Annexe 15

Listings des programmes

On trouve, dans les pages suivantes, les listings des programmes :

- initial.f
- pmultianneaux.f
- xpdessin.m

```
*************************************************************
*                    PROGRAMME INITIAL.F                    *
*************************************************************
* Le programme cree  une configuration de profils
* initiaux et la range dans le fichier profil.dat
* On a le choix entre plusieurs configurations initiales
* Il est a executer avant panneau.e


*  f77 -C initial.f  -o initial.e

*************************************************************
*                        SOUS ROUTINES                      *
*************************************************************

*-----------------------------------------------------------*
*                  echange avec le fichier                  *
*-----------------------------------------------------------*

      subroutine ouvrirf
      open(unit=30,name='profil.dat')
      return
      end

      subroutine fermerf
      close(30)
      return
      end

      subroutine saveprofil(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &        t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)



      do 1 j=1,n
      write(30,*)profil(1,j)

1     continue


      do 2 j=1,n
      write(30,*)profil(2,j)

2     continue

      do 3 j=1,n
      write(30,*)profil(3,j)

3     continue

      return
      end

*-----------------------------------------------------------*
*            routines de creation des profils               *
*-----------------------------------------------------------*


      subroutine ellipseyz(aa,bb,profil,centre)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &        t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),centre(3)
```

```
        real*8 aa,bb

        do 1 i=1,n
          profil(1,i)=0+centre(1)
          profil(2,i)=aa*cos((i-1)*2*3.1415926/n)+centre(2)
          profil(3,i)=bb*sin((i-1)*2*3.1415926/n)+centre(3)

1       continue

        return
        end

        subroutine ellipsexy(aa,bb,profil,centre)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
                 t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim),centre(3)
        real*8 aa,bb

        do 1 i=1,n
          profil(3,i)=0+centre(3)
          profil(2,i)=aa*cos((i-1)*2*3.1415926/n)+centre(2)
          profil(1,i)=bb*sin((i-1)*2*3.1415926/n)+centre(1)

1       continue

        return
        end

        subroutine trefle(aa,eps,profil)
* le premier parametre du  trefle est aa
* le second  parametre du  trefle est eps
* en polaire on a : r = ( 1 + eps * cos( 3*t ) ) *aa

        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
                ,t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)
        real*8 aa,eps

        do 1 i=1,n
          profil(1,i)=0
          profil(2,i)=aa*cos((i-1)*2*3.1415926/n)+aa*eps*cos((i-1)*2*3.1415926/n)*
                                                      cos(3*(i-1)*2*3.1415926/n)

          profil(3,i)=aa*sin((i-1)*2*3.1415926/n)+aa*eps*sin((i-1)*2*3.1415926/n)*
                                                      cos(3*(i-1)*2*3.1415926/n)
1       continue
        return
        end

        subroutine triangle(aa,profil)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
                ,t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)
        real*8 aa

        do 1 i=1,n
          profil(1,i)=0
          profil(2,i)=3*aa*cos((i-1)*2*3.1415926/n)+aa*cos(2*(i-1)*2*3.1415926/n)
          profil(3,i)=3*aa*sin((i-1)*2*3.1415926/n)-aa*sin(2*(i-1)*2*3.1415926/n)
1       continue
        return
        end

        subroutine lissa(profil)
```

```
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)


      do 1 i=1,n
        profil(1,i)=sin(2*((i-1)*2*3.1415926/n-3.1415926/4))
        profil(2,i)=2.*cos((i-1)*2*3.1415926/n)
        profil(3,i)=1.5*sin((i-1)*2*3.1415926/n)
1     continue
      return
      end


*-----------------------------------------------------------*
*     les differentes configurations de profils initiaux    *
*-----------------------------------------------------------*



      subroutine iellipse(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

* le demi grand axe de l'ellipse est aa
* le demi petit axe de l'ellipse est bb
      aa=2
      bb=1.5
      centre(1)=0
      centre(2)=0
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.03
      circul(1)=5
      debit(1)=0
      write(30,*)para(1)
      write(30,*)circul(1)
      write(30,*)debit(1)
      return
      end

      subroutine itrefle(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,eps


      aa=6
      eps=0.25

      call trefle(aa,eps,profil)
      call saveprofil(profil)

      para(1)=0.03
      circul(1)=5
      debit(1)=0

      write(30,*)para(1)
```

```
        return
        end

        subroutine itriangle(profil)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)


        aa=2.0

* en complexe on a : z = 3 * exp( i*t) + exp( -2*i*t )
        call triangle(aa,profil)
        call saveprofil(profil)

        para(1)=0.03
        circul(1)=5
        debit(1)=0

        write(30,*)para(1)
        write(30,*)circul(1)
        write(30,*)debit(1)
        return
        end

        subroutine ilissa(profil)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)


        call lissa(profil)
        call saveprofil(profil)

        para(1)=0.03
        circul(1)=5
        debit(1)=0

        write(30,*)para(1)
        write(30,*)circul(1)
        write(30,*)debit(1)
        return
        end

        subroutine icote(profil)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10
        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)
        real*8 aa,bb,centre(3)


        aa=1.5
        bb=1.5

        centre(1)=0
        centre(2)=-2.5
        centre(3)=0
        call ellipseyz(aa,bb,profil,centre)
        call saveprofil(profil)

        centre(1)=0
        centre(2)=2.5
```

```
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)


      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=5
      debit(2)=0

      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)
      return
      end

      subroutine iface1(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &         t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

      aa=1.5
      bb=1.5

      centre(1)=-0.25
      centre(2)=-0.375
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)

      centre(1)=0.25
      centre(2)=0.375
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=-5
      debit(2)=0

      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)

      return
      end

      subroutine iface2(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

      aa=1.5
      bb=1.5
      centre(1)=-0.25
      centre(2)=-1.30
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)

      centre(1)=0.25
      centre(2)=1.30
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=-5
      debit(2)=0

      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)


      return
      end


      subroutine iface3(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
               t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

      aa=1.5
      bb=1.5

      centre(1)=-0.25
      centre(2)=-1.375
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)

      centre(1)=0.25
      centre(2)=1.375
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=-5
      debit(2)=0

      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)

      return
      end


      subroutine ienlace(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

      aa=1.5
      bb=1.5

      centre(1)=0
      centre(2)=0
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)

      centre(1)=0
      centre(2)=-1.5
      centre(3)=0
      call ellipsexy(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=5
      debit(2)=0

      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)


      return
      end

      subroutine ienboite(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

      aa=1.5
      bb=0.5

      centre(1)=0
      centre(2)=0
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)

      centre(1)=0
      centre(2)=0
      centre(3)=0
      aa=0.5
      bb=1.5
      call ellipsexy(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.01
      circul(1)=5
      debit(1)=0

      circul(2)=5
      debit(2)=0
      write(30,*)para(1)

      write(30,*)circul(1)
      write(30,*)circul(2)
      write(30,*)debit(1)
      write(30,*)debit(2)


      return
      end

      subroutine ipellipse(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
               t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)
      real*8 aa,bb,centre(3)

* le demi grand axe de l'ellipse est aa
* le demi petit axe de l'ellipse est bb
      aa=2
      bb=1.5
      centre(1)=0
      centre(2)=0
      centre(3)=0
      call ellipseyz(aa,bb,profil,centre)
      call saveprofil(profil)
      para(1)=0.03
      circul(1)=5
      debit(1)=0
      write(30,*)para(1)
      write(30,*)circul(1)
      write(30,*)debit(1)
      return
      end

**************************************************************
*                         PRINCIPAL                          *
**************************************************************
      parameter(surdim=60,nsurdimt=600)
* la surdimension des matrices est surdim
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
              ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)


      real*8 profil(3,surdim)

      nvsurdim=surdim
      nvsurdimt=nsurdimt
      n=20
      write(*,*)'attention de bien avoir  n=',n,' <',surdim
* le nombre de points de discretisation est n


      call ouvrirf
      call iellipse(profil)
*     call itrefle(profil)
*     call itriangle(profil)
*     call ilissa(profil)
*     call icote(profil)
*     call iface1(profil)
*     call iface2(profil)
*     call iface3(profil)
*     call ienlace(profil)
*      call ienboite(profil)
*     call ipellipse(profil)
      call fermerf
      stop
```

```
        end
```

```
************************************************************
*             PROGRAMME PMULTIANNEAUX.F                    *
************************************************************
* Le programme utilise le  profil du fichier profil.dat
* et determine l'evolution de ce profil de vorticite
* Il enregistre le dernier profil dans profil.dat

* Il faut avoir rempli profil.dat avant de l'executer
* (a l'aide de initial.e  )
* Il n'y a pas de sortie graphique directe mais le
* programme cree le fichier pass.m qui sera repris par
* un programme matlab (xpdessin.m)
* qui se charge de faire le trace de l'evolution


*  f77 -C pmultianneaux.f  -o pmultianneaux.e

************************************************************
*                         SOUS ROUTINES                    *
************************************************************
*----------------------------------------------------------*
*                passage en matlab                         *
*----------------------------------------------------------*
        Subroutine  Ouvrir
        open(unit=20,name='pass.m')
        return
        end


        Subroutine  Fermer
        close(20)
        return
        end

        Subroutine Passmatrice(a,idim,jdim,nom,nvsurdimcnconf,nvsurdim)
        integer idim,jdim
        real*8 a(nvsurdimcnconf,nvsurdim)
        integer nvsurdimcnconf,nvsurdim
        character*(*) nom



        do 1 j=1,jdim
        write(20,*)nom,'(:,',j,')=['
        do 2 i=1,idim
        write(20,*) a(i,j)
2       continue
        write(20,*)'];'
1       continue

        return
        end

        Subroutine Passvecteur(a,idim,nom,nvsurdim)
        integer idim,i
        real*8 a(nvsurdim)
        character*(*) nom
        integer nvsurdim



        write(20,*)nom,'=['
        do 1 i=1,idim
        write(20,*) a(i)
1       continue
        write(20,*)'];'


        return
        end
```

```
      Subroutine passscalair(a,nom)
      integer a

      character*(*) nom



      write(20,*)nom,'=',a,';'



      return
      end

     Subroutine passscalairreal(a,nom)
      real*8 a

      character*(*) nom



      write(20,*)nom,'=',a,';'



      return
      end
*-------------------------------------------------------------*
*  chagement et stockage des profils initiaux et finaux       *
*-------------------------------------------------------------*
      subroutine ouvrirf
      open(unit=30,name='profil.dat')
      return
      end

      subroutine fermerf
      close(30)
      return
      end

      subroutine loadprofil(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)


      real*8 profil(3,nvsurdim)


      do 1 j=1,n
      read(30,*)profil(1,j)

1     continue


      do 2 j=1,n
      read(30,*)profil(2,j)

2     continue

      do 3 j=1,n
      read(30,*)profil(3,j)

3     continue

      return
      end

      subroutine ecrire(profil,total,nu)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &        t,epsi,ds,S0,t10
```

```
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     +       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),total(3,nvsurdimt)
      integer nu
      do 1 j=1,n
      do 1 i=1,3
      total(i,j+(nu-1)*n)=profil(i,j)
 1    continue
      return
      end

      subroutine lire(profil,total,nu)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     +         t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     +       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),total(3,nvsurdimt)
      integer nu
      do 1 j=1,n
      do 1 i=1,3
      profil(i,j)=total(i,j+(nu-1)*n)
 1    continue
      return
      end
      subroutine saveprofil(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     +        ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     +       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)



      do 1 j=1,n
      write(30,*)profil(1,j)

 1   continue


      do 2 j=1,n
      write(30,*)profil(2,j)

 2   continue

      do 3 j=1,n
      write(30,*)profil(3,j)

 3   continue

      return
      end



      subroutine sortie(profil)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     +        ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     +       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)


      do 1 j=1,n
```

```
1      continue


        do 2 j=1,n
        write(*,*)profil(2,j)

2      continue


        do 3 j=1,n
        write(*,*)profil(3,j)

3      continue
        return
        end
*------------------------------------------------------------*
*         operations sur des matrices et des vecteurs        *
*------------------------------------------------------------*


        subroutine egalmat(a,b)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &          ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &         ds,S0(10),t10(10)
        real*8 a(3,nvsurdim),b(3,nvsurdim)

        do 1 i=1,3
          do 1 j=1,n
           b(i,j)=a(i,j)
1         continue
        return
        end

        subroutine egalmatt(a,b)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
                ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)
        real*8 a(3,nvsurdimt),b(3,nvsurdimt)

        do 1 i=1,3
          do 1 j=1,n*nbra
           b(i,j)=a(i,j)
1         continue
        return
        end

        subroutine affectmatvect(a,b,i)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
                 t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 a(3,nvsurdim),b(3)

        do 1 j=1,3
         b(j)=a(j,i)
1       continue
        return
        end

        subroutine affectvectmat(a,b,i)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
                 t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)
```

```
      real*8 b(3,nvsurdim),a(3)

      do 1 j=1,3
       b(j,i)=a(j)
1     continue
      return
      end

************ addition de 2 vecteurs *************
      subroutine addition(a,b,c)
      real*8 a(3),b(3),c(3)

      do 1 i=1,3
      c(i)=a(i)+b(i)
1     continue
      return
      end

      subroutine soustraction(a,b,c)
      real*8 a(3),b(3),c(3)

      do 1 i=1,3
      c(i)=a(i)-b(i)
1     continue
      return
      end

      subroutine egalvect(a,b)
      real*8 a(3),b(3)

      do 1 i=1,3
      b(i)=a(i)
1     continue
      return
      end

      subroutine prodscal(k,a,b)
      real*8 a(3),b(3),k

      do 1 i=1,3
      b(i)=k*a(i)
1     continue

      return
      end

      subroutine norme(a,k)
      real*8 a(3),k

      k=a(1)*a(1)+a(2)*a(2)+a(3)*a(3)
      k=sqrt(k)

      return
      end

      subroutine scalaire(a,b,k)
      real*8 a(3),b(3),k

      k=a(1)*b(1)+a(2)*b(2)+a(3)*b(3)
      return
      end

      subroutine produitvectoriel(a,b,c)
      real*8 a(3),b(3),c(3)

      c(1)=a(2)*b(3)-a(3)*b(2)
      c(2)=a(3)*b(1)-a(1)*b(3)
      c(3)=a(1)*b(2)-a(2)*b(1)
      return
      end
      subroutine max(a,b)
      real*8 a,b

      if (a.gt.b) then
       b=a
      endif
      return
      end

      subroutine min(a,b)
      real*8 a,b

      if (a.lt.b) then
       b=a
      endif
      return
      end

*-----------------------------------------------------------*
*                sousroutines de discretisation             *
*                   et fonctions de calcul                  *
*-----------------------------------------------------------*
      subroutine dXs(a,c,d)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
              ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)
      real*8 a(3),c(3),d(3)

      call soustraction(c,a,d)
      call prodscal(0.5D00/ds,d,d)

      return
      end


      subroutine sigma(a,c,l)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
               t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)
      real*8 a(3),c(3),d(3),l

      call dXs(a,c,d)
      call norme(d,l)
      return
      end

      subroutine indice(i,ii,jj,kk)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
              ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      integer i,ii,jj,kk

      if (i.eq.1) then
          ii=n
          jj=1
          kk=2
      endif
      if (i.eq.n) then
          ii=n-1
          jj=n
          kk=1
      endif
      if ((i.ne.1).and.(i.ne.n)) then
          ii=i-1
```

```
            jj=i
            kk=i+1
        endif


        return
        end

        subroutine discret(i,profil,a,b,c)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &          t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &         ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim)
        integer i,ii,jj,kk
        real*8 a(3),b(3),c(3)

        call indice(i,ii,jj,kk)



        a(1)=profil(1,ii)
        a(2)=profil(2,ii)
        a(3)=profil(3,ii)

        b(1)=profil(1,jj)
        b(2)=profil(2,jj)
        b(3)=profil(3,jj)

        c(1)=profil(1,kk)
        c(2)=profil(2,kk)
        c(3)=profil(3,kk)

        return
        end

        subroutine longueur(profil,S)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &          ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &         ds,S0(10),t10(10)
        integer i
        real*8 profil(3,nvsurdim),l,S,a(3),b(3),c(3)

        S=0
        do 1 i=1,n


        call discret(i,profil,a,b,c)
        call sigma(a,c,l)
        S=S+l
1       continue
        S=S*ds
        return
        end


        subroutine dXss(a,b,c,d)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &          ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &         ds,S0(10),t10(10)
        real*8 a(3),b(3),c(3),d(3)

        call addition(c,a,d)
        call soustraction(d,b,d)
        call soustraction(d,b,d)
        call prodscal(1/ds/ds,d,d)
```

```
      return
      end

      subroutine coefficient(ni,profil,t1,coeff,cv,cw)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)
      real*8 profil(3,nvsurdim),t1,S,t11,delta,euler,cv,cw,coeff,deux
      integer ni

      call longueur(profil,S)
      t11=t1+t*S
      delta=4*viscosite*t11/S
      delta=sqrt(delta)
      euler=0.577215
      deux=2
      cv=log(para(ni)/delta)+0.5*(1+euler-log(deux))
      cw=S0(ni)/S
      cw=cw*cw*cw*cw
      cw=cw*(-2)*debit(ni)*debit(ni)/circul(ni)/circul(ni)*para(ni)*para(ni)/delta
                                                                         /delta
      coeff=cv+cw+log(1/para(ni))
      return
      end
*------------------------------------------------------------*
*                  sousroutine de calcul                     *
*                geometrique sur un profil                   *
*------------------------------------------------------------*
      subroutine caracteristique(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
                                 courbure,tors,dcourb)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
               t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),dprofil(3,nvsurdim),d2profil(3,nvsurdim),
             kb(3,nvsurdim),tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),
             fsigma(nvsurdim),courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim)
      real*8 a(3),b(3),c(3),d(3),e(3),f(3),g(3),h(3),m(3),no(3),l
      integer i,ii,jj,kk

      do 1 i=1,n
           call discret(i,profil,a,b,c)
           call dXs(a,c,d)
           call dXss(a,b,c,e)
           call norme(d,l)
           fsigma(i)=l
           call prodscal(1/l,d,f)
           l=l*l*l
           l=1/l
           call prodscal(l,d,g)
           call produitvectoriel(g,e,h)
           call norme(h,courbure(i))
           call prodscal(1/courbure(i),h,m)
           call produitvectoriel(m,f,no)
           call affectvectmat(d,dprofil,i)
           call affectvectmat(e,d2profil,i)
           call affectvectmat(f,tang,i)
           call affectvectmat(h,kb,i)
           call affectvectmat(m,binorm,i)
           call affectvectmat(no,norm,i)
1     continue

       do 2 i=1,n
           call discret(i,binorm,a,b,c)
           call dXs(a,c,d)
           call affectmatvect(norm,e,i)
           call scalaire(d,e,l)
           tors(i)=-1/fsigma(i)*l
```

```
 2       continue

          do 3 i=1,n
               call indice(i,ii,jj,kk)
               dcourb(i)=0.5*(courbure(kk)-courbure(ii))/ds
 3       continue

         return
         end

*---------------------------------------------------------*
*                sousroutines exprimant les                *
*                    fonctions de calcul                   *
*---------------------------------------------------------*
         subroutine permut(i,v,w)
         common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
                 ,t,epsi,ds,S0,t10

         real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
                ds,S0(10),t10(10)
         integer i,v(nvsurdim),w(nvsurdim)
         integer ll,j

         do 1 j=1,n
              ll=j+i-1
              if (ll.gt.n) then
                             ll=ll-n
              endif
              v(j)=ll
 1       continue
         w(1)=v(1)
         do 2 j=2,n
              w(j)=v(n-j+2)
 2       continue
         return
         end

         subroutine lambda(profil,fsigma,v,lamb)
         common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
                 ,t,epsi,ds,S0,t10

         real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
                ds,S0(10),t10(10)
         real*8 profil(3,nvsurdim),fsigma(nvsurdim),lamb(nvsurdim)
         real*8 cst,l
         integer i,v(nvsurdim)

         lamb(1)=0
         cst=fsigma(v(1))
         do 1 i=2,n/2+1
              l=fsigma(v(i))
              lamb(i)=lamb(i-1)+cst/2+l/2
              cst=l
 1       continue
         do 2 i=1,n/2+1
              lamb(i)=lamb(i)*ds
 2       continue
         return
         end

         subroutine fonctg(i,profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
                           courbure,tors,dcourb,v,lamb,gvect)
         common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
                  t,epsi,ds,S0,t10

         real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
                ds,S0(10),t10(10)

         real*8 profil(3,nvsurdim),dprofil(3,nvsurdim),d2profil(3,nvsurdim),
                kb(3,nvsurdim),tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),
                fsigma(nvsurdim),courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim),
                lamb(nvsurdim)
```

```
      integer i,v(nvsurdim)
      real*8 gvect(3)
      real*8 a(3),b(3),d(3),e(3),f(3),g(3)
      real*8 k

      call affectmatvect(profil,a,v(i))
      call affectmatvect(profil,b,v(1))
      call affectmatvect(dprofil,d,v(i))
      call soustraction(a,b,e)
      call norme(e,k)
      k=k*k*k
      k=1/k
      call prodscal(k,e,e)
      call produitvectoriel(e,d,f)

      call affectmatvect(kb,g,v(1))
      k=abs(lamb(i))
      k=1/k/2*fsigma(v(i))
      call prodscal(k,g,g)

      call soustraction(f,g,gvect)

      return
      end

      subroutine integral(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                    courbure,tors,dcourb,v,lamb,fg)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &         t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),dprofil(3,nvsurdim),d2profil(3,nvsurdim),
     &       kb(3,nvsurdim),tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),
     &       fsigma(nvsurdim),courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim),
     &       lamb(nvsurdim),fg(3)
      integer v(nvsurdim)

      integer i
      real*8 cstvect(3),gvect(3)

      fg(1)=0
      fg(2)=0
      fg(3)=0
      call fonctg(2,profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,courbure,tors,
     &            dcourb,v,lamb,cstvect)

      do 1 i=3,n/2+1
      call fonctg(i,profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,courbure,tors,
     &            dcourb,v,lamb,gvect)
      call addition(gvect,cstvect,cstvect)
      call prodscal(0.5D00,cstvect,cstvect)
      call addition(fg,cstvect,fg)
      call egalvect(gvect,cstvect)
1     continue
      call prodscal(ds,fg,fg)

      return
      end

      subroutine fonctglocal(i,profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                       courbure,tors,dcourb,gvect)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
     &         t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),dprofil(3,nvsurdim),d2profil(3,nvsurdim),
     &       kb(3,nvsurdim),tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),
     &       fsigma(nvsurdim),courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim)
```

```
      integer i
      real*8 gvect(3),cstss
      real*8 a(3),b(3),c(3),d(3)


      call affectmatvect(kb,a,i)
      call prodscal(tors(i),a,a)
      call affectmatvect(norm,b,i)
      call prodscal(dcourb(i)/fsigma(i),b,b)
      call addition(a,b,b)
      call affectmatvect(tang,c,i)
      call produitvectoriel(c,b,d)
      cstss=0.3333333333333
      call prodscal(cstss,d,gvect)
      return
      end

      subroutine integrallocal(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
                             courbure,tors,dcourb,v,lamb,fg)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
              t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim),dprofil(3,nvsurdim),d2profil(3,nvsurdim),
             kb(3,nvsurdim),tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),
             fsigma(nvsurdim),courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim),
             fg(3)
      integer v(nvsurdim)

      real*8 cstvect(3),gvect(3),lamb(nvsurdim)





      call fonctglocal(v(1),profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
                         courbure,tors,dcourb,gvect)

      call prodscal(lamb(2),cstvect,fg)

      return
      end

      subroutine fQ2(i,profil,b,f)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
             ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
             ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)

      integer i

      real*8 a(3),b(3),d(3),e(3),f(3)
      real*8 aa(3),bb(3),cc(3)
      real*8 k

      call affectmatvect(profil,a,i)

      call discret(i,profil,aa,bb,cc)
      call dXs(aa,cc,d)

      call soustraction(a,b,e)
      call norme(e,k)
      k=k*k*k
      k=1/k
      call prodscal(k,e,e)
      call produitvectoriel(e,d,f)
      return
      end

      subroutine termeQ2(ni,profil,b,Q2)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)

      real*8 profil(3,nvsurdim)

      integer ni
      real*8 Q2(3),b(3)
      integer i
      real*8 gvect(3)

      Q2(1)=0
      Q2(2)=0
      Q2(3)=0


      do 1 i=1,n

      call fQ2(i,profil,b,gvect)

      call addition(gvect,Q2,Q2)

1     continue
      call prodscal(alpha(ni)*ds,Q2,Q2)

      return
      end

      subroutine stermeQ2(ni,total,profil,b,SQ2)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10
      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)
      real*8 total(3,nvsurdimt)
      real*8 profil(3,nvsurdim)

      integer ni
      real*8 SQ2(3),Q2(3),b(3)
      integer i


      SQ2(1)=0
      SQ2(2)=0
      SQ2(3)=0


      do 1 i=1,nbra
      if (i.ne.ni) then
                 call lire(profil,total,i)
                 call termeQ2( i,profil,b,Q2)
                 call addition(SQ2,Q2,SQ2)
      endif
1     continue


      return
      end
*------------------------------------------------------*
*           sousroutine exprimant une equation         *
*                  du schema numerique                 *
*------------------------------------------------------*
      subroutine schema(ni,x0,Q2,coeff,profil,i,d,vit,err,dprofil,d2profil,kb,tang,
     &                  norm,binorm,fsigma,courbure,tors,dcourb,lambp,lambm,v,w)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &        ds,S0(10),t10(10)
      real*8 err,coeff
      real*8 x0(3),profil(3,nvsurdim),d(3),vit(3),Q2(3)
      integer i,v(nvsurdim),w(nvsurdim),ni

      real*8 dprofil(3,nvsurdim),d2profil(3,nvsurdim),kb(3,nvsurdim),tang(3,nvsurdim)
     &       ,norm(3,nvsurdim),binorm(3,nvsurdim),fsigma(nvsurdim),courbure(nvsurdim)
     &       ,tors(nvsurdim),dcourb(nvsurdim)
      real*8 lambp(nvsurdim),lambm(nvsurdim),Sp,Sm
      real*8 k,a(3),b(3),c(3),fg(3),fgp(3),fgm(3),Qf(3),Qetoile(3)


      call caracteristique(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                     courbure,tors,dcourb)
      call permut(i,v,w)

      call lambda(profil,fsigma,v,lambp)
      call lambda(profil,fsigma,w,lambm)

      Sp=lambp(n/2+1)
      Sm=lambm(n/2+1)
      k=Sp*Sm
      k=2*sqrt(k)
      k=log(k)-1
      call affectmatvect(kb,a,i)
      call prodscal(k,a,a)

      call integral(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,courbure,tors
     &          ,dcourb,v, lambp, fgp)

      call integral(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,courbure,tors
     &          ,dcourb,w, lambm, fgm)
      call addition(fgp,fgm,fg)

      call integrallocal(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                   courbure,tors,dcourb,v, lambp, fgp)
      call integrallocal(profil,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                   courbure,tors,dcourb,w, lambm, fgm)

      call soustraction (fgp,fgm,b)
      call addition(b,fg,fg)

      call addition(a,fg,Qf)

      call prodscal(alpha(ni),Qf,Qf)
      call addition(Qf,Q2,Qf)
      call affectmatvect(tang,c,i)
      call scalaire(Qf,c,k)
      call prodscal(k,c,c)
      call soustraction(Qf,c,Qetoile)

*     if (i.eq.1) then
*               write(*,*)'i',i
*                write(*,*)'Qet',Qetoile(1)
*                 write(*,*)'Qet',Qetoile(2)
*                write(*,*)'Qet',Qetoile(3)
*      endif

      call affectmatvect(kb,c,i)

      call prodscal(coeff*alpha(ni),c,c)

      call addition(Qetoile,c,d)
      call egalvect(d,vit)

      call prodscal(t,d,d)
      call addition(d,x0,d)

      call affectmatvect(profil,b,i)
      call soustraction(d,b,e)
      call norme(e,k)
      call max(k,err)
      return
      end


*--------------------------------------------------------*
*                sousroutine parcourant toutes les         *
*                   equations du schema numerique          *
*--------------------------------------------------------*
      subroutine recurrence(ni,profil0,t1,total,profil,profils,vitesse,err,dprofil,
     &                      d2profil,kb,tang,norm,binorm,fsigma,courbure,tors,dcourb,
     &                      lambp,lambm,v,w,cv,cw)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10


      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)
      real*8 total(3,nvsurdimt)
      real*8 profil(3,nvsurdim),profils(3,nvsurdim),vitesse(3,nvsurdim),
     &       profil0(3,nvsurdim),err
      real*8 d(3),vit(3),x0(3),t1,coeff,cv,cw,SQ2(3),b(3)

      integer v(nvsurdim),w(nvsurdim),ni

      real*8 dprofil(3,nvsurdim),d2profil(3,nvsurdim),kb(3,nvsurdim),
     &       tang(3,nvsurdim),norm(3,nvsurdim),binorm(3,nvsurdim),fsigma(nvsurdim),
     &       courbure(nvsurdim),tors(nvsurdim),dcourb(nvsurdim)
      real*8 lambp(nvsurdim),lambm(nvsurdim)

      err=0

      call lire(profil,total,ni)
      call coefficient(ni,profil,t1,coeff,cv,cw)


      do 2 i=1,n
      x0(1)=profil0(1,i)
      x0(2)=profil0(2,i)
      x0(3)=profil0(3,i)

      b(1)=profil(1,i)
      b(2)=profil(2,i)
      b(3)=profil(3,i)


      call stermeQ2(ni,total,profils,b,SQ2)

      call schema(ni,x0,SQ2,coeff,profil,i,d,vit,err,dprofil,d2profil,kb,tang,norm,
     &            binorm,fsigma,courbure,tors,dcourb,lambp,lambm,v,w)

      profil(1,i)=d(1)
      profil(2,i)=d(2)
      profil(3,i)=d(3)

      vitesse(1,i)=vit(1)
      vitesse(2,i)=vit(2)
      vitesse(3,i)=vit(3)
2     continue

      call ecrire(profil,total,ni)

      return
      end

      subroutine lambdaetendu(profil,fsigma,v,lamb)
      common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
     &        ,t,epsi,ds,S0,t10

      real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
     &       ds,S0(10),t10(10)
```

```
        real*8 profil(3,nvsurdim),fsigma(nvsurdim),lamb(nvsurdim)
        real*8 cst,l
        integer i,v(nvsurdim)

        lamb(1)=0
        cst=fsigma(v(1))
        do 1 i=2,n
             l=fsigma(v(i))
             lamb(i)=lamb(i-1)+cst/2+l/2
             cst=l
1       continue
        do 2 i=1,n
             lamb(i)=lamb(i)*ds
2       continue
        return
        end

        subroutine rayon(courbure,rmin)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)
        real*8 courbure(nvsurdim),g,rmin

        g=0
        do 1 i=1,n
          call max(courbure(i),g)
1       continue
        rmin=1/g
        return
        end

        subroutine distance(a,b,dis)
        real*8 a(3),b(3),c(3),dis

        call soustraction(b,a,c)
        call norme(c,dis)


        return
        end

        subroutine distance12(profil,profils,dmin)

        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 profil(3,nvsurdim),profils(3,nvsurdim),dmin,dis
        real*8 a(3),b(3)

        dmin=10000
        do 1 i=1,n
        call affectmatvect(profil,a,i)
        do 1 j=1,n
        call affectmatvect(profils,b,j)
        call distance(a,b,dis)
        call min(dis,dmin)
1       continue
        return
        end



        subroutine distancemin(profil,fsigma,rmin,dmin,v,w,lamb1,lamb2)
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha
               ,t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi
```

```
                ds,S0(10),t10(10)
         real*8 profil(3,nvsurdim),rmin,dmin,a(3),b(3),dis,cst
         integer v(nvsurdim),w(nvsurdim)
         real*8 lamb1(nvsurdim),lamb2(nvsurdim),discurve

         dmin=1000
         cst=3.14*rmin

         do 1 i=1,n-1
            call affectmatvect(profil,a,i)
            call permut(i,v,w)
            call lambdaetendu(profil,fsigma,v,lamb1)
            call lambdaetendu(profil,fsigma,w,lamb2)
            do 2 j=2,n-i+1
              call affectmatvect(profil,b,v(j))
              call distance(a,b,dis)

              if (lamb1(j).gt.lamb2(n-j+2)) then
                                          discurve=lamb2(n-j+2)
                                     else
                                         discurve=lamb1(j)
              endif

              if (discurve.gt.cst) then
                                     call min(dis,dmin)
              endif

2        continue

1        continue


         return
         end

*************************************************************
*               PROGRAMME    PRINCIPAL                      *
*************************************************************
        parameter(pi=3.1415926,surdim=61,nsurdimt=600,surdimcnbrconf=40,
                  nsurdimcnbrconft=400)
        integer conteur,cont
        common/ /nbra,n,nvsurdim,nvsurdimt,para,circul,debit,epaisseur,viscosite,alpha,
                 t,epsi,ds,S0,t10

        real*8 para(10),circul(10),debit(10),epaisseur(10),viscosite,alpha(10),t,epsi,
               ds,S0(10),t10(10)

        real*8 total(3,nsurdimt),total0(3,nsurdimt),vitessetotale(3,nsurdimt)
        real*8 tangtotal(3,nsurdimt),normtotal(3,nsurdimt),binormtotal(3,nsurdimt)
        real*8 profil1(3,surdim),profil2(3,surdim),vitesse1(3,surdim),
               profil01(3,surdim),err,t1(10),S(10)

        real*8 dprofil(3,surdim),d2profil(3,surdim),kb(3,surdim),tang(3,surdim),
               norm(3,surdim),binorm(3,surdim)
        real*8 fsigma(surdim),courbure(surdim),tors(surdim),dcourb(surdim)
        real*8 lambp(surdim),lambm(surdim)
        real*8 lamb1(surdim),lamb2(surdim)

        real*8 xx0(10,surdim),yy0(10,surdim),zz0(10,surdim)
        real*8 txx0(10,surdim),tyy0(10,surdim),tzz0(10,surdim)
        real*8 nxx0(10,surdim),nyy0(10,surdim),nzz0(10,surdim)
        real*8 bxx0(10,surdim),byy0(10,surdim),bzz0(10,surdim)
        real*8 vxx0(10,surdim),vyy0(10,surdim),vzz0(10,surdim)
        real*8 xx(surdimcnbrconf,nsurdimt),yy(surdimcnbrconf,nsurdimt),
               zz(surdimcnbrconf,nsurdimt)
        real*8 txx(surdimcnbrconf,nsurdimt),tyy(surdimcnbrconf,nsurdimt),
               tzz(surdimcnbrconf,nsurdimt)
        real*8 nxx(surdimcnbrconf,nsurdimt),nyy(surdimcnbrconf,nsurdimt),
               nzz(surdimcnbrconf,nsurdimt)
        real*8 bxx(surdimcnbrconf,nsurdimt),byy(surdimcnbrconf,nsurdimt),
               bzz(surdimcnbrconf,nsurdimt)
        real*8 vxx(surdimcnbrconf,nsurdimt),vyy(surdimcnbrconf,nsurdimt),
     &       vzz(surdimcnbrconf,nsurdimt)


      integer v(surdim),w(surdim),nbrdt,nbrconf,cnb,nconf,vv(surdim),ww(surdim)

      real*8 rmin,delta(10),dmin,para1(10,surdimcnbrconf),para2(10,surdimcnbrconf)

      real*8 para1min(10),para2min(10),para1var(10),para2var(10),para3min,
     &       para3(10,nsurdimcnbrconft),para3var(10,10)
      real*8 cv,cw

* la surdimension des matrices est surdim



*------------------------------------------------------------*
*       signification des constantes et des variables        *
*------------------------------------------------------------*
*  leurs valeurs est a modifier selon le jeu de valeurs que l'on veut faire:


*  n      : nombre de points de discretisation
*  t      : valeur du pas de temps
*  nbrdt  : nombre de pas de temps a effectuer
*  nbrconf : nombre de profil (configurations)  que l'on veut avoir
*  epsi   : valeur de l'erreur pour laquelle on arrete la recherche de zero

*  para   : petit parametre
*  circul: circulation
*  debit  : valeur de m0
*  epaisseur=para*sqrt(4*circul) :on choisit donc t20=1


*------------------------------------------------------------*
*                affectations initiale des                   *
*                  valeurs des variables                     *
*------------------------------------------------------------*
      nvsurdim=surdim
      nvsurdimcnbrconf=surdimcnbrconf
      nvsurdimt=nsurdimt
      nvsurdimcnbrconft=nsurdimcnbrconft

      nbra=1
      n=20
      nbrdt=700
      nbrconf=24
      write(*,*)'attention de bien avoir  n=',n,' <',surdim
      write(*,*)'attention de bien avoir  nbrconf=',nbrconf,' <',surdimcnbrconf

      cnb=int(nbrdt/nbrconf)
      nconf=0



      call ouvrirf
      do 77 j=1,nbra
      call loadprofil(profil1)
      call ecrire(profil1,total,j)
77    continue
      read(30,*)para(1)

      do 78 j=1,nbra
      read(30,*)circul(j)
78    continue
      do 79 j=1,nbra
      read(30,*)debit(j)

79    continue

      call fermerf

      viscosite=para(1)*para(1)*abs(circul(1))
      do 80 j=2,nbra

      para(j)=para(1)*sqrt(abs(circul(1)/circul(j)))
80    continue

      do 81 j=1,nbra
      epaisseur(j)=para(j)*sqrt(4*abs(circul(1)))
      alpha(j)=circul(j)/4/pi
81    continue
      t=0.02
      epsi=0.001
      ds=2*pi/n

*----------------------------------------------------------*
*               signification  des variables               *
*----------------------------------------------------------*
*  viscosite: viscosite
*  epaisseur: epaisseur
*  ds       : pas de discretisation en espace
*  cnb      : pas entre les differentes configurations enregistrees
*  profil   : matrice dans laquelle est mis les coordonnees x y et z
*             des noeuds de discretisation d'une ligne centrale
*  xx0,yy0,zz0 : matrices dans lesquelles sont mises les coordonnees x y
*                et z des noeuds de discretisation de la configuration initiale
*  xx ,yy ,zz  : matrices dans lesquelles sont mis les coordonnees x y
*                et z des noeuds de discretisation des cnconf configurations


*----------------------------------------------------------*
*        passage en matlab de la configuration initiale    *
*----------------------------------------------------------*


      do 82 j=1,nbra
      call lire(profil1,total,j)
      call caracteristique(profil1,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
     &                     courbure,tors,dcourb)
      do 83 i=1,n
      xx0(j,i)=profil1(1,i)
      yy0(j,i)=profil1(2,i)
      zz0(j,i)=profil1(3,i)
      txx0(j,i)=tang(1,i)
      tyy0(j,i)=tang(2,i)
      tzz0(j,i)=tang(3,i)
      nxx0(j,i)=norm(1,i)
      nyy0(j,i)=norm(2,i)
      nzz0(j,i)=norm(3,i)
      bxx0(j,i)=binorm(1,i)
      byy0(j,i)=binorm(2,i)
      bzz0(j,i)=binorm(3,i)
83    continue
      xx0(j,n+1)=profil1(1,1)
      yy0(j,n+1)=profil1(2,1)
      zz0(j,n+1)=profil1(3,1)
      txx0(j,n+1)=tang(1,1)
      tyy0(j,n+1)=tang(2,1)
      tzz0(j,n+1)=tang(3,1)
      nxx0(j,n+1)=norm(1,1)
      nyy0(j,n+1)=norm(2,1)
      nzz0(j,n+1)=norm(3,1)
      bxx0(j,n+1)=binorm(1,1)
      byy0(j,n+1)=binorm(2,1)
      bzz0(j,n+1)=binorm(3,1)
82    continue



      call Ouvrir
      call passscalair(nbra,'nbra')
      call passscalair(n,'nbr')
      call passscalair(nbrconf,'nbrconf')
      call passscalair(nbrdt,'nbrdt')
      call passvecteur(para,nbra,'para',10)
      call passvecteur(circul,nbra,'circul',10)
      call passvecteur(debit,nbra,'debit',10)
      call passscalairreal(t,'t')
      call passscalairreal(epsi,'epsi')
      call passmatrice(xx0,nbra,n+1,'xx0',10,nvsurdim)
      call passmatrice(yy0,nbra,n+1,'yy0',10,nvsurdim)
      call passmatrice(zz0,nbra,n+1,'zz0',10,nvsurdim)
      call passmatrice(txx0,nbra,n+1,'txx0',10,nvsurdim)
      call passmatrice(tyy0,nbra,n+i,'tyy0',10,nvsurdim)
      call passmatrice(tzz0,nbra,n+1,'tzz0',10,nvsurdim)
      call passmatrice(nxx0,nbra,n+1,'nxx0',10,nvsurdim)
      call passmatrice(nyy0,nbra,n+1,'nyy0',10,nvsurdim)
      call passmatrice(nzz0,nbra,n+1,'nzz0',10,nvsurdim)
      call passmatrice(bxx0,nbra,n+1,'bxx0',10,nvsurdim)
      call passmatrice(byy0,nbra,n+1,'byy0',10,nvsurdim)
      call passmatrice(bzz0,nbra,n+1,'bzz0',10,nvsurdim)



      do 84 j=1,nbra
      call lire(profil1,total,j)
      call longueur(profil1,S0(j))


      t10(j)=S0(j)*epaisseur(j)*epaisseur(j)/4/viscosite


      write(*,*)'S0(',j,')='
      write(*,*)S0(j)

      write(*,*)'t10(',j,')='
      write(*,*)t10(j)

      t1(j)=t10(j)
84    continue
      call passvecteur(t10,nbra,'t10',10)
      call passvecteur(S0,nbra,'S0',10)



      conteur=1
      do 85 j=1,nbra
      para1min(j)=100000
      para2min(j)=100000

85    continue
      para3min=100000
*------------------------------------------------------------*
*                boucle d'evolution du temps                 *
*------------------------------------------------------------*
      do 2 j=1,nbrdt


      err=100

      call egalmatt(total,total0)


      cont=0
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*
*     boucle interne d'iteration de la reherche  de zero     *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*


3       if  ((err.ge.epsi).and.(cont.ne.11)) then

         do 86 nu=1,nbra
```

```
          call recurrence(nu,profil01,t1(nu),total,profil1,profil2,vitesse1,err,
                          dprofil,d2profil,kb,tang,norm,binorm,fsigma,courbure,tors,
                          dcourb,lambp,lambm,v,w,cv,cw)


          call ecrire(vitesse1,vitessetotale,nu)

86        continue

          cont=cont+1
          goto 3
         endif

         if (cont.eq.11) then

            call egalmatt(total,total0)
            t=t/2
            write(*,*)'stop'
            write(20,*)'stop'
            write(20,*)j
            write(*,*)j
            err=100
            cont=0
            goto 3
         endif

       do 87 nu=1,nbra
       call lire(profil1,total,nu)
       call longueur(profil1,S(nu))
       t1(nu)=t1(nu)+t*S(nu)

87     continue


       do 88 nu=1,nbra
       call lire(profil1,total,nu)
       call caracteristique(profil1,dprofil,d2profil,kb,tang,norm,binorm,fsigma,
                            courbure,tors,dcourb)
       call ecrire(tang,tangtotal,nu)
       call ecrire(norm,normtotal,nu)
       call ecrire(binorm,binormtotal,nu)
       call rayon(courbure,rmin)
       delta(nu)=sqrt(4*viscosite*t1(nu)/S(nu))
       call distancemin(profil1,fsigma,rmin,dmin,vv,ww,lamb1,lamb2)
       para1var(nu)=rmin/delta(nu)
       para2var(nu)=dmin/2/delta(nu)

       call min(para1var(nu),para1min(nu))
       call min(para2var(nu),para2min(nu))
88     continue

       do 100 nu=1,nbra
       call lire(profil1,total,nu)
       do 100 nuu=1,nbra
       call lire(profil2,total,nuu)
       call distance12(profil1,profil2,dmin)
       para3var(nu,nuu)=dmin/(delta(nu)+delta(nuu))
       call min(para3var(nu,nuu),para3min)
100    continue

       if (j.eq.1) then

               do 89 nu=1,nbra
               call lire(vitesse1,vitessetotal,nu)

               do 90 il=1,n
               vxx0(nu,il)=vitesse1(1,nu)
               vyy0(nu,il)=vitesse1(2,nu)
               vzz0(nu,il)=vitesse1(3,nu)
```

```
 90                     continue
                        vxx0(nu,n+1)=vitesse1(1,1)
                        vyy0(nu,n+1)=vitesse1(2,1)
                        vzz0(nu,n+1)=vitesse1(3,1)

 89            continue
               call passmatrice(vxx0,nbra,n+1,'vxx0',10,nvsurdim)
               call passmatrice(vyy0,nbra,n+1,'vyy0',10,nvsurdim)
               call passmatrice(vzz0,nbra,n+1,'vzz0',10,nvsurdim)

               endif

               if (conteur.eq.cnb) then
                write(*,*)'j'
                write(*,*)j



               nconf=nconf+1

               do 91 nu=1,nbra
               para1(nu,nconf)=para1var(nu)
               para2(nu,nconf)=para2var(nu)
 91            continue

               do 101 nu=1,nbra
               do 101 nuu=1,nbra
               para3(nu,(nconf-1)*nbra+nuu)=para3var(nu,nuu)
 101           continue


               do 92 nu=1,nbra
               do 93 il=1,n
               xx(nconf,(nu-1)*(n+1)+il)=total(1,(nu-1)*n+il)
               yy(nconf,(nu-1)*(n+1)+il)=total(2,(nu-1)*n+il)
               zz(nconf,(nu-1)*(n+1)+il)=total(3,(nu-1)*n+il)
               txx(nconf,(nu-1)*(n+1)+il)=tangtotal(1,(nu-1)*n+il)
               tyy(nconf,(nu-1)*(n+1)+il)=tangtotal(2,(nu-1)*n+il)
               tzz(nconf,(nu-1)*(n+1)+il)=tangtotal(3,(nu-1)*n+il)
               nxx(nconf,(nu-1)*(n+1)+il)=normtotal(1,(nu-1)*n+il)
               nyy(nconf,(nu-1)*(n+1)+il)=normtotal(2,(nu-1)*n+il)
               nzz(nconf,(nu-1)*(n+1)+il)=normtotal(3,(nu-1)*n+il)
               bxx(nconf,(nu-1)*(n+1)+il)=binormtotal(1,(nu-1)*n+il)
               byy(nconf,(nu-1)*(n+1)+il)=binormtotal(2,(nu-1)*n+il)
               bzz(nconf,(nu-1)*(n+1)+il)=binormtotal(3,(nu-1)*n+il)
               vxx(nconf,(nu-1)*(n+1)+il)=vitessetotale(1,(nu-1)*n+il)
               vyy(nconf,(nu-1)*(n+1)+il)=vitessetotale(2,(nu-1)*n+il)
               vzz(nconf,(nu-1)*(n+1)+il)=vitessetotale(3,(nu-1)*n+il)
 93            continue
               xx(nconf,(nu-1)*(n+1)+n+1)=total(1,(nu-1)*n+1)
               yy(nconf,(nu-1)*(n+1)+n+1)=total(2,(nu-1)*n+1)
               zz(nconf,(nu-1)*(n+1)+n+1)=total(3,(nu-1)*n+1)
               txx(nconf,(nu-1)*(n+1)+n+1)=tangtotal(1,(nu-1)*n+1)
               tyy(nconf,(nu-1)*(n+1)+n+1)=tangtotal(2,(nu-1)*n+1)
               tzz(nconf,(nu-1)*(n+1)+n+1)=tangtotal(3,(nu-1)*n+1)
               nxx(nconf,(nu-1)*(n+1)+n+1)=normtotal(1,(nu-1)*n+1)
               nyy(nconf,(nu-1)*(n+1)+n+1)=normtotal(2,(nu-1)*n+1)
               nzz(nconf,(nu-1)*(n+1)+n+1)=normtotal(3,(nu-1)*n+1)
               bxx(nconf,(nu-1)*(n+1)+n+1)=binormtotal(1,(nu-1)*n+1)
               byy(nconf,(nu-1)*(n+1)+n+1)=binormtotal(2,(nu-1)*n+1)
               bzz(nconf,(nu-1)*(n+1)+n+1)=binormtotal(3,(nu-1)*n+1)
               vxx(nconf,(nu-1)*(n+1)+n+1)=vitessetotale(1,(nu-1)*n+1)
               vyy(nconf,(nu-1)*(n+1)+n+1)=vitessetotale(2,(nu-1)*n+1)
               vzz(nconf,(nu-1)*(n+1)+n+1)=vitessetotale(3,(nu-1)*n+1)
 92            continue


               conteur=0
               endif
```

```
      conteur=conteur+1


2     continue
*-----------------------------------------------------------*
*           passage en matlab des configurations            *
*-----------------------------------------------------------*
        write(*,*)'t'
        write(*,*)t


      call passmatrice(xx,nbrconf,(n+1)*nbra,'xx',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(yy,nbrconf,(n+1)*nbra,'yy',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(zz,nbrconf,(n+1)*nbra,'zz',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(txx,nbrconf,(n+1)*nbra,'txx',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(tyy,nbrconf,(n+1)*nbra,'tyy',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(tzz,nbrconf,(n+1)*nbra,'tzz',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(nxx,nbrconf,(n+1)*nbra,'nxx',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(nyy,nbrconf,(n+1)*nbra,'nyy',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(nzz,nbrconf,(n+1)*nbra,'nzz',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(bxx,nbrconf,(n+1)*nbra,'bxx',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(byy,nbrconf,(n+1)*nbra,'byy',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(bzz,nbrconf,(n+1)*nbra,'bzz',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(vxx,nbrconf,(n+1)*nbra,'vxx',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(vyy,nbrconf,(n+1)*nbra,'vyy',nvsurdimcnbrconf,nvsurdimt)
      call passmatrice(vzz,nbrconf,(n+1)*nbra,'vzz',nvsurdimcnbrconf,nvsurdimt)




      call passmatrice(para1,nbra,nbrconf,'para1',10,nvsurdimcnbrconf)
      call passmatrice(para2,nbra,nbrconf,'para2',10,nvsurdimcnbrconf)
      call passmatrice(para3,nbra,nbra*nbrconf,'para3',10,nvsurdimcnbrconf)
      call passvecteur(para1min,nbra,'para1min',10)
      call passvecteur(para2min,nbra,'para2min',10)
      call passscalairreal(para3min,'para3min')




      call fermer

      stop
      end
```

```
%-------------------------------------------------------------------%
%                 fichier : xpdessins                               %
%-------------------------------------------------------------------%


%*******************************************************************%
%                      introduction                                 %
%*******************************************************************%


% On trouve dans ce fichier les differentes sous routines matlab et le
% programme principal matlab qui permet de faire des visualisations
% graphiques. Ce fichier n'est pas executable car commme le veut
% matlab il faut mettre chaque procedure dans un fichier, mais le
% fait d' ecrire toutes les procedures dans un seul fichier permet
% de comprendre le lien des procedures entre elles ainsi qu'avec
% le programme principal.


% Pour lancer matlab4 sur sun faire :
% rsh mortimer xterm -e matlab -display $DISPLAY
% Pour avoir des precisions sur les instructions utiliser help


%*******************************************************************%
%                    les sous routines                              %
%*******************************************************************%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         utilitaires                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function xsoustitle(string)
h = get(gca,'title');
if isempty(h)
        h = text('horiz','center');
        set(gca,'title',h);
end
set(h+1,'string',string);

return
end

function [xi,yi,zi,nxi,nyi,nzi,bxi,byi,bzi]= xaffecte(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0)
xi=x0;
yi=y0;
zi=z0;
nxi=nx0;
nyi=ny0;
nzi=nz0;
bxi=bx0;
byi=by0;
bzi=bz0;
return
end

function sol=  xext(para3,i,j,nbra)
global nbr nbrconf nbrdt t para circul debit

    for l=1:nbrconf,
       sol(l)=para3(i,j+(l-1)*nbra);
    end;

return
end


function x=  xextract(xx,nu)
global nbr nbrconf nbrdt t para circul debit
    for i=1:nbr+1,
       x(:,i)=xx(:,(nu-1)*(nbr+1)+i);
    end:
```

```
return
end

function [x,y,z,nx,ny,nz,bx,by,bz]=xextractc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,nu)
x=xextract(xx,nu);
y=xextract(yy,nu);
z=xextract(zz,nu);

nx=xextract(nxx,nu);
ny=xextract(nyy,nu);
nz=xextract(nzz,nu);
bx=xextract(bxx,nu);
by=xextract(byy,nu);
bz=xextract(bzz,nu);

return
end

function [x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]=xextractcv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,
                                                               byy0,bzz0,nu)
x0=xx0(nu,:);
y0=yy0(nu,:);
z0=zz0(nu,:);

nx0=nxx0(nu,:);
ny0=nyy0(nu,:);
nz0=nzz0(nu,:);
bx0=bxx0(nu,:);
by0=byy0(nu,:);
bz0=bzz0(nu,:);
return
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                            rotation                                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x,y,z]= xrotax(ang1,x,y,z)
    Yv=y*cos(ang1)-z*sin(ang1);
    Zv=y*sin(ang1)+z*cos(ang1);
    Xv=x;
return
end

function [x,y,z]= xrotay(ang1,x,y,z)
    Zv=z*cos(ang1)-x*sin(ang1);
    Xv=z*sin(ang1)+x*cos(ang1);
    Yv=y;
x=Xv;
y=Yv;
z=Zv;
return
end

function [x,y,z]= xrotaz(ang1,x,y,z)
    Xv=x*cos(ang1)-y*sin(ang1);
    Yv=x*sin(ang1)+y*cos(ang1);
    Zv=z;
x=Xv;
y=Yv;
z=Zv;
return
end

function [x0,y0,z0]=  xaxo(x0,y0,z0,ang1,ang2,ang3)
global nbr nbrconf nbrdt t para circul debit
for i=1:nbr+1,

    [x0(i),y0(i),z0(i)]=xrotay(ang1,x0(i),y0(i),z0(i));
    [x0(i),y0(i),z0(i)]=xrotaz(ang2,x0(i),y0(i),z0(i));
    [x0(i),y0(i),z0(i)]=xrotay(ang3,x0(i),y0(i),z0(i));
```

```
end;
return
end

function [x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]= xaxon(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,
                                        ang1,ang2,ang3)
[x0,y0,z0]=  xaxo(x0,y0,z0,ang1,ang2,ang3);
[nx0,ny0,nz0]=  xaxo(nx0,ny0,nz0,ang1,ang2,ang3);
[bx0,by0,bz0]=  xaxo(bx0,by0,bz0,ang1,ang2,ang3);
return
end

function [xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz]=xaxonc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,
                                        ang1,ang2,ang3,nbra)
global nbr nbrconf nbrdt t para circul debit
for nu=1:nbra,

[x,y,z,nx,ny,nz,bx,by,bz]=xextractc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,nu);

for mm=1:nbrconf,
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]   =xaffecte(x(mm,:),y(mm,:),z(mm,:),nx(mm,:),ny(mm,:)
                                        ,nz(mm,:),bx(mm,:),by(mm,:),bz(mm,:));
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]= xaxon(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ang1,ang2,
                                  ang3);
x(mm,:)=x0;
y(mm,:)=y0;
z(mm,:)=z0;
nx(mm,:)=nx0;
ny(mm,:)=ny0;
nz(mm,:)=nz0;
bx(mm,:)=bx0;
by(mm,:)=by0;
bz(mm,:)=bz0;
end;

for i=1:nbr+1,
    xx(:,(nu-1)*(nbr+1)+i)=x(:,i);
    yy(:,(nu-1)*(nbr+1)+i)=y(:,i);
    zz(:,(nu-1)*(nbr+1)+i)=z(:,i);
    nxx(:,(nu-1)*(nbr+1)+i)=nx(:,i);
    nyy(:,(nu-1)*(nbr+1)+i)=ny(:,i);
    nzz(:,(nu-1)*(nbr+1)+i)=nz(:,i);
    bxx(:,(nu-1)*(nbr+1)+i)=bx(:,i);
    byy(:,(nu-1)*(nbr+1)+i)=by(:,i);
    bzz(:,(nu-1)*(nbr+1)+i)=bz(:,i);

end;
end;
return
end

function [xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0]=xaxoncv(xx0,yy0,zz0,nxx0,nyy0,nzz0
                                        ,bxx0,byy0,bzz0,ang1,ang2,ang3,nbra)

for nu=1:nbra,
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]=xextractcv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0
                                  ,nu);
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]= xaxon(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ang1,ang2,
                                  ang3);
xx0(nu,:)=x0;
yy0(nu,:)=y0;
zz0(nu,:)=z0;
nxx0(nu,:)=nx0;
nyy0(nu,:)=ny0;
nzz0(nu,:)=nz0;
bxx0(nu,:)=bx0;
byy0(nu,:)=by0;
bzz0(nu,:)=bz0;
end;

return
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                           dessin                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function sol=  xappartient(i,st,lg)
cct=0;
if lg==0
    cct=0;
else
    for j=1:lg,
           if i==st(j)
               cct=1;
            end
    end;
end
end;
sol=cct;
return
end

function [lg,st]= xinter(x0,y0,z0)

global nbr nbrconf nbrdt t para circul debit


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               recherche d'intersection                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

lg=0;
for i=1:nbr,
    vk=xpermut(i,nbr);
    disref=(y0(vk(2))-y0(i)).^2+(z0(vk(2))-z0(i)).^2;
    disref=sqrt(disref)/2;

    for j=4:nbr-2,
    dis=(y0(vk(j))-y0(i)).^2+(z0(vk(j))-z0(i)).^2;
    dis=sqrt(dis);
     if dis<disref




               lg=lg+1;
               if x0(i) < x0(vk(j))
                     st(lg)=i;
                else
                     st(lg)=vk(j);
                end
     end

    end;

end;

return
end



function [longint,int]= xinterann(xx0,yy0,zz0,nbra)

global nbr nbrconf nbrdt t para circul debit

for i=1:nbra,
    longint(i)=0;
end;
```

```
for i=1:nbra,

x0=xx0(i,:);
y0=yy0(i,:);
z0=zz0(i,:);
for j=1:nbra,
    if ~(j==i)


x01=xx0(j,:);
y01=yy0(j,:);
z01=zz0(j,:);
        for k=1:nbr,
          vk=xpermut(k,nbr);
          disref=(y0(vk(2))-y0(k)).^2+(z0(vk(2))-z0(k)).^2;
          disref=sqrt(disref)/0.8;
          for kk=1:nbr,
            dis=(y01(kk)-y0(k)).^2+(z01(kk)-z0(k)).^2;
            dis=sqrt(dis);
            if dis<disref & x0(k)<x01(kk)


                longint(i)=1+longint(i);
                int(i,longint(i))=k;
            end
          end;
        end;

     end
end;
if longint(i)==0
    int(i,1)=0;
end
end;
return
end


function sol=  xpermut(i,n)

for j=1:n,

             ll=j+i-1;
             if ll>n
                            ll=ll-n;
             end
             sol(j)=ll;
end;

return
end

function [x0c,y0c,z0c]= xcercle(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,la,jk,jkk)
    x0c=x0(jk)+ep*cos(la)*nx0(jkk)+ep*sin(la)*bx0(jkk);
    z0c=z0(jk)+ep*cos(la)*nz0(jkk)+ep*sin(la)*bz0(jkk);
    y0c=y0(jk)+ep*cos(la)*ny0(jkk)+ep*sin(la)*by0(jkk);

return
end

function xgraphel(x0,y0,z0)
global nbr nbrconf nbrdt t para circul debit
line(y0,z0)
return
end


function xgraphe(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,version,lg1,st1)
global nbr nbrconf nbrdt t para circul debit
```

```
[lg,st]= xinter(x0,y0,z0);

if ~(lg1==0)

       for h=1:lg1,
           st(lg+h)=st1(h);
       end;
       lg=lg+lg1;
end

m=8;

   for jk=1:nbr,
%pause(0.3)
   for jl=1:m+1,
   la=(jl-1)*2*pi/m;

[x0c(jl),y0c(jl),z0c(jl)]= xcercle(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,la,jk,jk);


    end;
    ddmax=0;
    for it=1:m,
       for itt=it+1:m,
         dd=(y0c(itt)-y0c(it)).^2+(z0c(itt)-z0c(it)).^2;
         dd=sqrt(dd);
         if dd>=ddmax
                   ddmax=dd;
                   stt(1)=it;
                   stt(2)=itt;
          end
       end;
    end;
        if stt(1)>stt(2)
                lh=stt(2);
                stt(2)=stt(1);
                stt(1)=lh;
        end

    vg=xpermut(stt(1),m);


    if x0c(vg(2)) > x0c(vg(m))
              no=stt(2)-stt(1)+1;

    else

              vg=xpermut(stt(2),m);
              no=stt(2)-stt(1)+1;
              no=m-no+2;
    end

 res=xappartient(jk,st,lg);
% res=0;
   if res==0

    for iv=1:no-1,

    yyy(1)=y0c(vg(iv));
    zzz(1)=z0c(vg(iv));
    yyy(2)=y0c(vg(iv+1));
    zzz(2)=z0c(vg(iv+1));

    line(yyy,zzz)
    end;

 if version==1 | version==3
              kt=no;
    else
              kt=2;
```

```
end
         for j=1:kt,
if version==1 | version==3
          la=(vg(j)-1)*2*pi/m;
   else

        la=(stt(j)-1)*2*pi/m;
end



[x01(jk),y01(jk),z01(jk)]= xcercle(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,la,jk,jk);
if version==1 | version==2
  [x01(jk+1),y01(jk+1),z01(jk+1)]= xcercle(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,la,jk+1,
                                          jk+1);

   else
    [x01(jk+1),y01(jk+1),z01(jk+1)]= xcercle(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,la,
                                            jk+1,jk);

end



         yyy(1)=y01(jk);
         zzz(1)=z01(jk);

         yyy(2)=y01(jk+1);
         zzz(2)=z01(jk+1);

         line(yyy,zzz)

    end;

    end;
 end;
return
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      multivues                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function sol= xlong(x0,y0,z0)

global nbr nbrconf nbrdt t para circul debit


disref=0;
for i=1:nbr,

    vk=xpermut(i,nbr);
    for j=2:nbr,
    dis=(x0(vk(j))-x0(i)).^2+(y0(vk(j))-y0(i)).^2+(z0(vk(j))-z0(i)).^2;
    dis=sqrt(dis);

    if dis>disref
        disref=dis;
    end;
end;
sol=disref;
return
end

function sol=xlong1(x0,y0,z0)
xmin=min(min(x0));
xmax=max(max(x0));
ymin=min(min(y0));
ymax=max(max(y0));
zmin=min(min(z0));
```

```
zmax=max(max(z0));
dx=xmax-xmin;
dy=ymax-ymin;
dz=zmax-zmin;
sol=dx^2+dy^2+dz^2;
sol=sqrt(sol);
return
end

function sol= xmoy(x,n)
sol=0;
    for i=1:n,
    sol=sol+x(i);
    end;
sol=sol/n;
return
end

function sol= xcentre(x,y,z,n)
sol(1)= xmoy(x,n);
sol(2)= xmoy(y,n);
sol(3)= xmoy(z,n);
return
end

function [x,y,z]= xcentrage(x,y,z,centre)
x=x-centre(1);
y=y-centre(2);
z=z-centre(3);
return
end

function xcarre(y0,z0,lc)
y=[-lc/2+y0,lc/2+y0,lc/2+y0,-lc/2+y0,-lc/2+y0];
z=[-lc/2+z0,-lc/2+z0,lc/2+z0,lc/2+z0,-lc/2+z0];
line(y,z);
return
end


function xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,
                      bxx,byy,bzz,ang1,ang2,ang3,version,num,nbra)
global nbr nbrconf nbrdt t para circul debit

lnum=length(num);

l=xlongl(xx0,yy0,zz0);
lc=2*l;
clg
if ~(lnum==1)
v=[-lc/2,lc/2+4*lc,-lc/2,ceil(lnum/4)*lc-lc/2];
else
   if num(1)==0
        v=[-lc/2,lc/2,-lc/2,lc/2];
    end
   if ~(num(1)==0)
        v=[lc/2,3*lc/2,-lc/2,lc/2];
    end
end
axis(v)


ep=0.1*l/2;

gk=0;
if num(1)==0
gk=1;
[xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0]=xaxoncv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,
                                                    byy0,bzz0,ang1,ang2,ang3,nbra);
[longint,int]= xinterann(xx0,yy0,zz0,nbra);

for nu=1:nbra,
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]=xextractcv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,
                                              bzz0,nu);
g= xcentre(x0,y0,z0,nbr);
gg(nu,:)=g;
end;
centre=xcentre(gg(:,1),gg(:,2),gg(:,3),nbra);

for nu=1:nbra,
lg1=longint(nu);
clear st1
if ~(lg1==0)
      st1=int(nu,:);
     else
      st1=0;
end
[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]=xextractcv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,
                                              bzz0,nu);
[x0,y0,z0]= xcentrage(x0,y0,z0,centre);
if version==5
  xgraphel(x0,y0,z0)
 else
  xgraphe(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,version,lg1,st1)
end
end;
end


if ~(lnum==1 & gk==1)
[xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz]=xaxonc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,ang1,ang2,
                                          ang3,nbra);


        s=0;
        ss=0;
    for kk=1+gk:lnum,
      mm=num(kk);
      for h=1:nbra,
        for hh=1:nbr+1,
        xx0(h,hh)=xx(mm,(h-1)*(nbr+1)+hh);
        yy0(h,hh)=yy(mm,(h-1)*(nbr+1)+hh);
        zz0(h,hh)=zz(mm,(h-1)*(nbr+1)+hh);
        nxx0(h,hh)=nxx(mm,(h-1)*(nbr+1)+hh);
        nyy0(h,hh)=nyy(mm,(h-1)*(nbr+1)+hh);
        nzz0(h,hh)=nzz(mm,(h-1)*(nbr+1)+hh);
        bxx0(h,hh)=bxx(mm,(h-1)*(nbr+1)+hh);
        byy0(h,hh)=byy(mm,(h-1)*(nbr+1)+hh);
        bzz0(h,hh)=bzz(mm,(h-1)*(nbr+1)+hh);
        end;
      end;
clear longint
clear int
      [longint,int]= xinterann(xx0,yy0,zz0,nbra);

        s=s+1;

for nu=1:nbra,

[x,y,z,nx,ny,nz,bx,by,bz]=xextractc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,nu);
  [x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]   =xaffecte(x(mm,:),y(mm,:),z(mm,:),nx(mm,:),
                                         ny(mm,:),nz(mm,:),bx(mm,:),by(mm,:),bz(mm,:));
g= xcentre(x0,y0,z0,nbr);
gg(nu,:)=g;
end;
centre=xcentre(gg(:,1),gg(:,2),gg(:,3),nbra);

for nu=1:nbra,
lg1=longint(nu);
clear st1
if ~(lg1==0)
      st1=int(nu,:);
     else
```

```
end

[x,y,z,nx,ny,nz,bx,by,bz]=xextractc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,nu);
  [x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]   =xaffecte(x(mm,:),y(mm,:),z(mm,:),nx(mm,:),
                                          ny(mm,:),nz(mm,:),bx(mm,:),by(mm,:),bz(mm,:));
 [x0,y0,z0]= xcentrage(x0,y0,z0,centre);
      y0=y0+s*lc;
      z0=z0+ss*lc;
     xcarre(s*lc,ss*lc,lc)
if version==5
  xgraphel(x0,y0,z0)
 else
  xgraphe(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,version,lg1,st1)
end
  end;



        if s==4
                      s=0;
                     ss=ss+1;
         end

     end;
end


return
end


function xmultivuesto(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,
                      bxx,byy,bzz,version,num,nbra)

ang1=-pi/2;
ang2=-pi/4;
ang3=35.2/180*pi;
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
            bzz,ang1,ang2,ang3,version,num,nbra);
return
end


function xmultivuestoe(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,
                       bxx,byy,bzz,version,num,nbra)
ang1=-pi/2;
ang2=0;
ang3=35.2/180*pi;
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
            bzz,ang1,ang2,ang3,version,num,nbra);
return
end


function xmultivuestom(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,
                       nzz,bxx,byy,bzz,version,num,nbra)

ang1=-pi/2;
ang2=-pi/6;
ang3=15/180*pi;
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
            bzz,ang1,ang2,ang3,version,num,nbra);
return
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              vues                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [ymin,ymax,zmin,zmax]=xfenetre(xi,yi,zi,x,y,z)
ymin=min([min(y),min(yi)]);
ymax=max([max(y),max(yi)]);
zmin=min([min(z),min(zi)]);
zmax=max([max(z),max(zi)]);
return
end



function [longinti,inti]= xinteranni(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,
                                        zz,nbra,num)


global nbr nbrconf nbrdt t para circul debit
lnum=length(num);
nb=nbra*lnum;

gk=0;
if num(1)==0
  gk=1;
  for i=1:nbra,
  vxx0(i,:)=xx0(i,:);
  vyy0(i,:)=yy0(i,:);
  vzz0(i,:)=zz0(i,:);
  end;
end

if ~(lnum==1 & gk==1)
   for i=gk+1:lnum,
   mm=num(i);
   for j=1:nbr+1,
   for k=1:nbra,
   vxx0((i-1)*nbra+k,j)=xx(mm,(k-1)*nbra+j);
   vyy0((i-1)*nbra+k,j)=yy(mm,(k-1)*nbra+j);
   vzz0((i-1)*nbra+k,j)=zz(mm,(k-1)*nbra+j);
   end;
   end;
end


[longint,int]= xinterann(vxx0,vyy0,vzz0,nb);


for i=1:lnum,
for j=1:nbra,
   longinti(i,j)=longint((i-1)*nbra+j);

   inti((i-1)*nbra+j,:)=int((i-1)*nbra+j,:);
end;
end;


return
end



function xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,
                byy,bzz,ang1,ang2,ang3,version,num,nbra,lab)
global nbr nbrconf nbrdt t para circul debit

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               vuesaxono de l'' ellipse en y-z                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

lnum=length(num);

l=xlongl(xx0,yy0,zz0);
ep=0.1*l/4;
```

```
[xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0]=xaxoncv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,
                                                     byy0,bzz0,ang1,ang2,ang3,nbra);
[xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz]=xaxonc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,ang1,ang2,
                                          ang3,nbra);



[ymin,ymax,zmin,zmax]=xfenetre(xx0,yy0,zz0,xx,yy,zz);
dy=ymax-ymin;
dz=zmax-zmin;
v=[ymin-0.1*dy,ymax+0.1*dy,zmin-0.1*dz,zmax+0.1*dz];
axis(v)

if lab==1

[longinti,inti]= xinteranni(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nbra,
                            num);

end

gk=0;
if num(1)==0

gk=1;
if lab==1
longint=longinti(1,:);
for i=1:nbra,
    int(i,:)=inti(i,:);
end;
end

for nu=1:nbra,

if lab ==1
lg1=longint(nu);
clear st1
if ~(lg1==0)
      st1=int(nu,:);
     else
      st1=1;
end
else
lg1=0;
st1=0;
end


[x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]=xextractcv(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,
                                              bzz0,nu);

if version==5
  xgraphel(x0,y0,z0)
 else
  xgraphe(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,version,lg1,st1)
end
end;
end

if ~(lnum==1 & gk==1)
for m=1+gk:lnum,

mm=num(m);
if lab==1
clear longint
clear int


  longint=longinti(m,:);
  for i=1:nbra,
     int(i,:)=inti(i+nbra*(m-1),:);
end;
end
```

```
for nu=1:nbra,

if lab==1
lg1=longint(nu);
clear st1
if ~(lg1==0)
      st1=int(nu,:);
     else
      st1=1;
end
else
lg1=0;
st1=0;
end


[x,y,z,nx,ny,nz,bx,by,bz]=xextractc(xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,nu);
  [x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0]  =xaffecte(x(mm,:),y(mm,:),z(mm,:),nx(mm,:),
                                          ny(mm,:),nz(mm,:),bx(mm,:),by(mm,:),bz(mm,:));
if version==5

  xgraphel(x0,y0,z0)
 else
  xgraphe(x0,y0,z0,nx0,ny0,nz0,bx0,by0,bz0,ep,version,lg1,st1)
end
end;
end;
end

function vuesto(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,
                byy,bzz,version,num,nbra,lab)
ang1=-pi/2;
ang2=-pi/4;
ang3=35.2/180*pi;

xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,ang1
      ,ang2,ang3,version,num,nbra,lab)
return
end

function vuestoe(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,
                 byy,bzz,version,num,nbra,lab)

ang1=-pi/2;
ang2=0;
ang3=35.2/180*pi;
xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,ang1,
       ang2,ang3,version,num,nbra,lab)
return
end

function vuestom(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,
                 byy,bzz,version,num,nbra,lab)

ang1=-pi/2;
ang2=-pi/6;
ang3=15/180*pi;
xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,
       ang1,ang2,ang3,version,num,nbra,lab)
return
end


%--------------------------------------------------------------------%
%                programme  matlab principal: xpdessin.m             %
%--------------------------------------------------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          introduction                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% le programme principal suivant utilise les procedures precedantes
% et permet de realiser facliement la sortie graphique d'une simulation
% Il est propose  une representation des differentes comnfigurations
% soit sous la forme d'une ''multivue'' soit sous celle d'une ''vue''
% avec ,dans l'une ou l'autre des representations, la possibilite de
% choisir la direction d'observation.Faite votre choix et recopier sur
% la fenetre matlab la partie correspondante a votre choix.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      preliminaires                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pass
% pour interpreter le fichier pass.m
% pour que pass.m soit non vide il faut avoir successivement
% execute initial.e puis pmultianneaux.e

% ou load ggellipse par exemple si on a deja fait pass et save ggellipse
global nbr nbrconf nbrdt t para circul debit


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 multivuesaxono                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[0:1:nbrconf];
xmultivuestoe(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
              bzz,4,num,nbra)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y-z')
ylabel('x')

dt=t*floor(nbrdt/nbrconf)
title('multivues de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 multivues en yz                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[0:1:nbrconf];
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
            bzz,0,0,0,5,num,nbra)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y')
ylabel('z')

dt=t*floor(nbrdt/nbrconf)
title('multivues de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 multivues en  yx                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[0:1:nbrconf];
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
            bzz,-pi/2,0,0,5,num,nbra)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y')
ylabel('x')

dt=t*floor(nbrdt/nbrconf)
title('multivues de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 multivues en  xz                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[0:1:nbrconf];
xmultivuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,
              bzz,0,pi/2,0,5,num,nbra)

kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'x')
ylabel('z')

dt=t*floor(nbrdt/nbrconf)
title('vue de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    vuesaxono                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num=[6,12,16];


xvuestom(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,4,
         num,nbra,0)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y-z')
ylabel('x')

dt=t*floor(nbrdt/nbrconf)
title('multivues de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    vues en yz                                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num=[0:1:nbrconf];

xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,
       0,0,0,5,num,nbra,0)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y')
ylabel('z')

dt=t*floor(nbrdt/nbrconf)
title('vue de l'' ellipse :  a=2 b=1.5 dt=0.58')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    vues en  yx                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num=[0:1:nbrconf];

xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,
       -pi/2,0,0,5,num,nbra,0)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'y')
ylabel('x')

dt=t*floor(nbrdt/nbrconf)
title('vue de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    vues en  xz                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num=[0:1:nbrconf];

xvuest(xx0,yy0,zz0,nxx0,nyy0,nzz0,bxx0,byy0,bzz0,xx,yy,zz,nxx,nyy,nzz,bxx,byy,bzz,
       0,pi/2,0,5,num,nbra,0)
kk=axis;
text(kk(2)+(kk(2)-kk(1))/100,kk(3),'x')
ylabel('z')

dt=t*floor(nbrdt/nbrconf)
title('vue de l'' ellipse :  a=2 b=1.5 dt=58e-2')
xsoustitle('circul=5    para=0,3   m0=0  t20=1')
```

```
%-----------------------------------------------------------------%
%                 fin du fichier xpdessins                        %
%-----------------------------------------------------------------%
```