Supervision work
Supervision 18 – Numerical Methods

Tudor Avram
Homerton College – tma33@cam.ac.uk

10th of March 2016

1 Exercise 2

(a) Let f be a function, r such as $f(r) = 0$ and x_0 a good estimate of r. $h = x_0 + r$. Thus, we can say that h is "small" and we can use the linear approximation to conclude that :

$$0 = f(r) = f(x_0 + h) \approx f(x_0) + h \times f'(x_0)$$

So, unless $f'(x_0)$ is close to 0,

$$h \approx -\frac{f(x_0)}{f'(x_0)}$$

We know that $r = x_0 + h$. Thus,

$$r \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

Let x_n a series of approximations of r. Therefore, the x_1 (the next approximation of r) has the value :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The next estimate x_2 is obtained from x_1 the same way x_1 was obtained from x_0 :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

By mathematical induction, if x_n is our current estimate, the next one, x_{n+1} is given by :

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

(b) In numerical methods, the order of an algorithm represents the way in which the truncation error is affected by reducing h. For a second-order method, the truncation error is divided by 4.
An iteration that has order of convergence 2 is also called a quadratic convergence.

$$\lim \frac{|x_{n+1}-r|}{|x_n-r|^2} = \alpha,$$

where $\alpha > 0$.

(c) For Newton Raphson, the order of convergence is $q = 2$.

2 Exercise 3

(a) The Taylor series expansion for cosine, ignoring cubic and higher powers is :

$$\boxed{\cos x = 1 - \frac{x^2}{2}}$$

(b) $[-\frac{\pi}{2}, \frac{\pi}{2}]$ (just a guess...)

(c) If we divide the domain in 2 sub-domains, $[-\frac{\pi}{2}, 0]$ and $[0, \frac{\pi}{2}]$, we can study the monotony of the function :
For $[-\frac{\pi}{2}, 0]$, x^2 is monotonically decreasing. Thus, $cos(x)$ is increasing. Similarly, on $[0, \frac{\pi}{2}]$ $cos(x)$ is monotonically decreasing, as x^2 is increasing.

(d) In my opinion, the best range reduction of cosine would be $[0, 2\pi]$, as its period is 2π.

3 Exercise 4

(a) I wrote te following code in Java :

```java
float n = (float) 3223.231;
float d = (float) 0.342;
for (int i = 1; i <= 30; i++) {
  System.out.println(Float.toString(n) + " " + |
    Float.toString(d) + " " + Float.toString((float)(n/d)));
  float f = (float) (2.0 - d);
  n *= f;
  d *= f;
}
```

The first 7 lines of the output are :

3223.231 0.342 9424.651
5344.1167 0.56703603 9424.651
7657.927 0.8125422 9424.651
9093.465 0.96485955 9424.651
9413.014 0.99876523 9424.651
9424.637 0.99999845 9424.651
9424.651 1.0 9424.651

and the following 23 lines coincide with the last one.

(b) The absolute error is 0 at every iteration, so not sure how to find the order of convergence ...

$$9424.651 = \frac{3223.231}{0.342}$$

Thus, the given iteration computes $\frac{n}{d}$.

4 Exercise 5

$$x^2 + 5x + 2 = 0$$
$$x(x+5) + 2 = 0$$
$$x(x+5) = -2$$

Thus, we can deduce the following two expressions:

$$(1)\ x = -\frac{2}{x+5}$$
$$(2)\ x = -\frac{2}{x} - 5$$

So, the two iterations in order fo fin the roots are :

$$\boxed{x_{n+1} = -\frac{2}{x_n+5} \text{ and } x_{n+1} = -\frac{2}{x_n} - 5}$$

5 Exercise 8

(b)

$$\arctan(x) \approx x - \frac{x^3}{3}$$

Thus, in base 10 there is one more significant figure added at every iteration. So in total we need 10 iteration for every significant figure in base 10. =>a value in the [30,40] interval of iterations in binary.

(c) The CORDIC algorithm works as it relies on the fact that the following series can take any value within [-2,2] :

$$\pm 1 \pm \frac{1}{2} \pm \frac{1}{4} \pm \frac{1}{8} \pm ... \ (1)$$

If instead I choose the series :

$$\pm 1 \pm \frac{1}{3} \pm \frac{1}{9} \pm \frac{1}{27} \pm ...$$

I can observe that it can no longer take any value within [-2,2]. So, if we multiply the series by 45^o we get that :

$$\pm 45^o \pm \frac{45^o}{3} \pm \frac{45^o}{9} \pm \frac{45^o}{27} \pm ...$$

does not represent any angle in $[-90^o, 90^o]$, so can't be used for CORDIC algorithm.
In conclusion, if $arctan(0.5) < \frac{90^o}{4}$ CORDIC does not work.

(d) By multiplying the series (1) by 45^o, we get the a new series, that can get any value within $[-90^o, 90^o]$(i.e. in the first quadrant). We know that, when calculating the cosine, any angle can be reduced to an angle in the first quadrant. So, any angle $\alpha > 90^o$ can be brought to the first quadrant and just make the calculations with our new value.
In the same time, if $45^0 < \alpha < 90^o$ the algorithm works without any issue.
In conclusion, the algorithm also works for an angle greater than 45^o.

6 Exercise 10

The partial derivatives for F1 are :

$$\frac{\partial f_1}{\partial x} = 1000 - 500y$$

and

$$\frac{\partial f_1}{\partial y} = 1000x - 500$$

The partial derivatives for F2 are :

$$\frac{\partial f_2}{\partial x} = \frac{2y(x-1)}{(x-y)^2}$$

and

$$\frac{\partial f_2}{\partial y} = \frac{2x(1-y)}{(x-y)^2}$$