Fluid Mechanics
EGME333
December 6, 2022
Design project

SINGLE PUMP SIZING AND PIPE SYSTEM OPTIMIZATION

Nathan Delos Santos and Carlos Pena
Department of Mechanical Engineering
California State University, Fullerton
Fullerton, CA, USA

PROBLEM DESCRIPTION

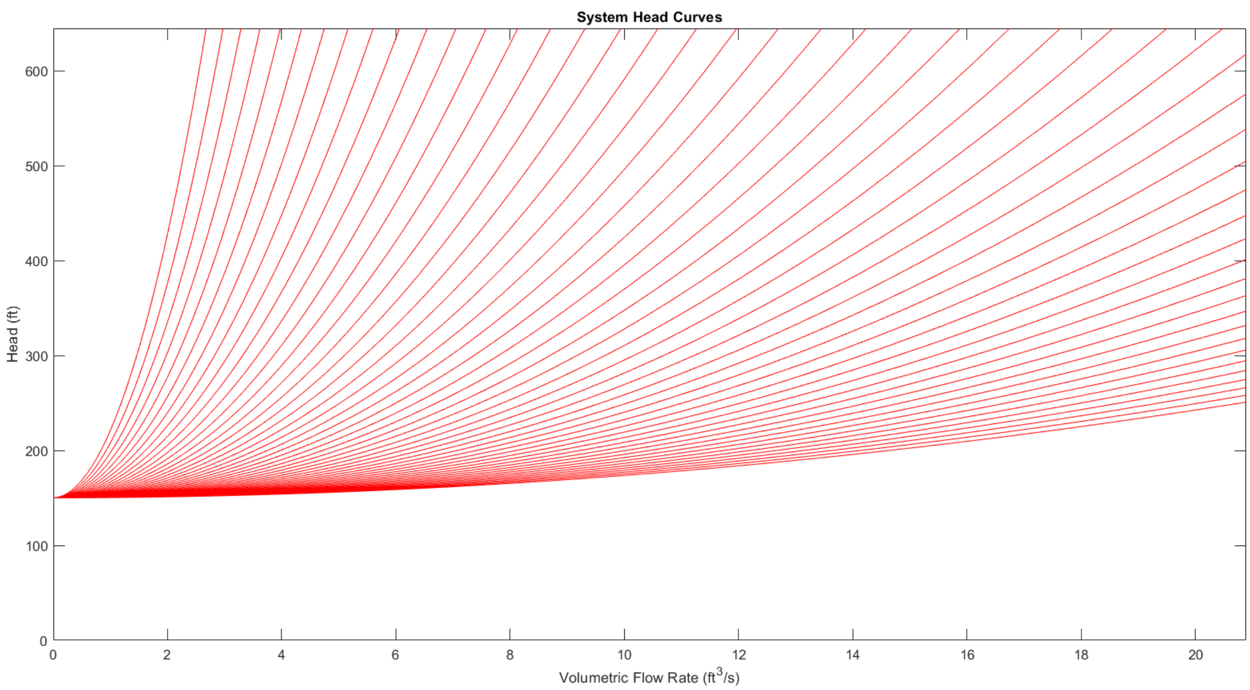
Our client requires a 1.5 Million gallon tank of water to be constantly filled from an aquifer 0.8 miles away, and 150 ft below the tank. To do this, there must be a pump to move the water, however, this pump must not run for more than 8 hours a day. Our client requires that this design last for 20 years. The design's parameters include the pipe's diameter and length, impeller diameter, operational speed, flow rate, pump head, pump power, valve head loss, and elbow head loss. We were also constrained to run the pump between 900 and 1800 rpm. Our selection of the components of the entire system was based on optimal performance, maximum efficiency, and best cost to get the job done. Our client also requires that the system have 4 butterfly valves, must have 10 90 degree elbows.

PUMP-PIPE SIZING

Before even sizing the pump, I needed to find the system the pump would have to make water flow through. To do this, I took the shortest length of pipe from the aquifer to the tank possible. I just used the pythagorean theorem with the height between the tank and the aquifer, and the distance between them. I then included the valves and the elbows. Unlike what Bernoulli's equation would have you believe, the flow down a straight pipe might not be the same, even if they have all the same dimensions. This is because of losses due to viscosity, known as head loss. Head loss is a way to put into perspective how much pressure a system loses, by comparing it with hydrostatic pressure - How much pressure can a column of liquid generate? How tall is that column? As flow rate increases through a pipe, the losses incurred increase. The losses that result from this resemble a curve, and are known as major loss. The losses are a function of the pipe diameter, and the flowrate, meaning at a specific diameter, and as you walk along the axis Q, flow rate, there will be a corresponding head loss. This amounts to taking the head loss at every single point for a given diameter, and then commencing each point with a curve. Each head loss is calculated with what is known as a friction factor, and relative roughness. The relative roughness is calculated with the surface roughness of the material divided by the hydraulic diameter of the pipe. Calculating the friction factor however, is much more difficult.

Because the function for friction factor is implicit, it must be solved with a computer. To do this, I had a function in my program that would iterate and guess a value for the friction factor until the equation became true, and then plugged that into the equation for major loss.

$$\frac{1}{\sqrt{F_{fric}}} = -\ 2 \cdot log(\frac{\epsilon / D_{Pipe}}{3.7} + \frac{2.51}{Re\sqrt{F_{fric}}})$$



From the graph and equation, it can be seen that a smaller diameter pipe produces higher head loss as flow rate increases compared to pipes of larger diameter. Because head loss is measured in distance, it can be added to the height between the tank and the aquifer. This can be interpreted as the viscosity adding an extra distance the fluid must climb to so that it can reach the tank.

In addition to the losses incurred from the pipes, there were losses from the 10 90 degree elbows and 4 butterfly valves that were required on the system. These are known as minor losses. Each component had a coefficient that determined its loss, and they were given to us by a graph.

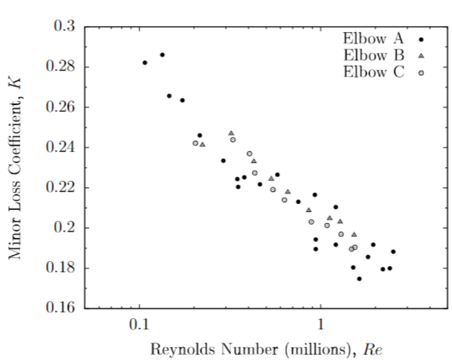


Figure 1: Measured loss K coefficients for 90° elbows.

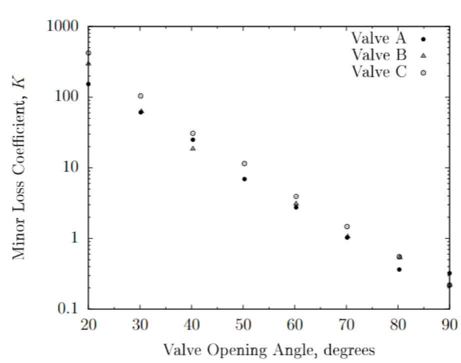
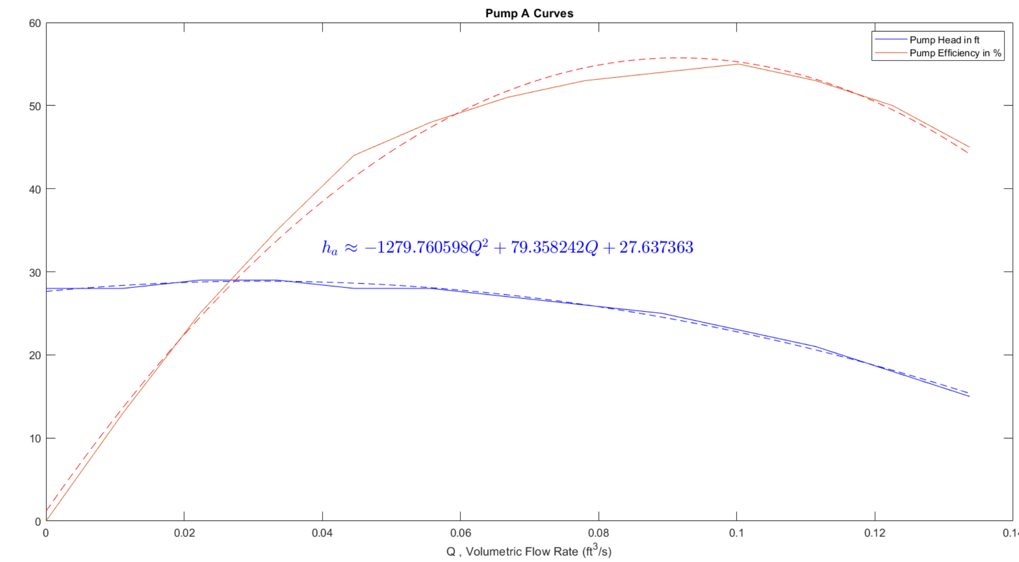
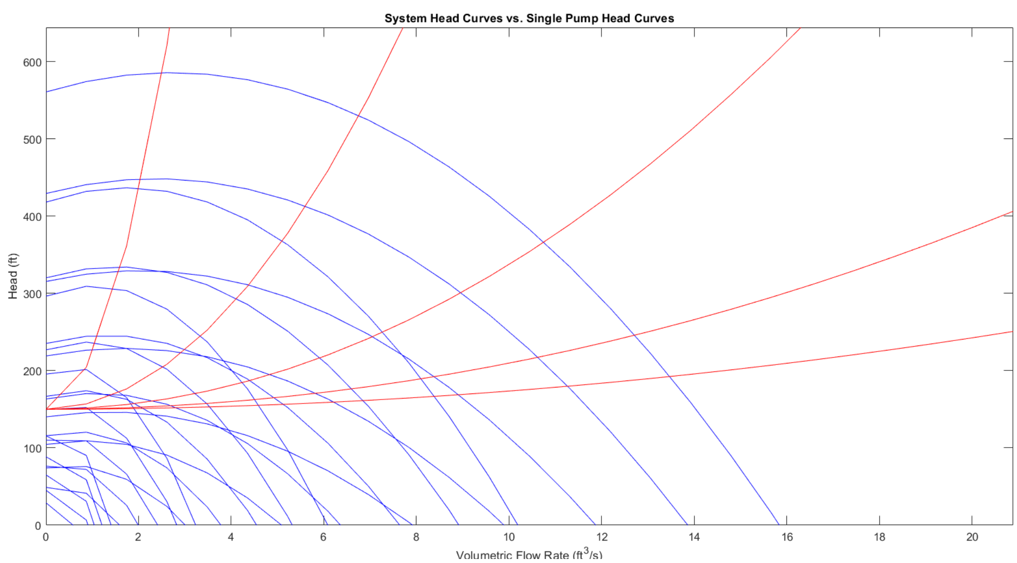


Figure 2: Measured loss K coefficients for three butterfly valves of different manufacturers.

PUMP-PIPE SIZING CONTINUED

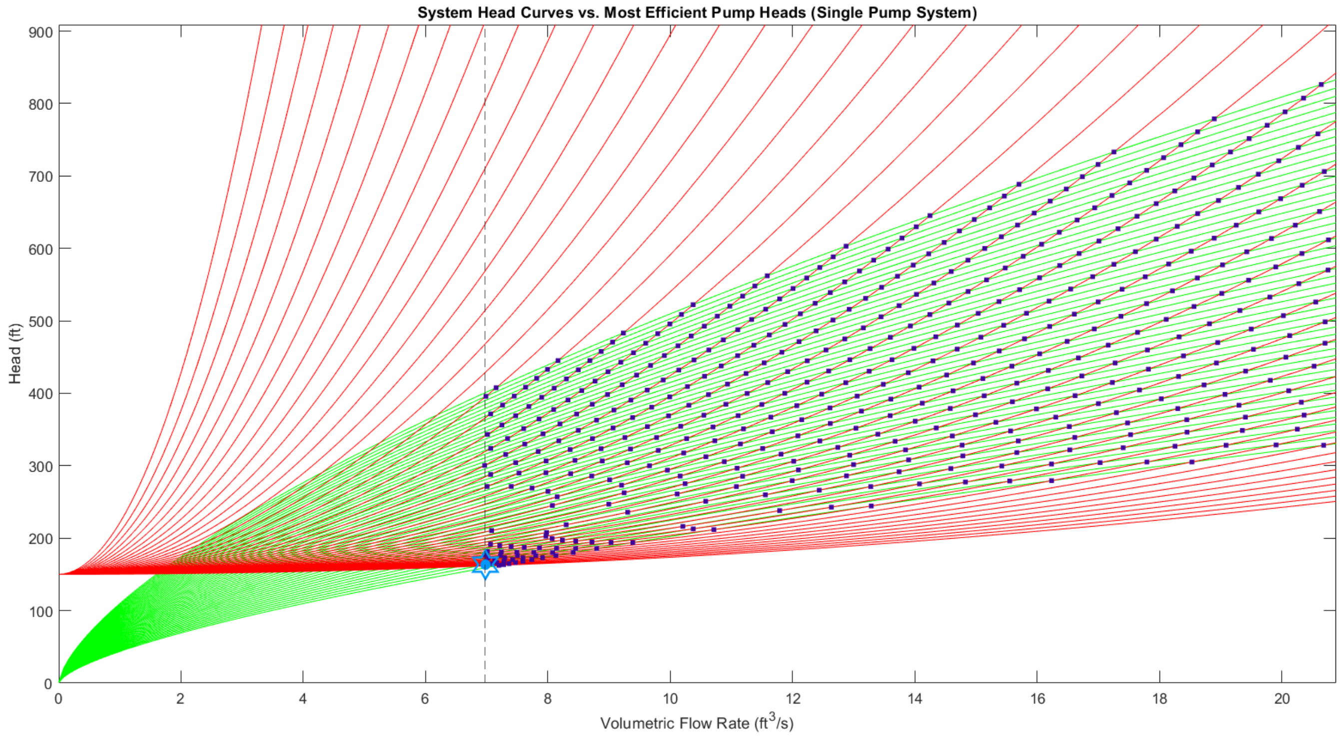
Pump A could only produce a head rise of 29 ft, not enough to overcome even the minimum height of 150 ft, and this wasn't produced at peak efficiency. This meant that pump A had to be scaled. When scaling pumps, you would expect to get a similar shape to the original pump. I expected the scaled plots to resemble the plots for pump A.





Although there are intersections, meaning the pump can supply what a pipe system requires, they may not necessarily intersect where the pump is operating most efficiently. If I were to plot all of the points where the pump was most efficient, and then connect them with lines (so basically connecting across various pump curves), they would look like the graph below

PUMP-PIPE SIZING CONTINUED



The green lines represent all the most efficient pump points for a set shaft speed. Whereas in the previous plot, I guessed both inputs for shaft speed and impeller diameter to generate each curve, I actually had to solve for diameter in this graph. In the previous graph, I scaled the flowrate in the curve fit equation parabolas. However, because I was working directly with flow rate for this graph, I could not guess three inputs. For a specific shaft speed and flow rate, there can be only one impeller diameter.

$$D_{imp} = \sqrt[3]{\frac{Q}{C_{Qmax\eta}\omega}} \quad , \; h_{max\eta} = C_{hmax\eta}\omega^2 D^2_{imp}$$

From this, we could solve for the head rise that the pump could produce at its most efficient point. We could have done the same thing, but instead solving for shaft speed. However, because we were already given a range for the possible rpms, it was easier to solve for impeller diameters.

After doing this for every point for a specific shaft speed, each shaft speed produced its own curve for the most efficient head rises. All that was left to do was find where the curves connecting the most efficient head rises intersected the system curves. From there, we calculated the cost of each intersection (each solution), based on its shaft speed, impeller diameter, and pipe diameter. Our optimal solution was one that met all requirements, but cost the least.

PUMP-PIPE SIZING CONTINUED

Because the pump was not allowed to operate for more than 8 hours a day, and the tank was 1.5 million gallons, the pump needed to supply a minimum flow rate of about 7 cubic feet per second. This means any intersections before the dashed line at the minimum flow rate would be invalid, and so they were deleted.

The reason why the intersections were the solutions has to do with the mechanical energy equation.

$$\frac{P_2}{\rho g} + \frac{V_2}{2g} + z_2 = \frac{P_1}{\rho g} + \frac{V_1}{2g} + z_1 + h_{pump} - h_{losses}$$

- Both ends are at atmospheric pressure, so $P_1 = P_2 = P_{atm} = 0$
- Both ends are reservoirs, so their surfaces practically don't move, so $V_1 = V_2 = 0$

$$z_2 = z_1 + h_{pump} - h_{losses}$$

- Differences in Z values is the height the fluid has to travel, so $z_2 - z_1 = h$

$$h = h_{pump} - h_{losses}$$

$$h_{pump} = h + h_{losses}$$

- Head losses are comprised of major losses and minor losses, so $h_{losses} = h_M + h_m$

$$h_{pump} = h + h_M + h_m$$

It is clear now why the intersection points are the solutions to this problem. The head that the pumps must generate must meet the total head, which are just comprised of losses that come from differences in height, viscosity losses from the pipes, and viscosity losses from the valves and the elbows.

PUMP-PIPE SIZING CONTINUED

Flow Rate	Head Rise	Shaft Speed	Impeller Diameter	Pipe Diameter	Shaft Power	Cost in US Dollars
9.0305	194.1327	96.1712	2.5278	1.2755	269.9638	1.7118e+06
15.5505	278.9051	96.1712	3.0298	1.2755	667.8801	4.0446e+06
8.4516	185.7454	96.1712	2.4726	1.2959	241.7431	1.5474e+06
17.7932	305.1138	96.1712	3.1690	1.2959	836.0100	5.0303e+06
8.0745	180.1788	96.1712	2.4352	1.3163	224.0362	1.4447e+06
19.9043	328.7942	96.1712	3.2897	1.3163	1.0078e+03	6.0370e+06
7.8009	176.0856	96.1712	2.4074	1.3367	211.5273	1.3725e+06
7.5907	172.9065	96.1712	2.3856	1.3571	202.1104	1.3184e+06
7.4200	170.3053	96.1712	2.3676	1.3776	194.5946	1.2756e+06
7.2803	168.1605	96.1712	2.3526	1.3980	188.5252	1.2412e+06
7.1637	166.3603	96.1712	2.3400	1.4184	183.5206	1.2131e+06
7.0634	164.8035	96.1712	2.3290	1.4388	179.2568	1.1894e+06
6.9774	163.4636	96.1712	2.3195	1.4592	175.6352	1.1694e+06
10.2052	216.2587	98.0946	2.6156	1.2347	339.8536	2.1188e+06
11.7878	238.0753	98.0946	2.7444	1.2347	432.1593	2.6602e+06
8.7181	194.7036	98.0946	2.4818	1.2551	261.3918	1.6596e+06
14.8260	277.4017	98.0946	2.9624	1.2551	633.3295	3.8403e+06

MAJOR HEAD LOSS MATRIX

Each Pipe Diameter is its Own System

Q

	D1	D2	D3	D4	D5	D6	D7	...
Q1	0.000362	0.000301	0.000253	0.000214	0.000182	0.000155	0.000133	
Q2	0.6692	0.550491	0.456328	0.381105	0.320387	0.270903	0.230542	
Q3	2.361235	1.93634	1.60082	1.332895	1.11763	0.943297	0.800418	
Q4	5.017986	4.10649	3.38946	2.817633	2.358845	1.98696	1.68408	
Q5	8.625594	7.049223	5.810636	4.82392	4.03315	3.394365	2.874513	
Q6	13.17865	10.75992	8.861031	7.349411	6.138928	5.161868	4.367347	
Q7	18.66782	15.23389	12.53337	10.39009	8.674517	7.286932	6.162368	
Q8	25.10012	20.47254	16.83484	13.94896	11.6344	9.77309	8.256857	
Q9	32.47771	26.47649	21.76089	18.01279	15.02345	12.60753	10.65131	
Q10	40.79918	33.22743	27.29547	22.59334	18.83423	15.80507	13.33945	
Q11	50.04386	40.75509	33.46211	27.69694	23.07688	19.35546	16.33561	
Q12	60.22041	49.04137	40.26455	33.29408	27.73968	23.25435	19.62578	
Q13	71.34324	58.09806	47.67601	39.42156	32.82806	27.51937	23.21336	
Q14	83.43252	67.87525	55.69798	46.05363	38.35011	32.13185	27.09008	
Q15	96.42007	78.43937	64.36564	53.19313	44.2725	37.1118	31.2724	
Q16	110.3498	89.76979	73.62548	60.84451	50.63984	42.42735	35.76897	
Q17	125.234	101.8765	83.55368	69.01383	57.43816	48.09823	40.54931	
Q18	141.0878	114.7151	94.03503	77.70875	64.64133	54.12927	45.61005	
Q19	157.7736	128.3434	105.2052	86.89495	72.28178	60.52645	50.9998	
Q20	175.5189	142.7064	116.9772	96.617	80.32732	67.26266	56.67509	
Q21	194.0849	157.7994	129.3473	106.8327	88.8196	74.37297	62.63352	
Q22	213.6491	173.7041	142.3115	117.539	97.71943	81.82438	68.90804	
..								

COST ANALYSIS

Different components incurred different costs. For example, while having a larger pipe diameter would result in less head loss, meaning the pump would require less energy, the cost for the pipe itself would go up, specifically, per length and per diameter. Similarly, a larger impeller diameter would be able to pump more water, however, at a larger cost to its manufacturing, and its cost to operate it. The valves and elbows didn't really contribute to the cost, although their cost did increase per every inch in diameter.

Component	Cost Relation
Pipe	$\$1\ per\ inch\ in\ diameter\ per\ foot\ of\ length$ $= \$12\ per\ foot\ in\ diameter\ per\ foot\ of\ length$
Pump & Impeller	$\$3500\ +\ \$1500\ per\ inch\ in\ diameter\ of\ impeller\ size$ $= \$3500\ plus\ \$18000\ per\ foot\ in\ diameter\ of\ impeller\ size$
Power	$\$0.10\ per\ kWh$
90 Degree Elbows	$\$50\ +\ \$50\ per\ inch\ of\ pipe\ diameter\ EACH$ $= \$50\ plus\ \$600\ per\ foot\ of\ pipe\ diameter\ EACH$
Butterfly Valves	$\$300\ +\ \$200\ per\ inch\ in\ pipe\ diameter\ EACH$ $= \$300\ +\ \$2400\ per\ foot\ in\ pipe\ diameter\ EACH$

Energy was the most expensive however. Although power only cost \$0.10 per kWh, the pump was running 8 hours every single day for 20 years. Our pump ran at an optimal power of about 175.6 kW. What is interesting is how this number was calculated. Because we were scaling from a similar pump, pump A, they both had similar efficiency curves, and both operated at a maximum efficiency of about 55%. From this, and by using dimensionless analysis, I was able to calculate the power coefficient for all similar pumps. By doing this, I was then able to solve for the power the pump consumed with a formula that included the flow rate, operating speed, impeller diameter, and the power coefficient.

- $\dot{W}_{max\eta Opt} = C_{\mathcal{P}max\eta}\rho\omega^{3}_{Opt}D^{5}_{impOpt}$

What is interesting of note here is how diameter and operating speed affect the power that the pump requires. In the equation to solve for power, the shaft speed term, ω_{Opt} , increases the requirement via a cubic relation, ω^{3}_{Opt} . On the other hand, the impeller diameter, D_{impOpt} , increases the requirement via a quintic relation D^{5}_{impOpt}. You would expect then to optimize the pump in a way as to minimize the quintic term, D^{5}_{impOpt} , meaning having as small an impeller diameter as possible, and to have the cubic term, ω^3_{Opt}, greater than the quintic term, meaning a higher value of shaft speed than a value for impeller, since a quintic term blows up faster than a cubic past values of 1. However, because our operating speed has high bounds, $94.2478\frac{rad}{s}$ to $188.4956\frac{rad}{s}$, and the optimal impeller diameter the program found was small in comparison to the operating speed, $2.3195\ ft$, ω^3_{Opt} blew up faster than D^5_{impOpt}.

DESIGN DESCRIPTION

	Measurement	Cost
Pipe Diameter, Length	$1.4592\ ft,\ 4226.6625\ ft$	$\$74,009.72$
Impeller Diameter	$2.3195\ ft$	$\$45,251.46$
Operation Speed	$96.1712\ rad/s$ $(918.3673\ rpm)$	
Flow Rate	$6.9774\ ft^3/s$	
Pump Head	$163.4636\ ft$	
Pump Power	$175.6352\ kW$	
Pump Energy Over 20 years, 8 hours/day	$10256768.1463\ kWh$	$\$1,025,676.81$
Valve Head Loss	$\times\ 4\ =\ 0.8\ ft$	$\times\ 4\ =\ \$15,208.16$
Elbow Head Loss	$\times\ 10\ =\ 3\ ft$	$\times\ 10\ =\ \$9,255.10$
		Total: $\$1,169,401.26$

For the entire 20 years of operation, the project would cost about $\$1,169,401.26$

APPENDIX A

- Pump A Data

Q (gpm)	h_a (ft)	Efficiency
0	28	0
5	28	13
10	29	25
15	29	35
20	28	44
25	28	48
30	27	51
35	26	53
40	25	54
45	23	55
50	21	53
55	18	50
60	15	45

- **Solving for Coefficients**
  - Solving for the power Pump A consumes at peak efficiency η_{max}
  - $\eta_{max} = \frac{\rho g Q_{max\eta A} h_{max\eta A}}{\dot{W}_{max\eta A}} \Rightarrow \dot{W}_{max\eta A} = \frac{\rho g Q_{max\eta A} h_{max\eta A}}{\eta_{max}}$
  - $\dot{W}_{max\eta A} = \frac{1.94\frac{slugs}{ft^3} \cdot 32.2\frac{ft}{s^2} \cdot 0.10026\frac{ft^3}{s} \cdot 23ft}{55\%} \Rightarrow \dot{W}_{max\eta A} \approx 261.91\frac{lb \cdot ft}{s}$
  - Solving for the Coefficient of Power $C_{\mathcal{P}max\eta}$
  - $C_{\mathcal{P}max\eta} = \frac{\dot{W}_{max\eta A}}{\rho \omega_A^3 D_{impA}^5} \Rightarrow C_{\mathcal{P}max\eta} = \frac{261.91\frac{lb \cdot ft}{s}}{1.94\frac{slugs}{ft^3} \cdot (58.667\pi\frac{rad}{s})^3 (0.4542ft)^5}$
  - $C_{\mathcal{P}max\eta} \approx 0.001116$

- Solving for the flow coefficient at peak efficiency $C_{Qmax\eta}$
- $C_{Qmax\eta} = \frac{Q_{max\eta A}}{\omega_A D_A^3} \Rightarrow C_{Qmax\eta} = \frac{0.10026\frac{ft^3}{s}}{58.667\pi\frac{rad}{s}\cdot(0.4542ft)^3}$
- $C_{Qmax\eta} \approx 0.00581$
- Solving for the head rise coefficient at peak efficiency $C_{hmax\eta}$
- $C_{hmax\eta} = \frac{gh_{max\eta}}{\omega_A^2 D_A^2} \Rightarrow C_{Qmax\eta} = \frac{32.2\frac{ft}{s^2}\cdot 23ft}{(58.667\pi\frac{rad}{s})^2\cdot(0.4542ft)^3}$
- $C_{Qmax\eta} \approx 0.10568$

- **Solving for Power Consumed by Optimized Pump**
  - Scaling to solve for the power our new optimal pump uses $\dot{W}_{max\eta Opt}$
  - $C_{\mathcal{P}max\eta} = \frac{\dot{W}_{max\eta Opt}}{\rho\omega_{Opt}^3 D_{impOpt}^5} \Rightarrow \dot{W}_{max\eta Opt} = C_{\mathcal{P}max\eta}\rho\omega_{Opt}^3 D_{impOpt}^5$
  - $\dot{W}_{max\eta Opt} = 0.001116 \cdot 1.94\frac{slugs}{ft^3}\cdot(96.1712\frac{rad}{s})^3\cdot(2.3195ft)^5$
  - $\dot{W}_{max\eta Opt} \approx 129541.8756\,\frac{lb\cdot ft}{s} \approx 175.6352\,kW$
- Matlab Code:

12/6/22 1:59 AM C:\Users...\Fluid_Mechanics_Project_V7.m 1 of 8

```
clear;
clc;
yearsInSeconds = 6.307e+8; %20 years in seconds
hoursADay = 8;
maxTime = yearsInSeconds / ( 24 / hoursADay);
powerCostRate = 0.10;%10 cents an hour
pumpInitialCost = 3500;
impellerSizeCostRate = 1500*12; % $1500 per inch, in feet
valveInitialCost = 300;
valveSizeCostRate = 200 * 12; % $200 per inch, in feet
elbowInitialCost = 50;
elbowSizeCostRate = 50 * 12; % $50 per inch, in feet
pipeSizeCostRate = 1*12; % $1 per inch of diam per foot of length, in feet

testCurvesPerVar = 50;
meshPoints = testCurvesPerVar*5;
flowRateMin = 6.96252894;
```

```
flowRateMax = flowRateMin*3;
Q =linspace(0.001,flowRateMax,meshPoints);
Dguess=linspace(0.5,1.5,testCurvesPerVar);%make sure the minumum is greater than the  resolution
density = 1.94;
dynamicViscosity = 0.000018588;
deltaP = 0;
b = 4224;
H = 150;
L = sqrt(H^2 + b^2);
E=4.92e-4;
g=32.2;

%https://www.engineeringtoolbox.com/minor-loss-coefficients-pipes-d_626.html
%minorLossComponents = ["90 deg elbow","butterfly valves"];
minorLossCoefficients = [0.3, 0.2];
minorLossComponentsQuantities = [10,4];
minorLossSumCoefficients = minorLossCoefficients.*minorLossComponentsQuantities;
minorLossMatrix = minorLossMatrixFunc(minorLossSumCoefficients,g, Dguess,Q);

resolution = 0.00001;
Re = [];
Re = reynoldsNumberMatrix(density, dynamicViscosity, Dguess, Q);
fFrictionMatrix = coleBrookMatrix( E, Dguess, Re, resolution);
majorLossMatrix = majorLossMatrixFunc(L,g,fFrictionMatrix,Dguess,Q);

totalHeadLossMatrix = majorLossMatrix + minorLossMatrix + H;

%%Pump Data
pump_A = readmatrix('Pump A Data');
pump_A(:,1) = pump_A(:,1) * (1/7.48)*(1/60);
impellerDiam_A = 0.454;
```



```
wGiven_A = 1760*pi/30;
Qfine_A = linspace(0, pump_A(end,1) ,100);
headCurveCoefficeints_A = polyfit(pump_A(:,1),pump_A(:,2), 2);
headCurve_A = polyval(headCurveCoefficeints_A,Qfine_A);
efficiencyCurveCoefficeints_A = polyfit(pump_A(:,1),pump_A(:,3), 2);
efficiencyCurve_A = polyval(efficiencyCurveCoefficeints_A,Qfine_A);

wInputs = linspace(900*pi/30 , 1800*pi/30, testCurvesPerVar);
newImpellerDiams = linspace(impellerDiam_A*2,2,testCurvesPerVar);

figure;
plot(pump_A(:,1) , pump_A(:,2),"b");
hold on;
plot(pump_A(:,1) , pump_A(:,3));
hold on;
```

```
plot(Qfine_A,headCurve_A,"b--")
hold on;
plot(Qfine_A,efficiencyCurve_A,"r--")
hold on;
headTrendLineEqn = sprintf('$%fQ^{2} + %fQ + %f $', headCurveCoefficeints_A(1),
headCurveCoefficeints_A(2),headCurveCoefficeints_A(3));
text( mean([pump_A(1,1) , pump_A(end,1)]), max( pump_A(:,2) ) + 5 , "$h_{a}$
"+"$\approx$ " + headTrendLineEqn, 'FontSize', 16, 'Color',
'b','HorizontalAlignment', 'center', 'VerticalAlignment', 'top',
'Interpreter','latex');
legend('Pump Head in ft','Pump Efficiency in %');
xlabel('Q , Volumetric Flow Rate (ft^{3}/s)')
title("Pump A Curves" )
hold off;

%max efficiency coefficients
maxEffIdx = find(pump_A(:,3)==max(pump_A(:,3)));
flowCoeffAtMaxEff = pump_A(maxEffIdx,1)/(wGiven_A * impellerDiam_A^3);
headRiseCoeffAtMaxEff = g*pump_A(maxEffIdx,2)/(wGiven_A^2 * impellerDiam_A^2);
powerMostEfficient_A = density * g *pump_A(maxEffIdx,1)*pump_A(maxEffIdx,2)/(pump_A
(maxEffIdx,3)/100);
powerCoeffAtMaxEff = powerMostEfficient_A/(density * wGiven_A^3 * impellerDiam_A^5);


%Actually, each combo of diameter and speed is its OWN PUMP CURVE!!!
%will store data for each pump: its w, D
count = 1;
newPumpHeadMatrix = [];
pumpCombos = []; %[w,D]
for i = 1:length(newImpellerDiams)
 for ii = 1:length(wInputs)
 newPumpHeadMatrix(count,:) = ( ( newImpellerDiams(i) * wInputs(ii) ) /  (wGiven_A
*impellerDiam_A) )^2* polyval(headCurveCoefficeints_A, ( (wGiven_A
*impellerDiam_A^3)/(wInputs(ii)*newImpellerDiams(i)^3) ) * Q);
```



```
 pumpCombos(count,1) = wInputs(ii);
 pumpCombos(count,2) = newImpellerDiams(i);
 count = count + 1;
 end
end
newPumpHeadMatrix;

BnewPumpHeadMatrix = newPumpHeadMatrix;%getting rid of all negatives to make ymax the  highest
positive pump head value * 1.1
BnewPumpHeadMatrix(BnewPumpHeadMatrix<0) = 0;

%plotting headloss curves, one curve for each pipe diameter (which is each its own  system)
```

```matlab
figure;
for i = 1:length(Dguess)
 plot(Q,totalHeadLossMatrix(:,i),"r")
 hold on;
end
xlabel('Volumetric Flow Rate (ft^{3}/s)')
ylabel('Head (ft)')
title("System Head Curves")
hold off;

%plotting both head loss and pump head curves
figure;
for i = 1:height(newPumpHeadMatrix)
 plot( Q , newPumpHeadMatrix(i,:) ,"b")
 hold on;
end
%yieldYL = get(gca, 'YLim');
%ylim([0,yieldYL(2)]);
hold on;
 [ ~ , columns] = size(totalHeadLossMatrix);

for i = 1:columns
 plot(Q,totalHeadLossMatrix(:,i), "r")
 hold on;
end
xlim([0,flowRateMax]);
ylim([0, max(BnewPumpHeadMatrix, [], 'all')*1.1]);
xlabel('Volumetric Flow Rate (ft^{3}/s)')
ylabel('Head (ft)')
title("System Head Curves vs. Pump Head Curves")
hold off;


%%Plotting only pump heads at max efficiency
%I cannot just guess diameter anymore. Previously, to scale the curves, I %just
scaled the Q values. However, because I am going point by point, and
```



```matlab
%my inputs are Q (the x axis),and w (given range of RPM),I must solve for D %Since
we were already given a range for w, its probably easier to just %solve for D instead of
solving for w

%testCurvesPerVar = 20;
Q =linspace(0.001,flowRateMax,meshPoints);
wInputs = linspace(900*pi/30 , 1800*pi/30, testCurvesPerVar);
newImpellerDiams = [];
mostEffPumpHeads = [];
for i = 1:length(wInputs)
```

```
for ii = 1:length(Q)
newImpellerDiams(i,ii) = ( Q(ii) / (flowCoeffAtMaxEff * wInputs(i) ))^(1/3);  mostEffPumpHeads(i,ii) =
newImpellerDiams(i,ii)^2 * wInputs(i)^2 *  headRiseCoeffAtMaxEff/g;
end
end
[ ~ , initLength ] = size(totalHeadLossMatrix);
figure;
lowestSystemCurveIdx = find(totalHeadLossMatrix(end,:) == min(totalHeadLossMatrix (end,:)));
for i = 1:length(wInputs)
plot( Q , mostEffPumpHeads(i,:),"g")
hold on;
end
Dguess=linspace(0.5,1.5,initLength);
for i = 1:length(Dguess)
plot(Q,totalHeadLossMatrix(:,i),"r")
hold on;
end

%finding intersections
intersections = [];%[Q,h,w,D,pipeDiam,power,cost] ---> [ft^3/s , ft , rad/s , ft , ft  , kW , $]
for i = 1:height(mostEffPumpHeads)%looping downwards
for ii = 1:initLength
[xx,yy] = polyxpoly(Q,mostEffPumpHeads(i,:),Q,totalHeadLossMatrix(:,ii));  %possibly 2
intersections,so will just append the row to the matrix  if isempty(xx) == false
for iii = 1:height(xx) %the Q where it intersects may not have an entry  for a specific D. w is an
imput here
%again, D = ( Q / (flowCoeffAtMaxEff * wInputs) ))^(1/3);  newD = ( xx(iii) / (flowCoeffAtMaxEff
* wInputs(i) ))^(1/3);  [powUsed,costToRun] = operationCost(Dguess(ii),L,newD,wInputs(i),
maxTime,density, powerCoeffAtMaxEff , powerCostRate, pipeSizeCostRate ,
[pumpInitialCost, impellerSizeCostRate], [valveInitialCost,valveSizeCostRate,
minorLossComponentsQuantities(2)], [elbowInitialCost,elbowSizeCostRate,
minorLossComponentsQuantities(1)]);
intersections = [ intersections ;xx(iii) , yy(iii), wInputs(i), newD  , Dguess(ii) ,powUsed, costToRun];
```



```
end
end
end
end
%would have been easier to limit the plot minimum to flowrate minimum %but
would have made an ugly graph imo
%removing all intersection points before the minimum flowrate
indecies = find(intersections(:,1)<flowRateMin);
intersections(indecies,:) = [];
xline(flowRateMin , "--");
plot(intersections(:,1) , intersections(:,2), "." , "MarkerSize" , 10,'color' ,[63,  5, 156]/256 )
hold on;
%plotting cheapest solution
```

```
bestCost = min(intersections(:,7))
bestIndex = find(intersections(:,7) == min(intersections(:,7)))
plot(intersections(bestIndex,1) , intersections(bestIndex,2), "." , "MarkerSize" ,  25,'color' ,[3, 152,
252]/256 )
plot(intersections(bestIndex,1) , intersections(bestIndex,2), "hexagram",  'MarkerSize',
 20,'color' ,[3, 152, 252]/256,'linewidth',1.5)

xlim([0,flowRateMax]);
ylim([0, max(intersections(:,2))*1.1]);
xlabel('Volumetric Flow Rate (ft^{3}/s)')
ylabel('Head (ft)')
title("System Head Curves vs. Most Efficient Pump Heads" )
hold off;

%Pipe Cost Only
 [~,pipeLineCost] = operationCost(intersections(bestIndex,5),L,newD,0,0,0,0,0,
pipeSizeCostRate,[0,0],[0,0,0],[0,0,0])
%Impeller Cost Only
 [~,impellerCost] = operationCost(0,0,intersections(bestIndex,4),0,0,0,0,0,0,
 [pumpInitialCost, impellerSizeCostRate], [0,0,0], [0,0,0])
%Valve Cost Only
 [~,valveTotalCost] = operationCost(intersections(bestIndex,5),0,0,0,0,0,0,0,0,0,[0,0],
[valveInitialCost,valveSizeCostRate,minorLossComponentsQuantities(2)],[0,0,0]) %Elbow Cost
Only
 [~,elbowTotalCost] = operationCost(intersections(bestIndex,5),0,0,0,0,0,0,0,0,0,[0,0],
[0,0,0],[elbowInitialCost,elbowSizeCostRate,minorLossComponentsQuantities(1)]) %Power Cost
Only
 [~,powerTotalCost] = operationCost(0,0,intersections(bestIndex,4),intersections
 (bestIndex,3),maxTime,density, powerCoeffAtMaxEff , powerCostRate,0,[0,0],[0,0,0], [0,0,0])

%Best Specs:
disp("Q = "+intersections(bestIndex,1) + " , h = " + intersections(bestIndex,2) + " ,  w = " +
intersections(bestIndex,3) + " , impD = " +intersections(bestIndex,4) + " ,  pipeD = " +
intersections(bestIndex,5) + " , power = " + intersections(bestIndex,6) +
```



```
 " , $ = " + intersections(bestIndex,7))
holdUpWait = pipeLineCost+impellerCost+valveTotalCost+elbowTotalCost+powerTotalCost

%%Functions
function [frictionFactorMatrix] = coleBrookMatrix(rough,diamArray,reynoldsMatrix, resol)
 fFricMatrix = [];
 for i = 1:height(reynoldsMatrix)
 for ii = 1:length(diamArray)
 closeEnough = false;
 fFric = 0;
 tol = 1e-12;%for floating point numbers
 %fprintf('(%f,%f) \n',i,ii)
```

```
 %fprintf('Re = %f \n',reynoldsMatrix(i,ii))
 %fprintf('D = %f \n',diamArray(ii))
 while closeEnough == false
 fFricPrev = fFric;
 if abs(1/sqrt(fFric) + 2*log10((rough/diamArray(ii))/3.7+2.51/ (reynoldsMatrix(i,ii)*sqrt(fFric)))) > tol
%checking inequality with floating point  numbers
 fFric = fFric + resol;
 end
 fFric;%delete the semicolon to see in real time
 %check if leftside - rightside < tolerance, OR
 %if it overshot, to use the previous value
 if abs(1/sqrt(fFric) + 2*log10((rough/diamArray(ii))/3.7+2.51/ (reynoldsMatrix(i,ii)*sqrt(fFric)))) <
 tol || abs(1/sqrt(fFric) + 2*log10 ((rough/diamArray(ii))/3.7+2.51/(reynoldsMatrix(i,ii)*sqrt(fFric))))
 > abs(1/sqrt (fFricPrev) + 2*log10((rough/diamArray(ii))/3.7+2.51/(reynoldsMatrix(i,ii)*sqrt
 (fFricPrev))))
 closeEnough = true;
 if abs(1/sqrt(fFric) + 2*log10((rough/diamArray(ii))/3.7+2.51/ (reynoldsMatrix(i,ii)*sqrt(fFric)))) >
abs(1/sqrt(fFricPrev) + 2*log10
((rough/diamArray(ii))/3.7+2.51/(reynoldsMatrix(i,ii)*sqrt(fFricPrev))))  fFric = fFricPrev;
 end
 end
 end
 fFric;
 fFricMatrix(i,ii) = fFric;
 end
 end
 frictionFactorMatrix = fFricMatrix;
end

function [ReNum] = reynoldsNumberMatrix(dens, dynVis, diamArray, volFlowArray)  for i =
1:length(volFlowArray)
 ReNum(i,:) = 4*dens/(pi * dynVis) * volFlowArray(i)./diamArray ;  %Vel =
4*volFlowArray(i)./(pi*diamArray.^2);
```



```
 %fprintf('V = %f) \n', Vel)
 end
end

function [hMajorMatrix] = majorLossMatrixFunc(Length, Gravity,fFricMatrix,
diameterArray,QArray)
 for i = 1:length(QArray)
 for ii = 1:length(diameterArray)
 hMajorMatrix(i,ii) = (8*Length/(pi^2 * Gravity)) * (fFricMatrix(i,ii) /diameterArray(ii)) *
(QArray(i)/(diameterArray(ii)^2))^2;
 end
 end
end
```

```
function [hMinorMatrix] = minorLossMatrixFunc(K_m_list,Gravity, diameterArray,QArray)  for i =
1:length(QArray)
 for ii = 1:length(diameterArray)
 for iii = 1:length(K_m_list)
 hMinorMatrix(i,ii) = 8*K_m_list(iii)/(pi*Gravity) * (QArray(i)/ (diameterArray(ii)^2))^2 ;
 end
 end
 end
end

function [pow , cost] = operationCost(pipeDiam,pipeLength,impellerDiam,shaftSpeed, time,dens,
powerCoeff , powCostRate, pipeCost ,impellerCostStuff, valveCostStuff,  elbowCostStuff)
 %Start Up Costs
 pipeTotal = pipeDiam * pipeLength * pipeCost;
 impellerTotal = impellerCostStuff(1) + impellerCostStuff(2) * impellerDiam;  valveTotal =
(valveCostStuff(1) + valveCostStuff(2) * pipeDiam)*valveCostStuff (3);
 elbowTotal = (elbowCostStuff(1) + elbowCostStuff(2) * pipeDiam)*elbowCostStuff (3);
 startUpCost = pipeTotal + impellerTotal + valveTotal + elbowTotal;
 %Calculating Power Used in kW
 power = (dens * powerCoeff * shaftSpeed^3 * impellerDiam^5); % is in lb ft/s.  need to convert to
kW
 power = power * 0.0013558179483314;
 pow = power;
 %power = density * 32.2

 %Calculating cost of Power
 toHours = time/3600;
 energyCost = powCostRate*toHours*power;

 cost = startUpCost + energyCost;
```

```
end
```