Differential_Calculus_Basic

```
function [d_dt] = Differential_Calculus_Basic(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Impact of Different Initial Strategies

y_0_0P4

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 1;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5.2 5]);

color_mat = [   [1   0   0  ];
                [1   0.5 0  ];
                [1   1   0  ];
                [0   1   0  ];
                [0   1   1  ];
                [0   0.5 1  ];
                [0   0   1  ];
                [0.5 0   1  ];
                [1   0   1  ];
            ];

for ii = unit: unit: 1 - unit

    for jj = 0.4

        [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 3], [ii, jj]);

        color = color_mat(round(ii / unit), :);

        plot(Y(:,1), Y(:,2), 'Color', color, 'LineWidth', 2);

        hold on;

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
```

```
% hold on;

legend( '{\it{x}}_{0} = 0.1', ...
        '{\it{x}}_{0} = 0.2', ...
        '{\it{x}}_{0} = 0.3', ...
        '{\it{x}}_{0} = 0.4', ...
        '{\it{x}}_{0} = 0.5', ...
        '{\it{x}}_{0} = 0.6', ...
        '{\it{x}}_{0} = 0.7', ...
        '{\it{x}}_{0} = 0.8', ...
        '{\it{x}}_{0} = 0.9', ...
        'NumColumns', 2, ...
        'FontSize', 12, ...
        'Location', 'SouthEast');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

y_0_0P7

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 1;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5.2 5]);

color_mat = [   [1    0    0   ];
                [1    0.5  0   ];
                [1    1    0   ];
                [0    1    0   ];
```

```
                [0      1      1    ];
                [0      0.5 1     ];
                [0      0      1    ];
                [0.5 0      1    ];
                [1      0      1    ];
            ];

for ii = unit: unit: 1 - unit

    for jj = 0.7

        [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 3], [ii, jj]);

        color = color_mat(round(ii / unit), :);

        plot(Y(:,1), Y(:,2), 'Color', color, 'LineWidth', 2);

        hold on;

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{x}}_{0} = 0.1', ...
        '{\it{x}}_{0} = 0.2', ...
        '{\it{x}}_{0} = 0.3', ...
        '{\it{x}}_{0} = 0.4', ...
        '{\it{x}}_{0} = 0.5', ...
        '{\it{x}}_{0} = 0.6', ...
        '{\it{x}}_{0} = 0.7', ...
        '{\it{x}}_{0} = 0.8', ...
        '{\it{x}}_{0} = 0.9', ...
        'NumColumns', 2, ...
        'FontSize', 12, ...
        'Location', 'SouthEast');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');
```

```
xlabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Impact_Beta

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
%       beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    global beta;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
```

```
- x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 0.4;
y_min = 0.4;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1   0   0   ];
                [1   0.5 0   ];
                [1   1   0   ];
                [0   1   0   ];
                [0   0.5 1   ];
                [0   0   1   ];
                [0.5 0   1   ];
                [1   0   1   ];
            ];

global beta;

beta_min = 0.16;
beta_max = 0.24;
beta_unit = 0.02;
```

```
for beta = beta_min : beta_unit : beta_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((beta - beta_min) / beta_unit + 1), :);

            plot(T, Y(:,1), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{\beta}} = 0.16', ...
        '{\it{\beta}} = 0.18', ...
        '{\it{\beta}} = 0.20', ...
        '{\it{\beta}} = 0.22', ...
        '{\it{\beta}} = 0.24', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);
```

```
xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1   0   0   ];
                [1   0.5 0   ];
                [1   1   0   ];
                [0   1   0   ];
                [0   0.5 1   ];
                [0   0   1   ];
                [0.5 0   1   ];
                [1   0   1   ];
            ];

for beta = beta_min : beta_unit : beta_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((beta - beta_min) / beta_unit + 1), :);

            plot(T, Y(:,2), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;
```

```
legend( '{\it{\beta}} = 0.16', ...
        '{\it{\beta}} = 0.18', ...
        '{\it{\beta}} = 0.20', ...
        '{\it{\beta}} = 0.22', ...
        '{\it{\beta}} = 0.24', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Impact_C_E

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
%       C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    global C_E;
```

```
    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 0.4;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1    0    0   ];
                [1    0.5  0   ];
                [1    1    0   ];
                [0    1    0   ];
                [0    0.5  1   ];
                [0    0    1   ];
```

```
                [0.5 0      1    ];
                [1      0      1    ];
            ];

global C_E;

C_E_min = 170;
C_E_max = 330;
C_E_unit = 40;


for C_E = C_E_min : C_E_unit : C_E_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((C_E - C_E_min) / C_E_unit + 1), :);

            plot(T, Y(:,1), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{C}}_E = 170', ...
        '{\it{C}}_E = 210', ...
        '{\it{C}}_E = 250', ...
        '{\it{C}}_E = 290', ...
        '{\it{C}}_E = 330', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
```

```
            'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1    0    0    ];
                [1    0.5  0    ];
                [1    1    0    ];
                [0    1    0    ];
                [0    0.5  1    ];
                [0    0    1    ];
                [0.5  0    1    ];
                [1    0    1    ];
            ];

for C_E = C_E_min : C_E_unit : C_E_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((C_E - C_E_min) / C_E_unit + 1), :);

            plot(T, Y(:,2), 'Color', color, 'LineWidth', 2);

            hold on;

        end
```

```
    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{C}}_E = 170', ...
        '{\it{C}}_E = 210', ...
        '{\it{C}}_E = 250', ...
        '{\it{C}}_E = 290', ...
        '{\it{C}}_E = 330', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Impact_F

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
```

```
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
%       F = 10;

    global F;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 0.4;
y_min = 0.4;
```

```
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [     [1     0     0    ];
                  [1     0.5 0    ];
                  [1     1     0    ];
                  [0     1     0    ];
                  [0     0.5 1    ];
                  [0     0     1    ];
                  [0.5 0     1    ];
                  [1     0     1    ];
              ];

global F;

F_min = 6;
F_max = 14;
F_unit = 2;


for F = F_min : F_unit : F_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((F - F_min) / F_unit + 1), :);

            plot(T, Y(:,1), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
```

```
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{F}} = 6', ...
        '{\it{F}} = 8', ...
        '{\it{F}} = 10', ...
        '{\it{F}} = 12', ...
        '{\it{F}} = 14', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1   0   0   ];
                [1   0.5 0   ];
                [1   1   0   ];
                [0   1   0   ];
                [0   0.5 1   ];
                [0   0   1   ];
                [0.5 0   1   ];
                [1   0   1   ];
            ];

for F = F_min : F_unit : F_max

    for ii = 0.5
```

```
            for jj = 0.5

                [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

                color = color_mat(round((F - F_min) / F_unit + 1), :);

                plot(T, Y(:,2), 'Color', color, 'LineWidth', 2);

                hold on;

            end

        end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{F}} = 6', ...
        '{\it{F}} = 8', ...
        '{\it{F}} = 10', ...
        '{\it{F}} = 12', ...
        '{\it{F}} = 14', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Impact_R

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
%   R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    global R;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%   dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%   dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);
```

```
    d_dt = [dx_dt; dy_dt];

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 0.4;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1    0    0   ];
                [1    0.5  0   ];
                [1    1    0   ];
                [0    1    0   ];
                [0    0.5  1   ];
                [0    0    1   ];
                [0.5  0    1   ];
                [1    0    1   ];
            ];

global R;

R_min = 485;
R_max = 685;
R_unit = 50;


for R = R_min : R_unit : R_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((R - R_min) / R_unit + 1), :);

            plot(T, Y(:,1), 'Color', color, 'LineWidth', 2);
```

```
                hold on;

            end

        end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{R}} = 485', ...
        '{\it{R}} = 535', ...
        '{\it{R}} = 585', ...
        '{\it{R}} = 635', ...
        '{\it{R}} = 685', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1   0   0  ];
                [1   0.5 0  ];
                [1   1   0  ];
                [0   1   0  ];
```

```
                [0      0.5 1     ];
                [0      0      1    ];
                [0.5 0      1    ];
                [1      0      1    ];
            ];

for R = R_min : R_unit : R_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((R - R_min) / R_unit + 1), :);

            plot(T, Y(:,2), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{R}} = 485', ...
        '{\it{R}} = 535', ...
        '{\it{R}} = 585', ...
        '{\it{R}} = 635', ...
        '{\it{R}} = 685', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');
```

```
xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Impact_R_prime

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
%       R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    global R_prime;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) *
(R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1
- x) * (lambda * R - C_G2 + F);
```

```
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%       dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%       dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X.

unit = 0.1;
x_min = 0;
x_max = 0.4;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [   [1      0       0   ];
                [1      0.5     0   ];
                [1      1       0   ];
                [0      1       0   ];
                [0      0.5     1   ];
                [0      0       1   ];
                [0.5    0       1   ];
                [1      0       1   ];
            ];

global R_prime;

R_prime_min = 1000;
R_prime_max = 1200;
R_prime_unit = 50;


for R_prime = R_prime_min : R_prime_unit : R_prime_max
```

```
        for ii = 0.5

            for jj = 0.5

                [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

                color = color_mat(round((R_prime - R_prime_min) / R_prime_unit + 1), :);

                plot(T, Y(:,1), 'Color', color, 'LineWidth', 2);

                hold on;

            end

        end

    end

    % [A, B] = Equilibrium_Calc();
    %
    % scatter(A, B, 5, 'filled', 'red');
    %
    % hold on;

    legend( '{\it{R}}{\it{''}} = 1000', ...
            '{\it{R}}{\it{''}} = 1050', ...
            '{\it{R}}{\it{''}} = 1100', ...
            '{\it{R}}{\it{''}} = 1150', ...
            '{\it{R}}{\it{''}} = 1200', ...
            'NumColumns', 1, ...
            'FontSize', 12, ...
            'Location', 'East');

    set(gca, 'FontSize', 14, 'FontName', 'Cambria');
    set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

    xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
    ylabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

    xlim([x_min, x_max]);
    ylim([y_min, y_max]);

    xticks(0: x_max / 5 :x_max);
```

```
yticks(0: y_max / 5 :y_max);

grid on;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5 3.5]);

color_mat = [    [1    0    0   ];
                  [1    0.5 0   ];
                  [1    1    0   ];
                  [0    1    0   ];
                  [0    0.5 1   ];
                  [0    0    1   ];
                  [0.5 0    1   ];
                  [1    0    1   ];
              ];

for R_prime = R_prime_min : R_prime_unit : R_prime_max

    for ii = 0.5

        for jj = 0.5

            [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 0.4], [ii, jj]);

            color = color_mat(round((R_prime - R_prime_min) / R_prime_unit + 1), :);

            plot(T, Y(:,2), 'Color', color, 'LineWidth', 2);

            hold on;

        end

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

legend( '{\it{R}}{\it{''}} = 1000', ...
```

```
        '{\it{R}}{\it{''}} = 1050', ...
        '{\it{R}}{\it{''}} = 1100', ...
        '{\it{R}}{\it{''}} = 1150', ...
        '{\it{R}}{\it{''}} = 1200', ...
        'NumColumns', 1, ...
        'FontSize', 12, ...
        'Location', 'East');

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('t','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```

Numerical_Simulation

Differential_Calculus_Basic

```
function [d_dt] = Differential_Calculus_Basic(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;

    x = z(1);
    y = z(2);

    %% Calculate the anticipated returns.
```

```
    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) *
(1 - lambda) * (R_prime - C_E - C_L);
    W_L2 = (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2)
+ (1 - x) * (lambda * R - C_G2);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda
* R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%     dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta
* C_L);
%     dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Differential_Calculus_Extended

```
function [d_dt] = Differential_Calculus_Extended(t, z)

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;
    F = 10;

    x = z(1);
    y = z(2);
```

```
    %% Calculate the anticipated returns.

    W_L1 = y * ((1 - lambda) * (R_prime - C_E - C_L) + beta * C_L) + (1 - y) * (1 - lambda) * (R_prime - C_E - C_L);
    W_L2 = y * ((1 - lambda) * R - F) + (1 - y) * (1 - lambda) * R;
    W_L_avg = x * W_L1 + (1 - x) * W_L2;

    W_G1 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1 + R_G2 - C_G1 - C_G2) + (1 - x) * (lambda * R - C_G2 + F);
    W_G2 = x * (lambda * (R_prime - C_E - C_L) + C_L + R_G1) + (1 - x) * lambda * R;
    W_G_avg = y * W_G1 + (1 - y) * W_G2;

    %% Calculate the partial derivatives.

%     dx_dt = x * (1 - x) * ((1 - lambda) * (R_prime - C_E - C_L - R) + y * beta * C_L);
%     dy_dt = y * (1 - y) * (x * (R_G2 - C_G1) - C_G2);

    dx_dt = x * (W_L1 - W_L_avg);
    dy_dt = y * (W_G1 - W_G_avg);

    d_dt = [dx_dt; dy_dt];

end
```

Equilibrium_Calc

```
function [A, B] = Equilibrium_Calc()

    %% Initialize the variables.

    lambda = 0.25;
    R = 585;
    R_prime = 1100;
    C_E = 250;
    C_L = 320;
    beta = 0.2;
    R_G1 = 40;
    R_G2 = 160;
    C_G1 = 85;
    C_G2 = 25;

    %% Calculate the equilibrium point;

    A = C_G2 / (R_G2 - C_G1);
```

```
    B = (1 - lambda) * (R + C_E + C_L - R_prime) / (beta * C_L);

end
```

Main_Function

```
%% Create a 2-D line plot of the data in Y versus the corresponding values in X (Basic).

unit = 0.05;
x_min = 0;
x_max = 1;
y_min = 0;
y_max = 1;

figure(1);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5.2 5]);

for ii = unit: unit: 1 - unit

    for jj = unit: unit: 1 - unit

        [T, Y] = ode23tb('Differential_Calculus_Basic', [0, 3], [ii, jj]);

        plot(Y(:,1), Y(:,2), 'Color', 'blue');

        hold on;

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman Italic');
```

```
xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;

%% Create a 2-D line plot of the data in Y versus the corresponding values in X
(Extended).

unit = 0.05;
x_min = 0;
x_max = 1;
y_min = 0;
y_max = 1;

figure(2);
set(gcf, 'Units', 'Inches', 'Position', [3 3 5.2 5]);

for ii = unit: unit: 1 - unit

    for jj = unit: unit: 1 - unit

        [T, Y] = ode23tb('Differential_Calculus_Extended', [0, 3], [ii, jj]);

        plot(Y(:,1), Y(:,2), 'Color', 'blue');

        hold on;

    end

end

% [A, B] = Equilibrium_Calc();
%
% scatter(A, B, 5, 'filled', 'red');
%
% hold on;

set(gca, 'FontSize', 14, 'FontName', 'Cambria');
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman');

xlabel('x','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman
```

```
Italic');
ylabel('y','FontSize', 16, 'FontWeight', 'light', 'FontName', 'Times New Roman
Italic');

xlim([x_min, x_max]);
ylim([y_min, y_max]);

xticks(0: x_max / 5 :x_max);
yticks(0: y_max / 5 :y_max);

grid on;
```