Application of the midpoint method for a system of two ordinary differential equations

Szymon Górski

1 Assignment

The goal of the project was to provide a solution for the following task:

The midpoint method ($Y_{k+1} = Y_{k-1} + 2hF_k$) for a system of 2 ordinary differential equations. Use the Heun method (the 2nd order method with $\gamma = 1$ — cf. Lecture Notes) to calculate the starting value Y_1.

The input given for the task is the system of equations as well as an initial value x_0 and values of y functions in x_0.

2 Mathematical background

2.1 Introduction

To calculate the values of y's in the first step (starting values) one should use the Heun method. This approximation technique is based on the evaluation of the first derivative of examined function in the previous step (in this case x_0). It is crucial to notice that one needs to be able to obtain these exact values, therefore to successfully apply this method one needs (1) be given exact values of all y's and its derivatives up to $y^{(m-1)}$, where m is the order of a differential equation or (2) be provided with formulas for the fist derivative of each y function. In this task we deal with the second option. Since there are exactly 2 functions given and two functions of the first order necessary to obtain the output, both functions given must be of the first order.

Note

Throughout the document the following notation is used:

- x_1 represents the value of x in the first step,
- f^1, y^1 index represents the first of two differential equations from the system,
- y'^1 represents the first derivative of the value above,
- Y represents the vector of derivatives of y's (in this case $Y = (y^1, y^2)$),
- y^1_0 is equivalent to $y^1(x_0)$,
- h represents a step increasing an argument ($x_{i+1} = x_i + h$).

One can derive an intuitive formula for this system:

$$\begin{cases} f^1(x, y^1, y^2) \\ f^2(x, y^1, y^2) \end{cases} \leftrightarrow \begin{cases} y^{1'} = a + by^1 + cy^2 + ... \\ y^{2'} = p + qy^1 + ry^2 + ... \end{cases}$$

where all variables used in equations represent functions of the same variable x. [1]

To solve the task, two second-order Runge–Kutta methods with two stages were used. The family of these methods is represented by the formula:

$$y^i_{n+1} \approx y^i_n + h\big((1 - \tfrac{1}{2\alpha})f^i(x_n, Y_n) + \tfrac{1}{2\alpha} f^i(x_n + \alpha h, y^i_n + \alpha h f^i(x_n, Y_n))\big),$$

where $\alpha = \frac{1}{2}$ is the midpoint method and $\alpha = 1$ is the Heun's method. [2]

2.2 The Heun's method

The Heun's method is a predictor-corrector method based on the Euler method. The explicit formula is used to predict a step:

$$\bar{y}^i(x + h) \approx y^i(x) + hf^i(x, Y(x)). \tag{1}$$

Next, predictions (1) are put into the correcting formula:

$$y^i(x + h) \approx y^i(x) + \frac{h}{2}\left(f^i(x, Y(x)) + f^i(x + h, \bar{Y}(x + h))\right).$$

2.3 The midpoint method

The midpoint method is another second-order method derived from the Euler method. It is based on an improved slope formula:

$$y^{i'}\left(x + \frac{h}{2}\right) \approx \frac{y^i(x + h) - y^i(x)}{h} \rightarrow y^i(x + h) \approx y^i(x) + hf^i\left(x + \frac{h}{2}, Y(x + \frac{h}{2})\right), \tag{2}$$

and its Taylor expansion:

$$y^i\left(x+\frac{h}{2}\right) \approx y^i(x) + \frac{h}{2}y^{i'}(x) = y^i(x) + \frac{h}{2}f^i(x, Y(x)). \tag{3}$$

Using formulas (2) and (3) one gets the explicit midpoint method:

$$\bar{y}^i\left(x+\frac{h}{2}\right) \approx y^i(x) + \frac{h}{2}f^i(x, Y(x)),$$

$$y^i(x+h) \approx y^i(x) + hf^i\left(x+\frac{h}{2}, \bar{Y}(x+\frac{h}{2})\right), \tag{4}$$

which can be successfully used for approximation of Y in a loop. [3] However, knowing values in both initial and first step (Y_0, Y_1) one can shrink the step size by two ($h := 2h$) and apply Y_1 directly to (4):

$$y^i(x+2h) \approx y^i(x) + 2hf^i\left(x+h, Y(x+h)\right).$$

3 Testing

To test the implementation of the methods the following 4 systems were used:

1. $\begin{cases} y^{1'} = -5y^1 - 2y^2, y^1(3) = 2 \\ y^{2'} = y^1 + 5y^2, y^2(3) = 5 \end{cases}$
2. $\begin{cases} y^{1'} = 2y^1 + 3y^2 + x, y^1(1) = 2 \\ y^{2'} = 5y^1 + 10y^2 - 10x, y^2(1) = 2000 \end{cases}$
3. $\begin{cases} y^{1'} = 2y^1 - 5y^2, y^1(0) = 1 \\ y^{2'} = 5y^1 + 10y^2, y^2(0) = 1 \end{cases}$
4. $\begin{cases} y^{1'} = 4y^1 + 4y^2, y^1(0) = 6 \\ y^{2'} = -8y^1 - 4y^2, y^2(0) = 12 \end{cases}$

Systems were approximated using both the solution presented and the built-in function `ode45`, which chose automatically the amount of steps (N) to perform the task. In all cases (except the second one) the length of the step was fixed to $h = 0.01$.

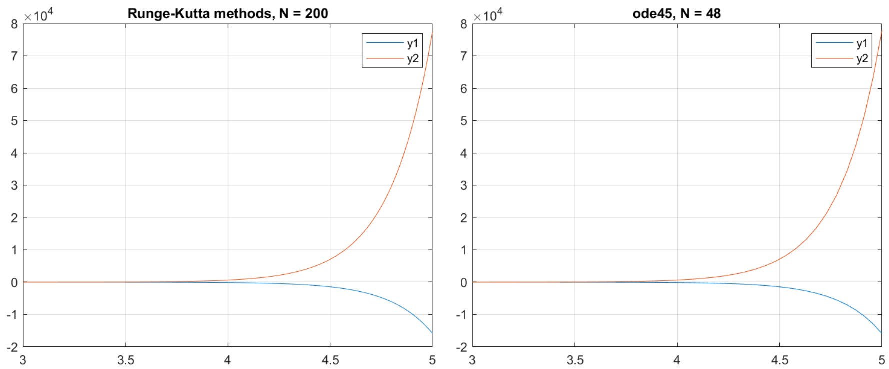

Figure 1.

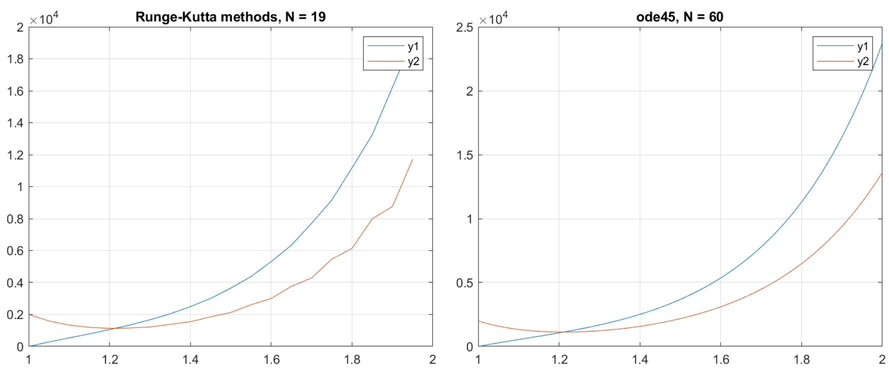

Figure 2. The step size was increased to $h = 0.05$ to show possible inaccuracies in approximations of functions quickly changing its values.

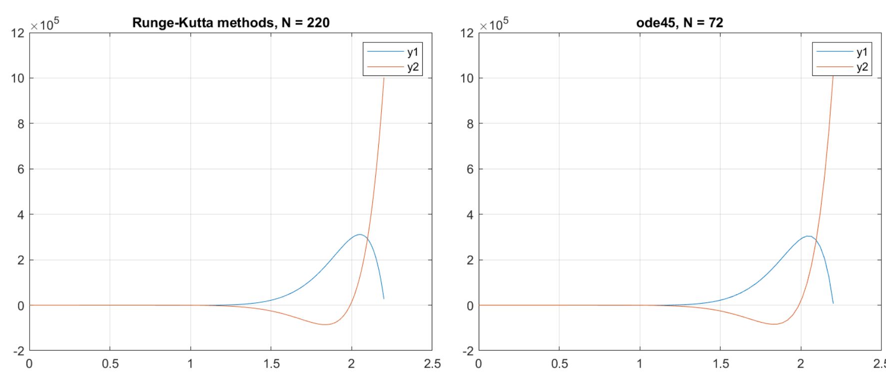

Figure 3.

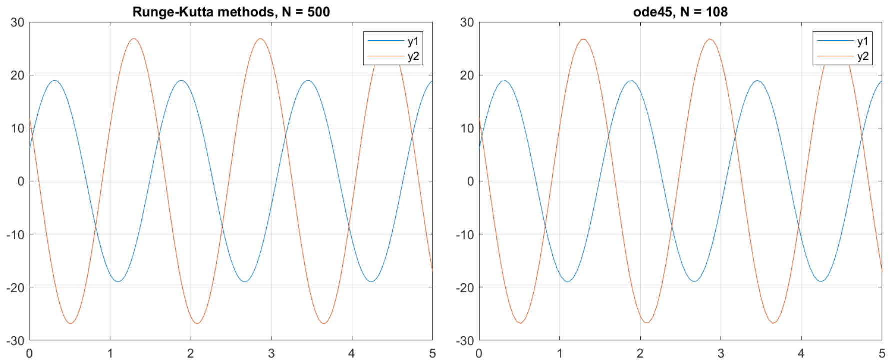

Figure 4.

Next, the systems were solved symbolically in the external software (using Wolfram language). The functions obtained are:

1. $$\begin{cases} y^1(x) \to \frac{1}{46} e^{\sqrt{23}(-x)-3\sqrt{23}} \left(27\sqrt{23}e^{2\sqrt{23}x} + 115e^{2\sqrt{23}x} - 27e^{6\sqrt{23}}\sqrt{23} + 115e^{6\sqrt{23}}\right) \\ y^2(x) \to \frac{e^{\sqrt{23}(-x)-3\sqrt{23}}\left(\sqrt{23}e^{2\sqrt{23}x} - 10e^{2\sqrt{23}x} + e^{6\sqrt{23}}\sqrt{23} + 10e^{6\sqrt{23}}\right)}{\sqrt{23}} \end{cases}$$

2. $$\begin{cases} y^1(x) \to e^{6x}(3\sin(3x) + \cos(3x)) \\ y^2(x) \to e^{6x}(\cos(3x) - 3\sin(3x)) \end{cases}$$

3. output omitted due to its length

4. $$\begin{cases} y^1(x) \to 12(\cos(4x) - 2\sin(4x)) \\ y^2(x) \to 6(3\sin(4x) + \cos(4x)) \end{cases}$$

The same software was used to plot the functions presented above:

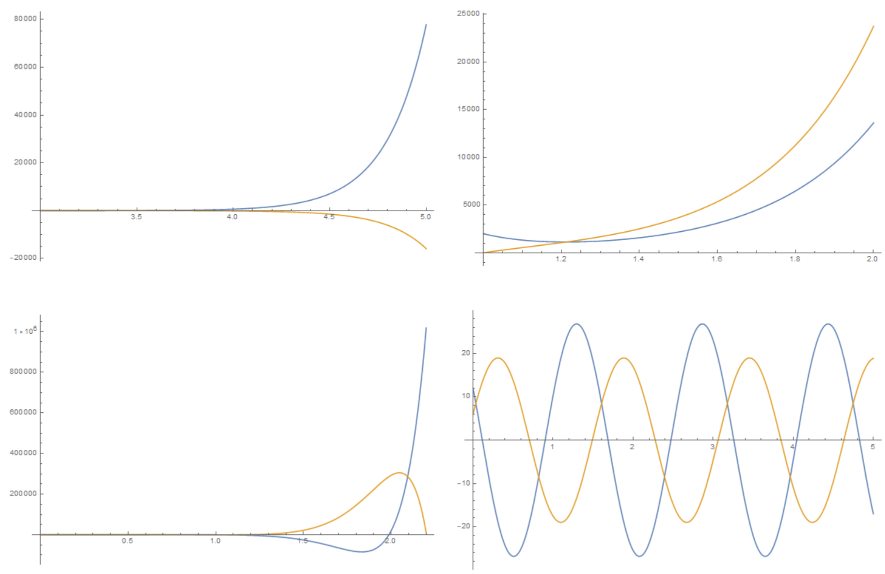

Figure 5. Plots generated using Wolfram language.

All outputs, except the second one, gave results nearly identical to presented approximation methods. With sufficient number of steps the approach managed to nearly perfectly estimate the value of complex polynomials, trigonometric as well as exponential functions.

References

[1] P. Keller, ”Metody Numeryczne 2”

[2] Wikipedia.org, ”Explicit Runge–Kutta methods”
`https://en.wikipedia.org/wiki/Runge-Kutta_methods#Explicit_Runge-Kutta_methods`

[3] Wikipedia.org, ”Derivation of the midpoint method”
`https://en.wikipedia.org/wiki/Midpoint_method#Derivation_of_the_midpoint_method`