Optimized Radiative Transfer

Åke Nordlund
Niels Bohr Institute and
Center for Star and Planet Formation
University of Copenhagen

Four subtopics

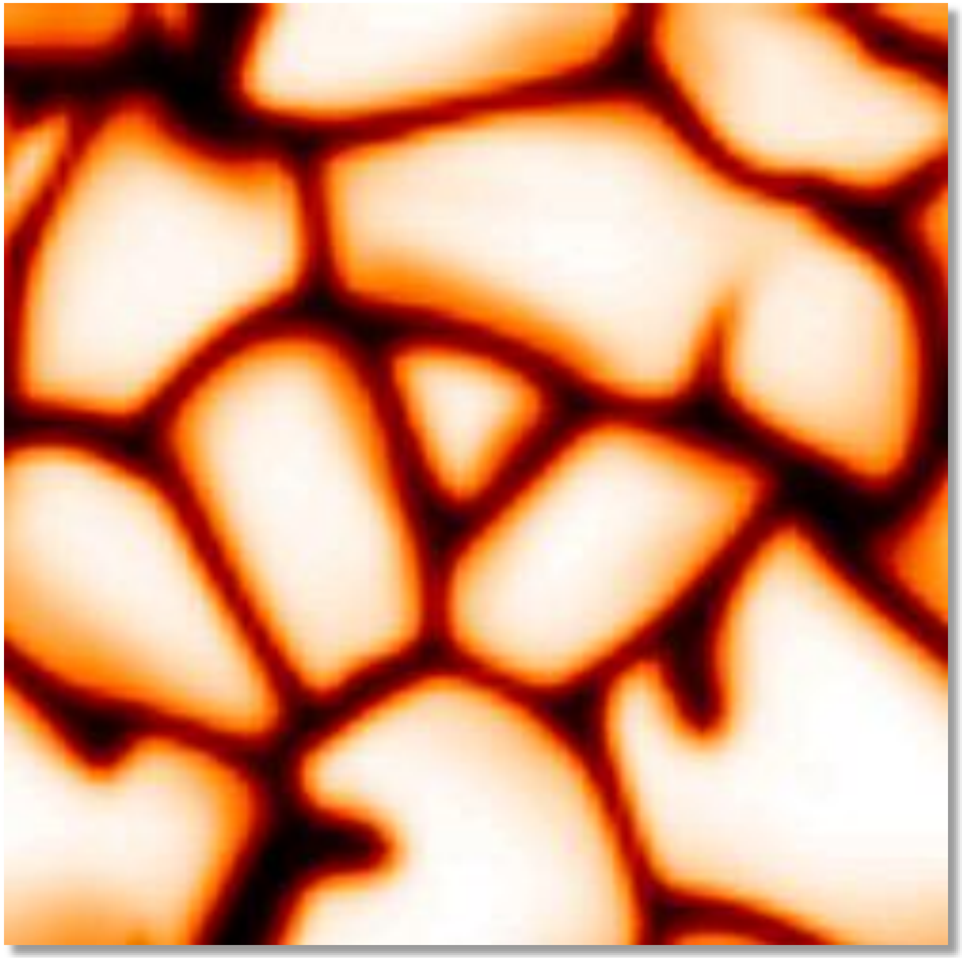
1. Quantifying Kolmogrov
2. Multi-group RT
3. Optimizing and Parallelizing RT
4. Time-dependent RT

Quantifying Kolmogorov

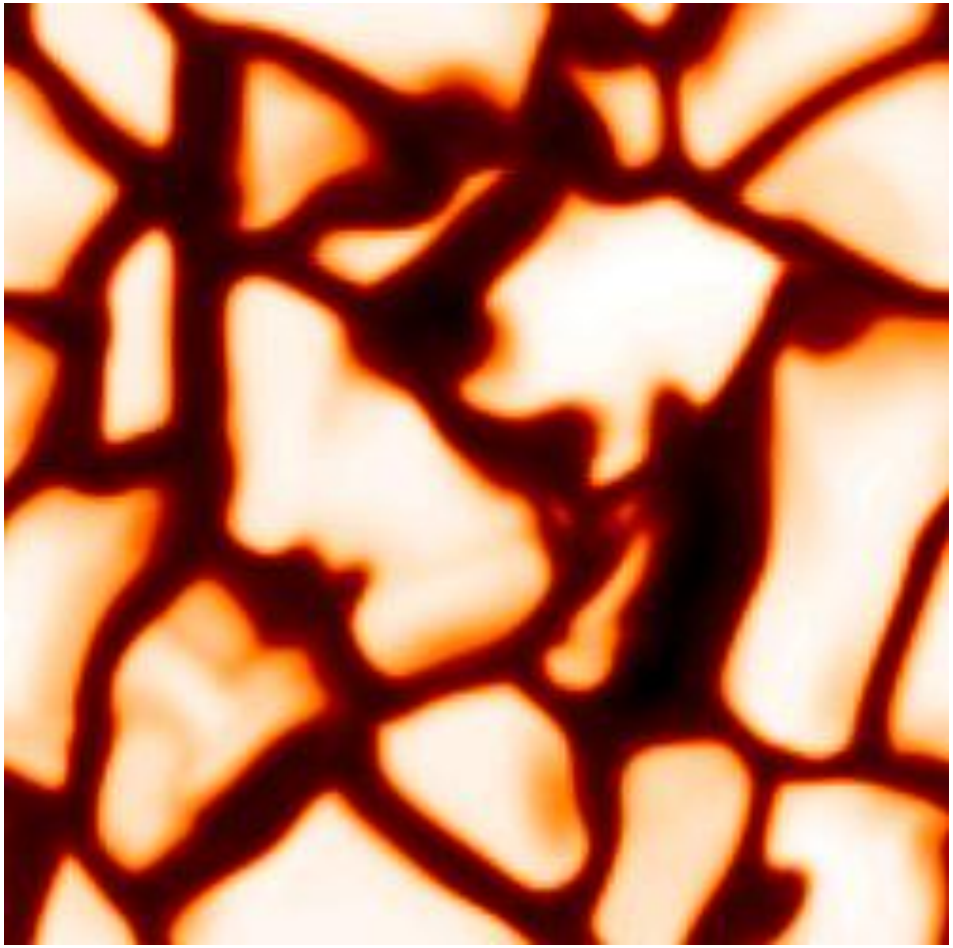
- Which resolution is ”sufficient”?

- How can we know?
  - Solar convergence study
  - Can be compared ***accurately*** with the Sun
    - RT diagnostics

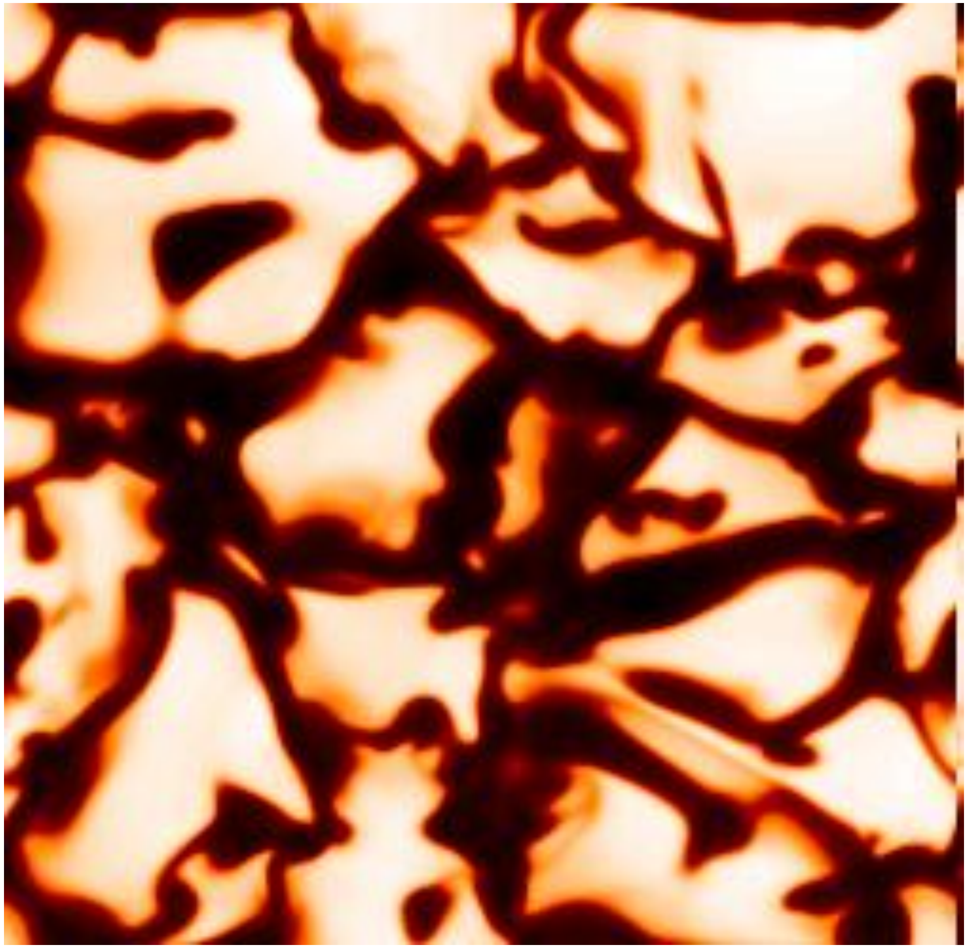
Emergent solar surface intensity mesh size: 96 km



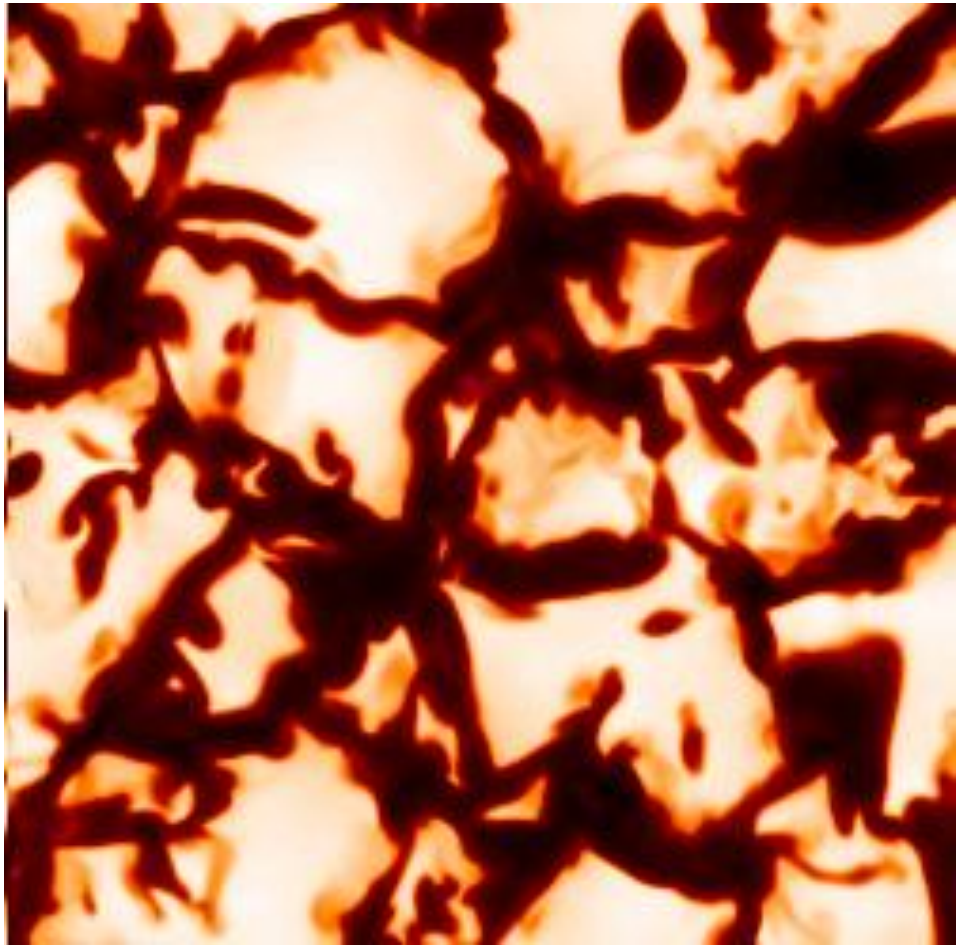
Emergent intensity mesh size: 48 km



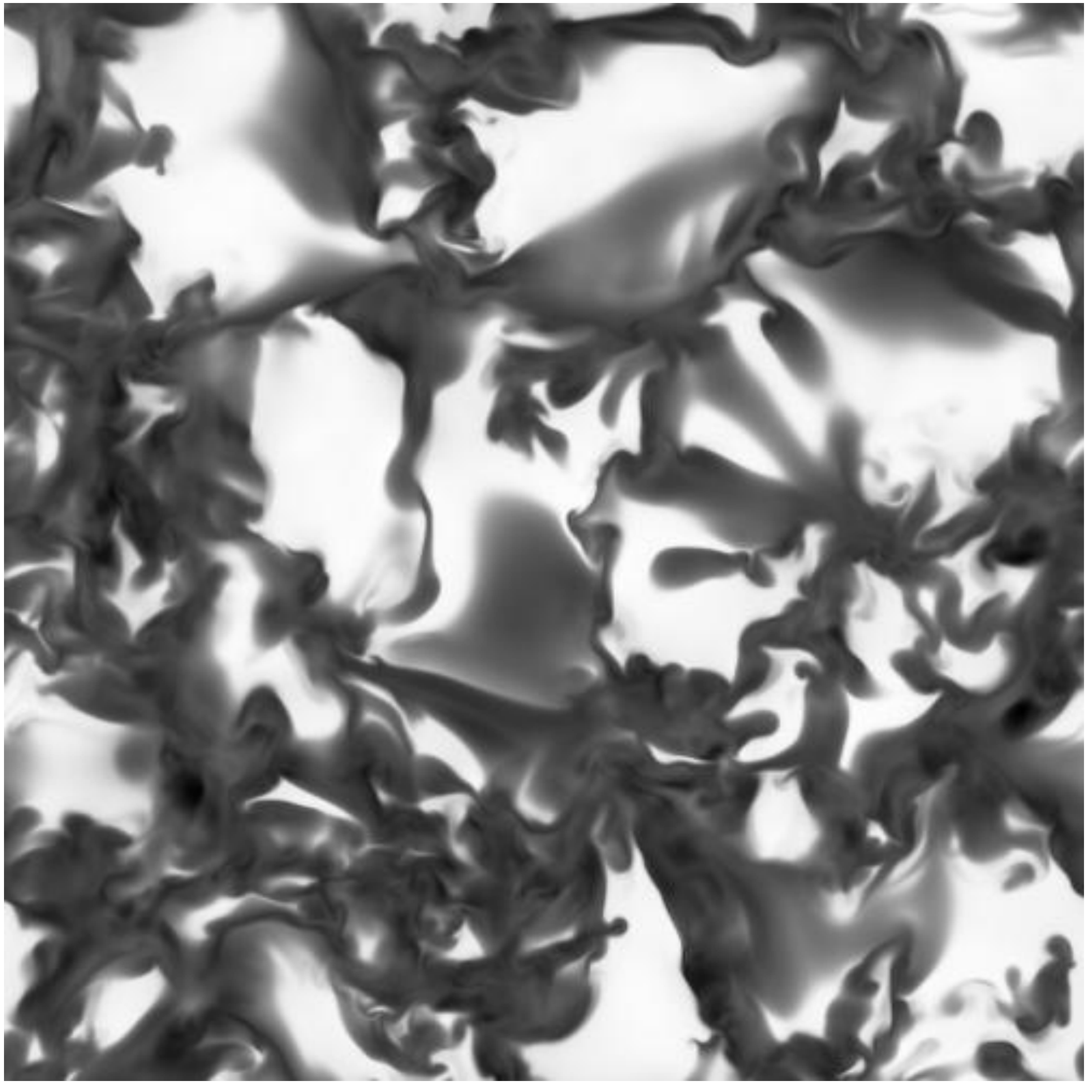
Emergent intensity mesh size: 24 km



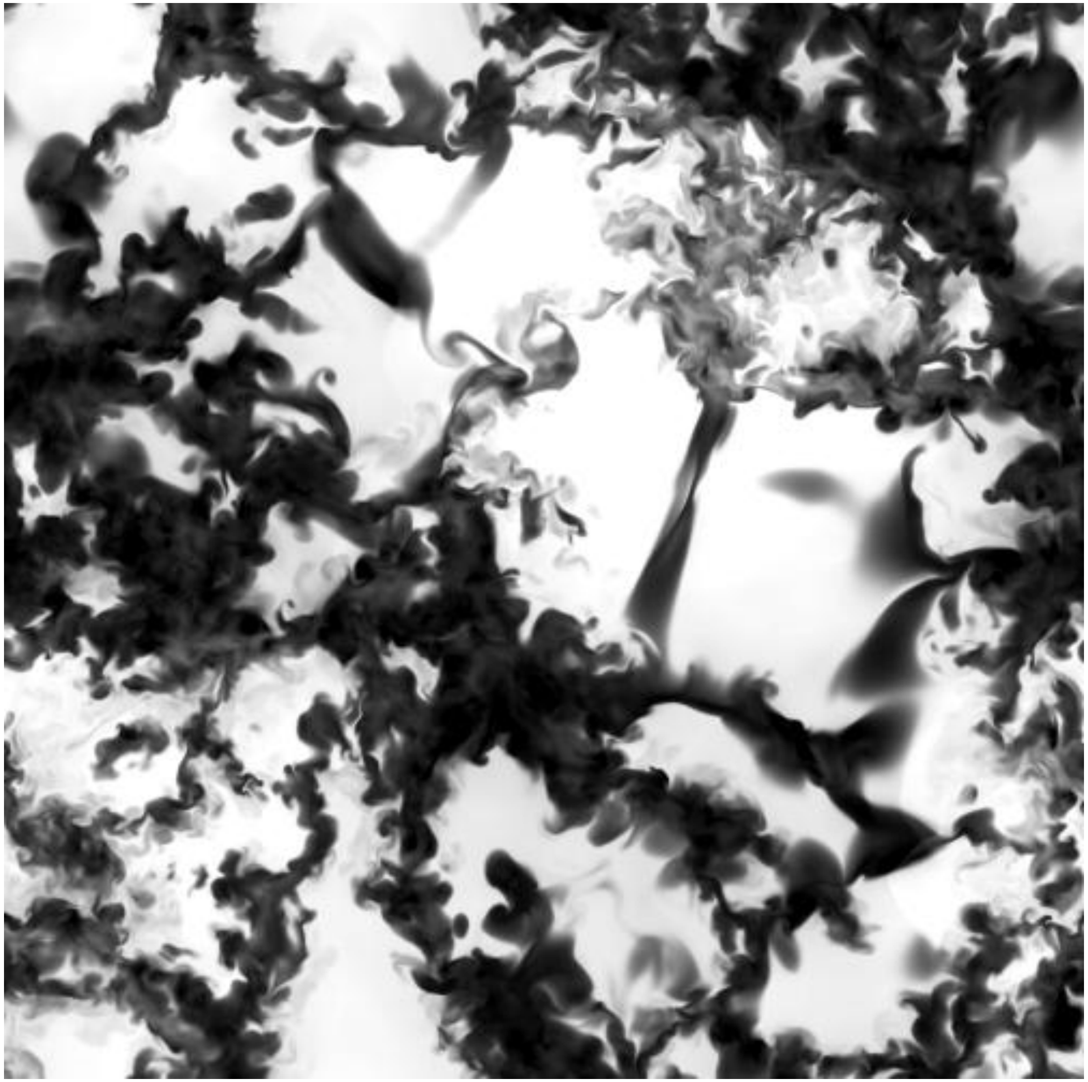
Emergent intensity mesh size: 12 km



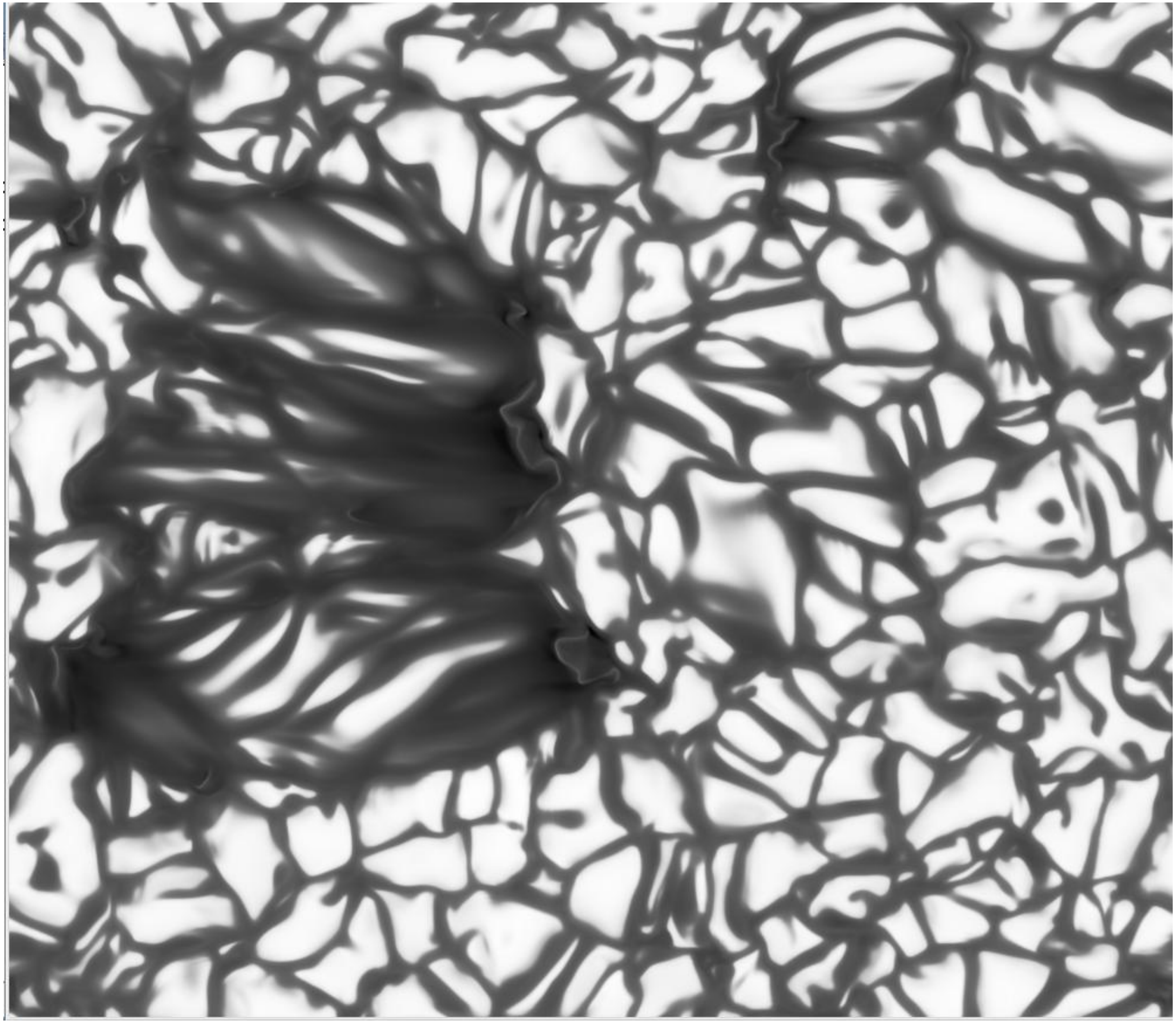
Temperature in continuum layers 12 km mesh size (504x504)



Temperature in continuum layers 6 km mesh size (1008x1008)



Active region emerging flux

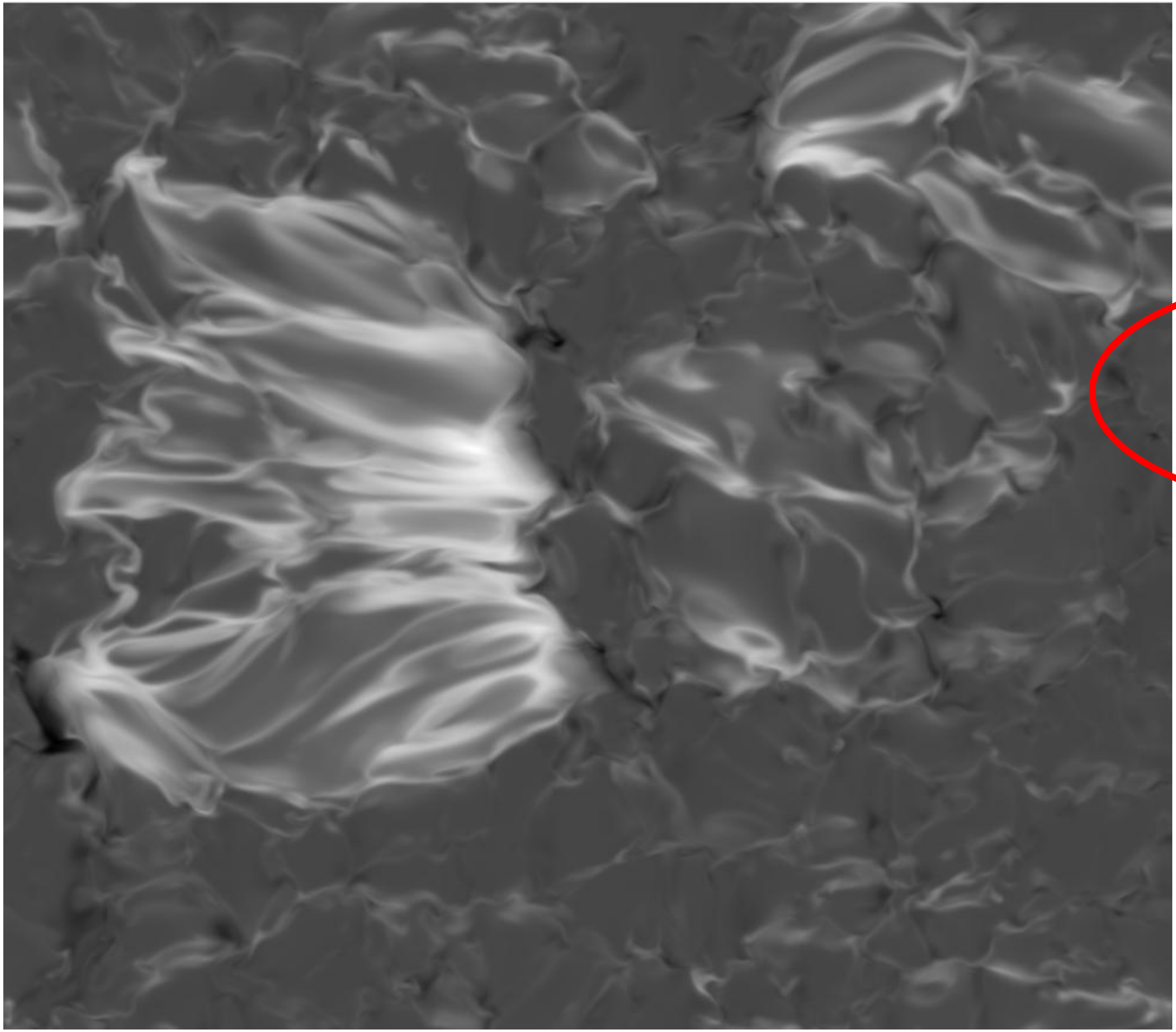


24x24x20 Mm

2000x2000x500

~12 km mesh

Active region emerging flux



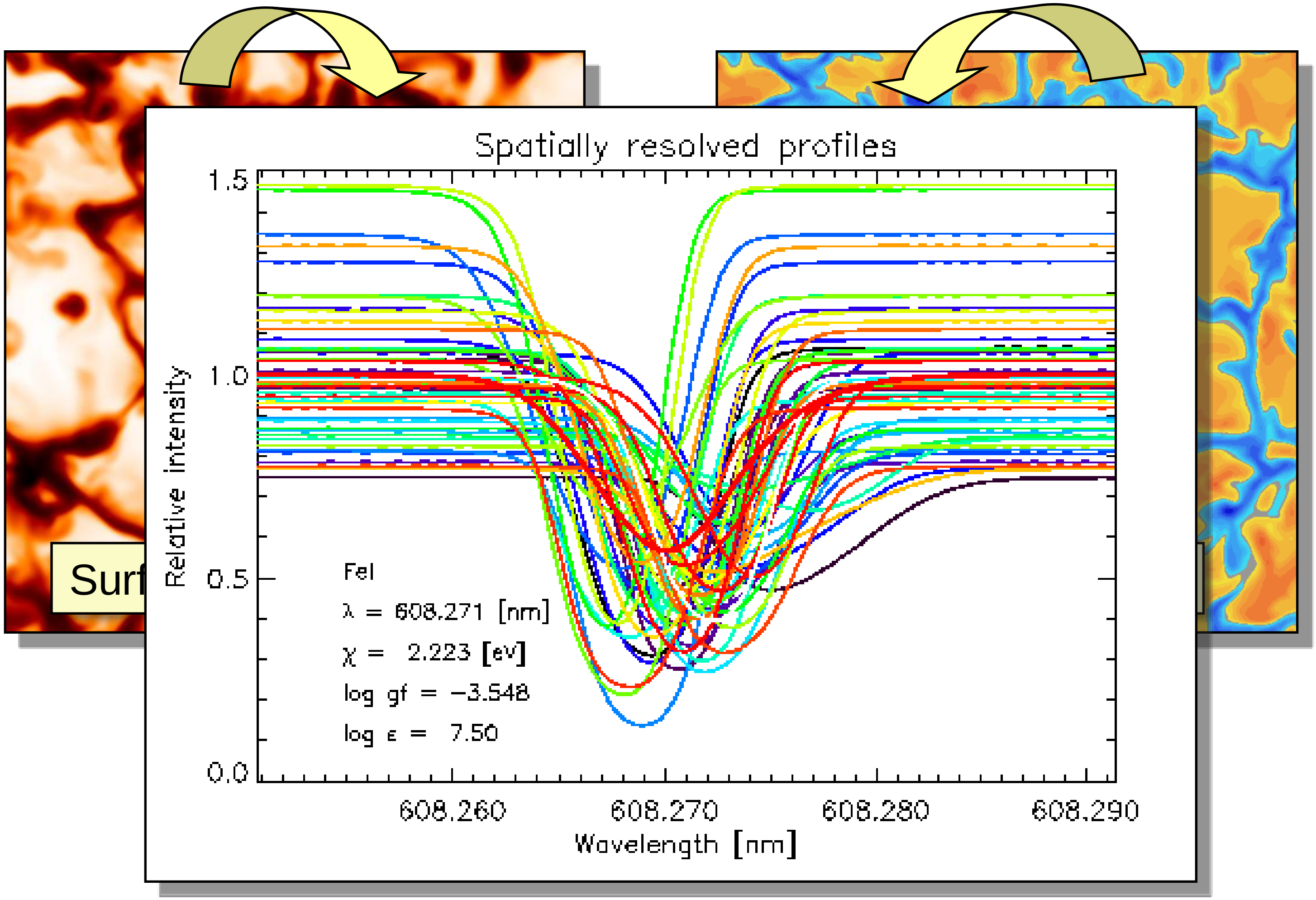
24x24x20 Mm

2000x2000x500

~12 km mesh

Synthetic spectral lines

Asplund et al (2002)



Spatially resolved spectral line profiles

Spectral line, with and w/o convective velocity field

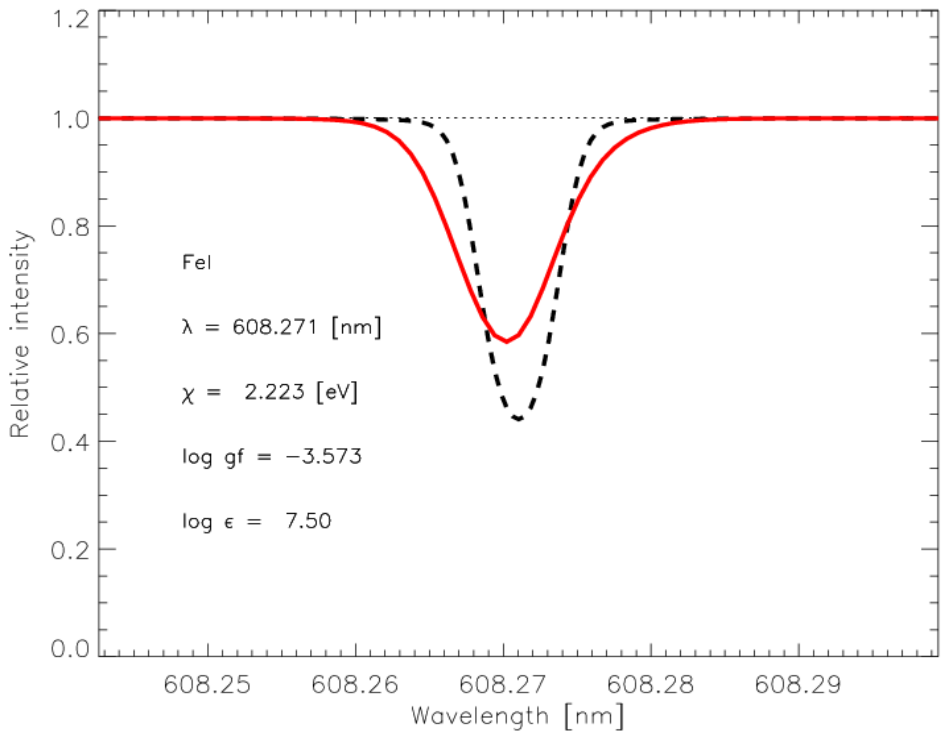


Figure 28: The predicted spatially and temporally averaged 3D LTE solar line profile of a typical Fe I line (solid line) compared with the corresponding calculation when ignoring all Doppler shifts arising from the photospheric velocity field (dashed line), demonstrating the importance of convective line broadening. The latter profile closely resembles 1D line profiles without application of the fudge parameters micro- and macroturbulence.

Synthetic spectral lines
Asplund et al (2002)

- Spectral line widths measure **velocity amplitudes**

- Convective blue-shifts and line shapes measure **temperature – velocity correlations**

- The total energy flux is what it is ($\Rightarrow$ the solar luminosity)
  - **No free parameters!**
    - **Teff, log g, chemical abundancies $\Rightarrow$ the Sun!**

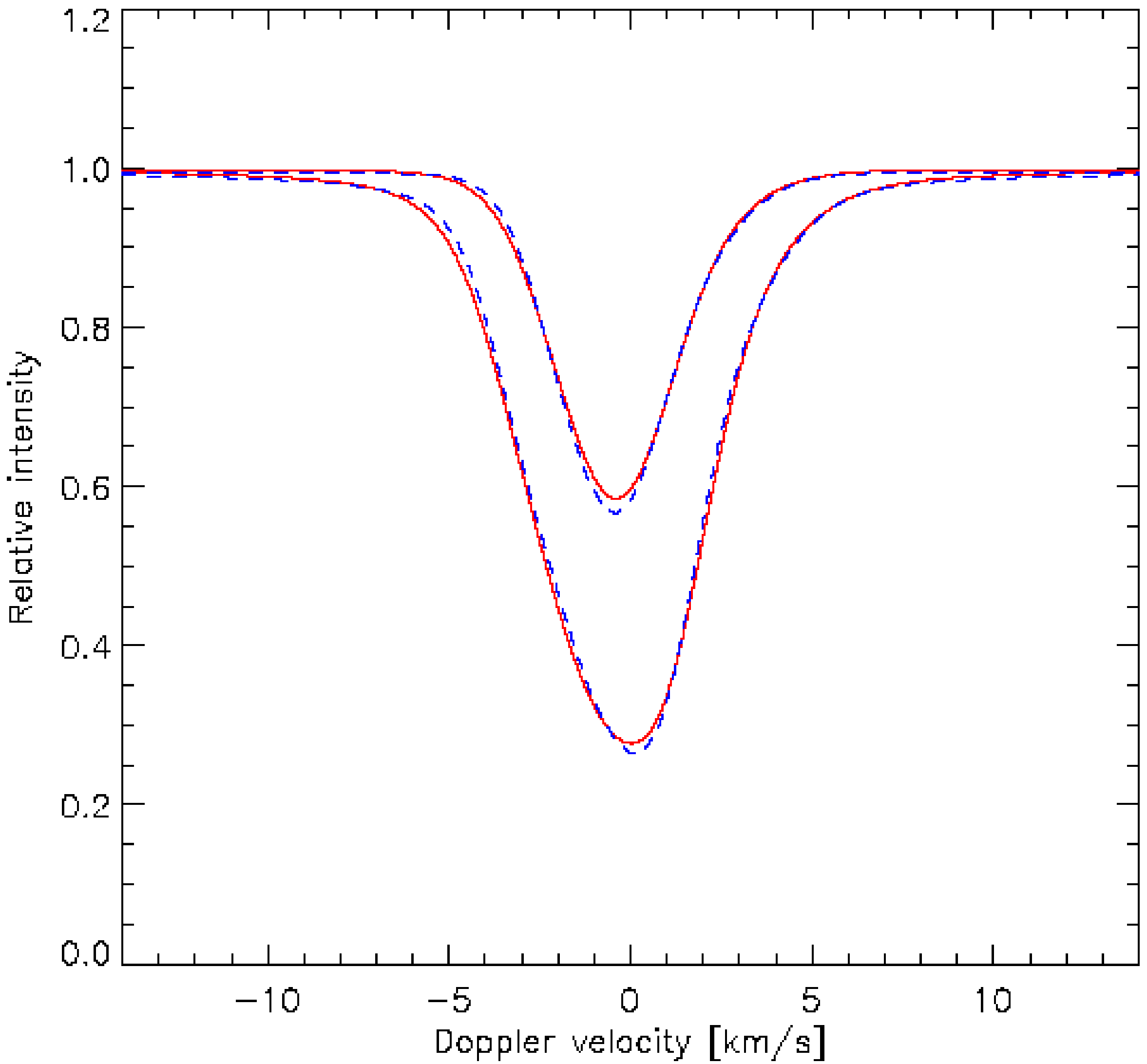


Fig. 5. The predicted Fe I 608.2 (weaker) and 621.9 nm (stronger) lines at two different resolutions of the solar convection simulation: 200×200×82 (solid) and 50×50×82 (dashed). All profiles have been computed with $\log \epsilon_{Fe} = 7.50$

3-D resolution: 50x50x82 & 200x200x82

This is a wonderful case; a test of the 'very large R_e' limit!

- The Sun:
  - has a **huge R_e**, and
  - knows how to 'compute' convection correctly
- The models:
  - have **only moderate R_e**, but
  - have realistic EOS, and detailed radiative transfer
  - "no free parameters"; fixed at solar values
- The models and the Sun agree to fraction of %!
  - Dravins & ÅN (1990), Asplund et al (2000)

Why is this evidence that nothing happens at Re = 10^{xx}?

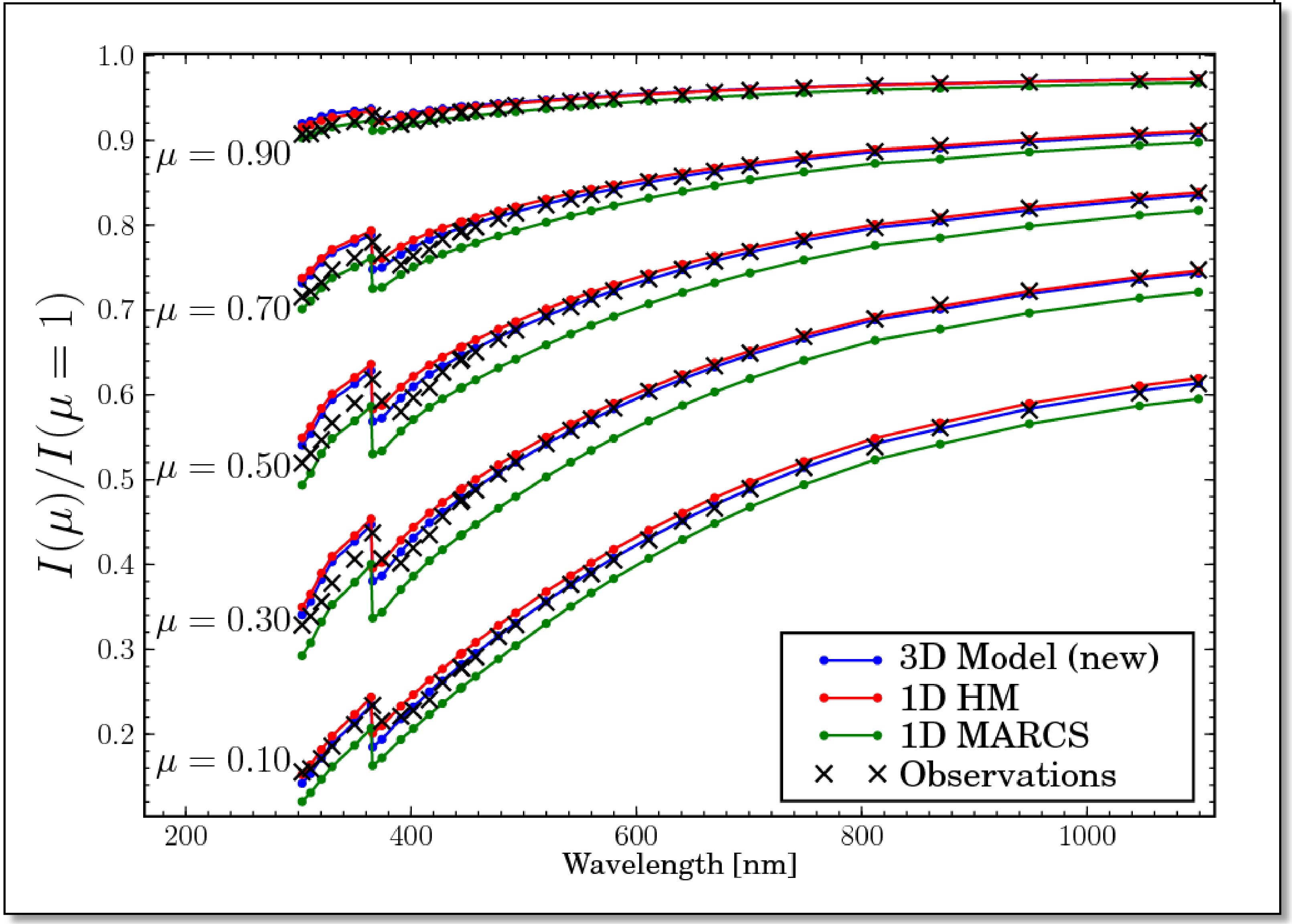
- What if turbulent dissipation had a transition to values different from those at moderate Re?

- The result would be a different balance btw buoyancy work and turbulent dissipation
  - Less dissipation $\Rightarrow$ higher velocity, lower ΔT
  - More dissipation $\Rightarrow$ lower velocity, higher ΔT

- Both line widths and line shifts would change!
  - **Would be inconsistent with observations!**

Multi-group Radiative Transfer

Pereira, Asplund & Trampedach (2009, in prep.)

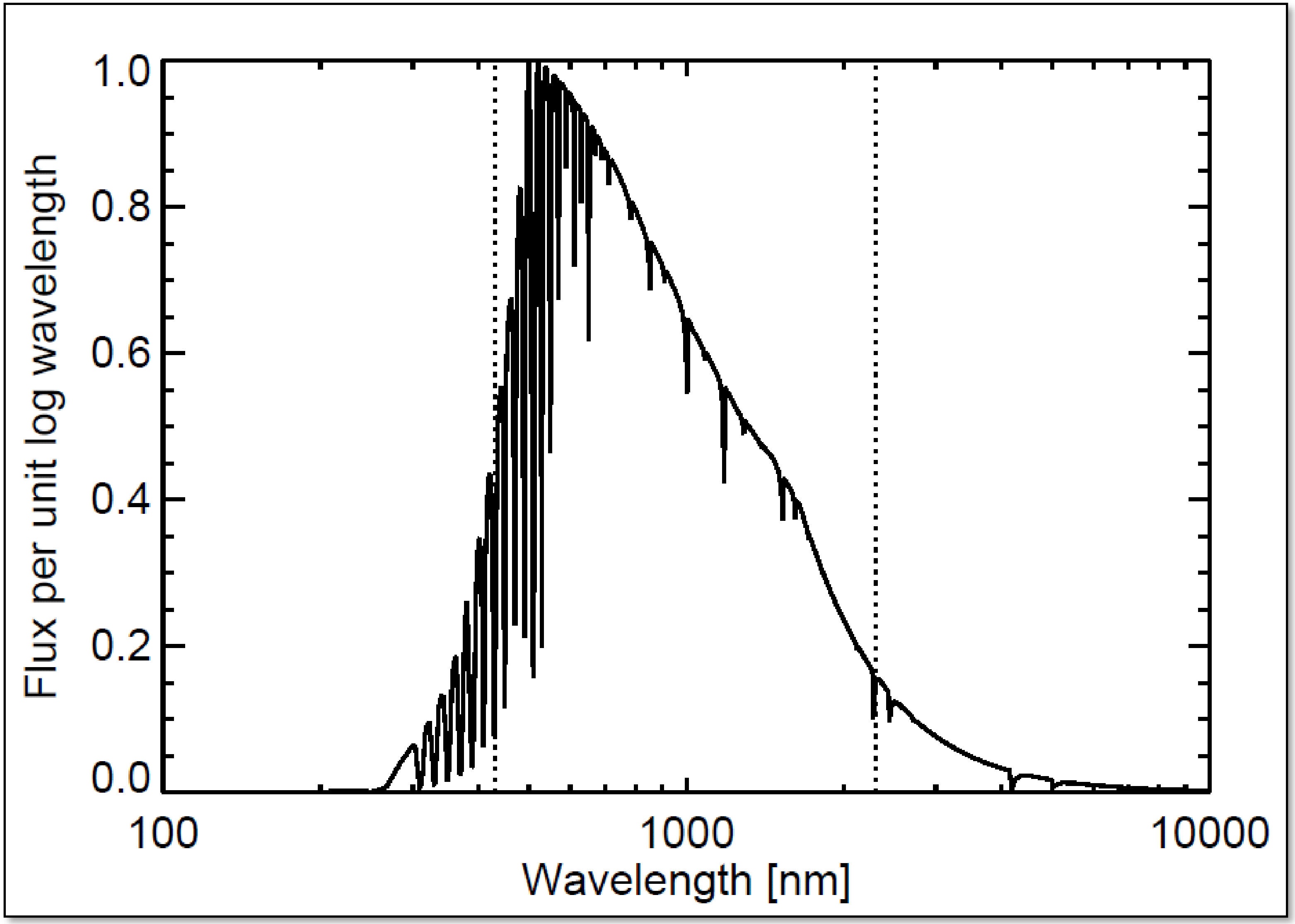


Basic radiative transfer; frequency dependence

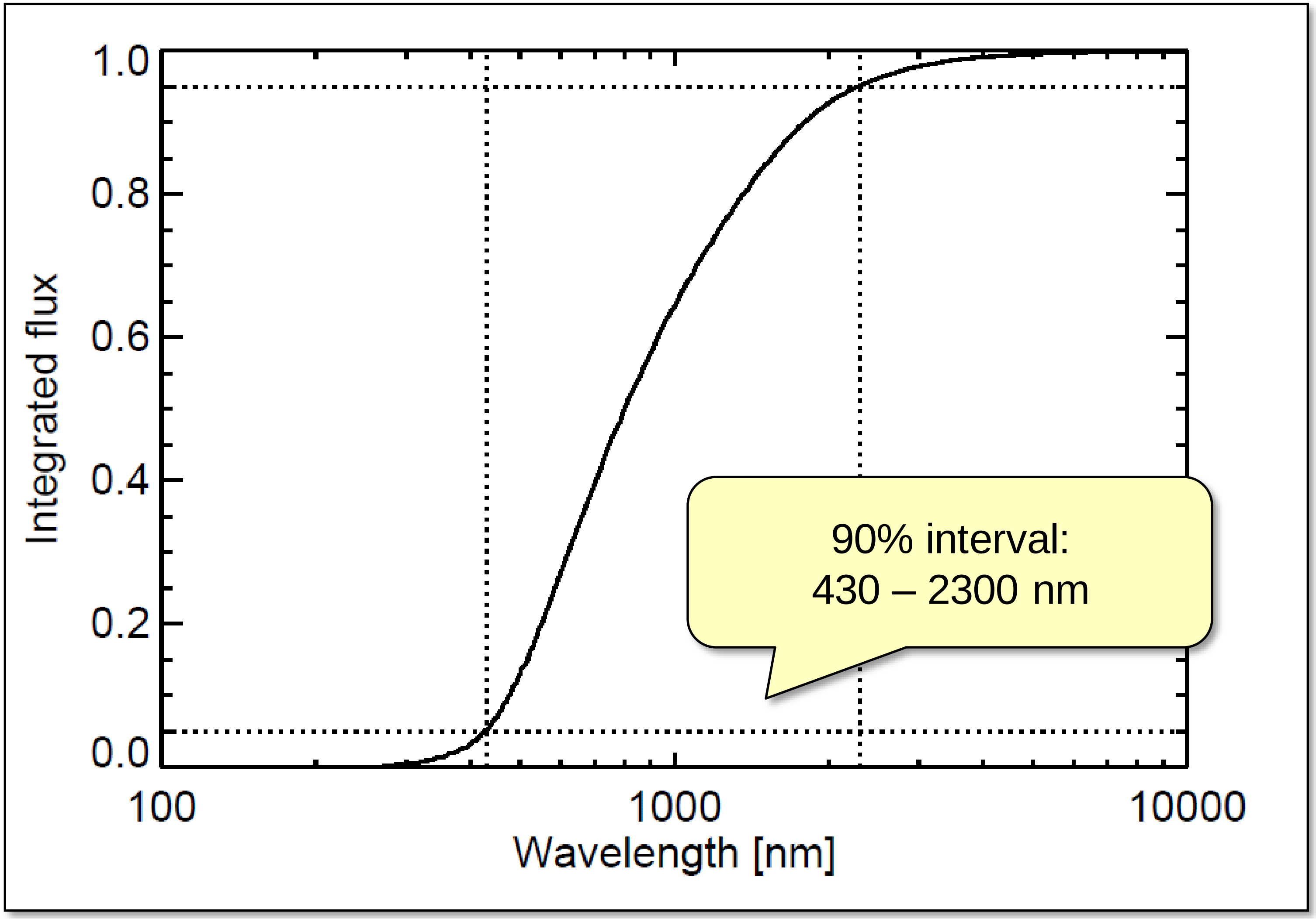
$$Q_{rad} = \int_{\nu} \rho \kappa_{\nu} (J_{\nu} - S_{\nu}) \, d\nu$$

- The mean intensity ***J-S***
  - goes inversely as the square of the opacity at large optical depth (diffusion limit)
  - is ~independent of opacity at low optical depth

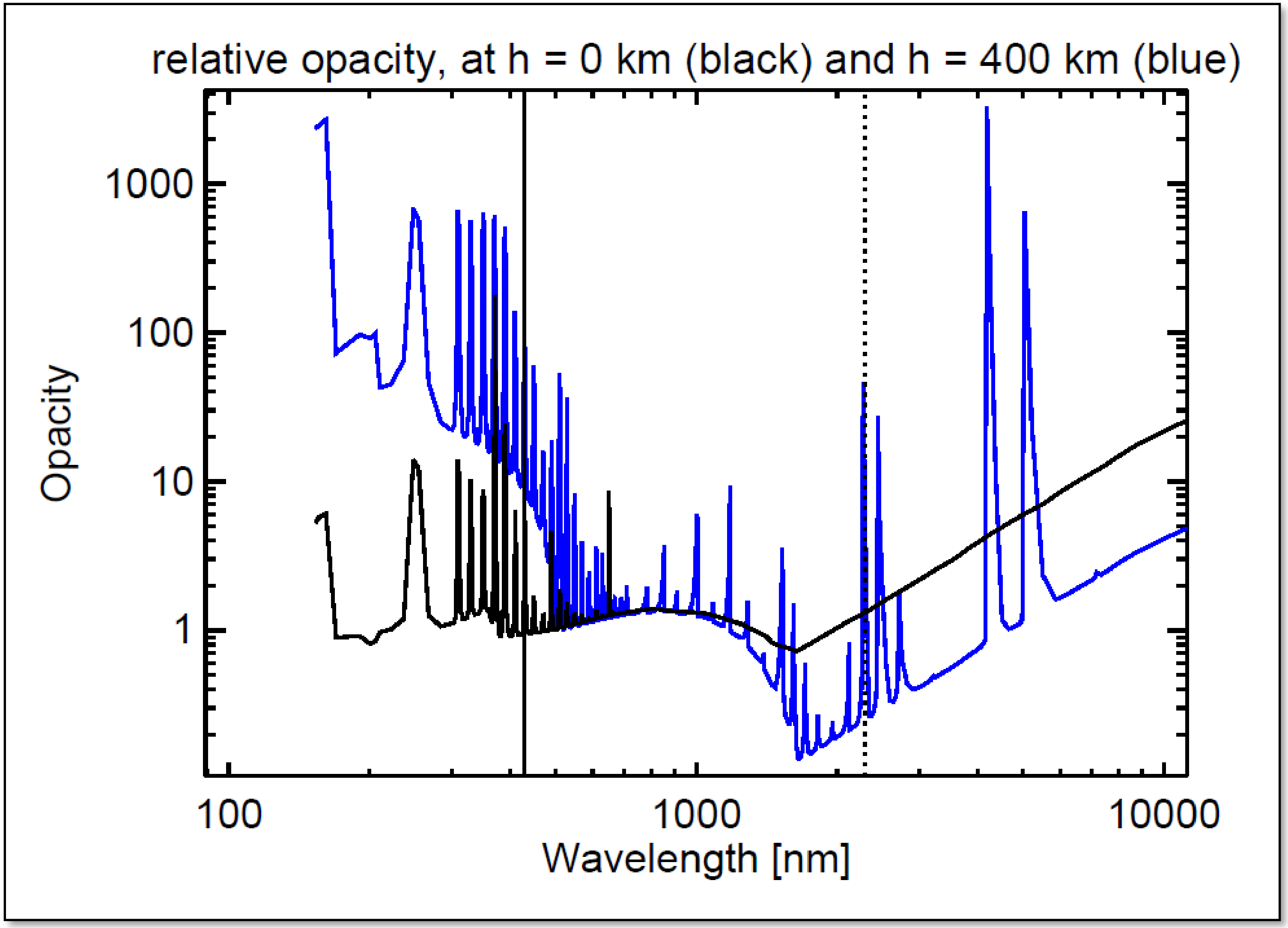
Solar flux; energy per unit log interval



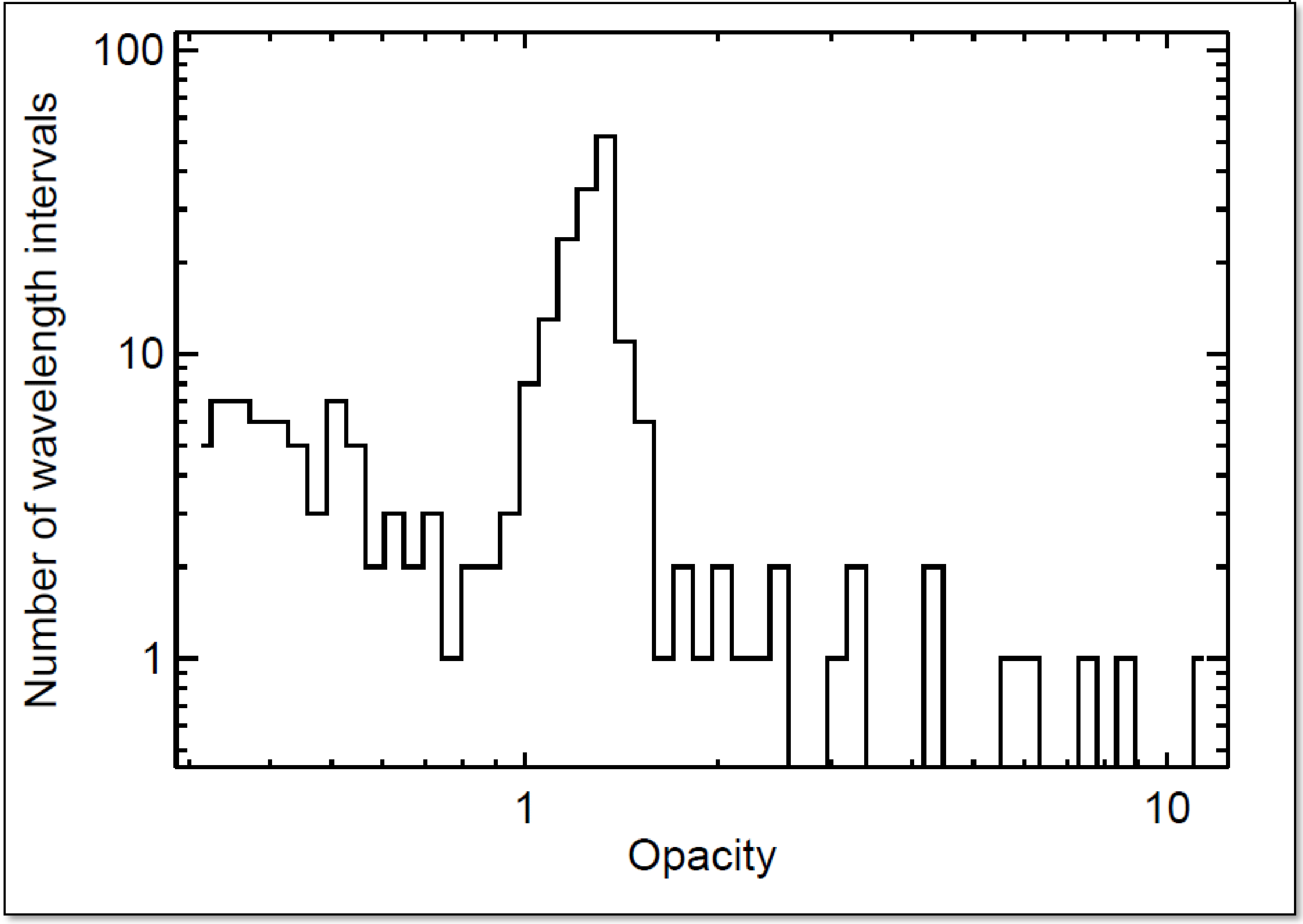
Integrated solar flux



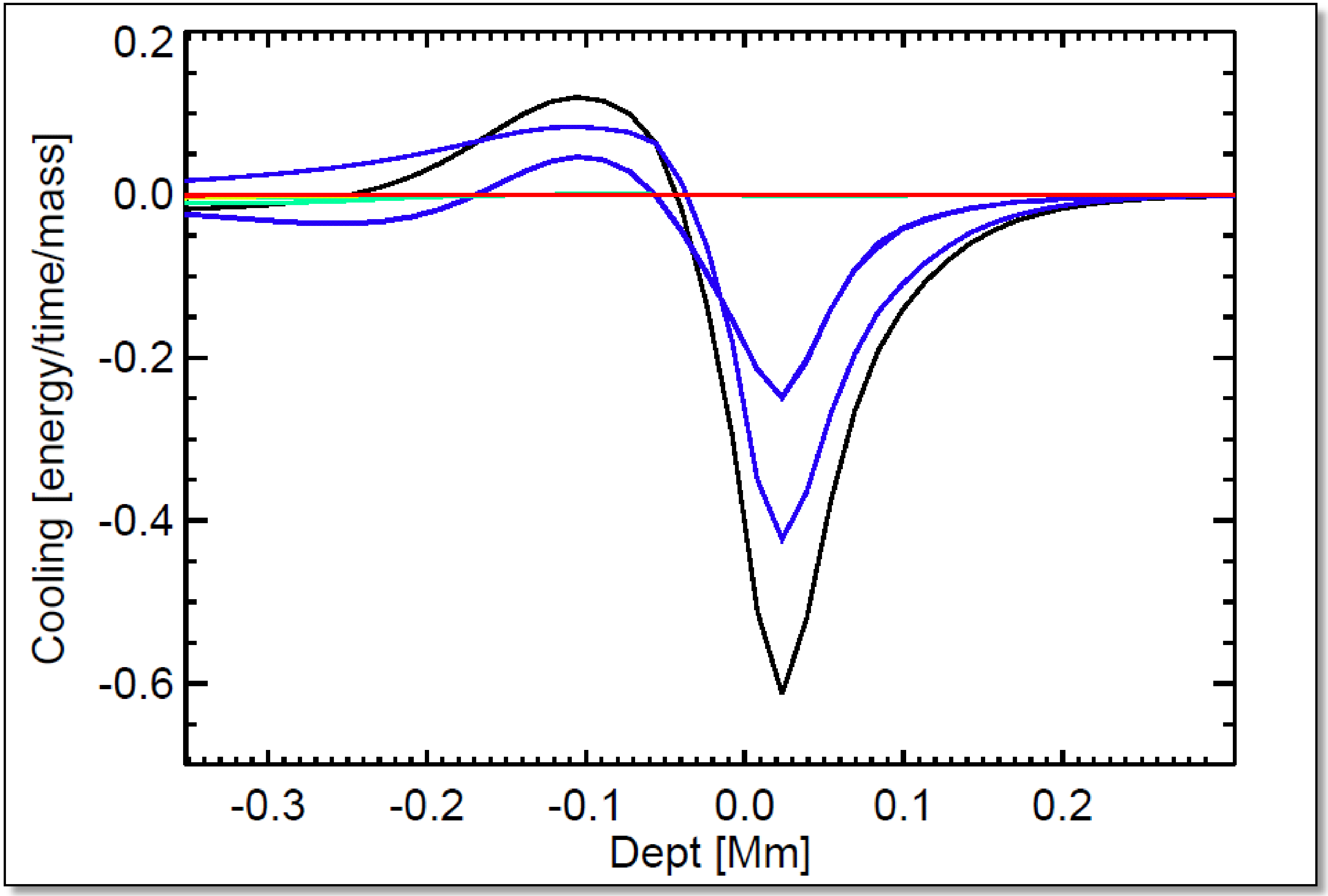
Wavelength dependencies at h=0 and h=400 km



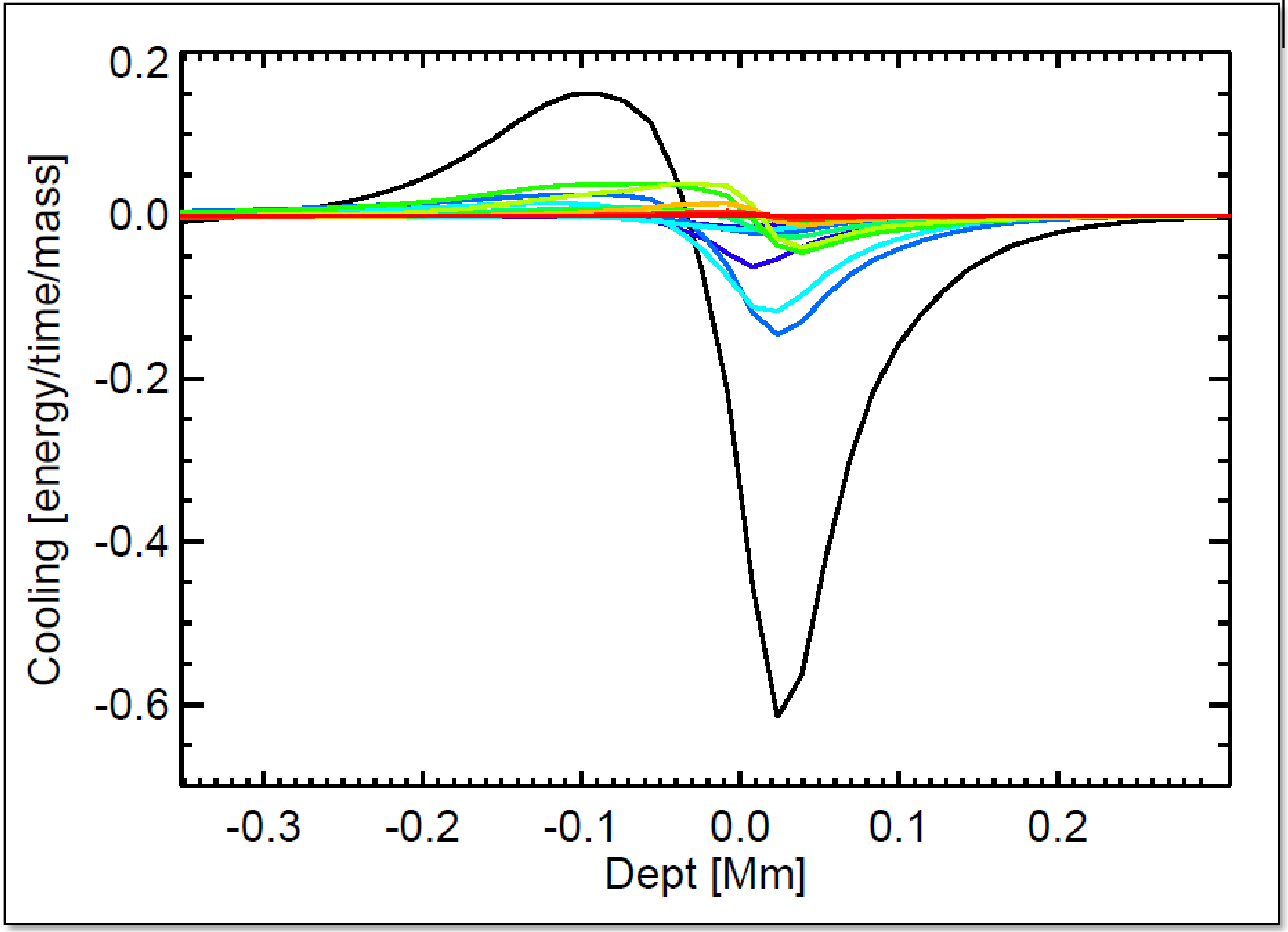
Histogram of opacities in 90% flux interval, monochr. tau = 1



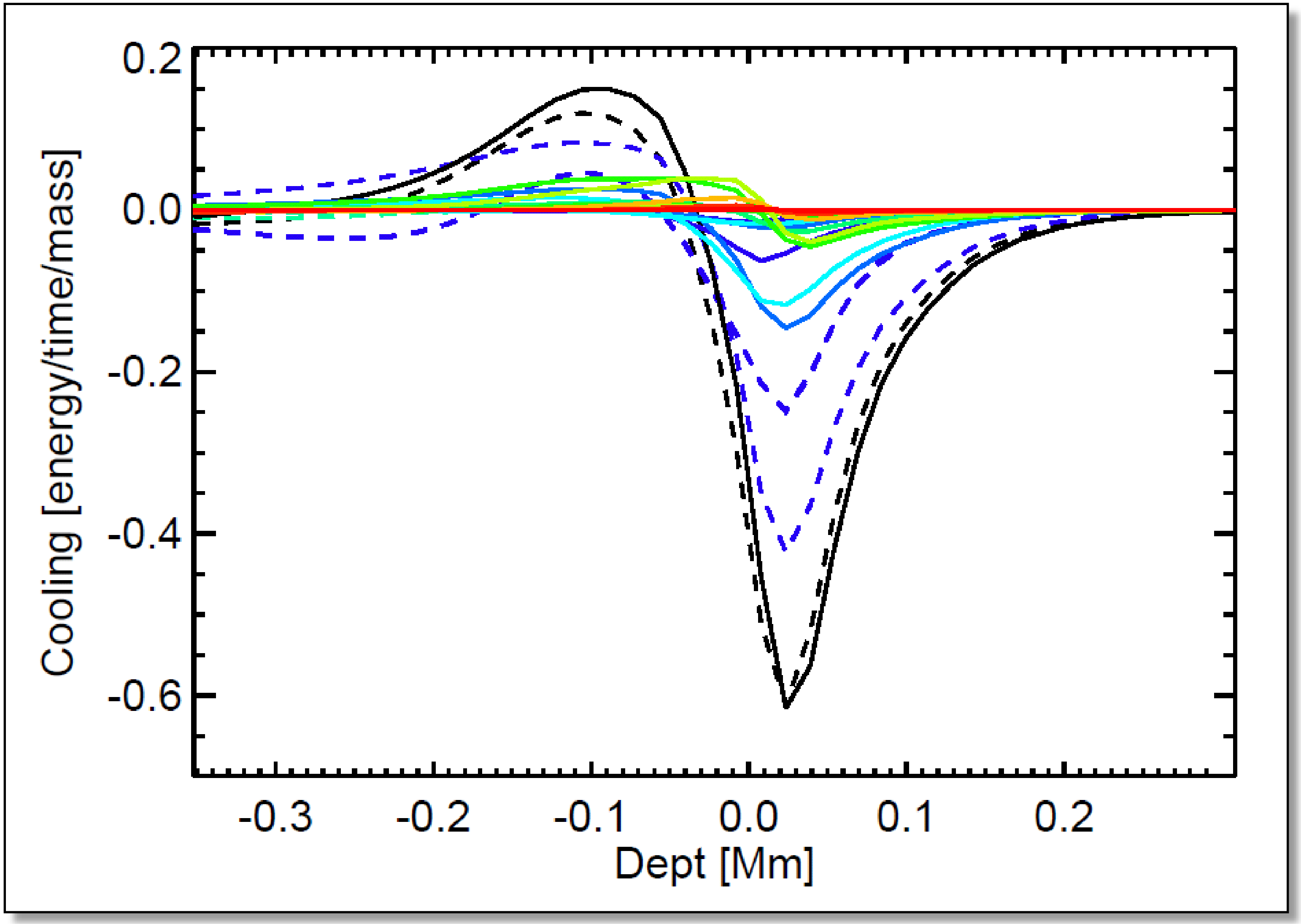
Bin contributions;
4 bins (black = sum, blue = bin1)



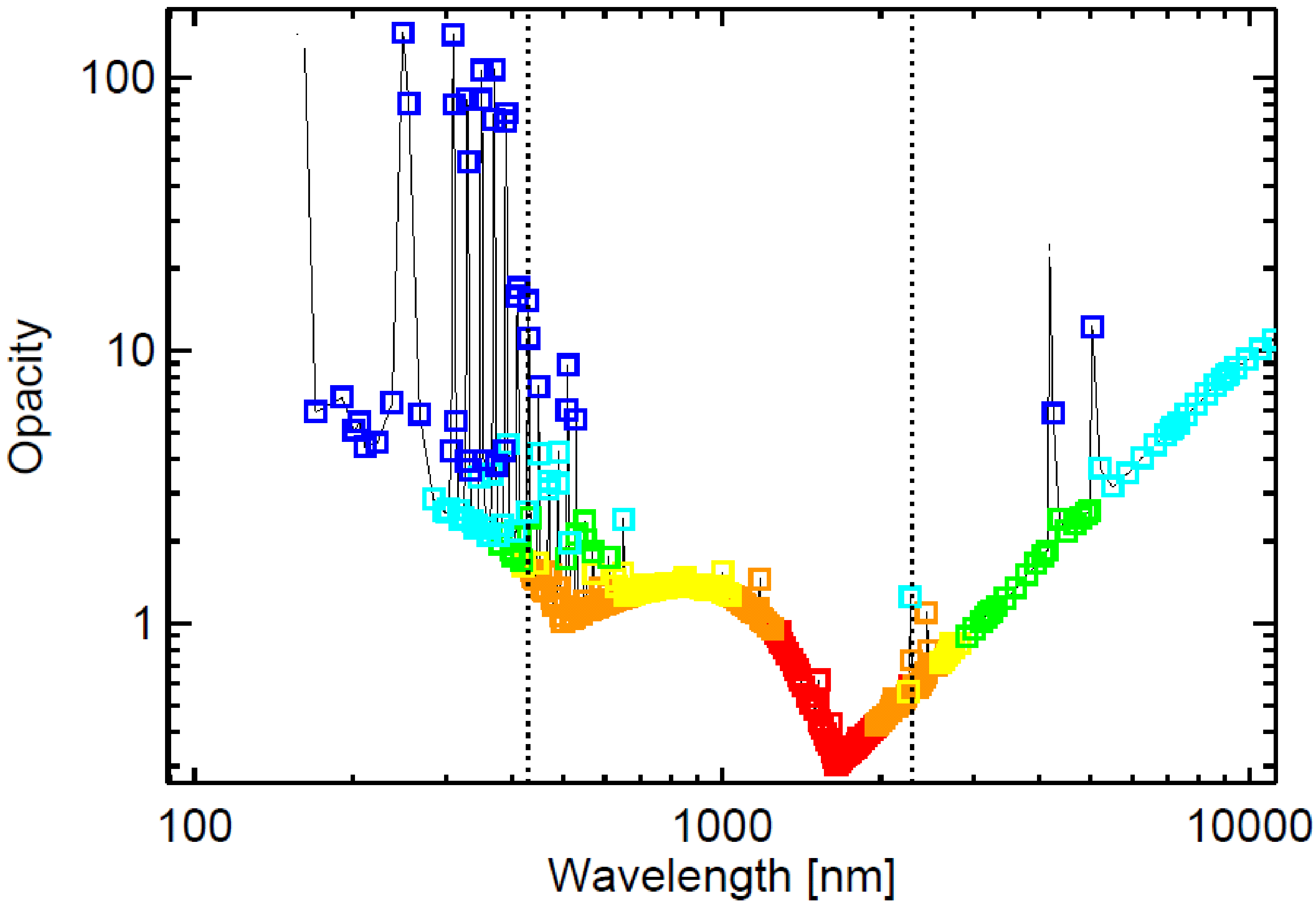
Bin contributions;
9 bins (black=sum, blue=bin1)



Bin contributions;
4 bins (dashed); 9 bins (full)



Bins as a function of wavelength



Optimizing and Parallelizing Radiative Transfer

Overview and C

- Topic: how to com
  formal problem

$$\frac{dI}{d\tau} = S - I$$

S is a completely general source function – can contain angle-scattering, frequency redistribition, etc.!

as rapidly and accurately as possible

-

The bottom line result is that, on typical current CPUs (Intel, Opteron) this should not take more than about ***5-10 nanoseconds*** per mesh point, frequency and angle!

Applications

- Ray tracing, diagnostics
- 3-D scattering and & NLTE problems
- Temperature equilbria
- Dynamical evolution with RT

Optimizing steps

- Speed up table lookup
  - Use linear interpolation
    - increased table size
  - Use cache-efficiency table order
    - increased table size again...
- Speed up interpolations
  - Use long characteristics!
    - re-use interpolation weights
- Speed up RT formal solvers
  - Use redundant indices
    - re-use exponential factors

Impact

- In any and all such problems, the faster the method ⇒ ***the more angles and frequencies!***
- Of course, computing κ, τ, and **S** takes time also..

> A reasonable goal is to be able to afford of the order of **50-100 angle-frequencies** per HD/MHD mesh point for a **doubling of the computing time**, on **problems where every mesh point is a source**!

Formal solution

$$I(\tau) = I_0\, e^{-|\tau-\tau_0|} + \int S(\tau')\, e^{-|\tau-\tau|}\, d\tau'$$

- Doubly useful:
  1. As a direct method
     - Very accurate, if S() is piecewise parabolic
     - But ***not*** the fastest method!

  2. As a basis for **domain decomposition**
     - Add 'remote' contributions separately!
     - Expensive part is ***entirely local!***

But, let's first look at direct integration

- Depending on CPU-type sometimes ”only” a factor 1.5 slower, sometimes a factor 5!
  - Accuracy comes in also; is resolution given by other factors or not?

- So, in general this method is based on ”just” integrating

$$I(\tau_{i+1}) = I(\tau_i) + \int_{\tau_i}^{\tau_{i+1}} S(\tau')\, d\tau'$$

Accurate optical depth increments

- Accurate optical depth calculation is often forgotten. Three alternatives:
  - trapezoidal (common choice – not the best!)

$$\Delta\tau_{i+½} = ½\,(\kappa_i + \kappa_{i+1})\,\Delta s$$

  - integral of exponential (better)

$$\Delta\tau_{i+½} = (\kappa_{i+1} - \kappa_i)\,/\,\ln(\kappa_{i+1}/\kappa_i)\,\Delta s$$

  - spline integral (best)

$$\Delta\tau_{i+½} = ½\left\{\kappa_{i+1}(1 + \tfrac{1}{6}d_{i+1}\Delta s) + \kappa_i(1 - \tfrac{1}{6}d_{i+1}\Delta s)\right\}\Delta s$$

A note about speed vs. accuracy on modern CPUs

- Employing more accurate – e.g. higher order – expressions is generally advantageous
  - **Cache speed vs. memory speed** $\Rightarrow$ relatively little extra time needed for extra operations
    - better load/store/compute balance
  - Improvements in precision are paid back **3-fold** (diagnostics) or **4-fold** (dynamics)

Accuracy requirements?

- For temperature bala...
  need solutions t...
  at large optical ...
  - S(τ) must be at ...
    parabolic), since the ... (τ→∞)
    are

$$I^{+}(\tau_i) = S(\tau_i) - S'(\tau_i) + S''(\tau_i) + \dots$$

$$I^{-}(\tau_i) = S(\tau_i) + S'(\tau_i) + S''(\tau_i) + \dots$$

...xt tem is
...r for
...ction,
...der in
sum!

Note that the 1st derivative term vanishes in the sum!

Lower orders are not just inaccurate but *wrong!*

- Doing, for example

$$I(\tau_{i+1}) = I(\tau_i)(1 - e^{-\Delta\tau}) + S(\tau_{i+½})e^{-\Delta\tau}$$

would lead to incorrect heating/cooling – catastrophically bad in optically thick regions!

Dual direction methods (1)

- The transfer equation for $\mathbf{I^+(\tau)}$, the intensity in the direction of increasing optical depth, is

$$\frac{dI^+}{d\tau} = S - I^+$$

and in the oppostite direction (still expressed in the same τ) it is

$$\frac{dI^-}{d\tau} = I^- - S$$

Why dual directions?

- Saves a factor of two in computing common factors!

Using the net intensity $Q = I - S$

It is almost always the ***difference between the intensity and the source function***, *e.g.* $\boldsymbol{Q^+ = I^+ - S}$ that we need. It obeys

$$\frac{dQ^+}{d\tau} = -\frac{dS}{d\tau} - Q^+$$

while for the net intensity $\boldsymbol{Q^-}$ in the opposite direction,

$$\frac{dQ^-}{d\tau} = Q^- - \frac{dS}{d\tau}$$

Why net intensity?

- Since $S'(\tau)$ and $S''(\tau)$ are anyway needed, nothing is lost, but **precision is maintained** at arbitrarily large $\Delta\tau$

Formal solutions: Integral method 2?

For a locally quadratic a…
at some discrete point τ_i

$$Q^{+}(\tau_i) = Q^{+}(\tau_{i-1})\, e^{-\Delta\tau}$$
$$+ S'(\tau_i)(1 - e^{-\Delta\tau}) \ldots$$

and, in the opposite direction,

$$Q^{-}(\tau_i) = Q^{-}(\tau_{i+1})\, e^{-\Delta\tau}$$

~~$-S'(\tau_i)(1 - e^{-\Delta\tau}) + S''(\tau_i)[(1 - e^{-\Delta\tau}) - \Delta\tau\, e^{-\Delta\tau}]$~~

Note the asymtotic behavior as $\Delta\tau \rightarrow \infty$!

Note also that this goes seemlessly to the standard case S=0!

Direct solution, integral form

- Three coefficients needed (may be reused!)

```
!-----------------------
!   Coefficients, dire
!-----------------------
    do iy=2,my
      ds(iy) = (s(iy)
```

```
!------------------------------------------------------------
!  Coefficients, IF method; (1e) +1a+2m
!------------------------------------------------------------
   do iy=2,my
     ds(iy) = (s(iy)-s(iy-1))/dtau(iy)
     if (dtau(iy) > 15.) then
       exp0 = 0.
```

```
!
!------------------------------------------------------------------------
!  Incoming rays (2a+3m)
!------------------------------------------------------------------------
  qminus(1) = Q0
  do iy=2,my
    qminus(iy) = qminus(iy-1)*ex0(iy)-dsdtau(iy)*ex1(iy)+d2sdtau2(iy)*ex2(iy)
  end do
!
!------------------------------------------------------------------------
!  Outgoing rays (2a+3m)
!------------------------------------------------------------------------
  qplus(my) = Q1
  do iy=my-1,1,-1
    qplus(iy) = qplus(iy+1)*ex0(iy+1)+dsdtau(iy)*ex1(iy+1)+d2sdtau2(iy)*ex2(iy+1)
    q(iy) = 0.5*(qplus(iy)+qminus(iy))
  end do
  q(my) = 0.5*(qplus(my)+qminus(my))
```

The Feautrier Method (this goes way back in RT!)

Defining

$$P = (I^{+} + I^{-})/2$$

and

$$R = (I^{-} - I^{+})/2$$

we have

$$\frac{dP}{d\tau} = R$$

and

$$\frac{dR}{d\tau} = P - S \ ,$$

which leads to

$$\frac{d^2 P}{d\tau^2} = P - S \ .$$

The Feautrier Equation for net radiative heating

At large optical depths the Feautrier formulation becomes susceptible to numerical round-off, since P approaches S. If one desires the difference P-S, e.g. to compute the heating / cooling due to radiation, it is advantageous to write the equation directly in terms of the difference $Q=P-S$, for which the RTE becomes

$$\frac{d^2Q}{d\tau^2} = Q - \frac{d^2S}{d\tau^2} .$$

At large optical depths, where Q is small, this goes to the diffusion approximation,

$$Q = \frac{d^2S}{d\tau^2} ,$$

without round-off error.

Speeds of integral and Feautrier methods (dual dir!)

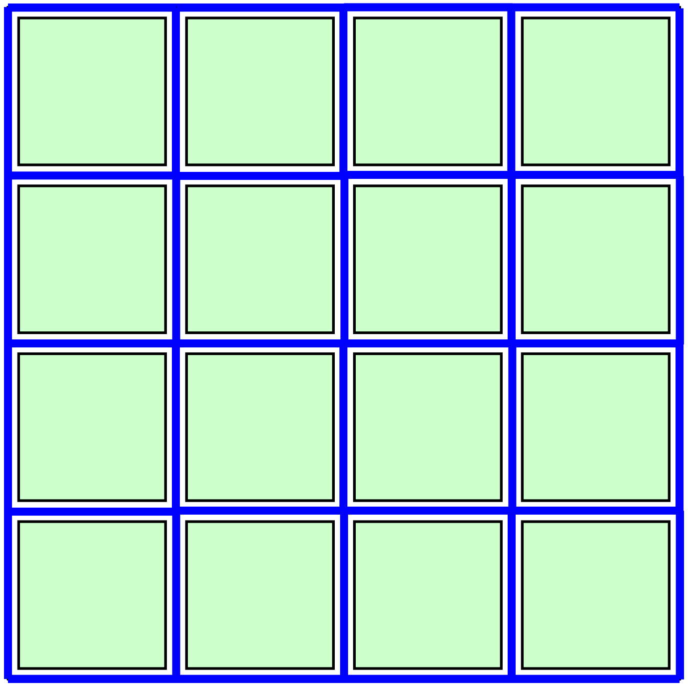
CPU type	Integral method	Feautrier method
Opteron/Intel recursive	25 ns/pt	30 ns/pt
Opteron/Intel vector (SSE)	14 ns/pt	9 ns/pt
Columbia recursive	68 ns/pt	32 ns/pt
Columbia vector	53 ns/pt	12 ns/pt

How to parallelize (Heinemann, et al. 2005; Rijkhorst et al 2005)

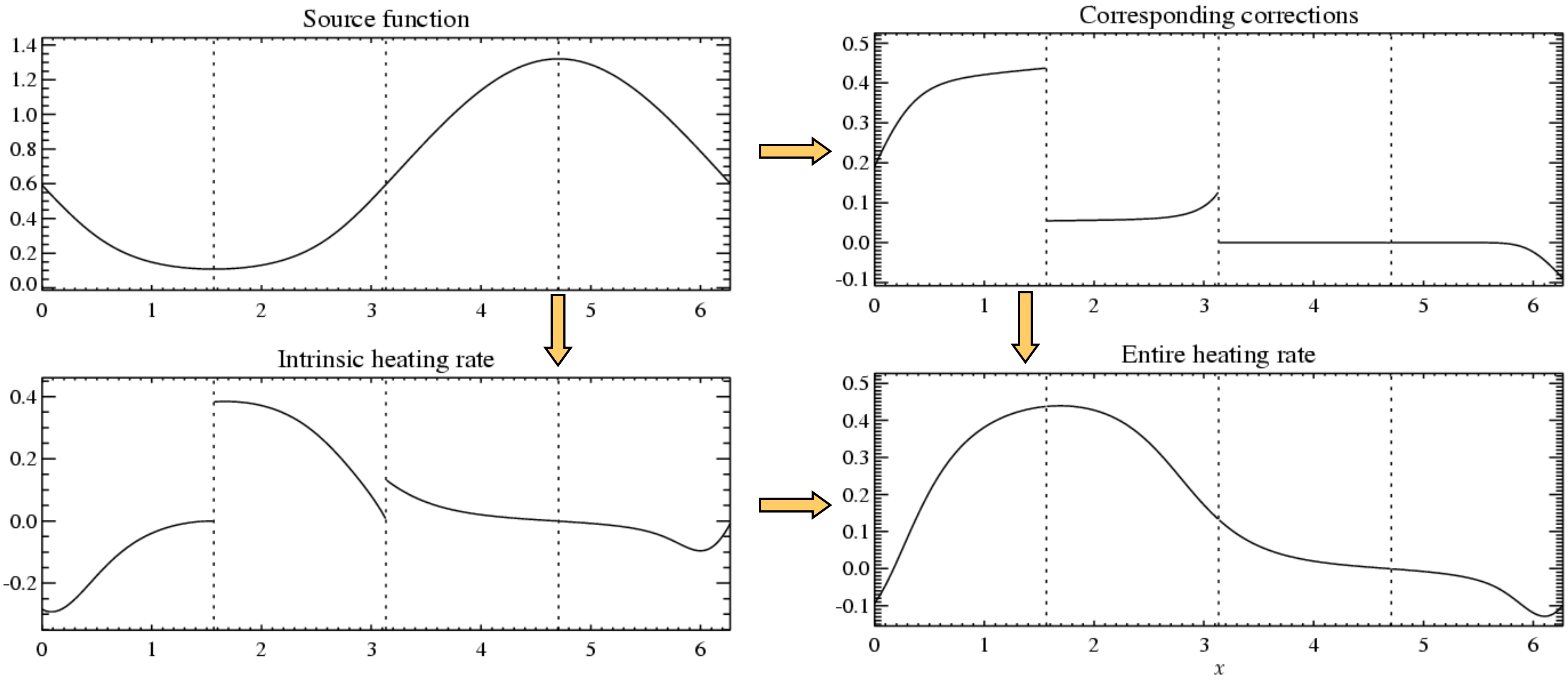
- Solve for the intensity generated internally in each domain, separately and in parallel
- Then propagate and accumulate the boundary intensities, modified only by trivial optical depth factors

About Node Wave Fronts and other parallelization aspects

- In cases where RT is handled separately it is NOT necessary to use node wave fronts – local contribution can then be scattered / gathered globally
  - Cf. Rijkhorst et al 2005
  - Heinemann et al 2005



Putting it together

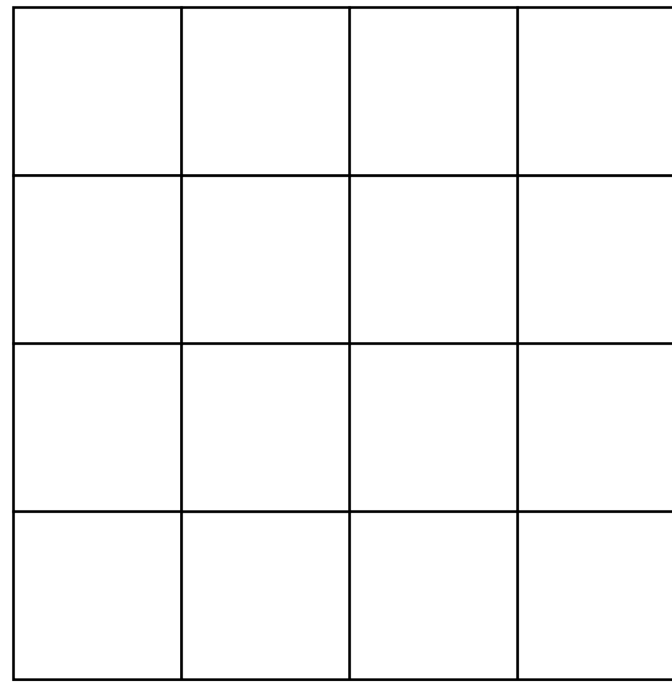


The Transfer Equation & Parallelization

Analytic Solution:

$$I(\tau) = I_0 e^{\tau_0 - \tau} + \int_{\tau_0}^{\tau} e^{\tau' - \tau} S(\tau') d\tau'$$

Processors



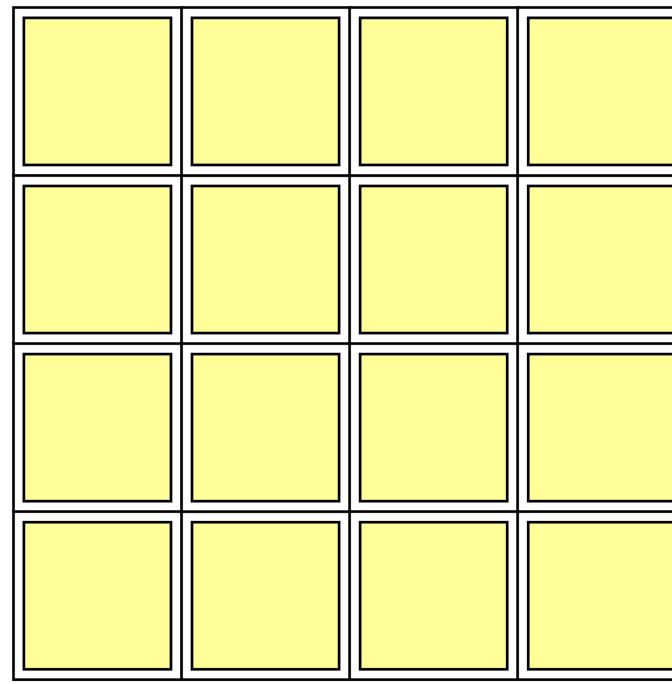
The Transfer Equation & Parallelization

Analytic Solution:

$$I(\tau) = I_0 e^{\tau_0 - \tau} + \int_{\tau_0}^{\tau} e^{\tau' - \tau} S(\tau') d\tau'$$

Intrinsic Calculation

Processors



Ray direction

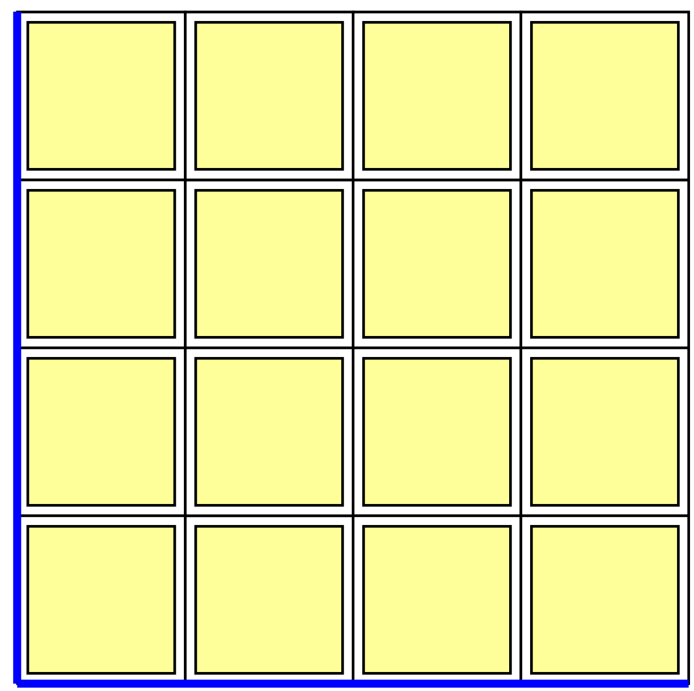
The Transfer Equation & Parallelization

Analytic Solution:

$$I(\tau) = I_0 e^{\tau_0 - \tau} + \int_{\tau_0}^{\tau} e^{\tau' - \tau} S(\tau') d\tau'$$

Communication

Processors



Ray direction

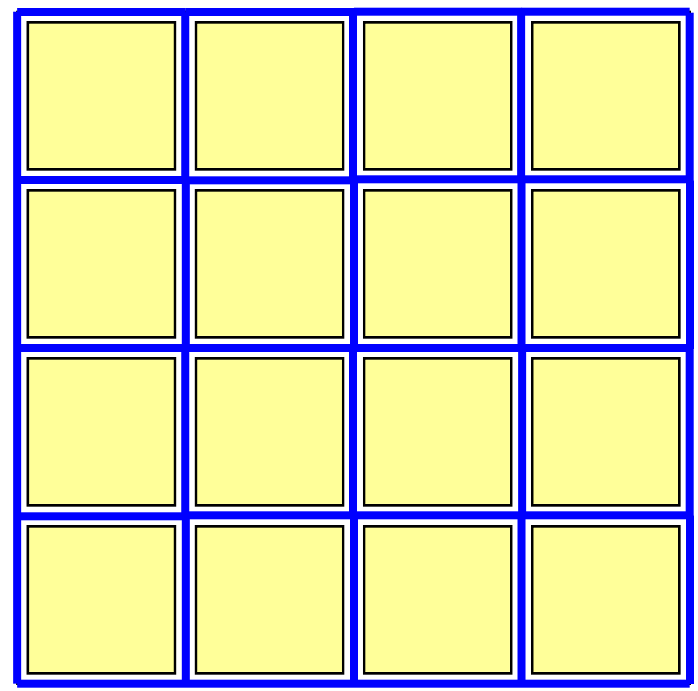
The Transfer Equation & Parallelization

Analytic Solution:

$$I(\tau) = I_0 e^{\tau_0 - \tau} + \int_{\tau_0}^{\tau} e^{\tau' - \tau} S(\tau') d\tau'$$

Communication

Processors



Ray direction

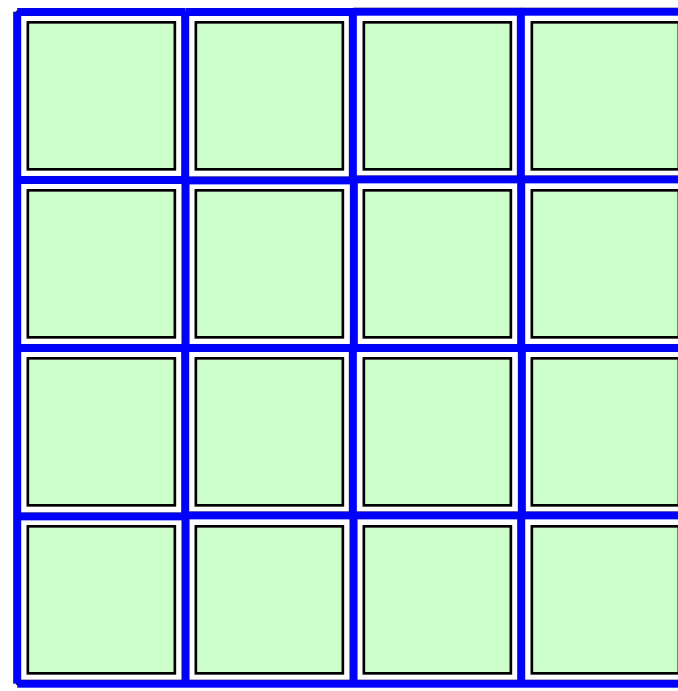
The Transfer Equation & Parallelization

Analytic Solution:

$$I(\tau) = I_0 e^{\tau_0 - \tau} + \int_{\tau_0}^{\tau} e^{\tau' - \tau} S(\tau') d\tau'$$

Intrinsic Calculation

Processors



Ray direction

Pencil code (Brandenburg et al) CPU-time per ray-point

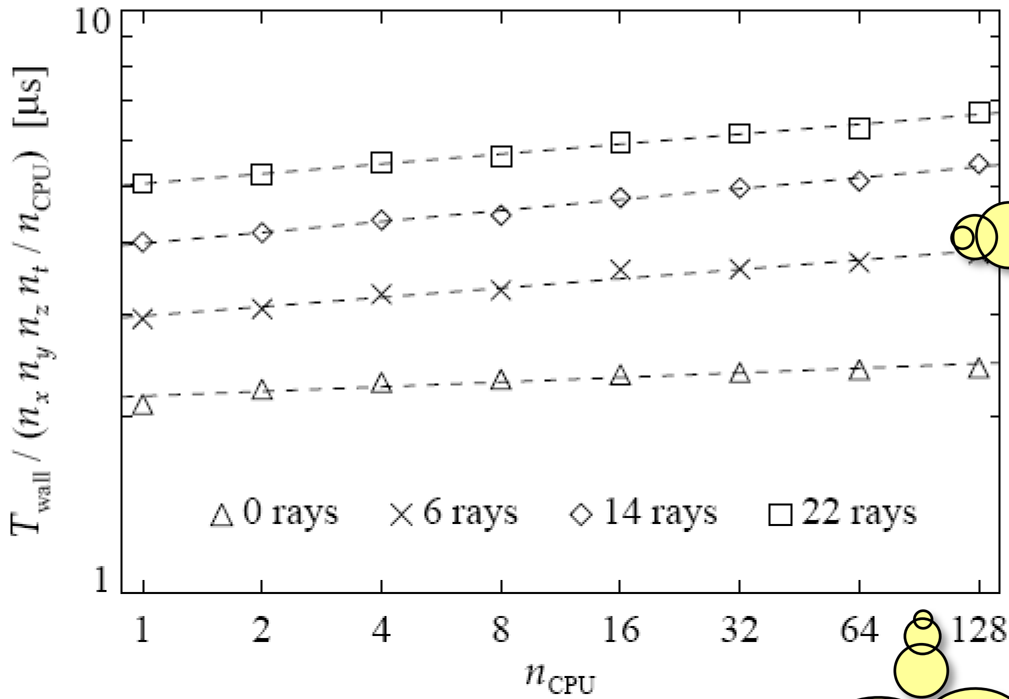


Figure 4. Wall clock time per grid point ... dimensional simulation of the solar sur... ray directions – in this case for columnar... periodic boundary conditions for the coo... lines denote least square fits. Perfect scaling would ... repre-sented by a horizontal line.

The actual RT speed has been greatly improved since!

Now scales up to 2048 cores on Pleiades at NASA/Ames

The scaling properties are great!

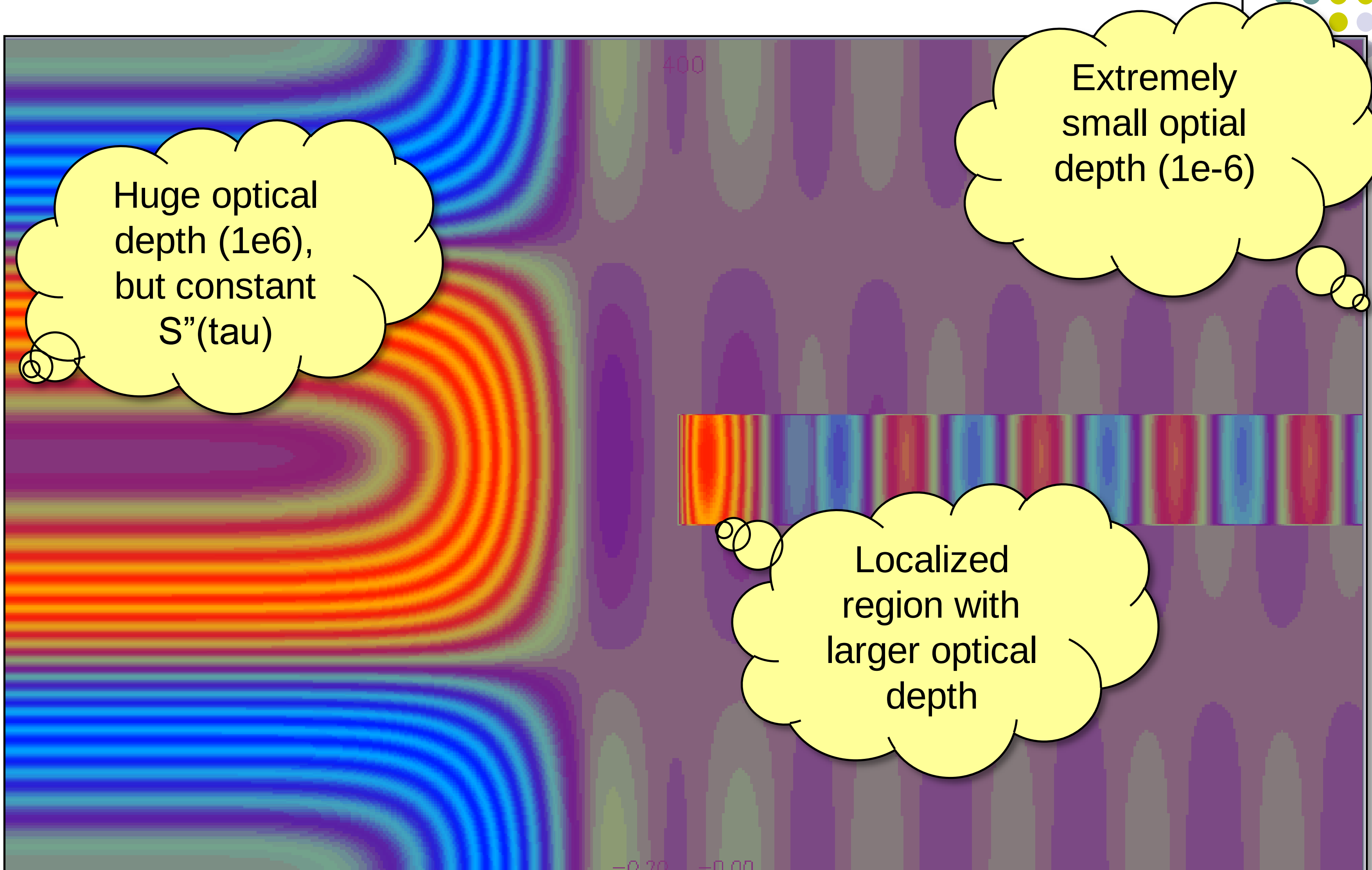
Perspective

- What we ***want*** to do is to include radiation, as an active ingredient and a diagnostic tool, whenever it is relevant – this is now becoming doable!

- In practice there is a **balance between *realism*** and ***cost***, which may be tilted towards realism by optimizing the methods!
  - maximize number of angle-frequencies!

Time-dependent Radiative Transfer

Non-trivial demo case



Summary

- RT is important / crucial in a number of astro-subfields
  - Cosmology, compact objects, molecular cloud formation star formation, planet formation, …
- RT-HD and RT-MHD is ***becoming practical*** in more or less all of these circumstances
- ***Speed of computation*** (of transfer ***and*** interpolations) is of premium importance!
  - Optimal choices depend on circumstances and also surprisingly much on hardware (CPU type)

Thanks for your attention!

Acknowledgments

The results presented here were made possible by hardware grants to ÅN from the **Danish Center for Scientific Computing**, and by Columbia / Pleiades computing resource grants to Bob Stein from **NASA / NAS**.

Visualizations partly done with **VAPOR from NCAR – see www.vapor.ucar.edu**