

Copyright

```
close all; format compact; clc;
fprintf("Engineer: Rodrigo Becerril Ferreyra\n");
fprintf("Company: California State University, Long Beach\n");
fprintf("Project Name: Task 4 Supplemental\n");
fprintf("Date: 09 October 2020\n");
```

```
Engineer: Rodrigo Becerril Ferreyra
Company: California State University, Long Beach
Project Name: Task 4 Supplemental
Date: 09 October 2020
```

Problem 1

```
fprintf("\nProblem 1\n");
clear variables;
% Fixed point method

f = @(x) 2.*sin(sqrt(x)) - x; % function to find root for
g = @(x) 2.*sin(sqrt(x)); % obtained by solving f for x

x = 0.5; % initial guess

last_x = x; x = g(x); % algorithm
iters = 25;

while (abs(last_x - x) >= 1e-6) && (iters > 0)
    last_x = x;
    x = g(x);
    iters = iters - 1;
end

fprintf("f(x) = 0 for x = %f\n", x);
```

```
Problem 1
f(x) = 0 for x = 1.972381
```

Problem 2

```
fprintf("\nProblem 2\n");
clear variables;
% Newton's method

%           x^4      x^3      x^2      x
f = @(x) polyval([0.0074, -0.284, 3.355, -12.183, 5], x);
%           x^3      x^2      x
fp = @(x) polyval([0.0074*4, -0.284*3, 3.355*2, -12.183*1], x); % derivative
of x

x = 16.75; % using x=16.15 gives a root of 0.468480

last_x = x; x = x - (f(x)/fp(x)); % algorithm
iters = 25;

while (abs(last_x - x) >= 1e-6) && (iters > 0)
    last_x = x;
    x = x - (f(x)/fp(x));
    iters = iters - 1;
end

fprintf("f(x) = 0 for x = %f\n", x);
```

Problem 2

f(x) = 0 for x = 18.894766

Problem 3

```
fprintf("\nProblem 3\n");
clear variables;
% Regula Falsi method
% Taken from An Introduction to Programming and Numerical Methods in MATLAB
Page 122 (PDF page 133)

% Function definition
R = 0.518; T = -50 + 273.15; p = 65e3;
pc = 4580; Tc = 191;
a = 0.427*(R^2)*(Tc^2.5)/pc; b = 0.0866*R*(Tc/pc);
f = @(v) (R*T)./(v-b) - a./(v.*(v+b).*T^0.5) - p;
% f = @(v) p - (R*T)./(v-b) + a./(v.*(v+b).*sqrt(T));

% Initial guesses
x0 = 0.0027; x1 = 0.0028; x2 = x0;
iters = 25;

% loop
while (abs(f(x2)) >= 1e-6) && (iters > 0)

    f0 = f(x0); f1 = f(x1);
    x2 = x0 - f0 * (x1-x0)/(f1-f0);
    if f(x2) * f0 < 0
        x1 = x2;
    else
        x0 = x2;
    end

    iters = iters - 1;

end

fprintf("The mass M is %f kg (v = %f m^3/kg)\n", 3/x2, x2);
```

Problem 3

The mass M is 1090.833338 kg (v = 0.002750 m³/kg)

Problem 4

```
fprintf("\nProblem 4\n");
clear variables;
% Secant method

% formula
R = 3; V = 30;
f = @(h) (pi.*h.^2 .* (3*R - h) ./ 3) - V;

% initial guesses
h0 = 0; h1 = 3;
h = h1-f(h1)*((h1-h0)/(f(h1) - f(h0))); % algorithm
last_h = 1000000;

iters = 25;

fprintf("%-2s %-6s %-10s\n", "#", "height", "error");
fprintf("%-2d %-6.4f %-10f%%\n", 1, h, 100);

while (abs(last_h - h) >= 1e-6) && (iters > 0)

    last_h = h;
    h0 = h1;
    h1 = h;
    h = h1-f(h1)*((h1-h0)/(f(h1) - f(h0)));
    iters = iters - 1;
    fprintf("%-2d %-6.4f %-10f%%\n", 26 - iters, h, abs(last_h - h)*100);

end
```

```
Problem 4
# height error
1 1.5915 100.000000%
2 1.9866 39.502557 %
3 2.0295 4.297278 %
4 2.0269 0.265516 %
5 2.0269 0.001310 %
6 2.0269 0.000000 %
```

Problem 5

```
fprintf("\nProblem 5\n");
clear variables;
% Bisection method

% function definition
x = 5000; y0 = 1000; v = 275; g = 9.81;
f = @(theta) x.*tan(theta) - g*x^2./(2*v^2 .* (cos(theta).^2)) + y0 - 250; %
the -250 s is because the target is 250 m above sea level

% initial guesses
a = 0.1; b = 0.2;
c = (a + b)/2;

if sign(f(a)) == sign(f(c))
    a = c;
elseif sign(f(b)) == sign(f(c))
    b = c;
else
    fprintf("Error a, b, c = %f %f %f\n", a, b, c);
end

iters = 25; last_c = 1000000;
while (abs(last_c - c) >= 1e-6) && (iters > 0)

    last_c = c;
    c = (a + b)/2;

    if sign(f(a)) == sign(f(c))
        a = c;
    elseif sign(f(b)) == sign(f(c))
        b = c;
    else
        fprintf("Error a, b, c = %f %f %f\n", a, b, c);
    end

    iters = iters - 1;

end

fprintf("Angle measure is %f radians or %f degrees.\n", c, c*180/pi);
```

Problem 5

Angle measure is 0.183367 radians or 10.506164 degrees.