HELIOS-K: The challenge of calculating opacity functions for 10^{10} molecular lines

Simon L. Grimm
Kevin Heng

Center for Space and Habitability
University of Bern

simon.grimm@csh.unibe.ch

Opacity functions

- Opacity functions are a key part of all sorts of (exo)planetary atmosphere characterisation

Opacity functions

- Opacity functions are a key part of all sorts of (exo)planetary atmosphere characterisation
- We need to calculate them for many different values of temperature and pressure

Opacity functions

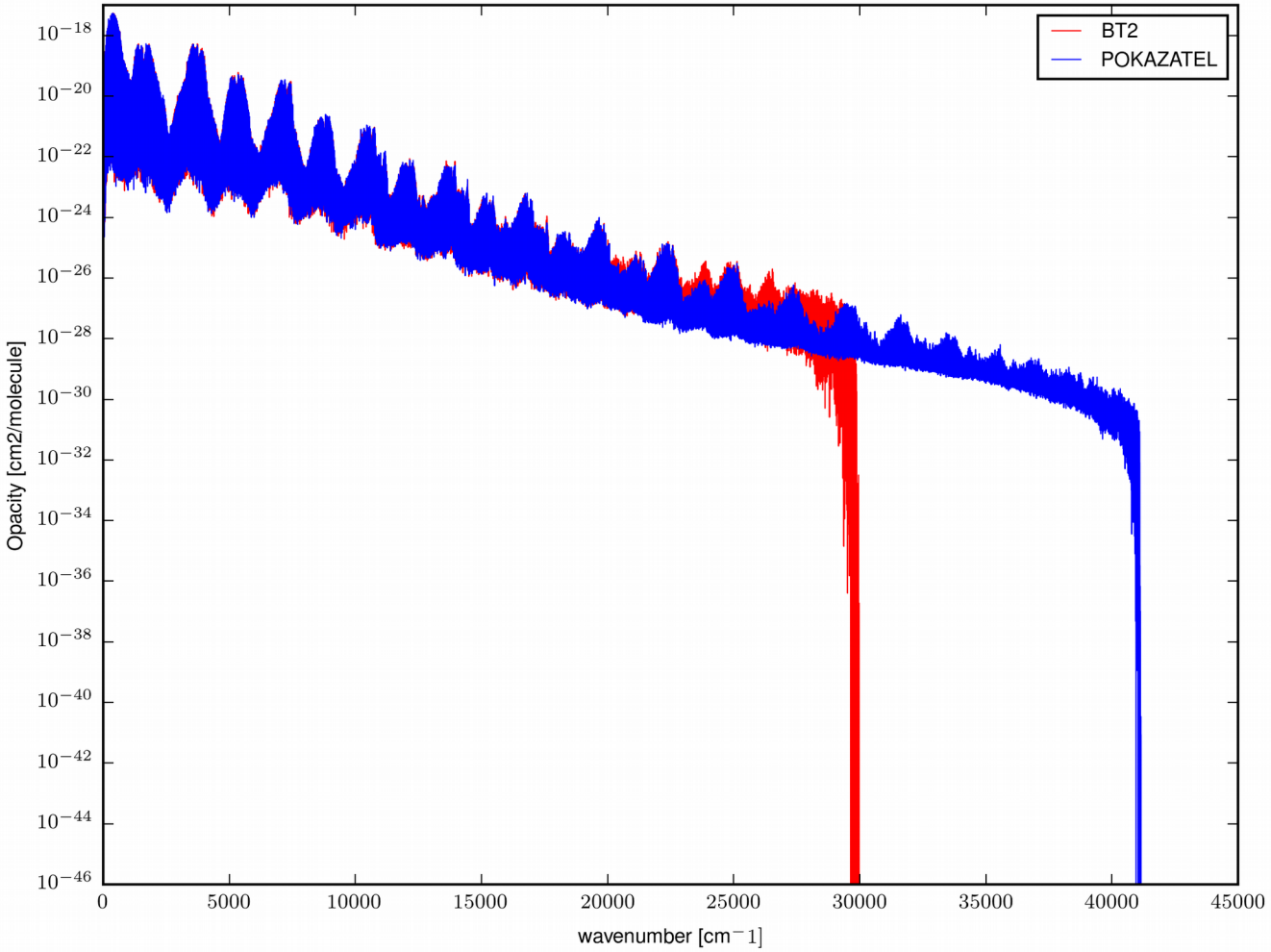
- Opacity functions are a key part of all sorts of (exo)planetary atmosphere characterisation
- We need to calculate them for many different values of temperature and pressure
- For a reasonable resolution in wavenumbers: ~0.01 cm^{-1}

Opacity functions

- Opacity functions are a key part of all sorts of (exo)planetary atmosphere characterisation
- We need to calculate them for many different values of temperature and pressure
- For a reasonable resolution in wavenumbers: ~0.01 cm^{-1}
- Different datasets: Hitran, HITEMP, Kurucz, ExoMol, ...

Example of a large line dataset: H20

BT2	T < 3000K	5*10^8 lines
POKAZATEL	T < 6000K	6*10^9 lines
CH3Cl		1.6 * 10^11 lines



Computational cost

- More than 10^{18} points to compute!
- Use the most efficient hardware to solve it!

Computational cost

- More than 10^{18} points to compute!
- Use the most efficient hardware to solve it!
- Use GPUs!

GPUs

- GPUs can be used to perform parallel computing very effitiently
- Different task can not be all independent
- Needs a special way to parallelise the work flow

GPUs

- GPUs can be used to perform parallel computing very effitiently
- Different task can not be all independent
- Needs a special way to parallelize the work flow

- Opacity functions are in principle easy to parallelise, but in practise, they are not

The opacity function

The opacity function

$$\kappa = S\Phi$$

Intensity * line shape

The opacity function

$$\kappa = S\Phi$$

Intensity * line shape

$$S = \frac{\pi e^2 g_i f_{ij}}{m_e c} \frac{e^{-hcF_i/kT}}{Q(T)} \left[1 - e^{-hc(F_j - F_i)/kT}\right]$$

Q(T) : partition function
F: Enery levels

f : Oscillator strength
g: degeneracy

The opacity function

$$\kappa = S\Phi$$

Intensity * line shape

$$S = \frac{\pi e^2 g_i f_{ij}}{m_e c} \frac{e^{-hcF_i/kT}}{Q(T)} \left[1 - e^{-hc(F_j - F_i)/kT}\right]$$

Q(T) : partition function
F: Enery levels

f : Oscillator strength
g: degeneracy

Line shape can be: Gaussian, Lorentzian, Voigt, non-Voigt, ...
We focus here on Voigt profiles.

ExoMol

Molecules

other oxides
- CO
- NO
- PO

ions
- LiH^+
- H_2^+
- HeH^+
- H_3^+

other hydrides
- NH
- CH
- OH
- HCl
- SiH
- SH

other diatomics
- PN
- KCl
- NaCl
- LiCl
- CN
- C_2
- H_2
- CS
- CP
- PS
- NS
- SiS

metal hydrides
- MgH
- NaH
- NiH
- AlH
- CrH
- CaH
- BeH
- TiH
- FeH
- LiH
- ScH

larger molecules
- CH_4
- NH_3
- HNO_3
- H_2O_2
- H_2CO
- PH_3
- SO_3
- SiH_4
- CH_3F
- CH_3Cl
- C_2H_4

metal oxides
- VO
- AlO
- TiO
- SiO
- CaO

triatomic molecules
- H_2O
- CO_2
- SO_2
- HCN
- N_2O
- H_2S

More than 10^10 lines

The VOMYT dataset for $^{51}V^{16}O$

Definitions file

51V-16O__VOMYT.def

Spectroscopic Model

51V-16O__VOMYT.model

VOMYT: partition function

New VO Linelist, utilising a spectroscopic model with the 13 lowest electronic states.

51V-16O__VOMYT.pf [265.6 KB]
Partition function for (51V)(16O) in the temperature range 1 – 8000 K in 1 K steps from the VOMYT data set.

References

1. McKemmish, L. K., Yurchenko, S. N., Tennyson, J., "ExoMol line lists XVIII – The high temperature spectrum of VO", *Monthly Notices of the Royal Astronomical Society* **463**, 771-793 (2016). [link to article] [16McYuTe.VO]

VOMYT: line list

New VO Linelist, utilising a spectroscopic model with the 13 lowest electronic states.

51V-16O__VOMYT.states.bz2 [6.4 MB]
States file for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__00000-05000.trans.bz2 [263.7 MB]
Transitions file in the region 0 – 5000 cm-1 for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__05000-10000.trans.bz2 [366.2 MB]
Transitions file in the region 5000 – 10000 cm-1 for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__10000-15000.trans.bz2 [439.6 MB]
Transitions file in the region 10000 – 15000 cm-1 for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__15000-20000.trans.bz2 [503.1 MB]
Transitions file in the region 15000 – 20000 cm-1 for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__20000-25000.trans.bz2 [587.6 MB]
Transitions file in the region 20000 – 25000 cm-1 for the VOMYT linelist for (51V)(16O)

51V-16O__VOMYT__25000-30000.trans.bz2 [676.7 MB]
Transitions file in the region 25000 – 30000 cm-1 for the VOMYT linelist of (51V)(16O)

51V-16O__VOMYT__30000-35000.trans.bz2 [452.4 MB]
Transitions file in the region 30000 – 35000 cm-1 for the VOMYT linelist for (51V)(16O)

References

1. McKemmish, L. K., Yurchenko, S. N., Tennyson, J., "ExoMol line lists XVIII – The high temperature spectrum of VO", *Monthly Notices of the Royal Astronomical Society* **463**, 771-793 (2016). [link to article] [16McYuTe.VO]

ExoMol structure

- *.def file (mass, number of files, T range)
- *.trans files (level indices, A-coefficient, ν)
- *.states files (energies, degeneracies)
- *.pf file (partition function)

Optimisation and precalculations

$$S = \frac{\pi e^2 g_i f_{ij}}{m_e c} \frac{e^{-hcF_i/kT}}{Q(T)} \left[1 - e^{-hc(F_j - F_i)/kT}\right]$$

$$S = L_{ij} \frac{e^{-c_1 F_i/T}}{Q(T)} \left[1 - e^{-c_1 \nu_{ij}/T}\right]$$

L is independent of T, P and ν
Preprocess ExoMol files, and store L in binary files

The Voigt-Profile

$$\Phi = \left(\frac{\ln 2}{\pi}\right)^{1/2} \frac{H_{\mathrm{V}}}{\Gamma_{\mathrm{D}}},$$

$$H_{\mathrm{V}} = \frac{a}{\pi} \int_{-\infty}^{+\infty} \frac{\exp\left(u'^2\right)}{\left(u - u'\right)^2 + a^2}\, du'$$

Convolution of a
Gaussia and a Lorentzian profile

The infinite integral must be solved numerically

The Voigt-Profile

$$\Phi = \left(\frac{\ln 2}{\pi}\right)^{1/2} \frac{H_{\mathrm{V}}}{\Gamma_{\mathrm{D}}},$$

$$H_{\mathrm{V}} = \frac{a}{\pi} \int_{-\infty}^{+\infty} \frac{\exp\left(u'^2\right)}{\left(u - u'\right)^2 + a^2}\, du'$$

Convolution of a
Gaussia and a Lorentzian profile

The infinite integral must be solved numerically

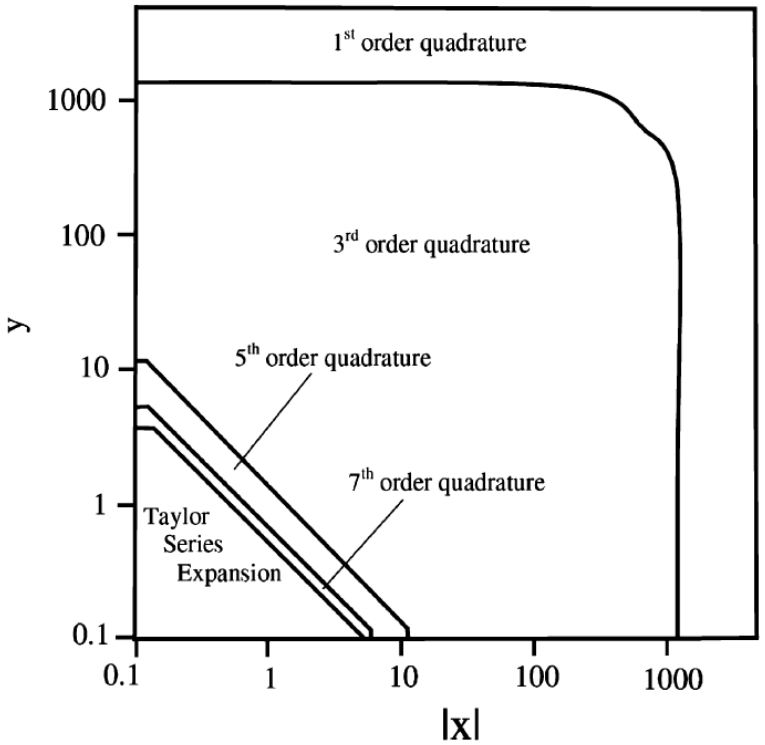
$$x = \sqrt{\ln 2}\, \frac{\nu - \nu_0}{\alpha_{\mathrm{D}}},$$

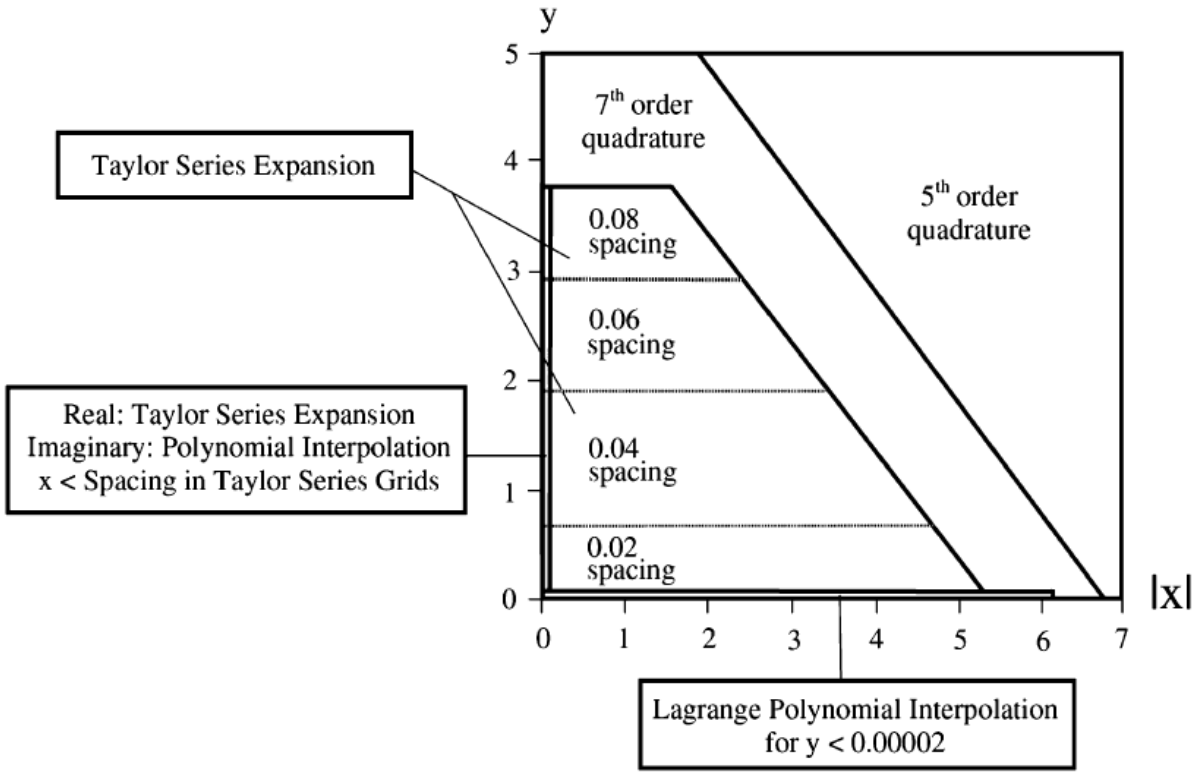
$$y = \sqrt{\ln 2}\, \frac{\alpha_{\mathrm{L}}}{\alpha_{\mathrm{D}}}.$$

Can be written as a two parameter function V(x, y)

Compute the Voigt function II: different methods for x and y regions

For small values of x and y, use a precomputed table and interpolate

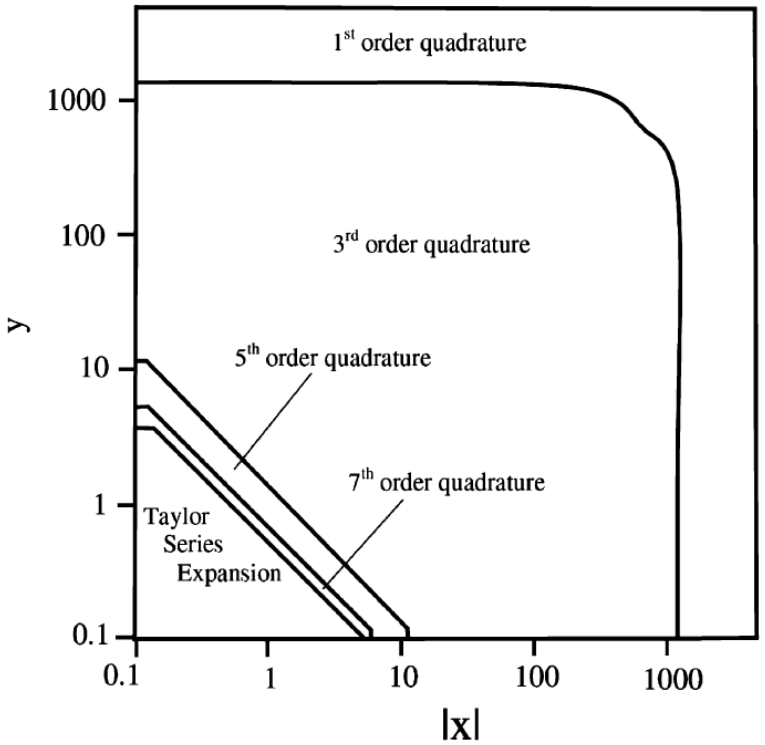


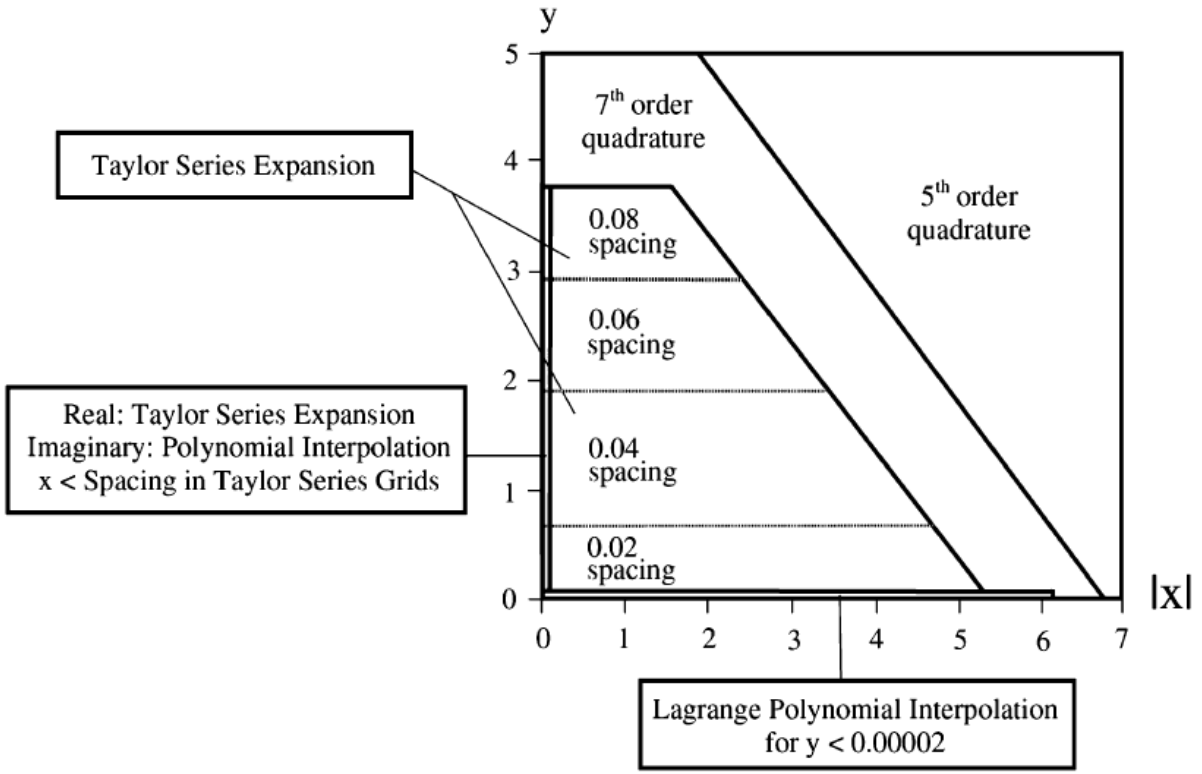


K.L. Letchworth, D.C. Benner / Journal of Quantitative Spectroscopy & Radiative Transfer 107 (2007) 173–192

Voigt function: different methods for x and y regions

For small values of x and y, use a precomputed table and interpolate





This method is not efficient for GPU computing, because it has too many different branches.

K.L. Letchworth, D.C. Benner / Journal of Quantitative Spectroscopy & Radiative Transfer 107 (2007) 173–192

Helios-K

- Use only 3 different regions
  - 1 order quadrature $1'000'000 < x^2 + y^2$
  - 3 order quadrature $100 < x^2 + y^2 < 1'000'000$
  - Algorithm 916 $x^2 + y^2 < 100$
- Hitran $< 10^6$ lines
- HITEMP $> 10^8$ lines
- ExoMol $> 10^{10}$ lines

  To cover all these databases needs different parallelisation techniques,

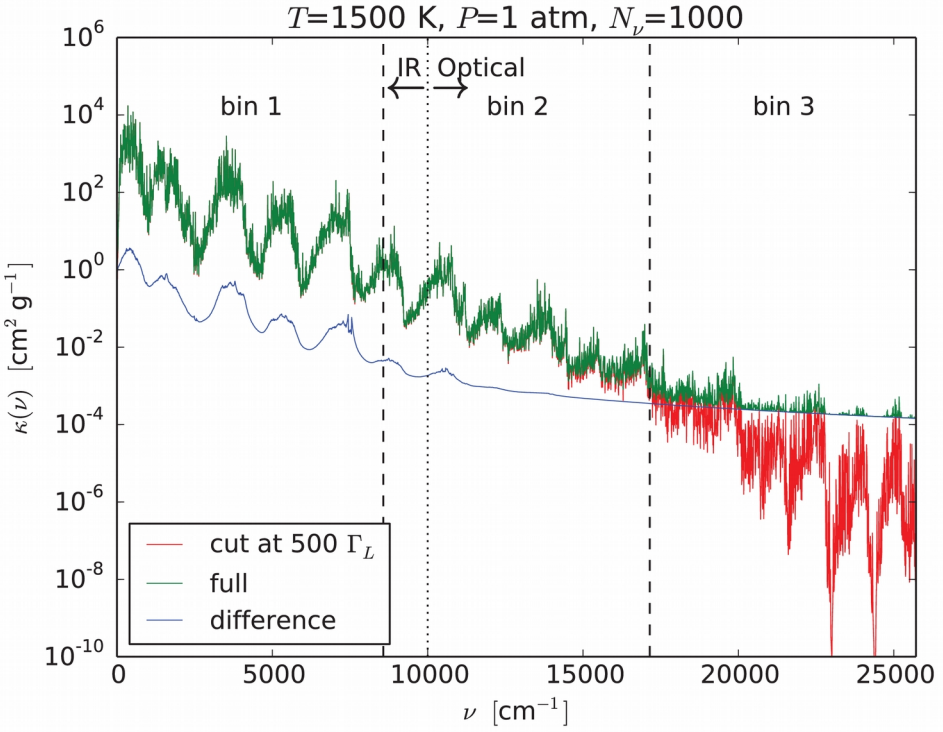
  because data transfer to compute ratio is very different.

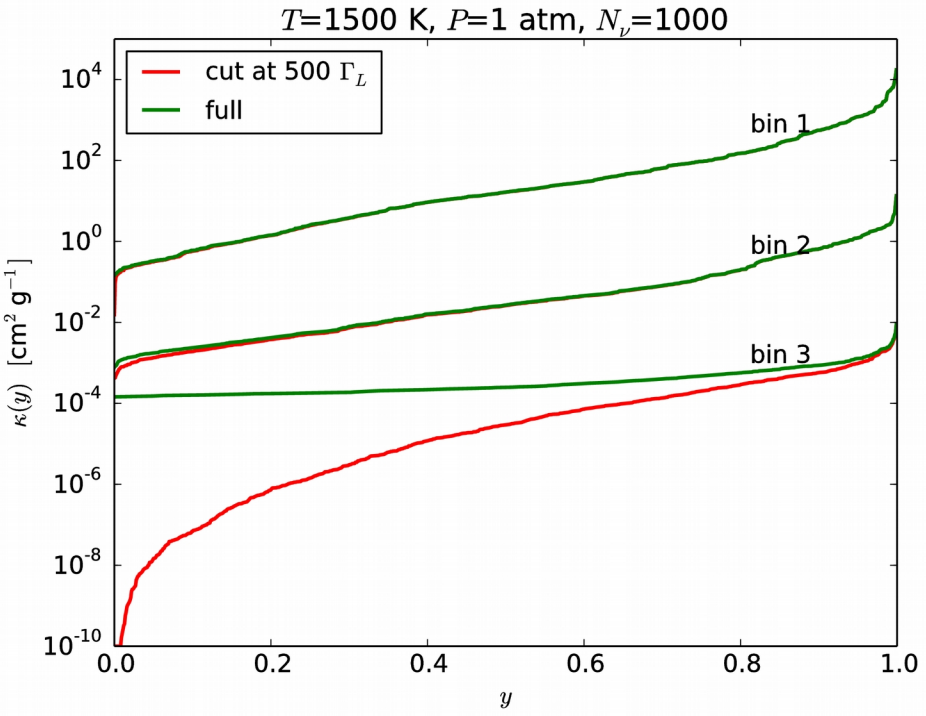
Once we have computed the opacities,

whats next?

How to use them?

The k-distribution method





The k-distribution method can be used to save a lot of storage space.
But it makes it hard to mix different molecular species.

The Bern opacity function grid

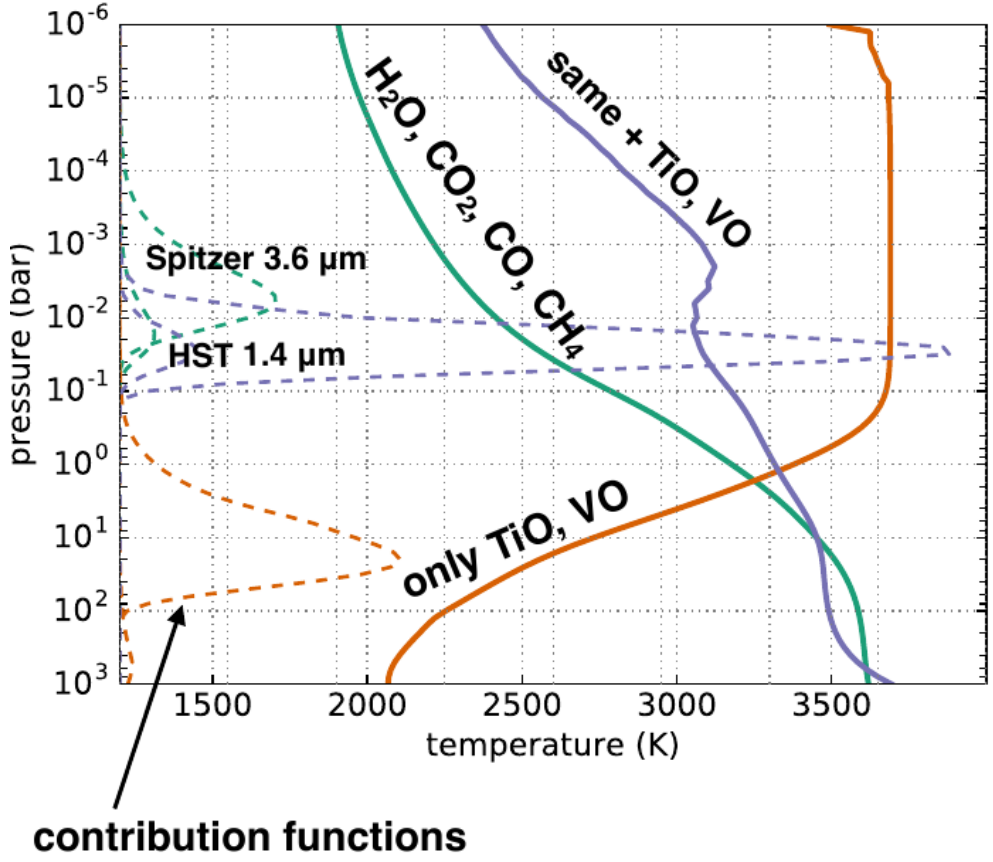
- All molecules from ExoMol (Hitran, ...)
- Grid of opacity functions in T and P
- Different versions of linelists
- Several Tbytes of Data

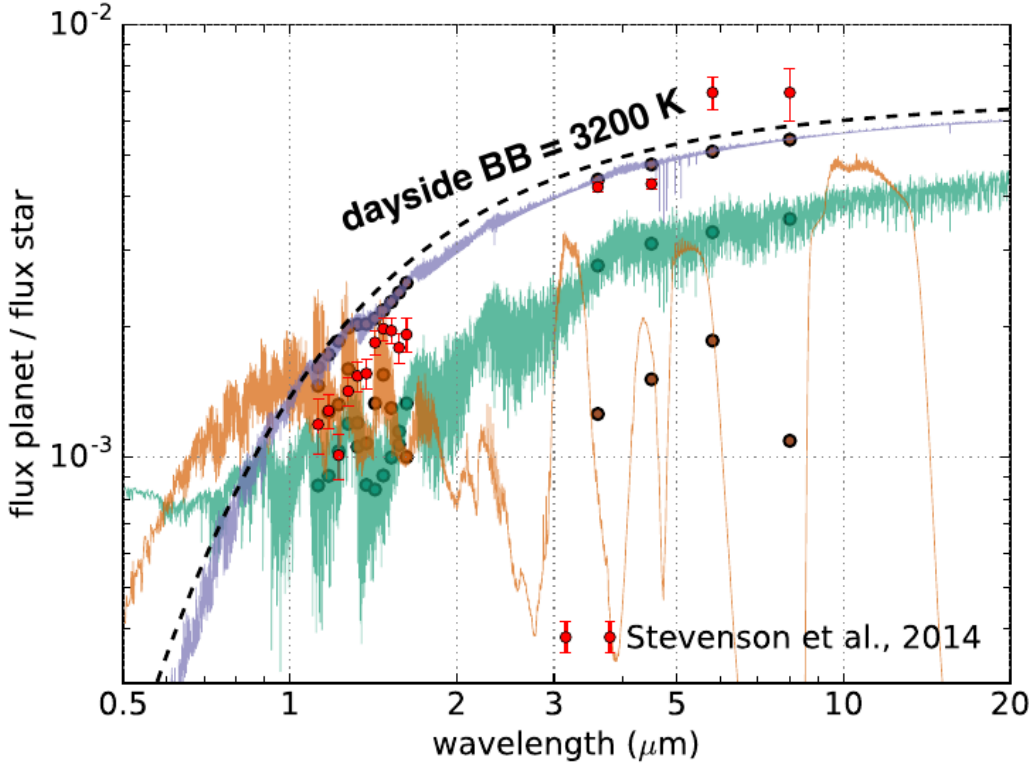
- Make everything open access
  on our data server via web interface.

Prelimiary 1D radiative transfer results

See poster from Matej Malik

Secondary eclipse of WASP-12b





Biggest challenges

- ExoMol releases constantly new and much larger line lists.
- Code must be even faster.
- Include other broadening effects
- Efficiently combine different species

Thank you