Root finding

$\star$ questions must be submitted to WebCT, as an extensively commented program and full report by Wednesday 31/10/12 4pm (usual late penalties will apply).

This worksheet aims to show you the standard methods for numerically finding the roots of an equation $f(x)$. That is, the values of x for which $f(x) = 0$. You will also need to use error analysis to understand the performance of each method.

It is often helpful when looking for the roots of an equation to have an idea of what the function looks like e.g. does it have multiple roots close by? How many roots are we looking for? To understand some of the issues involved in root finding we will stick to 1 dimensional problems. Higher dimensional problems can be solved in similar ways but come with their own set of issues, the difficulty of being able to visualise them being one of these.

Bisection method

The easiest (but relatively inefficient) way to tackle the problem is the *bisection method*. We want to find the root(s) of $f(x) = 0$. To start we 'bracket the root' by finding (either by guesswork or brute force) two values of x ($x_b < x_t$) such that $f(x)$ is positive for one of the values and negative for the other. As $f(x)$ must pass through 0 to go from positive to negative there must be at least one root in the interval between these two points. This constraint on x_b and x_t is equivalent to $f(x_b)f(x_t) < 0$.

We proceed by selecting the centre of the interval $x_m = \frac{(x_b+x_t)}{2}$ and test to see if

$$f(x_t)f(x_m) < 0$$

If this is true then x_m is a suitable lower bound (which is higher than x_b and so is more constraining) so we set $x_b = x_m$, else x_m is a suitable upper bound and so we set $x_t = x_m$. We continue repeating this until the interval containing the root reaches a certain size (i.e. if we want an accuracy of 3 decimal places then $x_t - x_b < 10^{-3}$).

Newton-Raphson Method

Another common method of root finding known as the Newton-Raphson method is more efficient under a wide range of circumstances. Starting with an initial guess for the root the next guess is given by,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Figure 1 shows what the algorithm does for 2 iterations. With an initial guess x_0 the tangent of the function at that point is found (red line) and followed down to the x-axis to give the next guess x_1. This is done again (blue line) to find the following guess x_2 and so on, to get better and better guesses. To see this is true you can just write down the function $y(x)$ of a straight

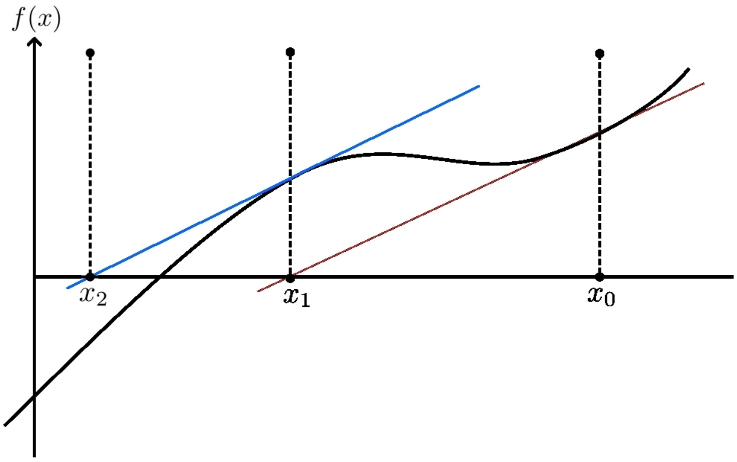
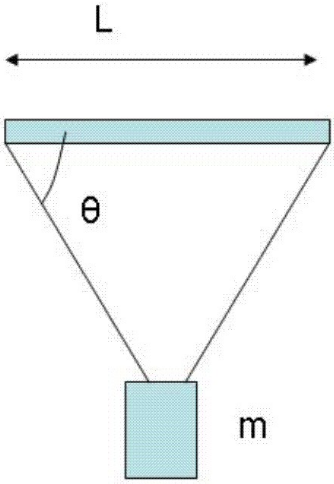


Figure 1: given an initial guess, x_0, the above diagram shows how the next approximations of the root are found by the Newton-Raphson method.

line that goes through the point $(x_n, f(x_n))$ and is tangent to $f(x)$ at that point. Then x_{n+1} is simply the value of x when $y(x) = 0$.

With a reasonable first guess and a suitable function the series of guesses will converge to the root. What we mean by '*reasonable*' and '*suitable*' for this method will be expanded on in question 2.

1. $\star$ Our problem is a classical mechanics one, which is easy to set up but cannot be solved exactly analytically. Consider a mass m suspended from a bench of length L by two equivalent springs, of unstretched length $L/2$ and of spring constant k



Show, by balancing forces, that the equation you need to solve is

$$kL(\tan\theta - \sin\theta) = mg$$

Assuming $m = 5$kg and $L = 0.6m$, $k = 1000$N/m find θ
using the bisection method outlined above. Tabulated output data should include: value of theta found, the size of the uncertainty, number of times you needed to apply the bisection method.

2. ⋆ In this question we will explore the performance of various methods to understand why you would use different algorithms depending on your problem.

   i) Consider the bisection method, we can analytically estimate the uncertainty in the value of the root after 1 iteration if we know the error of the result from the previous iteration. Derive the relation between the error at a step $n+1$, ϵ_{n+1}, in terms of the error at step n, ϵ_n.

   ii) By taylor expanding the function $f(x)$ around x_n and considering $f(x_{root})$ show the relation between ϵ_n and ϵ_{n+1} for the newton-raphson method. Discuss the conditions needed for the newton raphson method to converge.

   iii) Find the solution to

   $$x^3 + 8x^2 - 5x + 22 = 0$$

   both by using the bisection method and writing a program using GSL that uses the Newton-Raphson method. For the Newton-Raphson method try a few different initial guesses and choose 2 starting points that show different initial behaviours of the algorithm. Plot a graph that compares the evolution of the errors for these 3 runs and discuss the result, keeping in mind the analysis of the first parts of this question. Make sure you have found the solution to a high enough accuracy that this plot is meaningful.

3. Find all the roots of the following function

$$\frac{11}{4}\cos(x) - \ln(x) = 0$$

Demonstrate graphically that you have found all of the roots. NB Check that you are not asking the computer to do integer arithmetic!

4. Consider the standard Gaussian integral:

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-x^2/2}dx = 1$$

Now using a combination of a numerical integration routine, and a root finding method write a program which can find the value of $a > 0$ such that

$$\frac{1}{\sqrt{2\pi}}\int_{-a}^{a} e^{-x^2/2}dx = p$$

where p is a user input and is in the range $0 < p \leq 1$. The program should output the accuracy with which a has been evaluated.