Name: ______________________________

Date: _______________________________

Collaborators: ____

(Collaborators submit their individually written assignments together, in class, in person)

GitHub username: ___________________________

(Not needed if the real name is used as the username)

Question:	1	2	3	Total
Points:	50	15	5	70
Score:				

Instructor/grader comments:

Runge-Kutta midpoint method

1. (50 points) Consider the Initial Value Problem (IVP) for a first order differential equation:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y), \quad a \le t \le b, \quad y(a) = \alpha. \tag{1}$$

A Runge-Kutta method for solving Eq. (1), known as midpoint method, is the following algorithm:

$$\begin{aligned} k_1 &= hf(t_i, y_i), \\ k_2 &= hf(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}), \\ y_{i+1} &= y_i + k_2, \end{aligned} \tag{2}$$

where $i = 1, \dots, n-1$, $t_i = a + (i-1)h$, h is the integration step, $h = (b-a)/(n-1)$.

The goal of this assignment is to implement the midpoint algorithm and conduct numerical experiments to investigate its convergence.

(a) Accept the assignment in GitHub Classroom, launch the codespace, open the template of the notebook for the assignment, `midpoint.ipynb`.

(b) Use Markdown to fill in the blanks in the introductory part of the notebook.

(c) Write a function `myrkmid(fun, a, b, n, y1)` that accepts as the arguments the function of two variables, `fun(t, y)` (the right-hand side of Eq. (1)), the integration limits `a` and `b`, the number of nodes `n` to use, and the initial value $y(a) = y1$. The function should implement the Runge-Kutta midpoint method, and returns two vectors, t and y, where $t_i = a + (i-1)h$ and $y_i = y(t_i)$, $i = 1, \dots, n$ is the solution of the IVP at $t = t_i$.

(d) Consider the IVP,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y, \quad 0 \le t \le 5, \quad y(0) = 1, \tag{3}$$

with the exact solution

$$y_{\mathrm{ex}}(t) = e^t. \tag{4}$$

Solve IVP Eq. (3) using your function `myrkmid` for $n = 16$. On the same figure plot your numerical solutions and the exact solution Eq. (4). Provide the legend, grid, title, axes labels for your graph.

The expected graph is shown in Fig. 1.

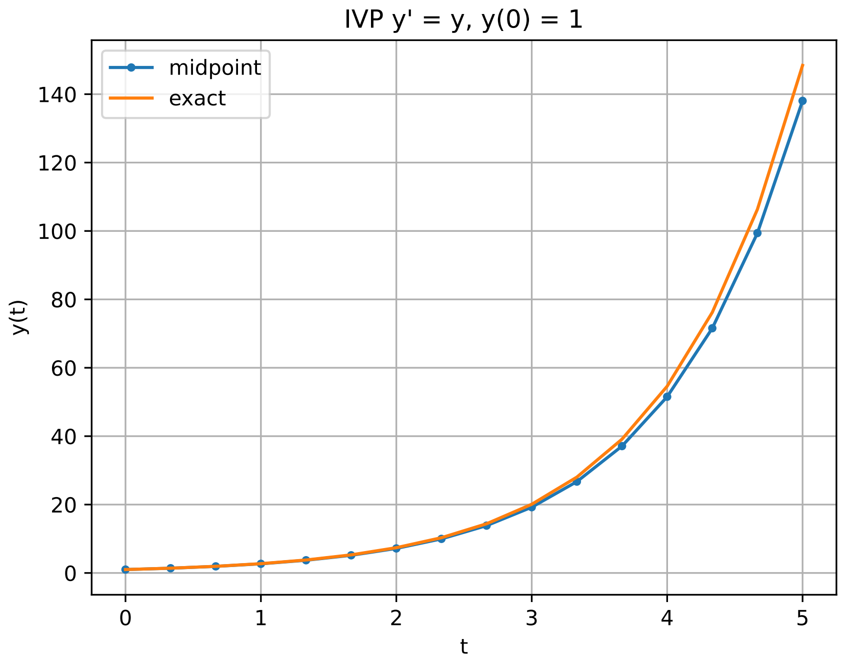

Figure 1: Expected graph in Problem 1(c).

(e) Solve IVP Eq. (3) using your function `myrkmid` for $n = 8, 16, \ldots, 2048, 4096$, i.e. $n(l) = 2^{l+3}$, $l = 1, \ldots, 9$.

Find the global errors of your solutions defined as the errors of solutions at the right hand end of the integration range:

$$\Delta(h_{n(l)}) = \left| y_{n(l)} - y_{\text{ex}}(b) \right|. \tag{5}$$

Plot Δ vs. the integration step h. By visual inspection determine the order of accuracy of the midpoint Runge-Kutta method. ((Use the appropriate style of plot axes.) Provide the legend, grid, title, axes labels for your graph.

The expected graph is shown in Fig. 2.

Describe your reasoning and the result of your numerical experiment.

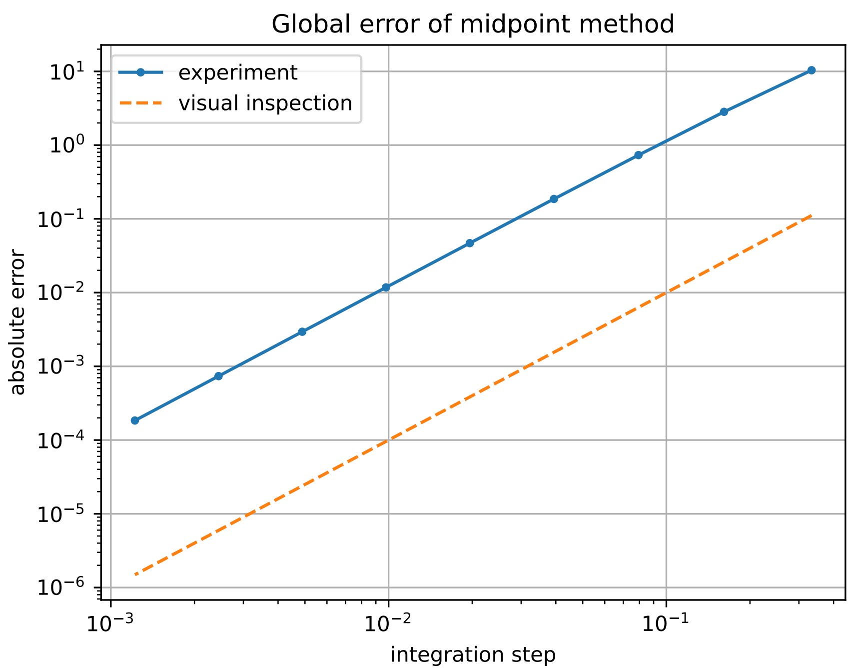

Figure 2: Expected graph in Problem 1(e).

Numerical integration

2. (15 points) **Sine integral** is one of the special mathematical functions defined as follows:

$$\mathrm{Si}(x) = \int_0^x \frac{\sin(t)}{t}\,\mathrm{d}t$$

Write a function, `mysinint(x)` that implements the sine integral. Your `mysinint(x)` must in turn call a Simpson's quadrature. Use the code for Simpson's quadrature that you wrote for HW3. Plot the graph of `mysinint(x)` for $1 \le x \le 3\pi$ using 100 data points. Use 101 points for Simpson's integration. Be careful to avoid the division by zero in the integrand of `mysinint(x)`.

The expected graph is shown in Fig. 3.

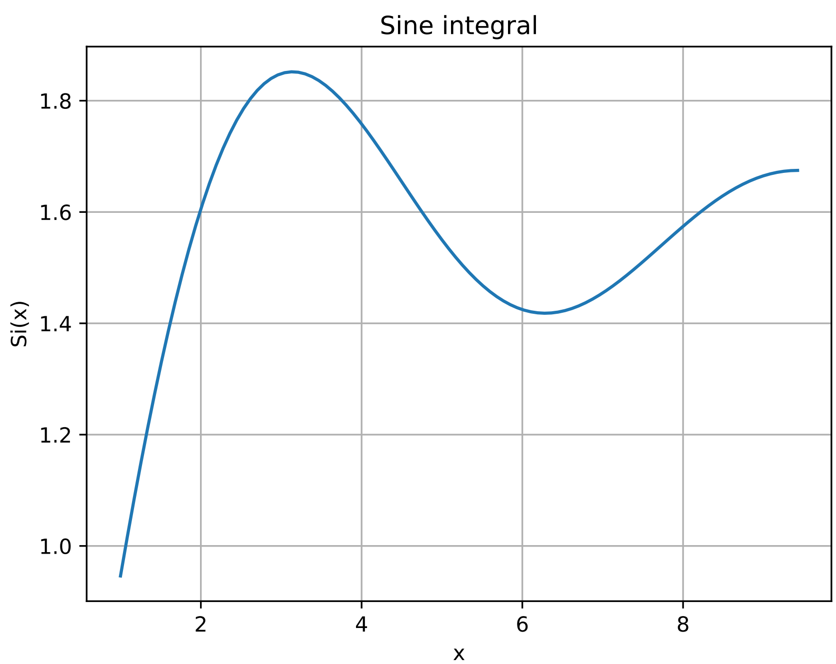

Figure 3: Expected graph in Problem 2.

3. (5 points) Clean the cells of all your jupyter notebook, save and close the notebook. Delete unneeded notebooks if you created any (e.g. `Untitled.ipynb`). Commit all your code changes, including changes to Project.toml and Manifest.toml files. Push your commits to the assignment's GitHub repository.