HW 4 - 131AH

ASHER CHRISTIAN 006-150-286

1. Exercise 1

Using a calculator of your choice, determine both the absolute and relative errors in the approximation of $p*$ by p. Please report all errors with 5 digits of accuracy.

1. *(a)* $p* = \sqrt{34}, p = 5.8$ The absolute error is

$$|p - p*| = |5.8 - \sqrt{34}| = 0.030952.$$

The relative error is

$$|\frac{p - p*}{p*}| = |\frac{5.8 - \sqrt{34}}{\sqrt{34}}| = 0.0053082.$$

2. *(b)* $p* = 9!, p = \sqrt{18\pi}(\frac{9}{e})^9$ The absolute error is

$$|p - p*| = |\sqrt{18\pi}(\frac{9}{e})^9 - 9!| = 3343.1.$$

The relative error is

$$|\frac{p - p*}{p*}| = 0.0092128.$$

2. Exercise 2

Find the third order Taylor polynomail $P_3(x)$ for the function $f(x) = \sqrt{x+1}$ about $x_0 = 0$. Approximate $\sqrt{0.5}, \sqrt{0.75}, \sqrt{1.25}$ and $\sqrt{1.5}$ using $P_3(x)$ and compute absolute errors with 3 digits of accuracy.

$$P_3(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + f^{(3)}(x_0)\frac{(x - x_0)^3}{3!}.$$

$$f(x_0) = \sqrt{x_0 + 1} = 1$$
$$f'(x_0) = \frac{1}{2\sqrt{x_0 + 1}} = \frac{1}{2}$$
$$f''(x_0) = -\frac{1}{4}(x_0 + 1)^{-\frac{3}{2}} = -\frac{1}{4}$$
$$f^{(3)}(x_0) = \frac{3}{8}(x_0 + 1)^{-\frac{5}{2}} = \frac{3}{8}$$
$$P_3(x) = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3.$$
$$|P_3(-0.5) - \sqrt{0.5}| = 0.00383.$$
$$|P_3(-.25) - \sqrt{.75}| = .000186.$$
$$|P_3(.25) - \sqrt{1.25)}| = 0.000130.$$

Date: 15.11.24.

$$|P_3(.5) - \sqrt{1.5}| = 0.00182.$$

3. Exercise 3

Use the IVT to show that there is a solution to the equation

$$g(x) = xcos(x) - 2x^2 + 3x - 1 = 0.$$

On the interval $[0.2, 0.3]$

$$g(0.2) = -0.28399 < 0.$$

$$g(0.3) = 0.0066009 > 0.$$

By IVT since $g(x)$ is continuous there exists some $c \in [0.2, 0.3]$ s.t. $g(c) = 0$

4. Exercise 4

Let $f(x) = 1.01e^{4x} - 4.62e^{3x} - 3.11e^{2x} + 12.2e^x - 1.99$

1. Use three-digit rounding arithmetic and the fact that $e^{1.53} = 4.62$ to evaluate $f(1.53)$ directly using only addition
   Let $x = e^{1.53}$

$$f(1.53) \approx Fl(Fl(Fl(Fl(1.01 * Fl(x * Fl(x * Fl(x * x))))$$

$$-Fl(4.62 * Fl(x * Fl(x * x)))) - Fl(Fl(3.11 * Fl(x * x)) + Fl(12 * x))) - 1.99).$$

$$= -6.79.$$

2. Rewrite $f(x)$ in nested form as done in class.

$$f(x) = e^x(e^x(e^x(e^x(1.01) - 4.62) - 3.11) + 12.2) - 1.99.$$

3. Redo the calculation in part (a) using the nested form Computing by hand the answer we get is -7.07
4. Given that $f(1.53) = -.761$ the absolute value error of the first estimate is .82 and the absolute value error of the second is .54. The second estimate is has less absolute value error and is therefore more accurate.

5. Exercise 5

Show that the sequence $p_n = (\frac{1}{10})^n$ *converges linearly to* $p = 0$

First we must show that $\lim_{n\to\infty} p_n$ exists and is equal to 0. Clearly the denominator of p_n grows without bound. Consider

$$\frac{|p_{n+1} - 0|}{|p_n - 0|^\alpha} = \frac{\frac{1}{10^{n+1}}}{\frac{1}{10^{\alpha n}}} = 10^{n(\alpha-1)-1}.$$

This is constant irrespective of n so its limit is the equal. If $\alpha = 1$ this value is $\frac{1}{10} < 1$ which satisfies linear convergence

Show that the sequence $p_n = 10^{-2^n}$ *converges quadratically to* $p = 0$

First we show that $\lim_{n\to\infty} p_n = 0$.This is clear as when represented as a fraction the denominator grows without bound. consider

$$\frac{(10^{2^n})^\alpha}{10^{2^{n+1}}} = 10^{\alpha 2^x - 22^x} = 10^{(\alpha-2)^x}.$$

clearly if $\alpha \leq 2$ then the limit converges either to 0 or 1 picking $\alpha = 2$ we get the limit $= 1$ and so it satisfies quadratic convergence.

6. Exercise 6

Suppose a numerical method at the k-th step produces an approximation x_k for the root of some continuous function f in the interval $[a_k, b_k]$. Find the root of the line that passes through the points $(a_k, f(a_k))$ and $(b_k, f(b_k))$
The equation for a line passing through those two points is given by

$$y = (x - a_k)\frac{f(b_k) - f(a_k)}{b_k - a_k} + f(a_k).$$

Solving this equation for x when $y = 0$ yields

$$x = -\frac{f(a_k)(b_k - a_k)}{f(b_k) - f(a_k)} + a_k.$$

Answering the question.

7. Exercise 7

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

double Bisection_Method(double start, double end, double (*
    f)(double));
double Regula_Falsi_Method(double start, double end, double
     (*f)(double));
double f1(double x);
double f2(double x);

double EPSILON = pow(10,-6);

int main(void){
    double s1 = 1.75; //start for part (a)
    double e1 = 2.95;  //end for part (a)
    double s2 = 1.0; //start for part (b) 1
    double e2 = 1.2; //end for part (b) 1
    double s3 = 1.0; //start for part (b) 2
    double e3 = 2.0; //end for part (b) 2
    cout << setprecision(16);
    cout << "Bisection part(a): " << Bisection_Method(s1,e1
        ,f1) << endl;
    cout << "Regula Falsi part(a): " << Regula_Falsi_Method
        (s1,e1,f1) << endl;
    cout << "Bisection part(b) 1st interval: " <<
        Bisection_Method(s2,e2,f2) << endl;
    cout << "Regula Falsi part(b) 1st interval: " <<
        Regula_Falsi_Method(s2,e2,f2) << endl;
    cout << "Bisection part(b) 2nd interval: " <<
        Bisection_Method(s3,e3,f2) << endl;
    cout << "Regula Falsi part(b) 2nd interval: " <<
        Regula_Falsi_Method(s3,e3,f2) << endl;
}
double f1(double x){
```

```
    return (x-1) * (x-2) * (x-3);
}
double f2(double x){
    return x*(pow(x,5) -1) - 1;
}

double Bisection_Method(double start, double end, double (*
    f)(double)){
    int iterations = 0;
    if((f(start) * f(end)) > 0){
        cout << "invalid start and end" << endl;
        return start;
    }
    double p1;
    do{
        p1 = (start + end)/2.0;
        if((f(p1) * f(start) > 0)){
            start = p1;
        }
        else{
            end = p1;
        }
        iterations++;
    }
    while(abs(f(p1)) >= EPSILON);
    cout << "Iterations: " << iterations << "   Approximate
         root: ";
    return p1;
}

double Regula_Falsi_Method(double start, double end,
    double (*f)(double)){
    if((f(start) * f(end)) > 0 ){
        cout << "invalid start and end" << endl;
        return start;
    }
    int iterations = 0;
    double p1;
    do{
        p1 = start - (f(start) * (end - start))/(f(end) - f
            (start));
        if((f(p1) * f(start) > 0)){
            start = p1;
        }
        else{
            end = p1;
        }
        iterations++;
    }
    while(abs(f(p1)) >= EPSILON);
    cout << "Iterations: " << iterations << "  Approximate
        root: ";
    return p1;
```

```
    }
```

The output of the code is the following:
Bisection part(a): Iterations: 19 Approximate root: 2.000000762939453
Regula Falsi part(a): Iterations: 5 Approximate root: 1.999999999762428
Bisection part(b) 1st interval: Iterations: 19 Approximate root: 1.134724044799805
Regula Falsi part(b) 1st interval: Iterations: 8 Approximate root: 1.134724105897137
Bisection part(b) 2nd interval: Iterations: 21 Approximate root: 1.134724140167236
Regula Falsi part(b) 2nd interval: Iterations: 92 Approximate root: 1.134724053091091
In conclusion, The Regula Falsi Method is not always better than the Bisection Method as detailed in the last case where it took significantly longer to find a solution, but on average it may be quicker as shown in the first two examples.