Assignment 02: On Iteration, Flow control and Lists.

PH1050 Computational Physics

Pranav S Ramanujam
EP23B038

1st year EP
Department of Physics, IIT Madras

Problem Statement

Part 1: To write a code to get the value of Machine Epsilon using flow of control functions such as "While", "For" and Do loops.

Part 2:Consider the Gaussian function modulated by x^2.

$$u = x^2 \frac{1}{\sigma \sqrt{2 \text{Pi}}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Taking "μ = 0", we need to find the area below the curve for various values of "σ" ranging from 10^-3 to 10^3. Essentially we need to Integrate "u" numerically using the function "NIntegrate" over suitable limits.

Aim

Part 1:To be able to discover the value of Machine Epsilon using control flow statements

Part 2:To find the area below the curve taking μ=0 and varying the value of σ from 10^-3 to 10^3 in neat steps.

Introduction

Part 1:

In this part to calculate MachineEpsilon, we define a While loop such that the condition in it is always "True" but we impose a second condition which is for the control to "break" or exit the loop when this condition holds true:
(1.0+(2.0)^n)-1.0==0 where (2.0)^(n-1) is the Machine Epsilon.
We could also do this by breaking when (1.0+(2.0)^n)==1.0, but in this case we will get Machine Epsilon of the order of 10^(-14) which is incorrect. We are getting such a value as the Machine Precision insufficient to detect values as small as 2^(-52) which is the value of MachineEpsilon, hence for this to work , we will have to explicitly increase the Precision to some value more than 16 (As MachineEpsilon is of the order of 10^(-16) which to be displayed as a non zero number requires atleast a precision of 16).

Part 2:

This part dealt with the Gaussian function modulated by x^2:

$$u = x^2 \frac{1}{\sigma \sqrt{2 \text{Pi}}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Taking “$\mu = 0$”, we needed to find the area below the curve for various values of “σ” ranging from 10^-3 to 10^3. We had to Integrate “u” numerically using the function “NIntegrate” over suitable limits.

Code Organization

Part 1:

1)Declare a While loop which uses a variable ‘n’=1 and decrements its value by 1 each time the loop is executed. The condition of the While loop is “True”.
2) The control exits/breaks out of the loop when
(1.0+(2.0)^n)-1.0==0
3) In case one uses the condition
(1.0+(2.0)^n)==1.0
then one must also SetPrecision to a value greater than 16 as only then Mathematica will not approximate the value of Machine Epsilon to zero as that number is of the order of 10^(-16).
4)2^(n-1) is the value of MachineEpsilon.

Part 2:

1)First we had to Plot the function u taking the values of σ and μ as 100 and 0 respectively. We do this using the ListLinePlot Function and the Replace All Function (To set the values of σ and μ). This is to get a feel of the nature of the Function.

2)Then we had to generate two lists of data for σ in the range 10^-3 to 10^3. The first list was evenly spaced in a decade: {0.001, 0.002,..., 0.009, 0.01, 0.02,...}.This had have 55 data points. The second list was supposed be logarithmically linear with about 120 points.
(This procedure helps to split the large range (0.001 to 1000) into small neat steps.)

3) Integrate “u” with “$\mu = 0$” for the values of “σ” given by list one over a suitable interval. Plot the integral value as a function of “σ”.

4) Repeat the same with the second list of values for “σ”.

5) See if we can find a relationship between the the integral value (area) and “σ”.

6) If we had time we could check our result by using the function “Plot”. Note “Plot” function has no option for specifying the steps.

Code for computation

Part 1: Discovering The Value Of Machine Epsilon

In[ ]:=
```
n=0;
While[True, If[(1+(2.0)^n)==1, Break[]]; n--]
(2.0)^n (*This is the value of Machine Epsilon, as the control exits the while loop only when 1+(2.0)^n is equal to 1, or when (2.0)^n becomes smaller than the smallest number that the computer can store*)
```

Out[ ]= 1.42109×10^{-14}

```
(*But this gives the wrong answer as the default Machine Precision approximates (2.0)^n as 0 when it reaches values of the order of 10^(-14)*)
```

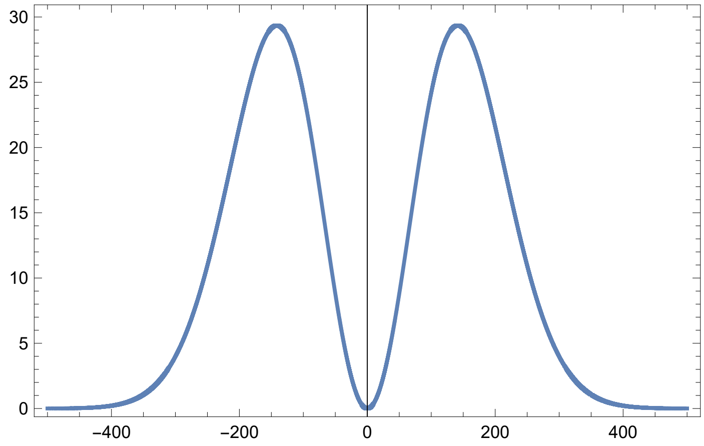
Part 2: Integration

In[ ]:=

```
f[x_]:=x^2*(1/(σ √2 Pi))*e^(-1/2 ((x-μ)/σ)^2) (*Definition of f[x]*)

(*Putting sigma=100 and mew=0*)
f[x_]:=x^2*(1/(100 √2 Pi))*e^(-1/2 (x/100)^2) (*It would have been better to use the /.(replace all)
Function*)
Lst1=Table[{x,f[x]},{x,-500,500,0.01}]; (*Plotting the function f[x] with certain fixed value
ListLinePlot[Lst1,Frame→True]
(*This is the first part to get the idea of the function*)
```

Out[ ]=



In[ ]:=
```
(*Generating lists of data for sigma*)

(*The first list*)
sigLst1={};
Do[AppendTo[sigLst1,n],{n,0.001,0.009,0.001}]
Do[AppendTo[sigLst1,n],{n,0.01,0.09,0.01}]
Do[AppendTo[sigLst1,n],{n,0.1,0.9,0.1}]
Do[AppendTo[sigLst1,n],{n,1,9,1}]
Do[AppendTo[sigLst1,n],{n,10,90,10}]
Do[AppendTo[sigLst1,n],{n,100,900,100}]
sigLst1

(*The second List*)
sigLst2={};
Do[AppendTo[sigLst2,10^(m)],{m,-3,3,0.05}];
sigLst2
```

Out[ ]=
```
{0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04,
 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5,
 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900}
```

Out[ ]=
```
{0.001, 0.00112202, 0.00125893, 0.00141254, 0.00158489, 0.00177828, 0.00199526,
 0.00223872, 0.00251189, 0.00281838, 0.00316228, 0.00354813, 0.00398107, 0.00446684,
 0.00501187, 0.00562341, 0.00630957, 0.00707946, 0.00794328, 0.00891251, 0.01,
 0.0112202, 0.0125893, 0.0141254, 0.0158489, 0.0177828, 0.0199526, 0.0223872,
 0.0251189, 0.0281838, 0.0316228, 0.0354813, 0.0398107, 0.0446684, 0.0501187,
 0.0562341, 0.0630957, 0.0707946, 0.0794328, 0.0891251, 0.1, 0.112202, 0.125893,
 0.141254, 0.158489, 0.177828, 0.199526, 0.223872, 0.251189, 0.281838, 0.316228,
 0.354813, 0.398107, 0.446684, 0.501187, 0.562341, 0.630957, 0.707946, 0.794328,
 0.891251, 1., 1.12202, 1.25893, 1.41254, 1.58489, 1.77828, 1.99526, 2.23872,
 2.51189, 2.81838, 3.16228, 3.54813, 3.98107, 4.46684, 5.01187, 5.62341,
 6.30957, 7.07946, 7.94328, 8.91251, 10., 11.2202, 12.5893, 14.1254, 15.8489,
 17.7828, 19.9526, 22.3872, 25.1189, 28.1838, 31.6228, 35.4813, 39.8107, 44.6684,
 50.1187, 56.2341, 63.0957, 70.7946, 79.4328, 89.1251, 100., 112.202, 125.893,
 141.254, 158.489, 177.828, 199.526, 223.872, 251.189, 281.838, 316.228, 354.813,
 398.107, 446.684, 501.187, 562.341, 630.957, 707.946, 794.328, 891.251, 1000.}
```

$$f[x\_] := x^2 * \left(1 \middle/ \left(\sigma \sqrt{2\,Pi}\right)\right) * e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2};$$

```
IntegLst = {};

For[l = 1, l < 55 , l++, b = NIntegrate[f[x], {x, -1000, 1000}]; AppendTo[IntegLst, b]];
IntegLst
(*Tried to store the values obtained by integrating the function over the limits -
 1000 to 1000 for various values of σ *)
```

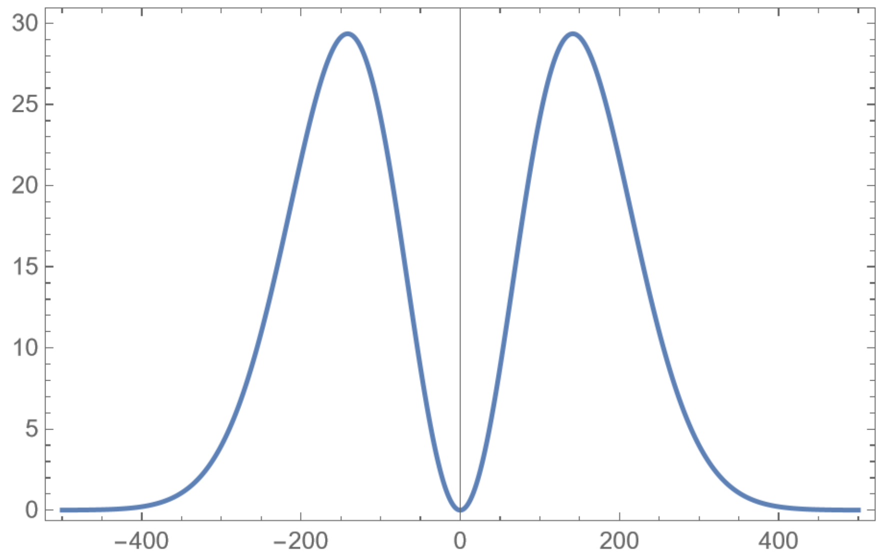
Results

Part 1:

The value of Machine Epsilon as computed by my code: 1.42109×10^{-14} (Wrong due to insufficient precision)

Part 2:

Out[ ]=



The plot of u i got putting σ=100 and μ=0 to get a rough idea of the nature of the Function.

{0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900}
This was the first of the 2 Tables containing various values of σ (54 data points)

{0.001, 0.00112202, 0.00125893, 0.00141254, 0.00158489, 0.00177828, 0.00199526, 0.00223872, 0.00251189, 0.00281838, 0.00316228, 0.00354813, 0.00398107, 0.00446684, 0.00501187, 0.00562341, 0.00630957, 0.00707946, 0.00794328, 0.00891251, 0.01, 0.0112202, 0.0125893, 0.0141254, 0.0158489, 0.0177828, 0.0199526, 0.0223872, 0.0251189, 0.0281838, 0.0316228, 0.0354813, 0.0398107, 0.0446684, 0.0501187, 0.0562341, 0.0630957, 0.0707946, 0.0794328, 0.0891251, 0.1, 0.112202, 0.125893, 0.141254, 0.158489, 0.177828, 0.199526, 0.223872, 0.251189, 0.281838, 0.316228, 0.354813, 0.398107, 0.446684, 0.501187, 0.562341, 0.630957, 0.707946, 0.794328, 0.891251, 1., 1.12202, 1.25893, 1.41254, 1.58489, 1.77828, 1.99526, 2.23872, 2.51189, 2.81838, 3.16228, 3.54813, 3.98107, 4.46684, 5.01187, 5.62341, 6.30957, 7.07946, 7.94328, 8.91251, 10., 11.2202, 12.5893, 14.1254, 15.8489, 17.7828, 19.9526, 22.3872, 25.1189, 28.1838, 31.6228, 35.4813, 39.8107, 44.6684, 50.1187, 56.2341, 63.0957, 70.7946, 79.4328, 89.1251, 100., 112.202, 125.893, 141.254, 158.489, 177.828, 199.526, 223.872, 251.189, 281.838, 316.228, 354.813, 398.107, 446.684, 501.187, 562.341, 630.957, 707.946, 794.328, 891.251, 1000.}
This was the second of the Two tables containing 120 data points.

Comments

Part 1:

The precision part was the tricky part.

Part 2:

This part was hard and involved many Functions used together.