

Math 151a, Homework 2

Lucas Wheeler

- 1.3.5 For this question we compute the approximation of arctan at the values specified, and calculate $\pi/4$ at those points to approximate π .

```
function [N] = arctanapprox(TOL, M)
    N = 1;
    sum = 0;
    sign = -1;
    while N <= M
        sign = -sign;
        term = sign* (4*((1/5)^(2*N - 1))/(2*N - 1)) - (((1/239)^(2*
            N - 1))/(2*N - 1));
        sum = vpa(sum + term);
        if abs(vpa(pi) - sum*4) < TOL
            return
        end
        N = N + 1;
    end
    N = Inf;
end
```

The code above with a tolerance of 10^{-4} as to have π accurate to the 10^{-3} outputs 3 terms needed to approximate π .

2. 1.3.6c

rate of convergence for $\lim_{n \rightarrow \infty} \left(\sin \frac{1}{n}\right)^2$

$$\left|\sin \frac{1}{n} - 0\right| = \left|\sin \frac{1}{n}\right| \leq \frac{1}{n} \implies \left|\left(\sin \frac{1}{n}\right)^2\right| \leq \frac{1}{n^2}$$

therefore

$$|P_n - P| < k * |\beta_n| \implies \left|\left(\sin \frac{1}{n}\right)^2\right| < k * \frac{1}{n^2}$$

Therefore, the sequence converges to 0 with rate $O(\frac{1}{n^2})$

3. 1.3.9

(a) if $F(h) = L + O(h^p)$ there $\exists k > 0$ s.t.:

$$F(h) - L \leq kh^p$$

for a small enough $h > 0$, if $0 < q < p$ and $0 < h < 1$ then $h^q > h^p \implies kh^q > kh^p$. Therefore

$$|F(h) - L| \leq kh^q \text{ and } F(h) = L + O(h^q)$$

n	h	h^2	h^3	h^4	
0.5	0.5	0.25	0.125	0.0625	
(b) 0.1	0.1	0.01	0.001	0.0001	The lowest rate of convergence is $O(h)$ and the
0.01	0.01	0.0001	1×10^{-6}	1×10^{-8}	
0.001	0.001	1×10^{-6}	1×10^{-9}	1×10^{-12}	

highest rate of convergence is $O(h^4)$. For a given h^n as $n \rightarrow \infty$ the rate of convergence increases.

4. 1.3.11 since $\lim_{x \rightarrow \infty} x_n = \lim_{x \rightarrow \infty} x_{n+1} = x$ and $x_{n+1} = 1 + \frac{1}{x_n}$, we can write

$$x = 1 + 1/x \implies x^2 - x - 1 = 0 \implies x = \frac{1}{2} (1 + \sqrt{5})$$

5. 1.3.15

- (a) There is a multiplication at each inner step at the process. This means there are $1 + 2 + 3 + \dots + n$ multiplications which occur. This is equivalent to $\frac{n*(n+1)}{2}$. The number of additions is similar, however, to calculate the first term you don't need to do addition but you do need to do multiplication, meaning there is one less addition step at each of the inner steps, however, there is an additional addition step at each of the outer steps to make up for it. Except for the first outer step. Therefore, the total number of additions is $\frac{(n+2)(n-1)}{2}$.
- (b) Change the form to be:

$$\sum_{i=1}^n a_i * \left(\sum_{j=1}^i b_j \right)$$

This results in the same number of additions, however it reduces the amount of multiplications to one for every step of the outer summation or n multiplications.

6. 2.1.12a

$$f(2.5) = 232.5586, f(-1.5) = -10.2539, \text{frac}2.5 - 1.52 = 0.5 \text{ and } f(0.5) = 0.5273$$

Therefore the next iteration will be on $[-1.5, 0.5]$

$$f(0.5) = 0.5273, f(-1.5) = -10.2539, \text{frac}0.5 - 1.52 = -0.5 \text{ and } f(-0.5) = -1.5820$$

Therefore the next iteration will be on $[-0.5, 0.5]$. The algorithm will then converge at the root $x = 0$.

7. 2.1.15 Running the following code for the bisection method on the function provided on the interval $[0, 1]$ since the water level will be between 0 and the radius r , with a tolerance ϵ that is arbitrarily small, and a maximum number of iterations is 7 as since $|p_n - p| < \frac{a-b}{2^n} \implies |p_n - p| < \frac{1}{2^n} \implies 0.01 < \frac{1}{2^n} \implies 2^{-7} < 0.01 < 2^{-6} \implies n = 7$.

```
function [x] = bisection(a, b, tol, NO)
    j = 1;
    FA = 10/12.4 * (0.5*pi - asin(a) - (a*sqrt(1-a^2))) - 1;
    FB = 10/12.4 * (0.5*pi - asin(b) - (b*sqrt(1-b^2))) - 1;
    if (FA*FB > 0)
        return
    end
    while j < NO
        p = a + (b-a)/2; % better way for writing p = (a+b)/2
        FP = 10/12.4 * (0.5*pi - asin(p) - (p*sqrt(1-p^2))) - 1; %
        evaluate the function at p
        if abs(FP) < tol
            % close enough to actual root, stop iteration
            break;
        elseif (FA*FP) > 0
            % continue search in right half interval
```

```

        a = p;
    else
% continue search in left half interval
        b = p;
    end
j = j + 1;
end
x = p;
fprintf('Iteration number = %d \n', j);
fprintf('p = %.4f \n', p);
fprintf('F(p) = %.4f \n', 10/12.4 * (0.5*pi - asin(p) - (p*sqrt(1-p^2)
)) - 1);

```

On the 7th iteration the algorithm outputs $h = 0.1719 \implies \text{depth} = r - h = 1 - 0.1719 = 0.8281$ and since $p = 0.8338 \implies |p_n - p| = |0.8281 - 0.8338| = 0.0057 < 0.01$

8. 2.1.17 $|p_n - p| < \frac{a-b}{2^n} \implies |p_n - p| < \frac{1}{2^n}$ since the interval is from 1 to 2. Therefore, we know that the error will be less than 2^{-n} . Since $\log_2(10^4) \approx 13.29 \implies 2^{-14} < 10^{-4} < 2^{-13}$ or the bound of number of iterations is $n \geq 14$.

```

function [x] = bisection(a, b, tol, NO)
    j = 1;
    FA = 10/12.4 * (0.5*pi - asin(a) - (a*sqrt(1-a^2))) - 1;
    FB = 10/12.4 * (0.5*pi - asin(b) - (b*sqrt(1-b^2))) - 1;
    if (FA*FB > 0)
        return
    end
    while j < NO
        p = a + (b-a)/2; % better way for writing p = (a+b)/2
        FP = 10/12.4 * (0.5*pi - asin(p) - (p*sqrt(1-p^2))) - 1; %
            evaluate the function at p
        if abs(FP) < tol
% close enough to actual root, stop iteration
            break;
        elseif (FA*FP) > 0
% continue search in right half interval
            a = p;
        else
% continue search in left half interval
            b = p;
        end
        j = j + 1;
    end
    x = p;
    fprintf('Iteration number = %d \n', j);
    fprintf('p = %.4f \n', p);
    fprintf('F(p) = %.4f \n', 10/12.4 * (0.5*pi - asin(p) - (p*sqrt(1-
p^2))) - 1);

```

by the code above the $p_{14} = 1.3248$

9. 2.1.20 $f(x) = (x - 1)^{10}$, $p_n = 1 + \frac{1}{n}$. When $n > 1$, $f(p_n) = f(1 + 1/n) = (1 - 1 + 1/n)^{10} = n^{-10}$. Since $n > 1$ the smallest $|f(p_n)|$ can be is $2^{-10} < 10^{-3}$
Conversely, $|p - p_n| = |1 - 1 + 1/n| = |1/n|$. This value: $|1/n| < 10^{-3} \implies n > \frac{1}{10^{-3}} = 1000$