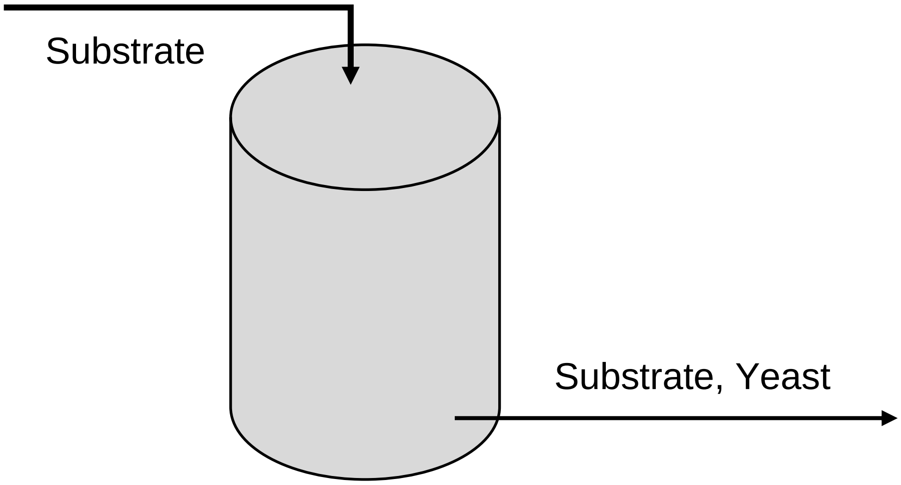
ABE 5570 | Prof. Okos

Fermenter Design Problem

Nathan LeRoy

Sept. 3rd, 2018

The System and Background



In this design problem, we are tasked with designing a fermenter that can produce 100 lbs of dry yeast product per hour. We are not given a volume. We are given growth equations for the cells and a substrate to cell yield:

$$\frac{dX}{dt} = uX, where\ u = \frac{u_{max}S}{k_s + S}$$

$$Y\ x/s = 0.5, thus\ \frac{dS}{dt} = -2\frac{dX}{dt} = -2uX, where\ u = \frac{u_{max}S}{k_s + S}$$

The inlet feed to our fermenter pumps in 0.2 g of substrate per gram of water. This is equivalent to 200 g of substrate per liter of water (ρ_w = 1000g/L). Thus, if we had a fermenter volume, we could calculate the total amount of substrate in the system initially. With this, and the substrate to yeast cell conversion ratio, we can calculate the final dry yeast yield. Knowing the mass of cells, we want to produce every hour, we can calculate the mass of substrate required initially in our system to achieve that. It should also be known that we want to use 95% of our initial substrate mass:

$$S_0(0.5)(0.95) = X_f$$

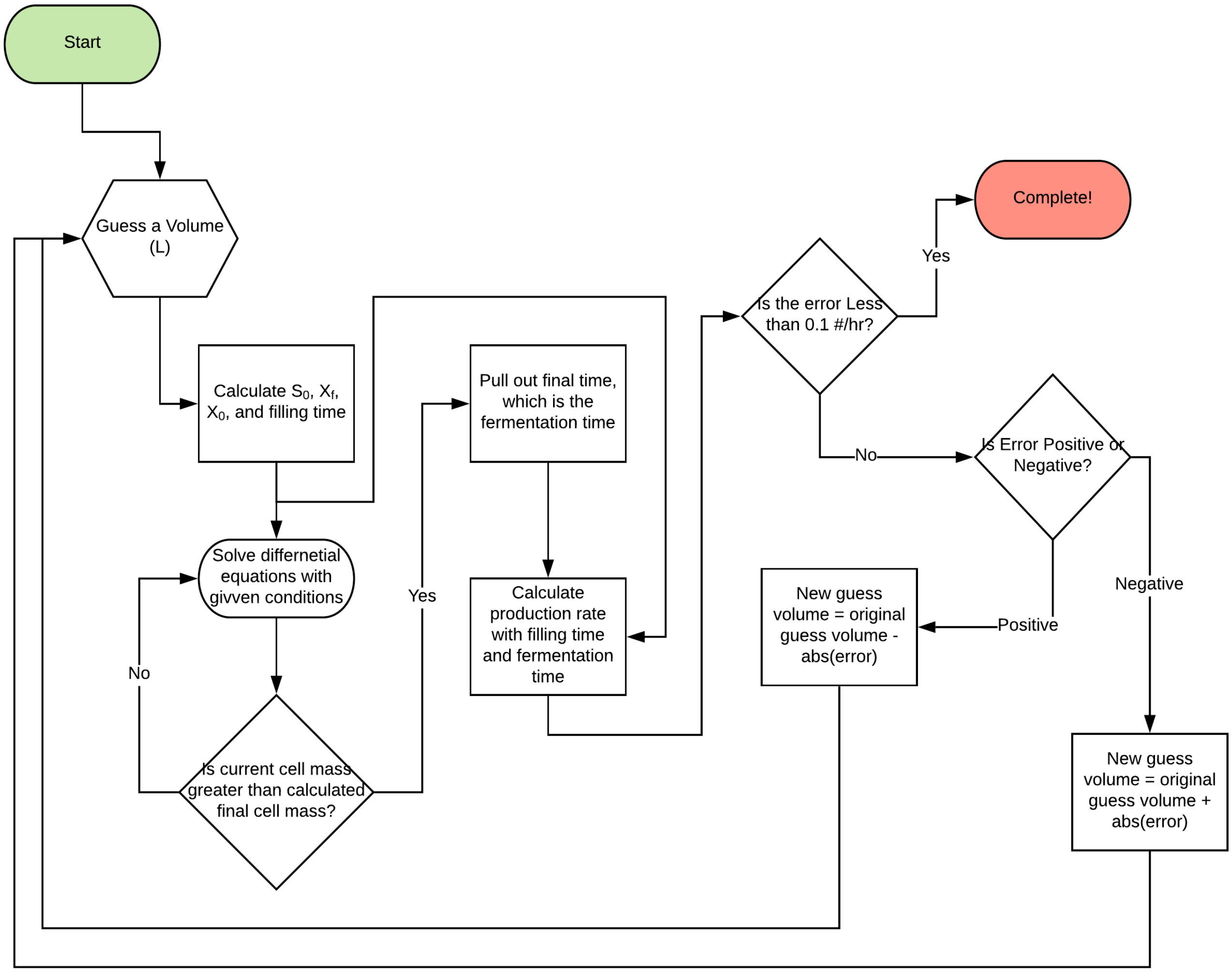
$$S_0 = \frac{100\ lbs/hr}{(0.5)(0.95)} = 210.52\ lbs/hr = 95{,}450\ g/hr$$

We are also given that the k_s value is 0.25 g/L_w, and our umax is 0.5 hr^{-1}. We **will assume** that the inflow rate into our fermenter is 2 L/s. This is equivalent to 7,200 L/hr. We will require this when calculating our filling time. So, if we know the volume, we can calculate the initial substrate mass, and then we can calculate the final cell mass. From there, we can calculate the initial cell mass (X_0 = 0.1*X_f). With these parameters, we can then solve the two given ODEs, and calculate the fermentation time to reach our final cell mass. Convert this value to pounds, divide by our filling and fermentation time and we can calculate the actual rate. If this does not match the assumed rate, then our volume guess was wrong, and we must go back to step one. The process is laid out below:

Algorithm Steps

1. Guess a Volume in liters
2. Calculate S_0, calculate X_f, calculate X_0, calculate the filling time
3. Solve differential equations using assumed initial conditions
4. Stop solver once final cell mass is reached and extract the fermentation time
5. Calculate rate with fermentation time and fill time
6. Compare to actual value, and calculate error
7. Re-create guess volume and go back to step (2).
8. Once rate is within pre-defined error limit, stop solver and note volume.

This can be represented pictorially as the following:


Start
Guess a Volume (L)
Calculate S_0, X_f, X_0, and filling time
Solve differnetial equations with givven conditions
Is current cell mass greater than calculated final cell mass?
No
Yes
Pull out final time, which is the fermentation time
Calculate production rate with filling time and fermentation time
Is the error Less than 0.1 #/hr?
Yes
Complete!
No
Is Error Positive or Negative?
Positive
New guess volume = original guess volume - abs(error)
Negative
New guess volume = original guess volume + abs(error)

Implementing the Process and Solving the ODE’s

The process on the previous page was carried out in MATLAB, and code was written to achieve it. The code requires an initial input volume guess from the user, and it uses a basic conditional block to converge onto a volume from there. The code will accept the initial guess input, calculate initial conditions, solve the ODEs using Eulers method, and based on the subsequent error, will re-guess a volume and try again. Eulers method is the most fundamental method to numerically solving ordinary differential equations. The basic formula is as follows:

$$Given\ a\ function, f(x,t)'s, derivative, \frac{df(x,t)}{dt} \dots$$

$$We\ can\ approximate\ the\ graph\ of\ the\ solution\ of\ the\ derivative\ with\ the\ formula:$$

$$f(x,t_{n+1}) = f(x,t_n) + h\frac{df(x,t)}{dt}, where\ h\ is\ an\ arbitrarily\ small\ step\ in\ time$$

This method was used to solve the ODE’s simultaneously, and they were continuously solved until the cell mass reached the desired final amount (which was calculated at the very beginning based on out assumed volume). A sample calculation of Euler’s method is given in the appendix. The code commented code is also given in the appendix.

The Results

When using a convergence algorithm, the entire MATLAB process takes about 25 seconds to implement. While somewhat slow, this is reasonable given that the program must repeatedly solve a system of differential equations using a relatively small step size.

The initial volume guess for our system was 2000 Litres. The program outputted the following results:

```
-=========================================================-
Volume: 2362.96 L
Fill Time: 0.33 hrs
Fermentation Time: 4.615 hrs
Calculated Rate: 99.90 #/hr
Error: -0.10 #/hr
Full Fill , Ferment, Empty Cycle Time: 9.89 hrs
Elapsed Implementation Time: 24.0789 sec in 125 iterations
-=========================================================-
```

We can see that the program converged to a volume of **2362.96 L**. This equates to a fill time of **0.33 hrs** or **~20 minutes**. The fermentation time was also found to be **4.615 hrs**. With these conditions, we can produce **99.90 #DS/hr** in our system. This is extremely close to the desired rate and is just within our error limit of **0.1 #DS/hr or of by 0.1 %.** A full cycle of filling, fermenting, and emptying will be **9.98 hrs**. This is reasonable for a facility to achieve. It took the solver exactly **24.08 seconds** and exactly **125 iterations** to converge to these data.

In addition to these data, we can plot the cell mass and substrate mass over time in kilograms.

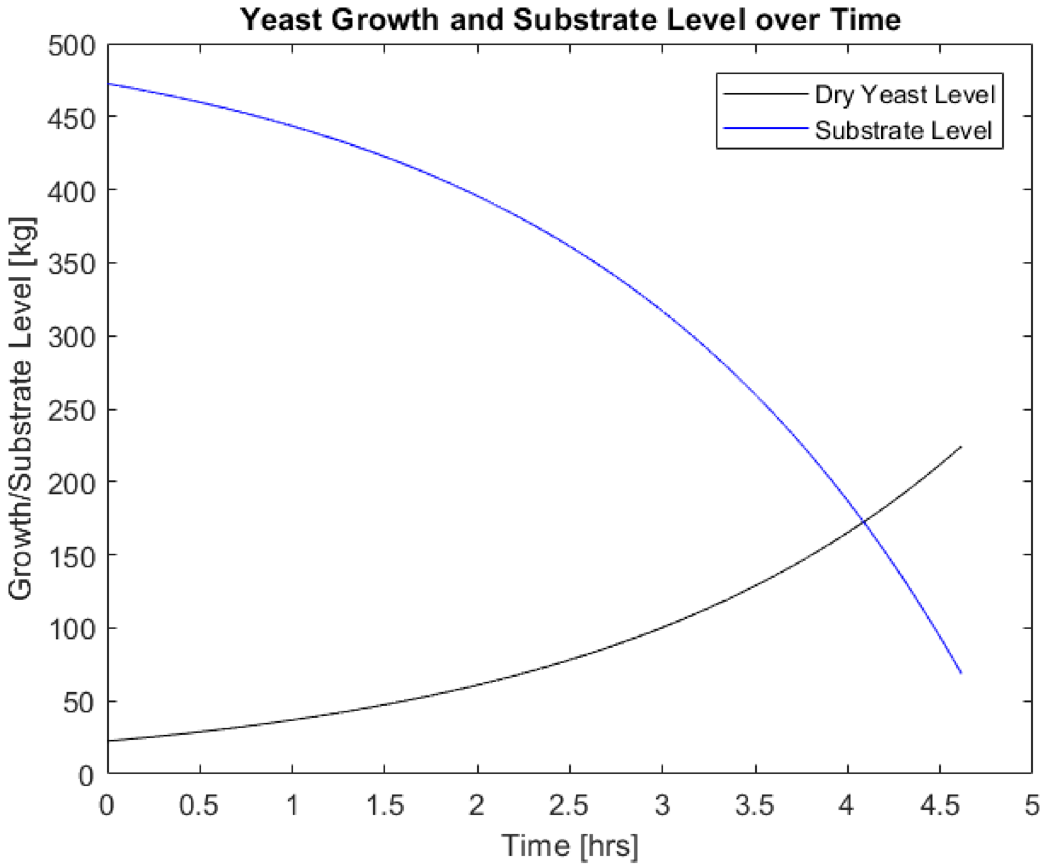


Figure 1. The cell mass and substrate mass over time in our fermenter in kilograms. As suspected, our cell mass increases over time, while the substrate mass correspondingly decreases over time.

Conclusions

- In conclusion, to produce 100 lbs of dry yeast product per hour, our fermenter will need to be 2362.6 L.
- With this and our assumed flow rate of 7,200 L/hr, our fill time will be 0.33 hrs, while our fermentation time will be 4.615 hrs.
- This allows us to produce 99.9 lbs of dry product per hour, which is only 0.1 % off of the target.
- Our program can converge to the proper volume with pretty much any guess, and all parameters can be tweaked to see how they change the outcome of our process.

Code

```
function main
tic
%clear commmand line and variable space
clc
clear all
q = 2 * 60 * 60; %L/hr
% GET THE GUESS VOLUME FROM USER %
guess_V = input('Enter a guess volume (L): ');
error = 100; % Assume very high error at begining to enter loop

% KEEP CHECKING THE ERROR UNTIL IT IS BELOW OUR DESIRED VALUE %
iterations = 0;
while abs(error) > 0.1 % we want to get close %
    iterations = iterations + 1;
    % CALCULATE CONDITIONS BASED OFF OF GUESS VOLUME %
    S0 = 200*guess_V; %grams
    X_f = S0*0.95*0.5; %grams
    X_0 = 0.1*X_f;
    fill_time = guess_V/q;

    % INITIALIZE MESHES %
    X_mesh = [];
    S_mesh = [];
    t_mesh = [];

    % POPULATE MESHES WITH INITIAL VALUES %
    X_mesh(1) = X_0;
    S_mesh(1) = S0;
    t_mesh(1) = 0;

    % START COUNTER AND DEFINE STEP SIZE %
    cntr = 1;
    h = 0.0001;

    % SOLVE DIFF EQs % WE ARE USING EULERS METHOD TO SOLVE EQUATIONS %
    while (X_mesh(cntr) < X_f) % Keep iterating until the maximum cell count is reached

         % EULERS METHOD X_n+1 = X_N + h*dXdt
         X_mesh(cntr + 1) = X_mesh(cntr) +
h*dXdt(X_mesh(cntr),S_mesh(cntr),t_mesh(cntr),guess_V);
         S_mesh(cntr + 1) = S_mesh(cntr) +
h*dSdt(X_mesh(cntr),S_mesh(cntr),t_mesh(cntr),guess_V);
         t_mesh(cntr + 1) = t_mesh(cntr) + h;
         cntr = cntr + 1;

    end

    %Extract fermentation time
    ferment_time = t_mesh(cntr); % Last time in the mesh

    %Calculate the rate based on the fill and ferment times
    calc_rate = X_f/(fill_time + ferment_time); %g/hr
    calc_rate = calc_rate*0.0022; %pounds/hr

    % Get the amount of error. 100 pounds per hour is desired.
    error = calc_rate - 100;
    % Gain variable for use in convergence algorithm
    kp = 1;

    % Converge towards a volume that creates a smaller error %
    if error < 0 % larger volume required
        guess_V = guess_V + abs(error)*kp;
    end

    if error > 0 % smaller volume required
        guess_V = guess_V - abs(error)*kp;
    end

end

%Plot the data
figure(1)
plot(t_mesh,X_mesh./1000,'-k');
hold on
plot(t_mesh,S_mesh./1000,'-b');
title('Yeast Growth and Substrate Level over Time')
xlabel('Time [hrs]');
ylabel('Growth/Substrate Level [kg]')
legend('Dry Yeast Level','Substrate Level')

time = toc
% OUTPUT RESULTS %
fprintf('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-\n');
fprintf('Volume: %0.2f L\n',guess_V);
fprintf('Fill Time: %0.2f hrs\n',fill_time);
fprintf('Fermentation Time: %0.3f hrs \n',ferment_time);
fprintf('Calculated Rate: %0.2f #/hr\n',calc_rate);
fprintf('Error: %0.2f #/hr\n',error);
fprintf('Full Fill , Ferment, Empy Cycle Time: %0.2f hrs\n', (fill_time + ferment_time)*2);
fprintf('Elapsed Implementation Time: %0.4f sec in %d iterations\n',time,iterations);
fprintf('-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-\n');
% DEFINE DERIVATIVES %

function X_slope = dXdt(X,S,t,guess_V)
% DEFINE CONSTANTS %
ks = 0.25 * guess_V; %g/L
umax = 0.5; %1/hr
Yx_s = 0.5; %x/s

u = (umax*S)/(ks + S);

X_slope = u*X;

end

function S_slope = dSdt(X,S,t,guess_V)
% DEFINE CONSTANTS %
ks = 0.25*guess_V; %g/L
umax = 0.5; %1/hr
Yx_s = 0.5; %x/s

u = (umax*S)/(ks + S);

S_slope = -1*2*u*X;
end
end
```