

MULTI_FRILLS: an Interactive Least-Squares Fitting Package

R.Osborn and T.G.Perring

March 1995

Rutherford Appleton Laboratory Chilton DIDCOT Oxfordshire OX11 0QX

Contents

| | |
|--|----|
| 1. Introduction..... | 2 |
| 2. Program Commands..... | 4 |
| 2.1 Manipulating Parameter Values and Data..... | 5 |
| 2.2 Displaying and Plotting parameters and data..... | 7 |
| 2.3 Other Commands..... | 10 |
| 3. Writing MULTI_FRILLS Applications..... | 13 |
| 3.1 Calling Programs..... | 13 |
| 3.2 Fitting Routines..... | 14 |
| 3.3 Linking Procedure..... | 15 |
| 3.4 Number of Parameters and Data Sets..... | 15 |
| 3.5 Example Program..... | 16 |
| 4. GENIE Transform Implementations..... | 19 |
| 4.1 Running GENIE Transform Applications..... | 19 |
| 4.2 Writing GENIE Transform Applications..... | 20 |
| 4.3 Linking Procedure..... | 21 |
| 4.4 Example Program..... | 21 |

1. Introduction

MULTI_FRILLS is a subroutine designed to be used in programs that perform interactive least squares fitting of a parametrised function to one or more sets of data points. It provides a flexible and easy-to-use interface to a least-squares fitting routine with the minimum of programming on the user's part. The simplest use of MULTI_FRILLS is to fit a function to a single set of data points, e.g. counts as a function of time in a single detector. More sophisticated programs fit several data sets simultaneously, with some parameters global to all data sets and others specific to a particular data set.

The user is required to provide:

- the calling program which initialises data points, parameter names and graph labels.
- a subroutine which calculates the parametrised function at each of the data points.

When subroutine MULTI_FRILLS has been called, the program operates in command mode. Following the prompt '#', the user types commands to

- list, remove or modify the data points
- set up, fix or constrain the initial parameters
- plot the data points and/or the calculated function for any of the data sets
- adjust least squares fitting constants
- perform the least squares fit

On completion, the fitted parameters are listed along with their estimated errors and a value for χ^2 . The program returns to command mode so that the user can plot the fit and choose new initial conditions (e.g. by clearing fixed parameters) before trying again. The results of fits are recorded in an ASCII file and lists of parameters may be stored in files for subsequent printing or input to another MULTI_FRILLS session. The plots may also be stored on disk in Postscript or LN03 files for output to a laser printer, or the calculated function written to ASCII files in tab delimited form suitable for input to proprietary programs such as Excel or SigmaPlot.

To optimize the function parameters, MULTI_FRILLS uses its own least squares fitting routine which employs a standard quasi-Newton algorithm. For a discussion of the general principles, see section 14.4 of "Numerical Recipes: The Art of Scientific Computing" (Cambridge University Press 1986). Using the calculated values passed back from the user-supplied subroutine, a χ^2 function is defined in the following way:

$$\chi^2 = \frac{\sum_{i=1}^{i=n_v} \left[\frac{1}{\sigma_i} (v_i - c_i) \right]^2}{(n_v - n_p)}$$

where v_i and σ_i are the i^{th} data value and error, c_i is the calculated value at that point, and n_v and n_p are the number of data values and fitting parameters respectively. The best fit parameters are those which minimize χ^2 . With this definition, the minimum of χ^2 should be close to $1.0 \pm \sqrt{2/(n_v - n_p)}$. Much larger values of χ^2 mean either that the model function is not an accurate representation of the data or that the data errors are underestimated, whereas smaller values mean that the data errors are too large. The fitting method will find the local χ^2 minimum; there may be a global minimum elsewhere in parameter space. However, since the user can plot the calculated function and compare it to the data for each data set before attempting a fit, it should be possible to define reasonable starting parameters for most models. The criteria for χ^2 convergence may be adjusted along with other fitting constants such as the maximum number of iterations. See the description of the ALTER FIT command in the next section for more details. Estimates of the parameter errors are provided by the diagonal elements of the covariance matrix, which assumes that they are normally distributed.

The next section describes all the MULTI_FRILLS commands in detail; if you are just using a MULTI_FRILLS application and not writing your own, this is the only section you need to read. For programmers, a description of how to write MULTI_FRILLS applications is given in Section 3 together with a simple example. A MULTI_FRILLS application may be written to work as a GENIE transform command; a description of how to do this is given in Section 4.

2. Program Commands

An important feature of this fitting package is the flexibility that the user has in the way a fit is performed. Commands may be given in any order, with the minimum of constraints. For instance, parameters may be fixed before one fit and cleared before the next. At any stage, the function, defined by the current set of parameters, may be plotted on the terminal screen for any of the several data sets. This section describes all the currently available MULTI_FRILLS commands for setting up the parameters, plotting the data and functions, and performing the least squares fits.

The following conventions apply:

- MULTI_FRILLS is insensitive to the case of the command line entered at the '#' prompt
- Every command or keyword may be abbreviated by its first letter.
- Numerical values should be separated by spaces or commas. A sequence of consecutive integers between n and m may be written as $n-m..$. If the user is prompted for a value, typing [RETURN] leaves it unchanged.
- Optional parameters are surrounded by square brackets. Values to be chosen by the user are surrounded by angle brackets.

MULTI_FRILLS works both on DECWindows screens and on terminals with both text and Tektronix 4014 graphics screens (e.g. Pericom terminals). On the latter, the terminal is automatically switched to the graphics screen after a plot command. However, the user must type [RETURN] after the plot is completed in order to return to text mode, as in GENIE.

At any stage in a MULTI_FRILLS session help can be obtained if the HELP command is given:

Help

The format of a specified command is listed at the terminal. If no command is specified, all the available commands are listed.

Format: H [<command>]

2.1 Manipulating Parameter Values and Data

Use

Selects the data set to be used for future commands. All commands that manipulate or display parameter values apply only to those that are global or specific to the data set identified by the most recent USE command. Similarly, commands that manipulate, display or plot data will do so only for that data set. For example, the DISPLAY PARAMETERS command lists only global parameters and those associated with the data set chosen with the USE command. Similarly, the REMOVE command, which removes points in a specified x-axis range, applies only to that data set. The GO command, which invokes the minimisation algorithm acts on all data sets at once. The default on entry is to use the first data set.

Format: U <n>

Set

The user is prompted for values of the specified parameters, and, optionally, their upper and lower limits. If no parameter number is specified, all the parameters associated with the selected data set are listed and the user is prompted to give the new values of each one in turn. If the user types a carriage return, the parameter is left unchanged.

Format: S [$\langle n1 \rangle$, [$\langle n2-n3 \rangle$, [$n4 \dots$]]]

Fix

The specified parameters are fixed in subsequent fits. They may be freed by the CLEAR PARAMETER command.

Format: F <n1>, [$\langle n2-n3 \rangle$, [$n4 \dots$]]

Bind

The value of each specified parameter is bound to one other parameter. The user is prompted for the number of the other parameter and the ratio of the two parameters i.e. if $p(i)$ is the specified parameter, and $p(j)$ is the parameter to which it is bound, the required ratio is $p(i) / p(j)$. If no ratio is given, it is set to the ratio of the current values of the parameters. Parameter binding is cleared by the CLEAR PARAMETER command.

Format: B <n1>, [<n2-n3>, [n4...]]

Limit

The specified parameters are constrained in subsequent fits by upper and lower values, for which the user is prompted. If the limits are to be removed, set both values to 0. In the least-squares fit, constrained parameters are transformed into a sine function so that parameter derivatives do not diverge even at the limiting values. Nevertheless, limiting parameters should only be used when absolutely necessary.

Format: L <n1>, [<n2-n3>, [n4...]]

Input

The parameter values and constraints are read in from a disk file of the same format as produced by the OUTPUT command below.

Format: I <disk file name>

Output Parameters

The current parameters are output to a disk file. If none is specified, the default file name is SYSSCRATCH:PAR.OUT. The parameters may be read subsequently from the disk file with the INPUT command.

Format: O P [<disk file name>]

Output Fit

The latest fitted parameters are output to a disk file along with their χ^2 value. If no disk file is specified, the default file name is SYSSCRATCH:FIT.OUT. The parameters may be read subsequently from the disk file with the INPUT command.

Format: O F [<disk file name>]

Remove

Data points between the specified x-values are removed from the fit. They may be restored by the CLEAR DATA command.

Format: R <xmin> <xmax>

Modify

The data points between the specified x-values are modified for subsequent fits. The user is prompted for new y-values. They may be restored to their initial values by the CLEAR DATA command.

Format: M <xmin> <xmax>

Clear Parameters

The specified parameters are freed to vary in subsequent fits. If no parameters are specified, all are freed.

Format: C P [<n1>,<n2-n3>,<n4...>]

Clear Data

The data between the specified limits are reset to their original values (if modified) and included in subsequent fits (if removed). If no limits are specified, all points are reset.

Format: C D [<xmin> <xmax>]

2.2 Displaying and Plotting Parameters and Data

Display Data

The data points within the specified range are listed at the terminal. If no range is specified, all points are listed.

Format: D D [<xmin> <xmax>]

Display Parameters

The specified parameters are listed at the terminal. If none are specified, all parameters are listed.

Format: D P [<n1>,<n2-n3>,<n4...>]

Display Calculation

The data points and the calculated values within the specified range using the current parameters are listed at the terminal. If no range is specified, all points are listed.

Format: D C [

Display Fit

The fitted parameters are listed at the terminal.

Format: D F

Display Matrix

The correlation matrix of the fitted parameters is listed at the terminal.

Format: D M

Plot Data

The data points are plotted at the terminal with the option of setting the axis limits. If DATA is preceded by OVER (i.e. PLOT OVER DATA), the data are overlaid on the previous plot.

Format: P [O] D [<xmin> <xmax> [<ymin> <ymax>]]

Plot Calculation

The data points and the calculated values using the current parameters are plotted at the terminal with the option of setting the axis limits. If CALCULATION is preceded by OVER (i.e. PLOT OVER CALCULATION), the calculated lines are overlaid on the previous plot and the data points are not replotted.

Format: P [O] C [<xmin> <xmax> [<ymin> <ymax>]]

Plot Fit

The data points and their fitted values are plotted at the terminal with the option of setting the axis limits. If FIT is preceded by OVER (i.e. PLOT OVER FIT), the fitted lines are overlaid on the previous plot and the data points are not plotted.

Format: P [O] F [<xmin> <xmax> [<ymin> <ymax>]]

Plot Residuals

The difference between the data points and their fitted values are plotted at the terminal with the option of setting the axis limits. If RESIDUALS is preceded by OVER (i.e. PLOT OVER RESIDUALS), the plotted points are overlaid on the previous plot.

Format: P [O] R [<xmin> <xmax> [<ymin> <ymax>]]

Plot <no. list>

The individual components of a fit are plotted separately with dashed lines. The numbers refer to the order in which the components are stored in the user-supplied function subroutine (see section 3, and common block /FR_CMPS/). If the list of component numbers is preceded by OVER (i.e. PLOT OVER <no. list>), the lines are overlaid on the previous plot.

Format: P [O] [<n1> , [<n2-n3> , [<n4...>]]]

Output Calculation

The calculated values of the fitting function and its components using the current parameters are written to an ASCII disk file in tab-delimited form. If no disk file is specified, the default file name is SYSSCRATCH:CAL.OUT. The format of the file is suitable to be read into Excel or SigmaPlot.

Format: O C [<disk file name>]

Keep

The plots on the screen are stored in a Postscript A4 (default), Postscript US letter or LN03 file, depending on the hard copy type selected with the ALTER HARDCOPY command. If no file name is specified, the default is SYSSCRATCH:FRILLS.PS for Postscript files and SYSSCRATCH:FRILLS.TEK for LN03 files. If a fit or calculation has been plotted, the parameters are included in the plot even though they do not appear on the terminal screen.

Format: K [<disk file>]

2.3 Other Commands

Alter Fits

Adjust least-squares fitting constants. The user is prompted for the new value of each one in turn. Type [RETURN] to leave them unchanged. The parameters that can be changed are: the derivative step length; the accuracy of χ^2 the accuracy(if χ^2 changes by less than this amount between two iterations then χ^2 is deemed to have converged); the maximum number of iterations; the maximum number of times the parameter step is halved if χ^2 diverges.

Format: A F

Alter Output

Controls the printing of output to the screen and to disk file during a fit. A list of options is provided if the output code is omitted. Type [RETURN] to leave the output code unchanged. The default is to print χ^2 to the screen for each iteration but no output to disk file.

Format: A O [<ncode>]

Alter Markers

Changes the marker type, size and colour used when plotting data. If no parameter values are entered on the line the user is prompted for the new value of each one in turn. Type [RETURN] to leave the values unchanged.

Format: A M [<ntype>, [<size>], <ncolour>]]]

Alter Lines

Changes the line type, thickness and colour used when plotting calculations or fits. If no parameter values are entered on the command line the user is prompted for the new value of each one in turn. Type [RETURN] to leave the values unchanged.

Format: A L [<ntype>, [<thickness>], <ncolour>]]

Alter Smoothing

Adjust the smoothness of calculated and fitted functions when plotted to the screen. The user is prompted for the new value of two parameters in turn: N_PLT, the number of points at which the function is calculated, and SMOOTH, which indicates whether or not a spline fitting algorithm is implemented. Type [RETURN] to leave them unchanged. The function is calculated at N_PLT (default 1000) x-coordinates equally spaced over the range of the plot. N_PLT should be reduced if the function is computationally intensive to evaluate. If the user sets N_PLT=0 the function is calculated only at those x-coordinates at which there is data and which have not been REMOVED. By default SMOOTH=0 and the calculated values are joined by straight lines. If the user sets SMOOTH=1 then from the N_PLT calculated data points a smooth curve is generated at 1000 equally spaced x-coordinates which passes through the calculated data points. This produces a smooth curve even if the function is evaluated at only a few points.

Format: A S

Alter Workstation

Alter the terminal emulation and workstation type. The user is prompted for new values. Type [RETURN] to leave the values unchanged. The default device emulation assumes either an Xwindows terminal or a Pericom with Tektronix 4014 emulation. Alternatively, Tektronix 4014 emulation suitable for an IBM compatible PC running Kermit can be chosen. The default workstation type produces graphical output using Xwindows/Motif or Tektronix 4014 drivers; the program translates DECW\$DISPLAY to determine which to use. The graphics driver can be changed by entering the corresponding GKS workstation type.

Format: A W

Alter Hardcopy

Alter the type of hardcopy output produced by the KEEP command. If no parameter value is given the user is prompted for a new value. Type [RETURN] to leave the hardcopy type unchanged. Three types are available: Postscript A4 (default type), Postscript US letter, and LN03 file.

Format: A H <ntype>

Title

The graph title is changed. The user is prompted for the new title if it is not entered on the command line. Type [RETURN] to leave the title unchanged.

Format: T <new-title>

Jump

A DCL sub-process is spawned. If a DCL command is given, it is executed and control returns to the MULTI_FRILLS session. Otherwise, a DCL process is started from which the user must log out in order to return to MULTI_FRILLS.

Format: J [<DCL command>]

Go

The non-fixed parameters are fitted to the data points. On completion, the fitted parameters are listed at the terminal along with a χ^2 value. The fitted points may optionally be plotted using the command P F, or listed at the terminal using the command D F. The GO command minimises χ^2 calculated over all the data sets.

Format: G

Exit

Control is passed back to the main program.

Format: E

3. Writing MULTI_FRILLS Applications

In this section all the essential initialisation required by subroutine MULTI_FRILLS is described, together with the input and output variables of the calculation subroutine. An example application is presented in Section 3.5 which illustrates the following description. The source code for the example program given in that section and the GENIE Transform examples in Section 4.4, together with that for MULTI_FRILLS and various ancillary programs is stored in the directory HET\$DISK1:[HETMGR.LIBRARY.MULTI_FRILLS.REF]. It is recommended that the user define the logical name FRILLS_SOURCES to point to this directory by typing at the terminal or adding to their LOGIN.COM the following line:

```
$ define frills_sources het$disk1:[hetmgr.library.multi_frills.ref]
```

3.1 Calling Programs

The program that calls MULTI_FRILLS should contain the line

```
INCLUDE 'FRILLS.INC'
```

which gives access to all the common blocks used by MULTI_FRILLS. The user needs to initialise the following variables:

| | |
|------------------|--|
| NPTOT | the total number of parameters |
| NAM(<i>ip</i>) | the names of the parameters (<i>ip</i> =1, NPTOT) |
| NS | the number of data sets |

then for each of the data sets (*is*=1, NS):

| | |
|--------------------------------------|--|
| ND(<i>is</i>) | the number of points in each data set |
| X(<i>i, is</i>) | the x-coordinates (<i>i</i> =1, ND(<i>is</i>)) |
| Y(<i>i, is</i>) | the y-coordinates (<i>i</i> =1, ND(<i>is</i>)) |
| YSIG(<i>i, is</i>) | the error bars (<i>i</i> =1, ND(<i>is</i>)) |
| XLAB(<i>is</i>), YLAB(<i>is</i>) | names used for the x and y axes when DISPLAYing data |
| XCAP(<i>is</i>), YCAP(<i>is</i>) | labels used for the x and y axes when PLOTting data |
| TITLE(<i>is</i>) | title |

Optionally, the user can define each parameter as either global to all data sets or specific to one data set, by filling the array $INS(ip) = 0$ (global) or $INS(ip) = is$ (specific to data set *is*). The default is for parameters to be global to all data sets. The array INS is used for display purposes only and is not used in the minimisation algorithm.

The include file FRILLS.INC gives access to many more variables which allow the ambitious programmer more control over the initial state of MULTI_FRILLS. Details are given in the next section. Whilst the use of include files simplifies the writing of MULTI_FRILLS applications it opens up the danger that the user will inadvertently use variable names that are also contained in the MULTI_FRILLS common blocks. The simplest way to reduce this possibility is for the user to avoid the IMPLICIT statement and explicitly declare all variables in the calling program. The DEC FORTRAN compiler will print a warning message if any variable has been declared twice. The compiler option FORTRAN/WARNINGS=DECLARATIONS will print a warning if there are any undeclared variables.

Diagnostic output from the fit is written to unit 41. If this is not explicitly opened by the calling program, it will sent to SYS\$SCRATCH:FRILLS.TXT

3.2 Fitting Subroutines

The subroutine defining the fitting function, in following example GAMFIT, must be explicitly declared as EXTERNAL in the calling program. The subroutine is expected to calculate the fitting function for only one data set at each call. It has no arguments passed to it directly, but it must contain the statement

```
INCLUDE 'FUNCTION.INC'
```

which gives access to the following variables:

| | |
|----------------|--|
| P(<i>ip</i>) | the current values of the parameters |
| IS | the index of the data set for which calculated values are required |
| NV | the number of points in the data set |
| X(<i>iv</i>) | the x-coordinates of the points (<i>iv</i> =1, NV) |

and the variables to contain the output from the fitting function:

| | |
|-----------------------------|--|
| YCAL(<i>iv</i>) | the calculated function at each of the x values (<i>iv</i> =1, NV) |
| NC | the number of separate function components that make up YCAL |
| YC(<i>iv</i> , <i>ic</i>) | the calculated function components (<i>iv</i> =1, NV; <i>ic</i> =1, NC) |

Filling NC and YC is optional. They permit the individual components of a multi-component function to be plotted separately with the PLOT command.

3.3 Linking Procedure

MULTI_FRILLS applications can be compiled and linked on ALPHA VMS computers. To link a compiled program to MULTI_FRILLS, use the following command:

```
$ @FRILLS_SOURCES:FL <source_code> <libraries>
```

where <source_code> is a comma separated list of FORTRAN files, and <libraries> is a comma separated list of object libraries. The command file will link to the MULTIF_FRILLS and graphics libraries; only those libraries needed by the users own source need to be included. An example is

```
$ @FRILLS_SOURCES:FL myprog,mysub mylib/lib,nagf/opt
```

To allow for reasonable numbers of parameters, data sets and the length of data sets MULTI_FRILLS requires a large amount of computer memory. The program may fail to compile as it might exceed the user's page file quota. Please contact ISIS Computer Support if this happens.

3.4 Number of Parameters and Size of Data Sets

The MULTI_FRILLS library currently has the following defaults:

| | |
|---------------------------------------|------|
| maximum number of data sets | 10 |
| maximum number of points per data set | 1000 |
| maximum number of parameters | 150 |

The number of parameters that are allowed to vary in a fit should be much less than 150 for acceptable running time.

The default limits are straightforward to change, but the MULTI_FRILLS library needs to be recompiled. If a customised version of the MULTI_FRILLS library is required, please contact T.G.Perring.

3.5 Example Program

The following program is an example of a MULTI_FRILLS application. The program reads in two data sets containing the measured linewidth of crystal field peaks as a function of temperature, one set for each of two crystal field transitions with different energies. The two spectra are fitted by the same functional form, which is different above

and below the superconducting transition temperature. The facility to pass separate components of the fitting function back to MULTI_FRILLS is used to return the functional form for $T > T_c$ extrapolated to $T < T_c$. This can then be compared with the correct form for $T < T_c$ using the commands PLOT CALCULATION and PLOT OVER 1. The USE command determines the data set that is being viewed.

```

program GAMFIT

    include 'FRILLS.INC'
    external GAMCAL
    integer i, j

c
c NAM(i) = name of parameter, P(i)
c INS(i) = spectrum number associated with that parameter
c         (0 for all spectra)
c
    data nam(1) //'Bkgd Constant'//,      ins(1) /1//,
    >    nam(2) //'Bkgd Constant'//,      ins(2) /2//,
    >    nam(3) //'Coupling constant'//,  ins(3) /0//,
    >    nam(4) //'Energy Gap'//,         ins(4) /0//,
    >    nam(5) //'Tc'//,                 ins(5) /0//

c
c
c TITLE  = graph title
c XCAP   = x-axis label
c YCAP   = y-axis label
c XLAB   = x identifier
c YLAB   = y identifier
c
c
    data xcap(1) //'Temperature (K)'//,  xcap(2) //'Temperature (K)'//,
    >    xlab(1)  //'T'//,                xlab(2)  //'T'//,
    >    ycap(1)  //'Gamma (11.8 meV)'//, ycap(2)  //'Gamma (14.2 meV)'//,
    >    ylab(1)  //'Gam'//,              ylab(2)  //'Gam'//,
    >    title(1) //'Crystal Field Linewidths E = 11.8 meV'//,
    >    title(2) //'Crystal Field Linewidths E = 14.2 meV'//

c

```

```

C
C NPTOT = no. of parameters          ND(i) = no. of data points
C
C      data nptot /5/
C
C
C Diagnostic output is written to a file on channel 41
C
C      open (unit = 41, file = 'SYS$SCRATCH:GAMFIT.TXT', status = 'NEW')
C
C
C Two sets of data to be fitted simultaneously
C
C      ns = 2
C
C
C The data points are read in from data files
C
C      do i=1,ns
C        if (i .eq. 1) open (unit = 10,
>          file = 'TM_GAM_11.DAT', status = 'OLD')
C        if (i .eq. 2) open (unit = 10,
>          file = 'TM_GAM_14.DAT', status = 'OLD')
C        read (10, fmt=*) nd(i)
C        do j = 1,nd(i)
C          read (10, fmt=*, end = 100) x(j,i), y(j,i), ysig(j,i)
C        end do
C        close (unit = 10)
C      end do
C
C      call multi_frills (GAMCAL)
C
C      close (unit = 41)
100 stop
end
C-----
C
C Subroutine for calculating the function to be fitted
C
C
C      subroutine gamcal
C
C        include 'FUNCTION.INC'
C
C        integer i
C        double precision Gam0, Eij, Mij, pJ, Delta, Tc,
>          pi, kT, Gam, Del_T, BCS
C        parameter (pi=3.14159265)
C
C Two components of the peak to be calculated
C
C      nc = 2
C
C Definition of parameters
C
C      if (is .eq. 1) then
C        Gam0 = p(1)
C        Eij = 11.8
C        Mij = 1.0
C      else
C        Gam0 = p(2)
C        Eij = 14.2
C        Mij = 0.88
C      end if
C      pJ = p(3)

```

```

Delta = p(4)
Tc = p(5)

do i = 1,nv
  kT = x(i)/11.6045
  if (kT/Eij .gt. 0.02) then
    Gam = 4.0*pi*Eij*(Mij*pJ)**2*
>      (1.0+2.0*exp(-Eij/kT))/(1.0-exp(-Eij/kT))
  else
    Gam = 4.0*pi*Eij*(Mij*pJ)**2
  end if
  yc(i,1) = Gam0 + Gam
  if (x(i) .lt. Tc .and. Tc .gt. 0.0) then
    Del_T = Delta * BCS(x(i)/Tc)
    if (kT / Del_T .gt. 0.05) then
      Gam = 2.0 * Gam / (1.0 + exp(Del_T/kT))
    else
      Gam = 0.0
    end if
  end if
  ycal(i) = Gam0 + Gam
  yc(i,2) = ycal(i)
end do

return
end

double precision function BCS (T)

double precision T

BCS = 0.99421561371 + 0.75673343851*T - 16.645329514*T**2 +
> 142.30924375*T**3 - 601.66550422*T**4 + 1383.5123877*T**5 -
> 1769.7613796*T**6 + 1181.302108*T**7 - 320.77574921*T**8

return
end

```

4. GENIE Transform Implementations

In this section an example of a program that fits several GENIE workspaces at once is given. The source code is in the file `FRILLS_GENIE:MULTI_RESFIT.FOR`. The program models the peak shape in all workspaces as the convolution of a Gaussian with an exponential decay, with the width the same for all spectra. The intensities and sloping background are different for each workspace. Up to four workspaces can be fitted in this example.

4.1 Running GENIE Transform Applications

To run a GENIE Transform application, the user should issue a GENIE command file:

```
>> @frills_sources:mfg
```

which will prompt the user for

- the name of the GENIE Transform program
- the number of workspaces
- the first input workspace number (it is assumed the workspaces are consecutive)
- the first output workspace number

Generally, `MULTI_FRILLS` programs for GENIE can act on any number of workspaces up to the maximum number for which the program was designed. With the example program `MULTI_RESFIT`, for instance, if after entry the user `FIXes` the parameters that act only on workspaces numbers 2,3 and 4, the program can be used to fit just one workspace.

It is possible to write GENIE applications that explicitly act on one workspace only. These should be invoked directly by the GENIE `TRANSFORM` command:

```
>> TRANSFORM Wm <program-name> Wn
```

4.2 Writing GENIE Transform Applications

The only difference between standalone and GENIE Transform applications is the way the data and graph captions are read into MULTI_FRILLS. To help the user write GENIE Transform applications there are several general purpose subroutines.

The example program illustrates how to call MULTI_FRILLS within a GENIE Transform that fits several workspaces at once. MULTI_FRILLS_GENIE effectively replaces the call to MULTI_FRILLS itself. It reads data (assumed to be in histogram mode), including titles and graph captions, from a file written when the command file FRILLS_SOURCES:MFG is issued in GENIE. The data is read into the MULTI_FRILLS common blocks before MULTI_FRILLS is called. After completion of the MULTI_FRILLS session it writes the function, calculated with the latest set of parameters, to disk, from where it is read into GENIE workspaces by FRILLS_SOURCES:MFG. The user need only provide the names and number of parameters, and the maximum number of data sets, i.e.

| | |
|------------------|--|
| NPTOT | the total number of parameters |
| NAM(<i>ip</i>) | the names of the parameters (<i>ip</i> =1, NPTOT) |
| NS | the maximum number of data sets |

and optionally fill the array array elements $INS(ip) = 0$ (global) or $INS(ip) = is$ (specific to data set *is*). The subroutine that calculates the fitting function should be written exactly as for standalone MULTI_FRILLS applications.

For those users who wish to manipulate or inspect the data before MULTI_FRILLS is called, two routines, MULTI_FRILLS_GENIE_IN and MULTI_FRILLS_GENIE_OUT, are provided. In this case the calling sequence would be

```
call multi_frills_genie_in
      .
      .
call multi_frills (fun)
      .
      .
call multi_frills_genie_out (fun)
```

where fun is the name of the function subroutine. Otherwise the GENIE Transform is written exactly as with MULTI_FRILLS_GENIE.

GENIE Transforms that act on one workspace only can be written with SINGLE_FRILLS_GENIE replacing the call to MULTI_FRILLS. Alternatively, if the user wishes to manipulate the data before entering MULTI_FRILLS the following calling sequence should be used:

```

call single_frills_genie_in
.
.
call multi_frills (fun)
.
.
call single_frills_genie_out (fun)

```

4.3 Linking Procedure

The linking procedure for all GENIE Transform applications is the same as that for standalone MULTI_FRILLS programs. See Section 3.4 for details.

4.4 Example Program

```

program MULTI_RESFIT

c Resolution function fitting of multiple sets chopper spectrometer data
c Up to 4 spectra can be simultaneously fitted to a Gaussian convoluted
c with a decaying exponential, with the peak area and the (linear)
c background fitted independently for each spectrum.
c
c Written by   R.Osborn
c             Materials Science Division
c             Argonne National Laboratory
c
c             T.G.Perring
c             ISIS Facility
c             Daresbury and Rutherford Appleton Laboratory

include 'FRILLS.INC'
external RESFUN

data nam(1)  /'Peak 1  Intensity'//,  ins(1)  /1/,
>  nam(2)  /'          bkgd const'//,  ins(2)  /1/,
>  nam(3)  /'          bkgd slope'//,  ins(3)  /1/,
>  nam(4)  /'Peak 2  Intensity'//,  ins(4)  /2/,
>  nam(5)  /'          bkgd const'//,  ins(5)  /2/,
>  nam(6)  /'          bkgd slope'//,  ins(6)  /2/,
>  nam(7)  /'Peak 3  Intensity'//,  ins(7)  /3/,
>  nam(8)  /'          bkgd const'//,  ins(8)  /3/,
>  nam(9)  /'          bkgd slope'//,  ins(9)  /3/,
>  nam(10) /'Peak 4  Intensity'//,  ins(10) /4/,
>  nam(11) /'          bkgd const'//,  ins(11) /4/,
>  nam(12) /'          bkgd slope'//,  ins(12) /4/,

>  nam(13) /'Gamma'//,  ins(13)/0/,
>  nam(14) /'Sigma'//,  ins(14)/0/,
>  nam(15) /'Centre'//,  ins(15)/0/

data nptot /15/

open (unit = 41, file = 'SYS$SCRATCH:RESFIT.LPT', status = 'NEW')

```



```

call multi_frills_genie (RESFUN)

close (unit=41)

stop
end

c-----

subroutine resfun

integer i
double precision area, bc, bs, gamma, sigma, centre, exp_decay
include 'FUNCTION.INC'
external EXP_DECAY

c Definition of parameters

area = p(3*is-2)           !Intensity of peak
bc   = p(3*is-1)           !Background constant
bs   = p(3*is)              !Background slope
gamma = p(13)               !Decay constant
sigma = p(14)               !Gaussian sigma
centre= p(15)               !Peak centre (1st moment)

c Component 1 contains peak, component 2 contains background

do i = 1,nv
  yc(i,1) = area * EXP_DECAY (x(i)-centre, gamma, sigma)
  yc(i,2) = bc + bs*x(i)
  ycal(i) = yc(i,1) + yc(i,2)
end do

return
end

```