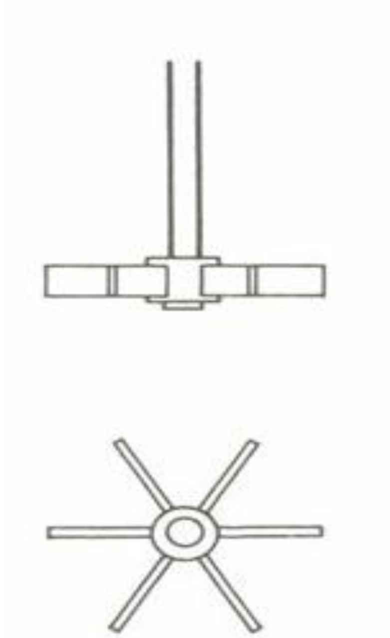
Fermenter Design Problem Part B

Nathan LeRoy | 9/11/2018 | ABE 557

Introduction

Once we know how large of a fermenter is required, we can begin to optimize and calculate fewer fundamental parameters of our system like **the power and type of agitator**, the **size and flow of heat exchanger**, **amount of air flow**, and **the size and power requirement of blower**. There are many factors that go into all these calculations. Some are defined by the thermodynamics of the system, or the biological profile, while others need to be chosen/assumed to facilitate and support a well optimized process.

Power and Type of Agitator



The team has decided to use a flat six blade open turbine (left). We chose this type of impeller because it fosters powerful mixing with lower rotational speeds. This is desirable to prevent shearing of our cells. One tradeoff is high impeller Reynolds numbers which forces high energy consumption.

To calculate this power consumption, we can use convenient charts like figure 3.4-5 from Geankopolis. For any impeller Reynolds number, N'_{Re}, we can find the power number, N_p, and subsequently c+alculate the power requirement:

$$N'_{Re} = \frac{D_a^2 N \rho}{\mu}$$

$$N_P = \frac{P}{\rho N^3 D_a^3}$$

The two parameters we must find are the rotational speed, N, and the agitator diameter, D_a. To find both, we must also calculate the tank diameter, D_t. We know that our tank height is twice the tank diameter, so we can develop an equation for tank diameter as follows. Note, our tank volume was calculated in the previous algorithm to be precisely 2362 L, and we will use this to calculate our tank diameter.

$$V = \pi R^2 h$$

$$V = \pi \left(\frac{D_t}{2}\right)^2 h$$

$$V = \pi \left(\frac{D_t}{2}\right)^2 2D_t$$

$$V = \frac{\pi D_t^3}{2}$$

$$\sqrt[3]{\frac{2V}{\pi}} = D_t$$

We can assume that our agitator or impeller diameter is half of our tank diameter. With this, our only required variable is the rotational speed. This is tightly coupled to the shear rate of our system. We must ensure care to not damage our cells, so this number must be calculated with cell-preservation in mind. It is known that cells can only withstand about 2,500 Pa of shear stress (SOURCE). This will be the starting point for calculating shear rate and subsequent rotational speed. Since our system is being assumed to have thermo-physical properties like water, we can assume Newtonian physics area at play. Thus, our shear stress and shear rate can be related with Newton's law of viscosity:

$$\tau = \mu_w \dot{\gamma}$$

Our shear rate can also be related to the rotational speed. The shear rate, for our system, will be defined as the linear decline in velocity from the impeller tip to the wall. This is the no-slip principle. Mathematically, this is written as:

$$\frac{v_{tip} - v_{wall}}{\Delta R} = \frac{v_{tip} - 0}{\Delta R} = \frac{v_{tip}}{R_t - R_i} = \frac{v_{tip}}{D_t/4}$$

The velocity of the tip is given through:

$$v_{tip} = \pi D_i N = \pi(0.5) D_t N$$

Plug this into our equation, and we can simplify to obtain a very simply formula that relates shear rate and impeller rotational speed:

$$\dot{\gamma} = 2\pi N$$

Finally, we can plug in our shear rate into Newtons law of viscosity and obtain an expression for N:

$$N = \frac{\tau}{2\pi\mu_w}$$

Using MATLAB to conduct our calculations, we can find that the rotational speed to obtain 2,500 Pa shear stress is ~400,000 RPS. This is obviously an absurdly fast speed. What does this mean for our system? It means that we can safely choose a rotational speed that fits our system without worrying about shearing our cells. The group feels that given our tank diameter, we will choose a rotational speed of about 1.5 – 2.0 rotations per second.

With this, we can now find our Power Consumption to be: **1.18 kW** (See MATLAB code for calculations).

Size and Flow of Heat Exchanger

The heat exchanger is required to pull out heat that is generated in our system as a result of cell respiration and/or fermentation. Our system will be operating at steady state, and we can use equations that describe such phenomena. First, we must identify the amount of heat being generated by the cells in our system. This can be approximated as such:

$$\dot{q}_{cells} \approx 0.12\, q_{O_2} X$$

Where X is the cell mass in the system, and q_{O2} is the oxygen uptake requirement of our cells. X can be calculated from the previous algorithm. With an initial substrate level, and known conversion/yield rates, X is found to be precisely:

$$X = 0.95 \times 0.5 \times S_0 \times V$$

Subsequently, we can find that our cells produce about 215,000 kcal/hr. Please see the MATLAB calculations for specifics. At steady state, we know that the energy generation from our cells is exactly the energy absorbed in our water-pip heat exchanger:

$$\dot{q}_{cells} = \dot{q}_{water}$$

$$\dot{q}_{cells} = \dot{m} c_p \Delta T, or$$

$$\dot{m} = \frac{\dot{q}_{cells}}{c_p \Delta T}$$

With a known mass flow rate, we can assume an exchanger diameter, and calculate the Reynolds number for the fluid flow within our pipes. This allows us to subsequently calculate Nusselt numbers, Prandtl numbers, and heat transfer coefficients:

$$Re = \frac{\rho v D}{\mu} = \frac{4\dot{m}}{\mu \pi D}$$

$$Pr = \frac{c_p \mu}{k_f}$$

$$Nu = 0.027 Re^{0.8} Pr^{1/3}, for\ laminar\ flow$$

$$Nu = \frac{hD}{k_f}$$

With all these dimensionless numbers, we can calculate the convective heat transfer coefficient inside our heat exchanger. It should be stated that the team assumed a negligible inner-tank convective heat transfer, and a negligible conductive heat transfer within the walls of our heat exchanger pipe.

With this information, we can now calculate the length of our heat exchanger. Again, we know that the heat generation from cells is precisely the heat absorption from our water. We can describe this mathematically as such:

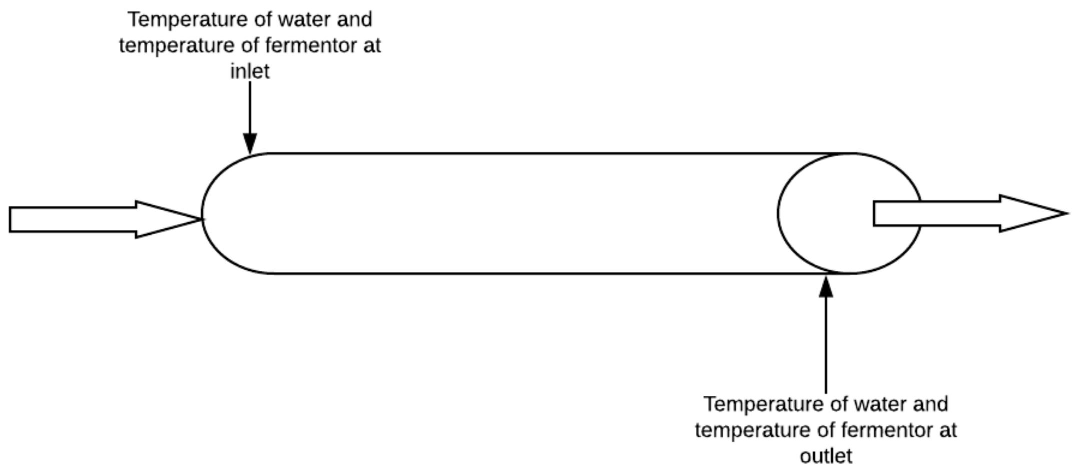
$$\dot{q}_{cells} = UA\Delta T_{lm}$$

$$\dot{q}_{cells} = U(\pi DL)\Delta T_{lm}$$

ΔT_{lm} is the log-mean temperature distribution in our heat exchanger pipe. It is defined as:

$$\Delta T_{lm} = \frac{\Delta T_{in} - \Delta T_{out}}{ln\, {}^{\Delta T_{in}}/_{\Delta T_{out}}}$$

The temperature differences at the inlet and outlet is the temperature difference between the heat exchanger water, and the system. See the diagram below:



We can rearrange the equations to obtain an expression for the length of our exchanger as such:

$$L = \frac{\dot{q}_{cells}}{\pi D U \Delta T_{lm}}$$

MATLAB was used for the calculations.

Amount of Airflow, Type of Blower, and Power Requirement

Airflow is essential for the survival of our cells in the system. In a large tank such as ours, we cannot solely rely on oxygen diffusion from the atmosphere to sufficiently supply our cells with the required amount of oxygen. We must sparge oxygen through the fermenter to adequately oxygenate out system. We are going to assume our fermenter is operating at steady state. For this, the oxygen transfer rate (OTR) into the system is equivalent to the oxygen uptake rate (OUR) of our cells. Mathematically, this is described as:

$$OUR = OTR$$

$$k_L a(C^* - C_L) = X \cdot q_{O_2}$$

$K_L a$ is the mazz transfer rate of oxygen, C^*, is the solubility of oxygen in the tank, C_L is the critical oxygen level in our tank, X is our cell mass, and q_{O2} is the cells' oxygen uptake requirement. We can obtain C^* from Henry's law:

$$P_{O_2} = H_{O_2} X_{O_2}$$

P is the partial pressure of oxygen in the gas, H is henry's law constant for oxygen, and X is the molar fraction of oxygen in our system. The group feels it would be best to sparge atmospheric air into our fermenter as opposed to pure oxygen. This is not only cheaper but will foster less heat generation. Using Henry's law and basic unit conversions, we can deduce that the solubility of oxygen in our fermenter is just under 8 mg/L. K_La can be calculated by solving the steady state oxygen transfer equation for.

We know the oxygen uptake requirement of hour cells. By normalizing for cell mass, we can obtain an equation that gives us the oxygen mass requirement per time in our fermenter. Finally, using the ideal gas law, we can find our volumetric flow rate of air into our fermenter.

$$\dot{\eta}_{O2} = q_{O2}$$

$$\dot{\eta}_{air} = \frac{\dot{\eta}_{O2}}{y_{O2}}$$

$$Q = \frac{\dot{\eta}_{air} RT}{P}$$

With a known volumetric flow rate, we can subsequently calculate an airflow velocity with an assumed cross-sectional area and find the pressure differential using Bernoulli's equation.

$$v_{air}^2 = \frac{P_2 - P_1}{\rho_{avg}}$$

Bernoulli's equation assumes incompressible flow, which we know for a gas is an extremely poor assumption. In this case, we will use the average densities between both pressures:

$$\rho_{avg} = \frac{\rho_1 + \rho_2}{2}$$

We can also relate both densities to each other and their respective pressures using the ideal gas law. This gives rise to the equation:

$$\frac{P_1}{\rho_1} = \frac{P_2}{\rho_2}$$

Use this relation and substitute into the Bernoulli equation, we now have an expression for the velocity as a function of the two velocities and final density:

$$v_{air} = \sqrt{\frac{4(P_2 - P_1)}{\rho_2 + \frac{P_1}{P_2}\rho_1}}$$

We can rearrange this equation to get an expression for the initial pressure of our gas:

$$P_1 = \frac{4P_2 - v_{air}^2 \rho_2}{\frac{v_{air}^2 \rho_2}{P_2} - 4}$$

With this new calculated pressure, we have the ability to calculate the total power required of our turboblower. This comes through shaft work:

$$-W_s = \frac{\gamma}{\gamma - 1}\frac{RT}{M}\left[\left(\frac{P_2}{P_1}\right)^{(\gamma-1)/\gamma} - 1\right]$$

Where R is the universal gas constant, T is temperature, M is the molecular weight, and gamma is a gas-specific value. For air, this value is 1.41 (Geankopolis, 2006). From here, we can finally calculate the pump power with the following equation. We will assume 80% efficiency in our pump:

$$Power(kW) = \frac{-W_s m}{\eta \cdot 1000}$$

Outputs

The final MATLAB output of our algorithm is:

```
Enter a guess volume (L): 2353
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
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, Empy Cycle Time: 9.89 hrs
Elapsed Implementation Time: 8.1527 sec in 41 iterations
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Power Consumption by Impeller: 1481.22 Watts
Heat Exchanger Length: 560.14 meters
Solubility of Oxygen: 7.86 mg/L
KLa: 11840278.79 hr^-1
Flow Rate of Air: 212.47 m^3/hr
Air Velocity: 187.87 m/s
Pressure Differential: 22.86 kpa
Shaft Work: 17180.51 J/kg
Blower Power Consumption: 1.48 kW
```

MATLAB 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');

% FERMENTER PART B %
% POWER CONSUMPTION %
D_t = (2*(guess_V/1000)/pi) ^ (1/3); % m
visc_w = 0.001; %pa.s
shear_stress = 2500; %pa
N = 2.0; %RPS
RPM = N*60; %RPM
D_i = 0.5*D_t; %m
rho_w = 1000; %kg/m^3
Re_im = (D_i^2*N*rho_w)/visc_w; %unitless
Power_im = 3; %from figure 3.4-5 geankopolis
Power = Power_im*rho_w*N^3*D_i^5;

% EXCHANGER FLOW AND SIZE %
X = 200*0.95*0.5*guess_V; %g dry weight
D_ex = 0.05; %meters (5 cm)
q_O2 = 8 * 32.02; %mg O2/g
q_cells = 0.12*q_O2*X; %kcal/hr
Cp_w = 1; %kcal/hjr
delT = 10; %K
m_dot = q_cells/(Cp_w*delT); %kg/hr
```

```
m_dot_s = m_dot/60/60; %kg/s
k_w = 0.6; %W/m-K
Re = (4*m_dot_s)/(pi*visc_w*D_ex); %Unitless
Pr = Cp_w*visc_w/k_w; %unitless
Nu = 0.027*(Re^0.8)*(Pr^(1/3)); %Unitless
h = Nu*k_w/D_ex; %W/m^2-K
T_in_f = 30; %C
T_in_w = 15; %C
T_out_f = 30; %C
T_out_w = 25; %C
delT_in = T_in_f - T_in_w; %C
delT_out = T_out_f - T_out_w; %C
delT_lm = (delT_in - delT_out)/(log(delT_in/delT_out)); %C
length = q_cells/(pi*D_ex*h*delT_lm); %m
p_O2 = 0.21; %atm
H_O2 = 4.75E4; %atm/mol
X_O2 = p_O2/H_O2; %unitless



% BLOWER RATE %
C_star = guess_V*(1/1)*1000*(1/18.018)*X_O2*32.02*1000*(1/2363); %mg/L
CL = 3; %mg/L
OUR = q_O2*X; %Uptake

T = 30; %C
RPM = N*60; %RPM
kla = OUR/(C_star-CL); %1/hr
R_gas = 0.082; %L-atm/mol-K
q_O2_mmol = 8; %mmol/hr-g
q_O2_mol = q_O2_mmol/1000; %mol/hr-g
p_O2 = 0.21;
molar_flow_rate_air = q_O2_mol*X/p_O2; %mol/hr
Q_air = molar_flow_rate_air*R_gas*(303)/1; %L/hr
Q_air_m3 = Q_air/1000; %m^3/hr
Q_air_m3_s = Q_air_m3/60/60; %m^3/s
d_turb = 0.02; %m (5 cm)
C_turb = pi*(d_turb/2)^2; %m^2
v_air = Q_air_m3/C_turb; %m/hr
v_air_s = v_air/60/60; %m/s
P_out = 101300; %Pa
rho_air = 1.164; %kg/m3
P_in = ((v_air_s^2)*rho_air+4*P_out)/(4-(v_air_s^2)*(rho_air/P_out)); %Pa
P_in_kpa = P_in/1000; %kpa
delP = P_in - P_out;
delP_kpa = delP/1000; %kpa
gamma = 1.41; %unitless
R = 8314; %J/mol-K
T = 303; %K
M = 28.97; %g/mol
effic = 0.8;
m_dot_air = Q_air_m3_s*rho_air; %kg/s

W_s = -1*(gamma/(gamma-1))*(R*T/M)*(((P_out/P_in)^((gamma-1)/gamma))-1); %J
Pow_blower = (W_s*m_dot_air)/(effic*1000); %kW

% OUTPUT %
fprintf('');
fprintf('Power Consumption by Impeller: %0.2f Watts\n', Power);
fprintf('Heat Exchanger Length: %0.2f meters\n', length);
fprintf('Solubility of Oxygen: %0.2f mg/L\n', C_star);
fprintf('KLa: %0.2f hr^-1\n',kla);
```

```matlab
fprintf('Flow Rate of Air: %0.2f m^3/hr\n',Q_air_m3);
fprintf('Air Velocity: %0.2f m/s\n',v_air_s);
fprintf('Pressure Differential: %0.2f kpa\n',delP_kpa);
fprintf('Shaft Work: %0.2f J/kg\n',W_s);
fprintf('Blower Power Consumption: %0.2f kW\n',Pow_blower);
% 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
```