ESP summer school

— VULCAN 1D photochemical kinetics

(Tsai et al. 2017)

Shang-Min “Shami”, Tsai
University of Oxford

CO_2 stability on Mars

$CO_2 + h\nu \rightarrow CO + O \quad h\nu < 200nm$

$CO + O + M \rightarrow CO_2 + M$ (spin-forbidden)

$O + O + M \rightarrow O_2 + M$

---

net: $2CO_2 \rightarrow 2CO + O_2$

CO_2 stability on Mars

$CO_2 + h\nu \rightarrow CO + O \quad h\nu < 200nm$

~~$CO + O + M \rightarrow CO_2 + M$ (spin-forbidden)~~

$O + O + M \rightarrow O_2 + M$

---

net: $2CO_2 \rightarrow 2CO + O_2$

CO_2 stability on Mars

$CO_2 + h\nu \rightarrow CO + O \quad h\nu < 200nm$

~~$CO + O + M \rightarrow CO_2 + M$ (spin-forbidden)~~

$O + O + M \rightarrow O_2 + M$

---

net: $2CO_2 \rightarrow 2CO + O_2$

But on Mars, CO / O2 ~ 1:2!

Exoplanet applications

- Determining the atmospheric composition

Exoplanet applications

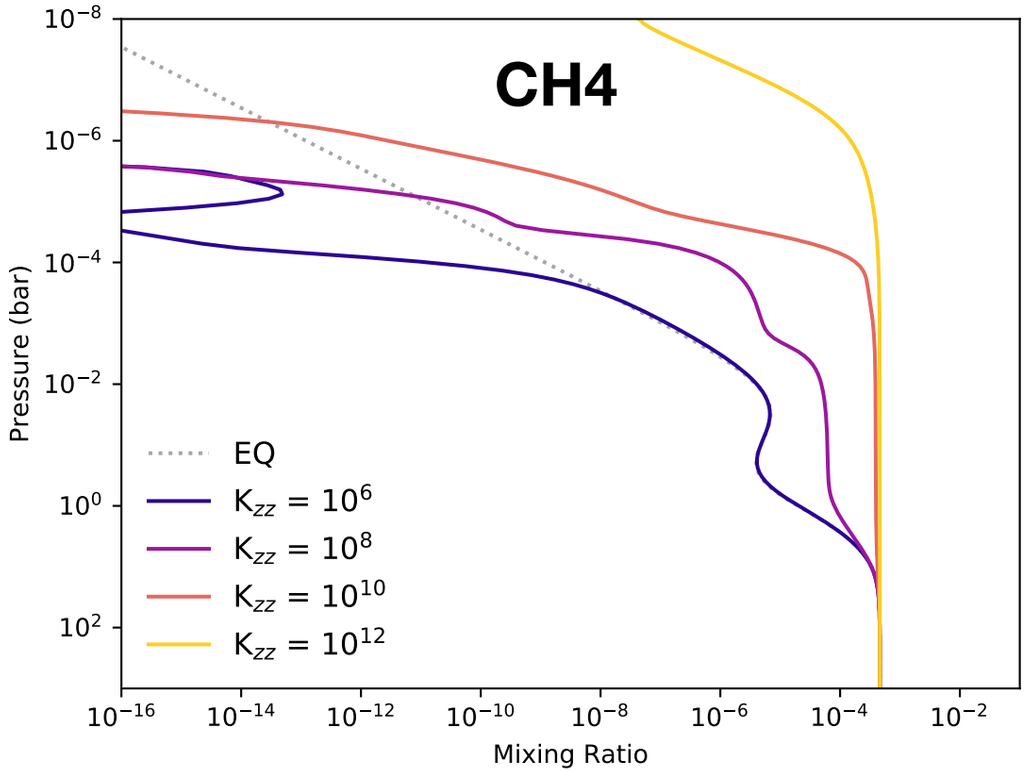
- Determining the atmospheric composition

Input:

- Atmospheric structure: T-P profile, mixing (eddy diffusion) etc.
- Elemental abundances: C/H, O/H, N/H etc.
- Boundary conditions

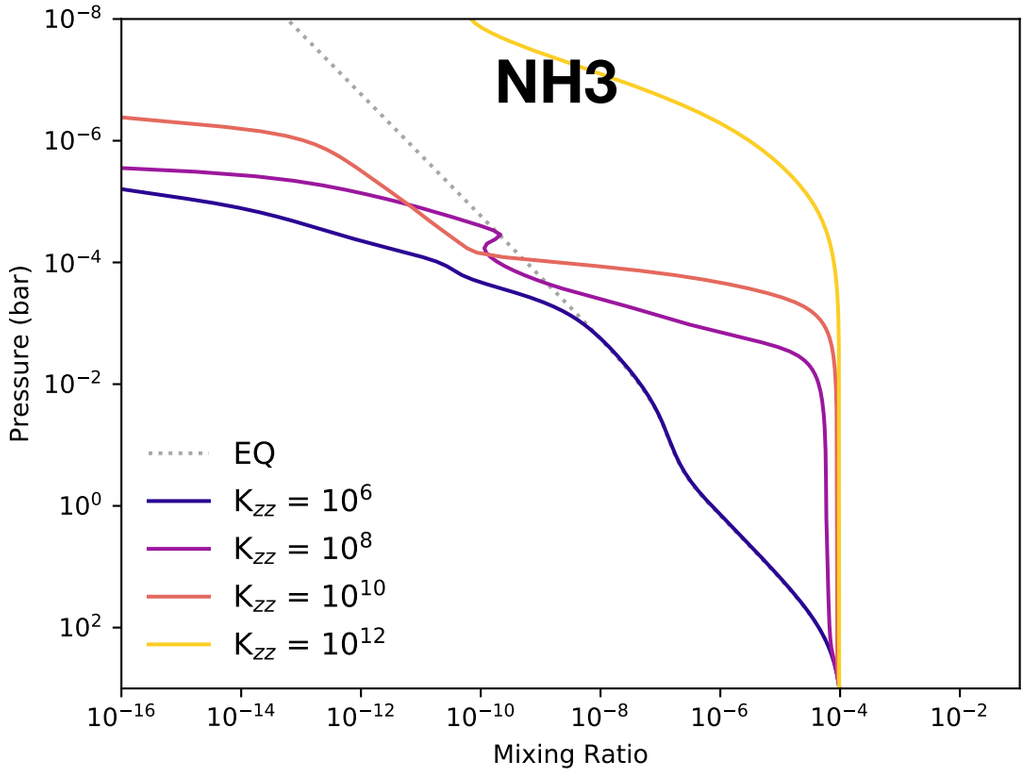
Exoplanet applications

- Determining the atmospheric composition
- Interpreting observations



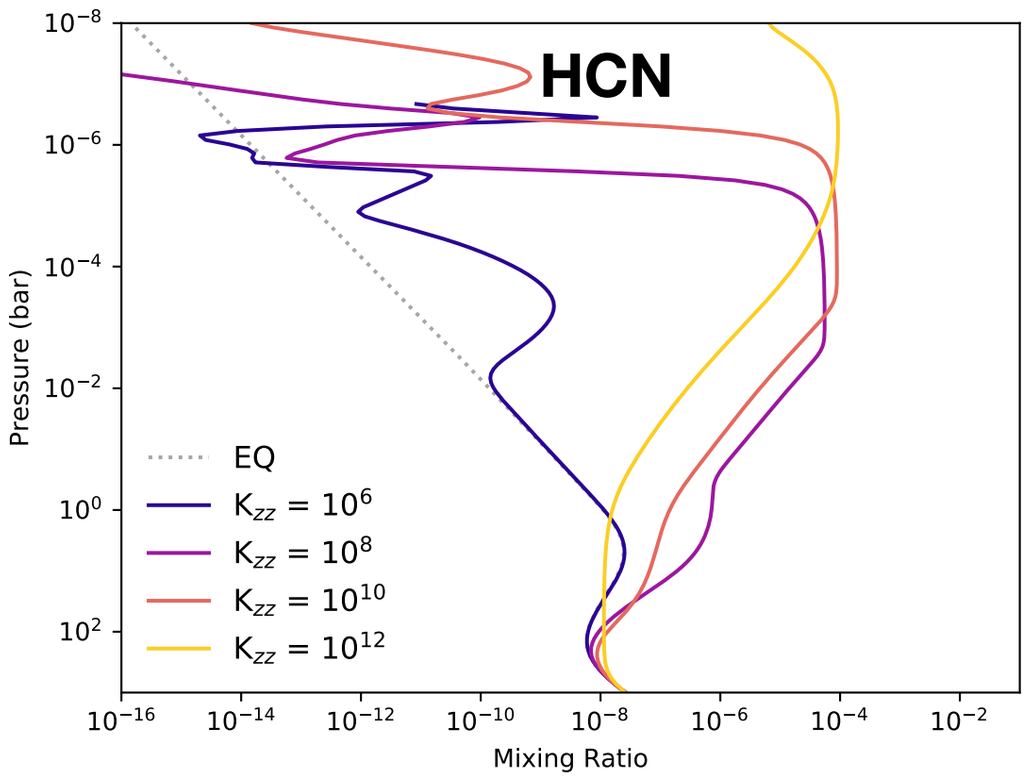
Exoplanet applications

- Determining the atmospheric composition
- Interpreting observations



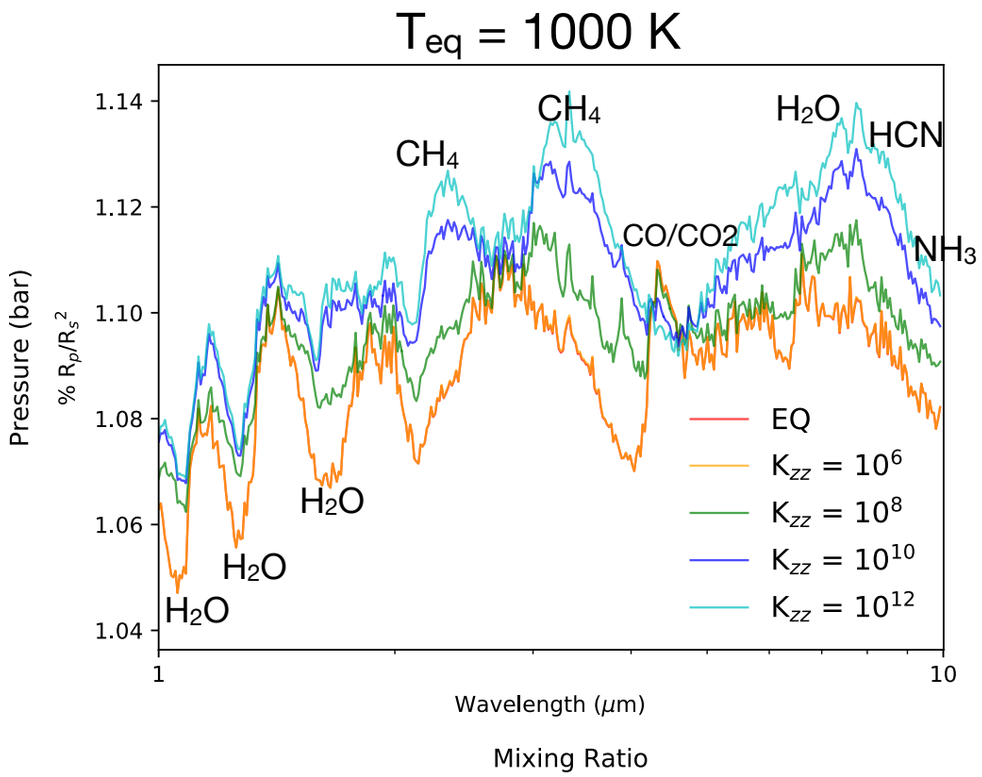
Exoplanet applications

- Determining the atmospheric composition
- Interpreting observations



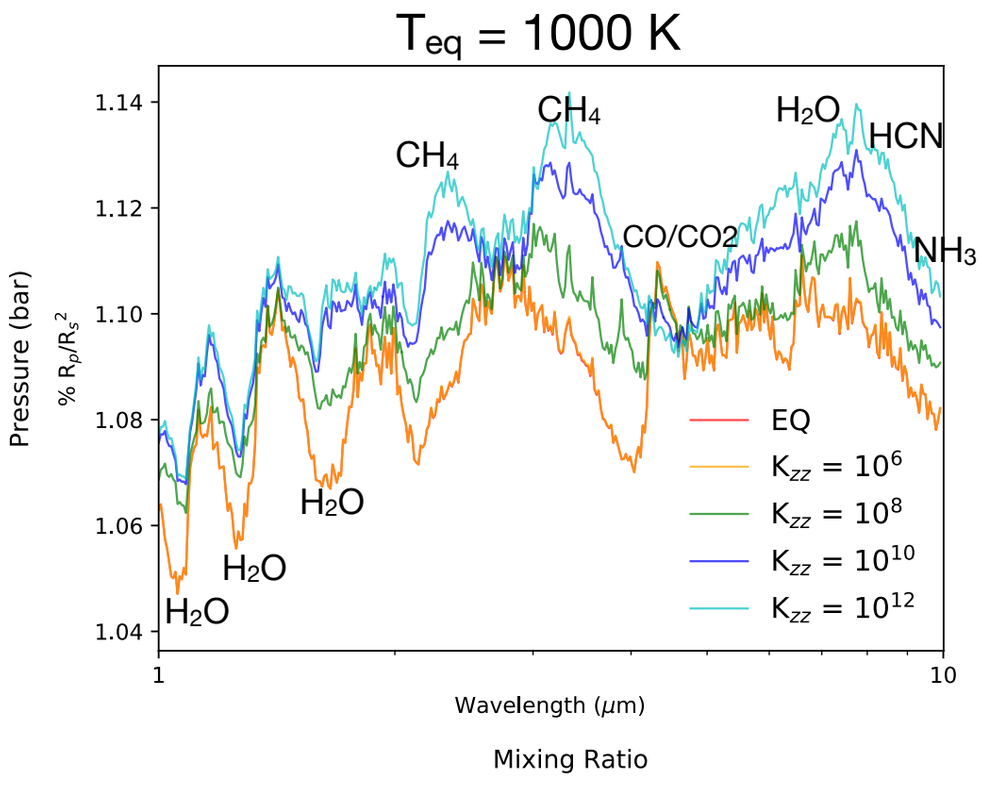
Exoplanet applications

- Determining the atmospheric composition
- Interpreting observations

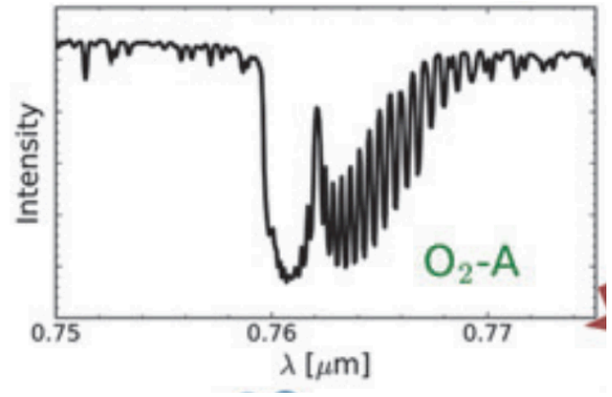


Exoplanet applications

- Determining the atmospheric composition
- Interpreting observations

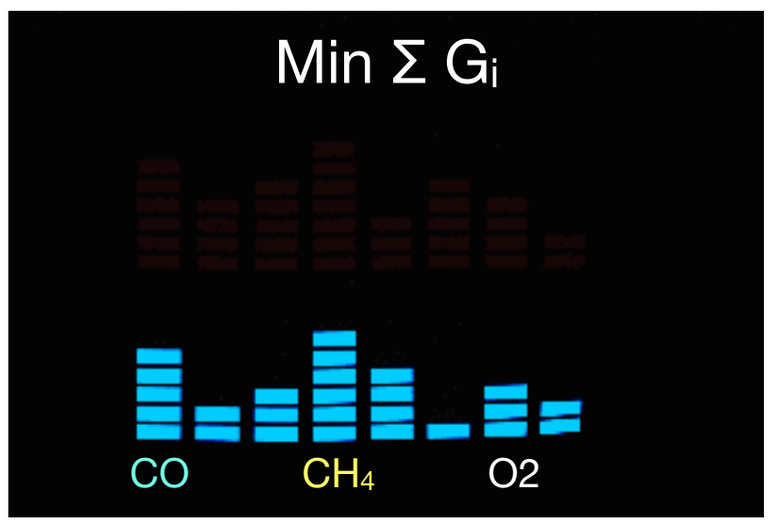


- False-positives of biosignatures



Two classes of chemical models

- Thermochemical equilibrium (**FastChem**)
  — minimising the Gibbs free energy



- Photochemical kinetics
  — reaction rates

$$\sum_i \frac{\partial n_i}{\partial t} + \nabla \phi_i = C_i$$

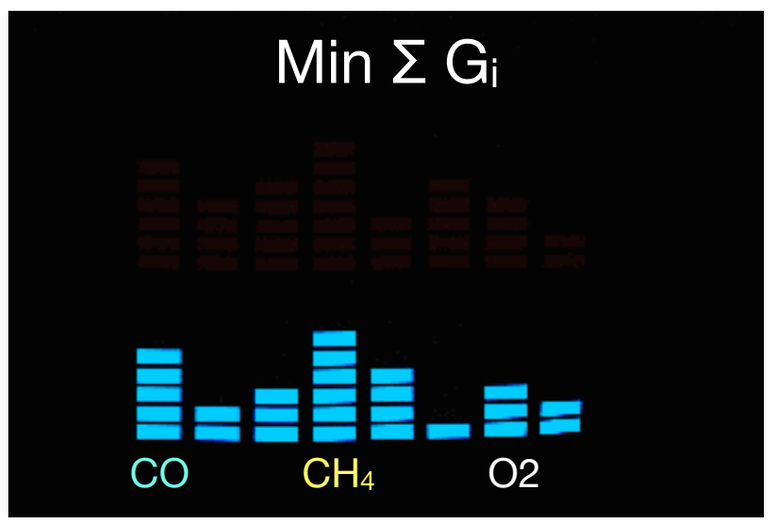
n_i: number density of i species

Φ_i: transport flux C_i: chemical sources

solves a set of coupled ODEs for the steady-state solutions

Two classes of chemical models

- Thermochemical equilibrium (**FastChem**)
  — minimising the Gibbs free energy



- Photochemical kinetics
  — reaction rates

$$\sum_i \frac{\partial n_i}{\partial t} + \nabla \phi_i = C_i$$

n_i: number density of i species

Φ_i: transport flux C_i: chemical sources

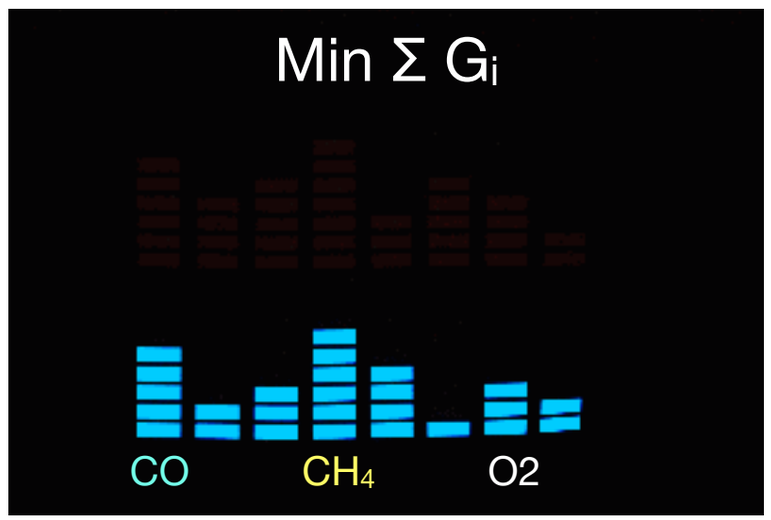
solves a set of coupled ODEs for the steady-state solutions

Two classes of chemical models

- Thermochemical equilibrium (**FastChem**)
  — minimising the Gibbs free energy

  pros: fast, large number of species

  cons: no temporal information



- Photochemical kinetics
  — reaction rates

$$\sum_i \frac{\partial n_i}{\partial t} + \nabla \phi_i = C_i$$

n_i: number density of i species

Φ_i: transport flux C_i: chemical sources

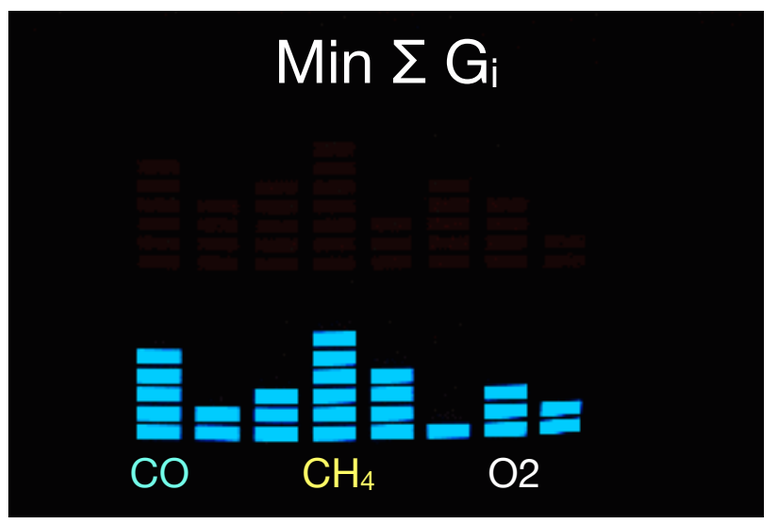
solves a set of coupled ODEs for the steady-state solutions

Two classes of chemical models

- Thermochemical equilibrium (**FastChem**)
  — minimising the Gibbs free energy

  pros: fast, large number of species

  cons: no temporal information



- Photochemical kinetics
  — reaction rates
  e.g. bimolecular reactions
  AB + C <—> AC + B
  e.g. photolysis reactions
  AB + hv —> A + B

  pros: handling disequilibrium processes

  cons: slow, uncertainty in rate coefficients

$$\sum_i \frac{\partial n_i}{\partial t} + \nabla \phi_i = C_i$$

n_i: number density of i species

Φ_i: transport flux C_i: chemical sources

solves a set of coupled ODEs for the steady-state solutions

Thermochemistry

- reversing forward reactions

AB + C —> AC + B

forward rate coefficient

$$\frac{k_f}{k_r} = K_{\mathrm{eq}} \left(\frac{k_B T}{P_0} \right)^{\Delta \mu}$$

“unknown”

The change of standard Gibbs free energy from reactants to products

$$K_{\mathrm{eq}} = \exp\left(-\frac{\Delta G^0}{\mathcal{R} T} \right)$$

Photochemistry

$$AB + h\nu \longrightarrow A + B$$

$$J(z,\lambda) = J_0(z,\lambda) + J_{scat}(z,\lambda)$$

$$J_0(z,\lambda) = \mu J(\infty,\lambda) e^{-\tau(z,\lambda)/\mu}$$

attenuate direct stellar beam

$$J_{scat}(z,\lambda) = \int_{4\pi} I(z,\lambda,\Omega) d\Omega$$

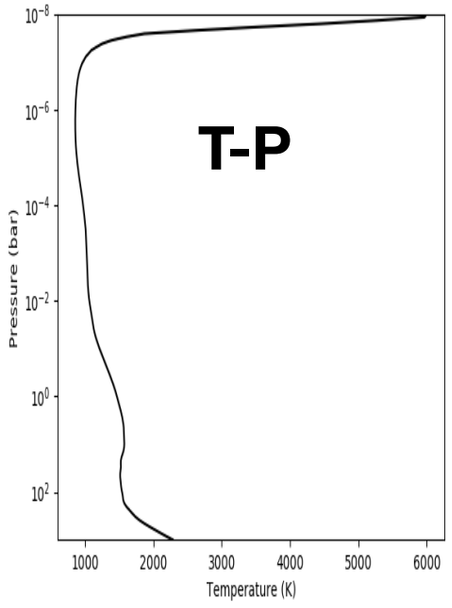
scattered radiation — intensity

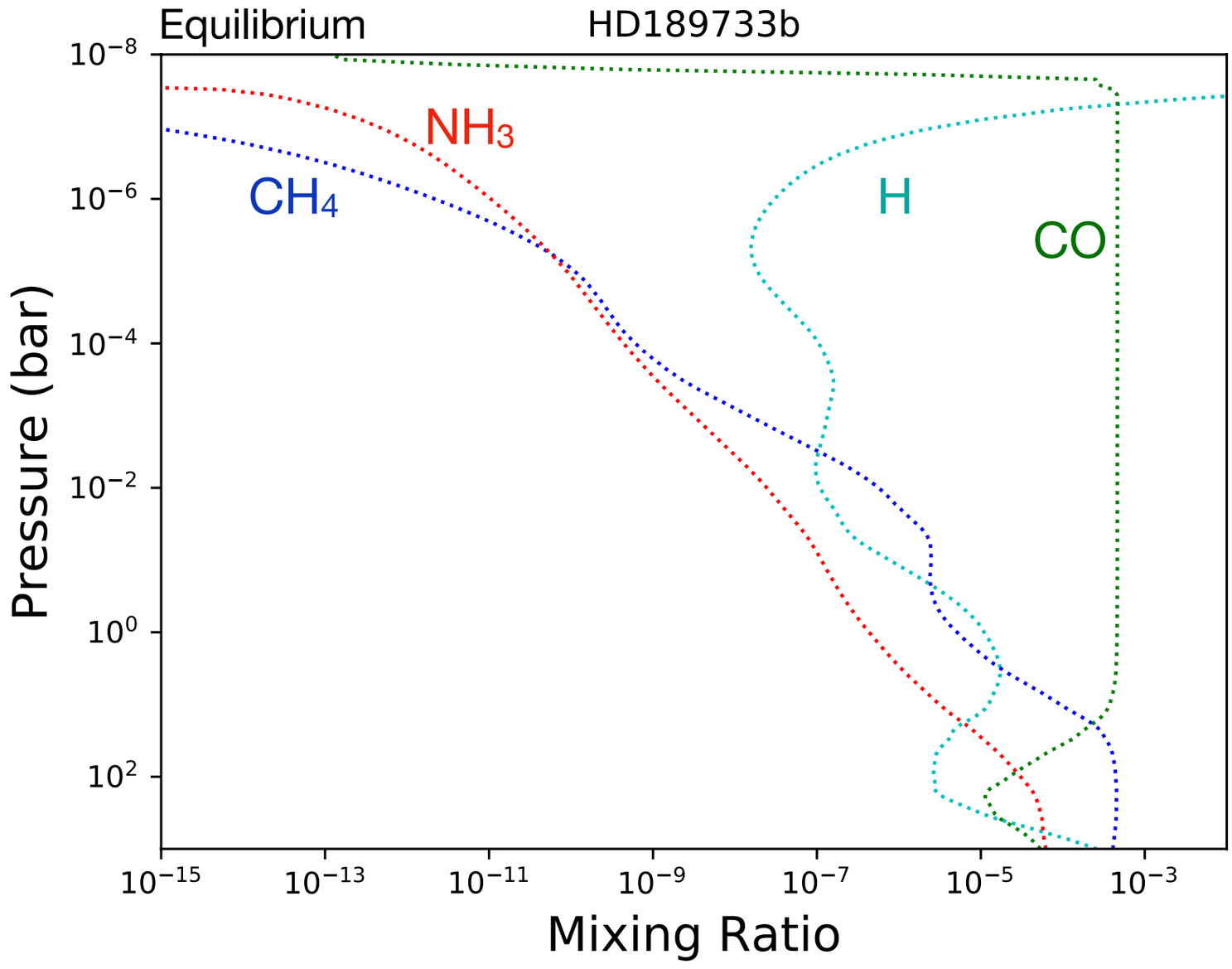
two-stream approximation (Malik et al. 2018) to solve for the scattered flux

$$k = \int_{\lambda} q(\lambda)\sigma_d(\lambda)J(z,\lambda)d\lambda$$

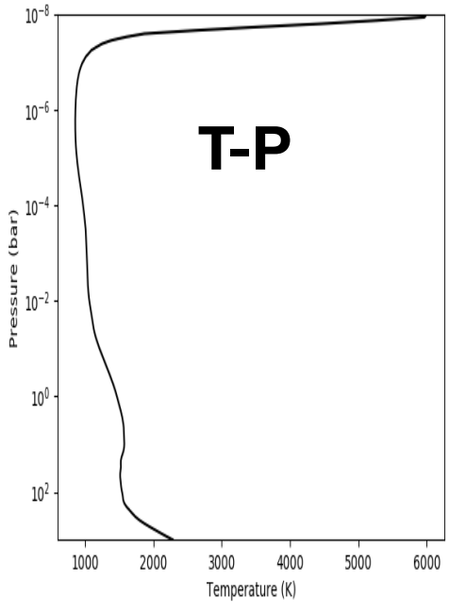
d [AB]/dt = - k [AB] = -d [A]/dt = -d [B]/dt

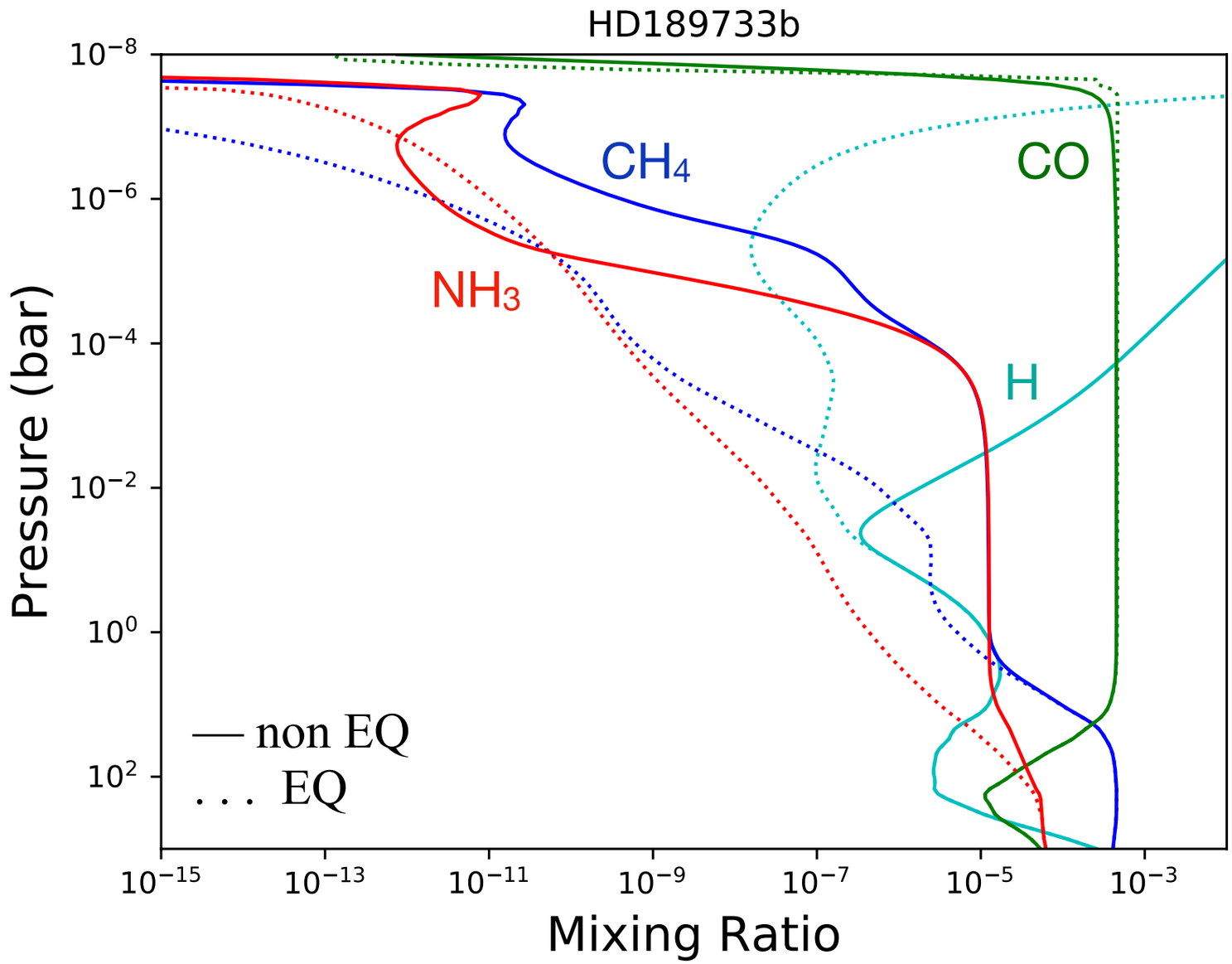
Disequilibrium regimes



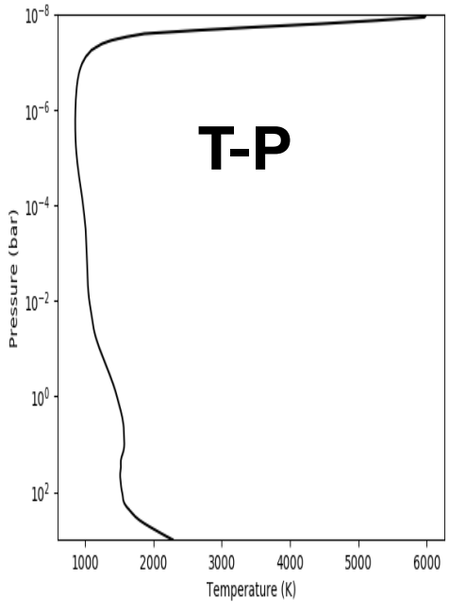


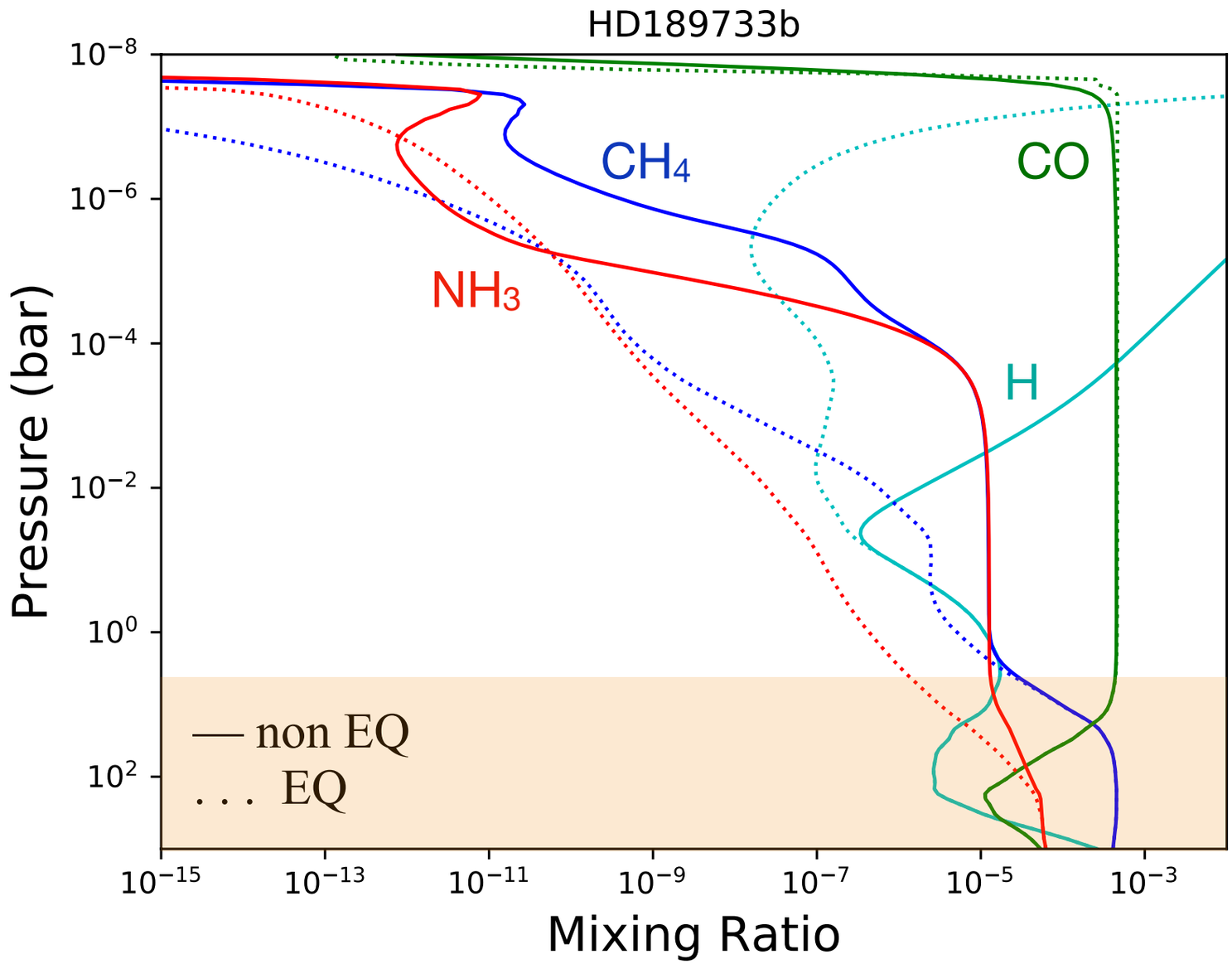
Disequilibrium regimes





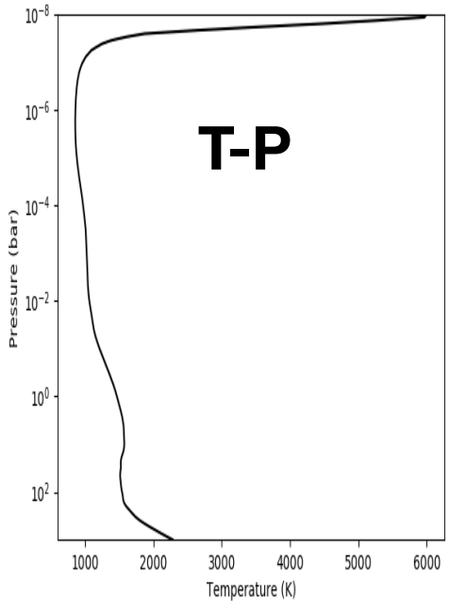
Disequilibrium regimes

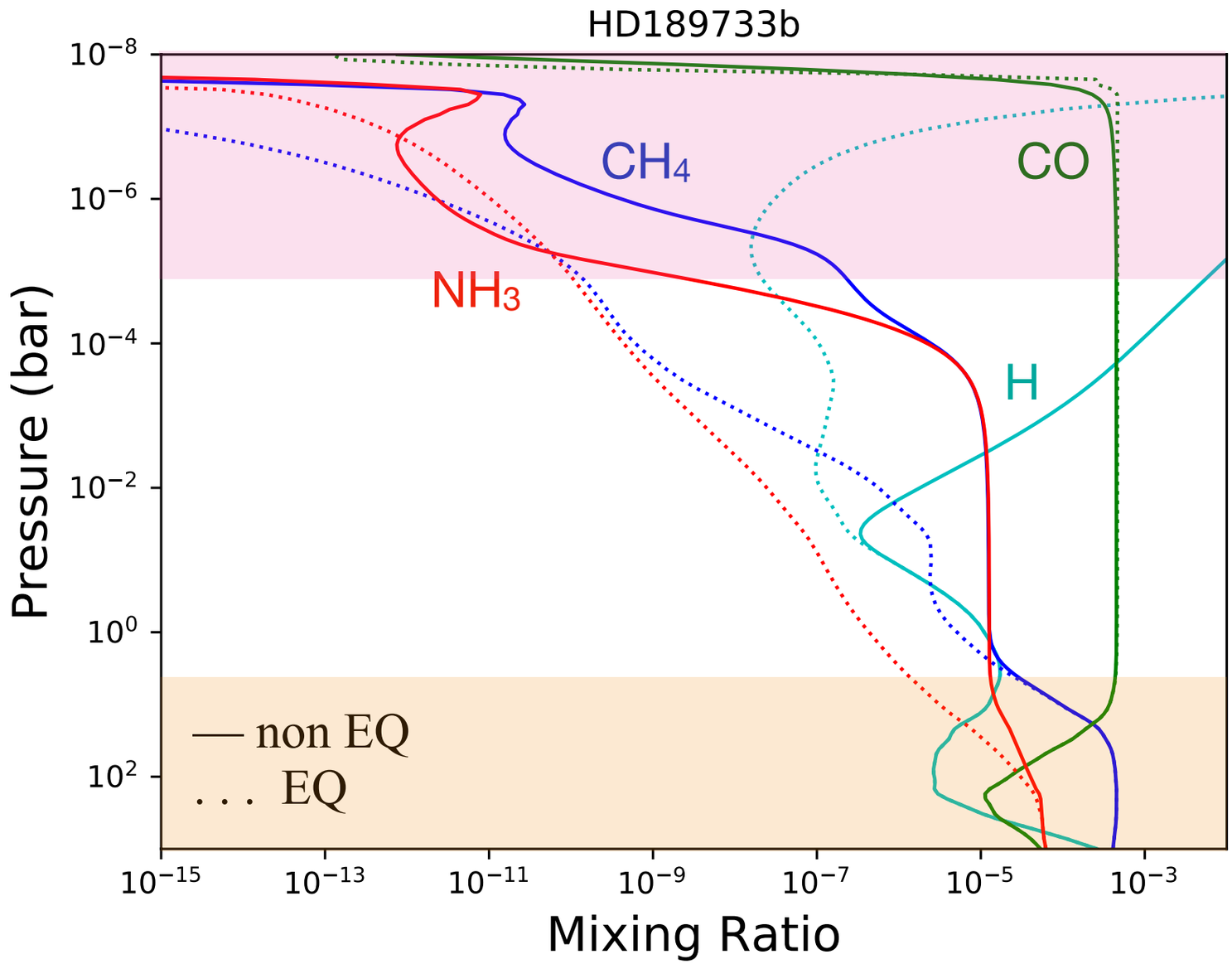




equilibrium
($\tau_{chem} < \tau_{dyn}$)

Disequilibrium regimes

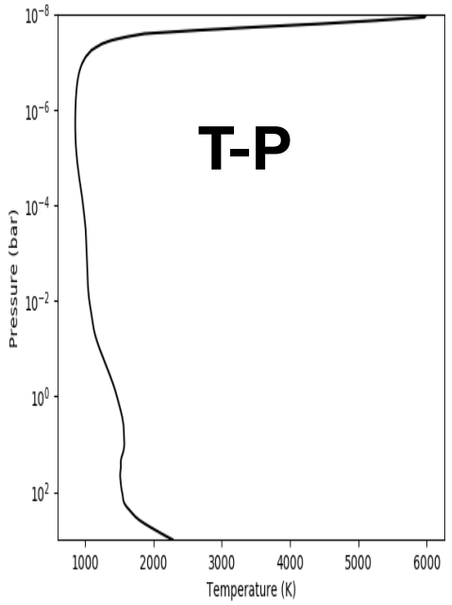


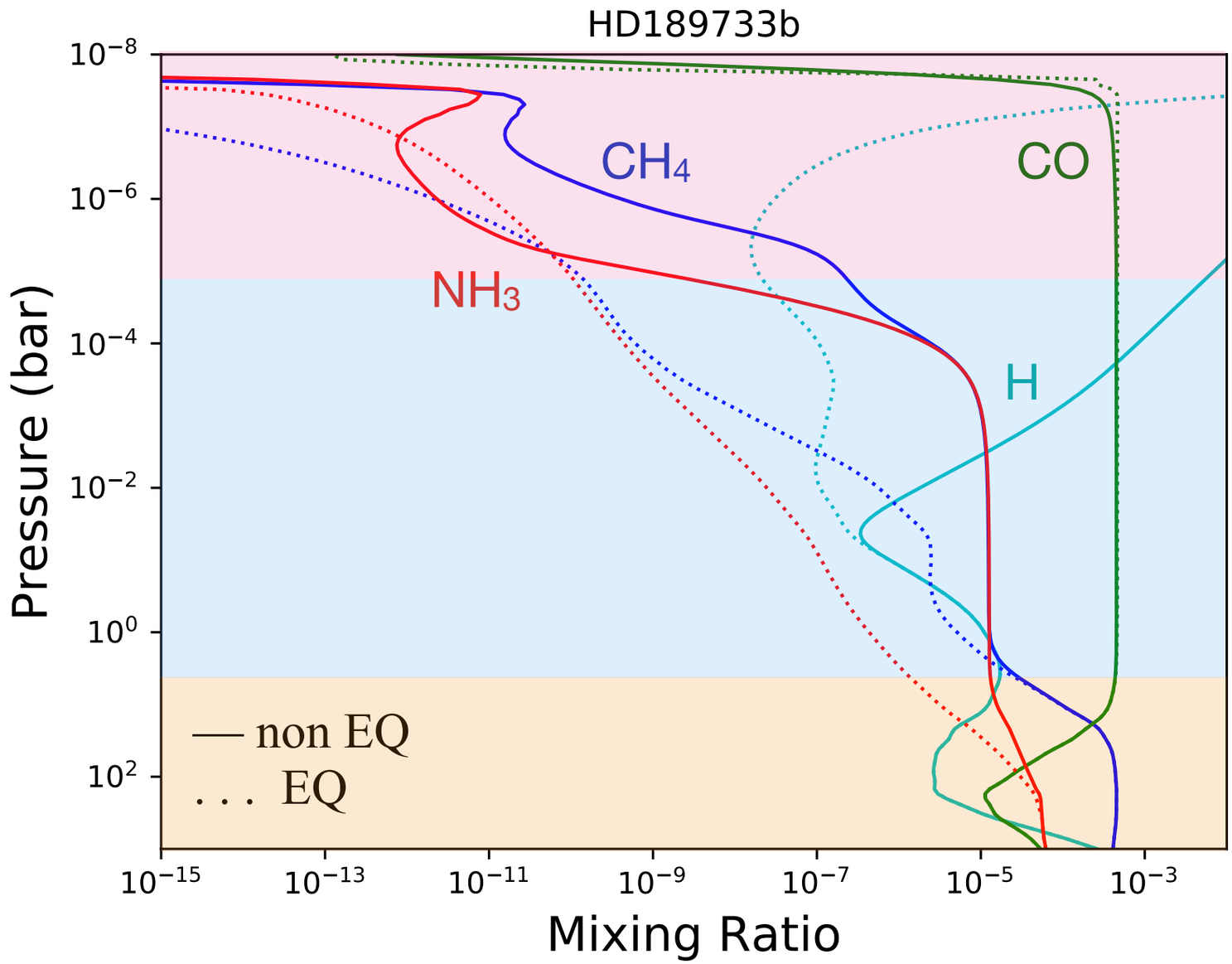


photolysis

equilibrium
($\tau_{chem} < \tau_{dyn}$)

Disequilibrium regimes



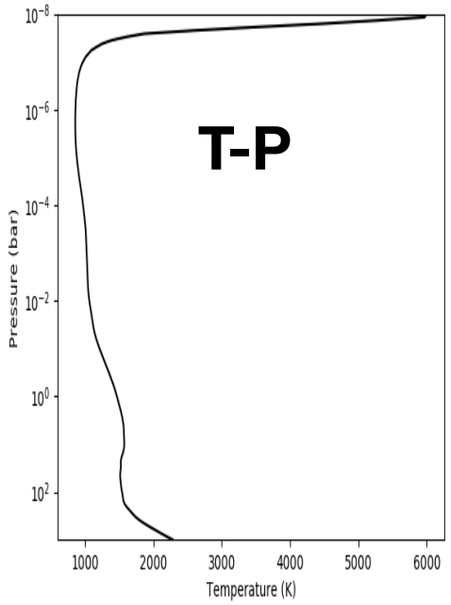


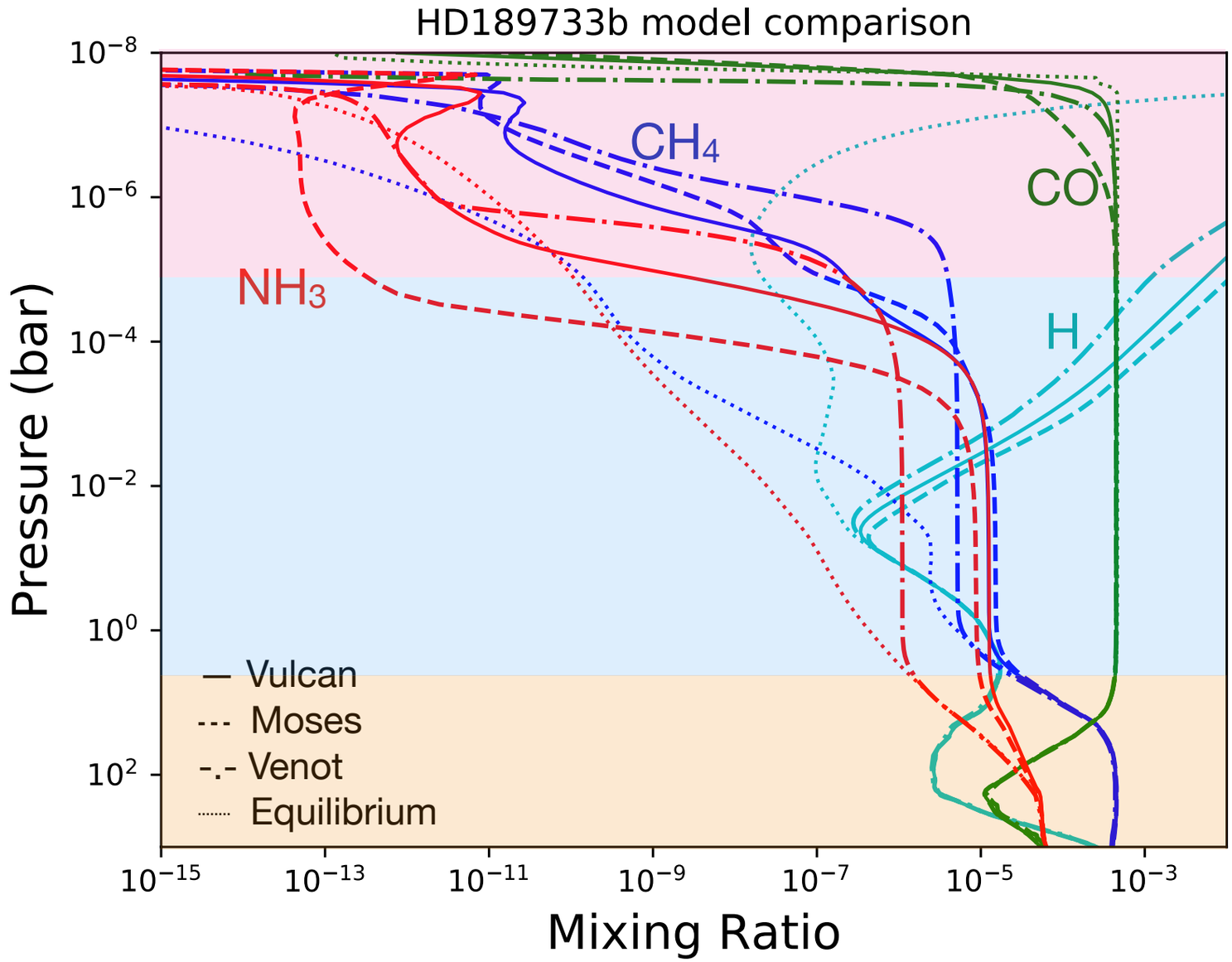
photolysis

vertical mixing
($\tau_{chem} > \tau_{dyn}$)

equilibrium
($\tau_{chem} < \tau_{dyn}$)

Disequilibrium regimes



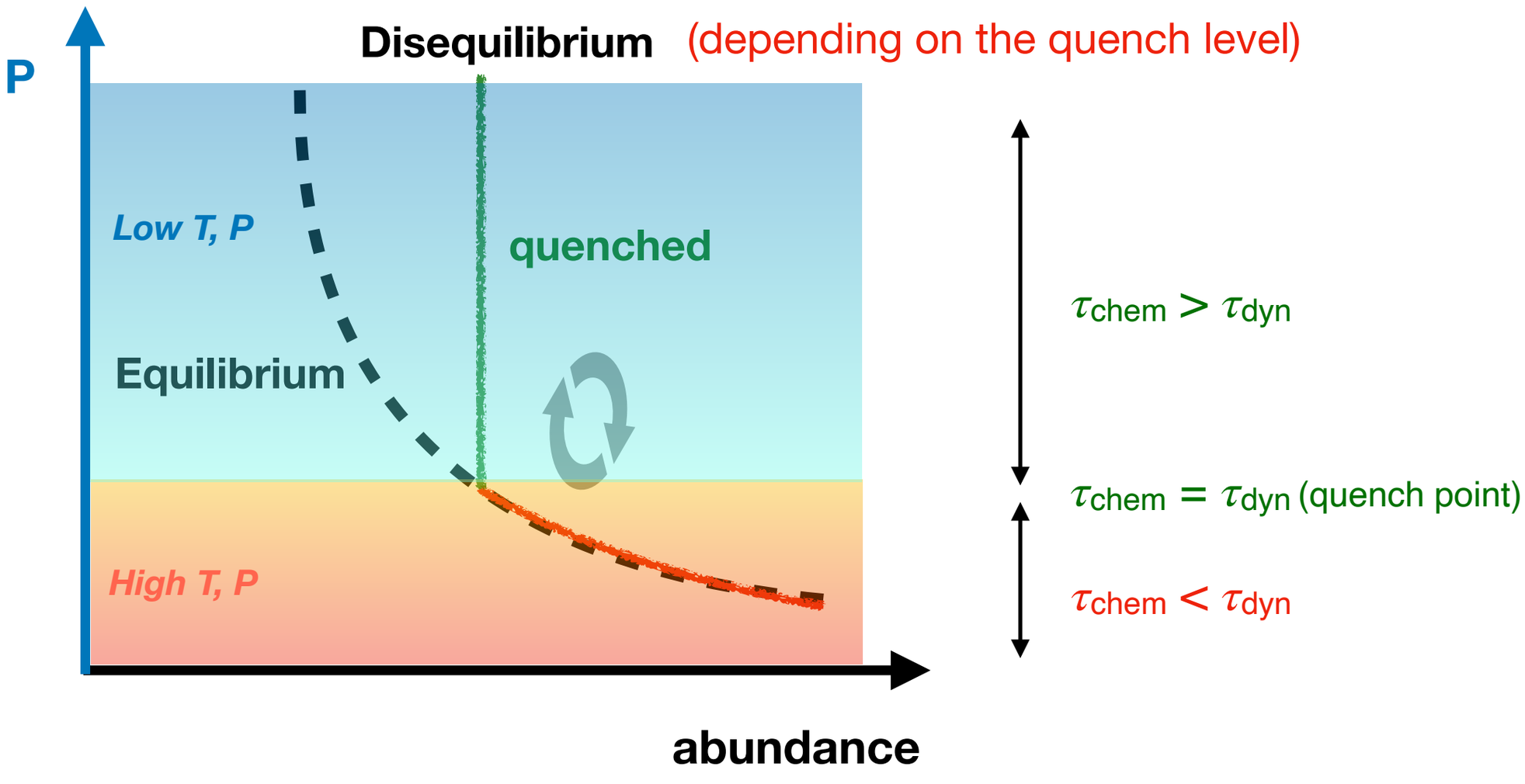


photolysis

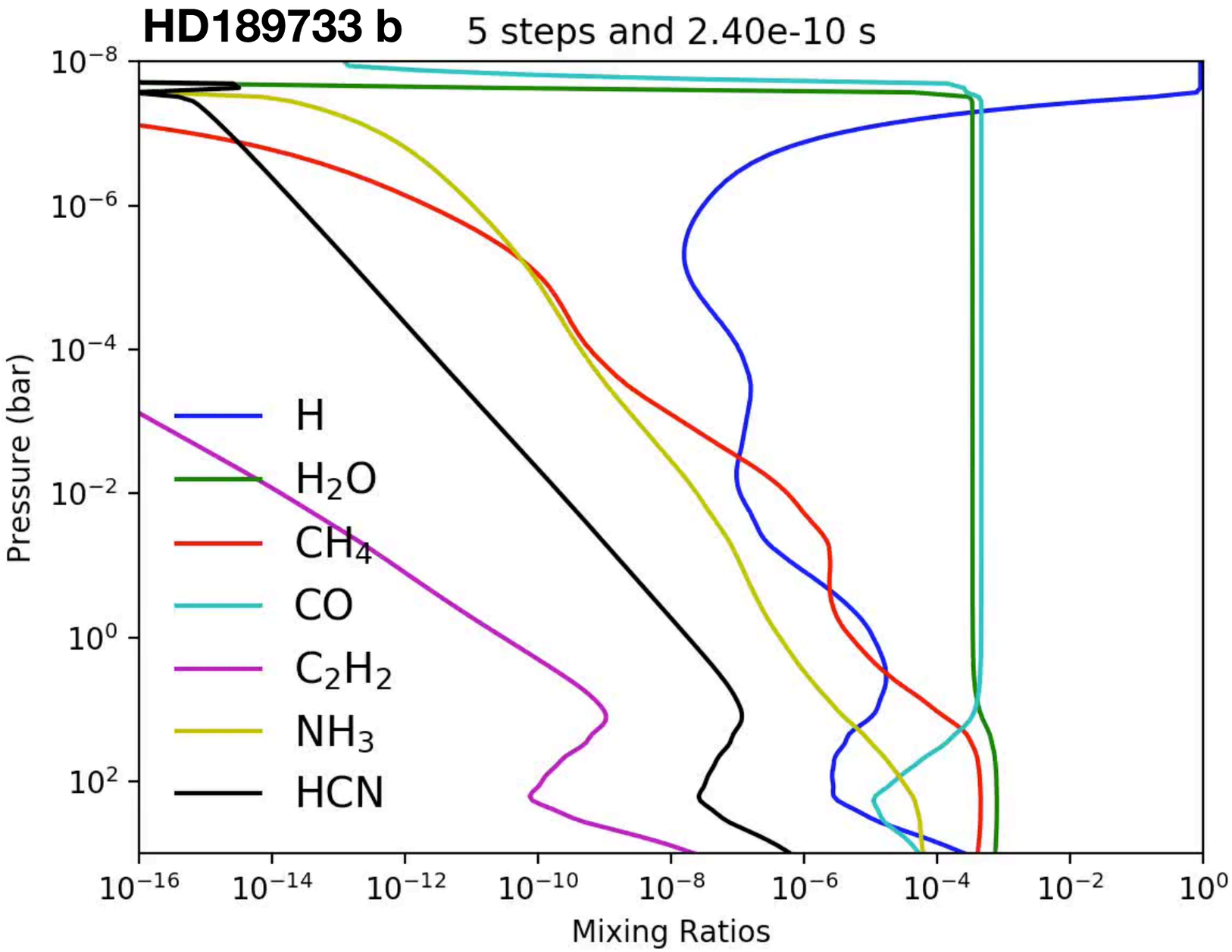
vertical mixing
($\tau_{chem} > \tau_{dyn}$)

equilibrium
($\tau_{chem} < \tau_{dyn}$)

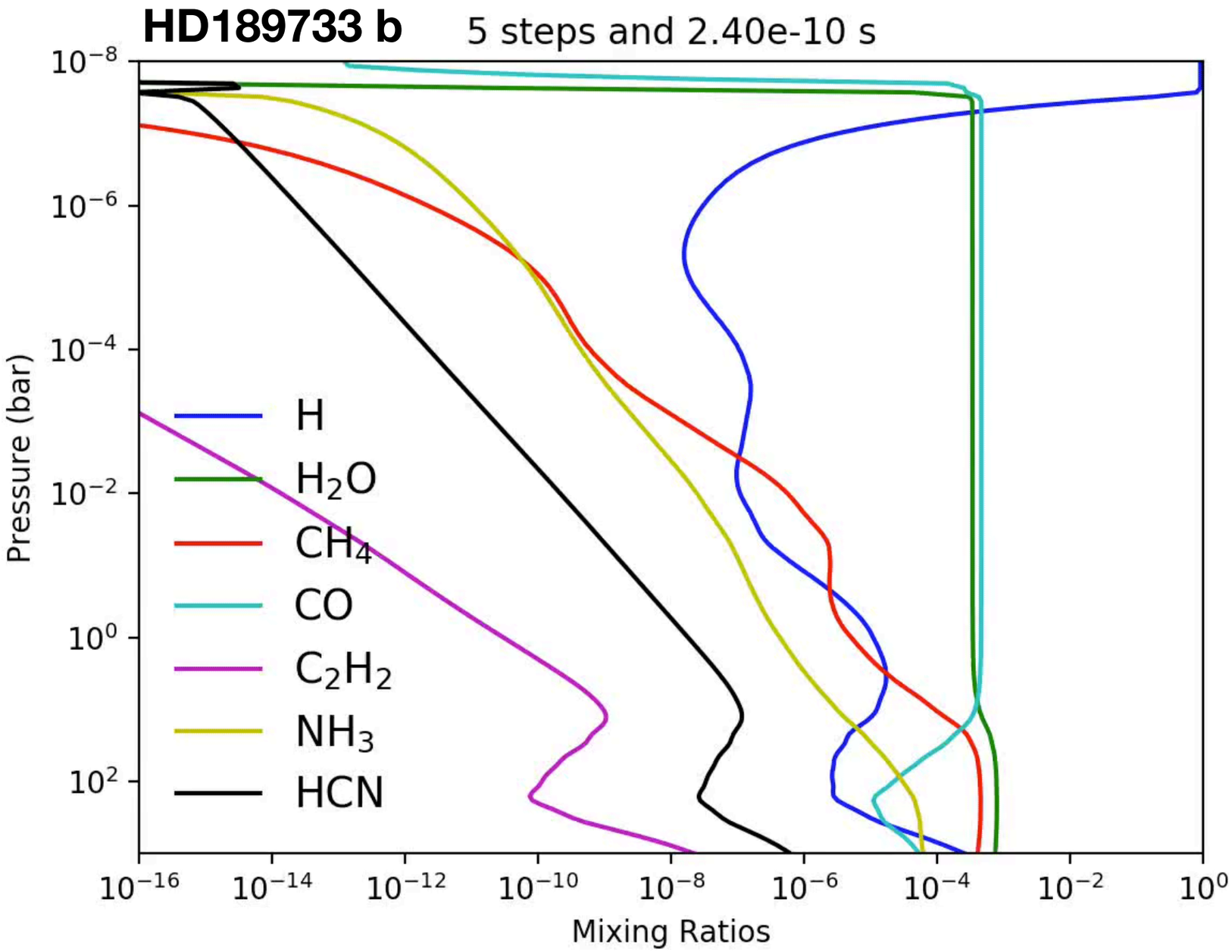
Transport-induced quenching



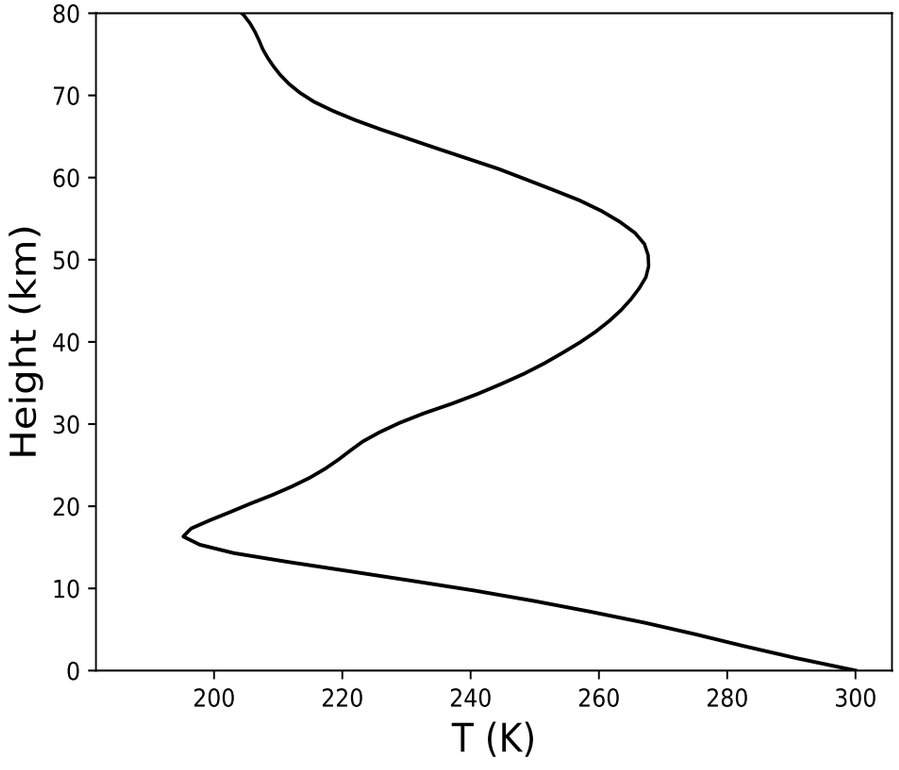
Model integration



Model integration



Earth validation



Bottom boundary (Hu et al 2012):

Fixed H_2O = 0.01

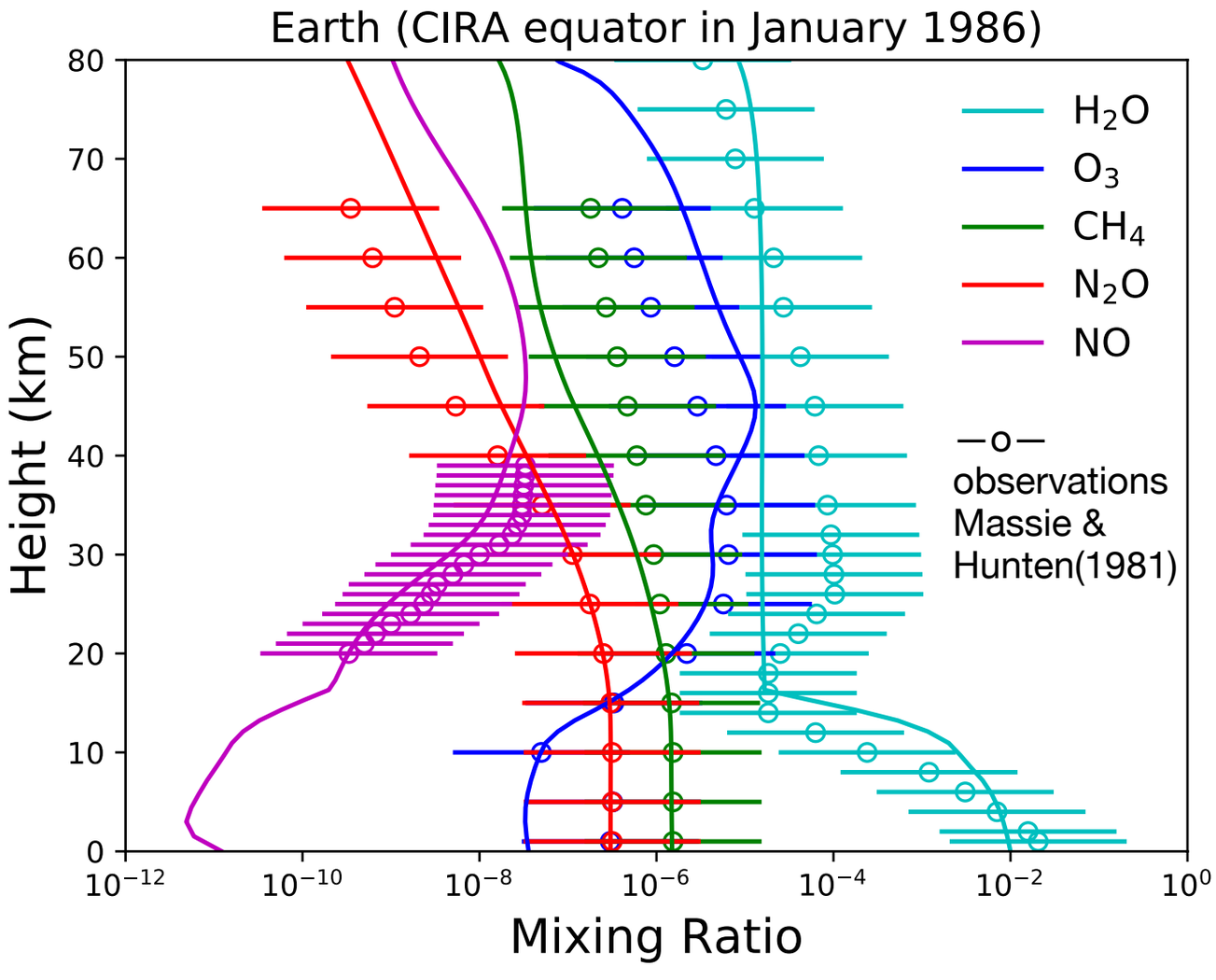
Surface Emission (cm-2 s-1):

CO: 3.7e+11

NH_3: 7.7e+9

NO: 1.0e+9

N_2O: 7.0e+9

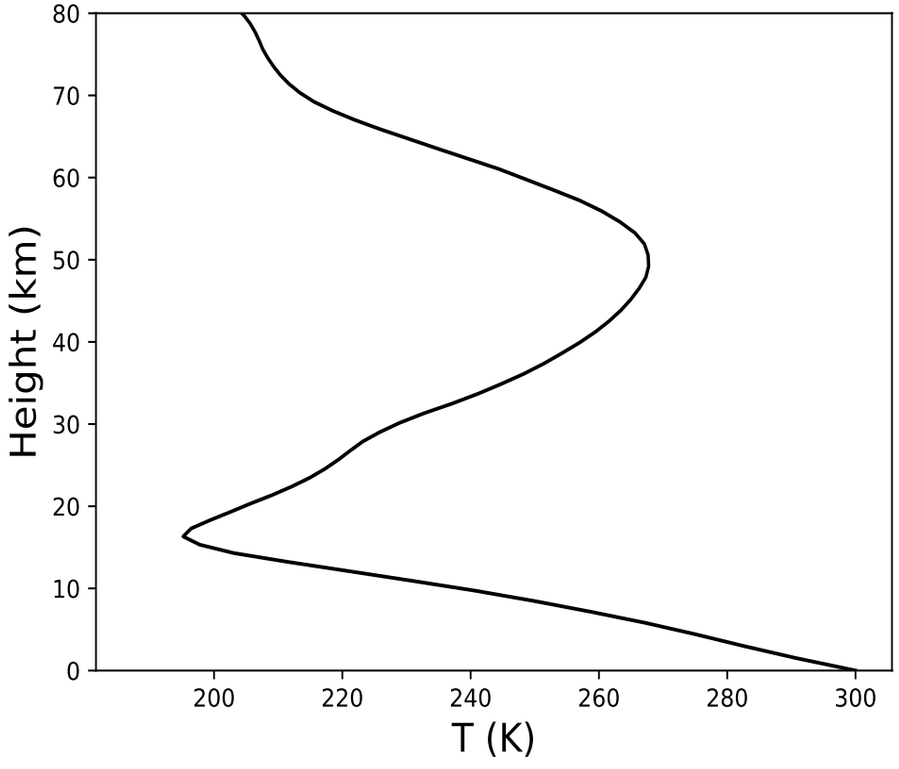


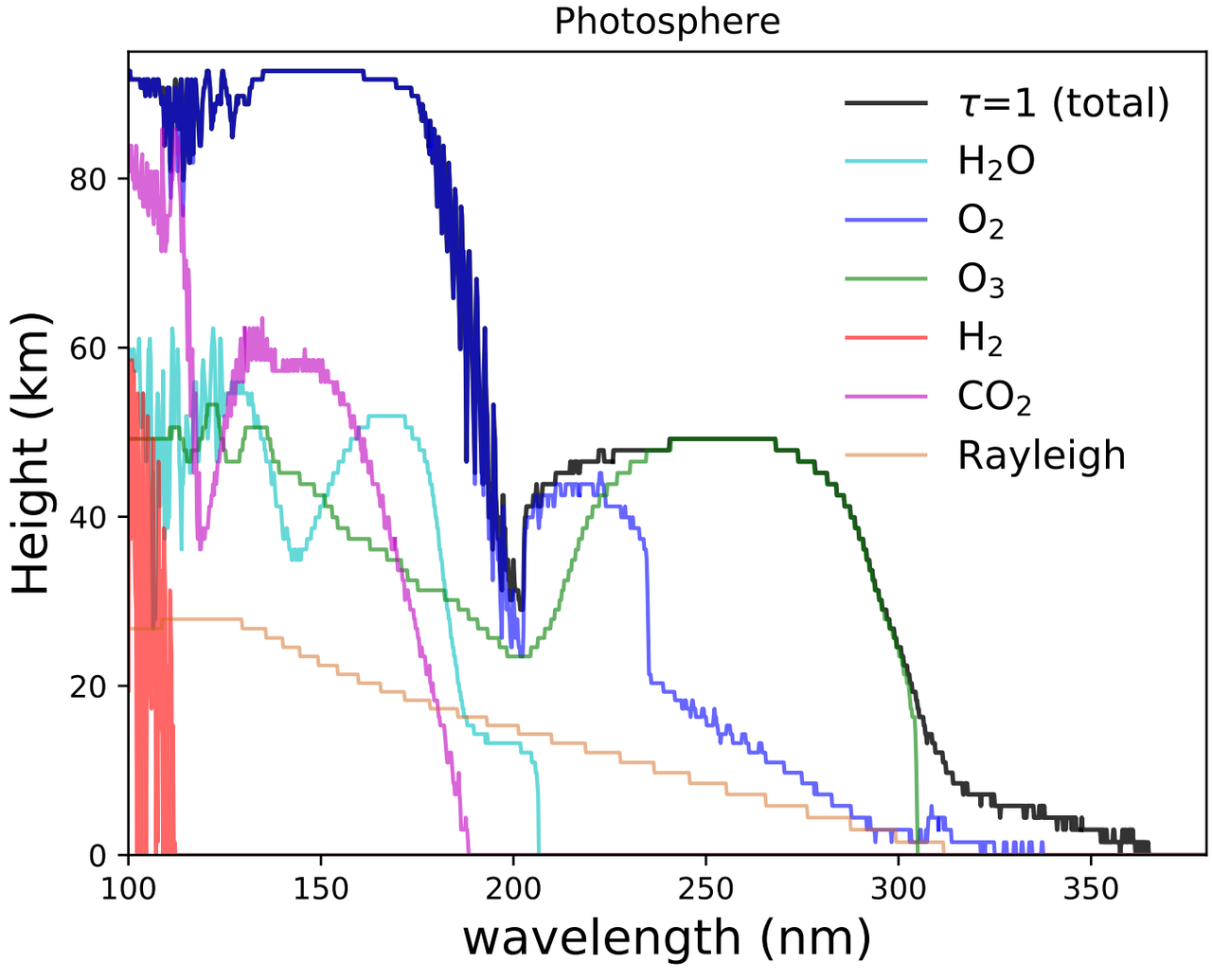
oxidising kinetics

condensation

boundary conditions

Earth validation





Bottom boundary (Hu et al 2012):

Fixed H_2O = 0.01

Surface Emission (cm-2 s-1):

CO: 3.7e+11

NH_3: 7.7e+9

NO: 1.0e+9

N_2O: 7.0e+9

oxidising kinetics

condensation

boundary conditions

Stiff ODE system

—chemical timescales can vary many orders of magnitude

$$\frac{dy_1}{dt} = 998y_1 + 1998y_2$$

$$\frac{dy_2}{dt} = -999y_1 - 1999y_2$$

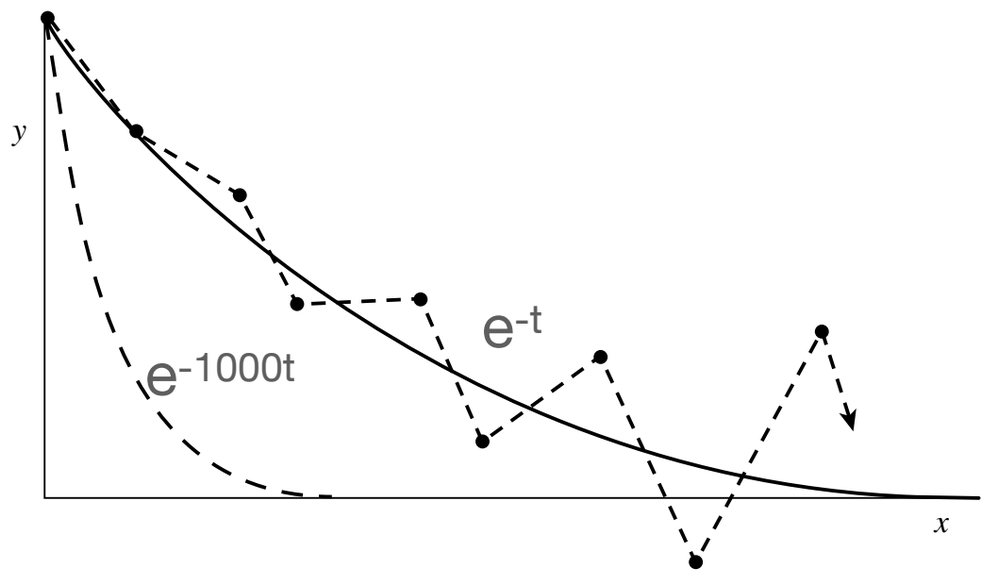
Initial condition: $y_1(0) = 1$, $y_2(0) = 0$

exact solution: $y_1(t) = 2e^{-t} - e^{-1000t}$

$y_2(t) = e^{-t} + e^{-1000t}$

τ = (1, 0.001)

(chemical lifetime)



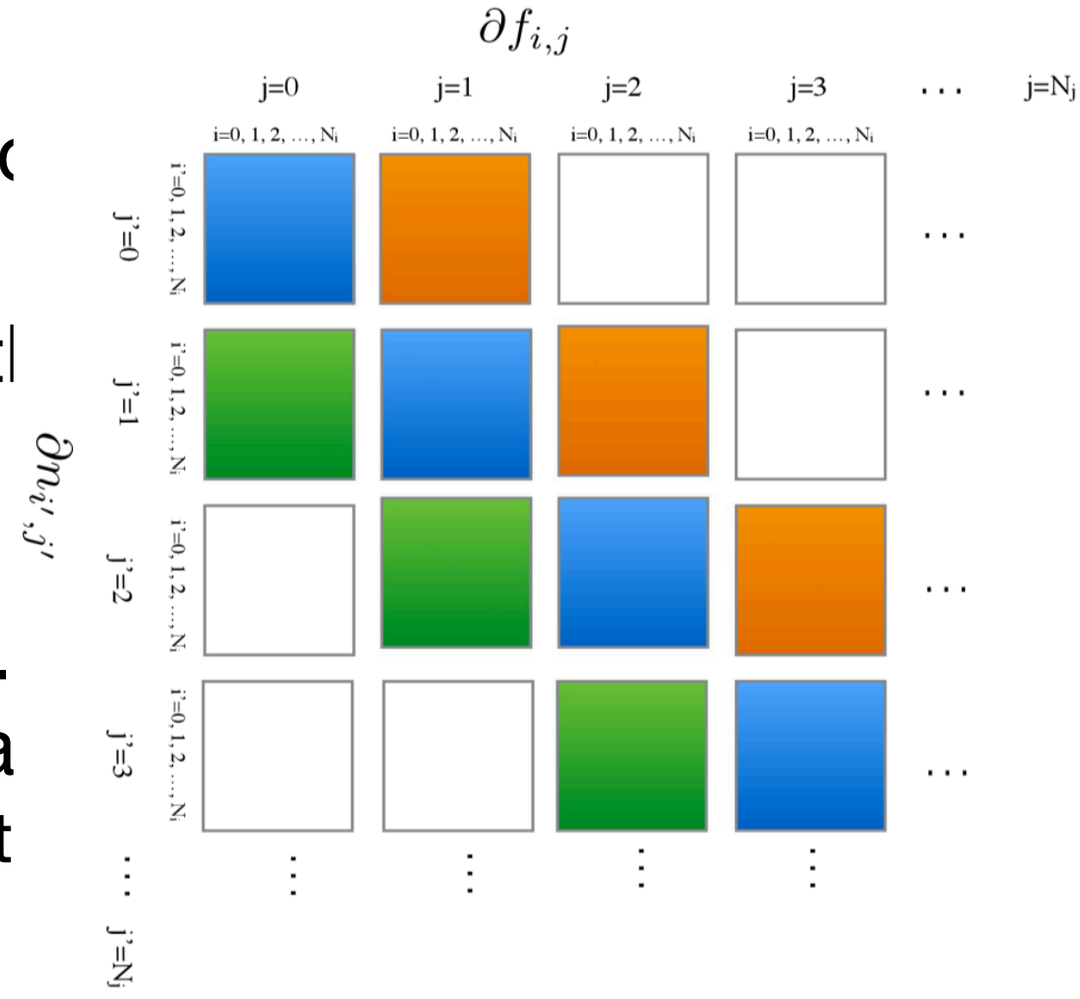
c.f. Numerical Recipies

Code details

- Developed in python 3, runs on CPU
- ODE solver: Rosenbrock method to solve the “stiff” system
- Matrix operation: scipy.linalg.solve_banded to solve Ax=B, where A is formatted as a banded matrix
  scipy.linalg.solve_banded is a python wrapper of LAPACK (Linear Algebra PACKage in Fortran 90)
- Can call FastChem to initialise compositions in chemical equilibrium

Code details

- Developed in python 3, runs c
- ODE solver: Rosenbrock met
  system
- Matrix operation: scipy.linalg.
  Ax=B, where A is formatted a
  scipy.linalg.solve_banded is a pyt
  Algebra PACKage in Fortran 90)



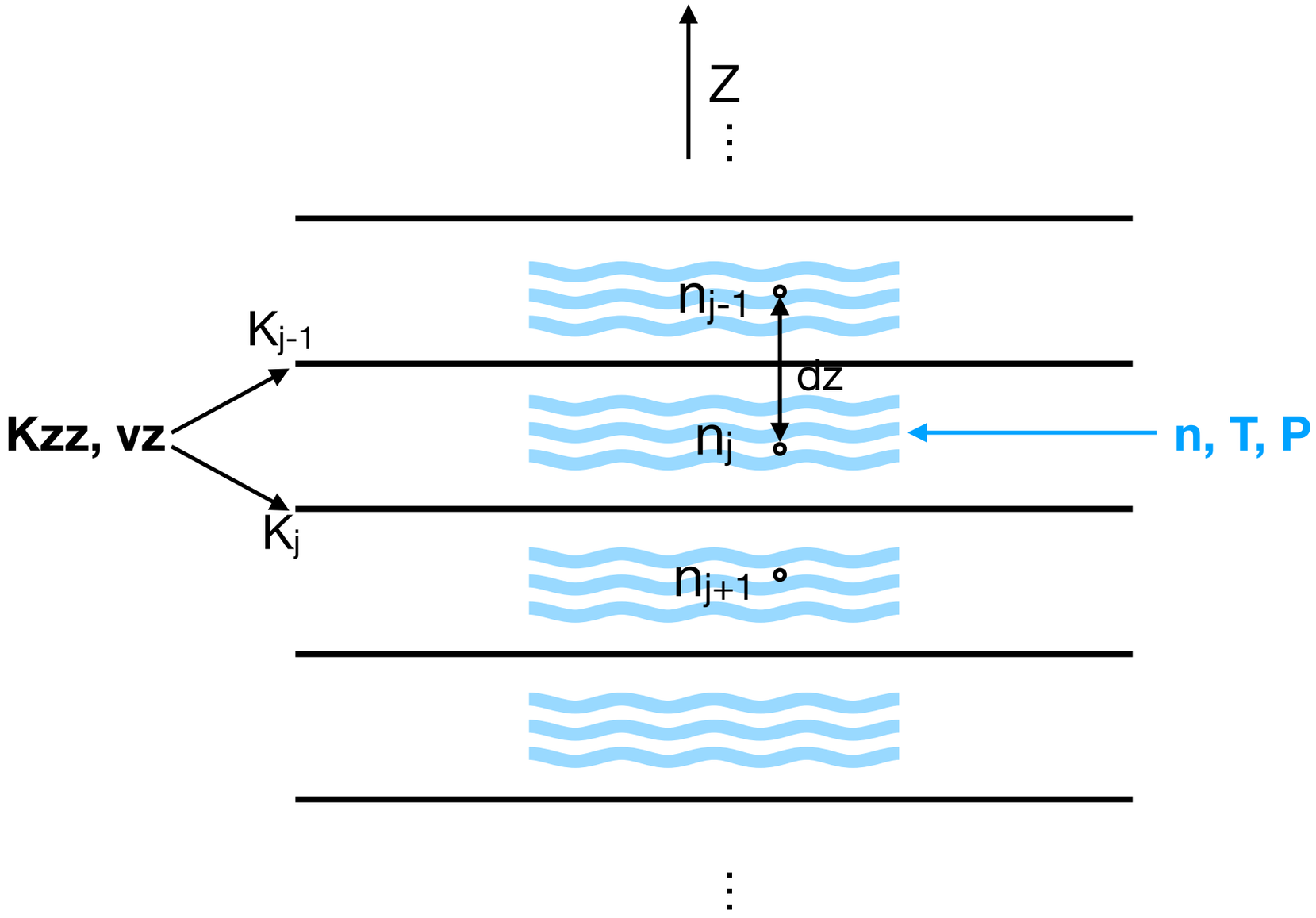
- Can call FastChem to initialise compositions in chemical equilibrium

Code details

- Developed in python 3, runs on CPU
- ODE solver: Rosenbrock method to solve the “stiff” system
- Matrix operation: scipy.linalg.solve_banded to solve Ax=B, where A is formatted as a banded matrix
  scipy.linalg.solve_banded is a python wrapper of LAPACK (Linear Algebra PACKage in Fortran 90)
- Can call FastChem to initialise compositions in chemical equilibrium

Staggered (Orel) grid

Staggered (Orel) grid



Kinetics resources

- Thermodynamics data: NASA polynomials (http://garfield.chem.elte.hu/Burcat/burcat.html)
- Reaction rates: NIST, KIDA
- UV photo cross sections: Leiden database, Phidrates

Installation

[Exoclime-Github]

Demo 0: hot Jupiter (HD189733b)

— Let me set up everything for you

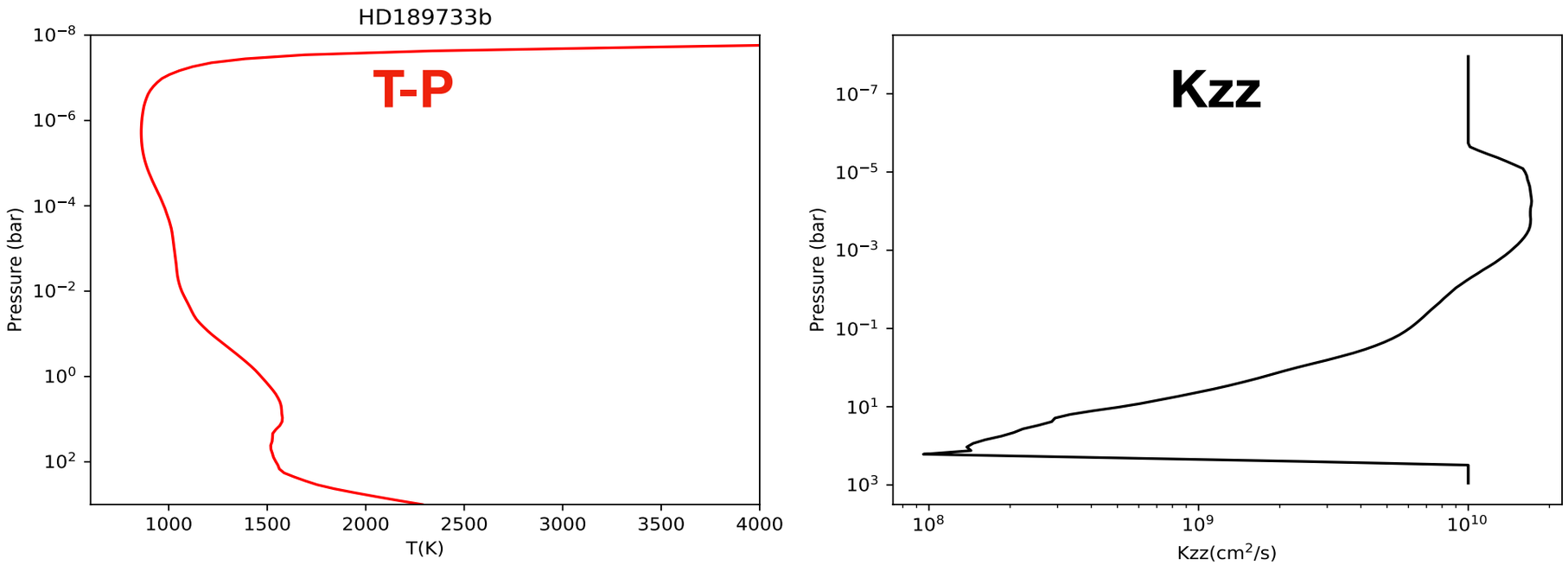
Input:

- Atmospheric structure (T-P, Kzz, g) and boundary conditions
- Stellar UV flux
- Elemental abundance
- Chemical network

Demo 0: hot Jupiter (HD189733b)

— Let me set up everything for you

Input:



Moses et al 2011
(SPARC/MITgcm)

Elemental abundance: ~ solar (Lodders et al. 2009) (O/H = 6.0618E-4 × 80%)
Stellar UV flux: epsilon Eridani (K2 V star) from CoolCAT

Demo 0: hot Jupiter (HD189733b)

—How to read and plot the output

- The numerical results are saved in binary files with pickle

- The configuration is copied into a txt file (*cfg_filename.txt*)

- The dictionary/variable structure can be found in store.py

- Plotting scripts are in plot_py/

  — Run the plotting script `plot_vulcan.py` in the folder `plot_py`, followed by three arguments: {output path} {comma-separated species} {plotname}.For example,

  ```
  python plot_vulcan.py ../output/HD189.vul H2O,CH4,CO,CO2,NH3,HCN HF189
  ```

  — Access numpy arrays

Demo 1: Sanity check

—How to switch off chemistry or mixing

vulcan_cfg_iso.py

- a: No chemical reactions — pure diffusion

  remove_list = list(range(1,285))
  use_Kzz = True

- b: No chemical reactions — pure advection

  remove_list = list(range(1,285))
  use_vz = True

- c: No atmospheric mixing — evolve to chemical equilibrium

  remove_list = [], use_Kzz = False, and use_vz = False

Demo 1: Editing the chemical network

create test_network.txt and add a few ***forward*** reactions

make_chem_funs.py is always
called prior to the main code to produce chem_funs.py,
unless -n is given: python vulcan.py -n

make_chem_funs.py checks:

- Element conservation
- Duplicate reactions

Demo 2: warm Neptune (GJ436b)

—How to change elemental abundances and astronomical parameters

vulcan_cfg_GJ436b.py

- a: setting 100X solar metallicity

  use_solar = False
  *O_H = 6.0618E-4 *100.*
  *C_H = 2.7761E-4 *100.*
  *N_H = 8.1853E-5 *100.*
  He_H = 0.09691

- b: setting C/O = 1

  use_solar = False
  *O_H = 6.0618E-4 *100.*
  C_H = O_H
  *N_H = 8.1853E-5 *100.*
  He_H = 0.09691

- c: initialising a CO2-dominated atmposphere

  ini_mix = 'const_mix'
  remove ‘He’
  const_mix = {'CO2':0.96, 'N2': 0.03, 'O2': 1.3E-3, 'H2O': 3E-4}
  scat_sp = [‘CO2']

- d: setting the zenith angle

  *sl_angle = 48 /180.*3.14159 (day-side)*
  *sl_angle = 80 /180.*3.14159 (terminator)*

  also try varying orbit_radius, g,… etc.

Demo 3: Earth

—How to set up the boundary conditions

- Default: no-flux boundary

- Options to set constant flux (top and bottom) or constant mixing ratio (bottom):

  atm_base = 'N2'
  use_topflux = False
  use_botflux = True
  (see BC_bot_Earth.txt)
  use_fix_sp_bot = {'H2O':0.01}

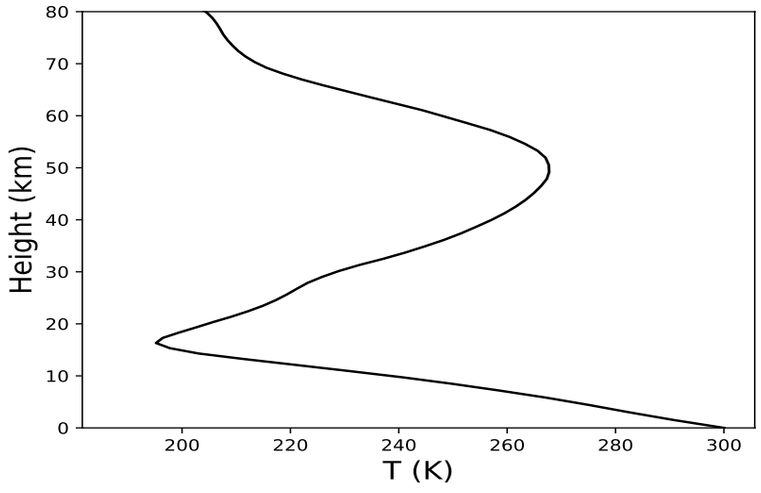
- Condensation of water:

  use_condense = True
  condesne_sp = ["H2O"]
  non_gas_sp = ["H2O_l_s"]

Species	Surface Emission[a] (molecule cm^{-2} s^{-1})	V_{DEP} (cm s^{-1})
CO	3.7×10^{11}	0.03[c]
CH_4	1.4×10^{11}	0.03[c]
NH_3	7.7×10^{9}	1[e]
N_2O	1.0×10^{9}	0[c]
NO	7.0×10^{9}	0.016[c]

H_2O: fixed 1% at the surface

CIRA 1986
(equator January)



Bonus Demo:

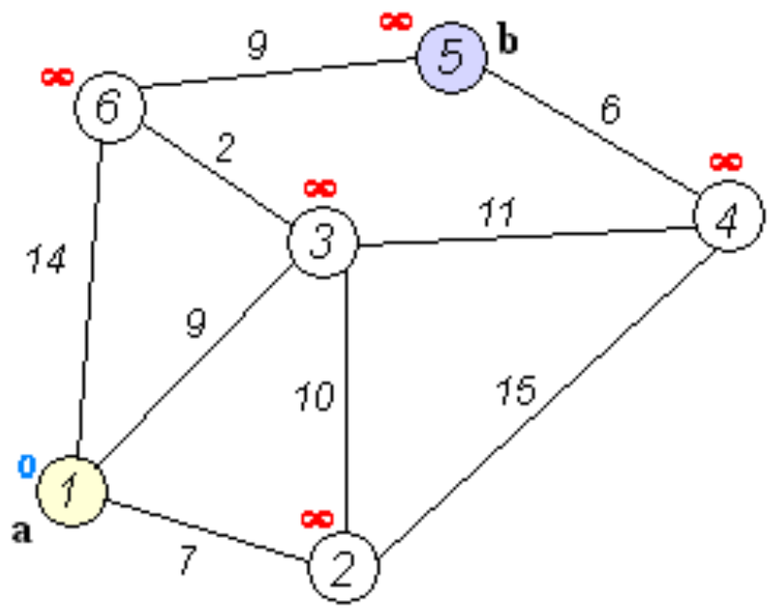
—How to diagnose results

—How to identify the “chemical pathways”

diagonose.py print out fastest reactions

fc_dynamical_Dijkstra.py identify the most efficient pathway and the control step within

(Tsai et al. 2018)



Bonus Demo:

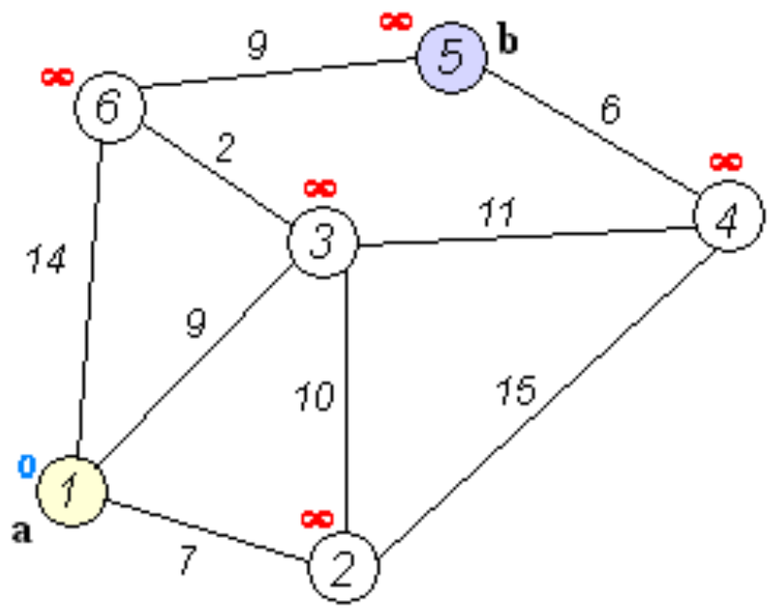
—How to diagnose results

—How to identify the “chemical pathways”

diagonose.py print out fastest reactions

fc_dynamical_Dijkstra.py identify the most efficient pathway and the control step within

(Tsai et al. 2018)



Bonus Demo:

—How to diagnose results

—How to identify the “chemical pathways”

diagonose.py print out fastest reactions

fc_dynamical_Dijkstra.py identify the most efficient pathway and the control step within

(Tsai et al. 2018)

Google Maps