DRAL
Daresbury Laboratory
Rutherford Appleton Laboratory

RAL Report
RAL-94-080

The Muon Data Analysis Program "RUMDA"

S H Kilcoyne

July 1994

Rutherford Appleton Laboratory Chilton DIDCOT Oxfordshire OX11 0QX

The Muon Data Analysis Program

"RUMDA"

S H Kilcoyne

July 1994

Table of Contents

1. INTRODUCTION

1.1. Computer programs

There are currently two data analysis programs available for muon users at ISIS. Both programs can be used for analysing MuSR and EMU data and can be run on [MUSR01], [EMU01] or set-up to run on a user's account.

1.1.1. **RUMDA** - "Reading University Muon Data Analysis" originally from Reading University and now controlled at ISIS by Sue Kilcoyne.

At present (mid 1994) this suite of programs is run using VAX/VMS and the ISIS plotting package "GENIE". It is possible to fit data to any function with a maximum of 10 variables.

1.1.2. **UDA** - "µ Data Analysis" written by Tore Sundqvist and Ola Hartmann of Uppsala University, Sweden.

This is a dashboard driven program which allows the user to plot and fit data files on the screen or as hard copies. It is possible to fit data to a combination of gaussian and/or lorentzian line shapes.
A manual describing this program can be found in the back of the MuSR User Guide.

1.2. Setting up a sub-directory for data analysis

Users are advised to work in an area dedicated to data analysis of their muon experiment. The easiest way of doing this is to create a sub-directory on the SCRATCH disk of either MUSR01 or EMU01 as shown below:

- Log in as either [MUSR01] or [EMU01]
- Set the default to a personal sub-directory on either SCRATCH$DISK:[MUSR01] or SCRATCH$DISK:[EMU01] by typing your establishment name. 50,000 blocks will then be available for data analysis. See your local contact if you have problems at this stage.
- Run SETUP.COM from this area by typing "SETUP".
- The command file SETUP.COM will copy all the files necessary to run RUMDA and UDA onto this area.
- As files on the scratch disk are deleted after 7 days therefore, any files you wish to save should be copied to your own account on the USER$DISK, or to a tape, as soon as possible.

Example

The following section shows the output on the screen when a RAL user logs onto MUSR01 on ISISE computer. Commands entered by the user are shown in **bold** type

```
         >> PUNCH VAX VMS V5.5-2  on Cluster Node ISISE <<

Username: MUSR01
Password: DIZITAL7              (the current passwords will be displayed in the cabins)


Welcome to VAX/VMS version V5.5-2 on node ISISE

   Last interactive login on Friday, 10-JUN-1994 16:17
   Last non-interactive login on Friday, 10-JUN-1994 16:16

   User [MUSR01] has 15283 blocks used, 4717 available,
   of 20000 authorized and permitted overdraft of 1000 blocks on USER$DISK


   ENTER YOUR ESTABLISHMENT NAME TO 'SET DEF' TO YOUR AREA.

    ON SCRATCH DISK TYPE SETUP TO ACCESS ANALYSIS PROGS

MUSR01>RAL


*********************************************************
       YOU ARE NOW WORKING ON THE SCRATCH DISK
     FILES IN THIS AREA WILL BE DELETED AFTER 7 DAYS
*********************************************************

RAL>
```

2. RUMDA

2.1. Introduction

Two equivalent suites of programs run independently from accounts [MUSR01] for analysing MuSR data and from [EMU01] to analyse EMU data. The intermediate or final spectra together with the results of a fit can be plotted using the graphics package GENIE (see either the PUNCH User Guide or report RAL-86-102 for further information about GENIE).

All file names generated for or by RUMDA have the general formula

<character><run number>.<extension>

The <character> will be R for a MuSR data file or L for a log file

The <run number> is the number generated by MCS for data collection.

The <extension> can be any one of the following:

.LOG for a text file containing information about the experimental set-up,

.DAT for a grouped data file,

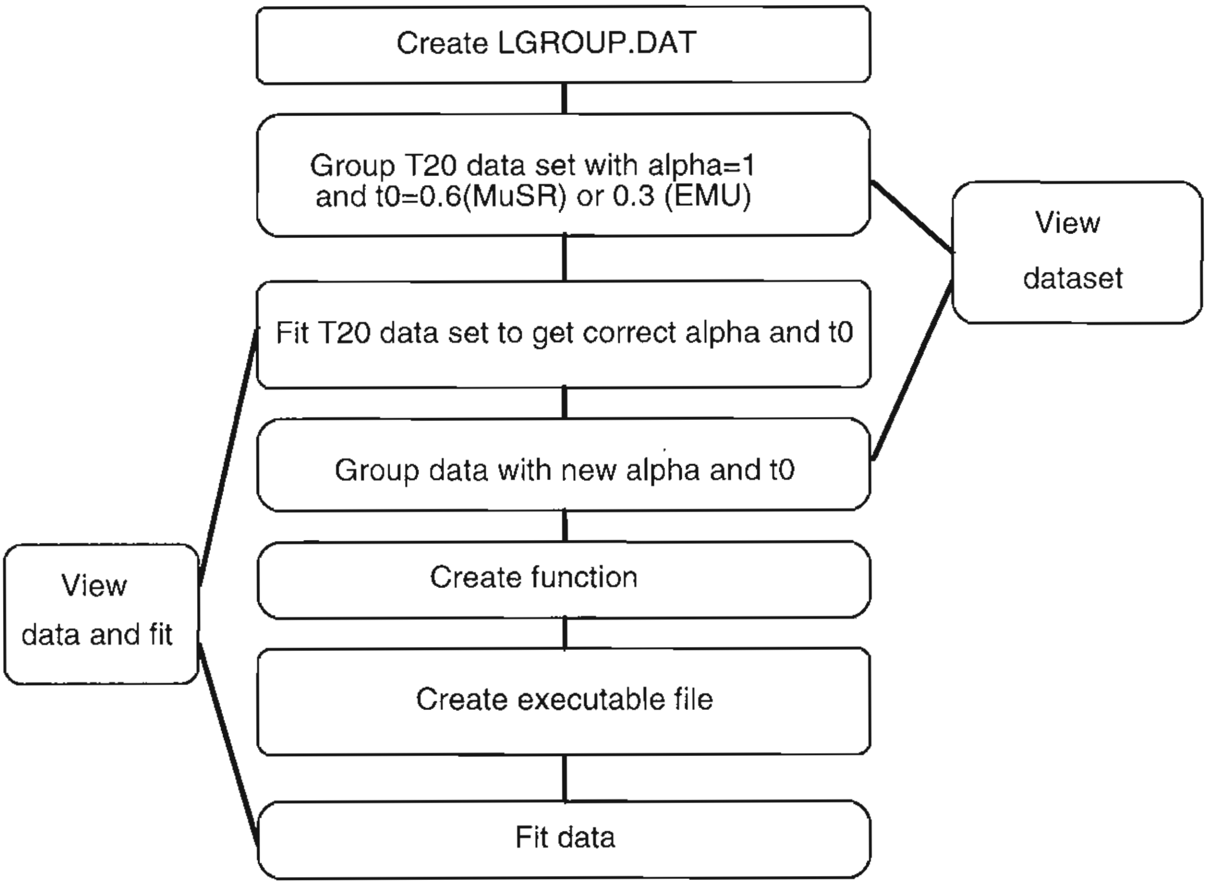
.OUT for the file containing the details of a fit,

.FIT for the fitted line shape

.PAR for the parameters determining the line shape.

2.2. Data analysis procedure : Flow chart of programs

The programs in RUMDA need to be run in the following order:


Create LGROUP.DAT
Group T20 data set with alpha=1 and t0=0.6(MuSR) or 0.3 (EMU)
Fit T20 data set to get correct alpha and t0
Group data with new alpha and t0
Create function
Create executable file
Fit data
View dataset
View data and fit

Each of these procedures is explained in detail in the following sections.

Step 1. Create LGROUP.DAT

A common method of analysing μSR data is to add the counts in several histograms together and fit the resultant. For example, in longitudinal geometry the sums of counts in the forward detectors are added together and those in the backward detectors are added together, resulting in two spectra (denoted F and B in step 2 below) from which the asymmetry can be calculated.

LGROUP.DAT is the file which determines this grouping of the initial 32 histograms. It has the format

```
 1 x (number of groups)
32 x (histogram number, group number)
```

e.g. to group the first 16 histograms into one forward detector and the second 16 into one backward detector LGROUP.DAT would read

```
2
1,1
2,1
3,1
.
.
16,1
17,2
18,2
19,2
.
.
32,2
```

Clearly, if a detector fails or is switched off during an experiment it should not be considered when analysing the data. Omitting the detector number from the list in LGROUP.DAT removes that histogram from the data analysis.

Three standard grouping files exist. If one of these is adequate for your needs copy it to LGROUP.DAT in the area you are working from.

LGROUP32_1.DAT	no grouping of histograms, used to check individual detectors
LGROUP2_16.DAT	2 groups with histograms 1-16 in group 1 and 17-32 in group 2
LGROUP2_8.DAT	2 groups with histograms 1-8 in group 1 and 9-16 in group 2 (needed for old (pre May 1993) DIZITAL data)

Step 2. Group T20 data

When working in longitudinal geometry each set of runs (after a sample change or change from CCR to cryostat for example) starts with a measurement in a small transverse field of approximately 20 gauss. By taking account of the relative efficiencies of the detectors in the forward, F, and backward, B, detector banks this measurement is used to generate a value for the calibration parameter, α; essential in the calculation of the ratio (F-αB)/(F+αB). These runs can be quite short (<5 Mevents) and are often referred to as "T20s". A T20 run can also be used to determine the time, t = 0 point in a spectrum.

For data collected in zero field or applied longitudinal field the grouping of histograms is achieved with the command LNG or EMULNG, depending on the instrument. The default is that dead time corrections (see section 4) will be made before the histograms are grouped. These corrections will not be carried out if a 1 is present at the end of the command line. For example; to group a single T20 run collected on MuSR and include dead time corrections the command is -

LNG Run number

Similarly, to group a single T20 run collected on EMU without dead time corrections the complete command would be -

EMULNG Run number Run number 1

(the second run number gives the option of grouping more than one file at a time)
The program then requests an initial guess at t0 from the user. This initial guess is modified by the fitting program in stage 3. As a guide values of 0.6 and 0.3 are suitable first guesses for MuSR and EMU data respectively. Finally, a value of alpha is requested; for the grouping of detectors in a silver spectrum alpha should be set to 1.

The program creates files TEMP.G1 and TEMP.G2, containing the sums of counts in group 1 and group 2, R(run number).LOG, containing the header block and R(run number).DAT, containing the ratio (F-B)/(F+B).

Computer programs

Command file	LNG.COM or EMULNG.COM
Program	MGROUP.EXE or EMUGROUP.EXE
Input files	Raw data file collected in small transverse field LGROUP.DAT
Output files	TEMP.G1, TEMP.G2 R(run number).DAT and L(run number).LOG

File formats

LGROUP.DAT	(see previous section)
TEMP.G#	N x (time, counts, σcounts)
L(run number).LOG	information in ASCII format
R(run number).DAT	N x (time, asymmetry, σasymmetry)

Step 3. Fit T20 data

In most cases the time dependence of the asymmetry for a T20 run will be a damped cosine function. The program FCL (Fit Cosine with Lorentzian damping) can be used to fit such a spectrum and calculate the time t0 and the calibration parameter α. This program is run by typing the command "T20FIT". In transverse fields of >50 gauss it may be difficult to fit the spectrum over the whole time range at the first attempt. If this occurs, fit the data between 0.5 and 3μs first and then fit over the whole range using the resulting .PAR file as the starting parameters.

On running the program the user is prompted for the following information:

Run number	The number determined by MCS during data collection
Input parameters	Parameters can be entered by the user (option (U)), from an old parameter file (option (F)) or from a parameter file generated by an earlier T20FIT to this dataset (option ( ))
Start and stop times	These times (in μs) determine the time-window over which the fit is performed
Maximum number of iterations	Maximum number of cycles, 0 gives a simulation
Field	Size of the transverse field in gauss
Δ(t-zero)	The uncertainty in the "first guess" value of t0 ($\neq 0$)

T20FIT then performs a non linear least squares fit to the data using the function

$$F(t) = a_o[\cos(\omega.(t+\delta t)) . \exp(-\lambda.(t+\delta t))] + B$$

where a_o is the initial asymmetry, ω is the precession frequency, $(t+\delta t)$ the time plus uncertainty on time and λ is the relaxation rate. B is the difference between the centre of gravity of the spectrum and the x axis. The program calculates the modification to t0 and the correct value of the calibration parameter α. These values should be recorded as they will be required in subsequent data analysis routines.

The record of the fit is saved in the file R(run number).OUT, and the fitted parameters in R(run number).PAR. The resulting line shape is saved in R(run number).FIT. The data and the fit can be plotted using GENIE, as described in section 3.

An example of this step is shown over the page.

Example

```
RAL>T20FIT

            **** PROGRAM TO FIT MUSR DATA ****

ENTER RUN NUMBER OF INPUT DATA FILE
2601
INPUT PARAMETERS BY USER (U), NAMED FILE (F) OR CURRENT .PAR FILE ( )
u
START AND STOP TIMES (MUSEC)
.5 10
ENTER MAXIMUM ITERATIONS (0 FOR SIMULATION)
100
TRANSVERSE FIELD  (G)?
20
ΔT-ZERO (mus)?
0.01
DAMPING RATE (MHZ)?
0.1


FILE NAME : R02601.OUT

FITTED PARAMETERS

  CHISQR    A 1     A 2     A 3     A 4     A 5
   3.478  19.458  -0.006   0.080   0.078   0.281
   1.620  19.524   0.009   0.113   0.069   0.121
   1.282  19.116   0.024   0.102   0.065   0.189
   1.224  19.210  -0.015   0.117   0.063   0.170
   1.188  19.092  -0.003   0.109   0.063   0.214
   1.166  19.132  -0.009   0.113   0.062   0.227
   1.153  19.119  -0.008   0.116   0.062   0.234
   1.144  19.096  -0.005   0.118   0.062   0.239
   1.137  19.074  -0.002   0.120   0.062   0.242
   1.132  19.053   0.001   0.122   0.063   0.246
   1.127  19.112  -0.006   0.128   0.063   0.236
   1.125  19.021   0.005   0.125   0.063   0.250
   1.123  19.008   0.006   0.126   0.063   0.252
   1.121  19.046   0.002   0.130   0.063   0.246
   1.119  18.980   0.010   0.128   0.063   0.256
   1.118  18.973   0.011   0.129   0.063   0.257
   1.117  18.986   0.010   0.131   0.063   0.255

NUMBER ITERATIONS = 17,  FINAL CHISQR =   1.1174655
```

```
FINAL VALUES OF PARAMETERS

             A(N)                ERROR
 A( 1)   18.98606300         0.1581956893
 A( 2)  0.9508864023E-02  0.1248449832E-01
 A( 3)  0.1312402785         0.2042313805E-02
 A( 4)  0.6345256418E-01  0.7911891444E-03
 A( 5)  0.2548668981         0.7690297440E-02


 FOR FITTED DATA SEE FILE : R02601.OUT
    FITTED DATA FOR GENIE : R02601.FIT
       NEW PARAMETER FILE : R02601.PAR


 ALPHA  =  1.1393 AND T0  =  0.0095


FORTRAN STOP
```

Computer programs

Program	FCL.EXE
Input files	Grouped data set, R(run number).DAT (Parameter file, R(run number).PAR (optional))
Output files	R(run number).FIT R(run number).PAR R(run number).OUT

Step 4. Group experimental data sets

Once the calibration parameters, α and t0 have been calculated from the T20 run the experimental datasets can be corrected and grouped as specified in LGROUP.DAT. This is carried out using either LNG or EMULNG , with the command

LNG First run number final run number dead time flag

The following points should be noted:

(i) a run of spectra can be grouped consecutively: to correct and group MuSR runs 2000 to 2050 with dead time corrections the command

LNG 2000 2050 0

would be used. The program will then prompt for the values of t0 and alpha calculated by T20FIT.

(ii) the correct value of t0 is calculated by taking the "first guess" value and subtracting the value obtained from fitting the 20 gauss transverse field spectrum. Values close to 0.62 for MuSR and 0.32 for EMU should be obtained.

(iii) The correct value of alpha is obtained by multiplying the initial value of alpha (usually 1) by the value calculated by the T20FIT program. For a thin, well centred sample alpha should lie between 0.9 and 1.1.

Pairs of files L(run number).LOG and R(run number).DAT will be created for each data set grouped.

Example

```
RAL>LNG 2599 2601
 T-ZERO (.61) ? : 0.6
 ALPHA ? : 1.14

 Run number, tzero, alpha and dead time flag
 0.6000000      1.140000          0
MUSR$DISK0:[DATA.MUSR]r02599.ral
created log file
created group  1
created group  2
Completed file      2599
Finished all files
FORTRAN STOP
```

Run number, tzero, alpha and dead time flag

```
 0.6000000      1.140000           0
MUSR$DISK0:[DATA.MUSR]r02600.ral
created log file
created group  1
created group  2
Completed file       2600
Finished all files
FORTRAN STOP

Run number, tzero, alpha and dead time flag
 0.6000000      1.140000           0
MUSR$DISK0:[DATA.MUSR]r02601.ral
created log file
created group  1
created group  2
Completed file       2601
Finished all files
FORTRAN STOP
```

Computer programs

Command file	LNG.COM or EMULNG.COM
Program	MGROUP.EXE or EMUGROUP.EXE
Input files	Raw data file LGROUP.DAT
Output files	TEMP.G1, TEMP.G2 L(run number).LOG R(run number).DAT

File formats

LGROUP.DAT	(see step 1)
TEMP.G#	N x (time, counts, σcounts)
R(run number).LOG	ASCII information file
R(run number).DAT	N x (time, asymmetry, σasymmetry)

Step 5. Create function to model data

The most common method of analysing longitudinal and zero field muon spin relaxation spectra is to generate and model the ratio of counts in the forward (F) and backward (B) detectors, R(t), where R(t) is defined by

$$R(t) = [N_F(t) - \alpha N_B(t)] / [N_F(t) + \alpha N_B(t)]$$

$$\text{or} \quad R(t) = a_0\, G_z(t)$$

where $G_z(t)$ is the longitudinal relaxation function and α is a calibration term to account for the relative efficiencies of the counters in the forward and backward detectors. RUMDA allows the user to select any functional form to describe the longitudinal relaxation. This is achieved by creating a short .FOR file which can be linked to a non linear least squares fitting routine, MUFIT2.

Directories created by SETUP.COM (see Introduction) will contain two simple function files; a gaussian function, GAUSS.FOR and an exponential, LOREN.FOR. These can be used as "skeleton" files by the users to create a function to describe the data.

Useful Functions

Gaussian $\quad R(t) = a_O \exp\left(-\frac{1}{2}\sigma^2 t^2\right)$

Lorentzian $\quad R(t) = a_O \exp(-\lambda t)$

Stretched exponential $\quad R(t) = a_O \exp(-\lambda t)^\beta$

Static gaussian Kubo-Toyabe $\quad R(t) = a_O\left[\frac{1}{3} + \frac{2}{3}\left(1 - \sigma^2 t^2\right)\exp\left(-\frac{1}{2}\sigma^2 t^2\right)\right]$

The <function name>.FOR file must be compiled before it can be linked to the fitting program. This is achieved by typing:

FOR <function name>

If the FORTRAN code has been typed correctly the computer will return with a prompt once the object (.OBJ) file has been created. However, if there are mistakes in the code these will be listed line by line at this point.

Examples of .FOR files for commonly used functions

GAUSS.FOR

```
      FUNCTION FUNCTN(X,I,A)
      DIMENSION X(1),A(1)
C
C  FUNCTION TO FIT GAUSSIAN
C
      XI=X(I)*X(I)
      A1=ABS(A(1))
      A2=A(2)*A(2)
C
      FUNCTN=A1*exp(-0.5*(A2*XI))
      RETURN
      END
```

LOREN.FOR

```
      FUNCTION FUNCTN(X,I,A)
      DIMENSION X(1),A(1)
C
C  FUNCTION TO FIT LORENTZIAN
C
      XI=X(I)
      A1=ABS(A(1))
      A2=ABS(A(2))
C
      FUNCTN=A1*exp(-(A2*XI))
      RETURN
      END
```

Step 6. Create executable file for fitting data

Two object files (.OBJ) should now exist in the directory

(i) the function to model the data and
(ii) the non linear least squares fitting routine MUFIT2.OBJ.

To create an executable file type

```
LINK  <Function name>,<MUFIT2>
```

This will create a file <Function name>.EXE which will perform the fit.

Example

```
RAL> dir *.obj

Directory SCRATCH$DISK:[MUSR01.RAL]

MUFIT2.OBJ;2                GAUSS.OBJ;1

Total of 2 files

RAL> LINK GAUSS,MUFIT2
RAL>
RAL>
RAL> dir gauss.exe

Directory SCRATCH$DISK:[MUSR01.RAL]

GAUSS.EXE;1

Total of 1 file

RAL>
RAL>
```

Step 7. Fitting data

Fitting the data can be carried out either

(i) interactively or
(ii) automatically as a pseudo batch job.

(i) Interactive Mode - Start the program running by typing

RUN <Function name>

The program begins by requesting the following information:

Run number	The number determined by MCS during data collection
Input parameters	Parameters can be entered by the user (option (U)), from an old parameter file (option (F)) or from a parameter file generated by an earlier fit to this data set (option ( ))
Start and stop times (in μs)	These times determine the time-window over which the fit is performed
Maximum number of iterations	Maximum number of cycles before terminating fit, 0 gives a simulation
Number of parameters	Number of variables in the function, (between 1 and 10)
A and DeltaA	Starting point and initial step size of parameter A(n) This is repeated for each variable in the function.

The program then starts the fit, displaying the results after each iteration. A record of the fit is saved in the file R(run number).OUT, and the fitted parameters in R(run number).PAR. The resulting line shape is saved in R(run number).FIT. The data and the fit can be plotted using GENIE, as described in section 3.

Example

```
RAL>RUN STREX


          **** PROGRAM TO FIT MUSR DATA ****


ENTER RUN NUMBER OF INPUT DATA FILE
2600
INPUT PARAMETERS BY USER (U), NAMED FILE (F) OR CURRENT .PAR FILE ( )
u
START AND STOP TIMES (MUSEC)
0.5 10
ENTER MAXIMUM ITERATIONS (0 FOR SIMULATION)
100
ENTER NUMBER OF PARAMETERS
4
ENTER A 1 AND DELTAA 1
.15,.01
ENTER A 2 AND DELTAA 2
.1,.01
ENTER A 3 AND DELTAA 3
1,.1
ENTER A 4 AND DELTAA 4
.05,.005

FILE NAME : R02600.OUT


   FITTED PARAMETERS

  CHISQR    A 1     A 2     A 3     A 4
    1.397  0.081  0.181  0.589  0.086
    1.271  0.109  0.134  0.242  0.082
    1.102  0.194  0.115  0.163  0.046
    1.064  0.233  0.100  0.166  0.030
    1.054  0.235  0.114  0.164  0.027
    1.016  0.276  0.100  0.133  0.014
    1.013  0.276  0.108  0.139  0.012
    1.002  0.305  0.099  0.123  0.002
    0.991  0.305  0.103  0.122  0.002
    0.988  0.305  0.101  0.124  0.002
    0.987  0.305  0.100  0.125  0.002


 NUMBER ITERATIONS = 11,  FINAL CHISQR =  0.98663586

 FINAL VALUES OF PARAMETERS
```

```
          A(N)                  ERROR
A( 1)   0.3049159050        0.5569971981E-03
A( 2)   0.1004454792        0.1478637918E-02
A( 3)   0.1246657446        0.1250753761E-02
A( 4)   0.1572212321E-02 0.7296421682E-03

FOR FITTED DATA SEE FILE : R02600.OUT
   FITTED DATA FOR GENIE : R02600.FIT
      NEW PARAMETER FILE : R02600.PAR

FORTRAN STOP
```

(ii) Pseudo-batch Mode - In the pseudo batch mode *consecutive* data sets can be fitted. This is a very quick way of modelling smoothly varying line shapes. The parameters saved in the .PAR file from one fit are used as the starting parameters for the next fit. The first spectrum must be fitted interactively and a .PAR file created; then start the automatic program running by typing:

FITB

The program then prompts for the program name, the first run number and the final run number. No further user input is required. The program starts fitting the data, run by run, displaying the results after each iteration. A record of each fit is saved in the file R(run number).OUT, and the fitted parameters in R(run number).PAR. The resulting line shapes are saved in R(run number).FIT.

Computer files

Command file for automatic mode	FITB
Program	<Function name>.EXE
Input files	Grouped data set, R(run number).DAT Parameter file, R(run number).PAR (optional for interactive mode, essential for automatic running)
Output files	R(run number).FIT R(run number).PAR R(run number).OUT

3. PLOTTING DATA AND FITS USING 'GENIE'

3.1 Initialising GENIE

Enter GENIE by typing : **GENIE**

A GKS screen will be created and the command file GENIEINIT.COM run.
A complete GENIE guide (RAL-86-102) is available from the computer support office in R3.

Example

After logging in type **genie** to initialise GENIE and generate a graphics window. The screen displays the program status at each stage of the initialisation:

```
RAL>GENIE

>> genieinit.com
>>
>> set format muon
>> Data will be read assuming muon MCS format
>> set inst r
>>  default instrument is now :R
>> set ext .ral
>>  default extension is now :.RAL
>> set work 40 2500
>>          *** WORK SPACE STATUS ***                          setting defaults
>>    40 work spaces   : length =  2500 channels
>>     1 graphics space: length =  2500 channels
>>    39 buffers       : length =  2500 channels
>> set dir [data.musr]
>>  default directory is now :[DATA.MUSR]
>> set disk musr$disk0:
>>  default disk is now :MUSR$DISK0:



>> a p 0 1 0 1
>> a b 3
>> t h                                                          determine size and
>>  no graph header will be displayed                           style of graphics
>> t m
>>  point plotting mode selected
>> readin:== @user$disk:[musr01.com]readin
>> readmcs:== @user$disk:[musr01.com]readmcs
>> vf:== @user$disk:[musr01.com]viewfit                    setting up aliases
>> s:== @user$disk:[musr01.com]small                       for command files
>> b:== @user$disk:[musr01.com]big
>> hc5:== @user$disk:[musr01.com]hc5
>> hc0:== @user$disk:[musr01.COM]hc0
>>
>>
     THIS VERSION OF GENIE IS NOW CAPABLE OF READING MUSR FILES
  STRAIGHT FROM DISK  TO DO SO TYPE "READMCS" AT THE GENIE PROMPT
>>
```

3.2. Displaying data

Once GENIE has been initialised it is possible to manipulate the data and display it in a suitable form. Ungrouped files can be read in and displayed using READMCS, one or more grouped files can be plotted using READIN, or a relaxation spectrum and its fit displayed using VIEWFIT.

3.2.1. "READMCS"

READMCS is a command procedure that will display spectra in GENIE collected straight from the instrument. It is NOT necessary to have a R<run number>.dat file for this program. At the GENIE prompt type

READMCS

you will now be prompted for the run number that you wish to plot i.e. '1879' or '2389', followed by a prompt for the value of alpha. READMCS then loads up the first 32 GENIE workspaces with individual detector data. Workspace 33 is loaded with the sum of the first 16 detectors and workspace 34 is loaded with the sum of the last 16 detectors (e.g. forward and backward grouping), workspace 35 is loaded with the sum of last 16 detectors multiplied by alpha and workspace is loaded with the sum of workspace 33 minus workspace 35 divided by workspace 33 plus workspace 35.

Example

```
>> READMCS
>>  @USER$DISK:[MUSR01.COM]READMCS
>> a p 0 1 0 1
>> a b 10
>> a m 4
>>
>>          To view a MCS file
>>
>>   FILENUMBER ? (i.e. 2000):
```

2384

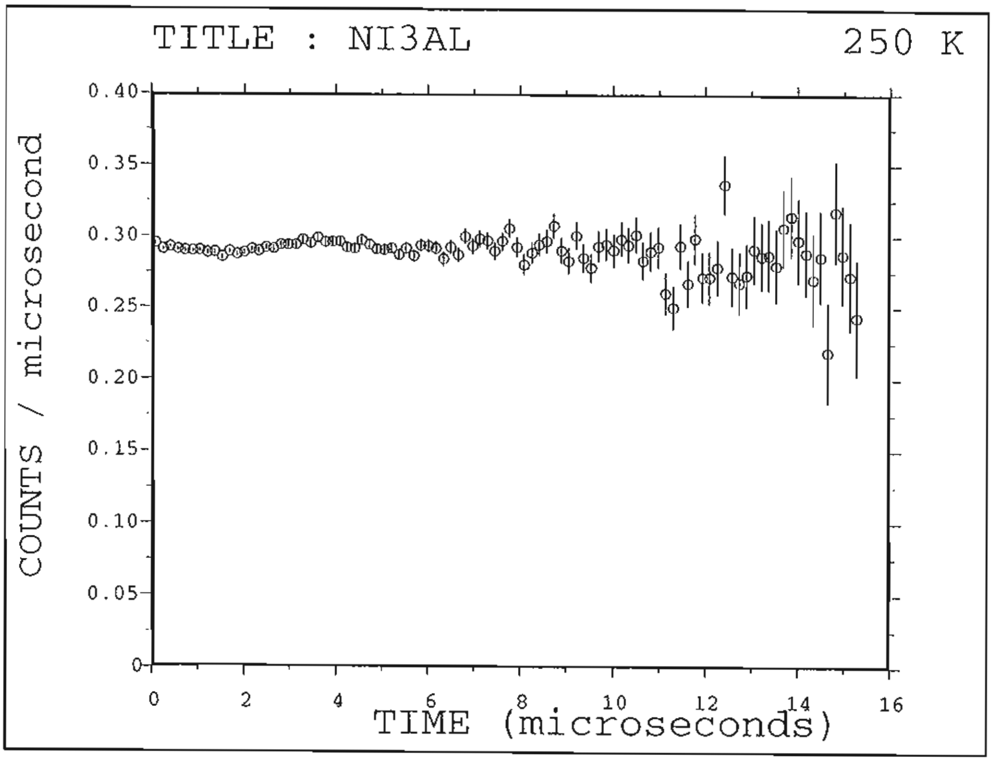
```
>>   ENTER VALUE FOR ALPHA:
1
>>
>> ASS 2384
>>
>> limit/default
>>
>>   W1 = S1

Load     Spectrum    1 into Workfile   1
>>   W2 = S2
Load     Spectrum    2 into Workfile   2
>>   W3 = S3
Load     Spectrum    3 into Workfile   3
>>   W4 = S4
.
.                    etc.
.
Load     Spectrum   30 into Workfile   30
>>   W31 = S31
Load     Spectrum   31 into Workfile   31
>>   W32 = S32
Load     Spectrum   32 into Workfile   32
>>
>>
>> W33 = (S1>S16)
Load     Spectrum    1 into Buffer     1
Add      Spectrum    2 to   Buffer     1
Add      Spectrum    3 to   Buffer     1
.
.                    etc.
.
Add      Spectrum   14 to   Buffer     1
Add      Spectrum   15 to   Buffer     1
Add      Spectrum   16 to   Buffer     1
Load     Buffer     1 into Workfile   33
>>
>>
>> W34 = (S17>S32)
Load     Spectrum   17 into Buffer     1
Add      Spectrum   18 to   Buffer     1
Add      Spectrum   19 to   Buffer     1
.
.                    etc.
.
Add      Spectrum   30 to   Buffer     1
Add      Spectrum   31 to   Buffer     1
Add      Spectrum   32 to   Buffer     1
Load     Buffer     1 into Workfile   34
>> W35 = (1 * W34)
Multiply  Workfile  34 into Buffer     1
Load     Buffer     1 into Workfile   35
>>
>>
>> W36 = (W33 - W35)/(W33 + W35)
Load     Workfile  33 into Buffer     1
Add      Workfile  35 to   Buffer     1 with multiplier     -1.0000000
Load     Workfile  33 into Buffer     2
```

```
Add     Workfile  35 to  Buffer    2
Divide   Buffer    2 into Buffer    1
Load     Buffer    1 into Workfile  36
>>
>>d w36
```

The ratio (F-αB)/(F+αB) is then displayed as a graph.

Figure 1. Output from READMCS



3.2.2. Plotting raw data

To plot the ratio (F-αB)/(F+αB) once a file has been grouped, type

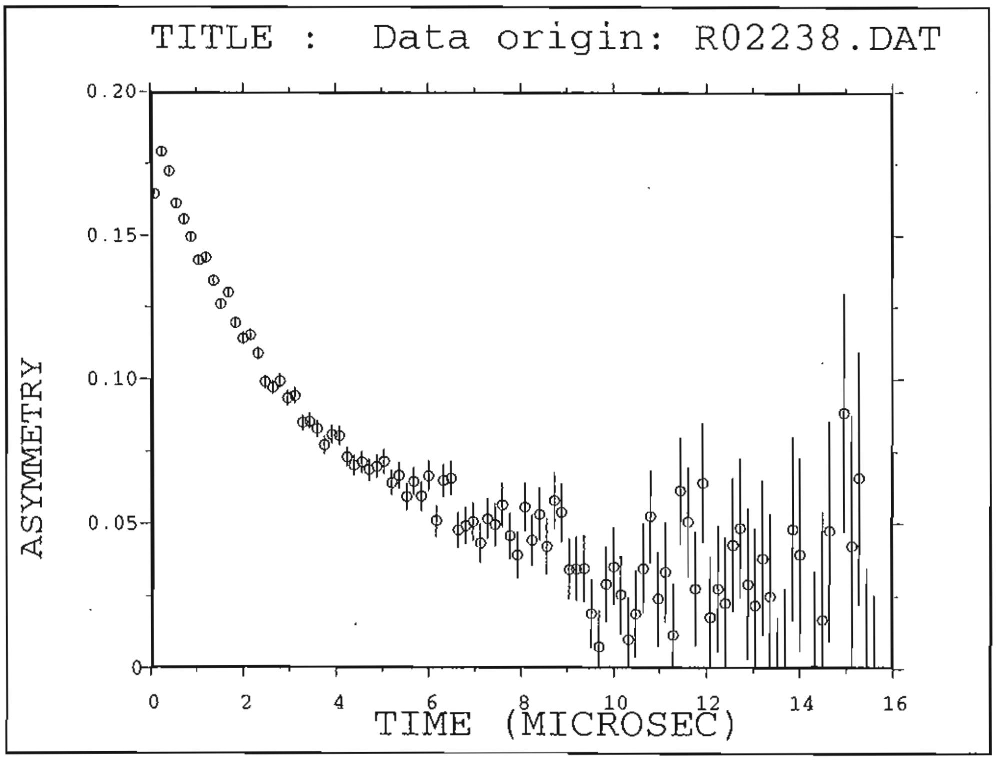
READIN

at the GENIE prompt (>>). You will be prompted for a run number and a GENIE workspace number. The command file displays the data on a default set of axes. These can be changed using standard GENIE commands. e.g. d/m w# xmin xmax ymin ymax displays the data as markers between the limits specified. All usual GENIE commands can also be used.

Example

```
>> READIN
>> @USER$DISK:[MUSR01.COM]READIN
>> a b 10
>> A M 4
>>          TO VIEW A DATA FILE
>>
>>   FILENUMBER ? (NO EXTENSION):
2238
>>   WORKSPACE? (NO W):
1
>> LO W1 R02238.DAT RP:MUIN
***   963 POINTS READ FROM FILE R02238.DAT                    ***
FORTRAN STOP
>> L/DEF
>> d/m wl 0 16
```

Figure 2. Output from READIN



3.2.3. Plotting fitted data and fitted curve

To view the raw data with the fit superimposed type

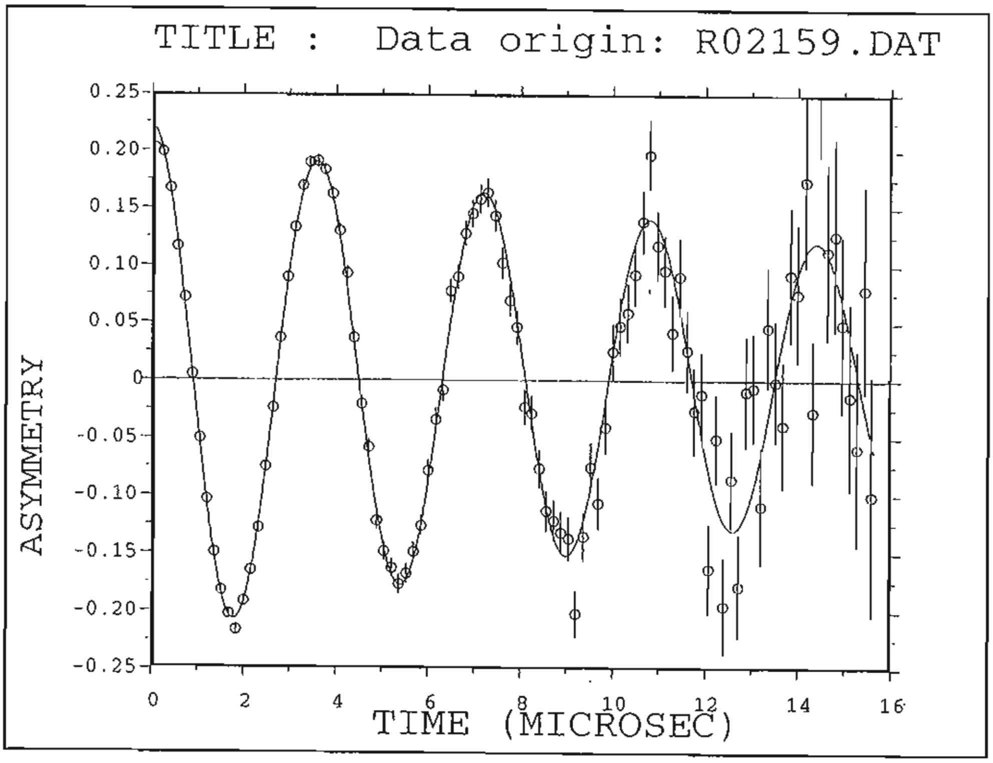
VF

at the GENIE prompt (>>). The program prompts for a run number only; no further information is required. The raw data is read into workspace 1 and the fit into workspace 2. A plot of the raw data, error bars and the fit is then displayed on a set of default axes. These can be changed using standard GENIE commands.

Example

```
>> VF
>>  @USER$DISK:[MUSR01.COM]VIEWFIT
>>
>>          TO VIEW A DATA FILE AND FIT (W1 AND W2)
>>
>>   FILENUMBER ? (NO EXTENSION):
2159
>> LO W1 R02159.DAT USER$DISK:[MUSR01.rumda.LPROG]MUIN
***  963 POINTS READ FROM FILE R02159.DAT                    ***
FORTRAN STOP
>> LO W2 R02159.FIT USER$DISK:[MUSR01.rumda.LPROG]MUIN
***  1001 POINTS READ FROM FILE R02159.FIT                   ***
FORTRAN STOP
>> A M 4
>> a b 10
>> D W1
>> D/M W1 0 16
>> P/E W1
>> P/L W2
```

Figure 3. Output from VF



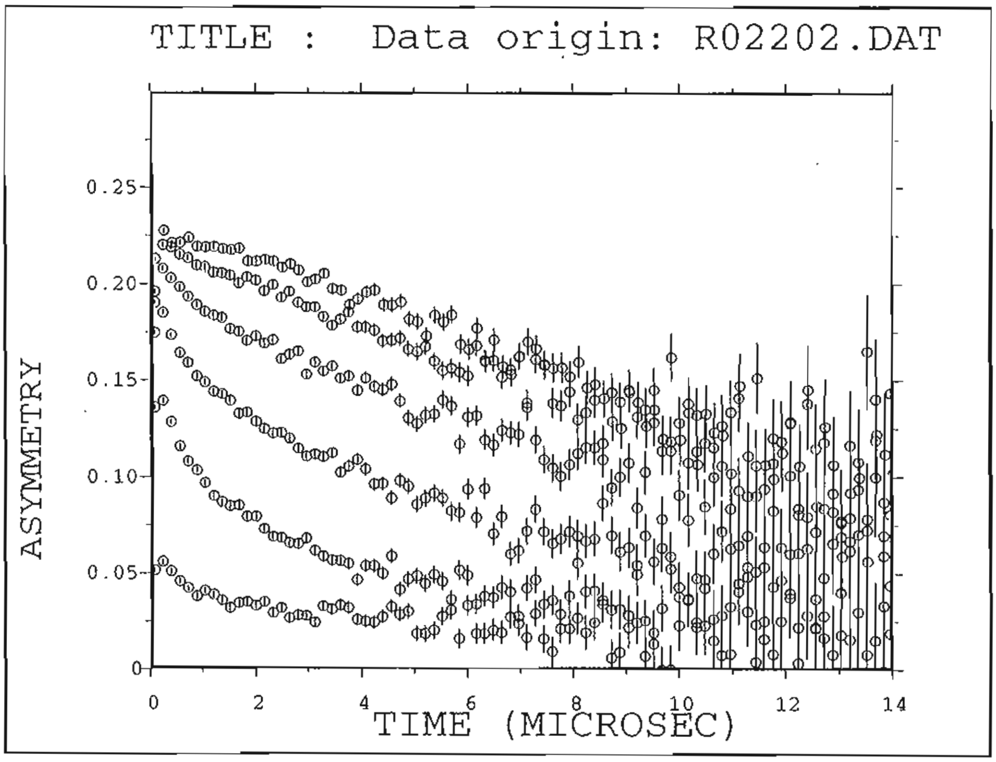
3.2.4 Multiplots of results

It is possible to create a muliplot of several datasets or of several datasets and their fits.

The command **@MULP** will multiplot a series of data sets, R(run number).DAT, with a variable increment in the run numbers of files to be plotted: i.e. it is possible to multiplot every third file between run numbers 2000 and 2030. A maximum of 40 files can be loaded into GENIE at any one time.

To multiplot several datasets, R(run number).DAT, and their fits, R(run number).FIT, use the command **@MULPF** at the GENIE prompt. A limit of 20 datasets and 20 fits exists.

Figure 4. Output from MULP.COM



3.3. Making a hard copy of a plot

Two command files exist, **HC0.COM and HC5.COM**. These will save a GENIE graphics screen as DEC_POSTSCRIPT.DAT. This file is then sent automatically to either laser printer 0 (Computer support office in R3) or laser printer 5 (MuSR cabin).

3.4. Listings of useful GENIE command files

If you wish to customise a command file please make a copy of the source file from **user$disk:[musr01.com]** and create your version with a new name. You can run the command file from inside GENIE by typing @<filename>.

<u>**READIN.COM**</u>

```
>! READIN.COM
> A P 0 1 0 1
> A B 10
> A M 4
!
!
!       TO VIEW A DATA FILE
!
$ INQUIRE IFILE "FILENUMBER ? (NO EXTENSION)"
$ INQUIRE IWORK "WORKSPACE? (NO W)"
!
> LO W'IWORK' R0'IFILE'.DAT RP:MUIN
> L/DEF
!
> D/M W'IWORK' 0 16
> P/E W'IWORK'
```

<u>**VIEWFIT.COM**</u>

```
>! VIEWFIT.COM
!!
!       TO VIEW A DATA FILE AND FIT (W1 AND W2)
!
$ INQUIRE IFILE "FILENUMBER ? (NO EXTENSION)"
!
> LO W1 R0'IFILE'.DAT USER$DISK:[MUSR01.RUMDA.LPROG}MUIN
> LO W2 R0'IFILE'.FIT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
!
> A M 4
> A B 10
> D W1
>!
> D/M W1 0 16
> P/E W1
> P/L W2
```

MULP.COM

```
>! MULP
!        TO VIEW A SERIES OF DATA FILES
!
$ INQUIRE NFILE "  NUMBER OF FILES ? "
$ INQUIRE IFILE "FIRST FILE NUMBER ? "
$ INQUIRE JFILE "    PLOT EVERY ...? "
!
> LO W1 R0'IFILE'.DAT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
> A M 4
> A B 10
> D/M W1 0 14 0 .3
!
$ DO I=1,'NFILE'
> LO W'I' R0'IFILE'.DAT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
> P/M W'I'
> P/E W'I'
!
$ IFILE=IFILE+'JFILE'
$ END DO
```

MULPF.COM

```
!      TO VIEW A SERIES OF DATA FILES AND FITS
!
$ INQUIRE NFILE "  NUMBER OF FILES ? "
$ INQUIRE IFILE "FIRST FILE NUMBER ? "
$ INQUIRE JFILE "    PLOT EVERY ...? "
!
> LO W1 R0'IFILE'.DAT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
> A M 4
> A B 10
> D/M W1 0 12 0 .3
!
$ DO I=1,'NFILE'
> LO W'I' R0'IFILE'.DAT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
> P/M W'I'
> P/E W'I'
> LO W'I' R0'IFILE'.FIT USER$DISK:[MUSR01.RUMDA.LPROG]MUIN
> A B 1
> P/L W'I'
!
$ IFILE=IFILE+'JFILE'
$ END DO
```

HC0.COM

```
>  @USER$DISK:[MUSR01.COM]HC0
>
>  K/H
>  J "PLASER0 DEC_POSTSCRIPT.DAT
```

4. DEAD TIME CORRECTIONS

4.1. Theory

A simple, non-paralyzable dead time model has been used to determine the dead time corrections of the muon spectra. In the non-paralyzable model the detector is dead for a time t_b after a count. Defining the variables as

$$\left.\begin{array}{l} n = \text{true interaction rate} \\ m = \text{recorded count rate} \end{array}\right\} \quad \text{in a single frame for a single PMT}$$

t_b = bin width

t_d = system dead time

then the real number of events arriving at the detector in a time bin t_b wide is $n.t_b = N$ per frame per PMT. The measured number of counts is $m.t_b = M$ per frame per PMT. The "lost" counts are therefore $(N-M) = t_b\,(n-m)$

However, the detector is "dead" for a time $m.t_b.t_d$, where t_d is the dead time. Thus the total number of lost counts is $n.m.t_b.t_d$ i.e.

$$N-M = N.(M/t_b).t_d$$

$$= N.M.c \quad (\text{where } c = \ t_d/t_b)$$

$$\therefore\ N = M\,/\,(1-M.c) \qquad \text{eq 1}$$

The true counts falling on a detector in a bin width t_b is

$$N = n_o.t_b.\exp(-\lambda t) + n_b.t_b$$

where $n_b.t_b$ is a negligible background and $\lambda = 1/t_\mu$. Using $N = N_o.\exp(-t/t_\mu)$ equation 1 can be rewritten as

$$M\,\exp(t/t_\mu) = N_o - N_o.c.M$$

where N_o is the true number of counts at time $t = 0$. The data can be replotted in the form $M.\exp(-t/t_\mu)$ vs. M which will yield a straight line with an intercept of N_o and a slope of -$N_o.c$. Therefore c, and hence t_d can be calculated.

4.2. Computer programs

Program	DEADTIM.FOR
Input files	Silver spectrum with good statistics (100M events) LGRPDT.DAT
Output files	DTPAR.DAT or DTEPAR.DAT DTFIT.PAR or DTEFIT.PAR DTEMP.G##, where ## is the histogram number DTFIT.G##, where ## is the histogram number

DEADTIM reads a raw silver data file and estimates the dead time corrections for the particular detector groupings listed in LGRPDT.DAT. The procedure is valid only if each histogram is analysed and corrected individually, but the program can be run with grouped data to estimate how significant dead time corrections might be.

DEADTIM performs the data manipulation and the linear fit, creating an output file DTPAR.DAT for MuSR or DTEPAR.DAT for EMU. (For grouped data the histogram number is replaced by group number and the filename is DTPARG.DAT.)

DEADTIM.EXE also creates a further file DTFIT.PAR containing all the fitted parameters in the following order: histogram number, N_0, σN_0, c, σ_c, regression coefficient

DTPAR.DAT (or DTEPAR.DAT) for the individual histograms is read by the grouping program through LNG (or EMULNG) to correct the experimental data. This file must always be available for the corrections. Usually, it will be created by the instrument scientist during the first calibration day of each cycle.

4.3. File formats

LGRPDT.DAT	1 x (number of groups) 32 x (histogram number, group number)
DTPAR.DAT	1 x (number of histograms) 32 x (histogram number , dead time in ns)
DTFIT.PAR	32 x (hist. no., N_0, σN_0, c, σ_c, regression coefficient)
DTEMP.G##	32 x (M, M.exp(t/t_μ), σ)