COMP116 – Work Sheet Three

Associated Module Learning Outcomes

1. basic understanding of the range of techniques used to analyse and reason about computational settings.
2. Ability to apply basic rules to differentiate commonly arising functions.

Question 1: Derivatives and Critical Points

For each of the following Real valued functions $f : \mathbb{R} \to \mathbb{R}$ describe:

a. Its **first** derivative with respect to x, i.e. $f'(x)$.

b. Whether after some **fixed** number, k say, repeatedly finding the derivative produces no change. e.g. if $f(x) = 3$, then $f'(x) = 0$ and $f''(x) = 0$, so in this case $k = 2$.

c. Comment on the **critical** points of $f(x)$. Note you are **not** asked to compute these only to give an informal justification as to whether these exist, are minima or maxima, indeterminate, etc.

1. $f(x) = x^2 + 2x - 8$.
2. $f(x) = x^3/3 \ + \ x \ + \ 10$.
3. $f(x) = \frac{x^2-5x+6}{x-2}$.
4. $f(x) = \sqrt{x}$.
5. $f(x) = 1/x^2$.
6. $f(x) = \sin^2(x) + \cos^2(x) + x^2$.
7. $f(x) = \sin(x^2) + \cos(x^2) + x^2$.
8. $f(x) = \exp(x \log x)$ (recall that $\log$ is Natural, i.e. base e unless explicitly stated to be otherwise).

Question 2 – Roots Revisited

A very basic method for discovering a root of a polynomial function was discussed in the first worksheet. An informal realization of the approach outlined would be

Algorithm 1 Simple root-finding method for polynomials with coefficients in $\mathbb{R}$

Input: $p(x) \in \mathbb{R}[X]$; $a, b \in \mathbb{R}$ with $p(a) < 0$ and $p(b) > 0$
Input: $MAX \in \mathbb{N}$; $\varepsilon \in \mathbb{R}$ with $\varepsilon > 0$;
$\{MAX$ is a "cut-off" to prevent indefinite looping; ε a "tolerance" threshold.$\}$
$low := a$; $high := b$; $k := 0$;
repeat
    $x := (low + high)/2$; {Find value in "middle", i.e. average of $\{low, high\}$.}
    if $p(x) < 0$ **then**
        $low := x$;
    else if $p(x) \geq 0$ **then**
        $high := x$;
    end if
    $k := k + 1$;
until $|p(x)| \leq \varepsilon$ or $k > MAX$
return x;

Algorithm 1 uses a simple "binary search" (if you have not met "binary search" already in Year 1, then you will certainly be introduced to it on COMP108: this is one of the most basic and effective data searching techniques). It can (provided suitable initial values a and b are identified) be applied as an approach to (try and) find not only roots of polynomials but also so-called "zeros" of arbitrary Real valued functions, i.e. for $f : \mathbb{R} \to \mathbb{R}$, values c with which $f(c) = 0$.

Following the discovery of Differential Calculus in the 17th Century, rather more sophisticated techniques were developed. One of these (the Newton–Raphson method) is still widely taught. As with the binary search technique in Algorithm 1 this can be adapted to arbitrary functions (subject to some technical conditions). Another very powerful general method is Halley's Method mentioned briefly in the first worksheet.

Halley's Method is presented in Algorithm 2.

One required property of $f(x)$ in order to use Halley's Method to find zeros of f is that not only is $f'(x)$ (the first derivative of f) well-defined but also $f''(x)$ (its second derivative). Depending on how the transitions from x_0 to x_1 through to x_k develop this *might* lead to complications, although it should be noted that it is *not* required that $f'(x)$ and $f''(x)$ are well defined for every $x \in \mathbb{R}$.

Algorithm 2 Halley's Method for finding roots of $f(x)$

Input: $f : \mathbb{R} \to \mathbb{R}$; {e.g. could be name of an implemented function.}
Input: $x_0 \in \mathbb{R}$; {An initial guess for $f(x_0) = 0$.}
Input: $MAX \in \mathbb{N}$; $\varepsilon \in \mathbb{R}$ with $\varepsilon > 0$;
{MAX is a "cut-off" to prevent indefinite looping; ε a "tolerance" threshold.}
$k := 0$;
repeat

$$x_{k+1} \ := \ x_k \ - \ \frac{2f(x_k)f'(x_k)}{2(f'(x_k))^2 - f(x_k)f''(x_k)}$$

    $k := k + 1$;
until $|f(x_k)| \leq \varepsilon$ or $k > MAX$
return x_k;

One significant advantage of Halley's Method for high-degree polynomials is that, unlike a number of methods, it does not require the computation of *square roots*, i.e. functions of the form $\sqrt{f(x)}$. This can be especially useful, in comparison with techniques such as Laguerre's (module textbook, pages 156-7), when dealing with polynomial functions.

a. Suppose $f(x) = \frac{\cos(5x+1)}{x+5}$.

VERY IMPORTANT: Measures are in **radians** and not degrees.

1. Why is it sufficient to consider *only* the behaviour of $\cos(5x+1)$ when seeking zeros of $f(x)$?
2. Using $a = -1$ (so that $\cos(5a + 1) \sim -0.654$) and $b = 1$ (with $\cos(5b+1) \sim 0.96$) what are the first **five** values found for $\cos(5x+1)$ using the method of Algorithm 1?
2. Similarly what are the first five values identified using Halley's method and an initial guess $x_0 = 0$ (that is to say, $(-1+1)/2$) for $\cos(5x+1)$.
3. If x_{binary} is the value discovered by Algorithm 1 and x_{halley} that found with Algorithm 2, which of these describes a "better zero" of $f(x)$, i.e. which of $\{|f(x_{binary})|, |f(x_{halley})|\}$ is smaller (closest to 0)?

b. For the degree 5 polynomial

$$p(x) = x^5 \ + \ 2x^4 \ - \ 11x^3 \ - \ 22x^2 \ + \ 30x \ + \ 60$$

1. Carry out the same comparison (again with 5 iterations) of Algorithm 1 and Algorithm 2, this time for the degree 5 polynomial $p(x)$. Use $a = -2.7$ (with $p(a) \sim -2.068$) and $b = 0$ (so that $p(b) = 60$) in Algorithm 1 and $x_0 = -1.35$ (i.e. $(-2.7 + 0)/2$) in Algorithm 2.
2. Which of the two seems to converge to a root of $p(x)$ more quickly?
3. Do Algorithm 1 and Algorithm 2 seem to converge to the *same* root?

Suggestion: You may it less cumbersome and error-strewn to write a short piece of code that will carry out this process.