Advances in Engineering Software **15** (1992) 27–32

Computer program to compute pressure forces in pipe flow junctions

R.J. Whitley

Department of Mathematics, University of California, Irvine, California 92717, USA

&

T.V. Hromadka II

Boyle Engineering Co., Newport Beach, California 92658 and Department of Mathematics, California State University, Fullerton, California 92634, USA

A frequently occurring and difficult to solve problem in civil engineering and water resources analysis is the analysis of hydraulic phenomena in storm flow pipe systems. In the design of storm drain flood control systems, the analysis of energy losses in pipeflow junction structures is of key interest. The general practice is to use a pressure-plus-momentum balance analysis in order to estimate the change in water surface (or hydraulic grade line) elevations through the pipe structure, from which an energy loss is subsequently estimated.

A computer program is presented for the evaluation of pressure forces within a pipe junction structure. The water profile or HGL is assumed to be a straight line between endpoint conditions. The program provides estimates of force for both a complete mathematical formulation as well as a popular estimate of the complete formula. The program can be substituted for other estimates in the analysis of storm drain systems.

INTRODUCTION

A frequently occurring and difficult to solve problem in civil engineering and water resources analysis is the analysis of hydraulic phenomena in storm flow pipe systems. In the design of storm drain flood control systems, the analysis of energy losses in pipeflow junction structures is of key interest. The general practice is to use a pressure-plus-momentum balance analysis[1] in order to estimate the change in water surface (or hydraulic grade line) elevations through the pipe structure, from which an energy loss is subsequently estimated.

To estimate the junction energy loss, the total pressure force acting upon the floodflow in the structure needs to be determined. It is this step of the analysis that has proved to be the most difficult to solve for civil engineers. In this paper, a computer program is developed that integrates the pressure force of the pipe junction, for both unsealed or sealed flow conditions, and the flow-directional component (the 'x' direction) is subsequently computed. This force component in the direction of flow, noted as F_x, is then available for use in balancing forces to change the momentum in the x-direction. The forces in the perpendicular directions are assumed balanced by supports (such as anchor blocks), and are not evaluated in the computation of forces in the x-direction.

Advances in Engineering Software 0965-9978/92/$05.00

MATHEMATICAL MODEL DEVELOPMENT

The mathematical model development is presented in detail in Ref. 2, but is reviewed in the following for the reader's convenience.

In order to develop the x direction of force in the pipe junction structure, it is assumed that water pressures are equal to hydrostatic pressures, and that the water surface or pressure profile gradients are constant in the structure, (i.e. the profiles are straight lines through the structure).

The geometry of the junction control volume (or pipe structure) is as indicated in Fig. 1. In (x, y, z) space the plane $x = 0$ cuts the structure outlet pipe at its downstream end, and this section of the pipe structure is a circle of radius r_1 with center at the point $(0, 0, r_1)$. When viewed from above, the central axis of the

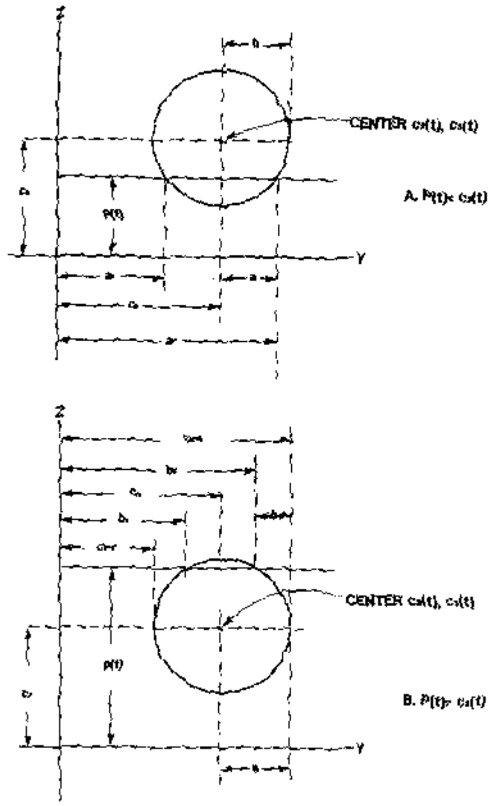

Fig. 3. Hydrostatic pressure in the plane $x = t$.

Also

$$a_1 = c_2 - (a_2 - c_2) = c_2 - a \tag{9}$$

To find the x-component of the force at a point on the bottom half of the structure, due to the pressure, the x-component of the outward pointing unit normal is needed. From eqn (3),

$$u_x = \frac{\partial u}{\partial x} = -2(y - c_2)\frac{\partial c_2}{\partial x} - 2(z - c_3)\frac{\partial c_3}{\partial x} - 2r$$

from which the component of the normal derivative in the x-direction is

$$\frac{-u_x}{[u_x^2 + 4r(x)^2]^{1/2}} \tag{10}$$

where the sign has been chosen to give a positive sign for a force acting towards the origin.

The element of surface area, dS, in terms of $dxdy$, is

$$ds = \frac{\{u_x^2 + 4r(x)^2\}^{1/2}}{|u_z|}dxdy \tag{11}$$

where

$$|u_z| = |2(z - c_3(x))| \tag{12}$$

Combining (5), (7), (8), (9), (10), (11), and (12), the x-component of the force on the bottom half of the structure is

$$Fxb = \gamma\int_o^L\int_{a_1(x)}^{a_2(x)}\{p(x) - zb(x, y)\}\{-u_x/|u_z|\}dydx \tag{13}$$

The integrand of (13) may be written as

$$\{p - c_3 + [r^2 - (y - c_2)^2]^{1/2}\} \tag{14}$$

times

$$\left\{-\frac{\partial c_3}{\partial x} + \left\{(y - c_2)\frac{\partial c_2}{\partial x} + r\frac{\partial r}{\partial x}\right\}/[r^2 - (y - c_2)^2]^{1/2}\right\} \tag{15}$$

Integrating with respect to y,

$$Fxb = 2\gamma\int_o^L fb(x)dx \tag{16}$$

where

$$fb(x) = \left(-(p - c_3)\frac{\partial c_3}{\partial x} + r\frac{\partial r}{\partial x}\right)a + \left(-\frac{\partial c_3}{\partial x}(r^2/2) + (p - c_3)r\frac{\partial r}{\partial x}\right)\sin^{-1}(a/r) - \frac{\partial c_3}{\partial x}(a/2)\sqrt{(r^2 - a^2)} \tag{17}$$

We now analyse the pressure on the top half of the structure. The pressure at the wetted point (t, y, zt) on the soffit of the structure, where

$$zt(t, y) = c_3(t) + [r(t)^2 - (y - c_2)^2]^{1/2} \tag{18}$$

is given by the specific weight of water γ times

$$p(t) - zt(t, y) \tag{19}$$

Define $b_2(t)$ so that in the plane $x = t$ the y-coordinates of the wetted points on the right half of the top half of the structure are the points

$$b_2(t) \leqq y \leqq r(t)$$

As indicated in Fig. 3,

$$b = b_2 - c_2 = \begin{cases} 0 & \text{if } p \geqq c_3 + r \\ [r^2 - (p - c_3)^2]^{1/2} & \text{if } c_3 \leqq p \leqq c_3 + r \\ r & \text{if } p \leqq c_3 \end{cases} \tag{20}$$

Similarly define b_1, so that in the plane $x = t$ the y-coordinates of the wetted points on the right half of the top half of the structure are the points

$$c_2 - r \leqq y \leqq b_1 \text{ and } b_2 \leqq y \leqq c_2 + r,$$

as indicated in Fig. 3.

COMPUTER PROGRAM

```
      program pipe

      real  area,delz,Fx,F12,p1,p2,r1,r2,temp1,temp2,temp3

      print *,"Program Pipe"
10    print *,"downstream pipe radius = ?"
      read *, r1
      print *,"upstream pipe radius  = ?"
      read *, r2
      print *,"downstream pressure height (from bottom of pipe) = ?"
      read *, p1
      print *,"upstream pressure height (from bottom of pipe) = ?"
      read *, p2
      print *,"tilt height delz of upstream end of pipe = ?"
      read *, delz
      temp1=Fx(delz,p1,p2,r1,r2)
      print *, "The Fx force = ", temp1
      temp2=F12(p1,p2,r1,r2)
      print *, "The F12 force = ", temp2
      print *,"The sum of the forces is  ",temp1+temp2
      temp3=(p2+delz-p1)*(area(p1,r1)+area(p2,r2))/2
      print *,"The approximate formula gives  ", temp3
      if (abs(temp1+temp2).gt.0) then
           print *,"relative error 100(Force-approx)/Force = ",
     &     100*(temp1+temp2-temp3)/(temp1+temp2),"%"
      endif
      goto 10
      end

      real function Fx(delz,p1,p2,r1,r2)
c     Robert Whitley, Dept of Math, Univ of CA, Irvine, CA 91717
c     T. V. Hromadka II, Williamson and Schmid, 17782 Sky Park Blvd,
c     Irvine, CA 92714

c     Computes the force in the x-direction a pipe exerts to oppose the
c     pressure due to water in the pipe.
c          The geometry of the pipe is as follows: In (x,y,z) space, the
c     plane x=0 cuts the pipe at its downstream end, and this section of
c     the pipe is a circle of radius r1 with center in the plane y=0.
c     The plane x=L cuts the pipe at its upstream end, and this section
c     of pipe is a circle of radius r2 with center in the plane y=0.
c     Furthermore the bottom of the pipe, consisting of the line joining
c     the lowest point on the downstream circle to the lowest point on
      the upstream circle, is taken to be at height 0 at x=0 and at
c     height "delz" at x=L. The body of the pipe is the minimal surface
c     spanning the downstream circle and the upstream circle, e.g. if
c     delz=0 and both circles have the same radius, it is a right
c     circular cylinder.
c          It happens that the final result is independent of L which is
c     taken to be 1 in the program.
c           Pressure heights p1 and p2 are given at x=0 and x=l=L, each
c     measured from the bottom of the pipe. If the pipe is full at
c     either end, the pressure height can be higher than the top of the
c     pipe. An assumption made is that the pressure height corresponding
c     to a point x between 0 and 1 is a linear interpolation of the
c     heights p1 and p2; e.g. if the end pressure heights are each below
c     the corresponding top of the pipe, the water surface is modeled
c     as a straight line.
c           The program takes the pressure at any point on the surface of
c     the pipe due to the pressure height above that point, and
c     multiplies that by an element of surface area dS=sec(gamma)dxdy to
c     obtain the force at that point and then multiplies that force by
c     the x-component of the normal to the surface at the point to get
c     the x-component of the force. This force in the x-direction is
c     then integrated over the wetted surface of the pipe to get the
c     force Fx. The integral is taken in two parts; one over the wetted
c     surface of the bottom half of the pipe, giving Fxb, and the other
c     over the wetted surfact of the top of the pipe, giving Fxt. Then
c     Fx=Fxb+Fxt.
c          The net force F12 in the x direction on the water surfaces at
c     both ends of the pipe is also computed. The total force in the x
c     direction is Fx+F12.

      real delz,fb,ft,Fxb,Fxt,p1,p2,r1,r2,simpson,tdelz,tr1,tr2,tp1,tp2
      common/tparms/tdelz,tp1,tp2,tr1,tr2
      external fb,ft
      tdelz=delz
      tr1=r1
      tr2=r2
      tp1=p1
      tp2=p2
      Fxb=-2.0*simpson(0.0,1.0,1E-4,fb)
      print *,"Fxb=",Fxb
      Fxt=-2.0*simpson(0.0,1.0,1E-4,ft)
      print *,"Fxt=",Fxt
      Fx=Fxb+Fxt
      return
      end

      real function fb(x)
      real a,c,csubx,p,r,rsubx,tdelz,temp1,temp2,temp3,
     &     tr1,tr2,tp1,tp2,x
      common/tparms/tdelz,tp1,tp2,tr1,tr2
      r=tr1+x*(tr2-tr1)
      rsubx=tr2-tr1
      c=tr1+x*(tr2-tr1+tdelz)
      csubx=tr2-tr1+tdelz
      p=tp1+x*(tp2-tp1+tdelz)
      if (p.le.(c-r)) then
           a=0.0
      endif
      if ((p.gt.(c-r)).and.(p.lt.c)) then
           a=sqrt(r*r-(p-c)*(p-c))
      endif
      if (p.ge.c) then
           a=r
      endif
      temp1=(-csubx*(p-c)+rsubx*r)*a
      temp2=-csubx*(r*r/2)+rsubx*r*(p-c)
      temp2=temp2*asin(a/r)
      temp3=-csubx*a*sqrt(abs(r*r-a*a))/2
      fb=temp1+temp2+temp3
      return
      end

      real function ft(x)
      real b,c,csubx,p,r,rsubx,tdelz,temp1,temp2,temp3,temp4,pi,
     &     tr1,tr2,tp1,tp2,x
      common/tparms/tdelz,tp1,tp2,tr1,tr2
      pi=3.1415926
      r=tr1+x*(tr2-tr1)
      rsubx=tr2-tr1
      c=tr1+x*(tr2-tr1+tdelz)
      csubx=tr2-tr1+tdelz
      p=tp1+x*(tp2-tp1+tdelz)
      if (p.le.c) then
           b=r
      endif
      if ((p.gt.c).and.(p.lt.(c+r))) then
           b=sqrt(r*r-(p-c)*(p-c))
      endif
      if (p.ge.(c+r)) then
           b=0.0
      endif
      temp1=(csubx*(p-c)-rsubx*r)*(r-b)
      temp2=csubx*(r*r/2)-rsubx*r*(p-c)
      temp2=temp2*asin(b/r)
      temp3=csubx*b*sqrt(abs(r*r-b*b))/2
      temp4=(pi/2)*((p-c)*r*rsubx-(r*r/2)*csubx)
      ft=temp1+temp2+temp3+temp4
      return
      end

      real function simpson(a,b,small,f)
      external f

      real a,b,f,small,h,oddsum,evensum,sum,I1,I2,sumf
      integer n

           n=4
           h=(b-a)/8.0
           evensum=sumf(a+2*h,2*h,n-2,f)
           oddsum=sumf(a+h,2*h,n-1,f)
           sum=oddsum+evensum
           I2=(h/3)*(f(a)+2*evensum+4*oddsum+f(b))
10         I1=I2
           n=2*n
           h=h/2.0
           evensum=sum
           oddsum=sumf(a+h,2*h,n-1,f)
           sum=oddsum+evensum
           I2=(h/3)*(f(a)+2*evensum+4*oddsum+f(b))
           if((abs(I1-I2).gt.small).and.(n.le.1024)) goto 10
           simpson=I2
      return
      end

      real function sumf(c,h,m,f)
c     sumf is is the sum of f(c+i*h), i=0,1,...,m.
      external f
      real c,h,f,tsum,targ
      integer m,i

           tsum=0.0
           targ=c
           do 10 i=0,m
                tsum=tsum+f(targ)
                targ=targ+h
10         continue
           sumf=tsum
      return
      end

      real function F12(p1,p2,r1,r2)
c     computes the force F1 on the downstream water face and the force
c     F2 on the upstream face, and returns F1+F2

      real  F1,F2,G,p1,p2,r1,r2

      F1=-G(p1,r1)
      print *, "F1 = ", F1
      F2=G(p2,r2)
      print *,"F2 = ", F2
      F12=F1+F2
      return
      end

      real function G(p,r)
c     computes the force on the wetted area at either pipe end

      real p,pi,r
      pi=3.1415926
      if (p .ge. (2.0*r)) then
           G=(p-r)*pi*r*r
      else if (p. le. 0.0) then
           G=0.0
      else
           G=(2.0/3.0)*exp((3.0/2.0)*log(r*r-(p-r)*(p-r)))
           G=G+(p-r)*r*r*asin((p-r)/r)
           G=G+(p-r)*(p-r)*sqrt(r*r-(p-r)*(p-r))
           G=G+((p-r)*r*r*pi/2.0)
      endif
      return
      end

      real function area(p,r)
c     returns the area of the circle x*x+(y-r)*(y-r)=r*r below the line
c     y=p.

      real p,pi,r
      pi=3.1415926

      if (p.le.0) then
           area =0.0
      endif
      if (p.ge.(2*r)) then
           area =pi*r*r
      endif
```