MTH 655, Winter 2007, LAB1

The goal of this assignment is to become familiar with the programming environment, to explore the finite precision and arithmetic, and to explore the first few concepts necessary in understanding Newton's method.

1. **Explore:** When solving the problem: find $\pi/4 < x < \pi/2$ such that $\sin x = 1$, what solution do you get ? What if you search for such a number on the computer ? (do not use arcsin, remember about looking at all possible digits). In other words, is there a positive number $\pi/4 < x < \pi/2$ such that $\sin x$ equals 1 on the computer ? How many such numbers exist ? **Explain**.

2. Let us have some sequence of iterates $\{x_n\}$ converging to some number x. That is, the sequence of errors in the n-th iteration $\varepsilon_n := |x - x_n|$ converges to 0.

We say that this iteration has q-convergence rate α if $\varepsilon_{n+1} \leq C\varepsilon_n^\alpha$. C is called a factor[1] **Note:** with $\alpha = 1$, the convergence is *linear*, $\alpha = 2$, the convergence is *quadratic*. See [Kelley, Chapter 4.1] for additional definitions and conditions. **Explain** the conditions.

A good estimate for α can be provided by calculating

$$\alpha \approx \frac{\log(\varepsilon_{n+1}) - \log(\varepsilon_n)}{\log(\varepsilon_n) - \log(\varepsilon_{n-1})}$$

for a few subsequent iterations. When x_n is close to x, the calculated ratio should be close to α. **Explain** why.

Now define

- $a_k = 1 + \frac{1}{2^k}$.
- $b_k = 3 + \frac{1}{k!}$.
- $c_1 = 2$; for $k > 1 : c_k = \frac{c_{k-1}^2+1}{2c_{k-1}}$
- $d_1 = 2$; for $k > 1 : d_k = d_{k-1} - \frac{d_{k-1}^2-1}{4}$

Explore: for each sequence, find its limit and convergence rate, if any. **Explain** using calculus.

Hint for a) MATH: Note $a_{k+1} - 1 = \frac{1}{2}(a_k - 1)$.

Hint for a) MATLAB: you can set up example one as follows

```
n=10;a=zeros(1,n);for j=1:n a(j)=1+1/(2^j);end;
```

Choose some good n here: remember about undeflow. (For some problems the convergence will be neither linear nor quadratic, see the textbook to identify what kind it is).

3. Newton's algorithm: an M-file as provided by link on the red sheet.

[1](Note: in some other sources q is called the order and C the rate. Sorry, I do not explain it, I just report it.)

Execute the algorithm `newton(2,1e-14)` that is, with initial guess 2, tolerance 1e-14. **Explore:** Use other tolerance and initial guesses. Test q-convergence rate.

Determine experimentally the interval around the true solution such that if initial guess is included in it, the convergence will be quadratic. **Explain** using information in class (forthcoming) and theory from [Kelley, Chapter 5].

Change the function to $f(x) = x^2 - 2x + 1$ and test convergence to the root. What problems do you anticipate ?

Improve the algorithm: consider the following issues:

- what happens when you call `newton(0,1e-8)` ? how to prevent it ?
- what to do if the algorithm is not converging ?
- change stopping criteria
- add testing for difference between subsequent iterates

Extra: The last part has an interesting feature in that a good *estimate* for ε_{n+1} is $\varepsilon_{n+1} \approx \upsilon_{n+1} = |x_{n+1} - x_n|$. Test ir ?

---

4. **Extra:** Stability of numerical derivatives:

Use the one-sided difference formula $D_h f(x) = \frac{f(x+h)-f(x)}{h}$ to approximate the derivative of $f(x) = \cos(x)$ at $x = .5$, with h ranging from $1E-1$ down to the machine epsilon (step by a factor of 1/10). Discuss behavior of the error. What is the behavior of central derivative ? (You may want to use `loglog` plot).