# Towards More Accurate and Efficient Integration of the UFS

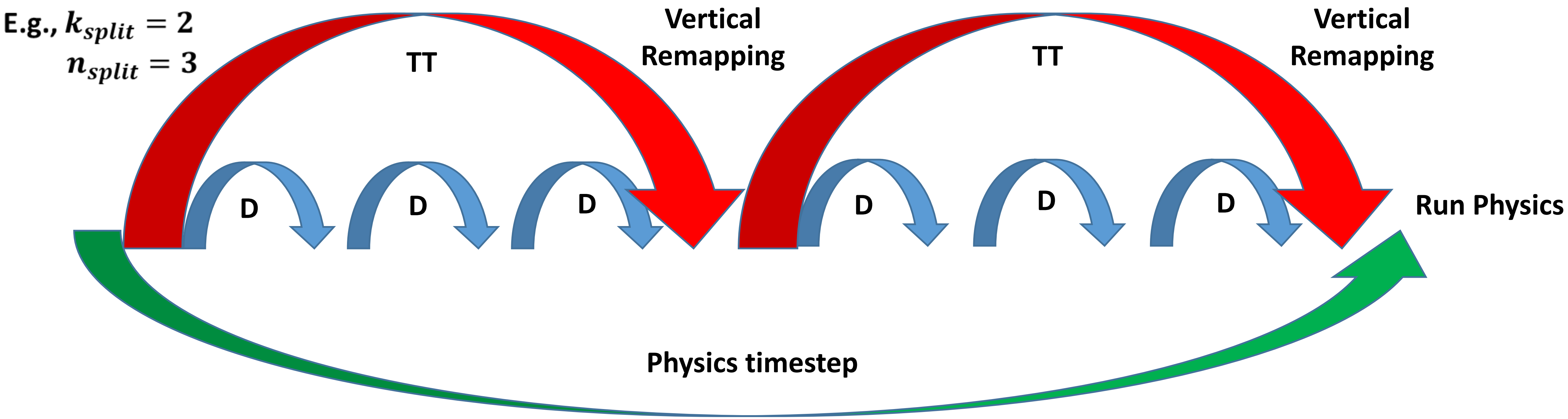
Kevin Viner

NOAA/NWS/NCEP/EMC

Physics and Dynamics Branch

# FV3 Time Stepping/Coupling

- Physics ($\Delta t_{atmos}$)
- Tracer Transport and Vertical Remapping ($\Delta t_{atmos}/k_{split}$)
- Dynamics ($\Delta t_{atmos}/\{k_{split} * n_{split}\}$)



# FV3 Stability Limits

- Horiztonal:
  - maximum stable $\Delta t_{dynamics} \sim \Delta x / (U + c_{acoustic})$

- Vertical: “No vertical CFL constraints”
  - A review by Griffies et al. 2020 (JAMES)
    - For the hydrostatic case:
      - “…there are no corresponding vertical CFL constraints for either tracer or velocity. Positive layer thicknesses are ensured for time-explicit methods if the horizontal (within layer) CFL constraint is maintained by the horizontal transport.”
    - For the non-hydrostatic case:
      - “The key condition is that one must enable regridding at a sufficiently fine time resolution to maintain monotonic vertical grid layering and thus to prevent grid singularities.”

| Application/Description | Dynamics time step (actual) | Maximum Stable Dynamics time step (linear analytic) |
|---|---|---|
| **HAFS_A**<br>Finest grid 2.0 km<br>k_split=4, n_split=10, dt_atmos=90 s, ptop=2 hPa | 2.25 seconds<br>(Equates to<br>U+c = 889 m/s) | 4.0 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **HAFS_B**<br>Finest grid 2.0 km<br>k_split=4, n_split=9, dt_atmos=90 s, ptop=2 hPa | 2.50 seconds<br>(Equates to<br>U+c = 800 m/s) | 4.0 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **RRFS_A**<br>Finest grid 3.0 km<br>k_split=2, n_split=5, dt_atmos=36 s, ptop=2 hPa | 3.60 seconds<br>(Equates to<br>U+c = 833 m/s) | 6 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **RRFS_B**<br>Finest grid 3.0 km<br>k_split=6, n_split=3, dt_atmos=60 s, ptop=2 hPa | 3.33 seconds<br>(Equates to<br>U+c = 909 m/s) | 6 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **GFS v15**<br>Finest grid 13.0 km (C768)<br>k_split=2, n_split=6, dt_atmos=225 s, ztop~55 km | 18.75 seconds<br>(Equates to<br>U+c = 693 m/s) | 22 seconds<br>(Assumes max<br>U+c = 600 m/s) |
| **GFS v16**<br>Finest grid 13.0 km (C768)<br>k_split=2, n_split=6, dt_atmos=150 s, ztop~80 km | 12.50 seconds<br>(Equates to<br>U+c = 1040 m/s) | 22 seconds<br>(Assumes max<br>U+c = 600 m/s) |

# What’s causing this inefficiency?

- Most cases work fine with the analytic time step size, but our operational systems need to be “bullet proof”
  - This requires an ~40% reduction in the acoustic time step size
  - For the GFS v16 application this equates to a potential ~25% reduction in atmospheric model efficiency

- Investigation has linked the crashes to non-monotonicity of LVC at the model top (we’re violating the Lagrangian stability condition)

- Recent discussions with NASA (FV3-based GEOS model) appear to reveal this is a known issue associated with the vertical remapping method being used in the UFS
  - Further investigation is required

# How often should we perform vertical remapping?

- - Most people I've talked to are fairly confused by the concept of the $k_{split}$ variable; they don't really understand how often vertical remapping is necessary or why it's necessary

  - A review by Griffies et al. 2020 (JAMES)
    - For the non-hydrostatic case:
      - "The key condition is that one must enable regridding at a <u>sufficiently fine time resolution</u> to maintain monotonic vertical grid layering and thus to prevent grid singularities."

  - What is sufficient? How do we know a priori?
    - We don't know! We don't!

  - Why don't we just test for monotonicity/layer thickness on the fly to trigger remapping steps?

| Application/Description | Dynamics time step (actual) | Maximum Stable Dynamics time step (linear analytic) |
|---|---|---|
| **HAFS_A**<br>Finest grid 2.0 km<br>k_split=4, n_split=10, dt_atmos=90 s, ptop=2 hPa | 2.25 seconds<br>(Equates to<br>U+c = 889 m/s) | 4.0 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **HAFS_B**<br>Finest grid 2.0 km<br>k_split=4, n_split=9, dt_atmos=90 s, ptop=2 hPa | 2.50 seconds<br>(Equates to<br>U+c = 800 m/s) | 4.0 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **RRFS_A**<br>Finest grid 3.0 km<br>k_split=2, n_split=5, dt_atmos=36 s, ptop=2 hPa | 3.60 seconds<br>(Equates to<br>U+c = 833 m/s) | 6 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **RRFS_B**<br>Finest grid 3.0 km<br>k_split=6, n_split=3, dt_atmos=60 s, ptop=2 hPa | 3.33 seconds<br>(Equates to<br>U+c = 909 m/s) | 6 seconds<br>(Assumes max<br>U+c = 500 m/s) |
| **GFS v15**<br>Finest grid 13.0 km (C768)<br>k_split=2, n_split=6, dt_atmos=225 s, ztop~55 km | 18.75 seconds<br>(Equates to<br>U+c = 693 m/s) | 22 seconds<br>(Assumes max<br>U+c = 600 m/s) |
| **GFS v16**<br>Finest grid 13.0 km (C768)<br>k_split=2, n_split=6, dt_atmos=150 s, ztop~80 km | 12.50 seconds<br>(Equates to<br>U+c = 1040 m/s) | 22 seconds<br>(Assumes max<br>U+c = 600 m/s) |

# Adaptive (in time) Remapping

- To start, set up time integrator as “while loop” instead of “for loop”
- Two options:
  - 1) Set a lower limit for layer thickness; if met, trigger a remapping step once the current dynamics step is finished (not guaranteed to finish the step)
  - 2) Store a copy of the state at the beginning of the dynamics step; if monotonicity is broken, replace the state, trigger a remapping step, and try again

- My hypothesis is that the necessary remapping frequency (for stability) is dependent upon a number of factors: grid resolution, terrain, flow characteristics, etc…it seems likely that for most relatively coarse resolution applications these issues are alleviated ($k_{split} \sim 1$)
  - Based on current UFS application values of $k_{split}$, this could save significant runtime

# Physics-Dynamics Coupling

**higher-order physics-dynamics coupling**

- $$\frac{x^{n+1} - x^n}{\Delta t} \cong D + P$$

- The model attempts a tighter coupling between the dynamics and physics in which they are integrated together (in some fashion) to boost the global accuracy of the system beyond first order.

**first-order physics-dynamics coupling, aka, time splitting**

- $$\frac{\tilde{x} - x^n}{\Delta t} = D$$

$$\frac{x^{n+1} - \tilde{x}}{\Delta t} = P$$

- The physics and dynamics are performed separately ("black boxes") with minimal communication. The global system converges at first order.

**Note that all fluxes from other components (ocean, land, ice, wave, etc) are or would be represented in the **P** operator

# “Dribbling” vs “Chunking”

**Dribbling (proposed) coupling**

- Physics (excluding microphysics) is run at the beginning of the time step (before dynamics); tendencies are stored
- Dynamics is integrated forward $\Delta t_{atmos}$. The physics tendencies are treated as static forcing terms on the RHS of the prognostic equations; they are applied at every “dynamics time step”
- After the dynamics the microphysics is called; the full microphysics tendencies are added to complete the time step

- The coupling frequency is >= the physics frequency

- GFS v16 coupling frequency would be 12.50 s

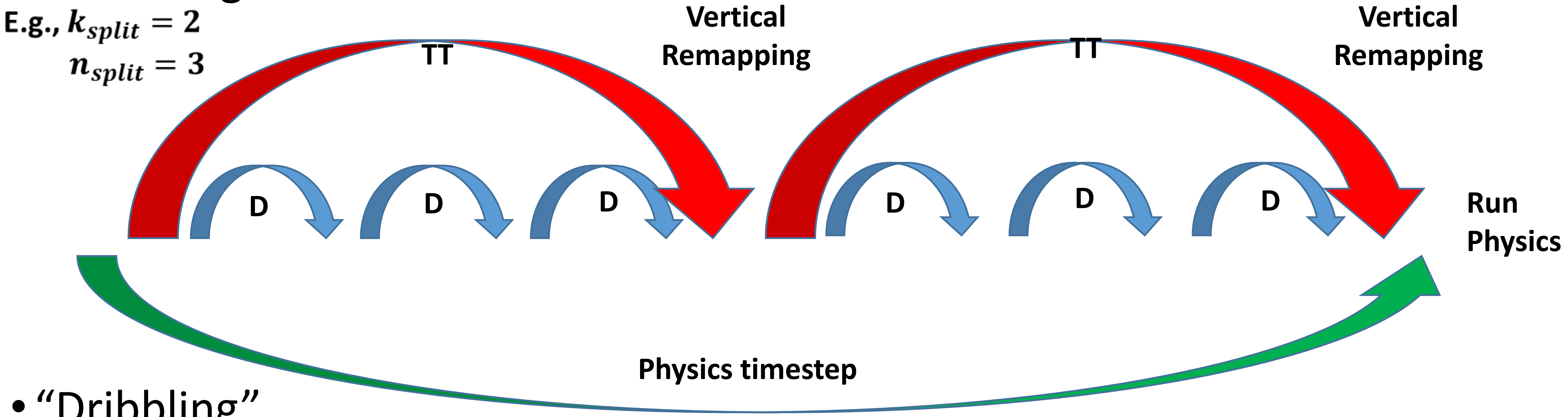
**Chunking (control) coupling**

- Dynamics is integrated forward $\Delta t_{atmos}$ (“physics time step”) by way of several “dynamics time steps”
- The full physics package is called (with microphysics last)
- Physics tendencies are added as a single increment to complete the time step

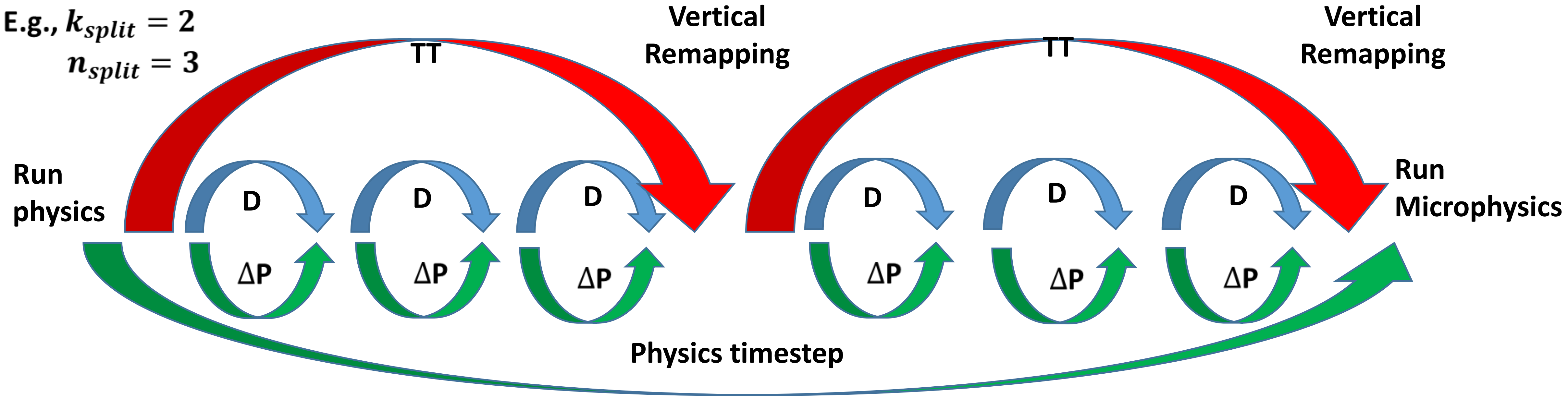
- The physics and coupling frequencies are the same

- GFS v16 coupling frequency is 150 s

- “Chunking”




- “Dribbling”



suite_FV3_HRRR

```xml
<group name="physics">
  <subcycle loop="1">
    <scheme>GFS_suite_interstitial_phys_reset</scheme>
    <scheme>GFS_suite_stateout_reset</scheme>
    <scheme>get_prs_fv3</scheme>
    <scheme>GFS_suite_interstitial_1</scheme>
    <scheme>GFS_surface_generic_pre</scheme>
    <scheme>GFS_surface_composites_pre</scheme>
    <scheme>dcyc2t3</scheme>
    <scheme>GFS_surface_composites_inter</scheme>
    <scheme>GFS_suite_interstitial_2</scheme>
  </subcycle>
  <!-- Surface iteration loop -->
  <subcycle loop="2">
    <scheme>mynnsfc_wrapper</scheme>
    <scheme>GFS_surface_loop_control_part1</scheme>
    <scheme>lsm_ruc</scheme>
    <scheme>flake_driver</scheme>
    <scheme>GFS_surface_loop_control_part2</scheme>
  </subcycle>
  <!-- End of surface iteration loop -->
  <subcycle loop="1">
    <scheme>GFS_surface_composites_post</scheme>
    <scheme>sfc_diag</scheme>
    <scheme>sfc_diag_post</scheme>
    <scheme>GFS_surface_generic_post</scheme>
    <scheme>rrfs_smoke_wrapper</scheme>
    <scheme>mynnedmf_wrapper</scheme>
    <scheme>rrfs_smoke_postpbl</scheme>
    <scheme>GFS_GWD_generic_pre</scheme>
    <scheme>drag_suite</scheme>
    <scheme>GFS_GWD_generic_post</scheme>
    <scheme>GFS_suite_stateout_update</scheme>
    <scheme>ozphys_2015</scheme>
    <scheme>h2ophys</scheme>
    <scheme>get_phi_fv3</scheme>
    <scheme>GFS_suite_interstitial_3</scheme>
    <scheme>GFS_suite_interstitial_4</scheme>
    <scheme>GFS_MP_generic_pre</scheme>
    <scheme>mp_thompson_pre</scheme>
    <scheme>mp_thompson</scheme>
    <scheme>mp_thompson_post</scheme>
    <scheme>GFS_MP_generic_post</scheme>
    <scheme>maximum_hourly_diagnostics</scheme>
    <scheme>phys_tend</scheme>
  </subcycle>
</group>
```

suite_FV3_HRRR_pdc

```xml
<group name="microphysics">
  <subcycle loop="1">
    <scheme>GFS_suite_stateout_reset</scheme>
    <scheme>get_prs_fv3</scheme>
    <scheme>get_phi_fv3</scheme>
    <scheme>GFS_suite_interstitial_3</scheme>
    <scheme>GFS_suite_interstitial_4</scheme>
    <scheme>GFS_MP_generic_pre</scheme>
    <scheme>mp_thompson_pre</scheme>
    <scheme>mp_thompson</scheme>
    <scheme>mp_thompson_post</scheme>
    <scheme>GFS_MP_generic_post</scheme>
    <scheme>maximum_hourly_diagnostics</scheme>
    <scheme>phys_tend</scheme>
  </subcycle>
</group>
<group name="physics">
  <subcycle loop="1">
    <scheme>GFS_suite_interstitial_phys_reset</scheme>
    <scheme>GFS_suite_stateout_reset</scheme>
    <scheme>get_prs_fv3</scheme>
    <scheme>GFS_suite_interstitial_1</scheme>
    <scheme>GFS_surface_generic_pre</scheme>
    <scheme>GFS_surface_composites_pre</scheme>
    <scheme>dcyc2t3</scheme>
    <scheme>GFS_surface_composites_inter</scheme>
    <scheme>GFS_suite_interstitial_2</scheme>
  </subcycle>
  <!-- Surface iteration loop -->
  <subcycle loop="2">
    <scheme>mynnsfc_wrapper</scheme>
    <scheme>GFS_surface_loop_control_part1</scheme>
    <scheme>lsm_ruc</scheme>
    <scheme>flake_driver</scheme>
    <scheme>GFS_surface_loop_control_part2</scheme>
  </subcycle>
  <!-- End of surface iteration loop -->
  <subcycle loop="1">
    <scheme>GFS_surface_composites_post</scheme>
    <scheme>sfc_diag</scheme>
    <scheme>sfc_diag_post</scheme>
    <scheme>GFS_surface_generic_post</scheme>
    <scheme>rrfs_smoke_wrapper</scheme>
    <scheme>mynnedmf_wrapper</scheme>
    <scheme>rrfs_smoke_postpbl</scheme>
    <scheme>GFS_GWD_generic_pre</scheme>
    <scheme>drag_suite</scheme>
    <scheme>GFS_GWD_generic_post</scheme>
    <scheme>GFS_suite_stateout_update</scheme>
    <scheme>ozphys_2015</scheme>
    <scheme>h2ophys</scheme>
  </subcycle>
</group>
```

I've done my best to avoid modifying the code inside atmos_cubed_sphere, but obviously changes were necessary. From a user perspective you need to know 3 things:

1) When you compile, use {suite_FV3_HRRR, suite_FV3_HRRR_pdc}
2) Add higher_order_pdc = .true. to atmos_model_nml
3) Change ccpp_suite = 'FV3_HRRR_pdc'

```
&atmos_model_nml
    avg_max_length = 3600.
    blocksize = 32
    ccpp_suite = 'FV3_HRRR_pdc'
    chksum_debug = .false.
    dycore_only = .false.
    ignore_rst_cksum = .true.
    higher_order_pdc = .true.
/
```

Many of the changes live in the top level FV3 code, e.g., module_fcst_grid_comp.F90:

```
!-----------------------------------------------------------------------
! *** call fcst integration subroutines

      if(Atmos%higher_order_pdc)then
        call update_atmos_radiation_physics (Atmos)

        call update_atmos_model_tendencies (Atmos, rc=rc)
        if (ESMF_LogFoundError(rcToCheck=rc, msg=ESMF_LOGERR_PASSTHRU, line=__LINE__, file=__FILE__)) return

        call update_atmos_model_dynamics (Atmos)

        call update_atmos_microphysics (Atmos)
      else
        call update_atmos_model_dynamics (Atmos)

        call update_atmos_radiation_physics (Atmos)
      endif

      call atmos_model_exchange_phase_1 (Atmos, rc=rc)
      if (ESMF_LogFoundError(rcToCheck=rc, msg=ESMF_LOGERR_PASSTHRU, line=__LINE__, file=__FILE__)) return

      if (mype == 0) write(*,'(A,I16,A,F16.6)')'PASS: fcstRUN phase 1, n_atmsteps = ', &
                                               n_atmsteps,' time is ',mpi_wtime()-tbeg1
!
!-----------------------------------------------------------------------
```

# Some details of the “Dribbling” implementation in FV3 (latest iteration)

- I am working on a new iteration:
  - Inline_q = .false.
  - Apply tendencies in native dynamic variables (where possible)
  - Physics tendencies are held fixed in time and horizontal, they are advected as scalars in the vertical using Lagrangian advection (this looks like a mass re-scaling) such that we don’t have to worry about boundary conditions.
  - The tendencies are applied on the C-grid and D-grid. This is done after the horizontal advection and requires further halo communication.
  - The tracer advection is in some ways decoupled from the dynamics across dt_atmos to prevent negatives and maintain conservation of dry mass
    - It was decoupled across k_split previously
    - The tracers will get the physics tendencies at the beginning of the time step (“chunking”) and are still advected at k_split, but the tracer proxy variables (q_con, kappa) will evolve independently across dt_atmos. This is then resolved at the end of the time step by computing new proxy values using the fully updated tracers; they are then used to convert theta_m to temperature prior to physics.

- Potential Deficiencies:
  - Cost: hopefully < 10% added cost
  - My results say vertical interpolation vs vertical advection of physics tendencies is not significantly different, but I’d like some outside opinions on this
  - How different are the proxy variables by the end of a time step? Have we traded a physics coupling problem for a tracer coupling problem?

# Conclusions

- It appears possible that we may be able to achieve greater stability, efficiency, and accuracy in the UFS with these three changes to the atmospheric model

  - Efficiency gains >25% may be possible, depending on the dynamics cost in the application
    - Could have implications for increasing ensemble sizes or decreasing computational resources

  - Accuracy gains from “dribbling” should increase with increasing $\Delta t_{atmos}$ , which also depends on application (largest for GFS and S2S). However, unless the coupling is prohibitively expensive or intractable, I don’t see a good reason to leave accuracy on the table