ExoPlex

Release 0.9

Cayman Unterborn

Aug 17, 2020

CONTENTS

CHAPTER

ONE

DOCUMENTATION FOR THE CODE

1.1 Introduction

Too be written

1.2 Modules

1.2.1 make_planet.py

1.2.2 planet.py

ExoPlex.planet.**compress_mass**(**args*)

This module iterates the density, mass within a sphere, adiabatic temperature and gravity integrals for a planet of Mass M until convergence is reached. Convergence is defined as the change from the previous run to the current is the difference in the density of all layers is <1e-6.

Parameters

Planet: dictionary

Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

Core_wt_per: float Composition of the Core

structural_params: list Structural parameters of the planet; See example for description

layers: list Number of layers for core, mantle and water

Planet: dictionary Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet

ExoPlex.planet.**compress_radius**(**args*)

This module iterates the density, mass within a sphere, adiabatic temperature and gravity integrals for a planet of radius R until convergence is reached. Convergence is defined as the change from the previous run to the current is the difference in the density of all layers is <1e-6.

Parameters

Planet: dictionary

Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

Core_wt_per: float Composition of the Core

structural_params: list Structural parameters of the planet; See example for description

layers: list Number of layers for core, mantle and water

Planet: dictionary Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet

ExoPlex.planet.**initialize_by_mass**(**args*)

This module creates the dictionary of lists for each planetary parameter (e.g., density) for a planet of the mass input by user.

Parameters

mass_planet: float

input radius of planet in Earth radii

structural_params: list Structural parameters of the planet; See example for description

compositional_params: list Structural parameters of the planet; See example for description

layers: list Number of layers for core, mantle and water

Planet: dictionary Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet

ExoPlex.planet.**initialize_by_radius**(**args*)

This module creates the dictionary of lists for each planetary parameter (e.g., density) for a planet of the radius input by user.

Parameters

radius_planet: float

input radius of planet in Earth radii

structural_params: list Structural parameters of the planet; See example for description

compositional_params: list Structural parameters of the planet; See example for description

layers: list Number of layers for core, mantle and water

Planet: dictionary Dictionary of initial guess of pressure, temperature, expansivity, specific heat and phases for modeled planet keys = 'radius','density','temperature','gravity','pressure', 'alpha','cp','Vphi"Vp','Vs','K'

1.2.3 run_perplex.py

ExoPlex.run_perplex.**run_perplex**(**args*)

This module runs PerPlex to produce mantle phase diagrams for a custom composition if the user opts to not use the premade grids.

Parameters

Mantle_wt_per [dictionary]

composition of mantle in oxides $[wt\%]$

compositional_params: list Structural parameters of the planet; See example for description

structural_params: list Structural parameters of the planet; See example for description

filename: string chosen filename for output file

UMLM: boolean True if creating grid for upper mantle, false if lower mantle

Returns

Phase diagram: file Mantle phase diagram for the chosen composition. Contains P, T, expansivity, density and specific heat. Stored in /Calc_Solutions/'+filename, where filename is the chosen user name

1.2.4 minphys.py

ExoPlex.minphys.**check_convergence**(*new_rho*, *old_rho*)

This module checks convergence by comparing the density layers of the previous and current runs

Parameters

new_rho: list densities calculated in current iteration

old_rho: list densities calculated in previous iteration

Returns

———-

converged: boolean True if converged

new_rho: list densities of current iteration

ExoPlex.minphys.**get_core_rho**(*Pressure*, *Temperature*, *Core_wt_per*)

This module calculates the density of the core given a list of pressures and temperatures and the light element composition of the core. See documentation for description of how light elements are treated.

Parameters

pressure: list

List of pressures to evaluate for density [bar]

temperature: list List of temperatures to evaluate for density [bar]

Returns

density: list list of calculated density of the core [kg/m^3]

ExoPlex.minphys.**get_gravity**(*Planet*, *layers*)

This module calculates the gravity profile of the planet using Gauss' Law of gravity given the density and radius lists. Parameters ————- Planet: dictionary

Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

layers: list Number of layers for core, mantle and water

Returns

gravity: list list of gravities in each shell for water, mantle and core layers [kg/m^2]

ExoPlex.minphys.**get_mass**(*Planet*, *layers*)

This module calculates the mass of each individual shell profile of the planet using the equation for mass within a sphere given the density and radius lists. Parameters ————- Planet: dictionary

Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

layers: list Number of layers for core, mantle and water

Returns

mass: list list of masses of each shell for water, mantle and core layers [kg/m^2]

ExoPlex.minphys.**get_pressure**(*Planet*, *layers*)

This module calculates the pressure profile of the planet using the equation for hydrostatic equilibrium given the density and radius lists. Parameters ————- Planet: dictionary

Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

layers: list Number of layers for core, mantle and water

Returns

pressure: list list of pressures in each shell for water, mantle and core layers [bar]

ExoPlex.minphys.**get_radius**(*Planet*, *layers*)

This module calculates the radius of each individual shell profile of the planet using the equation for density given the density and mass lists. Parameters ————- Planet: dictionary

Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

layers: list Number of layers for core, mantle and water

Returns

radius: list list of radius of each shell for water, mantle and core layers [kg/m^2]

ExoPlex.minphys.**get_rho**(*Planet*, *grids*, *Core_wt_per*, *layers*)

This module stitches together the lower and upper mantle grids and interpolates within them to determine the density of each material in each layer in the planet. It also calculates the density of the core for those pressures and temperatures within the core layers.

Parameters

Planet: dictionary

Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

Core_wt_per: float Composition of the Core

layers: list Number of layers for core, mantle and water

Returns

rho_layers: list list of densities for water, mantle and core layers [kg/m^3']

ExoPlex.minphys.**get_temperature**(*Planet*, *grids*, *structural_parameters*, *layers*)

This module calculates the adiabatic temperature profile of the planet using the equation for an adiabat given the density, gravity, pressure and radius lists as well as the grids of expansivity and specific heat at constant pressure.

Parameters

Planet: dictionary Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

structural_params: list Structural parameters of the planet; See example for description

layers: list Number of layers for core, mantle and water

Returns

temperature: list list of temperatures in each shell for water, mantle and core layers [kg/m^2]

ExoPlex.minphys.**get_water_Cp**(*Pressure*, *Temperature*)

This module calculates the specific heat at constant pressure of water (either as an ice or liquid) given a single pressure and temperature

Parameters

pressure: float

pressure to evaluate for Cp [bar]

temperature: list temperature to evaluate for Cp [bar]

Returns

Cp: float Calculated specific heat [J/(kg*K)]

ExoPlex.minphys.**get_water_alpha**(*Pressure*, *Temperature*)

This module calculates the thermal expansivity of water (either as an ice or liquid) given a single pressure and temperature

Parameters

pressure: float

pressure to evaluate for Cp [bar]

temperature: list temperature to evaluate for Cp [bar]

Returns

alpha: float Calculated thermal expansivity [1/K]

ExoPlex.minphys.**get_water_rho**(*Pressure*, *Temperature*)

This module calculates the density of water (either as an ice or liquid) given a list of pressures and temperatures

Parameters

pressure: list

List of pressures to evaluate for density [bar]

temperature: list List of temperatures to evaluate for density [bar]

Returns

density: list list of calculated density of water [kg/m^3]

1.2.5 functions.py

ExoPlex.functions.**find_Planet_mass**(*mass_planet*, *core_mass_frac*, *structure_params*, *compositional_params*, *grids*, *Core_wt_per*, *layers*)

This module contains functions to determine the a planet's radius when given mass and composition.

Parameters

radius_planet: float Radius of planet input by user $[m]$

core_mass_frac: float Core mass fraction from composition $[wt\%]$

structure_params: list Structural parameters of the planet; See example for description

compositional_params: list Structural parameters of the planet; See example for description

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

Core_wt_per: float Composition of the Core

layers: list Number of layers for core, mantle and water

Returns

Planet: dictionary Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

ExoPlex.functions.**find_Planet_radius**(*radius_planet*, *core_mass_frac*, *structure_params*, *compositional_params*, *grids*, *Core_wt_per*, *layers*)

This module contains functions to determine the a planet's mass when given radius and composition. Because we conserve the mass ratios of the planet we must iterate over core mass fraction to match the input composition.

Parameters

radius_planet: float Radius of planet input by user $[m]$

core_mass_frac: float Core mass fraction from composition $[wt\%]$

structure_params: list Structural parameters of the planet; See example for description

compositional_params: list Structural parameters of the planet; See example for description

grids: list of lists UM and LM grids containing pressure, temperature, density, expansivity, specific heat and phases

Core_wt_per: float Composition of the Core

layers: list Number of layers for core, mantle and water

Returns

Planet: dictionary Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planet

ExoPlex.functions.**find_filename**(*compositional_params*)

This module determines the closest compositional grid to pull from for a given composition

Parameters

compositional_params: list Structural parameters of the planet; See example for description

Returns

filename: string Name of file containing the grids based on this composition

ExoPlex.functions.**find_water_phase**(*Pressure*, *Temperature*)

This module determines the phase of water at a given pressure and temperature

Parameters

Pressure: float Pressure $[GPa]$

Temperature: float Temperature $[K]$

Returns

phase: string Phase of water present either water, ice VII, ice Ih, ice VI

ExoPlex.functions.**get_percents**(**args*)

This module calculates the bulk composition of the mantle given the elemental ratios provided in the input files

Parameters

compositional_params [list of floats] List containing molar ratios of planets, fraction of Fe in core, core composition and flags for writing individual phase and whether to use grids

Returns

Core_wt_per [dictionary] composition of core for individual elements:math:*[wt%]*

Mantle_wt_per [dictionary] composition of mantle in oxides $[wt\%]$

Core_mol_per [dictionary] composition of core in mole percent

core_mass_frac: float Mass of core divided by mass of planet

ExoPlex.functions.**get_phases**(*Planet*, *grids*, *layers*, *combine_phases*)

This module creates the output file of Pressure, Temperature and phase fractions

Parameters

Planet: dictionary Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planets

grids: list of lists Upper and lower mantle phase diagrams

layers: list number of core layers, mantle layers and water layers

combine_phases: boolean True if you'd like individual endmembers combined into constituent rocks, False if you'd like all endmembers

Returns

Phases [list] List of all phases present

new_names [list] composition of mantle in oxides $[wt\%]$

ExoPlex.functions.**make_mantle_grid**(*Mantle_filename*, *UMLM*, *use_grids*)

This module converts the PerPlex or premade grids into a dictionary of individual lists (e.g., pressure) for use by ExoPlex integrators

Parameters

Mantle_filename: string name of file either from PerPlex or premade grids

UMLM: boolean True for upper mantle grids, False for lower mantle grids

use_grids: boolean True is user is using premade grids, false if using perplex-derived grids

Returns

grid_dictionary: dictionary of lists dictionary of individual parameters taken from the phase diagram. Keys include: 'temperature','pressure','density','alpha','cp','phases'

ExoPlex.functions.**write**(*Planet*, *filename*)

This module creates the output file for reading later

Parameters

Planet: dictionary Dictionary of pressure, temperature, expansivity, specific heat and phases for modeled planets

filename: string chosen filename for output file

Returns

None

PYTHON MODULE INDEX

e

INDEX