UCLA JIFRESSE
Joint Institute for Regional Earth System Science & Engineering

NASA

A robust representation of rain microwave radiances for data assimilation and to estimate the 1st few radial modes of heating, vertical motion, precipitable water, total ice and rain

Post-doc #1: Jeff Steward, UCLA

Post-doc #2: Hsiao-Chieh Tseng, UC Davis

Post-doc #3: Sahra Kacimi, NASA/JPL

Tomi Vukicevic, ShuHua Chen, Svetla Hristova-Veleva and Ziad Haddad

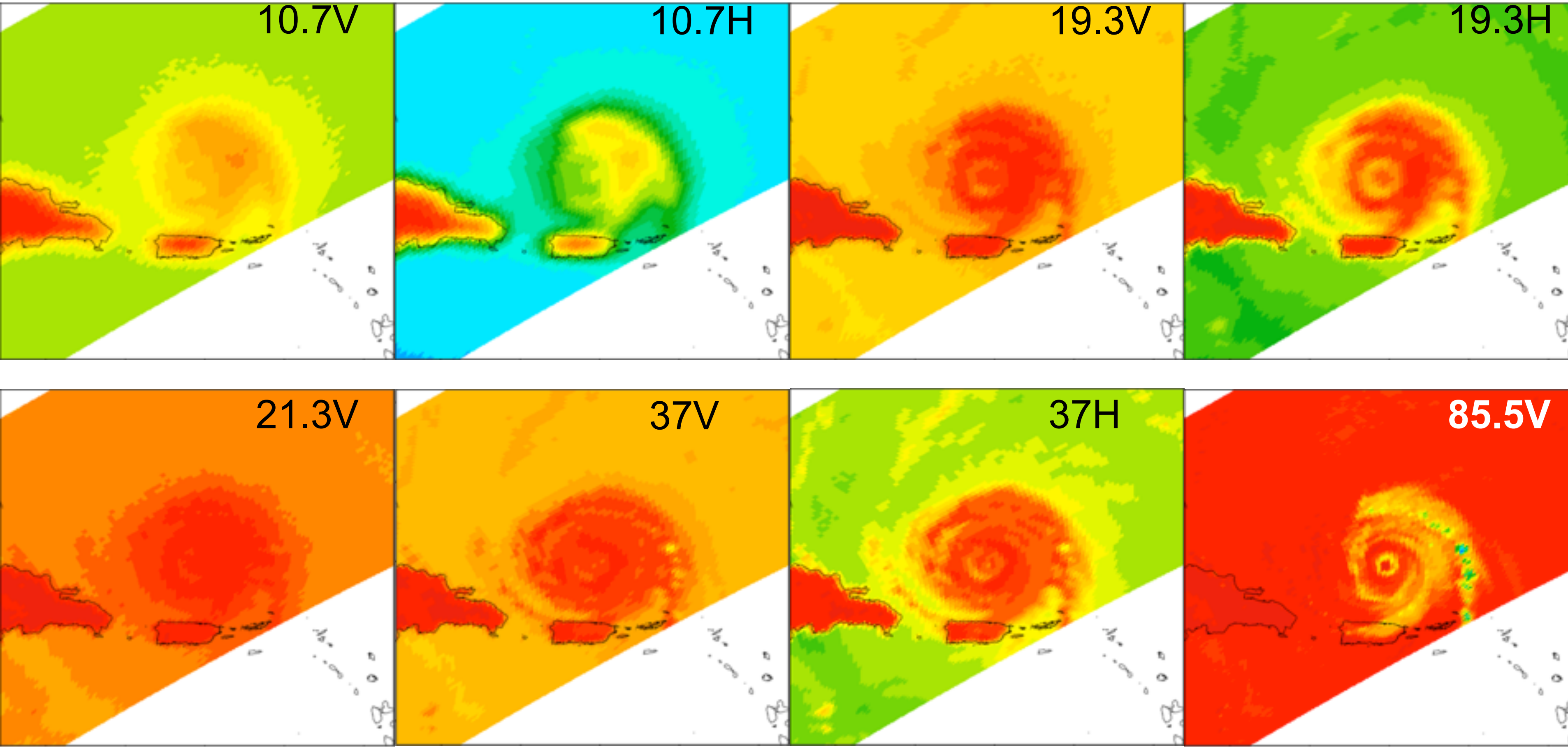
1) Microwave measurements over clouds are quite sensitive to the water content in its different phases, but

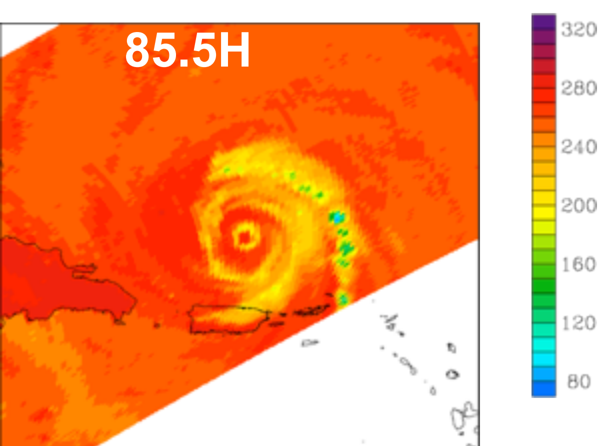
2) they are also sensitive to other variables: temperature, surface wind, hydrometeor size and shape

3) hydrometeor properties are difficult to characterize in an efficient expression for the dependence of the obs on the variables

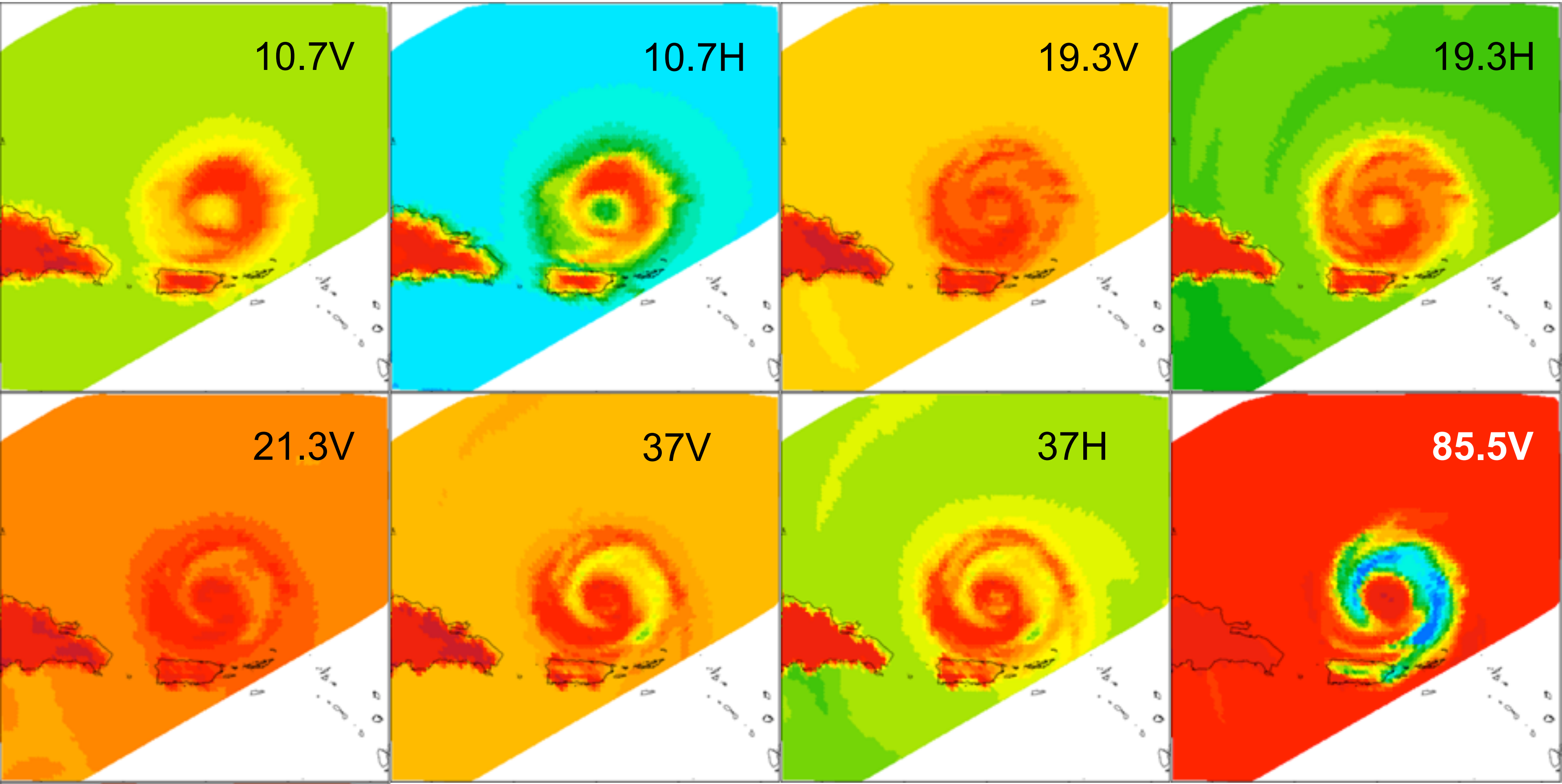
4) hydrometeor properties are not prognostic variables, they are poorly represented in the model

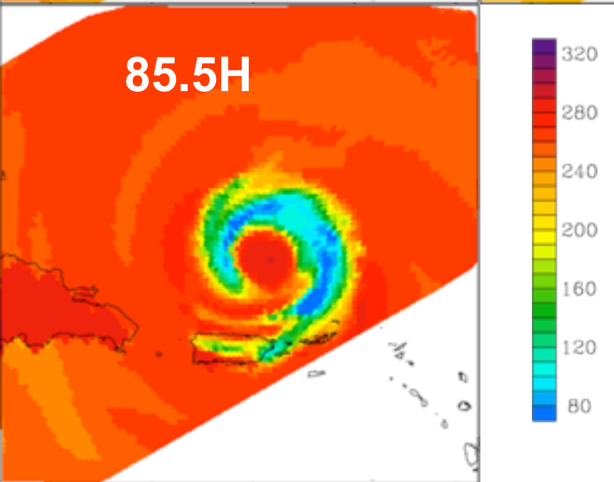
5) the measurements are noisy and they are not mutually independent pieces of information about the underlying variables, and the radiative transfer calculation is sensitive to these hard-wired hydrometeor parameters



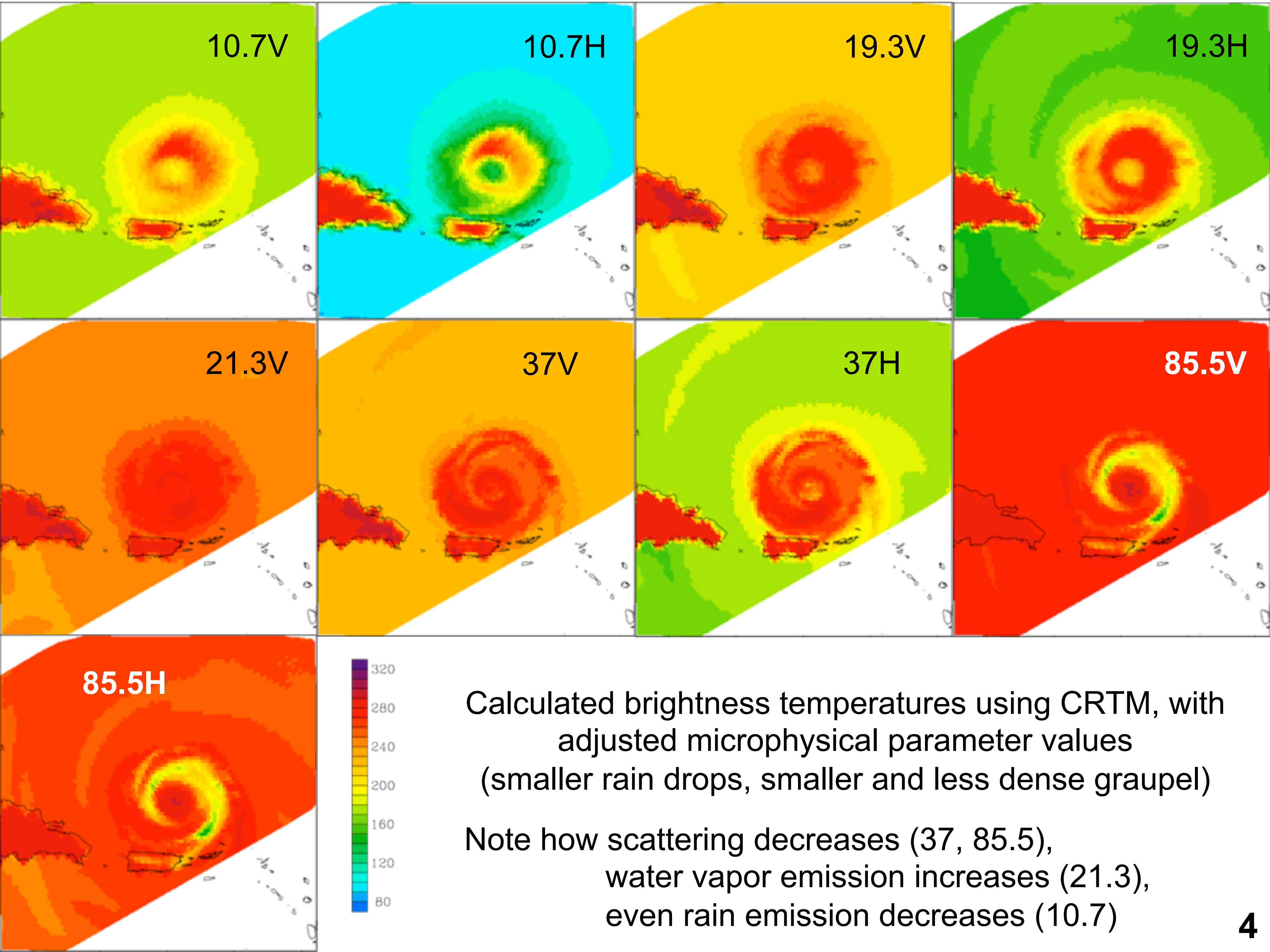


TRMM TMI pass over Earl, 30 August 2010
(ACTUAL MEASUREMENTS)



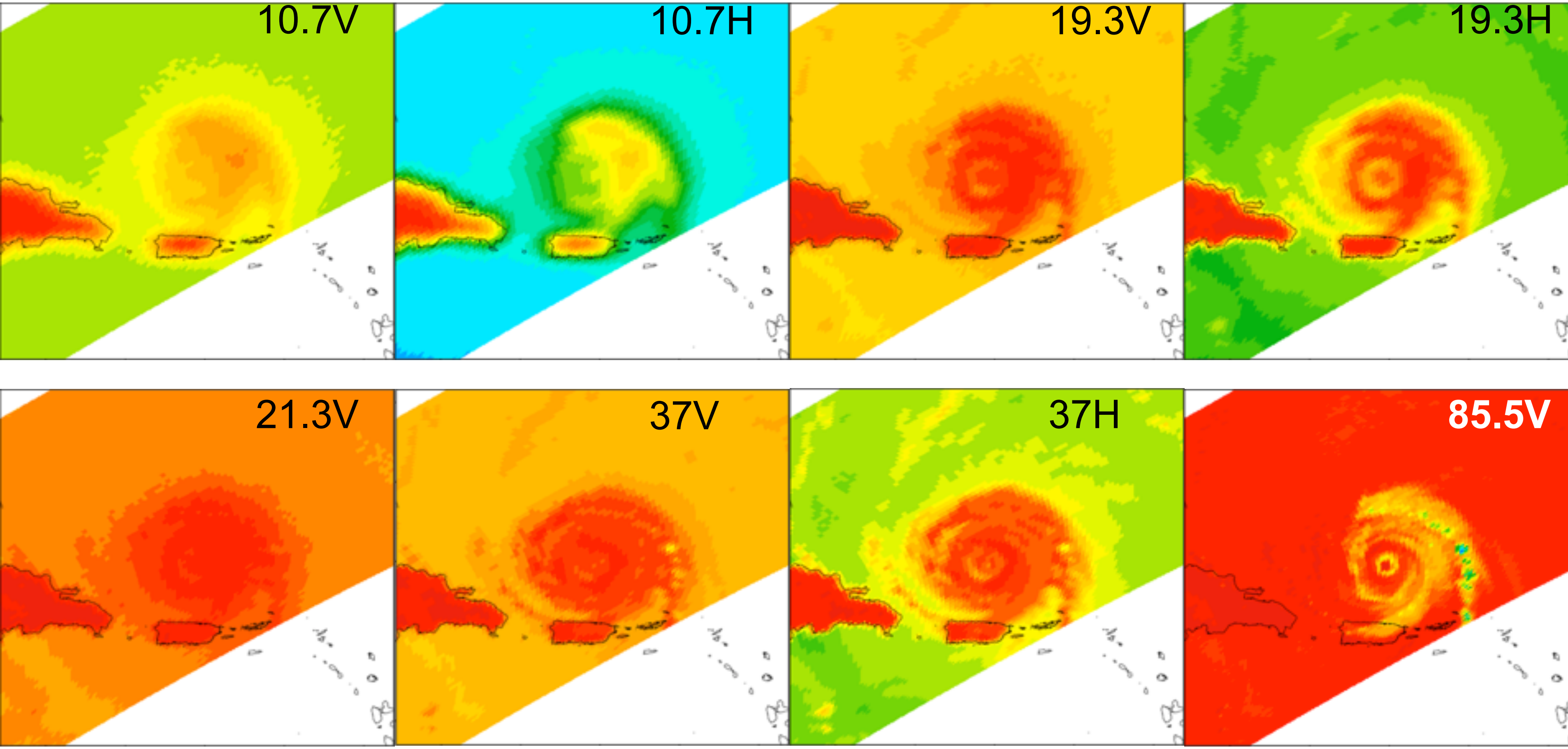


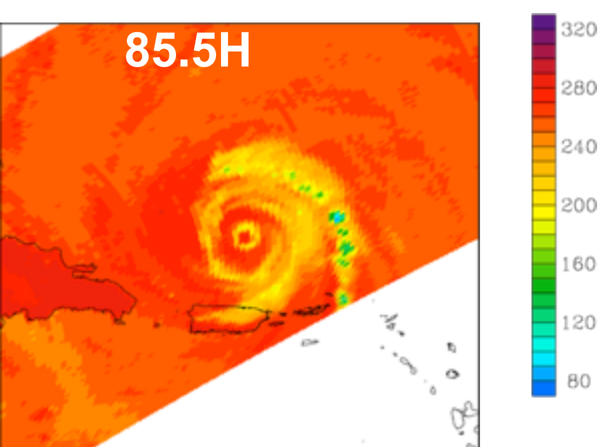
Calculated brightness temperatures using CRTM, with "HWISS model 2" microphysical parameter values at nearest forecast time



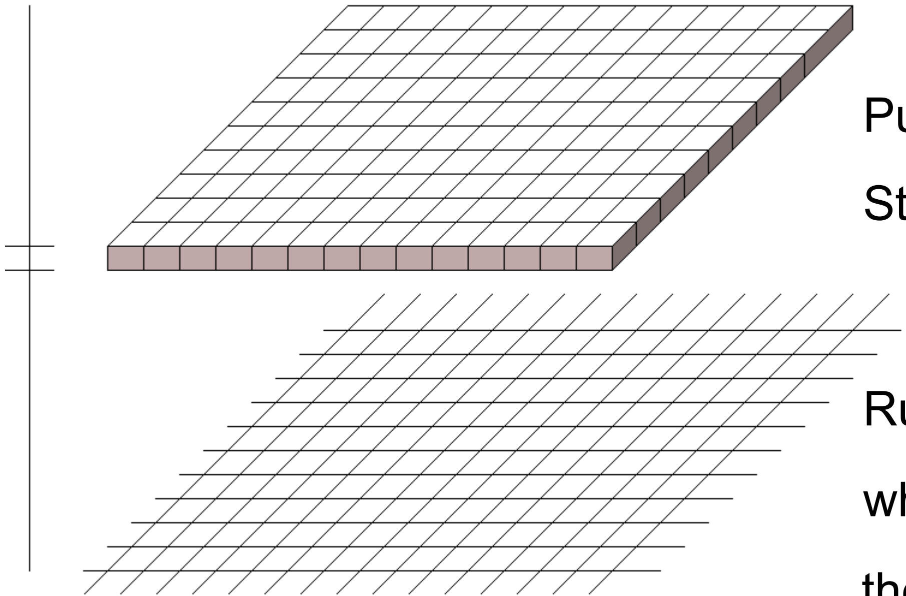
Calculated brightness temperatures using CRTM, with adjusted microphysical parameter values (smaller rain drops, smaller and less dense graupel)

Note how scattering decreases (37, 85.5),
water vapor emission increases (21.3),
even rain emission decreases (10.7)





TRMM TMI pass over Earl, 30 August 2010
(ACTUAL MEASUREMENTS,
DUPLICATE of slide 2)



Unknowns (variables): in each volume element,
$x = (T, p, u, v, w, q_{wv}, q_{cl}, q_{pl}, q_{ci}, q_s, q_g, q_h)$

Put them all together in one **X** living in $\mathbf{R}^{126,000,000}$

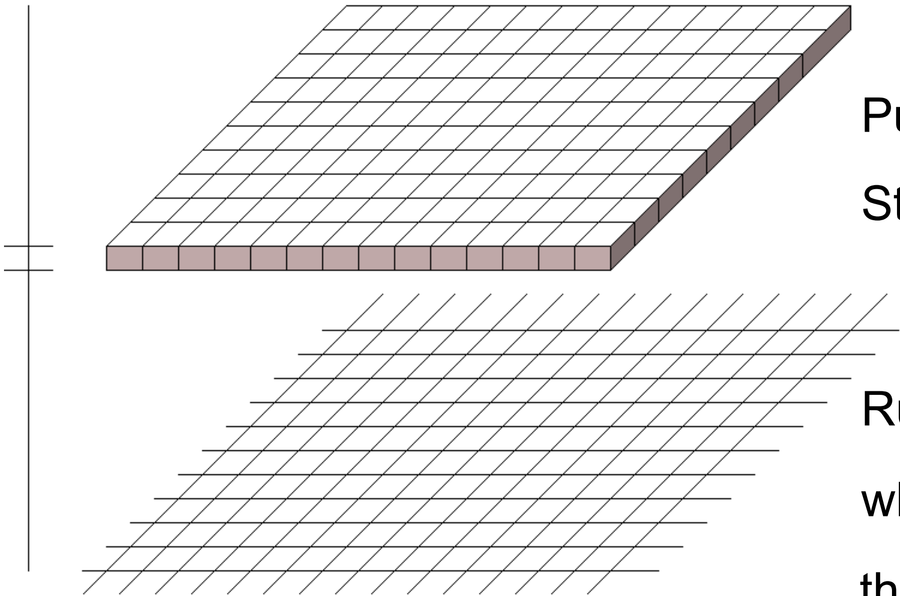
Start with $\mathbf{X} = \mathbf{X}_0$, initial condition known up to (Gaussian) error with imperfectly known higher moments (covariance)

Run dynamics $d\mathbf{X}_t = F(\mathbf{X}_t; \lambda)\, dt$

where F is nonlinear and has parameters,

then, at time s, observe $O = H(\mathbf{X}_s; \lambda')$ + error,
where nonlinear H depends on parameters λ'
whose dynamics are not known.

Goal: find "**X**" consistent with dynamics-only $\mathbf{X}_s$ and such that H(**X**) is consistent with O



Unknowns (variables): in each volume element,
$x = (T, p, u, v, w, q_{wv}, q_{cl}, q_{pl}, q_{ci}, q_s, q_g, q_h)$

Put them all together in one **X** living in $\mathbf{R}^{126,000,000}$

Start with $\mathbf{X} = \mathbf{X}_0$, initial condition known up to (Gaussian) error with imperfectly known higher moments (covariance)

Run dynamics $d\mathbf{X}_t = F(\mathbf{X}_t; \lambda)\, dt$

where F is nonlinear and has parameters,

then, at time s, observe $O = H(\mathbf{X}_s; \lambda') + \text{error}$, where nonlinear H depends on parameters λ' whose dynamics are not known.

Goal: find "**X**" consistent with dynamics-only $\mathbf{X}_s$ and such that H(**X**) is consistent with O

Parameters λ in the definition of the dynamics F: microphysics

- partitioning between cloud liquid, precipitating liquid, "small ice" (50 μm) and large ice
- hydrometeor size moments

Parameters λ' in the definition of the observation H:

- all of λ along with scattering efficiencies for different habits (snow, graupel, hail)

Ideally: try to express the brightness temps O independently from λ

⇒ Hard, because O depends on λ

⇒ so: one option is to have an empirical representation:

1. off-line, for a given $(x_1, x_2, \ldots, x_{504})$, calculate radiances with different λ,
2. store the answers in a large database
3. in real-time, for a given $(x_1, x_2, \ldots, x_{504})$, calculate radiances by referring to the database:

$$O(x_1, x_2, \ldots, x_{504}) = \Sigma \; T_b^{(n)} \exp(-[x_1-x_1^{(n)}]^2 - [x_2-x_2^{(n)}]^2 - \ldots - [x_{504}-x_{504}^{(n)}]^2)$$

Two obvious problems with this:

- too many variables (504), not all of which are important for O
- how many samples should be in database for it to be representative, and how many samples (n) should be included in every sum?

Ideally: try to express the brightness temps O independently from λ

⇒ Hard, because O depends on λ

⇒ so: one option is to have an empirical representation:

1. off-line, for a given (x_1, x_2, …, x_{504}), calculate radiances with different λ, different schemes, different sub-resolution
2. store the answers in a large database
3. in real-time, for a given (x_1, x_2, …, x_{504}), calculate radiances by referring to the database:

$$O(x_1, x_2, \ldots, x_{504}) = \Sigma \; T_b^{(n)} \exp\left(-[x_1-x_1^{(n)}]^2 - [x_2-x_2^{(n)}]^2 - \ldots - [x_{504}-x_{504}^{(n)}]^2 \right)$$

Two obvious problems with this:

- too many variables (504), not all of which are important for O
- how many samples should be in database for it to be representative, and how many samples (n) should be included in every sum?

Instead, try to distill the variables $(x_1, x_2, \ldots, x_{504})$ into a smaller number which would capture the main sensitivities of the brightness temperatures, namely

- the absorption/emission
- the scattering
- the surface temperature
- the wind speed

so about 4 distilled variables y_1, y_2, y_3, y_4:

$$O_i(x_1, x_2, \ldots, x_{504}) = \Sigma\, T_i^{(n)} \exp(-[y_1-y_1^{(n)}]^2-[y_2-y_2^{(n)}]^2-[y_3-y_3^{(n)}]^2-[y_4-y_4^{(n)}]^2)$$

(with y_1, y_2, y_3, y_4 calculated from $x_1, x_2, \ldots, x_{504}$),

instead of

$$O\,(x_1, x_2, \ldots, x_{504}) = \Sigma\ T_b^{(n)} \exp(-[x_1-x_1^{(n)}]^2 -[x_2-x_2^{(n)}]^2-\ldots-[x_{504}-x_{504}^{(n)}]^2)$$

i.e.

y_1	y_2	y_3	y_4	O_1'	O_2'	O_3'
$y_1^{(1)}$	$y_2^{(1)}$	$y_3^{(1)}$	$y_4^{(1)}$	$O_1'^{(1)}$	$O_2'^{(1)}$	$O_3'^{(1)}$
…	…	…	…	…	…	…
$y_1^{(n)}$	$y_2^{(n)}$	$y_3^{(n)}$	$y_4^{(n)}$	$O_1'^{(n)}$	$O_2'^{(n)}$	$O_3'^{(n)}$

instead of

x_1	x_2	…	x_{504}	O_1	…	O_9
$x_1^{(1)}$	$x_2^{(1)}$	…	$x_{504}^{(1)}$	$O_1^{(1)}$	…	$O_9^{(1)}$
…	…	…	…	…	…	…
$x_1^{(n)}$	$x_2^{(n)}$	…	$x_{504}^{(n)}$	$O_1^{(n)}$	…	$O_9^{(n)}$

Methodology

• Start with HWRF simulations (say HEDAS Earl 2010 h3vk, 2010-08-29-12Z to 2010-09-03-18Z), using stream ψ, potential χ, P, T, RH, W, q_{cliq}, q_r, q_{cli}, q_s, q_g, q_h at 42 vertical levels for a total of 504 variables $x_1, \ldots, x_{504}$

• for each of these 12million columns, forward-calculate $T_{b1}, \ldots, T_{b9}$

• find the principal components $x_1', \ldots, x_{504}'$ (each is a linear combo of $x_1, \ldots, x_{504}$) and the principal components $T_1', \ldots, T_9'$ (each a linear combo of $T_1', \ldots, T_9'$)

• Then we will have to find combos of $x_1',\ldots, x_{504}'$ that correlate most with combos of $T_1',\ldots, T_9'$

• Say these combos are x_1'', x_2'', x_3'' and T_1'', T_2'', T_3'': we finally need to express the latter in terms of the former, in a differentiable way (to be able to compute derivatives)

Methodology

- Start with HWRF simulations (say HEDAS Earl 2010 h3vk, 2010-08-29-12Z to 2010-09-03-18Z), using stream ψ, potential χ, P, T, RH, W, q_{cliq}, q_r, q_{cli}, q_s, q_g, q_h at 42 vertical levels for a total of 504 variables $x_1, \ldots, x_{504}$
- for each of these 12million columns, forward-calculate $T_{b1}, \ldots, T_{b9}$

- Step 1: find the principal components $x_1', \ldots, x_{504}'$
- Step 2: find the principal components $T_1', \ldots, T_9'$

- Step 3: find
  combos of $x_1',\ldots, x_{504}'$ that correlate most with combos of $T_1',\ldots, T_9'$

Methodology

- Start with HWRF simulations (say HEDAS Earl 2010 h3vk, 2010-08-29-12Z to 2010-09-03-18Z), using stream ψ, potential χ, P, T, RH, W, q_{cliq}, q_r, q_{cli}, q_s, q_g, q_h at 42 vertical levels for a total of 504 variables x_1, … , x_{504}
- for each of these 12million columns, forward-calculate T_{b1} , … , T_{b9}

- Step 1: find the principal components x_1', … , x_{504}'
- Step 2: find the principal components T_1', … , T_9'

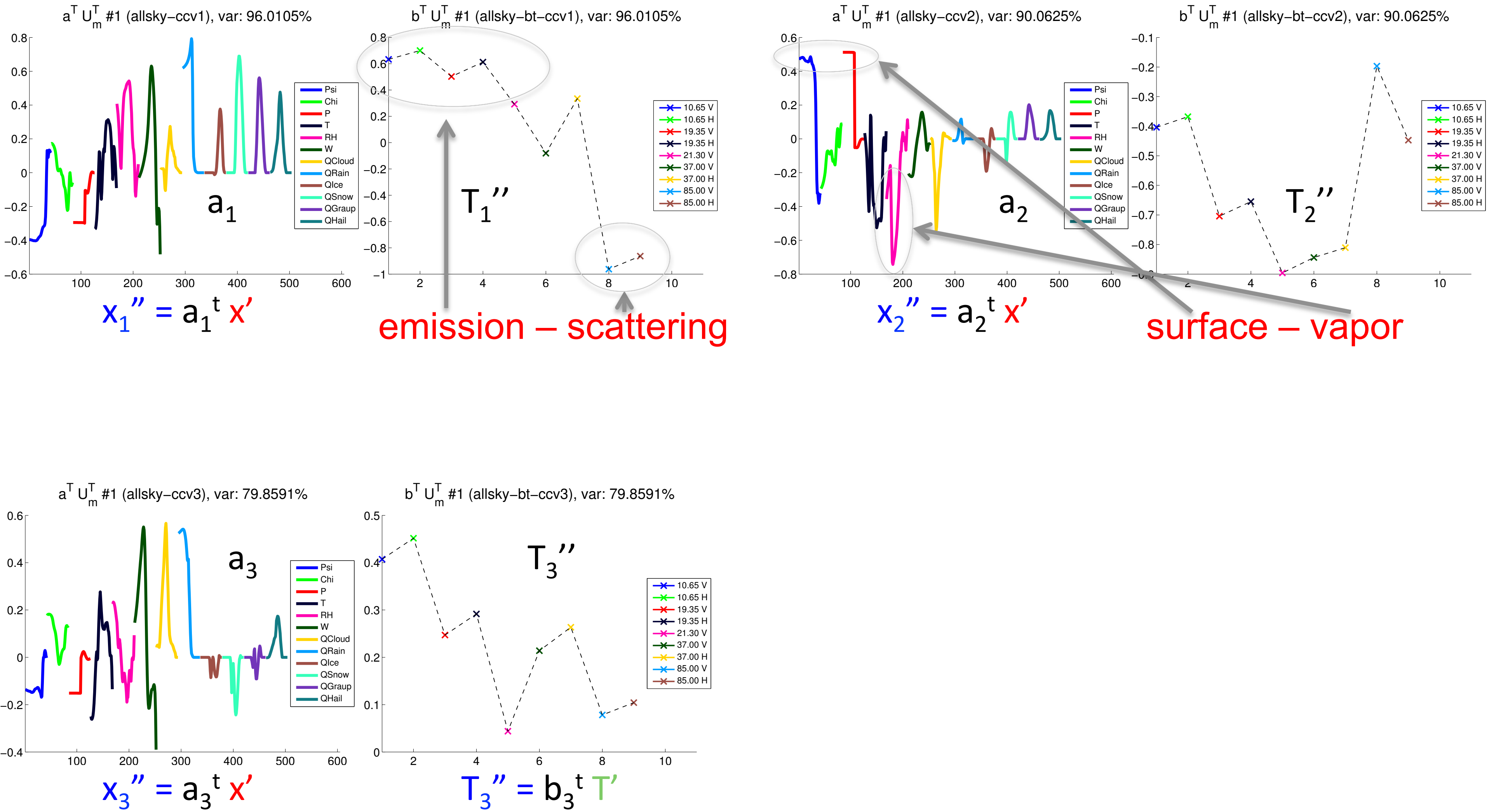
- Step 3: find
  combos of x_1',…, x_{504}' that correlate most with combos of T_1',…, T_9'

  and express T_1'', T_2'', T_3'' in terms of x_1'', x_2'', x_3''
  (with differentiable expression, in order to compute derivatives):

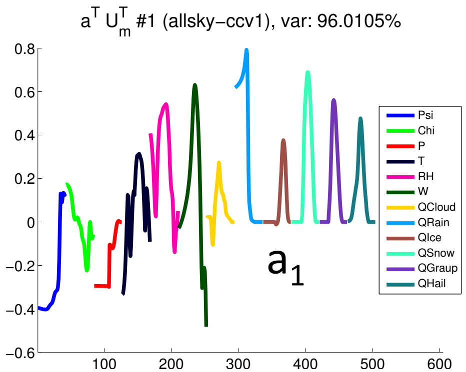
$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

where the weighted sum over n runs over the 12million training points

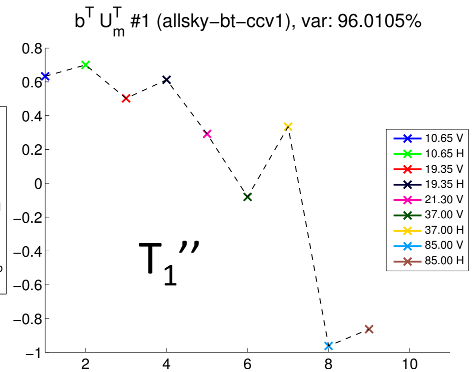
First part of step 3: here are the first 3 x” and T”



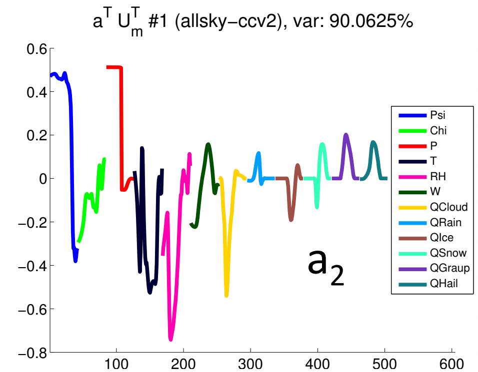
First part of step 3: here are the first 3 x” and T”



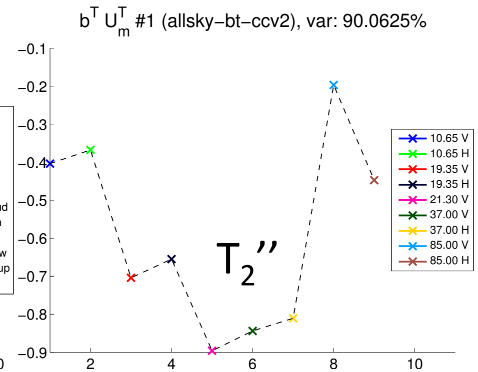
$x_1'' = a_1^t\, x'$



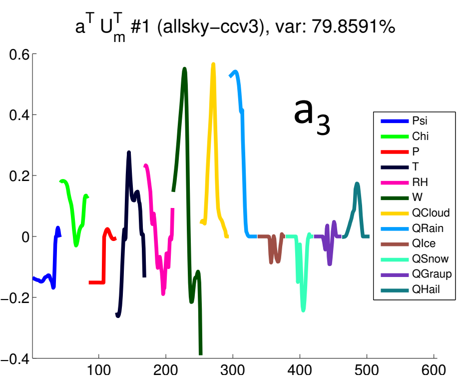
emission – scattering



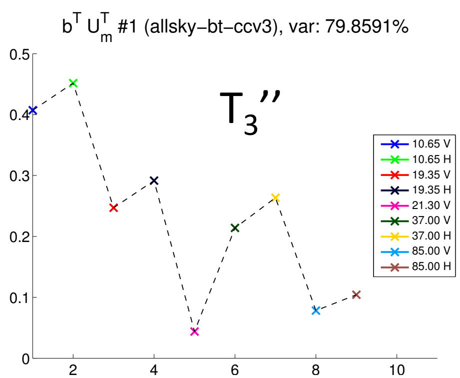
$x_2'' = a_2^t\, x'$



surface – vapor



$x_3'' = a_3^t\, x'$



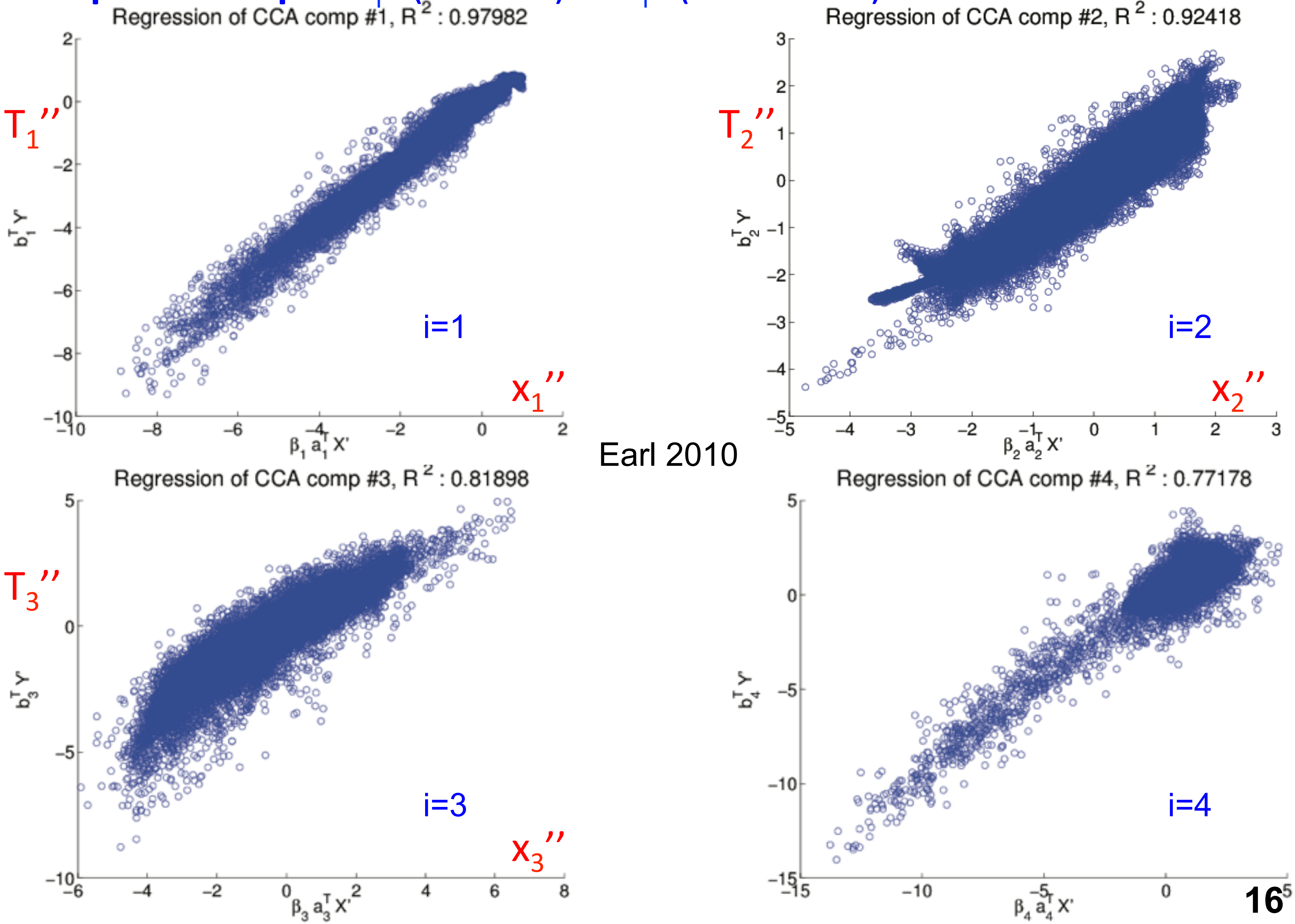
$T_3'' = b_3^t\, T'$

Most remarkable:
the operators H_1, H_2 , H_3
giving

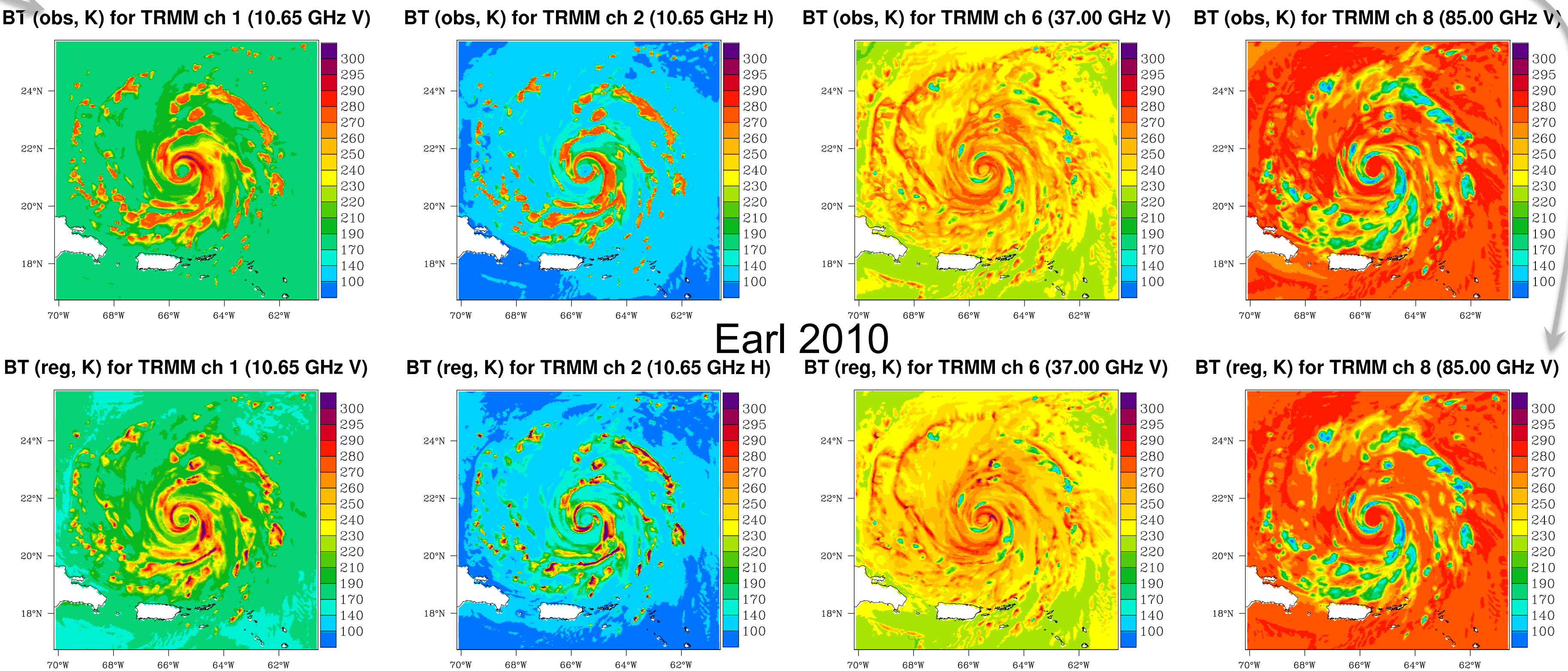
$$T_1'' = H_1(x_1'', x_2'', x_3'')$$
$$T_2'' = H_2(x_1'', x_2'', x_3'')$$
$$T_3'' = H_3(x_1'', x_2'', x_3'')$$

are not so nonlinear:

First part of step 3: T_i” (vertical) vs x_i” (horizontal)



First part of step 3: compare the actual T_b with approximates using x''



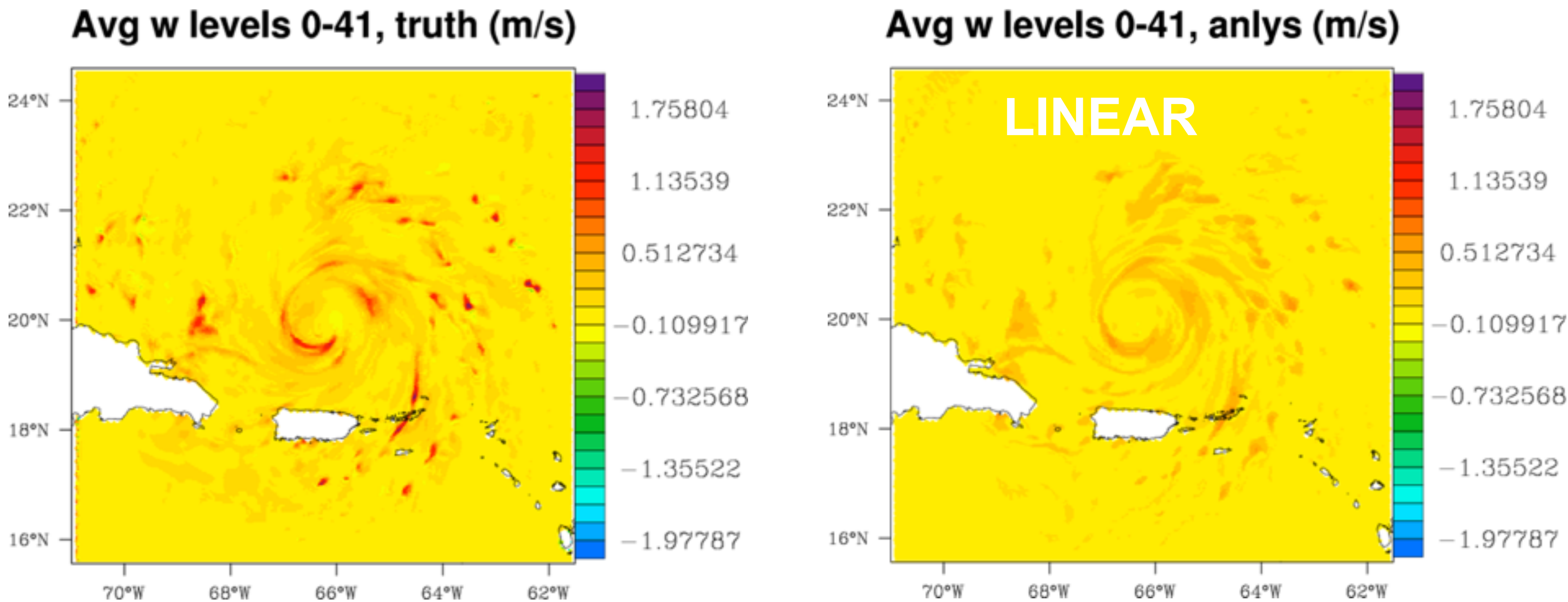
- Still not using the nonlinear representation of the observations (just the 3 combos of variables defined on slide 14 that maximize the linear correlation with corresponding combos of brightness temperatures)
- main problem: at the lowest and highest extremities of the ranges (high and low T_b)
- Let's test the performance of the non-linear differentiable expression for H

Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



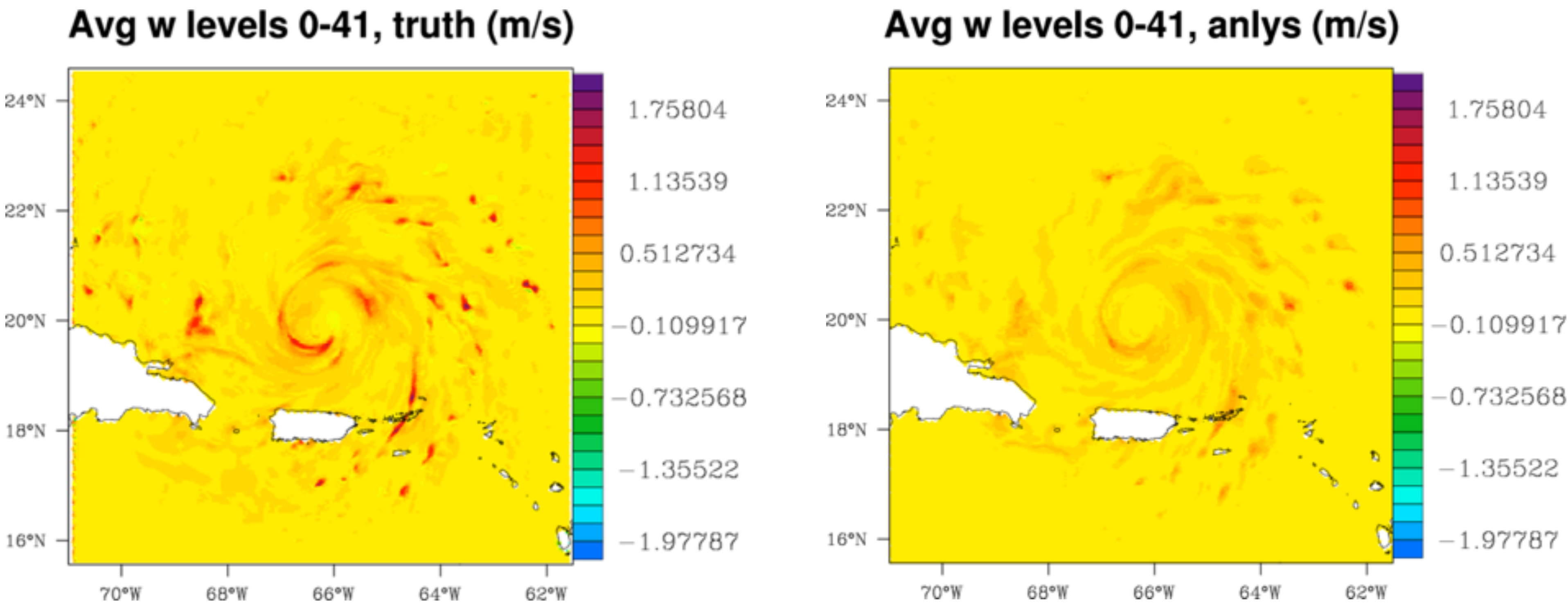
vertical component of wind

Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



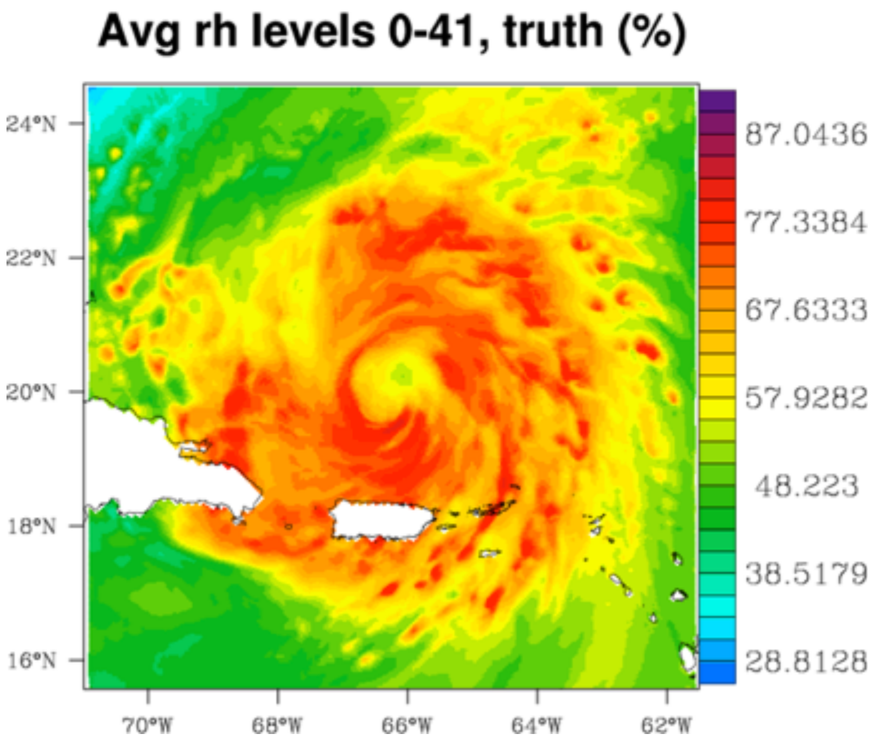
vertical component of wind

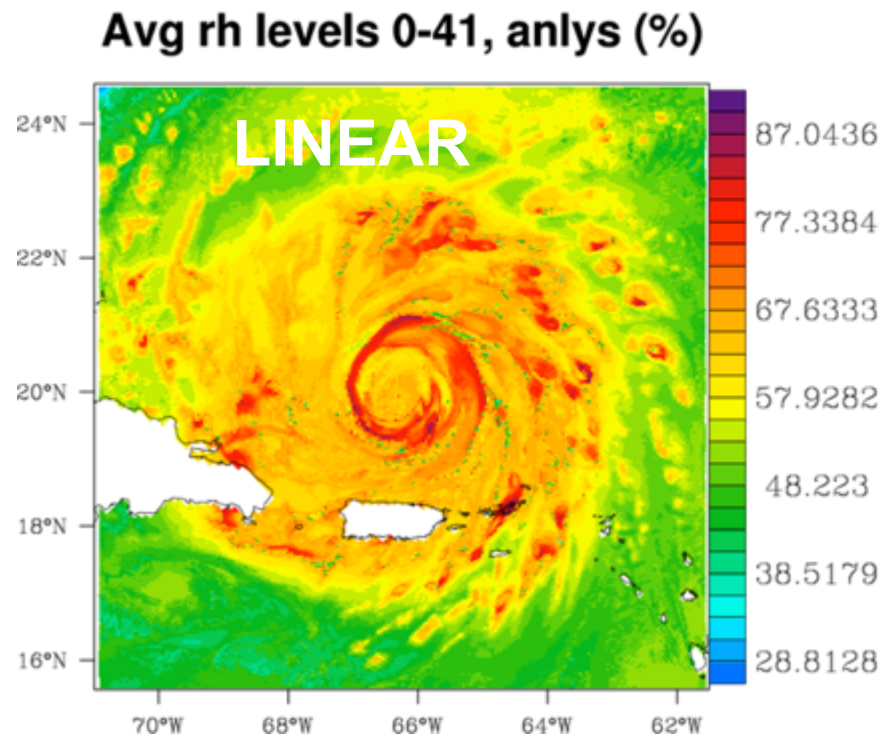
Second part of step 3: use nonlinear expression

$T_i'' \sim x_i''$

So let’s try an assimilation using this observation operator:
(start with 1D-var – not quite a “retrieval”, because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:





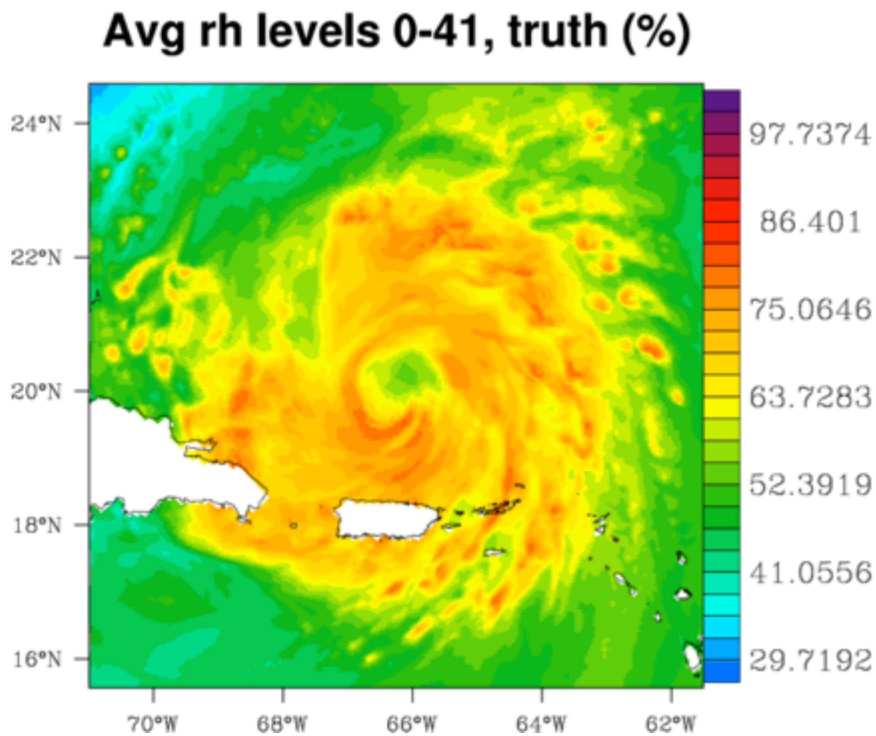
water vapor

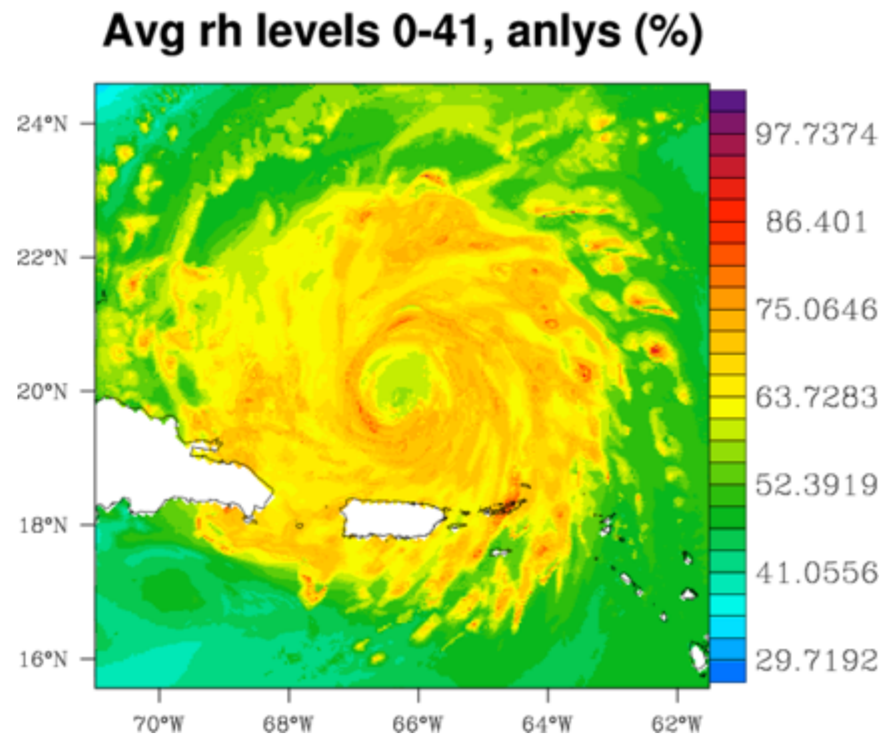
Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let’s try an assimilation using this observation operator:
(start with 1D-var – not quite a “retrieval”, because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:





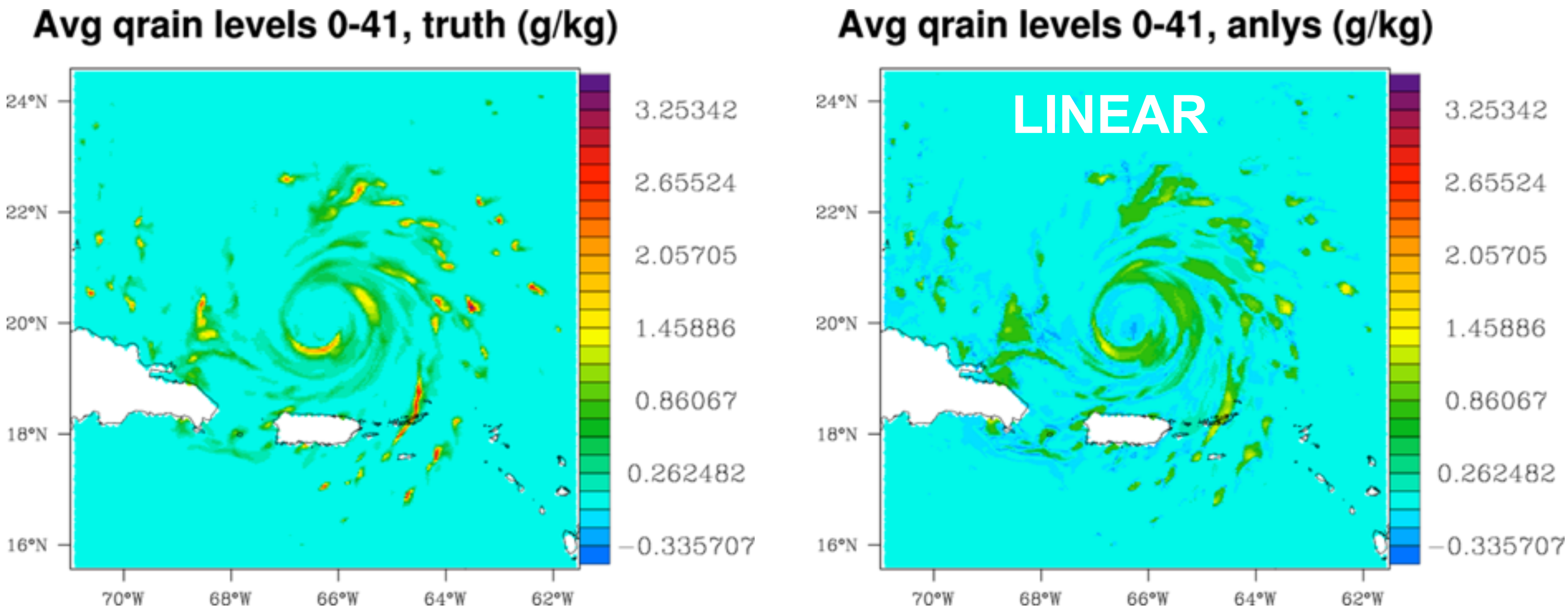
water vapor

Second part of step 3: use nonlinear expression

$T_i'' \sim x_i''$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



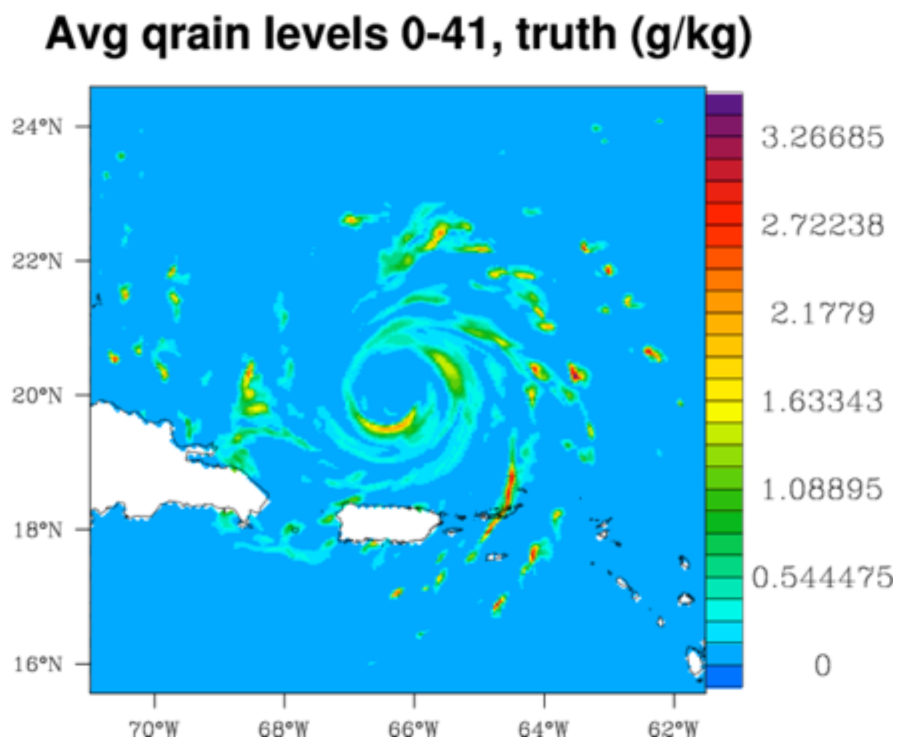
RAIN

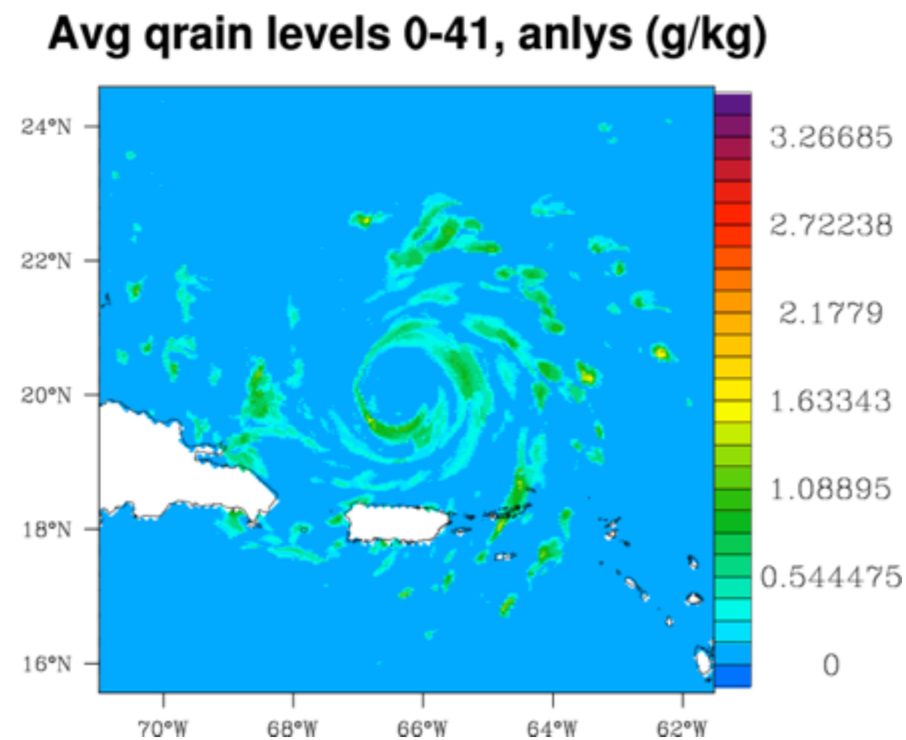
Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:





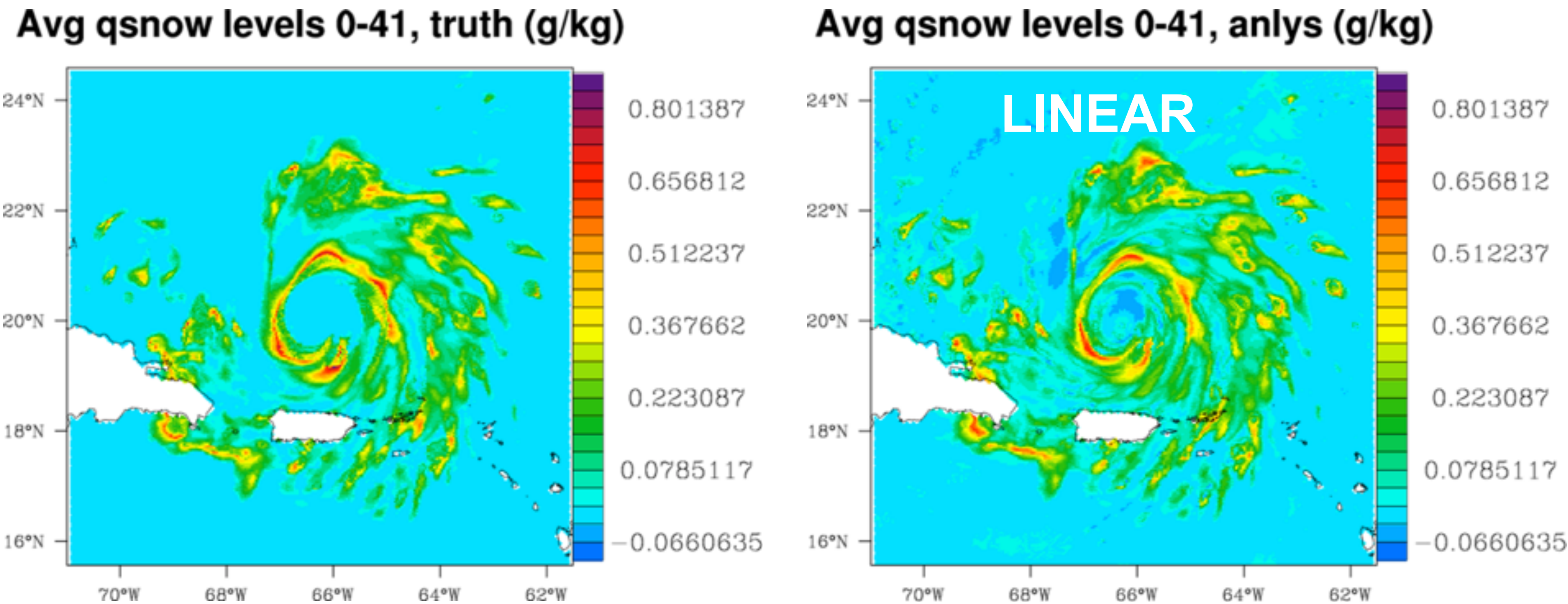
RAIN

Second part of step 3: use nonlinear expression

$T_i'' \sim x_i''$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



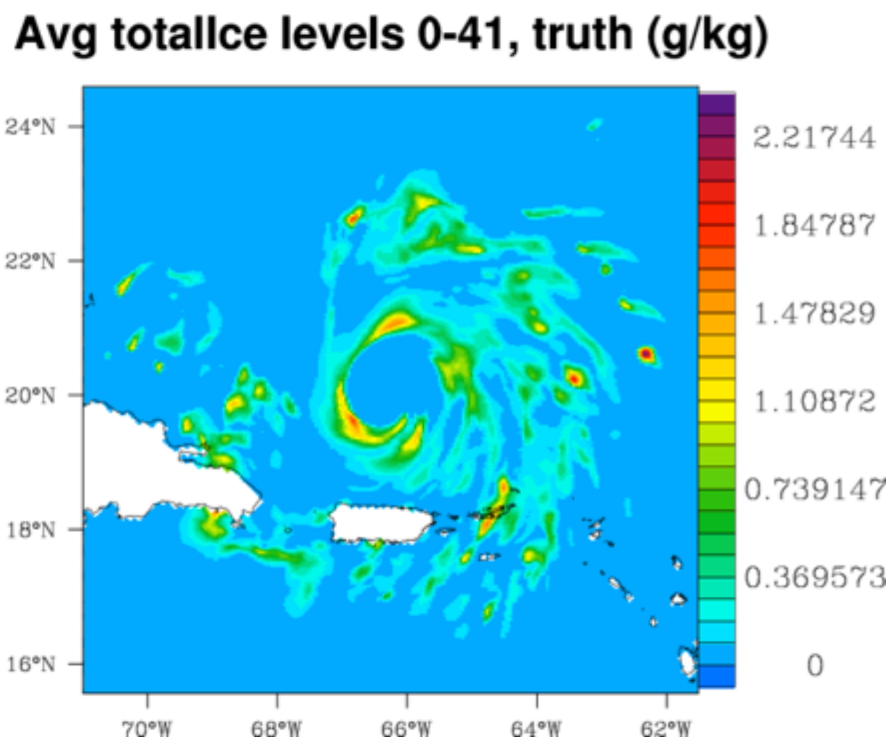
SNOW

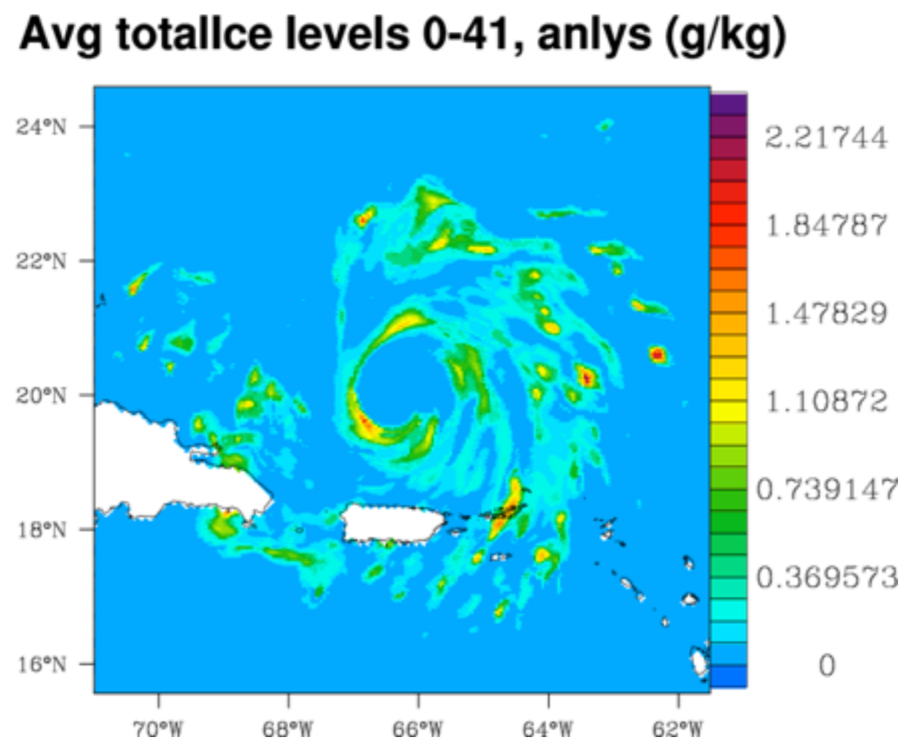
Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \ T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let’s try an assimilation using this observation operator:
(start with 1D-var – not quite a “retrieval”, because of the covariances)

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:





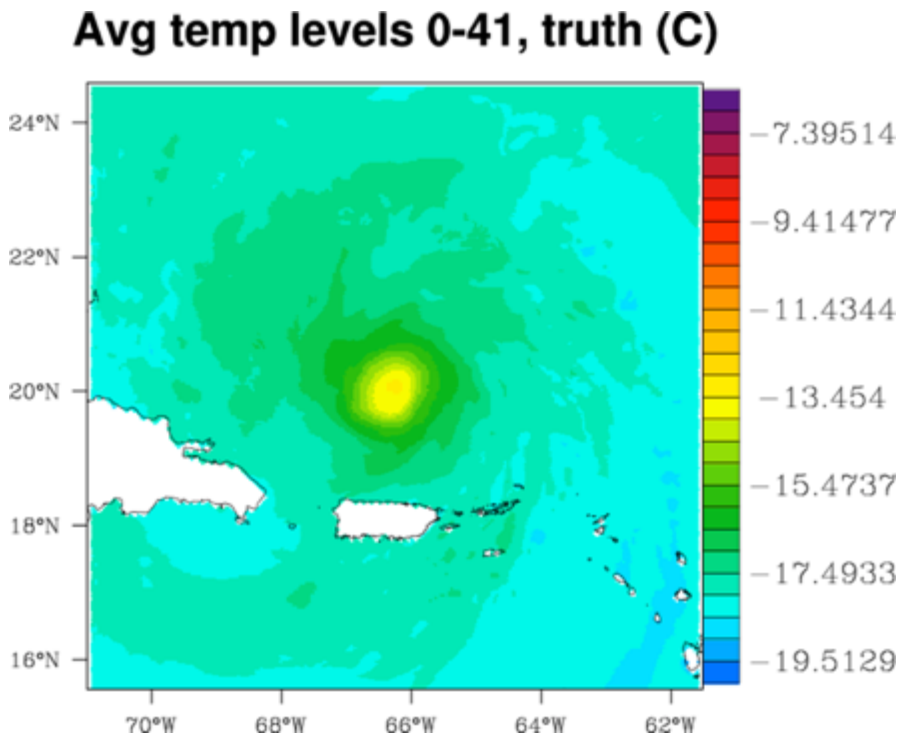
TOTAL ICE

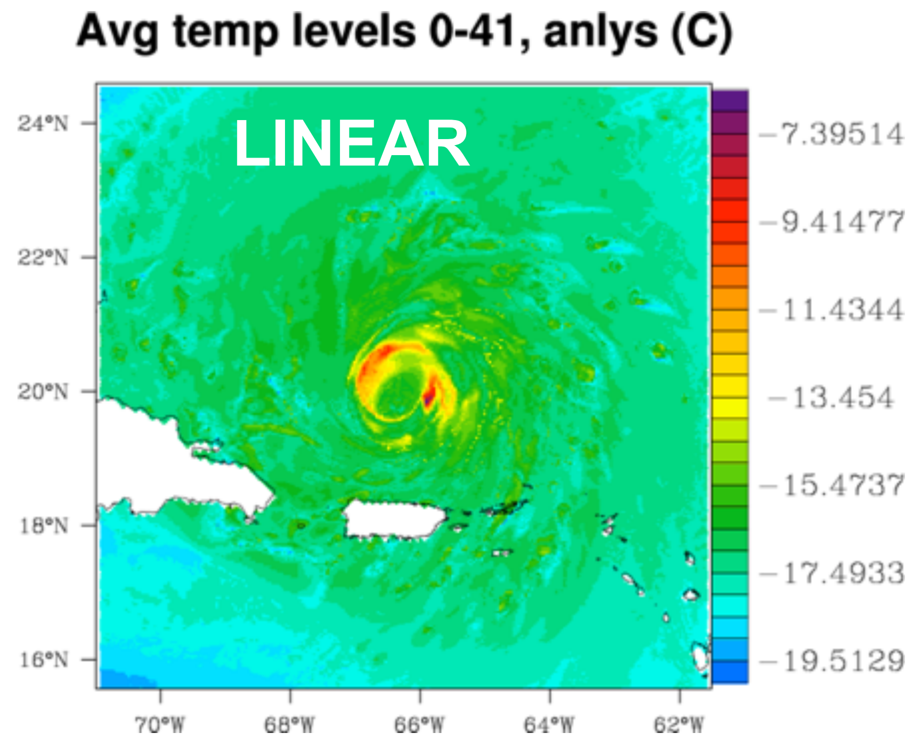
Second part of step 3: use nonlinear expression

$T_i'' \sim x_i''$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background, each variable having the global mean value at that height level:





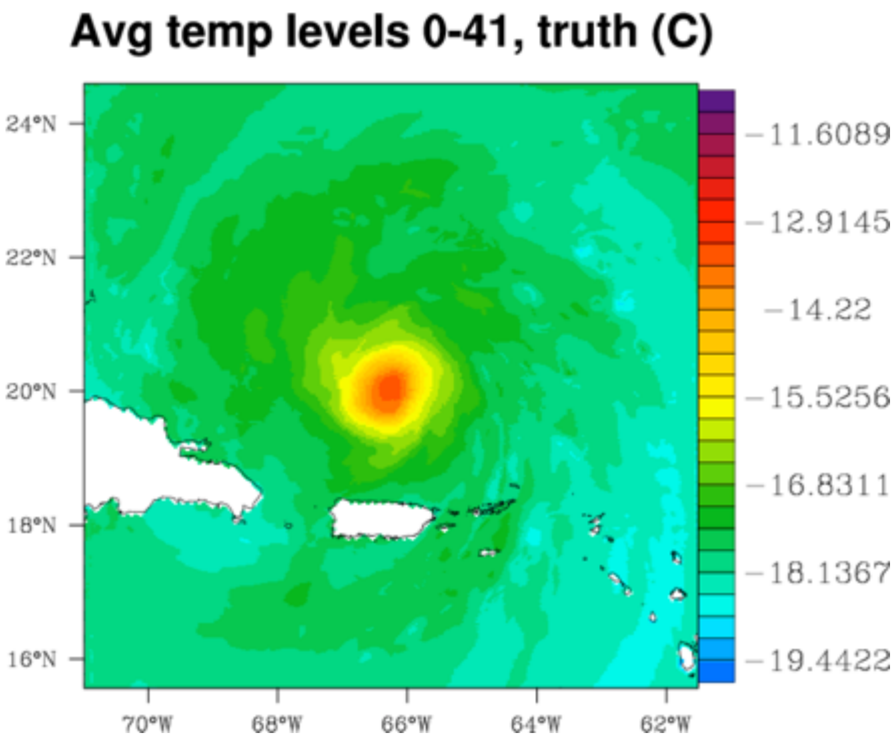
temperature

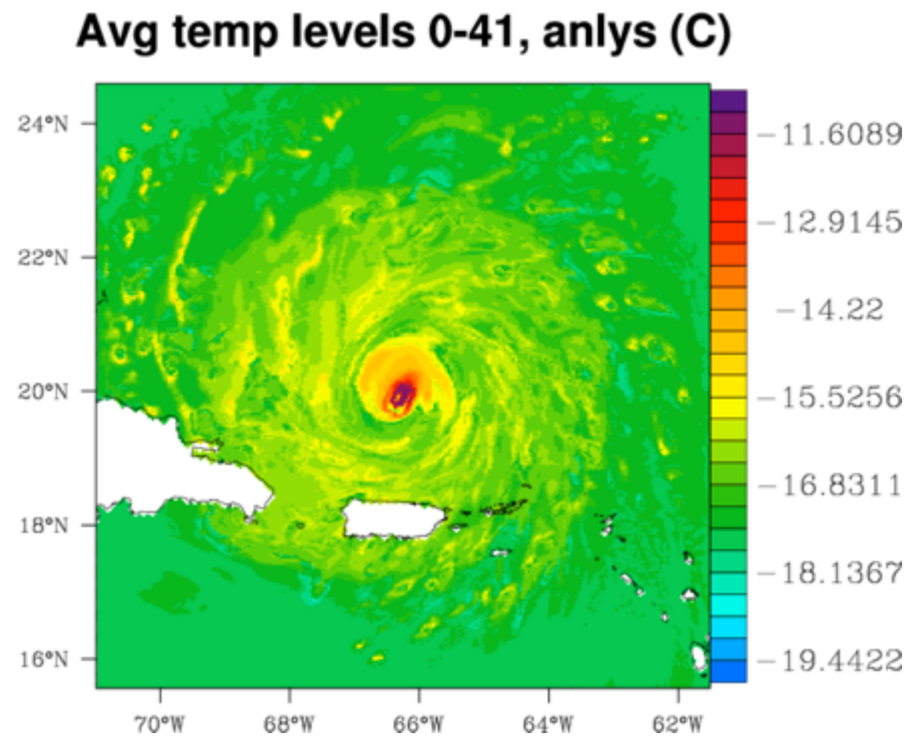
Second part of step 3: use nonlinear expression

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

So let's try an assimilation using this observation operator:
(start with 1D-var – not quite a "retrieval", because of the covariances)

Having started with a horizontally uniform background, each variable having the global mean value at that height level:



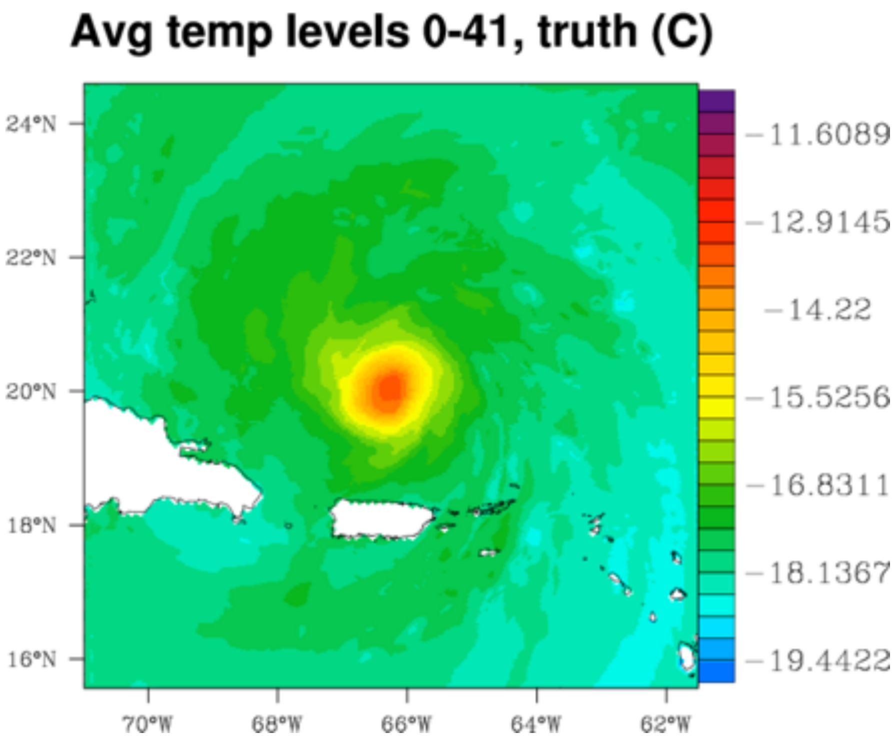


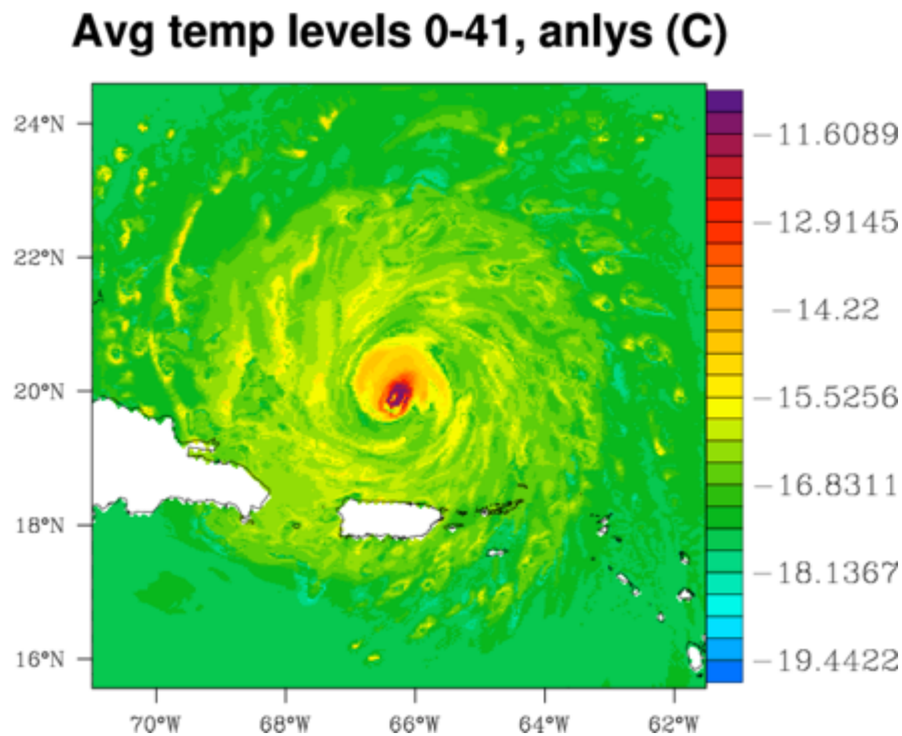
temperature

Look Ma!
a warm core!

Is this for real?
Can we really reconstruct most of the hurricane from window μwave??

Can we estimate vertical wind, and temperature anomaly, directly from the window-channel passive microwave (SSMIS, AMSR, TMI)???





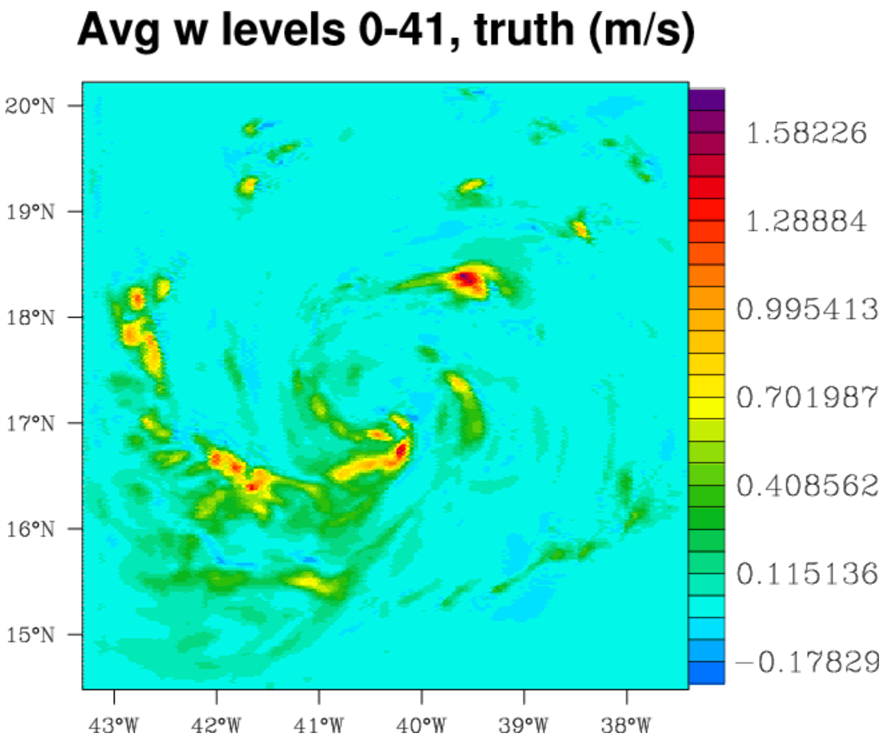
temperature

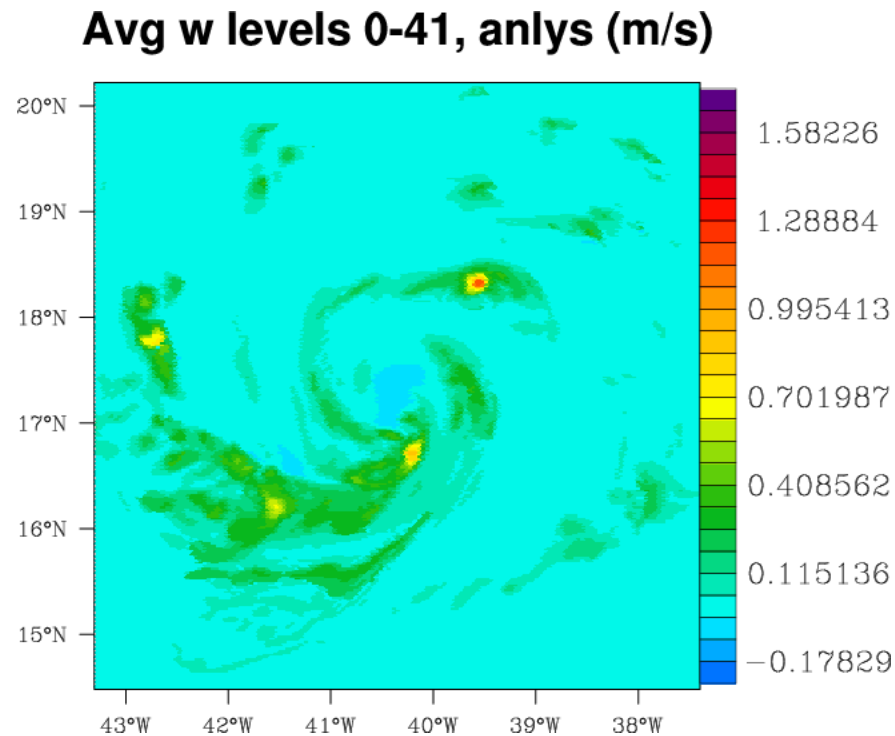
Look Ma!
a warm core!

Try the exact same operator, derived from Earl, on Igor:

assimilation using this observation operator:

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



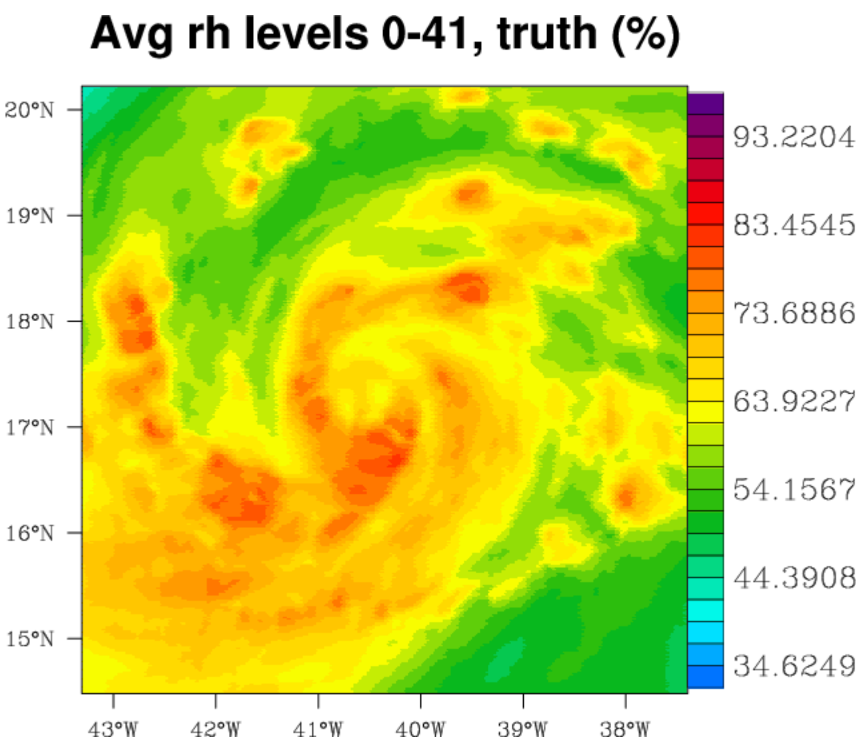


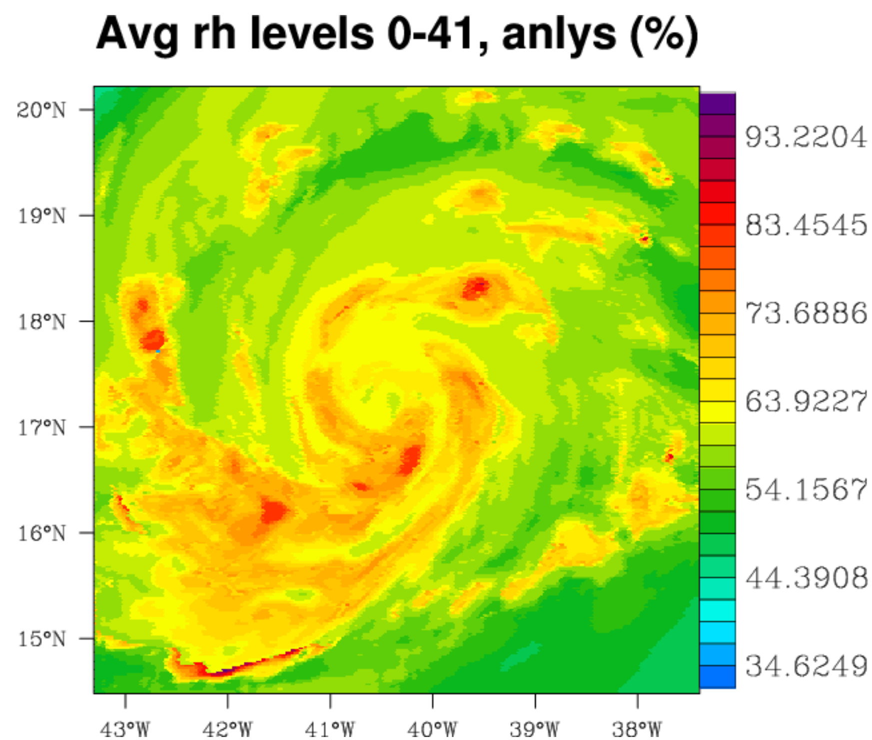
vertical component of wind

Try the exact same operator, derived from Earl, on Igor:

assimilation using this observation operator:

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



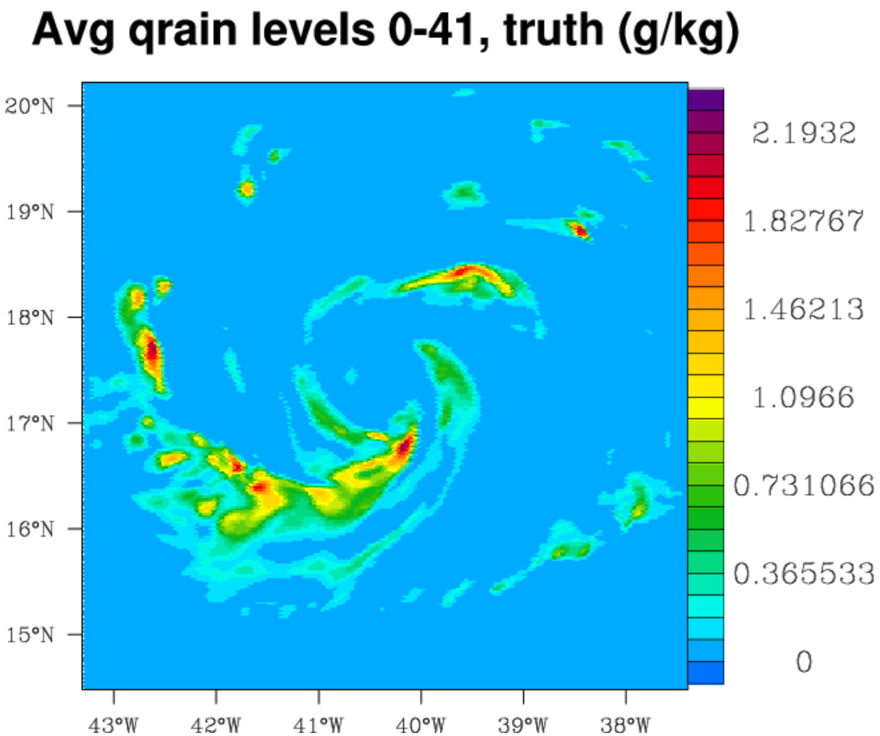


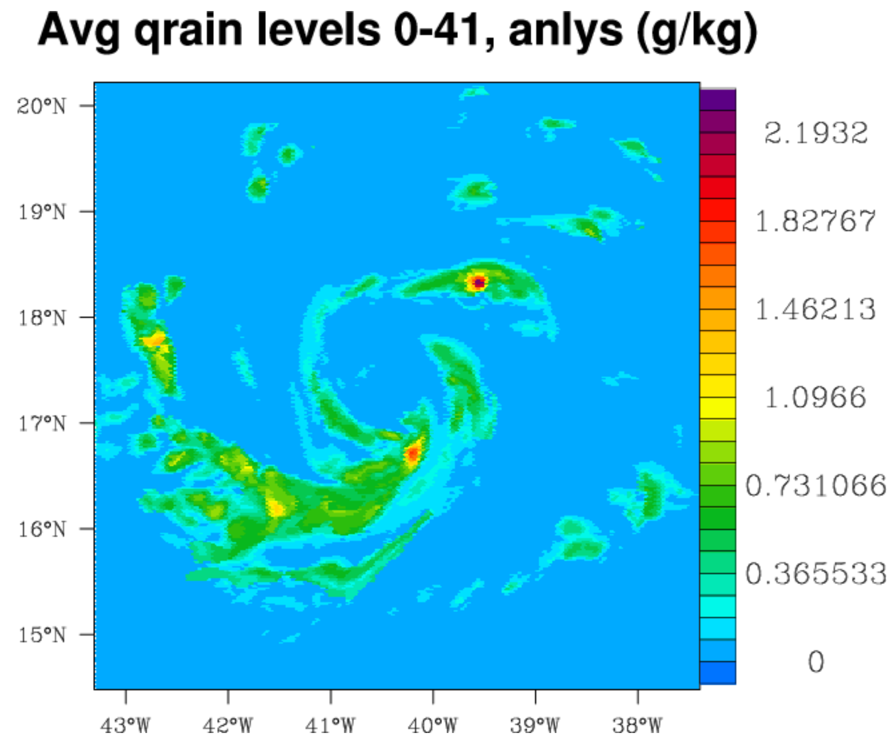
water vapor

Try the exact same operator, derived from Earl, on Igor:

assimilation using this observation operator:

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:



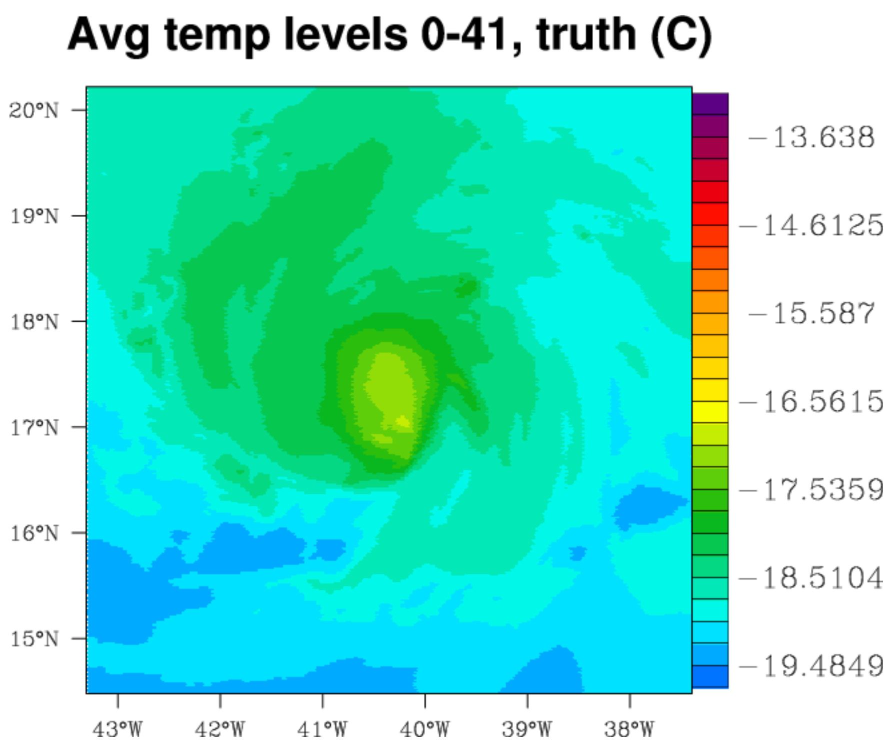


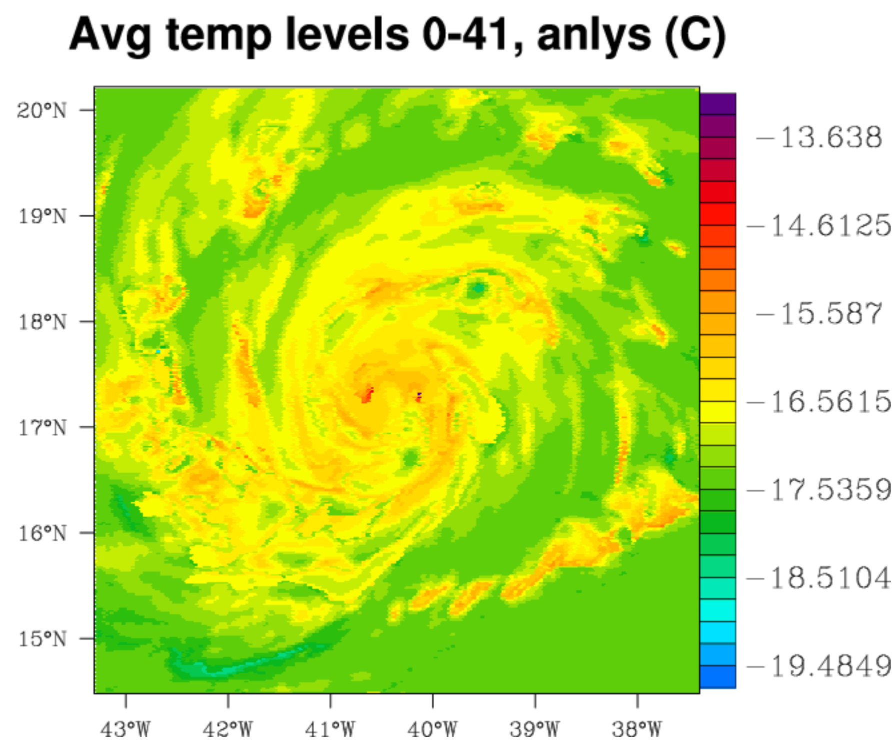
rain

Try the exact same operator, derived from Earl, on Igor:

assimilation using this observation operator:

Having started with a horizontally uniform background,
each variable having the global mean value at that height level:





temperature

still finds a warm core!
(but warm bias throughout)

So far, we have let the math define the transform variables:

$$T_1'' = H_1(x_1'', x_2'', x_3'')$$
$$T_2'' = H_2(x_1'', x_2'', x_3'')$$
$$T_3'' = H_3(x_1'', x_2'', x_3'')$$

Why not subjectively inject the physics, and impose different transform variables, dictated by expectation:

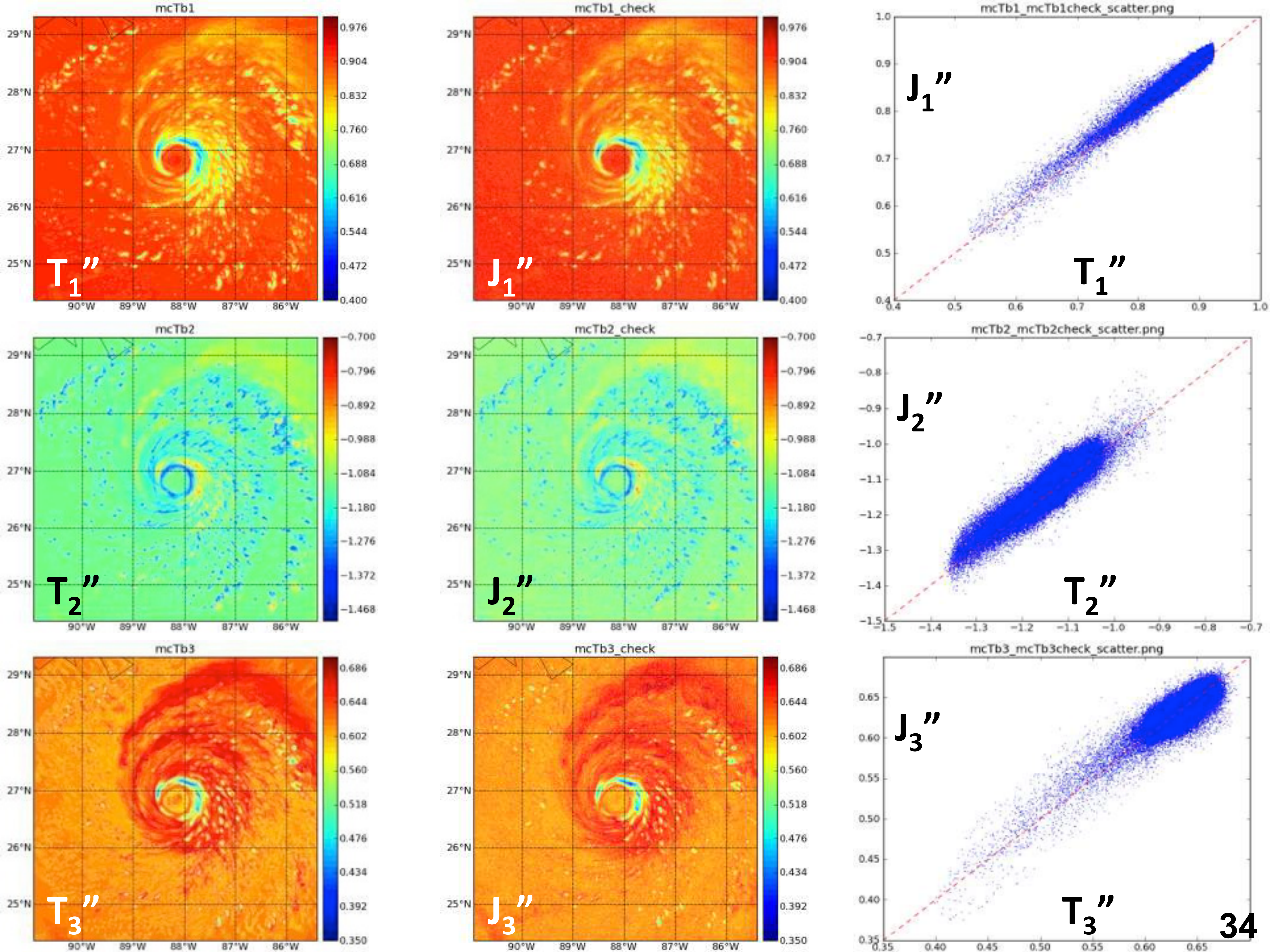
$$T_1'' = J_1(x_1'', x_{1wv}', x_{1rain}')$$
$$T_2'' = J_2(x_2'', x_{1wv}', x_{1rain}')$$
$$T_3'' = J_3(x_3'', x_{1wv}', x_{1rain}')$$

i.e. instead of

$$T_i''(x_1'', x_2'', x_3'') = \Sigma \; T_i''^{(n)} \exp(-[x_1''-x_1''^{(n)}]^2 -[x_2''-x_2''^{(n)}]^2 -[x_3''-x_3''^{(n)}]^2)$$

use

$$T_i''(x_1'', x_{1wv}', x_{1rain}'') = \Sigma \; T_i''^{(n)} \exp(-[x_i''-x_i''^{(n)}]^2 -[x_{1wv}'-x_{1wv}'^{(n)}]^2 -[x_{1rain}'-x_{1rain}'^{(n)}]^2)$$

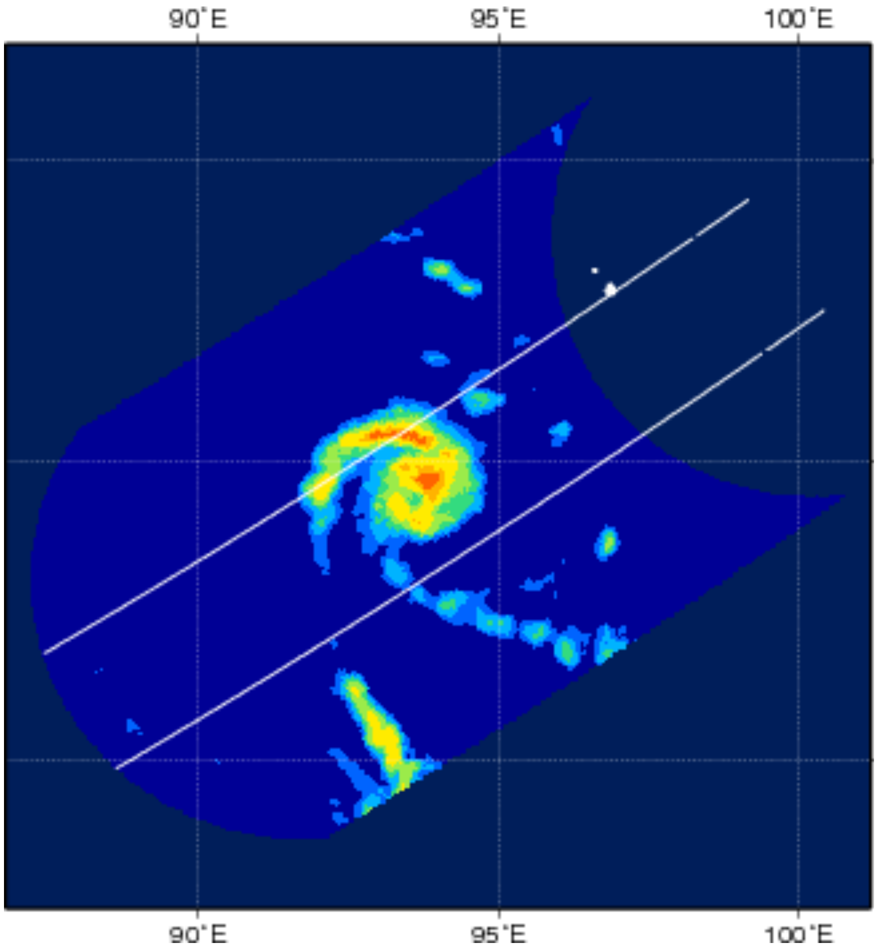

mcTb1
mcTb1_check
mcTb1_mcTb1check_scatter.png
mcTb2
mcTb2_check
mcTb2_mcTb2check_scatter.png
mcTb3
mcTb3_check
mcTb3_mcTb3check_scatter.png
T1”
J1”
J1”
T1”
T2”
J2”
J2”
T2”
T3”
J3”
J3”
T3”
29°N
28°N
27°N
26°N
25°N
90°W
89°W
88°W
87°W
86°W
0.976
0.904
0.832
0.760
0.688
0.616
0.544
0.472
0.400
−0.700
−0.796
−0.892
−0.988
−1.084
−1.180
−1.276
−1.372
−1.468
0.686
0.644
0.602
0.560
0.518
0.476
0.434
0.392
0.350

Jeff Steward is incorporating this operator into HWRF EnKF DAS

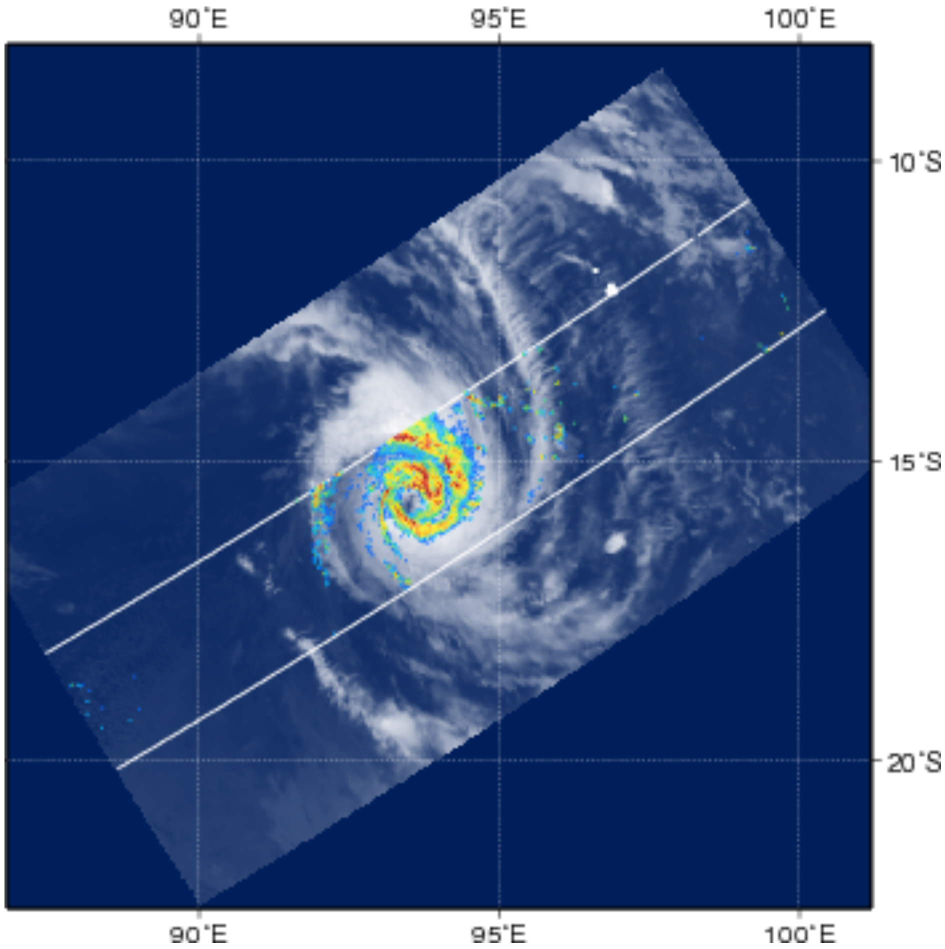
Horizontal Resolution:

passive microwave data have poor intrinsic resolution

Example: Tropical Cyclone Alenga, 7 December 2011



TMI



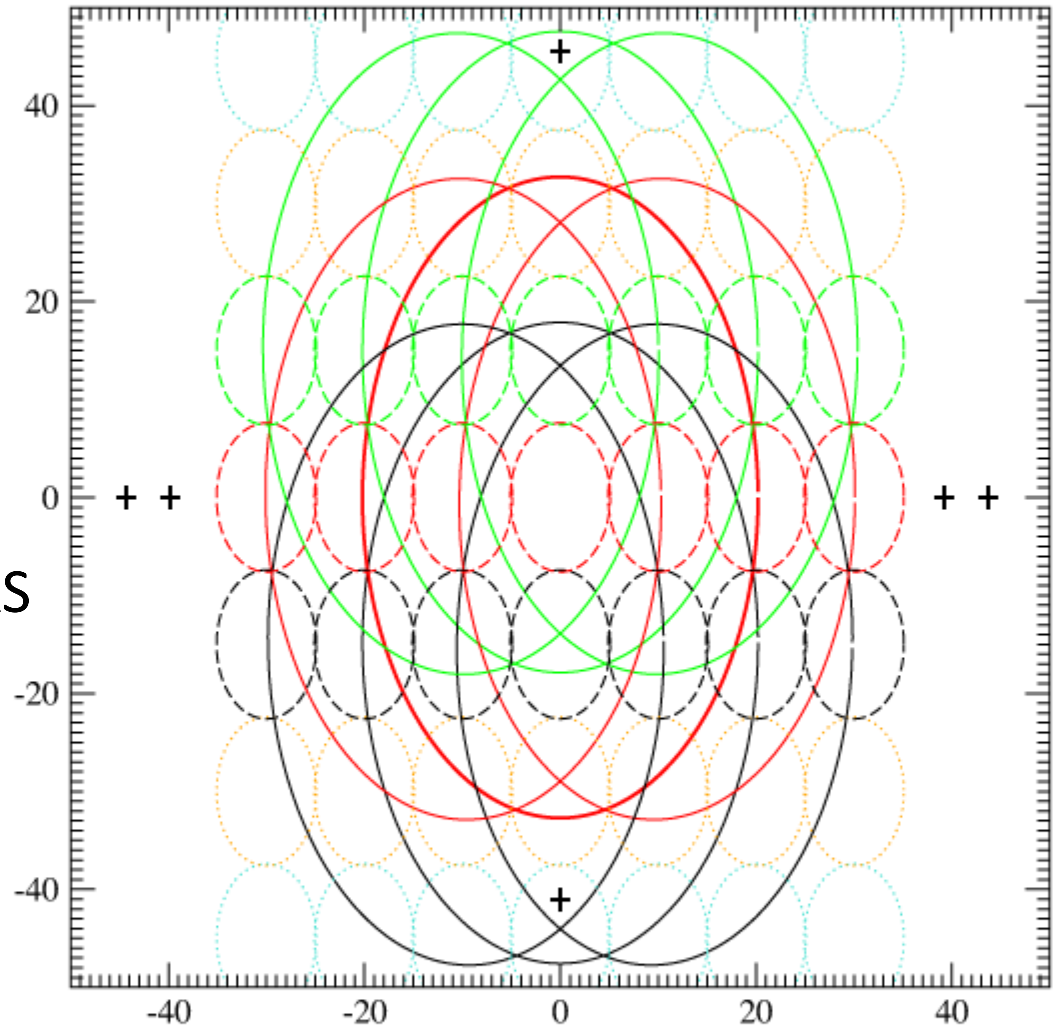
TRMM radar / IR

Horizontal Resolution:

All W-band channels (85-89 GHz) have < 10 km resolution

⇒ Use W-band measurements to sharpen resolution of lower-frequency channels

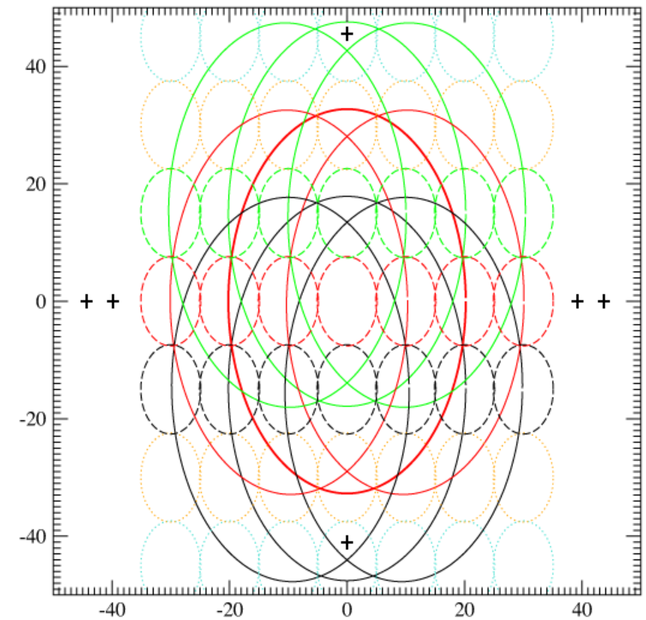
Example:
Megha-Tropiques's MADRAS scanning pattern
dashed = 89 & 157 GHz
solid = 18, 23 & 37 GHz



Horizontal Resolution:

All W-band channels (85-89 GHz) have < 10 km resolution

⇒ Use W-band measurements to sharpen resolution of lower-frequency channels



Try to solve for the high-resolution t_{hiR} given the coarse-resolution T_{loR} and the W-band w_{hiR}:

Assuming Gaussian probabilities, it is straightforward to show that

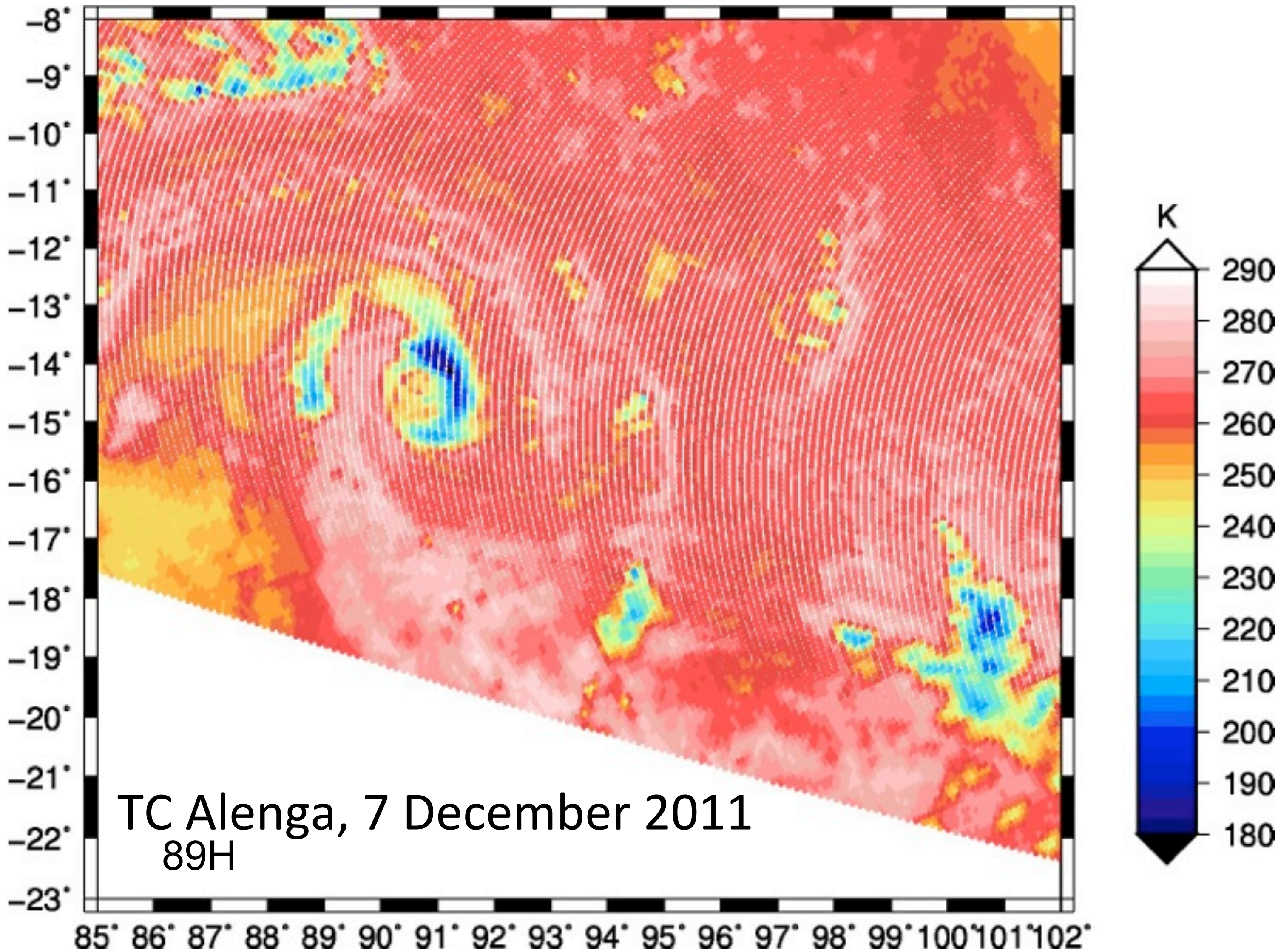
$$t_{hiR} = m(w_{hiR}) + (1 + C\,P^{*}\,E^{-1}\,P\,)^{-1}\,C\,P^{*}\,E^{-1}\,[T_{loR} - P\,m(w_{hiR})]$$

where m refers to the (average) relation between w and t with uncertainty matrix *C*, *P* is the antenna convolution matrix, and *E* is the error covariance you will allow in the convolution.

Horizontal Resolution:

Same example as before: Tropical Cyclone Alenga, 7 December 2011

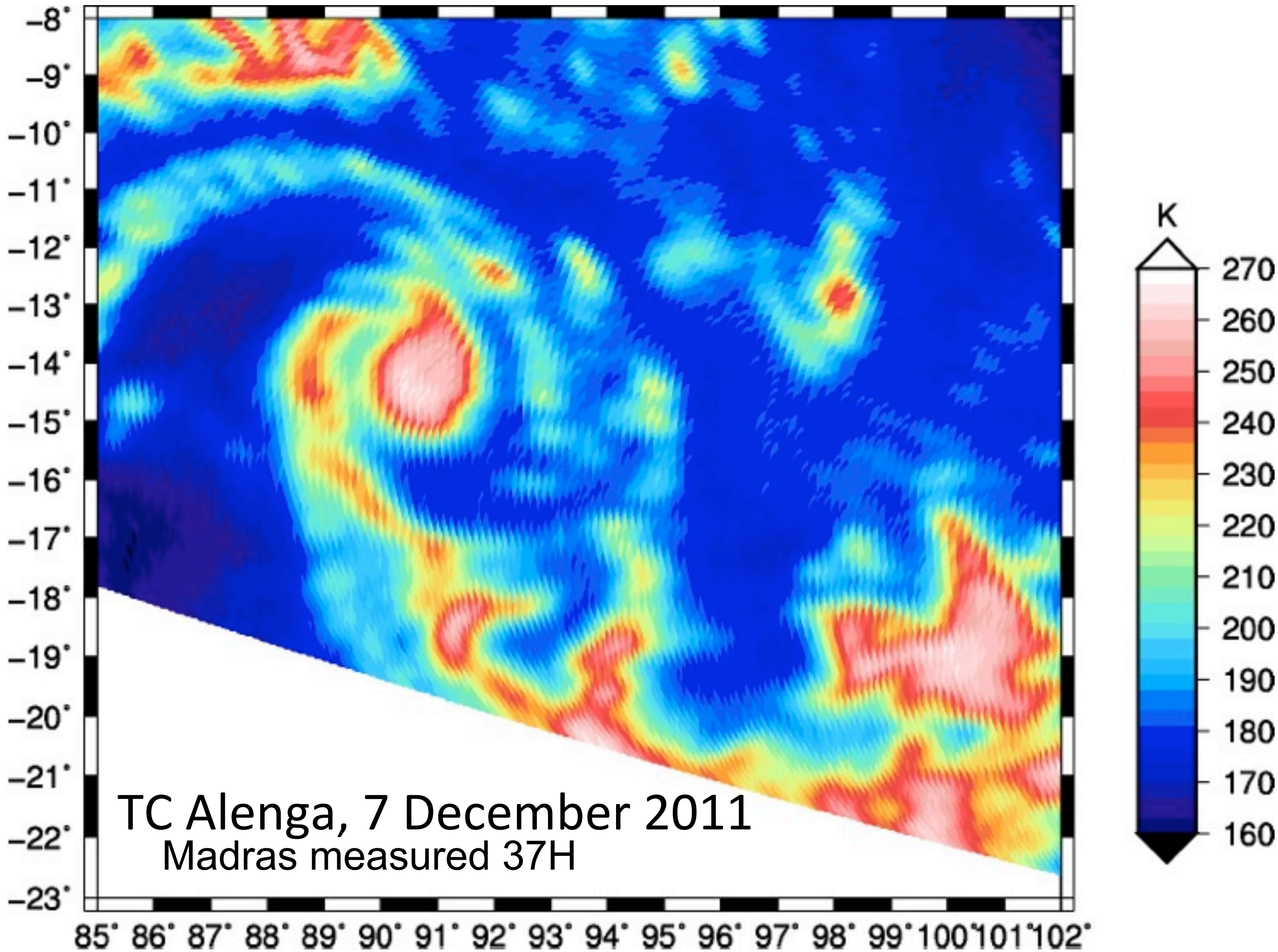
$$T_{hiRes} = t(89_{hiRes}) + (1 + C\,P^{*}\,E^{-1}\,P\,)^{-1}\,C\,P^{*}\,E^{-1}\,[\,T_{loRes} - P\,t(89_{hiRes})\,]$$



Horizontal Resolution:

Same example as before: Tropical Cyclone Alenga, 7 December 2011

$$T_{hiRes} = t(89_{hiRes}) + (1 + C\,P^*\,E^{-1}\,P\,)^{-1}\,C\,P^*\,E^{-1}\,[\,T_{loRes} - P\,t(89_{hiRes})\,]$$



Horizontal Resolution:

Same example as before: Tropical Cyclone Alenga, 7 December 2011

$$T_{hiRes} = t(89_{hiRes}) + (1 + C\,P^*\,E^{-1}\,P\,)^{-1}\,C\,P^*\,E^{-1}\,[\,T_{loRes} - P\,t(89_{hiRes})\,]$$