- You are asked to solve the following problems using Matlab.
- The tasks are inter-related so you should perform them in sequence.
- At the beginning of each session working on your project, start a `diary` file whose name is (for example) `Nov10.txt` where `Nov10` should be replaced by the date of the session.
- You should download the function m-file `bjk.m` from

  `http://jkcray.maths.ul.ie/ms4101/m-Files/bjk.m`.

1. Perform the following plotting and scripting tasks. (7%)

   (a) Write a **script** m-file `Run.m` that performs the following tasks.

   i. Plot the function `bjk` on the interval $[0, 30]$ with a suitable number of intermediate points.

   ii. Use your student ID as the second input parameter (a **vector** of 8 integers).

   iii. Include the x–axis by plotting a horizontal line from $(0, 0)$ to $(30, 0)$.

   iv. Annotate the plot with the title: **Plot of a combination of** $J_n(x)$ **on the interval** $[\mathbf{0}, \mathbf{30}]$.

   v. Use LaTeX formatting commands to typeset the parts of the title using mathematical notation.

   vi. Label the x–axis and y–axis suitably.

   (b) Based on your plot, add a line in your **script** m-file `Run.m` that creates a vector `myroots` which contains the approximate location of the root or roots of `bjk` in the interval $[0, 30]$.

   (c) Add a further line in `Run.m` that creates and displays a string variable `rootstr` which states the approximate location of the roots. (Use the `num2str` function to include the vector `myroots`.)

2. Perform the following Matlab programming tasks. (18%)

The Secant Method computes a succession of increasingly accurate estimates of a root of a specified function. The current approximation x_n is based on the two previous values x_{n-1} and x_{n-2} together with the function values $f(x_{n-1})$ and $f(x_{n-2})$ at these x–values.

The update rule is:

$$\begin{aligned} x_n &= x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}, \quad \text{or, equivalently:} \\ &\equiv \frac{x_{n-2}f(x_{n-1}) - x_{n-1}f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})} \end{aligned}$$

You can write the latter more simply as:

x3=(f2*x1-f1*x2)/(f2-f1);

(a) Write a **function** m-file `Secant.m` that uses this update rule to approximate a root of a specified function.

(b) `Secant.m` should have input parameters:

- `fun` (the name of the function m-file containing the function whose root is to be found).
- `x1` and `x2` (the two "bracketing" starting points).
- `acc` (the required accuracy).
- `ID`, your student ID.

(c) `Secant.m` should have output parameter `r`, the root found.

(d) `Secant.m` should use a `while` loop to apply the update rule above until the difference between successive x-values is less than the value of the input parameter `acc`.

(e) Amend your stop criterion so that the `while` loop also checks for `abs( fun(x)) < acc` and stops if this occurs.

(f) Now find values for each of the roots of **bjk** on the interval $[0, 30]$.

- You will add lines to your **script** m-file `Run.m` that call the **function** m-file `Secant.m`.
- You will:
  i. Use the value `@bjk` for the input parameter `fun`.
  ii. Choose values for `x1` and `x2` based on your plot of `bjk`.
  iii. Take the value `1.0e-8` for the input parameter `acc`.

3. Use LaTeXto write a short report (1–2 pages) in `pdf` format that: (5%)

   (a) includes a plot of your version of `bjk(x)` on the interval $[\mathbf{0}, \mathbf{30}]$,

   (b) includes your Matlab code (`Run.m` and `Secant.m`) using a **verbatim** or **lstlisting** environment,

   (c) explains your code,,

   (d) reports the values of the roots found using the Secant Method.

You should:

- Create a `zip` file named `123456789.zip` (where `123456789` should be replaced by your student ID) containing all the files in your current working folder, in particular your **pdf** file and your Matlab m–files and **diary** files.
- Upload your zip file when asked to do so at the end of the project period.