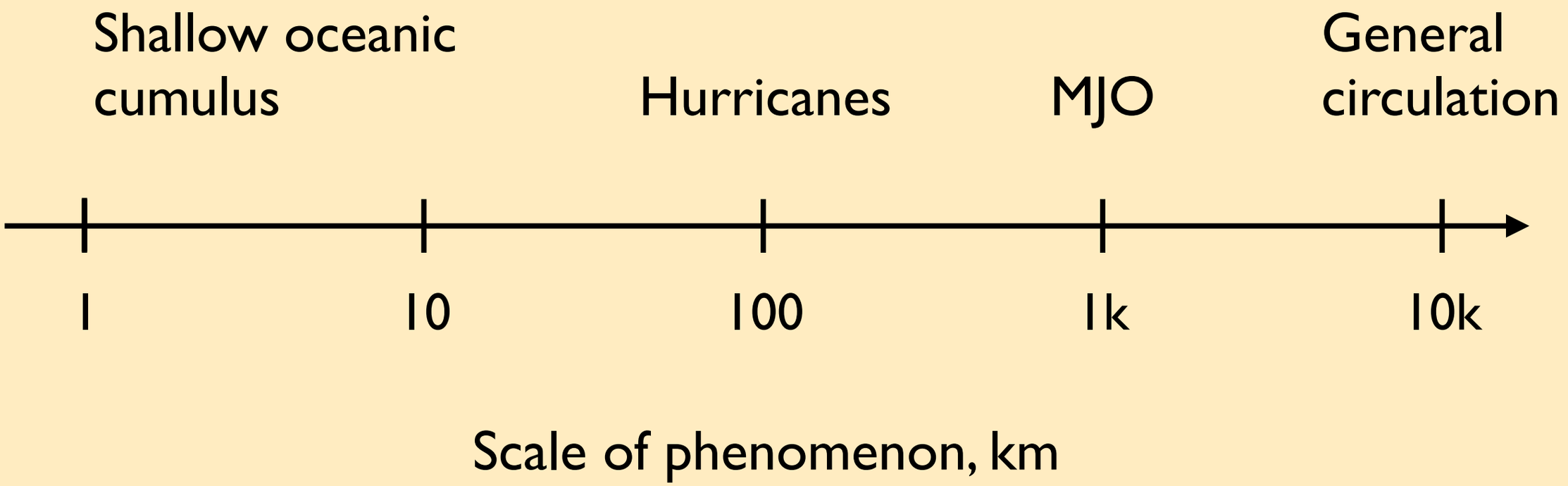
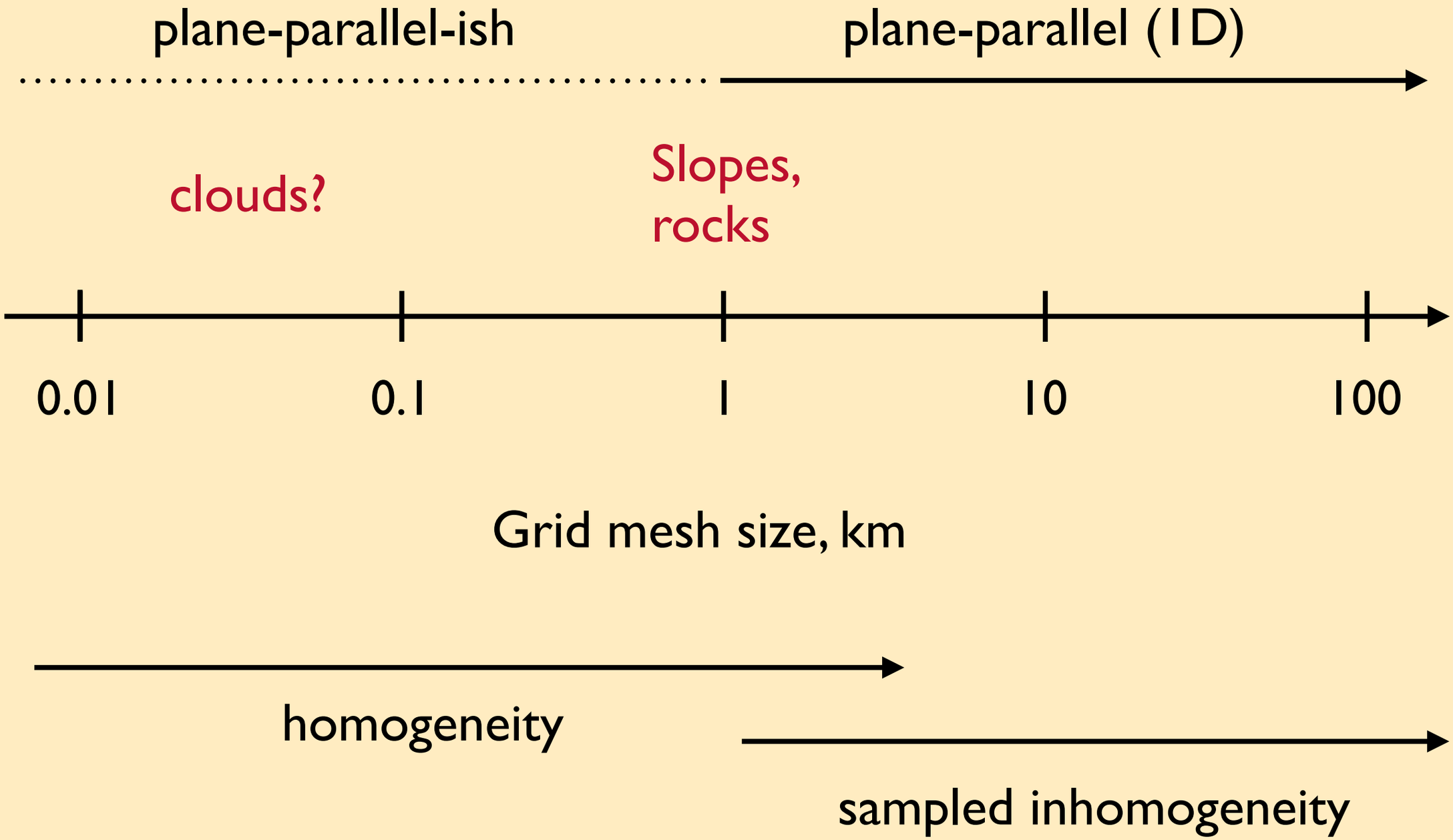
A new radiation model for atmospheric models across scales

Robert Pincus
University of Colorado
working with Eli Mlawer, Jennifer Delamere, and others at AER,
Brian Eaton and Jim Edwards at NCAR

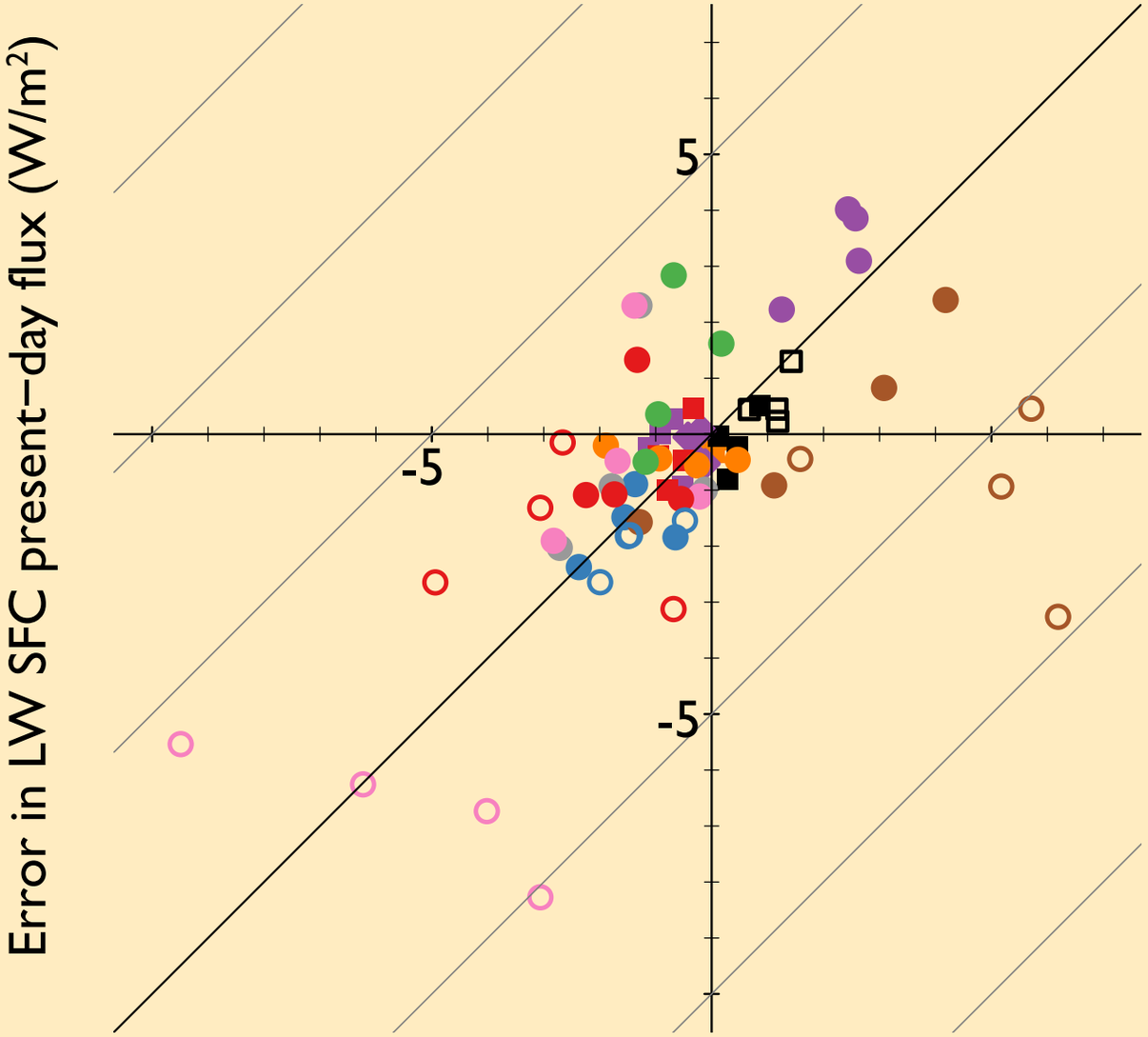
Radiation is central to atmospheric phenomena across scales including



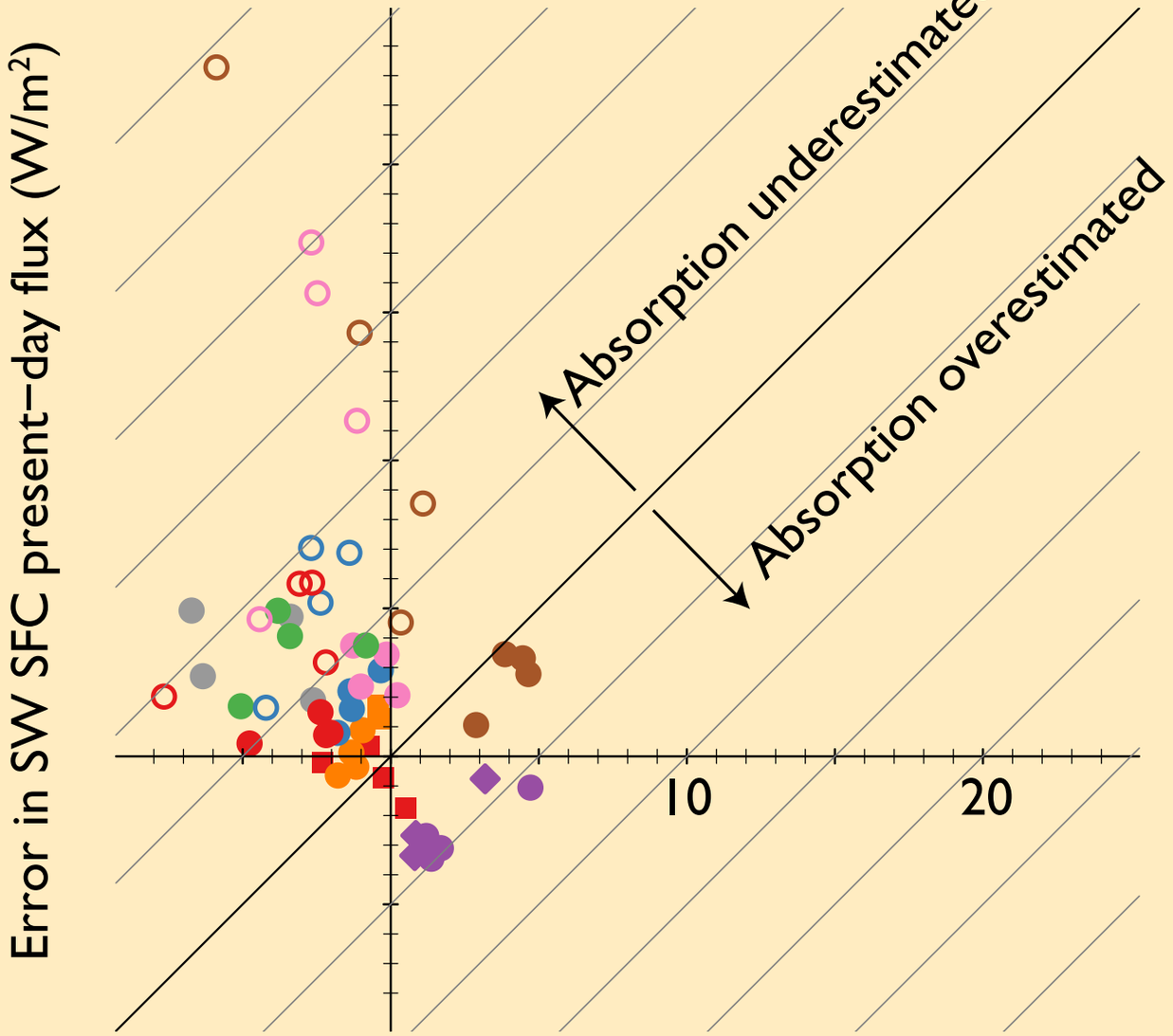
The same underlying (plane-parallel, homogenous) radiation model can be used for dynamical predictions across scales



Not all radiation codes are equally accurate



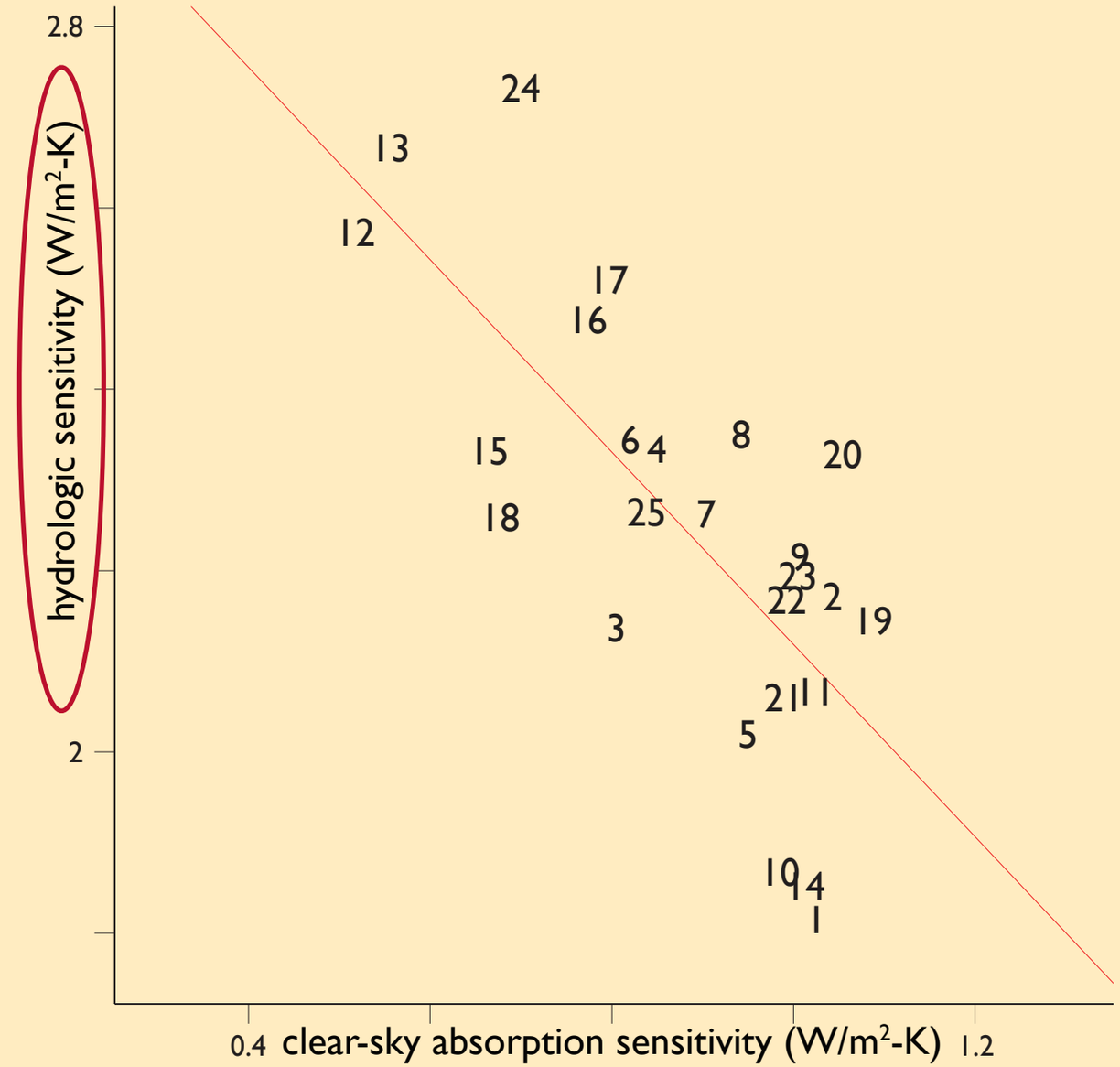
Error in LW TOA present−day flux (W/m²)

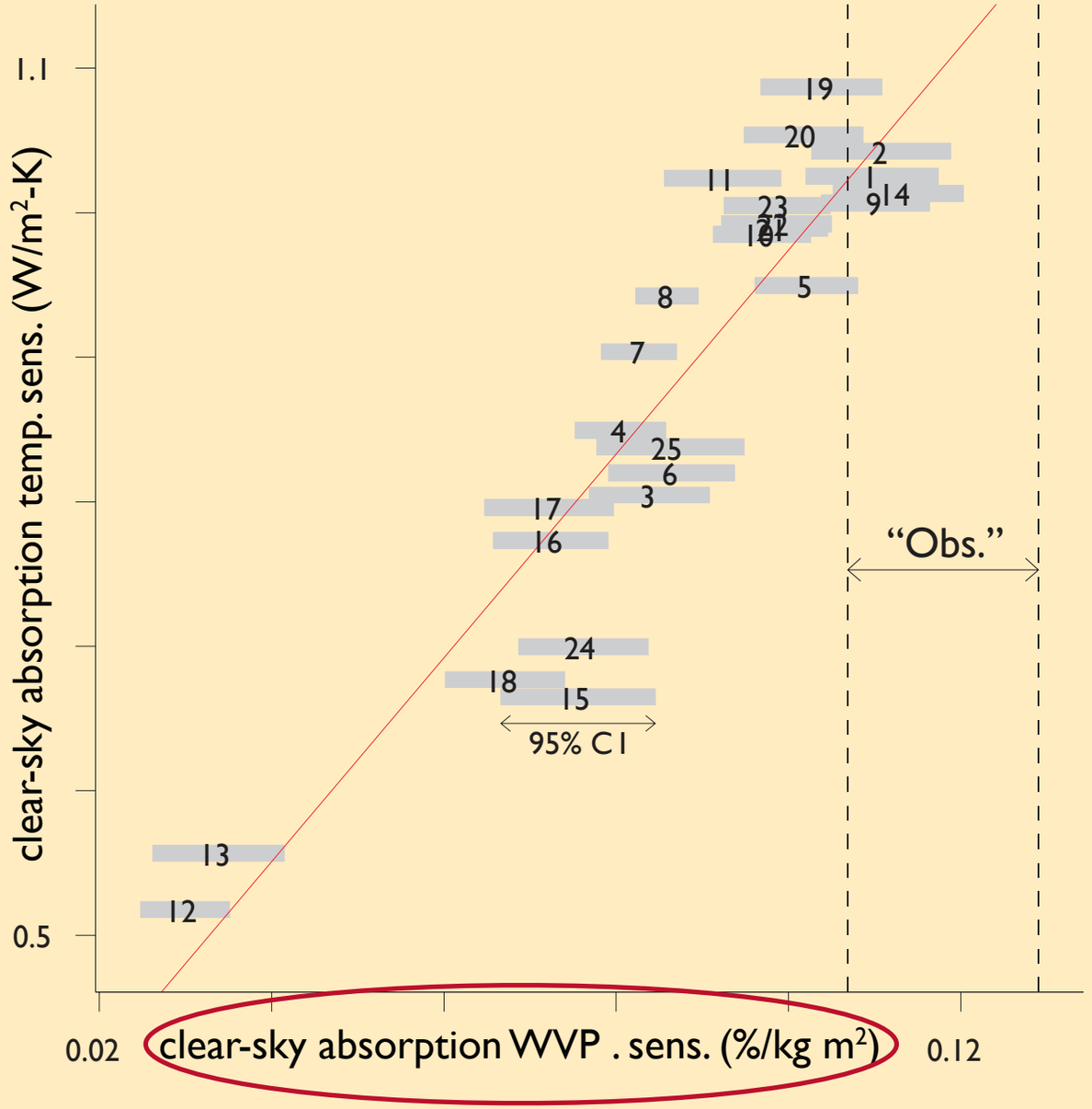


Error in SW TOA present−day flux (W/m²)

Pincus et al. 2015, 10.1002/2015GL064291

Inaccuracy has important impacts





after DeAngelis et al., 2015, 10.1038/nature15770

A radiation code for the next decade

With funding from ONR ESPC AOLI we (CU, AER) have been developing RRTMGP, a radiation code for the next decade. The emphasis is on

efficiency on modern and near-future computing platforms

flexibility for use in different contexts, e.g. across scales

These might seem like conflicting requirements but they can be treated at different levels in the code

What a radiation parameterization needs to do

Determine the spectrally-dependent optical state of the atmosphere from its physical state

Gases
Clouds, including scale-dependent McICA sampling
Aerosols (not currently scale-dependent but could be)

Determine layer properties

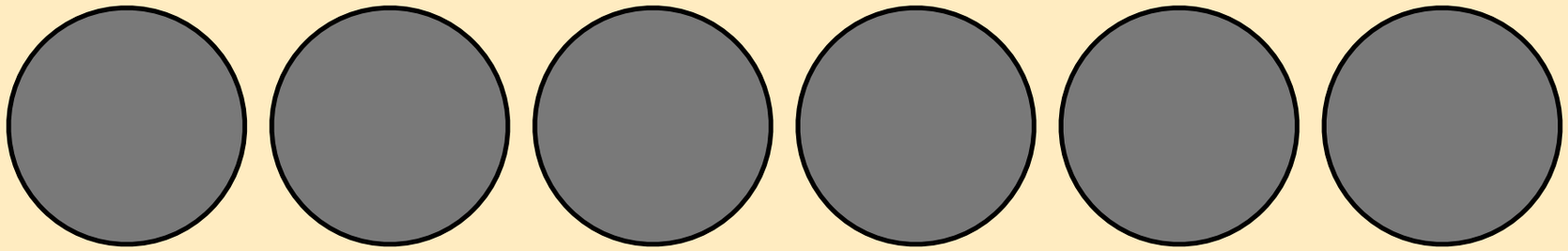
Compute transport

Spectral integration/reduction

A three layer cake

Interface: initialization, error handling, user choices, mapping physical to optical description

Brokering, flow control: e.g. level of detail, minimizing required calculations



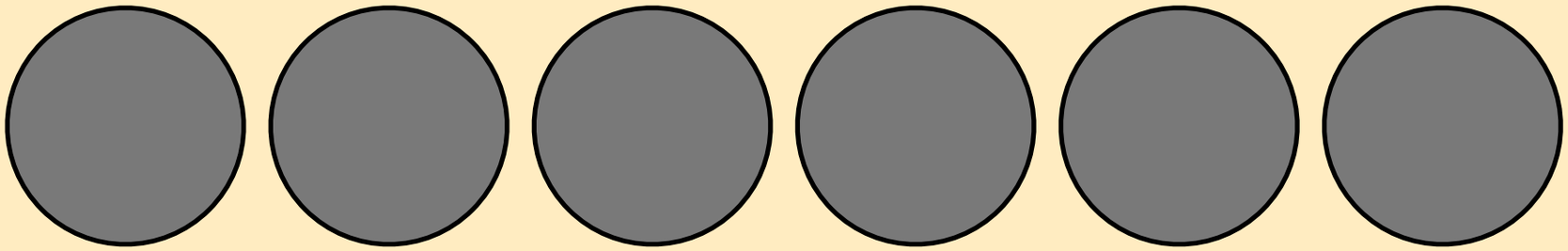
Computational kernels: simple, highly optimized code working on sanitized inputs

A three layer cake with icing

Interface: initialization, error handling, user choices, mapping physical to optical description

Data supplied at runtime

Brokering, flow control: e.g. level of detail, minimizing required calculations



Computational kernels: simple, highly optimized code working on sanitized inputs

What might take some getting used to

Much of the flexibility and some of the efficiency comes through the use of Fortran 2003 classes

Similar to derived types; can package data and procedures

RRTMGP requires some opaque classes for input, much like using a library (gas concentrations, *k*-distribution)

Outputs are an *extensible* class; if you want something beyond our simple choices you write code *on top of ours*

We are mindful that not every modeling center will love this

What really might take some getting used to

There are two ways to provide cloud and aerosol information

Directly, providing spectrally-resolved, McICA-sampled optical properties in a pre-defined class

Indirectly, by *extending* an *abstract* class that packages data you define with (at least) one procedure whose interface we've defined. This minimizes data transfer and localizes memory use.

“Where can I get some?”

RRTMGP comprises both the code and updates to the underlying science (spectroscopy, i.e. dependence of gas absorption on conditions)

The code is mostly done, modulo testing that will no doubt expose bugs, so you could look at implementation now

But RRTMGP won’t be useful until new data is available, likely by the end of 2016