An Adaptive Time-step for Increased Model Efficiency

Todd Hutchinson
WSI
May 8, 2007

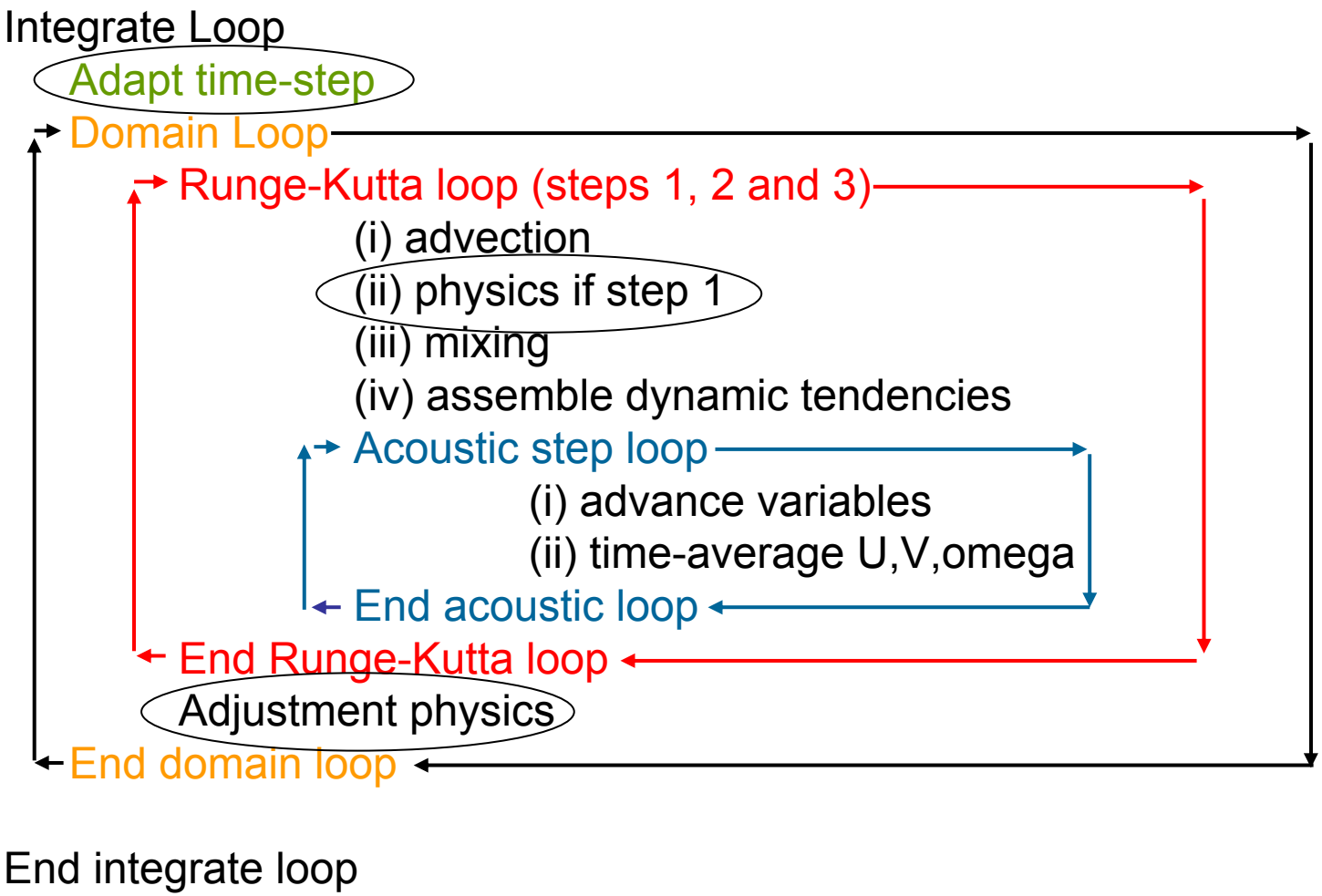
Background: WRF-ARW Time-Step

- Runge-Kutta 3rd order
- User-configurable via namelist parameter (time_step)
- Tradeoff:
  - Longer time-step = faster run-time
  - Shorter time-step = increased model stability
- Recommended time-step:
  - 6*dx
  - Conservative in order to assure stability
- Remains constant throughout a model run

Adapting Runge-Kutta: Overview

- Automatically adapt throughout a simulation
- Adapt to maximum value that can support underlying horizontal and vertical motions
- Adaptation assures stability
- Total Run-time reduced if average time-step exceeds 6*dx

Adapting Runge-Kutta: Code Modifications



Namelist entries

- To engage adaptive time-step, set time_step to -1, then

Namelist entry	**Symbol**	**Default**
min_time_step	dt_{min}	0
max_time_step	dt_{max}	3*dt_{start}
target_cfl	c_{target}	1.2
max_step_increase_pct	dt_{max_inc}	5
starting_time_step	dt_{start}	6*dx
step_to_output_time		.false.

Calculate time-step: Step 1: First guess

- Calculate max CFL over entire 3-D grid

  $C_{maxh} = U \cdot \Delta t / \Delta x$ $\quad$ $C_{maxv} = w \cdot \Delta t / \Delta z$

  C_{max}= maximum (C_{maxh} , C_{maxv})

  For C_{max} < ~1.2 model remains stable (Wicker and Skamarock, 2002)

- Adjust time-step so maximum CFL = target CFL

If $C_{max} < C_{target}$

$dt_n = (C_{target} / C_{max}) * dt_{n-1}$ } Increase dt

else } Decrease dt by 1.5

$dt_n = ((C_{target} - 0.5 * (C_{max} - C_{target})) / C_{max}) * dt_{n-1}$

Calculate time-step: Step 2: Limit time-step

- Limit time-step adaptation based on user (namelist) input

If $dt_n > \mathbf{dt_{max\_inc}} * dt_{n-1}$ then

$dt_n = \mathbf{dt_{max\_inc}} * dt_{n-1}$

} Limit increase of dt

If $dt_n > \mathbf{dt_{max}}$ then

$dt_n = \mathbf{dt_{max}}$

} Limit maximum dt

If $dt_n < \mathbf{dt_{min}}$ then

$dt_n = \mathbf{dt_{min}}$

} Limit minimum dt

Calculate time-step: Step 3: Step to output

- If user has selected to “step to output”, adjust time-step to fall exactly upon an output time:

If $t_{output} > dt_n$ and $t_{output} < 2*dt_n$ then

$dt_n = 0.5*t_{output}$

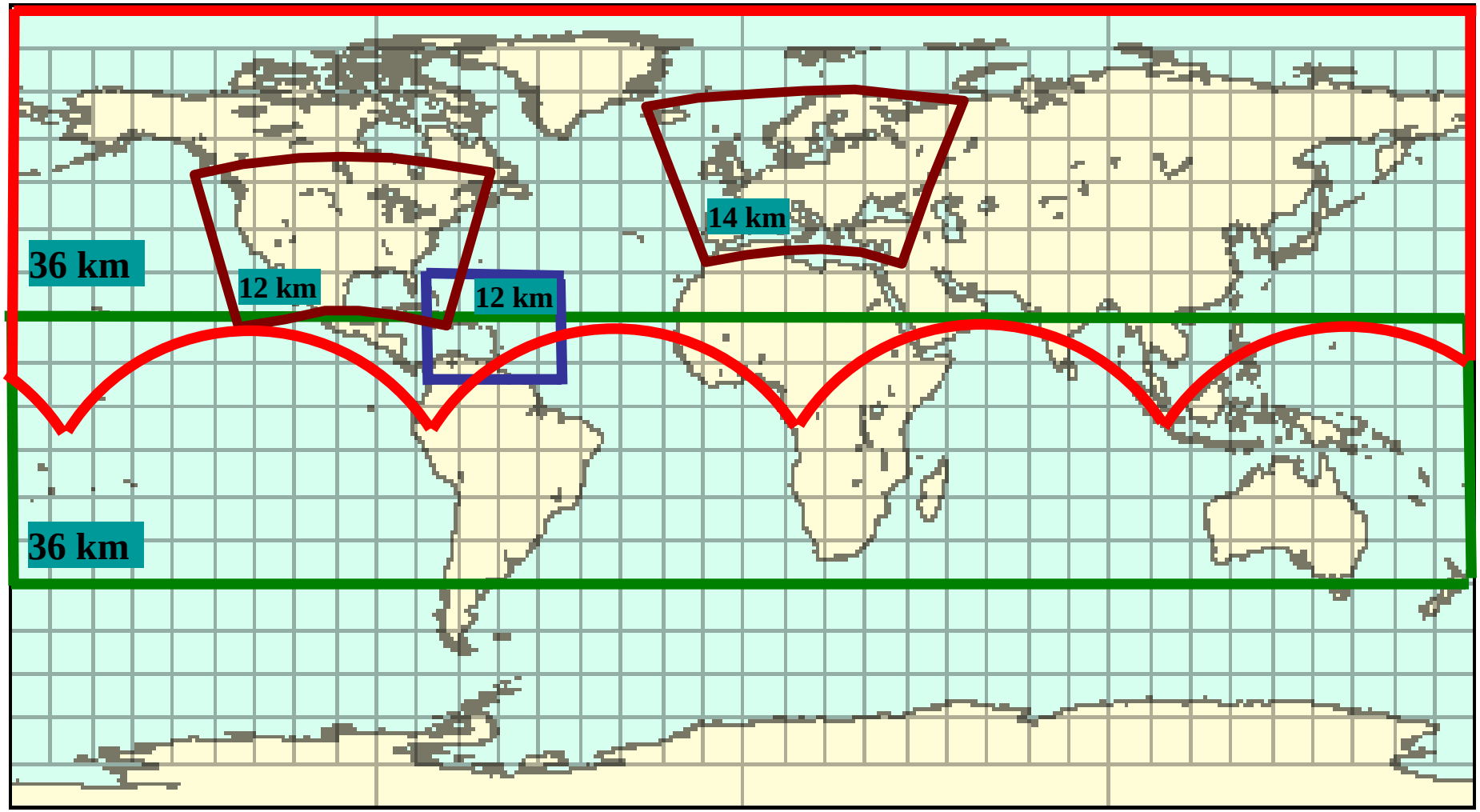
} 1-2 steps from output

Else if $dt_n \leq t_{output}$ then

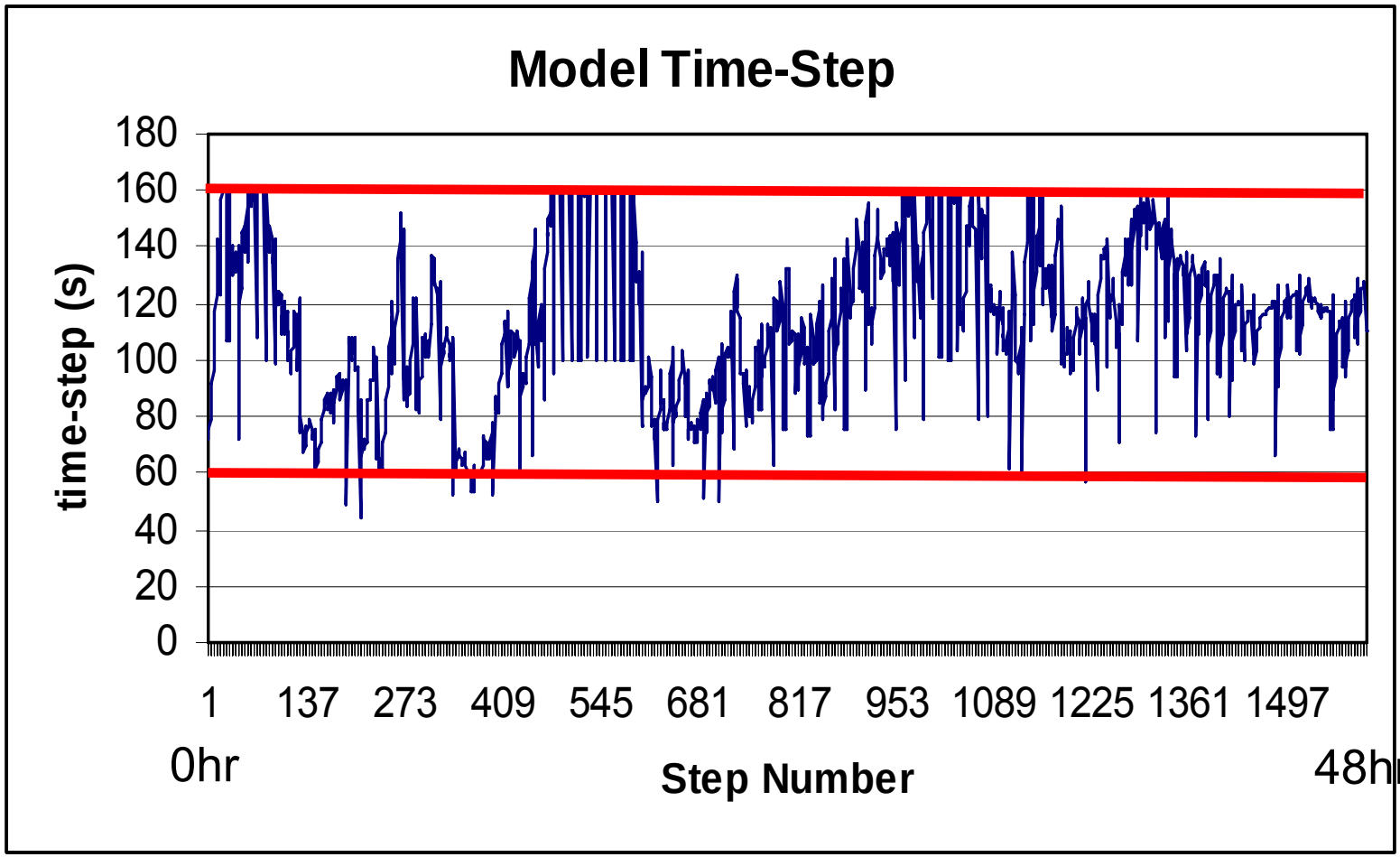
$dt_n = t_{output}$

} ≤1 step from output

WSI's Forecasting Domains

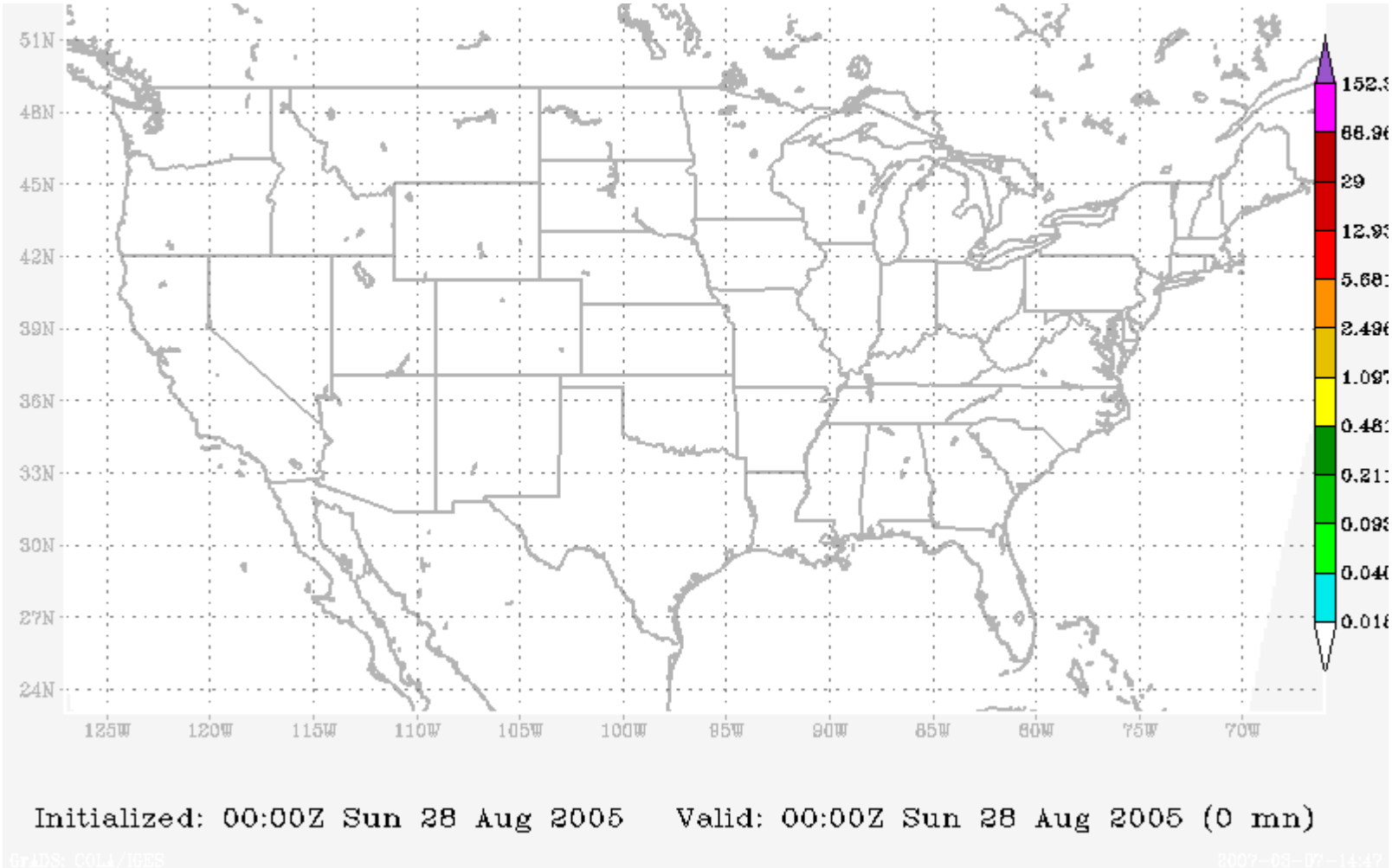


Model time-step for CONUS



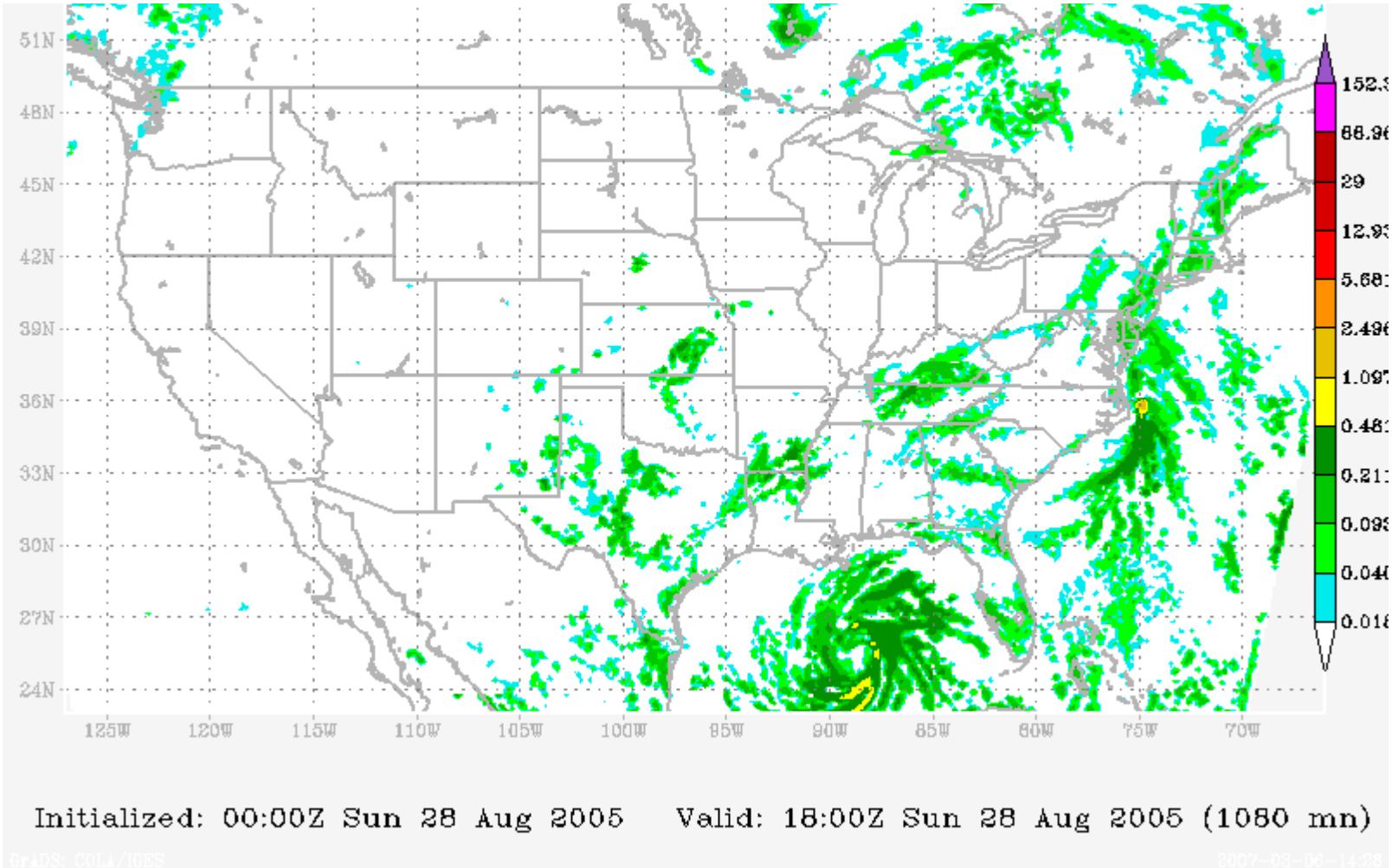
Testing and Results: CONUS Precipitation

- Static time-step (run-time: 145 minutes):



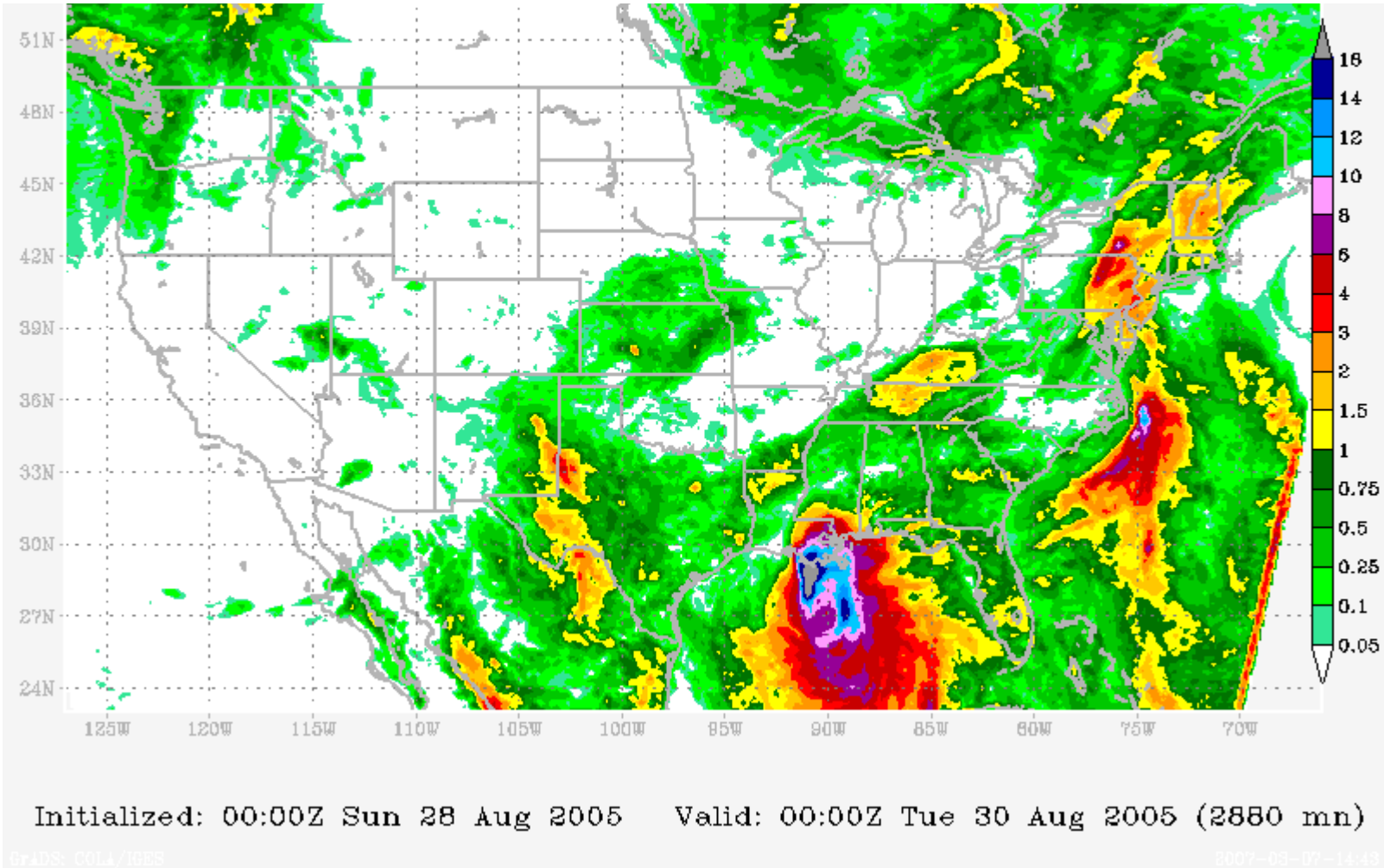
Testing and Results: CONUS Precipitation

- Adaptive time-step (run-time: 117 minutes):



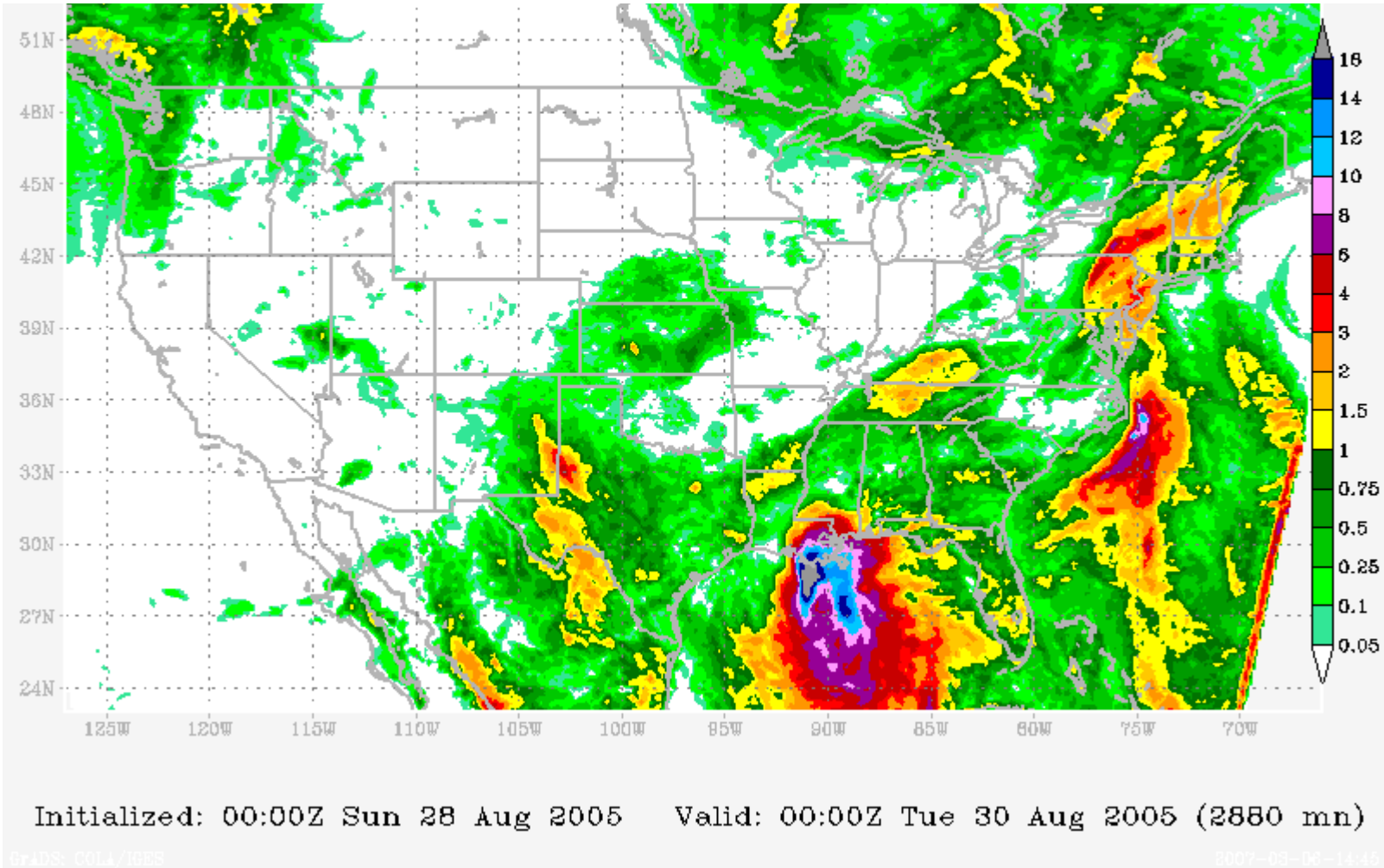
Results and Testing: CONUS Total Precip

- Static time-step:



Results and Testing: CONUS Total Precip

- Adaptive time-step:



Results and Testing: Statistical comparison

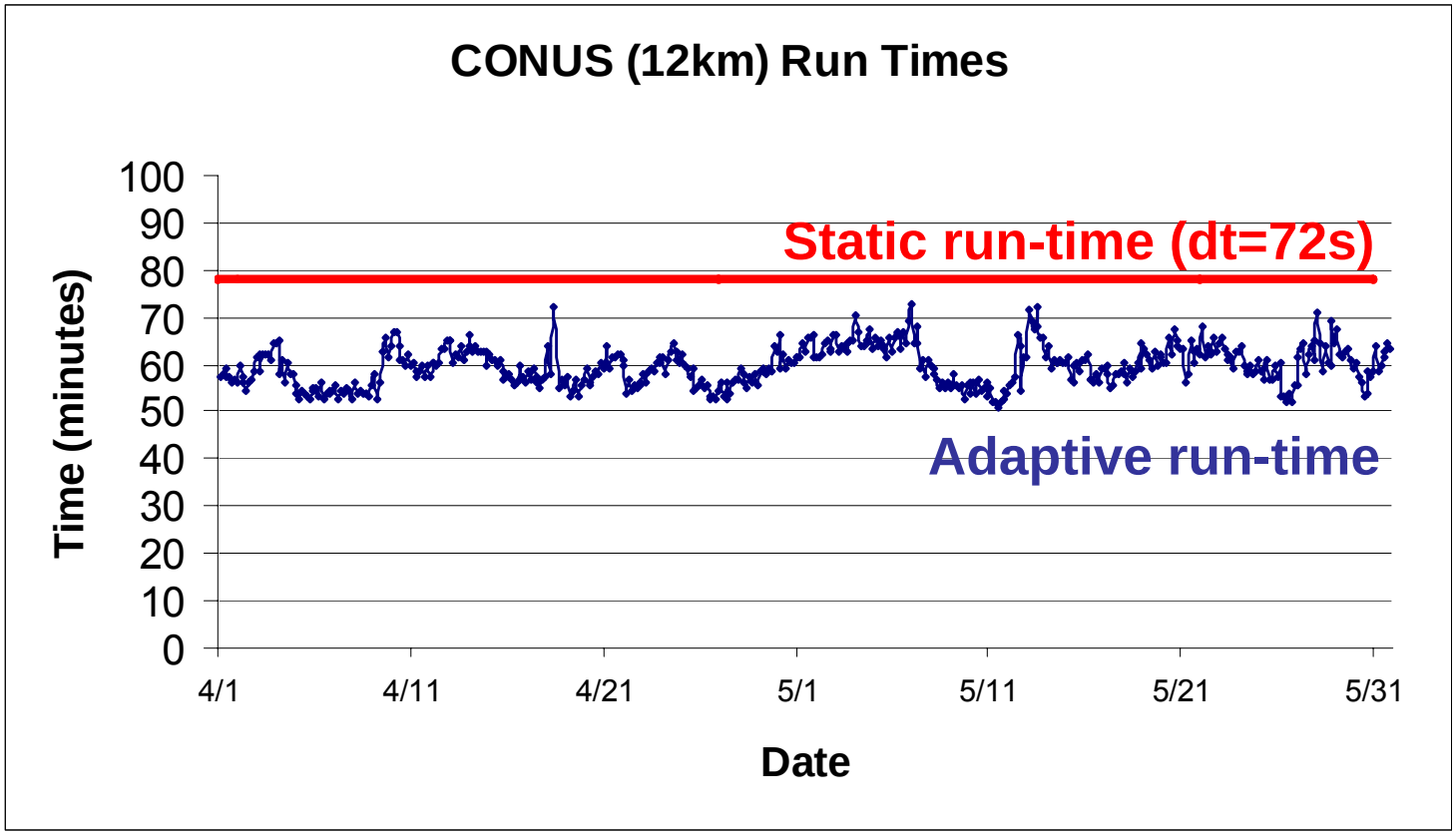
- Average differences (adaptive – static) for 12 cases

48 hr accumulation | avg of 12,24,36,48hr fcsts

Total PCP (%)	PCP stdev (%)	Cvg 1in PCP (%)	t2m (C)	t2m stdev (C)	u500 (m/s)	u500 stdev	v500 (m/s)	v500 stdev
4.11	3.10	8.27	-0.02	0.24	0.01	0.58	0.00	0.58

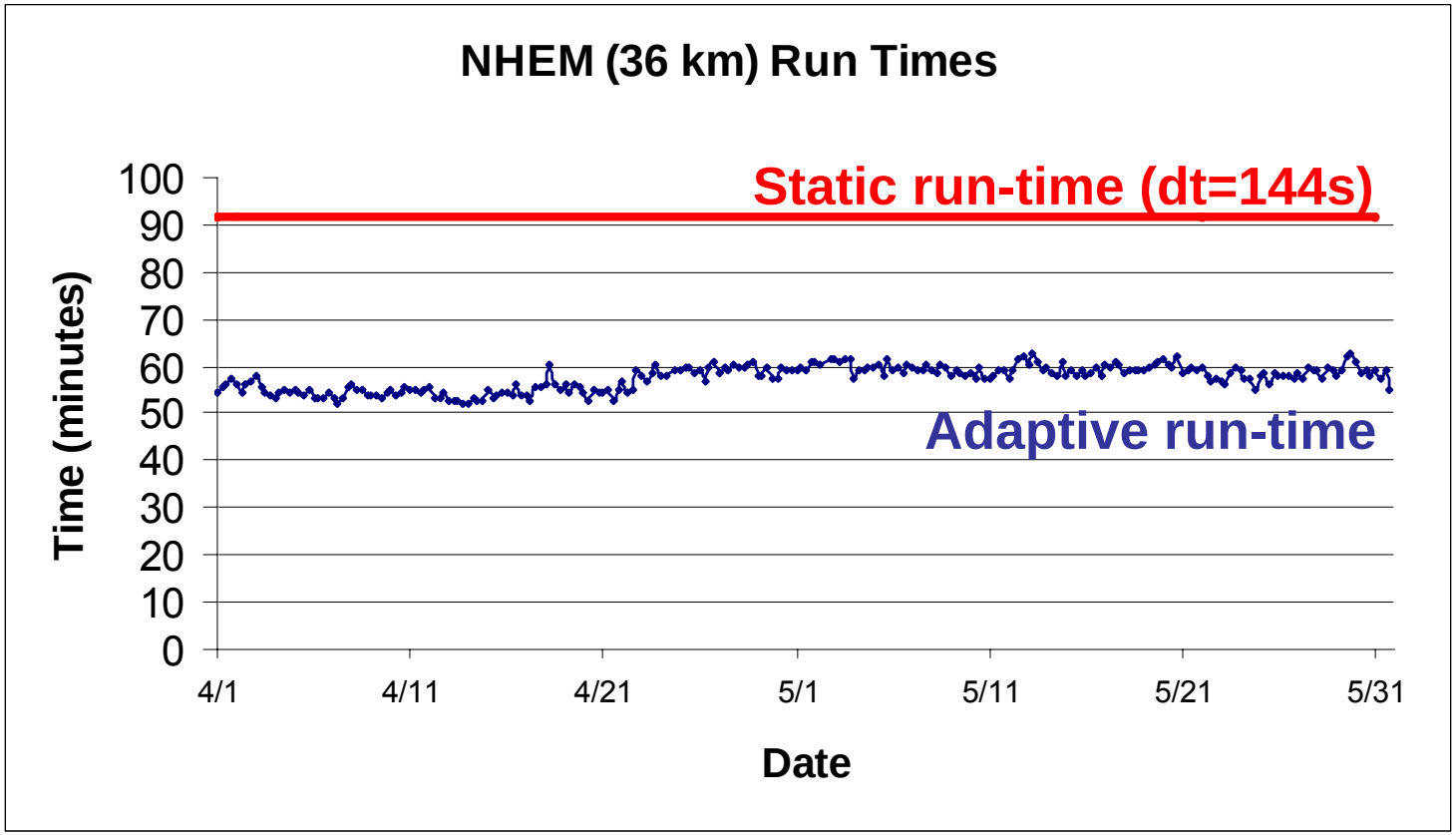
- Precip differences possibly due to Lin Microphysics
- Preliminary: Smaller precip diff when using WSM6

Results and Testing: Operational Run-times



- Range: 51-72 minutes
- Average: 60 minutes
- Stdev: 4.1 minutes
- Improvement over Static: 8-35%

Results and Testing: Operational Run-times



- Range: 52-63 minutes
- Average: 58 minutes
- Stdev: 2.6 minutes
- Improvement over Static: 32-43%

Summary and Future Work

- Adaptive time-step provides improved efficiency while maintaining model stability
- Minimal differences in forecasts between adaptive and static
  - Precipitation needs to be investigated further
- Implemented with version 2.1.2, porting to 2.2 is ongoing
- Testing at higher resolution (4km) is ongoing
- Can we adapt the acoustic time-step?
- Implement in WRF if there is sufficient interest