PHY407: Lab 3

Date: October 1st. 2021

Lab Partners: Brendan Halliday and Nikolaos Rizos

Contributions:

- Q1. Brendan Halliday
- Q2. Nikolaos Rizos
- Q3. Nikolaos and Brendan

Q1.a.i.

The following is the output for the approximate value of the integral using all three methods. Each iteration of the program uses $N = 2^n$ for increasing n from 3 to 11. The functions used to calculate the Trapezoidal and Simpson's rule were written in lab 2. The function used to calculate Gaussian quadratures was found in *Computational Physics* by Mark Newman. Specifically, the code found in gaussxw.py and gaussint.py were combined to define a function that approximates an integral of any well behaved function on any interval.

Integration for N = 8

Trapezoidal rule:
I = 3.138988494491089
Simpson's rule:
I = 3.1415925024587072
Gaussian Quadratures:
I = 3.1415926535191185

Integration for N = 16

Trapezoidal rule:
I = 3.140941612041389
Simpson's rule:
I = 3.1415926512248227
Gaussian Quadratures:
I = 3.1415926535897896

Integration for N = 32

Trapezoidal rule:
I = 3.141429893174975
Simpson's rule:
I = 3.141592653552836
Gaussian Quadratures:
I = 3.1415926535897905

Integration for N = 64

Trapezoidal rule:
I = 3.141551963485654
Simpson's rule:
I = 3.141592653589215
Gaussian Quadratures:
I = 3.141592653589792

Integration for N = 128

Trapezoidal rule:
I = 3.141582481063752
Simpson's rule:
I = 3.1415926535897847
Gaussian Quadratures:
I = 3.141592653589793

Integration for N = 256 points
Trapezoidal rule:
I = 3.1415901104582815
Simpson's rule:
I = 3.141592653589794
Gaussian Quadratures:
I = 3.141592653589792

Integration for N = 512

Trapezoidal rule:
I = 3.1415920178069214
Simpson's rule:
I = 3.141592653589787
Gaussian Quadratures:
I = 3.1415926535897967

Integration for N = 1024

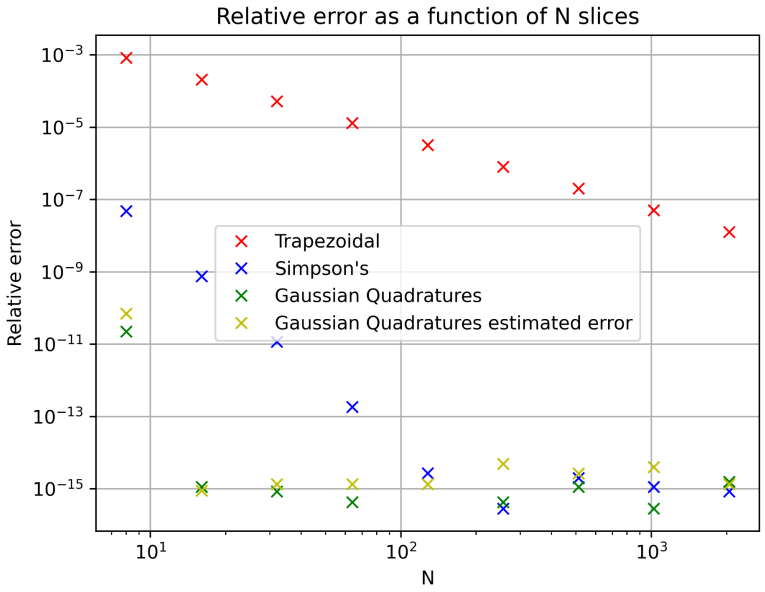
Trapezoidal rule:
I = 3.141592494644072
Simpson's rule:
I = 3.1415926535897896
Gaussian Quadratures:
I = 3.141592653589794

Integration for N = 2048

Trapezoidal rule:
I = 3.1415926138533594
Simpson's rule:
I = 3.141592653589796
Gaussian Quadratures:
I = 3.141592653589798

Q1.a.ii.

The following is is the output comparing the relative error for each N of all three methods:

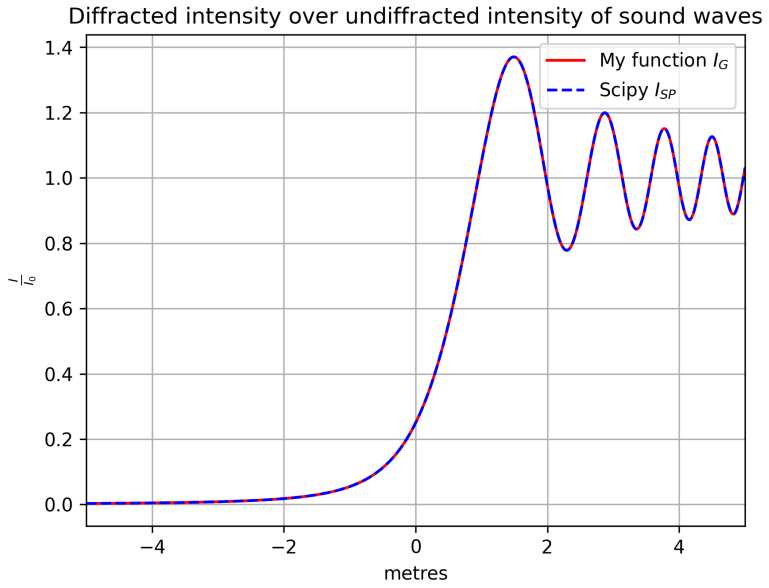


In the above graph, we see a clear downward trend for the relative error using the trapezoidal rule. A similar downward trend can be seen using Simpson's rule. However, this trend is steeper and implies that Simpson's rule achieves greater accuracy faster than the trapezoidal rule. This trend continues until N = 256. From N>256, the relative error plateaus at around 10^(-15) where the machine's precision has been reached.

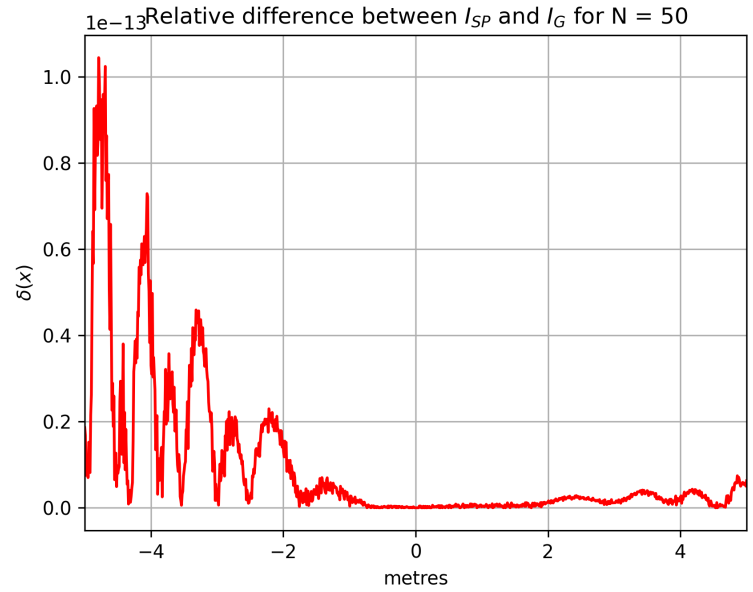
Looking at the relative error for Gaussian quadratures, we see a similar downward trend. Here, the slope is even steeper than Simpson's rule. However, we reach the machine's precision limit faster than Simpson's rule and more accurate results are not possible beyond N = 16. The estimated error calculated is plotted alongside the relative error for Gaussian quadratures. Qualitatively, the estimation is reasonably similar to that of the relative error. However, both the estimation and relative error straddle the limit of the machine's precision, thus it is difficult to compare the estimation's accuracy.

Q1.b.i.

Below is an overlaid comparison between my calculation of I/I_0 using Gaussian quadratures and Scipy's calculation:

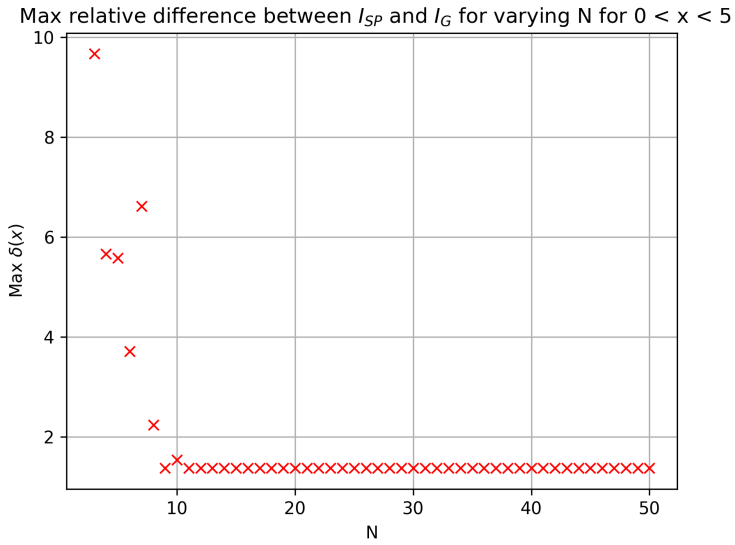


From this view, they seem to be pretty close. Below is a graph of the relative difference between the two graphs for each x.



And they are very close. Since I_SP is very close to zero for values less than zero, the relative difference is large.

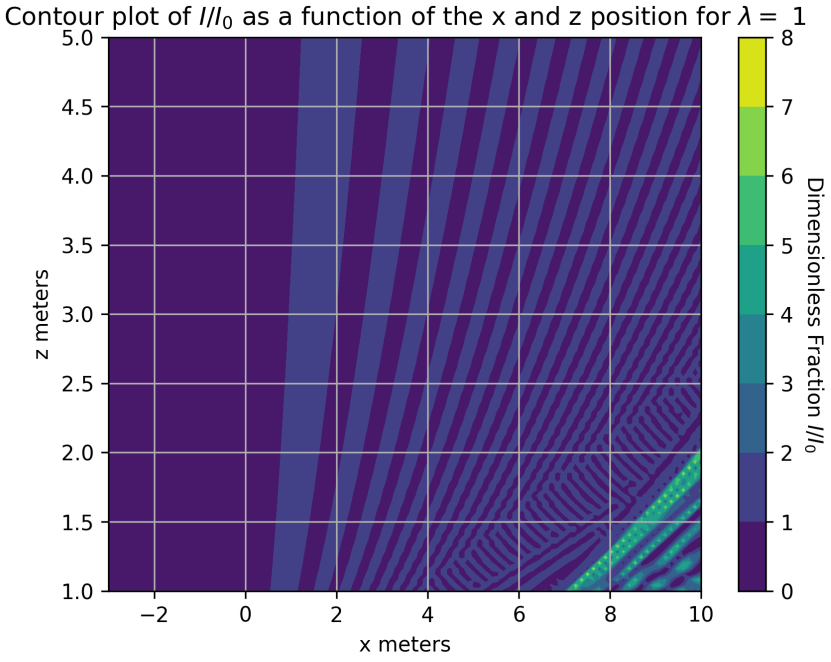
Q1.b.ii.

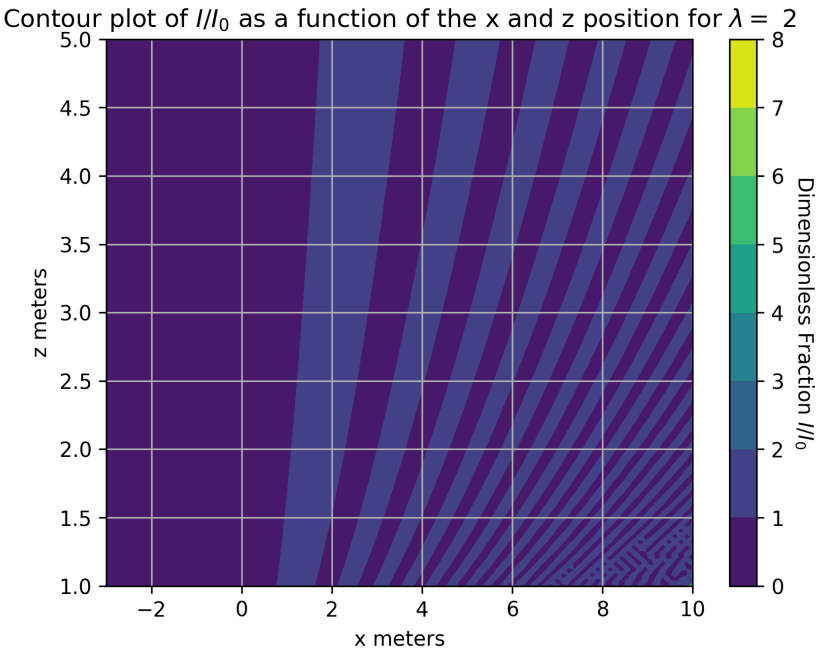


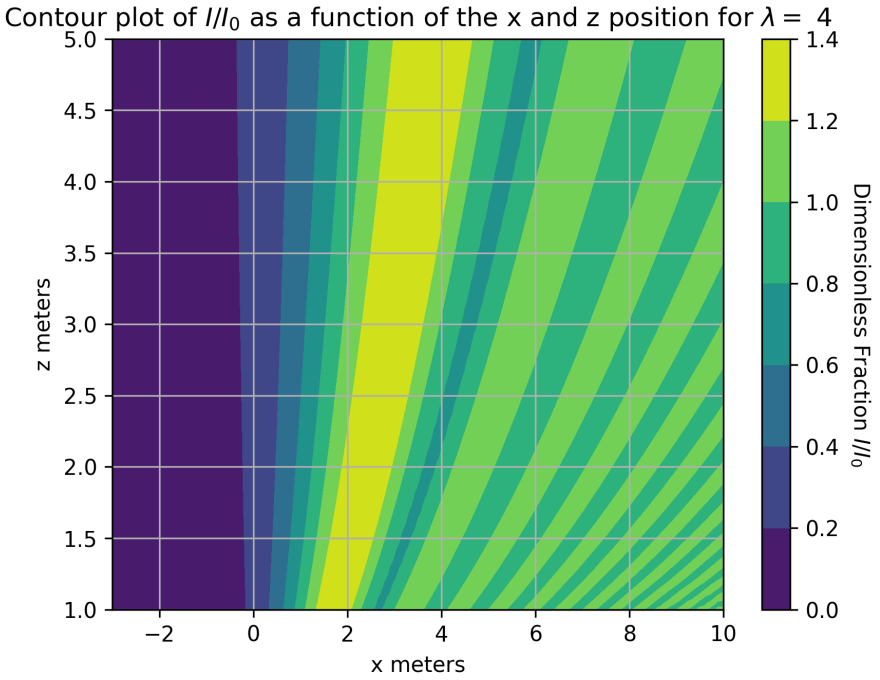
Roughly speaking, for N=11 and greater , we achieve a sufficiently small relative maximum difference between Gaussian quatradures and the expected value given by Scipy.

Q1.c.

Below are the contour plots for lambda = 1m, 2m and 4m. The domain is defined on 1m < z < 5m and -3m < x < 10m.







From the above images, we see how the barrier creates curved wavefronts as expected. However, depending on the value of the wavelength, we see more constructive or destructive interference between overlapping wavefronts. This feature is especially noticeable in the bottom right hand corner of the lambda = 1m graph.

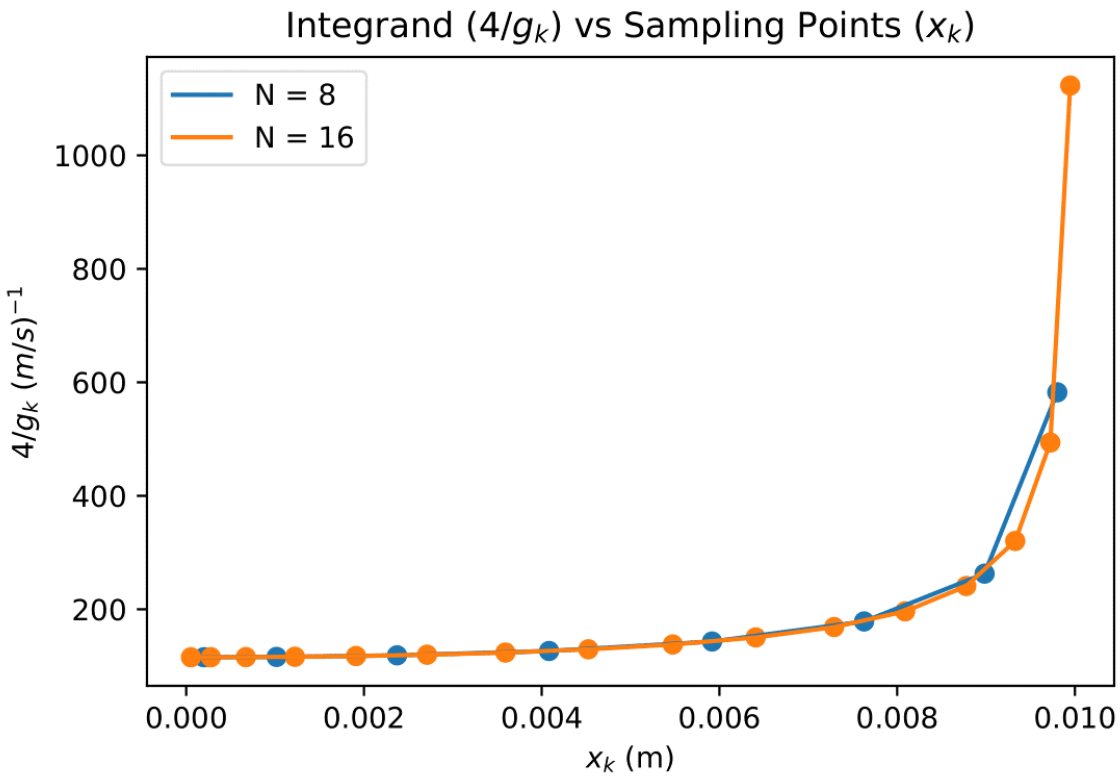
Q2.a.

The true value of the period in the classical limit for an initial displacement , was computed using , (m = 1kg, k = 12 N/m), and was evaluated to be T=1.8137993642342176 (s).

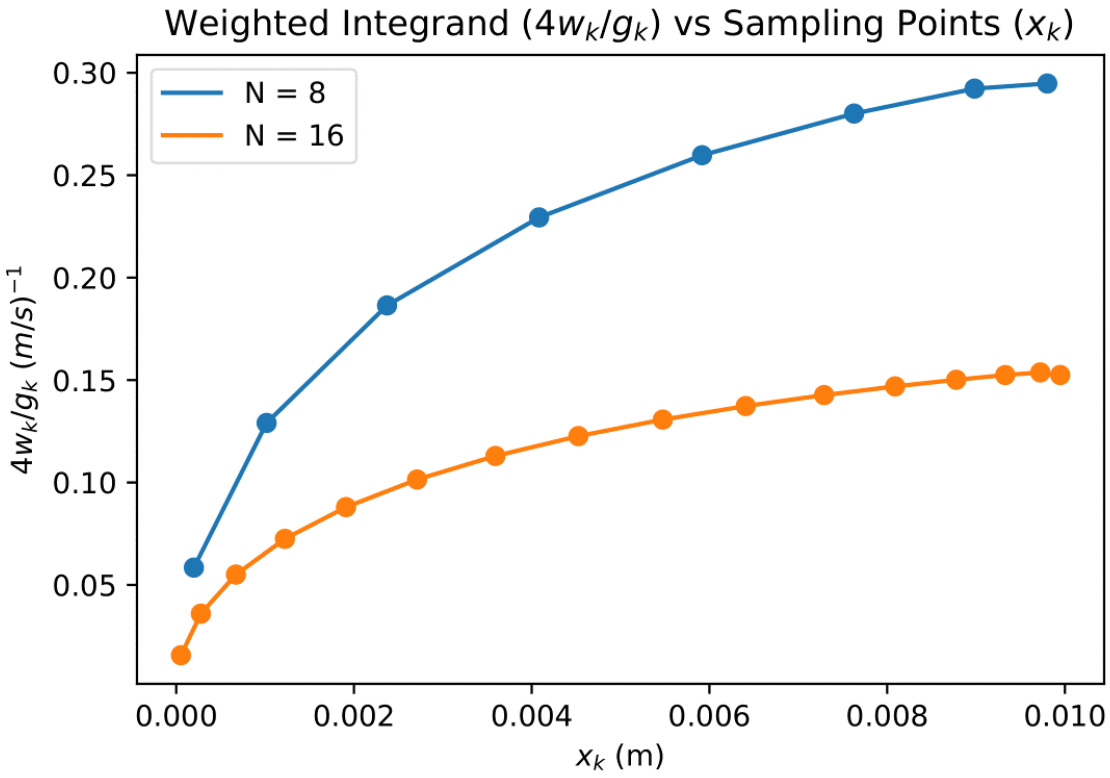
When using the method of Gaussian quadrature to estimate the integral of Eq.(7) (0m<x<0.01m) for a mass and spring system equal to the above, the estimate of the period, and the relative error of the period estimate (compared to the expected value of T) for N = 8 samples used in the Gaussian quadrature method, were evaluated to be: 1.7301762343365563 (s) and 0.046103847838190645, correspondingly.

When for the same system and the same method of estimating the period integral, we used N = 16 samples, the estimate of the period for the same initial displacement, and the relative error of the estimate compared to the true expected value of the period, were evaluated to be: 1.770715490242243 (s) and 0.023753384658487007, correspondingly. The relative error for N = 16 seems to be half that of N = 8.

We now plot the value of the integrand over the sample points, and the weighted integrand over the same sample points derived by evaluating the zeroes of the Legendre polynomials.



We observe that the two integrands take about the same values. As the value of the upper bound of the integral is approached though for N = 16 samples, the integrand seems to take a value twice as large as the values of N = 8. The sample points move closer on the edges (on the x-axis) for both sample sizes.



In this plot, we see that adding the weights makes the value at each sample point noticeably smaller. We also observe that the weighted integrand values for N = 8 are bigger when compared to the weighted integrand values for N = 16. This difference between the 2 seems to also increase, in favor of N = 8. Again, we observe that the sampling points are more closely spaced at the edges of the x-axis. As the limit of integration is approached, the weighted integrand values for N = 16 are about half as big as the ones for N = 8. This is interesting as the normal integrand for N = 16 took a value twice as big as we approached the displacement limit, but in contrast, when approaching the same limit, the weighted integrand takes values half as large as the weighted integrand for N = 8.

It seems that this method captures more points at the extremes of the integration interval, placing more focus on how the function behaves near its bounds of integration. It also seems that adding the weight to the method decreases how much each value of the integrand affects the end result, as we end up summing more but smaller values instead of less but larger values. This might reduce the error of the integration, as we end up summing more smaller values (which in turn have less error) rather than summing less, larger, more inaccurate values. Also, increasing the weight makes the weighted values smaller, also indicating that summing more smaller values, might result in smaller error (as we saw that the error in the larger sample size is half as big as the error in the smaller sample size).

Q2.b.

2) b) For a classical particle on a spring:

$$m\frac{d^2x}{dt^2} = -kx \Rightarrow m\cdot x'' = -kx \Rightarrow mx'' + kx = 0 \Rightarrow x'' + \frac{k}{m}x = 0 \ (1)$$

characteristic polynomial of (1):

$r^2 + \frac{k}{m} = 0 \Rightarrow r^2 = -\frac{k}{m} \Rightarrow r = \pm\sqrt{\frac{k}{m}}\,i$. General solution to (1):

- $x(t) = A\cdot\cos\left(\sqrt{\frac{k}{m}}\,t\right) + B\sin\left(\sqrt{\frac{k}{m}}\,t\right)$ (2)
- $x'(t) = u(t) = -A\sqrt{\frac{k}{m}}\sin\left(\sqrt{\frac{k}{m}}\,t\right) + B\sqrt{\frac{k}{m}}\cos\left(\sqrt{\frac{k}{m}}\,t\right)$ (3)

we know that at $t_0 = 0$, $x(0) = x_c$ and $u(0) = 0$:

(2) $\Rightarrow x(0) = x_c \Rightarrow A = x_c$

(3) $\Rightarrow u(0) = 0 \Rightarrow B\sqrt{\frac{k}{m}}\cos(0) = 0 \Rightarrow B = 0$

thus (2) and (3) now become:

$x(t) = x_c\cos\left(\sqrt{\frac{k}{m}}\,t\right)$ and $u(t) = -x_c\sqrt{\frac{k}{m}}\sin\left(\sqrt{\frac{k}{m}}\,t\right)$

we also know that at t_1, $x(t_1) = 0$, $u(t_1) = c$

thus: $x(t_1) = 0 \Rightarrow x_c\cos\left(\sqrt{\frac{k}{m}}\,t_1\right) = 0$ but we are told that $x_c > 0$, thus $\cos\left(\sqrt{\frac{k}{m}}\,t_1\right) = 0$. Since we are in the first quarter of the oscillation it has to hold that:

$\cos\left(\sqrt{\frac{k}{m}}\,t_1\right) = \left(\frac{\pi}{2}\right) \Rightarrow \sqrt{\frac{k}{m}}\,t_1 = \frac{\pi}{2} \Rightarrow \boxed{t_1 = \frac{\pi}{2}\sqrt{\frac{m}{k}}}$

now: $u(t_1) = c \Rightarrow -x_c\sqrt{\frac{k}{m}}\sin\left(\sqrt{\frac{k}{m}}\cdot\sqrt{\frac{m}{k}}\cdot\frac{\pi}{2}\right) = c \Rightarrow$

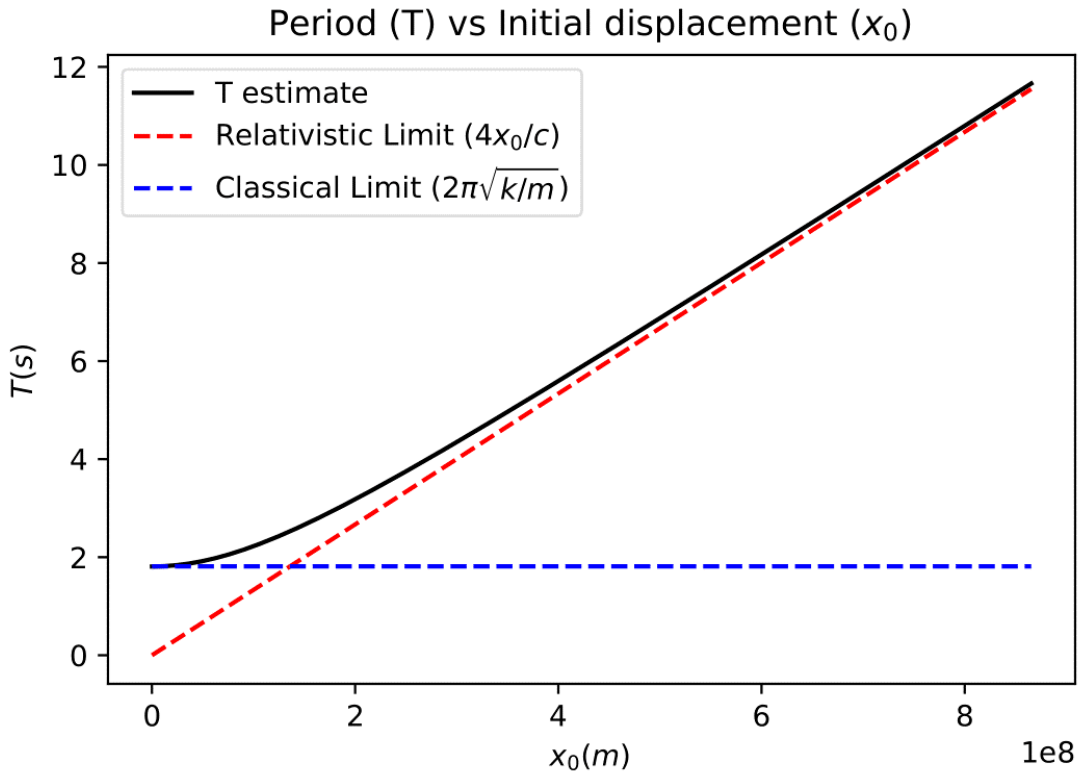
$\Rightarrow -x_c\sqrt{\frac{k}{m}} = c \Rightarrow -x_c = c\sqrt{\frac{m}{k}} \Rightarrow x_c = -c\cdot\sqrt{\frac{m}{k}}$. The negative sign indicating that initially the particle was displaced to the left (negative) amplitude position.

thus: $\boxed{x_0 = x_c = c\sqrt{\frac{m}{k}}}$

Q2.c.

For the small amplitude case (), the percentage error of the period estimate was evaluated to be: 0.1954853607529289 4 % for N = 200 samples. We expect the Gaussian quadrature method to give highly accurate results, especially for large sample sizes. The reason this is not the case here (as we can see, even with quite large sample size the percent error is quite larger than expected), is because the function we are integrating is not a polynomial. It is the square root of a quotient of polynomials. Thus, the actual estimate of the integral is not going to be nearly as accurate as this method would prove to be for an actual polynomial.

In any case, the method manages to capture the classical period value with relative accuracy (they are close enough), and we can thus see how the period changes according to the initial displacement. Below we plotted the period estimate for values ranging from 1m to values comparable to the numerical value of c.



As previously stated, even though the estimates of the period are not the most accurate, the method manages to capture the constant linear behavior of the period for small displacements, as well as the increasing linear behavior of the period for large displacements. For small displacements the estimates of the period seem to perfectly coincide with the classical (non-changing) value of the period computed in part a) of the exercise, and as the displacement increases, the period tends to asymptotically coincide with the relativistic form of the period. The method thus is not useful for accurately capturing singular values of the period in either the relativistic, or the classical cases, but it does manage to correctly capture the general behavior of the period as we shift from the one case (linear) to the other (relativistic), according to the change in displacement.

Q3.b.

This is the printed output for the code:

The 17 step sizes are: [1.e-16 1.e-15 1.e-14 1.e-13 1.e-12 1.e-11 1.e-10 1.e-09 1.e-08 1.e-07
1.e-06 1.e-05 1.e-04 1.e-03 1.e-02 1.e-01 1.e+00]

The forward difference estimates of the derivative value of f(x) for x = 0.5, corresponding to the above step values are: [-1.11022302 -0.77715612 -0.77715612 -0.77937656 -0.77882145
-0.77879925
-0.77880147 -0.7788008 -0.77880079 -0.77880082 -0.77880117 -0.77880468
-0.77883972 -0.77918953 -0.78262986 -0.81124457 -0.67340156]

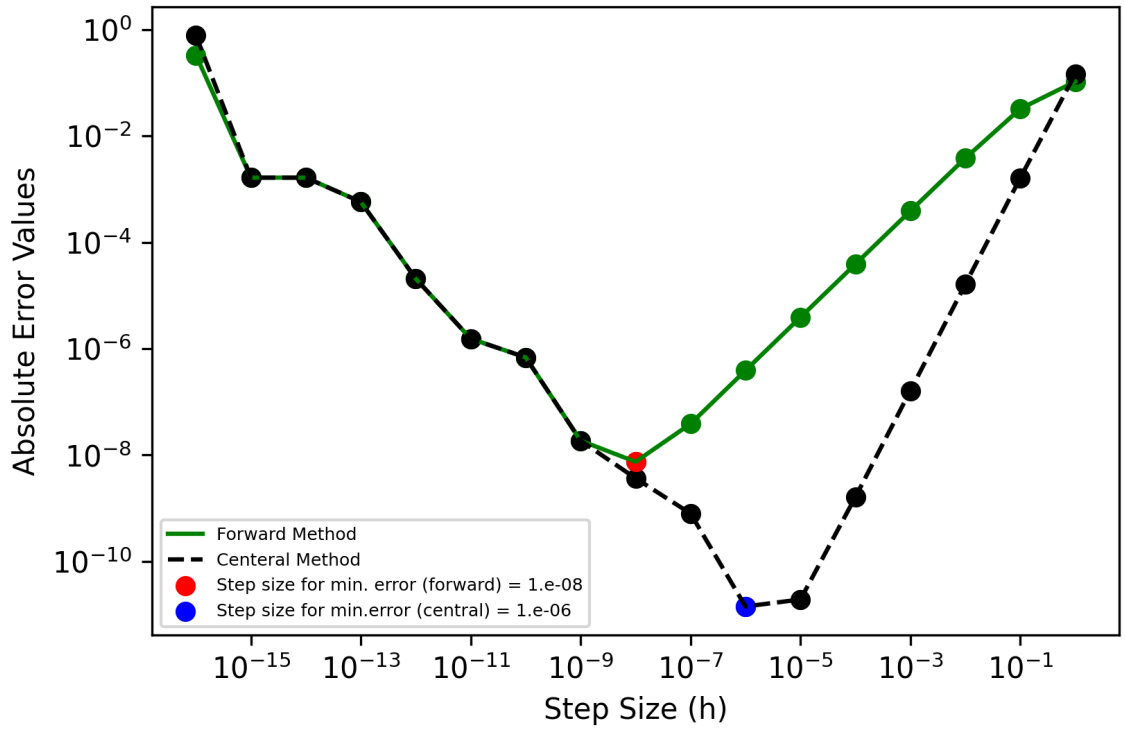
The errors between the analytic method and the numerical method are: [3.31422242e-01
1.64466583e-03 1.64466583e-03 5.75780215e-04
2.06687031e-05 1.53575735e-06 6.84688699e-07 1.85548844e-08
7.45265416e-09 3.85388988e-08 3.89369375e-07 3.89393268e-06
3.89335494e-05 3.88751359e-04 3.82907406e-03 3.24437869e-02
1.05399225e-01]

Q3.c.

The step size corresponding to the minimum error (forward method) is indeed h = 1e-08 as shown in the plot given in the part d. Looking at the plot, we see that the first term (rounding error) of equation 5.91 dominates for values of h less than 1e-08 since the shape looks like 1/h. The second term (truncation error) dominates for values greater than h = 1e-08 since the graph is linear with a positive slope.

Q3.d.

Below is the plot for absolute error values for both methods with minimum error indicated by the blue and red dot.



Clearly both methods increase for smaller values of h since we are dividing by values of h that approach zero. For values of h greater than the minimum, the error in the central method shows parabolistic characteristics. Equation 5.99 from Newman has a second term equal to (1/24)h^2|f'''(x)|, which is clearly dominant for values of h greater than the minimum. For values of h greater than the minimum, the error in the forward method shows linear characteristics instead.

As shown from the plot, the central difference method does not always beat the forward difference method.

Bibliography:

Newman, M. (2012). *Computational physics*. Mark Newman.