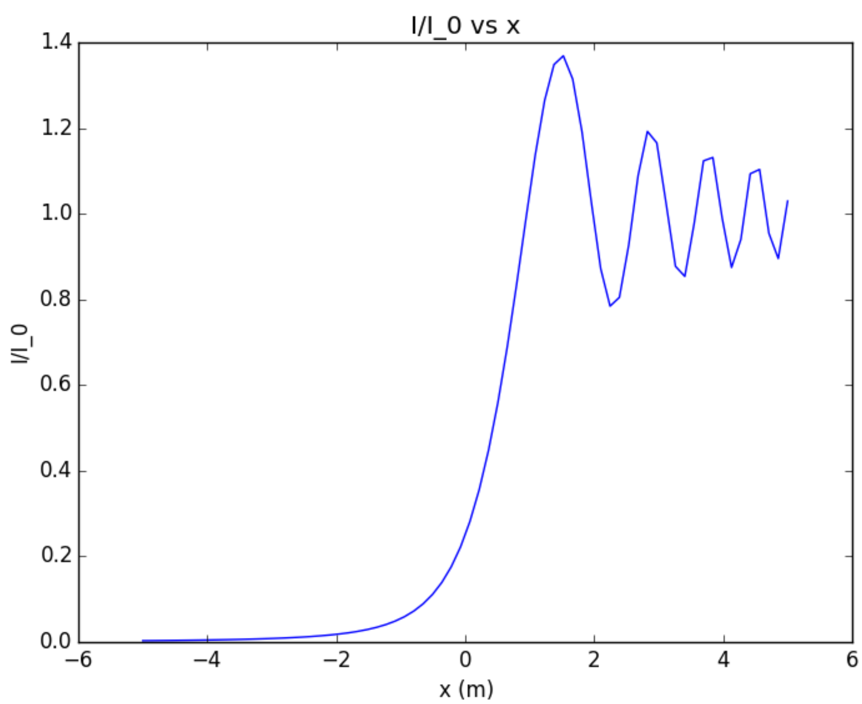
PHY407H1 Lab 3

Eric Yeung*
Department of Physical Sciences, University of Toronto, Toronto M1C 1A4, Canada
(Dated: October 5, 2015)

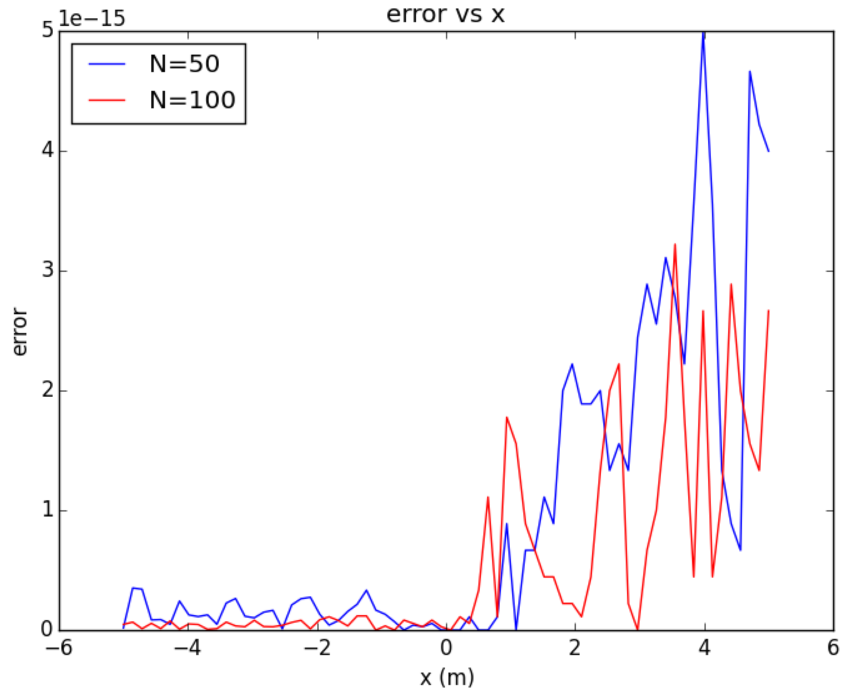
QUESTION 1

a) See lab3_q1a.py by Chi.



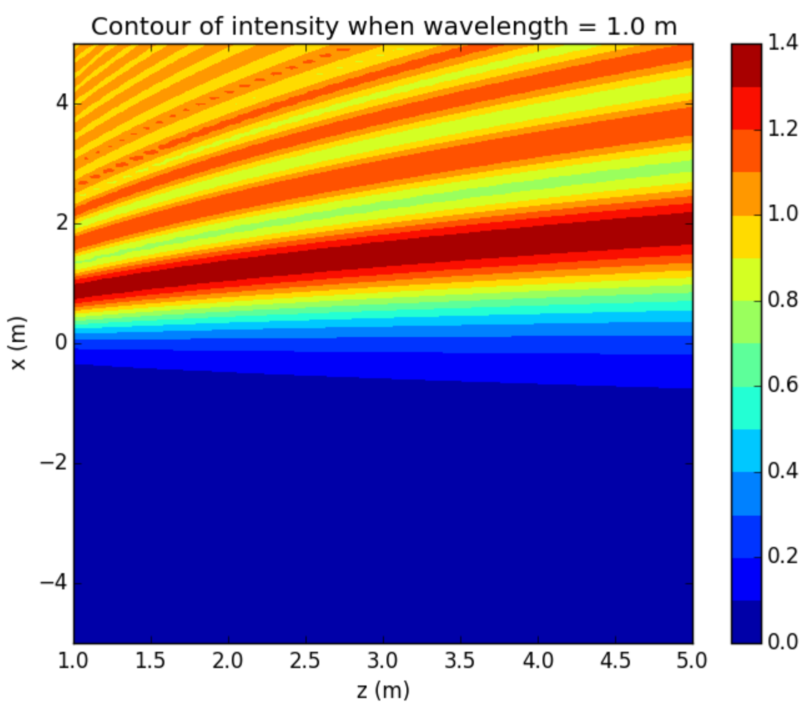
* eric.yeung@mail.utoronto.ca

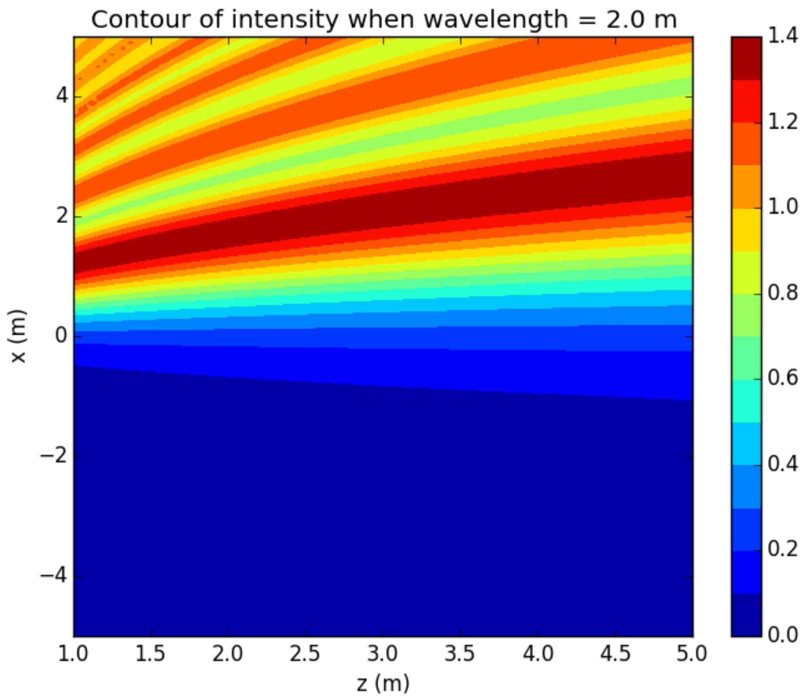
b) See lab3_q1b.py by Chi.

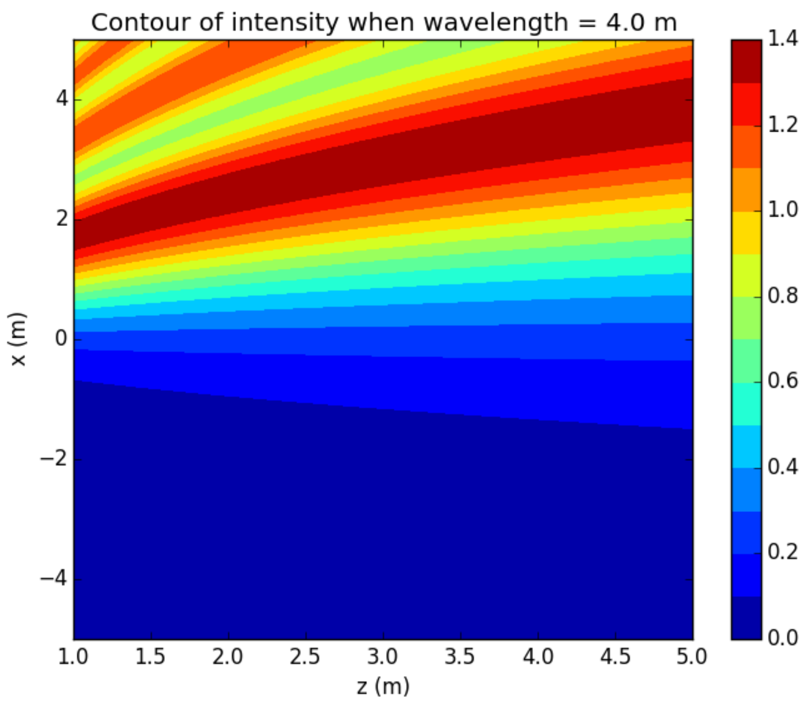


I can see from the plot that, when compared to the numerical round off error of 1e-16, the error for N = 50 points is a lot further away from the numerical round off error. The error for N = 100 points is a lot closer to 1e-16 and smaller is better in this case, and one would expect that you would get less error when one doubles the points.

c) See lab3_q1c.py by Chi.





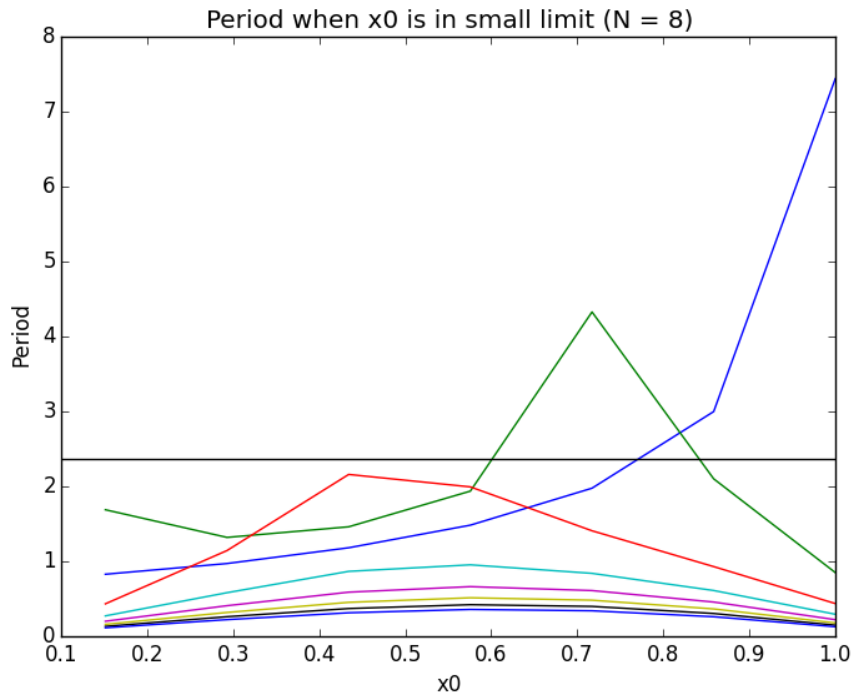


As the wavelength increases, it seems like the less intense areas disappear and are replaced by more intense areas. For the $\lambda = 1$ case, the top of the graph has a lot of yellow and orange bands. In the $\lambda = 2$ case, there are less yellow/orange bands but they are slightly thicker. In the $\lambda = 4$ case, one can only see 2 orange bands and 4 yellow bands in the graph, but they are- of course- much thicker. In a way, this resembles "zooming in" on the density plot.

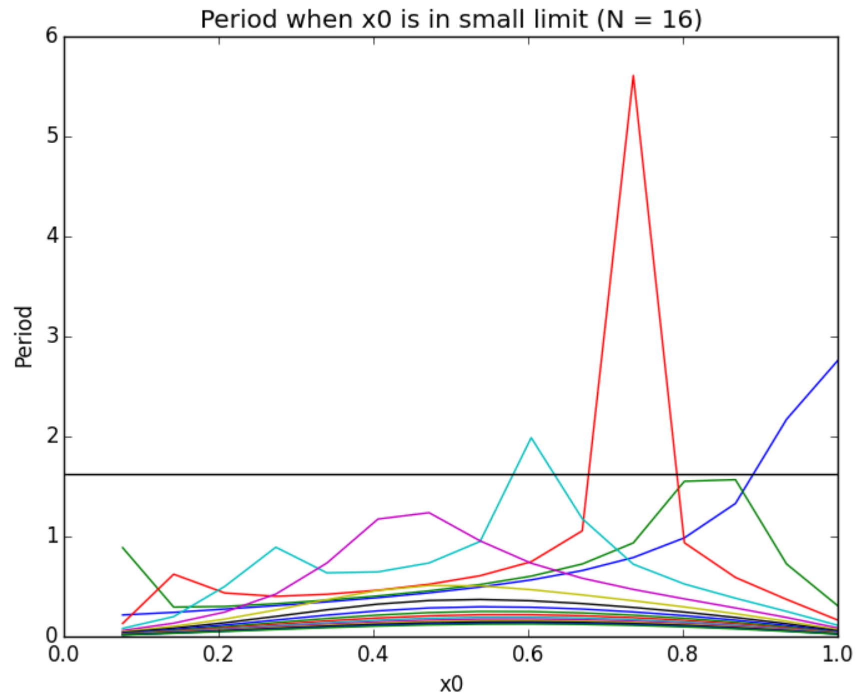
QUESTION 2

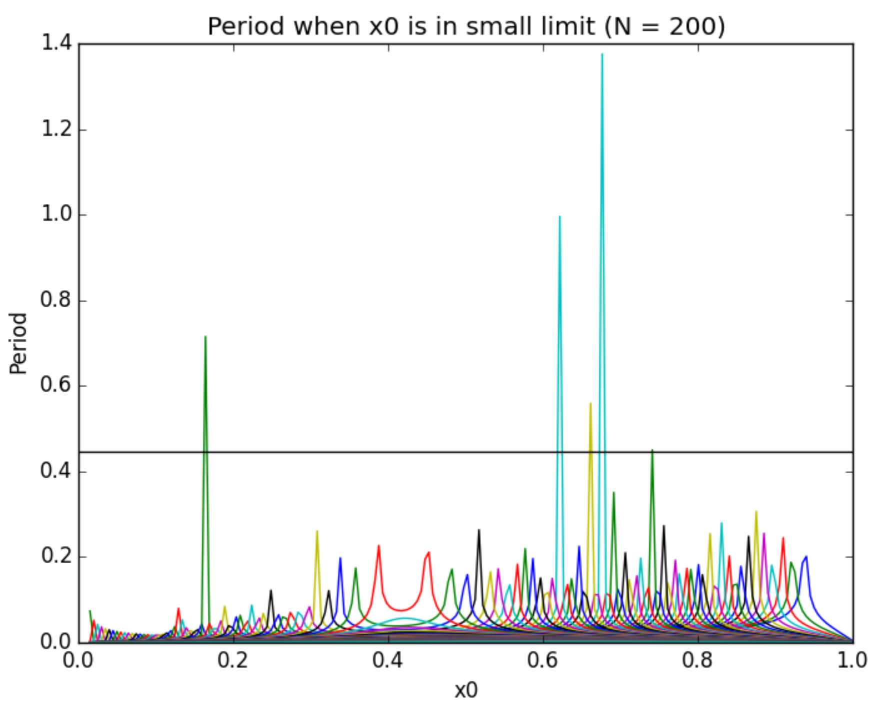
a) See lab3_q2a.py. The known classical limit is

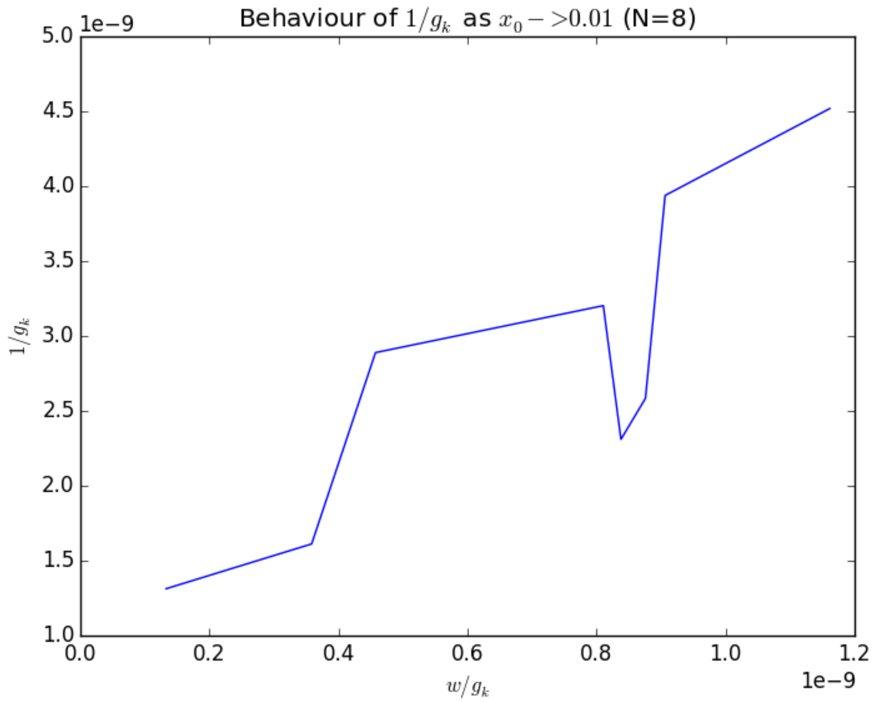
$$T = 2\pi\sqrt{\frac{m}{k}} \tag{1}$$

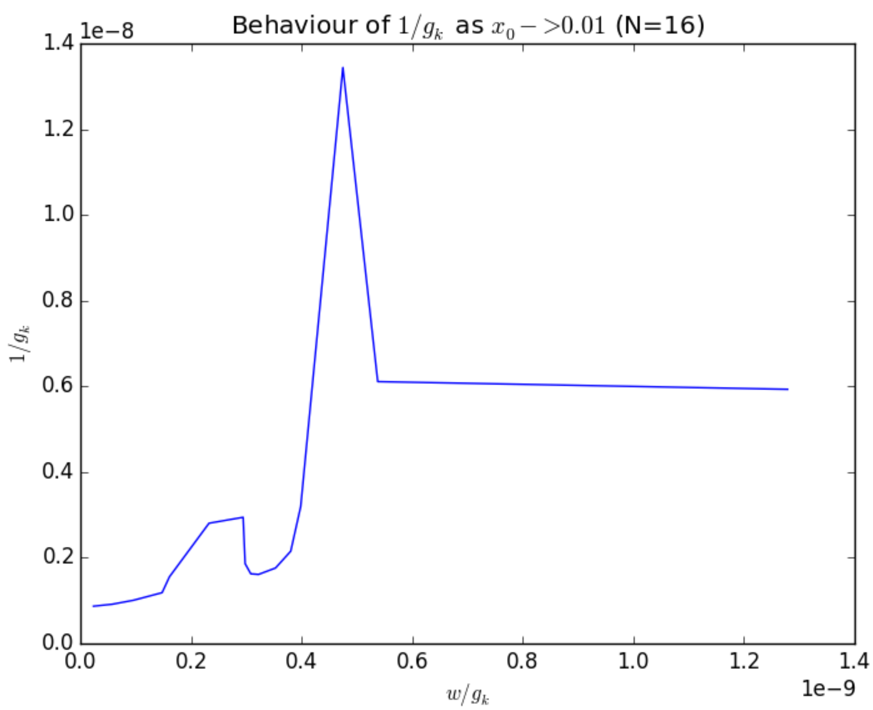


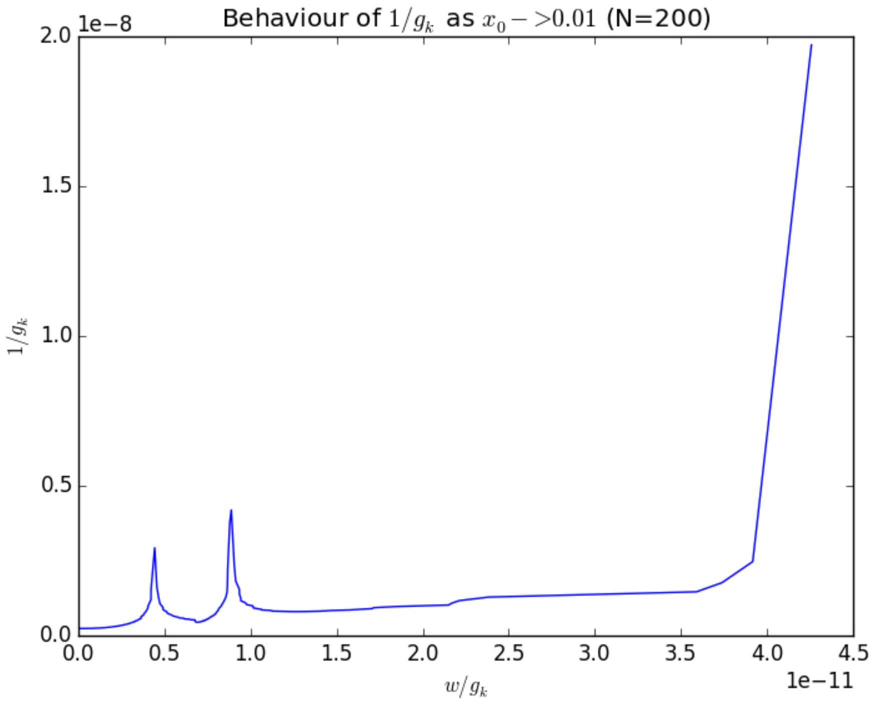
The integrals my code were giving me were around $\approx 1e-9$ in magnitude which was 1e-9 times too small, as the classical period should be 2.37482082345. Even wolfram alpha gave a value of 6.04610^{-9} for the integral. I then multiplied the period by 1e9 to correct this anomaly.


Period when x0 is in small limit (N = 16)
Period
x0
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0


Period when x0 is in small limit (N = 200)
Period
x0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0.0
0.2
0.4
0.6
0.8
1.0


Behaviour of $1/g_k$ as $x_0->0.01$ (N=8)
1e−9
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
$1/g_k$
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e−9
w/g_k


Behaviour of $1/g_k$ as $x_0->0.01$ (N=16)
1e−8
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
$1/g_k$
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e−9
w/g_k



It seems like when x_0 approaches the limit of integration, $1/g_k$ does not have as many spikes anymore. The function $1/g_k$ is not monotonic, but it does tend to increase or get further away from 0. As g(x) gets further away from 0, the integral converges. And the more the integral converges, the more accurate the calculation is.

For the N = 8 case, the fractional error is estimated to be ≈ 0.8. For the N = 16 case, the fraction error is estimated to be ≈ 0.5 as the fractional error should decrease when one increases the number of points.

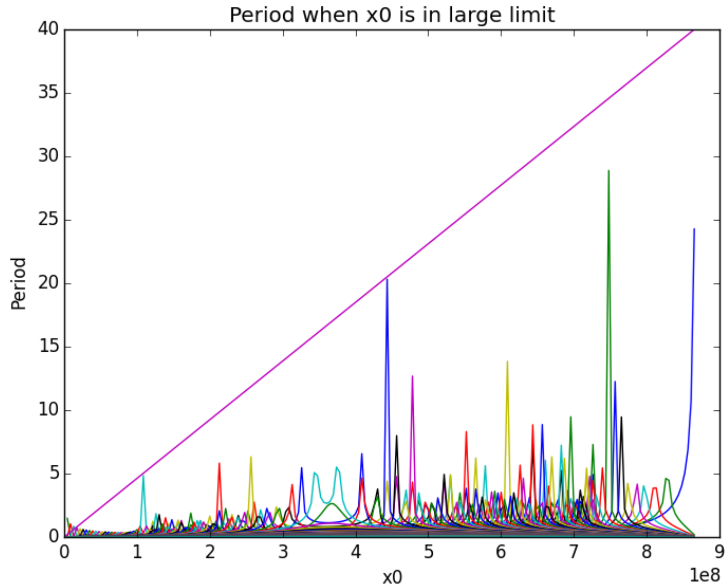
b) See lab3_q2b.py. I estimate the percentage error for the small amplitude N = 200 case to be of the magnitude $\mathcal{O}(10^{-4})$.

The relativistic limit from the second lab is

$$T \to \frac{4x_0}{c} \tag{2}$$

The plot of x_0, where $1 < x < 10x_c$, and the period T is shown below.

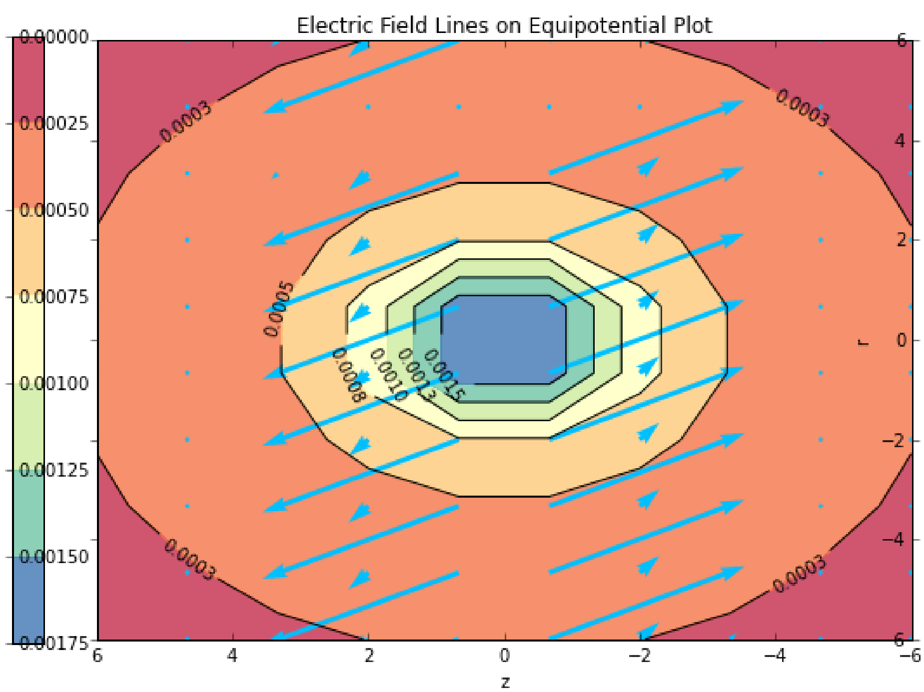
As one can see, unlike in the classical limit, the relativistic limit grows as a function of x_0. There are also seems to be more sharp peaks in the case where $1 < x_0 < 10x_c$.



QUESTION 3

Referring and using the potential function from Lab 2, I calculated the gradient of the potential and subsequently the electric field. See lab3_q3.py.

For the first points in the array [0], the forward difference formula was used for both the r and z partial derivatives. For the last points in the array [-1], the backward difference formula was used. Then, for the values in between [1:-2], central differences were used. Because $h \to 0$, I had to make the delta small but not too small. This is because python interprets it as a 0 at around $\approx 1e-16$.



The value of 1 for the spacing was used because that gave me the best resolution where I could see all the vectors.