Amateur-ish 8th order Runge-Kutta implementation

The file `rk.py` contains the source code for a semi-functional implementation of an 8th order Runge-Kutta integrator with 9th order error approximation and adaptive step capabilities.

Installation

The function has a few dependencies that can be installed with `pip`. Maybe in the future I will provide a `requirements.txt` file, but for now, you can install the dependencies with the following command:

```
pip install numpy rich
```

Rich is a package used for Progress Bars, pretty printing, and other nice things. It is not necessary for the function to work, but it is nice to have. Currently there is no way to disable it, but I will add that in the future I suppose.

Usage

The function is used as follows:

```
from rk import *

# Define the function to be integrated
def ivp(x, y):
    """Test IVP to try out integrators.
    Has the analytical solution 2 * np.exp(-2*t) + np.exp(t)
    for y(0) = 5
    """
    return -2*y + 3 * np.exp(x)

# Define the initial conditions
x0 = 0
y0 = 5
x = 10
stepSize = 0.0005
solution, errors= RK8_9(ivp, x0, y0, x, stepSize)
```

The function generally returns two arrays where the columns of the arrays are the values of the solution and the errors at each step.

Parameters

The function has optional parameters mostly pertaining to debugging and testing. They are as follows:

Parameter	Type	Description

Parameter	Type	Description
`outputSteps`	`bool`	If `True`, the function also return a third array of the steps used for calculation.
`debug`	`bool`	If `True`, the function will print out the values of the solution and the errors at each step.
`exitOnFail`	`bool`	If `True`, the function will raise a ValueError on Failure/Warning.
`disableDivCheck`	`bool`	If `True`, the function will not check for divergence.
`overrideInternalSettings`	`dict[str, float]`	If not 'None', will use parameters in given dict. See below.

Overriding Internal Settings

The function has a few internal settings that can be overridden. They are as follows:

divergenceToleance

The tolerance for divergence. If the error is greater than this value, the function will warn the user or exit, if so set by `exitOnFail`. The default value is `1e10`.

eps

The adaptive step size tolerance. If the error is less than this value, the step size will be increased and vice versa. The default value is `1e-8`.

stepDownSafetyFactor

The factor by which is used in the calculation of the decrease of the step size when its error is too large. The default value is `0.9`.

stepUpSafetyFactor

The factor by which is used in the calculation of the increase of the step size when its error is too small. The default value is `1.1`.

Example

Below is an example of how to override the internal settings:

```
override = {
    "divergenceTolerance": 1e12,
    "eps": 1e-6,
    "stepDownSafetyFactor": 0.8,
    "stepUpSafetyFactor": 1.2
}
```