Table of Contents

Part 1

Finds a solution using Newton's method to sinx _ e^#x = 0

```
fprintf('Part 1\n')

%Set tolerance here
tolerance = 10^-6;
currentIteration = 0;
%Set initial value here
p0 = 0;
solutionFound = false;

while ~solutionFound
    currentIteration = currentIteration + 1;
    p = p0 - ((sin(p0)-exp(-p0))/(cos(p0)+exp(-p0)));
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Tolerance:  %e,  Approximation:  %.6f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
```

```
Part 1
Tolerance:  1.000000e-06,  Approximation:  0.588533,  Iterations:  5
```

Part 2

Finds an approximation using Newton's method to root(3)

```
fprintf('\nPart 2\n')

%Set tolerance here
tolerance = 10^-8;
currentIteration = 0;
%Set initial value here
p0 = 2;
solutionFound = false;

while ~solutionFound
```

```
    currentIteration = currentIteration + 1;
    p = p0 - ((p0^2) - 3)/(2*p0);
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
fprintf("Newton's method required 4 iterations while Bisection
 required 26.\n")
```

```
Part 2
Tolerance:  1.000000e-08,  Approximation:  1.73205081,  Iterations:  4
Newton's method required 4 iterations while Bisection required 26.
```

Part 3

Finds the interest rate needed to achieve a specified financial goal

```
fprintf('\nPart 3\n')

%Set tolerance here
tolerance = 10^-8;
currentIteration = 0;
%Set initial value here
p0 = .07; % Value chosen based off of typical stock market returns
solutionFound = false;

while ~solutionFound
    currentIteration = currentIteration + 1;
    fp0 = ((6000/p0)*(((1+p0)^30)-1)) - 1000000;
    fprimep0 = (6000/(p0^2))*((-(1+p0)^30)+((30*p0*((1+p0)^29))+1));
    p = p0 - (fp0)/(fprimep0);
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
interestRate = p*100;
fprintf('%.6f%% is the minimum interest rate assuming it is compounded
 yearly\n', interestRate)


currentIteration = 0;
%Set initial value here
p0 = .07; % Value chosen based off of typical stock market returns
solutionFound = false;
```

```
while ~solutionFound
    currentIteration = currentIteration + 1;
    fp0 = ((1000/p0)*(((1+p0)^(30*12))-1)) - 1000000;
    fprimep0 = ((360000*(1+p0)^359)/p0)-((1000*((1+p0)^360)-1)/
(p0^2));
    p = p0 - (fp0/fprimep0);
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
interestRate = p*100;
fprintf('%.6f%% is the minimum interest rate assuming it is compounded
 monthly\n', interestRate)
```

```
Part 3
Tolerance:  1.000000e-08,  Approximation:  0.10069281,  Iterations:  6
10.069281% is the minimum interest rate assuming it is compounded
 yearly
Tolerance:  1.000000e-08,  Approximation:  0.00498037,  Iterations:
 32
0.498037% is the minimum interest rate assuming it is compounded
 monthly
```

Part 4

Finds when to inject the drug and what concentration

```
fprintf('\nPart 4\n')

fprintf('The maximum concentration occurs 3 hours after injection
 (found by hand using calculus)\n')

%Set tolerance here
tolerance = 10^-8;
currentIteration = 0;
%Set initial value here
p0 = 1;
solutionFound = false;

while ~solutionFound
    currentIteration = currentIteration + 1;
    p = p0 - ((3*p0/exp(1))-1)/(3/(exp(1)));
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Part A\n')
```

```
fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
fprintf('%.8f should be given initially\n', p)

startingConcentration = p;

currentIteration = 0;
%Set initial value here
p0 = 4; % Max is at 3, and the decrease begins after the max, so a
 number after 3 must be chosen because if 3 is chosen the derivative
 is 0
solutionFound = false;

while ~solutionFound
    currentIteration = currentIteration + 1;
    p = p0 - ((startingConcentration*p0*exp(-p0/3)-(1/4))/
(startingConcentration*exp(-p0/3)*(1-(p0/3))));
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Part B\n')
fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
fprintf('%.0f minutes after the initial injection the second dosage
 should be given\n', p*60)
```

```
Part 4
The maximum concentration occurs 3 hours after injection (found by
 hand using calculus)
Part A
Tolerance:  1.000000e-08,  Approximation:  0.90609394,  Iterations:  2
0.90609394 should be given initially
Part B
Tolerance:  1.000000e-08,  Approximation:  11.07790359,  Iterations:
 6
665 minutes after the initial injection the second dosage should be
 given
```

Part 5

Approximate the zero of the function f(x) = x^2#2xe^#x+e^#2x

```
fprintf('\nPart 5\n')

%Set tolerance here
tolerance = 10^-8;
currentIteration = 0;
%Set initial value here
p0 = 1;
solutionFound = false;
```

```
while ~solutionFound
    currentIteration = currentIteration + 1;
    p = p0 - ((p0^2)-(2*p0*exp(-p0))+exp(-2*p0))/((2*p0)-2*(exp(-p0)-
(p0*exp(-p0))+exp(-2*p0)));
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Using Newton\''s Method\n')
fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)


currentIteration = 0;
%Set initial value here
p0 = 1;
solutionFound = false;

while ~solutionFound
    currentIteration = currentIteration + 1;
    fp0 = (p0^2)-(2*p0*exp(-p0))+exp(-2*p0);
    fprime0 = (2*p0)-2*(exp(-p0)-(p0*exp(-p0))+exp(-2*p0));
    fdoubleprime0 = 2 + (2*((2*exp(-p0))-(p0*exp(-
p0))+(2*exp(-2*p0))));
    p = p0 - (fp0*fprime0)/((fprime0^2)-(fp0*fdoubleprime0));
    if abs((p-p0)/p) < tolerance
        solutionFound = true;
    end
    p0 = p;
end

fprintf('Using Newton\''s Modified Method\n')
fprintf('Tolerance:  %e,  Approximation:  %.8f,  Iterations:  %d\n',
 tolerance, p, currentIteration)
fprintf('The modified Newton''s method takes substanially less
 iterations than Newton''s method to converge\n')
```

```
Part 5
Using Newton's Method
Tolerance:  1.000000e-08,  Approximation:  0.56714329,  Iterations:
 27
Using Newton's Modified Method
Tolerance:  1.000000e-08,  Approximation:  0.56714329,  Iterations:  4
The modified Newton's method takes substanially less iterations than
 Newton's method to converge
```

Published with MATLAB® R2020a