FFFwNPF

Fee - Foe - Fum's Well-nigh Perfect Fit

FFFWNPF

A: Purpose

This is an efficient program to either

(i) Minimize X^2:

$$X^2 = \sum_{i=1}^{NT\emptyset T} f_i^2 \quad (\theta_1 \ldots \theta_{NPAR}) \tag{1}$$

as a function of the NPAR parameters $\theta_1 \ldots \theta_{NPAR}$. Typically $f_i = (t_i - e_i)/\sigma_i$ where the experimental values e_i have error σ_i and theoretical estimates $t_i = t_i(\theta_1 \ldots \theta_{NPAR})$.

(ii) or Maximize Likelihood:

$$-2\ell nL = -2 \sum_{i=1}^{n} \ell n\, q_i \tag{2}$$

Here we have n observations of probability q_i.

There are three basic varieties.

(a) q_i is natural probability of event:

$$L = \prod_{i=1}^{n} (q_i) \tag{2a}$$

(b) The natural probability p_i is not normalized and

$$q_i = p_i/\sigma \tag{2b}$$

where $\sigma = \int_v p\,dv$: the integrated probability over the volume v in which events are observed.

(c) The case (b) corresponds to this possibility that p_i is the unnormalized natural probability but the normalization σ is not to be fitted. If we wish to fit σ then:

$$L \propto \exp\left[-\sigma t\right] \prod_{i=1}^{n} (p_i) \tag{2c}$$

where t is the "events/μb" normalized so that

$$\langle n \rangle = \sigma t \tag{3c}$$

This can be written in the form (2) by putting

$$q_i = p_i \exp\left[-\sigma t/n\right] \tag{4c}$$

Given the initial values of θ and a subroutine to calculate f_i/p_i and their derivatives $\partial(f_i/p_i)/\partial\theta_j$, the program returns the best values of θ and the errors in their determination.

$$\mathrm{err}_j = 2\left|\left\{\left[\frac{\partial^2\psi}{\partial\theta_\ell \partial\theta_m}\right]^{-1}\right\}_{jj}\right|^{\frac{1}{2}} \tag{5}$$

which is change in θ_j which should change a charge of 1 in ψ. (Here $\psi = X^2$ or $-2\ell n\ L$).

The advantages of having one routine to handle both X^2 and minimum unlikelihood fits is described in Section E.

Section B describes the use of the program. Section C describes the method used in parameter variation. Section D lists the preset variables the user can change to affect parameter search (this is identical in either X^2 or m. l. use).

B: Use of FFFWNPF

(i) To initialize, one must use CALL PRESET and then optionally

(ii) reset TITLE by input on 8A10 format

(iii) reset fitting options by CALL GCFIN which reads namelist $GCFFIT...$END with specification of preset parameters for fitting routine. These are always given standard values before namelist input and so one need only input those that are to be changed. This is detailed in Section D.

Then one must

(iv) set initial values of PARAM the parameter values. PARAM is in the big common block /MAYMAD/ whose size can be changed as number of parameters/data values f_i change but is typically

```
 COMMON/MAYMAD/TITLE(10),NDIM,OUTAPE,PARAM(40),FIRST(40,2),
1 SECOND(40,40,2),FUNC(1000),TEST(1002),FBEST(1000),VECTOR(40,40),
2 EXTRA(40),XBEST(40),SHIFT(40),SPARE(40,40),SAVEV(40,40),
3  AGNES(40),SAVEE(40),STATSA(40)
 INTEGER OUTAPE
 DIMENSION FIND(1)
 EQUIVALENCE (FIND,PARAM)
```

Other interesting variables in /MAYMAD/ are given in (viii).

(v) If maximum likelihood mode of operation desired, one must set variables in CØMMØN/MAXLIK/LIKØPT, EVMUB.

Here LIKØPT = 0 : X^2 mode (preset)

= 1 : as eqn. (2a)

= 2 : as eqn. (2b)

= 3 : as eqn. (2c)

LIKØPT = 3 also needs one to set EVMUB = t where t enters in eqns. (2 → 4c).

(vi) Finally one calls FFFWNPF by:

CALL GCFGØ(NPAR,NTØT,CALCFG,N)

where:

NPAR = number of theoretical parameters.

NTØT is number of f_i in X^2 mode. In m.l. mode, NTØT is ignored unless derivative testing requested when latter is done for first NTØT events.

N = 0 always. N ≠ 0 used to specify special large core store mode which saved /MAYMAD/ there and allowed overwriting of small core /MAYMAD/. This option is not implemented.

CALCFG , which must be declared EXTERNAL, is name of user routine which calculates f_i or p_i . This goes as follows:

(vii) SUBRØUTINE CALCFG(VIØLAT) must

a) Set logical variable VIØLAT = .FALSE. if parameters are in an acceptable range. or set VIØLAT = .TRUE. if parameters are out of range and have been altered to accord with sensible limits. These parameters are stored in PARAM(1..>NPAR) in /MAYMAD/.

b) For each f_i/p_i CALCFG must (sequentially)

CALL ACCUM (DERIV,VALUE)

where VALUE = f_i/p_i

DERIV (j) = $\partial(f_i/p_i)/\partial\theta_j$

In the case of LIKOPT = 2,3, FFFWNPF assumes that in first call to ACCUM, VALUE = σ and DERIV(j) = $\partial\sigma/\partial\theta_j$.

(viii) At the end of run, FFFWNPF returns in

FUNC (1..NTØT) : Best values of f_i (X^2 mode only)

PARAM (1..NPAR) : Best parameter values

TEST (1..NPAR) : Corresponding parameters errors

TEST (NPAR + 1) : X^2 or $-2\ell nL$

Note TEST is used in derivative testing as a temporary store; it must have a dimension $\geq$ NTØT.

(ix) Note that if TRIPE - see Section D - = .TRUE., program will deluge you with output but if TRIPE = .FALSE., it will give you only timing information.

C: Method Used

The basis of the calculational method is the usual trivial second-order expansion.

$$\psi = \psi_0 + A(\underline{\theta} - \underline{\theta}_0) + (\underline{\theta} - \underline{\theta}_0)^T B(\underline{\theta} - \underline{\theta}_0) \qquad (C0)$$

Here the vector $\underline{\theta}$ holds theoretical parameters and $\underline{\theta}_0$ is some guess for $\underline{\theta}$. $\psi = X^2$ or $-2\ell nL$ as usual,

rigorously

$$A_i = \partial\psi/\partial\theta_i\big|_{\underline{\theta} = \underline{\theta}_0} \qquad (C1)$$

$$B_{ij} = \tfrac{1}{2}\, \partial^2\psi/\partial\theta_i\partial\theta_j\big|_{\underline{\theta} = \underline{\theta}_0} \qquad (C2)$$

Rather than (C2), one employs X^2 :

$$B_{ij} = \sum_{\ell} \partial f_\ell/\partial\theta_i \;\; \partial f_\ell/\partial\theta_j \qquad (C3)$$

dropping the $f_\ell \;\; \partial^2 f_\ell/\partial\theta_i\partial\theta_j$ part of B.

m.l. method:

$$B_{ij} = \sum_{k} (\partial q_k/\partial\theta_i)(\partial q_k/\partial\theta_j)(q_k)^2 \qquad (C4)$$

dropping a $\partial^2 q_k/\partial\theta_i\partial\theta_j$ term (C4)

In each case, the term dropped can be shown to be negligible near the true minimum $\theta = \theta_{min}$. Further the approximate forms (C3), (C4) have important properties not shared by (C2).

(i) B is positive definite

(ii) B can be calculated using only the <u>first</u> derivative of f or q.

Solving (C0) gives the new guess for minimum.

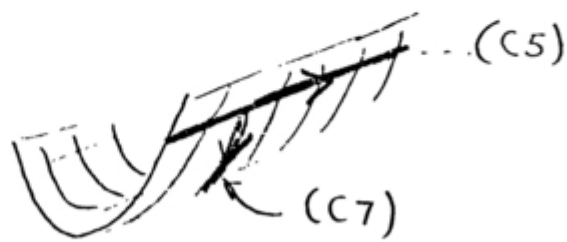
$$\underline{\theta} - \underline{\theta}_0 = -\tfrac{1}{2} B^{-1} A \qquad \text{(C5)}$$

The iteration of (C5) is called "simple $2^{\underline{nd}}$ -order method". A modification of (C5) was suggested by Powell. It puts

$$\underline{\theta}(\lambda) - \underline{\theta}_0 = -\tfrac{1}{2} \lambda B^{-1} A \qquad \text{(C6)}$$

where $\lambda = 1$ gives previous method, (C6) gives $\psi = \psi(\lambda)$ and Powell minimizes ψ with respect to the single parameter λ to find new guess for minimum. This is intelligent because there are very good and efficient algorithms for minimizing functions of one variable. So (C6) is termed "Powell's method".

In practice, it does not work for often the plot of ψ v. θ has the form sketched below:



This valley implies ψ is essentially independent of one combination of parameters (that marked "(C5)") but depends strongly on another "(C7)". The "second order" and "Powell's method" search only along the badly determined direction and never converge.

There are two simple solutions. The best is termed the "Eigenvalue method". If one diagonalizes B, the direction "(C5)" represents a combination of parameters corresponding to a small eigenvalue of B while "(C7)" represents an eigenvector with a large eigenvalue. The "Eigenvalue method" essentially varies all the large eigenvectors first and then the small ones, i.e., it solves

$$\underline{\theta} - \underline{\theta}_0 = -\tfrac{1}{2} B'^{-1} A' \qquad \text{(C7)}$$

where B', A' are found from B and A by deleting all components of eigenvectors with small eigenvalues. Pictorially it first goes down the steep side of the valley and only then wanders around the flat undetermined bottom. This method nearly always converges and is not sensitive to poorly determined parameters and/or too many parameters.

A less accurate formulation of this idea is the well-known "method of steepest descents". This puts

$$\underline{\theta} - \underline{\theta}_0 = \mu A \qquad \text{(C8)}$$

and finds μ by minimizing (C0) with this direction for $\underline{\theta} - \underline{\theta}_0$, i.e.,

$$\mu = -\tfrac{1}{2} \; A^T A / A^T B A \tag{C9}$$

This will avoid the divergence difficulties of the first two methods but it can essentially vary in only one direction. Clearly (this is borne out in practice) the "eigenvalue" method will cope better with the many parameter situation with many directions of various degrees of determination (i.e., various sizes of eigenvalues).

Historically we tried all these methods and found the eigenvalue method best. However, as described in Section D, the program FFFWNPF still suffers from the ability to use all four.

Also note that the "eigenvalue method" actually utilizes Powell's idea, i.e., we put - rather than (C7) -

$$\underline{\theta} - \underline{\theta}_0 = -\tfrac{1}{2} \lambda \; B'^{-1} \; A' \tag{C7*}$$

and find λ by minimizing ψ as a function of one variable.

FFFWNPF has built in a set of powerful routines for minimizing as a function of one variable. (Taken directly from the routines coded by Powell - these worked unlike his many parameter routines). One could use these separately if all you need is to minimize as a function of one variable. However, the full package - at the cost of

some overhead - does work if only one parameter and so no special action is necessary.

D: NAMELIST GCFFIT

Variable	Preset	Meaning
MAXIT	20	No. of function evaluations allowed.
TIMMAX	2.	Maximum time allowed in minutes.
STUPID	1. E50	Stop if initial ψ > STUPID. Useful for restart jobs to check no foolish parameter punching error.
TRIPE	F if GCFNO T if GCFIN	If T, write output. If F, suppress output.
IDTEST	1	= 1, no derivative test. = 2,3 derivative test all parameters and stop. = 4,5 as 2,3 but go onto to fit afterwards. If IDTEST = 3,5 then only parameter i ($1 \leq i \leq$ NPAR) with IDERIV(i) = -1 will be tested and then step length of numerical differentiation will be taken as PSTEP(i).
PSTEP IDERIV	Unset	see IDTEST
IEIG1	0	Ignore first IEIG1 eigenvalues.
SCASET	F	If T, STYPE and SCPREV have been set.
STYPE ($1 \leq i \leq$ NPAR) an integer variable	Unset	If ith element has value: -2 fix ith eigenvalue. n > 0 vary with nth scale factor together with other eigenvalues of same n. 0 vary as usual.

SCPREV ($1 \leq j \leq$ MLOOP)	Unset	Specifies scale factor of jth eigenvalue set. If value > 0, it is scale factor. If ≤ 0, there is no preset scale factor in this group.
NEWPAR ($1 \leq L \leq 9$)	-1	List of upto 9 new parameter numbers. STYPE as read in corresponds to old parameters. These extra parameters must be varied appropriately.
NOERR	5	No. of consecutive boundary violations before error STOP.
IRANDY	2	No. of function evaluations allowed for different methods.
IPOWL	5	
ISTEEP	3	
IEIGEN	5	
STEEP	T	If F , use eigenvalue before steepest descent method. Only allow MSTEEP calls of STEEP. Stop STEEP section when target ψ DELTA > shift.
MSTEEP	3	
DELTA	.05	
DOPOWL	F	If T , an ordinary POWELL fit is attempted before STEEP or EIGEN whether or not preset conditions allow it.
RANGE	.1	In EIGEN, vary together those eigenvalues in a range RANGE of value.
MLOOP	5	Maximum no. of divisions allowed for separate eigenvalue variation.
ISPLIT(i) ($1 \leq L \leq$ MLOOP)	-1	If set fails then if ISPLIT(i) = -1 fix parameters = 0 deem parameters unvaried and put back into hat $\geq$ 1 keep parameters together but scale by 1/4.
FACTOR	2.	Target ψ = FACTOR* No. of degrees of freedom (X^2 mode). Target ψ = FACTOR (maximum likelihood mode).
CHEMIN	.01	Stop when $\Delta\psi \leq$ CHEMIN* Target ψ .
OVERAL	1.	Scale all shifts by OVERAL.
DONT	F	If T , never scale shifts.
OVRRID	F	If T , ignore any CALL GCFNO.

Notes:

(i) $\psi = X^2$ or $-2\ell nL$ depending on mode.

(ii) The different methods are described in Section C. In that notation:

IRANDY is number of "Simple 2^{nd}-order method" trials

IPOWL is number of "Powell method" trials

ISTEEP is number of "steepest descent" trials

IEIGEN is number of "Eigenvalue method" trials

Here trials = number of CALCFG calls allowed in each call of this section of FFFWNPF.

The methods are tried in the above order unless number of trials = 0 when method skipped or STEEP = F when "eigenvalue method" used before "steepest descent".

(iii) Various parts of the program think X^2 = number of data points is a good fit and think about stopping. For maximum likelihood options, these sections are skipped.

(iv) Call GCFNØ resets parameters to preset values. Call GCFIN first sets parameters to preset values and then reads namelist. If OVRRID is set = .TRUE. in this namelist, further GCFNØ calls are ignored, i.e., parameters left at values after namelist. Note CALL GCFGØ does not destroy parameter values.

E: Combination of X^2 and m. l. method in fitting experimental data

The advantage of the m.l. method is that it is the best, i.e., it should give the best possible values for the parameters with a given set of data.

The advantages of the X^2 method are a) quicker on the computer than the m.l. method and b) it is easy to judge the good of fit and areas where theory and experiment disagree from examination of the residuals f_i . b) is summarized in the usual result:

$$\langle X^2 \rangle = NT\emptyset T - NPAR \qquad (E1)$$

standard deviation of $X^2 = \sqrt{2}\,\langle X^2 \rangle$

(E1) assumes that X^2 is calculated using the parameters found by minimizing X^2 .

However, it is easy to show that if we call $\bar{X}^2 = X^2$ calculated with parameters determined by the m.l. method then:

$$NT\emptyset T \geq \begin{cases} \langle \bar{X}^2 \rangle \\ \text{Standard Deviation } \bar{X}^2/\sqrt{2} \end{cases} \geq NT\emptyset T - NPAR \qquad (E2)$$

whence as - in cases of interest - $NT\emptyset T$ and $NT\emptyset T - NPAR$ are pretty much the same, it follows that $\langle \bar{X}^2 \rangle$ is as good a judge of the m.l. fit as $\langle X^2 \rangle$ is of the X^2 fit.

So the following procedure can be recommended:

Find X^2 and use X^2 minimization to deduce first guess at parameters.

Use m.l. method to improve X^2 parameters.

Find X^2 to judge goodness of m.l. fit.

This appears to combine the advantages of both methods.