Ay190 – Worksheet 03 Writeup
David Vartanyan
Date: January 21, 2014

1 Exercise 1

For problem 1, I follow Professor Ott's example in class. I also define the integrands as a list with the ith element in the list corresponding to the ith integral interval in case one would be interested in plotting the integral. We will see, as expected, that midpoint and trapezoidal integral estimates have similar errors while Simpson's rule has much smaller error.

1.1 Exercise 1a

The exact integral is:

$$\int_0^{\pi} \sin(x)\,\mathrm{d}x = 2 \tag{1}$$

Below, h indicates number of integral steps used. The py file in Github corresponds to $h = 100$. This is easily changed.

1.1.1 Integrals

	Midpoint	Trapezoid	Simpson's
h=200	2.0000207690	1.99995846214	2.00000000004
h=100	2.00008391911	2.0000000007	1.99983216389

1.1.2 Errors

	Midpoint	Trapezoid	Simpson's
h=200	2.07689960372e-05	-4.15378626695e-05	4.31352731312e-11
h=100	8.39191093354e-05	-0.000167836106008	7.04220681769e-10

1.1.3 Convergence

Midpoint	Trapezoid	Simpson's
0.247488280103	0.24749062438	0.061252494066

Indeed, note that for the midpoint and trapezoid estimates, both of which are 2nd order, doubling the number of steps (or halving the step size, alternately) results in 1/4th the error, as expected.

For the Simpson's estimate, which is 4th order, doubling the number of steps results in 1/16th or .0625 the error. Ths our results are convergent.

1.2 Exercise 1b

The exact integral is:

$$\int_0^{\pi} x \sin(x)\, \mathrm{d}x = \pi \tag{2}$$

As for 1a: below, h indicates number of integral steps used. The py file in Github corresponds to $h = 100$. This is easily changed.

1.2.1 Integrals

	Midpoint	Trapezoid	Simpson's
h=200	3.14162527745	3.14152740607	3.14159265366
h=100	3.14172447342	3.14132901725	3.1415926547

1.2.2 Errors

	Midpoint	Trapezoid	Simpson's
h=200	3.26238626873e-05	-6.52475221039e-05	6.77542466576e-11
h=100	0.000131819828693	-0.00026363633882	1.10618980642e-09

1.2.3 Convergence

Midpoint	Trapezoid	Simpson's
0.247488280108	0.24749062438	0.0612501093975

Thus our results are convergent by the same argument presented for 1a.

2 Exercise 2

Consider electron number density:

$$n_e = \frac{8\pi (k_B T)^3}{(2\pi\hbar c)^3} \int_0^{\infty} \frac{x^2}{e^x + 1}\, \mathrm{d}x \tag{3}$$

2.1 Exercise 2a

Integrating equation (3) for 100 nodes gives $n_e = 1.90215974998 \times 10^{35}$ electrons per cm^3.

I solve using the Gauss-Chebyev method.

Below we tabulate our integrand without the numerical cofactor. Our github py file solves for $n = 2$ nodes.

Nodes	Integrand
n=5	1.80202715322
n=10	1.80309513198
n=100	1.80308535474

Nodes	Integrand Error
n=5	-0.00105820157676
n=10	9.77717911321e-06
n=100	-6.05464567371e-11

Thus our model is convergent.

2.2 Exercise 2b

We write our equation over a differential energy, $\mathrm{d}E = \mathrm{d}(pc)$, so we have:

$$n_e = \frac{8\pi}{(2\pi\hbar c)^3} \int_0^{150} \frac{x^2}{e^{\beta x} + 1} \, \mathrm{d}x \tag{4}$$

To script for the Gauss-Laguerre model, I had to make use of the int command, which was new to me.

Below we tabulate the integrand itself, without the numerical cofactor. Beyond $n = 4$, increasing then number of nodes does not affect the result. We are likely dominated by roundoff error. Our Github py file is for $n = 2$ nodes.

Nodes	Integrand
n=1	14101.1854368
n=2	14100.6431677
n=4	14100.6539862
n=500	14100.6539862

The exact integral, using Wolfram Alpha, returns 14100.7 for the integrand without cofactor.

Nodes	Integrand Error
n=1	0.485436838491
n=2	-0.0568322618419
n=4	-0.0460138192102

We see convergence that at larger n is dominated by roundoff error.

Including the cofactor, we have $n_e = 1.85943058591 \times 10^{35}$ electrons per cm^3, a few percent off from our result in 3a.