

CS-6210: HW 5

James Brissette

November 26, 2018

1 Chapter 2

- 2.6 a) Using the bisection method, we can calculate the value of $\sqrt{\alpha}$ by looking first at the non-linear form of the function:

$$x = \sqrt{\alpha}$$
$$f(\alpha) \equiv x^2 - \alpha = 0$$

Using this equation we can use the bisection method to choose two points. We make sure the points we choose, a_1 and b_1 , yield solutions on either side of the x-axis (e.g. $f(a)f(b) < 0$) to ensure there is a zero somewhere in between (an assumption we can only make if the function is continuous and smooth). We then take the bisection at point c_1 where c_i is defined as $(a_i + b_i)/2$ and identify the sub interval that contains the zero (e.g. changes signs) and update our points. If it's in the left interval, b_2 becomes c_1 and $a_2 = a_1$ or else if it's in the right interval, $a_2 = c_1$ and $b_2 = b_1$.

We repeat this process until our error is below our tolerance.

- b) Using Newton's method, we similarly solve in terms of x in order to get our equation into a form in which we can differentiate it. Newton's method follows the form:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Following this, we can substitute our equation into f and its calculated derivative into f' and calculate the next value of x . For example, using the function given in the problem, starting with $\alpha = 36$ and $x = 5$ we could iterate through and solve as follows:

x	$error$
5	11
6.1000000000000000e + 00	-1.2099999999999994e + 00
6.000819672131148e + 00	-9.836737436181409e - 03
6.000000055980886e + 00	-6.717706284575797e - 07
6.0000000000000001e + 00	-1.421085471520200e - 14
6	0

See below for the MATLAB script used to generate this example:

```
function [output] = ch2q6(x0,alpha)
tol = 1e-16;
error = 1e10;
x = x0;
while abs(error) > tol
    f = x^2 - alpha;
    fp = 2*x;
```

```

    x = x - (f / fp);
    error = 0 - (x^2 - alpha)
end
output = x;
end

```

- 2.8 a) Based on the rate of convergence of the computed solution in the data table, this looks like the Newton method. We see our first iteration gives us an error of $2.5e-01$, followed by $2.5e-02$, $3.04e-04$, $4.64e-08$ and $1e-15$. Using the formula given in 2.15, the calculated γ here is approximately 2.66096, 2.19537, 2.085077, and 2.045413 for each respective iterative error. We know that given an initial starting point close to the solution the Newton method converges at a rate of $\gamma \approx 2$ what we see here are results consistent with what we would expect from this method.

- 2.21 a) Since we are given a function f for which we know there must be some value y such that plugging y into f we get zero, and we are given a function $y + x = e^{-6y}$ that is difficult to solve for y in terms of x , we can use bisection to solve for the value of y in the first equation that is necessary for us to get plugging in some value of x into the second. This will allow us to find the correct value of x .

The first step is to find the value of y that gives us $f = 0$, and to do this we take a starting interval for our bisection to look over. We can quickly see that in the trivial case if we plug in 0 we get 1 and this gives us a convenient upper bound, b . If we plug in -1 we see we get -3 and this is a convenient lower bound, a . We might also have found this by plotting the curve and estimating these bounds.

We then begin bisecting the interval until we find the correct value of y and we subsequently plug that into our second function to solve for the appropriate value of x .

- b) Using Newton's method is a little more challenging in that we need to solve for the first derivative. By writing f as $f(y(x))$ we see that we want to solve for f in terms of x . If we take the function $y + x = e^{-6y}$ and solve it for y it gives us an equation that we then use in place of f in Newton's method. Since y is solved in terms of x we differentiate it for the denominator and we proceed to solve.
- c) Using the method described in part *a* I used the following script to solve for y and then subsequently used that y to solve for x . This gave me a value of $y = -3.221853546260856e-01$ and $x = 7.233170030518265e+00$

```

function [output] = ch2q21()
a = -1;
b = 0;
c = (a+b)/2;

tol = 1e-16;
error = 1e10;

while abs(error) > tol
    if (a^3 + 3*a + 1)*(c^3 + 3*c + 1) < 0
        b = c;
    else
        a = c;
    end
    c = (a + b)/2;
    error = 0 - (c^3 + 3*c + 1);
end

```

```
output = c;  
end
```

2.22 a) The second-order Taylor approximation centered at x_i takes the form

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2$$

Using the quadratic formula to solve for x we have

$$(x - x_i) = \frac{-f'(x_i) \pm \sqrt{f'(x_i)^2 - 2f''(x_i)f(x_i)}}{f''(x_i)}$$

$$x = x_i - \frac{f'(x_i) \pm \sqrt{f'(x_i)^2 - 2f''(x_i)f(x_i)}}{f''(x_i)}$$

- b) There are a few things that seem to be potential issues. First, calculating the square root every time requires solving both the first and second derivatives which may not be feasible on complicated functions and which otherwise may be very computationally expensive. Next, the \pm introduces additional logic that must be evaluated to determine which sign to use. Third, given the author's comment that this approximation may be more accurate when you are close to the solution, it begs the question whether these additional calculations will amplify your error when poor initial starting points are selected.
- c) Given that you are close to the solution, it depends on what side of the root you are on. On the one hand you may need to add the add the sqrt term to get closer, or else you may need to subtract it off. However, if you are close to the solution, it may not matter given the rate of convergence for the Newton method.
- d) We know from the Newton Method that $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$. By rearranging the terms we can solve for the equation given in the problem as follows:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)}$$

$$(x_{i+1} - x_i)(x_{i+1} - x_i) = -(x_{i+1} - x_i) \frac{f(x_i)}{f'(x_i)}$$

$$(x_{i+1} - x_i)^2 = -(x_{i+1} - x_i) \frac{f(x_i)}{f'(x_i)}$$

Substituting this into our second-order Taylor approximation gives the following:

$$0 = f(x_i) + f'(x_i)(x_{i+1} - x_i) - \frac{1}{2}f''(x_i)(x_{i+1} - x_i) \frac{f(x_i)}{f'(x_i)}$$

$$0 = f(x_i) + (x_{i+1} - x_i) \left[f'(x_i) - \frac{1}{2}f''(x_i) \frac{f(x_i)}{f'(x_i)} \right]$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i) - \frac{1}{2}f''(x_i) \frac{f(x_i)}{f'(x_i)}} * \frac{2f'(x_i)}{2f'(x_i)}$$

$$x_{i+1} = x_i - \frac{2f(x_i)f'(x_i)}{2f'(x_i)^2 - f''(x_i)f(x_i)}$$

2.25 a)

2 Chapter 8

- 8.6 1. Starting with $y = v_1 x^{v_2}$, we can re-write y in terms of x as $g(x) = v_1 x^{v_2}$. Taking the log of both sides, we get the form given in 8.29:

$$G(X) = V_1 + V_2 X$$

Taking the log of the right side we see $G(X) = \log(v_1) + v_2 \log(x)$, and accordingly we see $V_1 = \log(v_1)$, $X = \log(x)$, and $V_2 = v_2$. From $X = \log(x)$ we conclude that for any $X_i = \log(x_i)$ and from taking the log of the left side of the equation, $G(X) = \log(g(x)) = \log(y) = Y$ and so $Y_i = \log(y_i)$