Bayesian Data-Analysis Toolbox
Release 4.23, Manual Version 3

G. Larry Bretthorst
Biomedical MR Laboratory
Washington University School Of Medicine,
Campus Box 8227
Room 2313, East Bldg.,
4525 Scott Ave.
St. Louis MO 63110
http://bayes.wustl.edu
Email: gbretthorst@wustl.edu

September 18, 2018
Appendix E

Writing Fortran/C Models

In this Chapter we are going to describe how to write Fortran and C models. We are going to do this primarily for Fortran and we will briefly discuss how to do this in C. Obviously, if you are going to write Fortran or C models to be used by this system, then you must have Fortran and C installed on your system. If this is not the case, there will be nothing of use to you in this Chapter. First, there are two different types of models used in the Ascii packages, one's that marginalize out the amplitudes and those that do not. Conceptually, those that do not marginalize out the amplitudes are easiest to understand. Let's suppose the data are a simple exponential decaying data set that contains a constant offset. The signal equation, \( S(t_i) \) for this data would be simple

\[
d_i = S(t_i) + \sigma_i \quad \text{(E.1)}
\]

\[
S(t_i) = M_\infty + (M_0 - M_\infty) \exp\{-\alpha t_i\} \quad \text{(E.2)}
\]

where this model is written as an inversion recovery model: \( M_0 \) is the amplitude at time \( t = 0 \), \( M_\infty \) is the amplitude at time \( t = \infty \) and \( \alpha \) is the decay rate constant. For a single exponential model, the data, the \( d_i \), are a single column of numbers. Similarly, that abscissa, the \( t_i \), are also a single column of numbers. Finally, there are three parameters \( M_0, M_\infty \) and \( \alpha \) that Markov chain Monte Carlo simulations must estimate.

E.1 Model Subroutines, No Marginalization

The system model, system models are models which ship with the software, which implements a single exponential plus a constant with no marginalization is the ExpOneConst_NoMarg.f model. Figure E.1 is a copy of ExpOneConst_NoMarg.f without most of the comments. These were stripped out for no other reason that to make the code fit on a single page. Lines 01 through 12 are the interface to the model subroutines. This interface is exactly the same for every model and we will describe what each of these parameters are shortly. Lines 13 through 25 are the Fortran declarations for the interface, the arguments on the call list, and as such are generally things you should not change. Finally, Lines 27 through 36 are the lines of code written by me to implement the single exponential plus a constant model. Lines 30 to 32 pull out the three parameters of interest and place them in temporary work areas. This is done for readability. The three parameters are the decay
Figure E.1: Writing Models A Fortran Example

```fortran
01 Subroutine Model(CurSet, ! The current data set number
02 C NoOfParams, ! The number of nonlinear parameters
03 C NoOfDerived, ! The number of derived parameters
04 C TotalDataValues, ! The number of hyper-complex data values
05 C MaxNoOfDataValues,! The largest number of data values in all sets
06 C NoOfDataCols, ! The current number of data column
07 C NoOfAbscissaCols, ! The current number of abscissa columns
08 C NoOfModelVectors, ! The number of model vectors
09 C Params, ! The input model parameters
10 C Derived, ! The output derived parameters
11 C Abscissa, ! The abscissa values
12 C Signal) ! The output model signal
13 Implicit None
14 Integer, Intent(In):: CurSet
15 Integer, Intent(In):: NoOfParams
16 Integer, Intent(In):: NoOfDerived
17 Integer, Intent(In):: TotalDataValues
18 Integer, Intent(In):: MaxNoOfDataValues
19 Integer, Intent(In):: NoOfDataCols
20 Integer, Intent(In):: NoOfAbscissaCols
21 Integer, Intent(In):: NoOfModelVectors
22 Real (Kind=8), Intent(In):: Params(NoOfParams)
23 Real (Kind=8), Intent(Out):: Derived(NoOfDerived)
24 Real (Kind=8), Intent(In):: Abscissa(NoOfAbscissaCols,MaxNoOfDataValues)
25 Real (Kind=8), Intent(InOut)::Signal(NoOfDataCols,MaxNoOfDataValues)
26
27 Integer CurEntry
28 Real (Kind=8) DecayRate1,Amp,Const
29
30 DecayRate1 = Params(1)
31 Amp = Params(2)
32 Const = Params(3)
33
34 Do CurEntry = 1, TotalDataValues
35   Signal(1,CurEntry) = Const*Amp*Exp(-DecayRate1*Abscissa(1,CurEntry))
36 EndDo
37
38 Return
39 End
```

Figure E.1: This is an example of a Fortran routine to analyze a single exponential plus a constant without marginalization. The code from line 27 through 37 is what I entered to produce this model. Lines 30 to 32 pull out the three parameters of interest and place them in temporary work areas. This is done for readability. Lines 34 through 36 generate the exponential evaluated at the abscissa values and store the resulting model signal in the output “Signal” vector.
Figure E.2: Writing Models A C Example

```c
#include <stdio.h>
#include <math.h>

void model_(int *CurSet,
         int *NoOfParams,
         int *NoOfDerived,
         int *TotalDataValues,
         int *MaxNoOfDataValues,
         int *NoOfDataCols,
         int *NoOfAbscissaCols,
         int *NoOfModelVectors,
         double Params[],
         double Derived[],
         double Abscissa[],
         double Signal[])
{

    int CurEntry;
    double Rate, AmpZero, AmpInfty;

    Rate = Params[0];
    AmpZero = Params[1];
    AmpInfty= Params[2];

    if (Rate == 0.0)
        {Derived[0] = 0.0;}
    else
        {Derived[0] = 1.0 / Rate;}

    for (CurEntry = 0; CurEntry < *TotalDataValues; CurEntry++)
    {
        Signal[CurEntry]=AmpInfty+(AmpZero-AmpInfty)*exp(-Rate*Abscissa[CurEntry]);
    }

    return;
}
```

Figure E.2: This is an example of a CC routine to analyze a single exponential plus a constant without marginalization. However, in this code we have written the model as an inversion recovery model. The code from line 18 through 36 is what I entered to produce this model. Lines 21 to 23 pull out the three parameters of interest and place them in temporary work areas. This is done for readability. Lines 25 through 28 are examples of how one might set a derived parameter, in this case a decay time. Note that care was taken to avoid possible divide by zeros in the event the decay rate constant is allowed to go to zero. Lines 30 through 34 generate the exponential evaluated at the abscissa values and store the resulting model signal in the output “Signal” vector.
rate constant and the two amplitudes. Line 34 is a Fortran loop construct that tells the compiler it is to loop over the code between the “Do’ and the “EndDo”, while doing this the Fortran integer variable ”CurEntry” is varied from 1 to the TotalDataValues in steps of 1. Lines 35 generates the exponential plus the constant at the abscissa value specified by the index “CurEntry”. Finally, as indicated Line 46 terminates the Do loop.

Figure E.2 is CC version of this code. The structure of the code is almost identical to the Fortran. However, note the name of the model is “model_”. When Fortran compiles a subroutine it automatically appends this underscore to a model name and consequently when writing CC modes, the name must be “model_”. Lines 21 through 23 pull the parameters out of the input parameter vector and place them in temporary work areas. Lines 26 through 28 are an example of how one might set a derived parameter, in this case the inverse of the decay rate constant, or the decay time. Note that the code is careful to check that the value of the decay rate is not zero, thus avoiding a possible divide by zero. Lines 30 through 33 generate the exponential signal and place them in the signal vector, just as the Fortran does.

E.2 The Parameter File

Now an interesting question comes up here as to how it is known that parameter 1 is the decay rate constant, and that 2 and 3 are the amplitude and constant? The answer to this is that each model file must be accompanied by a parameter file and that file describes the parameters and their prior probabilities. The order of the parameters in the prior probabilities list defines the order of the parameters in the “Params” vector in the model subroutine. The parameter file associated with this exponential model is shown in Fig. E.3. In general terms the parameter file consists of three parts, the top part defines some structural features of the model, things like how many model vectors, data columns, abscissa columns and number of priors. Normally, these parameters are not used by the model subroutine, but they are used by the package to determine what parameters must be passed to the subroutine. The middle part of the parameter file contains the prior probabilities for each parameter, including prior probabilities for amplitudes that are marginalized. Finally the bottom part of this file contains a list of the derived parameter names. In this example this list has no entries because there are no derived parameters. Here is a brief description of what each line in parameter file does:

**Number of Abscissa** tell the packages how many abscissa columns this model uses. Something as simple as this single exponential plus a constant only uses a single abscissa. However, routines like the diffusion tensor analysis with a “B” matrix take 6 abscissa. As a reminder, when more than one data column or abscissa are present the file format for Ascii data files becomes a bit more complicated, see Chapter A for a description of these files.

**Number of model vectors** tell the packages how many amplitudes are being marginalized from the posterior probability. For models like this one, where no amplitudes are marginalized, the number of model vectors is zero.

**Number of data cols** tell the packages how many data columns must be present in the Ascii data file.

**Number of Priors** is the number of input priors on the following lines. In this case 3 priors follow.
Figure E.3: Every model file is accompanied by a parameter file. The parameter file shown here is for the CC model shown in Fig. E.2. Of course, the format of the parameter file is the same for both C and Fortran programs. However, in the exponential decaying signal discussed earlier, the Fortran model did not have any derived parameters while the C version did. The top part of this file contains some configuration parameters. The middle part contains the parameter names and a description of their prior probabilities. These prior probabilities, the three lines starting with “DecayRate” describe the parameter, their ranges and their prior probabilities. The last part is a list of the derived parameters. See the text for a more extensive description of this file.

DecayRate is name of the parameter. The name is used to assign output file names to the probabilities. So if you were to run this model, there would be a file with “DecayRate” in the name and that file would contain the posterior probability for the decay rate parameter. The lines starting with the DecayRate are used to define parameters and their prior probabilities. We are going to call these three lines, the prior definitions and we will use this terminology in describing these prior probabilities. The name of this parameter is “DecayRate” and this name is used in the outputs from the packages. However, to define a prior you have to have more than then name, you need the other fields on this line:

**Low** The first number on the prior definitions is the lowest, the smallest, allowed value of the parameter, in this case the decay rate constant. This Low bound is a hard bound and the Markov chain Monte Carlo simulation restricts the decay rate to be greater than or equal to the low value.

**Mean or Peak** is the second number on the prior definitions and it is used as the mean value in a Gaussian prior probability. It is used as the peak value for a positive prior probability and it is used as the parameter value when the prior type is set to parameter. For all other prior types, this parameter is ignored.

**High** is the third number on the prior definition and is the highest value the decay rate parameter is allowed to take on.

**StdDev** is fourth number on the prior definitions is the standard deviation of a Gaussian prior probability and this field, while present on other priors, is not used unless the prior type is Gaussian.

**Positive(e)** this quantity is the type of prior probability that is to be used and may be set by the user. Valid values for the prior type are: Gaussian, Uniform, Parameter, Exponential
and Positive. These prior types are set using a pull down menu, so you cannot set one incorrectly on the interface. The “(e)” tells the interface that the prior type is editable; while “(ne)” indicates the the prior type cannot be changed.

**NotOrdered(ne)** tells the package that this parameter is not an ordered parameter. Valid values are NotOrdered, LowHigh and HighLow. The “ne” tells the interface that this field is Not Editable, i.e., cannot be changed. As with the Positive entry, the ”(ne)” can also take on “(e)” meaning the ordering relationship can be changed. When Ordering is used, one must order at least two parameters. Ordering less will result in an error.

**NonLinear** is an indicator that tells the packages that this parameter is to be treated as if it appears in the model in a nonlinear way and as a result the parameter must be varied in the Markov chain Monte Carlo simulation. The other valid values in this field are “Parameter” and “Amplitude”. Parameter tells the packages that this is just a single number given by the mean, and that the Markov chain Monte Carlo simulations do not vary this parameter. Amplitude tells the packages that this parameter is an amplitude and is to be marginalized from the posterior probability.

Note that the second and third prior are clearly amplitude even though they have been declared as NonLinear parameters. When the prior type is declared as “NonLinear” the parameter is simulated in the Markov chain Monte Carlo simulation. However, when a parameter is declared as “Amplitude” the amplitude is marginalized from the posterior probability. Marginal probability density functions are often more sharply peaked then nonmarginal distributions, but when the marginalization was done the amplitudes were integrated from minus to plus infinity, thus the marginal probability can effectively constrain an amplitude to either negative or positive values and for some model this is not appropriate. Consequently, we allow the user to specify the amplitudes when needed.

The Fortran code shown in Fig. E.1 is a Fortran subroutine using a fixed format. The code consists of an interface, Lines 01 through 25, some user parameter declarations, Lines 27 and 28, and the code to generate the exponential decay, Lines 30 through 36. We are going to describe each of these three items separately starting with defining the interface.

### E.3 The Subroutine Interface

In Fortran the interface to a subroutine is often pretty simple, consisting of little more than a list of interface parameters and their definitions. In Fig. E.1 Lines 01 through 12 are the interface. They tell Fortran that this subroutine receives 12 arguments. Here is a description of these arguments and what they are used for:

**CurSet** is a 4 byte signed integer and contains the number of the current data set. The data set number is passed for the simple reason that the model could be data set dependent. If the model is data set dependent, it is up to the user to generate the appropriate signal function for the current data set.

**NoOfParams** is a 4 byte signed integer indicating the number of nonlinear priors specified in the “.params” file. Note that in general this is not the same thing as the number of prior in the .params file. However, for nonmarginalization models, as this one is, the number of parameters is the same as the number of priors. If this number is 5, then there will be 5 values in the Params vector that are used as the NonLinear parameters in the model.
**NoOfDerived** is a 4 byte signed integer containing the number of derived parameters in the model. Note that Derived is specified as an output parameter and dimensioned by NoOfDerived. Because of this dimension, when the model is called, if the number of derived is zero, a one is passed to this routine. This is done simply to avoid run time errors on some system. When a model does not generate any derived parameters, the Derived vector should not be touched or manipulated in any way.

**TotalDataValues** is a 4 byte signed integer containing the number of hyper-complex data values in CurSet. Note that this number is data set dependent and different data sets can have differing number of data values and abscissa values.

**MaxNoOfDataValues** is a 4 byte signed integer containing the maximum number of data values in all data sets. This number is used to dimension the Abscissa and the Signal parameters.

**NoOfDataCols** is the number of data columns in this data set. Usually this is just one. However, for complex data this would be 2 and for more complicated types of data, this could be any number.

**NoOfAbscissaCols** is a 4 byte signed integer similar to the NoOfDataCols except this is the number of abscissa columns. Again this is usually one, but things like diffusion tensors can have 3 or 6 or more depending on the type of data.

**NoOfModelVectors** is a 4 byte signed integer containing the number of declared amplitude parameters in the .params file. For nonmarginalized models, as this one is, the number of model vectors is zero.

**Params** is a real vector containing NoOfParams parameters. Specifically, if the parameter file contains 5 NonLinear parameters, then Params is a vector of 5 parameters, one for each parameter in the parameter file. Note that this comment is true of the packages that use Ascii models, but not necessarily true of all packages that read Ascii files. Note that the code declares this parameter as input, so you must not change the values in this vector. Having said that, you can change this parameter from “Intent(In)” to “Intent(InOut)” and then you can change these parameters. An example of when you might want to do that is when the input parameters are not ordered and the processing requires them to be ordered, so they are sorted in place.

**Derived** is a real output vector of containing space for NoOfDerived parameters. Inside of this subroutine NoOfDerived is always greater than or equal to one and the work area passed to this subroutine contains at least one entry. The reason for this is simply because Derived is declared as an output vector and as a result some Fortran compilers insist that you set its value, even when there are no derived parameters. Consequently, we pass a work area large enough to hold at least one derived parameter and the subroutine can set this parameter to satisfy the Fortran compilers. Derived parameters are a way of outputting functions of the various parameters. For example in this exponential model, one might be interested in the decay time as well as the decay rate. In that case one could compute something like Derived(1) = 1/Params(1) as a derived parameter and the package will output both Params(1) and Derived(1).

**Abscissa** is a real input vector containing the abscissa for the current data set. Note that the abscissa is a multicolumn vector the length of the current data set. In the model discussed
here, only a single abscissa is present so the abscissa is referenced as “Abscissa(1,CurEntry)” where the 1 means the first abscissa column and “CurEntry” is the current time or abscissa point. However, for something like a B vector there would be three abscissa, lets call them Bx, By and Bz, then these three abscissa would be referred to by Abscissa(1,CurEntry), Abscissa(2,CurEntry) and Abscissa(3,CurEntry) for each of the three abscissa values and which Abscissa value is Bx, By or Bz is something determined by the user.

**Signal** is an 8 byte vector used to output the model signal. This output signal is is a two dimensional vector, the first dimension being the number of data columns, and the second is the maximum number of data values. Please note that the total number of data values and the maximum number of data values are in general different. Consequently, when you generate a signal, you generate a vector containing the number of data values. You do not generate a vector containing the maximum number of data values.

When the packages call a Fortran or C model, they pass the subroutine 12 arguments. Arguments 1 through 11, excluding the derived parameters, are input arguments and are set when the model is executed. And as a general rule, it is unwise to change these values. However, the derived parameters, argument 10, and the signal vector, argument 12, are outputs that must be set by the model subroutine. Note that the signal vector may have multiple columns and if the signal vector is 5 columns, then you must compute all 5 signals for all abscissa vectors.

### E.4 The Subroutine Declarations

Fortran lines 27 and 28 and C lines 18 and 19 are declarations. In Fortran and C, these declarations tell the compiler if a variable is a real, complex or integer number. And how many bytes of storage a variable is to occupy. The various parameters that are passed to the model subroutines are either 4 byte integers are 8 byte real numbers. For those unfamiliar with this terminology a real number is a number in scientific notation.

In Fortran model codes I have written, I use a feature of Fortran called *Implicit None*. This feature tells Fortran not to assume the type of data based on its naming convention and causes Fortran to look for a declarations of the variable type. In C all variables must be declared so issues of default variable types do not occur. You can see in Fig. E.2 that the declarations are specified at the time the parameters are declared on the argument list. However, in Fortran these declarations are separate and you must declare all variables in any Model routine you write you write. In Fortran, there are three types of declarations that might be used: Integer, Real and Complex. It is my understanding that C does not support a complex definition and you must program you complex arithmetic by hand, this is not true in Fortran. All of your Fortran declarations must go after, line 25, but before the start of any executable code. The order of these declarations can sometimes matter, but usually not. The exception is when you define a parameter used in a dimension, then the parameter must be defined first. Figure E.4 contains some examples of the types of Fortran declarations you might need.

Line 01 declares a 4 byte real number named FourByToNumber and this quantity could be used as a variable in the calculations done in the Model subroutine. Line 02 declares an 8 byte real number named FourByToNumber. Line 03 declares an integer called MyLoopVariable. Line 04 declares a second integer called NoOfParams. Line 05 decals that NoOfParams is to be assigned the value 4 and this value is a constant that cannot be changed. Line 06 is an example of a single one
Figure E.4: Writing Models Fortran Declarations

01 Real (Len=4) FourByToNumber
02 Real (Len=8) EightByToNumber
03 Integer MyLoopVariable
04 Integer NoOfParams
05 Parameter (NoOfParams=4)
06 Real (Kind=8) Vector(NoOfParams)
07 Real (Kind=8) Vector2D(2, NoOfParams)
08 Complex(Kind=8) ComplexVector(NoOfParams)

Figure E.4: Line 01 declares a 4 byte real number named FourByToNumber. Line 02 declares an 8 byte real number named EightByToNumber. Line 03 declares an integer called MyLoopVariable. Line 04 declares a second integer called NoOfParams. Line 05 decays that NoOfParams is to be assigned the value 4 and this value is a constant that cannot be changed. Line 06 is an example of a single one dimensional vector of containing NoOfParams. Line 07 is an example of a single two dimensional vector of containing NoOfParams entries, but each entry consists of two 8 eight byte variables. Finally, Line 08 is an example of a complex variable. This variable contains NoOfParams complex numbers and because each complex number consists of a real and imaginary part this complex vector takes exactly as much storage as Vector2D and what’s more the storage arrangement of these vectors is identical.

dimensional vector of containing NoOfParams. Line 07 is an example of a single two dimensional vector of containing NoOfParams entries, but each entry consists of two 8 eight byte variables. Finally, Line 08 is an example of a complex variable. This variable contains NoOfParams complex numbers and because each complex number consists of a real and imaginary part this complex vector takes exactly as much storage as Vector2D and what’s more the storage arrangement of these vectors is identical.

E.5 The Subroutine Body

The body of the Fortran Model subroutine starts after Line 30 through the end of the file. In the C routine the body starts on line 21 and proceeds to the end of the file. The body of the code is where you put the instructions to compute your model signal. Ignoring the integer variables, you receive as you input the parameters, contained in the “Params” vector and the abscissa and you must compute the model signal for each value of the abscissa and place this value in the signal vector. For example in the single exponential plus a constant model, the output signal vector is a single column vector containing an exponential plus a constant.

Additionally, the body of the code must compute the derived parameters and place them in the derived vector. A derived parameter is some function of the input parameters that the user wishes to output. For example, in this model we process the exponential signal using a decay rate constant. However, we could have used a decay time constant. The two formulations are exactly identical, but the probability density functions for a decay rate are not the same as the probability density functions for a decay time. If we had desired to see the probability density function for the decay time, we could have defined a derived parameter, and by adding a single line of code to the above,
“Derived(1) = 1.0/DecayRate1” we could output both probability density functions.

Any valid executable statement can be in the body of the Fortran or CC code including function, subroutine calls and IO statements. However, with I/O extreme care must be exercised to make sure that you do your IO only a single time and save the results in variables having the save attribute because, your model function will be called millions to times by the Markov chain Monte Carlo simulation.

If there are function or subroutine calls in your model subroutine, these routines must be part of the same physical file. For example, suppose the loaded model is named test.f and it calls a routine named “MyModel,” then test.f must consists of the “test” model subroutine and the “MyModel” subroutine. If your model calls multiple subroutines, then these subroutines must also be a part of the same physical source code. Finally, if you use a model which calls multiple subroutines or functions in a model selection calculation, then all subroutine and function names called by all of your models must be unique.

E.6 Model Subroutines With Marginalization

There are two basic types of model, those which do not marginalize out any amplitudes and those that do. Model routines that marginalize amplitudes are different in their functional requirements. Typically, the relationship between the data and the model is given by:

\[ d_k(t_i) = \sum_{j=1}^{m} A_{jk} G_j(\Omega, t_i) + \sigma_{ki} \]  

(E.3)

where there are multiple data sets, \( k \in \{1, 2, \ldots, n\} \), and each data set is modeled as the same functional form but having a unique set of amplitudes for each data set. We usually call the functions, \( G_j(\Omega, t_i) \), model vectors because each \( G_j(\Omega, t_i) \) is an \( N \) dimensional vector in \( t_i \). Finally, we are using \( \Omega \) to stand for the collection of all of the nonlinear parameters in the model.

When generating a subroutine to process this model using marginalization, it is the \( G_j(\Omega, t_i) \) that must be computed, not the sum, because in a marginal probability distribution there are no amplitudes. Rather one computes the \( G_j(\Omega, t_i) \) and uses these quantities in the calculation for the posterior probability, see Chapter ?? for more on how marginal probabilities are computed. To make this more concrete, suppose we have a single exponential plus a constant model:

\[ d(t_i) = M_\infty + (M_0 - M_\infty) \exp\{-\alpha t_i\}. \]  

(E.4)

This equation is not in the form given by Eq. E.3. However, it is simple to rearrange it into this form:

\[ d(t_i) = M_0 \exp\{-\alpha t_i\} + M_\infty (1 - \exp\{-\alpha t_i\}) \]  

(E.5)

and in this form, the two amplitudes are given by \( A_1 = M_0 \) and \( A_2 = M_\infty \) and the two model equations, \( G_1(\Omega, t_i) \) and \( G_2(\Omega, t_i) \) are given by

\[ G_1(\Omega, t_i) = \exp\{-\alpha t_i\} \]  

(E.6)

and

\[ G_2(\Omega, t_i) = 1 - \exp\{-\alpha t_i\}. \]  

(E.7)
Formally, the model now reduces to

\[
    d(t_i) = A_1 G_1(\Omega, t_i) + A_2 G_2(\Omega, t_i)
\]  

(E.8)

and it these model functions, \(G_1(\Omega, t_i)\) and \(G_2(\Omega, t_i)\), that must be programmed into a model subroutine using marginalization. As illustrated in Fig. E.5, the two model functions are programmed into the model subroutine. Because it is the model functions, the \(G_j(\Omega, t_i)\), that must be programmed, the name of the output signal vector has been changed from “Signal” to “Gij”. Note that the dimension of the Gij vector is given by the number of data columns, the maximum number of data values and finally the number of model vectors. This Gij vector must be filled in for each data column, for each data value in the current set and for each model function. Also note that in this model subroutine there is only a single parameter, the decay rate constant; the amplitudes are not present in the parameter vector. Only the parameters that are not marginalized appear in the parameter vector. The order of the parameters in this vector, is the order given in the parameter file. In the parameter file, the amplitude parameters must appear after all other parameters. They cannot appear before any nonlinear parameters. If they do appear out of order, the program that implements the calculation will issue an error on the console and in the mcmc.values report and the stop.

In addition to computing the model vectors, the \(G_j(\Omega, t_i)\), rather than the signal, \(S(t_i)\), there are also changes in the parameter file, see Fig. E.6. In particular the number of model vectors is now set to 2 because there are two \(G_j(\Omega, t_i)\). As noted, the nonlinear parameters, in this case the decay rate constant, must come before the amplitudes and again this is illustrated in Fig. E.6. The amplitude parameters are now designated as “Amplitudes” and, finally, the range on the amplitudes is set to a large number covering both negative an positive values. This range is not actually used by the program that implements the calculation, rather the program requires the prior range for an amplitude to be \((-\infty \leq A_j \leq \infty)\), because that was the range used when the amplitudes were marginalized from the posterior probability. Consequently, if its is important to impose a prior range on an amplitude, you should use a nonmarginalization routine where you can specify the exact prior probabilities. One last note, in Fig. E.6 the prior type and the ordering relationship for the amplitudes are set to “ne”, not editable, because the prior type and the ordering relationships are built into the calculations and cannot be changed by the users. Even if you tried to set these parameters to “editable”, and even if the interface permits you to do this, the programs that implement the calculation will not use that information.
Figure E.5: Writing Models Fortran Example

```fortran
01 Subroutine Model(CurSet, ! The current data set number
02 C NoOfParams, ! The number of nonlinear parameters
03 C NoOfDerived, ! The number of derived parameters
04 C TotalDataValues, ! The number of hyper-complex data values
05 C MaxNoOfDataValues, ! The largest number of data values in all sets
06 C NoOfDataCols, ! The number of data column
07 C NoOfAbscissaCols, ! The number of abscissa columns
08 C NoOfModelVectors, ! The number of model vectors
09 C Params, ! The input parameters
10 C Derived, ! The output derived parameters
11 C Abscissa, ! The abscissa values
12 C Gij) ! The output Gij vector
13 Implicit None
14 Integer, Intent(In):: CurSet
15 Integer, Intent(In):: NoOfParams
16 Integer, Intent(In):: NoOfDerived
17 Integer, Intent(In):: TotalDataValues
18 Integer, Intent(In):: MaxNoOfDataValues
19 Integer, Intent(In):: NoOfDataCols
20 Integer, Intent(In):: NoOfAbscissaCols
21 Integer, Intent(In):: NoOfModelVectors
22 Real (Kind=8), Intent(In):: Params(NoOfParams)
23 Real (Kind=8), Intent(Inout):: Derived(NoOfDerived)
24 Real (Kind=8), Intent(In):: Abscissa(NoOfAbscissaCols,MaxNoOfDataValues)
25 Real (Kind=8), Intent(Inout):: Gij(NoOfDataCols,MaxNoOfDataValues,NoOfModelVectors)
26 Integer CurEntry
27 Real (Kind=8) DecayRate1
28
29 DecayRate1 = Params(1)
30
31 Do CurEntry = 1, TotalDataValues
32   Gij(1,CurEntry,1) = Exp(-DecayRate1*Abscissa(1,CurEntry))
33   Gij(1,CurEntry,2) = 1-Exp(-DecayRate1*Abscissa(1,CurEntry))
34 EndDo
35
36 Return
37 End
```

Figure E.5: This is a single exponential plus a constant model when the amplitudes are marginalization from the posterior probability. As explained in the text, the model equation must be written in the form

\[ d_i = A_1 G_1(\Omega, t_i) + A_2 G_2(\Omega, t_i) + \cdots \]

In a marginalized model, the two model functions, \( G_1(\Omega, t_i) \) and \( G_2(\Omega, t_i) \), are programmed into the model.
Figure E.6: Writing Models The Parameter File

1 Number of Abscissa
2 Number of model vectors
1 Number of data cols
3 Number of Priors
DecayRate 0.000E+00 1.000E+00 1.000E+01 1.000E+00 Positive(e) NotOrdered(ne) NonLinear
AmpInit -1.000E+06 0.000E+00 1.000E+06 3.000E+05 Gaussian(ne) NotOrdered(ne) Amplitude
AmpFinal -1.000E+06 0.000E+00 1.000E+06 3.000E+05 Gaussian(ne) NotOrdered(ne) Amplitude
0 Number of Derived parameters

Figure E.6: The parameter file for a model subroutine using marginalization is shown here. The major modification are that the number of model vectors is now 2, one for each of the two model functions shown in Fig. E.5 lines 32 and 33. Additionally, note that the amplitudes very over a large range, and this range in the programs that implements this calculation is taken to be minus to plus infinity. Also note that the prior type and the ordering relationships are not editable. The marginalization is switched on by specifying the parameter type as a Amplitude and Amplitude parameters must be the last entries in the prior list.
Bibliography


479


[45] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953), “Equation of State Calculations by Fast Computing Machines,” Journal of Chemical Physics. The previous link is to the American Institute of Physics and if you do not have access to Science Sitations you may not be able to retrieve this paper.


Index

$A_k$ definition, 349
$H_{jk}(t_i)$ definition, 349
$\lambda_k$ definition, 349
$g_{jk}$ eigenvalue, 349

Abscissa, 437
  Computational, 436
  Generating, 427
  Loading, 39
  Multicolumn, 437
  Number Of Columns, 458
  Total Data Values, 456

Aliases, 113, 126
Amplitudes orthonormal definition, 349

Analyze Image Pixel Package, 411
  Modification History, 413
  Phased Images, 397

Reports
  Bayes Accepted, 425
  Using, 425

Viewers
  Fortran/C Models, 423
  Image, 423
  Prior Probabilities, 425

Widgets
  Build, 423
  Find Outliers, 423
  Get Statistics, 425
  System, 423
  User, 423

Ascii Data Viewer, 53

Assigning Probabilities, 118

Bandwidth, 111, 127
Bayes Analyze Package, 155
  Levenberg-Marquardt, 171
    Step, 194
  Algorithm, 175
  Amplitudes, 197, 198
  Bayes Model, 159, 161
  Bayesian Calculations, 167
  Bruker, 162
  Build BA Model, 159
  Covariance, 174
  Default Parameters Settings, 155
  Error Messages, 200
  Fid Model Viewer, 160
  Interface, 156
  Likelihood
    Gaussian, 158
    Student’s $t$-distribution, 158
INDEX

Log File, 193, 195
Lorentzian lineshape, 161
Marking Resonances, 157
Model
  \( J_0 \), 165
  \( J_p \), 165
  \( J_s \), 165
Amplitude, 163, 164
Bessel Function, 163
Constants Models, 157
Correlated, 157, 162, 164
Equation, 161, 164, 164
First Order Phase, 157, 162, 164
First Point, 162, 164
Gaussian, 163
Imaginary Constant, 164
Multi-Exponential, 163
Multiple Data Sets, 165
Multiplet Order, 164
Multiplet Orders, 164
Multiplets, 162
Multiplets of Multiplets, 164
Non-Lorentzian, 163
Offsets, 162
Real Constant, 164
Relative Amplitude, 164–166
Resonance Frequency, 165
Shim Order, 163
Shimming, 166
Shimming Order, 164
Uncorrelated, 157, 162, 164
Zero Order Phase, 157, 162, 164
Model Interface, 160
Multiplets, 158
Newton-Raphson, 171
Noise File, 158
Noise Standard Deviation, 158
Outputs
  Bayes.accepted File, 177
  bayes.log.mnnn File, 177, 193, 193
  bayes.model.mnnn File, 177, 185, 197, 197
  bayes.noise File, 180
  bayes.noise.mnnn File, 158, 180
  bayes.output.mnnn File, 176, 186, 186
  bayes.params File, 176, 177
  bayes.params.mnnn File, 176, 177, 177
  bayes.probabilities.mnnn File, 177, 190, 190
  bayes.status.mnnn File, 177, 196, 200
  bayes.summary1.mnnn File, 177, 198, 198
  bayes.summary2.mnnn File, 177, 199, 199
  bayes.summary3.mnnn File, 177, 200, 200
Global Parameters, 182, 183
Model File, 184
Probabilities file, 191
Zero Order Phase, 182
Parameter File
  Activate Shims, 180
  Analysis Directory, 178
  By Fid, 181
  Data Type, 180
  Default Model, 181
  Directory Organization, 180
  Fid Model Name, 178
  File Version, 178
  First Fid, 181
  First Order Phase, 180, 183
  Imaginary Constant, 184
  Last Fid, 181
  lh, 182
  Maximum Candidates, 182
  Maximum New Resonances, 182
  Model Fid Number, 181
  Model Name, 184
  Model Names, 181
  Model Number, 184
  Model Points, 181
  Multiplets of Multiplets, 185
  Noise Start, 181
Numerical Parameters, 178
  Prior Odds, 182
  Procpar, 178
  Real Constant, 184
  Relative Amplitude, 183
  Resonance Model, 185
  Shim Order, 182
  Spectrometer Frequency, 182
  Text Parameters, 178
  Total Complex Data Values, 181
  Total Data Values, 181
  Total Sampling Time, 182
  True Reference, 182
INDEX

Units, 180
Use Noise StdDev, 180
User Reference, 182
Prior Probabilities, 167
Probabilities File, 191
Product Rule, 168
Relative Amplitude, 167
Remove Resonances, 159
Reports
Bayes Status, 155
Save/Reset, 159
Search, 166
Levenberg-Marquardt, 166
Short Parameter Description, 195
Siemens, 162
Status File, 196
Steepest Descents, 173
Sum Rule, 168
Summary File, 198
Summary Reports, 176
Summary2, 199
Summary3, 201
Units, 161
Using, 157
Varian/Agilent, 162
Widgets, 155
By, 158, 176
First Point, 157, 163
From, 158, 176
Imag Offset, 163
Imaginary Offset, 157
Mark, 159
Max New Res, 157
New, 159
Noise, 158
Phase, 157
Primary, 158
Real Offset, 157, 163
Remove, 159
Remove All, 159
Reset, 159, 193
Restore, 159
Save, 159
Secondary, 159
Shim Order, 157, 163
Signal, 158
To, 158, 176
Bayes Find Resonances Package, 239
Bayesian Calculations, 241
Current Fid, 239
Model Equation, 241
Number of data sets, 239
Phase Model
Automatic, 239, 242
Common, 239, 242
Independent, 239, 242
Prior Probabilities, 243–245
Reports
Bayes Accepted, 241, 246
Condensed, 246
Console log, 246
McMC Values, 246
Prob Model, 246
Using, 239, 241
Viewers
Fid Data, 240
Fid Model, 240, 246
File, 246
Plot Results, 246
Text, 246
Widgets
Build FID Model, 240, 241, 246
Constant, 239, 242
First Trace, 239
Last Trace, 239
Model Fid Number, 241
Phase Model, 239, 242
Bayes Home Directory, 45, 49
Bayes Manual pdf, 469
Bayes Metabolite Package
Widgets
Shift Left, 222
Shift Right, 222
Bayes Metabolite Package, 219
Aligning Resonances, 221
Bayesian Calculation, 225
Metabolite Locations, 221
Model Equation, 223
Reports
Bayes Accepted, 221, 238
Condensed, 238
Console log, 238
INDEX

<table>
<thead>
<tr>
<th>Topic</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>McMC Values</td>
<td>238</td>
</tr>
<tr>
<td>Prob Model</td>
<td>238</td>
</tr>
<tr>
<td>Viewers</td>
<td></td>
</tr>
<tr>
<td>Fid Data</td>
<td>219</td>
</tr>
<tr>
<td>Fid Model</td>
<td>221, 236</td>
</tr>
<tr>
<td>File</td>
<td>222, 238</td>
</tr>
<tr>
<td>Metabolite</td>
<td>221</td>
</tr>
<tr>
<td>Plot Results</td>
<td>238</td>
</tr>
<tr>
<td>Text</td>
<td>238</td>
</tr>
<tr>
<td>Widgets</td>
<td></td>
</tr>
<tr>
<td>Fid Model</td>
<td>221</td>
</tr>
<tr>
<td>Fid Model Viewer</td>
<td>221</td>
</tr>
<tr>
<td>Load System Metabolite File</td>
<td>219</td>
</tr>
<tr>
<td>Load System Resonance File</td>
<td>221</td>
</tr>
<tr>
<td>Load User Metabolite File</td>
<td>219</td>
</tr>
<tr>
<td>Load User Resonance File</td>
<td>221</td>
</tr>
<tr>
<td>Shift Left</td>
<td>221</td>
</tr>
<tr>
<td>Shift Right</td>
<td>221</td>
</tr>
<tr>
<td>Bayes Model</td>
<td>159, 159</td>
</tr>
<tr>
<td>Bayes Test Data Package</td>
<td>427</td>
</tr>
<tr>
<td>Parameters</td>
<td>431</td>
</tr>
<tr>
<td>Reports</td>
<td></td>
</tr>
<tr>
<td>Bayes Accepted</td>
<td>428</td>
</tr>
<tr>
<td>Condensed</td>
<td>429</td>
</tr>
<tr>
<td>McMC Values</td>
<td>429, 431–433</td>
</tr>
<tr>
<td>Viewers</td>
<td></td>
</tr>
<tr>
<td>Fortran/C Models</td>
<td>427</td>
</tr>
<tr>
<td>Image</td>
<td>428</td>
</tr>
<tr>
<td>Prior Probabilities</td>
<td>427</td>
</tr>
<tr>
<td>Text Data</td>
<td>430</td>
</tr>
<tr>
<td>Text Results</td>
<td>429</td>
</tr>
<tr>
<td>Widgets</td>
<td></td>
</tr>
<tr>
<td># Images</td>
<td>427</td>
</tr>
<tr>
<td># Slices</td>
<td>427</td>
</tr>
<tr>
<td>Abscissa</td>
<td>427</td>
</tr>
<tr>
<td>ArrayDim</td>
<td>427</td>
</tr>
<tr>
<td>Build</td>
<td>427</td>
</tr>
<tr>
<td>Get Job</td>
<td>428</td>
</tr>
<tr>
<td>Max Value</td>
<td>427</td>
</tr>
<tr>
<td>Noise SD</td>
<td>427</td>
</tr>
<tr>
<td>Parameter Ranges</td>
<td>428</td>
</tr>
<tr>
<td>Pe</td>
<td>427</td>
</tr>
<tr>
<td>Ro</td>
<td>427</td>
</tr>
<tr>
<td>Run</td>
<td>428</td>
</tr>
<tr>
<td>Set (server)</td>
<td>428</td>
</tr>
<tr>
<td>Status</td>
<td>428</td>
</tr>
<tr>
<td>Bayes.accepted</td>
<td></td>
</tr>
<tr>
<td>Body</td>
<td>77</td>
</tr>
<tr>
<td>Header</td>
<td>76</td>
</tr>
<tr>
<td>Behrens-Fisher Package</td>
<td>311</td>
</tr>
<tr>
<td>Bayesian Calculations</td>
<td></td>
</tr>
<tr>
<td>Derived Probabilities</td>
<td>320</td>
</tr>
<tr>
<td>Different Mean And Same Variance</td>
<td>318</td>
</tr>
<tr>
<td>Different Mean And Variance</td>
<td>319</td>
</tr>
<tr>
<td>Parameter Estimation</td>
<td>321</td>
</tr>
<tr>
<td>Same Mean And Different Variance</td>
<td>317</td>
</tr>
<tr>
<td>Same Mean And Variance</td>
<td>315</td>
</tr>
<tr>
<td>Model Equation</td>
<td></td>
</tr>
<tr>
<td>Different Mean And Same Variance</td>
<td>318</td>
</tr>
<tr>
<td>Different Mean And Variance</td>
<td>319</td>
</tr>
<tr>
<td>Same Mean And Different Variance</td>
<td>317</td>
</tr>
<tr>
<td>Same Mean And Variance</td>
<td>315</td>
</tr>
<tr>
<td>Number of data sets</td>
<td>311</td>
</tr>
<tr>
<td>Parameter Listing</td>
<td>323</td>
</tr>
<tr>
<td>Prior Probabilities</td>
<td></td>
</tr>
<tr>
<td>Different Mean And Same Variance</td>
<td>318</td>
</tr>
<tr>
<td>Different Mean And Variance</td>
<td>319</td>
</tr>
<tr>
<td>Same Mean And Different Variance</td>
<td>317</td>
</tr>
<tr>
<td>Same Means And Same Variance</td>
<td>315</td>
</tr>
<tr>
<td>Reports</td>
<td></td>
</tr>
<tr>
<td>Bayes Accepted</td>
<td>311, 322</td>
</tr>
<tr>
<td>Condensed</td>
<td>322</td>
</tr>
<tr>
<td>Console Log</td>
<td>322, 323</td>
</tr>
<tr>
<td>McMC Values</td>
<td>322, 323</td>
</tr>
<tr>
<td>Prob Model</td>
<td>322</td>
</tr>
<tr>
<td>Using</td>
<td>311</td>
</tr>
<tr>
<td>Viewers</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>322</td>
</tr>
<tr>
<td>Plot Results</td>
<td>322, 324</td>
</tr>
<tr>
<td>Prior Probabilities</td>
<td>311</td>
</tr>
<tr>
<td>Text</td>
<td>322</td>
</tr>
<tr>
<td>Widgets</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>311</td>
</tr>
<tr>
<td>Big Endian</td>
<td>471, 473</td>
</tr>
<tr>
<td>Big Magnetization Transfer Package</td>
<td>259</td>
</tr>
<tr>
<td>Bayesian Calculations</td>
<td>259</td>
</tr>
<tr>
<td>Files</td>
<td></td>
</tr>
<tr>
<td>Bayes Analyze</td>
<td>264</td>
</tr>
</tbody>
</table>
Fid, 263
Peak Pick, 262
Model Equation, 261
Number of data sets, 259
Prior Probabilities, 261
Reports
Bayes Accepted, 259, 262
Condensed, 262
Console log, 262
McMC Values, 262
Prob Model, 262
Using, 259
Viewers
Ascii Data, 259
File, 262
Prior Probabilities, 259
Text, 262
Widgets
Find Outliers, 259
Big Peak/Little Peak Package, 207
Bayesian Calculations, 209
Fid Analyzed, 207
Model Equation, 210
Metabolites, 209
Solvent, 210
Number of data sets, 207
Prior Probabilities
Metabolite, 207
Solvent, 207
Removing Resonances, 207
Reports
Bayes Accepted, 209, 216
Condensed, 216
Console log, 216
McMC Values, 216
Prob Model, 216
Using, 207
Viewers
File, 216
Model, 209
Plot Results, 216
Prior Probabilities, 207
Text, 216
Widgets
Metabolite, 207
Solvent, 207
Binned Density Function Estimation, 355
Binned Histogram Package
Reports
Bayes Accepted, 357
Viewers
Ascii, 355
Binned Histograms Package
Using, 357
Viewers
Prior Probabilities, 355
Bloch-McConnell Equations, 267, 277
Changing the Bayes Home Directory, 469
Compilers, 29
CC, 29, 455
Fortran, 29, 455
Correlations, 91
Diffusion Tensor Package, 247
Ascii File Formats, 247, 254, 255
Bayesian Calculations, 249
Prior Probabilities
\( \Delta, 254 \)
\( \Gamma, 254 \)
\( \delta, 254 \)
\( \sigma, 253 \)
Amplitudes, 253
Eigenvalues, 253
Euler Angles, 253
Likelihood, 253
Parameter, 254
Reports
Bayes Accepted, 247, 255
Condensed, 255
Console log, 255
McMC Values, 255
Prob Model, 255
Symmetries, 253
Using, 247
Viewers
File, 247, 255
Plot Results, 255
Prior Probabilities, 247, 253
Text, 255
Widgets
Abscissa Options, 248
INDEX

Find Outliers, 247
Include Constant, 247, 248, 255
Tensor Number, 247, 248, 255
Use $b$ Matrix, 255
Use $b$ Vectors, 255
Use $g$ Vectors, 254
Discrete Fourier Transform, 110, 113, 123

Enter Ascii Model Package, 329
Bayesian Calculations, 332
Marginalization, 332
No Marginalization, 331
Fortran/C Models, 330, 335
Model Equation
Marginalization, 331
No Marginalization, 331
Output Names
Derived, 335
Parameters, 335
Reports
Bayes Accepted, 331, 335
Bayes Params, 335
Condensed, 335
Console log, 335
McMC Values, 335
Prob Model, 335
Using, 331
Viewers
Ascii Data, 329
File, 335
Fortran/C Models, 329
Plot Results, 335
Prior Probabilities, 329
Text, 335
Widgets
Build, 329
Find Outliers, 329
System, 329
User, 329

Enter Ascii Model Selection Package, 341
Bayesian Calculations
Marginalization, 346
No Marginalization, 344
Fortran/C Models, 341, 343, 353
Model Equation, 343
No Marginalization, 343
With Marginalization, 347
Output Names
Derived, 354
Parameters, 353
Reports
Bayes Accepted, 343, 353
Condensed, 353
Console log, 353
McMC Values, 353
Params File, 353
Prob Model, 353
Using, 343
Viewers
Ascii Data, 341
File, 353
Fortran/C Models, 341
Plot Results, 353
Prior Probabilities Not Used, 341
Text, 353
Widgets
Build Not Used, 341
Find Outliers, 341
System, 341
User, 341

Errors In Variables Package, 303
Ascii File Formats
Errors In X and Y Known, 303, 309
Errors In X Known, 303, 309
Errors In Y Known, 303, 309
Errors Unknown, 303, 309
Bayesian Calculations, 305
Data Error Bars, 303
Files
Ascii, 303
Bayes Analyze, 303
Peak Pick, 303
Model Equation, 305
Number of data sets, 303
Reports
Bayes Accepted, 305, 309
Condensed, 309
Console log, 309
McMC Values, 309
Prob Model, 309
Using, 305
Viewers
Ascii Data, 303
File, 309
Plot Results, 309
Text, 309
Widgets
  Given Errors In, 303
  Order, 303
Exponentials
  Given Package, 137
  Inversion Recovery Package, 151
  Magnetization Transfer Package, 267
  Unknown Number of Package, 143
Fid Data Viewer, 53
Fid Model Viewer, 68
File Format
  Ascii, 436
File Viewer, 80
Files
  4dfp, 59, 428, 430, 470, 471
    Header, 473
    Reading, 471
  Abscissa, 39, 77, 470
  afh, 53
  ASCII, 35, 36
  Ascii, 53, 54, 435
    k-space, 437
    Abscissa, 435, 436, 437
  Data, 435
    Image, 436
  Bayes Analyze, 36
  Bayes.accepted, 51, 76
  Bayes.params, 76, 79
  Bayes.prob.model, 447
  BayesManual.pdf, 469
  Condensed, 77, 78
  Console.log, 76, 79, 465
  dir.info, 470
  fid, 470, 470
    ASCII, 36
    ffh, 56
    Model, 68, 70
    propar, 470
  Siemens Raw, 36
  Siemens Rda, 36
  Spectroscopic, 53
  Varian fid, 36
Fortran/C Models, 42, 455, 457, 458, 465–467
Images
  4dfp, 38
  Binary, 38
  Bruker 2dseq, 38
  Bruker stack, 38
  DICOM, 38
  FDF, 38
  Multi-Column Text, 38
  Siemens IMA, 38
k-space
  Text, 36
    Varian fid, 36
  mcmc.values, 76, 449
Model Listing, 77
  prob.model, 76
  procpar, 470
  Raw, 36
  RDA, 36
Statistics, 65
  System.err.txt, 469
  System.out.txt, 469
  Varian fid, 36
  WaterViscosityTable, 469
Fortran/C Model Viewer, 93
Popup Editor, 93
Fortran/C Models, 42, 330, 335, 353, 455
  Abscissa, 463
  Body, 463
    Abscissa, 457
  Declarations, 462
  Derived Parameters, 457, 459, 463
  Edit/Create New Model, 42, 455
  I/O, 464
Marginalization, 464
  \( G_f(\Omega, t_i) \), 464
  Amplitude Range, 465
  Example, 465, 466
  Model Vectors, 465
  Ordering Amplitudes, 465
  Parameter File, 465, 467
  Parameter Order, 465
  Parameters, 465
  Model Files, 455
Model Selection, 464
No Marginalization, 457
$S(t)$, 455
Example, 456
Parameter File, 458, 459, 465
Parameters, 463
Signal, 463
Subroutine Interface, 460
  Abcissa, 462
  Current Set, 460
  Derived Parameters, 461
  Maximum No Of Data Values, 461
  Number Of Abscissa Columns, 461
  Number Of Data Columns, 461
  Number Of Derived Parameters, 461
  Number Of Model Vectors, 461
  Number Of Parameters, 460
  Parameters, 461
  Signal, 462
  Total Complex Data Values, 461
Subroutines and Functions, 464

Frequency Estimation, 114, 132

Given Exponential Package, 137
  Bayesian Calculations, 140
Files
  Ascii, 137
  Bayes Analyze, 137
  Peak Pick, 137
Model Equation, 139
Number of data sets, 139
Prior Probabilities, 139–141
Reports
  Bayes Accepted, 137, 141
  Condensed, 141
  Console log, 141
  McMC Values, 141
  Prob Model, 141
Symmetries, 141, 148
Using, 137
Viewers
  File, 141
  Plot Results, 141
  Prior Probabilities, 137, 139
  Text, 141
Widgets

Constant, 137, 139
Find Outliers, 137
Given Order, 27
Include Constant, 27
Order, 137, 139

Given Polynomial Order Package, 285
  Bayesian Calculations, 288
Files
  Ascii, 285
  Bayes Analyze, 285
  Peak Pick, 285
Gram-Schmidt, 287
Model Equation, 287
Number of data sets, 285
Prior Probabilities, 289
Reports
  Bayes Accepted, 285, 291
  Condensed, 291
  Console log, 291
  McMC Values, 291
  Prob Model, 291
Scatter Plots, 292
Using, 285
Viewers
  File, 290
  Plot Results, 291
  Text, 290
Widgets
  Set Order, 285

Histograms
  Binned, 381
  Kernel Density, 381

Image Model Selection Package, 415
  Abcissa, 415
  Fortran/C Models, 415, 417
Reports
  Bayes Accepted, 417
Using, 417
Viewers
  Fortran/C Models, 415
  Image, 415
Widgets
  Noise SD, 415
  System, 415
Use Gaussian, 415
User, 415
Image Viewer, 59
Images
  Flip
    Horizontal, 63
    Vertical, 63
  Grayscale, 63
  ImageJ, 63
  Original, 63
Inversion Recovery Package, 151
  Bayesian Calculations, 153
  Model Equation, 153
  Number of data sets, 153
  Prior Probabilities, 153
Reports
  Bayes Accepted, 151, 154
  Condensed, 154
  Console Log, 154
  McMC Values, 154
  Prob Model, 154
Using, 151
Viewers
  Plot Results, 154
  Prior Probability, 151
Widgets
  Find Outliers, 151
Kernel Density Function Package, 361
  Ascii File Format, 361
  Bayesian Calculations, 369
  Data Requirements, 361
  Data, Model And Residuals, 369
  Kernels, 369
    Biweight, 362
    Cosine, 362
    Epanechnikov, 362
    Exponential, 362
    Gaussian, 362, 370
    nonnegative, 361
    Real Valued, 361
    Triangular, 362
    Tricube, 362
    Triweight, 362
    Uniform, 362
  Likelihood, 371
Number of data sets, 364
Plots
  Expected Density Function, 367, 368
  Mean Density Function, 367, 368
  Posterior Probability for the Kernel Type, 365
  Posterior Probability for the Number Of Kernels, 366
  Scatter Plots of Model Averaged Density Function, 368
  Standard Deviation of the Mean Density Function, 367, 368
Prior Probabilities
  Kernel Center, 371
  Kernel Smoothing Parameter, 371
  Kernel Type, 370
  Number Of Kernels, 370
Reports
  Bayes Accepted, 364
  Condensed, 372
  McMC Values, 372
  Prob Model, 372
Using, 364
Viewers
  Ascii, 361
Widgets
  Kernel Type, 364
  Output Size, 364
Levenberg-Marquardt, 171
Linear Phasing Package, 395, 409
  Interface, 397
  Model Equation, 398
Widgets
  cf, 403
  Display, 403
  Display Array Element, 403
  fn, 403
  fn1, 403
  Image Type, 402
  Load An Image, 402
  np, 403
  nv, 403
  Process, 403
Load Working Directory, 33
Logical Independence, 117
INDEX

Magnetization Transfer Kinetics Package, 275
  Arrhenius Plot, 281
  Bayesian Calculation, 278
  Boltzmann’s Constant, 277
  Eyring Equation, 275, 276, 277, 280
  Model Equation, 277
  Plank’s Constant, 277
  Prior Probabilities, 279
  Reports
    Bayes Accepted, 277, 281
    Condensed, 281
    Console log, 281
    McMC Values, 281
    Prob Model, 281
  Sum and Difference Variables, 280
  Transmission coefficient, 277
  Universal Gas Constant, 277
  Using, 277
  van’t Hoff Plot, 281
  Viewers
    Ascii File, 275
    File, 281
    Prior Probabilities, 275
    Text, 281
  Widgets
    Load, 275, 281
    Set, 275
    Uncertainty, 275

Magnetization Transfer Package, 265
  Bayesian Calculations, 267
  Files
    Ascii, 265
    Bayes Analyze, 265
    Inversion Recovery, 272
    Peak Pick, 265
  Model Equation, 267
  Number of data sets, 265
  Prior Probabilities, 265, 270
  Reports
    Bayes Accepted, 267, 272
    Condensed, 272
    Console log, 272
    McMC Values, 272
    Prob Model, 272
  Three Column Data, 265
  Using, 267

Viewers
  Ascii Data, 265
  Fid Data, 272
  File, 271
  Plot Results, 262, 272, 281
  Prior Probabilities, 265
  Text, 271

Widgets
  Find Outliers, 265

Marginalization, 100
  Bayes Analyze Package, 174
  Behrens-Fisher, 315
  Big Magnetization Transfer, 261
  Big Peak/Little Peak, 211
  Diffusion Tensors, 252
  Enter Ascii Model Package, 331
  Errors In Variables, 306
  Fortran/C Models, 464
  Given Exponential, 139
  Inversion Recovery, 153
  Linear Phasing, 399
  Magnetization Transfer, 269
  Magnetization Transfer Kinetics, 278
  Metabolic Analysis, 225
  Nonexhaustive Hypotheses, 101
  Nuisance Hypotheses, 100
  Nuisance Parameter, 100
  Unknown Number of Exponentials, 146

Markov chain Monte Carlo, 132, 439
  Acceptance Rate, 444
  Annealing Schedule, 91, 442
    Dynamic, 443
    Linear, 442
  Killing Simulations, 443
  Maximum Posterior Probability, 91
  Metropolis-Hastings, 439
  Mixing, 91
  Monte Carlo Integration, 440
  Multiple Simulations, 441
  Posterior Probability, 440
  Random Number Generators, 440
  Repeats, 91
  Sampling, 91
  Simulated Annealing, 442
  the Proposal, 444
MaxEnt Density Function Estimation Package,  
   373  
   Data Requirements, 381  
Plots  
   Contour/Scatter, 375, 379  
   Number Of Multipliers, 375, 378  
Reports  
   Bayes Accepted, 375  
   Console Log, 375  
Using, 375  
Viewers  
   Ascii, 373  
   Plot, 375, 378  
   Prior Probabilities, 373  
Widgets  
   Histogram Size, 373  
   Order, 373  
Maximum Entropy Method Of Moments, 102,  
   377, 381  
   Advantages, 386  
   Problems, 386  
   Review, 381  
Maximum Entropy Method Of Moments Package  
   Bayesian Calculations, 387  
Plots  
   Data, Model and Residuals, 380  
Menus  
   Files, 24, 35  
      4dfp, 37, 38  
      Abscissa, 35, 39  
      ASCII, 35, 36  
      Binary, 38  
      Bruker, 37  
      Bruker 2dseq, 38  
      Bruker Stack, 38  
      DICOM, 37, 38  
      FDF, 37, 38  
      fid, 36, 37  
      General Binary, 37  
   Images, 35  
   Import Working Directories in Batch, 40  
   Import Working Directory, 40  
   Load Images, 36, 37, 59  
   Load Working Directory, 35  
   Multi-Column Text, 37, 38  
   Save Working Directory, 35, 39  
Siemens IMA, 37, 38  
Single-Column Text, 38  
Spectroscopic Fid, 35  
Test Data, 35, 39  
Text k-space, 36  
Text k-space fid, 37  
User Manual, 35, 39  
Help, 24  
Packages, 22, 24, 33, 40  
Settings, 46  
   Add Server, 48  
   Auto Configure Server, 48  
   McMC Parameters, 24, 46, 48  
   Min Annealing Steps, 48, 48  
   Port number, 48  
   Preferences, 49, 63  
   Remove Server, 48, 49  
   Repetitions, 46, 48  
   Server Name, 48  
   Server Setup, 24, 26, 48  
   Set Window Size, 49  
   Simulations, 46, 48  
   View Server Installation Info, 48, 49  
Spectroscopy fid, 36  
Utilities, 24, 50  
   Memory Monitor, 50  
   Software Updates, 50  
   System Information, 50  
WorkDir  
   Creating, 22, 33, 46  
   Deleting, 22, 33, 46  
   List, 24, 46  
   Loading, 46  
   Name, 46  
   Popup, 47  
Model Comparison  
   Big Peak/Little Peak Package, 211  
model orthonormal definition, 349  
Mouse  
   Control-left, 59  
   Fid Data Viewer  
      Left, 56  
      Right, 56  
   Shift-left, 59  
Multiplets  
   J-Coupling
INDEX

Center, 159
Primary, 159
Secondary, 159

Newton-Raphson, 171
Noise Standard Deviation, 64
Non-Linear Phasing Package, 405
Calculations, 407
Model Equation, 405, 407
Widgets
Process, 409
Write Ascii images, 409
Write imaginary images, 409
Nuisance Parameter, 100, 115, 135
Nyquist Critical Frequency, 111, 127

orthonormal, 349
Outliers, 475
Mean Parameter, 477
Model, 475
Prob Number of, 476
Proposal, 475
Red dot, 477
Weighted Average, 477

Packages
Analyze Image Pixel Unique, 423
Bayes Analyze, 20, 43, 57, 155, 200
Bayes Find Resonances, 21, 239
Bayes Test Data, 427
Behrens-Fisher, 21, 44, 311
Big Magnetization Transfer, 20, 43, 259
Big Peak/Little Peak, 20, 43, 207
Binned Density Function Estimation, 355
Binned Histograms, 21, 44
Diffusion Tensors, 20, 40, 247
Enter ASCII Model, 42
Enter Ascii Model, 20, 329
Enter ASCII Model Selection, 42
Enter Ascii Model Selection, 20, 341
Errors In Variables, 21, 44, 303
Find Resonances, 43
Given Exponential, 20, 40, 137
Given Polynomial Order, 285
Image Model Selection, 415
Image Pixel, 21, 45, 411
Image Pixel Model Selection, 22, 45
Inversion Recovery, 20, 40, 151
Kernel Density Function, 361
Linear Phasing, 21, 44, 395
Magnetization Transfer, 20, 42, 265
Magnetization Transfer Kinetics, 20, 43, 275
Maximum Entropy Method Of Moments, 21, 44, 373
Metabolic Analysis, 21, 43, 219
Non-Linear Image Phasing, 21, 45, 405
Polynomials
of Given Order, 21, 44
of Unknown Order, 21, 44
Test ASCII Model, 42
Test Ascii Model, 20, 337
Unknown Number of Exponentials, 20, 40, 143
Unknown Polynomial Order, 293
Parameter File, 42
Number Of
Abscissa, 458
Data Columns, 458
Model Vectors, 458
Priors, 458
Prior Probability, 459
Amplitude, 460
High, 459
Low, 459
Mean, 459
NonLinear, 460
Ordered, 460
Parameter File, 459
Peak, 459
Prior Type, 460
Standard Deviation, 459
Phase Cycling, 162
Plot Results Viewer, 71
Plots
Data and Model, 81
Data, Model and Residuals, 81
Expected Log Likelihood, 88
Logarithm of the Posterior Probability, 91
Maximum Entropy Histogram, 84
Maximum Entropy Histograms, 83
McMC Samples, 83, 85
Parameter Vs Posterior Probability, 86, 87
INDEX

Test Ascii Model Package, 337
  Reports
    Bayes Accepted, 339
    Mcmc Values, 339
  Using, 339, 428
Viewers
  Ascii Data, 337
  Fortran/C Models, 337
  Prior Probabilities, 337
Widgets
  Build, 337
  Find Outliers, 339
  System, 337
  User, 337
Thermodynamic Integration, 445, 449

Uninstall, 49
Unknown Number of Exponentials Package, 143
  Bayesian Calculations, 145
  Model Equation, 145
  Reports
    Bayes Accepted, 143, 148
    Condensed, 148
    Console Log, 148, 149
    McMC Values, 148
    Prob Model, 148
  Using, 143
  Viewers
    File, 148
    Plot Results, 149, 150
    Prior, 143
    Text, 148
Widgets
  Constant, 143
  Find Outliers, 143
  Order, 143
Unknown Polynomial Order Package, 293
  Bayesian Calculations, 295
Files
  Ascii, 293
  Bayes Analyze, 293
  Peak Pick, 293
Model Equation, 295
Number of data sets, 293
Reports
  Bayes Accepted, 293, 299
Condensed, 299
Console Log, 298, 299
McMC Values, 299
Polynomial Order Plot , 301
Prob Model, 299
Using, 293
Viewers
  File, 299
  Text, 299
Widgets
  Set Order, 293, 294
  Unknown Order, 293, 294

Viewers, 27, 52
ASCII Data, 36
Ascii Data, 27, 53, 56, 63, 137, 265, 275, 285, 293, 311, 329, 337, 341
  Expanding Plot, 53
  Printing, 53
  Right click, 53
Bayes Model, 160
Fid Data, 27, 265
fid Data, 53, 56, 285, 293
  Auto Range, 59
  Autoscale, 56
  Clear Cursors, 56
  Clear Data, 57
  Copy, 59
  Cursor, 56
  Data Info, 57
  Expand, 56
  fn, 57
  Full, 56
  Get Peak, 56
  Phase Popup, 57
  Print, 59
  Properties, 59
  Referencing, 59
  Save As, 57, 59
  Set Preference, 57
  Units, 59
  Zoom, 59
Fid Model, 27
fid Model, 68, 186
  Build BA Model, 70, 159
  Data, 71
Horizontal, 71
Model, 71
Overlay, 71
Report, 71
Residual, 71
Stacked, 71
Trace, 71
Vertical, 71
File, 28, 80
Fortran/C Models, 93, 330
Image, 27, 59, 415
  Autoset Grayscale, 61
  Copy Selected, 62
  Delete All, 61
  Delete Selected, 61
  Display Full, 61
  Element Selection, 60
  Export, 62
  Get Statistics, 64, 65
  Get Threshold Statistics, 65
  Grayscale, 63
  Image Selection, 60
  List, 59
  Load Selected Pixels, 61
  Max, 64
  Mean, 64
  Min, 64
  Right Click, 61
  RMS, 64
  Save Displayed, 62
  Save Statistics, 65
  Sdev, 64
  Set Image Area, 62
  Show Histogram, 61
  Show Info, 62
  Slice, 62
  Slice Selection, 60
  Statistics, 60
  Value, 64
  View Selected Pixels, 61
  Viewer Settings, 62
  Viewing, 62
  X Pos, 64
  Y Pos, 64
Plot Results, 28, 71
Prior, 27, 65
Prior Probabilities, 138, 312
Text, 141, 271, 281, 290, 309, 322, 335, 353
Text Results, 26, 28, 52, 74
  Bayes Analyze, 176
Widgets
  Analyze Image Pixel Package
    Build, 411
    Find Outliers, 411
    Get Statistics, 413
    System, 411
    User, 411
  Analyze Image Pixel Unique Package
    Build, 423
    Find Outliers, 423
    Get Statistics, 425
    System, 423
    User, 423
  Ascii Data Viewer
    Delete, 53
    Left-mouse, 53
    Right-mouse, 53
  Bayes Analyze Package
    By, 158, 176
    First Point, 163
    From, 158, 176
    Imag Offset, 163
    Mark, 159
    Max New Res, 157
    New, 159
    Noise, 158
    Phase, 157
    Primary, 158
    Real Offset, 163
    Remove, 159
    Remove All, 159
    Reset, 159, 193
    Restore, 159
    Save, 159
    Secondary, 159
    Shim Order, 157, 163
    Signal, 158
    To, 158, 176
  Bayes Find Resonances Package
    Build FID Model, 240, 241, 246
    Constant, 239, 242
INDEX

First Trace, 239
Last Trace, 239
Model Fid Number, 241
Phase Model, 239, 242
Bayes Metabolite Package
Fid Model, 221
Fid Model Viewer, 221
Load System Metabolite File, 219
Load System Resonance File, 221
Load User Metabolite File, 219
Load User Resonance File, 221
Shift Left, 221, 222
Shift Right, 221, 222
Bayes Test Data Package
# Images, 427
# Slices, 427
Abscissa, 427
ArrayDim, 427
Build, 427
Get Job, 428
Max Value, 427
Noise SD, 427
Pe, 427
Ro, 427
Run, 428
Set (server), 428
Status, 428
System, 427
User, 427
Big Magnetization Transfer Package
Find Outliers, 259
Big Peak/Little Peak Package
Metabolite, 207
Solvent, 207
Diffusion Tensor Package
Abscissa Options, 248
Find Outliers, 247
Include Constant, 247, 248, 255
Tensor Number, 247, 248, 255
Use b Matrix, 255
Use b Vectors, 254, 255
Use g Vectors, 254
Enter Ascii Model Package
Find Outliers, 329
System, 329
User, 329
Enter Ascii Model Selection Package
Find Outliers, 341
System, 341
User, 341
Errors In Variables Package
Given Errors In, 303
Order, 303
Fid Data Viewer
Autoscale, 56
Clear Cursors, 56
Cursor A, 56
Cursor B, 56
Delta, 56
Display Type, 56
Expand, 56
Full, 56
Get Peak, 56
Left-mouse, 56
Options, 57, 59
Right-mouse, 56
Trace, 70
Fortran/C Model Viewer
Abscissa Spinner, 93
Add Prior, 96
Allow/Disallow Editing, 97
Cancel and Exit, 96
Changing Models, 94
Code, 93, 94
Compile Results, 97
Compiling, 96
Create/Edit Model, 93
Data Columns Spinner, 93
Derived, 96
Edit/Create New Model, 93, 94
High, 97
Low, 97
Mean, 97
Model, 96
Model Vectors, 93
Name (parameter), 97
Order, 97
Parameter Type, 97
Parameters button, 93, 94, 96
Prior Type, 97
Priors, 96
Remove All (priors), 96
Remove Prior, 96
Remove Selected Model, 93
Save and Load, 96
Standard Deviation, 97

Given Exponential Package
Constant, 137, 139
Find Outliers, 137
Order, 137, 139

Given Polynomial Order Package
Set Order, 285

Global
Bayes Find Outliers, 27
Cancel, 26, 51
Edit Servers, 26
Get Job, 26, 51, 137, 143, 151, 155, 209, 221, 241, 247, 259, 267, 277, 285, 293, 305, 311, 329, 337, 343, 357, 364, 373, 413, 415, 425, 428
Reset, 27
Restore Analysis, 22
Save, 27
Set (server), 26, 52, 137, 143, 151, 155, 207, 221, 239, 247, 259, 265, 277, 285, 293, 305, 311, 329, 337, 343, 355, 364, 373, 375, 413, 415, 425, 428

Image Model Selection Package
System, 415
User, 415

Image Viewer
Element Number, 62
Get Statistics, 64
Get Threshold Statistics, 65
Grayscale, 63
Save Statistics, 65
Slice Number, 62
Value, 64
X Pos, 64
Y Pos, 64

Inversion Recovery Package
Find Outliers, 151

Kernel Density Function Package
Kernel Type, 364
Output Size, 364

Linear Phasing Package
cf, 403
Display, 403
Display Array Element, 403
fn, 403
fn1, 403
Image Type, 402
Load An Image, 402
np, 403
nv, 403
Process, 403

Magnetization Transfer Kinetics Package
Load, 275, 281
Set, 275
Uncertainty, 275

Magnetization Transfer Package
Find Outliers, 265

MaxEnt Density Function Estimation Package
Histogram Size, 373
Order, 373

Non-Linear Phasing Package
Process, 409
Write Ascii images, 409
Write imaginary images, 409

Prior Viewer
High, 65
Low, 65
Mean, 65
Prior Type, 67

Server
Edit, 52
Name, 26, 52, 52
Set (server), 48
Setup, 48, 52

Test Ascii Model Package
Find Outliers, 339
System, 337
User, 337

Text Results Viewer
Copy, 74
INDEX

Down arrow, 74
Enable Editing, 74
Print, 74
Save (a copy), 74
Save As, 74
Settings, 74
Up arrow, 74

Unknown Number of Exponentials Package
  Constant, 143
  Find Outliers, 143
  Order, 143

Unknown Polynomial Order Package
  Set Order, 293, 294
  Unknown Order, 293, 294

WorkDir
  Creating, 22, 33, 46
  Deleting, 22, 33, 46
  List, 24, 46
  Loading, 46
  Name, 46
  Popup, 47