High-Tune: RenDeRer

Najda Villefranque[1]
najda.villefranque@lmd.ipsl.fr

Vincent Eymet[2]
vincent.eymet@meso-star.com

Vincent Forest[2]
vincent.forest@meso-star.com

[1]CNRS/LMD, [2]|Méso|Star>

April 14th 2021

Overview

Monte Carlo rendering engine of scenes with inhomogeneous media

- Backward path-tracing algorithm [Villefranque et al., 2019]

Pure CPU solver

- C ANSI [Kernighan, 1988]
- GNU/Linux 64-bits
- Mixed parallelisation (OpenMP/OpenMPI)

Command line interface

UNIX philosophy in mind

Write programs that do one thing and do it well

- Write readable, simple, small, transparent and robust programs

Write programs that work together

- Use composition

Write programs that handle text streams...

- *... because that is a universal interface*

"When in doubt, use brute force" *(K. Thompson)*

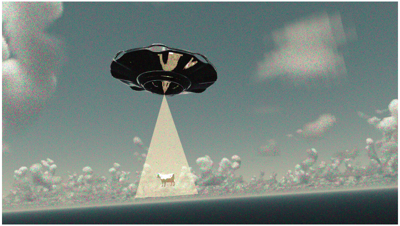
- Monte Carlo!

Libre Software [GPL, 2007]

Hacked by its community

- Study, modify and extend both data and code

(a) Spectral reflectivity support

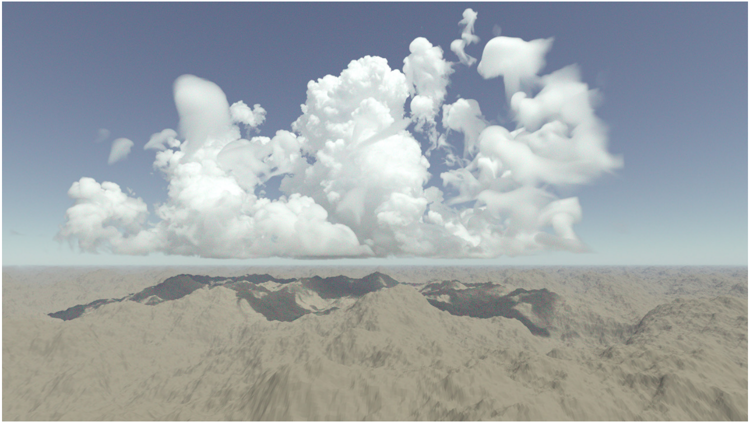


(b) Handling laser sheet [Sans et al., 2021]

History

2018, 2019[1]

- Rendering of cloudy atmospheres [Villefranque et al., 2019]
- Arbitrary ground geometries

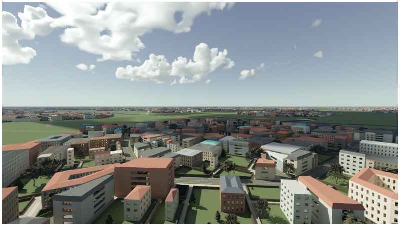


[1]High-Tune ANR-16-CE01-0010 - CNRS/CNRM, CNRS/LMD, LAPLACE

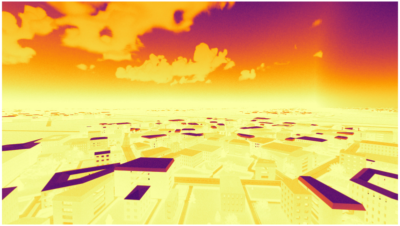
History

2020[2]

- Infrared rendering of cloudy atmospheres
- Solve upward and downward atmospheric fluxes (visible & infrared)
- Ground geometry with spectral reflectivity [Kotthaus et al., 2013]



(a) CIE XYZ rendering



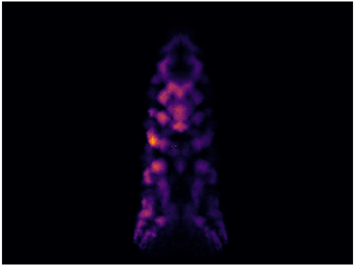
(b) Infrared rendering in 9 μm and 10 μm

[2] ModRadUrb Ademe MODEVAL-URBA-2019 - CNRS/CNRM, |Méso|Star>

To come up

2020, 2021[3]

- Radiative transfer in soot aggregates illuminated by a laser [Sans et al., 2021]



Monochromatic shortwave rendering at 532 nm

2021[4], ...

- Line sampling: a whole new class of radiative transfer algorithms

[3] Astoria ANR-18-CE05-0015 - ONERA, CNRS/RAPSODEE, CORIA

[4] MCG-Rad ANR-18-CE46-0012 - IRIT, CNRS/LMD, LAPLACE

Data

Input data

- 1D atmospheric profile
- 3D cloud field
- Water droplets properties
- Ground geometry and materials

Output data

- Image of per pixel Monte Carlo estimations

Input data - 1D atmospheric profile

RRTM-G optical properties computed from an 1D profile of T, P, q

- Handle horizontal variation of q by pre-processing N water vapor concentrations
- Absorption coefficient (visible & infrared)
- Scattering coefficient (visible)

k-distributions

- Computed with `ecRad` [Hogan and Bozzo, 2016]
- Tabulated per x_{H_2O} to handle local variations
- Interpolated at runtime according to x_{H_2O}

Input data - 3D cloud field

Computed from Large Eddy Simulations
[Lafore et al., 1997, Lac et al., 2018]

Per cell data

- Water vapor mixing ratio in kg of water per m^3 of dry air
- Liquid water in suspension mixing ratio in kg of water per m^3 of dry air
- Pressure in Pascal
- Temperature in Kelvin

Input data - Water droplets properties

Computed from a Mie code [Mishchenko et al., 2002]

- Discretised over the visible & infrared spectral ranges
- Integrated onto one water droplet distribution
  - In `htspk`: lognormal distribution with an effective radius of 10 μm and a standard deviation of 0.1 μm

Per wavelength data

- Massic absorption cross-sections
- Massic scattering cross-sections
- Asymmetric parameters of the equivalent Henyey-Greenstein phase function

Input data - Ground surface

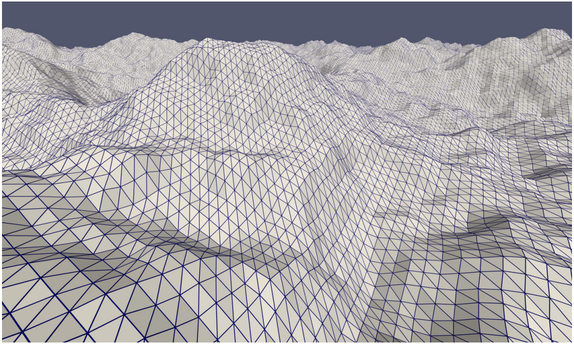
Arbitrary geometry stored in an `obj`-like format

Spectrally varying reflectivity (visible & infrared)

- 2 types of BRDF: lambertian and specular

```
# List of vertices
v 0 0 367.196
v 19.5312 0 353.952
v 39.0625 0 347.371
v 58.5938 0 345.122
v 78.125 0 340.736
...
v 9980.47 10000 390.285
v 10000 10000 367.196

# List of triangles
usemtl air:rock
f 1 514 2
f 2 514 515
f 2 515 3
f 3 515 516
...
f 262655 263168 262656
f 262656 263168 263169
```



Output data - Rendered image

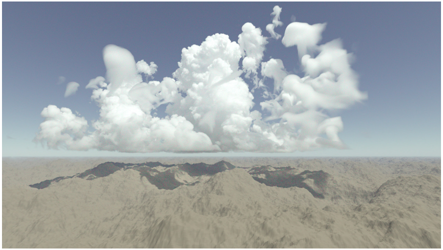
Array of per pixel Monte Carlo estimations

Camera image

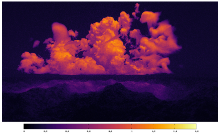
- CIE XYZ [CIE, 1931]
- Shortwave
- Longwave

Flux map

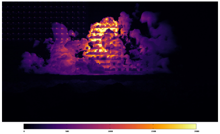
- Shortwave
- Longwave



(a) XYZ radiance



(b) X standard error



(c) Time per realisation

Beyond the renderer

Web site

- `www.meso-star.com/projects/high-tune/high-tune.html`

Reference documentation

- `man` pages

High-Tune: Starter Pack (`htspk`)

- Set of "ready for use" input data
- `www.meso-star.com/projects/high-tune/starter-pack.html`

High-Tune: Post Process (`htpp`)

- Post treatment of the generated images

(1931).
Commission International de l'Éclairage Proceedings.
Cambridge University Press, Cambridge.

(2007).
GNU General Public License.
https://www.gnu.org/licences/gpl.html.

Hable, J. (2010).
Filmic Tonemapping Operators.
filmicworlds.com/blog/filmic-tonemapping-operators.

Hogan, R. and Bozzo, A. (2016).
A new radiation scheme for the IFS.
ECMWF Technical Memorandum number, 787.

Kernighan, B. W. (1988).
The C Programming Language.
Prentice Hall Professional Technical Reference, 2nd edition.

Kotthaus, S., Smith, T., Wooster, M., and Grimmond, S. (2013).
Spectral library of impervious urban materials.

Lac, C., Chaboureau, J.-P., Masson, V., Pinty, J.-P., Tulet, P., Escobar, J., Leriche, M., Barthe, C., Aouizerats, B., Augros, C., Aumond, P., Auguste, F., Bechtold, P., Berthet, S., Bielli, S., Bosseur, F., Caumont, O., Cohard, J.-M., Colin, J., Couvreux, F., Cuxart, J., Delautier, G., Dauhut, T., Ducrocq, V., Filippi, J.-B., Gazen, D., Geoffroy, O., Gheusi, F., Honnert, R., Lafore, J.-P., Lebeaupin Brossier, C., Libois, Q., Lunet, T., Mari, C., Maric, T., Mascart, P., Mogé, M., Molinié, G., Nuissier, O., Pantillon, F., Peyrillé, P., Pergaud, J., Perraud, E., Pianezze, J., Redelsperger, J.-L., Ricard, D., Richard, E., Riette, S., Rodier, Q., Schoetter, R., Seyfried, L., Stein, J., Suhre, K., Taufour, M., Thouron, O., Turner, S., Verrelle, A., Vié, B., Visentin, F., Vionnet, V., and Wautelet, P. (2018).
Overview of the Meso-NH model version 5.4 and its applications.
Geoscientific Model Development, 11(5):1929–1969.

Lafore, J. P., Stein, J., Asencio, N., Bougeault, P., Ducrocq, V., Duron, J., Fischer, C., Héreil, P., Mascart, P., Masson, V., Pinty, J. P., Redelsperger, J. L., Richard, E., and de Arellano, J. V.-G. (1997).

The Meso-NH Atmospheric Simulation System. Part I: adiabatic formulation and control simulations.
Annales Geophysicae, 16(1):90–109.

Mishchenko, M. I., Travis, L. D., and Lacis, A. A. (2002).
Scattering, Absorption, and Emission of Light by Small Particles.
Cambridge University Press, Cambridge.

Sans, M., Eymet, V., Villefranque, N., El Hafi, M., Fourner, R., and Forest, V. (2021).
Null-collision meshless Monte-Carlo - A new backward integral formulation designed for laser-source emission in absorbing/emitting inhomogeneous media.
https://perso.imt-mines-albi.fr/~msans/.

Villefranque, N., Fournier, R., Couvreux, F., Blanco, S., Cornet, C., Eymet, V., Forest, V., and Tregan, J.-M. (2019).
A path-tracing monte carlo library for 3-d radiative transfer in highly resolved cloudy atmospheres.
Journal of Advances in Modeling Earth Systems, 11(8):2449–2473.

Converting a intensity spectral signal to a color

Rendering in the XYZ colorimetry space [CIE, 1931]

$$X = \int_0^{+\infty} \bar{x}(\lambda) L_\lambda(\vec{x}, -\vec{u}) d\lambda \qquad (1)$$

$$Y = \int_0^{+\infty} \bar{y}(\lambda) L_\lambda(\vec{x}, -\vec{u}) d\lambda \qquad (2)$$

$$Z = \int_0^{+\infty} \bar{z}(\lambda) L_\lambda(\vec{x}, -\vec{u}) d\lambda \qquad (3)$$

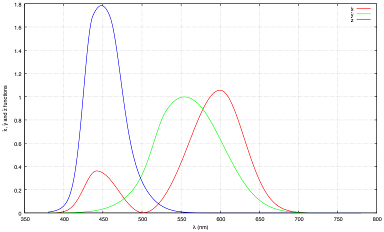


Figure: Functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ et $\bar{z}(\lambda)$

XYZ → sRGB conversion : "High-Tune: Post-Process"

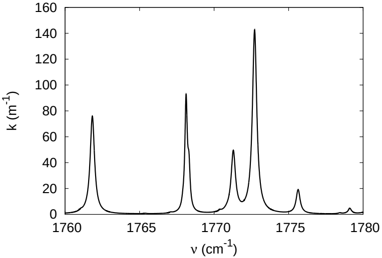
- https://gitlab.com/meso-star/htpp.git
- Tone mapping [Hable, 2010]
- XYZ → sRGB conversion : $C_{sRGB} = [M].C_{XYZ}$
- gamma correction

The k-distribution spectral model (1/4)

The spectral integration needs to be performed over the $[\nu_{min}, \nu_{max}]$ interval :

$$\frac{1}{\nu_{max} - \nu_{min}} \int_{\nu_{min}}^{\nu_{max}} A(\nu) d\nu \tag{4}$$

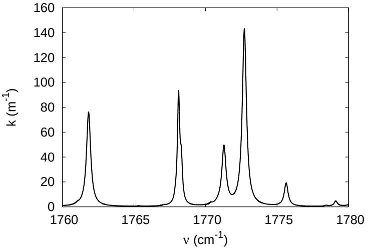
With A the monochromatic radiative quantity.

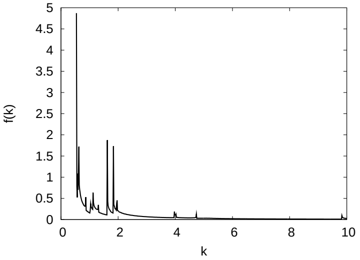


The k-distribution spectral model (2/4)

$$\frac{1}{\nu_{max} - \nu_{min}} \int_{\nu_{min}}^{\nu_{max}} A(\nu) d\nu = \int_0^{+\infty} A(k) f(k) dk \qquad (5)$$

$f(k)$: distribution function for $k(\nu)$
$f(k)dk$: fraction of ν_{max}-ν_{min} where $k \in [k; k + dk]$

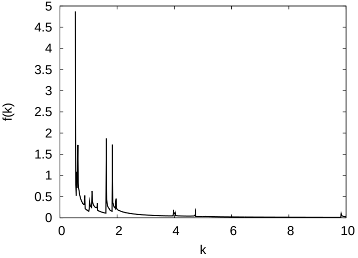


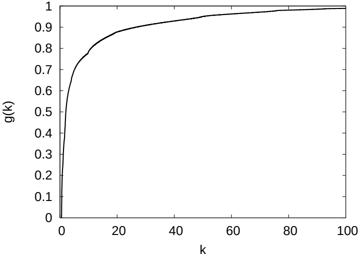


The k-distribution spectral model (3/4)

$$\frac{1}{\nu_{max} - \nu_{min}} \int_{\nu_{min}}^{\nu_{max}} A(\nu) d\nu = \int_0^{+\infty} A(k) f(k) dk = \int_0^1 A(k(g)) dg \quad (6)$$

with $g(k) = \int_0^k f(k')dk'$ the cumulative of $f(k)$





The k-distribution spectral model (4/4)

$$\frac{1}{\nu_{max} - \nu_{min}} \int_{\nu_{min}}^{\nu_{max}} A(\nu) d\nu = \int_0^1 A(k(g)) dg \approx \sum_{i=1}^{N} A(k(g_i)) \omega_i \quad (7)$$

g_i : quadrature abscissae
ω_i : quadrature weight
$k(g_i) \equiv k_i$: values of k

Advantage

Instead of performing thousands of monochromatic radiative transfer computations, only $N \approx 10$ independant computations are required for a spectral interval a few tenths of cm^{-1} wide.

Disadvantage

For a heterogeneous medium, a small error (≈ 5 %) is introduced: when T or P change, the $k(\nu)$ spectrum is modified and a given value of g_i/k_i no longer corresponds to the same pool of frequencies. The $k(\nu)$ spectrum is therefore supposed to be homothetic when thermodynamic conditions are modified, in order to assume a radiative transfer computation for a given k_i is equivalent to a series of monochromatic radiative transfer computations ("correlated-k" hypothesis, more or less relevant).

Using High-Tune: RenDeRer

Install

Prerequisites

- git and git-lfs
- GNU Compuler Collection (version $\geq$ 4.8)
- CMake (version $\geq$ 3)
- NetCDF library and headers (version $\geq$ 4)
- OpenMPI library and headers
- AsciiDoc to generate the man pages (optional)

Install htrdr

```
~ $ git clone -b High-Tune-0.6.1 \
    https://gitlab.com/meso-star/star-engine.git High-Tune-0.6.1
~ $ mkdir High-Tune-0.6.1/build; cd High-Tune-0.6.1/build
~/High-Tune-0.6.1/build $ cmake ../cmake
~/High-Tune-0.6.1/build $ make
```

Install htspk

```
~ $ wget https://www.meso-star.com/projects/high-tune/downloads/\
    High-Tune-Starter-Pack-0.6.0.tar.gz
~ $ tar xzvf High-Tune-Starter-Pack-0.6.0.tar.gz
```

Setup the working environment

Register `htrdr` against the current GNU/Bash shell

```
~ $ source ~/High-Tune-0.6.1/local/etc/high_tune.profile
~ $ htrdr -h
~ $ man htrdr
~ $ export HTSPK=~/High-Tune-Starter-Pack-0.6.0
```

Create the working directory

```
~ $ mkdir Tuto
~ $ cd Tuto
~/Tuto $ echo "Hello, world!"
```

Clear sky rendering

First rendering

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=128 \
  -C pos=0,0,100:tgt=0,1,100 \
  -o clear_sky.txt
```

Display the rendered image

```
~/Tuto $ htpp -h
~/Tuto $ man htpp
~/Tuto $ htpp clear_sky.txt | display -
~/Tuto $ htpp -o clear_sky.ppm clear_sky.txt
~/Tuto $ display clear_sky.ppm
~/Tuto $ htpp -i exposure=0.2 -o clear_sky_0.2.ppm clear_sky.txt
~/Tuto $ display clear_sky.ppm clear_sky_0.2.ppm
```

Overwrite the default sun position

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=128 \
  -C pos=0,0,100:tgt=0,1,100 \
  -D 90,20 \
  -o clear_sky.txt -f

~/Tuto $ htpp -i exposure=0.2 clear_sky.txt | display -
```

Update the camera

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=128 \
  -C pos=0,0,100:tgt=0,1,100.5:up=0,0,1:fov=60 \
  -D 90,20 \
  -o clear_sky.txt -f

~/Tuto $ htpp -i exposure=0.2 clear_sky.txt | display -
```

Cloud field rendering

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=4 \
  -C pos=0,0,100:tgt=0,1,100.5:up=0,0,1:fov=60 \
  -D 90,20 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp \
  -O DZVAR_octrees.cache \
  -o sky_DZVAR.txt

~/Tuto $ htpp -i exposure=0.2 sky_DZVAR.txt | display -
```

Infinitly repeat the clouds

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=4 \
  -C pos=0,0,100:tgt=0,1,100.5:up=0,0,1:fov=60 \
  -D 90,20 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -O DZVAR_octrees.cache \
  -o sky_DZVAR.txt -f

~/Tuto $ htpp -i exposure=0.2 sky_DZVAR.txt | display -
```

Dump the octrees

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp \
  -O DZVAR_octrees.cache \
  -d \
  -o octrees.txt

~/Tuto $ csplit \
  -f cloud_octree_ \
  -b %02d.vtk \
  -z \
  --suppress-matched \
  octrees.txt \
  /^---$/ *

~/Tuto $ paraview cloud_octree_11.vtk
```

Update the optical thickness criterion

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=640x480:spp=4 \
  -C pos=0,0,100:tgt=0,1,100.5:up=0,0,1:fov=60 \
  -D 90,20 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -T 0.01 \
  -O DZVAR_octrees_0.01.cache \
  -o sky_DZVAR_0.01.txt

~/Tuto $ htpp -i exposure=0.2 sky_DZVAR_0.01.txt | display -
```

Update the ground surface

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/mountain.mtls \
  -g $HTSPK/models/mountain.obj -R \
  -i def=640x480:spp=4 \
  -C pos=0,0,600:tgt=0,1,600.1:up=0,0,1:fov=60 \
  -D 90,60 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -O DZVAR_octrees.cache \
  -o sky_DZVAR_mountain.txt

~/Tuto $ htpp -i exposure=0.2 sky_DZVAR_mountain.txt | display -
```

Infrared rendering

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/mountain.mtls \
  -g $HTSPK/models/mountain.obj -R \
  -i def=640x480:spp=128 \
  -C pos=0,0,600:tgt=0,1,600.1:up=0,0,1:fov=60 \
  -D 90,60 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -s lw=9000,10000:Tref=590 \
  -O DZVAR_octrees_lw_9000nm_10000nm_590K.cache \
  -o sky_DZVAR_mountain_lw.txt
```

Display heat map

Display expected value

```
~/Tuto $ htpp -v -m pixcpnt=0 sky_DZVAR_mountain_lw.txt | display -
~/Tuto $ htpp -v -m pixcpnt=0:gnuplot -o heat_map.gp sky_DZVAR_mountain_lw.txt
~/Tuto $ gnuplot heat_map.gp > heat_map.png
~/Tuto $ display heat_map.png
```

Display standard error

```
~/Tuto $ htpp -v -m pixcpnt=1:gnuplot sky_DZVAR_mountain_lw.txt | gnuplot - | display -
```

Computing ascending flux in longwave

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=512x512:spp=128 \
  -D 0,90 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -s lw=4000,10000 \
  -O DZVAR_octrees_lw_4000nm_10000nm.cache \
  -p pos=3200,3200,80000:tgt=3200,3200,0:up=0,1,0:sz=6400,6400 \
  -o sky_DZVAR_plane_lw_ascending_flux.txt

~/Tuto $ htpp -v -m default sky_DZVAR_plane_lw_ascending_flux.txt | display -
```

Computing descending flux in longwave

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=512x512:spp=128 \
  -D 0,90 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -s lw=4000,10000 \
  -O DZVAR_octrees_lw_4000nm_10000nm.cache \
  -p pos=3200,3200,1:tgt=3200,3200,2:up=0,1,0:sz=6400,6400 \
  -o sky_DZVAR_plane_lw_descending_flux.txt

~/Tuto $ htpp -v -m default sky_DZVAR_plane_lw_descending_flux.txt | display -
```

Computing descending flux in shortwave

```
~/Tuto $ htrdr -v \
  -a $HTSPK/ecrad_opt_prop.txt \
  -M $HTSPK/materials/plane.mtls \
  -g $HTSPK/models/plane.obj -R \
  -i def=512x512:spp=32 \
  -D 0,90 \
  -m $HTSPK/Mie_LUT_Cloud-2-10-0.010.nc \
  -c $HTSPK/clouds/DZVAR.1.ARMCU.008.diaKCL.htcp -r \
  -s sw=380,780 \
  -O DZVAR_octrees_sw_380nm_780m.cache \
  -p pos=3200,3200,1:tgt=3200,3200,2:up=0,1,0:sz=6400,6400 \
  -o sky_DZVAR_plane_sw_descending_flux.txt

~/Tuto $ htpp -v -m range=0,900 sky_DZVAR_plane_sw_descending_flux.txt | display -
```

The `ht-run.sh` GNU/Bash script

Build a `htrdr` command line from an input file that describes the scene to render

```
~/Tuto $ cat $HTSPK/scenes/city
~/Tuto $ cat $HTSPK/ht-run.sh | more
~/Tuto $ bash $HTSPK/ht-run.sh $HTSPK/scenes/city
~/Tuto $ htpp -i exposure=0.2 city_1280x720x256.txt | display -
```