Stellar/Planetary Atmospheres

Part 08: 3D RT

Peter Hauschildt
yeti@hs.uni-hamburg.de

Hamburger Sternwarte
Gojenbergsweg 112
21029 Hamburg

14. März 2018

Topics

- 3D radiative transfer
  - basic idea
  - continuum transfer (scattering)
  - parallelization
  - line transfer (2-level atom)
  - periodic BCs

Framework

- Cartesian coordinates
- voxel grid ("cells")
  - physical data constant within a voxel
  - solve RT inside the voxels
- grid coordinates, e.g., $[-n_x, +n_x]$
- $\rightarrow$ voxel coordinates from $(-n_x, -n_y, -n_z)$ to $(+n_x, +n_y, +n_z)$

Data size

- small grid:
- $n_x = n_y = n_z = 32$
- $\rightarrow (2 * 32 + 1)^3 = 274\,625$ voxels
- compare to typically 64 to 128 grid points in 1D!
- large grid:
- $n_x = n_y = n_z = 128$
- $\rightarrow (2 * 128 + 1)^3 = 16\,974\,593$ voxels
- $\rightarrow$ one scalar physical variable (T) $\rightarrow$ 129 MB

Framework

- solution through operator splitting

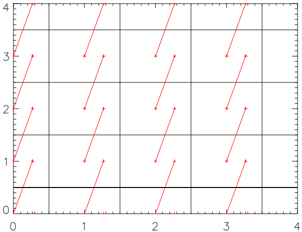
$$[1 - \Lambda^*(1 - \epsilon)]\,\bar{J}_{\mathrm{new}} = \bar{J}_{\mathrm{fs}} - \Lambda^*(1 - \epsilon)\bar{J}_{\mathrm{old}},$$

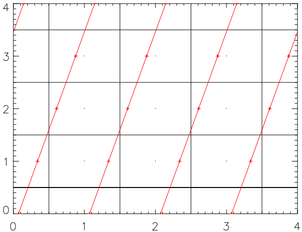
  where $\bar{J}_{\mathrm{fs}} = \Lambda S_{\mathrm{old}}$.
- need method to perform FS
- need method to construct Λ^*

Formal Solution

- basically 'any' method could be used
  - short-characteristics
  - full-characteristics ("ray-tracing")
- parameterized by direction (θ, ϕ)
- set of charactistics covering the border
- tracked through the voxel grid
- until they leave through exit border
- cover solid angles $\Omega = (\theta, \phi)$ to compute J etc

characteristics





Formal Solution

- along characteristic $\rightarrow$
- piecewise parabolic interpolation & integration (PPM)
- piecewise linear interpolation & integration (PLM)
- along a characteristic the RTE is

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

Formal Solution

- $\rightarrow$ discretized formal solution

$$
\begin{aligned}
I(\tau_i) &= I(\tau_{i-1}) \exp(\tau_{i-1} - \tau_i) \\
&\quad + \int_{\tau_{i-1}}^{\tau_i} S(\tau) \exp(\tau - \tau_i)\, d\tau \\
I(\tau_i) &\equiv I_{i-1} \exp(-\Delta\tau_{i-1}) + \Delta I_i
\end{aligned}
$$

- i labels the points (steps) along a characteristic
- τ_i: optical depth along a characteristic

$$
\Delta\tau_{i-1} = (\chi_{i-1} + \chi_i)|s_{i-1} - s_i|/2
$$

Formal Solution

- Source function is interpolated along i:

$$\Delta I_i = \alpha_i S_{i-1} + \beta_i S_i + \gamma_i S_{i+1}$$

Formal Solution

- parabolic interpolation

$$
\begin{aligned}
\alpha_i &= e_{0i} + [e_{2i} - (\Delta\tau_i + 2\Delta\tau_{i-1})e_{1i}] \\
&\qquad /[\Delta\tau_{i-1}(\Delta\tau_i + \Delta\tau_{i-1})] \\
\beta_i &= [(\Delta\tau_i + \Delta\tau_{i-1})e_{1i} - e_{2i}]/[\Delta\tau_{i-1}\Delta\tau_i] \\
\gamma_i &= [e_{2i} - \Delta\tau_{i-1}e_{1i}]/[\Delta\tau_i(\Delta\tau_i + \Delta\tau_{i-1})]
\end{aligned}
$$

Formal Solution

- linear interpolation

$$\begin{aligned}\alpha_i &= e_{0i} - e_{1i}/\Delta\tau_{i-1} \\ \beta_i &= e_{1i}/\Delta\tau_{i-1} \\ \gamma_i &= 0\end{aligned}$$

- with

$$\begin{aligned}e_{0i} &= 1 - \exp(-\Delta\tau_{i-1}) \\ e_{1i} &= \Delta\tau_{i-1} - e_{0i} \\ e_{2i} &= (\Delta\tau_{i-1})^2 - 2e_{1i}\end{aligned}$$

- and $\Delta\tau_i \equiv \tau_{i+1} - \tau_i$

Formal Solution

- integration over Ω:
  - trapez or Simpson quadrature
  - Monte-Carlo integration:

$$J = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} I \sin\theta \, d\theta d\phi$$

  - replaced by MC sum:

$$J \approx \frac{1}{2\pi^2} \sum I \sin\theta$$

Formal Solution

- sum over all solid angle points (θ, ϕ)
- (θ, ϕ) randomly selected, equal weight
- also works for precribed (θ, ϕ) grids
- advanced: use Sobol sequence to cover solid angle better

Computation of Λ^*

- α, β, and γ $\rightarrow$
- construct diagonal and tri-diagonal Λ^* in 1D
- non-local Λ^* operators $\rightarrow$ excellent convergence
- highly desirable to implement non-local Λ^* in 3D!

Computation of Λ^*

- tri-diagonal operator in the 1D case
- $\rightarrow$ nearest neighbor Λ^* in 3D
- 1D: considers interaction of point with two direct neighbors
- 3D: interaction of a voxel with the $3^3 - 1 = 26$ surrounding voxels
- (or 6 neighbors in strictly face-centered view)
- 3D: storage requirements large but worth it!

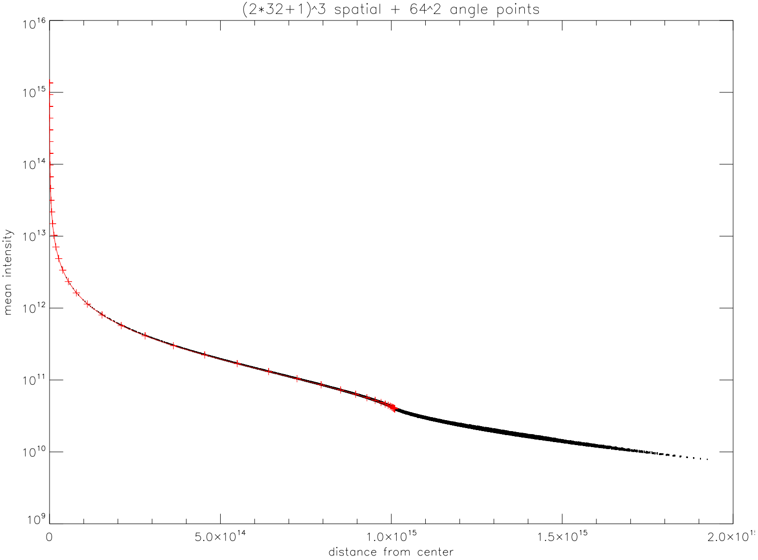
Operator splitting step

- must solve linear system for non-local Λ^*

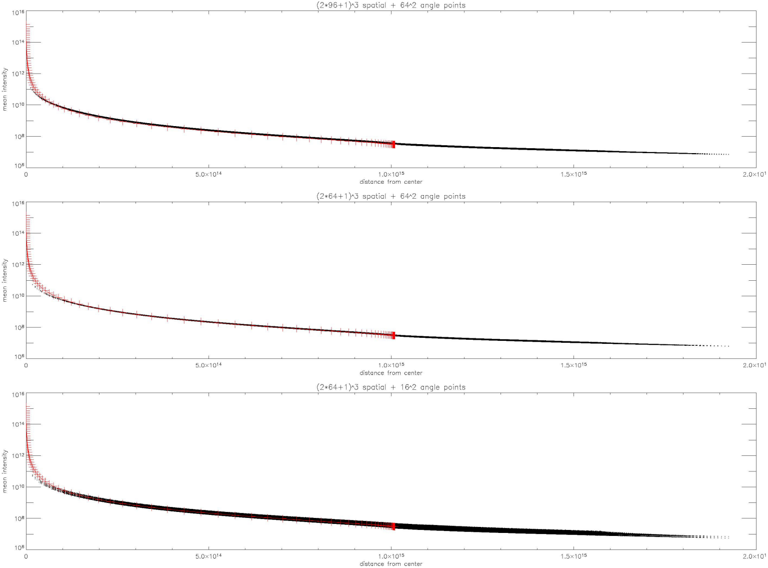
$$[1 - \Lambda^*(1-\epsilon)]\,\bar{J}_{\mathrm{new}} = \bar{J}_{\mathrm{fs}} - \Lambda^*(1-\epsilon)\bar{J}_{\mathrm{old}}$$

- for grids with $n_x = n_y = n_z > 16$ this cannot be done directly
- use iterative solvers
  - Jordan
  - Gauss-Seidel
- can use additional convergence acceleration

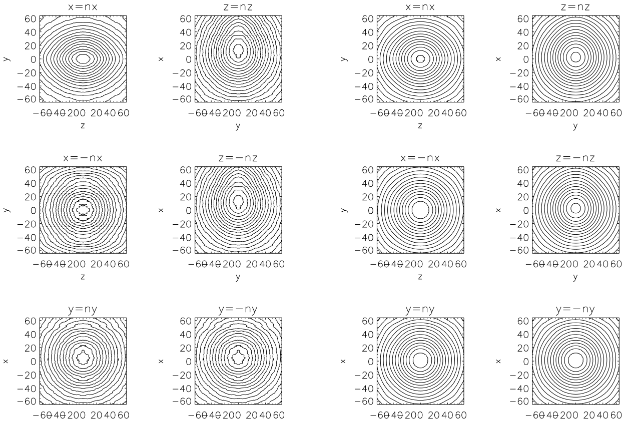
results: LTE


(2*32+1)^3 spatial + 64^2 angle points
mean intensity
10^{16}
10^{15}
10^{14}
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
0
5.0×10^{14}
1.0×10^{15}
1.5×10^{15}
distance from center

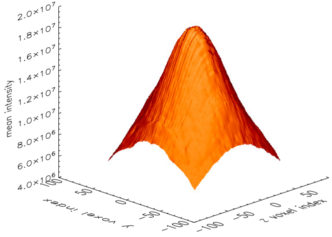
results: $\epsilon = 10^{-4}$

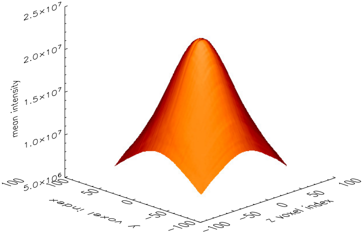


results: $\epsilon = 10^{-4}$

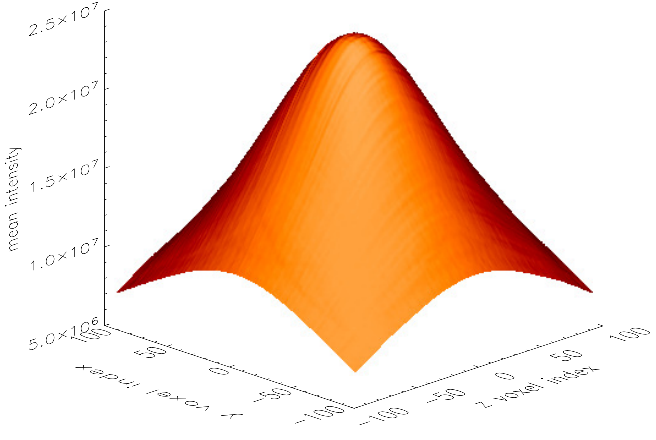


results: $\epsilon = 10^{-4}$

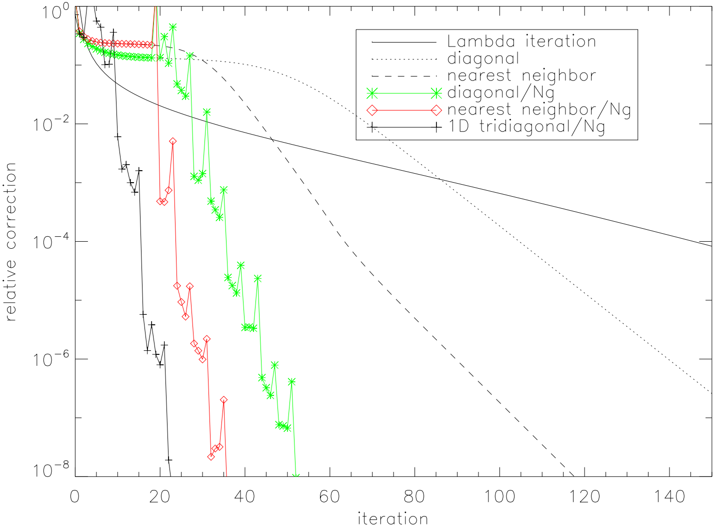




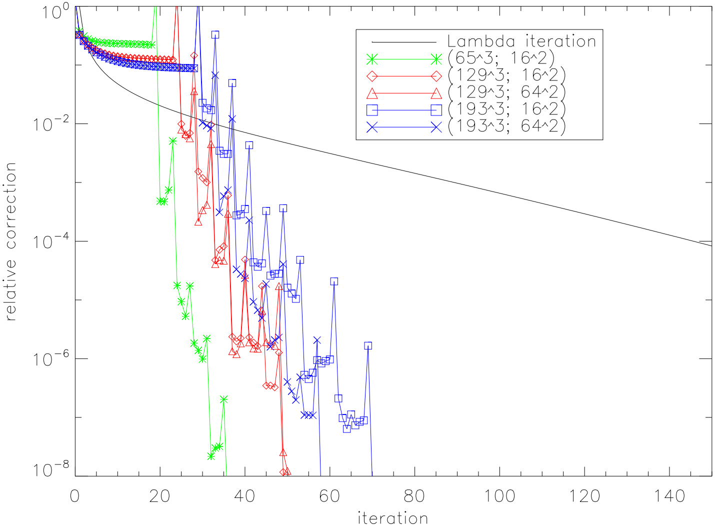
results: $\epsilon = 10^{-4}$



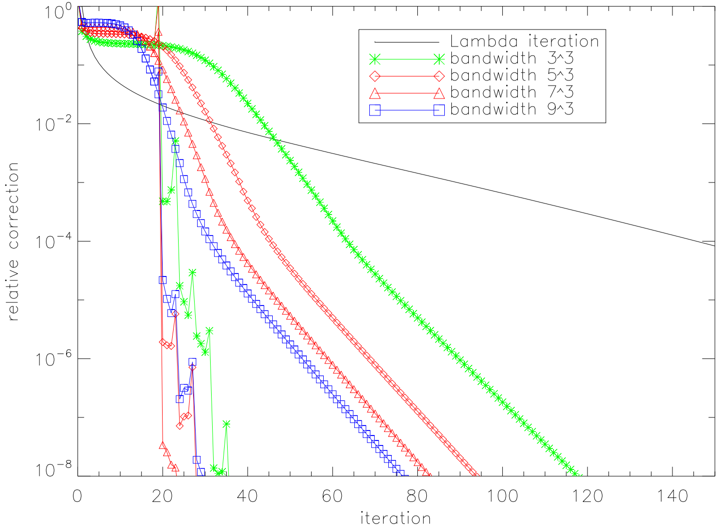
results: convergence, $\epsilon = 10^{-4}$



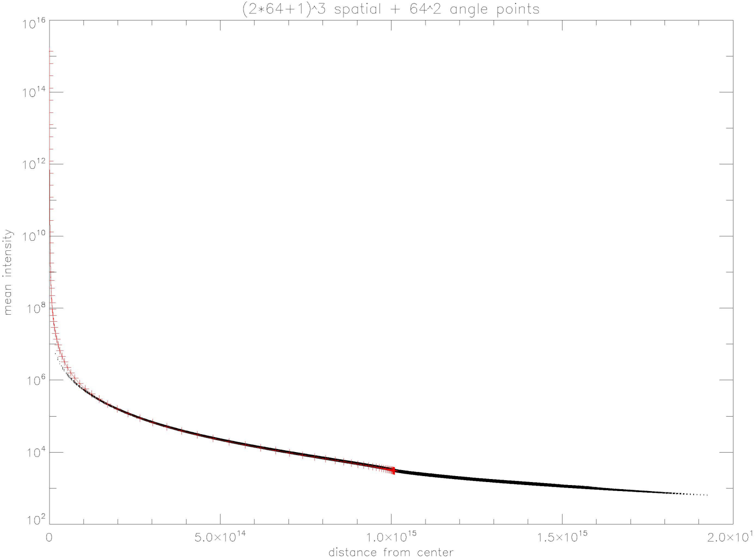
results: convergence, $\epsilon = 10^{-4}$



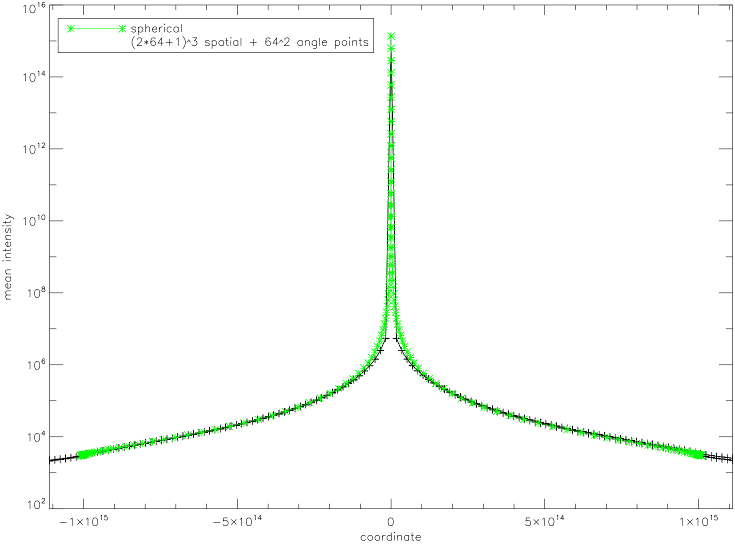
results: convergence, $\epsilon = 10^{-4}$



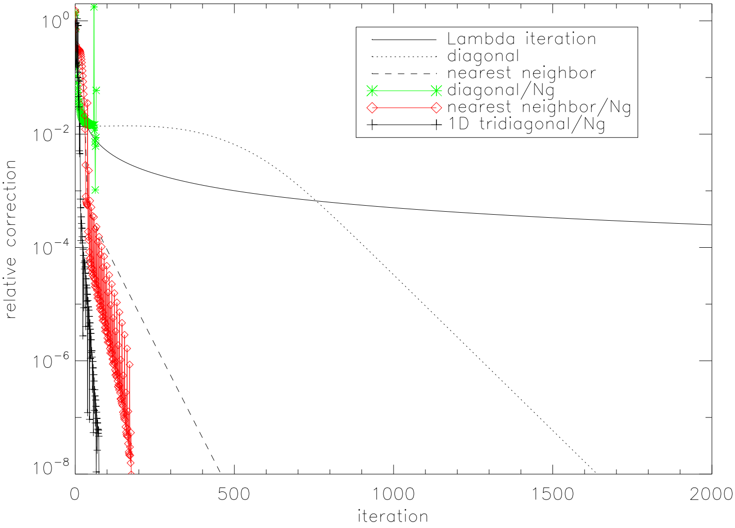
results: $\epsilon = 10^{-8}$



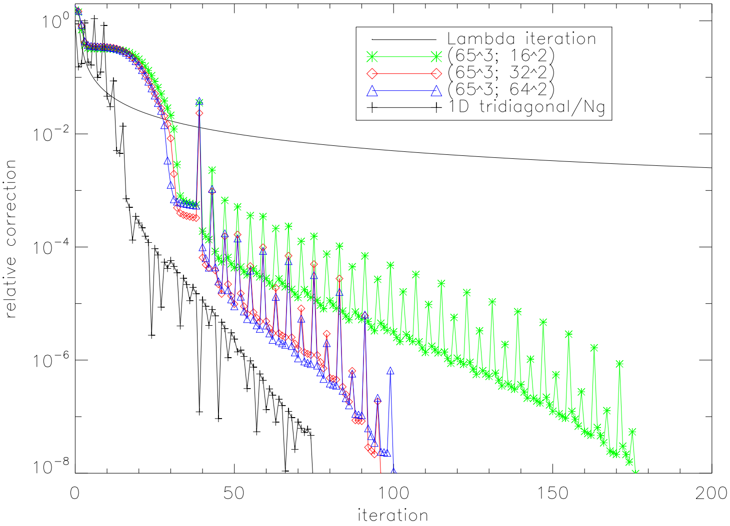
results: $\epsilon = 10^{-8}$



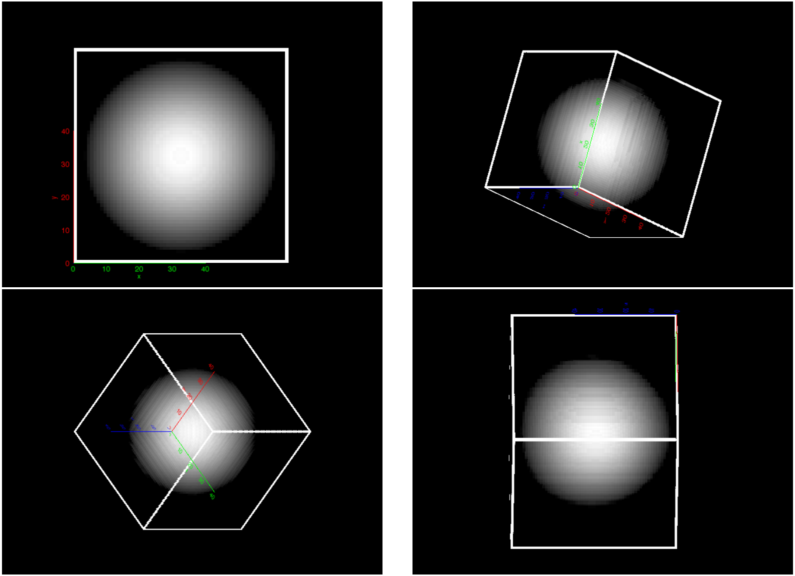
results: $\epsilon = 10^{-8}$



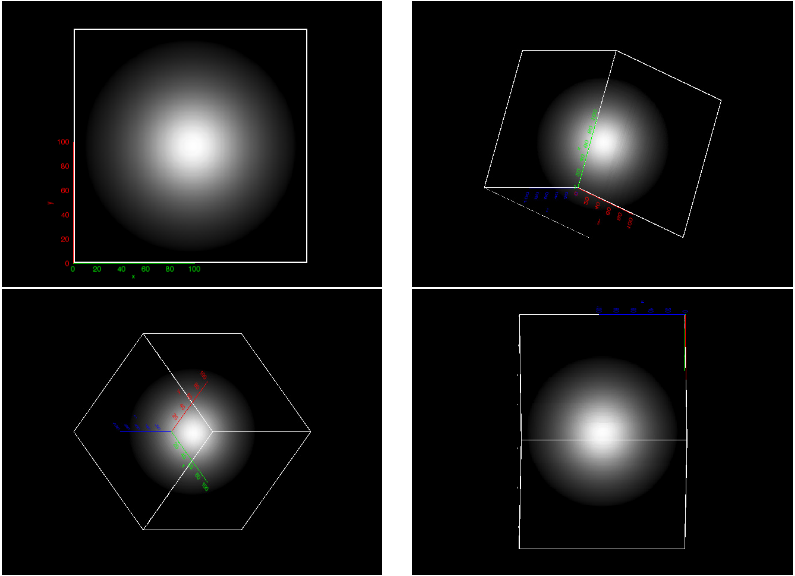
results: convergence, $\epsilon = 10^{-8}$



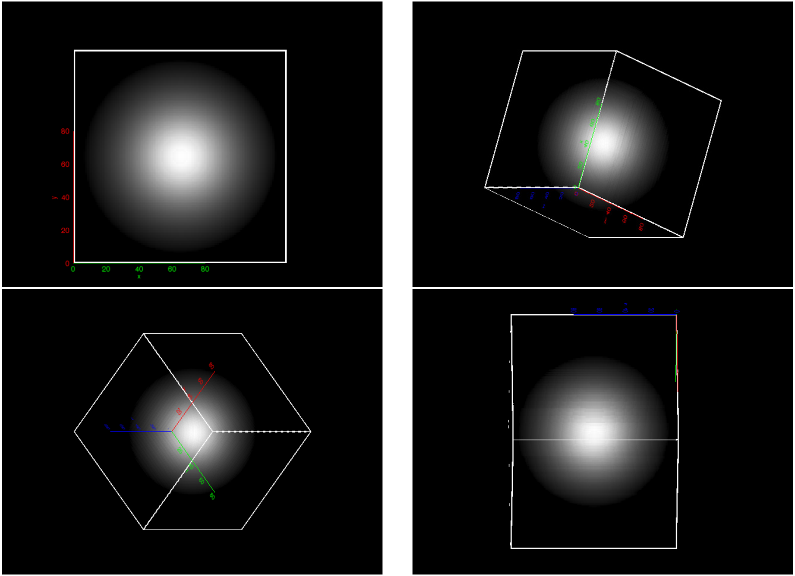
visualization: LTE



visualization: $\epsilon = 10^{-4}$



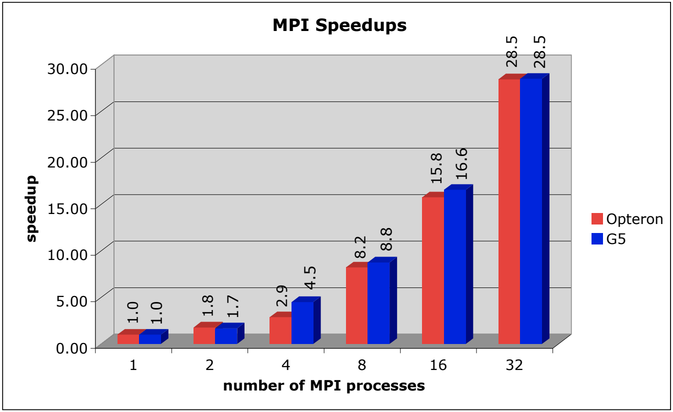
visualization: $\epsilon = 10^{-8}$



Parallelization

- many possible approaches
- simplest: all solid angle points can be done in parallel
- needs communication only at the very end
- SMP parallelization: different characteristics for single Ω

results: parallelization



line transfer

- same approach as in 1D
- uses 3D framework for wavelength dependent calculations
- then just compute

$$\bar{J} = \int \phi(\lambda) J_\lambda \, d\lambda$$

and

$$\bar{\Lambda}^* = \int \phi(\lambda) \Lambda^* \, d\lambda.$$

line transfer

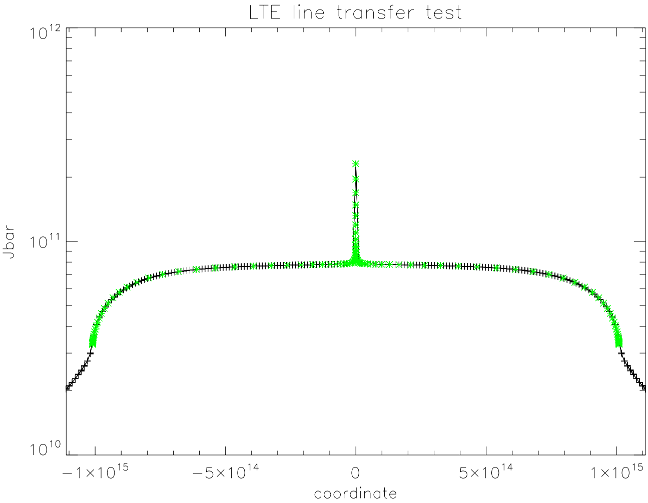
- operator splitting step for the line →

$$\left[1 - \bar{\Lambda}^*(1-\epsilon)\right] \bar{J}_{\rm new} = \bar{J}_{\rm fs} - \bar{\Lambda}^*(1-\epsilon)\bar{J}_{\rm old}$$

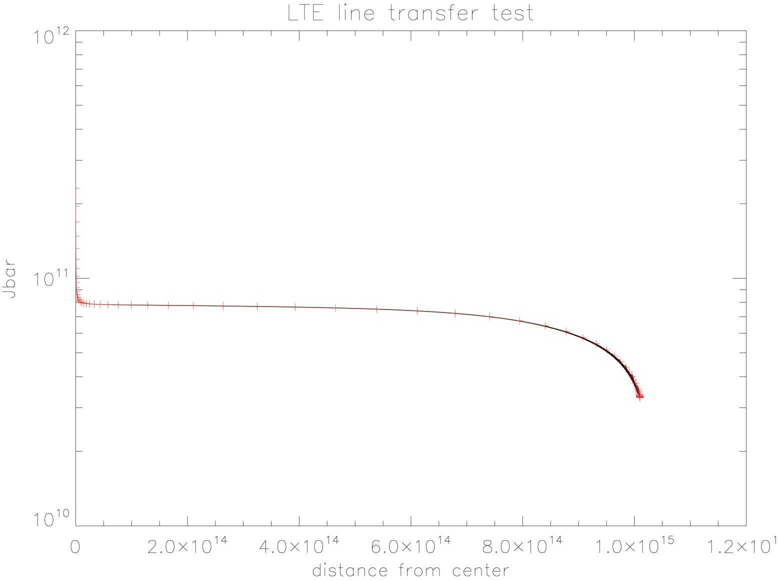
line test

- line parameterized
- $\chi_l/\chi_c = 10^6$: strong line
- Gauss profile
- 32 wavelength points

results: LTE

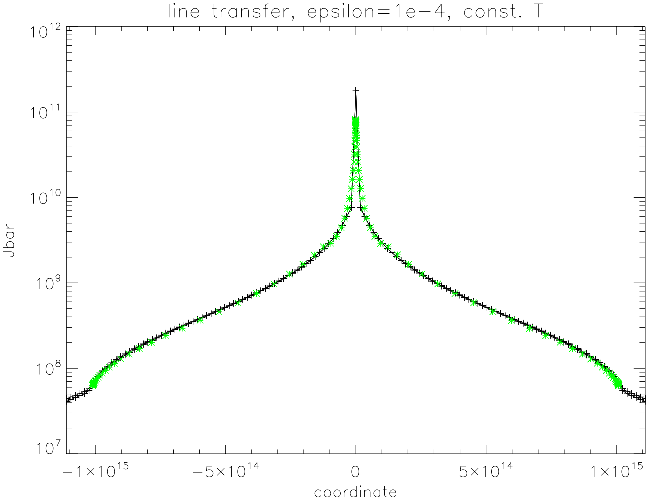

LTE line transfer test
10^{12}
10^{11}
10^{10}
Jbar
-1×10^{15}
-5×10^{14}
0
5×10^{14}
1×10^{15}
coordinate

results: LTE

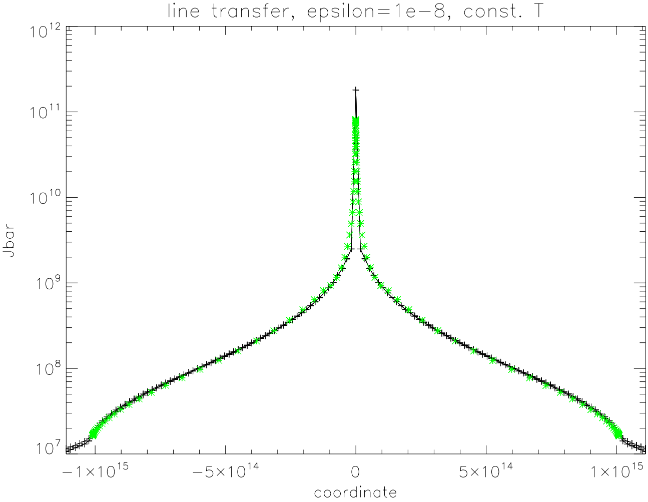


LTE line transfer test
10^{12}
10^{11}
10^{10}
Jbar
0
2.0×10^{14}
4.0×10^{14}
6.0×10^{14}
8.0×10^{14}
1.0×10^{15}
1.2×10^{1}
distance from center

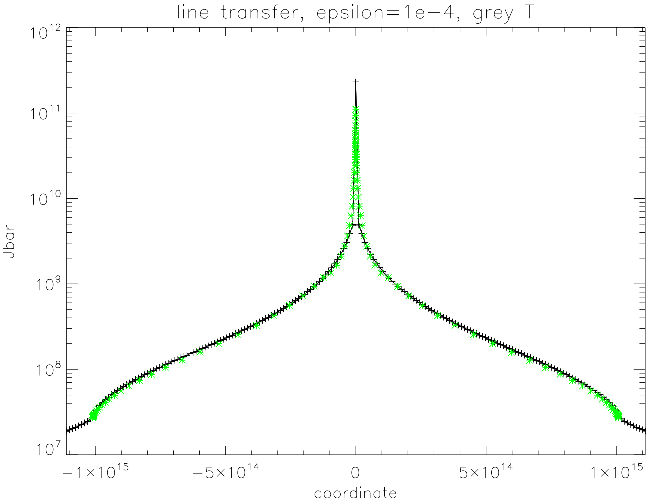
results: $\epsilon_l = 10^{-4}$



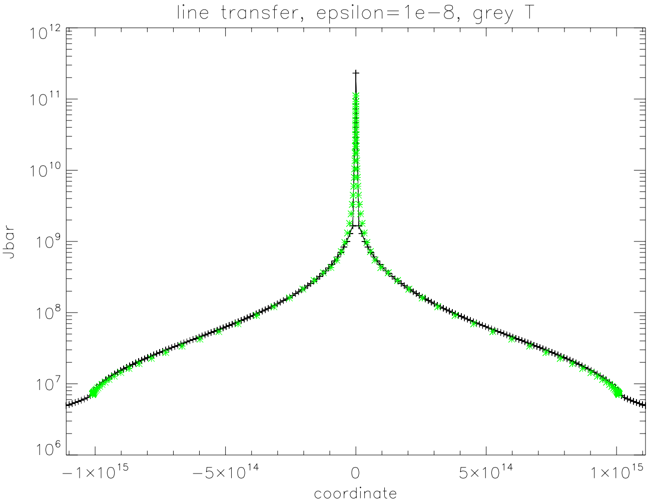
results: $\epsilon_l = 10^{-8}$



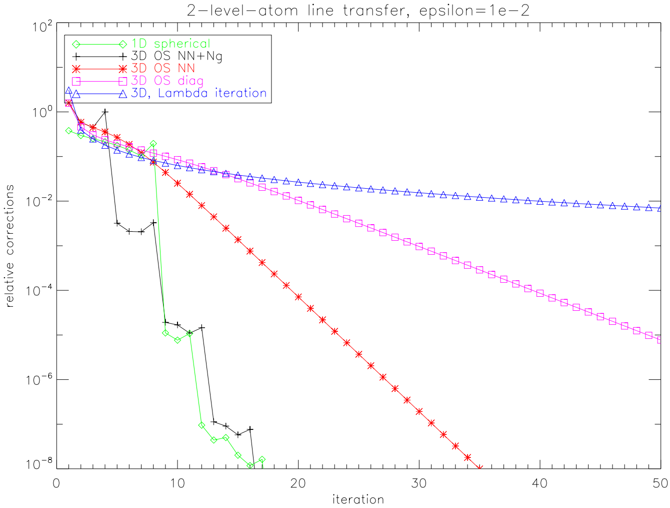
results: $\epsilon_l = 10^{-4}$



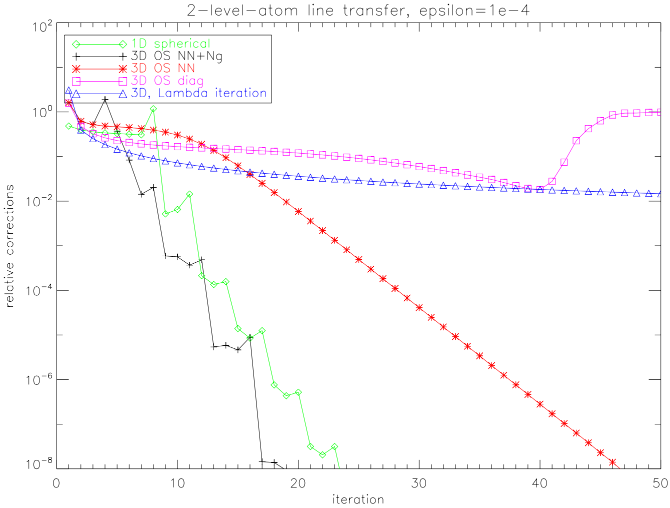
results: $\epsilon_l = 10^{-8}$



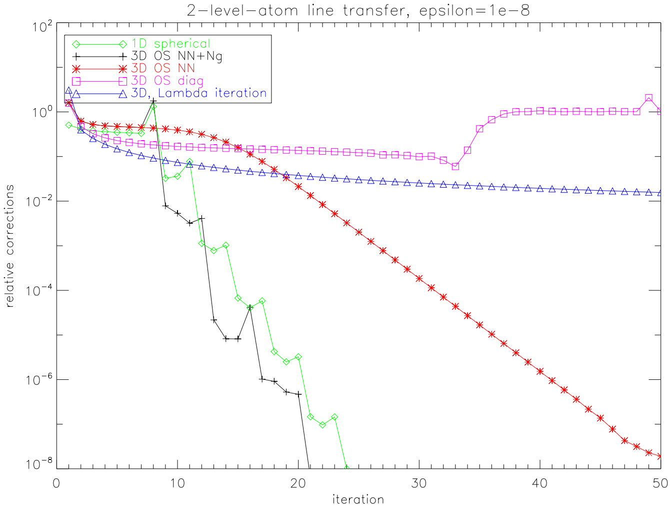
convergence: $\epsilon_l = 10^{-2}$



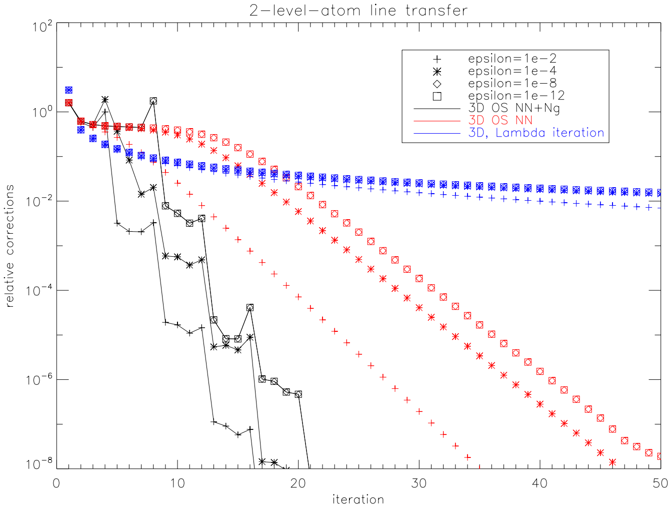
convergence: $\epsilon_l = 10^{-4}$



convergence: $\epsilon_l = 10^{-8}$



convergence



Parallelization

- each wavelength can be treated seperately!
- $\rightarrow$ additional parallelization over wavelength
- can be combined with solid angle parallelization

results: parallelization

N_{worker}	N_{cluster}	FS+Λ^*+OS step	FS+OS step
128	1	3018	1143
64	2	2595	1072
32	4	2340	1032
16	8	2308	1018
8	16	2264	1052
4	32	2318	1054

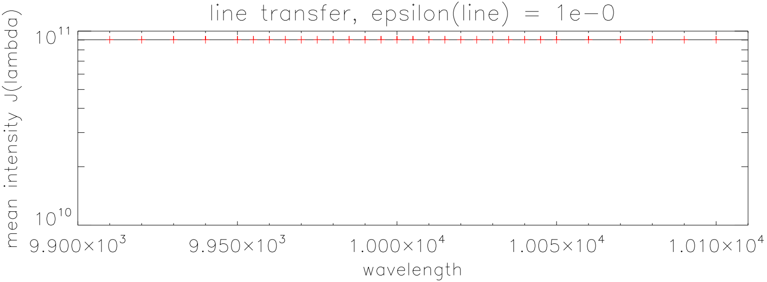
Periodic Boundary Conditions

- typically in stellar models
- PBCs in x and y (horizontal)
- z axis vertically (radial)
- PBCs implemented as wrap-around
- $\rightarrow$ trivial!
- though beware the evils of θ!

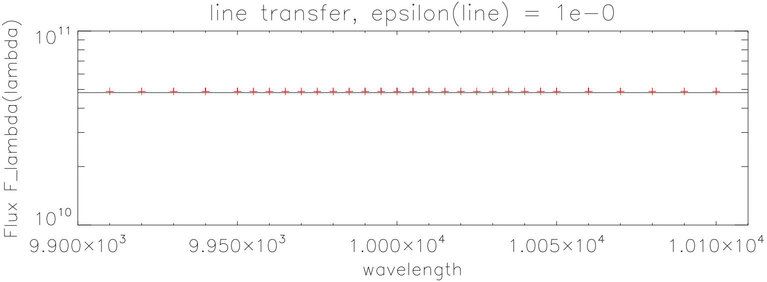
PBC testing

- can be directly compared to plane parallel model
- compare line spectra for test models

PBCs: LTE

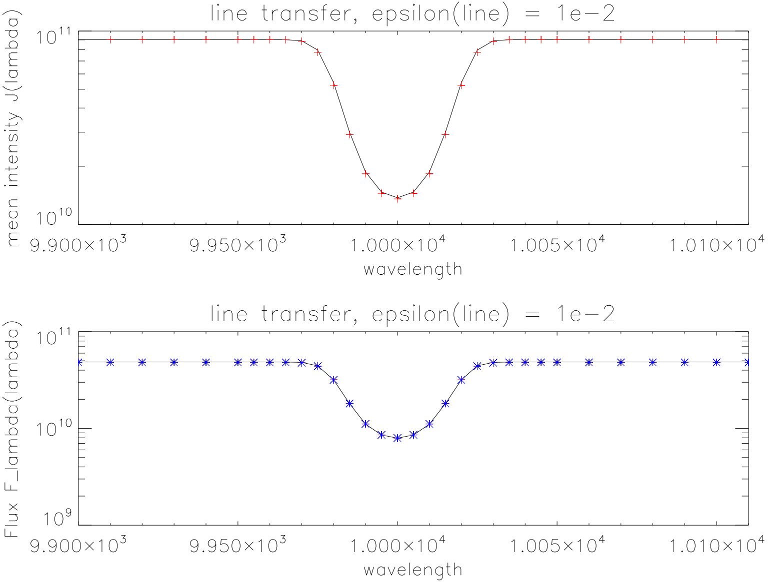


line transfer, epsilon(line) = 1e−0
mean intensity J(lambda)
10^11
10^10
9.900×10^3
9.950×10^3
1.000×10^4
1.005×10^4
1.010×10^4
wavelength

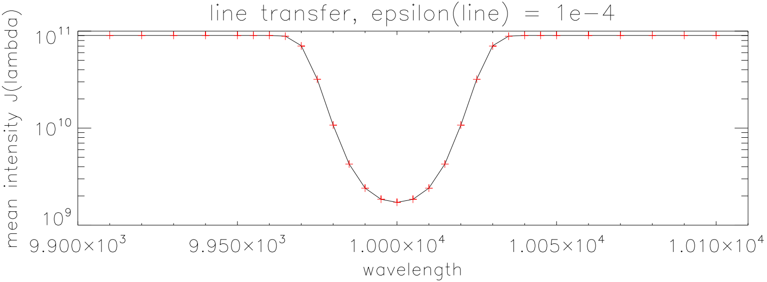


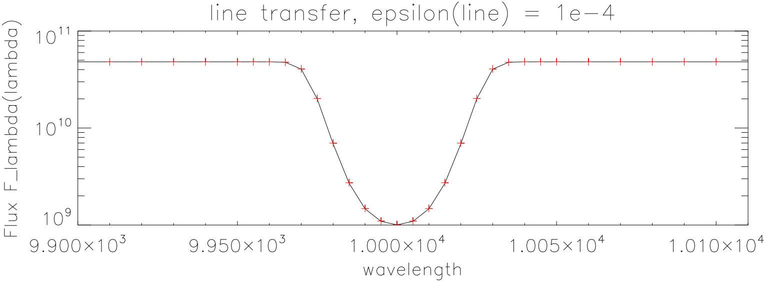
line transfer, epsilon(line) = 1e−0
Flux F_lambda(lambda)
10^11
10^10
9.900×10^3
9.950×10^3
1.000×10^4
1.005×10^4
1.010×10^4
wavelength

PBCs: $\epsilon_l = 10^{-2}$

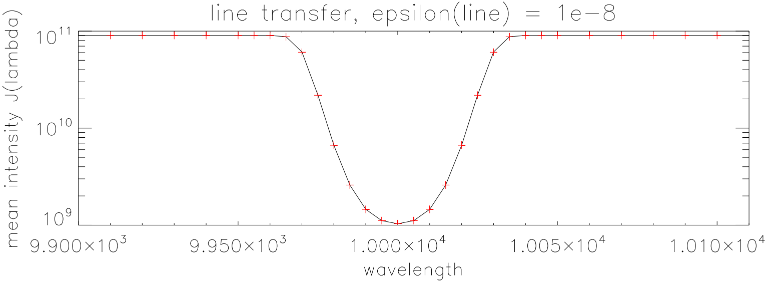


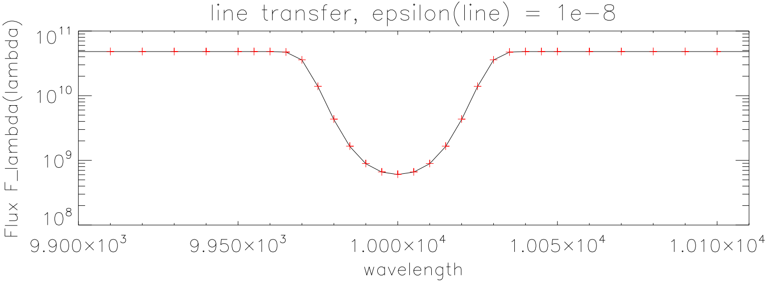
PBCs: $\epsilon_l = 10^{-4}$



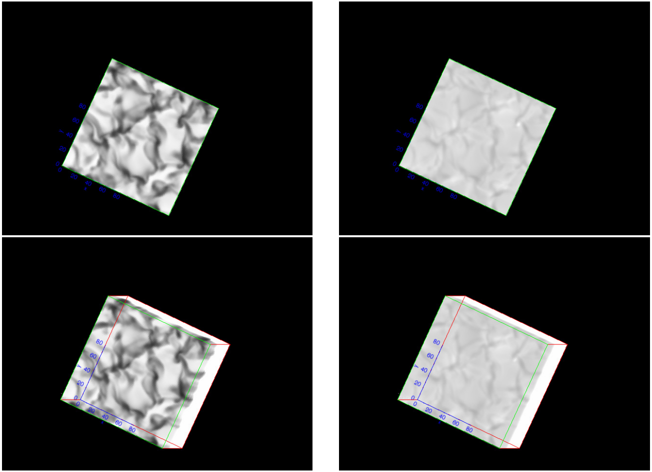


PBCs: $\epsilon_l = 10^{-8}$

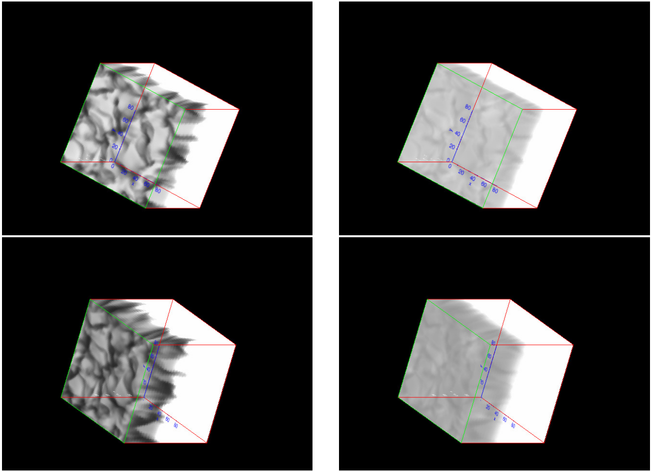




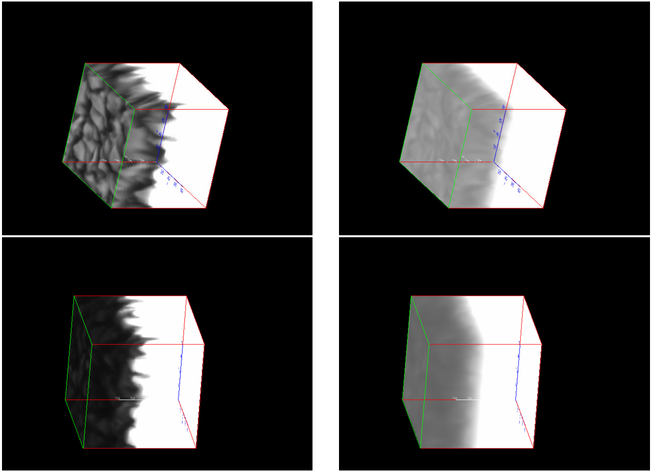
visualization: hydro model



visualization: hydro model



visualization: hydro model



visualization: line transfer hydro model