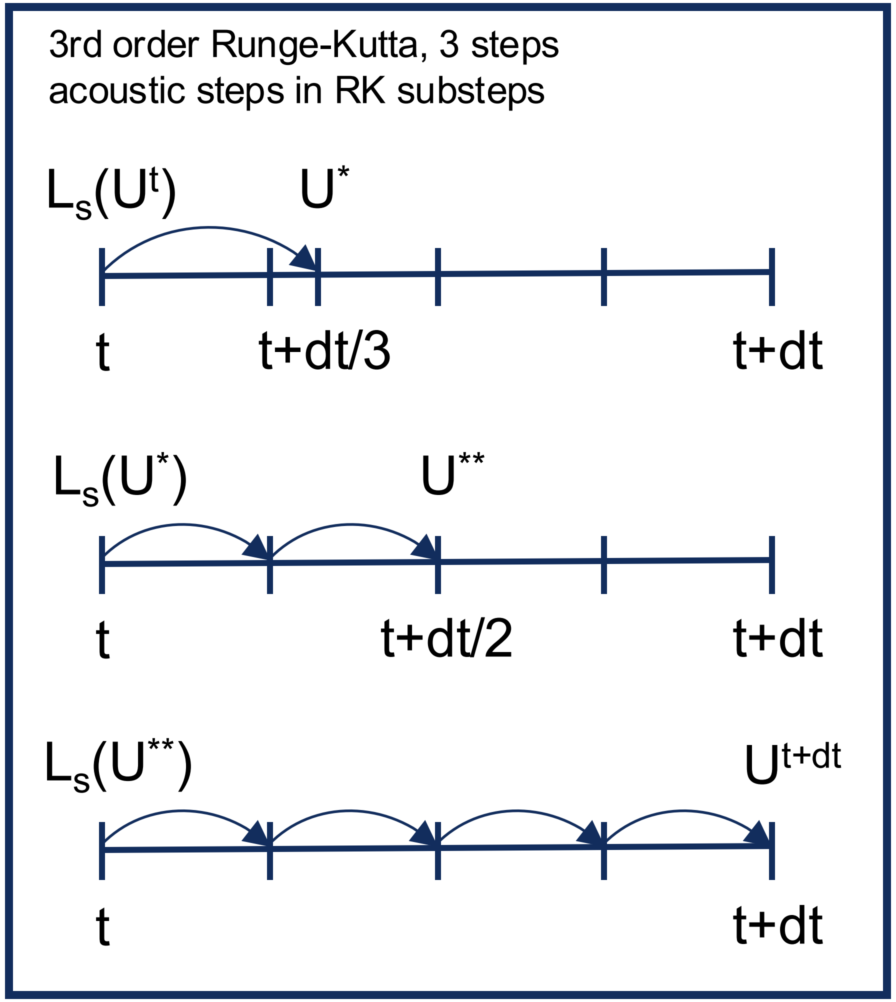
Time Integration: Acoustic Modes

$$U_t = L_{fast}(U) + L_{slow}(U)$$

Split-explicit time integration

fast: acoustic waves and gravity waves.
slow: everything else.

- RK3 is 3rd order accurate for linear eqns, 2nd order accurate for nonlinear eqns.
- Stable for centered and upwind advection schemes.
- Stable for Courant number Udt/dx < 1.73
- Three $L_{slow}(U)$ evaluations per timestep.



Time Integration

Default time integration

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```

Dynamics are integrated first (config_split_dynamics_transport = .true.), typically with multiple Runge-Kutta timesteps (dynamics_split_steps > 1)

Scalar transport is integrated separately, after the dynamics

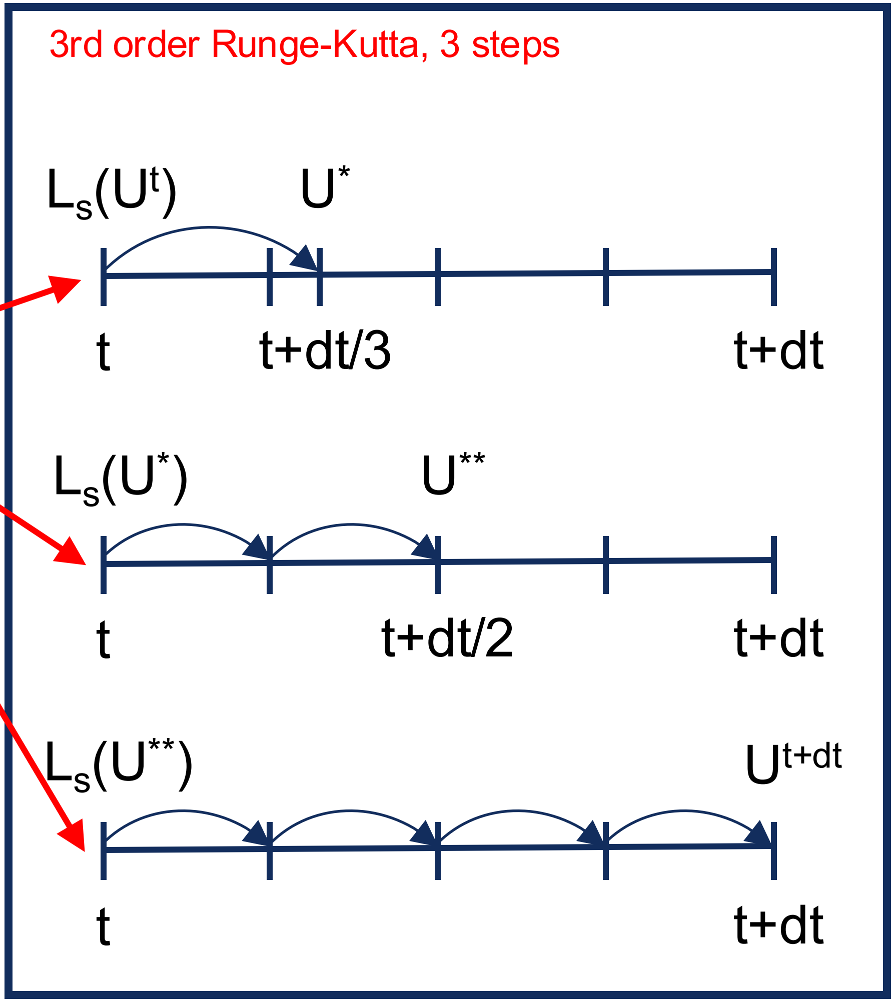
Time Integration

Default time integration

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```



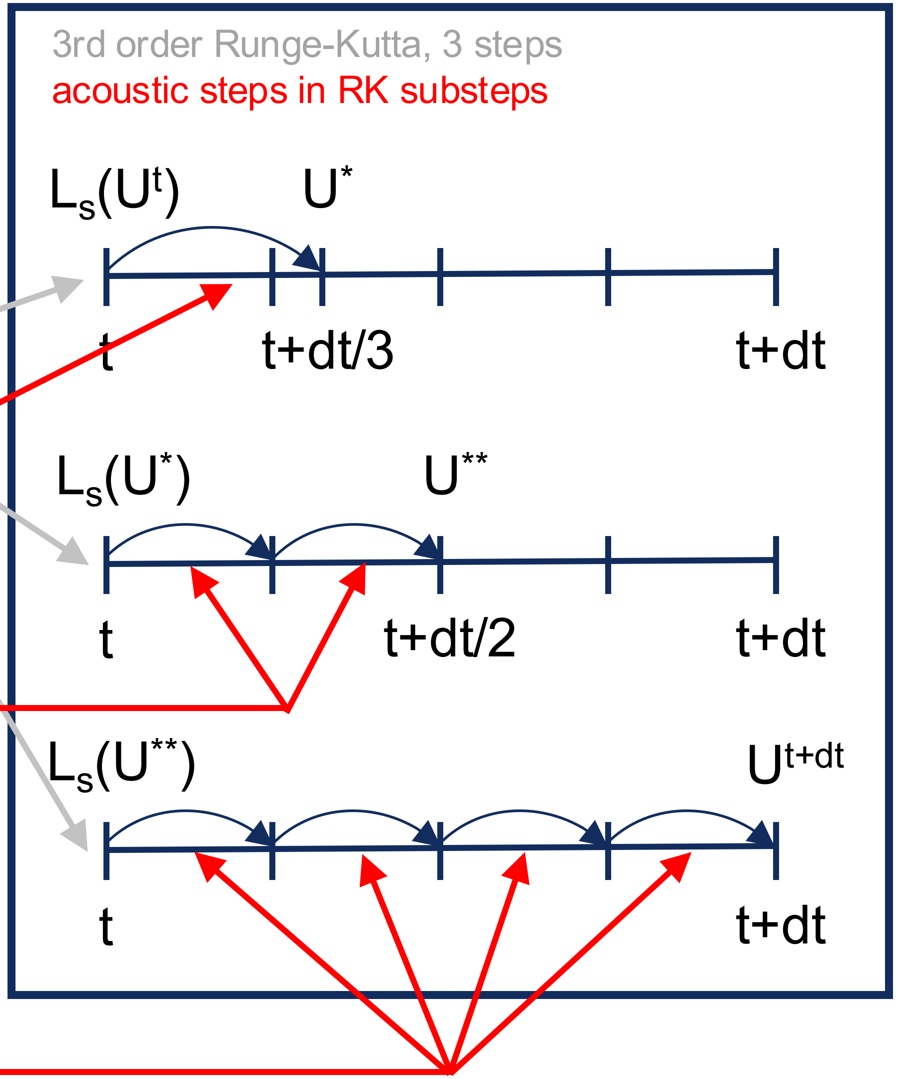
Time Integration

Default time integration

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```



Time Integration

MPAS
Model for Prediction Across Scales

Default time integration

3rd order Runge-Kutta, 3 steps
acoustic steps in RK substeps

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```

Forward-Backward acoustic mode time integration:

(1) Explicit integration of the horizontal momentum. There is stability constraint on the acoustic timestep.

(2) Implicit (in time) integration of the vertically-propagating acoustic modes and gravity waves. There is no stability constraint on the timestep. It uses the result from (1).

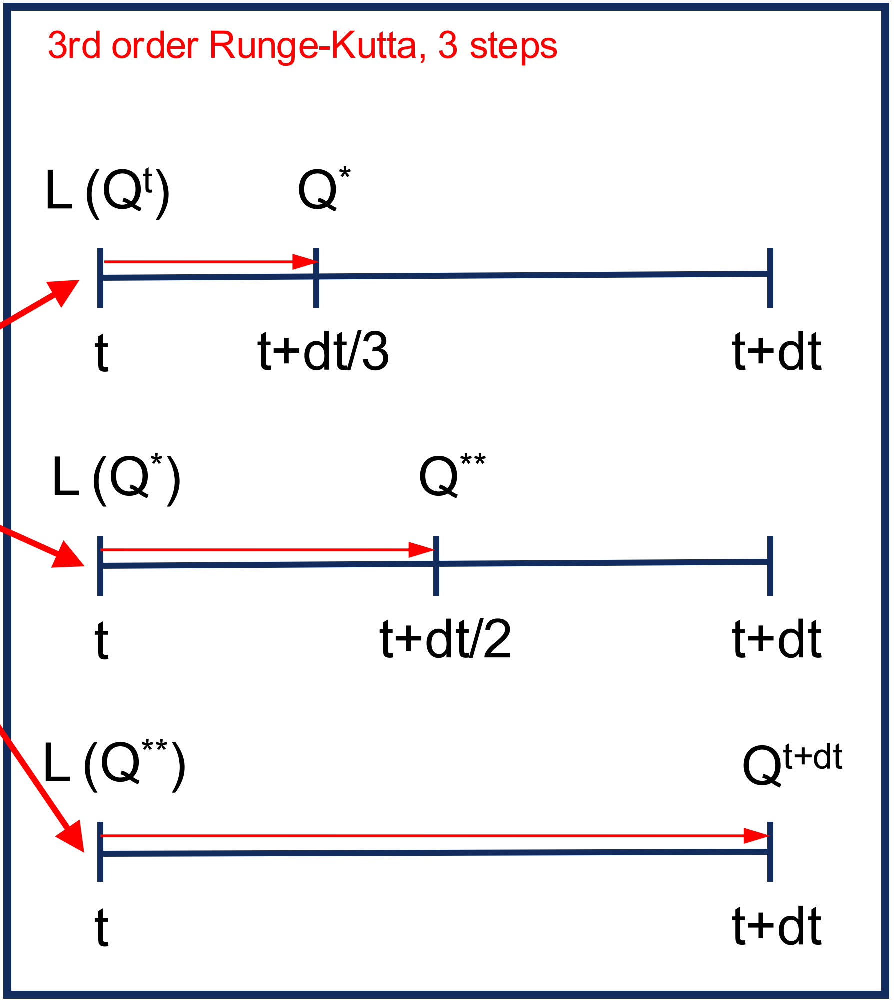
Time Integration

Default time integration

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```



Time Integration

Default time integration

```
Call physics

Do dynamics_split_steps
        Do rk3_step = 1, 3
                compute large-time-step tendency
                Do acoustic_steps
                        update u
                        update rho, theta and w
                End acoustic_steps
        End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
        scalar RK3 transport
End scalar_rk3_step
```

Allows for smaller dynamics timesteps relative to scalar transport timestep and the main physics timestep.

We can use any transport scheme here (we are not limited to RK3)
Scalar transport and physics are the expensive pieces in most applications.

Time Integration

MPAS Model for Prediction Across Scales

Default time integration

Call physics

Do dynamics_split_steps
    Do rk3_step = 1, 3
        compute large-time-step tendency
        Do acoustic_steps
            update u
            update rho, theta and w
        End acoustic_steps
    End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
    scalar RK3 transport
End scalar_rk3_step

&nhyd_model
config_dt = 90
config_start_time = ”2010-10-23_00:00:00"
config_run_duration = ”5_00:00:00”
config_split_dynamics_transport = true
config_dynamics_split_steps = 3
config_number_of_sub_steps = 2

Default time integration

In the file “namelist.atmosphere”

MPAS
Model for Prediction Across Scales

Time Integration

Default time integration

Call physics

Do dynamics_split_steps
    Do rk3_step = 1, 3
        compute large-time-step tendency
        Do acoustic_steps
            update u
            update rho, theta and w
        End acoustic_steps
    End rk3_step
End dynamics_split_steps

Do scalar_rk3_step = 1, 3
    scalar RK3 transport
End scalar_rk3_step

&nhyd_model
config_dt = 90
config_start_time = ”2010-10-23_00:00:00"
config_run_duration = ”5_00:00:00”
config_split_dynamics_transport = true
config_dynamics_split_steps = 3
config_number_of_sub_steps = 2

Default time integration

$$\Delta t \text{ (dynamics)} = \frac{\text{config\_dt}}{\text{config\_dynamics\_split\_steps}}$$

$$\Delta t \text{ (acoustic)} = \frac{\Delta t \text{ (dynamics)}}{\text{config\_number\_of\_sub\_steps}}$$

$$\Delta t \text{ (scalar transport)} = \text{config\_dt}$$

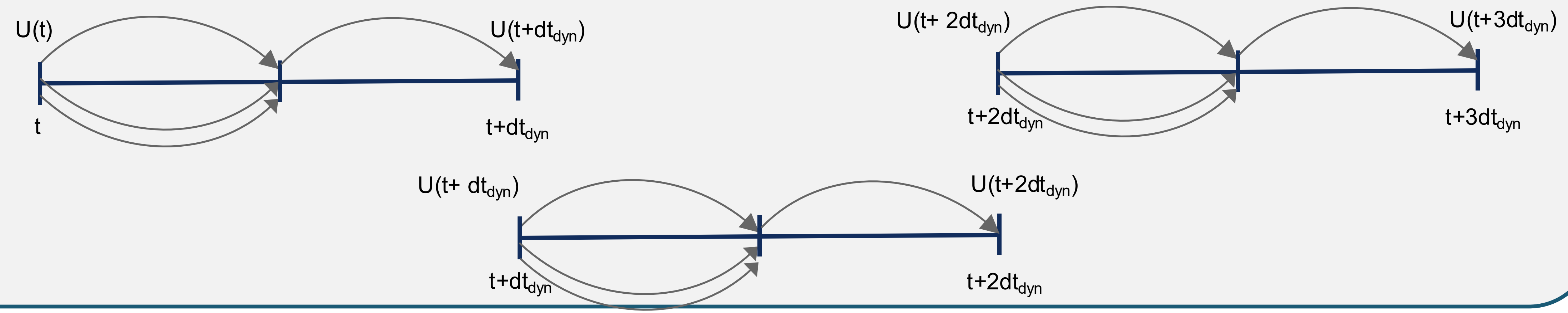
MPAS
Model for Prediction Across Scales

Time Integration

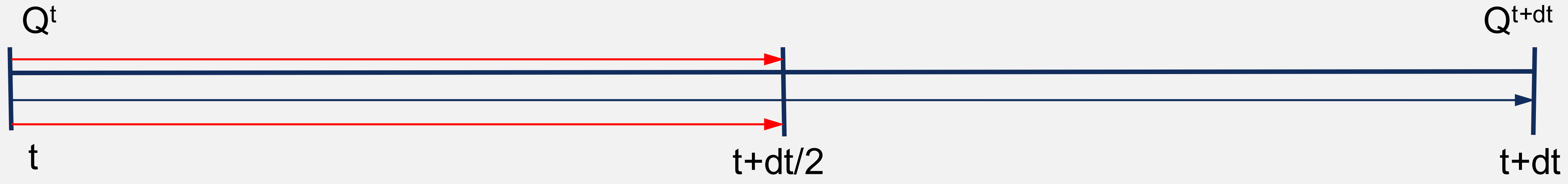
Default configuration summary

config_dynamics_split_steps = 3, config_number_of_sub_steps = 2,
config_time_integration_order = 2

Dynamics timestep



Scalar transport timestep



Time Integration
Option: The WRF approach

config_split_dynamics_transport = true/false
~~config_dynamics_split_steps = 3~~
config_number_of_sub_steps = 6
(acoustic_steps)

Time integration option in MPAS

```
Call physics

Do rk3_step = 1, 3
        compute large-time-step tendency
        Do acoustic_steps
                update u
                update rho, theta and w
        End acoustic_steps
        scalar RK3 transport
End rk3_step
```

Δt (dynamics) = Δt (scalar transport)
= config_dt

$$\Delta t \text{ (acoustic)} = \frac{\Delta t \text{ (dynamics)}}{\text{config\_number\_of\_sub\_steps}}$$

Time Integration

Testing the Timestep Configuration

&nhyd_model
config_dt = 90 ← *Timestep in seconds*

Similar to WRF, the model timestep (in seconds) initially should be set to be 6 times the finest nominal mesh spacing in km. For example – 15 km fine-mesh spacing would use a 90 second timestep.

We have found that a larger timestep is often stable.

If MPAS integrations become unstable (producing NaNs) after just a few timesteps, the issue may be the acoustic modes.

1) Reduce the main timestep (*config_dt*) and see if the simulations are stable.
2) If stable with a reduced timestep, try the original timestep with a reduced acoustic timestep: *config_number_of_sub_steps* > 2 (even integer)
3) The acoustic and dry dynamics timestep can also be reduced by increasing *config_dynamics_split_steps* > 3 (can be odd or even)
4) If none of these work, then the problem is likely not the dynamics. Check the initial conditions.

Time Integration
References

Runge-Kutta scheme:

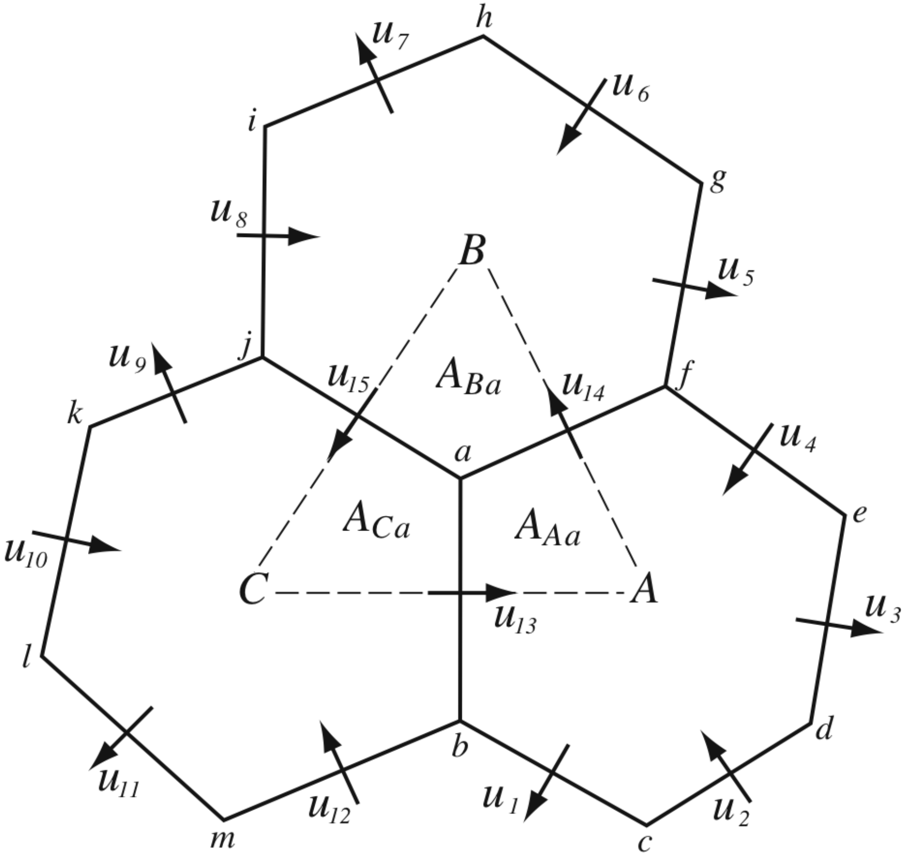
Wicker, L. J., and W. C. Skamarock, 2002: Time Splitting Methods for Elastic Models Using Forward Time Schemes. *Mon. Wea. Rev.,* **130**, 2088-2097. https://doi.org/10.1175/1520-0493(2002)130<2088:TSMFEM>2.0.CO;2

Detailed presentation on the acoustic time splitting:

Klemp. J. B., W. C. Skamarock, and J. Dudhia, 2007: Conservative Split-Explicit Time Integration Methods for the Compressible Nonhydrostatic Equations. *Mon. Wea. Rev.*, **135**, 2897-2913, doi:10.1175/MWR3440.1
(specifically section 2 and Appendix section (a) which deal with height-coordinate models, i.e. MPAS)

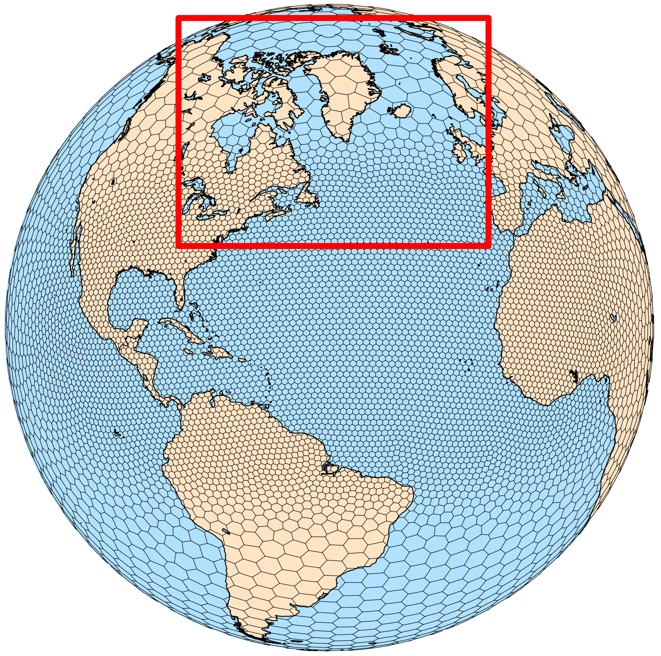
MPAS
Model for Prediction Across Scales

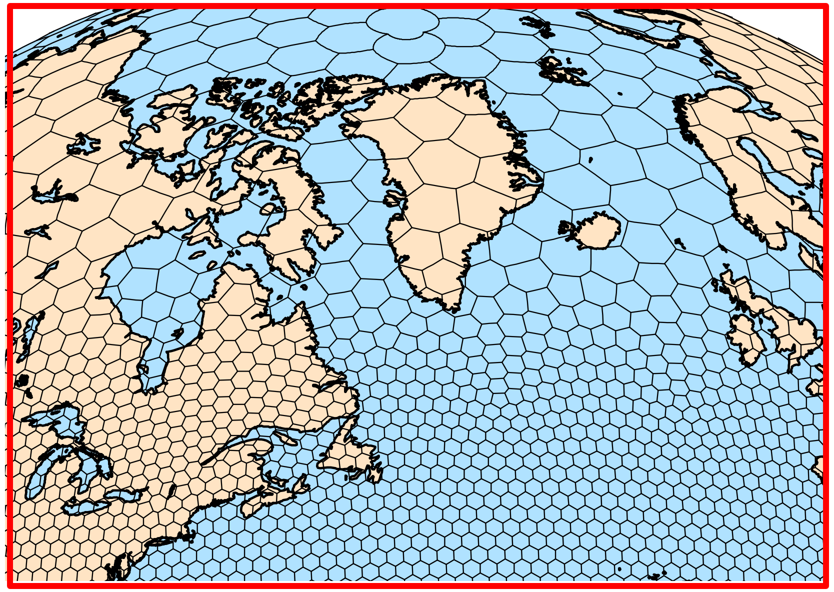
MPAS Horizontal Mesh



Unstructured spherical centroidal Voronoi meshes

- Mostly *hexagons*, some pentagons (5-sided cells) and heptagons (7-sided cells).
- Cell centers are at cell center-of-mass (centroidal).
- Cell edges bisect lines connecting cell centers; perpendicular.
- C-grid staggering of velocities (velocities are perpendicular to cell faces).
- Uniform resolution – traditional icosahedral mesh.





MPAS
Model for Prediction Across Scales

MPAS Nonhydrostatic Atmospheric Solver

Equations

- Prognostic equations for coupled variables.
- Generalized height coordinate.
- Horizontally vector-invariant equation set.
- Continuity equation for dry air mass.
- Thermodynamic equation for coupled potential temperature.

Variables: $(U, V, \Omega, \Theta, Q_j) = \tilde{\rho}_d\,(u, v, \omega, \theta, q_j)$ $\qquad \tilde{\rho}_d = \rho_d/\zeta_z$

Vertical coordinate: $z = \zeta + A(\zeta)h_s(x, y, \zeta)$

Prognostic equations:

$$\frac{\partial \mathbf{V}_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial \mathbf{z}_H p}{\partial \zeta}\right] - \eta\,\mathbf{k}\times\mathbf{V}_H - \mathbf{v}_H\nabla_\zeta\cdot\mathbf{V} - \frac{\partial \Omega \mathbf{v}_H}{\partial \zeta} - \rho_d\nabla_\zeta K + \mathbf{F}_{V_H}$$

$$\frac{\partial W}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\frac{\partial p}{\partial \zeta} + g\tilde{\rho}_m\right] - \left(\nabla\cdot\mathbf{v}\,W\right)_\zeta + F_W$$

$$\frac{\partial \Theta_m}{\partial t} = -\left(\nabla\cdot\mathbf{V}\,\theta_m\right)_\zeta + F_{\Theta_m}$$

$$\frac{\partial \tilde{\rho}_d}{\partial t} = -\left(\nabla\cdot\mathbf{V}\right)_\zeta$$

$$\frac{\partial Q_j}{\partial t} = -\left(\nabla\cdot\mathbf{V}\,q_j\right)_\zeta + F_{Q_j}$$

Gradient operators

Nonlinear Coriolis term

Diagnostics and definitions:

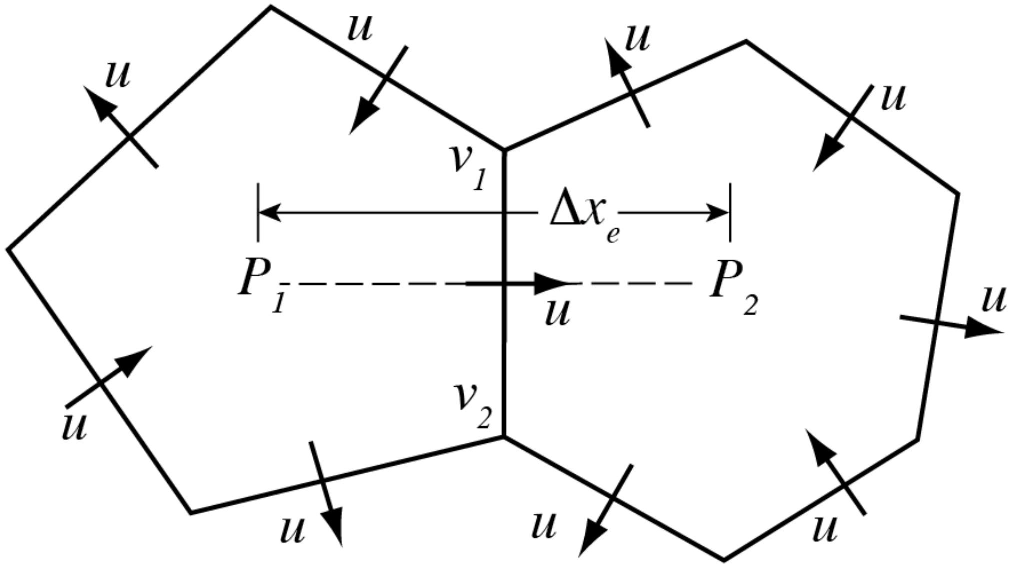
$$\frac{\rho_m}{\rho_d} = 1 + q_v + q_c + q_r + \ldots$$

$$p = p_0\left(\frac{R_d\zeta_z\Theta_m}{p_0}\right)^\gamma \qquad \theta_m = \theta[1 + (R_v/R_d)q_v]$$

Operators on the Voronoi Mesh
Pressure and KE gradients

$$\frac{\partial \mathbf{V}_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial \mathbf{z}_H p}{\partial \zeta}\right] - \eta\,\mathbf{k}\times\mathbf{V}_H - \mathbf{v}_H \nabla_\zeta\cdot\mathbf{V} - \frac{\partial \Omega \mathbf{v}_H}{\partial \zeta} - \rho_d \nabla_\zeta K + \mathbf{F}_{V_H}$$

On the Voronoi mesh, P_1P_2 is perpendicular to v_1v_2 and is bisected by v_1v_2, hence $P_x \sim (P_2\text{-}P_1)\Delta x_e^{-1}$ is 2nd order accurate.



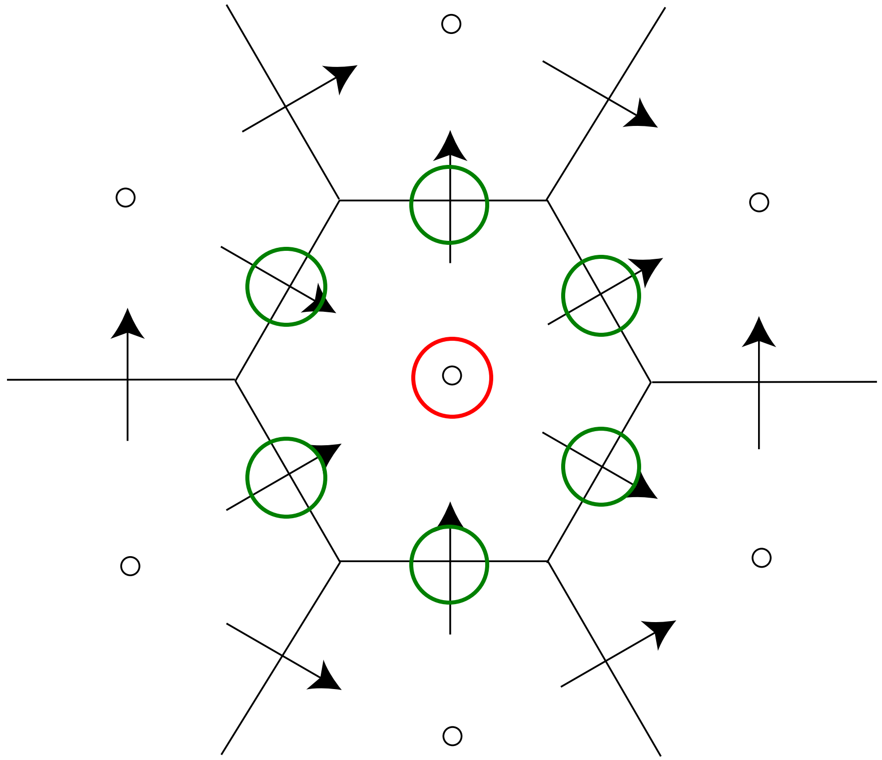
Operators on the Voronoi Mesh
Pressure and KE gradients

Cell center kinetic energy: KE_i

$$KE_i = (1-\beta) \sum_{e_i} w_{e_i} u_{e_i}^2 + \beta \sum_{v_j} w_{v_j} KE_{v_j}$$

Vertex kinetic energy: KE_v

$$KE_v = \sum_{e_v=1}^{3} w_{e_v} u_{e_v}^2$$



Operators on the Voronoi Mesh

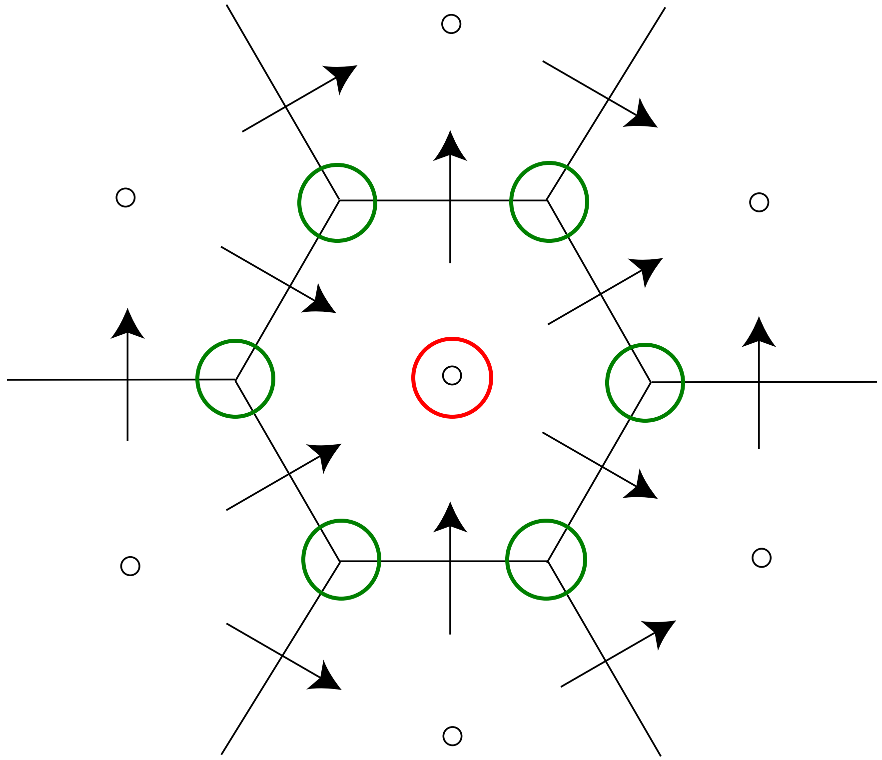
Pressure and KE gradients

Cell center kinetic energy: KE_i

$$KE_i = (1-\beta)\sum_{e_i} w_{e_i} u_{e_i}^2 + \beta \sum_{v_j} w_{v_j} KE_{v_j}$$

Vertex kinetic energy: KE_v

$$KE_v = \sum_{e_v=1}^{3} w_{e_v} u_{e_v}^2$$



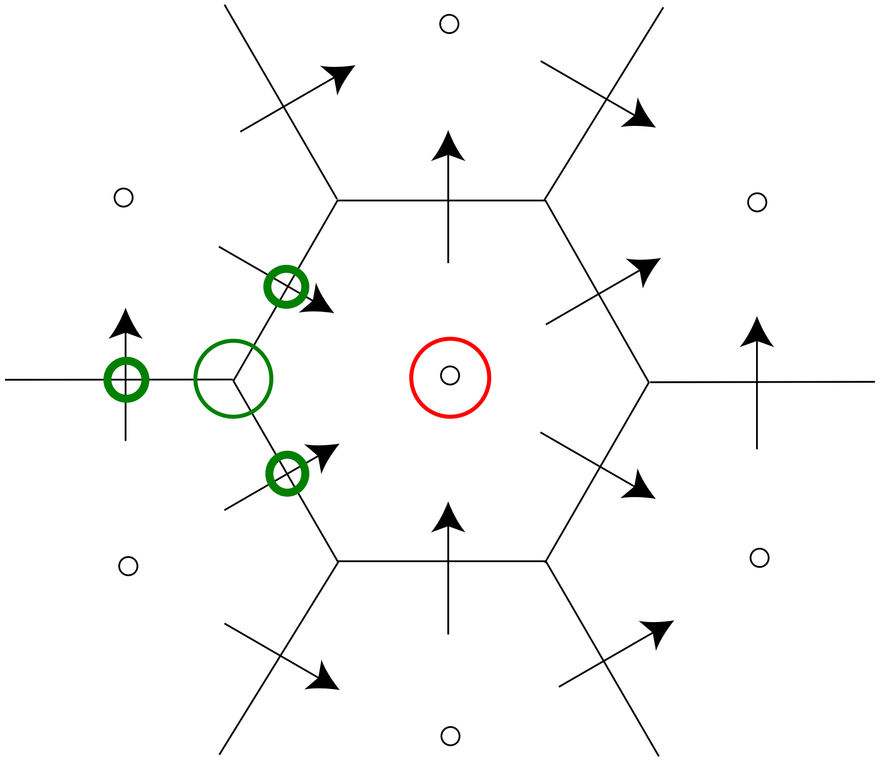
Operators on the Voronoi Mesh
Pressure and KE gradients

Cell center kinetic energy: KE_i

$$KE_i = (1-\beta)\sum_{e_i} w_{e_i} u_{e_i}^2 + \beta \sum_{v_j} w_{v_j} KE_{v_j}$$

Vertex kinetic energy: KE_v

$$KE_v = \sum_{e_v=1}^{3} w_{e_v} u_{e_v}^2$$



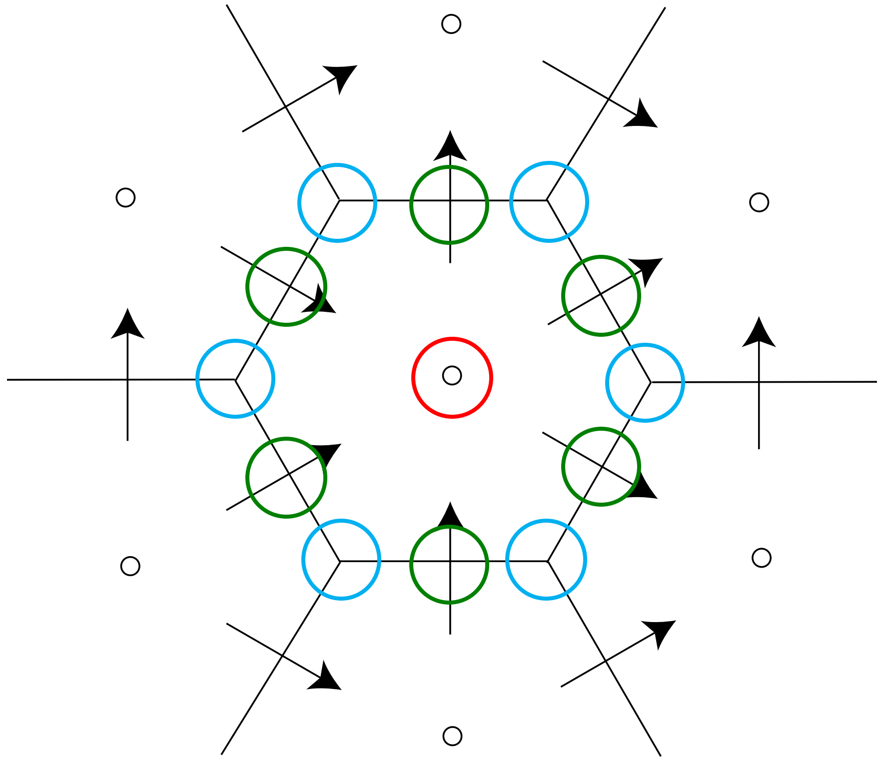
Operators on the Voronoi Mesh
Pressure and KE gradients

Cell center kinetic energy: KE_i

$$KE_i = (1-\beta)\sum_{e_i} w_{e_i} u_{e_i}^2 + \beta \sum_{v_i} w_{v_j} KE_{v_j}$$

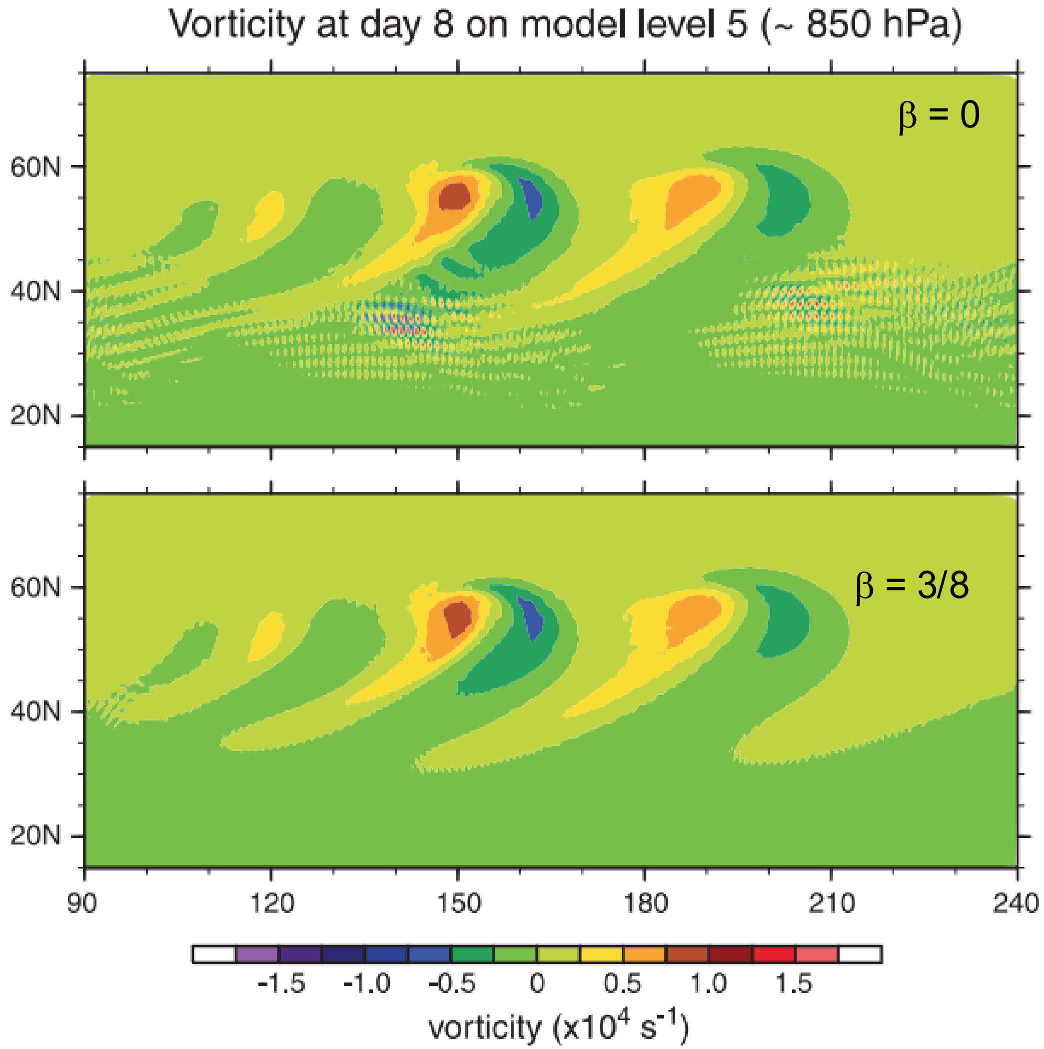
Vertex kinetic energy: KE_v

$$KE_v = \sum_{e_v=1}^{3} w_{e_v} u_{e_v}^2$$



Operators on the Voronoi Mesh
cell-center KE evaluation

MPAS uses $\beta = 3/8$



Operators on the Voronoi Mesh
'Nonlinear' Coriolis force

$$\frac{\partial \mathbf{V}_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial \mathbf{z}_H p}{\partial \zeta}\right] - \eta\,\mathbf{k}\times\mathbf{V}_H - \mathbf{v}_H \nabla_\zeta \cdot \mathbf{V} - \frac{\partial \Omega \mathbf{v}_H}{\partial \zeta} - \rho_d \nabla_\zeta K + \mathbf{F}_{V_H}$$

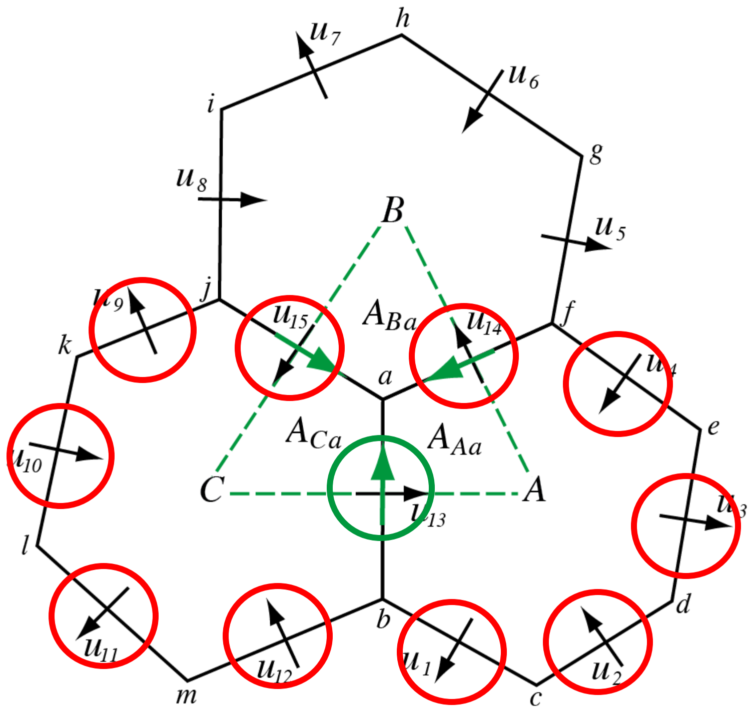
Tangential velocity reconstruction:

$$v_{e_i} = \sum_{j=1}^{n_{e_i}} w_{e_{i,j}} u_{e_{i,j}}$$

Nonlinear term:

$$[\eta\, \boldsymbol{k}\times V_H]_{e_i} = \sum_{j=1}^{n_{e_i}} \frac{1}{2}(\eta_{e_i} + \eta_{e_{i,j}})\, w_{e_{i,j}} \rho_{e_{i,j}} u_{e_{i,j}}$$

The general tangential velocity reconstruction produces a consistent divergence on the primal and dual grids, and allows for PV, enstrophy and energy* conservation in the nonlinear SW solver.



Operators on the Voronoi Mesh
'Nonlinear' Coriolis force

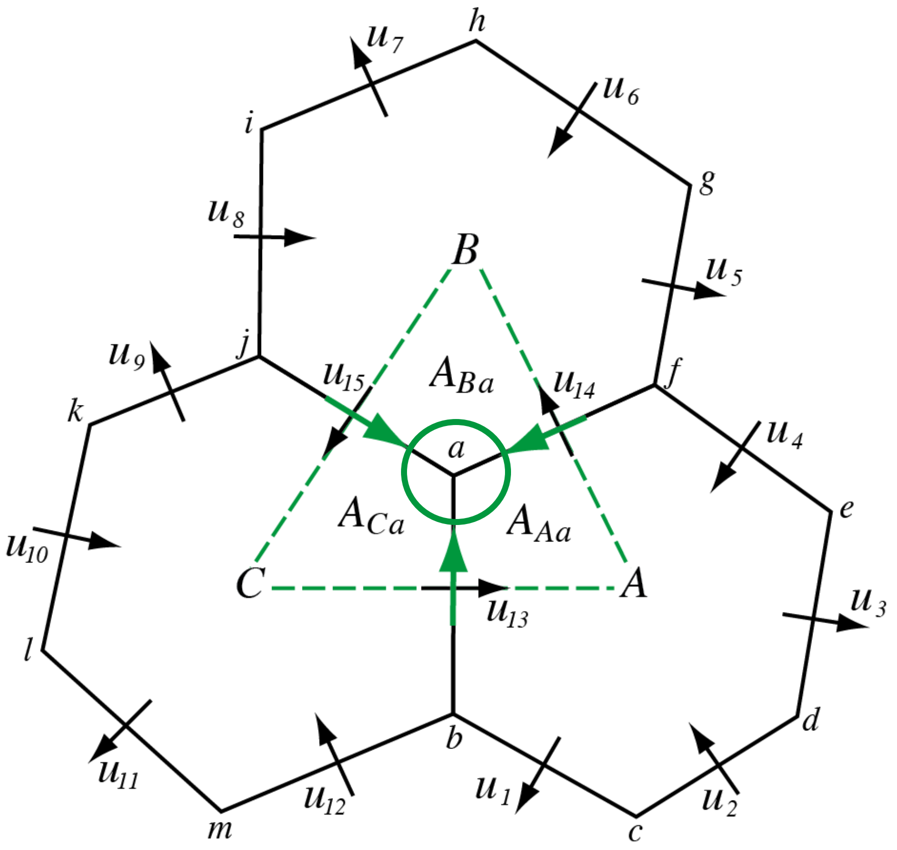
$$[\eta \boldsymbol{k} \times \mathbf{V}_H]_{e_i} = \sum_{j=1}^{n_{e_i}} \frac{1}{2}(\eta_{e_i} + \eta_{e_{i,j}}) w_{e_{i,j}} \rho_{e_{i,j}} u_{e_{i,j}}$$

Example: absolute vorticity at e_{13}

$$\eta_{13} = \frac{1}{2}(\eta_a + \eta_b)$$

Example: absolute vorticity at vertex a

$$\eta_a = f_a + \frac{\left(u_{13}\left|\overrightarrow{CA}\right| + u_{14}\left|\overrightarrow{AB}\right| + u_{15}\left|\overrightarrow{BC}\right|\right)}{Area(ABC)}$$



Configuring the dynamics

Upwind estimate of vorticity at the cell faces using a timestep of *config_apvm_upwinding* Δt

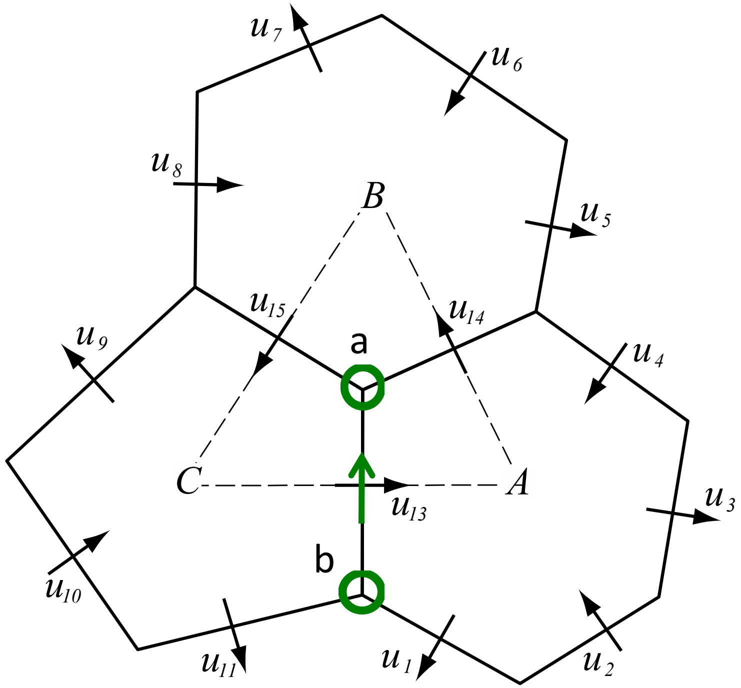
(namelist.atmosphere)

```
&nhyd_model
    config_apvm_upwinding = 0.5
```

$$\frac{\partial \mathbf{V}_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial \mathbf{z}_H p}{\partial \zeta}\right] - \eta\, \mathbf{k} \times \mathbf{V}_H$$
$$- \mathbf{v}_H \nabla_\zeta \cdot \mathbf{V} - \frac{\partial \Omega \mathbf{v}_H}{\partial \zeta} - \rho_d \nabla_\zeta K + \mathbf{F}_{V_H}$$

$$[\eta\, \boldsymbol{k} \times \boldsymbol{V}_H]_{e_i} = \sum_{j=1}^{n_{e_i}} \frac{1}{2}(\eta_{e_i} + \eta_{e_{i,j}})\, w_{e_{i,j}} \rho_{e_{i,j}} u_{e_{i,j}}$$

Vorticity at cell faces (at u points)



APVM: Anticipated Potential Vorticity Method (Sadourny 1985)

Upwinding the vorticity here will result in dissipation of the vorticity.

Vorticity at edge 13:

config_apvm_upwinding = 0, $\eta_{13} = (\eta_a + \eta_b)$

config_apvm_upwinding = 0.5, $\eta_{13} = (\eta_a + \eta_b) - 0.5\, \Delta t\, (u_e \eta_x - v_e\, \eta_y)$

MPAS
Model for Prediction Across Scales

Spatial Discretization in MPAS
references

Dynamics

Skamarock, W. C., J. B. Klemp, M. G. Duda, L. Fowler, S.-H. Park, and T. D. Ringler, 2012: A Multi-scale Nonhydrostatic Atmospheric Model Using Centroidal Voronoi Tesselations and C-Grid Staggering. Mon. Wea. Rev., 140, 30903105. doi:10.1175/MWR-D-11-00215.1

Ringler, T. D., J. Thuburn, J.B. Klemp, W. C. Skamarock, 2010: A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids. J. Comp. Phys., 229, 3065-3090. doi:10.1016/j.jcp.2009.12.007

MPAS
Model for Prediction Across Scales

The Most Important Takeaway from this Lecture

Testing the Timestep Configuration

```
&nhyd_model
    config_dt = 90
```

← *Timestep in seconds*

Similar to WRF, the model timestep (in seconds) initially should be set to be 6 times the finest nominal mesh spacing in km. For example – 15 km fine-mesh spacing would use a 90 second timestep.

We have found that *a larger timestep is often stable.*

If MPAS integrations become unstable (producing NaNs) after just a few timesteps, the issue may be the acoustic modes.

1) Reduce the main timestep (*config_dt*) and see if the simulations are stable.
2) If stable with a reduced timestep, try the original timestep with a reduced acoustic timestep: *config_number_of_sub_steps* > 2 (even integer)
3) The acoustic and dry dynamics timestep can also be reduced by increasing *config_dynamics_split_steps* > 3 (can be odd or even)
4) If none of these work, then the problem is likely not the dynamics. Check the initial conditions.