11C.4

Options for Large-Eddy Simulations of Hurricane Boundary Layers in CM1 (and Other Numerical Models)

George Bryan
National Center for Atmospheric Research

35th Conference on Hurricanes and Tropical Meteorology
New Orleans, LA
11 May 2022

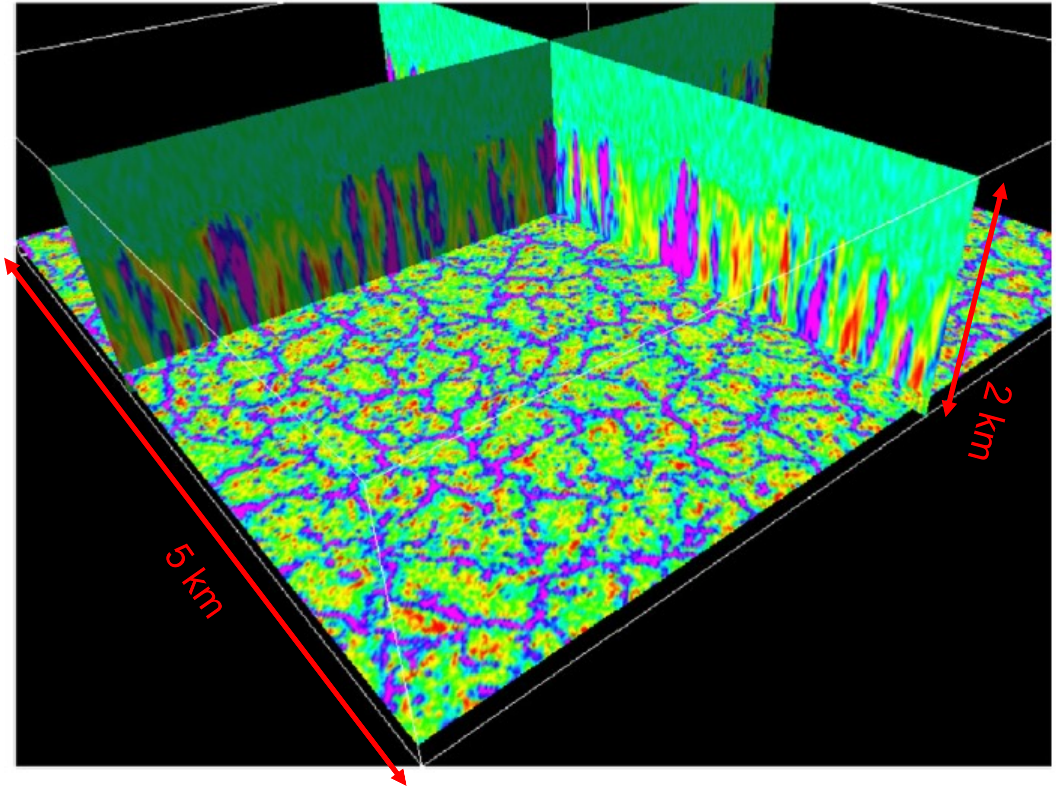
Acknowledgements: NSF PREEVENTS, ONR TCRI

(also thanks to Dan Stern, Rich Rotunno, Xiaomin Chen, …)

NCAR is sponsored by the National Science Foundation

Large-Eddy Simulation (LES)

Vertical velocity, convective boundary layer



Moeng and Sullivan (2015)
(*Encyclopedia of Atmospheric Sciences*)

- Integrate governing equations using turbulence-permitting resolution
  - Δx of ~100 m or less
- Advantage:
  - A more accurate representation of turbulent processes
- Disadvantages:
  - Need to account for larger scales
  - High resolution (even $\Delta x < 10$ m) doesn't solve everything

Outline

Three Options in CM1:

1. Small-domain, high-resolution "patch"
2. The "eddy recycling" method
3. Subgrid turbulence model

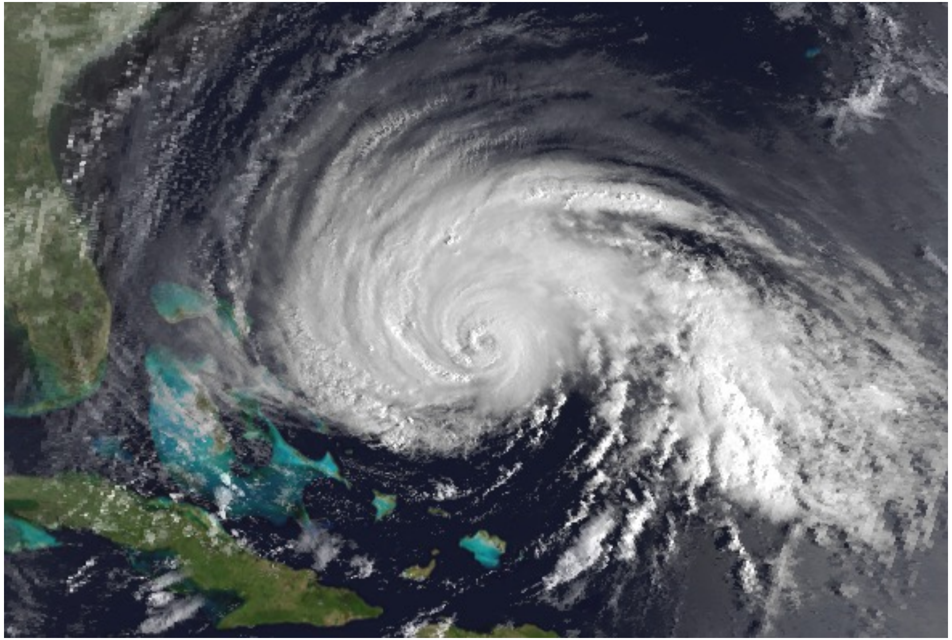
Outline

Three Options in CM1:

1. Small-domain, high-resolution “patch”
2. The “eddy recycling” method
3. Subgrid turbulence model

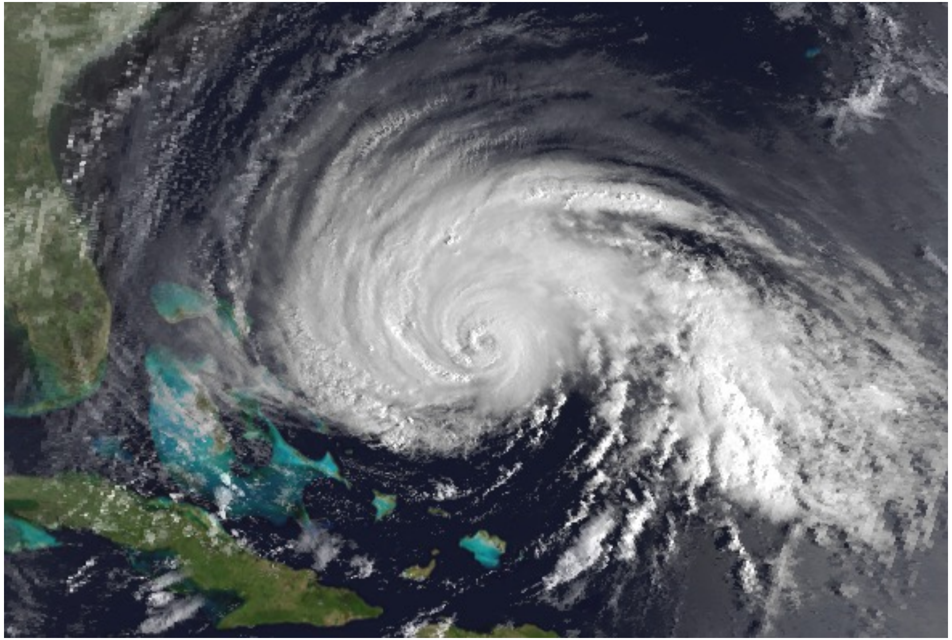
Problem:

- Hurricanes are huge! (hundreds of km)
- So, 100 m grid spacing for ~2-day simulation requires millions of supercomputer core-hours



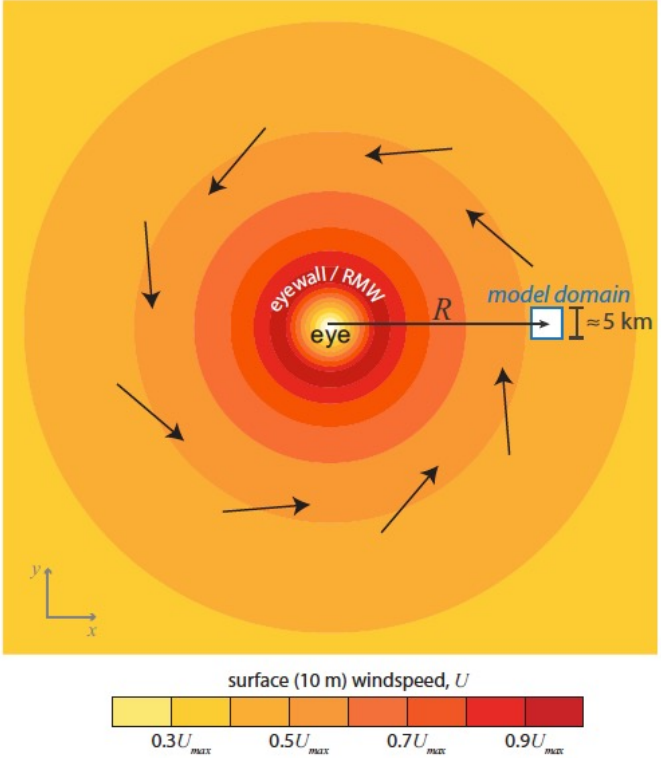
Problem:

- Hurricanes are huge! (hundreds of km)
- So, 100 m grid spacing for ~2-day simulation requires millions of supercomputer core-hours



Solution:

Simulate a “patch” of the TC boundary layer



Bryan et al. (2017a)

Bryan et al. (2017a, BLM)

radial velocity: $$\frac{\partial u}{\partial t} = \ldots \boxed{+\langle v\rangle \frac{V}{R}} \boxed{+\frac{\langle u\rangle^2}{R}} \boxed{-\left(fV+\frac{V^2}{R}\right)}$$

tangential velocity: $$\frac{\partial v}{\partial t} = \ldots \boxed{-\langle u\rangle \frac{V}{R}} \boxed{-\langle u\rangle \frac{\partial V}{\partial R}}$$

centrifugal acceleration
& radial advection
& pressure-gradient acceleration

User must specify 3 terms:
- R (radius from center of TC)
- $V(z)$ ("gradient wind")
- $dV/dR\ (z)$ (radial gradient of V)

Caveat: applies only *outside eyewall* (r > RMW)

Chen et al. (2021, JAS)

$$\frac{\partial \theta}{\partial t} = \ldots + \frac{\theta_{\mathrm{ref}} - \langle\theta\rangle}{\tau_n}$$

$$\frac{\partial q_v}{\partial t} = \ldots + \frac{q_{\mathrm{ref}} - \langle q_v\rangle}{\tau_n}$$

These "nudging" terms account crudely for radial advection, subsidence, microphysics, radiation…

User must specify vertical profiles of θ and q_v

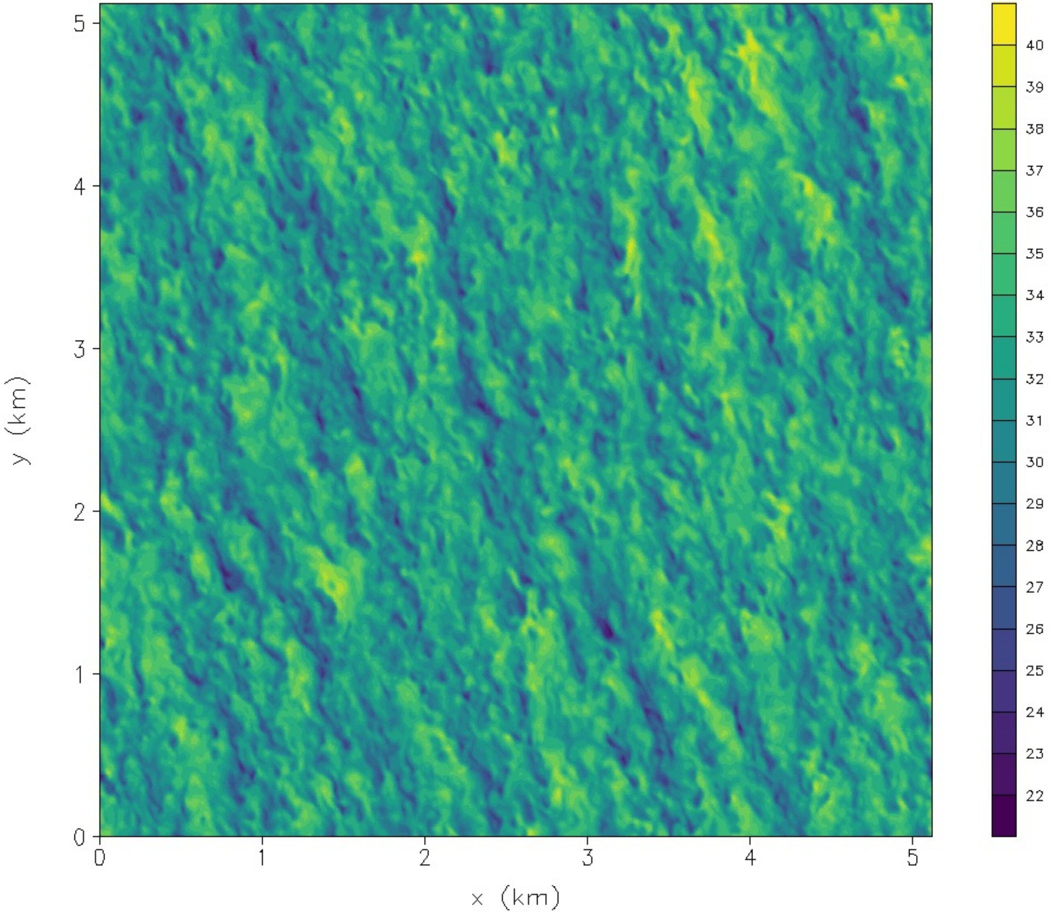
CM1 default (LES Hurricane test case):
- θ: constant lapse rate
- q_v: constant relative humidity

Angled brackets: < > denotes horizontal average

Horizontal windspeed (m s^{-1})
at 100 m ASL

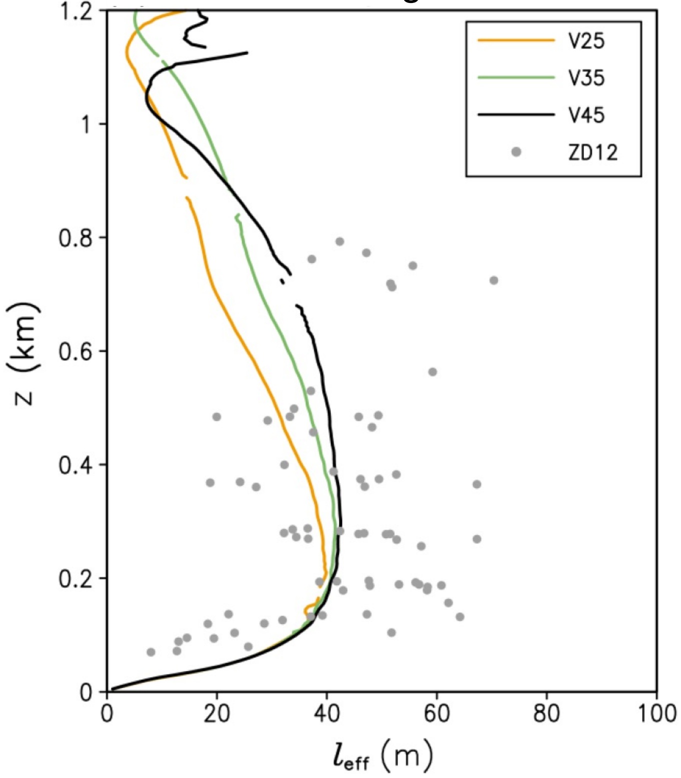
CM1:
$\Delta x = \Delta y = 10$ m,
$\Delta z = 5$ m

Input parameters:
$R = 40$ km
$V = 38$ m s^{-1}
$dV/dR = -8 \times 10^{-4}$ s^{-1}
(radial decay parameter $n = 0.8$)

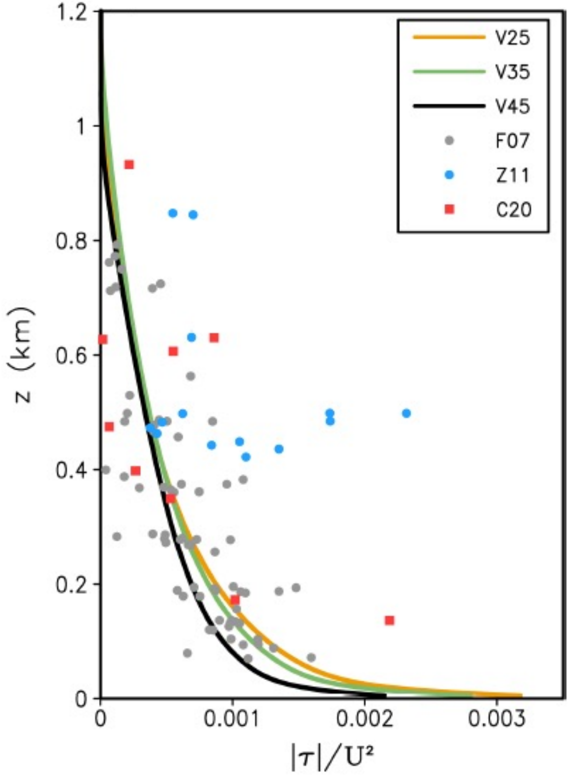


Simulations (colored lines) compare well with obs. (dots)

Effective length scale

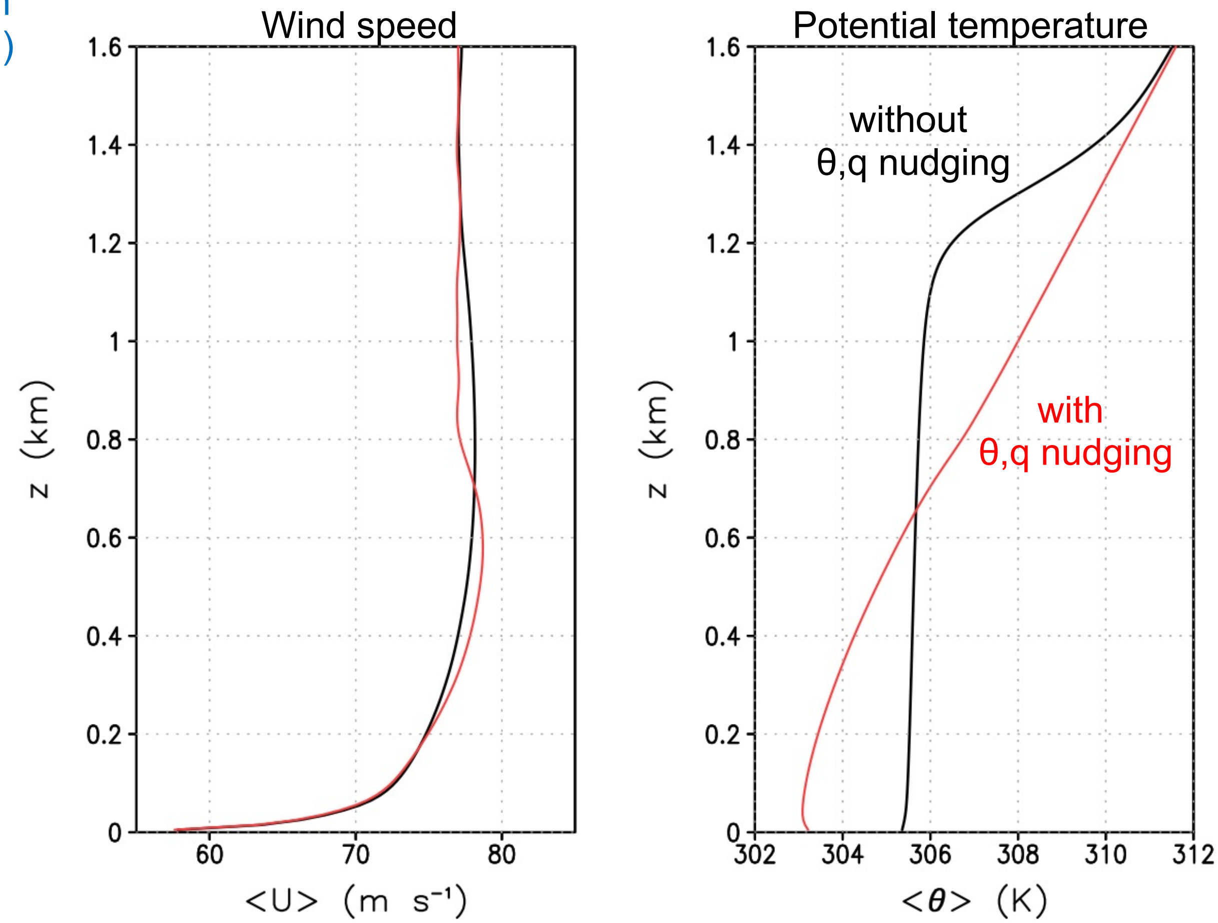


Vertical flux of horiz. momentum



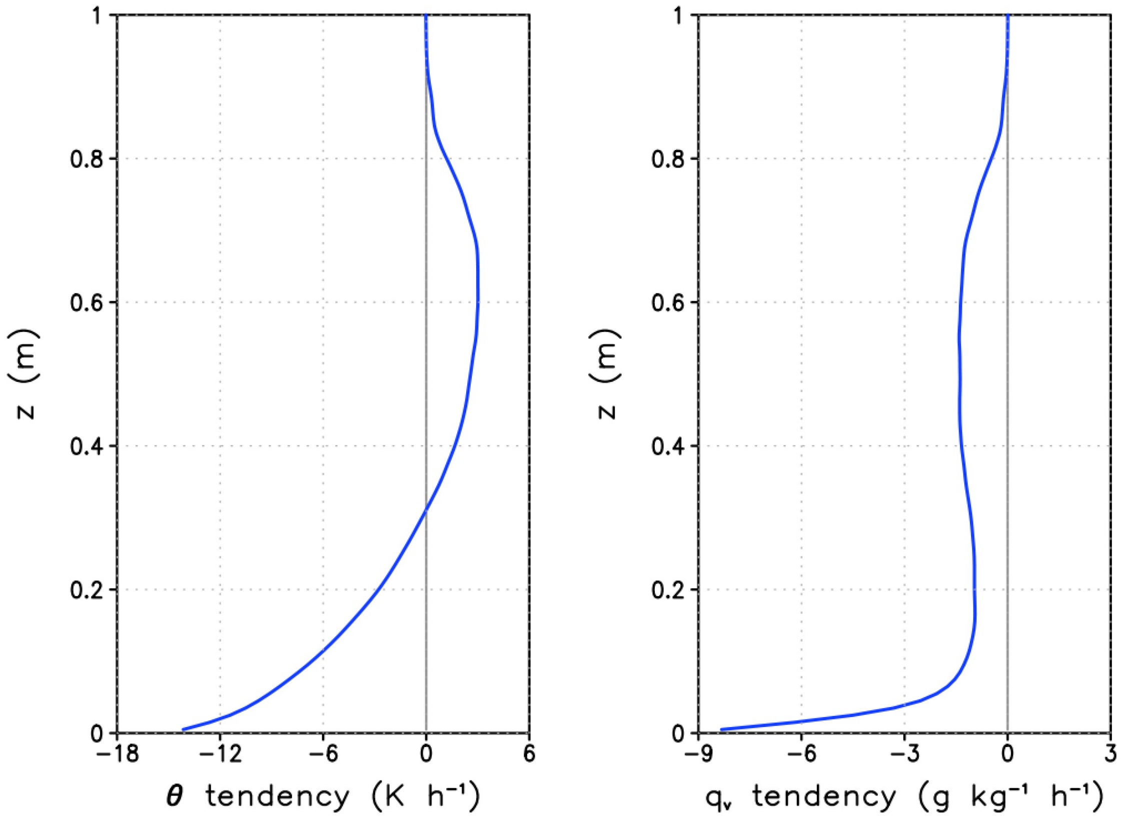
see Bryan et al. (2017a), Chen et al. (2021, JAS) (also, the next talk!)

• simulations from Chen et al. (2021)



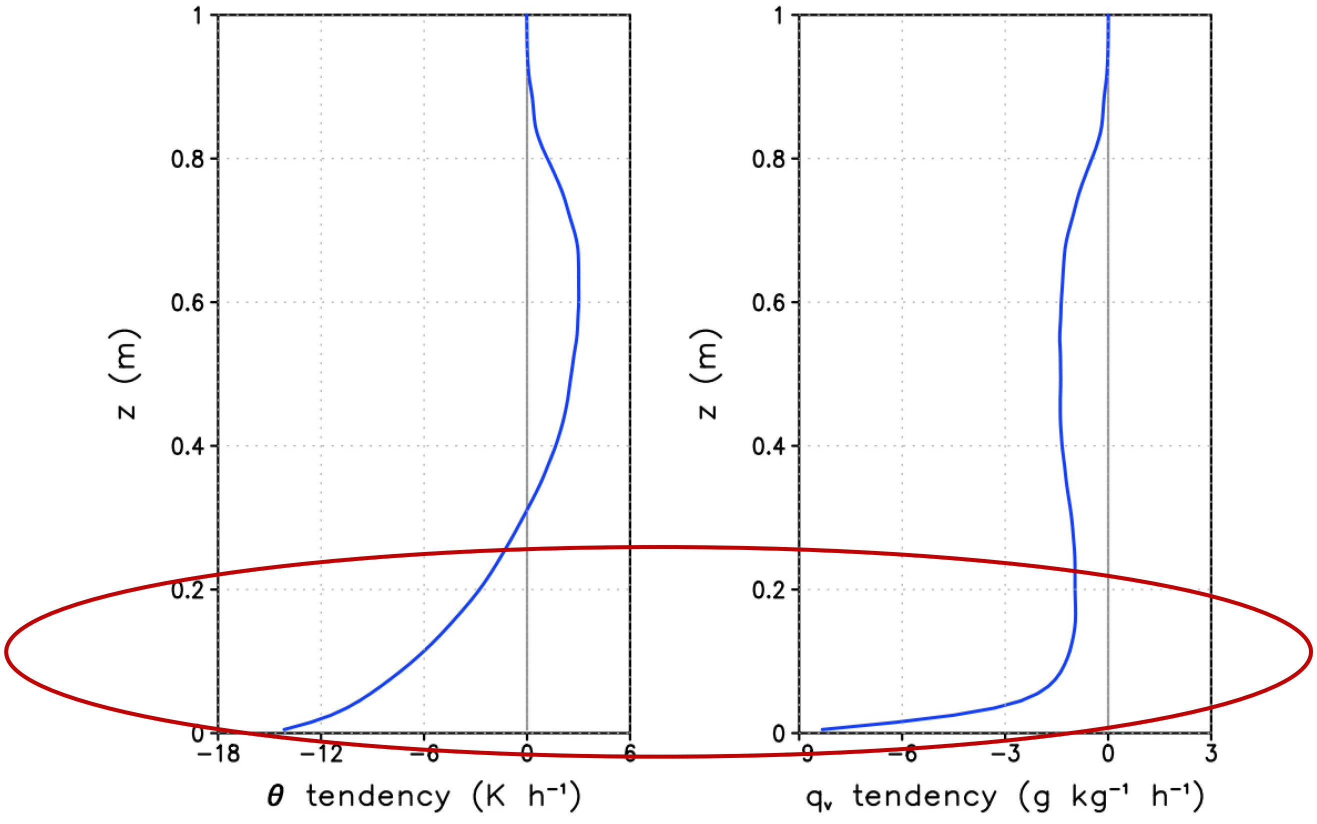
• simulations from
Chen et al. (2021)

Nudging tendencies



• simulations from Chen et al. (2021)

Nudging tendencies

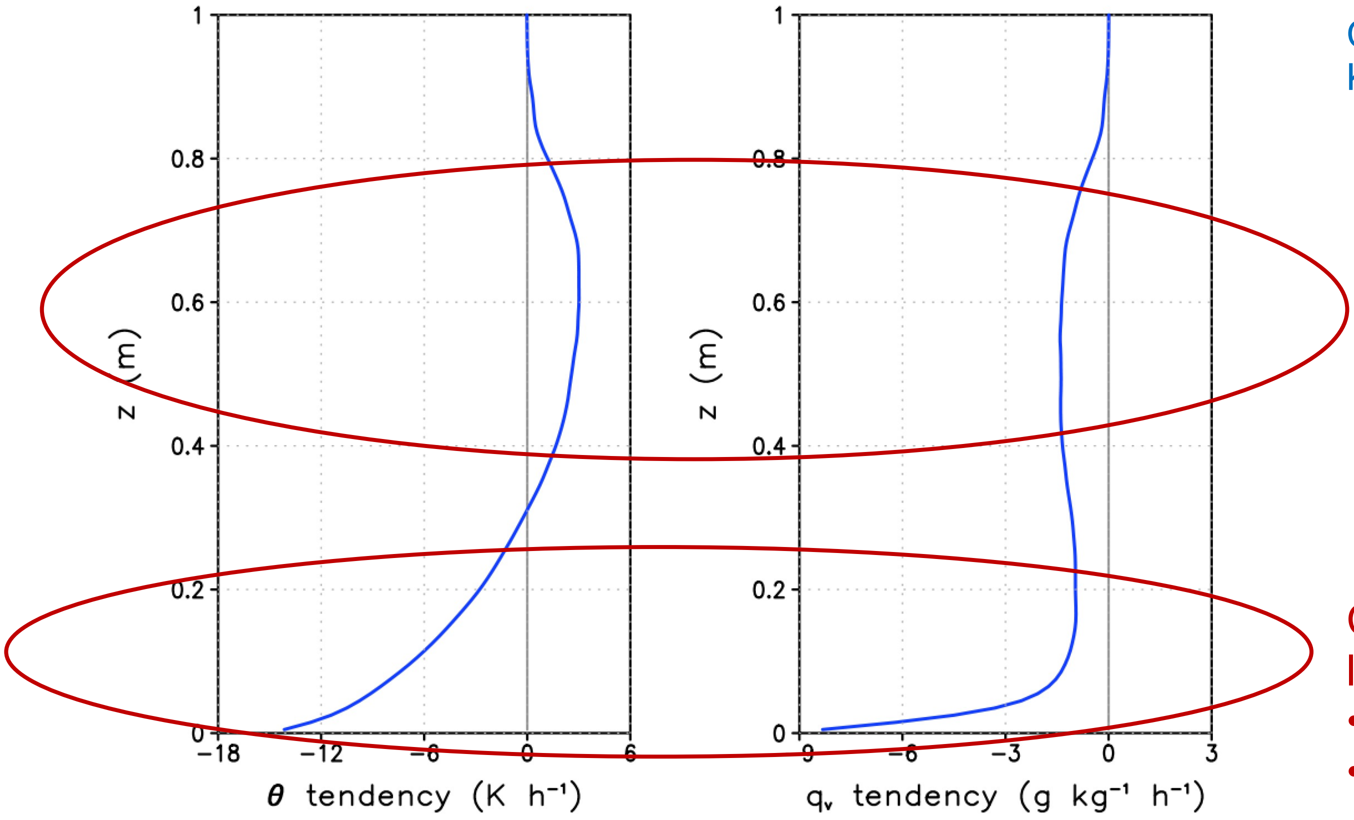


Cooling and drying in lower PBL:

- Radial Advection
- Evaporation (for θ)

• simulations from
Chen et al. (2021)

Nudging tendencies



Consistent with
Kepert et al. (2016)

Warming and drying in upper PBL:
- Clouds/convection
- Subsidence

Cooling and drying in lower PBL:
- Radial Advection
- Evaporation

The Small-Domain (“patch”) Hurricane LES Technique

CM1

- Pre-configured namelists in “run/config_files”
- Since `cm1r21`: user settings are now in “namelist.input” file (eg, hurr_vg)
- Can rotate winds & mesoscale-tendency terms (hurr_rotate)

Other models:

- Should be easy to add
- Primarily need “mesoscale tendency terms”
  - Horizontal winds: Bryan et al. (2017a, BLM)
  - Temperature, moisture: Chen et al. (2021, JAS)
- Code to calculate domain-average profiles: <u>, <v>, <theta>, <qv>

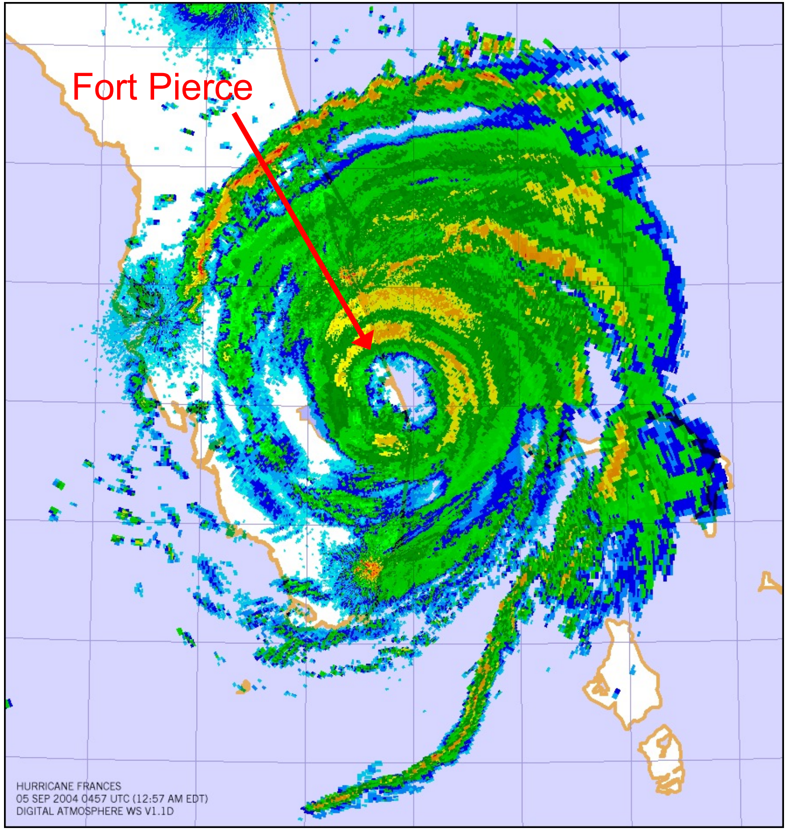
Outline

Three Options in CM1:

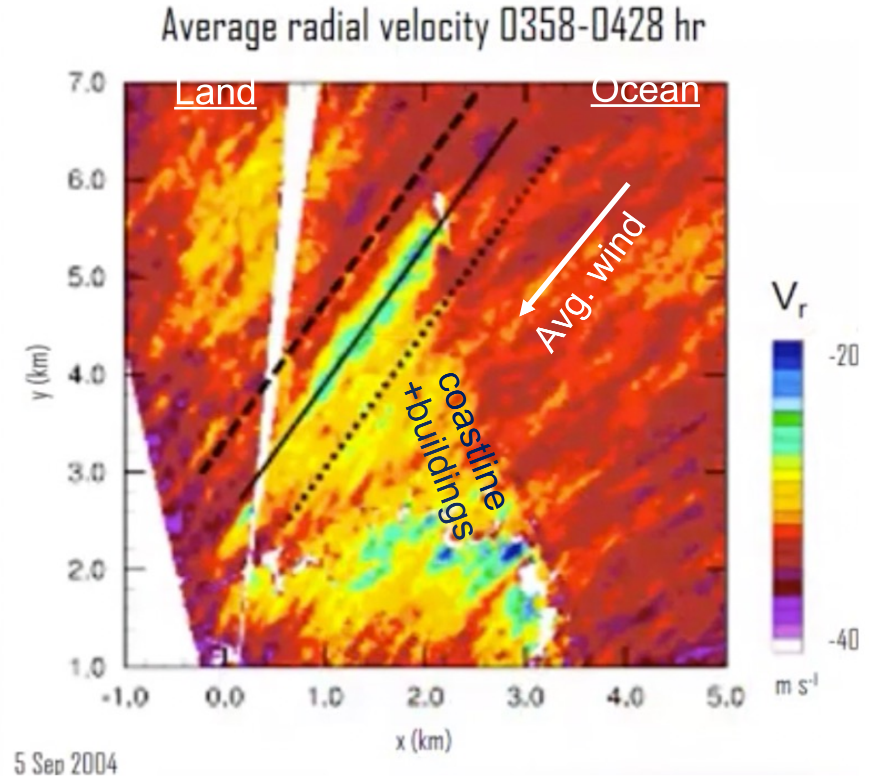
1. Small-domain, high-resolution “patch”
2. The “eddy recycling” method
3. Subgrid turbulence model

Inhomogeneous Conditions

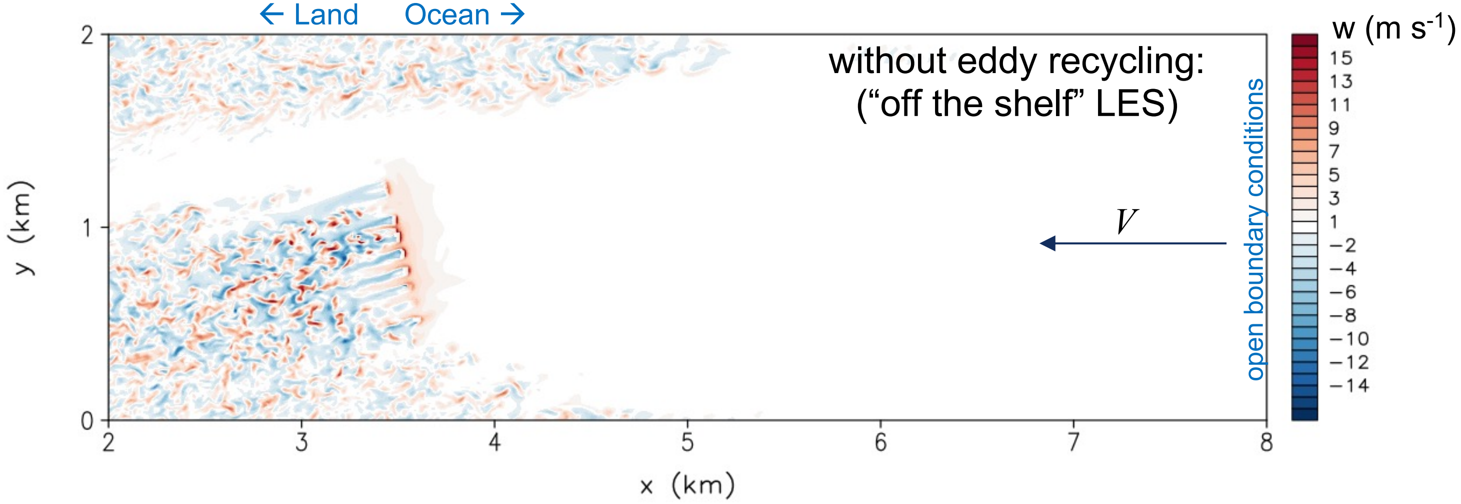
Hurricane Frances (2004)

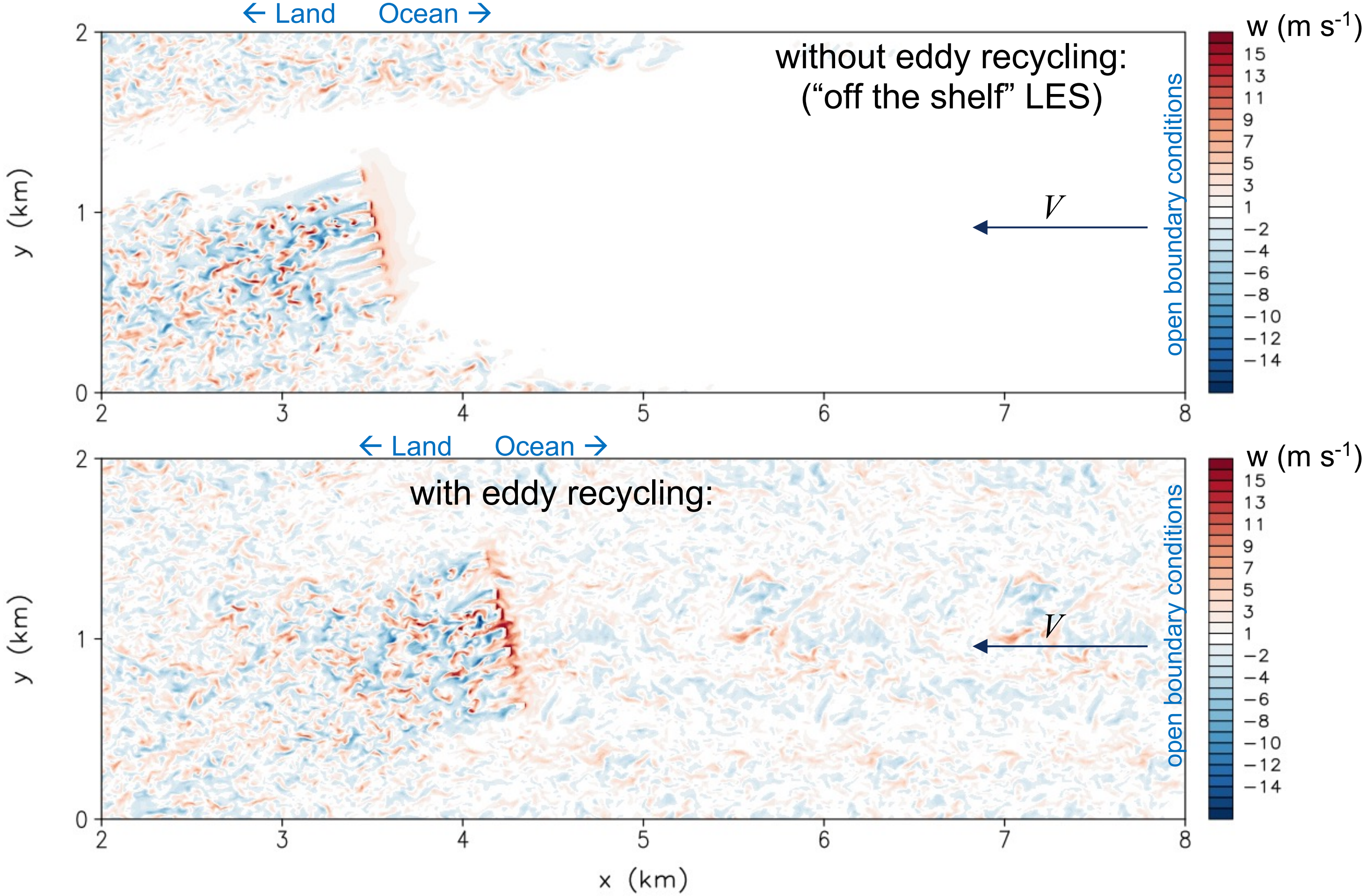


Kosiba (2014, Hurricanes Conf.)
Doppler on Wheels radial velocity:

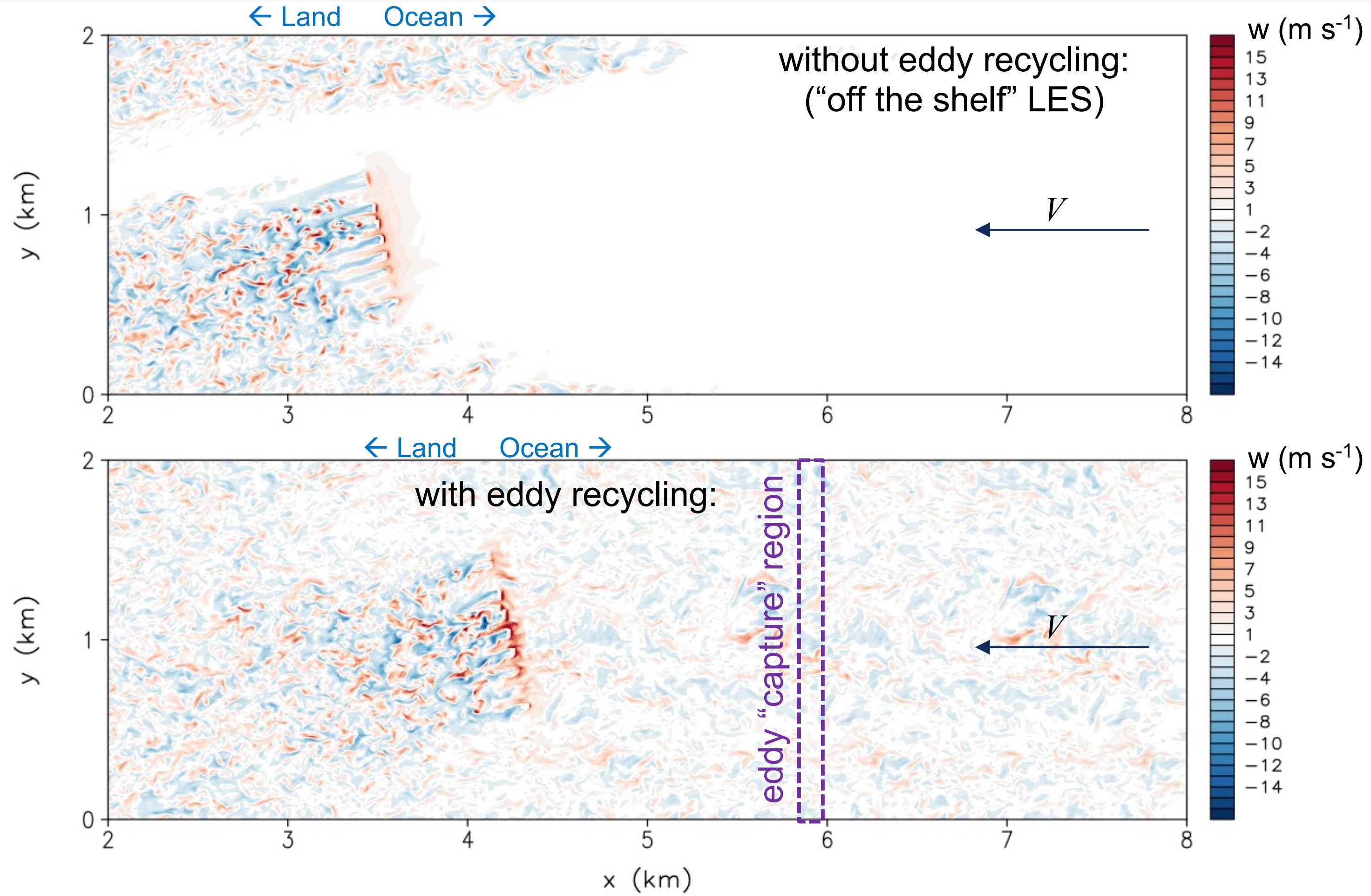


- Not suitable for periodic lateral boundary conditions!

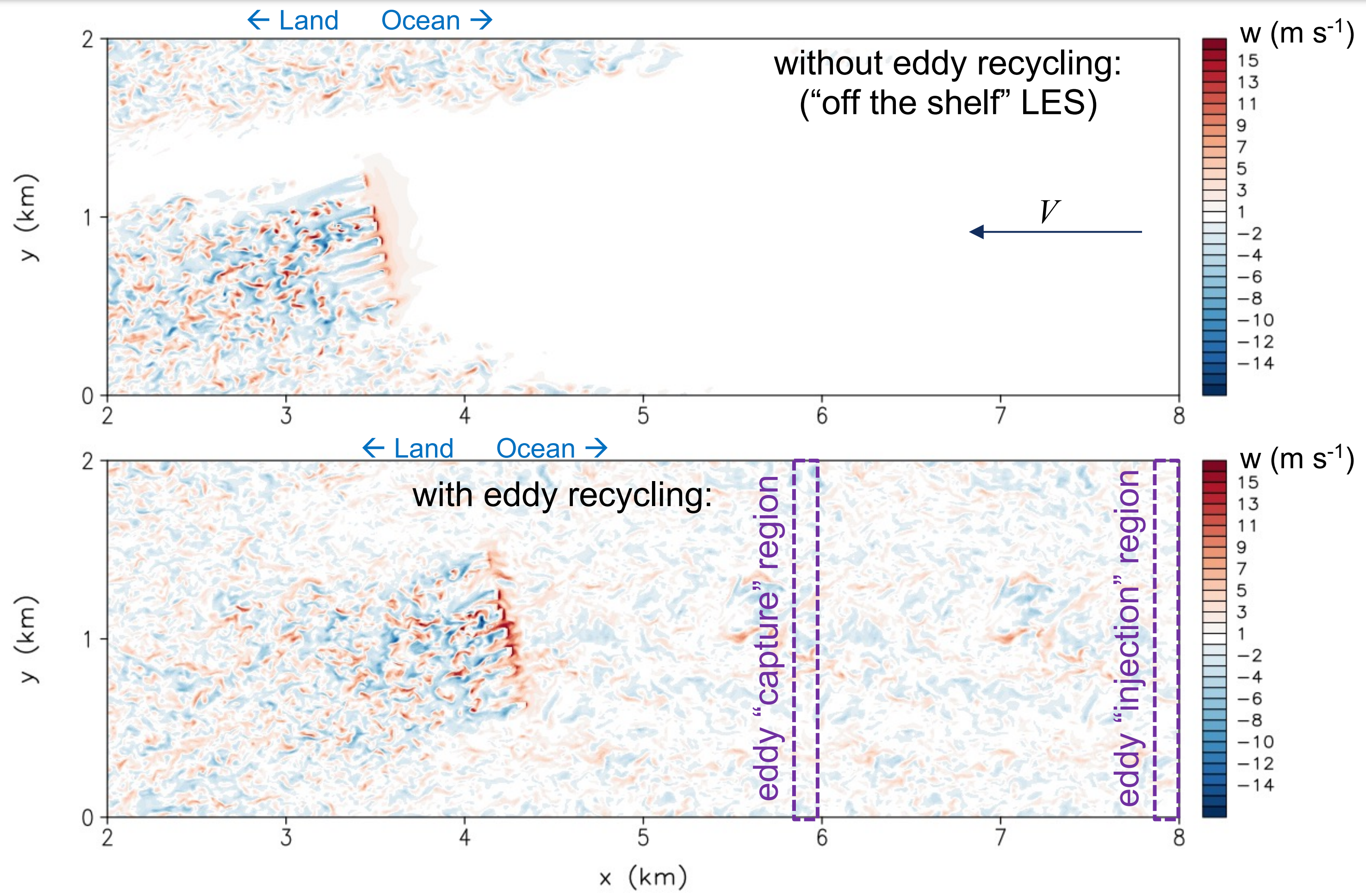

← Land
Ocean →
without eddy recycling:
("off the shelf" LES)
open boundary conditions
V
w (m s-1)
15
13
11
9
7
5
3
1
−2
−4
−6
−8
−10
−12
−14
y (km)
x (km)
2
1
0
2
3
4
5
6
7
8



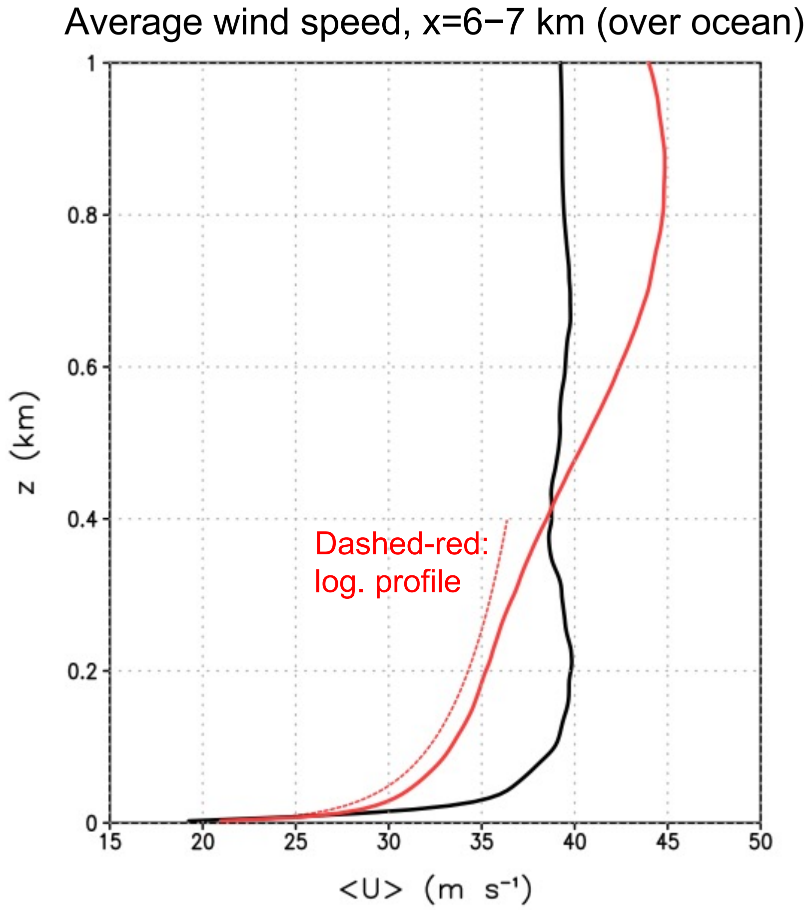
← Land
Ocean →
without eddy recycling:
("off the shelf" LES)
open boundary conditions
V
y (km)
w (m s-1)
15
13
11
9
7
5
3
1
−2
−4
−6
−8
−10
−12
−14
← Land
Ocean →
with eddy recycling:
open boundary conditions
V
y (km)
x (km)



← Land
Ocean →
without eddy recycling:
("off the shelf" LES)
V
w (m s-1)
15
13
11
9
7
5
3
1
−2
−4
−6
−8
−10
−12
−14
y (km)
2
1
0
2
3
4
5
6
7
8
← Land
Ocean →
with eddy recycling:
eddy "capture" region
V
w (m s-1)
y (km)
x (km)



← Land
Ocean →
without eddy recycling:
(“off the shelf” LES)
V
w (m s-1)
15
13
11
9
7
5
3
1
−2
−4
−6
−8
−10
−12
−14
y (km)
2
1
0
2
3
4
5
6
7
8
← Land
Ocean →
with eddy recycling:
eddy “capture” region
eddy “injection” region
w (m s-1)
y (km)
x (km)



Black:
without eddy injection

Red:
with eddy injection

The “Eddy Recycling” Technique

CM1

- Available in `cm1r21.0`
- See pre-configured namelists:
  `config_files/hurricane_les_within_mm`
  `config_files/les_HurrCoast`
- A work in progress; feedback welcome

Other models:

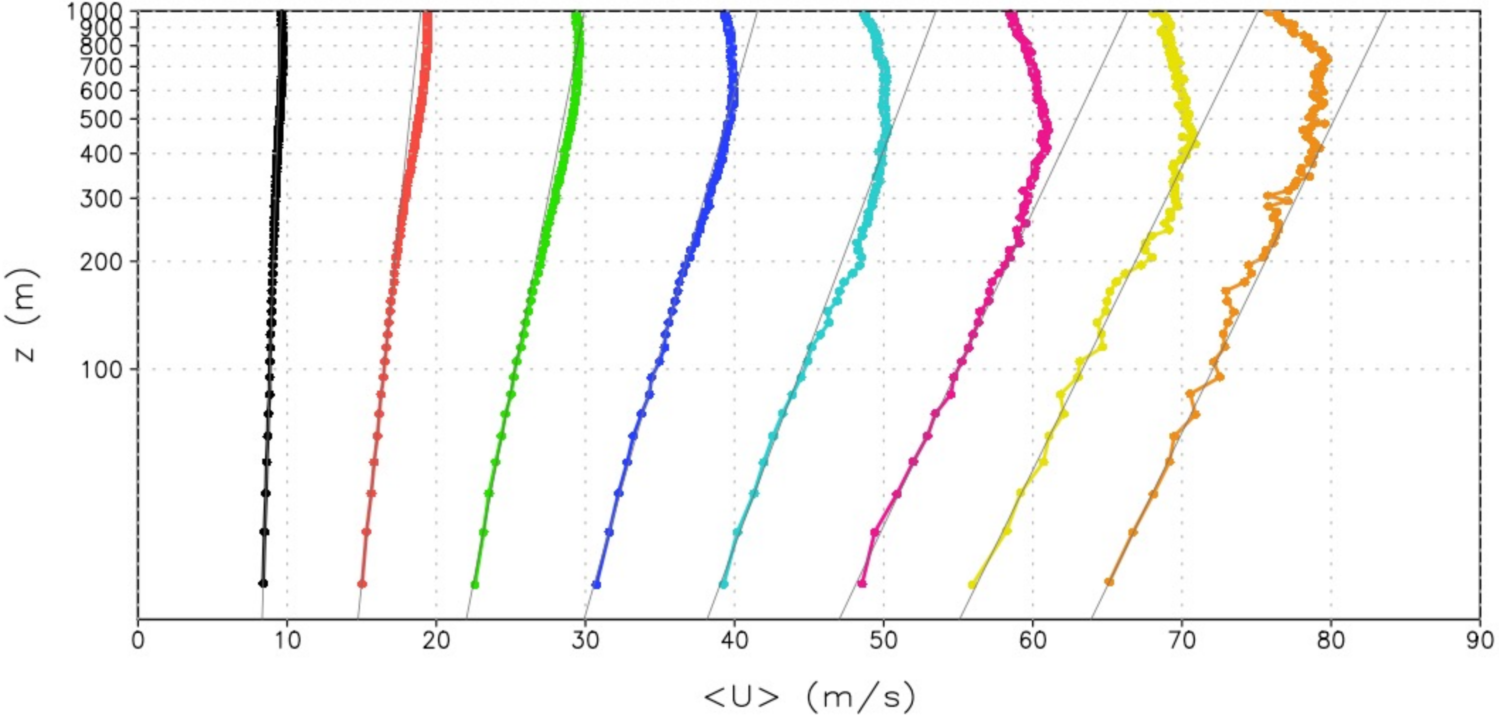
- Not easy to add…
- Need code to identify eddy “capture” and “injection” zones
- Need code to apply tendencies in “injection” zone
- Distributed-memory (MPI) communication can be a real hassle

Outline

Three Options in CM1:

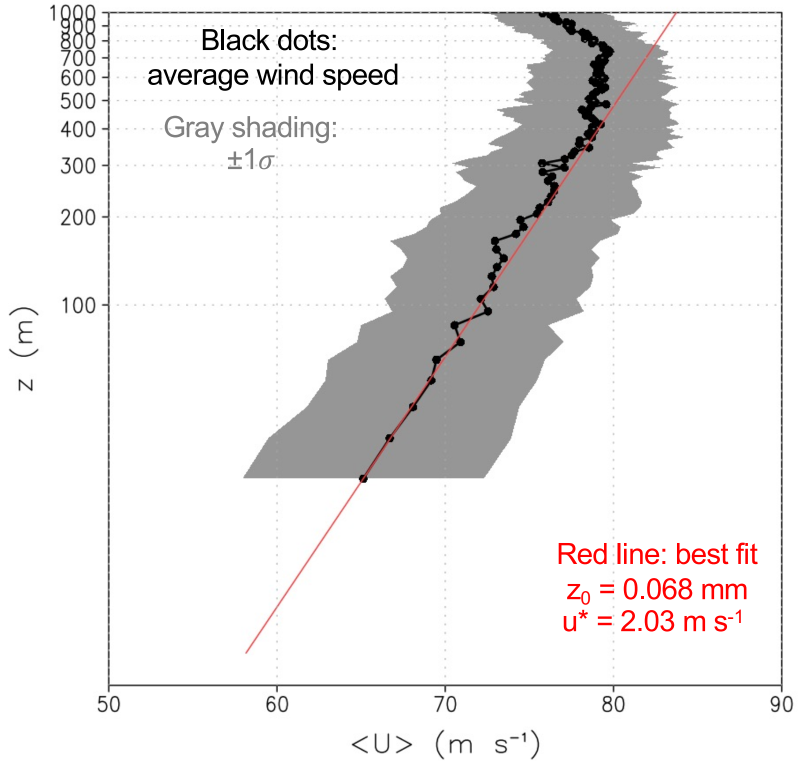
1. Small-domain, high-resolution “patch”
2. The “eddy recycling” method
3. Subgrid turbulence model

Composites: average wind speed
(uses database from Wang et al. 2015, BAMS)

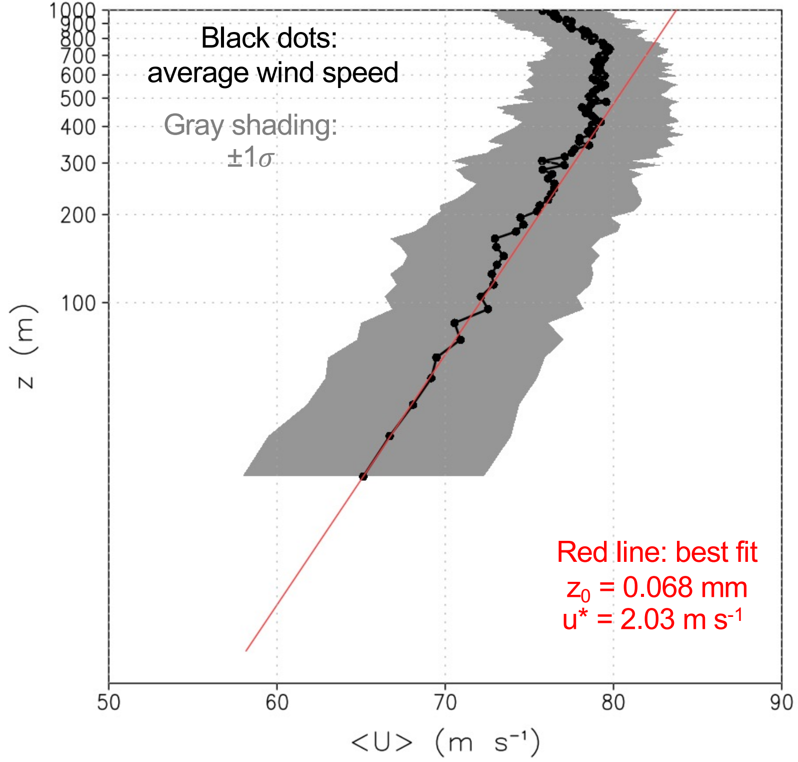


see also Richter et al. (2021, JAS)

Composite wind speed from dropsondes (N = 98)

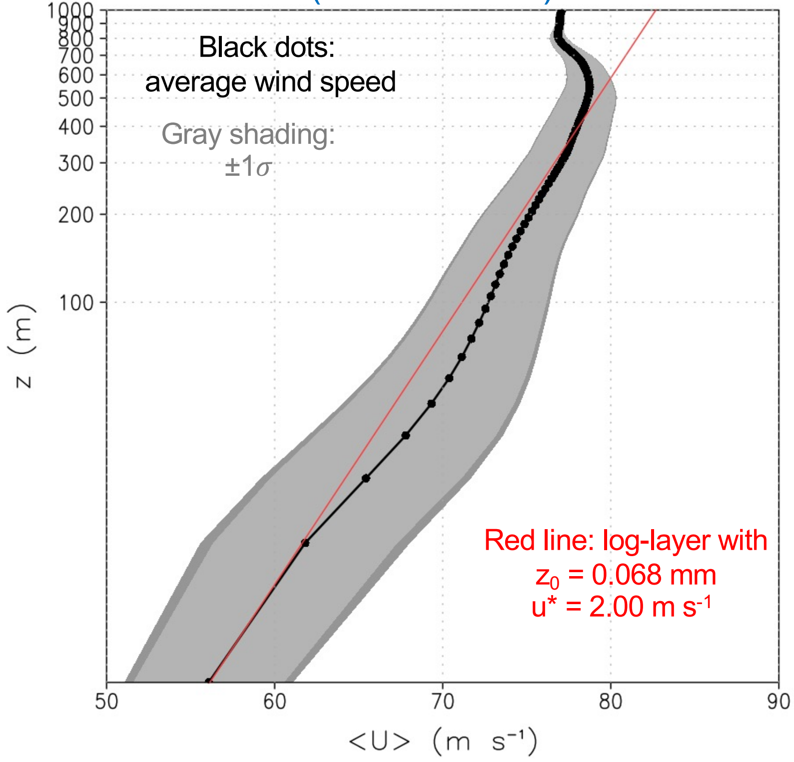


Composite wind speed from dropsondes (N = 98)



Average wind speed from CM1-LES (Δx = 20 m)

Standard ("off-the-shelf") SGS model



- Overprediction of shear near surface is a common problem with LES (eg, Brasseur and Wei, 2010)

See Appendix of Bryan et al. (2017a, BLM)

(motivated by Sullivan et al., 1994, BLM)

For subgrid turbulence terms

$$\frac{\partial u_i}{\partial t} = -u_j \frac{\partial u_i}{\partial x_j} - c_p \theta \frac{\partial \pi'}{\partial x_i} + \delta_{i3} g \frac{\theta'}{\theta_0} + \varepsilon_{ij3} u_j f + \boxed{\frac{1}{\rho} \frac{\partial \tau^t_{ij}}{\partial x_j}}$$

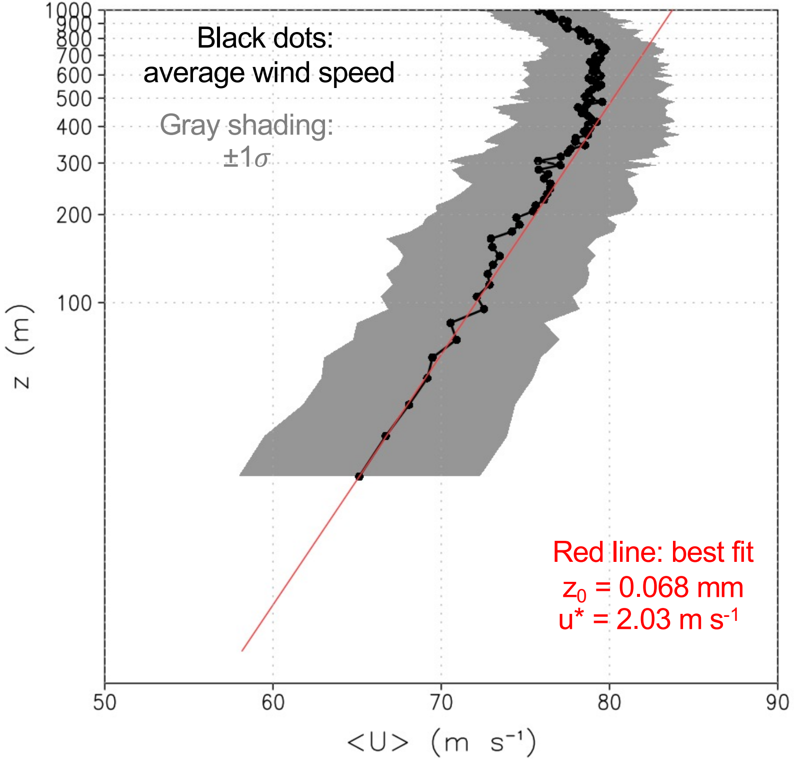
… add a second eddy viscosity term

$$\tau^t_{13} = \rho K_m \gamma \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \boxed{\rho K_w \frac{\partial \tilde{u}}{\partial z}},$$

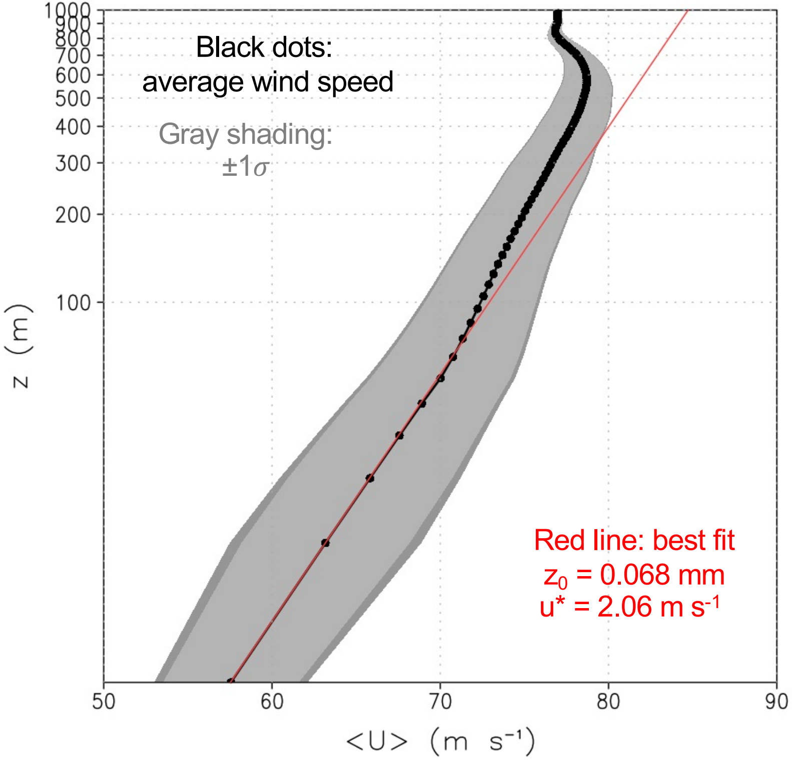
$$\tau^t_{23} = \rho K_m \gamma \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \boxed{\rho K_w \frac{\partial \tilde{v}}{\partial z}},$$

K_w is determined diagnostically so that <U> is logarithmic near surface (z < 100 m)

Composite wind speed from dropsondes (N = 98)



Average wind speed from CM1-LES (Δx = 20 m)
Two-Part SGS model



The “Two-Part” Subgrid Turbulence Model

CM1

- Available since `cm1r20`
- sgsmodel = 4
- Since `cm1r21` can use time-average instead of space-average (t2p_avg = 2)
  - So we can use it for non-homogeneous cases

Other models:

- Should be fairly easy to add
- Need code to calculate average fields
  - Either spatial averages or time averages
- Code to calculate K_w is not complicated
  - (happy to share CM1 code)
- Minor addition to subgrid turbulence code

Thanks!

CM1 webpage: https://www2.mmm.ucar.edu/people/bryan/cm1
(or just google “CM1”)

[These slides are now available on the CM1 webpage]

Acknowledgements: NSF PREEVENTS, ONR TCRI

(also thanks to Dan Stern, Rich Rotunno, Xiaomin Chen, …)