

```

Declare Sub CalSpatial Lib "MeasCalcs.dll" (ByRef DataIn As Complex, ByVal AngleIn() As Double, _
    ByVal Anglenum As Integer, ByVal Freqs() As Double, _
    ByVal FreqNum As Integer, ByRef Data As Complex, _
    ByVal NumCoef As Integer, ByVal aradius As Double, _
    ByVal katrans As Double, ByVal LFgain As Double, ByVal
aradius As Double, _
    ByVal dist As Double, ByVal afreq As Double, ByVal aQ As
Double,
    ByVal avalue As Integer)
Declare Function CalPolar Lib "MeasCalcs.dll" (ByRef DataIn As Complex, _
    ByVal NumCoef As Integer, _
    ByVal angl As Double, _
    ByVal freq As Double, _
    ByVal farnum As Double, _
    ByVal Velnum As Double) As Complex
Declare Function CalVelocity Lib "MeasCalcs.dll" (ByRef DataIn As Complex, _
    ByVal NumCoef As Integer, _
    ByVal angl As Double, _
    ByVal freq As Double, _
    ByVal MeasRad As Double) As Complex
Declare Function CalPower Lib "MeasCalcs.dll" (ByRef DataIn As Complex, _
    ByVal NumCoef As Integer, _
    ByVal freq As Double, _
    ByVal farnum As Double) As Double

Declare Sub FFT Lib "MeasCalcs.dll" _
    (ByVal Measure_pt() As Double, ByRef NumPts_ As Integer)
Declare Sub LinToLog Lib "MeasCalcs.dll" _
    (ByRef Data_in As Complex, ByVal X_in As Double, _
    ByVal numpts_ As Integer, _
    ByRef DataOut As Complex, ByVal NumLogPts As Integer, _
    ByRef afreq As Double)

Declare Sub Smooth Lib "MeasCalcs.dll" _
    (ByRef Data_in As Double, ByVal Freqs() As Double, ByVal numpts_ As
Integer, _
    ByVal atype As Integer)

```

Here is the FORTRAN side of the call for FFT

```

SUBROUTINE FFT(Data_, n)

REAL(8), intent(in):: Data_(n)
integer(4), intent(in):: n

```

And on the next page the call for LogToLin

```

subroutine LinToLog(    DataIn,      &    ! FFT data in at all angles
                      DeltaF,      &    ! the frequency delta
                      NumFFTPts,  &    ! the number of FFT points
                      Dataout,    &    ! The interpolated log frequency output
                      NumLogPts,  &    ! the number of data points being output
                      Freqs       )    ! the Array of output frequencies


!DEC$ ATTRIBUTES REFERENCE :: DataIn
!DEC$ ATTRIBUTES Value :: NumFFTPts, DeltaF
!DEC$ ATTRIBUTES REFERENCE :: Dataout
!DEC$ ATTRIBUTES REFERENCE :: Freqs
!DEC$ ATTRIBUTES Value :: NumLogPts


integer, intent(in):: NumFFTPts                ! input array size constants
integer, intent(in) :: NumLogPts                ! number of
smoothed points - log scaled
COMPLEX(8), intent(in):: DataIn(0:NumFFTPts)    ! input frequency data !
changed
COMPLEX(8), intent(out) :: DataOut(0:NumLogPts) ! output data array ! changed
real(8), intent(in) :: DeltaF
real(8), intent(in) :: Freqs(0:NumLogPts)
real(8)  Freqin(0:NumFFTPts)

```