Solution to Problem VId.5.3:

```
c
      implicit real*8(a-h,o-z)
      dimension vol(2),h(2),k(2),t(2),v(2),x(2),a(2),dvdh(2),dvdp(2),
     1          am(2),regmul(2),timsrg(50),dmsrg(50),hsrg(50),
     2          qdoth(3),thih(3),tloh(3),taoh(3)
c
      data      time,htc21,vcd,ispray,ichoke/0.0,1500.,0.61,1,1/
c
c     Get Input Data
c
      call input(p,vol,pzrx,dmsui,hsui,htc21,dmsp,hsp,
     1           psph,pspl,nvalve,areav,phch,plch,iwall,uai,htce,
     2           ccw,ccf,delw,delf,akw,akf,row,rof,te,ttt,dt,np,
     3           nsrg,timsrg,dmsrg,hsrg,iheatr,qdoti,qdoth,thih,tloh,
     4           taoh)
c
c     Open Output Files
c
      open(12,file='out1.out')
      open(13,file='out2.out')
      open(14,file='out3.out')
      open(15,file='out4.out')
      open(16,file='out5.out')
c
c     Determine Initial Values
c
      call steady(p,hf,hg,k,t,v,x,a,vf,vg,dvdh,dvdp,ts,
     1            h,hsui,vol,volt,tw,tf,akw,akf,am,htce)
c
c     Set up Wall Geometry
c
      call geom(pzrx,volt,delw,delf,radb,radxw,radw,radxf,radf,height,
     1          row,rof,ccw,ccf,capcw,capcf)
c
c     Determine Water Level Based on Water Volume
c
      call wlev(height, vol(1),a(1),pzrx,s1,s2,colev)
      s1ua=s1
c
c     Determine Steady-state Wall Temperature
c
      kwall=1
      call wallht(kwall,iwall,uai,t,tw,tf,te,hf,hg,v(2),
     1            radb,radxw,radw,radxf,radf,height,
     2            capcw,capcf,s1,s2,akw,akf,dt,htce,
     3            qdotw,qdotw1,qdotw2)
c
c     Write Initial Conditions
c
      iout=1
      call output(p,vol,h,dmsui,hsui,qdoti,dmsp,dt,ttt,ip,np,
     1            time,am,t,ts,tw,tf,te,htc21,htce,qdotw1,qdotw2,
     2            x,a,s1,s2,dm12,dm21,dmsu,dm21wc,dmch,dm2s,
     3            iout,iflag,dm21ro,uai,nvalve,areav,iwall,
```

```
     4           psph,pspl,phch,plch)
c
c       -------------  START OF TRANSIENT  -------------------
c
1     continue
c
c     Counter for Print outs
c
      ip=ip+1
c
c     Null Transient for 1.00 seconds
c
      if(time.le.1.00) go to 109
c
c     Determine Surge Flow and Enthalpy
c
      if(nsrg.ne.0) call intrpl(nsrg,timsrg,dmsrg,hsrg,time,dmsui,hsui)
c
c     set up heaters
c
      call heater(iheatr,time,qdoti,qdoth,thih,tloh,taoh,t(1),p,qdot)
c
c Boil-off due to; Liquid Flush (Outsurge or PORV/SV),f eaters & Wall
c
      call surge(dmsui,dmsu,hsui,h(1),hsu)
      dm12=0.0
      call mfpool(dt,x(1),am(1),vf,vg,pzrx,ts,a(1),dm12)
c
c A simple Rain out model added.  Investigate more  8/31/92
c
      dm21ro=0.0
      if(x(2).lt.1.0.or.a(2).lt.1.0) dm21ro=(1.-x(2))*am(2)/dt
c
c Vapor condensation on the interface
c
      call intcond(am(2),t,ts,dmsu,pzrx,hf,hg,q21,dm21,htc21)
c
c Call Wall Heat Transfer To or From Wall
c
      qdotw1=0.0
      qdotw2=0.0
      dm21wc=0.0
      if(iwall.eq.0) go to 1000
      if(uai.eq.0.0) go to 71
      s1=vol(1)/pzrx
      s2=height-s1
c     qdotw1=(uai*(t(1)-te)/3600.00)
      qdotw1=0.0
      qdotw2=(uai*(t(2)-te)/3600.00)*(s1/s1ua)
      go to 74
71    continue
      call wlev(height,vol(1),a(1),pzrx,s1,s2,colev)
      kwall=2
      call wallht(kwall,iwall,uai,t,tw,tf,te,hf,hg,v(2),
     1            radb,radxw,radw,radxf,radf,height,
     2            capcw,capcf,s1,s2,akw,akf,dt,htce,
     3            qdotw,qdotw1,qdotw2)
```

```
c
c     Wall Condensation
c
74    continue
      if(qdotw2.gt.0.0) dm21wc=qdotw2/(hg-hf)
1000  continue
c
c     ckech for spray and spray condensation
c
      call spray(ispray,p,psph,pspl,h(2),hsp,hf,dmsp,dm1s,dm2s,
     1           en1s,en2s,am(2),dt)
c
c     ckech for PORV or SV
c
      call stats(am,vol,h,hch,istat,regmul)
      call choke(ichoke,nvalve,areav,p,phch,plch,hch,hf,dmch,vcd)
109   continue
c
c     perform transient calculation
c
      call trans(dt,dmsu,hsu,dm21,hf,dm12,hg,dm1s,dm2s,qdot,q21,
     1          en1s,en2s,v,h,dvdh,dvdp,
     2          dmch,hch,regmul,
     3          vol,volt,am,p,iflag,x,a,vf,vg,
     4          t,ts,qdotw1,qdotw2,dm21wc,dm21ro)
c
      time=time+dt
      iout=2
      call output(p,vol,h,dmsui,hsui,qdoti,dmsp,dt,ttt,ip,np,
     1            time,am,t,ts,tw,tf,te,htc21,htce,qdotw1,qdotw2,
     2            x,a,s1,s2,dm12,dm21,dmsu,dm21wc,dmch,dm2s,
     3            iout,iflag,dm21ro,uai,nvalve,areav,iwall,
     4            psph,pspl,phch,plch)
c
      if(time.le.ttt) go to 1
      stop
      end
cEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
c
      subroutine trans(dt,dmsu,hsu,dm21,hf,dm12,hg,dm1s,dm2s,qdot,q21,
     1                 en1s,en2s,v,h,dvdh,dvdp,
     2                 dmch,hch,reg,
     3                 vol,volt,am,p,iflag,x,a,vf,vg,
     4                 t,ts,qdw1,qdw2,dm21wc,dm21ro)
      implicit real*8(a-h,o-z)
      dimension reg(1),vol(1),am(1),h(1),x(1),a(1),v(1),dvdh(1),
     1          dvdp(1),t(1),alf(2),bet(2),hdot(2),vldot(2),k(1)
c
      data c/.18509/
c
      iflag=0
      hdot1=0.0
      hdot2=0.0
      vldot1=0.0
      vldot2=0.0
c
      alf(1)=dmsu+dm21-dm12+dm1s-reg(1)*dmch+dm21wc+dm21ro
```

```
      bet(1)=dmsu*hsu+dm21*hf+qdot+q21-dm12*hg+en1s-reg(1)*dmch*hch
     1    +dm21wc*hf-qdw1+dm21ro*hf
      alf(2)=-dm21+dm12+dm2s-reg(2)*dmch-dm21wc-dm21ro
      bet(2)=-dm21*hf-q21+dm12*hg+en2s-reg(2)*dmch*hch-dm21wc*hf-qdw2
     1       -dm21ro*hf
c
      s1=alf(1)*v(1)+alf(2)*v(2)
      s2=dvdh(1)*(bet(1)-alf(1)*h(1))+dvdh(2)*(bet(2)-alf(2)*h(2))
      s3=c*(dvdh(1)*vol(1)+dvdh(2)*vol(2))
      s4=am(1)*dvdp(1)+am(2)*dvdp(2)
c
      pdot=-(s1+s2)/(s3+s4)
c
      do 1 i=1,2
      if(am(i).gt.0.0.and.vol(i).gt.0.0)
     1 hdot(i)=((bet(i)-alf(i)*h(i)+c*vol(i)*pdot)/am(i))
      if(am(i).gt.0.0.and.vol(i).gt.0.0)
     1 vldot(i)=alf(i)*v(i)+am(i)*(dvdh(i)*hdot(i)+dvdp(i)*pdot)
1     continue
c
c
c
c    INTEGRATION OF MAJOR PARAMETERS
c
      p=p+pdot*dt
c
      do 2 i=1,2
      h(i)=h(i)+hdot(i)*dt
      vol(i)=vol(i)+vldot(i)*dt
      am(i)=am(i)+alf(i)*dt
2     continue
c
      volt=vol1+vol2
c
c    The following for stand-alone PZR model and an imposed outsurge
c
      if(am(1).lt.0.0.or.vol(1).lt.0.0.and.dmsu.lt.0.0) go to 999
      do 3 i=1,2
      if(am(i).gt.0.0.or.vol(i).gt.0.0)
     1 call prp(p,h(i),hf,hg,k(i),t(i),v(i),x(i),a(i),vf,vg,
     2                    dvdh(i),dvdp(i),ts)
3     continue
      return
999   iflag=8
      return
      end
```