School of Mathematics and Statistics
MAST30028 Numerical Methods & Scientific Computing
Week 3

Drag and drop the folder `Week3` from `L: \MAST30028` to `C:\...\MATLAB` and include it in the path. Now MATLAB knows how to find the files in `Week3`.

1 Simple plotting

This relates to material presented in Lecture 4 (Plotting in MATLAB).

Exercise Set 1

Write MATLAB commands to create the following kinds of plots:

a. Plot $\sin\theta$ versus θ for 60 points in the interval $0 \le \theta \le 2\pi$. Connect the points with a dashed line *and* label the points with open circles.

b. Create a plot that will reveal whether the data $\{y_n\}$ decay geometrically with n e.g. $y_n = 4.3(0.5)^n$

c. Create a plot that will reveal whether the data $\{y_n\}$ decay algebraically with n e.g. $y_n = 7.2n^{-2}$

d. Plot $y = x^2; y = \sin(x^2)$ and $y = \sin(x)$ on the same graph. Label them with arrows pointing to the respective curve (for example, have an arrow from the label '$y = x^2$' to the curve $y = x^2$). I suggest you do this interactively with Plot Tools.

e. Use a parametric plot to plot the ellipse

$$\frac{x^2}{2^2} + \frac{y^2}{3^2} = 1$$

2 Stochastic simulation

This relates to material presented in Lecture 5 (Probability Revision and random numbers).

Pseudorandom numbers

Exercise Set 2

a. Write a MATLAB command that generates 20 random integers between 4 and 12.

b. Write a MATLAB command that generates 30 random numbers uniformly distributed between -2 and 6.

c. Write a MATLAB command that generates 40 random numbers drawn from a normal distribution with mean 2 and standard deviation 3.

d. Write a MATLAB command that generates 50 random numbers drawn from a normal distribution with mean 3 and variance 4.

e. run the script `demorand_revised` and understand the results. Repeat several times.

f. modify `demorand_revised` to explore simulating random integers between 1 and 10.

Stochastic simulations

Recall the basic structure of a simulation program:

```
set the number of repetitions
set any inputs for the random experiment
for each repetition
     run the random experiment
     collect quantities of interest
end
compute summary data
```

Exercise Set 3

a. Modify the code given in `deMere1.m` to simulate de Méré's modified bet i.e. determine the probability of throwing at least 1 double six in 24 throws of 2 fair dice. This is your first *simulation program.*

   De Méré thought this probability must be larger than 0.5 but found by bitter experience that 25 throws were required. Using probability laws, find the exact probability.

   Your program should print the difference between the simulation result and the exact answer.

   STOP AND THINK BEFORE SOLVING THIS PROBLEM. THERE ARE MANY WAYS TO SOLVE THIS PROBLEM BUT ONE WAY IS SMARTER AND EASIER THAN THE REST!

b. Write a driver function that runs your simulation many times and outputs the different results. Experiment with various numbers of repetitions.

c. Kevin plays roulette and bets \$1 on 'red' on each spin of the wheel. As the wheel has a '0', Kevin's chance of winning \$1 is $p = 18/37$ ($q = 1 - p$ is the chance of losing \$1). We assume that together the casino and Kevin have 'T' dollars in total. Kevin starts with 'k' dollars — the casino with the remainder. The game ends when either Kevin or the casino runs out of money.

   (i) Let $q_k = Pr$(Kevin is ruined|Kevin starts with k dollars). It can be shown that

   $$q_k = \frac{(q/p)^T - (q/p)^k}{(q/p)^T - 1}$$

   Write a MATLAB function which simulates N games of roulette following the rules above, with k (Kevin's starting capital), $T - k$ (the casino's starting capital) and N as input arguments. The program function should calculate an estimate for q_k (the probability of ruin for the gambler) and calculate the mean number of bets in a game.

   (ii) Check your program by calculating q_k using the formula in part (i) for two different examples and compare with your program estimates. Roughly how many repetitions do you need to be within 1% of the correct answer?

   (iii) If (as is usually the case) the casino has a lot more capital than Kevin, the mean duration of the game can be shown to be approximately $\frac{k}{q-p}$. Use this result to check the mean duration calculated by your program.