Taller

Física Computacional

```
      dimension v(0:19,20),x(20),c(0:20),work(20),iwork(20)
     1 ,y(0:19,20),ca(0:19)
      b=1.
      a=-1.
      n=3
      h=(b-a)/fLoat(n-1)
      xx=0.5
      xy=0.0
      x(2)=0.
      x(3)=xx
      x(1)=-x(3)
      do 15 j=1,20
      do 1 i=1,3
      v(0,i)=1.
      y(0,i)=1.
    1 continue
      do 2 k=1,n-1
      do 2 i=1,n
      v(k,i)=x(i)**k
      y(k,i)=v(k,i)
    2 continue
      do 3 k=0,n-1
      c(k)=(b**(k+1)-a**(k+1))/float(k+1)
    3 continue
      do 4 k=0,n-1
    4 ca(k)=c(k)
      call sgefs(v,20,n,ca,0,work,iwork)
      do 5 i=0,n-1
      ! print 6,ca(i)/h,(y(i,k),k=1,n),c(i),x(i+1)
    5 continue
    6 format(f8.4,12f6.1)
      do 8 k=0,8
      sum=0.
      do 9 i=0,n-1
      if (k.ne.0) tem=x(i+1)**k
      if (k.eq.0) tem=1.
    9 sum=sum+ca(i)*tem
      ! print 10,k,sum,(b**(k+1)-a**(k+1))/float(k+1)
   10 format(i5,5f10.2)
    8 continue
      av = (abs(xx - xy))/2.
      xy = xx
      diff = ( (xx**4)/(xx**2) ) - (3./5.)
      if (diff.gt.0) xx=(xx - av)
      if (diff.lt.0) xx=(xx + av)
      temp = (xx**4)/(xx**2)
      print 6, xx
   15 continue
      ! print 6,( (ca(0))*(xx**4) )
      ! print 6,( ca(0) + ca(1) + ca(2) )
      stop
      end

      include 'sgefs.for'
```

```
      dimension ys(10)
      ys(1)=   0.57847
      ys(2)=   0.55518
      ys(3)=   0.13365
      ys(4)=   0.68883
      ys(5)=   0.82248
      ys(6)=   0.51131
      ys(7)=   0.33379
      ys(8)=   0.84509
      ys(9)=   0.17888
      ys(10)=  0.02397
      do 1 i=1,10
      x=ran3(u)
      print 34,x,ys(i)
   34 format(2f10.5)
    1 continue
      stop
      end

      function ran3(d)
      integer *4 k,i,j,l,m
      data i/646143019/,j/1048729165/,l/0/,m/1073741823/
      save
      if (l.gt.0) goto 10
      fact=4./256.**4
      l=l+1
   10 continue
      k=iand(i+j,m)
      i=j
      j=k
    4 ran3=k*fact
      return
      end
```

```
c Solucion de la ecuacion radial de Schroedinger para una particula libre ( V(r)=0 )
c Se considera un proton viajando a una energia de E = 10.0 MeV

      implicit real*8 (a-h,o-z)
      dimension r(0:200)
      dimension aj(0:40),an(0:40)
      dimension w(200),u(0:15,0:200)

      n=200
      en=10.0
      ak=((1/20.9)*(en))**0.5     ! onda incidente, ec. 10.12 de la pag. 210
(Computational Physics)
      h=0.05
      r(0)=0.

      do i=1,n
      r(i)=r(i-1)+h
      enddo

      do 1 il=0,15
      call ajn(ak*h,aj,an)
      sl=il
      u(il,0)=0.              ! onda viajando inicialmente en el origen, fijado a cero
en el punto i=0
      u(il,1)=aj(il+1)*ak*h   ! onda viajando posteriormente en el punto i=1, valor
razonable

      do 2 i=1,n-1

      pot=0.0          ! potencial fijado a cero, caso de una particula libre

      w(i)=ak**2-(pot/20.9)-sl*(sl+1.)/r(i)**2
      u(il,i+1)=2.*u(il,i)-u(il,i-1)-h**2*w(i)*u(il,i)

      if (il.eq.15) print 25, r(i),u(il,i)  ! imprime la distancia radial "r(i)" y las
"u's" para un valor de "l"

  25  format(f6.2,e14.5)
  2   continue

c      print 26, il,u(il,i)  ! imprime el momento orbital y las soluciones de u's en
el ultimo punto de la malla, n=200

  26  format(i5,e16.5)
  1   continue

      end

c      include 'pleg.for'          ! polinomios de Legendre
      include 'ajn.for'           ! funciones de Bessel
```

```
c Solucion de la ecuacion radial de Schroedinger para un potencial de Coulomb
c Se considera un proton viajando a una energia de E = 10.0 MeV

      implicit real*8 (a-h,o-z)
      dimension r(0:200)
      dimension aj(0:40),an(0:40)
      dimension w(200),u(0:15,0:200)

      n=200
      en=10.0
      ak=((1/20.9)*(en))**0.5     ! onda incidente, ec. 10.12 de la pag. 210
(Computational Physics)
      VC=1.44                     ! constante de Coulomb en unidades de [r]=[fermis] y
[m]=[MeV]
      h=0.05
      r(0)=0.

      do i=1,n
      r(i)=r(i-1)+h
      enddo

      do 1 il=0,15
      call ajn(ak*h,aj,an)
      sl=il
      u(il,0)=0.              ! onda viajando inicialmente en el origen, fijado a cero
en el punto i=0
      u(il,1)=aj(il+1)*ak*h   ! onda viajando posteriormente en el punto i=1, valor
razonable

      do 2 i=1,n-1

      pot=VC/r(i)          ! potencial repulsivo de Coulomb, ecuacion (10.193), pag.
246 (Computational Physics )

      w(i)=ak**2-(pot/20.9)-sl*(sl+1.)/r(i)**2
      u(il,i+1)=2.*u(il,i)-u(il,i-1)-h**2*w(i)*u(il,i)

      if (il.eq.15) print 25, r(i),u(il,i)  ! imprime la distancia radial "r(i)" y las
"u's" para un valor de "l"

  25  format(f6.2,e14.5)
  2   continue

c      print 26, il,u(il,i)  ! imprime el momento orbital y las soluciones de u's en
el ultimo punto de la malla, n=200

  26  format(i5,e16.5)
  1   continue

      end

c      include 'pleg.for'          ! polinomios de Legendre
      include 'ajn.for'           ! funciones de Bessel
```

```
c package contains two routines, one to calculate simple Pn(x) and
c another to calculate Pnm(x) (associated Legendre functions)
      subroutine pleg (x,pl,lmax)
      implicit real*8 (a-h,o-z)

c calculates legendre polys for x=cos theta for l=0 (in 1)
c to l=lmax (actually lmax-1)
      dimension pl(50)
      pl(1)=1.
      pl(2)=x
      if (lmax.lt.3) return
      do 10 il=3,lmax
   10 pl(il)=((2.*il-3.)*x*pl(il-1)-(il-2.)*pl(il-2))/(il-1.)
      return
      end
      subroutine pl (m,x,p)
c calculates associated Legendre polys for single m and l=0 to 51
      dimension p(51)
      c=x
      s=sqrt(1.-x**2)
      if (m) 10,10,20
   10 p(1)=1
      p(2)=c
      go to 50
   20 do 30 l=1,m
   30 p(l)=0.0
      a=1.0
      am=m
      do 40 i=1,m
      ai=i
   40 a=a*(2.0*ai-1.0)
      p(m+1)=a*s**m
      p(m+2)=a*(2.0*am+1.0)*s**m*c
   50 n=m+2
      do 60 l=n,50
      a=2*l-1
      b=l+m-1
      d=l-m
   60 p(l+1)=(a*c*p(l)-b*p(l-1))/d
      return
      end
```

```
c calculates the spherical Bessel and Neumann functions
      subroutine ajn (x,aj,an)
      implicit real *8 (a-h,o-z)
      dimension aj(2), an(2)
      if (x.ne.0.0) go to 20
      aj(1)=1.0
      ain=1.0
      an(1)=ain
      do 10 i=2,25
      aj(i)=0.0
   10 an(i)=ain
      return
   20 s=dsin(x)
      c=dcos(x)
      aj(1)=s/x
      an(1)=-c/x
      aj(2)=s/x**2-c/x
      an(2)=-c/x**2-s/x
      do 30 lp=3,25
      al=lp-1
      aj(lp)=(2.0*al-1.0)*aj(lp-1)/x-aj(lp-2)
   30 an(lp)=(2.0*al-1.0)*an(lp-1)/x-an(lp-2)
      if (x.gt.6.0) return
      acon=1.0
      z=.50*x**2
      do 50 lp=1,25
      al=lp-1
      s=1.0
      t=1.0
      do 40 i=1,25
      aa=i
      t=-t*z/(aa*(2.0*al+2.0*aa+1.0))
   40 s=s+t
      aj(lp)=acon*s
      acon=acon*x/(2.0*al+3.0)
   50 continue
      return
      end
```

```
      dimension a(5,6), b(5,6), y(5), x(5)
      y(1) = 0.468
      y(2) = 0.695
      y(3) = 0.398
      y(4) = 0.913
      y(5) = 0.483

      a(1,1) = 0.546
      a(1,2) = 0.447
      a(1,3) = 0.242
      a(1,4) = 0.194
      a(1,5) = 0.795
      a(2,1) = 0.380
      a(2,2) = 0.276
      a(2,3) = 0.581
      a(2,4) = 0.108
      a(2,5) = 0.416
      a(3,1) = 0.721
      a(3,2) = 0.022
      a(3,3) = 0.853
      a(3,4) = 0.068
      a(3,5) = 0.312
      a(4,1) = 0.151
      a(4,2) = 0.759
      a(4,3) = 0.186
      a(4,4) = 0.597
      a(4,5) = 0.757
      a(5,1) = 0.192
      a(5,2) = 0.509
      a(5,3) = 0.041
      a(5,4) = 0.411
      a(5,5) = 0.632
      a(1,6) = y(1)
      a(2,6) = y(2)
      a(3,6) = y(3)
      a(4,6) = y(4)
      a(5,6) = y(5)


      open (55,file='gaussian.out',status='unknown')


      call gauss (a, 5, 6, 5)
      call m_display (a, 5, 6)

      stop
      end

      subroutine gauss(a,n,m,ldim)
      dimension t(1000),a(ldim,1)
      do 10 i=1,n-1
      con=1./a(i,i)
      do 5 k=1,m+1-i
    5 t(k)=-con*a(i,k+i-1)
      call elim (a(i,i),t,n-i,m+1-i,ldim,con)
   10 continue
```

```
      return
      end

      subroutine elim(a,t,nn,m,ldim,con1)
      dimension a(ldim,1),t(1)
      do 4 i=2,nn+1
      con=a(i,1)
      do 3 j=1,m
    3 a(i,j)=a(i,j)+con*t(j)
      !a(i,1)=con1*con
    4 continue
      return
      end

      subroutine m_display (a, l_row, l_col)

       dimension a(l_row, l_col)
       do i=1,l_row
        print 10,(a(i,j),j=1,l_col)

        write(55,10) (a(i,j),j=1,l_col)

   10    format (6F10.3)
       enddo

       print *, ""
       return
      end
```

```
c Minimos cuadrados para la obtencion de los parametros a1 y a2 de la
c ecuacion lineal y= a1 + a2*x. Solucion: a1= -0.07 y a2= 8.62

      implicit real*8(a-h,o-z)
      DATA SX/0./, SY/0./, X2/0./, XY/0./
      OPEN(5,FILE="mcuadrados.dat")
      READ(5,*) N
    1 FORMAT(I5)
      DO 170 I=1,N
      READ(5,*) X,Y
    2 FORMAT(2F10.0)
      SX=SX+X
      SY=SY+Y
      X2=X2+X*X
      XY=XY+X*Y
  170 CONTINUE
      XM=SX/N
      YM=SY/N
      A1=(N*XY-SX*SY)/(N*X2-SX*SX)
      A0=YM-A1*XM
      OPEN(6,FILE="mcuadrados.out")
      WRITE(6,3) A0, A1
    3 FORMAT(' ',2F10.7)
      STOP
      END
```