

## Annexes 1 : Programme Libre Libre (L-L)

```

program LL
dimension inert (50), raid (50),
C(50),ALPHA(50),D(50),OM(50),x(50,50)
real inert, raid, C, alpha, D, OM, X
real, parameter: ESPI = 1.0e-8
open (11,file='donne1.dat',status='old')
open (12,file = 'rest.dat')
print*
print '(a\)', 'donner le nombre de disques N > 1 : '
read*,n
print*
print'(a,i2)', 'N=',N
L=n-1

!print*, 'saisie des moments inerties'
!print*
do i=1,n
! print*, 'inert(',i,')='
  read(11,*) inert (i)
enddo
!print*

!print*, 'saisie des rigidités torsionnelle'
!print*
do i=1,L
! print*, 'raid(',i,')='
  read(11,*) raid(i)
enddo

! calcul des éléments de la [C0] = f(c0(i) et
alpha0(i+1))
C(1 )=raid(1)/inert(1)
d(1) =c (1)
print*
do i=2,L
  j=i-1
  c(i)=(raid(j)+raid(i))/inert(i)
  d(i)=c(i)
  alpha(i)=(raid(j))**2/(inert(i)*inert(j))
end do
C(n)=raid(L)/inert(n)
d(n)=c(n)
alpha(n)=raid(1)**2/(inert(n)*inert(1))
S1 = 0
S2 = 0
do k = 1,n
  S1 = S1 + abs(c(k) - d(k))
  S2 = S2 + c(k)
  d(k) = c(k)
end do

!text d'arrêt d'iteration
if (Q - ESPI <= 0) then
print*
print*, "text d'arret d'iteration satisfait (Q -
espi)<= 0"
print*
  goto 1
end if
end do

1 print*, "Arret d'iteration a NB =",nb-1

```

```

! calcul des pulsations propres
do j = 1,n
  OM(j) = SQRT(ABS(C(j)))
end do
print*
print*,"Les pulsation propres sont:"
print*

do i = 1,L
  ii = (L + 1) - i
print*,'OM('i,')='om(ii),'rad/s'
print*
end do
print*,'OM('i,')='om(N),'rad/s'
  print*

!calcul des déformées modales X(disque, mode)
do i = 1,N
  j = (N + 1)- i
  x(1,i) = 1
  x(2,i) = 1 - (C(j) * inert(1) / raid(1))
  do k = 2,L
    x(k+1,i)=(inert(k)/raid(k))*((((raid(k-
1)+raid(k))/inert(k)-c(j))*x(k,i)-(raid(k-1)*x(k-
1,i))/inert(k))
  end do
end do

!instruction pour l'affichage des déformées
modales X(disque, mode)
print*
print*,"Les déformées modales X(disque,
mode)sont:"
print*
do i=1,L
  ii= (L + 1) - i! j'ai remplace n-i par (n+1)-i
write(*,*) 'Mode',i ;
write(*,*)'OMEGA('i,')='om(ii),"rad/s'
print*
write(*,3) 'Disque','Amplitude relative'
3 format(1x,A8,6x,A20)
print*
do j=1,N
  k = i + 1
write(*,4) j,x(j,k)
4 format(4x,I3,10x,F12.5)
enddo
print*
enddo
write(*,*) 'Mode',N ;
write(*,*)'OMEGA('N,')='om(N),"rad/s'
print*
write(*,3) 'Disque','Amplitude relative'
!3 format(1x,A8,6x,A20)
print*
do j=1,N
write(*,4) j,x(j,1)
!4 format(4x,I3,10x,F12.5)
END DO
end program LL

```

---

**Annexes 2 : Programme Encastré-Libre (E-L)**

```

program EL
dimension
inert(50),raid(50),C(50),ALPHA(50),D(50),OM(
50),x(50,50)
real inert,raid,C,alpha,D,OM,X
real, parameter:: ESPI = 1.0e-8
open(11,file='donne1.dat',status='old')
open(12,file = 'rest.dat')
print*
print'(a)', 'donner le nombre de disques N > 1 : '
read*,n
print*
print'(a,i2)', 'N=',N
L = n-1
!print*

!print*, 'saisie des moments inerties'
!print*
do i=1,n
    !print*, 'inert(',i,')='
    read(11,*)inert (i)
enddo

!calcul des elements de la matrice [C]
do NB = 1,250
A = c(1)
do i=1,L
    B = alpha(i+1)/A
    C(i) = A + B
    A = c(i + 1) - B
    alpha(i+1) = A * B
!print*
!print*, 'saisie des rigidites torsionnelles'
!print*
do i=1,n
    !print*, 'raid(',i,')='
    !print*

    read(1,*) raid(i)
enddo

! calcul des éléments de la [C0] = f(c0(i) et
alpha0(i+1))
do i = 1,L
    j = i + 1
    c(i) = (raid(i) + raid(j)) / inert(i)
    d(i) = c(i)
    alpha(j) = (raid(j))**2 / (inert(j) * inert(i))
end do
c(n) = raid(n) / inert(n)
d(n) = c(n)
alpha(n) = (raid(n))**2 / (inert(n) * inert(L))
end do
C(n) = A

!text de convergence
S1 = 0
S2 = 0
do k = 1,n
    S1 = S1 + abs(c(k) - d(k))
    S2 = S2 + c(k)

```

---

<pre> d(k) = c(k) end do  !text d'arrêt d'iteration if (Q - espi &lt;= 0) then print* print*,"text d'arret d'iteration satisfait (Q - espi)&lt;= 0" print* goto 1 end if end do print* 1 print*,"Arret d'iteration a NB =",nb-1  ! calcul des pulsation propre do j = 1,n OM(j) = SQRT(ABS(C(j))) end do  !instruction pour l'affichage des pulsation propres print* print*,"Les pulsation propres sont:" print* !print* !print*,'OM(',j,')=',om(j),'rad/s' !print* do i = 1,N ii = (N + 1) - i print*,'OM(',i,')=',om(ii),'rad/s' print* end do </pre>	<pre> !calcul des déformées modales X(disque, mode) do i = 1,N j = (N+1) - i x(1,i) = 1 x(2,i) = 1 + (raid(1) /raid(2)) -(C(j) * (inert(1) / raid(2))) do k = 2,L x(k+1,i)=(inert(k)/raid(k+1))*((((raid(k)+raid(k +1))/inert(k))-c(j))*x(k,i))-(raid(k)*x(k- 1,i))/inert(k)) end do end do  !instruction pour l'affichage des déformées modales X(disque, mode) print* print*,"Les déformées modales X(disque, mode)sont:" print* do i=1,N ii= (N + 1) - i! write(*,*) 'Mode',i ; write(*,*)'OMEGA(',i,')=',om(ii),"',rad/s' print* write(*,3) 'Disque','Amplitude relative' 3 format(1x,A8,6x,A20) print* do j=1,N write(*,4) j,x(j,i) 4 format(4x,I3,10x,F12.5) enddo print* enddo end program EL </pre>
---	---

**Annexes 3 : Programme Encastré-Encastré (E-E)**

```

program EE
dimension
inert(50),raid(50),C(50),ALPHA(50),D(50),OM(
50),x(50,50)
real inert,raid,C,alpha,D,OM,X
real, parameter::espi = 1.0e-8,pi= 3.141592654
open(11,file='donne1.dat',status='old')
open(12,file = 'rest.dat')
print*
print'(a)', 'donner le nombre de disques N > 1 : '
read*,n
print*
print'(a,i2)', 'N=',N
L = n+1
!print*

!print*, 'saisie des moments inerties'
!print*
do i=1,N
  ! print*, 'inert(',i,')='
  read (11,*) inert (i)
enddo
!print*

!print*, 'saisie des rigidites torsionnelles'
!print*
do i=1,L
  !print*, 'raid(',i,')='
  read (11,*) raid(i)
enddo

```

```

! calcul des éléments de la [C0] = f(c0(i) et
alpha0(i+1))
do i = 1,N - 1
  j = i + 1
  c(i) = (raid(i) + raid(j)) / inert(i)
  d(i) = c(i)
  alpha(j) = (raid(j))**2 / (inert(j) * inert(i))
end do
c(n) = (raid(n) + raid(L)) / inert(N)
d(n) = c(n)
do NB =1,250
  A = c(1)
  do i=1,N-1
    B = alpha(i+1)/A
    C(i) = A + B
    A = c(i + 1) - B
    alpha(i+1) = A * B
  end do
  C(n) = A

!text de convergence
S1 = 0
S2 = 0
do k = 1,n
  S1 = S1 + abs(c(k) - d(k))
  S2 = S2 + c(k)
  d(k) = c(k)
end do

!text d'arrêt d'iteration
if (Q - espi <= 0) then
print*

```

<pre> print*, "text d'arret d'iteration il faut que (Q - espi) soit &lt;= 0" print*   goto 1 end if end do print* 1 print*, "Arret d'iteration a NB =", nb-1  ! calcul des pulsations propres do j = 1, n   OM(j) = SQRT(ABS(C(j))) end do  !instruction pour l'affichage des pulsation propres print* print*, "Les pulsation propres sont:" print* !print* !print*, 'OM(', j, ')=' , om(j), 'rad/s' !print* do i = 1, N   ii = (N + 1) - i   print*, 'OM(', i, ')=' , om(ii), 'rad/s'   print* end do  !calcul des déformées modales X(disque, mode) do i = 1, N   j = (N+1) - i </pre>	<pre> x(1,i) = 1 x(2,i) = 1 + (raid(1) /raid(2)) -(C(j) * inert(1) / raid(2)) do k = 2, L -2  x(k+1,i)=(inert(k)/raid(k+1))*((((raid(k)+raid(k +1))/inert(k))-c(j))*x(k,i))-(raid(k)*x(k- 1,i))/inert(k))    end do end do  !instruction pour l'affichage des déformées modales X(disque, mode) print* print*, "Les déformées modales X(disque, mode)sont:" print* do i=1, N   ii= (N + 1) - i!   write(*,*) 'Mode', i ;   write(*,*) 'OMEGA(', i, ')=' , om(ii), " ,rad/s'   print*   write(*,3) 'Disque', 'Amplitude relative'   3 format(1x,A8,6x,A20)   print*   do j=1, N     write(*,4) j, x(j,i)     4 format(4x,I3,10x,F12.5)   enddo   print*   enddo end program EE </pre>
--	--

## Annexes 4 : Biographie de M. Heinz Rutishauser



**Heinz Rutishauser** (né le 30 janvier 1918 à Weinfelden et mort le 10 novembre 1970 à Zurich) est un mathématicien suisse, pionnier de l'analyse numérique moderne et précurseur de l'informatique.

**Nationalité :** Suisse,

**Formation :** École polytechnique fédérale de Zurich,

**Activités :** Mathématicien, informaticien, professeur d'Université.

---