Introduction to Scientific Computing Lab 3

Functions

2017

1 Objectives

After completing these exercises you will be able to:

- Break up your problem to make your code simpler
- Use functions where appropriate

2 Notes

Work individually

Create a directory for each week so you can come back to your codes in the future. Create files for each of the different exercises and name them in a logical manner, so `exercise1.c` for example.

When changing the source code, ensure you have **saved it before compiling** otherwise the compiler will only see the old file

When you have completed all exercises, ask a demonstrator to assess your work. They will test your code and ensure it is formatted well with good commenting, structure and variable names. This is a useful feedback mechanism, so listen to what the demonstrator has to say and their recommendations for improving your code.

In case of an error, read the compiler error. This will often tell you the line (or close to the line) where the error is occurring. Fix it, test and repeat for the errors you have. If you are getting nowhere then it can often be useful to copy the error into google, or use some keyword searches. If you are really stuck on one error then call over an assistant who will be able to point you in the right direction.

3 Deflection of a Cantilevered Beam Under Uniform Loading Conditions

In this exercise, we will consider developing a code to find the weight of fuel that can be added to a wing before the wing-tip hits the ground. To do this, we will first need to construct functions to calculate the deflection of a wing under a uniform loading. A wing can be approximated as a cantilevered beam of length s under spanwise uniform loading of q N/m. The beam can be modelled by a rectangular cross section of width c (the aerofoil chord) and height t (the aerofoil thickness).

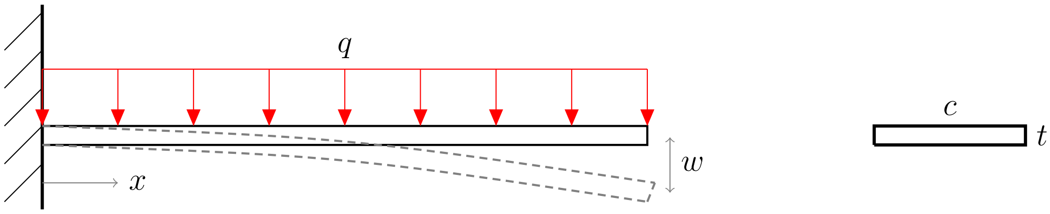

Figure 1: Cantilever beam under uniform loading

From beam bending theory, the deflection at the tip of the beam is given by:

$$w = \frac{qs^4}{8EI} \tag{1}$$

where s is the semi-span of the wings, E is the Young's modulus, I is the second moment of area and q is made up of the weight of the wing, W_{wing}, and the weight of the fuel, W_{fuel}:

$$q = \frac{W_{wing} + W_{fuel}}{s} \tag{2}$$

This problem can clearly be solved analytically, but we will solve using a bisection approach.

4 Exercises

4.1 Exercise 1

Create a `main` program that stores variables that represent the physical quantities given in table 1 and hard-code values as shown in table 1.

We now need to calculate some quantities that will be used to obtain the deflection of the beam. You will need to add a number of functions, which are detailed below. **You should name your functions sensibly**.

Table 1: Variables to include

Variable	Nomenclature	Value
Thickness	t	0.07m
Chord	c	0.6m
Semi-span	s	13.5m
Young's Modulus	E	69×10^{9}Pa
Wing weight	W_{wing}	1150N
Wing height	h	0.5m

Function 1

Add a function to your program that has input arguments c and t and outputs the second moment of area, I, of a rectangular beam (equation 3). Ensure you test that the output value of I is $1.72 \times 10^{-5}\text{m}^4$.

$$I = \frac{ct^3}{12} \tag{3}$$

Function 2

Add a function to your program that has input arguments W_{wing}, W_{fuel} and s and outputs the loading per unit span, q, using equation 2. Ensure you test that the output value of q is 85.2N/m for zero fuel.

Function 3

Add a function to your program that has input arguments W_{wing}, W_{fuel}, s, E, c and t and outputs the tip deflection of the wing (equation 1). You will need to call functions 1 and 2 within this function. Ensure you test that the output value of w is 0.299m for zero fuel.

4.2 Exercise 2

We now need to use bisection to find W_{fuel} such that the tip deflection is not greater than the height of the wing-tip. Hence we want to find W_{fuel} such that $w(W_{fuel}) = h$[‡]. Bisection is similar to Newton's method in that it is a method of finding the inverse of a function without algebra (or in the case where we can't use algebra), but unlike Newton's method, we don't need the gradient.

Assuming we have an increasing function (i.e. if $A > B$ then $w(A) > w(B)$) and that the solution lies within an upper and lower bound U, L, bisection proceeds as follows:

1. Generate new test point $W_{test} = \frac{1}{2}(U + L)$
2. Evaluate $w(W_{test})$

[‡]This notation simply says that the deflection is a function of the fuel weight and for some value of fuel weight, this equals the tip height.

   (a) If $w > (h + \epsilon)$ then replace U with W_{test}
   (b) If $w < (h - \epsilon)$ then replace L with W_{test}
   (c) Else (i.e. $h - \epsilon \leq w \leq h + \epsilon$) then $W_{fuel} = W_{test}$

3. Repeat

After a while, the solution gets to within some small precision, ϵ, of the required solution and the algorithm can exit.

Use initial values of $L = 0N$ and $U = 10,000N$ to calculate W_{fuel} to within a precision of $\epsilon = 1 \times 10^{-4}$. Output the final answer to the user.

An example of the iteration history is given below:

```
   Lower      Upper      Wfuel       Def.
 -----------------------------------------
    0.00   10000.00    5000.00     1.5984
    0.00    5000.00    2500.00     0.9486
    0.00    2500.00    1250.00     0.6237
    0.00    1250.00     625.00     0.4613
  625.00    1250.00     937.50     0.5425
  625.00     937.50     781.25     0.5019
  625.00     781.25     703.12     0.4816
  703.12     781.25     742.19     0.4918
  742.19     781.25     761.72     0.4968
  761.72     781.25     771.48     0.4994
  771.48     781.25     776.37     0.5007
  771.48     776.37     773.93     0.5000

Weight of fuel is     773.93N with 0.5000m deflection
```