管網計算ライブラリの利用例　仮想管路の作成と計算　Visual C#

名前空間　HydroObjectsがライブラリ

仮想的な管路網を作成して計算する例
複数の格子状のブロックを二次圧一定バルブ
または二次圧一定ポンプで水源につないだ管路網

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using System.Diagnostics;
namespace Demo
{
    public partial class Form1 : Form
    {
        private HydroObjects.IPipeNetwork m_PipeNetwork;
        public Form1()
        {
            InitializeComponent();
            textBox1.Text = "10";
            textBox2.Text = "40";
        }
        private void button1_Click(object sender, EventArgs e)
        {
            int nBlock, nSize, i, j, lBlock;
            int nNode = 0, nEdge = 0;

            nBlock = Int32.Parse(textBox1.Text); //10;
            nSize = Int32.Parse(textBox2.Text);
            m_PipeNetwork = new HydroObjects.PipeNetwork();
            //水源節点_
            HydroObjects.BoundaryH sourcenode = new HydroObjects.BoundaryH();
            sourcenode.Type = HydroObjects.enumBoundaryHType.NodeTypeSource;
            sourcenode.ID = "S";
            sourcenode.H = 100;
            m_PipeNetwork.AddNode(sourcenode); nNode++;
            for (lBlock = 0; lBlock < nBlock; lBlock++)
            {
                string sBlock = lBlock.ToString("000");
                //水源ー入口
                if ((lBlock % 2) == 0)
                {
                    //二次圧一定バルブ
                    HydroObjects.ConstantHeadValve consthvalve = new HydroObjects.ConstantHeadValve();
                    consthvalve.HeadValue = 80.0;
                    consthvalve.ID = "V" + sBlock;
                    consthvalve.StartNodeID = "S";
                    consthvalve.EndNodeID = sBlock + "001001";
                    m_PipeNetwork.AddEdge(consthvalve);
                }
                else
                {
                    //二次圧一定ポンプ
                    HydroObjects.ConstantHeadPump consthpump = new HydroObjects.ConstantHeadPump();
                    consthpump.HeadValue = 120.0;
                    consthpump.ID = "P" + sBlock;
                    consthpump.StartNodeID = "S";
                    consthpump.EndNodeID = sBlock + "001001";
                    m_PipeNetwork.AddEdge(consthpump);
                }
                nEdge++;
                //格子内節点
                for (i = 1; i <= nSize; i++)
                {
                    for (j = 1; j <= nSize; j++)
                    {
                        if (i == 1 && j == 1)
                        {
                            //二次側節点
                            HydroObjects.BoundaryH outlet = new HydroObjects.BoundaryH();
                            outlet.Type = HydroObjects.enumBoundaryHType.NodeTypeOutlet;
                            outlet.ID = sBlock + i.ToString("000") + j.ToString("000");
                            m_PipeNetwork.AddNode(outlet);
                            if ((lBlock % 2) == 0) outlet.H = 80.0;
                            else outlet.H = 120.0;
                        }
                        else
                        {
                            //一般節点
                            HydroObjects.PipeNode pipenode = new HydroObjects.PipeNode();
                            pipenode.ID = sBlock + i.ToString("000") + j.ToString("000");
                            pipenode.Q = 0.001;
                            m_PipeNetwork.AddNode(pipenode);
                        }
                        nNode++;
                    }
                }
                //横格子
                for (i = 1; i <= nSize; i++)
                {
                    for (j = 1; j <= nSize - 1; j++)
                    {
                        HydroObjects.Pipe pipe = new HydroObjects.Pipe();
                        if (i == 1) pipe.Diameter = 0.5;
                        else pipe.Diameter = 0.2;
                        pipe.LossFormula = HydroObjects.enumPipeLossFormula.Hazen_Williams;
                        pipe.LocalLossRule = HydroObjects.enumPipeLocalLossRule.LossCoef;
                        pipe.LocalF = 0.0;
                        pipe.Length = 20;
                        pipe.CoefC = 110.0;
                        pipe.ID = "EW" + sBlock + i.ToString("000") + j.ToString("000");
                        pipe.StartNodeID = sBlock + i.ToString("000") + j.ToString("000");
                        pipe.EndNodeID = sBlock + i.ToString("000") + (j + 1).ToString("000");
                        m_PipeNetwork.AddEdge(pipe); nEdge++;
                    }
                }
                //縦格子
                for (i = 1; i <= nSize - 1; i++)
                {
                    for (j = 1; j <= nSize; j++)
                    {
                        HydroObjects.Pipe pipe = new HydroObjects.Pipe();
                        if (j == 1) pipe.Diameter = 0.5;
                        else pipe.Diameter = 0.2;
                        pipe.LossFormula = HydroObjects.enumPipeLossFormula.Hazen_Williams;
                        pipe.LocalLossRule = HydroObjects.enumPipeLocalLossRule.LossCoef;
                        pipe.LocalF = 0.0;
                        pipe.Length = 20;
                        pipe.CoefC = 110.0;
                        pipe.ID = "SN" + sBlock + i.ToString("000") + j.ToString("000");
                        pipe.StartNodeID = sBlock + i.ToString("000") + j.ToString("000");
                        pipe.EndNodeID = sBlock + (i + 1).ToString("000") + j.ToString("000");
                        m_PipeNetwork.AddEdge(pipe); nEdge++;
                    }
                }
            }
            label1.Text = "管網完成¥n節点数" + nNode + "  管路数?" + nEdge;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Stopwatch sw = new Stopwatch();
            sw.Reset();
            sw.Start();
            HydroObjects.SteadyStateModel model = new HydroObjects.SteadyStateModel();
            model.PipeNetwork = m_PipeNetwork; model.License="              ";
            model.MaxIteration = 100;
            model.ConvergenceCriteria = 0.000001;
            model.EventInterval = 1;
            model.Computing += new HydroObjects.SteadyStateModel.ComputingEventHandler(ComputingEvent);
            label1.Text = "閉路探索";
            label1.Refresh();
            //計算実行
            HydroObjects.ReturnCode ret = model.Compute(1);
            sw.Stop();
            if (ret != HydroObjects.ReturnCode.COMP_SUCCEEDED)
            {
                string msg = model.GetErrorMessage();
                label1.Text = "エラー" + msg;
            }
            else
            {
                label1.Text = "計算時間：" + sw.Elapsed.ToString("m¥"分¥"s¥¥.fff") + "秒";
            }
            model.Computing -= new HydroObjects.SteadyStateModel.ComputingEventHandler(ComputingEvent);
        }
        //実行中のイベントハンドラー[
        private bool ComputingEvent(int IterationCount)
        {
            label1.Text = "繰り返し:" + IterationCount.ToString();
            label1.Refresh();
            return false;
        }
    }
}
```

管路網オブジェクト PipeNetwork を作成

水源は BoundayH オブジェクト　Type を NodeTypeSource;にする

二次圧一定バルブは ConstantHeadValve オブジェクト

全ての枝は、始点と終点の節点名を指定

二次圧一定バルブは ConstantHeadPump オブジェクト

二次圧一定の吐出側節点は、BoundaryH オブジェクト　Type を NodeTypeOutlet;にする

パイプの接続点は、PipeNode オブジェクト
Q は取り出し流量

管路は Pipe オブジェクト

定常計算(SteadyStateModel　)オブジェクトを作って管路網を代入

計算の実行

実行ライセンスが必要

計算の実行中の繰り返しごとにイベントが発生
計算の進行を表示できます