BayMEM - A computer program for application of the Maximum Entropy Method in reconstructions of electron densities in arbitrary dimension.

User Manual

Lukáš Palatinus
Sander van Smaalen

version: 05/01/2005

Contents

Chapter 1

Introduction

The Maximum Entropy Method (MEM) is a versatile statistical method for reconstruction of images of virtually any type. One of its applications is the reconstruction of the electron density distributions from the X-ray diffraction data. As a special case, combination of the MEM and superspace approach offers new possibilities for studies of the modulation functions of modulated structures.

BayMEM is a computer program that has been developed for applications of the MEM in charge-density reconstructions of both ordinary and modulated crystal structures.

This manual is intended to provide practical guide to the usage of BayMEM; it will not focus on the theory of the MEM and on details of different algorithms and types of MEM available in the program. The reader can find the theoretical information in special literature: The basic foundations of the MEM are described in Jaynes (1996). A collection of articles encompassing the wide variety of applications of the MEM in science was is compiled in von der Linden et al. (1998). The various applications of the MEM to the crystallographical problems are described in a review article by Gilmore (1996). The description of the Sakata-Sato algorithm is given in Sakata & Sato (1990). The Cambridge algorithm was first published by Skilling & Bryan (1984). The commercial set of subroutines MEMSys5, that implements the Cambridge algorithm and that BayMEM provides interface with, has its own extensive user manual (Gull & Skilling, 1999*b*). The first version of BayMEM has been described in a PhD. thesis by Schneider (2001). Further developments of BayMEM are described in a PhD. thesis by Palatinus (2003). An article on the theory of MEM in superspace with description of BayMEM and examples of application was published by van Smaalen et al. (2003).

A variety of methods exist, that can enhance the performance of the MEM. Many of them are available in BayMEM. Among them is the concept of static weighting (De Vries et al., 1994), the generalized F-constraints (Palatinus & van Smaalen, 2002), the two-channel entropy formalism (Papoular et al., 1996), and the prior-derived F-constraints (Palatinus & van Smaalen, 2003).

Chapter 2

Basic operation of BayMEM

The operation of BayMEM can be considered to be split into following steps.

1. Reading the data: The data from the input file is read and checked for consistency. Dynamic arrays for holding the data are allocated. The format of the input file is described in Chapter 5.

2. Initializing the MEM iteration: Before start of the iteration, three essential steps are necessary. First, the prior density is created or read from an external file. Second, the symmetry of the pixel grid is analyzed and the asymmetric unit is found. And finally, the set of unique reflections given in the input file is expanded into the whole sphere to facilitate Fast Fourier Transform performed later during the iteration.

3. The iteration: The heart of the program. More about the iteration process will be described in following subsections. BayMEM can work with two different MEM algorithms. The Sakata-Sato algorithm (Sakata & Sato, 1990) has been implemented as a part of the code of BayMEM. The MEMSys5 set of subroutines, that implements the Cambridge algorithm (Skilling & Bryan, 1984), is commercial and must be purchased separately (Gull & Skilling, 1999*a*). BayMEM provides interface with the MEMSys5 package. Following chapter discusses issues specific for the two algorithms.

4. Writing the output: After the iteration has been finished, BayMEM writes out the electron density and other files containing information about the MEM calculation. The output is described in Chapter 6.

These four steps are performed automatically by the program. The task of the user is just to prepare the input file.

Chapter 3

Algorithms

3.1 Sakata-Sato algorithm

The Sakata-Sato algorithm is based on an approximate solution of the MEM equation (Sakata & Sato (1990), Kumazawa et al. (1995), **??**). The crucial point in the performance of the algorithm is selection of the value of the Lagrange multiplier λ. In the zeroth-order single-pixel approximation used by the Sakata-Sato algorithm the value of λ is not critical for the convergence of the algorithm, unless it is too large. Too large values lead to divergence of the algorithm. On the other hand, too small values of λ only decrease the speed of convergence. Selection of the value of λ is user's responsibility and does not follow from the theory.

BayMEM offer two modes of handling λ. The first mode is the fixed-λ mode. The user selects the value of λ at the beginning of the iteration and the value is fixed during the iteration. If divergence is encountered in this mode, BayMEM terminates. The second mode is called the automatic λ-control. The starting value of λ is increased by an arbitrary factor f_i every cycle (currently $f_i = 1.1$). If divergence is encountered, λ is decreased by a factor f_d (currently $f_d = 0.75$) and the cycle is repeated. At the same time, the factor f_i is lowered (currently $f_i^{new} = (f_i^{old} + 1)/2$), so that the increments of lambda are not so large in following cycles.

In the majority of cases, the automatic λ-control gives the best possible performance of the Sakata-Sato algorithm. However, in some exceptional cases the automatic λ-control fails and divergence occurs, that cannot be avoided by any decreasing of λ. In those cases, the only solution is to turn-off the automatic λ-control and set λ to a fixed value. See description of the keyword `settings` (Section 5.3) for information on how to select different λ-modes.

The Sakata-Sato algorithm works well in most cases. However, its convergence is not guaranteed. Sometimes, the speed of the convergence becomes so low, that the calculation must be stopped before the final value is reached. If the difference between the current value of the constraint and the desired stopping value is not large, the problem is usually not dramatic, because the electron density changes only very little in the last stages of the iteration. Note also, the some problems with convergence are not due to the algorithm, but due to inconsistencies in the input data. See Chapter 8 for description of possible problems.

3.2 MemSys5 package

MemSys5 package is a general MEM system applicable to any MEM problem, not only crystallographic. The package is commercial and must be purchased separately (Gull & Skilling, 1999*a*). The interface with MemSys5 is provided as a part of BayMEM. The interface is written so as to avoid modifications to the code of MemSys5 package as much as possible. However, modifications changes could not be avoided entirely. These changes

have to be made in the source code of the MemSys5 package; without them BayMEM will not work with MemSys5 properly. The changes are listed here. The line numbers refer to the version 1.2 of MemSys5 package, released on September 6^{th}, 1999.

- All declarations of the floating-point numbers should be changed from REAL to DOUBLE PRECISION. This can be done by replacing all occurrences of text `'REAL '` with text `'DOUBLE PRECISION '` (note the ending spaces!) in files memsys5.for, vector.for and memsys.inc.

- file memsys5.for, line 896: replace code

  ```
  IF((METHD1.LT.1).OR.(4.LT.METHD1)) STOP ' Illegal METHOD(1) value'
  ```

  by code

  ```
  IF((METHD1.LT.1).OR.(5.LT.METHD1)) STOP ' Illegal METHOD(1) value'
  ```

- file memsys5.for, line 3409: Between lines 3409 and 3410:

  ```
  CALL MENT4(ST,DEF, PS,PGRADS,PSUM)
  END IF
  ```

  this code must be inserted:

  ```
  ELSEIF (METHD1.EQ.5) THEN
  CALL MENT5(ST,DEF, PS,PGRADS,PSUM)
  ```

- file memsys5.for, subroutine MENT1: Replace the code of subroutine MENT1 (between lines 3418 and 3440) by this code:

```
      SUBROUTINE MENT1(ST,DEF,S,GS,SUM)
* One block of standard entropy
      IMPLICIT CHARACTER (A-Z)
      DOUBLE PRECISION ST(0:*),DEF,S,GS,SUM
      DOUBLE PRECISION ZERO,A,C
      PARAMETER (ZERO=0.0D0)
      IF (DEF.GT.ZERO) THEN
        CALL MFILL(ST,2,DEF)
        CALL MMUL(ST,2,4,2)
        CALL MSUM(ST,2,A)
        CALL MDIV (ST,1,4,2)
        CALL MEXP(ST,2,2)
        CALL MSMUL(ST,2,DEF,2)
      ELSE
        CALL MMUL(ST,3,4,2)
        CALL MSUM(ST,2,A)
        CALL MDIV (ST,1,4,2)
        CALL MEXP(ST,2,2)
        CALL MMUL(ST,2,3,2)
      ENDIF
      CALL MDOT(ST,2,4,SUM)
      C=A/SUM
      CALL MSMUL(ST,2,C,2)
      CALL MDOT(ST,2,1,C)
      S=SUM-A-C
      CALL MMUL(ST,1,1,1)
      CALL MDIV(ST,2,4,2)
      CALL MSUM(ST,2,SUM)
```

```
      CALL MDOT(ST,2,1,GS)
      CALL MSQRT(ST,2,1)
      CALL MMUL(ST,2,4,2)
      END
```

- file memsys5.for, line 3441: Insert code of subroutine MENT5 here:

```
      SUBROUTINE MENT5(ST,DEF,S,GS,SUM)
* One block of standard entropy without normalization
      IMPLICIT CHARACTER (A-Z)
      DOUBLE PRECISION ST(0:*),DEF,S,GS,SUM
      DOUBLE PRECISION ZERO,EPS,A,C
      INTEGER I
      PARAMETER (ZERO=0.0D0,EPS=1.0D-13)
      IF (DEF.GT.ZERO) THEN
        CALL MFILL(ST,2,DEF)
        CALL MMUL(ST,2,4,2)
        CALL MSUM(ST,2,A)
        CALL MDIV (ST,1,4,2)
        CALL MEXP(ST,2,2)
        CALL MSMUL(ST,2,DEF,2)
      ELSE
        CALL MMUL(ST,3,4,2)
        CALL MSUM(ST,2,A)
        CALL MDIV (ST,1,4,2)
        CALL MEXP(ST,2,2)
        CALL MMUL(ST,2,3,2)
      ENDIF
      CALL MDOT(ST,2,4,SUM)
      CALL MDOT(ST,2,1,C)
      S=SUM-A-C
      CALL MMUL(ST,1,1,1)
      CALL MDIV(ST,2,4,2)
      CALL MSUM(ST,2,SUM)
      CALL MDOT(ST,2,1,GS)
      CALL MSQRT(ST,2,1)
      CALL MMUL(ST,2,4,2)
      END
```

- file memsys5.for, lines 3661-3663: Replace code

```
5     CALL MSUB(ST,21,25,24)
      CALL MMUL(ST,24,22,24)
      CALL MDOT(ST,24,24,PLHOOD)
```

  with code

```
5     CALL MSUB(ST,21,25,24)
      CALL MMUL(ST,24,22,24)
      CALL MMUL(ST,24,31,28)
      CALL MDOT(ST,28,28,PLHOOD)
```

Following changes are not necessary for proper performance of the program, but they remove some unnecessary operations on the data and thus speed up the operation of the program:

- file vector.for, lines 540-563: the original code between lines 540 and 563:

```
      IF (MORE.EQ.0) THEN
* Initialise and count disc buffers if using dynamic block sizes .....
        MORE=1
        NBUF=0

                ⋮

*   held on disc
          CALL VSTACK(1,KCORE)
          IF (ACTION) CALL UFETCH(ST,KCORE,KB(J)+IOFF,LENGTH)
        ENDIF
      ENDIF
```

  should be replaced by:

```
       KCORE=KB(J)
       LENGTH=KL(J)
```

- file vector.for, lines 567-589: the original code between lines 567 and 589:

```
* If using dynamic block sizes instead of fixed LBLOCK ..............
      IF (MORE.EQ.1) THEN
* Calculate block size, and set up stack
        LBLOCK=KL(J)
        IF (NBUF.GT.0) THEN
          LBLOCK=MIN(LBLOCK,LWORK/NBUF)

                ⋮

*   if held on disc
          IF (ACTION) CALL USTORE(ST,KCORE,KB(J)+IOFF,LENGTH)
          CALL VSTACK(-1,KCORE)
        ENDIF
      ENDIF
* Any more elements?
      IF (IOFF+LENGTH.GE.KL(J)) MORE=0
```

  should be completely removed.

Operation of MemSys5 is extensively described in the MemSys5 user manual (cite). The Cambridge algorithm guaranties convergence to the proper MaxEnt solution under normal circumstances. If the computation with Cambridge algorithm does not converge, the reason is usually in the input data and the data should be checked for errors (see Section 8). However, the MemSys5 set of subroutines is complex and relatively rigid, and therefore difficult to adapt to non-standard problems. For this reason, some variations of the constraints are not implemented in BayMEM with the Cambridge algorithm. This concerns G-constraints and the generalized F-constraints. These constraints work only with the Sakata-Sato algorithm.

Important: MemSys5, version 1.2, does not have a built-in option for normalization of the MEM distribution, although this option is described in the user manual. Therefore, the normalization must be achieved by adding the F(000) structure factor to the dataset with value equal to the number of electron given in the input file. For details on handling the F(000) see description of the keyword `fbegin - endf` (Section 5.3).

Chapter 4

Technical details

4.1 Programming language and system requirements

The program BayMEM is written in the programming language Fortran 90. It has been compiled and tested on two computers:

- Compaq AlphaStation ES40 with 500MHz 64-bit Alpha EV6 RISC processor and with Compaq Fortran Compiler V5.5-1877-48BBF
- Silicon Graphics Fuel with 500MHz IP35 MIPS R14000 processor and with MIPSPro Fortran compiler V7.4

The program obeys Fortran 90 standards and should therefore be compilable with any F90 compiler. The program does not have any special system requirements. It does not use graphical interface and the input can be edited with any plain text editor such as vi, nedit or emacs. However, it should be noted that the requirements for the RAM are quite high, in order of GB for large problems.

4.2 Execution

The program is executed with command

`BayMEM input_filebase [ncycles]`

where `input_filebase` is the name of the ASCII file containing input parameters (see sections 5.2, 5.3 and 5.4) without the extension `.BayMEM`. The extension `.BayMEM` is automatically added by the program; every input file must have this extension. If the input_filebase is omitted, the program will prompt for it interactively. The optional parameter `ncycles` defines the maximal number of MEM iterations. After BayMEM performs `ncycles` iterations, it stops regardless of the degree of convergence of the job. If `ncycles` is omitted, the value `MAXCYCLES` from the module GlobalDefinitions in file Variables_mod.f90 is used (currently `MAXCYCLES` = 100000).

Chapter 5

Specification of the input

5.1 Types of input

There are two types of input. The basic input is the ASCII input file: it contains all the necessary parameters of the BayMEM run. In following sections, the expression "input file" means always the ASCII input file. The second type is a file containing the reference electron density (prior). This input is used only if the keyword `inputdensity` has another value than `flat`. Currently, BayMEM supports electron density files in three different formats (see section 6.1). The file with the reference electron density is referred to as a "prior density file" or "prior density" in following text.

5.2 Format of the ASCII input file

The input file is a free-format file based on keywords. Each keyword represents a specific parameter of the MEM calculation and must be given a value.

Multiple spaces anywhere in the file are handled as a single space. If the sign `'#'` or `'!'` occurs anywhere in the line, the rest of the line after this sign is treated as comment and not interpreted. Blank lines anywhere in the input file are ignored. The length of the interpreted part of the line is 132 characters, any text exceeding this length is ignored.

5.3 Specification of keywords

There are two basic types of keywords. The first type is followed by one or more values on the same line:

```
keyword value1 [value2 value3...]
```

The second type has the form:

```
initial_keyword
line 1
line 2
...
final_keyword
```

Each line may contain one or more values.

The name of the keyword of the first type is a single word without spaces. The name of the keyword of the second type is a pair of initial and final word (separated by a hyphen in the following text).

Each value can be a constant of type real, integer or character. The type of the parameters and their allowed values are specified. Alternative values are separated by slashes.

The keywords are either compulsory or optional. The compulsory keywords must be specified for the analysis to proceed. The optional keywords can be omitted. If an optional keyword is omitted, the default value is used. Compulsory keywords are indicated by "compulsory keyword - no default value" in the item "default".

The item "description" describes the function of the keyword, its influence on the output and relations to other keywords.

5.3.1 name: 2channel

- value: yes/no
- default: no
- description: Activates or deactivates the two-channel entropy formalism (Papoular et al., 1996). That allows to reconstruct the maps with both positive and negative densities, for example the difference electron densities.

5.3.2 name: algorithm

- value: S-S/MEMSys [+ optional algorithm-specific settings]
- default: compulsory keyword - no default
- description: This keyword selects one of the two algorithms presently available in BayMEM. S-S selects the Sakata-Sato algorithm, MemSys selects the Cambridge algorithm implemented in the MEMSys5 package. Each algorithm has its own specific settings.

  For `algorithm MEMSys` the settings are:

  method: Integer. 4 for the "historical" maximum entropy ($\chi^2 = N$), 1, 2 and 3 for different variants of the "Bayesian" maximum entropy (Gull & Skilling, 1999*b*). All choices are possible, but only method 4 has been extensively tested and the other methods did not prove to be useful in crystallographic problems.

  NRAND: Integer. Number of random vectors used in the calculation of conjugate gradient in the MEMSys package. NRAND=1 is safe for the vast majority of cases. For more details see Gull & Skilling (1999*b*).

  aim: Real number. The stopping criterion $aim = C_{final}$, where C is the value of constraint. Usually 1.0. Lower values mean closer fit of the MEM density to the experimental structure factors.

  RATE: Real number. Sets the user-definable factor, that influences the sizes of the steps along the conjugate-gradient vector. For more details see (Gull & Skilling, 1999*b*). Usually between 1.0 and 5.0 at the beginning of the iteration. `RATE` can be increased interactively during the iteration (see Section 7.2).

  internal accuracy: Real number. Defines the internal accuracy in the calculation of the conjugate gradient. The recommended value is 0.05. The precise value is not crucial for the performance of the algorithm. Too small values do not improve the accuracy, but slow down the iteration.

  For `algorithm S-S` the settings are:

  lambda: Real. The initial estimate of the lagrange multiplier λ. λ is always positive. If the parameter lambda is given negative, the absolute value is taken and fixed, e.g. BayMEM will operate in the fixed-λ mode (Section 3.1). If the string AUTO (case

sensitive) occurs instead of a number, BayMEM will estimate the starting value of λ automatically.

aim: Real. The stopping criterion $aim = C_{final}$, where C is the value of constraint. Usually 1.0. Lower values mean closer fit of the MEM density to the experimental structure factors.

The settings after the specification of the algorithm can be omitted. In that case the default settings are used. The defaults are:

`algorithm S-S` ∼ `algorithm S-S AUTO 1.0` `algorithm MEMSys` ∼ `algorithm MEMSys 4 1 1.0 1.0 0.05`

5.3.3 name: cell

- value: $a\ b\ c\ \alpha\ \beta\ \gamma$
- default: compulsory keyword - no default
- description: Lattice parameters of the structure.

5.3.4 name: centers - endcenters

- value: Each line contains one centering vector.
- default: no centering vectors
- description: Defines the centering vectors of the (super)space group. The dimension of the centering vectors must correspond to the dimension of the structure defined by the keyword `dimension`. The components can be given both as fractions and as fractional number.

5.3.5 name: centro

- value: yes/no
- default: compulsory keyword - no default
- description: The value `yes` corresponds to a centrosymmetric structure. The value of `centro` must be consistent with the symmetry operators given in keyword `symmetry - endsymmetry`!

5.3.6 name: conorder

- value: even positive integer
- default: 2
- description: Defines the order of the generalized F-constraint (Palatinus & van Smaalen, 2002). The generalized F-constraint of order n is defined as:

$$C_{F_n} = -1 + \frac{1}{m_n(Gauss)} \frac{1}{N_F} \sum_{i=1}^{N_F} \left(\frac{|F^i_{obs}(\vec{H}) - F^i_{calc}(\vec{H})|}{\sigma^i(\vec{H})} \right)^n \tag{5.1}$$

$m_n(Gauss)$ is the value of the n^{th} central moment of the Gaussian distribution. The generalized F-constraint is implemented only in the S-S algorithm. $n = 2$ corresponds to the standard χ^2 constraint. If `conorder` other than `2` is combined with the MEMSys algorithm, the program terminates with an error message.

5.3.7 name: conweight

- value: Hn or Fn, n is a number between -50 and 50
- default: $n = 0 \sim$ no weighting
- description: Static weighting according to De Vries et al. (1994). The F-constraint with the static weighting is defined as:

$$C_w = -1 + \frac{1}{N_F} \sum_{i=1}^{N_F} w(F_{obs}) \left(\frac{|F^i_{obs}(\vec{H}) - F^i_{calc}(\vec{H})|}{\sigma^i(\vec{H})} \right)^2 \tag{5.2}$$

The weighted G-constraints can be defined analogically. Weighting factor is defined as $w = 1/|\vec{H}|^n$ or $w = |\vec{F}|^n$, depending on the value of the keyword `conweight`. $|\vec{H}|$ is the length of the diffraction vector and $|\vec{F}|$ is the amplitude of the structure factor of every reflection. The power n can be any number, but numbers between 2 and 5 proved to be the most efficient. The weighting on $|\vec{F}|$ is not applicable to G-constraints, because the separate intensities of reflections in one group are not known.

5.3.8 name: correction

- value: none/normalize/cut/flat/raise
- default: none
- description: The prior density used in BayMEM must bepositive everywhere, with exception of the two-channel entropy method. If the prior density does not fulfil this requirement, BayMEM offers several possibilities to make the prior density positive everywhere. The meaning of the values is:

  none: No correction.

  normalize: The density is normalized to the number of electrons given in the input file - $\rho_i^{corr} = \frac{N_{el}}{\sum \rho_i^{uncorr}} \rho_i^{uncorr}$.

  cut: If the minimum of the density is negative, then all pixels with $\rho_i < \text{abs}(\rho_{min})$ are assigned the value of ρ_{min}.

  flat: Values less than zero are set to value corresponding to an equally distributed density $\rho_{eq} = \frac{N_{el} V_{UC}}{N_{pix}}$.

  raise: If the minimum of the density is negative, then the whole density is raised by $1.5 * \text{abs}(\rho_{min})$.

  The charge of the prior density must be equal to the total charge of the resulting ρ_{MEM}. Therefore, the prior densities obtained by the corrections `cut`, `flat` or `raise` are subsequently normalized to the number of electrons given in the input file.

5.3.9 name: dimension

- value: positive integer
- default: compulsory keyword - no default
- description: This keyword defines the dimension of the structure. Dimension must be larger or equal to the value of `realdimension`. Apart from this restriction, the dimension is arbitrary. However, for dimensions larger that the parameter MAXDIM in the module Globaldefinitions the parameter MAXDIM must be changed and the program recompiled. The current value of MAXDIM is 8.

5.3.10 name: electrons

- value: real number
- default: compulsory keyword - no default
- description: Gives the number of electrons in the unit cell. The resulting electron density will be normalized to this number of electrons. It the value F(000) is given in the list of structure factors (see keyword `fbegin - endf`, it must be equal to the value of `electrons`. Zero or negative value of the number of electrons is possible only with two-channel entropy (see keyword `2channel`).

5.3.11 name: expandedlog

- value: yes/no
- default: no
- description: If the value is yes, the log-file will contain the full list of all symmetry-expanded reflections. If the value is no, only the symmetry independent reflections are listed.

5.3.12 name: extra - endextra

- value: each line contains indices of one reflection with optional expected value of the structure factors of that reflection.
- default: no extra reflection
- description: The maximum entropy method is able to estimate the values of the structure factors that have not been used in the MaxEnt optimization. The estimated values of the structure factors of reflections given in the `extra - endextra` list are written to the output file `jobname.BMout` at the end of the calculation. If an expected value of structure factor is given, this value is also written in the output file and the expected and estimated values can be directly compared.

5.3.13 name: fbegin - endf

- value: each line contains the indices, real and imaginary part of the structure factor and ats estimated standard uncertainty of one reflection from the input dataset.
- default: compulsory keyword - no default
- description: This keyword serves for definition of the input data. The format of each line is free, the order is {indices A B σ(—F—)}. A and B are the real and imaginary components of the structure factor. Number of indices must be consistent with the dimension of the structure defined in keyword `dimension`. F(000) must be present in the data set (Section 3.2). If F(000) is present as the first structure factor in the input data, its value and sigma are read by BayMEM, otherwise BayMEM adds this structure factor automatically, and any later occurrence of F(000) will be reported as doubled reflection. The standard deviation of F(000) will be set to one third of the smallest standard uncertainty found in the input data. F(000) is not included in calculation of the starting and final R-values and values of the F-constraint, but it is included in the value of χ^2 reported by MemSys.

5.3.14 name: file

- value: valid specification of filename without extension
- default: The first command-line argument to BayMEM at start
- description: The user can change the base of all output files generated by BayMEM. The default base is the base of the input file (filename without the extension .BayMEM).

5.3.15 name: gbegin - endg

- value: one or more groups of intensities (see keyword `ggroup`)
- default: no G-group
- description: The so called G-groups are groups of two or more reflections, where only sum of their intensities in known. Several such groups can be placed between the keywords `gbegin` and `gend`. Each group starts with keyword `ggroup`.

5.3.16 name: ggroup

- value: first line: `ggroup` G $\sigma(G)$, where G is the "group amplitude and $\sigma(G)$ is the standard uncertainty of G; following lines: $ndim$ integers representing the indices of reflections in the G-group.
- default: no ggroups
- description: `ggroup` defines one G-group. The "group amplitude" is defined as:

$$G = \sqrt{\sum_{j=1}^{N_g} \left(\frac{m_j}{\sum m_j} |F_j|^2 \right)} \tag{5.3}$$

N_g is the number of reflections in the G-group, m_j is the multiplicity of the reflection j, $|F_j|$ is the amplitude of the structure factor of reflection j. Each line following the line with `ggroup` contains indices of one reflection in the G-group. The format of the whole ggroup is:

```
gbegin
# first ggroup
ggroup g-amplitude g-sigma
h k l ...
h k l ...

   ⋮

h k l ...
# second ggroup
ggroup g-amplitude g-sigma

   ⋮

endg
```

5.3.17 name: initialdensity

- value: flat/jana/BMascii/BMbinary
- default: compulsory keyword - no default
- description: `initialdensity` defines the type of the prior density τ. If `initialdensity` is set to `flat`, the prior density is assigned a uniform value of electrons/NPix (NPix being the total number of pixels in the unit cell; see keywords `electrons` and `voxel`). Any other `initialdensity` has to be accompanied by specification of the prior density file (see keyword `initialfile`). Formats are: `jana` = single precision .m81 format of the program package JANA2000; `BMascii` = ascii format of BayMEM, transferable between platforms; `BMbinary` = double precision binary format of BayMEM, usually non-transferable between different platforms.

5.3.18 name: initialfile

- value: valid filename shorter or equal in length to 132 characters
- default: compulsory keyword - no default, if `initialdensity` is other than `flat`,
- description: Specifies the file containing the input electron density map. See keyword `initaldensity`.

5.3.19 name: memcheck

- value: positive integer
- default: no memcheck output
- description: The correctness of the MaxEnt solution can be checked by testing the set of equations:

$$\frac{\partial S}{\partial \rho_i^{MEM}} = \lambda \frac{\partial C}{\partial \rho_i^{MEM}} (i = 1, \cdots, N_{pix}^{au}) \tag{5.4}$$

or its equivalent in reciprocal space:

$$\frac{\partial S}{\partial F_j^{MEM}} = \lambda \frac{\partial C}{\partial F_j^{MEM}} (j = 1, \cdots, N_F) \tag{5.5}$$

where S is the entropy of the MEM density and C is the constraint. If the pairs $\frac{\partial S}{\partial \rho_i^{MEM}}$ vs. $\frac{\partial C}{\partial \rho_i^{MEM}}$ or $\frac{\partial S}{\partial F_j^{MEM}}$ vs. $\frac{\partial C}{\partial F_j^{MEM}}$ are plotted, they should be aligned on a straight line. The keyword `memcheck` can be used to produce a file jobname.BMcheck, that contains list of the pairs $\frac{\partial S}{\partial \rho_i^{MEM}}, \frac{\partial C}{\partial \rho_i^{MEM}}$ and $\frac{\partial S}{\partial F_j^{MEM}}$, $\frac{\partial C}{\partial F_j^{MEM}}$. This file can be used to plot the the corresponding graphs. The list of every pixel of the asymmetric unit (eq. 5.4) could be extremely long. Therefore, only some pixels are selected. The number of selected pixels is given by the value of the keyword `memcheck`. The pairs corresponding to the equation in reciprocal space (eq. 5.5) are listed completely.

5.3.20 name: outputfile

- value: a valid filename of a non-existing file
- default: jobname.*ext*, where *ext* is a format-specific extension

- description: Specifies the filename of the output electron density. If the keyword is omitted, then the name of the output file is created from the jobbase (see keyword `file`) and a format-specific extension, which is `.asc` for `outputformat BMascii`, `.raw` for `outputformat BMbinary` and `.m81` for `outputformat jana/janap1` (for description of the formats see Section 6.1). The file must not exist. BayMEM will never overwrite an existing density file. Instead of that, the output density is written to a file named bmmapXX.*ext*, where XX is a serial number starting from 00. A warning is written to the logfile jobname.BMlog. If all hundred files (bmmap00.*ext*-bmmap99.*ext*) exist, the program will try to ask the user for the filename on the terminal. If the terminal, which BayMEM has been started from, does not exist anymore, the program will stop without writing any output density file.

5.3.21 name: outputformat

- value: jana/janap1/BMascii/BMbinary
- default: compulsory keyword - no default
- description: Supported formats of the output density files are: `jana` and `janap1` = single precision .m81 format of the program package JANA2000; `BMascii` = double precision ascii format of BayMEM, transferable between platforms; `BMbinary` = double precision binary format of BayMEM, usually non-transferable between different platforms. The files written with format `jana` contain pixels with coordinates 0...N_i-1 in each direction i. Format janap1 contains all pixels with coordinates 0...N_i in each direction, e.g. with the redundant border of the unit cell. For more description of the formats of the output density see Section 6.1.

5.3.22 name: priorsf

- value: three real numbers, optional positive integer
- default: no adding of prior structure factors
- description: This keyword provides a possibility to include structure factors calculated from the prior density in the input data set (Palatinus & van Smaalen, 2003). The format of the keyword is: `priorsf` $[sin\theta/\lambda]_{min}$ $[sin\theta/\lambda]_{max}$ `sigma [maxindex]`. All structure factors between $[sin\theta/\lambda]_{min}$ and $[sin\theta/\lambda]_{max}$, that are not present in the input data, are calculated from the prior density and added to the data set as so-called "P-constraints". These P-constraints behave exactly like the F-constraints in the MEM iteration, but they are not included in the calculation of χ^2 and therefore do not influence the stopping point of the convergence. The optional parameter `maxindex` tells the program, that only the reflections with the maximal satellite index smaller or equal to the value of `maxindex` will be added to the dataset. Default value of `maxindex` is 0. Currently, maxindex works correctly only for ordinary modulated structures and not for composites, because the definition of a satellite reflection is slightly different in the two cases.

5.3.23 name: qvectors - endqvectors

- value: Each line contains the components of one q-vector. There must be $dim - rdim$ q-vectors between the start- and end-keyword. $rdim$ is the number of real-space dimensions (keyword `realdimension`).
- default: compulsory keyword - no default; not applicable if $dim = rdim$
- description: This keyword contains coordinates of the q-vectors.

5.3.24 name: realdimension

- value: positive integer smaller than or equal to *dim* (see keyword `dimension`)
- default: 3
- description: Defines the dimension of the "real space". Normally this is 3. For some special applications (two-dimensional diffraction on surfaces) other values than 3 can be chosen.

5.3.25 name: regularwidth

- value: *dim* positive real numbers
- default: regularization function not used
- description: BayMEM has the capability of introducing a correlation between the neighboring pixels. The correlation is introduced by convolution of the "density" with a nD normalized Gaussian distribution. More about this topic can be found in (cite Schneider). The Gaussian can have different widths in different directions. The widths w are given as argument to `regularwidth`. The units of w are $d_{min} = \frac{1}{|\vec{H}_{max}|}$, where $|\vec{H}_{max}|$ is the longest reciprocal vector present in the input dataset.

5.3.26 name: spacegroup

- value: text shorter or equal to 132 characters
- default: empty
- description: Symbol of the (super)space group. Currently not used in the program.

5.3.27 name: symmetry - endsymmetry

- value: each line contains definition of one symmetry operator
- default: compulsory keyword - no default
- description: This keyword contains the complete definition of the symmetry with exception of the centering vectors. Each line contains one symmetry operator. The format of the symmetry operators corresponds to the conventions used in the International Tables for Crystallography. If the symmetry operator is $\{R|\tau\}$, then the format of each entry is:

  $\tau_1 + R_{11}\text{x1} + R_{12}\text{x2} + \ldots + R_{1n}\text{x}n \qquad \tau_2 + R_{21}\text{x1} + R_{22}\text{x2} + \ldots + R_{2n}\text{x}n \qquad \ldots$

  Example - n-glide perpendicular to b:

  ```
  1/2+x1 -x2 1/2+x3
  ```

  The translation components can be given either as fractions or decimal numbers.

5.3.28 name: symtable

- value: yes/no
- default: no

- description: BayMEM calculates so called symmetry table at the beginning of each run. This symmetry table contains all information about the symmetry of the pixel grid. This calculation is quite time-consuming. To save computer time in subsequent calculations with the same grid and symmetry, the symmetry table can be written out in a file called jobname.BMsymtb. BayMEM will write the file jobname.BMsymtb only if setting `symtable yes` is present in the input file. If the file jobname.BMsymtb exists, BayMEM reads the symmetry information from that file instead of calculating it. Generally, this option is useful only if many calculations with the same symmetry setting are planned. Note that the .BMsymtb file can be very large, up to several GB for large grids.

  Once the file jobname.BMsymtb exists, BayMEM will read it regardless of the value of `symtable`. This setting influences only writing of the file.

5.3.29 name: title

- value: text shorter or equal to 132 characters
- default: compulsory keyword - no default
- description: The title of the job. It is used in the output files to identify the individual calculations. It is recommended, but not necessary, to change the title with every new calculation.

5.3.30 name: voxel

- value: *dim* positive integers
- default: compulsory keyword - no default
- description: Defines the division of the unit cell. Each number corresponds to the number of pixels along one axis of the (superspace) unit cell. The division must obey the symmetry of the unit cell, i.e. a center of any pixel must be mapped by the symmetry operators onto itself or onto a center some other pixel. This means that for example the numbers of pixels along the directions of the screw axes 2_1,3_1,4_1,6_2 and 6_1 must be multiples of 2, 3, 4, 3 and 6, respectively. Numbers with small prime factors should be preferred, to take the full advantage of the speed of the Fast Fourier Transform. Combinations of powers of two or three are especially favorable. The largest prime factor of all divisions must be smaller than 23.

5.4 Examples of typical input files

5.4.1 Example 1.

This is an example of a simple input file for the calculation of a 3D electron density of a monoclinic crystal using a flat prior density. The Sakata-Sato algorithm is selected, while the alternative setting for the MEMSys algorithm is commented out by hashes (#):

```
title oxalic acid
dimension 3
initialdensity flat
outputfile example1.m81
outputformat jana

algorithm S-S AUTO 1.0
#algorithm MEMSys 4 1 1.0 1.0 0.05

cell 6.1005 3.4999 11.9554 90.0 105.781 90.0
voxel 64 32 128
spacegroup P21/n
```

```
centro yes
electrons 132

symmetry
     x1     x2     x3
 1/2-x1 1/2+x2 1/2-x3
    -x1    -x2    -x3
 1/2+x1 1/2-x2 1/2+x3
endsymmetry

fbegin
   2   0   0    25.3459292     0.0000000     0.2727374
   4   0   0    10.4668808     0.0000000     0.1478727

                      :
                      :

   5   4  -1     4.1934419     0.0000000     0.1320773
   1   5  -1     1.9409568     0.0000000     0.1921479
   2   5  -1    -0.6319270     0.0000000     0.5103144
endf
```

5.4.2 Example 2.

The following input file can be used for determination of accurate charge density of a 4D structure. It uses prior density and prior-derived F-constraints (keyword `priorsf`). The generalized F-constraint of order 4 is used. The input and output formats are BMascii. This format allows easy transfer of the densities between different platforms.

```
title Example 2
dimension 4

initialfile prior_density.asc
initialdensity BMascii
outputfile example2.asc
outputformat BMascii

algorithm MEMSys 4 1 1. 1. 0.05
conorder 4
priorsf 0.89 1.5 0.01 1

cell 7.5281 5.8851 10.437 90. 90. 90.
voxel 128 100 162 32
spacegroup Pnma(a00)0ss
electrons 248
qvectors
0.4794 0. 0.
endqvectors
centro yes
centers
# the 0 0 0 0 centering vector can be omitted, here it is given just for illustration
0.0 0.0 0.0 0.0
endcenters
symmetry
     x1     x2     x3     x4
     x1 1/2-x2     x3 1/2+x4
 1/2+x1     x2 1/2-x3 1/2+x4
 1/2+x1 1/2-x2 1/2-x3     x4

    -x1    -x2    -x3    -x4
    -x1 1/2+x2    -x3 1/2-x4
 1/2-x1    -x2 1/2+x3 1/2-x4
 1/2-x1 1/2+x2 1/2+x3    -x4
endsymmetry

fbegin
   0   0   0   0     248.0000000       0.0000000       0.1000000
   2   0   0   0     -37.4445880       0.0000000       0.2325634
   4   0   0   0       8.2858164       0.0000000       0.0570781

                      :
                      :

   1   2  18   1       0.1344953       0.0000000       0.2110144
   2   2  18  -1      -0.1234955       0.0000000       0.2823512
   2   2  18   0       1.9918284       0.0000000       0.0583801
endf
```

Chapter 6

Description of the output

The most important output of the MEM calculation is the optimized electron density. Apart from this output, there are several other files containing information about the input data, data processing, symmetry, iteration and results. All files except for the density file have names of the form jobname.BM*. If several runs are performed with the same jobname, all the "jobname.BM*" files except for .BMsymtb are appended, not overwritten.

6.1 Electron density

6.1.1 Format BMascii

In this format, the electron density of all pixels in the unit cell is written with six values per line in exponential format, preceded by a four-line header. The pixels are sorted first by increasing first coordinate, then second, then third etc.

The file has this form:

```
dimension real_dimension
pixel_division
lattice parameters a b c alpha beta gama volume
minimum and maximum of the map
density values, six numbers per line (multiple occurrence)
```

⋮

For example:

```
  4   3
     128      100      162       32
   5.41530   12.33200    6.78930   90.00000   90.00000   90.00000     453.40
 1.041386E-02  4.063997E+01
2.348115E-02 2.380756E-02 2.477587E-02 2.641150E-02 2.877340E-02 3.185024E-02
3.561728E-02 4.012891E-02 4.537618E-02 5.122452E-02 5.765039E-02 6.476191E-02
7.245219E-02 8.037046E-02 8.832136E-02 9.632531E-02 1.042432E-01 1.116911E-01
```

⋮

6.1.2 Format BMbinary

This is a double precision binary format of BayMEM. The order of numbers is the same as in the ascii format, only the minimum and maximum of the map is omitted in the binary format. Dimension, real dimension and pixel division is written as integers, all following numbers are written as double precision floating point numbers.

6.1.3 Format jana, janap1

This is the format of the crystallographic software package Jana. Its standard extension is .m81. It stores the electron density in single-precision direct-access binary format. It is beyond the scope of this manual to fully describe this format. The density file written in the format **jana** contains only the pixels in one unit cell ($0...N_i - 1$ in each direction, N_i is the pixel division in the direction i); format **janap1** contains also the border of the unit cell, e. g. pixels with coordinates $0...N_i$ in each direction i.

6.2 File jobname.BMout

This file contains all important information about the input data, data processing and results of the MEM run. It consist of three parts:

- Summary of the input data: This part contains the most important information collected by the program from the input file. Its content is self-explainatory. Following BMout file was produced by the input file in Example 2. (Section 5.4.2:

```
Title: Example 2
Input: example2.BayMEM
Dimension of superspace:   4
"Real" dimensions:         3
Cellparameters (a b c alpha beta gamma Volume):              7.52810   5.88510  10.43700 \\
                                                           90.00000  90.00000  90.00000    462.40
Reciprocal cellparameters (a b c alpha beta gamma Volume):  0.1328356 0.1699206 0.0958130 \\
                                                           90.00000  90.00000  90.00000 2.162644E-03
Pixel (a b c...):                   128   100   162    32
Electrons per unit cell:           248.0000
Type of initial density file:    BMascii
Initial density filename:  prior_density.asc
Initial density correction type: none
Algorithm type is set to MEMSys
The settings are:
    Method: 4
    NRand :  1
    Aim   : .10000D+01
    Rate  : .10000D+01
    Utol  : .50000D-01
Constrained moment order:  2


Spacegroup: Pnma(a00)0ss
Centrosymmetric structure: No Friedel-pairs are computed while expanding reflections!

q-vectors:  1
  1:      0.4794000     0.0000000     0.0000000
Symmetry operations:     8

Symmetry operations:     1 through    3
  1.0  0.0  0.0  0.0   0.000   1.0  0.0  0.0  0.0   0.000   1.0  0.0  0.0  0.0   0.500
  0.0  1.0  0.0  0.0   0.000   0.0 -1.0  0.0  0.0   0.500   0.0  1.0  0.0  0.0   0.000
  0.0  0.0  1.0  0.0   0.000   0.0  0.0  1.0  0.0   0.000   0.0  0.0 -1.0  0.0   0.500
  0.0  0.0  0.0  1.0   0.000   0.0  0.0  0.0  1.0   0.500   0.0  0.0  0.0  1.0   0.500

Symmetry operations:     4 through    6
  1.0  0.0  0.0  0.0   0.500  -1.0  0.0  0.0  0.0   0.000  -1.0  0.0  0.0  0.0   0.000
  0.0 -1.0  0.0  0.0   0.500   0.0 -1.0  0.0  0.0   0.000   0.0  1.0  0.0  0.0   0.500
  0.0  0.0 -1.0  0.0   0.500   0.0  0.0 -1.0  0.0   0.000   0.0  0.0 -1.0  0.0   0.000
  0.0  0.0  0.0  1.0   0.000   0.0  0.0  0.0 -1.0   0.000   0.0  0.0  0.0 -1.0   0.500

Symmetry operations:     7 through    8
 -1.0  0.0  0.0  0.0   0.500  -1.0  0.0  0.0  0.0   0.500
  0.0 -1.0  0.0  0.0   0.000   0.0  1.0  0.0  0.0   0.500
  0.0  0.0  1.0  0.0   0.500   0.0  0.0  1.0  0.0   0.500
  0.0  0.0  0.0 -1.0   0.500   0.0  0.0  0.0 -1.0   0.000

 F-Constraints input/expanded:     3970   30045
 P-Constraints input/expanded:    15270  118968
 G-Constraints input/expanded:        0       0
```

- Information about initial and final status of the iteration: Contains starting and ending time of the iteration, initial and final R-values and related quantities. Part of the

BMout file corresponding to Example 2 is show here. The lines beginning with # are comments and are not present in the file:

```
 Date of iteration start: 30.01.2003 Time: 14:03.58

Initial state:

# Non-weighted and weighted R-value for all reflections, F-constraints and G-constraints

R =  0.9725     Rw =  0.7779
RF=  0.9725     RwF=  0.7779
RG=  0.0000     RwG=  0.0000
Entropy:  0.0000000E+00

#Sum of calculated (mem) and observed (obs) amplitudes of F- and G-constraints

Sum Fmem= 2.480E+02     Sum Gmem= 0.000E+00
Sum Fobs= 9.031E+03     Sum Gobs= 0.000E+00
Sum all F and G mem= 2.480E+02
F-Constraint 2.373E+03     G-Constraint 0.000E+00


 Date of iteration end: 01.02.2003 Time: 09:41.56
 After         38 Cycles of iteration
 Elapsed CPU time:      2628 min 28 sec

Final state:
R =  0.0137     Rw =  0.0159
RF=  0.0137     RwF=  0.0159
RG=  0.0000     RwG=  0.0000
Entropy: -3.0085513E+02
Sum Fmem= 8.914E+03     Sum Gmem= 0.000E+00
Sum Fobs= 9.031E+03     Sum Gobs= 0.000E+00
Sum all F and G mem= 8.914E+03
F-Constraint:  9.964E-01     G-Constraint:  0.000E+00

 F-Constraints input/expanded:     3970   30045


# P-constraints denote the F-constraints calculated from the prior density (see keyword priorsf)

 P-Constraints input/expanded:    15270  118968
 G-Constraints input/expanded:        0       0
```

- List of the reflections: The list contains information about all input reflections and (optional) extra reflection (see keyword `extra - endextra`:

```
F-Constraints:
     #     h   k   l...     A/Gobs             B           A/Gmem             B             DeltaF   \\
                                                                        Sigma       DeltaF/Sigma   sinth/l
     1:    2   0   0   0   -37.4445880     0.0000000      -37.7395357     0.0000000       -0.2949477 \\
                                                                    0.2325634       -1.2682464  0.132835
     2:    4   0   0   0     8.2858164     0.0000000        8.1621250     0.0000000        0.1236914 \\
                                                                    0.0570781        2.1670549  0.265671

                      ⋮
```

All input F-constraints are listed. DeltaF=$|F_{obs}| - |F_{calc}|$, sinth/l denotes $\sin(\theta)/\lambda$. reflections with DeltaF>3sigma and DeltaF>6sigma are marked with # and ! at the end of the line.

6.3 File jobname.BMlog

This is the log-file for all messages and informations produced by BayMEM during its run-time. The file can be separated into several parts:

- report from the reading of the input file: This part contains error messages about missing or doubled compulsory keywords, and information about missing or doubled optional keywords. The part has form:

```
Error!!! Statement either missing or doubled: cell
Error!!! Statement either missing or doubled: voxel
Info: Statement not included or doubled: expandedlog
Info: Statement not included or doubled: file
Info: Statement not included or doubled: conorder

  2 errors found in the input file!
There are   3 infos!
```

If a missing or doubled compulsory keywords are encountered, BayMEM terminates with an error message written to the standard output. It is recommended to always check this part of the BMlog file. Even non-fatal infos can indicate a problem in the input file.

- Summary of the input data: This part is almost identical with the corresponding part of the .BMout file (Subsection 6.2).
- List of all input reflection with components of the structure factor or square-root of the intensity of the G-constraints, amplitude of the structure factor, sigma, assignment to a g-group (zero for F-constraints; see keyword `gbegin - endg`), multiplicity of each reflection, and some other diagnostic information.
- Information about the total number of voxels and number of voxels in asymmetric unit:

```
Voxels in unitcell       :        66355200
Voxels in asymmetric unit:         8294402
```

- A list of reflections is produced, that contains all reflections that are equivalent by symmetry to other reflections in the input file. If such equivalent reflections are found, the program terminates with an error message written to the standard output and to the BMlog file.
- If the prior-derived F-constraints are used, this list will contain also the structure factors calculated from prior density. The format is identical with the format of experimental F-constraints.
- If setting `expandedlog yes` is present in the input file, then the BMlog file will contain a full list of all reflections expanded by the symmetry operators from the input file. This listing is suppressed by default. In the default case, only this summary is printed:

```
Listing of expanded reflections suppressed, summary follows:

 Totals of expanded reflections:
  F-Constraints:    30045
  P-Constraints:   118968
  G-Constraints:        0
  Alltogether  :   149013
```

- Record of the progress of the iteration: The style of this part depends of the type of algorithm in use.

  Sakata - Sato algorithm: A record identical to the record in the BMout file is written to the BMlog file at the beginning of the iteration. If automatic determination of the starting λ is allowed (keyword `settings`), following information occurs in the BMlog file:

  `Automatic calculation of starting lambda:  the value set to 0.3144E-02`

  Following statistics is written into the log file after each cycle:

```
Cycle:        392
Lambda: 0.1308E-01
Test:  0.2313     Charge increase factor:  1.0002
R = 7.596E-03     Rw = 6.674E-03
RF= 7.596E-03     RwF= 6.674E-03
RG= 0.000E+00     RwG= 0.000E+00
Entropy: -3.5944897E-01      Entropy shift: -4.512E-04
       L=-3.6019225E-01
Constrained moment number: 4
Sum Fmem= 4.799E+03     Sum Gmem= 0.000E+00
Sum Fobs= 4.801E+03     Sum Gobs= 0.000E+00
Sum all F and G calc= 4.799E+03
F-constrained moment 5.246E-02     G-constrained moment 0.000E+00
Combined FG Constraint :           0.998     Constraint shift:  5.136E-03
Combined FPG Constraint:           0.057     Constraint shift:  2.891E-04
Aim:                            1.000
FCon(1)= 5.231E-02     FCon(2)= 7.993E-02     GCon(1)= 0.000E+00     GCon(2)= 0.000E+00
FCon(3)= 1.688E-01     FCon(4)= 5.246E-02     GCon(3)= 0.000E+00     GCon(4)= 0.000E+00
FCon(5)= 1.635E+00     FCon(6)= 2.888E-02     GCon(5)= 0.000E+00     GCon(6)= 0.000E+00
FCon(7)= 2.806E+01     FCon(8)= 1.153E-02     GCon(7)= 0.000E+00     GCon(8)= 0.000E+00
```

The majority of the text is self-explanatory or has been explained in Section 6.2. The meaning of the remaining statements is given here:

Lambda: The Lagrange multiplier. For more description see chapter 3.1.

Test: Test=$1 - \cos(\nabla S.\nabla C)/(|\nabla C|.|\nabla S|)$, i.e. one minus the cosine of the angle between the gradients of the entropy and of the constraint. The angle is zero for the ideal MaxEnt solution and consequently the test should be close to zero, too. However, there is no guarantee that the Sakata-Sato algorithm leads to an ideal MaxEnt solution. Therefore, a larger deviation of test from zero does not necessarily indicate a problem in the computation and/or input data. The higher value of test may be caused by the inadequacy of the approximations used in the Sakata-Sato algorithm.

Charge increase factor: Ratio of total charge of the new ρ_{MEM} and ρ_{MEM} of the previous cycle before the new density is normalized. Large values of this factor indicate too large change between the two cycles. If the automatic lambda-control is enabled, the last cycle will automatically be repeated with a decreased value of lambda, if the charge increase factor exceeds 100. If the automatic lambda-control is disabled, only a warning is printed to the BMlog file.

Entropy: The total entropy of the unit cell:

$$S = -\sum_{i=1}^{N_{pix}} \rho_i \ln \frac{\rho_i}{\tau_i} \tag{6.1}$$

L: The total maximized Lagrangian:

$$L = S - \lambda C \tag{6.2}$$

Constrained moment number: Order of the generalized constraint (see keyword `conorder`).

Combined FG constraint: Constraint calculated only from the experimental data present in the input file. If this value becomes smaller than aim (keyword `settings`), the iteration is considered to be converged.

Combined FPG constraint: Constraint calculated from all data including the prior-derived F-constraints (keyword `priorsf`). This quantity is the measure of the convergence of the algorithm. If the constraint shift between the FPG constraints of successive cycles is negative, the calculation is considered to diverge. This line occurs only if the prior-derived F-constraints are used.

FCon(i), GCon(i): Values of the ith moments of the distribution of the normalized residuals. The odd values should remain close to zero, the even values should converge

to one. These values allow the user to estimate the quality of the distribution of the normalized residuals before the end of the iteration.

MEMSys package: Prior to beginning of the iteration, check of consistency of the transformation routines is performed. Output of this check is a real number. If the check is successful, the number is very small, comparable to the numerical accuracy of the calculation. For double precision calculations, the value of check should not exceed 10^{-14}. The result of the check is written to the BMlog file:

```
Result of input data check:  0.3803E-17
```

Important: It is strongly recommended to check this number at the beginning of the calculation, especially if a completely new dataset is used. If this check fails, the calculation will not lead to correct results. For more information see Section 8.

The information logged at each cycle is produced by the MEMSys5 package. User should refer to MEMSys5 manual for description of the output (Gull & Skilling, 1999*b*). Here only the description of the most important values will be given. The standard form of the output is:

```
Iteration     5
  Entropy ===  -1.4524E-01    Test === 0.0214    Chisq ===  7.6539E+03
  Omega   ===   0.260387      dist === 0.2746    Alpha ===  7.3870E+03
  Ntrans  ===     74                             Code  ===  001010
```

The most important indicators are `Test`, `Omega` and `Code`. Test is defined as:

$\text{Test} = 1 - \cos(\nabla S.\nabla C)/(|\nabla C|.|\nabla S|)$

i.e. one minus the cosine of the angle between the gradients of entropy and constraints. Test is zero for the ideal MaxEnt solution. In the MEMSys algorithm, this value is a crucial indicator of the quality of the MaxEnt solution. The value should be low at the end of the iteration, say less than 0.1. Omega is an indicator of the progress of the iteration. Iteration is stopped only if `Omega` is equal to $1.\pm$ internal accuracy (see keyword `settings`). Code is a string of ones and zeroes. Iteration will not stop before all digits in Code are zero. For the meaning of individual positions in `Code` please refer to the MEMSys 5 user manual.

Note, that one cycle of MEMSys iteration is not comparable with one cycle of iteration of Sakata-Sato algorithm. In each iteration cycle of MEMSys several "subcycles" are performed. The number of total direct and inverse Fourier transforms performed during one iteration cycle is given as Ntrans in the last line of the output of each cycle.

6.4 File jobname.BMhst

One of the basic assumptions underlying the principle of the F- and G-constraints is that the noise in the data is distributed randomly with a Gaussian distribution. Thus, the proper solution should produce normalized residuals $(|F_{obs}| - |F_{calc}|)/\sigma(F_{obs})$ that have a Gaussian distribution. The file jobname.BMcheck contains a representation of the histogram of normalized residuals for an easy assessment of the quality of the distribution of the normalized residuals. This is an example of the histogram:

```
Start of histogram
Title: Sample histogram
Prior type: flat
10.06.2003 Time: 09:31.08
Weighting: No
Constrained moment:  2
Rescaled by 100/ 105
```

```
|D-F|/sigma(D): # of appearances : graphical representation
   -4.6    1 |X
   -4.4    0 |
   -4.2    0 |
   -4.0    0 |
   -3.8    0 |
   -3.6    0 |
   -3.4    0 |
   -3.2    0 |\
   -3.0    1 |\
   -2.8    2 |X\
   -2.6    1 |X \
   -2.4    3 |XXX \
   -2.2    2 |XX    \
   -2.0    9 |XXXXXXXXX  \
   -1.8    7 |XXXXXXX        \
   -1.6   20 |XXXXXXXXXXXXXXXXXX    \
   -1.4   34 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\
   -1.2   39 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     \
   -1.0   49 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     \
   -0.8   75 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\XXXXXXX
   -0.6   87 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\XXXXXXXXX
   -0.4   96 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|XXXXXXXXXX
   -0.2  104 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|XXXXXXXXXXXX
    0.0  105 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|XXXXXXXXXXXX
    0.2  101 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|XXXXXXXXXX
    0.4   98 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|XXXXXXXXXXXX
    0.6   79 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/X
    0.8   69 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/XX
    1.0   45 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX         /
    1.2   27 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXX               /
    1.4   31 |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  /
    1.6   17 |XXXXXXXXXXXXXXXXX       /
    1.8   24 |XXXXXXXXXXXXXXXX/XXXXXX
    2.0    8 |XXXXXXXX   /
    2.2    8 |XXXXXXX/
    2.4    4 |XXXX/
    2.6    3 |XX/
    2.8    2 |X/
    3.0    1 |/
    3.2    0 |/
    3.4    0 |
    3.6    2 |XX
    3.8    0 |
    4.0    0 |
    4.2    0 |
    4.4    0 |
    4.6    1 |X
    4.8    1 |X
    5.0    1 |X
    5.2    0 |
    5.4    0 |
    5.6    0 |
    5.8    0 |
    6.0    1 |X
    6.2    0 |
    6.4    0 |
    6.6    0 |
    6.8    0 |
    7.0    0 |
    7.2    0 |
    7.4    0 |
    7.6    0 |
    7.8    0 |
    8.0    0 |
    8.2    0 |
    8.4    0 |
    8.6    0 |
    8.8    1 |X
Statistics:
 #       F           G       Combined
 1   7.397E-01   0.000E+00   7.397E-01
 2   1.007E+00   0.000E+00   1.007E+00
 3   2.443E+00   0.000E+00   2.443E+00
 4   3.552E+00   0.000E+00   3.552E+00
 5   6.807E+01   0.000E+00   6.807E+01
 6   3.451E+01   0.000E+00   3.451E+01
 7   4.244E+03   0.000E+00   4.244E+03
 8   3.434E+02   0.000E+00   3.434E+02
End of histogram
```

The header contains basic information about the MEM run. The time refers to the time of writing of the histogram and is almost equal to the time of writing the output electron density. "Weighting" and "Constrained moment" refer to the static weighting (keyword `conweight` and to the order of the generalized F-constraint (keyword `conorder`), respectively.

The main body of the histogram consists of a representation of the distribution of normalized residuals. The interval between the minimum and maximum normalized residue is divided into intervals of 0.2. The number at the beginning of each line denotes the center of each interval. The next number in line gives the number of normalized residuals falling into that interval. The number of crosses (X) in each line corresponds to number of residuals in each interval, rescaled by a factor given in the header of the histogram, if the number of normalized residuals in any interval exceeds 100. Three types of slashes (\ | /) outline the ideal Gaussian distribution.

After the main body of the histogram, a list of moments of the distribution is printed, separated into contribution of F-constraints and G-constraints. The even moments are normalized so, that the expected value of the ideal Gaussian distribution is 1. Ideal Gaussian value of the odd moments is zero.

6.5 File jobname.BMcheck

This file allows the user to graphically check the quality of the MaxEnt solution. The file is produced only if the keyword `memcheck` is present in the input file. For closer description of the underlying mathematics see description of the keyword `memcheck`. The file has this format:

```
# Algorithm: MEMSys
# direct space gradient
# number   dS/drho       dC/drho   multiplicity
   686  -0.10099E+01   0.26331E-03    8
  1372  -0.10160E+01  -0.17682E-03    8
  2058  -0.10429E+01  -0.45650E-03    8
  2744  -0.10190E+01  -0.53921E-03    8
  3430  -0.10327E+01  -0.22974E-03    8
  4116  -0.10345E+01  -0.21474E-04    8
  4802  -0.10134E+01  -0.34958E-04    8
  5488  -0.96380E+00   0.14663E-03    8
  6174  -0.99438E+00   0.11397E-03    8

                      ⋮

# reciprocal space gradient:
#    j    dS/dF(j)      dC/dF(j)   multiplicity
     1   0.98691E-02  -0.26485E+01    1
     2  -0.19613E-01  -0.32598E+02    2
     3   0.93420E-02   0.21646E+02    2
     4   0.45182E-02   0.10621E+02    2
     5  -0.10662E-02  -0.14466E+01    2
     6  -0.21129E-02  -0.18777E+01    2
     7   0.30789E-03   0.14683E+01    2
     8  -0.46036E-02  -0.11276E+02    2
     9  -0.42390E-02  -0.78690E+01    2
    10  -0.44431E-03  -0.10764E+01    2
    11  -0.11022E-02  -0.37930E+01    2
    12  -0.28685E-03  -0.82155E+00    2

                      ⋮
```

The pixels in the direct-space gradient are sampled so, that the total number of samples corresponds to the number given by the keyword `memcheck`. The list of reflections in the reciprocal-space part is complete. If a chart of second vs. third columns is plotted, the points should form a straight line in case of an ideal MaxEnt solution.

6.6 File jobname.BMsymtb

This is a binary file containing the information about the symmetry of the discrete unit cell. It is intended for internal use in BayMEM. This file is written only if enabled in the input file (keyword `symtable`). If the file jobname.BMsymtb exists, the symmetry information is read from the file and the time-consuming calculation of the symmetry information is not performed. Having this file can be useful, if many runs with identical symmetry settings are planned. However, note that the BMsymtb file can be very large (several GB for more-dimensional unit cells). The file is never deleted by BayMEM.

Chapter 7

Run-time interaction with the program

7.1 Program-to-user communication

Almost all messages from the program are written into the BMlog file (Section 6.3). This is because the MaxEnt runs might take very long time and the terminal BayMEM has been run from might have been closed before the MaxEnt run has finished. Therefore, user should always check the BMlog file for possible warnings and error messages, especially in the initial stages of the runs. There are few exceptions, when an error message is written both to the BMlog file and to the standard output. These exceptions concern problems encountered during the reading of data, e.g. at the very beginning of the run.

7.2 User-to-program communication

All information necessary to run BayMEM is given in the input file. However, there is a limited set of commands, that can be passed to the program during its run-time. The commands must be written in the file jobname.BMcom. This file is checked by the program and if some known command is found, it is performed and the file is deleted. No multiple commands are allowed in the BMcom file. The commands are case-sensitive. The most convenient way to pass a command to BayMEM is to use command echo:

```
$ echo "command" > jobname.BMcom
```

The allowed commands are:

- `DENSITY` *filename*: If this command is found, the current ρ_{MEM} is written to the file specified in the command and the iteration continues.
- `STOP`: If this command is found, the iteration is stopped, like if the maximal allowed number of cycles were exceeded. The output density and all output files are written.
- `RATE` *value*: This command applies only to the algorithm `MEMSys`. It allows the user to change the value of parameter `RATE` specified originally in the keyword `settings`. Higher values of `RATE` speed up the convergence, but too high values can cause failure of the algorithm. For more information on this problem see Section 8. The information about the change of RATE is written in the BMlog file.

Chapter 8

Troubleshooting

- **Problem:** BayMEM terminates very quickly without writing any output density.

  Solution: Check the jobname.BMlog file for possible reports on errors in the data, like some missing or mistyped keywords or non-unique set of reflections.

- **Problem:** The 'input data check' in the file jobname.BMlog returns too large value.

  Solution: This indicates some inconsistency in the input parameters or data. Most probable reason is that the symmetry operators do not form a space group. Check also, whether the keyword `centro` is consistent with the symmetry operators. All symmetry operators of the space group must be listed, including those related by the center of symmetry. Another possibility is that the list of reflections in the input file contains reflections, that are systematically extinct. Such reflections must not be present in the input list! Such a situation can occur, if the reflections have been indexed in other symmetry (or only other setting of the space group) that the symmetry operators refer to.

- **Problem:** The Cambridge algorithm (MemSys5 package) converges smoothly up to certain value of omega, but then oscillates around that value without reaching the final value omega=1.

  Solution: This usually indicates inconsistency in the number of electrons given in various places. The total number of electrons given by the keyword `electrons` must be consistent with the value of F(000) in the input reflection list and - if applicable - with the number of electrons in the prior electron density. BayMEM does not automatically normalize the prior electron density to the expected number of electron, it is user's responsibility to supply consistent prior density.

- **Problem:** The Cambridge algorithm converges very slowly, the change in Omega between cycles is very small.

  Solution: If none of the previously described problems applies, you may have chosen too low value of RATE. Try to increase it either in the input file (keyword `settings`) or during the iteration (recommended; see Section 7.2). In general, more symmetrical structures can have higher RATE, up to 15 or 20 in extreme cases. **Important:** Do not increase RATE too much, better increase it by small amount several times. Watch the value of Test. If Test starts to increase, do not increase Rate further. Increasing Rate too much can (and probably will) lead to serious problems with convergence.

  If the value of RATE is already high and the convergence is still slow, consider increasing the standard uncertainty of the F(000) structure factor (keyword `fbegin - endf`. But do not increase it too much, or you end up with a problem described below!

- **Problem:** The Cambridge algorithm converges, but the value of Test is high.

  Solution: The two most probable reasons for this problem are too large value of RATE or too large value of $\sigma(F(000))$. Try to decrease them and repeat the calculation. Under special circumstances (with very informative prior and/or inaccurate data), the condition on the value of the constraint is reached sooner than the proper MEM path is found. This can be sometimes also solved by decreasing RATE, or by decreasing the internal accuracy (keyword `settings`.

- **Problem:** The Sakata-Sato algorithm converges very slowly, virtually stops converging.

  Solution: The convergence of Sakata-Sato algorithm is not guaranteed. However, slow convergence can be caused by various errors in the input file. It is recommended to try calculation with the Cambridge algorithm, if possible. If the Cambridge algorithm converges, the problem is in the Sakata-Sato algorithm. If none of the two algorithms converge, the problem is in the data. In that case (or if the Cambridge algorithm is not available), try to see solutions of previous problems.

Bibliography

De Vries, R. Y., Briels, W. J. & Feil, D. (1994), 'Novel treatment of the experimental data in the application of the maximum-entropy method to the determination of the electron-density distribution from x-ray experiments', *Acta Crystallogr. A* **50**, 383–391.

Gilmore, C. J. (1996), 'Maximum entropy and bayesian statistics in crystallography: a review of practical applications', *Acta Crystallogr. A* **52**, 561–589.

Gull, S. F. & Skilling, J. (1999*a*), MEMSYS5 *v1.2 program package, September 6, 1999*, Maximum Entropy Data Consultants Ltd., Suffolk, U.K.

Gull, S. F. & Skilling, J. (1999*b*), *Quatified Maximum Entropy,* MEMSYS5 *Users' Manual*, Maximum Entropy Data Consultants Ltd., Suffolk, U.K.

Jaynes, E. T. (1996), *Probability theory: The Logic of Science.*, http://www.bayes/wustl.edu/etj/postscript, Fragmentary Edition.

Kumazawa, S., Takata, M. & Sakata, M. (1995), 'On the single-pixel aprroximation in maximum-entropy analysis', *Acta Crystallogr. A* **51**, 47–53.

Palatinus, L. (2003), *Ph.D. thesis*, University of Bayreuth, Bayreuth, Germany.

Palatinus, L. & van Smaalen, S. (2002), 'The generalized f-constraint in the maximum entropy method – a study on simulated data', *Acta Crystallogr. A* **accepted**.

Palatinus, L. & van Smaalen, S. (2003), 'The prior-derived f-constraint: Suppressing the artefacting in the maximum entropy method.', *Acta Crystallogr. A* **in preparation**.

Papoular, R. J., Vekhter, Y. & Coppens, P. (1996), 'The two-channel maximum-entropy method applied to the charge density of a molecular crystal: α-glycine', *Acta Crystallogr. A* **52**, 397–407.

Sakata, M. & Sato, M. (1990), 'Accurate structure analysis by the maximum-entropy method', *Acta Crystallogr. A* **46**, 263–270.

Schneider, M. (2001), *Ph.D. thesis*, University of Bayreuth, Bayreuth, Germany.

Skilling, J. & Bryan, R. K. (1984), 'Maximum entropy image reconstruction: general algorithm', *Mon. Not. R. Astr. Soc.* **211**, 111–124.

van Smaalen, S., Palatinus, L. & Schneider, M. (2003), 'Maximum entropy method in superspace', *Acta Crystallogr. A* **??**, ???–???

von der Linden, W., Dose, V., Fisher, R. & Preuss, R., eds (1998), *Maximum Entropy and Bayesian Methods*, Kluwer Academic Publishers, Dordrecht.