

# (j)edit

*an editor which allows interactive exploration of the properties of elementary  
functions in the computing environment IPython/Jupyter*

# system purpose/recap

interactive application

education tool

mathematical analysis

visualization tool

numerical computing

jupyter notebook application

# technical overview

## **coding language**

Python 3.7.4

## **libraries**

Numpy

Scipy

matplotlib

ipywidgets

## **environment**

Anaconda

Jupyter Notebook

# modules overview

## analysis module

computing. algorithms. plotting. function management.

## gui module

main board. outputs. menu. observing. logger.

## settings module

configuration interface. yaml loading.

## runner module

software running. task management.



# analysis module

## computing. algorithms.

- recalculating functions based on user input
- derivatives
- roots (zero points) – Newton, Halley, Secant
- local extrema (using derivatives)
- inflex points (using derivatives)
- monotonic intervals (using derivatives)
- concave intervals (using derivatives)

### <<Analysis>> ComputationsManager

- main\_function: void()
- main\_derivatives: void()
- zero\_points: void()
- extremes: void()
- inflex\_points: void()
- monotonic: void()
- concave: void()

# analysis module

## graph plotting.

- matplotlib
- plotting parts of graph based on user choice

<<Analysis>>

Plotter

- plot\_main\_function: void()
- plot\_asymptotes: void()
- plot\_derivatives: void()
- plot\_zero\_points: void()
- plot\_extremes: void()
- plot\_inflex\_points: void()
- plot\_intervals: void()
- plot\_all: void()

# analysis module

## function management.

- class Function
- class Manager
- loading user data
- preparing data to plot
- updating data/plot
- interface for GUI module

<<Analysis>>

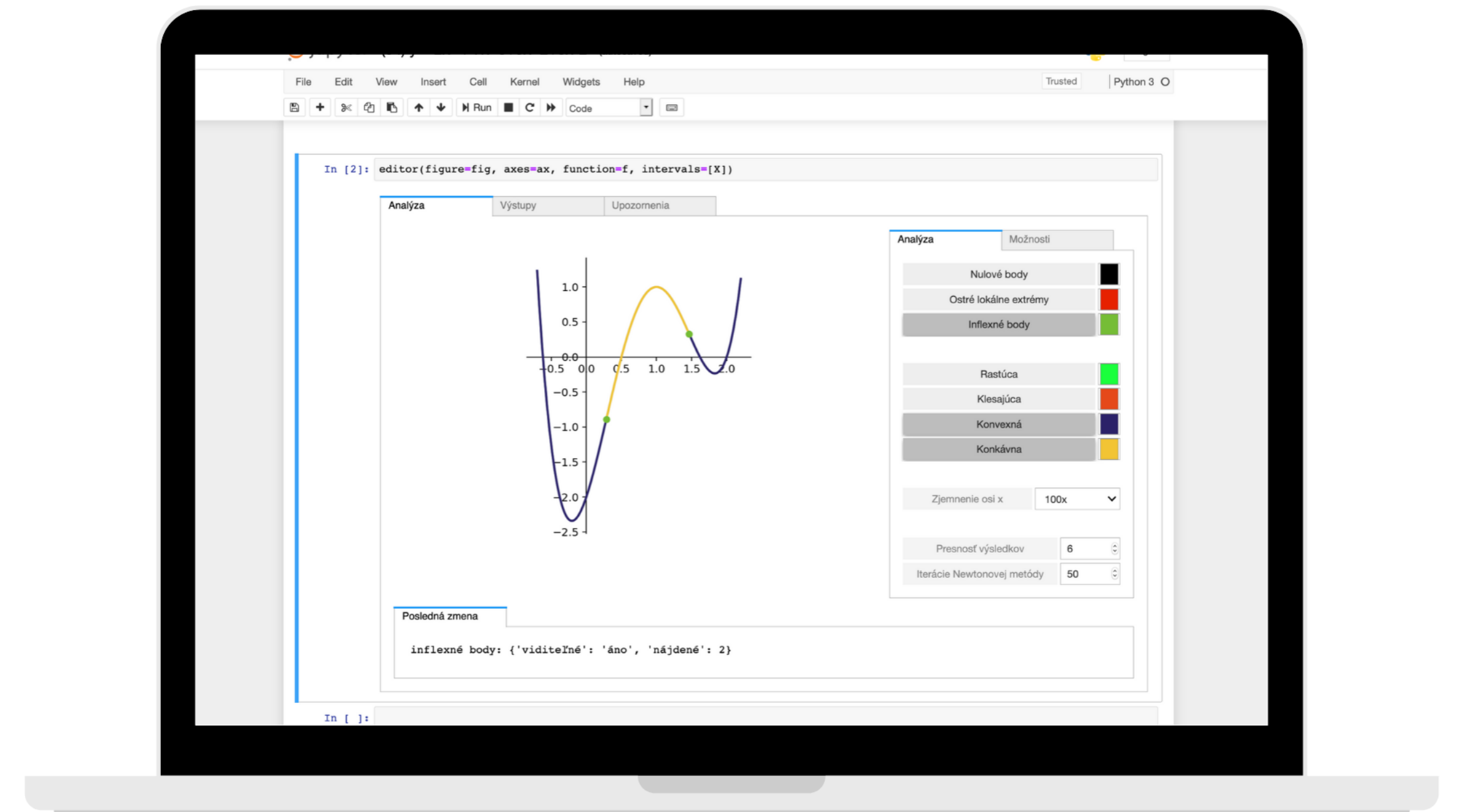
Function

- init\_function\_details: void()
- init\_plot\_parameters: void()
- init\_derivatives: void()
- init\_zero\_points: void()
- init\_analysis: void()
- init\_refinement: void()
- plot: void()

# gui module

main board. tabs. outputs. menu.

- ipywidgets
- main board divided to tabs
- main menu divided to tabs
- interactive elements
- outputs – plot, logs, warnings
- logic





# gui module

## observing.

- class Observer
- changed\_ methods for observing
- saving user actions
- handling next steps

<<GUI>>

Observer

- changed\_grid: void()
- changed\_color: void()
- changed\_zero\_points: void()
- changed\_inflex\_points: void()
- ...
- start()

# gui module

## logging. printing.

- class Logger, LoggerMessage, LoggerStack
- saving information about user actions
- saving calculated values
- saving warnings during calculations
- printing to output
- exporting txt/JSON

<<GUI>>

Logger

- new\_message: LoggerMessage
- set\_order: void()
- write: void()
- to\_file: void()

# settings module

configuration interface. yaml loading.

- configuration file YAML
- loading configuration file
- interface for configuration

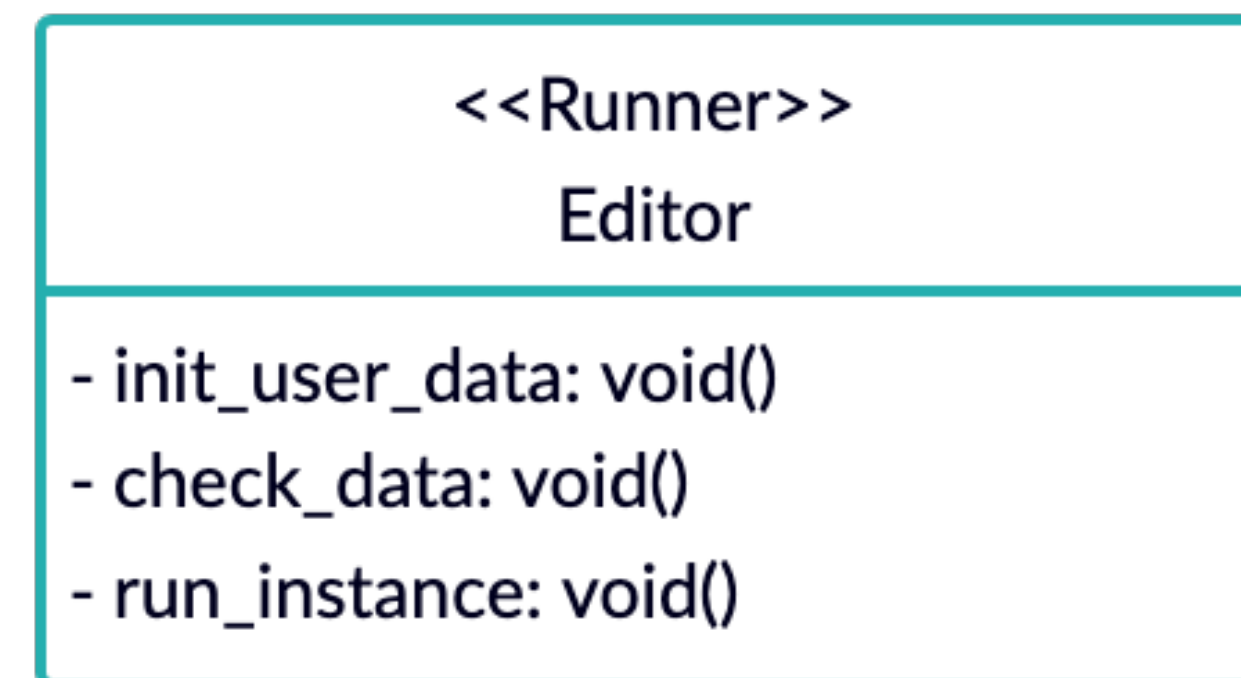
<<Settings>>  
Settings

- load\_yaml: void()  
- get\_data: dict

# runner module

**software running. task management.**

- running Editor instances
- loading and checking user data
- loading program elements



**thank you for your attention**