Contents


```
% CSTR in Series with given volumes. Dilutions varied.

clc
clear
close all

% Plot Configuration for labeling of vessel 1 figure
plot_cfg = struct( xlabel = "Dilution, D1 (1/h)", ...
    ylabel = ["Concentrations (g/L)", "Concentrations (g/L)", "Concentrations (g/L)", ...
    "Productivities (g/h)", "Productivities (g/h)"], ...
    legend = ["Substrate", "Cell", "Product", "Cell Productivity (DX)", "Productivity (PD)"], ...
    lineStyle  = ["-", "-", "-", "-", "-"], lineColor = ["r", "b", "g", "b", "g"], ...
    layout = [2 1], subplotIndex = [1, 1, 1, 2, 2], yAxis = ["L", "L", "L", "L", "L"], ...
    y_max = [inf, inf]);

% Cell and Product Model Constants
mumax  = 0.48;      % max growth rate constant; 1/h
YXS    = 0.20;      % gX/gS, g cell per g substrate
YPS    = 0.5;       % gP/gx, g product per g dry cell
alpha  = 3.3;       % productivity; gP per gX*h
beta   = 0.05;      % productivity; 1/h
KS     = 1.5;       % gS/L, g substrate per L
kd     = 0.01;      % death rate constant; 1/h

% Inlet Concentrations & Maximum Dilution
SF = 75.0;          % g/L, inlet substrate concentration
D_max = 0.99*mumax; % maximum dilution factor (g/l) -- do not go higher than mumax

% Model Equations for Cell and Product for Vessel #1
n_steps   = 100;                      % linspace observation
D1 = linspace(0.00, D_max, n_steps)';    % dilution rate column vector
[S1, X1, P1, DX1, PX1] = vessel_1(D1, kd, KS, mumax, SF, alpha, beta, YXS, YPS);

% Plot results for Vessel #1
y1 = [S1, X1, P1, DX1, PX1];
plot_results(D1, y1, plot_cfg)

% Additional parameters for 2nd vessel in series
V1 = 80;            % vessel #1 volume; L
V2 = 170;           % vessel #2 volume; L

alpha2   = 3.3;      % vessel #2 productivity; gP per gX*h
beta2    = 0.05;     % vessel #2 productivity; 1/h
SF_prime = 75;       % vessel #2 g/L, inlet substrate concentration
D2plot_mult = 1.5 ;  % range of D2 relative to D_max
D2 = linspace(0.0, D2plot_mult*D_max, n_steps+20)';  % Vessel #2 dilution rates
```

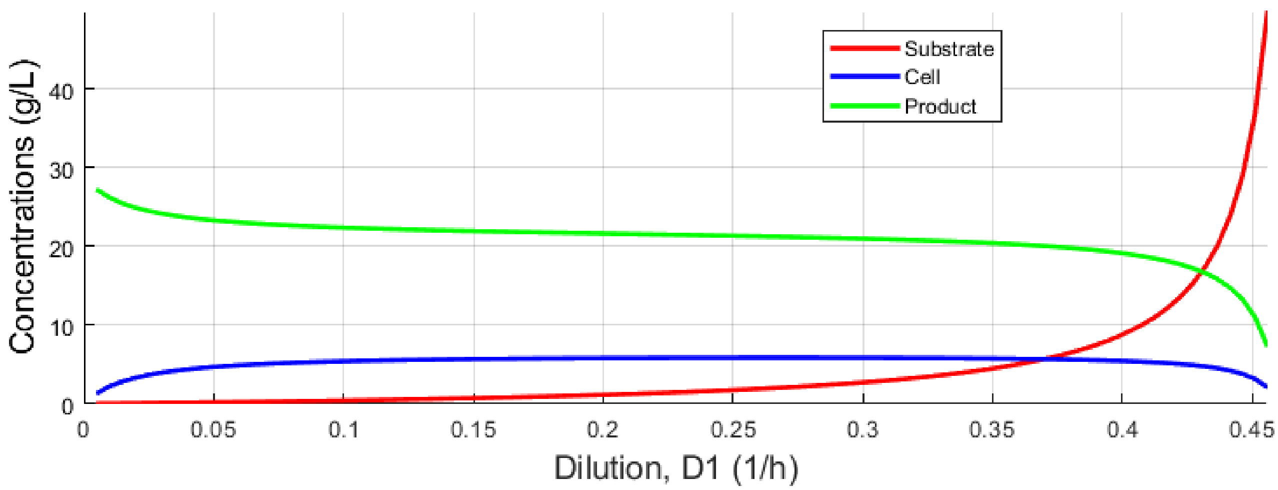
```matlab
% Matrix of dilutions for which calculations will be run
[m.D2, m.D1] = meshgrid(D2, D1);

m.F       =  V1 * m.D1;          % Flow rate from the first vessel (V1 * dilution rate D1)
m.F_prime = (V2 * m.D2) - m.F;   % Net flow into the second vessel

% Set any F_prime <= 0 to NaN (reverse flow is not permissable)
if any(m.F_prime <= 0, "all")
    inv_idx = m.F_prime <=0;
    m.F_prime(inv_idx) = NaN;
end

% initialize matrices for vessel 2 iterative calculations
[m.mu2, m.S2, m.X2, m.P2, m.count] = deal(NaN(size(m.D1)));

fprintf('Computation Progress: %3d %3d\n',0,0)

max_count = 0;      % variable to track maximum number of iterations for convergence
max_iter  = 250;    % maximum iterations allowed
toler     = 1e-4;   % convergence tolerance (1e-4 == 0.01%)

for k1 = 1:length(D1)        % loop through vessel #1 dilution domain
    for k2 = 1:length(D2)    % loop through vessel #2 dilution domain

        fprintf('\b\b\b\b\b\b\b\b %3d %3d',k1, k2)

        % variable extraction
        D1_k1 = D1(k1);    % also stored as m.D1(k1,k2)
        P1_k1 = P1(k1);
        X1_k1 = X1(k1);
        S1_k1 = S1(k1);
        D2_k2 = D2(k2);    % also stored as m.D2(k1,k2)
        F_k1  = m.F(k1,k2);
        F_prime_k2 = m.F_prime(k1, k2);

        % if any variable extraction is nan, then go to next iteration
        if any(isnan([D1_k1, P1_k1, X1_k1, S1_k1, D2_k2, F_prime_k2]))
            continue
        end

        X2 = X1_k1;   % Initial guess of X2
        delta = 1e9;  % Arbitrary big number greater than tolerance
        count = 0;    % Set iteration counter

        while delta > toler && count < max_iter
            mu2 = D2_k2 - ((F_k1 * X1_k1) / (V2 * X2)) + kd;
            S2 = KS * mu2 / (mumax - mu2);
            qp2 = alpha2 * mu2 + beta2;
            X2_num = (F_k1 / V2*S1_k1) + (F_prime_k2 / V2*SF_prime) - D2_k2*S2;
            X2_new = X2_num / (mu2/YXS + qp2/YPS );
            delta = abs((X2_new - X2) / X2);
            X2 = X2_new;
            count = count + 1;
        end

        % skip results where mu2 is outside physical reality
        if mu2 <=0 || mu2 >= mumax
            continue
```

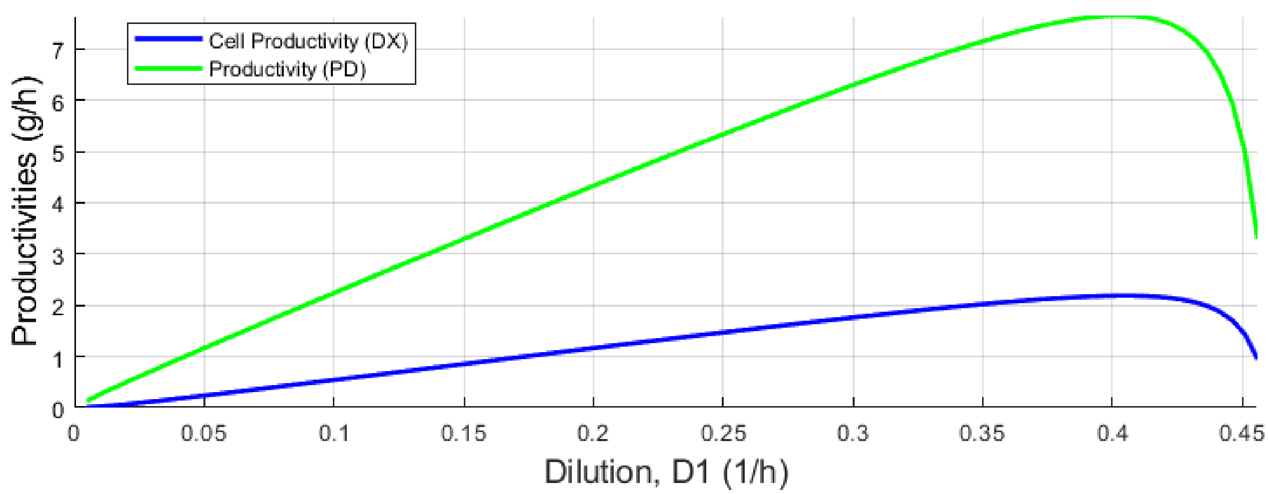
```
        end

        % keep track of maximum number of iterations
        if count ~= max_iter && count > max_count
            max_count = count;
        end

        % store good solutions (i.e., convergence achieved)
        if count ~= max_iter
            P2 = (F_k1 * P1_k1) / (V2*D2_k2) + (qp2 * X2 / D2_k2);
            [m.P2(k1,k2), m.mu2(k1,k2), m.S2(k1,k2), m.X2(k1,k2), ...
                m.count(k1,k2)] = deal(P2, mu2, S2, X2, count);
        end

    end  % ~ vessel #2 dilution domain
end      % ~ vessel #1 dilution domain

fprintf('\nMax Count: %3d\n', max_count)

% Remove invalid values (<0) for S2, X2, P2
inv_idx = (m.S2 < 0) | (m.X2 < 0) | (m.P2 < 0) ;
if any(inv_idx, "all")
    % warning('Invalid values found: %d indices set to NaN.', sum(inv_idx, 'all'));
    [m.X2(inv_idx), m.S2(inv_idx), m.P2(inv_idx)] = deal(NaN);
end

% Calculate Productivities
m.XD2 = m.X2 .* m.D2;
m.PD2 = m.P2 .* m.D2;

% Plot vessel 2 results
surf_conf = struct( ...
    'axis_1', "Dilution, D2 (g/L)", 'axis_2', "Dilution, D1 (g/L)", ...
    'subplot_labels', ["F_prime", "mu2", "S2", "count"], ...
    'colormap', jet, 'view', [0, 90] );

% first figure set (F_prime, mu2, S2, count)
plot_surf_wrap(m.D1, m.D2, m, surf_conf);

% Update surf_conf for the second set of plots
surf_conf.subplot_labels = ["X2", "P2", "XD2", "PD2"];

% second figure set (X2, P2, XD2, PD2)
plot_surf_wrap(m.D1, m.D2, m, surf_conf);
```

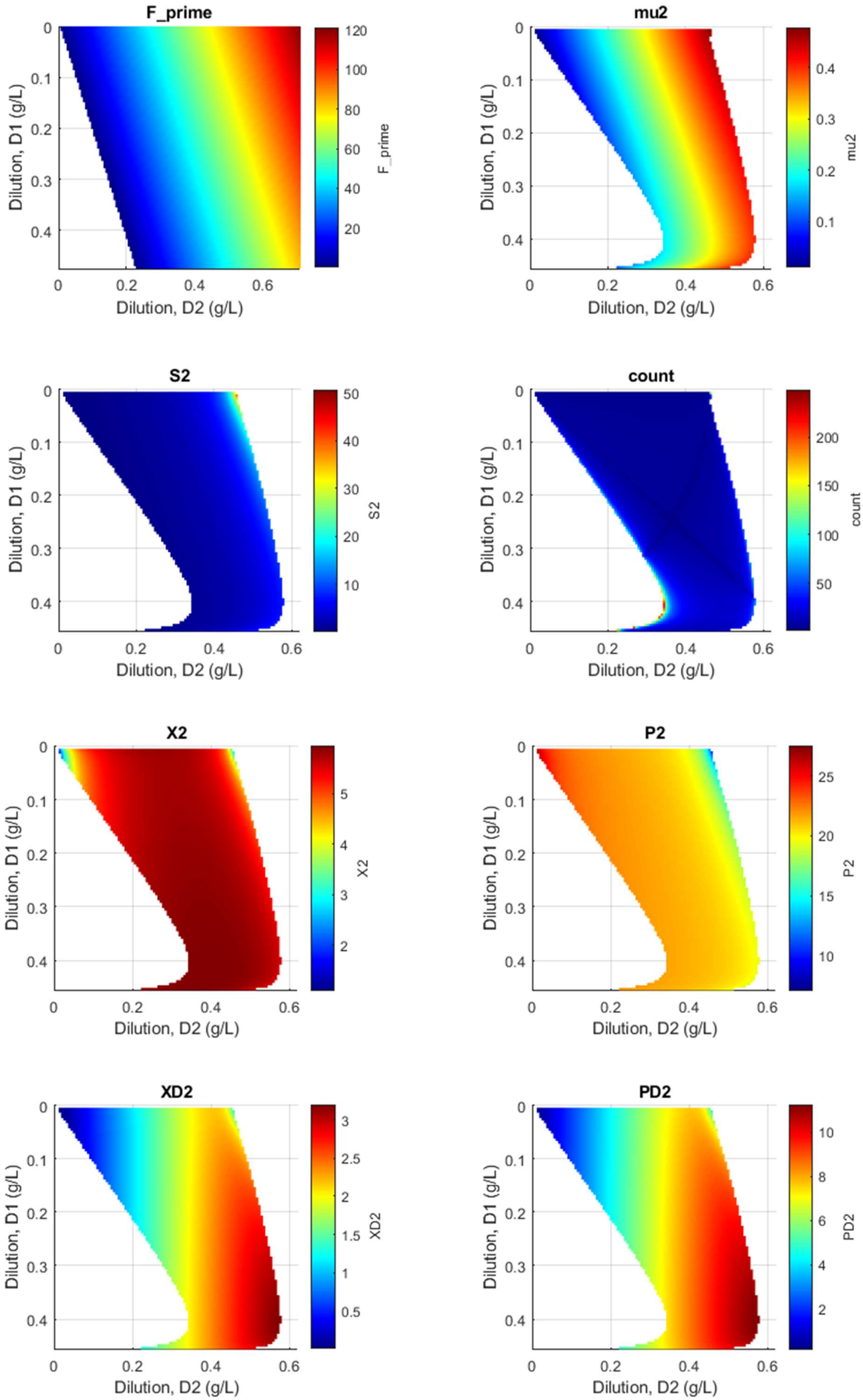
Vessel 1 Calculations

```
function [S, X, P, XD, PD] = vessel_1(D, kd, KS, mumax, SF, alpha, beta, YXS, YPS)

% Model equations for Cell and Product
mu   = D + kd;                     % growth rate
S    = (mu * KS) ./ (mumax - mu);  % substrate concentration
qp   = alpha * mu + beta;          % product formation rate

if any(qp == 0, "all")
    warning('Product formation rate, qp, equal to 0!')
end
```

```
% Cell and Product Concentrations
X_num = D .* (SF - S);             % cell concentration numerator
X_den = ((mu / YXS) + (qp / YPS)); % cell concentration denominator
X =  X_num ./ X_den;               % cell concentration
P      qp .* X ./ D;               % product concentration
XD     D .* X;                     % cell productivity
PD     D .* P;                     % product productivity

% Remove invalid values for S, X, P, XD, and PD
inv_idx = (X <= 0) | (S <= 0) | (P < 0);
if any(inv_idx, "all")
    % warning('vessel_1:inv_idx', '%d invalid indices found. Values set to NaN.', sum(inv_idx, 'all'));
    [S(inv_idx), X(inv_idx), P(inv_idx), XD(inv_idx), PD(inv_idx)] = deal(NaN);
end

end
```






F_prime
Dilution, D1 (g/L)
Dilution, D2 (g/L)
F_prime
mu2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
mu2
S2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
S2
count
Dilution, D1 (g/L)
Dilution, D2 (g/L)
count
X2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
X2
P2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
P2
XD2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
XD2
PD2
Dilution, D1 (g/L)
Dilution, D2 (g/L)
PD2