MATH20411   Bessel functions of order zero

Consider the second-order ODE

$$x^2 \frac{d^2 X}{dx^2} + x \frac{dX}{dx} + \omega^2 x^2 X = 0 \tag{1}$$

for $X(x)$. Equation (1) is called Bessel's equation of order zero and the general solution is

$$X(x) = c_1 J_0(\omega x) + c_2 Y_0(\omega x) \tag{2}$$

where c_1, c_2 are arbitrary constants, and J_0 and Y_0 are the so-called **Bessel functions of order zero**, of the first and second kinds, respectively. (You may not have met these functions before but you can find them in many textbooks and MATLAB has routines called `besselj` and `bessely` to evaluate them.)

We encounter equation (1) and Bessel functions, when solving the wave equation on a disk. When applying separation of variables to solve PDEs in two variables, be aware that the ODEs obtained may not be simple, and may not have standard solutions like $\sin(x)$ and $\cos(x)$ or e^x and e^{-x}. We will discuss separation of variables for the wave equation on a disk in lectures. You can spend some time familiarising yourself with Bessel functions by doing the following simple computations in MATLAB. Remember that you can ask MATLAB about one of its in-built functions, and find examples of how to call it, by typing `help` followed by the name of the function. For example,

```
>> help besselj
```

Exercise 1 Plot the Bessel functions $J_0(x)$ and $Y_0(x)$ in MATLAB on the interval $[0, 100]$. Hint: use the MATLAB functions `besselj` and `bessely`.

Solution The MATLAB commands to plot the graphs of J_0 and Y_0 on $[0, 100]$ are

```
x=linspace(0,100,1000);  % sets up some values of x in the interval [0,100]
y1=besselj(0,x);         % evaluates the function besselj of order 0, at x values
y2=bessely(0,x);         % evaluates the function bessely of order 0, at x values
plot(x,y1); plot(x,y2);  % plot results
```

See Fig 1. below. Observe that both $J_0(x)$ and $Y_0(x)$ have many roots in the interval $[0, 100]$. If we extend the interval, we find more roots, and in fact J_0 and Y_0 have infinitely many roots (like sin and cos). Note that the graph should be smooth - not pointy. If you use too few points for plotting, you will miss important features - such as what happens to $Y_0(x)$ near $x = 0$, and possibly even miss some of the roots.

Exercise 2 Find (any way you like) the first 5 positive roots $\alpha_n, n = 1, 2, \ldots 5$, of J_0. Hint: you may find the MATLAB function `fzero` helpful.

Solution The MATLAB function `fzero` will find the roots of a function (that you define - or one of its own in-built functions), close to an initial guess that you provide. From the graph of J_0 you should be able to provide good initial guesses for the first five roots. If you zoom in on your graph of J_0, you should see roots close to $x = 2, 6, 8, 11, 14$. Calling `fzero` with the initial guess $x = 2$ returns the first root

```
X = fzero(@(x) besselj(0,x),2)
 X =
    2.4048
```
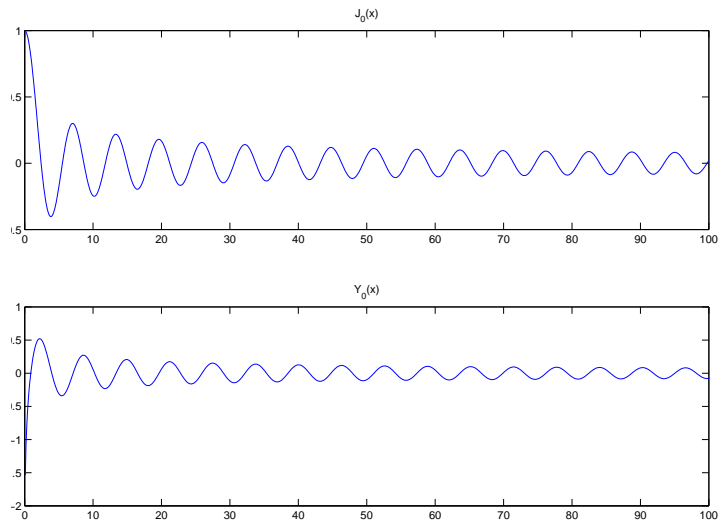
Figure 1: Plot of $J_0(x)$ (top) and $Y_0(x)$ (bottom) on [0, 100]

(to four d.p). Similarly, we find the next roots (to 4 d.p.) are:

$$\alpha_1 = 2.4048, \alpha_2 = 5.5201, \alpha_3 = 8.6537, \alpha_4 = 11.7915, \alpha_5 = 14.9309.$$

You can find the correct calling sequence for `fzero` by typing `help fzero` or you can even get quite a good approximation by zooming in on the graph.
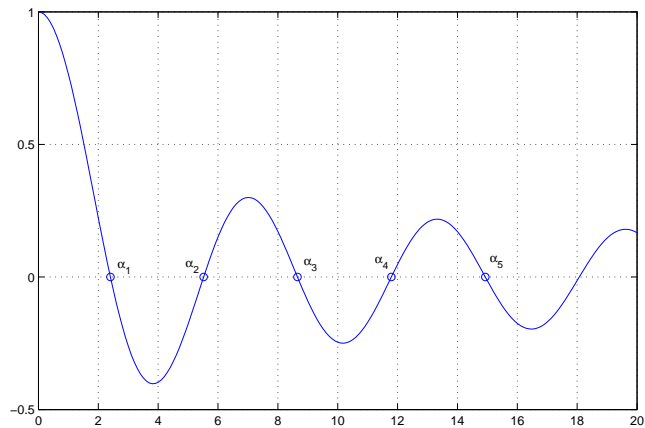

Figure 2: J_0 on [0,20] and first five positive roots α_n marked with circles