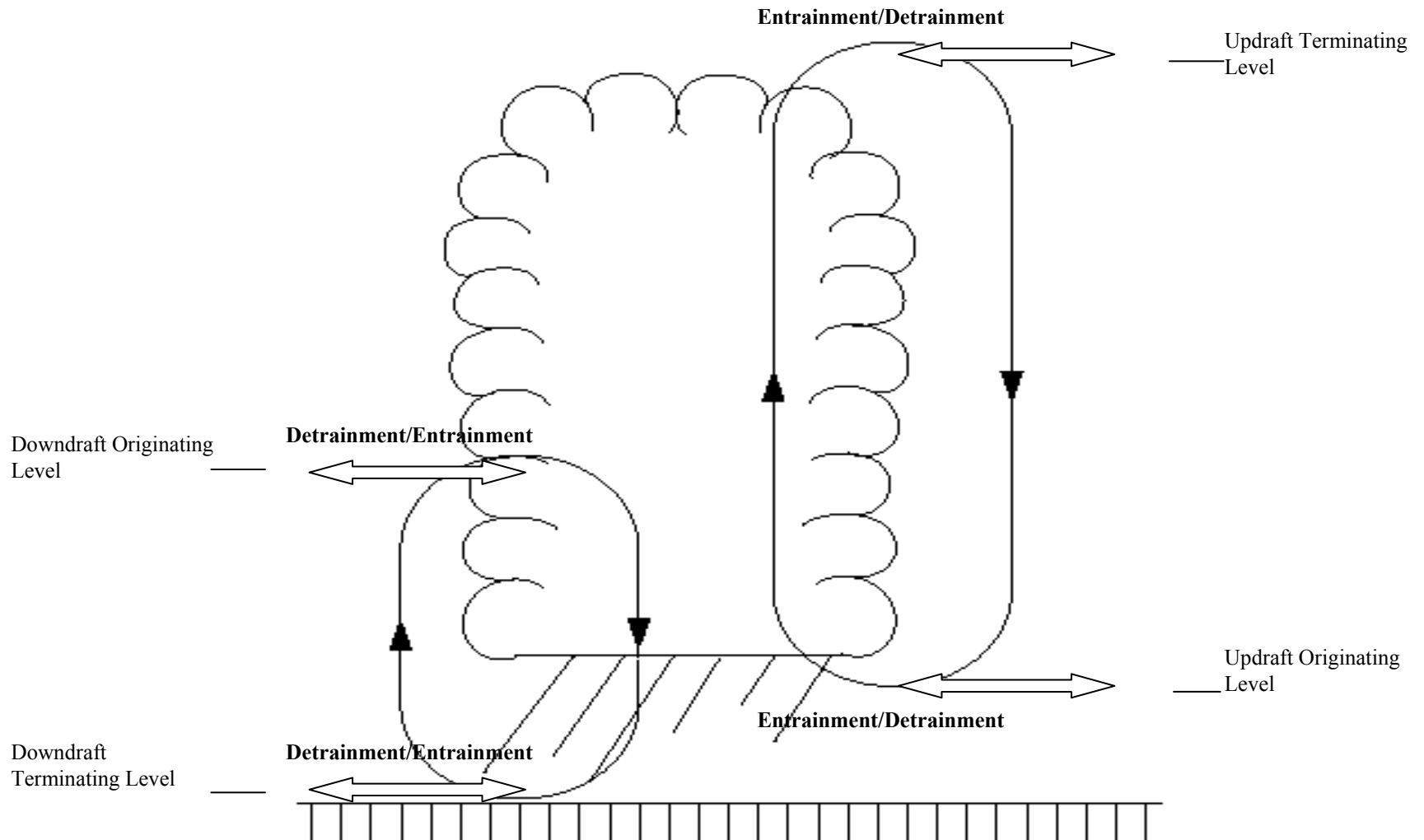


# Grell Convection Scheme in RAMS

Jih-Wang Wang

2006/4/26

# *Grell Scheme for convection and transport*



# Grell Scheme Procedure

$$z_T \quad (\text{where } h_u(z_B) = \tilde{h}^*(z_T))$$

1. Find out

$$z_B \quad (LCL)$$

$$z_0 \quad (\text{where } \tilde{h}(z) \text{ minimum})$$

2. Check criteria to start or not

3. Determine mass flux based on ABE

4. Parameterize entrainment/detrainment rate and mass flux ratio

5. Quasi-equilibrium adjustment

6. Compute precipitation amount

# Grell Scheme Features

- Entrainment/Detrainment based on cloud radius = 12km
- Entrainment/Detrainment happens only at originating/terminating level
- Mass-flux scheme, i.e., rearrangement of mass in a vertical column
- One type cloud, shallow cumulus parameterization usually accompanied
- Virtually instantaneous adjustment scheme

# Ensemble Approach

- 5 different closure types with regard to cloud base mass flux – dynamic control
- 144 combinations to parameterize entrainment/detrainment rate – static control
- 6 combinations for mass flux ratio
- Sensitivity tests against closure types, mass flux ratio, and entrainment/detrainment rate
- Who knows whether it's better or not?

# My Understanding of LUT

$$[X]^\tau \rightarrow \langle \textit{parameterization} \rangle \rightarrow [X]^{\tau+1}$$

$$[X]^\tau \rightarrow \left\langle \begin{array}{l} [x_1] \rightarrow \langle \textit{eq1} \rangle \rightarrow [y_1] \\ [x_2] \rightarrow \langle \textit{eq2} \rangle \rightarrow [y_2] \\ [x_3] \rightarrow \langle \textit{eq3} \rangle \rightarrow [y_3] \\ \dots \end{array} \right\rangle \rightarrow [X]^{\tau+1}$$

$$[X]^\tau \rightarrow \left\langle \begin{array}{l} [x_1] \rightarrow \langle \textit{LUT1} \rangle \rightarrow [y_1] \\ [x_2] \rightarrow \langle \textit{LUT2} \rangle \rightarrow [y_2] \\ [x_3] \rightarrow \langle \textit{LUT3} \rangle \rightarrow [y_3] \\ \dots \end{array} \right\rangle \rightarrow [X]^{\tau+1}$$

$$[X]^\tau \rightarrow \langle \textit{LUT} \rangle \rightarrow [X]^{\tau+1}$$

```
pef= 0.591-0.39*V SHEAR (D+ 0.953*(V SHEAR (D**2) - 0.0496*(V SHEAR (D**3)))
```

```
if (pef>edtm ax) pef=edtm ax
```

```
if (pef<bedtm in) pef=edtm in
```

!— cloud base precip efficiency

```
zkbc=z(ikbcon(i))*3.281e-3
```

```
prezk= 0.2
```

```
if (zkbc>3.) then
```

```
prezk = 9.6729352+zkbc*(-.70034167+zkbc*(.162179896+zkbc* &  
(-.12569798E-2+zkbc*(4.2772E-4-zkbc*5.44E-6))))
```

```
endif
```

```
if (zkbc>25) then
```

```
prezk=2.4
```

```
endif
```

```
pefb = 1./0.+prezk)
```

```
if (pefb>edtm ax) pefb=edtm ax
```

```
if (pefb<bedtm in) pefb=edtm in
```

```
EDT(D) = 1.-5*(pefb+pef)
```

```
einc = edt(i)/float(m axens2+1.)
```

```
do k=1,m axens2
```

```
edtc(ik)=edt(i)+float(k-m axens2/2-1.)*einc
```

```
edtc(ik)=edt(i)-float(k)*einc
```

```
enddo
```

{ VSHEAR (wind shear)  
z (cloud base height) } → ⟨code⟩ → edtc (ratio between downdraft and updraft mass flux)

$$m_0 = \frac{\beta I_1 m_b}{I_2} = \xi m_b$$

```

pef=(1.591-.639*V S H E A R (I)+.0953*(V S H E A R (I)**2) - .00496*
(V S H E A R (I)**3))

```

```

if (pef.gt.edtm ax) pef=edtm ax

```

```

if (pef.lt.edtm in) pef=edtm in

```

```

!--- cloud base precip efficiency

```

```

zkbc=z(i-kbcon(i))*3.281e-3

```

```

prezk=.02

```

```

if (zkbc.gt.3.) then

```

```

    prezk = .96729352+zkbc*(-.70034167+zkbc*(.162179896+zkbc*

```

```

        (-1.2569798E-2+zkbc*(4.2772E-4-zkbc*5.44E-6)))

```

```

endif

```

```

if (zkbc.gt.25) then

```

```

    prezk=2.4

```

```

endif

```

```

pefb = 1./(1.+prezk)

```

```

if (pefb.gt.edtm ax) pefb=edtm ax

```

```

if (pefb.lt.edtm in) pefb=edtm in

```

```

E D T (I) = 1.-.5*(pefb+pef)

```

```

einc = edt(i)/float(m axens2+1)

```

```

do k=1,m axens2

```

```

    edtc(i-k)=edt(i)+float(k-m a

```

```

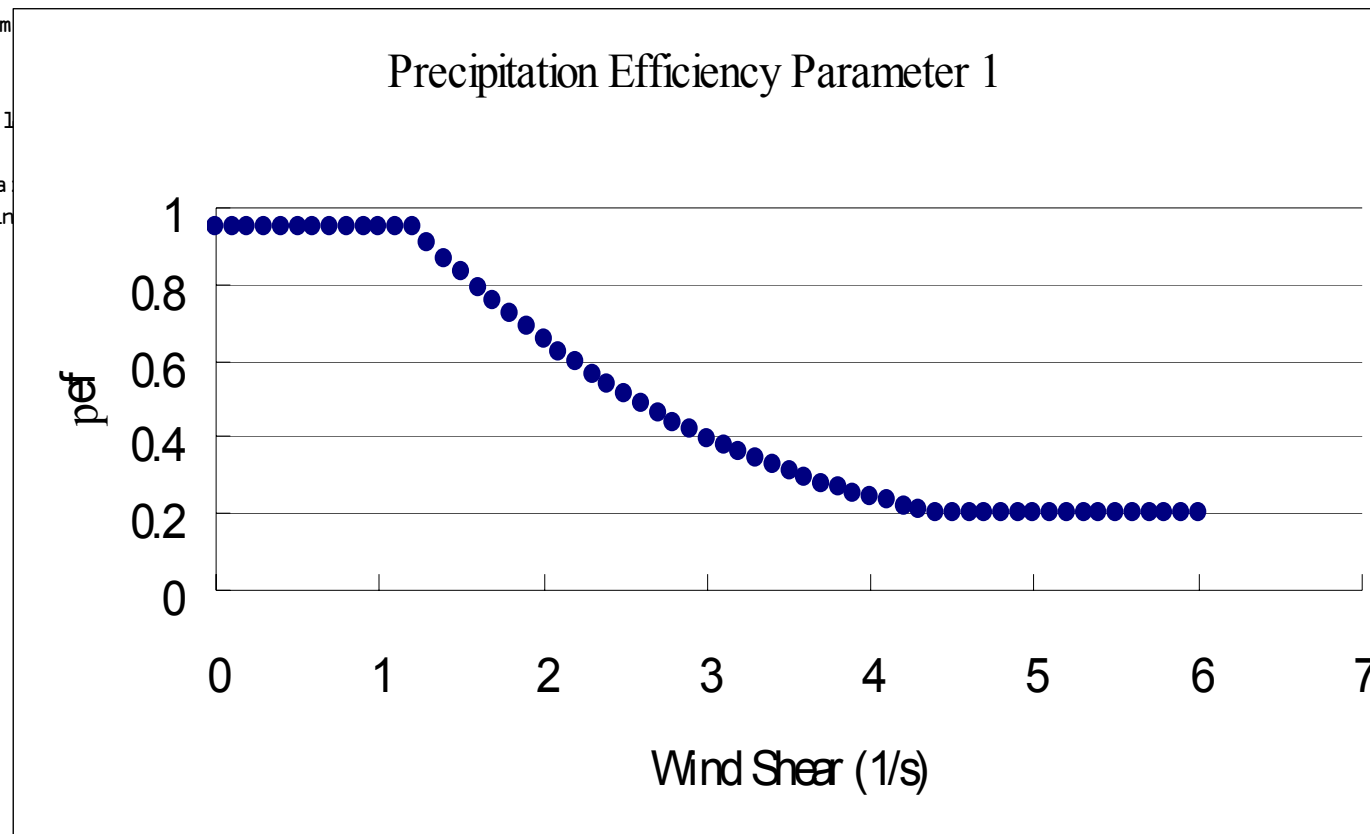
    edtc(i-k)=edt(i)-float(k)*einc

```

```

enddo

```



```

p ef = (1.591-.639*V S H E A R (I)+.0953*(V S H E A R (I)**2) - .00496*
(V S H E A R (I)**3))
if (p ef.gt.edtm ax) p ef= edtm ax
if (p ef.lt.edtm in) p ef= edtm in

```

!--- cloud base precip efficiency

```

zkbc=z(i,kbcon(i))*3.281e-3
prezk=.02
if (zkbc.gt.3.) then
prezk = .96729352+zkbc*(-.70034167+zkbc*(.162179896+zkbc*

```

&

```

(-1.2569798E-2+zkbc*(4.2772E-4-zkbc*5.44E-6))))
endif

```

```

endif
if (zkbc.gt.25) then
prezk=2.4
endif

```

```

endif

```

```

p efb = 1./(1.+prezk)
if (p efb.gt.edtm ax) p efb= edtm ax
if (p efb.lt.edtm in) p efb= edtm in

```

```

E D T (I) = 1.-.5*(p efb+p ef)

```

```

einc = ed t(i)/float(m axens2+1)

```

```

do k=1,m axens2
ed t(i-k)= ed t(i)+ float(k-m axens2)
ed t(i-k)= ed t(i)-float(k)*einc
enddo

```

```

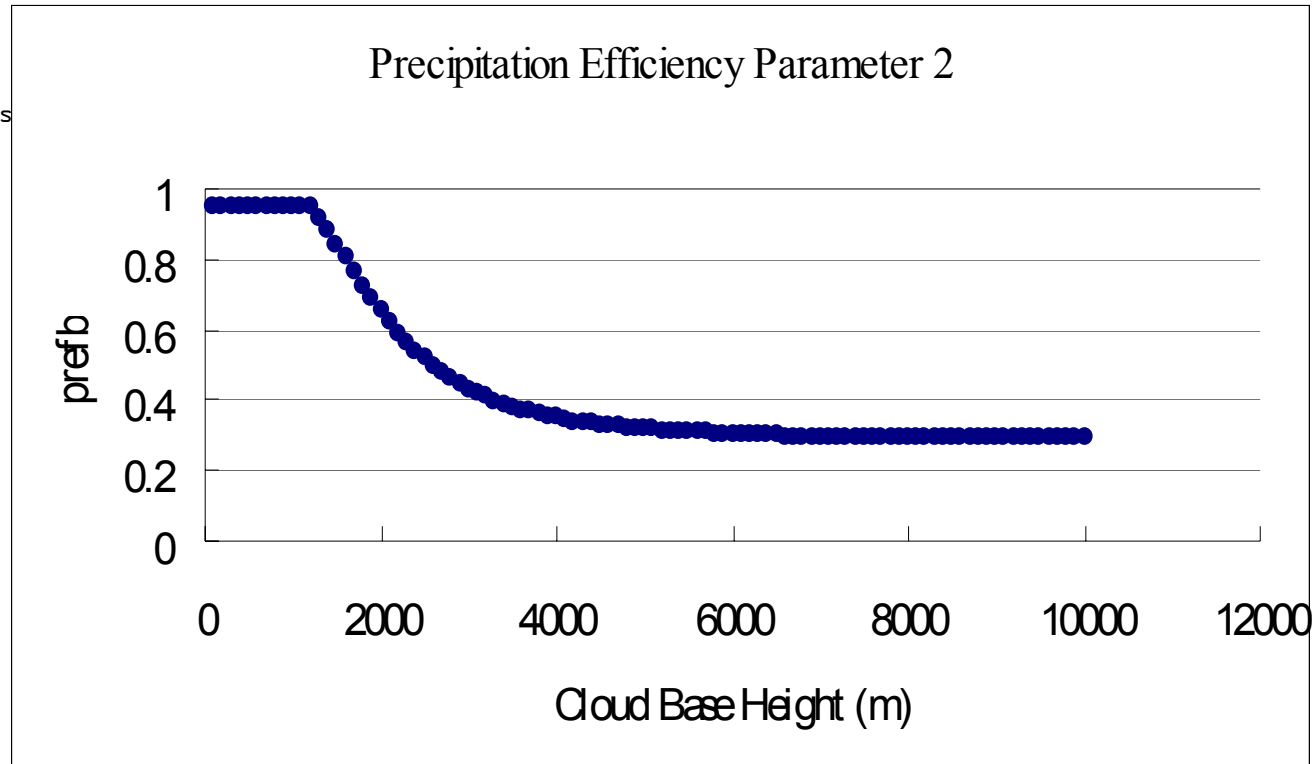

```

```


```

```


```



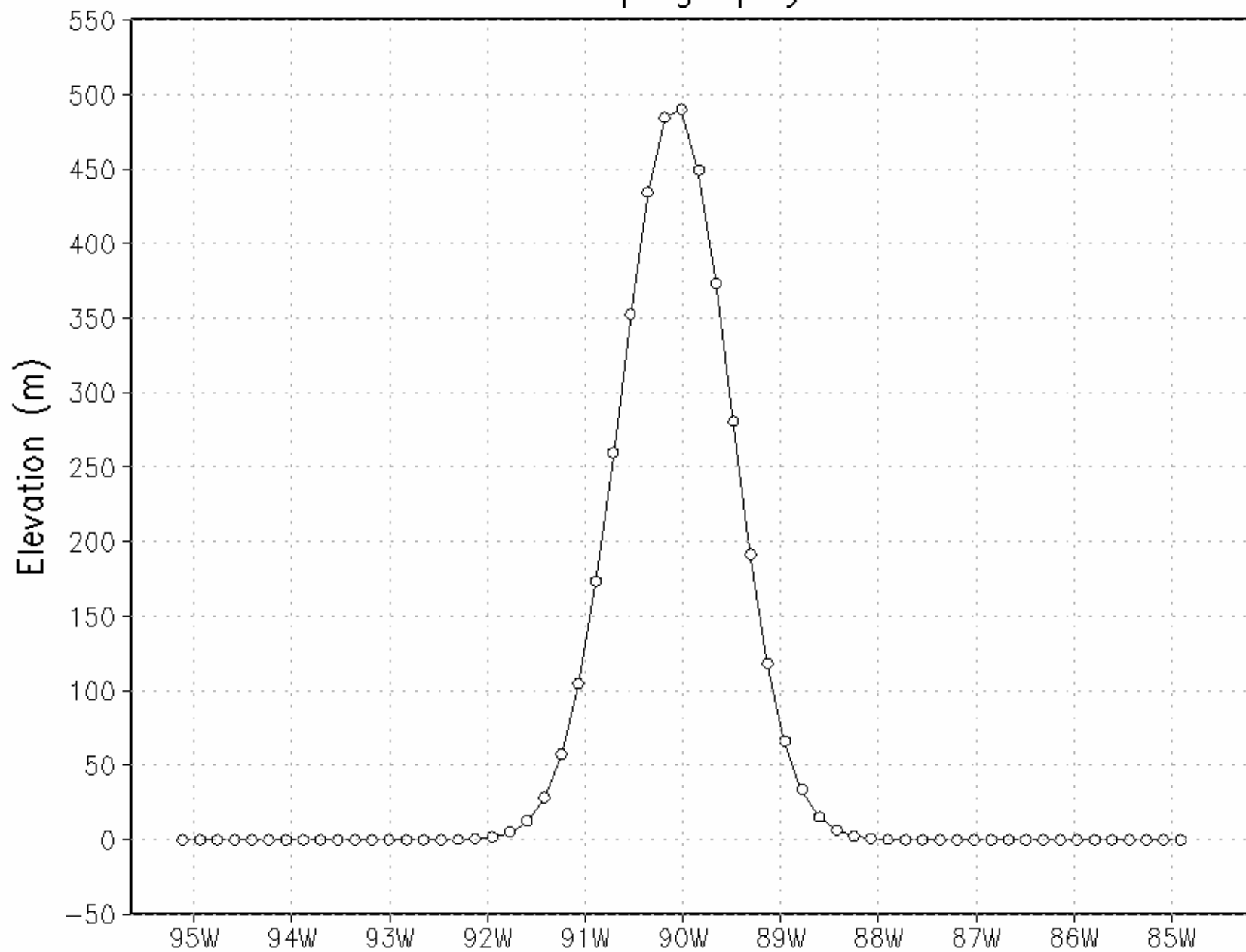
# Procedure to Implement LUT Approach

- Find out the range for VSHEAR and z
- Given VSHEAR and z, run the piece of code offline to get the LUT for  $edtc=f(VSHEAR,z)$
- Replace the piece of code with LUT reading code and linear interpolation
- Check simulation results
- Estimate model efficiency change and resources usage

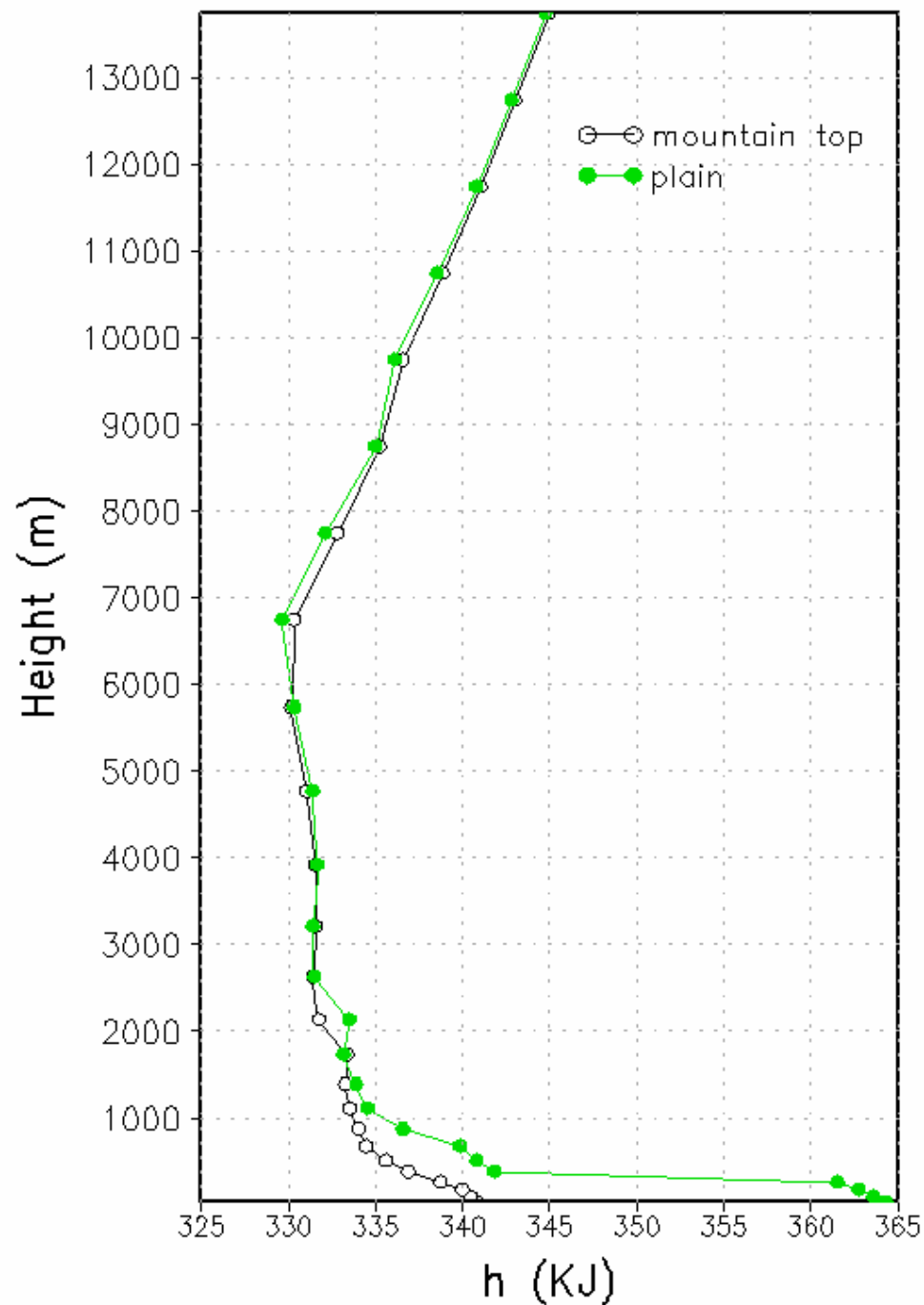
# RAMS Model Options and Setups

- **Basic Equations**  
Nonhydrostatic; compressible
- **Grid Stagger/Structure**  
Arakawa C grid; One Grid
- **Convective Parameterization**  
Grell scheme
- **Surface Layer Flux**  
None
- **Boundary Condition**  
Cyclic for momentum; zero gradient inflow/outflow for others
- **Radiation Parameterization**  
Harrington scheme
- **Initialization**  
Unstable sounding; horizontally homogeneous
- **Terrain**  
N-S mountain range in the middle (Gaussian function)
- **Grid Spacing**  
16 km x 16 km x (60 m ~ 1000 m)
- **Time Step**  
30 secs
- **Grid Cell Number**  
60 x 60

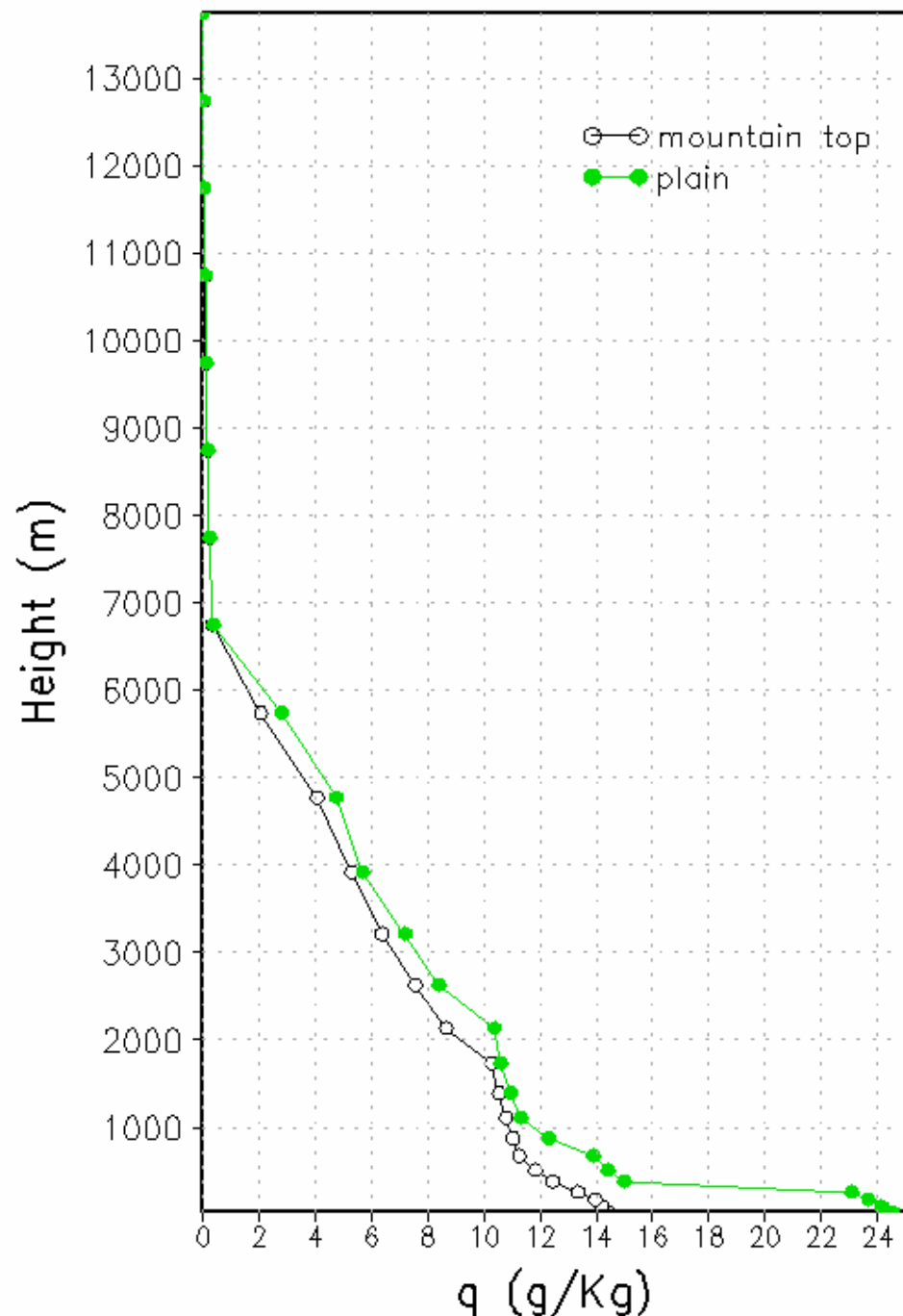
# Topography



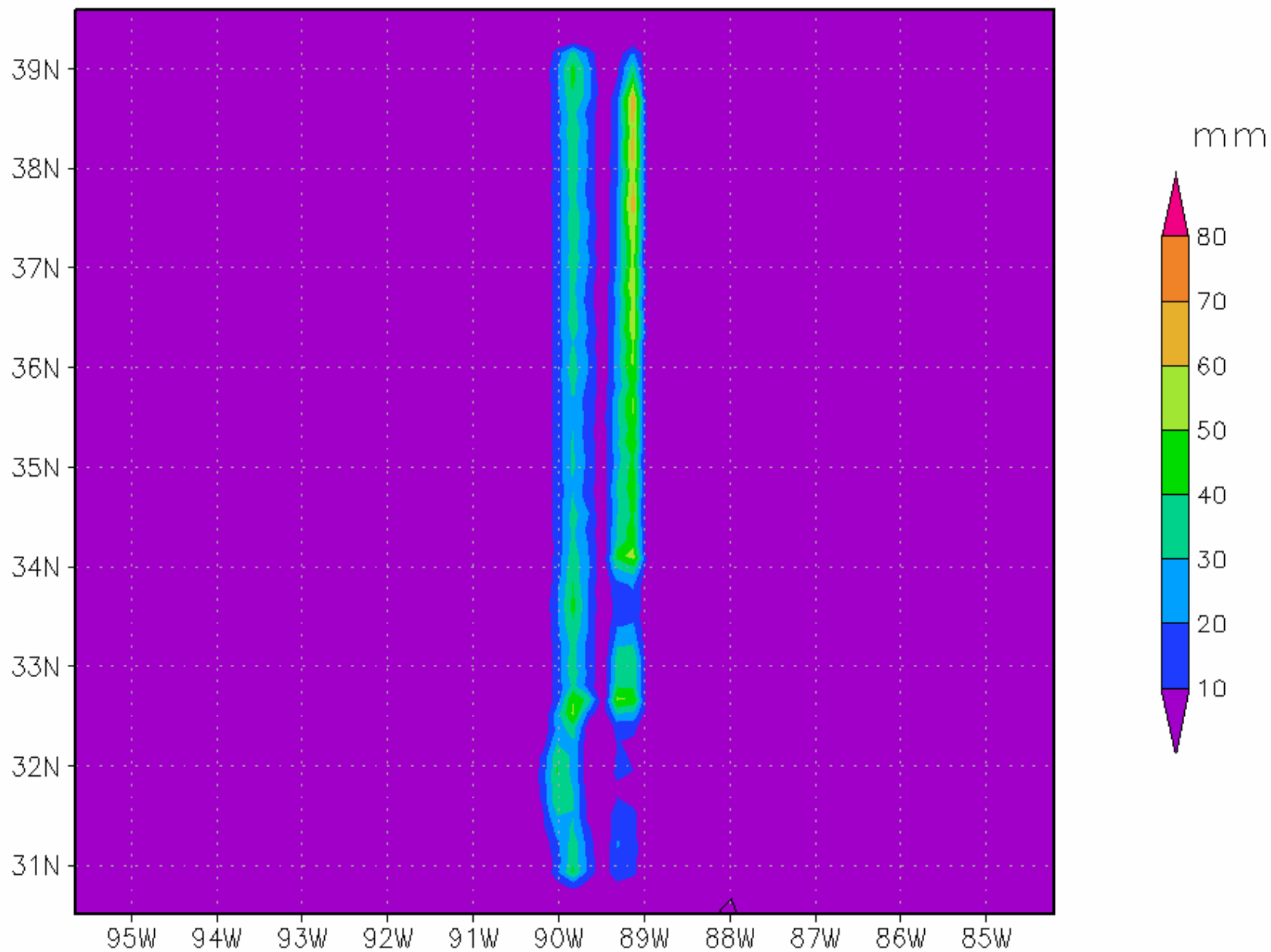
Initial Moist Static Energy



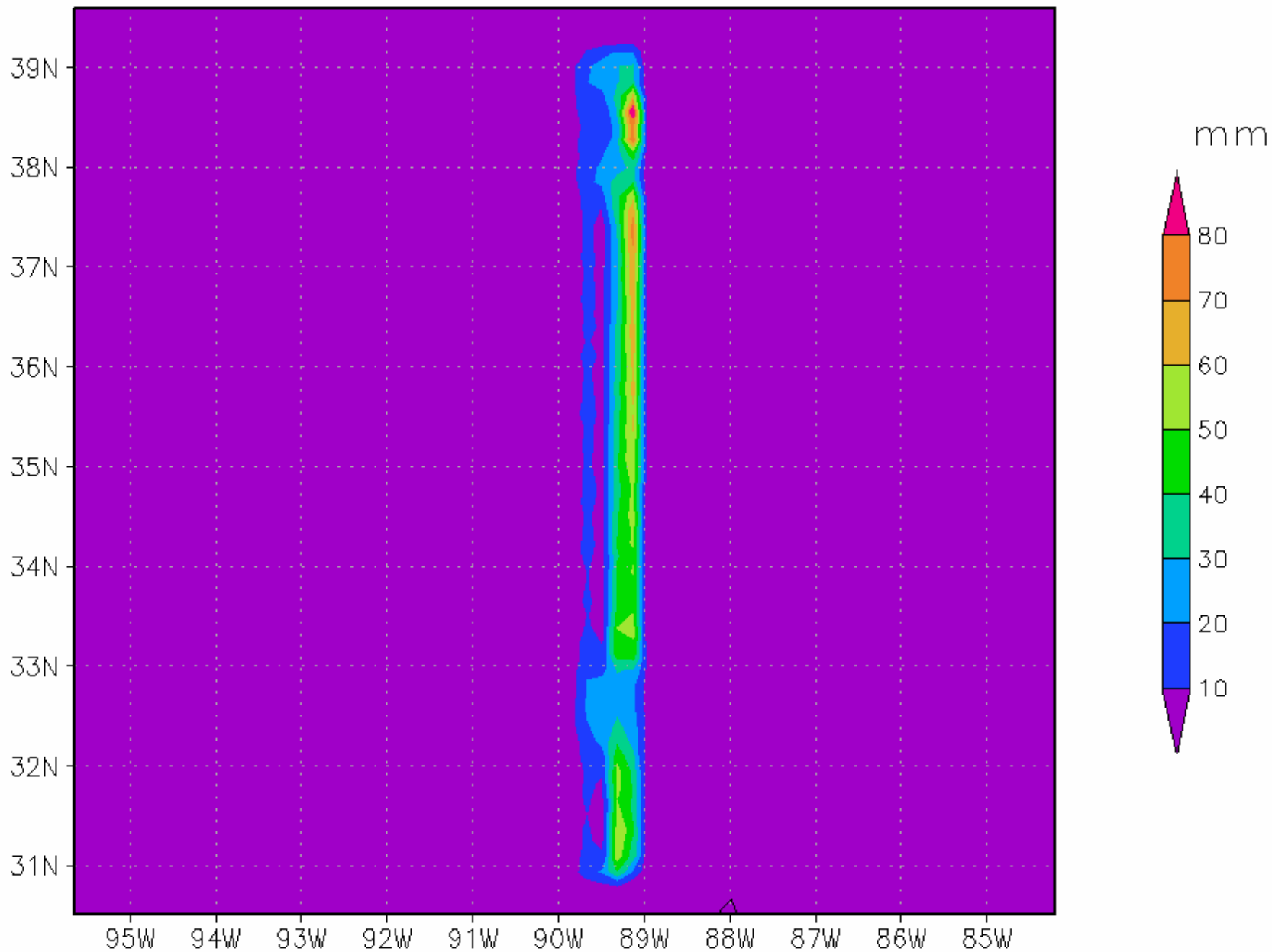
Initial Mixing Ratio



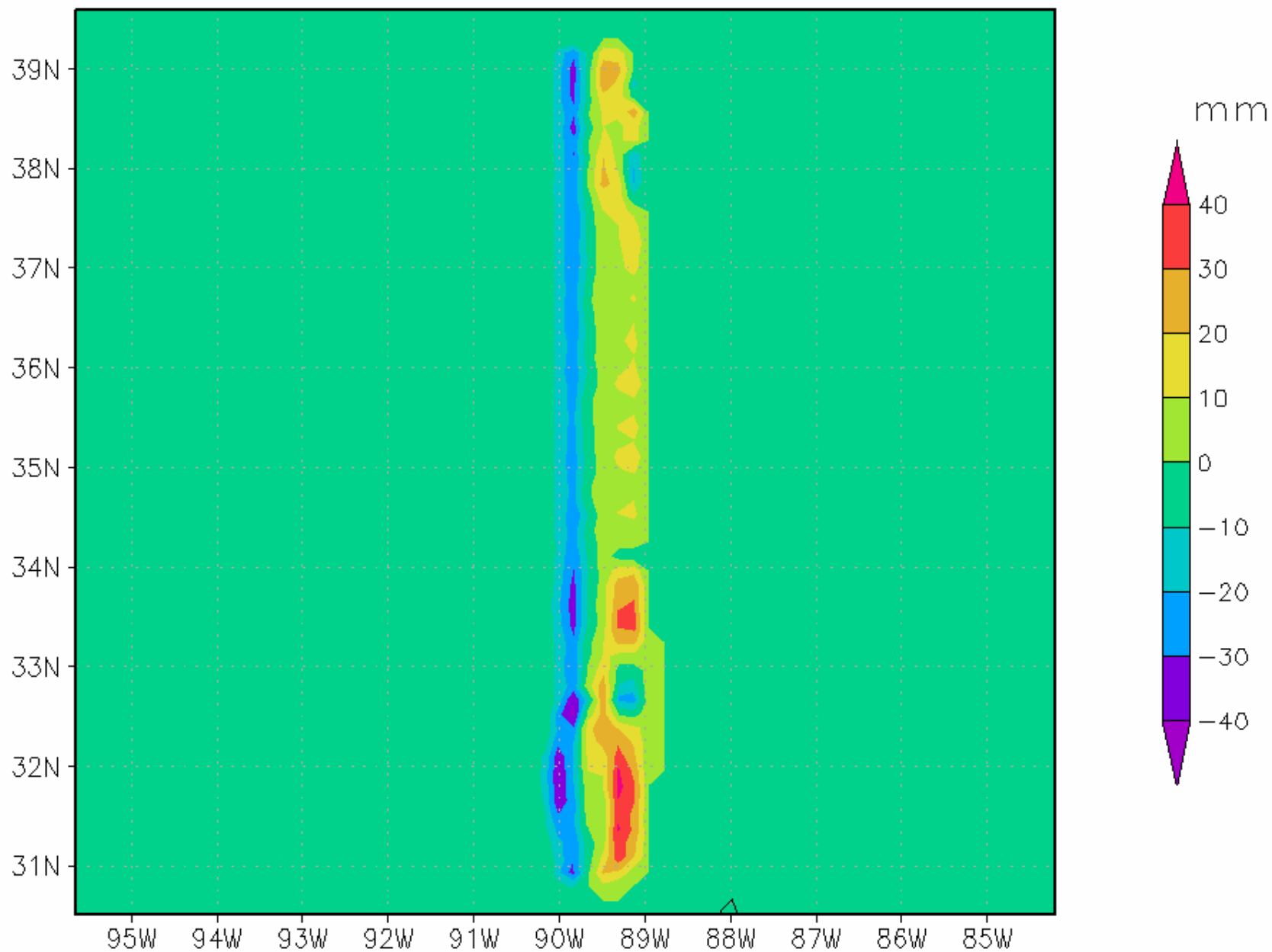
# Convective Precipitation – Explicit



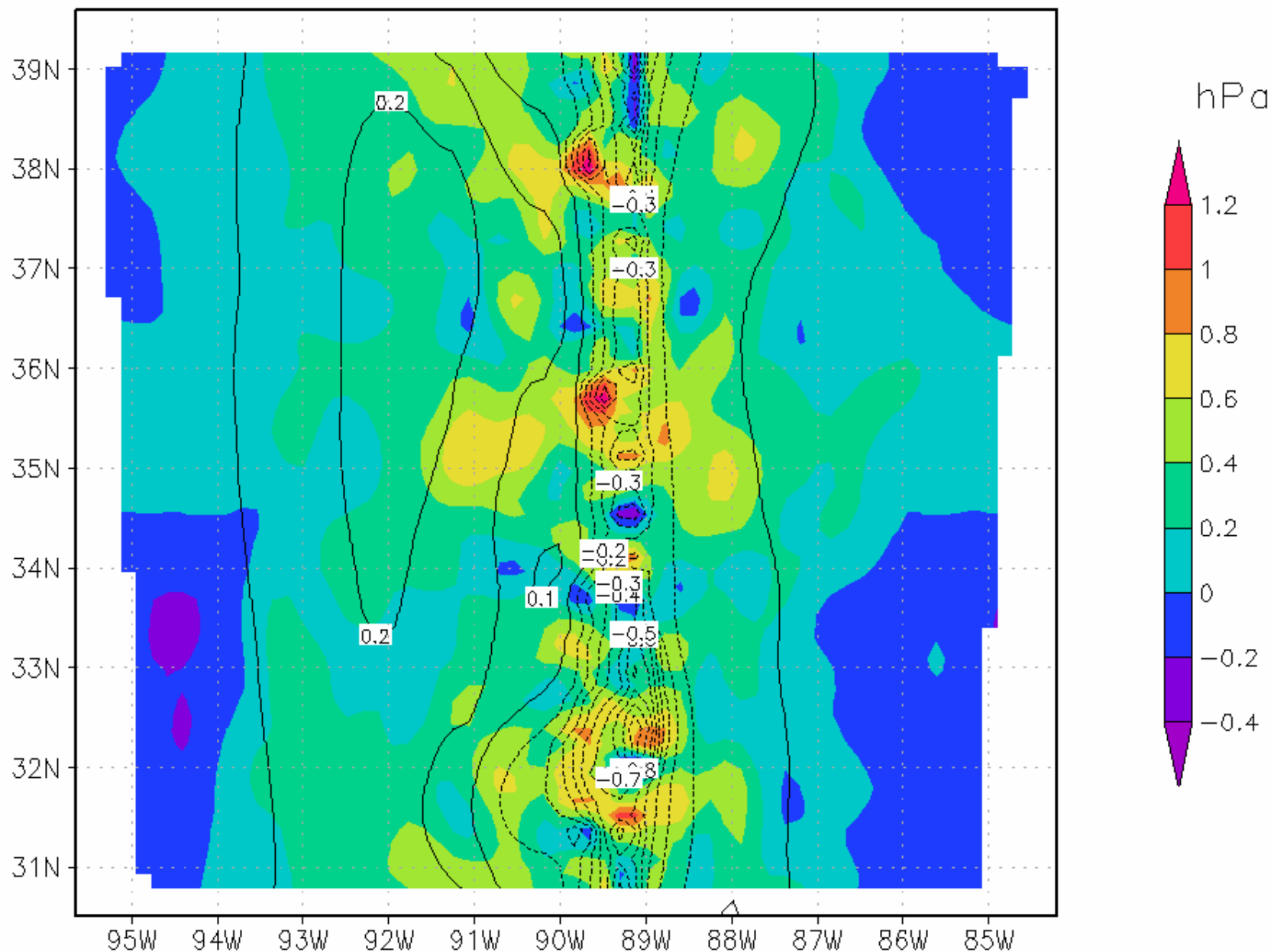
# Convective Precipitation – LUT



# Convective Precipitation Difference

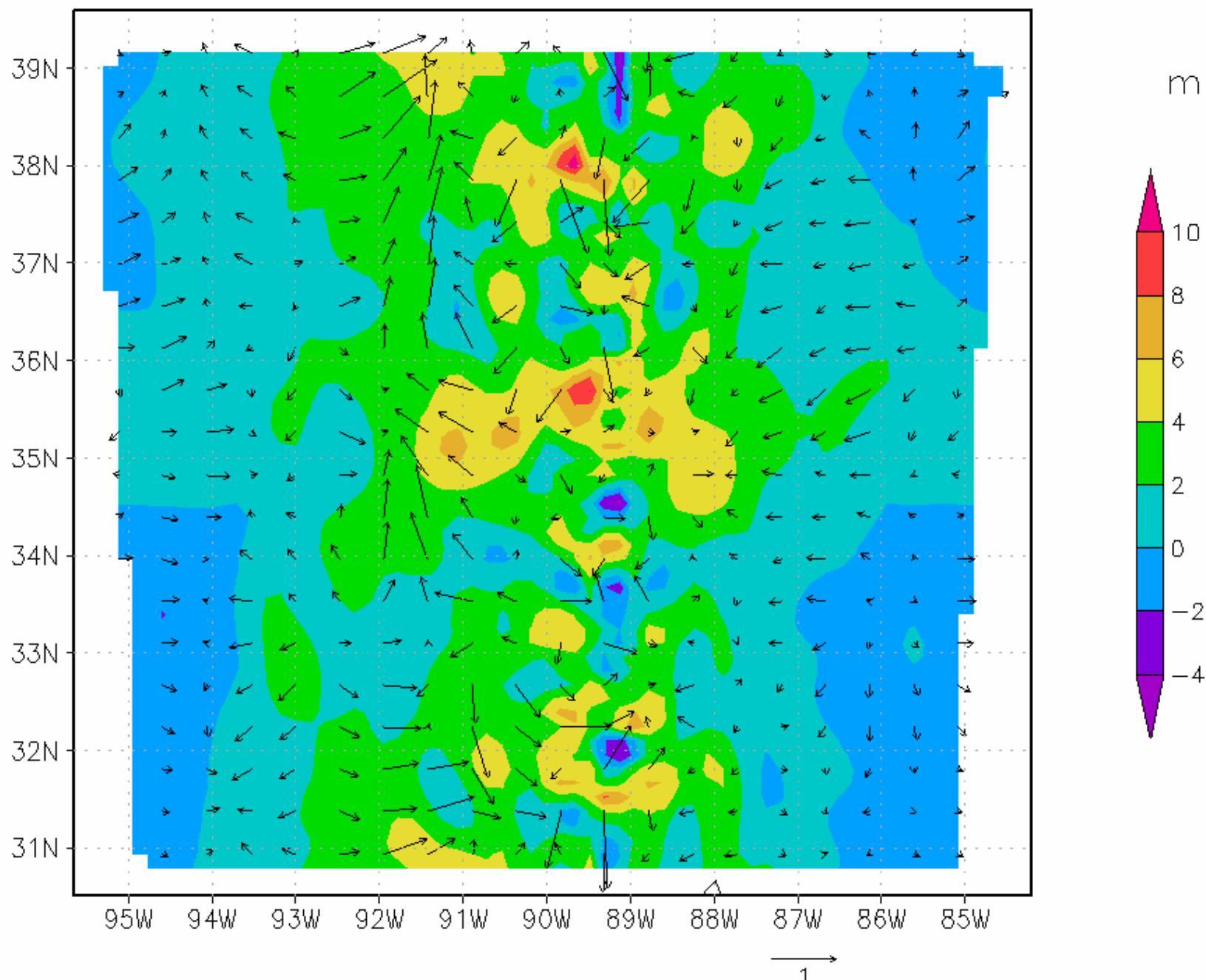


# Sea Surface Pressure and Lowest Level Temperature Difference

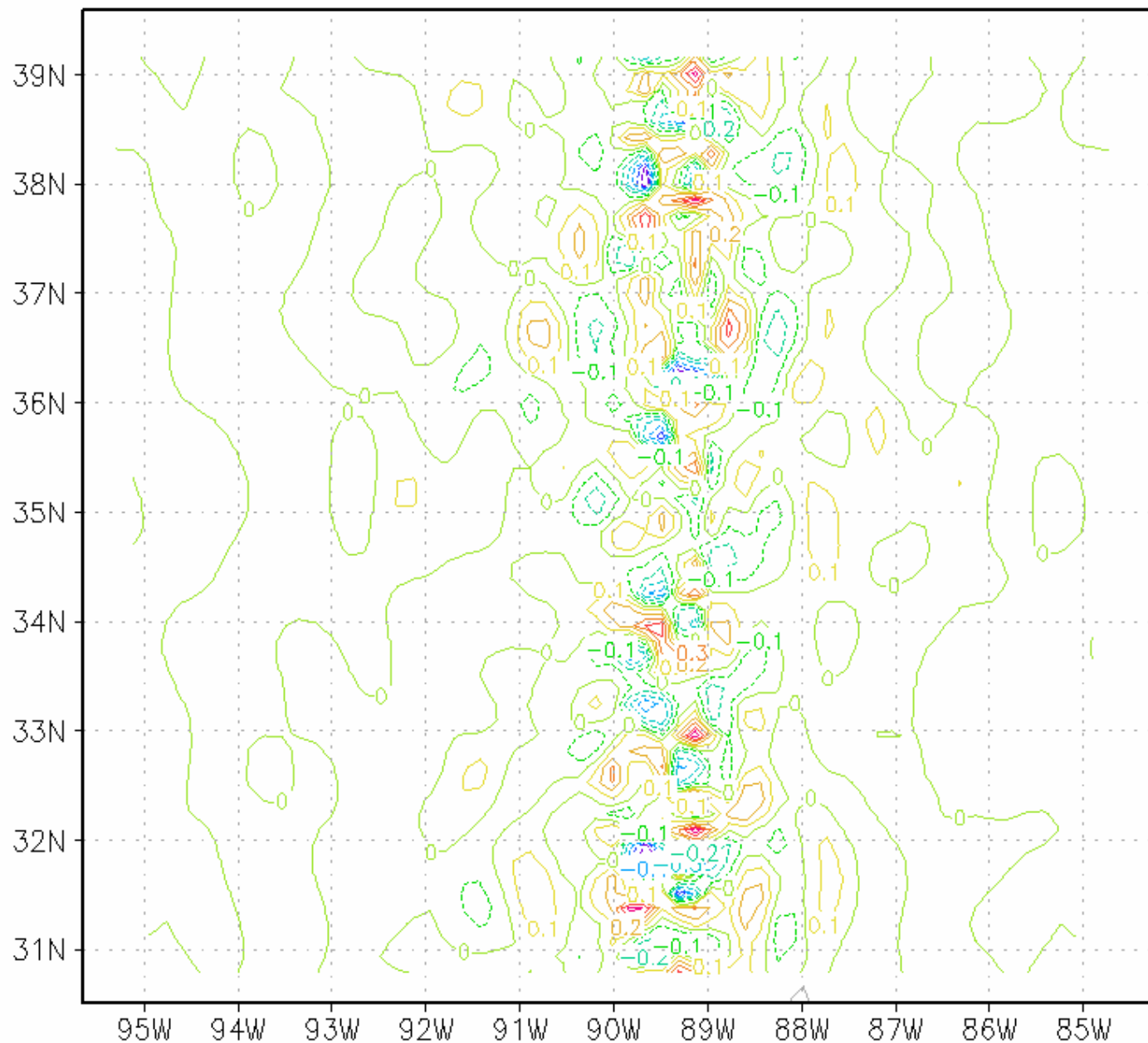




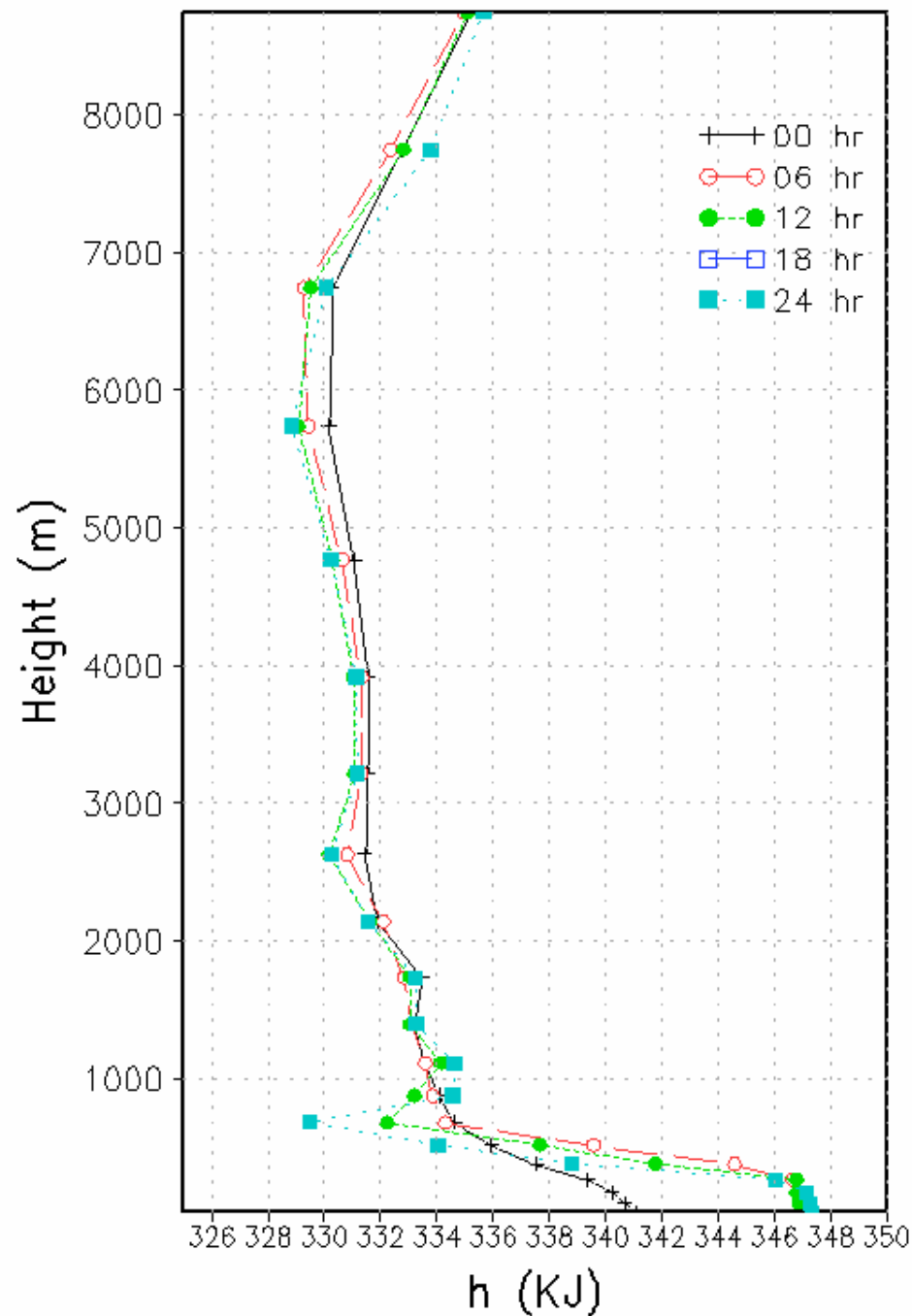
# Geopotential Height and Wind Vector Difference at 850 hPa



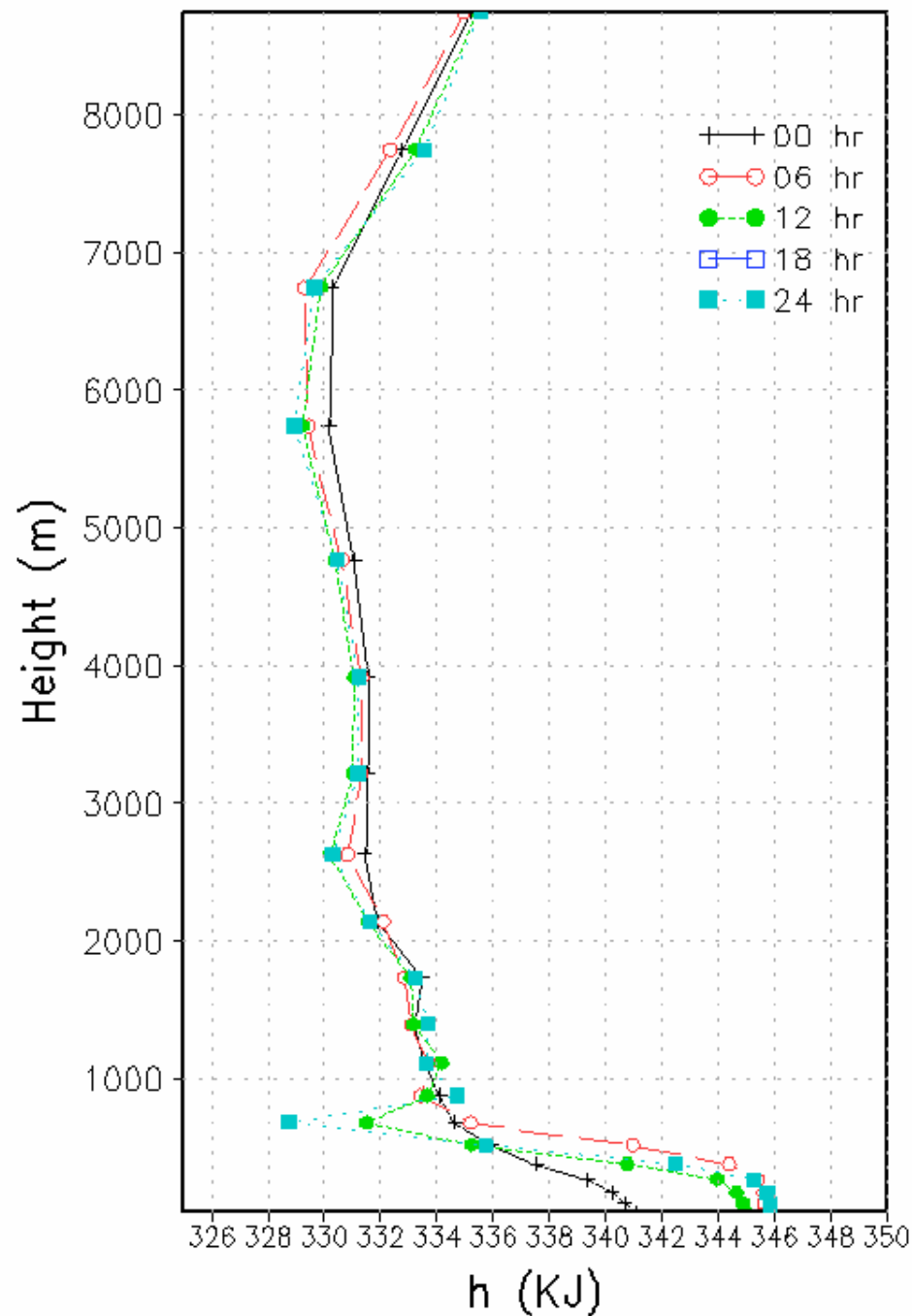
# Vertical Velocity Difference at 500 hPa



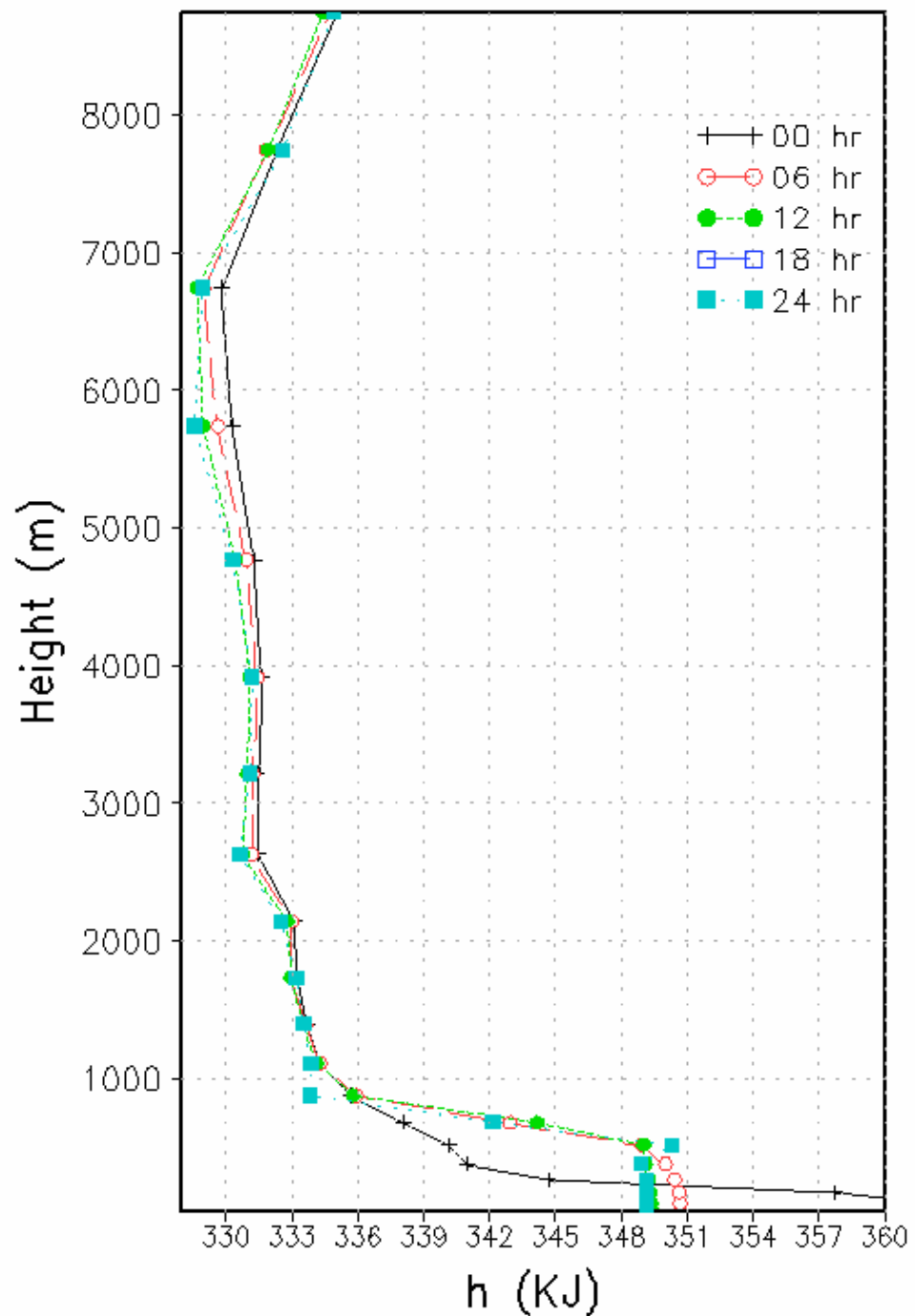
h at Lon=-89.836, Explicit



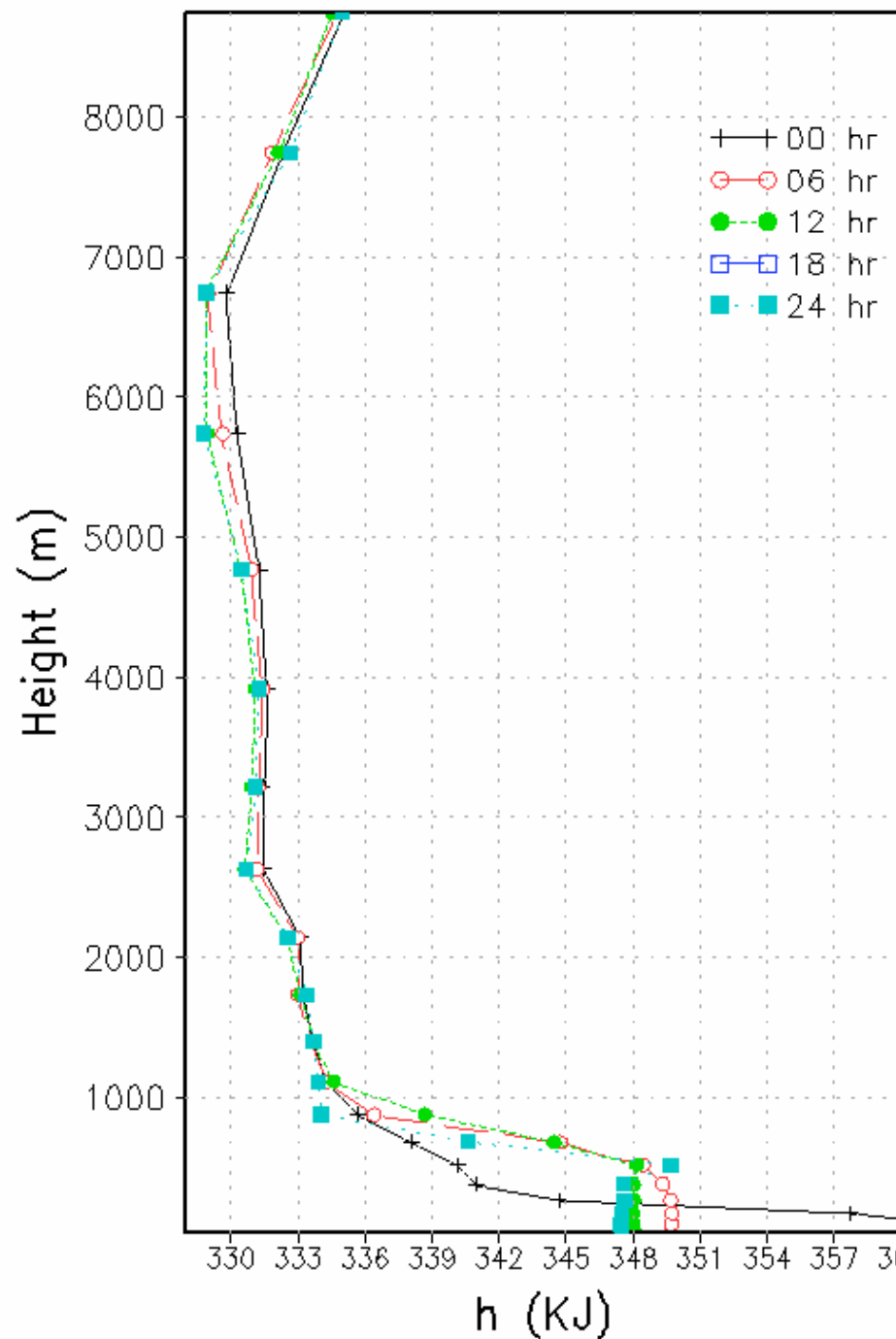
h at Lon=-89.836, LUT



h at Lon=-89.132, Explicit



h at Lon=-89.132, LUI



# Efficiency and Resources Usage

- 4mins/100mins faster in LUT than in Explicit for this sensitivity test
- Need ~ 64 Kbytes memory to store the table

# Conclusion

- More than a dozen tunable parameters
- Very sensitive to mass flux ratio
- LUT method is a little faster than the original code, but needs ~64 K memory for each node

# Reference

- Grell, G. A., J. Dudhia, and D. R. Stauffer, A description of the Fifth-generation Penn State/NCAR Mesoscale Model (MM5), *NCAR Tech Note TN-398 + STR*, 122 pp., 1994.
- G. A. Grell and D. Devenyi, 2002. A generalized approach to parameterizing convection combining ensemble and data assimilation techniques, *Geophysical Research Letters*, 29(14), No. 38.
- Arakawa, A., 2004. Review Article: The cumulus parameterization problem: Past, Present, and Future, *Journal of Climate*, 17(13), 2493-2525.
- Grell, G. A., 1993. Prognostic evaluation of assumptions used by cumulus parameterizations, *Monthly Weather Review*, 121, 764-787.