

1 !Tweak-Begin. F95  
2 !2018. 10. 09. 1015cdt JMS- Introducing: "Tweak- Engine",  
3 ! \*\*\*\*=  
4 ! &: Global Variables.  
5 !  
6 ! Jeffrey M Setterholm, 8095 230th St. E., Lakeville, Minnesota 55044, USA  
7 ! I have authored the four Fortran \*.f95 source code files listed below.  
8 ! I hereby place these four files:  
9 ! Tweak-Begin. f95, Tweak- Engine. F95, Tweak-User. F95, & Tweak-Vis. f95  
10 ! and the algorithms which are demonstrated therein,  
11 ! in the public domain (a. k. a.: "free").  
12 ! Disclaimer:  
13 ! \*\*\*\*=  
14 ! \*\*\*\*\* Individual cognition is always flawed, \*\*\*\*\*  
15 ! \*\*\*\*\* including yours & mine. \*\*\*\*\*  
16 ! \*\*\*\*\* - So: - \*\*\*\*\*  
17 ! \*\*\*\*\* Use this code at your own risk. \*\*\*\*\*  
18 ! \*\*\*\*=  
19 !  
20 ! This code realizes my present understanding of:  
21 !  
22 ! Your : Best-Fitting the Continuous Parameters of  
23 ! Your : Linear/Non-Linear Algorithmic Models of  
24 ! Arbitrarily Large Datasets  
25 ! with Extreme Accuracy.  
26 !  
27 ! When YouTweak. f95 is tailored for your purposes:  
28 !  
29 ! You can determine your math model's adequacy,  
30 ! or need for improvement, in accounting for your data.  
31 !  
32 ! If you enjoy searching, - and-  
33 ! when these algorithms are subsequently perfected,  
34 ! you'll find that:  
35 !  
36 ! Mother nature's error evaluations of math models  
37 ! are delightfully impartial, non-intrusive, and deeply insightful...  
38 ! much like the smiles & frowns of the great teachers in the Himalayas.  
39 !  
40 ! \*\*\*\*=  
41 !  
42 ! Tweak-Begin. f95 organizes the key variables within Tweak. exe  
43 ! An overview of the Engine (timestamps & sizes are approximate):  
44 !  
45 ! Lines Bytes  
46 ! 10/08/2018 05:30 AM 3, 560 Tweak. gui <- Absoft's GUI  
47 ! 10/08/2018 11:54 AM ~470 23, 907 Tweak-Begin. f95 <- This file  
48 ! Modul e TweakRec  
49 ! Modul e KPrec  
50 ! Modul e LZrec  
51 ! Modul e MDrec  
52 ! Modul e MDrec  
53 ! Subroutine TweakAlllocateAll  
54 ! 10/08/2018 08:11 AM ~1260 60, 540 Tweak- Engine. F95 Includes:  
55 ! Program Tweak  
56 ! Subroutine EvalFit(RSSL, iP)  
57 ! Subroutine PrintIter(iP)  
58 ! Subroutine RSSPartials(jpUL)  
59 ! Subroutine DatapointPartials  
60 ! Subroutine Invert(N, A, ValMin, iRank...)  
61 ! Subroutine PrintA(N, A, Noise, iRank, ...)  
62 ! Subroutine SelectStepMult(iP)  
63 ! Function om(Value1, Value0, iP)  
64 ! Subroutine FloatWrite(R16In, a40out)  
65 ! Subroutine FDate23(DaTi me)  
66 ! Subroutine Beamer(n, nTot)  
67 ! 10/08/2018 08:14 AM ~400 19, 167 Tweak-User. F95 <- Your Model goes here  
68 ! Subroutine YouTweak(Model)  
69 ! Subroutine YouTweak3D  
70 !

```

71 ! 10/08/2018 07:22 AM ~850 50,436 Tweak-Vi.s.f95 <- 3D Visualization
72 !
73 !
74 !
75 !
76 !
77 !
78 !
79 !
80 ! 10/08/2018 12:27 PM 1,037,824 Tweak.exe
81 ! 10/08/2018 11:58 AM 1,621 Tweak-User.nml <- For mods @ runtime
82 !
83 ! 10/08/2018 12:03 PM 141,391 Tweak.txt <- Reports the process
84 ! 10/08/2018 06:18 AM 1,301 Tweak-Params.txt <- Reports parameters
85 !
86 ! 10/08/2018 12:03 PM 1,412,272 Tweak-3dDraw.3dv <- 3D output.
87 !
88 ! ~9 File(s) ~3,089,439 bytes
89 !
90 ! Subdirectory /Images has 23 files of five solution trajectories.
91 !
92 ! ****
93 !
94 ! Five "modules" follow herein. The summarizing 'use' statements are:
95 !--- global s
96 !use Tweakrec, only: jPhase, jMode , cVersion, cDateTi me & !Tweak's FYI
97 ! ,jB0n, jUnClamp, jStepMult, jPrev & !Solution strategy
98 ! ,jPU10, jPU, jPB ,cFloat40 & !Printout-Alphanumeric
99 ! ,jPU3d, TLrec, TL, TLprev, TL2, TLsave & !Printout-3D
100 ! ,jTLmorph, TLiter, omjSave & !
101 ! ,jItertot, jIter, jDone & !Iteration control
102 ! ,RSS, RSSbase, Weight, Delta, offon & !Tweaking & errors
103 ! ,AbsDet, NoiseFloor, iRank, kPChanged & !Inverter outputs
104 ! ,jMinTot, jMin, StepMult, omj & !Minimization passes
105 ! ,B, BTB, BTZ & !Allocated matrices
106 ! ,What, Why, How, Who, When, Where1 & !Project context
107 ! ,jUserPhase, jUserConfig, cjUserFile & !Use in YouTweak()
108 ! ,TweakNml & !Runtime reconfig.
109 !Use KPrec, only: Kptot, Kp, Kp2, Pr ,Pu, PstepMag & !Parameters - to fit to-
110 !Use LZrec, only: Lztot, Lz, Zr ,Zu, Zu2, Z0 & !Outputs - of your -
111 !use MDrec, only: Mdtot, Md, MdMax, Datae & !Dataset
112 !use NXrec, only: Nxtot, Nx, Xr & !Independent variables
113 !
114 ! ^*: These are allocated records.
115 !
116 !
117 !Table of Contents:
118 !Module TweakRec !Program control and various summary values:
119 !Module KPrec !Parameters:
120 !Module LZrec !Outputs:
121 !Module MDrec !dataset storage & access:
122 !Module MDrec !dataset storage & access:
123 !
124 !Subroutine TweakAllocateAll !Allocates Tweak's arrays.
125 !----- 7 9
126 !////////////// 127
127 !
128 Module TweakRec !Program control and various summary values: !use TweakRec
129 ! = j - integers & various reals
130 !
131 !Tweak's FYI
132 ! character: :jPhase*80 !Marks key phases of the fitting process. Purely: FYI
133 ! For Your Information
134 ! = "0: Allocates records based on Kptot, Lztot, Mdtot, & Nxtot" c
135 ! in YouTweak(0)
136 ! = "1: You quantify your system model in the allocated arrays" c
137 ! in YouTweak(1)
138 ! YouTweak(2, 3, & 4) - exercised by the Tweak engine.
139 ! = "2: Top-of-each iteration: establish baseline residual errors" c
140 ! = "3: Populate BTB(Kptot, Kptot) and BTZ(Kptot)" c

```

```

141 !          ="4: Normalize and invert BtB(Kptot, Kptot)"c
142 !          ="5: Inverters stepsize: Pr. Pstep = BtBinv(denormalized)*BtZ"c
143 !
144 !          ="3, 4, & 5: RSSPartials uses RSS error gradient instead."c
145 !
146 !          ="6: Find an error-reducing actual stepsize"c
147 !          ="7: Iterative pass done. Your intervention opportunity:"c
148 !                      in YouTweak(5)
149 !          ="8: Tweak is done iterating. Export the results & close"c
150 !                      in YouTweak(6)
151
152 integer*4::j Mode !Interface of Tweak() to YouTweak().           Purely: FYI
153 !          = 0: Define Nxtot, Lztot, Mdtot, & Kptot:
154 !          = 1: Initialize Xr(), Zr(), Dr(), Pr(), & Datae():
155 !          = 2: Md=0 ~ "Rewind" your dataset:
156 !          = 3: Md=Md+1 - Datae(*,Md) -> Xr, Xdata and Zr, Zdata:
157 !          = 4: Exercise your system. Compute Zr, Z using Pr, P:
158 !          = 5: End of iterative pass -intervention opportunity:
159 !          = 6: Iterating done. Final printout(s) opportunity:
160
161 character::cVersion*60 & !Present software version
162 !          ="Tweak.Exe, version: 1.0, 2018.10.09, JMS, Public Domain"c
163
164 character::cDateTime*23!Current date & time.
165
166 !Solution strategy:
167 integer*4::j B0n      = 3 !Iteration# to start calling DatapointPartials()
168 !                      For jIter<jB0n the call is to RSSPartials() instead
169 !          = 1 : Will probably work well for linear models.
170 !          > 1 : A few iterations using RSSPartials()
171 !                      hastens the convergence of non-linear models.
172
173 integer*4::j UnClamp = 3 !Number of iterations to restrict Pr. Pstep to <= 1.
174
175 integer*4::j StepMult= 2!Minimization algorithm - computes StepMult.
176 !          = 1! - StepMult=1.0 . e.g.: using inverter's result
177 !          = 2! - calls SelectStepMult()
178
179 character*9::omj &
180 !          ='undefined' !Expresses the value1/Value0 ratio of positive numbers
181 !                      in exponential form, & places the signed exponent
182 !                      on the left side of the result.
183 !          !Exponents > 0 represent ratios >= 1.
184 !          !Exponents <=0 represent ratios < 1.
185 !          !1. > mantissa >= .1, except when Value1 = 0.
186 !          !"__sme.nnn"
187 !          !"__smme.nnn"
188
189 integer*4::j Prev     =-1!Use previous parameter values in Tweak-Params.txt?
190 !          =-1: no
191 !          = 0: ask
192 !          = 1: yes
193 !          Ordinarily you'll solve each problem from scratch.
194 !          For problems whose iteration time is more than a few
195 !          minutes, easy resumption is convenient.
196 !Printout-Alphanumeric:
197 integer*4::jpU10    =10 !Unit number- for Tweak's parameter reads & writes
198 !                      to/from "Tweak-Params.txt"
199 integer*4::jpU       =11 !Unit number- Output of general information
200 !          = 6 : outputs to DOS screen
201 !          >=11 : writes/appends "Tweak.txt"
202 integer*4::jpD       = 4 !Print Detail level, use: "if(jpD >= ____)"
203 !          = 0 : Disabled, e.g: when jpU<6
204 !          = 1 : User selected important information/program controls
205 !          = 2 : Enables screen/keyboard/program interactions
206 !          = 3 : Report/summarize- final iteration
207 !          = 4 :                  - each iteration
208 !          = 5 :                  - some main program info.
209 !          = 6 :                  - completed matrices & minimization
210 !          = 7 :                  - some subroutine details

```

```

211 ! = 8 : - in-depth details
212 character::cFloat40*40 !used for: call FloatWrite(R16In, cFloat40) printing
213
214 !Printout-3D: -----
215 !YouTweak can interface with 3D visual data collector Draw3dJTL().
216 ! A minimal interface follows.
217 !If no calls to Draw3dJTL() are made, then TweakVis.f95 isn't needed
218 ! to compile & link Tweak.exe.
219 integer*4::jP3d = 13 !Unit number- for exporting 3D viewing coordinates.
220
221 integer*4::jTlMorph = 1!Morph3dJTL() mode control.
222 ! Morph3dJTL() centralizes the arbitrary rescale
223 ! (~morphing) of TL XYZ plot points.
224 !
225 !
226 ! Example uses:
227 ! = 0: Z-axis- no change
228 ! = 1: - log10(-Z) | linear | +log10(+Z) :Z
229 ! = 2: - 2. - 1./Z | linear | - 2. + 1./Z
230 ! = 3: - User defined morph of X, Y, & Z.
231
232 type:: TLrec ! "Thick Line" record
233 integer*4:: iC ! DOS color of the line [0, 15] to the next dot
234 ! = 0: move-to without drawing
235 ! = -1: closes the file
236 real*4 :: Th ! Line thickness [1.0 - 20.0] & width in pixels
237 real*16 :: XYZ(3) ! (X, Y, Z) "next dot"
238 end type TLrec;
239
240 type(TLrec):: TL !Draw3dJTL- Your "next dot" input
241 type(TLrec):: TLprev ! - the most recent dot added
242 type(TLrec):: TL2 ! - "next dot" used by AlphaJS
243 type(TLrec):: TLSave !A place to save TLprev during dot digressions.
244 type(TLrec):: TLiter(0:100)
245 character*9:: omjSave(0:100) !omj values of the first 100 iterations
246 !-----
247 !Iteration control:
248 integer*4::jIterTot=30!Fit iterations- total
249 integer*4::jIter ! - current #
250 integer*4::jDone != 0: will do another iteration
251 !> 0: reduced error not found.
252
253 !Evaluation:
254 real*16 :: RSS !Root-Sum-Squared- present
255 real*16 :: RSSbase ! - previous iteration
256 !Numerical Partial Derivative default values:
257 real*16 :: Weight = 1._16 !Nominal output weighting factor
258 ! = 0._16 - set individually
259 real*16 :: Delta = 1.e-6_16 !Numerical partial tweak-
260 ! - adjusted in YouTweak(5)
261 real*16 :: offOn = 0._16 !Defaults to all- on - set individually
262 ! = 1._16 ! - off- set individually
263
264 !Inversion output values:
265 real*16 :: AbsDet !Determinant- absolute value
266 real*16 :: NoiseFloor !Inversion noise floor
267 integer*4:: iRank !BtB's rank, < Ktot if linearly dependent
268 integer*4:: kPChanged !# of parameters whose values changed.
269
270 !Minimization passes to determine the step multiplier:
271 integer*4::jMinTot = 20
272 integer*4::jMin
273 real*16 :: StepMult !Pr. Pnew = Pr. Pbase + SetpMult * Pr. Pstep
274
275 real*16 , allocatable:: B(:, :) !(Lztot, Ktot) Numerical partials array
276 real*16 , allocatable:: BtB(:, :) !(Ktot, Ktot) Outer product array
277 real*16 , allocatable:: BtZ(:,:) !(Ktot, )
278
279 !Project Context: Consider having your dataset(s) define these:
280 character:: What *60 ! & !Title = "Title this particular parameter fit."c

```

```

281 character:: Why *60 ! & !Objective= "Briefly describe your objective(s)."c
282 character:: How *60 ! & !Approach = "Briefly describe your approach."c
283 character:: Who *60 ! & !Author(s)= "The model & dataset authors."c
284 character:: When *60 ! & !date/time= "The model/dataset date?"c
285 character:: Where1*60 ! & !Location = "Describe the location."c
286 !
287 ^^ : 60 characters max
288
289 !User-assignable variables, for use in YouTweak()
290 integer*4:: jUserPhase
291 integer*4:: jUserConfig
292 character:: cjUserFile*60
293
294 !Name list interface to TweakRec - for runtime reconfiguration of the .exe:
295 NAMELIST / TweakNml /
296     jB0n, jUnClamp, jStepMult
297     ,jPjU, jPjD
298     ,jTLmorpH
299     ,jItertot
300     ,jDelta
301     ,jMintot
302     ,jWhat, jWhy, jHow, jWho, jWhen, jWhere1
303     ,jUserPhase, jUserConfig, cjUserFile
304 !Contains
305 End Module TweakRec
306 !/////////////////////////////////////////////////////////////////
307
308 Module KPrec      !Parameters:                                     !Use KPrec
309 !       ... the "parameters" of "parameter identification".
310 !       = P's with K indices
311 !Size & indexes:
312 integer*4:: Kptot    !Parameter- Total number
313 integer*4:: Kp        !          - # in use
314 integer*4:: Kp2       !          - # in use
315
316 !Parameter properties:
317 type:: Prec          !
318 real*16 :: P          !
319 real*16 :: Pbase       !
320 real*16 :: offon & !
321 !           = 1. _16!
322 real*16 :: Delta       !
323 ! real*16 :: Pstep      !
324
325 real*16 :: Pstep(4, 2)!           - Step size- (1, 1): from RSS
326 !                                         (1, 2): scaled/bounder
327 !                                         (2, 1): from BtBInv*BtZ "Inv"
328 !                                         (2, 2): scaled/bounder
329 !                                         (3, 1): <>unused>>
330 !                                         (3, 2): <>unused>>
331 !                                         (4, 1): input to minimizer
332 !                                         (4, 2): Minimizer's net tweak
333 real*16 :: Pnew       !
334 real*16 :: Vstep       !
335 integer*4:: Inverted   !
336 !
337 integer*4:: Kp          !
338 integer*4:: iSparse(2)  !
339 character:: Pname*16   !
340 real*16 :: BtBnorm !y Diagonal of BtB()- used as a normalizing factor
341 end type Prec
342 type(Prec)             :: Pu, P0
343 type(Prec), allocatable:: Pr(:)      !(KpTot)
344
345 real*16 :: PstepMag !Magnitude of the Pr. Pstep(s) vector
346 !Contains
347 End Module KPrec
348
349 !Adjustment of Delta is shown in YouTweak(5)
350 !/////////////////////////////////////////////////////////////////

```

```

351
352 Module LZrec      ! Outputs:                                     ! Use LZrec
353 !           = Z's with L indices:
354 ! Size & indexes:
355   integer*4::Lztot    ! Output- Total number
356   integer*4::Lz       ! - # in use
357 ! Output properties:
358   type::Zrec          ! - type:
359   real*16::Z           !y - value- in use
360   real*16::Zbase      !y - baseline at iteration start
361   real*16::Wt          !y - Weight of output in the fit, Nominally 1.0
362   real*16::WZ          !y - Weight*value
363   real*16::Zdata      !y - value- of/from the current Md datapoint
364 !                                           Managed & used by YouTweak()
365   real*16::RMS         !y - error- RMS- present
366   real*16::RMSbase    !y - at iteration start
367   real*16::Zmax        !y - max- value- computed by EvalFit()
368   integer*4::Mdmax     !y - # of this Zr(Lz)
369   integer*4::Lz          ! - Name
370   integer*4::ispare(2) !
371   character::Zname*16 !
372   integer*4::L          !
373 end type Zrec
374 type(Zrec)           ::Zu, Zu2, Z0
375 type(Zrec), allocatable::Zr(:) !(Lztot)
376 !Contains
377 End Module LZrec
378 !////////////////////////////////////////////////////////////////
379
380 Module MDrec      ! Dataset storage & access:                      ! use MDrec
381 !           = Datae with M indices:
382 ! Size & indexes:
383   integer*4::Mdtot     ! Datapoints- total number in the Dataset
384   integer*4::Md          ! - # in use
385   integer*4::Mdmax      !
386   real*16, allocatable::Datae(:, :) !(0:Nxtot+Lztot, Mdtot)
387 !                                         ^0 stores the dataset index as a real*16
388 !Contains
389 End Module MDrec
390 !////////////////////////////////////////////////////////////////
391
392 Module NXrec      ! Independent variables of your dataset:          ! use NXrec
393 !           = X's with N indices:
394
395 ! Size & indexes:
396   integer*4::Nxtot     ! Variables- Total number
397   integer*4::Nx          ! - # in use
398 ! Variables properties:
399   type::Xrec            !
400   real*16::Xdata        !
401 !                                         - type:
402   integer*4::Kcoes      !
403   integer*4::Nx          !
404   integer*4::ispare(2) !
405   character::Xname*16 !
406 end type Xrec
407 type(Xrec)           ::Xu, X0
408 type(Xrec), allocatable::Xr(:) !(Nxtot)
409 !Contains
410 End Module NXrec
411 !////////////////////////////////////////////////////////////////
412 !-----7 9
413
414 Subroutine TweakAllocateAll !Allocates Tweak's arrays.
415 ! 2018.09.25. 2010cdt JMS- Traveler2/Athlon64/WinXPPro-32/APF9.0-32
416 use TweakRec, only:B, BtB, BtZ                                ! Globals
417 use KPrec
418 use LZrec
419 use MDrec
420 use NXrec

```

```

421 !---  

422 implicit none  

423 !---  

424 integer*4::iAlloc  

425 !----  

426 if(Kptot<=0) then; write(6, "(1x, 'Tweak halt! Kptot=', i9)") Kptot ; stop; endif  

427 if(Lztot<=0) then; write(6, "(1x, 'Tweak halt! Lztot=', i9)") Lztot ; stop; endif  

428 if(Mdtot<=0) then; write(6, "(1x, 'Tweak halt! Mdtot=', i9)") Mdtot ; stop; endif  

429 if(Nxtot<=0) then; write(6, "(1x, 'Tweak halt! Nxtot=', i9)") Nxtot ; stop; endif  

430 !-----  

431 allocate(Pr(Kptot), stat=iAlloc) !Parameters array  

432 if(iAlloc /= 0) then  

433 pause 'TweakAllocateAll: Pr(Kptot) allocation error. `Enter` to exit.'  

434 stop  

435 endif!(iAlloc/=0)  

436 Pr(1:Kptot)=P0; Pr.Pname=char(32)  

437 !-----  

438 allocate(Zr(Lztot), stat=iAlloc) !Outputs array  

439 if(iAlloc /= 0) then  

440 pause 'TweakAllocateAll: Zr(Lztot) allocation error. `Enter` to exit.'  

441 stop  

442 endif!(iAlloc/=0)  

443 Zr(1:Lztot)=Z0; Zr.Zname=char(32)  

444 !-----  

445 allocate(Datae(0:Nxtot+Lztot, Mdtot), stat=iAlloc) !Data array  

446 if(iAlloc /= 0) then  

447 pause 'TweakAllocateAll: Datae() allocation error. `Enter` to exit.'  

448 stop  

449 endif!(iAlloc/=0)  

450 Datae=0._16  

451 !-----  

452 allocate(Xr(0:Nxtot), stat=iAlloc) !Independent variables array  

453 if(iAlloc /= 0) then  

454 pause 'TweakAllocateAll: Xr(Nxtot) allocation error. `Enter` to exit.'  

455 stop  

456 endif!(iAlloc/=0)  

457 Xr(0:NxTot)=X0; Xr.Xname=char(32)  

458 !-----  

459 allocate(B(Lztot, Kptot), stat=iAlloc) !B: Numerical partials array  

460 if(iAlloc /= 0) then  

461 pause 'TweakAllocateAll: B(Lztot, Kptot) alloc. error. `Enter` to exit.'  

462 stop  

463 endif!(iAlloc/=0)  

464 B=0._16  

465 !-----  

466 allocate(BtB(Kptot, Kptot), stat=iAlloc) !Bt*B: Outer product array  

467 if(iAlloc /= 0) then  

468 pause 'TweakAllocateAll: B(Lztot, Kptot) alloc. error. `Enter` to exit.'  

469 stop  

470 endif!(iAlloc/=0)  

471 BtB=0._16  

472 !-----  

473 allocate(BtZ(Kptot), stat=iAlloc) !Bt*Z array  

474 if(iAlloc /= 0) then  

475 pause 'TweakAllocateAll: BtZ(Kptot) allocation error. `Enter` to exit.'  

476 stop  

477 endif!(iAlloc/=0)  

478 BtZ=0._16  

479 ! ----- Array allocations completed.  

480 End Subroutine TweakAllocateAll  

481

```

return