Evaluating Runge-Kutta Normalization Error
$N = 51$, $\beta = 0.5$ $T = 1000$

Solver: python **scipy.integrate.solve-ivp** solver is used with the default Runge-Kutta *'RK45'* integration method.
RK45: Explicit Runge-Kutta method of order 5(4). The error is controlled assuming accuracy of the fourth-order method, but steps are taken using the fifth-order accurate formula (local extrapolation is done) [1]

Step size: The method is adaptive and the error is controlled with the relative and absolute tolerances parameters

Relative and absolute tolerances. The solver keeps the local error estimates less than atol + rtol * abs(y). Here **rtol** controls a relative accuracy (number of correct digits). But if a component of y is approximately below **atol**, the error only needs to fall within the same atol threshold, and the number of correct digits is not guaranteed [1]

rtol	atol	norm	error	comp time (s)
e-3	e-5	1.0851	-0.0851	83.8
e-3	**e-6**	**1.0198**	**-0.0198**	**19.5**
e-3	e-7	1.0188	-0.0188	19.4
e-3	e-8	1.0187	-0.0187	19.5
e-4	e-5	0.9988	1.16e-3	90.6
e-4	e-6	0.9996	3.98e-4	21.7
e-4	e-7	0.9996	4.13e-4	21.4
e-4	e-8	0.9996	4.04e-4	21.5
e-5	e-5	0.9989	1.09e-3	82.8
e-5	e-6	0.9998	1.81e-4	20.5
e-5	e-7	0.9999	1.33e-4	20.6
e-5	e-8	0.9999	1.21e-4	20.8
e-6	e-5	0.9998	2.48e-4	92
e-6	e-6	0.9999	5.78e-5	22.6
e-6	e-7	0.9999	2.47e-5	23.2
e-6	e-8	0.9999	1.68e-5	23.2

[1] *Scipy Documentation Page*