CS3200 – Assignment 5 Written Report
Skylar Shyu

Question 1:

Question 1 requested an implementation of Tsexact, which is the calculation of the analytical solution to the given coffee cup problem. I used the solution to the in-class assignment where we find the constant (T_c – T_s) and used it to compute the output using the following formula:

$$T(t) = (T_C - T_S)e^{-rt}$$

It should be mentioned that in here, as well as throughout the assignment:

- *T(t)* is the temperature of the coffee after time *t* has passed
- T_s is the temperature of the surrounding atmosphere (or ambient temperature)
- T_c is the temperature of the coffee
- *r* is the coefficient of the temperature 'retention ability' of the coffee cup

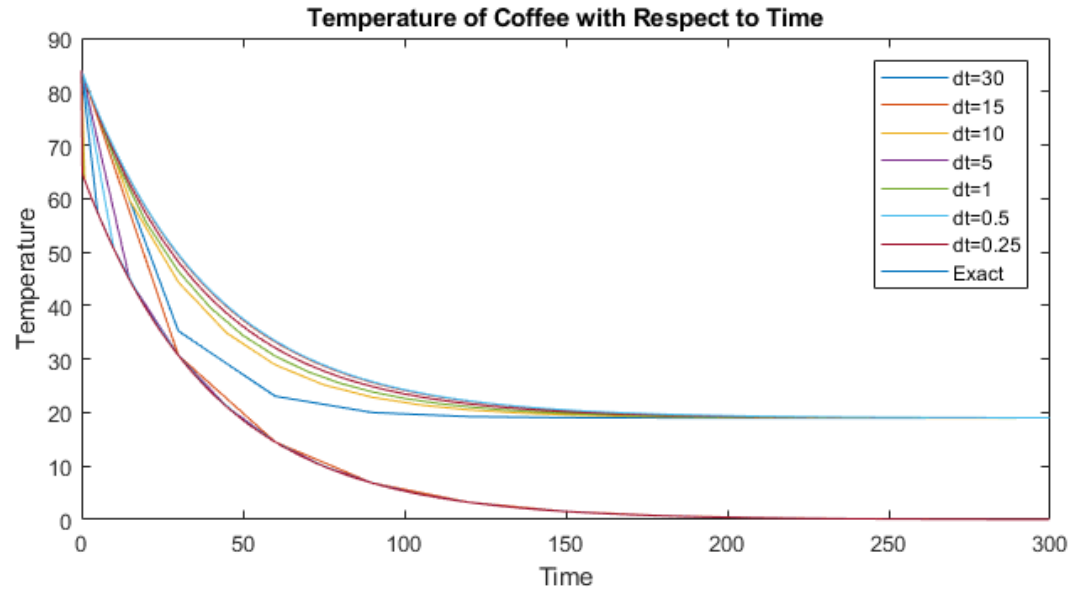
Question 2:

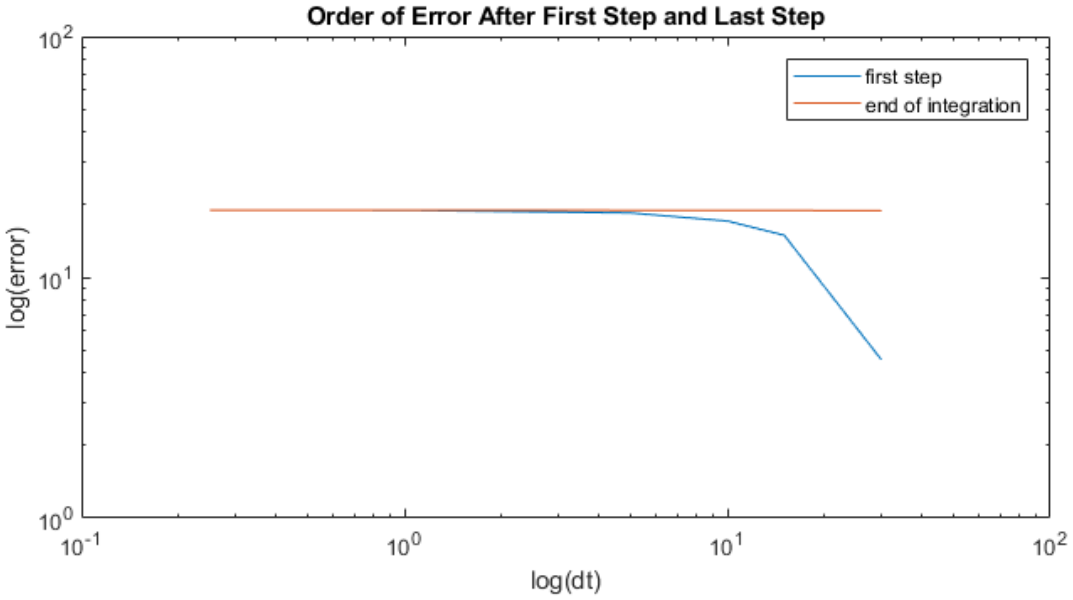
Here, the cup temperatures were plotted with different step-sizes. The temperatures were from the provided Forward Euler Method (Tc) and compared with the temperature obtained from the analytical method (tsexact). The first subplot in Figure 1 demonstrates this.

The order of error after the first step and at the end of the integration are plotted in the second subplot of Figure 1. As seen in the subplot, increases in the step-size is correlated to decreases in error. This is plausible since an increase in step size is also an increase in the granularity of temperature calculation. With more calculations being made over a smaller interval, the more accurate the results would be.

CS3200 – Assignment 5 Written Report
Skylar Shyu

Figure 1: Forward Euler Solution and Errors (r = 0.025)





Question 3:

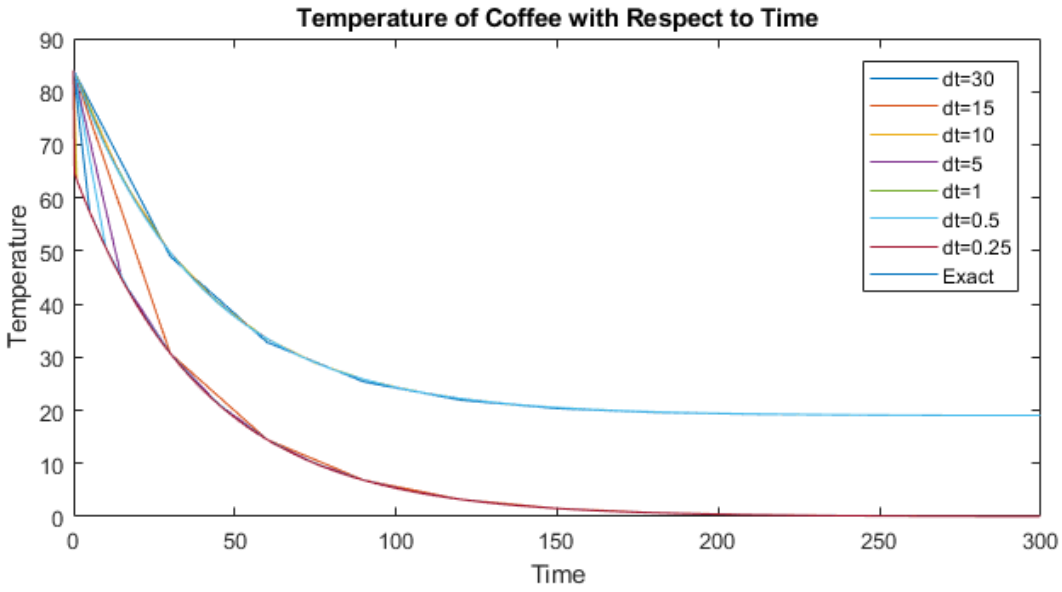
Similar to Question 1, Question 3 required a solution produced using the ODE23 method given by Moler. I used the equations given in the assignment sheet and coded them all into a function. This function also calculates the error at each step, and the information is used as the predicted error to answer Question 5.

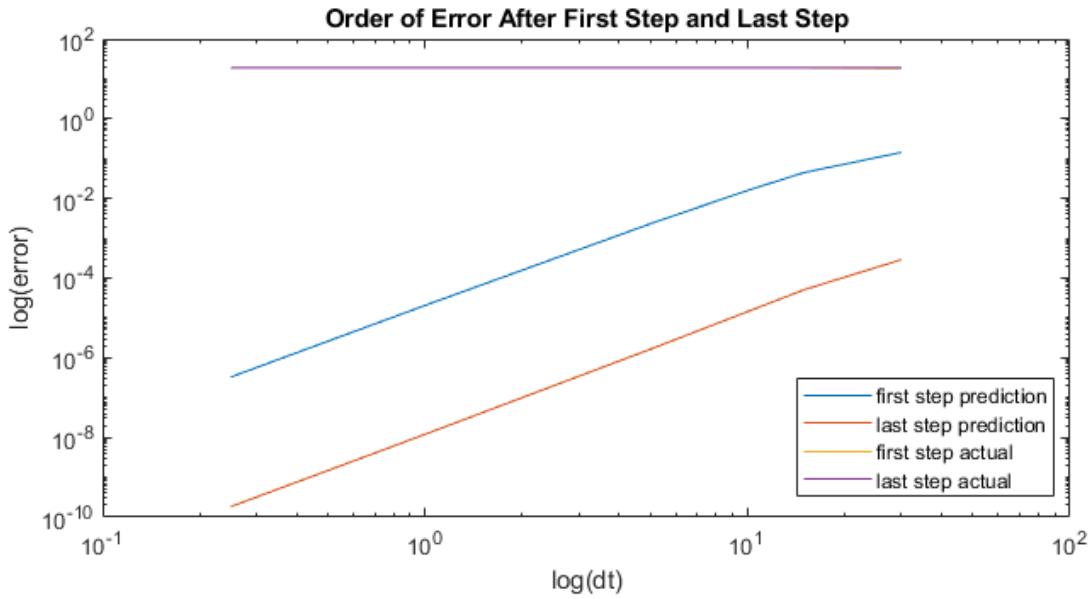
CS3200 – Assignment 5 Written Report
Skylar Shyu

Question 4:

Using the same method for Question 2, the coffee temperatures were plotted with different step-sizes. The smaller the step size, the more accurate the result was to the analytical solution. Graph may be seen below.

Figure 2: ODE23 Solutions and Errors (r = 0.025)





* because the scale is so big, the first step actual overlaps first step prediction

CS3200 – Assignment 5 Written Report
Skylar Shyu

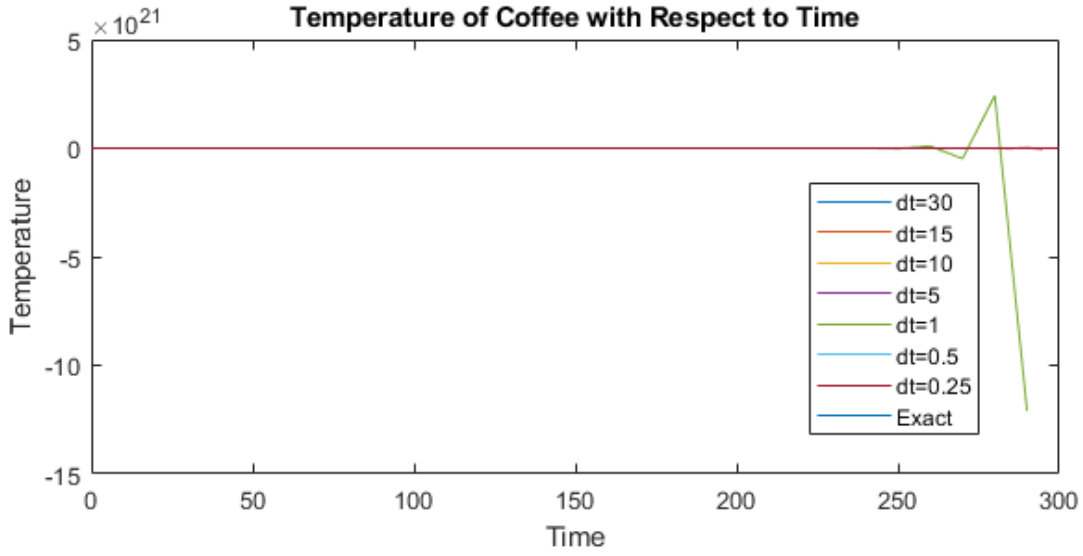
Question 5:

The bottom subplot of Figure 2 is the error plot for after the first step and at the last step. The predicted errors were calculated via the ODE23 method. The actual error numbers were calculated by taking the absolute value of the difference between the analytical solution and the solution produced by ODE23.

On the first step, the line for the actual error and predicted error overlap. Thus, there is very little difference in error for the first step, mostly due to the scale of the subplot.

Question 6:

Figure 3: Forward Euler Solution and Errors (r = 0.6)



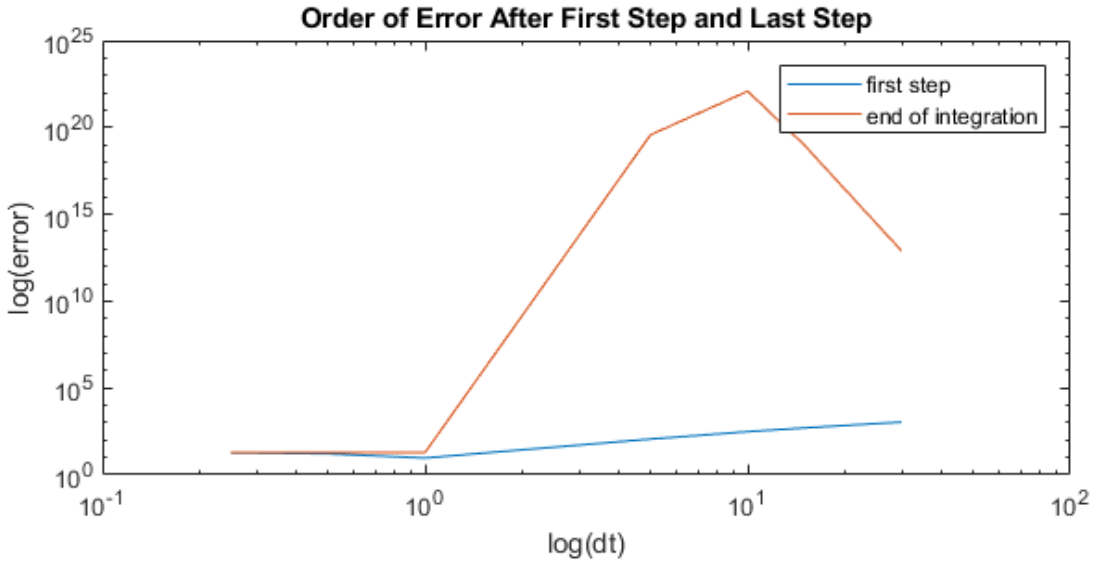
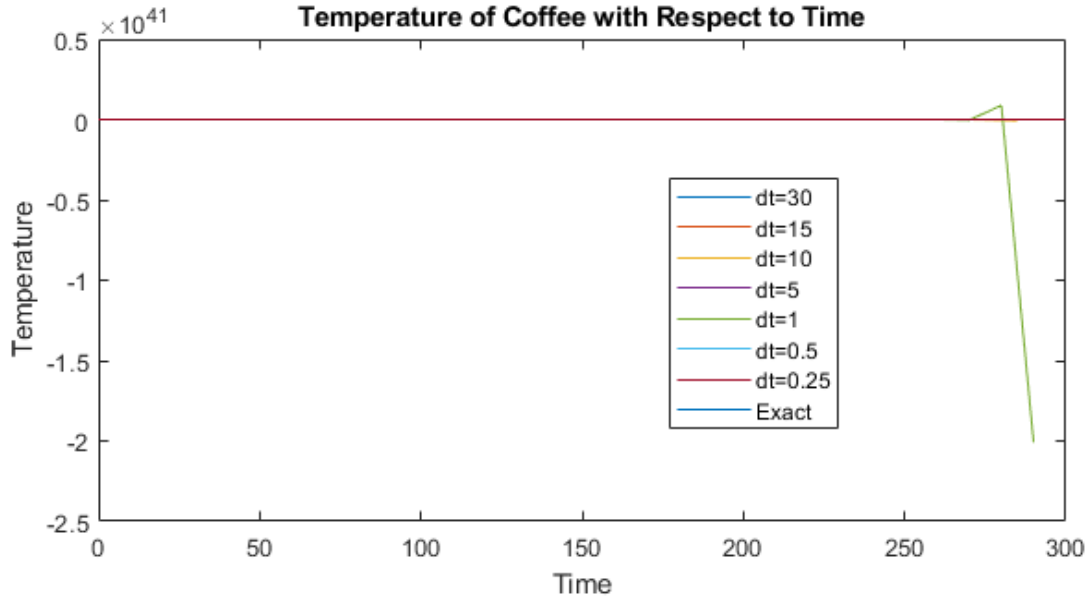
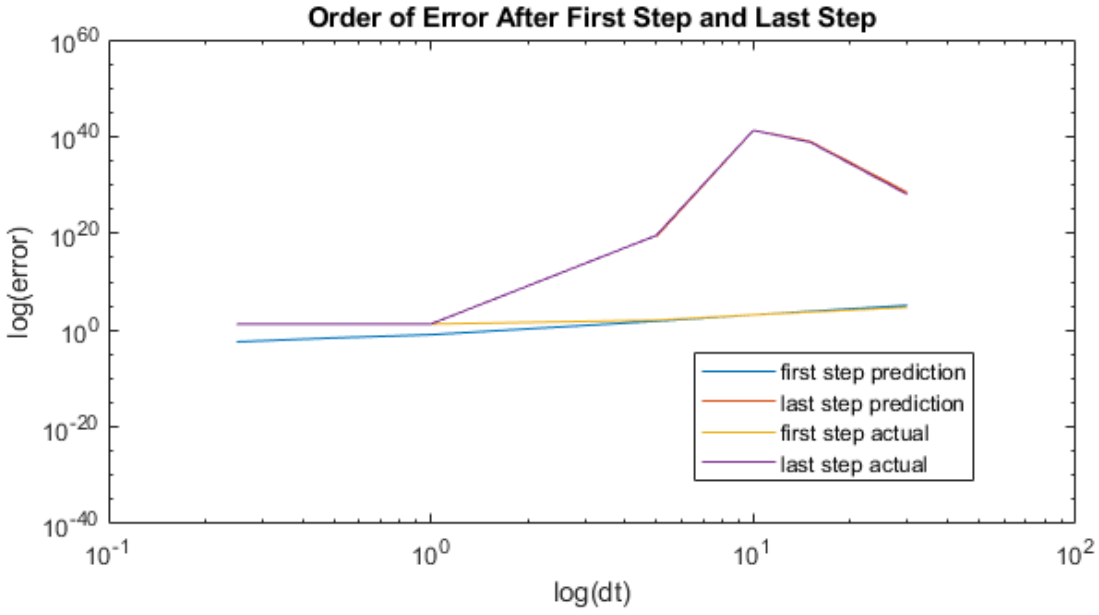


Figure 4: ODE23 Solutions and Errors (r = 0.6)





When the r-value is changed in both the Forward Euler and ODE23 generated solution, yes, the error estimators blow up for both. The first step error estimations are always reasonable (close to 0), but the last step error gets out of hand.

Sources of Information:

CS3200 – Assignment 5 Written Report
Skylar Shyu

Sources of information include the slides posted on Canvas, as well as MatLab's documentation on routines such as subplot (I've been avoiding using subplot until this assignment). Also, I did browse the Discussion forum on Canvas for tips and help, as well as going to TA hours to obtain assistance for clarification.

I did confuse myself regarding the coffee problem, and used another class's pdf to sort myself out:
http://home.lagrange.edu/jernstberger/courses/Jan2009/Intm3390/Lecture9/MATLAB-Lecture9.pdf

I did not collaborate with any classmates on this assignment.

forward_eulersol.m was provided to us to modify and obtain the solutions needed to answer the assignment. I did not write *forward_eulersol.m*. ode23_sol.m is based off *forward_eulersol.m* (as in I pretty much made a copy of *forward_eulersol.m* and modified it so it used the ODE23 method instead of the euler method).

As far as source codes went, I implemented the functions *Tsexact.m* and *ODE23.m*. For Tsexact, I worked off the in-class assignment solution to obtain the equation to implement. For ODE23.m, I went off the method given under Question 3 and essentially implemented all the S1, S2, S3, etc.