Maximum Entropy Fourier Analysis in μSR: A Manual for MaxError

Draft in process! Large holes!

Manual located at

USER$DISK:[TANYA.TEX.MAXENT.MAN]MAXERR_MANUAL.TEX

For program version:

CMALPHA1::[TANYA.FOR.MAX99.SEP]MAX_MAIN_99SEP.EXE

001033

19990921

September 21, 1999

T.M. Riseman and E.M. Forgan,

School of Physics and Astronomy,
University of Birmingham,
Birmingham B15 2TT
United Kingdom

October 23, 1999

Contents

1 Preface **1**

1.1 How does MaxError differ from Maximum Entropy? 2
1.2 The name MaxError: A warning! 2
1.3 Overview of this manual's structure 2
1.4 Degree of availability 3
1.5 Legal matters 4

2 Installation **5**

2.1 Conventions used in this manual 6
2.2 Version and directory structure 6
2.3 Documentation 6
2.4 Source code 6
2.5 μSR data formats available 8
2.6 Logicals and files needed to run MaxError 8
2.7 Output Files 9
2.8 Limitations of MaxError program 9

3 Brief Mathematical Introduction to ME **11**

3.1 Introduction 12
3.2 The μSR situation 12
3.3 Fourier Transform of μSR data 13
3.4 Use which technique to get $P(b)$? 13
3.5 ME: Animal, mineral or vegetable? 14
3.6 How many parameters can you use in a fit? 15
3.7 How entropy helps 16
3.8 The Search Algorithm in Maximum Entropy 16
3.9 ME fiddle factors: the default level 16
3.10 ME Fiddle factors: σ looseness 17
3.11 Difference between ME and a Fourier Transform 17
3.12 Basic choices: Binning and time window 17
3.13 Fancy choices: Apodization and variable binning 18
3.14 Output: Enhancement factor 18
3.15 Output: Hey, what about error estimation? 18
3.16 Output: The noise level and P_d 18

4 Concise Advice on Usage **21**
4.1 Summary of advice: Program choices 22
4.1.1 Conventions used in I/O 22
4.1.2 The markers file VMARKER_%%%.DAT 22
4.1.3 Before starting MaxError 23
4.1.4 Choices made at the beginning of MaxError 23
4.1.5 Choices made at the end of MaxError 26
4.2 Low statistics data: May the noise be with you 27
4.3 Summary of advice: Is the result OK? 28
4.3.1 Numerical indications that results are good 28
4.3.2 Graphical indications that results are good 29
4.3.3 Numerical indications that results are poor 29
4.3.4 Graphical indications that results are poor 30
4.4 To do 31

5 Graphically Viewing Output **33**
5.1 Output Files 34
5.2 Viewing using DB 34
5.2.1 An example of using DB 35
5.2.2 Assessment 35
5.3 Viewing using spreadsheet programs via DB_SPREAD.EXE 36
5.3.1 An example of using DB_SPREAD 36
5.3.2 Assessment 37
5.4 Viewing using Physica (and other math programs) 37
5.4.1 An example of using macros in PHYSICA 38
5.4.2 Assessment 38

6 Advice that MaxError outputs **41**
6.1 Introduction 42
6.2 Version number info 42
6.3 Group removal 42
6.4 Time binning adjustment 42
6.5 Apodization of σ errors 43
6.6 Number of FFT points 43
6.7 Field range selection 44
6.8 Is assumption of Gaussian statistics satisfied? 44
6.9 Modified parameters outside ME 45
6.10 Reports from the ME search engine 45
6.11 Success? 46
6.12 When are errors calculated? 47
6.13 Peaks 47
6.14 Outliers 47
6.15 Time range OK? 49
6.16 Time bin suggestions 49
6.17 General reminder about how ME was done 50
6.18 Recount of data input 51
6.19 Recount of how MaxError was performed 52
6.20 Recount of internal indicators 52
6.21 Recount of histogram parameters 53
6.22 Recount of advice based on statistics of run 54

6.23 Recount of calculated moments 54

7 Advice (and Speculation) on MaxError Usage 57
7.1 How many parameters can you use in a fit? 58
7.1.1 Overdetermined theory functions 58
7.1.2 What should you compare χ^2 to? 58
7.2 How entropy helps 58
7.2.1 Noise suppression due to S 58
7.2.2 Speculation: Alternatives to the entropy term 58
7.3 The Search Algorithm in Maximum Entropy 59
7.4 ME fiddle factors: the default level 59
7.4.1 The *previous* conventional wisdom about P_d 59
7.4.2 The (probably) correct value of P_d 59
7.4.3 How is P_d affected by time binning? 60
7.4.4 What happens if you fiddle P_d? 60
7.4.5 How to test the value of P_d 60
7.5 ME Fiddle factors: σ looseness 61
7.6 Difference between ME and a Fourier Transform 61
7.7 Basic choices: Binning and time window 61
7.8 Fancy choices: Apodization and variable binning 61
7.9 Output: Enhancement factor 61
7.10 Output: Hey, what about error estimation? 61
7.10.1 Are the errors correct? Does it matter? 61
7.11 Output: The noise level and P_d 62

8 The Basics of Maximum Entropy 63
8.1 How entropy helps 64
8.2 The Search Algorithm in Maximum Entropy 64
8.3 ME fiddle factors: the default level 64
8.4 ME Fiddle factors: σ looseness 64
8.5 Difference between ME and a Fourier Transform 64
8.6 Basic choices: Binning and time window 64
8.7 Fancy choices: Apodization and variable binning 64
8.8 Output: Enhancement factor 64
8.9 Output: Hey, what about error estimation? 64
8.10 Output: The noise level and P_d 64

9 Unfinished business: Promises, things to do, etc. 65
9.1 Specific chores 66
9.1.1 MaxError chores 66
9.1.2 Other related programming chores 66
9.1.3 Specific editing/writing chores 66
9.1.4 Specific Latex chores 67
9.2 General chores 67
9.2.1 General editing chores 67
9.2.2 General Latex chores 67

List of Figures

3.1 S and χ^2 Space 19

4.1 A typical markers file VMARKER_BRA.DAT 23

List of Tables

2.1 Summary of where files are kept on CMALPHA1 6
2.2 Source code files 7
2.3 μSR data formats 8
2.4 μSR data formats logicals 8
2.5 Files needed to run MaxError 9
2.6 Example output files 9

5.1 Example output files 34

Chapter 1

Preface

1.1 How does MaxError differ from Maximum Entropy?

This is the manual for a program called MaxError, for making maximum entropy Fourier analysis of μSR data.

Maximum entropy Fourier analysis of μSR data is an alternative to traditional Fourier transform. Both serve to produce a lineshape corresponding to the distribution of local fields which created the observed μSR spectra. However, as will be described later on in greater detail (Sec. ??), the maximum entropy technique naturally suppresses noise, thereby giving a cleaner lineshape than traditional Fourier transforms.

The maximum entropy engine was provided by Prof. Geoff Daniell, University of Southampton, and the original code (usually referred to as MaxEnt) was written by Geoff Daniell and Brian Rainford[2], also of the University of Southampton. A divergent version at use at ISIS is called Maximum Entropy Signal Analysis (MESA)[?].

We will refer to the maximum entropy engine as ME for short.

What makes this version MaxError much more powerful than previous versions is that it

- Uses information from the ME search algorithm to automatically estimate errors on the results.
- Automatically provides reasonable and justifiable values for ME parameters which were previous considered to to be essentially arbitrary "fiddle" factors, such as σ-looseness and the default value.
- Offers additional control with choice of binning, the time range and (shocking!) apodization.
- It contains code to produce tradition Fourier transforms under the same conditions of binning and time range, to allow the curious to compare the effects of the two techniques.

1.2 The name MaxError: A warning!

We named this program somewhat humorously MaxError, to emphasize its new ability to estimate errors associated with the results and to differentiate it from other program versions. As it is a very powerful program with many parameters left adjustable even when the optimal value is automatically provided, the name also cynically reflects the potential for user to make many mistakes. Consider its humorous name to be a warning as well!

1.3 Overview of this manual's structure

This manual attempts the ambitious task of being

- an installation guide (Chapter 2),
- a elementary introduction to maximum entropy and its application to μSR (Chapter 3),

- a nuts-and-bolts manual on how to use the program, with all the program prompts clearly indicated in `typewriter font` and what default answers you should usually use (Chapter 4),
- how to graphically view the results (Chapter 5),
- a source of advice on what the intermediate and final results mean and how to use it as well as one can (Chapter 4 giving bare bones advice, Chapter 6 reviewing the advice MaxError provides automatically, and Chapter 7 giving more detailed advice, explanations and speculation),
- and, finally, a detailed description of the mathematics used in the code, where it has not been covered earlier (Chapter 8).

In an attempt to avoid overwhelming the user, the rather different aspects have been separated out into different parts of the manual. Extensive cross referencing between the chapters should help some one jump "horizontally" to increasing depths of explanation for a given topic. (If I pull it off, maybe even the chapter and section titles will correspond in the different parts!) The user of the hard copy will be able to use the table of contents and the index to help with this horizonal movement. The PDF file on one's PDF viewer will show the cross referencing as clickable links, allowing this manual to be used as a convenient on-line resource.

If you want to USE MaxError rather blindly and quickly, go directly to Chapter 4.

1.4 Degree of availability

There are several issues involved with when we decide to make this program "publically" available. Firstly, we would prefer to make sure that it is reasonably stable, debugged, and that we have (mostly) run out of important features to add. This takes time, of course, and it is very important to test the program on both simulated data and "in-the-field", to use a military analogy.

The second issue resolves around how easy it is to use correctly. If the program is widely available but used very poorly by the majority of users, it will reflect very badly on both the program and (much worse!) on μSR. Conceptually, maximum entropy is a great deal more sophisticated than traditional Fourier transform. People often have trouble using ordinary Fourier transforms to best advantage. The maximum entropy technique has potentially even more adjustable parameters than a Fourier transform, so the degrees of freedom can become overwhelming.

Keeping this difficulty in mind, we have made progress towards automatically providing reasonable and justifiable values for most of the adjustable parameters. Now, the most of the remaining routinely adjustable parameters are exactly those for a traditional Fourier transform with apodization. So, in principle, the program should be usable by people who are competent at traditional Fourier transforms. However, we don't have much experience training a lot of people in the use of this program, and don't really want to anyway. Hopefully, this manual will take the place of a live teacher, for the most part. But in some sense, it will take some field testing to see how well novices pick it up, before the program is safe to release to general use.

However, it is my (T.M.R.) personal hope that this program will eventually become widely used and help improve the overall quality of μSR data analysis.

1.5 Legal matters

All trademarks are acknowledged??

This software is supplied "as is" without warranties as to correctness, merchantability or fitness for a particular purpose. No responsibility is accepted for loss or adverse consequences arising from the incorrect functioning of, misuse of, or *inability to use* this software. Use of this software constitutes an acceptance of the conditions set out in this section.

Chapter 2

Installation

files	Section	directories
source code	2.4	DKA200:[TANYA.FOR.MAX99.SEP]
documentation	2.3	DKA200:[TANYA.TEX.MAXENT.MAN]*.TEX, *.BIB
examples	5.1	DKA200:[TANYA.TEX.MAXENT.MAN.EXAMPLES.19990921]
data	2.5, 2.6	usually DATA$DISK:[PSI...], DATA$DISK:[RAL...]
Viewing programs	Chapter 5	
DB	5.2	DATA$DISK:[PROGRAMS.TRIUMF.MSR.BIN]DB.EXE
		USER$DISK:[TANYA.TEX.PROGRAMS_DOC]DB_MANUAL.TEX
db_spread	5.3	USER$DISK:[TANYA.FOR.DBDUMB]DB_SPREAD.EXE
Spreadsheet	5.3	your PC or Mac
Physica	5.4	DATA$DISK:[PROGRAMS.TRIUMF.PHYSICA]PHYSICA.EXE

Table 2.1: Summary of where files are kept on CMALPHA1

2.1 Conventions used in this manual

All program I/O excerpts are in a `typewriter font`. User provided answers to program prompts are in *`slanted typewriter font`*. All file names are in a sans serif font. All system commands (DCL in VMS, DOS in MS-windows, shell commands in Unix) and some directory listings are in a `typewriter font`.

2.2 Version and directory structure

Note: This is the MaxError manual's chapter on installing MaxError version 19990921. It should only be included for 19990921 = 19990921.

Note: unless indicated otherwise, all file directories referred to are those on the computer CMALPHA1.PH.BHAM.AC.UK, see Table 2.1 for a summary of where files are kept. The directory structure associated with *your* installation will vary.

2.3 Documentation

This MaxError manual was produced using the main LaTex file

DKA200:[TANYA.TEX.MAXENT.MAN]MAXERR_MANUAL.TEX

which then includes additional files containing individual chapters, a BibTex file and figures. Hopefully, you are either reading a PS print out or previewing it as PDF file with hyperlinks.

2.4 Source code

MaxError, version CMALPHA1::[TANYA.FOR.MAX99.SEP]MAX_MAIN_99SEP.EXE, date 19990921, version number 001033 runs under

- Alpha-chip Open-VMS,

using only DEC FORTRAN and no graphics packages. In the past we have ported a very old version of MaxEnt (from which this version of MaxError

```
Directory DKA200:[TANYA.FOR.MAX99.SEP]

MAX_FFT_99AUGA.FOR;10          62  24-SEP-1999 17:36:48.75
MAX_FMT_99SEP.FOR;7           188  22-SEP-1999 13:51:08.83
MAX_GUTS_99JULV.FOR;15         63   4-AUG-1999 19:31:25.32
MAX_INVERTM_98NOV.FOR;1         5  22-NOV-1998 13:09:31.74
MAX_MAIN_99AUGA.FOR;18         44  25-AUG-1999 14:43:37.97
MAX_MAIN_99SEP.FOR;2           44  21-SEP-1999 16:46:00.51
MAX_MOMENTS_99AUGA.FOR;2       63  25-AUG-1999 14:12:10.00
MAX_MUSR_99SEP.FOR;14         346  24-SEP-1999 10:22:09.12
MAX_NUMERICAL_RECIPES.FOR;1    13  12-AUG-1999 15:54:15.10
VMSTIME.FOR;1                   7  22-NOV-1998 13:11:31.85

DATA_FMT.INC;1                  5  22-NOV-1998 13:12:09.08
MAXERRORS_EAPOD.INC;15          4  25-AUG-1999 16:21:44.12
MAXERRORS_FMT.INC;11           19   5-AUG-1999 18:14:09.95
MAXERRORS_FMT_VBIN.INC;13       2   3-AUG-1999 16:28:46.95
MAXERRORS_GUTS.INC;10          11  21-JUL-1999 15:30:47.01
OUTLIERS.INC;4                  2  21-SEP-1999 16:28:49.34
PSI_HEADER.INC;1                5  22-NOV-1998 13:12:36.51
```

Table 2.2: Source code files

evolved from) to MS-Power FORTRAN and to Linux. We hope to port MaxError to Linux in the near future.

MaxError is currently only a text-based I/O program. In the past there were some very crude routines for plotting using the PGPLOT package, but the development effort to make that usable is not presently worthwhile. Windowing environments on VMS, UNIX, windows, NT, etc. are now standard, and therefore a do-it-all (calculate and plot) program which runs in a single terminal session is no longer necessary. We suggest that you view the output files graphically in a separate window, see Sec. ??.

The subdirectory that the code is contained in is named for year and month of the last modifications. All files needed for compilation are in this directory. The FORTRAN code files names end with a text string indicating which year and month they were last modified. The FORTRAN code files and include files for this version are are listed in Table 2.2.

Note that all system-dependent code are in files which start with the name of the operating system, *e.g.* VMSTIME.FOR.

They are compiled using the DEC utility MMS (or MadGoat equivalent MMK) using the VMS make file "DESCRIP.MMS". Just type "MMS" at the DCL prompt, **two times.** This is necessary because there are some warnings which stop the first make in the file MAX_FMT_*.FOR, due to misaligned fields in the include file PSI_HEADER.INC. This is due to the awkward design of the current PSI data format. If you don't have the utility MMS available, you may make a compile command file which uses the fortran compile options
`FORTRAN /warn/ext/LIST/show=include/align=record=pack`,
where the "align" option is critical for the PSI data to be read in properly.

# Code	Lab	format	File name	Enter as
1	PSI/RAL	USR	USR99999.USR	99999
2	PSI	PSI	DELTAT_FLC_9999.dat	9999
4	TRIUMF	TRI	099999.TRI	99999
6	PSI, LE-μSR	PAW	99999.dat	99999
7	PSI, LE-μSR	BRA	R9999_9.NEMU	R9999_9

Table 2.3: μSR data formats. Notes: USR: Occasionally unreadable due to bug in production code. TRI: newer TRIUMF format MUD not implemented yet. PAW: Only online histograms, use BRA in preference.

format	Logical name	Example assignment
USR	DATAUSRDIR	`$ assign user$disk:[tanya.for.fake99] dataUSRdir:`
PSI	DATAPSIDIR	`$ assign DATA$DISK:[PSI.PIM3.TD_MUSR.1997] dataPSIdir:`
TRI	DATATRIDIR	
PAW	DATAPAWDIR	
BRA	DATABRADIR	`$ assign DATA$DISK:[PSI.PIE3.SLOWMU98.RUNXI_YBC0700P_1] dataBRAdir:`

Table 2.4: μSR data formats logicals, typically defined in file SETUP_MAX.com.

2.5 μSR data formats available

Data formats available in MaxError are listed in Table 2.3. The include file DATA_FMT.INC defines which data formats are implemented in the source code MAX_FMT_99SEP.FOR. Unimplemented formats which have names already are RAL (for ISIS) and MUD (the current TRIUMF format). They should be implemented shortly.

Note: Ports of MaxError to operating systems other than VMS will probably *not* support PSI and RAL and possibly TRI data formats, due to file structure incompatibilities. Conversion of data either to either USR or MUD (preferred) formats will be necessary.

2.6 Logicals and files needed to run MaxError

Rather than require the user to type the data file name including the directory information, MaxError defines logicals internally to the program to allow the user to simply give the data file name without directory information. To allow data of more than one data type to be analysed in the same directory, both the logical name for the directory information and the file name for the phase information contain a three character code for the data type: USR, PSI, RAL, TRI, MUD, PAW and BRA.

Under **Open-VSM**, logicals DATA%%%DIR are defined for each data type, which indicate which directory the data is stored, see Table 2.4. These logical assignments are typically defined in a command file called SETUP_MAX.com.

MaxError requires information about the number of histograms, which bins to use as T0, T1, and T2 (either not provided in the data format or might need to be over-rided), phases to start with, etc. This information is contained in a "markers" file called VMARKER_%%%.DAT, where %%% stands for one of the three character codes for the data formats, *e.g* PSI. This file is required for

file	purpose
SETUP_MAX.com	Logical assignment of DATA%%DIR (e.g. DATABRADIR)
CMALPHA1::[TANYA.FOR.MAX99.SEP]MAX_MAIN_99SEP.EXE	program executable
VMARKER_%%.DAT (e.g. VMARKER_BRA.DAT)	Markers and phase info
data file (e.g. DATABRADIR:R1844_1.NEMU	data

Table 2.5: Files needed to run MaxError

file name	Note	Comment
R1844_1.ADB	a	summary of moments
R1844_1.MDB	b,a	$P(b)$, moments and progress info
SQUASH_CNTS_ZFT_R1844_1.DB	b	Effective orthogonal $a(t)$
SPEC01.DB	b	Data and ME $a(t)$ for histo. 1
SPEC03.DB	b	Data and ME $a(t)$ for histo. 3
Optional files		
FFT_0%_R1844_1_CNTS_ZFT.DB	b,c	FFT of experimental $a(t)$
FFT_0%_R1844_1_MECNTS_ZFT.DB	b,c	FFT of ME $a(t)$

Table 2.6: Example MaxError output files. Notes: **a:** DB fragment. **b:** Complete DB file. **c:** Percent sign (%) represents the number of the traditional FFT attempt; each attempt may be done with a different apodization. These optional files are for the purpose of comparing MaxError results with a traditional Fourier transform (see Sec. ??).

MaxError to run. Its purpose is to reduce the amount of typing the user does, since all the information could be entered interactively. See Secs. ?? for more information on its contents.

Table 2.5 summarizes the files needed to run MaxError.

2.7 Output Files

Example output files resulting from running MaxError with a data file r1844_1, which is data type `BRA`, using markers file VMARKER_BRA.DAT are listed in Table 2.6. The files are all in DB format, which is described briefly in a separate chapter (Sec. 5.2).

2.8 Limitations of MaxError program

MaxError can not be used for data taken in fields higher than a few kilogauss, due to the code being mostly single precision.

MaxError can not presently encompass negative as well as positive field ranges, although this should be a simple extension.

Presently, MaxError can only treat signals with a single phase. It cannot handle data such as fast-formation and slow formation muonium with different phases or a mixture of muonium and bare μ signals.

Maximum histogram lengths `maxchans` and maximum number of groups `maxgroups` may be changed by editing the include file MAXERRORS_GUTS.INC and recompiling all of the code. Note that in practice you will be limited by the memory of your computer.

Chapter 3

Brief Mathematical Introduction to ME

3.1 Introduction

The purpose of this chapter is to give a very brief and simple description of what maximum entropy Fourier analysis of μSR data is, and how it differs from fitting and from a traditional Fourier transform. It contains the basic equations, but there is an emphasis on an intuitive description of the approach over heavy mathematics, which is covered in Part ??. It is intended mainly as an introduction for the novice, setting up up the context for following chapter(s) ?? on using the program.

There are many introductions to the Maximum Entropy technique (references Ref. [?]). The first application to μSR was done by Rainford and Daniells[?]. Most introductions stress the technique's basis in logic and Baesian analysis in order to justify the choice and use of an entropy term. We present an alternative approach, where we assume the entropy term and discuss how it is actually used in a way that is analogous to χ^2 fitting, so as to be more intuitively approachable by experimentalists.

3.2 The μSR situation

Low energy surface muons are produced 100% polarized at a cyclotron's production target. A muon implanted in a sample precesses at a Lamor frequency proportional to the local field $\nu = \gamma_\mu/(2\pi)b$, where $\gamma_\mu = 2\pi 135.5$ MHz/T is the muon gyromagnetic ratio. After a time, of the order of the muon lifetime $\tau_\mu = 2.2$ μs, the muon decays into a positron and two neutrinos. The positron is emitted in a direction correlated with the direction that the muon spin was pointing at the time of its decay. The positron is detected by one of several scintillating counters arranged around the sample.

Each positron detected by a given detector j is histogrammed according to the time taken to decay. The raw μSR data spectra for a counter j looks like

$$N_j(t) = N_j(0)e^{-t/\tau}[1 + A_j(t)] + B_j \tag{3.1}$$

where τ is the muon lifetime of 2.2 μs, $N_j(0)$ is a normalization factor reflecting the total number of muons recorded and B_j is a time-independent background signal. $A_j(t)$ is the "asymmetry" of our signal, which is an oscillating signal with an average frequency $\bar{\nu} = (\gamma_\mu/2\pi)\bar{b}$ and initial phase ϕ_j. The initial asymmetry $A_j(0) \approx 0.2$ reflects both the degree to which the positrons convey the muon's polarization and the geometry of the detector. The decay of oscillating signal $A(t)$ reflects the depolarization of the muon spin due to the distribution of local fields $P(b)$ in the sample.

The spectra $N_j(T)$ are histogrammed according to $t_i = i\Delta t$, with $\Delta t \geq 1.25$ ns, typically. The errors are independent, with the error being the square root of the number of events in the bin:

$$\sigma_{N_j}(t_i) = \sqrt{N_j(t_i)}.$$

Therefore the errors associated with the asymmetry are

$$\sigma_{A_j}(t_i) = \frac{\sqrt{N_j(t_i)}}{N_j(0)e^{-t/\tau}} \approx \frac{e^{+t/2\tau}}{\sqrt{N_j(0)}}.$$

This feature of the errors increasing exponentially with time due to the muon lifetime is particular to μSR.

3.3 Fourier Transform of μSR data

If we have N detectors, each with a different phase ϕ_j relative to the initial muon spin direction, we can build a summary complex asymmetry (termed a "squashogram" by W.R. Dickson[?])

$$\begin{aligned} A(t) &= A_r(t) + iA_i(t) \qquad (3.2) \\ &= A_0 \sum_j^N \frac{A_j(t)}{A_j(0)} \exp(-i\phi_j) \\ &= A_0 \sum_j^N \left[\frac{A_j(t)}{A_j(0)} \cos(\phi_j)\right] - iA_0 \sum_j^N \left[\frac{A_j(t)}{A_j(0)} \sin(\phi_j)\right] \qquad (3.3) \end{aligned}$$

where A_0 is the average asymmetry of all the detectors, weighted by the number of events in each detector. The "real" and "imaginary" asymmetries $A_r(t)$ and $A_i(T)$ correspond to fictional detectors with phases of 0 and 90 degrees. Reconstructing the asymmetries for individual counters looks like

$$A_j(t) = A_j(0)\cos(\phi_j)A_r(t) + A_j(0)\sin(\phi_j)A_i(t). \qquad (3.4)$$

The Fourier transform of the distribution of Lamor frequencies $P(\nu)$ is the complex asymmetry

$$A_r(t) + iA_i(t) = \int_{-\nu_{max}}^{\nu_{max}} P(\nu)\exp(i2\pi\nu t)\, d\nu \qquad (3.5)$$

and the inverse

$$P(\nu) = \int_0^{t_2} [A_r(t) + iA_i(t)]\exp(-i2\pi\nu t)\, dt, \qquad (3.6)$$

where $t_2 \approx 4\tau_\mu$ is the typical time window of the experiment and $\nu_{max} = 1/(2\Delta t)$ the the maximum frequency resolution. Note that physically $P(\nu)$ is strictly real and positive. Because the frequency is simply $\nu = (\gamma_\mu/2\pi)b$, we will refer to $P(b)$ rather than $P(\nu)$ from now on.

Note that the area under $P(b)$ is the average asymmetry.

3.4 Use which technique to get $P(b)$?

In summary, what we really want our raw data to tell us is this $P(b)$, from one of three techniques:

1. Fitting the original sprectra (either Eq. 3.1 or Eq. 3.3) to a justifiable theoretical function $A_t(t)$ with a limited number of fit parameters, which implies a specific $P_t(b)$.

2. Extract the experimental $P(b)$ via a traditional descrete Fourier transform. Fourier transforms have difficulty with statistical noise, and are not designed to cope with a varying $\sigma(t)$. (See Sec. ??).

3. "Fit" the most *probable* $P(b)$ to the data, using a Maximum entropy technique. (The goal of this manual!)

If one knows which theoretical function is appropriate, then the first technique is appropriate and preferred. However, if one doesn't know which theory function to use (either the theory doesn't exist or is unwieldy to calculate) or if one wants a pretty picture to assist with an intuitive understanding of the data, one of the last two techniques is appropriate.

Which is better to use: Fourier transform or Maximum Entropy? Well, if one has clean, high statistics data, probably either one will be good enough**??**, if one doesn't care about error analysis (a feature of MaxError but not Fourier transforms). For low statistics data or difficult data, MaxErrors will be superior.

3.5 ME: Animal, mineral or vegetable?

What is "maximum entropy Fourier analysis of μSR data"? Animal, mineral or vegetable? Just joking. (The game of twenty questions always starts with the question "Animal, mineral or vegetable?")

Animal = Information Theory: A traditional explanation emphases the origins of maximum entropy in logic ("Bayesian theory"[3]) and in information theory, originating with Shannon[?]. Information theory just steals the concept and mathematics of entropy from physics and uses it to narrow down the most probable configuration of something (like $P(b)$ or a fuzzy photo) when you don't have enough information to get it by obvious brute force analysis methods. It is just like how it is much more likely that all the oxygen in a room will be approximately evenly distributed rather than all in one half. The difficulty of the traditional explanation is that is usually doesn't resonate with experimentalist' previous experiences with data analysis, which is why I don't use it to explain ME. Worse, it tends to encourage blind faith in the results, partly because ME *must* give the the most likely $P(b)$ and partly because most people can't easily follow the mathematical notation used, for instance in Ref. [3].

Mineral = Fourier transform: Since we are using "maximum entropy Fourier analysis of μSR data" to get $P(B)$, it is just like a traditional Fourier transform, only using a lot more CPU cycles. This description is a rather practical one, but the most superficial. In fact, a lot of the same choices are made with both MaxErrors and a Fourier transform: binning, time range, phases, and even apodization. Experience with traditional Fourier transforms really helps one use MaxError to best advantage and understand how you can make a real mess of it.

Vegetable = Fitting: Fourier transforms take the raw spectra and Fourier transform it to get a $P(b_i)$. MaxError goes the opposite way, by proposing a theoretical $P(b_i)$, with descrete binning, and then inverse Fourier transforming it to calculate a χ^2 with the raw spectra. Then a special search algorithm is used to adjust the values of $P(b_i)$ to minimize χ^2. Hey, MaxError is just a fitting routine! Ok, that makes sense.

Oops, I left out one point. There are too many fit parameters to get a good fit (the problem is overdetermined). Now at this point, things get a little unfamiliar. So we need to add something else: entropy. The χ^2 is constrained to the number of points in the raw spectra, which is what is expected when there is good agreement between theory and data. Then the entropy equation (using the values of $P(b_i)$) is maximized, giving us the most *probable* lineshape. Basically, the values of $P(b_i)$ are kept small unless they are needed (justified) to keep χ^2 to a reasonable value. What is a bit odd conceptually is that the lineshape $P(b_i)$ is its own theory function, and still manages to be (mostly) "model independent"!

Animal, mineral and vegetable: I think all three ways are perfectly good ways to think about *what* "maximum entropy Fourier analysis of μSR data" is. In fact, all three are necessary aspects of understanding it. Start from which ever aspect you find easiest.

3.6 How many parameters can you use in a fit?

Traditionally, experimental μSR asymmetries $A(t_i)$ $(1 \leq i < M)$ are fit using a theoretical fitting function $A_{j,0}G(t)$ with a finite number of parameters $n_p \ll MN$ by minimizing

$$\chi^2 = \sum_j^N \sum_i^M \frac{[A_j(t_i) - A_{j,0}G(t_i)]^2}{\sigma_j^2(t_i)}.$$

For example, in the case of random dipolar fields for a sample in an applied field, both $G(t)$ and its Fourier transform $g(b)$ would be gaussian with $\langle A_o \rangle g(b) = P(b)$. If the wrong theoretical function is chosen, $\chi^2/MN \gg 1$. If the theoretical function has too many fit parameters to be supported by the data, that is the problem is overdetermined, $\chi^2/MN < 1$ and the fit program has difficulty finding convergence. If the theoretical function is consistent with the data then $\chi^2/MN \approx 1$.

However, there are many cases where a suitable candidate for the theoretical function is unknown, not available or is unwieldy. An example is the asymmetry function due to the field distribution produced by the melted flux line lattice in a high T_c superconductor. This is also an example of where the fit function $G(T)$ is of secondary interest, compared to its Fourier transform $P(b)$.

Maximum Entropy as applied to μSR takes quite a different approach. It considers a "trial" field distribution histogram $P_{ME}(b_i)$, where the amplitude of each histogram bin i is in effect a fitting parameter. Using Eqs. 3.6 and 3.3, the "trial" distribution is then Fourier transformed to provide a *model-independent* theoretic fit function $A_{j,ME}(t)$ from which χ^2 can be calculated:

$$\chi^2 = \sum_j^N \sum_i^M \frac{[A_j(t_i) - A_{j,ME}(t_i)]^2}{\sigma_j^2(t_i)}. \tag{3.7}$$

In effect, each parameter $P_{ME}(b_i)$ provides a non-decaying cosine to $A_{j,ME}(t)$. Even though the range of fields b_i is limited in practice by the user to less than the maximum of $2M$ points, the number n_{ME} of "fit parameters" $P_{ME}(b_i)$ is still quite large, much larger than number of fit parameters used in a successful traditional fit by one to three orders of magnitude.

This is clearly an over-determined problem. Without anything more, minimizing χ^2 will give $\chi^2/MN < 1$ and $A_{j,ME}(t)$ will slavishly follow every little bit of statistical noise in each spectra j. If we impose the constraint that $\chi^2 = MN$

in order to try to find a consistent theory function, we find a infinite number of solutions lying in a multi-dimensional circle around minimum (see Fig. 3.1.) We need some criterion by which to select the most likely solution on that circle.

Sec. 3.6	Sec. 7.1	Sec. ??

3.7 How entropy helps

The maximum entropy method introduces an entropy term (analogous to entropy in statistical mechanics) which is rigorously justified from a information theory perspective

$$S = -\sum_i \frac{P(b_i)}{eP_d} \ln \frac{P(b_i)}{eP_d} \tag{3.8}$$

where the default value P_d is related to the statistical noise in the spectra (see Sec. 3.9.) Because the logarithm of a negative number is undefined, S requires that all $P(b_i) > 0$. The χ^2 term is added using a Lagrange multiplier α to form the constrained entropy S_c

$$S_c = \alpha S - \frac{1}{2l^2}\chi^2. \tag{3.9}$$

The Lagrange multiplier imposes the constraint that

$$\chi^2 \equiv l^2 MN, \tag{3.10}$$

where the practical "looseness" adjustment factor l is introduced, in part, to account for imperfect timing in the experimental electronics which is not reflected in the errors $\sigma(t)$.

Sec. 3.7	Sec. 7.2	Sec. 8.1

3.8 The Search Algorithm in Maximum Entropy

This is not necessary to understand the search algorthm in great detail at an introductory level. See Sec. ?? for a detailed explaination.

In short, the trial function $P_{ME}(b_i)$ is initialized with the value P_d for each i. With each iteration, S and χ^2 are calculated, and used to select the next set of values for the trial function $P_{ME}(b_i)$ [4, 1]. The fit is complete once S_c is maximized (Eq. 3.9) and $\chi^2 = l^2MN$ (Eq. 3.10). If, instead, a maximum number of iterations is reached, MaxError is considered to have failed its converge criterion.

Sec. 3.8	Sec. 7.3	Sec. 8.2

3.9 ME fiddle factors: the default level

The default level P_d has a very real statistical meaning, reflecting the statistical noise $\approx N_k^{1/2}$ of the N_k *discrete* muons per time bin k in the original spectra. Since the muon lifetime makes N_k vary a great deal, the important question is which bin/time do we use to estimate P_d? Consider the fractional error in the asymmetry at a characteristic time $t_c = 2\pi\tau_\mu$:

$$\frac{\sigma(t_c)}{A_o} \approx \frac{\sqrt{\exp(t_c/\tau_\mu)}}{\sqrt{\sum_j N_j(0)}}. = \frac{\exp(\pi)}{\sqrt{\sum_j N_j(0)}}. \tag{3.11}$$

If we assume that this fractional error is representative of the whole asymmetry $A(t)$, then the average fractional error for $P(b)$ is the same, to within a prefactor of the order of 1. If we average $P(b_i)$ over the range $i_{min} \leq i \leq i_{max}$, we get an average value of $A_o/(\delta b * (i_{max} - i_{min} + 1))$. Let us make the *a priori* assumption that the asymmetry will be about 0.2, then the average value of $P(b)$ per bin is $0.2/(\delta b * (i_{max} - i_{min} + 1)$ and our estimate of the noise in $P(b)$ is

$$P_{d,a\,priori} = \frac{e^\pi}{\sqrt{\sum_j N_j(0)}} \times \frac{0.2}{\delta b * (i_{max} - i_{min} + 1)} \tag{3.12}$$

See Sec. ?? for why you should use the value of P_d provided and more in depth discussion.

Sec. 3.9	Sec. 7.4	Sec. 8.3

3.10 ME Fiddle factors: σ looseness

The sigma looseness factor l in Eq. 3.10 is practically introduced by modifiying Eq. 3.7 by multiplying the spectra errors σ by l:

$$\chi^2 = \sum_j^N \sum_i^M \frac{[A_j(t_i) - A_{j,ME}(t_i)]^2}{l^2 \times \sigma_j^2(t_i)}. \tag{3.13}$$

When $l > 1$, what happens is that in effect the balance of power in the constrained entropy (Eq. 3.9) is changed to favor the entropy over the χ^2. Using values of $l > 1$ may visibly smooth out wiggles in $P(b)$. It is considered by many users as ME's "better" alternative to apodization in a traditional Fourier transform. However, it does introduce a systematic distortion of the lineshape. See Secs. ?? for advice on if, how, and when to use this looseness factor.

Sec. 3.10	Sec. 7.5	Sec. 8.4

3.11 Difference between ME and a Fourier Transform

Sec. 3.11	Sec. 7.6	Sec. 8.5

3.12 Basic choices: Binning and time window

Sec. 3.12	Sec. 7.7	Sec. 8.6

3.13 Fancy choices: Apodization and variable binning

Sec. 3.13	Sec. 7.8	Sec. 8.7

3.14 Output: Enhancement factor

Sec. 3.14	Sec. 7.9	Sec. 8.8

3.15 Output: Hey, what about error estimation?

Sec. 3.15	Sec. 7.10	Sec. 8.9

3.16 Output: The noise level and P_d

...

Then an estimate of the "independent" errors $e(b)$ for $P(b)$ is made using the covariance matrix (Sec. ??), and all points for which the fractional error $e(b)/P(b) < 1$ are deemed significant. For convenience, a new multiplicative factor m_d is chosen so that value of $m_d P_d$ is where the signal is equal than the noise, that is

$$P(B) > m_d P_d \quad \text{when} \quad e(b)/P(b) < 1.$$

While default level P_d is what we have to select, the noise level $m_d P_d$ is in fact the more important quantity for things such as moments derived from the lineshape, since we will only include those points with $e(b)/P(b) < 1$.

Sec. 3.16	Sec. 7.11	Sec. 8.10

Figure 3.1: S and χ^2 Space

Chapter 4

Concise Advice on Usage

4.1 Summary of advice: Program choices

4.1.1 Conventions used in I/O

Because there is a lot of output written to the screen, all requests for responses from the user start with `>>`, to make them more visible. The answer entered might appear at the end of the reqest, or, more commonly, on the next line. If a default value response is assumed, it is included in square brackets `[]` at the end of the end of the request. The default values are selected simply by hitting carriage return. We have attempted to make the program tolerant of entry mistakes, rather than crashing. If an error is detected, the request will be repeated.

Debugging statements (and some general information) from subroutines are usually indicated by the subroutine name followed by a colon at the beginning of the line. This is obviously useful to the programmer (T.M.R.) and should be included in bug reports (preferably cut-and-paste a section including the failure, rather than a vague description!)

4.1.2 The markers file VMARKER_%%%.DAT

See Fig. 4.1 for an example of a typical markers file VMARKER_%%%.DAT. Note that the text strings in single quotes are not required when the file is read in. They are added by the program MaxError when optionally overwriting the file to make the file more human readable.

The number of groups (which are just the number of histograms usually), the marker bins `it0 it1 it2`, and the phases are all required by the program to run properly and *cannot* be overrided interactively. It is generally assumed that the marker bins are probably the same for a whole set of data. If one wishes to change the time window for individual runs, it is preferable to specify it in μs (rather than via the marker bins) for each run by using parameter `t2` on the line indicated by `'t1, t2, tbin range (musec)'`. Note that the time range will be truncated to whichever (`t2` or `it2`) gives the shorter range; so make sure that `it2` is in fact the last good bin of the histogram.

The number of groups (which are just the histograms) are given by the first line and *must* agree with the number of histograms in the data file, except for `RAL` data (which will probably have a multiple of the number of groups). The number of `'it0 it1 it2 bins'` lines and the number of phases lines must match the number of groups.

When MaxError is run, one can optionally "remove" groups, that is ignore them. In that case, the values given corresponding to those removed groups are ignored. In this example, groups 2 and 4 will be removed, so the fact that `t1 = t2` and the phase is zero for those groups is of no consequence. That was done intentionally to remind ourselves that those groups where removed.

The last three lines provide default values which may be overrided interactively in the program. The third to last line covers the time range to fit (`t1` and `t2`), and the time binning `tbin`, all in μs. The second to last line gives a default time to start variable binning. In version 19990921of MaxError, only two values make sense: 0 μs for start variable binning at the beginning of the time range and a value larger than $t2$ (like 20 μs) which means don't do variable binning at all. The last line gives the default field range in Telsa to calculate $P(b)$ over;

```
         4    '# of groups (to check that it matches data)'
      1039       1059       10000   'it0 it1 it2 bins for group           1'
      1039       1059        1059   'it0 it1 it2 bins for group           2'
      1049       1069       10000   'it0 it1 it2 bins for group           3'
      1049       1069        1069   'it0 it1 it2 bins for group           4'
         0          0 'ib1, ib2 backgrds bins (ib1=ib2 fit)
         0          0 'ib1, ib2 backgrds bins (ib1=ib2 fit)
         0          0 'ib1, ib2 backgrds bins (ib1=ib2 fit)
         0          0 'ib1, ib2 backgrds bins (ib1=ib2 fit)
 -27.52730     'phase (degrees) for group           1'
 0.0000000E+00 'phase (degrees) for group           2'
 -165.6973     'phase (degrees) for group           3'
 0.0000000E+00 'phase (degrees) for group           4'
 0.0000000E+00   9.000000      5.1000003E-02 't1, t2, tbin range (musec)'
  20.00000          'Time start var binning (usec); If > T2 no var binning'
 2.0000001E-03  1.2000000E-02 'field range (T) (f1=f2 use whole range)'
```

Figure 4.1: A typical markers file VMARKER_BRA.DAT

the MaxError lineshape intensity is identically zero outside this range.

Any extra lines at the end are ignored.

4.1.3 Before starting MaxError

- *Before:* Create a file called VMARKER_%%%.DAT, where %%% are three characters representing the data type, e.g. BRA, (Sec. 2.6). See Sec. 4.1 for an example.

- *Before:* Inspect each histogram of the raw data to determine time zero bin `IT0` the first good bin `IT1` and the last good bin `IT2`. Put that information in the file VMARKER_%%%.DAT.

- *Before:* Try to have a run with a very narrow lineshape to determine phases. Maxerror it several times to get a best estimate for the phases. Don't fix the phases and remember to save the results in the file VMARKER_%%%.DAT at the end! Then fix the phases for similar runs which are wider. This is particularly necessary for asymmetric lineshapes.

- *Before:* Remember to initialize the logical DATA%%%DIR so the program can find the directory where the data is kept.

4.1.4 Choices made at the beginning of MaxError

The following advice is in the order that choices appear in the program, which is not necessarily the most user-friendly order, conceptually. This advice concerns the choices made to set up the conditions before the MaxError search engine starts.

- `>> Use time binning convolution [Y]:`
  Use time binning convolution — don't worry about what that means!

- `>> How many runs with the same conditions[1]?`
  Only do one run at a time.

- `>> Enter run name for 1th run [ ]`
  Enter the run number in the format required for its data type.

- `>> Chose an option [ 1:USR]:`
  Select the correct data format.

- `>> Enter background signal asymmetry [ 0.0000000E+00]`
  Don't subtract off a "background signal".

- `>> Remove group # (1- 4)?  [neg #=none] :`
  Only remove those histograms/groups which are empty of data (unusual).

- `>> Enter time range &bin size(us)[ 0.0E+00 9.0 5.1E-02]`

  - These default numbers correspond to the values given on the line with the text string `'t1, t2, tbin range (musec)'` in the markers file VMARKER_%%%.DAT (Sec. 4.1.2).
  - The time range used is entered as starting time `t1`, end of the time window `t2`, and time binning `tbin`. Enter zero for the start time `t1` unless you wish to over ride the IT0 bin in the file VMARKER_%%%.DAT. [Default value from file VMARKER_%%%.DAT.]
  - If not using apodization, select a time window `t2` which ends about where the signal either disappears in the scatter (low statistics data) or completely relaxes. The key is to avoid the outliers at long times. [Default value from file VMARKER_%%%.DAT.]
  - If apodizing the errors σ, use the whole time window available `t2` out to approximately the last good bin in data histograms.
    [Default value from file VMARKER_%%%.DAT.]
  - Select a time binning `tbin` which "looks" good. For low statistics data, make sure that the bins at the end of the time range have at least 10 counts per bin on average, to insure the assumption of Gaussian statistics. The higher the statistics, the less binning needed; there seems to be an ideal that we haven't figured out yet; however values which produce $N_j(0)$ (`exp_T0=`) between 800 and 1600 (perhaps 1000 is ideal) seem to work best. High statistics runs might not need any binning.
    [Default value from file VMARKER_%%%.DAT.]

- `>> Enter time to start exp(-T/tau) binning (us)[ 20.00000 ]`
  Don't use variable time binning (set time > the time window `T2`).
  [Default value from file VMARKER_%%%.DAT.]

- `>> Enter sigma apodization time (us):  [ 0.0000000E+00]`
  If apodizing the errors σ, use the whole time window available. Then select an apodization time which is approximately $\sqrt{2}$ *smaller* than the the time at which the signal either disappears in the scatter (low statistics data) or completely relaxes. Apodizing is preferable to selecting the time window, because it tends to reduce wiggles in the lineshape!

- `>> Enter sigma looseness factor [ 1.000000 ]`
  Use a σ looseness factor $l = 1$ *unless*

- You know that there is a problem with the histogramming memory, in which case all data should probably be analysed using the same value.
- If MaxError doesn't converge and the data is high statistics and sensitive to slight imperfections in the data (perhaps due to histogramming memory problems).
- You have a particularly "naughty" low statistics run.
- With low statistics data, even very large values of $l > 1.1$ will barely make a difference if you haven't found an optimal time binning. Keep looking.
- Use apodization instead of $l > 1$ to deal with wiggles in broad lineshapes, preferably.

Sec. 3.10, p. 17	Sec. 4.1.4, p. 25	Sec. 7.5, p. 61	Sec. 8.4, p. 64

- `>> Enter test criterion [ 1.0000000E-03]`
  The smaller the test criterion value, the harder it is for MaxError to converge. The suggested value for the test criterion value should work fine for typical runs. You may be able to reduce it. You may need to increase it if you use very small time binning or very wide field ranges.

- `>> Do you want to keep the phases fixed [Y]:`
  Fix the phases for runs with *wide* lineshapes, using the phases determined from a narrow run. (Phases stored in file VMARKER_%%%.DAT and not entered interactively!) This is particularly necessary for asymmetric lineshapes.

- `>> Enter power for number of points:  [ 9]:`
  - If the signal is not fully relaxed by the end of the time window, use about a factor of 4 more FFT points than suggested, so that the linewidth will be accurate. That is, the suggested power +2.
  - If the signal is fully relaxed by the end of the time window, use either the number of FFT points suggested (for speed) or higher (to give extra interpolated points for prettiness). Either way won't make much difference with the errors of the moments. Either use the suggested power or power + 1 or power + 2.

- `>> Enter min and max fields (in Telsa):  [ 2.0E-03, 1.20E-02]`
  Use a field range which includes all of your expected lineshape, plus a bit extra on each side. Plotting $P(b)$ log scale will help you double check that there is a reasonable region on either side where the signal is below the noise level and has settled in around the default level P_d. When determining phases on narrow lineshapes, make the range symmetric about the average field and at least 0.02 T wide or wide enough that the phase has stabilized.
  [Default value from file VMARKER_%%%.DAT.]

- `>> Enter default level twiddle factor [ 1.000000 ]`
  Use the default level suggested, by using the twiddle factor of 1.0.

- `>> Fit using sqrt(theory) as the errors ?  [y]`
  Use MaxError fit ("theory") rather than "data" for spectra errors σ. Otherwise you will need a larger σ looseness factor l to get *exactly* equivalent results and the exact value of l will be more dependent on statistical fluctuations, the linewidth, etc., which is extra complications which we don't need!

4.1.5 Choices made at the end of MaxError

This advice concerns the choices after the MaxError search engine has finished.

- `>> Save these phases and time range [N]:`
  This saves many choices to the file VMARKER_%%%.DAT. Remember to use it when you allowed the phases to float and you want to save the results.

- `>> Use final fitting enhancement factor 1.056365 ?  [y]`
  Yes. This helps correct an artificial loss of asymmetry due to how maximum entropy works with μSR data. See Sec. ??.

- `>> Write time data for each group (y/n/h for only one hist)?  [y]`
  This option writes out all (y) or one (h) or none (n) of the files SPEC0%.DB. Note: Group means histogram (except for `RAL` data).

- `>> Bin by what factor?  [1]`
  This is an *additional* time binning factor, to make the output SPEC0%.DB more visible. Just use 1, unless you have high statistics data that isn't already binned. If there isn't some binning, this could be a very large file!

- `>> MAIN: Perhaps now output FFT of data asymmetry?`
  `>> Output a traditional FFT for comparison?  [N]: y`
  This is to create a file FFT_0%_R1844_1_CNTS_ZFT.DB (using the data) which contains a tradition Fourier transform, so you can compare it to the MaxError results. Additional Fourier transforms (with different apodizations) will have incrementally larger numbers where the "%" appears.
  Note: the file Squash_cnts_zft_r1844_1.db is created automatically containing the asymmetry squashograms (which is just the asymmetry function for orthogonal counters, $\phi = 0$, 90 Degrees, Eq. 3.3), which are used to make the traditional Fourier transforms. If there isn't some binning, this could be a very large file!

- `>> Field range truncated to ME"s range?  [Y]:`
  If you don't truncate, FFT_0%_R1844_1_CNTS_ZFT.DB will be a very large file, filled mostly with noise.

- `>> Enter time range for apodization tapod [blank=stop]`
  This is the apodization of the asymmetry (not the σ errors!) to be used in the traditional Fourier transform.

- `>> MAIN: Perhaps now output FFT of ME asymmetry?`
  `>> Output a traditional FFT for comparison?  [N]:`
  Usually, it is a waste of time to do a traditional Fourier transform of the MaxError's theoretical asymmetry. This creates a file FFT_0%_R1844_1_MECNTS_ZFT.DB.

- `>> Again? [y]`
  To run MaxError again or exit the program.

4.2 Low statistics data: May the noise be with you

... May the noise be with you... Intended as a variation of "May the force be with you", from Star Wars. Sort of tongue-in-cheek "Zen Buddhism and the art of MaxEnting."

Maybe move this section to the general advice chapter.

Low statistics data is obviously more difficult to handle. The following discussion should give you some hints as to what to try to improve your output.

- Low stats data is more sensitive to statistical fluctuation, obviously. Therefore, for all other things being equal, one run might look "nicer" than other just due to dumb chance.
  ... May the noise be with you...

- With considerable time binning (say 0.06 - 0.1 μs per bin), you have relative few time bins left in the histograms. The file `*.MDB` file says the number is `Used_pts` binned time points. You don't expect to determine the total χ^2 to better than $\pm$ (`Used_pts`)$^{1/2}$, this is the source of the error associated with `Total_ChiSq`. Fewer time points, expecially in comparison to the number of field bins, suggests that you should expect more fluctuations.
  ... May the noise be with you...

- *In general:* You can't get something for nothing: You can't determine χ^2 better by reducing the binning arbitarily! You *must* bin hard enough to insure that there are enough events per bin (10+) to satisfy the assumption of Gaussian statistics and you must bin hard enough to get optimal values for $N_j(0)$ (`exp_t0`) in the range of 800 and 1600 events per bin. Note that for a full time range $T_2 = 10$ μs and `exp_t0=` 1000, you have $N_j(0)\exp(-T_2/\tau_m u) = 10.6$ events per bin at the the end of the time range. The two requirements appear to approximately the same! Binning any harder seems to give MaxError *more* difficulty; it's as if MaxError likes to have the right amount of statistical noise.
  ... May the noise be with you...

- *In general:* Any particular binning will produce a typical number ($\pm$ statistical fluctuations of course) of data outliers which are more than two standard deviations from " true" lineshape (approximately the MaxError results).
  $N_{2\mathrm{StdDev}} = (1.0 - 0.9545)$`Used_pts`
  $N_{3\mathrm{StdDev}} = (1.0 - 0.9973)$`Used_pts`
  High statistics data binned "optimally" will have lots of outliers, which will tend to destructively interfere in the MaxError lineshape. Low statistics data will have only a few outliers and it will be by chance if they interact "nicely" or "badly" in the MaxError lineshape.
  ... May the noise be with you...

- Given that you are in the "optimal" binning range, you have some latitude in selecting the time binning. *Keep trying different time binnings until you find one that gives nice output!* The best choice will be quite similar to several runner-ups. This sounds a bit like cheating, in that you are "choosing" the answer that suits you best. In some sense, by your choice, you are applying an extra *a priori* of single peaked data, smoothness, etc. On the other hand, other criterion such as the ease of convergence, a large enhancement factor, the number of outliers being reasonable, the fact that certain noise features can be related to certain outliers, etc., could be used by the computer program rather than you to select the best choice. In other words, the process could easily be made automatic and appear more impartial.
  ... May the noise be with you...

- *In general:* The reduced constrained entropy S_c (`RedConEntrpy`) has a maximum value of -0.5 (corresponding to being entirely dominated by the χ^2 contribution). Increasingly more negative values correspond to the MaxError solution being increasingly more dominated by the entropy, in which case the lineshape should look smoother. (This can be achieved artificially by increasing the σ looseness factor l above a value of 1.) If you perversely *decrease* the looseness factor below 1, you will find that you continue to get convergence as long as $S_c < -0.5$. Any smaller value of l and MaxError will fail to converge. The *minimum* looseness factor that satisfies convergence is typically about 0.8. By turning "off" the use of the entropy term, this just reflects the fact that when just fitting to a large number of non-relaxing cosines (the $P(b_i)$), the fitting problem is grossly over determined. Which is why we need to use the entropy term.

- With low statistics data, you don't expect to determine the total χ^2 to better than $\pm$ $(\texttt{Used\_pts})^{1/2}$. Therefore, there will be a large fluctuation in the value of the reduced entropy `RedConEntrpy` run to run, all other things being equal. Runs which by chance have small ideal values of $\chi^2 = \texttt{Used\_pts} - \sqrt{\texttt{Used\_pts}}$ will be easy to fit and report large enhancements `EnhanceUsed` > 1 and `RedConEntrpy` $\ll -0.5$. Runs which by chance have large ideal values of $\chi^2 = \texttt{Used\_pts} + \sqrt{\texttt{Used\_pts}}$ will be hard to fit and report small enhancements `EnhanceUsed` $>\approx 1$ and `RedConEntrpy=` near -0.5. Those runs with large ideal values of χ^2 may be so hard to get to converge that it is justified to use $l > 1$, which will only work if the right time binning has been found!
  ... May the noise be with you...

- Don't forget to make good choices for the time range and apodization, to get rid of unnecessary outliers!

... May the noise be with you...

4.3 Summary of advice: Is the result OK?

4.3.1 Numerical indications that results are good

Is the convergence OK? If things are going well,

- The resulting test of convergence `Test_result=` will be at least an order of magnitude smaller than the test criterion `Test_criterion=`.
- The error associated with the Lagrange multiplier `LagrangeMult=` will several orders of magnitude smaller than the value.
- The reduced χ^2 `RedcdTotChiSq=` will much closer to the value of one than its error would suggest.
- The enhancement `EnhanceUsed=` will be several percent larger than 1.

4.3.2 Graphical indications that results are good

- Plotting up $P(b)$ shows all noise spikes below the noise level `Noise=`.
- Plotting up $P(b)$ in log scale shows symmetric behavior below the noise level `Noise=` on either side of the lineshape. The values below the noise level `Noise=` cluster around the default value P_d `default_level=` and generally are not clipped at $0.001P_d$ (which would indicate points which would like to be negative).
- The plot of $\nabla\nabla\chi^2$ shows little or no wiggles corresponding to the time window's sinc function. (A plateau or glitches due to the choice of field range may appear with low field data. Negative values at large relative fields seem to be related to this.) $\nabla\nabla\chi^2$ should lie between the Lorentzian behavior corresponding to $2\pi\tau_\mu$ (the case of no apodization) (dashed line) and the amount of apodization (dashed-dot line).

4.3.3 Numerical indications that results are poor

If things are going poorly:

- The resulting test of convergence `Test_result=` will be close in value to the test criterion `Test_criterion=`, or larger if the test fails.
- The error associated with the Lagrange multiplier `LagrangeMult=` will be large. (This is not reliable when a point in $P(b)$ is very close in value to P_d; this gives a large error automatically.)
- The further the reduced χ^2 `RedcdTotChiSq` is from the value of one (especially compared to its error), the worse the MaxErrrors results, and may be the symptom that MaxError defines as failure to converge. In very bad cases, `RedcdTotChiSq` > 1.05 and MaxError thrashes while trying to reduce it. Try checking the phases, checking the field range, changing the time binning (***... May the noise be with you...***) or increase the apodization or increase the looseness.
- If the asymmetry (`AvAsym` or `AvAsymWeighted`) is much too small (or MaxError completely thrashes with huge χ^2), your initial phases may be completely wrong or your field range doesn't cover your signal.
- If the asymmetry (`AvAsym` or `AvAsymWeighted`) is slightly too small after enhancement, it may be due to time binning too hard or that the lineshape is particularly wide. In the case of wide lineshapes, this is usually unavoidable.

- If the asymmetry is too large (`AvAsym` or `AvAsymWeighted`), there might be too much noise in $P(b)$; look at the plot!

If it is converging too easily for poor choices, some of the above signs will be bad, especially enhancement factors `EnhanceUsed` close to 1 and reduced χ^2 `RedcdTotChiSq` far from 1.

4.3.4 Graphical indications that results are poor

- Plotting up $P(b)$ shows a very narrow lineshape with only one or two points on the peak and perhaps a larger linewidth than expected. Redo again with more FFT points.

- Plotting up $P(b)$ shows some or many noise spikes above the noise level `Noise`; usually due to either too long a time range (lots of big outliers) or too little apodization (lots of big outliers) or poor binning (low statistics data often needs just the right binning to avoid certain outliers). In dire cases, increase the looseness as a last resort. Look at numerical indications for hints (especially the convergence criterion `RedcdTotChiSq` and `Test_result`.)

- Plotting up $P(b)$ shows some or many noise spikes above the noise level `Noise` which seem to have a certain periodicity; These either

  - are due to specific outliers (try changing the binning of low stats data).
  - if they are always the same periodicity, are due to poor estimates of the random "flat" background and the number of events at T0; try a longer time range, or variable time binning using the whole time range, or estimating the backgrounds from negative time bins rather than fitting.
  - if they are always the same periodicity, are due to the random background not being flat at all. Get a new beamline or try above suggestions to get a better "average estimate".

- The convergence criterion `RedcdTotChiSq` > 1 and `EnhanceUsed` very close to one are associated with *many* narrow noise spikes above the noise level `Noise`; This is is a sign that MaxError has failed to converge and needs help.

- Plotting up $P(b)$ in log scale shows a lopsided behavior below the noise level `Noise`: one side high and the other low relative to the default level P_d `default_level` (and perhaps clipped to $0.001P_d$). This suggests *either* the phases are incorrect *or*, in the case of naturally asymmetric lineshapes, you need a wider field range.

- The plot of $\nabla\nabla\chi^2$ shows wiggles due to too short a time window. (A plateau or glitches due to the choice of field range may appear with low field data.) If $\nabla\nabla\chi^2$ is very wide due to severe apodization, narrow features in $P(b)$ might be broadened somewhat.

4.4 To do

Write equivalent chapter on the advice that ME prints out as it goes along.

Chapter 5

Graphically Viewing Output

file name	Note	Comment
R1844_1.ADB	a	summary of moments
R1844_1.MDB	b,a	$P(b)$, moments and progress info
SQUASH_CNTS_ZFT_R1844_1.DB	b	Effective orthogonal $a(t)$
SPEC01.DB	b	Data and ME $a(t)$ for histo. 1
SPEC03.DB	b	Data and ME $a(t)$ for histo. 3
Optional files		
FFT_0%_R1844_1_CNTS_ZFT.DB	b,c	FFT of experimental $a(t)$
FFT_0%_R1844_1_MECNTS_ZFT.DB	b,c	FFT of ME $a(t)$

Table 5.1: Example MaxError output files. Notes: **a:** DB fragment. **b:** Complete DB file. **c:** Percent sign (%) represents the number of the traditional FFT attempt; each attempt may be done with a different apodization. These optional files are for the purpose of comparing MaxError results with a traditional Fourier transform (see Sec. ??).

5.1 Output Files

Example output files resulting from running MaxError with a data file r1844_1, which is data type `BRA`, using markers file VMARKER_BRA.DAT are listed in Table 5.1. The files are all in DB format, which is described briefly below (Sec. 5.2).

5.2 Viewing using DB

Output files from MaxError are formated in Jess Brewer's DB format, which is an ASCII-based mini data base program. The MaxError output files ending with *.MDB have an additional category called "MaxHes". DB with this additional category is available in Alpha-VMS ports of DB originating from Tanya Riseman, using the PGPLOT package for the graphics. For a current version of this DB, see the PSI μSR web site

http://WWW1.psi.ch/~reid/thirdparty.html.

The DB structure is, in short, ntuples of numbers, one row of the ntuple consisting of a fixed number of numerical entries per line. Each number entry arranged as value, positive error, negative error. An optional text string ended by an equal sign "=" may be used to label each number entry; this makes DB very useful (and unusual) as entries can be looked at, easily identified, and changed in an ordinary ASCII editor. The formatting is flexible, and a backslash may be used a line continuation. See the file

DKA200:[TANYA.TEX.PROGRAMS_DOC]DB_MANUAL.TEX

for more information about using DB.

The file *.MDB contains the main information; that is the MaxError lineshape $P(B)$. Additional information about calculated moments, all values used used in the calculation, the intermediate results, etc., are contained as DB-style fragments in the `ABSTRACT` section of the file.

You may start to generate a new DB file which summarizes a series of MaxError files *.MDB by a VMS DCL command like

```
$ sea r%%%%_1.mdb "AvAsym=", "mean_t=", "binfieldRes_t=", -
"fieldRes_predicted=", "run=", "sqrtM2_T=" /out=listing.db.
```

(Or use `grep` in Unix.) Then edit it to make it a proper DB file.

Common values of interest, especially the moments of $P(b)$ are pre-summarized in the DB fragment file *.ADB, for your convenience. Create a summary DB file by including a series of those.

See Secs. ?? for detailed descriptions of the contents of the files listed in Table 5.1.

5.2.1 An example of using DB

```
$ show def
  DKA200:[TANYA.TEX.MAXENT.MAN.EXAMPLES.19990921]
DB>@SPEC01.DB
 title
 spec01.db  SLO r1844_1   : 700 nm YBCO, online 90.       108.19991005  12:37:07
 labels
  Time (musec)
  a(t) Theory
  a(t) data
  Difference a(t)
  ratio*(datum(i,j)-guess(i,j))/sigma(i,j)

  Comments
   binfactor=          1
  group =            1
  Use variable binning = F

 data t th data diff test
\-e
DB> plot t data
Point type?
DB>plot t th
Point type? 2
DB>
DOPLOT>
DOPLOT>x 0, 5
DOPLOT>y -.3, .3
DOPLOT>
DOPLOT>
DB> exit
$
```

5.2.2 Assessment

DB is a quick way to graphically view the contents of DB files, and it is easy to create new summary DB files using a simple ASCII editor. DB is fine for superimposing several graphs on the same axes but cannot presently do multiple graphs on the same page (like PHYSICA) without resorting to merging them using a program called EDGR and the amount of extra text information one can tack on (via CAPTION) is very limited.

5.3 Viewing using spreadsheet programs via DB_SPREAD.EXE

If you wish to plot results from DB files using a spreadsheet program, from any DB file you can create a file *.SPR which contains the information in columns format using the program DB_SPREAD.EXE. The program DB_SPREAD.EXE is limited to output of 132 characters per line, so you will be limited in the number of columns you can do at a time.

If each ntuple is written in one physical line with no optional entry labels and no continuation lines, then, in principle, one can read the DB file directly into a spreadsheet. This is the case for *.MDB files, except that you have to skip about 500 lines to get to where the data starts.

5.3.1 An example of using DB_SPREAD

This is an example of how to use `DB_SPREAD` where we chose to converts only columns labeled `t`, `th` and `data` but not `diff` or `test` when creating the file SPEC01.SPR from SPEC01.DB.

```
$ show def
  DKA200:[TANYA.TEX.MAXENT.MAN.EXAMPLES.19990921]
$ DB_spread
Default file name =

File name to fit (C/R for previous)?
SPEC01.DB
db_read: keytitle_=TITLE
 spec01.db  SLO r1844_1   : 700 nm YBCO, online 90.       108.19991005  12:37:07
db_read: keylabels_=LABELS
db_read: keycomment_=COMMENTS
db_read: keydata_=DATA
 Enter output spreadsheet file name
*.SPR file name, at present   (C/R to keep):
SPEC01.spr

Using *.SPR file:SPEC01.spr
SPEC01.spr
 spec01.db  SLO r1844_1   : 700 nm YBCO, online 90.       108.19991005  12:37:
Time (musec)
Use variable t                ? [y]
 a(t) Theory
Use variable th               ? [y]
y
a(t) data
Use variable data             ? [y]
y
Difference a(t)
Use variable diff             ? [y]
```

```
n
ratio*(datum(i,j)-guess(i,j))/sigma(i,j)
Use variable test                ? [y]
n
Label string length          12
t,th,data,e
Line length needed =           53
$
```

The columns of data in the output file start at line 4, and the file looks like:

```
SPEC01.spr
  spec01.db  SLO r1844_1   : 700 nm YBCO, online 90.       108.19991005  12:37:
 t,th,data,e
   0.510000E-01, 0.189200    , 0.179113    , 0.720E-02
   0.102000    , 0.133084    , 0.135231    , 0.712E-02
   0.153000    , 0.671291E-01, 0.587775E-01, 0.700E-02
...
    8.87400    , 0.696963E-03,-0.554499E-01, 0.106E+04
    8.92500    , 0.744115E-03,-0.838427E-01, 0.120E+04
```

5.3.2 Assessment

Spreadsheet files have the drawback that the lineshape information tends to get separated from the information about how the lineshape was made; that is, all the extra information in the `ABSTRACT` section of the *.MDB file gets discarded. I would recommend thinking about how you will track of that information (via the original *.MDB files, printouts, logbook, etc...) to assist you when you try to reproduce or improve a preliminary analysis. Spreadsheets have the obvious advantage of being the tool of *your* choice (not mine) so you can optimize the graphics output for publication, cut-and-paste into documents, etc.

5.4 Viewing using Physica (and other math programs)

Rather than writing computer code to interpret and display DB files, one can write macros in a math program to search for text strings and read in the data contained.

We have written macros for use with the program Physica, see its web site

http://www.triumf.ca/physica/html/homepage.html.

Physica is supported on VAX/VMS, Alpha/VMS, Digital Unix (both 3.x and 4.0), DECstation/ULTRIX, SGI/IRIX, SUNOS, Solaris, HP-UX, IBM-AIX, and PC Linux.

An example Physica macro is

DKA200:[TANYA.TEX.MAXENT.MAN.EXAMPLES.19990921]MDB.PCM

which prints or plots out *everything* that we can think that might be useful under most circumstances. (In practice, we use a highlighter pen to mark off information that is of specific interest, e.g. time range when we do a series varying the time range.) It calls calls sub-macros EXTRACT_NUM.PCM, FIND_STRING.PCM, GGCHI2_CURVES.PCM, PLACE_TXT.PCM and SPEC.PCM. It used in PHYSICA by

```
PHYSICA: @mdb.pcm 'R1844_1.MDB' 0 2
```

where `'R1844_1.MDB'` is the MaxError output file (note the backwards and forward single quotes), the second parameter is 1 for logscale and 0 for linear, and the third parameter is the number of removed groups (2 out of 4 groups here). In addition to the file R1844_1.MDB, it reads in some or all of the files SPEC0%.DB and the markers file VMARKER_BRA.DAT. This macro

- Plots up the $P(B)$ lineshape at the top of the page,
- Reports many interesting resulting numerical values from MaxError, including moments, in the middle.
- Reports the T0, T1, and T2 bins defined in VMARKER_BRA.DAT.
- Using SPEC0%.DB, plots up one or two of the asymmetry spectra with "apodized" errors bars and the ME theory curve, so one can check that the binning, time range and apodization is suitable, in the lower left hand corner.
- Plots up $\nabla\nabla\chi^2$ in the lower right hand corner, so one can see how the correlations between bins in $P(b_i)$ are affected by the choices made.

Secs. ?? will discuss the interpretation of this information in greater detail.

5.4.1 An example of using macros in PHYSICA

```
$ show def
  DKA200:[TANYA.TEX.MAXENT.MAN.EXAMPLES.19990921]
$ physica
PHYSICA: device post a4
PHYSICA: @mdb.pcm 'R1844_1.MDB' 0 2
PHYSICA:hard
Device= postscript
P - Print on a queue
S - Save into a file
A - Auxiliary port output
command >> s
filename (default= postscript.plt) >> R1844_1.ps
 Hardcopy file: r1844_1.ps
PHYSICA: clear
PHYSICA: @mdb.pcm 'R1844_1.MDB' 1 2
PHYSICA:
```

5.4.2 Assessment

The drawback to this approach is that macro(s) tend to be very dependent on the version of MaxError used (particularly when things get rearranged or renamed) and on minor details like the number of groups/histograms used. Therefore, the macros will be less flexible than using DB directly or spreadsheets. The first strength of using macros is that you can extract information directly from text strings in the `COMMENTS` and `ABSTRACT` sections of the DB files, which DB ignores. The second strength lies in the ability to arrange multiple plots with that text information adjacent. It is not clear how much programming effort one saves over using FORTRAN and PGPLOT, once the plots get fancy.

Chapter 6

Advice that MaxError outputs

6.1 Introduction

Very rough draft! No time to edit!

This chapter shows some of the output advice that MaxError automatically produces when running. This chapter's aim is to explain it, highlight the important information, and point out which advice is probably nonsense at the moment.

6.2 Version number info

```
$ run MAX_MAIN_99SEP.EXE
MaxErr version
CMALPHA1::[TANYA.FOR.MAX99.SEP]MAX_MAIN_99SEP.exe
       1033
   19990921
```

Just to make sure you are using the version you want to.

6.3 Group removal

```
>> Remove group # (1- 4)? [neg #=none] :2
>> Remove group # (1- 4)? [neg #=none] :4
>> Remove group # (1- 4)? [neg #=none] :
Reading file vmarker_BRA.dat
          1 Fitting Background bins.
          2 Fitting Background bins. REMOVED GROUP
          3 Fitting Background bins.
          4 Fitting Background bins. REMOVED GROUP
Phase(           1)  =   -27.52730
Phase(           2) =   0.0000000E+00 REMOVED GROUP
Phase(           3)  =   -165.6973
Phase(           4) =   0.0000000E+00 REMOVED GROUP
```

When you optionally remove groups (2 and 4 in this case), the program reminds you of which information from the markers file VMARKER_%%%.DAT is going to be ignored. In the output file *.MDB those ignored items are indicated as follows: `FlatBkgrdFit= 0.0000000E+00,,,` and `PhaseNotUsed`. Other output values for the removed groups will be zero as a consequence, such as `GoodBinsPerGrp`, `EvntsPerGroup`, `ChiSqPerGroup`, `RedChiPerGrp`, `AsymFree`, `RelAmp`, and `exp_T0`.

6.4 Time binning adjustment

```
>> Enter time range &bin size(us)[  0.0000000E+00   9.000000      5.1000003E-02]

read_markers: Modifying time res from   1.0000000E-03 to   5.1000003E-02
read_markers: using binning =          51
```

The program reports that the time binning is being increased from the original resolution to something *approximately* equal to what you requested. The value used is always be an odd integer multiple of the original resolution in the data file. Odd multiples insure that the "time binning convolution" is done correctly.

6.5 Apodization of σ errors

```
 Time range used is    8.925000     us
 Apod of errors is according to exp(+0.5*(t/t_apod))
 The following are preset values for apodization of errors
 which           name   (t_apod/t_max)   t_apod_us     looseness_correction
            0 None      9.9999999E+14  8.9250000E+15   1.000000
            1 Weak      0.6263080       5.589799      0.7342517
            2 Medium    0.4769893       4.257130      0.6470073
            3 Strong    0.4010303       3.579195      0.5936351
            4 Teds      0.5000000       4.462500      0.6619936

>> Enter sigma apodization time (us): [  0.0000000E+00]
2.0
 Using apod custom  =   2.000000     us.  lfac=  0.4424219
```

The looseness correction `lfac` < 1 is a factor used internally to ensure that results with and without apodization are comparable. It arises from the fact that by apodizing, you are effectively *reducing* the time window of the data, so the effective number of data points is being reduced. Since we are constraining the total χ^2 to the number of data points, we need this factor $lfac$ to fix this bookkeeping problem.

The ideal apodization time is determined by the statistics of your data and the linewidth of the signal. The suggested values corresponding to "weak", "medium" and "strong" are Gaussian approximations to three "ideal" apodization functions recommended by So-and-so??. They are included for the convenience of people familiar with TRIUMF's μSR FFT program.

6.6 Number of FFT points

```
       175 of          176 # of points in time spectrum are used.
 Ideal # of points in time spectrum is fft pts/2
      e.g. default is 16384 fft pts for 8192 time pts
 NOTE: No increase in accuracy of moments by oversampling!
 # of pts in FFT is 2**N. Acceptable range         128 <= 2**N <=      131072
 # of pts in FFT is 2**N. Choose N           7 <= N <=          17
 Default res is:   2.8256018E-04 Tesla for points:         512 = 2^           9
>> Enter power for number of points: [           9]:
11
 Using # of points: 2**          11 =         2048
 Tesla per channel=  7.0640046E-05
```

This is where you get to select your field resolution in the lineshape $P(b_i)$. The suggested value is based on the time window `T2` and time binning `tbin` that you selected, with the restriction that the time range is zero padded out to a power of 2 for speed of Fourier transform. For the default value of the power suggested, the resolution will be 2.8256018E-04 Tesla. If you select a power such as 11 instead of the suggested value of 9, the resolution used will be $(2.8256018E-04)/2^{11-9}$ T, which is a factor of four smaller.

Note that using higher than the value for the power will merely interpolate the points in $P(b_i)$, roughly speaking. The only case where one *needs* a higher power is when the lineshape's asymmetry function $a(t)$ is not fully relaxed by the end of the time range.

6.7 Field range selection

```
Using values from markers.dat for b range
However, maximum B range allowed is   0.0000000E+00  7.2300084E-02 (Telsa)
>> Enter min and max fields (in Telsa): [  2.0000001E-03,   1.2000000E-02]

# of Frequency channels:         143
Using field range   1.9779212E-03  1.2008808E-02
b1, b2, n1, n2  1.9779212E-03  1.2008808E-02          29         171
```

Here you are told both the maximum field range (the Nyquest frequency) possible for $P(b_i)$ based on the time binning selected `tbin` and what the default values are from the markers file `VMARKER_%%%.DAT`. You may enter any range within the maximum range. Afterwards, field bin size is determined from the zero-padded time range (from the FFT power you gave in Sec. 6.6). Then the exact starting and ending fields are calculated (`Using field range 1.9779212E-03 1.2008808E-02`) and the number of points in that range.

6.8 Is assumption of Gaussian statistics satisfied?

```
Will estimate default level from statistical noise in
raw data, assuming a typical asymmetry of   0.2000000
 *** For this time binning =  5.1000003E-02 [us]
 *** Assumption of Gaussian errors breaks down after
 *** time = tau*log(min_expt0/10) ~    11.90346     [us]
...
 **THINK** This assumption is satisfied.
          You have four options:
 *** 1. Leave as is.
 *** 2. If signal is NOT fully relaxed, maybe increase time range
        from    8.976001     to as high as   11.90346     [us]
 *** 3. Decrease time bin from   5.1000003E-02 to as low as  1.3455235E-02 [us]
 *** 4. If there are wiggles in the ME lineshape, try increasing
       the time bin and hope that binning of random noise might smooth it.
Time range to where signal/noise =1 is    12.94047     assuming asy=
 0.2000000
This signal-to-noise time range gives B-resolution of   9.0744667E-05
```

We assume that Gaussian statistics probably holds as long as the average number of events per bin is 10 or higher. This bit of advice tell you at what time range this assumption breaks down. For this example, it breaks down after `11.90346` μs, which is longer than, but not a great deal longer than, the time range of `8.976` μs, being used.

It also estimates the time `12.94047` μs where the signal-to-noise probably drops below 1.0 in the asymmetry $a(t)$ per histogram, assuming a total asymmetry of 0.2 and no relaxation.

You obviously want to insure that the time range you use is equal to or longer longer than both these critical time ranges. You may want to start over again (by aborting MaxError with control C) with either a different time range or a different binning if they are not satisfied. At present there is no graceful way to change your mind at this point. These conditions should probably be relaxed if you are using apodization.

6.9 Modified parameters outside ME

```
MODBAK: Items modified this cycle marked by *:
  exp(To)      +exp_chng    back         +back_chng   amp          phi
 *  2298.2729 *    35.0735 *    12.7686 *    -1.9849     1.001998   -27.527300
 *  2267.6692 *    13.4942 *    10.6457 *    -0.9859     0.998002  -165.697296

MODBAK: CHi^2 per histogram, raw and normalized
  1678.304       4.767910
  1575.898       4.476982
MODBAK: Total Chi^2, raw and normalized
  3254.202       4.622446
MODAMP: Items modified this cycle marked by *:
  exp(To)      +exp_chng    back         +back_chng   amp          phi
    2298.2729       0.0000      12.7686        0.0000 *   0.989087   -27.527300
    2267.6692       0.0000      10.6457        0.0000 *   1.010913  -165.697296
```

Between each call to the ME search engine, the phases ϕ_j and relative signal amplitudes are refined (subroutine **MODAMP** for fixed phases or **MODAB**) and the random "flat" backgrounds B_j and $N_j(t = 0)$ **exp_TO** are refined (subroutine **MOD_EXPTO** for fixed backgrounds or **MODBAK** for fitted backgrounds). The values of ϕ_j and the relative amplitudes are given in the columns `phi` and `amp` with an asterisk if they have been modified. The value of B_j and $N_j(t = 0)$ are given in the columns `exp(To)` and `back` and their change since the last cycle in the columns `+exp_chng` and `+back_chng` with an asterisk if they have been modified.

The χ^2 values reported are not important, they will settled down to a normalized value that is larger than 1, but which tracks the values of χ^2 in the ME search engine.

6.10 Reports from the ME search engine

```
 cycle number            1          Enhancement=    1.016179
It Test        Entropy    ChiSqTargt ChiSqNow   Alpha      +-Dev.Alph SConstraind
 1 0.1713E+00 -.7904E+05 0.7040E+03 0.6403E+03 0.4135E-02 0.1303E-03 -.6470E+03
 2 0.3438E-01 -.7412E+05 0.7040E+03 0.7035E+03 0.1362E-01 0.1533E-01 -.1362E+04
 3 0.1816E-01 -.7399E+05 0.7040E+03 0.7036E+03 0.1462E-01 0.1648E-01 -.1434E+04
 4 0.2041E-01 -.7392E+05 0.7040E+03 0.7036E+03 0.1406E-01 0.1632E-01 -.1391E+04
 5 0.1237E-01 -.7385E+05 0.7040E+03 0.7041E+03 0.1470E-01 0.1721E-01 -.1437E+04
...
19 0.1830E-02 -.7373E+05 0.7040E+03 0.7035E+03 0.1470E-01 0.5965E-02 -.1436E+04
```

```
20 0.1332E-02 -.7371E+05 0.7040E+03 0.7041E+03 0.1471E-01 0.3767E-02 -.1437E+04
   0.1095E-02 -.7372E+05 0.7040E+03 0.7038E+03 0.1471E-01 0.2582E-02 -.1437E+04
```

The cycle number is the number of times the ME search engine has alternated with refinements of the values of the phases ϕ_j, and relative signal amplitudes, the random "flat" backgrounds B_j and $N_j(t = 0)$ `exp_T0`.

Each line is a separate call, with up to 20 iterations. Each time, the search suggests new values for $P(b_i)$. The columns indicate

- **It** Number of the iteration. The last unnumbered row contains the final values.
- **Test** The results of the convergence test, reported as `Test_result`. Compare these values to the test criterion that you specified (reported as `Test_criterion`), Values that are at least an order of magnitude lower are very good.
- **Entropy** This is the unnormalized entropy, which is not very informative. On the first iteration, it may have a positive value, which is nonsense, but harmless.
- **ChiSqTargt** This is the target χ^2, which is the number of data points in the time domain.
- **ChiSqNow** This is the current total χ^2. When it is approximately equal to the target χ^2, the ME search is going well!
- **Alpha** This is the Lagrange multiplier for the constrained entropy (Eq. ??). The value is very dependent on many things, like the default value, etc., so it by itself is not very informative. It is *not* used by the ME search engine, but is calculated after the fact as an average over all the alphas needed for each bin of $P(b_i)$
- **+-Dev.Alph** This is the estimated error in alpha, based on the scatter of values when it is averaged. If it is several orders of magnitude smaller than `alpha`, the ME search engine is doing well.
  Warning: larger values of `Test_criterion` will tend to produce sloppier convergences and bigger errors for `alpha`.
- **SConstraind** This is the constrained entropy, involving the entropy, alpha, and the total χ^2. Until it is normalized, it isn't a very useful indicator.

(In a future version, more of these items will be normalized, which will make them more useful.)

6.11 Success?

```
 getting there in one iteration, so stopping ...
# of runs=           1. Phases fixed= T
>> Save these phases and time range [N]:
```

Apparently successful convergence.

WARNING: Sometimes ME exceeds a set number of iterations and exits without complaint, yet there was a failure to converge. This needs to be fixed.

6.12 When are errors calculated?

```
About to calculate MaxEnt errors...
calc_maxerrors: Inverting matrix         143
```

If the number of field bins **num_b_pts** is larger than the value of **maxerrors** defined in the include file MAXERRORS_GUTS.INC, usually 500, then no covariance matrix is calculated and no errors are estimated. This will probably cause difficulties with PHYSICA macros. In any case, MaxError will be very slow if you exceed this practical number!

The larger **num_b_pts** is, the longer it takes to invert the matrix. You will notice a pause in the program at this point.

6.13 Peaks

```
The            1 largest peaks are in order:
 FIELD           FREQ          AMPLITUDE
 5.9337639E-03  0.8042279       160.6705
```

MaxErrors reports all the peaks (up to a maximum of 50) which are above the noise level, to the screen. It only reports the two largest peaks to the output file *.MDB.

6.14 Outliers

```
 ~< Means that the noise period is larger than
    than B-range window  1.0030886E-02, so the phase will be distorted.
 *  Asterisk indicates maximum values per histogram.
    Maximum Standard Deviations and Impulses suggest
 which outliers produce wiggles in ME.
    Impulse is estimated amplitude of oscillation in FFT.
        Impulse = asy_outliers(iout,j)/(res)
 Expect    8.007997     outliers > 2 std devs out of          176 pts per group
 Expect   0.4751949     outliers > 3 std devs out of          176 pts per group
Hist  1 Found      8.4454 or     12 while expecting      3.5429 or      8.0080 outliers
SUM  T_bin Time_[us]  B_period_[T]     B_bins   StdDev     dAsym       Impulse
 1      52  2.6010     0.2836683E-02     40.16   2.0676 *   0.08258 *   1.61915
 1      42  2.0910     0.3528557E-02     49.95   2.0673     0.05410     1.06080
 1      32  1.5810     0.4666801E-02     66.06   2.5239     0.04323     0.84770
 1      29  1.4280     0.5166815E-02     73.14   2.4359     0.04000     0.78440
 1      28  1.3770     0.5358178E-02     75.85 * 4.2723     0.06953     1.36326
 1      25  1.2240     0.6027950E-02     85.33   3.4617     0.05461     1.07080
 1      17  0.8160     0.9041926E-02    128.00   2.4009     0.02951     0.57858
 1      16  0.7650     0.9644721E-02    136.53   2.0413     0.02385     0.46768
 1      12  0.5610 ~< 0.1315189E-01    186.18   2.1533     0.02081     0.40811
 1      11  0.5100 ~< 0.1446708E-01    204.80   2.3321     0.02188     0.42895
 1      10  0.4590 ~< 0.1607453E-01    227.56   3.9855     0.03672     0.72001
 1       9  0.4080 ~< 0.1808385E-01    256.00   2.7723     0.02540     0.49809
...
...
Hist  3 Found     10.2133 or     13 while expecting      3.5429 or      8.0080 outliers
SUM  T_bin Time_[us]  B_period_[T]     B_bins   StdDev     dAsym       Impulse
```

```
3      39  1.9380    0.3807127E-02      53.89   2.0658      0.05075     0.99511
3      35  1.7340    0.4255024E-02      60.24   3.8474 *    0.08178 *   1.60359
3      31  1.5300    0.4822360E-02      68.27   2.5156      0.04417     0.86608
3      30  1.4790    0.4988648E-02      70.62   3.0544      0.05092     0.99849
3      24  1.1730    0.6290035E-02      89.04   2.3510      0.03082     0.60439
3      16  0.7650    0.9644721E-02     136.53   3.5610      0.04603     0.90263
3      15  0.7140 ~< 0.1033363E-01     146.29   3.2635      0.04195     0.82256
3      13  0.6120 ~< 0.1205590E-01     170.67   2.5394      0.03173     0.62216
3      12  0.5610 ~< 0.1315189E-01     186.18   3.8616      0.04712     0.92384
3       8  0.3570 ~< 0.2066726E-01     292.57 * 4.6780      0.04867     0.95436
3       7  0.3060 ~< 0.2411180E-01     341.33   2.6090      0.02577     0.50537
3       6  0.2550 ~< 0.2893416E-01     409.60   2.0962      0.01964     0.38509
3       3  0.1020 ~< 0.7233541E-01    1024.00   2.3829      0.01960     0.38430
...
...
```

This section reports the number of data "outliers" which lie more than 2 standard deviations from the MaxError curve, which is assumed to be close to the ideal lineshape. It tells you how many are expected for a normal treatment (second `8.0080`), plus modified estimates based on the apodization (first `3.5429`):
`while expecting 3.5429 or 8.0080 outliers`.
As long as the number of outliers is approximately equal to the number predicted, to within the expected scatter, the lineshape is probably generally OK, other than maybe the usual difficulty with noisiness. If the number is much, much larger, there are serious problems with basic things like the phases.

These outliers will have the *most* effect on traditional Fourier transforms of the data, done the same way as MaxError. The larger **`dAsym`** is, the worse oscillatory noise in the traditional Fourier transform. However, despite being somewhat discounted by MaxError's natural noise suppression tendencies, the MaxError result will still be slightly kinked by each of the outliers. Therefore, the information given about the outliers will be useful in identifying the worse offending outliers but will be very poor at estimating the size of the effect observed (e.g. the size of noise wiggles on the lineshape.) Each histogram/group is listed separately.

The columns are labeled:

- **SUM** or **ALL:** Short for summary or all outliers listed. The number below indicated which group.
- **T_bin** The number of time bin where the outlier occurs, relative to the zero time bin.
- **Time_[us]** The time at which the outlier occurs.
- **B_period_[T]** Since a delta function in the time domain will produce a pure cosine signal in the frequency domain, this columns gives the expected period of the cosine produced by the outlier.
- **B_bins** The expected period expressed in field bins, so you can count off the observed period of wiggles and identify which outlier is producing it.

- **StdDev** How many standard deviations out the outlier is; the largest standard deviation is indicated by an asterisk. The largest deviations might produce the most kinking of the MaxError results.

- **dAsym** The asymmetry difference between the MaxError results and the data; the largest difference is indicated by an asterisk. The largest differences will produce the biggest noise in a traditional Fourier transform.

- **Impulse** A failed attempt to translate `dAsym` into the size of the effect expected in $P(b_i)$.

With high statistics data and optimal binning, there will be *many* outliers, which will tend to interfere with each other, producing a fuzzy low level noise in the MaxError lineshape, assuming that the time range and apodization have be chosen appropriately. Persistent wiggles at with a field periodicity corresponding exactly to the time window `T2` indicate problems with the determination of the random flat backgrounds B_j and $N_j(t=0)$, assuming reasonable choices for the time window and/or apodization.

With low statistics data, there are only a few outliers, especially with larger time binnings. Therefore, the noise in $P(b_i)$ will tend to be more oscillatory than for high statistics data.

With low statistics data, certain outliers will tend to appear for many different time binnings, producing the same noisy spikes or the same splitting of the lineshape's peak. To distinguish a real (but unexpected) feature of the data from a stubborn outlier, search for a time binning which doesn't have that particular outlier in it, by visually inspecting the files SPEC0%.DAT. The true features should remain and the noise, if not improved, should at least have moved. Remember to look at both graphical and numerical indicators of good convergence to help you judge which time binning appears to be the most representative of the "true" lineshape.

6.15 Time range OK?

```
...
 *** Smallest recommended time range based on second moment:
 ***   T_range >=   8.363091     +-    2.098408     [us]
 *** Largest recommended time range based on limit of Gaussian Stats:
 ***   T_range <   11.91801     [us]
...
```

The first part reminds you very roughly what time range (or apodization) you might want to use based on the width of the lineshape. If the linewidth is very noisy, the estimate of the second moment will be too large, and this will give you nonsense.

The second part just reminds you again the consequences of the time binning on the time window possible, based on maintaining the assumption of Gaussian statistics.

6.16 Time bin suggestions

```
 *** Smallest time bin size:  (limit Gaussian Stats)
```

```
 ***   t_bin>  1.3366411E-02 [us]
 *** "LastSigPt" time bin tolerates a 5% loss
 ***   at the highest significant field =    1.0030886E-02 [T]
 *** "Range" time bin tolerates a 5% loss
 ***   at the highest field in B-range =   1.2008808E-02 [T]
 *** FracAsy ~ (1-exp(-Std_dev)) =sin(pi 135.5 B_high bin)/(pi 135.5 B_high bin)
 Std_devs    FractionAsym  TBin(LastSigPt) TBin(Range) [us]
         2  0.8646647      0.2109808      0.1762311
         3  0.9502130      0.1279663      0.1068895      ***Recommended***
         4  0.9816844      7.7615470E-02  6.4831749E-02
         5  0.9932621      4.7076195E-02  3.9322473E-02
         6  0.9975212      2.8553303E-02  2.3850409E-02
         0  0.9921107      5.1000003E-02  0.0000000E+00 Current for LastSigPt
         0  0.9887044      0.0000000E+00  5.1000003E-02 Current for Range
 *** Use   0.1279663     > tbin >    0.1068895     [us]
```

This bit of advice is intended to suggest to you a range of time binnings (**tbin**) which might be acceptable, if you decide to run MaxError over again.

The first line recommends what the smallest time binning which will satisfy the assumption of Gaussian statistics for the time range used.

If one bins too hard, the basic oscillatory behavior of the asymmetry $a(t)$ starts getting washed out, and the average asymmetry will start to drop. If the lineshape is at an average field much higher than its linewidth, the whole lineshape will be affected approximately the same, with just the average asymmetry dropping. For lineshapes in low field, the higher field portions will be much more affected than the lower field portions, thus possibly producing a *distortion* as well as an overall drop in asymmetry.

We assume that if you decide that you are willing to tolerate up to a certain loss (say 5 % or 2 %) in the highest field bins of $P(b_i)$, you can then read off the table what range of time binnings will be consistent with that. For instance, if you are willing to tolerate up to 2 % loss, you might consider time binnings between 7.7615470E-02 μs and 6.4831749E-02 μs (row number 4). The larger time binning corresponds to the highest field point of $P(b_i)$ which lies above the noise level. Depending on how good the current lineshape is, this may or may not be useful. The smaller time binning corresponds to the highest field in the field range, which is probably a conservative estimate.

The last two lines of the table tell you approximately what asymmetry loss you are suffering with the current choice of time binning.

Note: the time binning convolution option in the program was intended to circumvent this problem (Sec. ??). It does seem to maintain the phases better (good!) when the time binning is adjusted, but mysteriously does not recover this lost asymmetry. It should not be depended on to recover asymmetry lost due to over-vigorous time binning. Because over-vigorous binning (that is, much larger than needed to just barely satisfy the assumption of Gaussian statistics) doesn't suit MaxError anyway, it is probably better to be conservative by keeping the asymmetry losses low.

6.17 General reminder about how ME was done

```
!>> CMALPHA1::[TANYA.FOR.MAX99.SEP]MAX_MAIN_99SEP.exe
!>> MaxEnt Version Number        1033
```

```
!>> MaxEnt Version Date      19990921
!>> Data Format BRA
!>> Using time binning convolution:  T
!>> Using variable time binning:  F
!>> Using apod of errors:  T t_apod=   2.000000     lfrac=  0.4424219
!>> MODBAK=Guess_W_Enh_                  MODAMP=Guess_Sigm2_
```

This information is just a summary of the version of MaxError and generally how it was done (logicals T=true and F=false).

6.18 Recount of data input

```
!>> Input Data Use:
 run=     r1844_1             ,,, \
 temp=       30.00000    ,,, \
 ngroups=           4,,, \
 ngroupsUsed=           2,,, \
 deadTime_us=   0.0000000E+00,,, \
 deadTime_us=   0.0000000E+00,,, \
 deadTime_us=   0.0000000E+00,,, \
 deadTime_us=   0.0000000E+00,,, \
 backbins_ib1_ib2=           0,,,            0,,, \
 backbins_ib1_ib2=           0,,,            0,,, \
 backbins_ib1_ib2=           0,,,            0,,, \
 backbins_ib1_ib2=           0,,,            0,,, \
 data_T_Res_us=       1.0000000E-03,,, \
 TRange_TRes_us=      0.0000000E+00,,,     9.000000    ,,,   5.1000003E-02,,, \
 VBin_start_us=       20.00000    ,,, \
 Lastgood_orig=            10000,,, \
 T0_T1_T2_orig=        1039,,,        1059,,,        10000,,, \
 T0_T1_T2_orig=        1039,,,        1059,,,         1059,,, \
 T0_T1_T2_orig=        1049,,,        1069,,,        10000,,, \
 T0_T1_T2_orig=        1049,,,        1069,,,         1069,,, \
 Binning=          51,,, \
Lastgood_bin=          176,,, \
 T0_T1_T2_bin=           1,,,           2,,,          176,,, \
 T0_T1_T2_bin=           1,,,           2,,,          176,,, \
 T0_T1_T2_bin=           1,,,           2,,,          176,,, \
 T0_T1_T2_bin=           1,,,           2,,,          176,,, \
 used_pts=         350,,, \
 GoodBinsPerGrp=          175,,, \
 GoodBinsPerGrp=            0,,, \
 GoodBinsPerGrp=          175,,, \
 GoodBinsPerGrp=            0,,, \
 Lastgood_VarBin=          176,,, \
 T0_T1_T2_VarBin=           1,,,           2,,,          176,,, \
 T0_T1_T2_VarBin=           1,,,           2,,,          176,,, \
 T0_T1_T2_VarBin=           1,,,           2,,,          176,,, \
 T0_T1_T2_VarBin=           1,,,           2,,,          176,,, \
 Fitted_events=        193662,,, \
 total_events=         200927,,, \
 EvntsPerGroup=        101885,,, \
 EvntsPerGroup=             0,,, \
```

```
 EvntsPerGroup=        99042,,, \
 EvntsPerGroup=            0,,, \
```

This is a recount of the run number, temperature, binning and number of events.

6.19 Recount of how MaxError was performed

```
!>> How MaxEnt Done:
 VarTimeBin=   F,,, \
 TimeBinConvol=   T,,, \
 UserDeconvol=   F,,, \
 SigamApod=   T,,, \
 looseFitSigma=        1.000000    ,,, \
 ratio=  0.7050951    ,,, \
 facFake=  0.3119495    ,,, \
 lapod_factor=  0.4424219    ,,, \
Apod_us_custom  =   2.000000    ,,, \
 def_twiddle=       1.000000    ,,, \
 default_level=    0.1059752    ,,, \
 intn_def_level=   1.7092185E-02,,, \
 mult_def=          14.51721    ,,, \
 Noise=             1.538463    ,,, \
 NormNumEvents=   1.000000    ,,, \
 FFTpwr=              11,,, \
 FFTpoints=         2048,,, \
 freqRes_Mhz=  9.5741414E-03,,, \
 fieldRes_T=   7.0640046E-05,,, \
 b_first_T=    1.9779212E-03,,, \
 b_last_T=     1.2008808E-02,,, \
 num_b_pts=            143,,, \
 backSubtract_asy=     0.0000000E+00,,, \
 backSubtract_B=     0.0000000E+00,,, \
 backSubtract_sigma=     0.0000000E+00,,, \
 backSubtract_pwr=      2.000000    ,,, \
```

This section contains a lot of information about the choices made in how MaxError was done: time binning convolution, apodization, σ looseness factor used, binning resolution of the field range, and field range used. It also reports some important numbers calculated by the program: the default level `default_level` and the noise level `Noise`.

6.20 Recount of internal indicators

```
!>> Entropy and ChiSq From MaxEnt:
 MaxEntCycles=              5,,, \
 MaxEntIter=                1,,, \
 Test\_criterion=    1.0000000E-03,,, \
 Test\_result=      4.0531505E-04,,, \
 effUsedPts=              704,,, \
 LagrangeMult=      1.4818776E-02,  7.7878911E-05,, \
 Entropy=           -73673.76    ,,, \
```

```
ConstrEntrpy=      -1443.990    ,    1.000000    ,, \
RedConEntrpy=      -2.051122    ,   1.4204546E-03,, \
Total_ChiSq=        704.4702    ,   26.53300    ,, \
RedcdTotChiSq=      1.000668    ,  3.7688918E-02,, \
ChiSqPerGroup=     155.2445    ,,, \
ChiSqPerGroup=    0.0000000E+00,,, \
ChiSqPerGroup=     196.9915    ,,, \
ChiSqPerGroup=    0.0000000E+00,,, \
RedChiPerGrp=     0.8820712    ,,, \
RedChiPerGrp=     0.0000000E+00,,, \
RedChiPerGrp=      1.119270    ,,, \
RedChiPerGrp=     0.0000000E+00,,, \
NGood_RHS=  0.0000000E+00,   0.0000000E+00,, \
NGood_LHS=  0.0000000E+00,   0.0000000E+00,, \
Diff_NGood=  0.0000000E+00,   0.0000000E+00,, \
```

This section recounts the success (or failure) of MaxError to converge. It records χ^2, the entropy, the lagrange multiplier, the constrained entropy, and their errors when available. It reports the number of cycles and iterations that MaxError used (the lower the better) and the result of the convergence test.

The last three lines all return zeros because the code was disabled because it was very slow (due to matrix inversions) and the results puzzling; perhaps the equation does not apply.

6.21 Recount of histogram parameters

```
!>> Info From MaxEnt:
!>> Flat backgrounds fitted using +times. MODBAK using Guess_W_Enh_
!>> Phases fixed. Amps fitted. MODAMP using Guess_Sigm2_
 EnhanceUsed=     1.056365    ,,, \
 AvAsymFree=    0.2666622    ,,, \
 AsymFree=      0.2611028    ,,, \
 AsymFree=      0.0000000E+00,,, \
 AsymFree=      0.2722216    ,,, \
 AsymFree=      0.0000000E+00,,, \
 RelAmp=        0.9854285    ,,, \
 RelAmp=        0.0000000E+00,,, \
 RelAmp=         1.014572    ,,, \
 RelAmp=        0.0000000E+00,,, \
 exp\_T0=         2297.828    ,,, \
 exp\_T0=        0.0000000E+00,,, \
 exp\_T0=         2269.155    ,,, \
 exp\_T0=        0.0000000E+00,,, \
 FlatBkgrdFit=   12.77069    ,,, \
 FlatBkgrdFit=  0.0000000E+00,,, \
 FlatBkgrdFit=   10.67028    ,,, \
 FlatBkgrdFit=  0.0000000E+00,,, \
 PhaseFixed=    -27.52730    ,,, \
 PhaseNotUsed=    0.0000000E+00,,, \
 PhaseFixed=    -165.6973    ,,, \
 PhaseNotUsed=    0.0000000E+00,,, \
```

This section recounts the resulting values for familiar parameters describing the data: phase, random backgrounds B_j, number of events at zero time $N_j(t = 0)$, the relative amplitude and the asymmetry per histogram.

6.22 Recount of advice based on statistics of run

```
BinfieldRes_T=   7.0640046E-05,,, \
 mult_trange=   6.283185    ,,, \
 fieldRes_T_2piTau=  8.5065483E-05,,, \
 Trange_2piTau=   13.80441    ,,, \
 fieldRes_T_STN=  9.2421993E-05,,, \
 Trange_STN=   12.70562    ,,, \
 fieldRes_GausStats_max=  9.8529745E-05,,, \
 Trange_GausStats_max=   11.91801    ,,, \
 fieldRes_M2=  8.8223501E-04,   2.7839240E-05,, \
 Trange_M2_min=   8.363091    ,   0.2636380    ,, \
fieldRes_predicted=  1.1312729E-04,,, \
 FracAsyLossFromBin=  1.1295617E-02,,, \
 Tbin_1PerLoss_Suggest=   4.7904555E-02,,, \
 Tbin_5PerLoss_Suggest=   0.1071179    ,,, \
```

This section speculated on what the field resolution of the lineshape might be and makes some suggestions as to what time binning and time window you might want to use. I think that most of the suggested field resolutions are nonsense, except `fieldRes_predicted`. The only really important item is `fieldRes_predicted`, which I believe is the field resolution based on the time binning and the statistics. It seems to be approximately correct with very narrow lineshapes, where it should be more critical. (With wide lineshapes, the natural linewidth swamps the effect of the intrinsic MaxError resolution.) It does need to be tested on fake data. Treatments like apodization of the errors σ and shorter time windows will broaden the lineshape slightly, but *much* smaller than the comparable amount in a traditional Fourier transform. I don't know yet how to estimate the increase in the field resolution due to those effects.

6.23 Recount of calculated moments

```
!>>
!>> Calculations Based on MaxEnt:
 Areadb_to_asym=      1.000000    ,,, \
 IntnAreadbToAsym=    6.200211    ,,, \
 AvAsym=            0.2666622    ,,, \
 AvAsymWeighted=     0.2650071    ,    2.2830397E-03,, \
NptsMoms=      78.00000    ,   0.0000000E+00,, \
Area_P_X_db=  4.0461030E-02,   3.4857227E-04,, \
Mean_T=       6.3386816E-03,   1.8625457E-05,, \
sqrtM2_T=     8.8223495E-04,   2.7783952E-05,, \
crtM3_T=      9.1671362E-04,   5.3626522E-05,, \
Alpha_skew=    1.039081    ,   3.3393744E-02,, \
Skewness=      1.121885    ,   0.1081646    ,, \
```

This is the summary of the moments and the skewness, which is what many users are interested in primarily. Estimated errors are based on the covariance matrix. The errors will appear as zero if there are too many points in the field range over which $P(B_i)$ is calculated.

Chapter 7

Advice (and Speculation) on MaxError Usage

7.1 How many parameters can you use in a fit?

7.1.1 Overdetermined theory functions

With ordinary fitting, if the theoretical function has too many fit parameters to be supported by the data, that is the problem is overdetermined, $\chi^2/MN < 1$ and the fit program has difficulty finding convergence. Practically speaking, when there are too many fit parameters, you notice that the fitting program complains about having trouble converging, the results of a series of fits (say, for different temperature) will have a more scatter than is reasonable, and, if you have a good fitting program, the errors will be huge.

7.1.2 What should you compare χ^2 to?

If there are N_p fit parameters, actually we should consider the value of $\chi^2/(MN-N_p)$, but for $MN \gg N_p$, this is an unimportant refinement.

Sec. 3.6	Sec. 7.1	Sec. ??

7.2 How entropy helps

7.2.1 Noise suppression due to S

The most important feature of the entropy term (Eq. 3.8) is that $\ln P(b_i)$ is *only* defined when $P(b_i) > 0$. Therefore, the entropy term automatically does some noise reduction in the regions of $P(b_i)$ where there is no signal. In contrast, noise in a traditional Fourier transform has both positive and negative values of $P(b_i)$ in regions where there is no signal. This results from the fact that for a given outlier in the raw spectra acts like a delta function, and its Fourier transform is a pure cosine.[?]

7.2.2 Speculation: Alternatives to the entropy term

If you don't care about the information theory justifications, you could use

$$S = -\sum_i \ln \frac{P(b_i)}{eP_d} \tag{7.1}$$

or even

$$S = -\sum_i \sqrt{\frac{P(b_i)}{eP_d}} \tag{7.2}$$

instead of the standard definition of the entropy (Eq. 3.8) and still get similar behavior. The possibilities are endless and, for the most part, pointless. Note that Eq. 3.8 and these alternatives all share in common the fact that the entropy term makes no distinction about the relative ordering of the field bins. Any correlations present between field bins i are only found *during* the search algorithm and expressed via the correlation matrix (or in $\nabla\nabla\chi^2$).

Some possibilities which *might* be worth pursuing are alternatives which impose smoothness criterion for adjacent bins in $P(b_i)$[?, ?]. This would discourage ME lineshapes which have inappropriate wiggles in them, particularly those which alternate high-low. This might be appropriate for experiments in which one expects a single smooth lineshape, but might complicate the analysis of experiments with multiple well defined peaks (muonium, different knight shifts in adjacent materials, different diamagnetic shifts in adjacent superconducting and normal metals).

I personally believe that *understanding* the sources of the wiggles and using the current ME better is the preferable approach, since any fancy variations will likely further limit the uses of the program, cloud the source of the difficulties, and introduce a whole new range of "fiddle" factors we need to come to an understanding of how to use and how they might distort the data. I also think that people can dream up variations about two to three orders of magnitude faster than they can understand the ramifications of the variations.

Other possibilities are including convoluting with ad-hoc lineswidths and/or something related to the muon lifetime, reflecting the correlations that it introduces. (Even worse in my opinion.)

Sec. 3.7	Sec. 7.2	Sec. 8.1

7.3 The Search Algorithm in Maximum Entropy

Sec. 3.8	Sec. 7.3	Sec. 8.2

7.4 ME fiddle factors: the default level

7.4.1 The *previous* conventional wisdom about P_d

The default level P_d is described in the literature *to date* as your default "model" for what $P(b_i)$ for each i might be, in the absence of actually measuring anything. So, depending on your purpose, you might make it some tiny number (close to zero), something related to the average asymmetry spread out over all the field bins i or even the full asymmetry for each of the field bins i. This is as close as you usually get in maximum entropy techniques to something that is like a theoretical model. It is called an "a priori", meaning an assumption that you incorporate into your analysis ahead of time. But the conventional wisdom about P_d makes it sounds pretty arbritrary.

7.4.2 The (probably) correct value of P_d

Well, we think think that that the conventional wisdom for P_d is all **WRONG** in Sec. 7.4.1!

The default level P_d (Eq. 3.12) has a very real statistical meaning, reflecting the statistical noise $\approx N_k^{1/2}$ of the N_k *discrete* muons per time bin k in the original spectra. We won't call it a noise level (because it turns out that it is smaller than that, see Sec. 3.16). It is more like a measure how wrong $P(b_i)$ will be if the measured number of events in a single time bin k of the raw spectra is "wrong" by about the expected error $\approx N_k^{1/2}$ compared to the expected value of N_k (assuming that all the other bins have about the correct value). So we could call P_d the "one bin in the time spectra is wrong by $N^{1/2}$" level or "wrong by $N^{1/2}$" level for short.

Our choice of a characteristic time $t_c = 2\pi\tau_\mu$ for estimating P_d in Eq. 3.12, seems a bit arbritrary. We noted that in Eq. ?? in Sec. ?? that t_c characterizes the correlations between different bins of $P(b_i)$, as seen in the Lorentzian behavior of $\nabla\nabla\chi^2$, reflecting the influence of the muon lifetime on the MaxError lineshape. By an intuitive leap of faith, we tried $t_c = 2\pi\tau_\mu$ here for P_d, since it seems to be a characteristic time scale for μSR. The value of $t_c \approx 13.8\,\mu$s is only a little bit longer than the typical experimental time window used. However, for the current argument, assuming another characteristic time such as the muon lifetime will work as well. In practice, we tried both, and found somewhat better convergence with the value t_c for single choice of data file. (Not very rigorous!) But, which time scale works better may be rather dependent on the random nature of the particular run chosen due to limited statistics (some might prefer slightly higher than the ideal P_d and other prefer slightly lower P_d.) Moreover, as we will argue below in Sec. 7.4.4, the response of MaxError to changes in P_d is so weak, that practically speaking we need only get the right order of magnitude.

We could of course refine Eq. 3.12 by putting in the experimental asymmetry instead of the typical value of 0.2, but it won't make much difference.

Our argument for P_d is a bit crude. It may be that someone will be able to put it on firmer grounds, and come up with a better equation. However, as long as it gives values within an order of magnitude of the current values, it won't make much difference practically.

7.4.3 How is P_d affected by time binning?

Because the normalization factors $N_j(0)$ in spectra (Eq. 3.1) are relative to the binning δt, this treatment (Eq. 3.12) indicates that additional binning $n\Delta t$ of the time spectra can be useful in reducing the noise in $P(b)$ by a factor $\sqrt{n}$ in low statistics data.

7.4.4 What happens if you fiddle P_d?

As one can see from Eq. 3.8, changing P_d only changes the entropy logrythmically slowly, which is *very* slow. So the exact value of P_d is not important, only the order of magnitude. The program maxError allows you to multiply P_d as calculated from Eq. 3.12 by your choice of multiplicative factor. It's only allowed so that you can increase or decrease P_d by a couple of orders of magnitude and observe that the program then has trouble converging.

In practice, just use the value provided.

7.4.5 How to test the value of P_d

You can lower the convergence test criterion (see Sec. ??) to a value that is one to three orders of magnitude smaller than the suggested value. Then you will find that you must get the value of P_d (by adjusting its multiplicative twiddle (??) factor) to within a much more narrow range to achieve convergence. But, this is may be rather dependent on the random nature of the particular run chosen due to limited statistics (some will might prefer slightly higher than the ideal P_d and other might prefer slightly lower P_d.) So to be completely vigorous, you should do this many times on different real data or different fake data of the exactly the same statistics.

Sec. 3.9	Sec. 7.4	Sec. 8.3

7.5 ME Fiddle factors: σ looseness

Sec. 3.10, p. 17	Sec. 4.1.4, p. 25	Sec. 7.5, p. 61	Sec. 8.4, p. 64

7.6 Difference between ME and a Fourier Transform

Sec. 3.11	Sec. 7.6	Sec. 8.5

7.7 Basic choices: Binning and time window

Sec. 3.12	Sec. 7.7	Sec. 8.6

7.8 Fancy choices: Apodization and variable binning

Sec. 3.13	Sec. 7.8	Sec. 8.7

7.9 Output: Enhancement factor

Sec. 3.14	Sec. 7.9	Sec. 8.8

7.10 Output: Hey, what about error estimation?

7.10.1 Are the errors correct? Does it matter?

One thing that I would like to stress regarding the errors and the covariance matrix was my original intention of getting something about the right size, with relatively little regard to mathematical rigorness. I wanted "error bars" for experimental reasons foremost. I think that the maths are probably correct (maybe a factor of 2 hiding somewhere) but I never really intended to get that detailed. Even if they are technically correct, I am not sure how fiddle factors like the sigma looseness factor and (more recently) apodization of sigma affect the accuracy (or meaning) of the covariance matrix.

My current temptation is say that in particular $\nabla\nabla\chi^2$ (which is used in part to make the covariance matrix) absorbs most of the abuse we subject the data to. That is, $\nabla\nabla\chi^2$ usually broadens rather than the lineshape when we bin poorly or apodize, etc. $\nabla\nabla\chi^2$ wiggles when the time range is cut too short (without apodization) and the lineshape doesn't wiggle. So as far as the cult of ME goes, my latest variation is that $\nabla\nabla\chi^2$ is the sacrificial lamb for all our sins, which is why the lineshapes come out so nice. However, the indirect cost is that the errors come out larger.

I think that NOBODY has a deep, clear understanding of implications of the covariance matrix in ME yet. The more theoretically minded people have their set opinions (mostly disagreeing with my interpretation at one level or another), but us experimentalists are learning through experience (and fake data) the real usefulness of it.

I tend to be a bit skeptical, so I don't make a very good member of the cult of ME. But I hope that that pessimistic attitude means that my work on ME is better than it would be otherwise.

I think the covariance matrix gives a good guide to the errors, but I am not certain that it is completely free of being biased by how MaxError is run. Under certain conditions, the errors seem too small. More experience is needed to fully assess the degree of accuracy (or alternatively, how to run MaxError better.)

Errors from MaxError are much more convenient (being automatic) and probably better (or no worse) than errors estimated by hand by other means. They are certainly better than no error estimation at all (which is the usual default)!

Sec. 3.15	Sec. 7.10	Sec. 8.9

7.11 Output: The noise level and P_d

Sec. 3.16	Sec. 7.11	Sec. 8.10

Chapter 8

The Basics of Maximum Entropy

8.1 How entropy helps

Sec. 3.7	Sec. 7.2	Sec. 8.1

8.2 The Search Algorithm in Maximum Entropy

Sec. 3.8	Sec. 7.3	Sec. 8.2

8.3 ME fiddle factors: the default level

Sec. 3.9	Sec. 7.4	Sec. 8.3

8.4 ME Fiddle factors: σ looseness

Sec. 3.10	Sec. 7.5	Sec. 8.4

8.5 Difference between ME and a Fourier Transform

Sec. 3.11	Sec. 7.6	Sec. 8.5

8.6 Basic choices: Binning and time window

Sec. 3.12	Sec. 7.7	Sec. 8.6

8.7 Fancy choices: Apodization and variable binning

Sec. 3.13	Sec. 7.8	Sec. 8.7

8.8 Output: Enhancement factor

Sec. 3.14	Sec. 7.9	Sec. 8.8

8.9 Output: Hey, what about error estimation?

Sec. 3.15	Sec. 7.10	Sec. 8.9

8.10 Output: The noise level and P_d

Sec. 3.16	Sec. 7.11	Sec. 8.10

Chapter 9

Unfinished business: Promises, things to do, etc.

9.1 Specific chores

9.1.1 MaxError chores

- When porting MaxError to Linux, etc., make sure that ALL system dependent code is isolated in separate files, with names that either start or (better) end with the OS name.

- When porting MaxError to Linux, how do system logicals like dataPSIdir?

- Add RAL data type, the dead time correction, the calibration info, and the extra convolution.

- Use the extra arrays needed for RAL addition to improve the current problem with random round errors accumulating (best seen in fake data with no noise).

- Improve background signal subtraction code, the way Ted wants it (with its own random noise).

- Add MUD data format.

- Make gradgradchi2.db file (like the spec0%.db with no run number associated with it) to make looking at $\nabla\nabla\chi^2$ easier, with automatic plotting commands at the end for theory curves sinc and apodization. This is redundant with MaxHes part of the *.MDB file, but I should have put that in DB format originally!

- check factors of 2 in covariance matrix.

- Look over negative error code (might be mangled, due to confusion over problems in logscale in DB!)

- Compare results of using whole covariance matrix and the estimated "independent" errors in fits in DB, using fake data.

9.1.2 Other related programming chores

- Write/verify PSI2MUD code and RAL2MUD code for Linux port, since those data formats probably won't port well.

- Fix bug in USR production code or ask people if they have fixed it?. Ivan? ISIS?

- Fix logscale in DOPLOT used by DB and MSRFIT.

9.1.3 Specific editing/writing chores

- Add subsection to section 2.2 for where files are on the PSI computer.

9.1.4 Specific Latex chores

- Why doesn't ref "part:math" work?
- Why get error for tilde in URLs?
- Try keep random font size changes to a minimum.
- Try to arrange tables so they fit in the margins
- Check that book page size makes sense, both printed and for PDF.

9.2 General chores

9.2.1 General editing chores

- Spell check
- check grammar and readability
- Cut out stream of consciousness
- Make sure that the 4 part structure of manual is in fact helpful.
- Check that system of naming come latex files after the version date of the maxerror program makes sense and helps with updating.

9.2.2 General Latex chores

- Maintain systematic system of naming labels for parts, chapters, sections, tables, figures and equations.
- Bibliography and citations.
- index entries.
- fill in cross references which are currently ??.
- Put in clear "navigation bar" of horizontal cross references from one part to another, to make the most of the hyper link aspects of the manual.
- Put in "refpage" as well as "ref", so that can use the page numbers instead of section numbers.
- Cross reference all "Note:", "Warning:", and future development promises (to this chapter perhaps?) so can keep track of things and update more systematically?
- Check that system of naming come latex files after the version date of the maxerror program makes sense and helps with updating.

Bibliography

[1] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in Fortran: The Art of Scientific Computing, Second Edition*, pages 809–817. Cambridge, 1992.

[2] B.D. Rainford and G.J. Daniell. μSR frequency spectra using the maximum entropy method. *Hyperfine Interaction*, **87**:1129–1134, (1994).

[3] D.S. Sivia. *Data Analysis; a Bayesian Tutorial*, page 137ff. Oxford Science Publications, 1996.

[4] J. Skilling and R.K. Bryan. Maximum entropy image reconstruction: General algorithm. *Monthly Notices of the Royal Astromonical Society*, **211**:111–124, (1984).