```fortran
PROGRAM scattcross

IMPLICIT NONE

!storage and variables
INTEGER, PARAMETER :: m=7
REAL, DIMENSION(m) :: theta, sigma, costheta, xdata, ydata
REAL, PARAMETER :: pi=3.14159265358979, convdr=pi/180.0
REAL, DIMENSION(4):: wg,xg
 DATA wg / 3.62683783378361983E-01, 3.13706645877887287E-01, 2.22381034453374471E-01, 1.01228536290376259E-01 /
 DATA xg / 1.83434642495649805E-01, 5.25532409916328986E-01, 7.96666477413626740E-01, 9.60289856497536232E-01 /

integer, DIMENSION(m):: iperm
INTEGER :: ndata, ier, i
REAL:: x,y


!aquire data
OPEN(15,file='scattcross.dat', FORM='FORMATTED',STATUS='OLD')
READ(15,'(I4)') ndata
do i=1, ndata
        READ (15,'(F5.0,1X,F7.3)') theta(i),sigma(i)
        costheta(i)= cos(convdr*theta(i))
 PRINT *, i, theta(i), sigma(i)

END DO
        CLOSE (15)




!enforce ordered nodes, increasing

 CALL SPSORT(costheta,ndata,iperm,1,ier)
!print *, 'here', ier
   DO i=1, ndata

        xdata(i)=costheta(iperm(i))
        ydata(i)=sigma(iperm(i))
        write(*,'(I4,2E15.7,I4)') iperm(i),xdata(i), ydata(i), ier

   end do

!make list of interpolated data, for plot
 open(16,file='scattcross.out',FORM='FORMATTED',STATUS='UNKNOWN')
 do  i=-120,+120

     x=.01*i
    y=func(x)
!linear spline interpolation
    write (16, '(2E15.7)') x,y
 end do

!----------------------------------------------------------------------------------------
     CONTAINS

    FUNCTION func(x)
!LINEAR spline interpolation of data
```

```

IMPLICIT NONE

!Declarations

REAL:: func
REAL, INTENT(IN):: x
REAL:: pl
INTEGER:: i,i1,i2

!find data interval for x
do i=2,ndata
   if (xdata(i).GE.x) EXIT
END DO
i=MIN(i,ndata)
i1=i-1
i2=i

!linear interpolation,Lagrange
func=ydata(i1)*(x-xdata(i2))/(xdata(i1)-xdata(i2))+ydata(i2)*(x-xdata(i1))/(xdata(i2)-xdata(i1))

RETURN

END FUNCTION func
!--------------------------------------------------------------------------------------


!-----------------------------------------------------------------
subroutine legpol(l,x,pl)

implicit none

integer, parameter:: lmax=4

integer, intent(in):: l
real, intent(in):: x
real, intent(out):: pl
integer:: i
real pl1, s
if(l.lt.0 .or. l.gt.lmax)then
   print*, '<legpol>|',l
   stop
end if
pl=1.
if(l.eq.0)return
pl=x
if(l.eq.1)return

pl1=1.
do i=2,1
   s=(2*i-1)*x*pl-(i-1)*pl1
   pl1=pl
   pl=s/float(i)
end do
return
end subroutine legpol
end program

```