HW04 Taylor's Polynomials and Newton's Method

Problem 1 (2 points)

(Analytical) Find the 3rd-order Taylor's Polynomial for $f(x) = \sin(x)$ centered at $a = 0$ (*i.e.*, the Maclaurin polynomial). What is the error term, $E_3(x)$?

Solution

$f(x) = \sin(x)$

$f'(x) = \cos(x)$

$f''(x) = -\sin(x)$

$f^{(3)}(x) = -\cos(x)$

$T_3(x) = \sin(0) + \cos(0)(x) - \frac{\sin(0)}{2}x^2 - \frac{\cos(0)}{6}x^3$

$= 0 + x - 0 - \frac{x^3}{3!}$

$E_3(x) = \frac{\sin(\xi_n(x{:}a))}{4!}x^4$

Problem 2 (2 points)

(Analytical) Find the 4rd-order Taylor's Polynomial for $f(x) = \ln(x)$ centered at $a = 1$. What is the error term, $E_4(x)$?

Solution

$f(x) = \ln(x)$

$f'(x) = \frac{1}{x}$

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

$f^{(3)}(x) = 2\frac{1}{x^3}$

$f^{(4)}(x) = -6\frac{1}{x^4}$

$T_4(x) = \ln(1) + (x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \frac{1}{4}(x-1)^4$

$E_4(x) = \frac{1}{5(\xi_n(x:a))^5}x^5$

Problem 3 (2 points)

(Analytical) Use Newton's Method to get an iterative method for finding the square root of a number a. Simplify your formula so that there are no subtractions. Show your derivation in a **Markdown** cell. Hint: what function has $\sqrt{a}$ as a root?

Solution

Our function is $x^2 - a$. Newton's method says that $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

Thus, we have $x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n}$

$x_{n+1} = \frac{2x_n^2 - x_n^2 + a}{2x_n}$

$x_{n+1} = \frac{x_n^2 + a}{2x_n}$

Problem 4 (2 points)

(In Julia) Use the formula you got for your square root solver in Problem 1 to approximate the square root of 2. For simplicity, you can do a for-loop over five to ten iterates of your formula. Print out your iterates to observe the convergence.

Solution

In [1]:
```
function iter_method(x, a)
    numerator = x*x +a
    denominator = 2*x
    return numerator/denominator
end
```

Out[1]:
```
iter_method (generic function with 1 method)
```

In [2]:
```
x = 1
a = 2

for i = 1:10
    x = iter_method(x, a)
    println(x)
end
```

```
1.5
1.4166666666666667
1.4142156862745099
1.4142135623746899
1.414213562373095
1.414213562373095
1.414213562373095
1.414213562373095
1.414213562373095
1.414213562373095
```

Problem 5 (2 points)

(In Julia) Use Newton's Method to find the two distinct roots of $f(x) = x^3 + x^2$, the double root at $x = 0$ and the simple root at $x = -1$, by choosing appropriate initial values x_0 (other than 0 or -1). Print out your iterates to observe the convergence. For the double root, modify the problem so that you get quadratic convergence. You may use a for-loop with 12 iterations to observe the convergence.

Solution

$f'(x) = 3x^2 + 2x$

In [3]:
```julia
function iter_method(x)
    numerator = x ^ 3 + x ^ 2
    denominator = 3 * (x ^ 2) + 2 * x
    return x - (numerator/denominator)
end
```

Out[3]:
```
iter_method (generic function with 2 methods)
```

In [4]:
```julia
x = -2

for i = 1:12
    x = iter_method(x)
    println(x)
end
```

```
-1.5
-1.2
-1.05
-1.0043478260869565
-1.0000373203955963
-1.000000002785312
-1.0
-1.0
-1.0
-1.0
-1.0
-1.0
```

For quadratic convergence, $f'(x) \neq 0$ for x near 0

Let's factor out the x term, to get $f(x) = x^2 + x$.

This still has a root at 0, and $f'(x)$ is not 0 for x near 0, so we can use Newton's

In [5]:
```
function iter_method(x)
    numerator = x ^ 2 + x
    denominator = 2x + 1
    return x - (numerator/denominator)
end
```

Out[5]:
```
iter_method (generic function with 2 methods)
```

In [6]:
```
x = 1

for i = 1:12
    x = iter_method(x)
    println(x)
end
```

```
0.33333333333333337
0.06666666666666665
0.00392156862745098
1.5259021896696368e-5
2.3283064370807974e-10
5.421010862427522e-20
0.0
0.0
0.0
0.0
0.0
0.0
```