Users's guide NEMMO

Input Parameters

In order to efficiently carry out the analysis of the thermal evolution of a magmatic ocean, with the `Stage1Analysis` and `Stage2Analysis` classes, it is essential to provide the appropriate input parameters. These parameters define the initial conditions and specific characteristics of the model under study. Table 1 lists the required input parameters and their descriptions.

Variable	Description
`r_body`	Radius of the body [m]
`r_core`	Core radius of the body [m]
`albedo`	Reflectivity of the body, a dimensionless value
`rho`	Density of the material [$\mathrm{kg\,m^{-3}}$]
`gravity`	Gravitational acceleration [$\mathrm{m\,s^{-2}}$]
`initial_heat_production`	Initial heat production rate [$\mathrm{W\,kg^{-1}}$]
`c0`	Initial composition in anorthite component (fraction)
`ce`	Eutectic composition (fraction)
`k_crust`	Thermal conductivity of the crust [$\mathrm{W\,m^{-1}\,K^{-1}}$]
`D`	Partition coefficient for heat-producing elements (dimensionless)
`heat_source`	`False` for a global magma ocean, `True` for a non-global magma ocean
`r_flottability`	`None` for a global magma ocean; Radius at the depth of the magma ocean [m]
`distance_sun_object`	Distance from the Sun to the body [m]
`n_factor`	Factor to increase precision in cumulate resolution (dimensionless)
`overturn`	`False` for no overturn, `True` for with overturn, `None` for without overturn and no flux from cumulates
`t_overturn`	Factor to modify the initial heat flux of the overturn (dimensionless)

Table 1: Input Parameters for the `Stage1Analysis` and `Stage2Analysis` Class

Running Stage#1

To perform the analysis of the stage # 1, follow these instructions:

1. **Import the Class:** Begin by importing the `Stage1Analysis` class from the `evolution.py` module:

```
from evolution import Stage1Analysis
```

2. **Define the Problem:** Create an instance of the `Stage1Analysis` class to define your problem. Ensure that you provide the required input parameters (see the Input Parameters section for details).

```
stage1 = Stage1Analysis(...)
```

3. **Run the Analysis:** Execute the analysis by calling the method on the instance you created. This method will return the results of the analysis.

```
analysis = stage1.run_stage1_analysis()
```

4. **Review the Output:** The results of the analysis can be found in Table table 2.

Getter Functions for Temporal Evolution (ndarray)	
get_time_history()	Time [s]
get_r_history()	Radius [m]
get_T_history()	Temperature [K]
get_Ts_history()	Surface temperature [K]
get_h_solid_history()	Heat production in the solid [$W\,m^{-3}$]
get_h_lmo_history()	Heat production in the cumulates [$W\,m^{-3}$]
Getter Functions for Radial Profile (ndarray)	
get_hr_history()	Radial distribution of heat-producing elements [$W\,m^{-3}$]
get_T_profil()	Temperature [K]

Table 2: Getter functions for output stages. Temporal evolution and radial profile outputs.

Running Stage #2

To perform the analysis for Stage 2, follow these instructions:

1. **Import the Class:** Begin by importing the `Stage2Analysis` class from the `evolution.py` module:

```
from evolution import Stage2Analysis
```

2. **Define the Problem:** Create an instance of the `Stage2Analysis` class to define your problem. Ensure that you provide the required input parameters (see the Input Parameters section for details).

```
stage2 = Stage2Analysis(...)
```

3. **Run the Analysis:** Execute the analysis by calling the method on the instance you created. This method will return the results of the analysis.

```
analysis = stage2.run_stage2_analysis()
```

4. **Review the Output:** The results of the analysis can be found in Table table 3.

Getter Functions for Temporal Evolution (`ndarray`)	
`get_time_history()`	Time [s]
`get_radius_history()`	Two variables: • Crust radius [m] • Cumulates radius [m]
`get_temp_history()`	Two variables: • Surface temperature [K] • Core temperature [K]
`get_h_history()`	Three variables: • Heat production of the LMO [$\mathrm{W\,m^{-3}}$] • Heat production of the crust [$\mathrm{W\,m^{-3}}$] • Heat production of the cumulates [$\mathrm{W\,m^{-3}}$]
`get_drdt_history()`	Two variables: • Growth rate of the crust [$\mathrm{m\,s^{-1}}$] • Growth rate of the cumulates [$\mathrm{m\,s^{-1}}$]
`get_flux_history()`	Five variables: • Flux of the crust [W] • Flux of the cumulates [W] • Flux of the LMO [W] • Latent heat [W] • Overturn heat flux [W]
`get_boundary_temp()`	Two variables: • Boundary temperature at the bottom of the crust [K] • Boundary temperature at the top of the cumulates [K]
Getter Functions for Radial Profile (`ndarray`)	
`get_crust_profil()`	Three variables: • Radius [m] • Temperature profile of the crust [K] • Heat production profile of the crust [$\mathrm{W\,m^{-3}}$]
`get_solid_profil()`	Three variables: • Radius [m] • Temperature profile of the cumulates [K] • Heat production profile of the cumulates [$\mathrm{W\,m^{-3}}$]
Getter Function for the Overturn Constants (`float`)	
`get_overturn_constant()`	Three variables: • Heat stored in the cumulates [J] • Initial flux [W] • Decay constant [$\mathrm{Myr^{-1}}$]

Table 3: Getter Functions for Temporal Evolution, Radial Profile, and Overturn Constants