Balancing accuracy, efficiency, and flexibility in a radiative transfer parameterization for dynamical models

Robert Pincus, University of Colorado
Eli Mlawer, AER

Radiation parameterization has a unique set of challenges

The radiation problem is extremely well-understood at a fundamental level.

Radiation parameterization is (mostly) convergent, making a series of well-defined approximations to a benchmark approach. The impact of each can be assessed (e.g. Barker et al. 2015, 10.1175/JAS-D-15-0033.1)

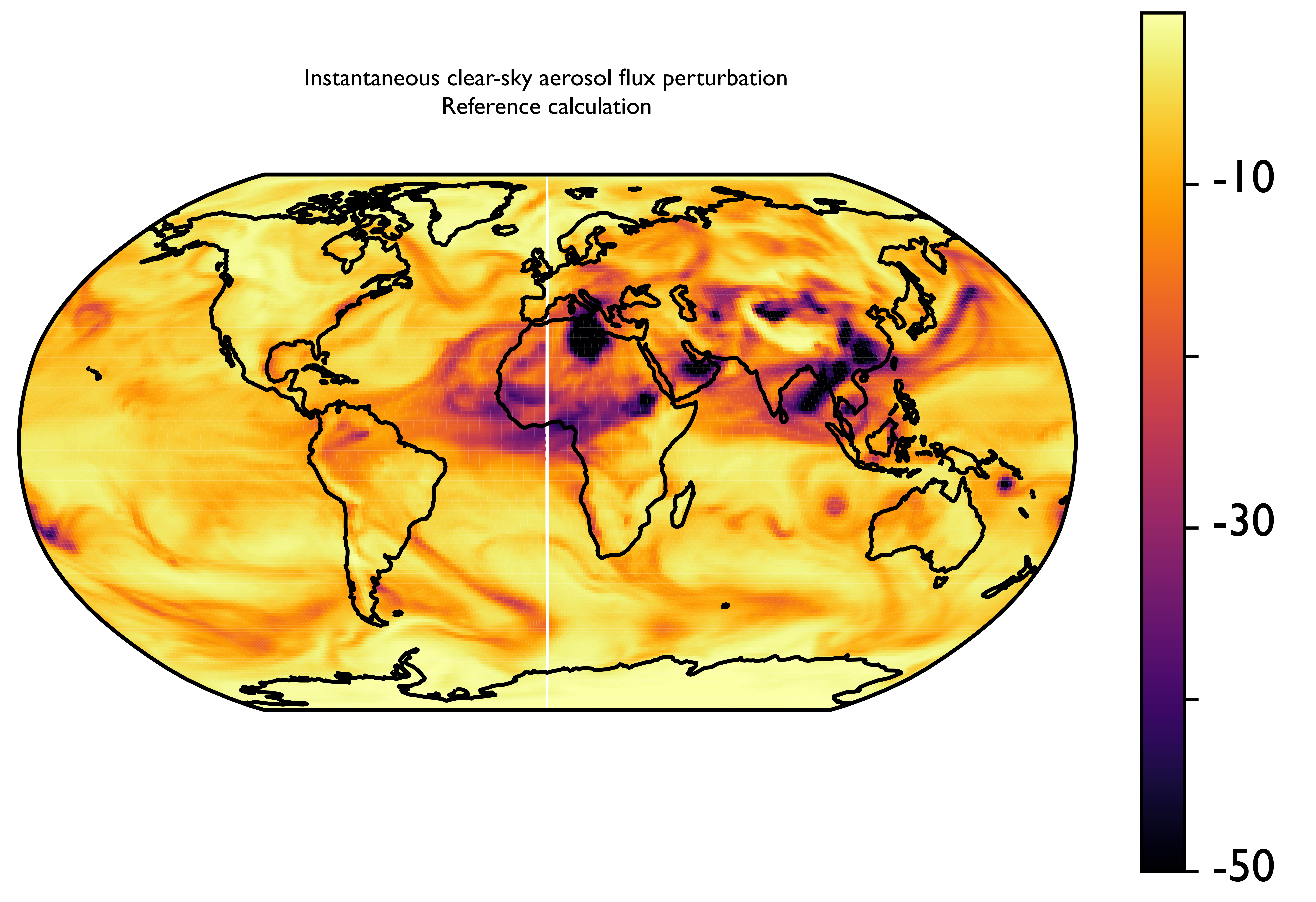
Radiation in any domain requires its own coupling:
between the physical state of the earth system and its optical description

Errors in radiative fluxes arise from some combination of

errors in state
incorrect coupling of physical and radiative states
approximation errors

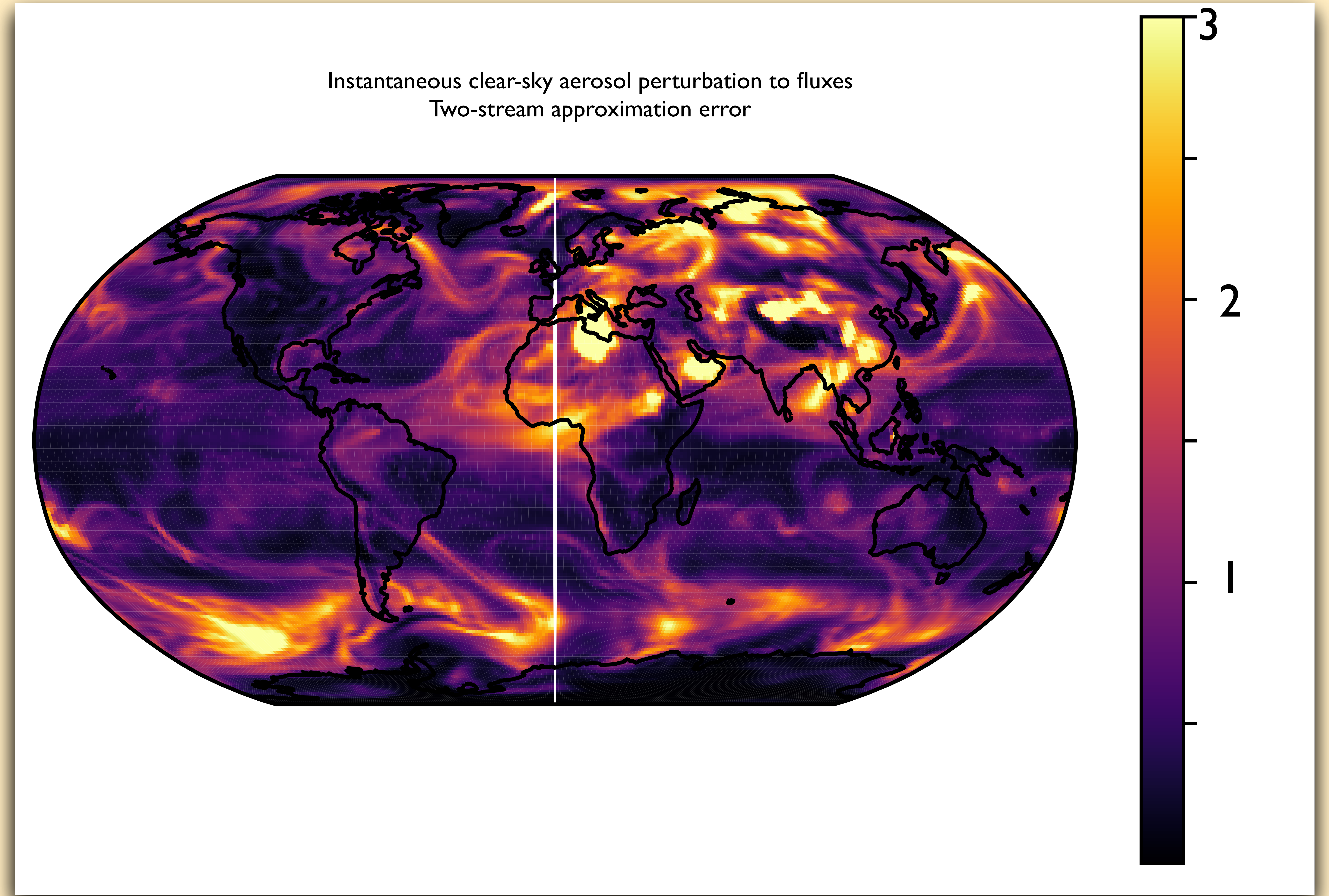
In climate models state error dominates; this is less clear for NWP

Approximation error is well-characterized…



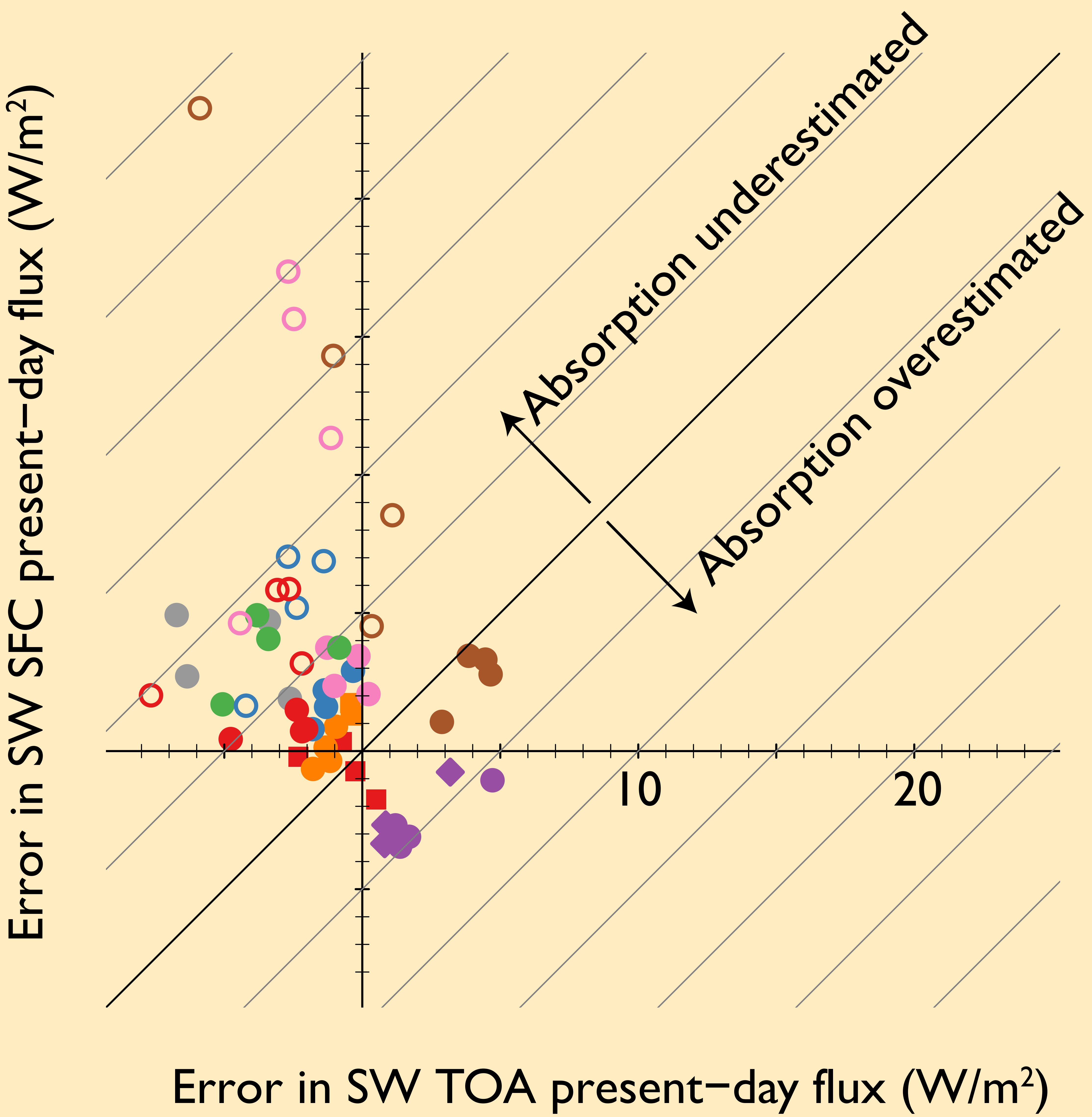
Freidenreich,
Jones, Paynter:
RFMIP Aerosol IRF

Approximation error is well-characterized…

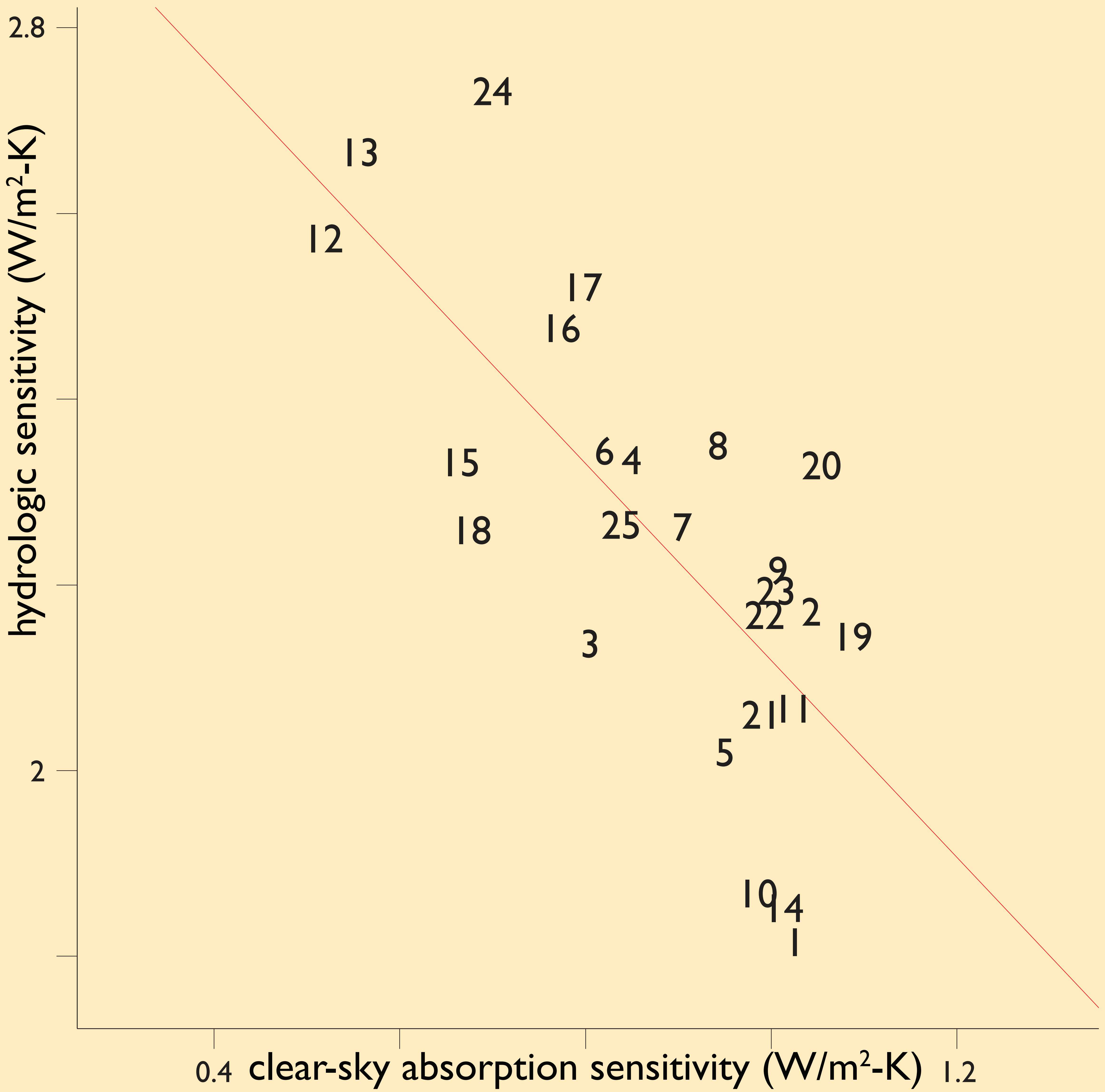


Freidenreich,
Jones, Paynter:
RFMIP Aerosol IRF

… but emergent behavior can be discouraging



Pincus et al., doi:10.1002/2015GL064291



after DeAngelis et al., doi:10.1038/nature15770

Why such a mess?

We got here because

"every" atmospheric model needs a radiation code, but
most modeling efforts don't have deep in-house radiation expertise because
the field is mature: spectroscopic understanding changes slowly and new ideas are rare, so
people stick with the code they have

But the codes we have are black boxes to most users, who assume that they are accurate in all circumstances, e.g. 4xCO_2, runaway greenhouse, exoplanets… where accuracy is in fact unknown.

Codes don't change because ideas develop slowly… but computers change fast, and the complexity of radiation codes makes them difficult to adapt to new-generation architectures

A multiplicity of goals

We have developed a radiation parameterization for computing broadband fluxes that seeks a balance among

- Accuracy via data and algorithms
- Efficiency expansively defined
- Flexibility across applications, models, computing architectures

Our goal is to be useful in many contexts including

- developers or centers without local radiation expertise
- offline radiation calculations
- …

RTE+RRTMGP

We (U. Colorado, AER) have developed a new radiation tool for atmospheric models, based on RRTMG and PSrad

RTE (Radiative Transfer for Energetics) for flux calculations

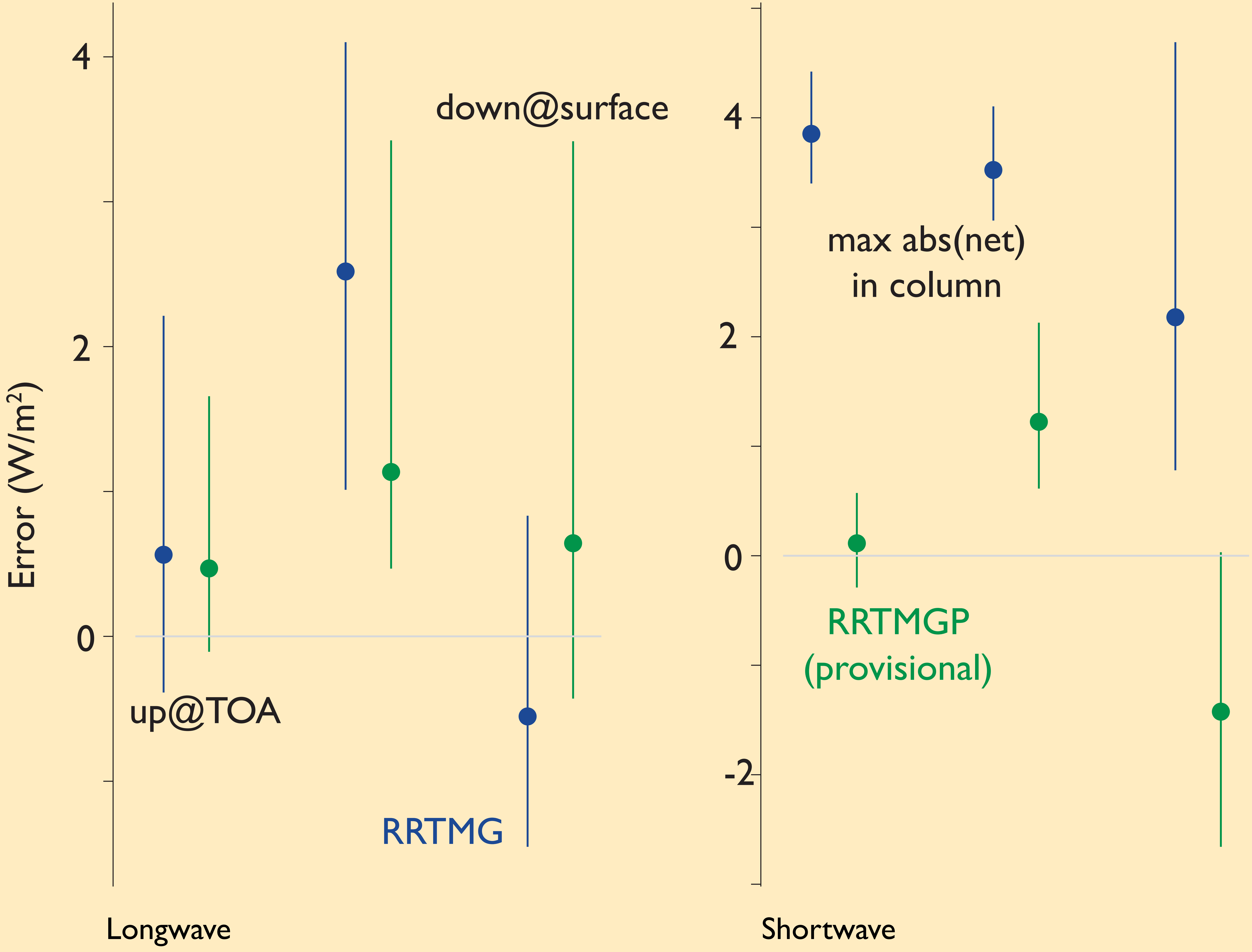
- One-dimensional plane-parallel radiative transfer equations, using either
- absorption/emission calculations or the
- two-stream approximation to compute layer properties and adding to compute transport
- extensible to multi-stream methods

RRTMGP (RRTM for GCMs, Parallel) for gas optical properties

- Correlated *k*-distribution to treat spectrally-dependent absorption and scattering by gases, built using
- up-to-date spectroscopy based on current AER line parameters + MT_CKD water vapor continuum
- encompassing code and data (currently a high-resolution general purpose *k*-distribution)
- using much simpler interpolation algorithms than RRTMG

RTE and RRTMGP can be used (almost) independently

Accuracy: Errors in Garand atmospheres



Designing a radiation library

Strict separation of concerns: gas optics, condensate optics, transport, reduction

Data and code are disjoint. Data can be targeted to applications

Trailing edge algorithms: most have been used in the community for 20+ years

Multiple columns: exposes more parallelism, matches problem size to architecture

Small (~120 lines), high-efficiency kernels with language-interoperable interfaces: accessible, replaceable, traceable (because unit testing)

Solvers

```
two_stream_sw(ncol, …
  tau, …,
  Rdif, …) bind(C)

adding(ncol, …
  Rdif, …,
  flux_up, …)

solver_sw(ncol, …
  tau, …,
  flux_up, …)
```

Optical properties

```
validate(ncol, …
  tau, w0, g, …)

delta_scale(ncol, …
  tau, w0, g, f)

increment(ncol, …
  tau1, …,
  tau2, …)
```

Gas optics

```
gas_optics(ncol, …
  p_lay, t_lay,
  gas_concs, …,
  tau, …)
```

Easing interactions with the radiation library

Kernels are bound together with classes (currently Fortran 2003). We use classes to

- bundle related data, code
- hide unimportant details
- provide facilities (e.g. adding clouds to clear-sky)
- increase efficiency by minimizing data transfer
- pre-process e.g. validate inputs

We will see how much people hate the classes.

They do provide a way to safely provide access to fine-grained calculations without moving lots of data around.

- User-extensible output (flux) class brings user code to fully-detailed calculations
- allows perfect spectral and spatial control over summary (broadband, PAR, …)

Solvers

```
two_stream_sw(ncol, …
  tau, …,
  Rdif, …) bind(C)

adding(ncol, …
  Rdif, …,
  flux_up, …)

solver_sw(ncol, …
  tau, …,
  flux_up, …)
```

```
call rte_sw(op,
  mu0, incident_flux,
  fluxes)
```

Optical properties

```
validate(ncol, …
  tau, w0, g, …)

delta_scale(ncol, …
  tau, w0, g, f)

increment(ncol, …
  tau1, …,
  tau2, …)

class :: op_2str
  real :: tau(ncol,…)
```

```
call op%delta_scale()
```

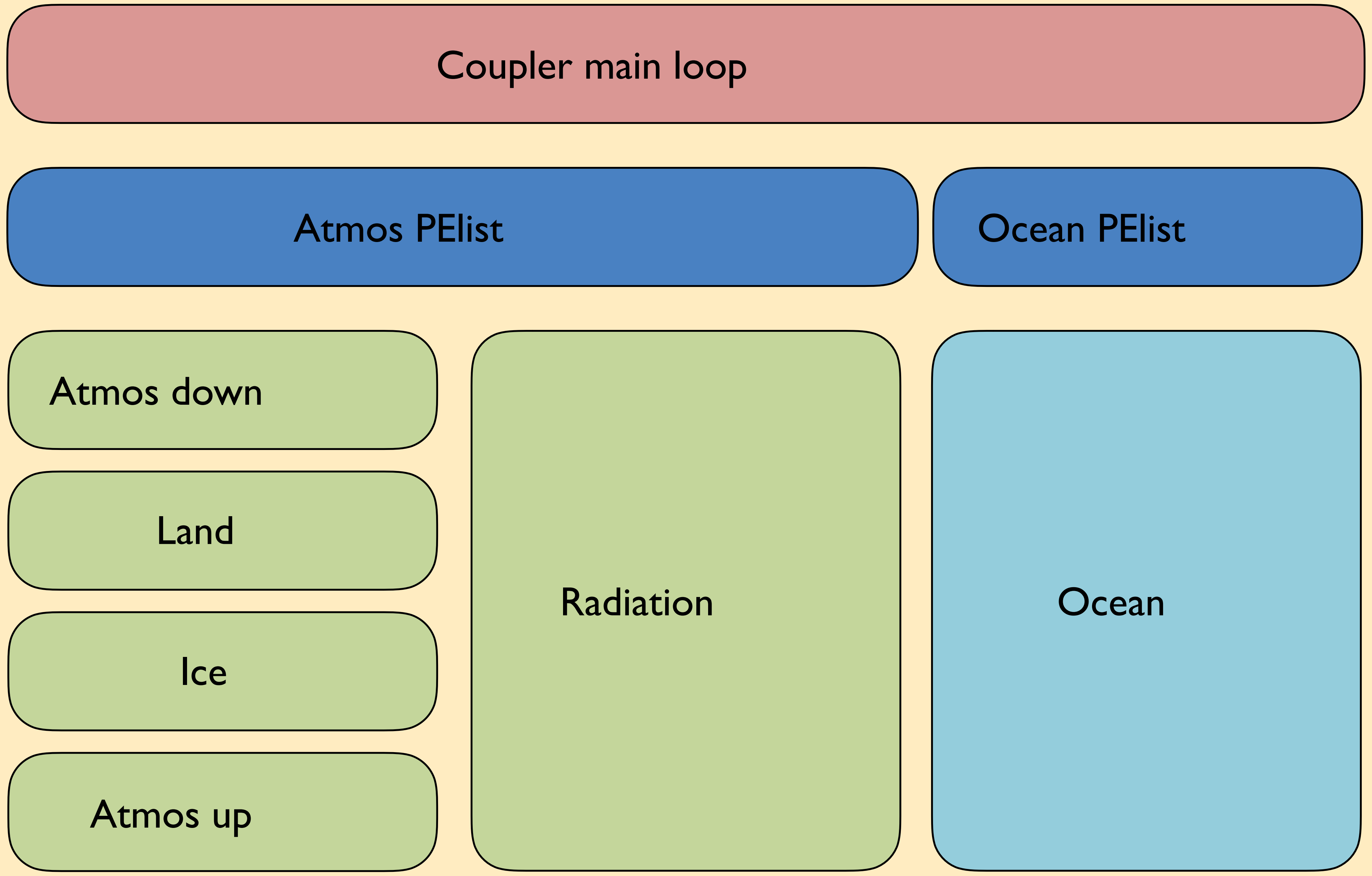
Gas optics

```
gas_optics(ncol, …
  p_lay, t_lay,
  gas_concs, …,
  tau, …)

class :: gas_optics
  real :: kmajor(…)

init(kmajor,…)
```

```
call go%init(kmajor,)

call go%gas_optics()
```

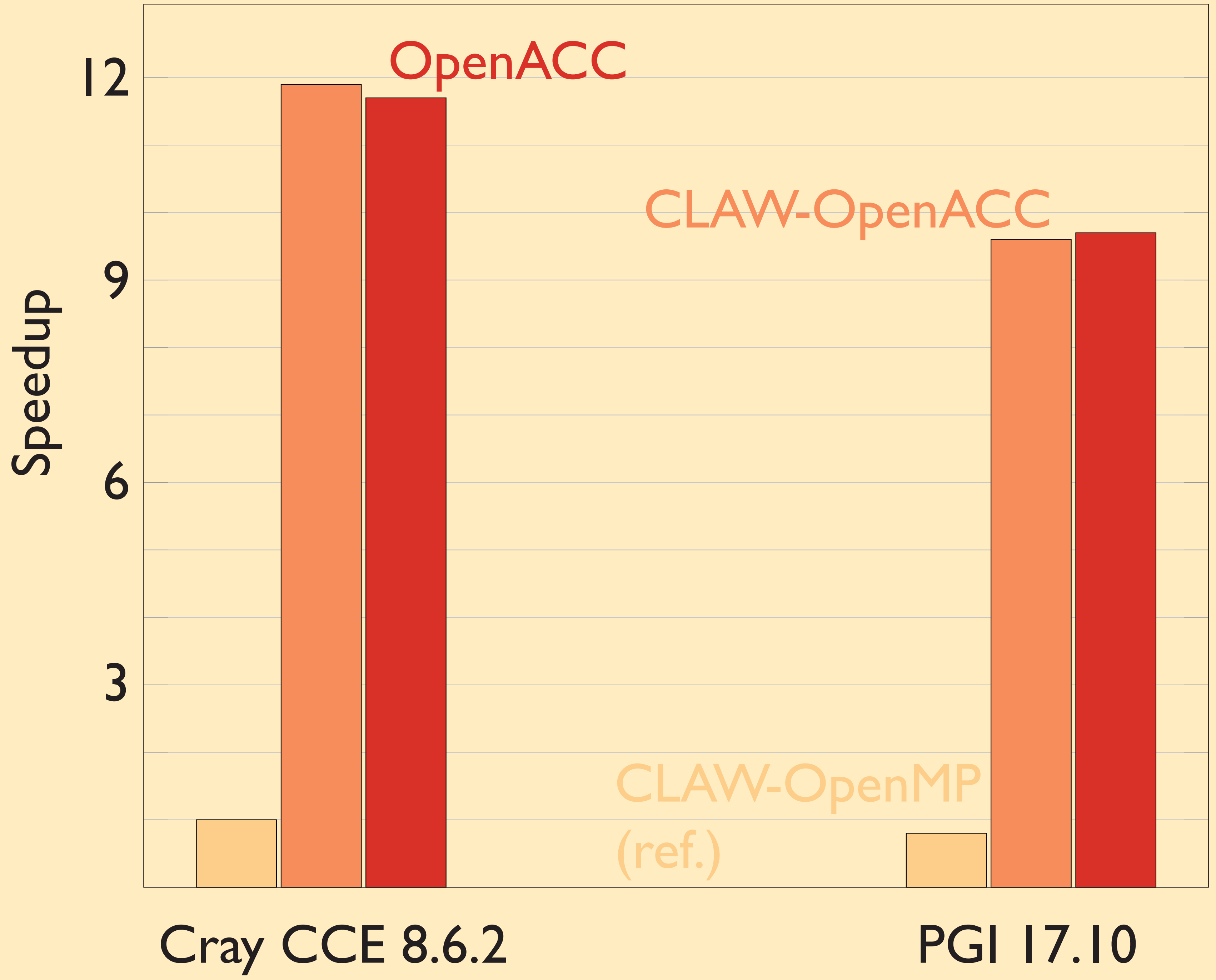
Using kernels and classes for agility

We gave minimize data transfers to make it easier to use dedicated computational resources for radiation…



after V. Balaji et al., 2016: 10.5194/gmd-9-3605-2016

… including co-processors, because radiation has so much fine-grained parallelism



Valentin Clément, ETH (PASC 18)

“With freedom comes responsibility”

The intent of the class/kernel structure is to provide the efficiencies of a black box with the flexibility required to address a range of applications.

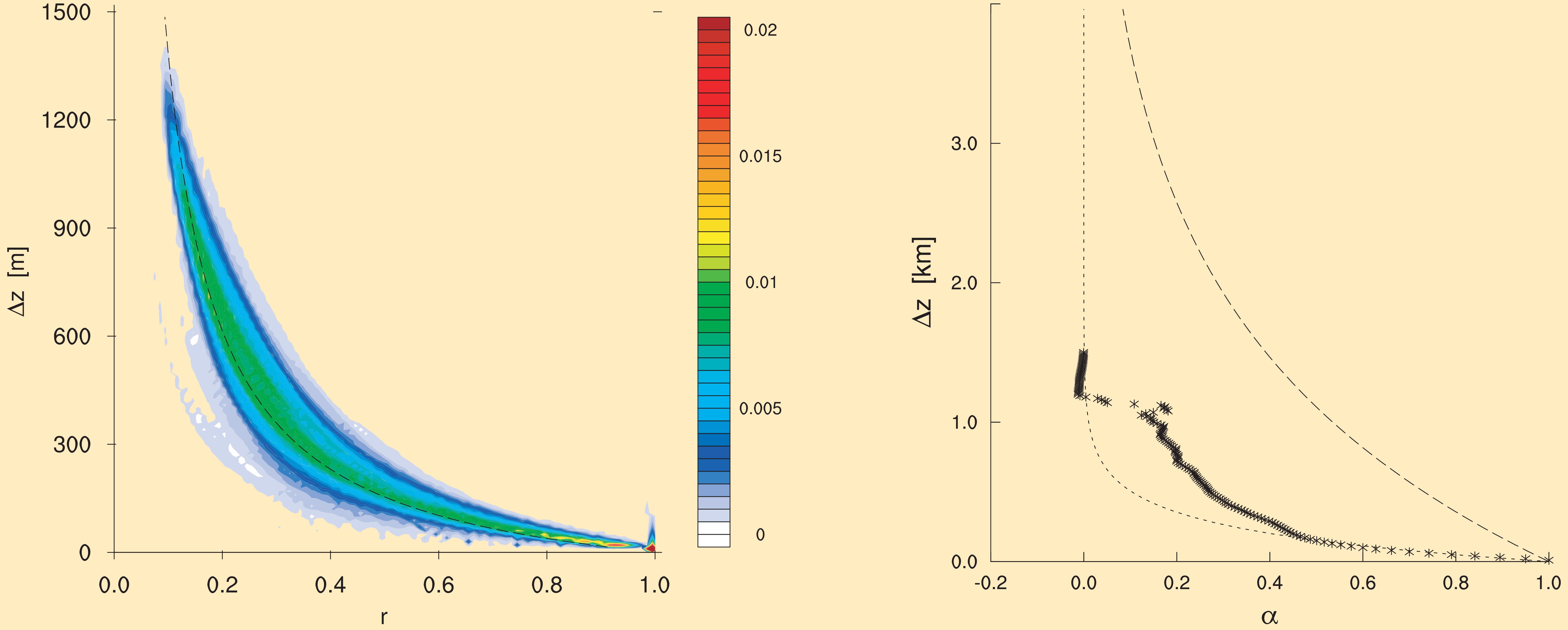
As one result RTE focuses on the radiative transfer problem: inputs are described in purely optical terms.

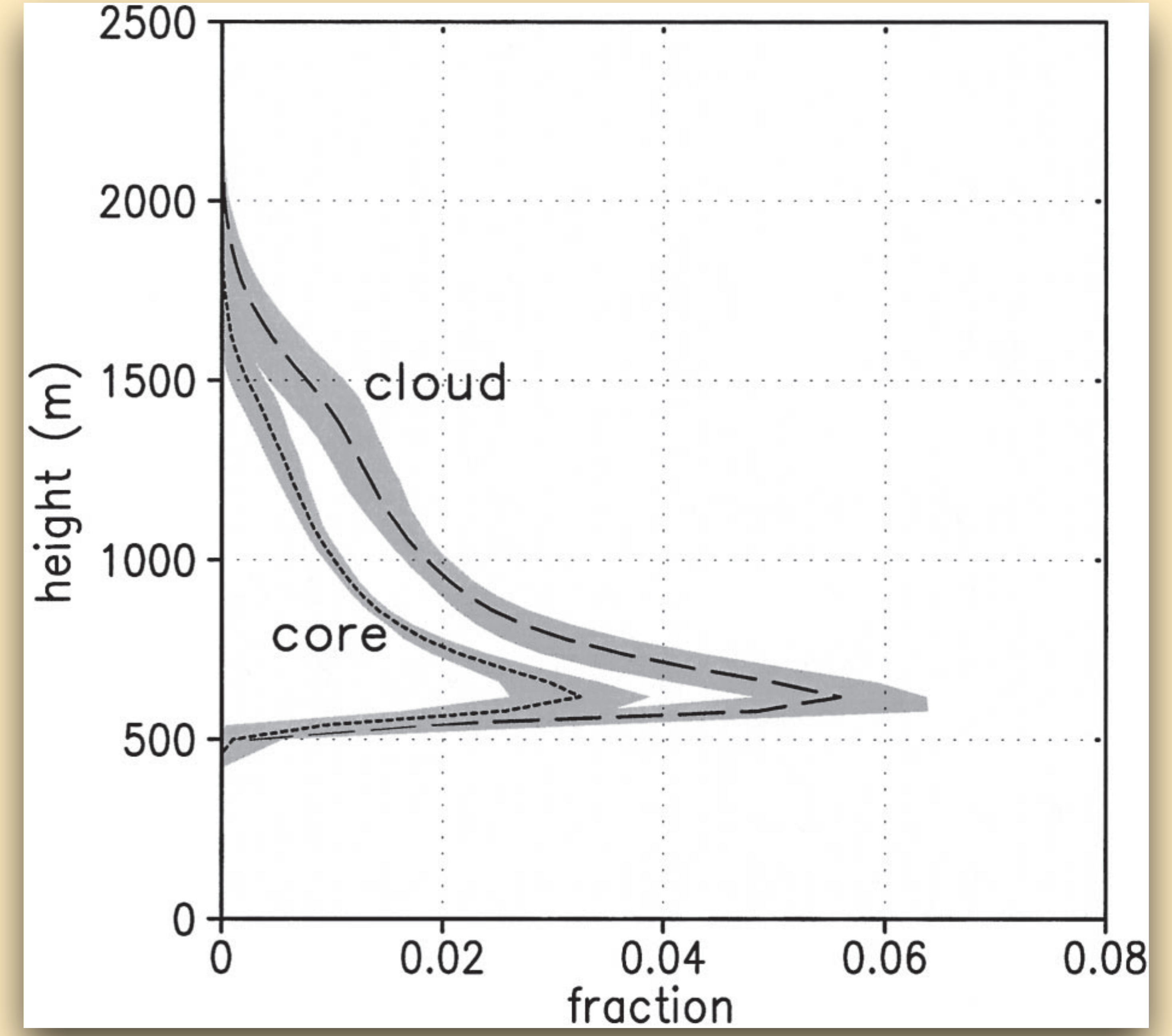
Users have more freedom and responsibility in coupling than usual: by requiring radiative inputs we require users to be explicit about microphysics, macrophysics, and overlap for clouds and aerosols.

Or: by stressing flexibility we’re punting these issues to the user (though we provide* a simple example implementation of cloud optics)

We’re doing this because the problem is not well-posed: the mapping of physical state to optical state introduces ambiguity (or perhaps uncertainty).

We expect users to make choices based on empirical relationships.



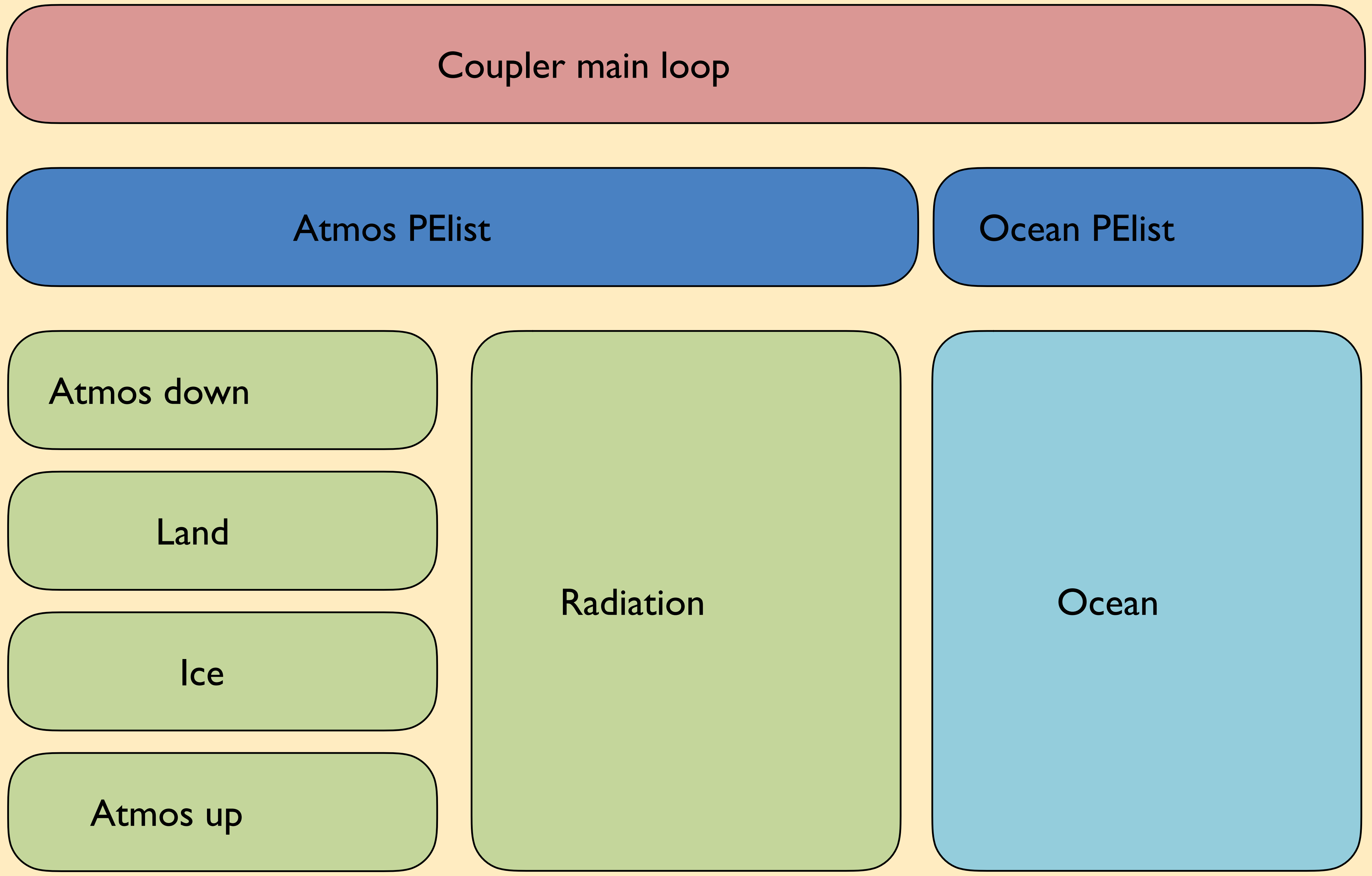


Neggers et al. 2011:
10.1029/2011JD015650

Siebesma et al. 2003:
10.1175/1520-0469(2003)60<1201:ALESIS>2.0.CO;2

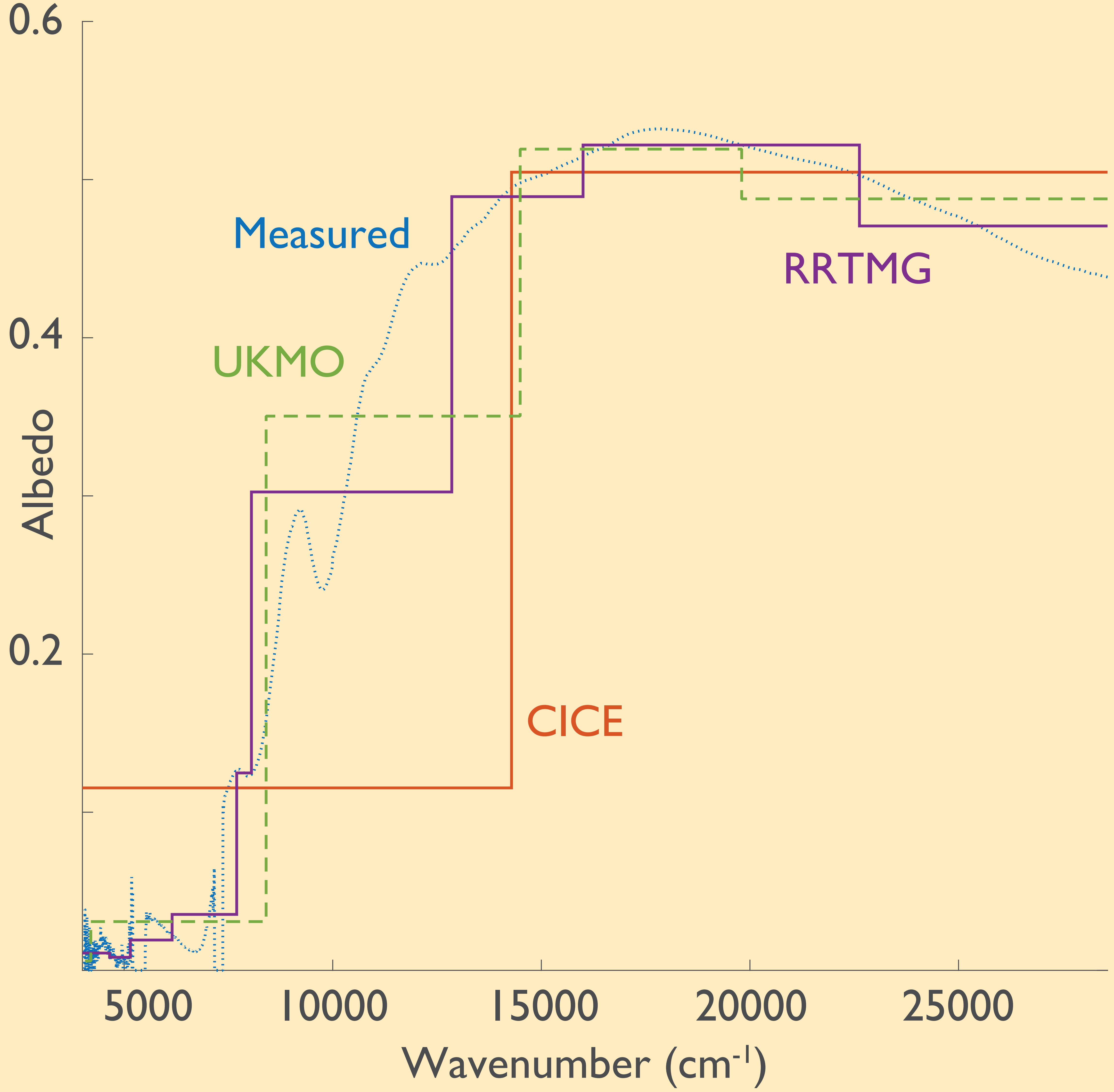
Reimagining coupling for better forecasts

As long as we are thinking about treating radiation as its own process…



after V. Balaji et al., 2016: 10.5194/gmd-9-3605-2016

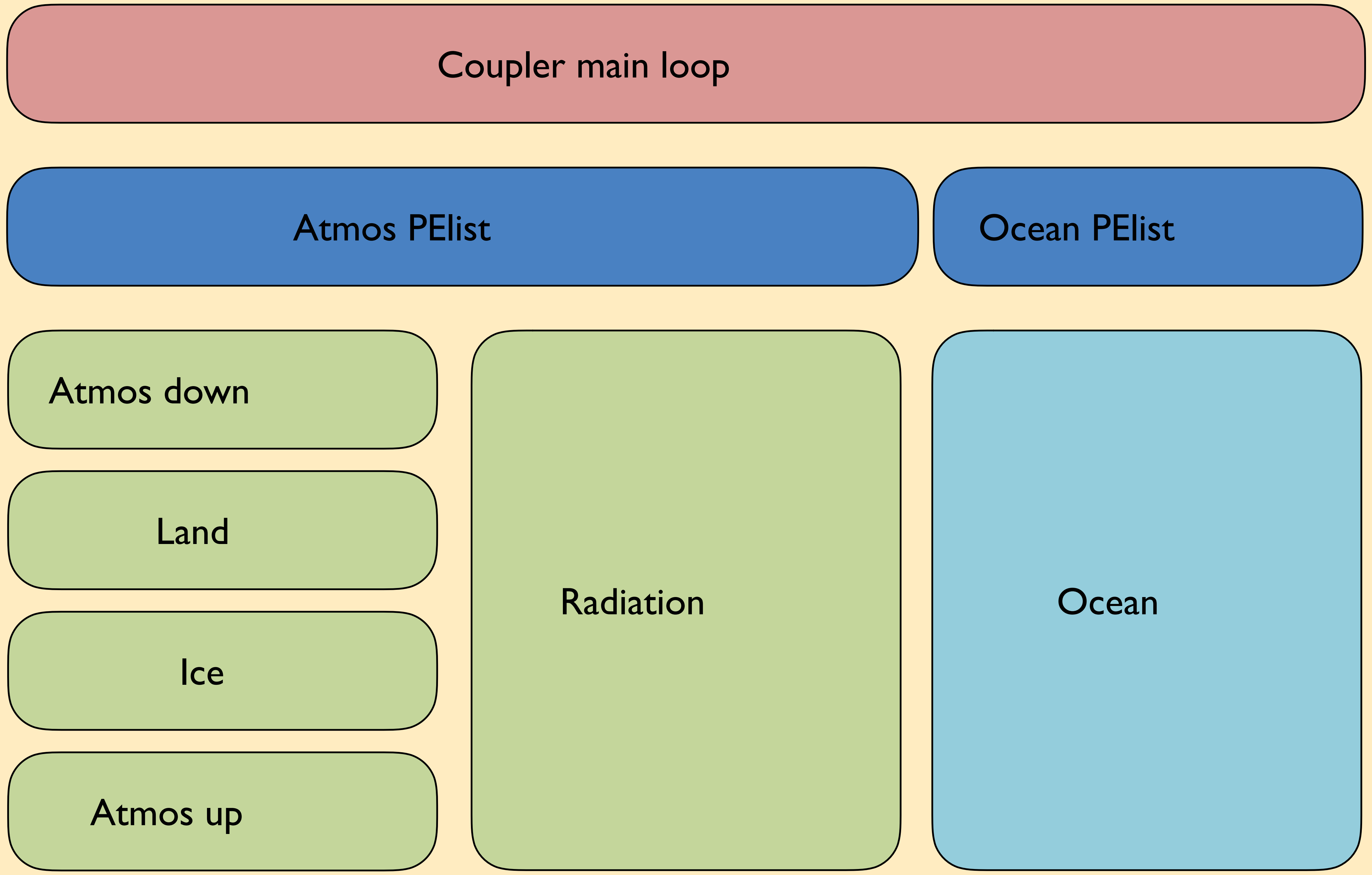
… might we consider coupled ocean-atmosphere radiation?

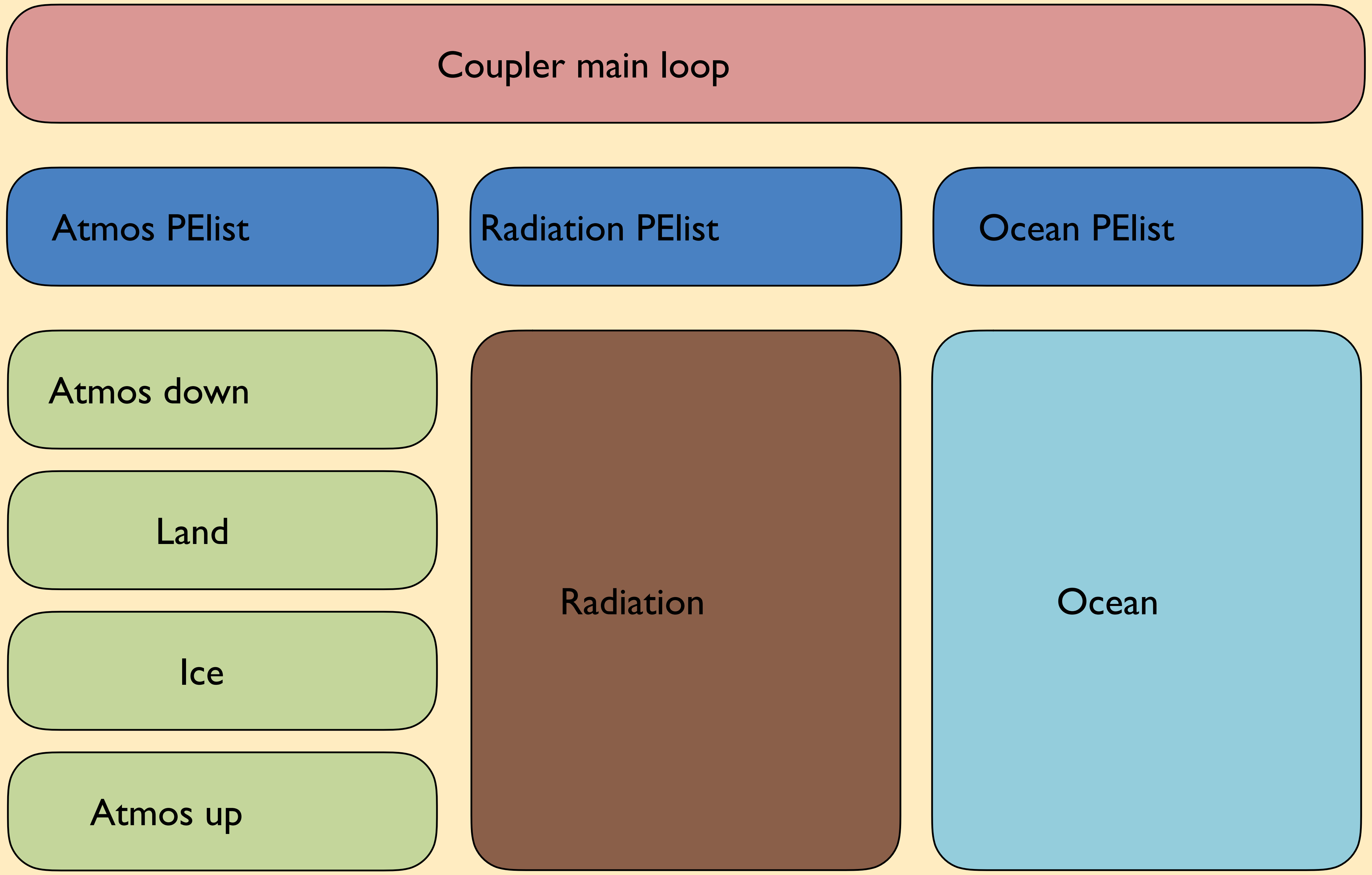


Reimagining coupling for better forecasts

As long as we are thinking about treating radiation as its own process…

… might we consider coupled ocean-atmosphere radiation?





after V. Balaji et al., 2016: 10.5194/gmd-9-3605-2016

On building targeted data

1) The initial release of RRTMGP includes a general-purpose *k*-distribution (pre-industrial to 4xCO2) and high accuracy across a range of metric chosen by us.

Equivalent accuracy for more focussed applications (e.g. NWP) could be achieved with fewer spectral points

2) The view of radiation codes as black boxes has encouraged people to use parameterizations far outside the realm of applicability

RRTMGP takes a hard line: it doesn't run outside the (expansive) training range

The community would benefit from a toolchain that enabled the construction of custom *k*-distributions with control over

- sets of atmospheric conditions being tuned for
- spectral partitioning
- cost functions
- …