

```

0001 clear()
0002 clearglobal()
0003
0004 //--- Input and defs -----
0005
0006 thispath = "C:\Users\Henry\Desktop\quasi-steady-sim-yogurt\";
0007
0008 exec(thispath+'func-thermo.sce', -1) // Thermo funcs and consts
0009 exec(thispath+'input_vars.sce', -1) // Input data (Will change)
0010 //exec(thispath+'func-simul.sce', -1) // Put all your functions here!!!!
0011
0012 // Initial Values
0013 T = [308.16176 308.16176 316.15001 320.61251 320.2396 308.16176];
0014
0015 // Air Temperatures through Rock Bed
0016 Tr = [T_b0-15 T_b0-10 T_b0-8 T_b0-4];
0017
0018 // Air Temperatures through Yogurt
0019 Ts = [T_y0 T_y0-5 T_y0-7 T_y0-10];
0020
0021 //Yogurt stacks averages temperatures
0022 Ty = [293.15+10 293.15+12 293.15+15];
0023
0024 // Change in Yogurt stacks temperatures
0025 dTy = [ 0 0 0 ];
0026
0027 // rockbed average temperature
0028 Tb = [326.15 326.15-5 326.15-10];
0029
0030 // Change in rockbed temperatures
0031 dTb = [ 0 0 0 ];
0032
0033 // Pressure drop across Rockbed
0034 Pd = 0;
0035
0036 // Pressure drop across yogurt stack
0037 Py = 0;
0038
0039 // number of iteration
0040 num_iteration = 10
0041
0042 // Volume flow rate;
0043 Qr = 3.1734; // 2.647
0044 rpm_set = 900
0045 Qr_ = Qr/n_fan
0046 Q_star = Qr_*rpm_ref/rpm_set
0047 Ab = W_b*H_b;
0048
0049 pressure_pump = -13.649*Q_star^2+73.946*Q_star+1055.2
0050 pressure_drop = 0;
0051 for k=1:num_iteration
0052     // FAN
0053     T(2) = T(1); // no temperature change accross fan
0054     Tr(1) = T(2); //
0055     Pd = 0; // Pressure drop accross Rockbed
0056     Py = 0; // Pressure drop accross yogurt
0057     for j = 1:n_b,
0058         T_avg = 0.5*( Tr(j)+Tr(j+1) ); // average
0059         temperature rho_avg = rho_air( T_avg) ; // density of air
0060         mu_avg = mu_air( T_avg ) ; // dynamic
0061         viscosity of air

```

```

0061         cp_avg = cp_air( T_avg ); // specific heat
        capacity of air
0062         k_avg = k_air( T_avg ); // thermal
        conductivity of air
0063         G_avg = rho_avg*(Qr/Ab); // mass flow rate
        per unit area
0064         Re_avg = G_avg*d_p/(mu_avg); // Reynolds Num
0065
        f_avg = 4.466*((Re_avg)^0.2)*((psi)^0.696)*((eps_b)^-2.945)*exp( 11.85*( log(psi) )^2 ); //
        friction factor
0066
        Nu_avg = 0.437*((Re_avg)^0.75)*((psi)^3.35)*((eps_b)^-1.620)*exp( 29.03*( log(psi) )^2 ); //
        Nusselt Number
0067         h_avg = (k_avg*Nu_avg)/(d_p^2); // Convection
        coefficient
0068         Tr(j+1) = Tb(j) - ( Tb(j)-Tr(j) )*exp( (-h_avg*( L_b/n_b ) )/( cp_avg*G_avg ) );
0069
0070         // change in temperature in rockbed
0071
        dTb(j) = (cp_avg*G_avg*(Tr(j)-Tr(j+1)))/(2640*820*(1-Tr(j+1))*(L_b/n_b))
0072         // Pressure drop across control volume
0073         Pd = -f_avg*G_avg^2*(L_b/n_b)/(rho_avg*d_p) + Pd;
0074         disp(Pd)
0075
0076     end
0077     Pd = eta_r*Pd // Pressure loss accross
        rock bed
0078     T(3) = Tr(n_b+1);
0079
0080     // Heater
0081     T_avg = (T(3)+T(4))/2; // Average Temperature
0082     cp_avg = cp_air(T_avg); // Heat capacity air
0083     rho_avg = rho_air(T_avg); // density of air
0084     q_h = KP*(T_set-Ts(1))+KI*0 // heat added
0085     T(4) = T(3) + q_h/(cp_avg*rho_avg*Qr) // temperature change
0086
0087     // Environemental Losses
0088     T_avg = (T(4)+T(5))/2; // average temperature
0089     cp_avg = cp_air(T_avg); // Heat capacity air
0090     rho_avg = rho_air(T_avg); // density of air
0091     q = rho_avg*Qr*cp_avg*(T(4)-T(5));
0092
0093     T(5) = T_env + (T(4)-T_env)/(exp(h_env*A_env/(rho_avg*Qr*cp_avg)));
0094     Ts(1) = T(5);
0095
0096     // Yogurt Warming
0097     for i = 1:n_s,
0098         T_avg = 0.5*(Ts(i)+Ts(i+1)); // Average Temperature
0099         rho_avg = rho_air(T_avg); // density of air
0100         mu_avg = mu_air(T_avg); // dynamic viscosity of air
0101         cp_avg = cp_air(T_avg); // Heat capacity air
0102         k_avg = k_air(T_avg); // thermal conductivity of
        air
0103         Pr_avg = Pr_air(T_avg); // Prantel Number of Air
0104         Pr_avg_e = Pr_air(Ts(i+1)); // Prantel Number of air for
        next control volume
0105         nu_avg = nu_air(T_avg); // Nusselt Number of air
0106         As = N_Ts*S_Ts*H_y; // Effective Area
0107         mu_s = Qr/As;
0108         Re_avg = mu_s*D_y/(nu_avg); // Reynolds
        Numbers

```

```

0109
0110         if Re_avg > 10e3 || Re_avg < 2*10e5 then           // Determine
correct Nusselt number equation based on Re
0111             Nu_d = 0.72*F_s*Re_avg^0.63*Pr_avg^0.365*(Pr_avg/Pr_avg_e)^0.25;
// Nusselt Number
0112         elseif Re_avg > 2*10e5 || Re_avg < 2*10e6 then
0113             Nu_d = 0.033*F_s*Re_avg^0.8*Pr_avg^0.4*(Pr_avg/Pr_avg_e)^0.25;
// Nusselt Number
0114         end,
0115         m_y = (%pi)/4*rho_avg*(D_y)^2*H_y*N_Ls*N_Ts;
0116         h_avg = Nu_d*k_avg/(D_y);
0117         A_y = %pi*D_y*N_Ls*N_Ts;
0118
0119         Ts(i+1) = Ty(i) - (Ty(i) - Ts(i))*exp( -1*(A_y*h_avg)/(rho_avg*Qr*cp_avg) );
0120
0121         // change in yogurt stack temperatures
0122         dTy(i) = (rho_avg*Qr)/(3520*m_y)*(Ts(i)-Ts(i+1))
0123         // Pressure drop
0124         // Fix friction factor
0125         Py = Py + -N_Ls*0.3*(rho_avg*mu_s^2)/2;
0126     end
0127     Py = eta_s*Py;
0128     T(6) = Ts(n_s+1);
0129     if abs( (T(1)-T(6))/(T(6))) < 1e-8 then
0130         k = num_iteration;
0131     else
0132         T(1) = T(6);           // closed system
0133     end
0134
0135     pressure_drop = Py+Pd;
0136 end
0137
0138 disp('k')
0139 disp(k);
0140 disp('T')
0141 disp(T);
0142 disp('Pressure pump')
0143 disp(pressure_pump)
0144 disp('pressure drop')
0145 disp(pressure_drop)
0146 disp('Pressure pump - pressure drop')
0147 disp(pressure_pump + pressure_drop)
0148
0149 disp('change in rockbed temperature')
0150 disp(dTb)
0151 disp('change in yogurt temperature')
0152 disp(dTy)

```