SESG6025: Lab 5 - Integration

Aims: This lab introduces some further features of Matlab/ Python and discusses Monte Carlo integration.

Objectives: Perform some numerical integration using Matlab/ Python

1) Before attempting the questions, read the Matlab/ Python `help` on:
`exp, trapz, quad, function, for, rand, mean, std, abs`

2) Plot the function e^x using 10 equally spaced points between 0 and 4. Show circles for the points and join up the points with a straight line.

3) Using the trapezium rule find the integral of the function e^x using 10 points over the range 0 to 4.

4) The exact answer is given by:

$$\int_0^4 e^x = e^x\Big|_0^4 = e^4 - e^0 = \texttt{exp(4) - exp(0)}$$

Evaluate the integral as in (2) using N = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, and 200 equally spaced points over the range and plot the absolute percentage value of the error = $\frac{\left|I_{exact} - I_{trapezium}\right|}{I_{exact}} \times 100\%$ as a function of the number of points used, N.

5) Evaluate the integral in (4) using the '`quad`' function and a tolerance of `1e-3`.

6) Repeat (4) using the function '`humps`' and the limits 0 to 4. The exact answer is given by

```
I_exact = 10*atan(37)+5*atan(31/2)-24+10*atan(3)+5*atan(9/2);
```

(If you want to show this, then use '`type`' to find out what function '`humps`' represents, and the look at the help for '`int`' to perform the symbolic integration)

7) Evaluate the integral in (6) using the '`quad`' function and a tolerance of `1e-3`.

8) Monte Carlo integration. It is possible to estimate an integral using the following formula:

$$I = \int_{x=a}^{b} f(x)\,\mathrm{d}x \approx (b-a)\langle f(x_i)\rangle \pm \frac{(b-a)}{\sqrt{N}}\sqrt{\left\langle f(x_i)^2\right\rangle - \langle f(x_i)\rangle},$$

where the x_i are chosen randomly between a and b, N is the number of sample points, and:

$$\langle f(x_i)\rangle = \texttt{mean(f)} \text{ and } \sqrt{\left\langle f(x_i)^2\right\rangle - \langle f(x_i)\rangle} = \texttt{std(f)}.$$

Using for x_i 10000 random numbers in the range 0 to 4, estimate the integral calculated in (4) and give the one standard deviation error bound for it (this is just the the ± part).

(Hint: To scale random numbers in the range 0 to 1 to be in the range a to b, you should use `(b - a)*rand(N,1) + a`, where $b > a$ and N is the number of points to generate.)

Why might you ever want to use this method? For multi-dimensional integrals in d dimensions, the error for the trapezium rule falls off as $N^{(-2/d)}$, where N is the number of points used/ function evaluations made. However, the $N^{(-1/2)}$ dependence of the error for the Monte Carlo integration is independent of the number of dimensions; hence for large d, the Monte Carlo method is actually more accurate for a given number of function evaluations. For Simpson's rule the error goes as $N^{(-4/d)}$ in d dimensions. By using quasi-random sequences it is possible to arrange for the error to drop of as $N^{(-1)}$ independent of d.

Please ask if you need help.

Prof Simon Cox, sjc@soton.ac.uk. Building 25/2037 Phone Ext 23116.