Module `compose`

Functions

```
def all_MvsR(first_initial_pressure=33, last_initial_pressure=36, file_initial_letters='',
folder_initial_letters='', add_low_density=False)
```

Compute and plot radius versus mass for various models based on the equations of state files found in the "compose_eos" directory that match the given initial letters for both file names and folder names. If no filters are applied, it plots all models.

Parameters

`first_initial_pressure` : `float`
Exponent in base 10 of the first initial pressure to integrate. Default is 33.

`last_initial_pressure` : `float`
Exponent in base 10 of the last initial pressure to integrate. Default is 36.

`file_initial_letters` : `str`
Starting letters to filter files by name. Default is empty, meaning no filter.

`folder_initial_letters` : `str`
Starting letters to filter folders by name. Default is empty, meaning no filter.

`add_low_density` : `bool`
Boolean variable to indicate if low density - crust equation of state must be included.

```
def all_eos(file_initial_letters='',
folder_initial_letters='')
```

Plot the equation of state for different models based on the csv files found in the "compose_eos" directory that match the given initial letters for both file names and folder names. If no filters are applied, it plots all models with color corresponding to the folder.

Parameters

`file_initial_letters` : `str`
Starting letters to filter files by name. Default is empty, meaning no filter.

`folder_initial_letters` : `str`
Starting letters to filter folders by name. Default is empty, meaning no filter.

```
def all_v(file_initial_letters='',
folder_initial_letters='')
```

Compute and plot sound speed versus pressure for various models based on the equations of state files found in the "compose_eos" directory that match the given initial letters for both file names and folder names. If no

filters are applied, it plots all models.

Parameters

file_initial_letters : str
Starting letters to filter files by name. Default is empty, meaning no filter.

folder_initial_letters : str
Starting letters to filter folders by name. Default is empty, meaning no filter.

```
def all_z_and_I(first_initial_pressure=33, last_initial_pressure=36, file_initial_letters='',
folder_initial_letters='', add_low_density=False)
```

Compute and plot redshift and moment of inertia versus initial pressure for various models based on the equations of state files found in the "compose_eos" directory that match the given initial letters for both file names and folder names. If no filters are applied, it plots all models.

Parameters

first_initial_pressure : float
Exponent in base 10 of the first initial pressure to integrate. Default is 33.

last_initial_pressure : float
Exponent in base 10 of the last initial pressure to integrate. Default is 36.

file_initial_letters : str
Starting letters to filter files by name. Default is empty, meaning no filter.

folder_initial_letters : str
Starting letters to filter folders by name. Default is empty, meaning no filter.

```
def compute_moment_inertia(r, m)
```

Compute the Newtonian moment of inertia of a sphere.

Parameters

r : float
Radius in km.

m : float
Mass in solar masses.

Returns

float
Moment of inertia in g cm^2.

```
def compute_redshift(r, m)
```

Compute the gravitational redshift using the general relativity formula.

Parameters

r : float
Radius in km.

m : float
Mass in solar masses.

Returns

float
Dimensionless gravitational redshift.

```
def compute_sound_speed(range_p,
range_e)
```

Computes the speed of sound, i.e. the square root of the derivative of pressure with respect to energy density.

Parameters

range_p : NumPy array
A NumPy array containing pressures.

range_e : NumPy array
A NumPy array containing energy densities.

Returns

NumPy array:
A NumPy array containing the corresponding speeds of sound in units of c.

```
def numerical_derivative(range_x,
range_y)
```

Computes the numerical derivative using a 5-point method.

Parameters

```
range_x : NumPy array
    The independent variable.
range_y : NumPy array
    The dependent variable.
```

Returns

NumPy array:
A NumPy array containing the computed numerical derivatives.

```
def search_file_path(file_initial_letters='',
folder_initial_letters='')
```

Search for csv files in the "compose_eos" directory that match the given initial letters for both file names and folder names.

Parameters

file_initial_letters : str
Starting letters to filter files by name. Default is empty, meaning no filter.

folder_initial_letters : str
Starting letters to filter folders by name. Default is empty, meaning no filter.

Returns

list : A list of lists, where each sublist contains the file path, folder name and file name
of each matching file.

Functions