**qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

|  |
| --- |
| Sprinkler Building Design  Fluid Mechanics for HVAC  Utmost Engineering  Tariq Ramlall, Sai Sharan Kasa, Srivighnesh Rajagopalan, Akanksha Chauhan |

**Introduction**

This document serves as the basis of design for the installation of a sprinkler system for the residential building on East 9th street. This document will contain a brief explanation of our method of design, the reasoning behind why sprinklers were places where they were, blueprints for where each sprinkler will be installed, and calculations for the pressure and flow rate of each sprinkler in the hydraulic zone.

The building is 8 stories and residential. The first floor is served mainly as a garage; floors 2 through 7 each house two apartments each; the eighth floor houses a single large apartment; the roof has one small enclosed area leading to a staircase; the cellar is mostly an open area used for recreation for the building's tenants.

Our goal is to provide continuous sprinkler coverage so residents may safely exit the building from the main staircase. We also wish to design our system efficiently enough that we reach the desired pressure at the most remote sprinkler head without needing to use a pump.

We will check the pressure and other factors at each sprinkler head using the software Hass. We will also check the calculations manually to ensure the design is safe and reliable.

**Methodology**

The code we are following is NFPA13, but because of the nature of our building we will say that our hydraulic zone covers 900 square feet. Because the building contains over ten apartments, we will consider the building to be an Ordinary Hazard.

We have chosen to use 12x12 sprinkler heads exclusively for our sprinkler system. The K value for these sprinkler heads is 4.9, and the manufacturer of these heads is Reliable.

Because we are providing continuous coverage, our sprinkler heads cannot be more than 12 feet apart. We also require that sprinkler heads are positioned at least 4 inches away from the wall. In addition, each sprinkler head must be at least one foot away from an elbow or a tee pipe.

Every room should be covered except for these exceptions: bathrooms under 55 feet squared don’t require a sprinkler head, closets under 12 feet squared don’t require a sprinkler head, electrical rooms don’t require a sprinkler head, mechanical rooms don’t require a sprinkler head.

For our calculations, our hydrant test flow rate is 500 gpm, our residual pressure is 83 psi, and our static pressure is 85 psi. Our RPZ is -12.

**Calculations**

The point of doing the calculations is to provide a check to see if we used the Hass software correctly, as well as ensure our design is safe and accurate. To check our manual calculations, we make use of two important equations.

The first is equation is the relationship between the pressure and the flow rate:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

The second equation is the Hazen Williams equation which calculations the friction loss, h.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

L is the pipe length, C is the roughness, Q is the flow rate, and d is the pipe diameter.

We can modify the second equation to convert the units to psi and incorporate our roughness constant as such:

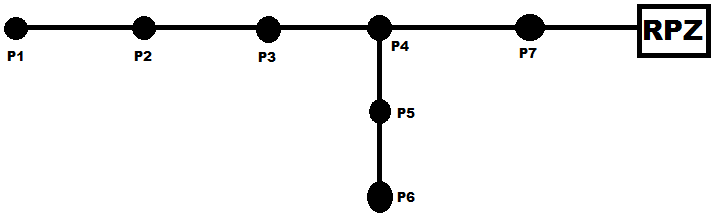
|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Additionally, we will make use of the fact that the pressure at a node is equal to the pressure of next node away from it (further away from the RPZ) plus the pressure loss in that pipe. The equation looks like:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

**Manual Calculations**

Let's go through an example of a sprinkler system to see how we'll do the manual calculations.



We will take the pressure of the most remote sprinkler head (in this example, P1) from the Hass software. So we're given P1.

We can easily calculate P2 since we have P1. We first calculate the flow of P1 by rewriting Equation (1) to:

We have K1and P1 so Q1 is calculated. From here, we use Equation (3) to find the friction loss between the pipes:

We have L1-2, we have d1-2, and we just calculated Q1. We can now find P2 using Equation (4):

Using the same method, we can calculate P3, P4 and P7.

To calculate P5 and P6 is a little trickier since they come off of a branch. However, the same formulas apply.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | (5) | |
|  |  | | | (6) |
|  | |  | (7) | |

(Notice that in Equation (7), the friction loss is subtracted from the previous node since we are moving further away from the source.)

P5 is unknown and Q5 is unknown. However, we have enough equations relating P5 and Q5 to solve for them. Replace the Q5 in Equation (6) with the Q5 in Equation (5) to get:

If you plug that h4->5 into Equation 7 and simplify things, then:

This is a polynomial equation where P5 is the only unknown. We can solve this and find P5.

P6 is solved the same way that P5 was solved.

Using the calculations shown in this section, all of the calculations for our hydraulic zone can be calculated.

**Program**

Although we are doing these calculations ‘manually,’ we use the same equations over and over. To save time and avoid mistakes, we’ve created a simple program to do the calculations provided you input the parameters (the ‘knowns’) of our equations. This program will not only simplify things for us, but it will provide consistency and make sure we use the same parameters and equations for each calculation.

The program is written in C++ and can be compiled using Visual Studio C++ in Windows or a g++ compiler in Linux.

The source code is provided here:

// Program to calculate the new pressure at a sprinkler head.

#include <iostream>

#include <cmath>

using namespace std;

void OutputAnswer(double pressure, double newq)

{

cout << "\nThe pressure (P) at this node is " << pressure << endl;

cout << "The flow (Q) at this node is " << newq << endl;

cout << "\nTo run the program again type any character and press enter. ";

char zzz;

cin >> zzz;

cout << endl << endl;

}

//Function for branches.

void CalculateBranch(double pressureprevious, double kprevious, double k, double pipeLength, double diameter)

{

// P5 + [P5^1.85/2]\*[0.00064\*L\*(K5^1.85)/(d^4.8655)] - P4 = 0

double pressure = 7;

for (double count = pressure; count <= pressureprevious ; count + 0.0001)

{

pressure = pressure + 0.0001;

double LHS = pressure + pow(pressure, 1.85/2)\*0.00064\*pipeLength\*pow(k, 1.85)/pow(diameter, 4.8655) - pressureprevious;

//double check = trunc(LHS \* 100) / 100;

//if (check == 0.000)

if (LHS >= -0.001 && LHS <= 0.001)

{

double newq = k\*sqrt(pressure);

OutputAnswer(pressure, newq);

break;

}

}

}

void CalculateNode(double pressureprevious, double kprevious, double k, double pipeLength, double diameter)

{

// Find flow of previous pipe.

double qprevious = kprevious\*sqrt(pressureprevious);

double h = 0.00064\*pipeLength\*pow(qprevious,1.85)/pow(diameter, 4.8655);

// Pressure at the current node is pressure of previous node + pressure loss at that node.

double pressure = h + pressureprevious;

// New Q.

double newq = k\*sqrt(pressure);

OutputAnswer(pressure, newq);

}

void GetInfo()

{

// Ask user which node we're dealing with.

int x;

cout << "If the node we're calculating is on a branch, type 1. \nIf the node isn't on a branch, type 2. ";

cin >> x;

if (x == 1 || x == 2)

{

}

else

{

cout << "\nError. Please enter the number 1 or the number 2." << endl;

GetInfo();

}

// Ask for the input values.

double pressure;

cout << "\nEnter the pressure of the previous node. ";

cin >> pressure;

double pk;

cout << "Enter the K value for your previous node. ";

cin >> pk;

double k;

cout << "Enter the K value for your current node. ";

cin >> k;

double pipeLength;

cout << "Enter the pipe length in feet (convert any inches to a decimal). ";

cin >> pipeLength;

double diameter;

cout << "Enter the diameter of your pipe (inches). ";

cin >> diameter;

// Pass the values into the calculate function.

if (x == 1)

{

CalculateBranch(pressure, pk, k, pipeLength, diameter);

}

else

{

CalculateNode(pressure, pk, k, pipeLength, diameter);

}

}

int main()

{

GetInfo();

return main();

}