```fortran
Subroutine YouTweak(Mode)
!2018.10.09.1020cdt JMS- Your interface to/with Tweak-Engine.

!  Jeffrey M. Setterholm, 8095 230th St. E., Lakeville, Minnesota 55044, USA
!   I have authored the four Fortran *.f95 source code files listed below.
!                       I hereby place these four files:
!       Tweak-Begin.f95, Tweak-Engine.F95, Tweak-User.F95, & Tweak-Vis.f95
!               and the algorithms which are demonstrated therein,
!                            in the public domain.
!Disclaimer:
!       ***************************************************************
!       **********   Individual cognition is always flawed,  **********
!       **********           including yours & mine.         **********
!       **********                  - So: -                  **********
!       **********       Use this code at your own risk.     **********
!       ***************************************************************

!Table of Contents:
!Subroutine YouTweak(Mode)
!Subroutine YouTweak3D

!              Generates either of two "golden dataset" at runtime:
!              Zr.Z =       Pr(1).P  * Xr(1).X **      Pr(2).P    Non-Linear
!              Zr.Z = Log10(Pr(1).P) + Xr(1).X * Log10(Pr(2).P)       Linear

!              Using:
!                           Pr(1).P = 3.1_16
!                           Pr(2).P = 2.200000001_16

!Programming environment: Traveler2/Athlon64/WinXPPro-32/APF9.0-32
 !---                                                                 globals
!Defined in Tweak-Begin.f95; all named here, as an overview:
 use Tweakrec,only: jPhase,jMode  ,cVersion,cDateTime & !Tweak's FYI
                   ,jBOn,jUnClamp,jStepMult,jPrev     & !Solution strategy
                   ,jpU10,jpU,jpD           ,cFloat40 & !Printout-Alphanumeric
                   ,jpU3d,TLrec,TL,TLprev,TL2,TLsave  & !Printout-3D
                   ,jTLmorph,TLiter,omjSave           & !         "
                   ,jItertot,jIter,jDone              & !Iteration control
                   ,RSS,RSSbase,Weight,Delta,offon    & !Tweaking & errors
                   ,AbsDet,NoiseFloor,iRank,kPChanged & !Inverter outputs
                   ,jMintot,jMin,StepMult,omj         & !Minimization passes
                   ,B,BtB,BtZ                         & !Allocated matrices
                   ,What,Why,How,Who,When,Where1      & !Project context
                   ,jUserPhase,jUserConfig,cjUserFile & !Use in YouTweak()
                   ,TweakNml                            !Runtime reconfig.
!Tweak-Engine interfaces with the Pr.* and Zr.* using Md & Mdtot.
 Use KPrec,   only: Kptot,Kp,Kp2,  Pr   ,Pu,PstepMag    !Parameters -to fit to-
 Use LZrec,   only: Lztot,Lz,      Zr   ,Zu,Zu2,Z0      !Outputs    - of your -
!YouTweak interfaces with and manages Xr.* and Datae(:,:) ~independently.
 use MDrec,   only: Mdtot,Md,MdMax,Datae                !Dataset
 use NXrec,   only: Nxtot,Nx,      Xr                   !Independent variables
 !---
 implicit none                                                       !arguments
  integer*4:: Mode
  !---                                                               !internals
  integer*4:: Init,i
  real*16  :: Coe1,Coe2,X,Z
   !-----                                                             !end defs
   jMode = Mode !FYI

   select case(Mode)
!   ------------------------------------------------------------------

    case(:0) !Define Nxtot, Lztot, Mdtot, & Kptot: -------------------
     jBOn       =   4 !4    ! jIter< jBOn uses- BmZPartials()
!                           !      >=         - DatapointPartials()
     jUnClamp   =   4 !4    ! jIter< jUnClamp - Pr.Pstep(1&2)- <=1.
!                           !      >=         -             - as is.
     jStepMult  =   2       ! [1:StepMult=1., 2:SelectStepMult()]
!    jPrev      =  -1       ! [-1,0,1]
```

```

     jpU         =  11       ! [0,6,11]
     jpD         =   6 !6    ! [1,2,3,4,5,6,7,8]
     jItertot    =  40       ! [0,...]
     jMintot     =   2       ! [1,...]  0:no minimization.
     Weight      =   1._16   ! Enables equal weighting of all output errors.
!    Delta       =   1.e-6_16! >0., small
     offon       =   1._16   ! Enables all parameters for fitting.

     jTLmorph    =   1       ! onm vertical scaling of 3D error surface

     Kptot =  2 !Number of- parameters
     Lztot =  1 !         - outputs
     Mdtot = 10 !         - datapoints
     Nxtot =  1 !         - independent variables

     What  = "Non-Linear Fit:   Z = P1 * X**P2"c
     Why   = "Introduces the analytical power of `Tweak-Engine`."c
     How   = "Uses Tweak`s `'YouTweak()' interface."c
     Who   = "Model & Datae: Jeff Setterholm"c
     When  = "Tue Oct 09,2018 08:00"c
     Where1= "Lakeville, MN 55044, USA"c
                                                           return !Pass 0
!   -----------------------------------------------------------------

    case(1) !Initialize Xr(), Zr(), Dr(), Pr(), & Datae(): ----------
            !Only called once per run.

     !Synthesize the dataset: Z=3.1*(X**2.200000001), X=[2,3,4,...,11]
     do Md=1,Mdtot
       Datae(0,Md) = quad(Md)                                 !0:Id#
       Datae(1,Md) = quad(Md)+1._16                           !1:X
       Datae(2,Md) = 3.1_16 * Datae(1,Md) **2.200000001_16  !2:Z
     enddo!Md
!    Knowing the exact answer, you can assess Tweak's accuracy.

     write(Xr(0).Xname,"('0:Id#' )") !Independent variable names
     write(Xr(1).Xname,"('1:X'   )")
     write(Zr(1).Zname,"('1:Z'   )") !Output name
     write(Pr(1).Pname,"('1:Coe1')") !Parameter names
     write(Pr(2).Pname,"('2:Coe2')")

     if(Init == 0) then
!      Your initial parameter values:
!      **********************
       Pr(1).Pbase = .0000_16
       Pr(2).Pbase = .0000_16
!      **********************
       if(jpU >= 5) then                         !Display name & value.
         write(jpU,"(/'YouTweak(1):')")
         write(jpU,"( ' Pr( ).Pbase=  Value:',36x,'Name:')")
         do Kp=1,KpTot
           write(jpU,"(i5,1x,e41.32,4x,16a1)")Kp,Pr(Kp).Pbase &
                                ,(Pr(Kp).Pname(i:i),i=1,len_trim(Pr(Kp).Pname))
         enddo!Kp
       endif !jpU>=5
           Init=1
     endif!Init=0

                                                               return !Initdata
!   -----------------------------------------------------------------

    case(2) !Md=0  ~ "Rewind" your dataset: -------------------------
     if(Md /= 0) pause "Md not zero - a potential problem. Press enter."
        Md  = 0
                                                           ;return !Rewind data
!   -----------------------------------------------------------------

    case(3)  !Md=Md+1 - Datae(*,Md) -> Xr.Xdata and Zr.Zdata: -------

```

```
     Xr.Xdata   = Datae(      0:Nxtot      ,Md)
     Zr.Zdata   = Datae(Nxtot+1:Nxtot+Lztot,Md)

     select case(jUserConfig)
      case(1) !Exponential model:
       X = Xr(1).Xdata
       Z = Zr(1).Zdata
      case(2)!    Logarithmic model:
       X = Log10(Xr(1).Xdata)
       Z = Log10(Zr(1).Zdata)
      case default
       pause"YouTweak: jUserConfig not = [1,2]. Press enter to halt."; stop
     end select !jUserConfig

     if((jpU > 5).and.(jpD>=8)) &
       write(jpU,"( 'YouTweak(3): Md=',i2,'  X=',f17.9,'  Z=',f17.9)") &
                                  Md     ,   X        ,   Z
                                                           return !Increment
!   -----------------------------------------------------------------

    case(4) !Exercise your system. Compute Zr.Z using Pr.P: ---------
     Coe1 =  Pr(1).P
     Coe2 =  Pr(2).P

     select case(jUserConfig)
      case(1); Zr.Z = Coe1*(X**Coe2)  !Exponential model
      case(2); Zr.Z = Coe1+ coe2*X    !Logarithmic model
      case default
       pause"YouTweak: jUserConfig not = [1,2]. Press enter to halt."; stop
     end select !jUserConfig

!    Net Output Error & Weighting are done here:
     Zr.Z = (Z - Zr.Z) * Zr.wt

     if((jpU > 5).and.(jpD>=8)) then
       write(jpU,"( '        (4):',6x,    ' C1=',f17.9,' C2=',f17.9)") &
                                          Coe1        ,Coe2
       write(jpU,"(37x,' Zr.Z=',f17.9)") Zr.Z
     endif !(jpU>5,jpD>8)
                                                           return !Update
!   -----------------------------------------------------------------

    case(5) !End of iterative pass -intervention opportunity: -------


!    Report the iteration's ending RSS value:
     if(jpU > 5) write(jpU,"('YouTweak(5): RSS  =',e12.6)") RSS
!    Update and report the numerical partial's Delta value:
     if(Delta >       RSS*10._16)  &
        Delta = 10._16**(floor(log10(RSS))+2._16)
     if(jpU > 5) write(jpU,"(13x,'Delta=',e12.6)") Delta

!    Exporting the solution trajectory to a .3dv visualization file:
     TL.iC = 15; TL.TH = 2.0;
     TL.XYZ = (/ Pr(1).Pbase, Pr(2).Pbase, (-RSS) /)    ;call Morph3dJTL(TL,0)
     if(jIter == 0) TL.iC = 0                           ;call Draw3dJTL( TL,0)
!    Annotating the solution trajectory will be done in case(6).
     if(jIter<=100) TLiter(jIter)=TLprev

                                                    ;return !fit iteration done
!   -----------------------------------------------------------------

    case(6) !Iterating done. Final printout(s) opportunity: ---------
!             You may want to report the results in your format of choice.
!             Case(6) is only called once per run.
     if(jpD > 0) then
!      You can write to unit jpU as follows,
!        or open, write, & close another file (not Unit=jpU & not Unit=10).
       write(jpU,"(/,40('/'))")
       write(jpU,"( 'A YouTweak(6)`Final printout entry.')")
     endif !jpU=11

     call YouTweak3D !This subroutine completes a custom 3D drawing.
                                                           return !Run done
    !                                                      --------------------
    case default;                           stop 'YouTweak: Mode out of range.'
    !                                                      --------------------
   end select !Mode
end Subroutine YouTweak
!----------------------------------------------------------------------------7-9

Subroutine YouTweak3D
!2018.10.08.0755cdt JMS- Custom 3D visualization of the YouTweak model above.
!                      - Traveler2/Athlon64/WinXPPro-32/APF9.0-32

!Use "C" executable: 3DEnv.exe to visualize the Tweak-3dDraw.3dv created here.
!Change App-F5-3dvFileName.ini so that the first line is: "Tweak-3dDraw.3dv"
!Run the .exe, & press F5 to enter 3dv mode.
!  Then, press "M" for an overview of control options.
!  "B" followed by "+"'s steps & recenters you along the iterative solution.
!  "S" and "s" allow you to scale-in and scale-out (generalized "zooming").
!  "~" generates .bmp full color screen dumps at up-to 4K resolution.
!  Type "Q" to quit.

!Output file Tweak-3dJTL.txt is also produced... for your use in exporting
!  the 3D line data into your CAD file format of choice.

 !---                                                                 globals
!Defined in Tweak-Begin.f95; all named here, as an overview:
 use Tweakrec,only: jPhase,jMode  ,cVersion,cDateTime & !Tweak's FYI
                   ,jBOn,jUnClamp,jStepMult,jPrev     & !Solution strategy
                   ,jpU10,jpU,jpD           ,cFloat40 & !Printout-Alphanumeric
                   ,jpU3d,TLrec,TL,TLprev,TL2,TLsave  & !Printout-3D
                   ,jTLmorph,TLiter,omjSave           & !         "
                   ,jItertot,jIter,jDone              & !Iteration control
                   ,RSS,RSSbase,Weight,Delta,offon    & !Tweaking & errors
                   ,AbsDet,NoiseFloor,iRank,kPChanged & !Inverter outputs
                   ,jMintot,jMin,StepMult,omj         & !Minimization passes
                   ,B,BtB,BtZ                         & !Allocated matrices
                   ,What,Why,How,Who,When,Where1      & !Project context
                   ,jUserPhase,jUserConfig,cjUserFile & !Use in YouTweak()
                   ,TweakNml                            !Runtime reconfig.
 Use KPrec,   only: Kptot,Kp,Kp2,  Pr   ,Pu,PstepMag    !Parameters -to fit to-
 !---
 implicit none                                                       !arguments
  !---                                                               !internals
  integer*4:: i,j
  real*16  :: PosLLCq(3),RpyDq(3),SizeHq
  character:: cLabel*80
  integer*4:: iColor
  real*16  :: RSSL
   !-----                                                            !end defs
!*** Add a wealth of detail to the solution trajectory visualization:

!    Show error layers- RSS=0. surface in white, RSS=1.,10.,& 100. in gray:
     PosLLCq = (/  0._16,  .5_16,  0._16 /)  !Define text- location
     RpyDq   = (/  0._16,  0._16, 90._16 /)  !           - & orientation
     SizeHq  =    .1_16                      !Character size
     TL.Th   =   1.0          !Line width in pixels (temporarily has no affect)
     do i=0,5; TL.iC       = 0          !Line color - for move-without-drawing
               TL.XYZ( 3 ) = quad(-i)   !Z-coordinate
               TL.XYZ(1:2) = (/ 0._16, 0._16 /)           ;call Draw3dJTL(TL,0)
               TL.iC = 8;    if(i == 0) TL.iC = 15 !Line color
               TL.XYZ(1:2) = (/ 4._16, 0._16 /)           ;call Draw3dJTL(TL,0)
               TL.XYZ(1:2) = (/ 4._16, 4._16 /)           ;call Draw3dJTL(TL,0)
               TL.XYZ(1:2) = (/ 0._16, 4._16 /)           ;call Draw3dJTL(TL,0)
               TL.XYZ(1:2) = (/ 0._16, 0._16 /)           ;call Draw3dJTL(TL,0)

!      Label the error layer:
       if(i==0) cLabel="RSS=    0. Linear ^"c
       if(i==1) cLabel="RSS=    1. Log10LF ^"c
       if(i==2) cLabel="RSS=   10. Log10LF ^"c
       if(i==3) cLabel="RSS=  100. Log10LF ^"c
       if(i==4) cLabel="RSS= 1000. Log10LF ^"c
       if(i==5) cLabel="RSS=10000. Log10LF ^"c
                             PosLLCq(3)=-quad(i)-.03_16  !Text Z location
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 15, TL.Th , 0)
     enddo!i

!    Label the parameter axes:
     PosLLCq = (/  .2_16,  0._16,  .3_16 /)
     RpyDq   = (/  0._16,  0._16,  0._16 /)
     SizeHq  =    .15_16
     cLabel="Parameter#1 +>"c
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 13, 1. , 0)
     PosLLCq = (/  0._16,  .2_16,  .3_16 /)
     RpyDq   = (/  0._16,  0._16, 90._16 /)
     SizeHq  =    .15_16
     cLabel="Parameter#2 +>"c
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 13, 1. , 0)

!    Tag the first 100 iterations
     RpyDq   = (/  0._16, 45._16, 90._16 /)
     SizeHq  =     .1_16
     do i= 0,jIter-1                                           ;if(i > 100)exit
!      Draw the label connecting line:
       TL  = TLiter(i); TL.iC     =  0                    ;call Draw3dJTL(TL,0)
                        TL.XYZ(2) = TL.XYZ(2) + .5_16
                        TL.XYZ(3) = TL.XYZ(3) - .5_16
                        TL.iC     = 11                    ;call Draw3dJTL(TL,0)
!      Write the 3D label:
                  PosLLCq = TL.XYZ
       write(cLabel,"('Iter',i3)") i
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 11, 1. , 0)
     enddo!i

!    List the first 100 om = values
     RpyDq   = (/  0._16, 0._16, 90._16 /)
     PosLLCq = (/  4._16, 4.1_16, 0._16  /)
     SizeHq  =     .09_16
     write(cLabel,"('    ^iter ^om change')")
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 13, 1. , 0)
     do i= 0,jIter-1                                           ;if(i > 100)exit
       PosLLCq = (/  4._16, 4.1_16, -quad(i+1)*.15_16  /)
       if(i.ne.jBOn) write(cLabel,"(i5,2x,a9)") i, omjSave(i)
       if(i == jBOn) write(cLabel,"(i5,2x,a9,' -jBon')") i, omjSave(i)
       call AlphaJS( cLabel, PosLLCq, RpyDq, SizeHq, 13, 1. , 0)
     enddo!i

!    What,Why,How,Who,When,Where1:
     PosLLCq = (/  0._16,  4._16,  .2_16 /)
     RpyDq   = (/  0._16,  1._16,  0._16 /)
     SizeHq  =    .10_16

     cLabel = What;   PosLLCq(3) = .2_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)
     cLabel = Why;    PosLLCq(3) = .4_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)
     cLabel = How;    PosLLCq(3) = .6_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)
     cLabel = Who;    PosLLCq(3) = .8_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)
     cLabel = When;   PosLLCq(3) =1.0_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)
     cLabel = Where1; PosLLCq(3) =1.2_16
       call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)

!    Free/Public Domain label:
     PosLLCq = (/  0._16,  .1_16,  .6_16 /)
     RpyDq   = (/  0._16,  0._16, 90._16 /)
     SizeHq  =    .2_16
     write(cLabel,"(a1,'Free software',2a1,'Public Domain')") &
                   char(224),char(13),char(10)
     call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq,  7, 1., 0)

     PosLLCq = (/  0._16,  .1_16,  1.2_16 /)
     SizeHq  =    .11_16
     call FDate23(cLabel); cLabel="Runtime: "//cLabel(1:16)//char(0)
     call AlphaJS(cLabel, PosLLCq, RpyDq, SizeHq, 15, 1., 0)

!    Export a 21 x 21 grid of ( Pr.P(1), Pr.P(2), -RSS ) error points to 3dv:
         TL.Th = 1.0
     do i=0,20;    iColor=0
       do j=0,400
         TL.iC =   iColor
         Pr(1).P = 4._16 *i/20._16
         Pr(2).P = 3._16 *j/300._16
         call EvalFit(RSSL, 0)
         TL.XYZ = (/ Pr(1).P, Pr(2).P, (-RSSL) /)    ;call Morph3dJTL(TL    ,0)
         if((j ==   0).and.(mod(i,5) == 0)) then
           TLsave = TL; TLsave.XYZ(3)=0._16          ;call  Draw3dJTL(TLsave,0)
                   iColor = 8 !Start with a dark gray vertical line
           TL.iC = iColor
         endif!Beginning of line with mod(j,5)=0
                                                           call Draw3dJTL(TL,0)
                   iColor = 9 !Dark blue connecting lines
         if((j == 400).and.(mod(i,5) == 0)) then !End with dark gray vertical:
            TL.iC = 8;  TL.XYZ(3)=0._16                   ;call Draw3dJTL(TL,0)
         endif!End of line with mod(j,5)=0
       enddo!j
     enddo!i

     do j=0,20;    iColor=0
       do i=0,400
         TL.iC =   iColor
         Pr(1).P = 4._16 *i/400._16
         Pr(2).P = 3._16 *j/15._16
         call EvalFit(RSSL, 0)
         TL.XYZ = (/ Pr(1).P, Pr(2).P, (-RSSL) /)    ;call Morph3dJTL(TL    ,0)
         if((i ==   0).and.(mod(j,5) == 0)) then
           TLsave = TL; TLsave.XYZ(3)=0._16          ;call  Draw3dJTL(TLsave,0)
                   iColor = 8 !Start with a dark gray vertical line
           TL.iC = iColor
         endif!Beginning of line with mod(j,5)=0
                                                           call Draw3dJTL(TL,0)
                   iColor = 9 !Dark blue connecting lines
         if((i == 400).and.(mod(j,5) == 0)) then !End with dark gray vertical:
            TL.iC = 8;  TL.XYZ(3)=0._16                   ;call Draw3dJTL(TL,0)
         endif!End of line with mod(j,5)=0
       enddo!i
     enddo!j

!    Close the .3dv accumulator & write .3dv file "Tweak.3dv":
     TL.IC = -1;  TL.TH = 0.; TL.XYZ=0._16                ;call Draw3dJTL(TL,0)
!*** Creation of Tweak.3dv is completed.

                                                                         return
end Subroutine YouTweak3D
!---------------------------------------------------------------------------7-9
```