

Project 1 MM5016

Contents

1	Introduction	1
2	Graphs	2
2.1	Euler's method	2
2.2	2nd order Runge Kutta	3
2.3	4th order Runge Kutta	4
2.4	Adam-Bashforth method	5
3	Discussion	6

1 Introduction

In this document, 4 numerical methods for solving a specific ODE will be visualised and briefly discussed. The numerical methods being: Euler's method, second order Runge-Kutta, fourth order Runge-Kutta and the Adam-Basforth method. The ODE that these methods will try to approximate will be the solution to:

$$\frac{du}{dt} = \cos(\pi t) + u(t),$$

with initial value $u(0) = 2$. The analytical solution can be yielded by multiplying with an integrating factor as such:

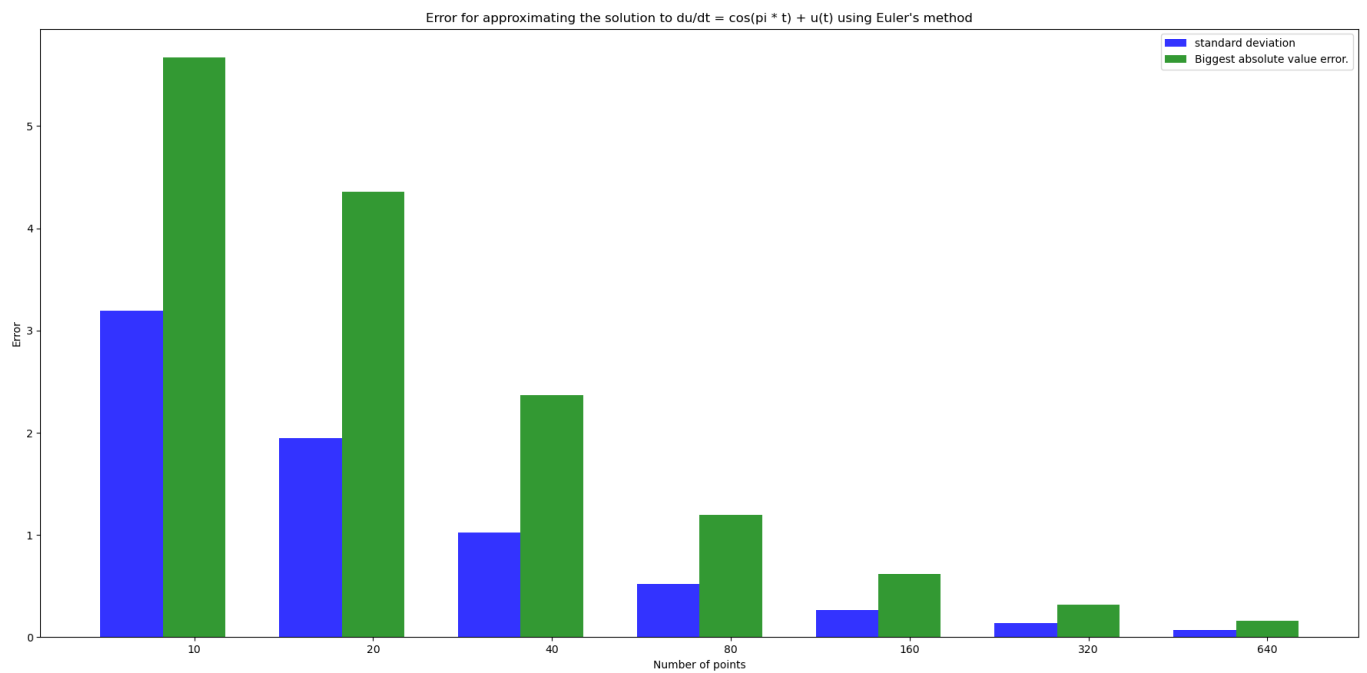
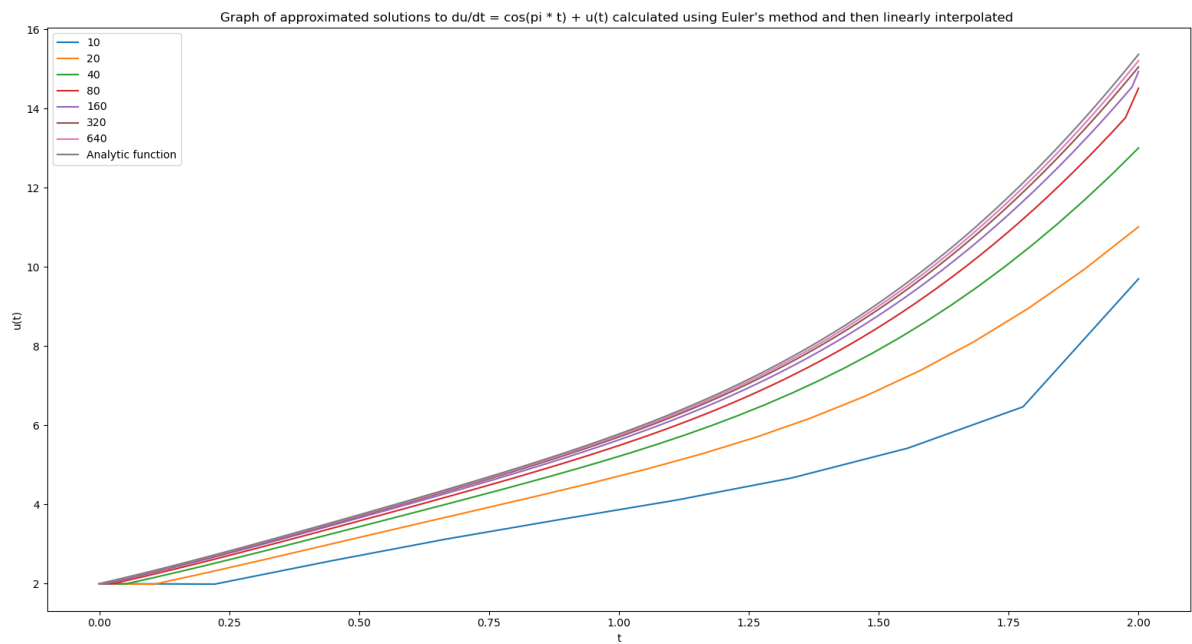
$$\begin{aligned}\frac{du}{dt} &= \cos(\pi t) + u(t) \\ \implies \frac{du}{dt} - u(t) &= \cos(\pi t) \\ \implies e^{-t} \frac{du}{dt} - e^{-t} u(t) &= e^{-t} \cos(\pi t) \\ \implies \frac{d}{dt}(e^{-t} u(t)) &= e^{-t} \cos(\pi t) \\ \implies e^{-t} u(t) &= C + \int e^{-t} \cos(\pi t) dt \\ \implies u(t) &= e^t C + e^t \int e^{-t} \cos(\pi t) dt.\end{aligned}$$

The integral on the right can be evaluated using repeated partial integration, so an analytical solution exists, and it will be the one we will compare with the approximated solutions from the numerical methods.

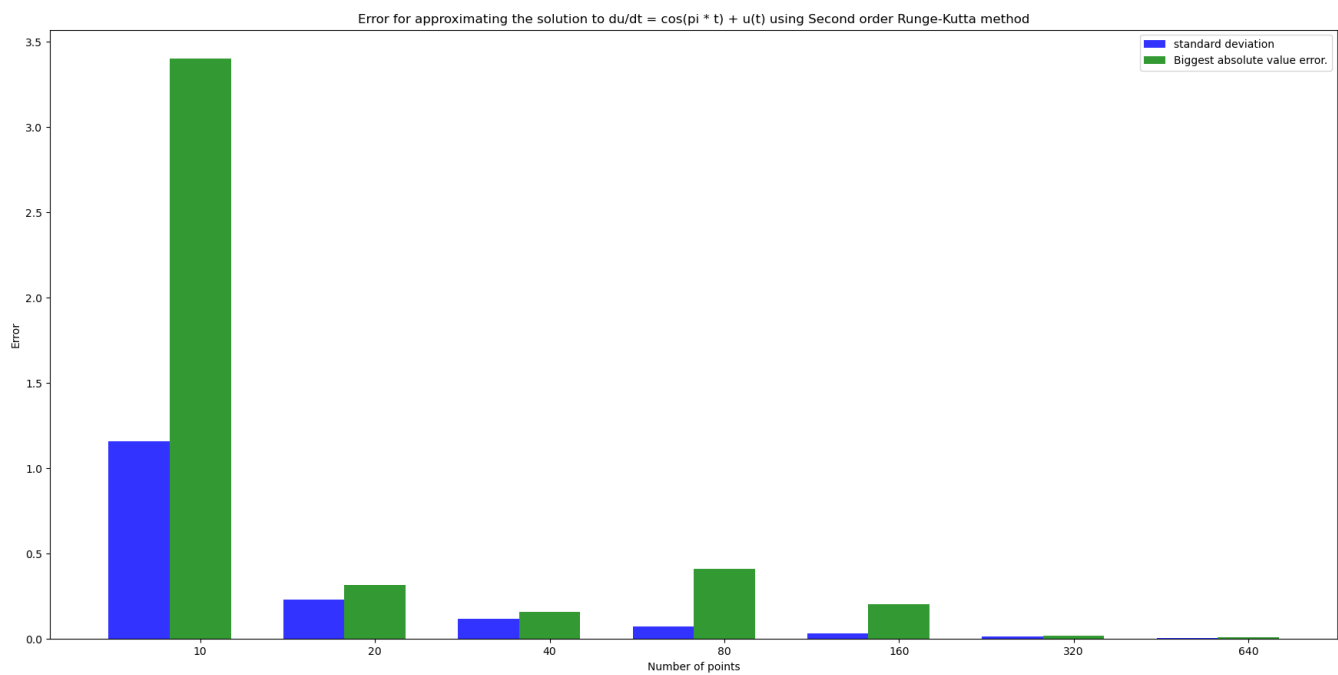
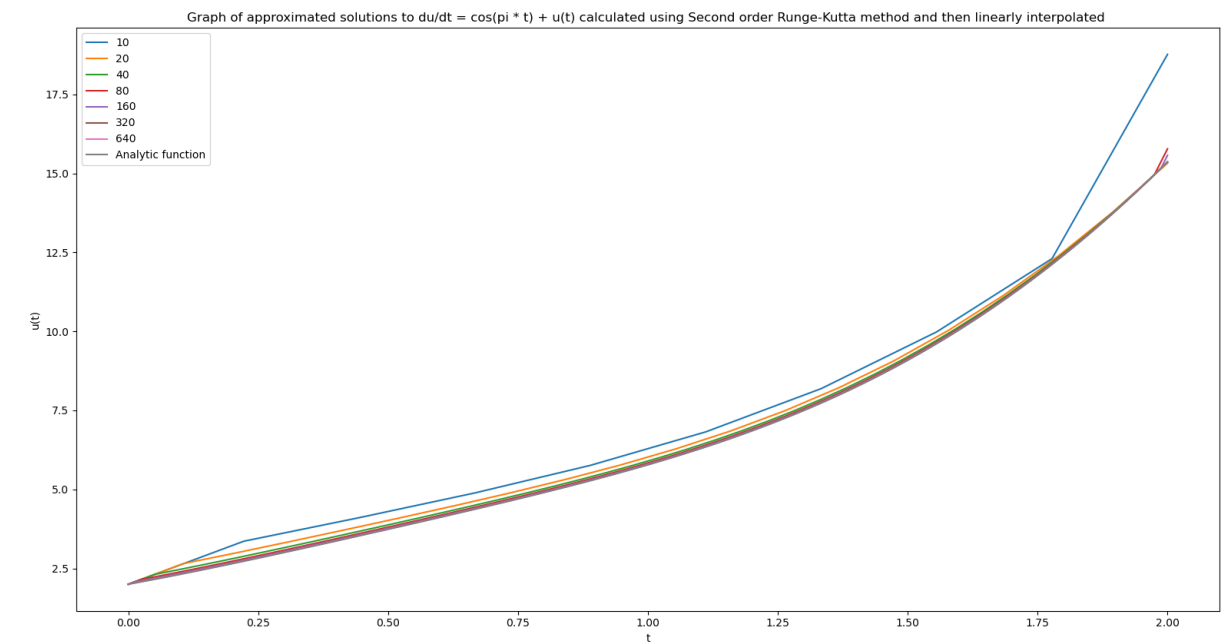
All the following graphs can be generated by running the included python 3 scripts in a terminal.

2 Graphs

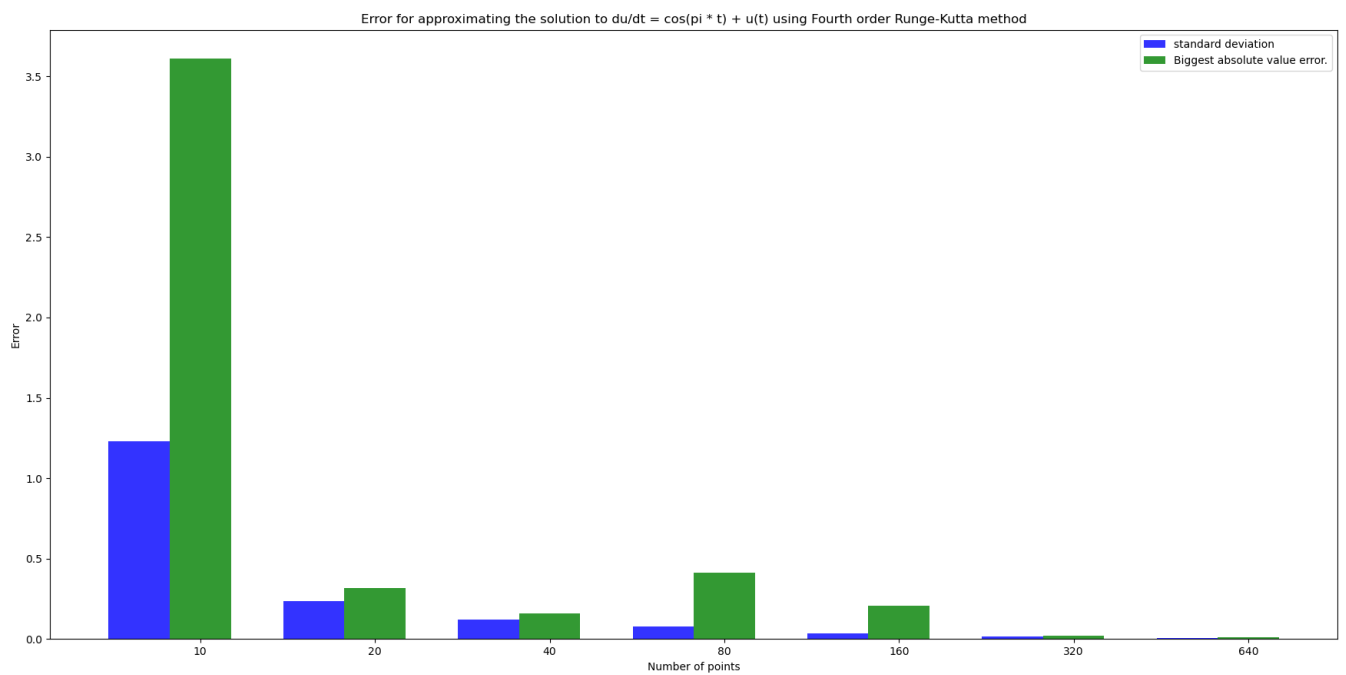
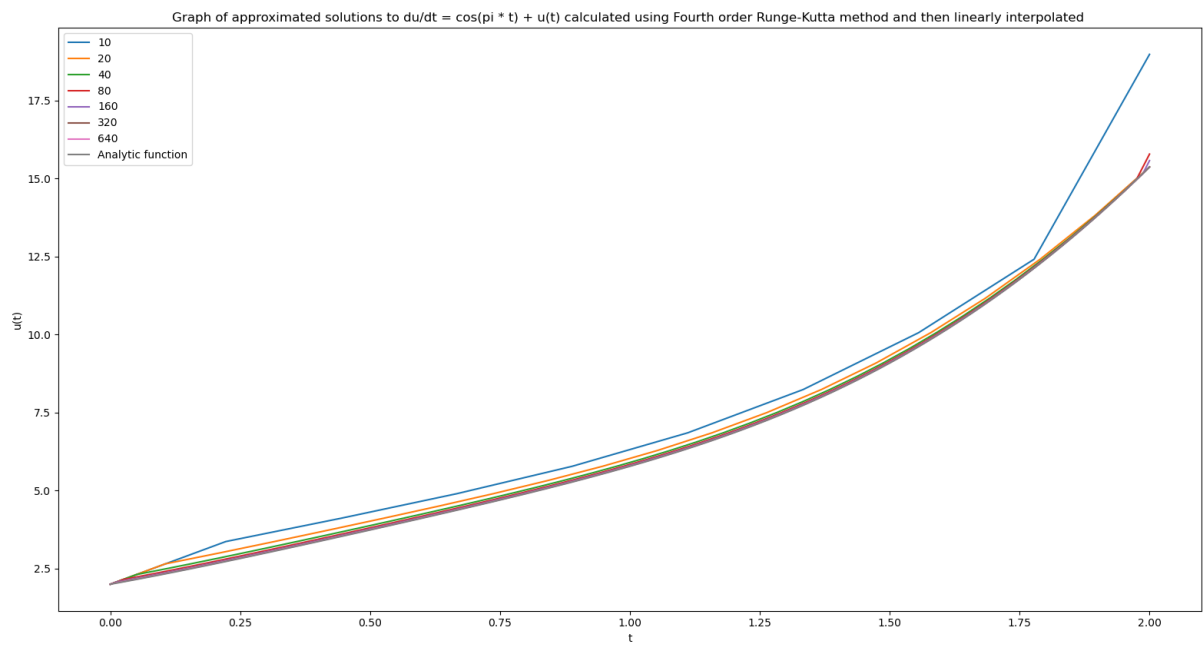
2.1 Euler's method



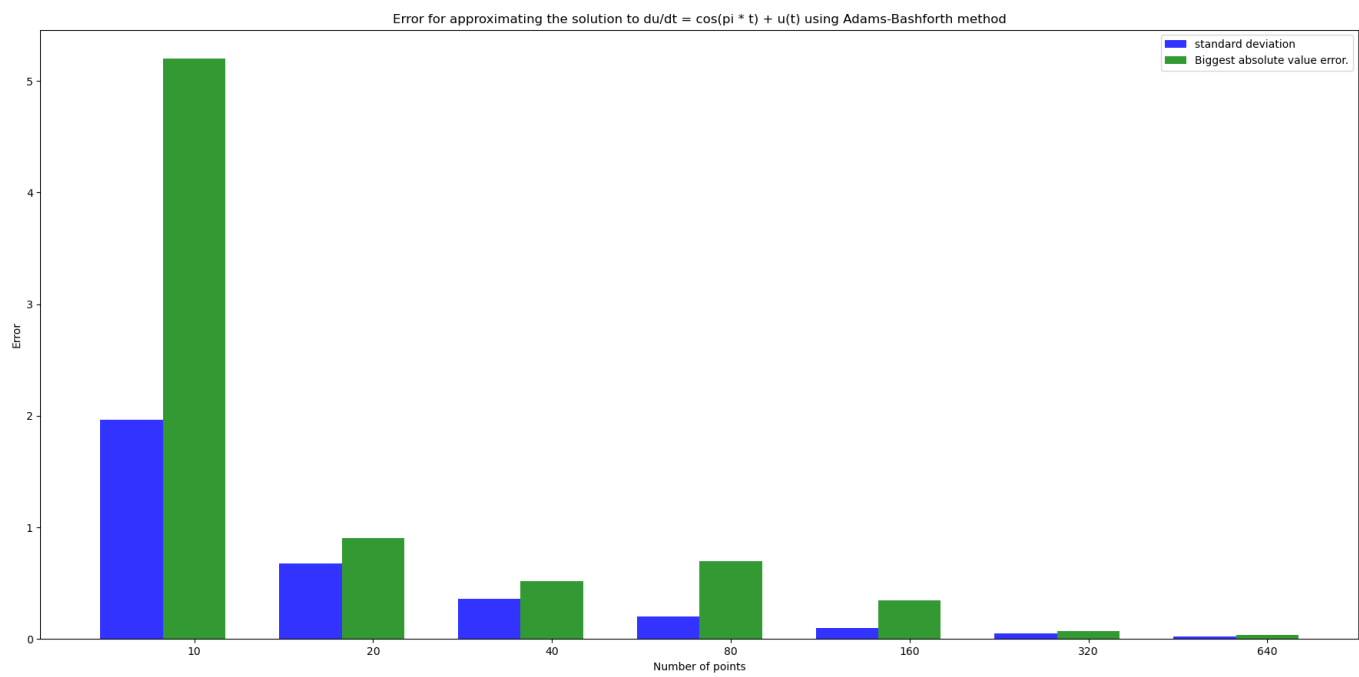
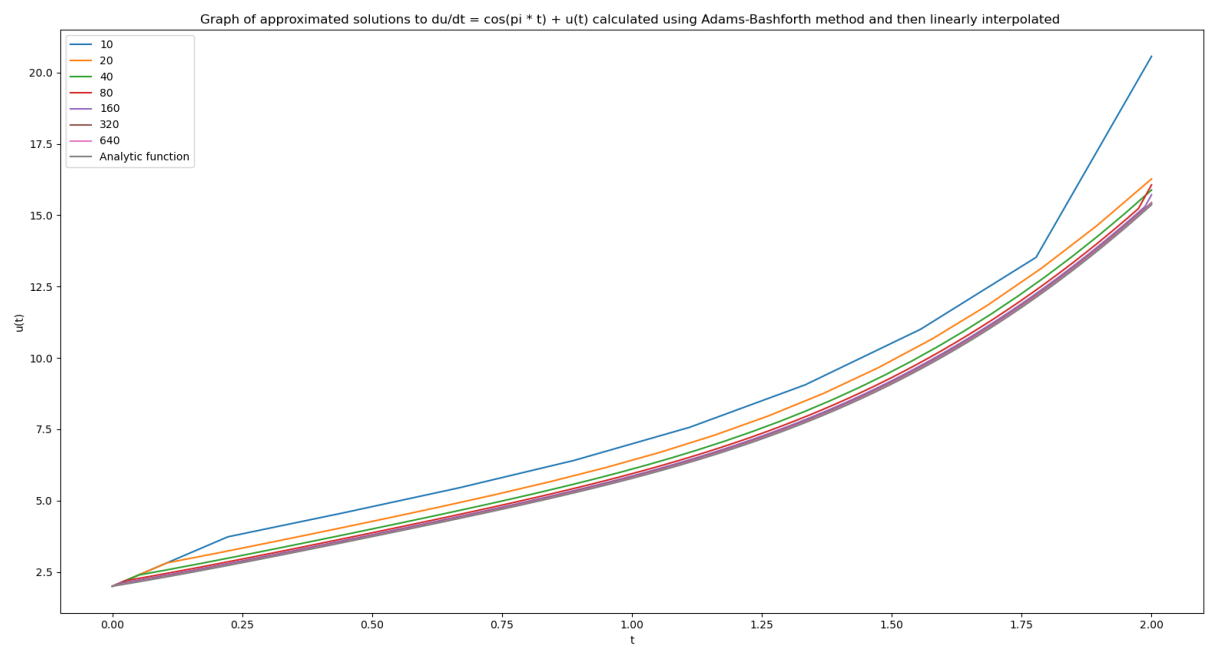
2.2 2nd order Runge Kutta



2.3 4th order Runge Kutta



2.4 Adam-Bashforth method



3 Discussion

As expected more points for all methods yields more accurate results, both regarding average accuracy and minimizing the range of the error. We can see that compared to Euler's method, the other methods converge quicker. And this makes sense as the euler method's step forward depends on the the slope ahead being the same incline or descent (derivative), as at the current point. The other methods uses the derivative around the point in order for the step forward to be more accurate rather than stepping forward relying on the slope ahead having constant derivative.

Comparing the Runge-Kutta methods and the Adam-Bashforth method yield no drastic difference. The Adam-Bashforth method implemented in the code uses two steps, so it can be optimized with higher steps. This version of Adam-Bashforth also doesn't use a predictor, so using a multistep method that uses a predictor, would probably be better than the predictor itself. For example by using the Moulton method with some Runge-Kutta would most definetly yield better results than the Runge-Kutta method by itself.

Across the methods we also se that the ratio between average accuracy and range of error seems to be the same for all methods. This perhaps relies more on the function itself, as for example a function that has oscilating features when zoomed in will greatly affect this ratio.