



COherent NUclear Scattering from Single crystals

***Software for the evaluation of
Synchrotron Mössbauer Spectra***

Wolfgang Sturhahn

wolfgang@nrixs.net

About CONUSS:

- developed 1983-1986 by E. Gerdau and W. Sturhahn at the University of Hamburg
 - ☆ coherent elastic nuclear and electronic Bragg scattering
 - ☆ explain first NRS experiments (Gerdau et al. PRL 54, 1985)
 - ☆ FORTRAN code implemented on IBM 360 mainframe (MVS-VM)

- improved 1986-today by W. Sturhahn at
 - the University of Hamburg (1986-1993)
 - the ESRF (1992)
 - the APS (1992-2010)
 - ☆ forward scattering (SMS a.k.a. NFS) added in 1991
 - ☆ ported to Sun UNIX in 1992
 - ☆ extended data handling capability (fitting) added in 1996
 - ☆ ported to Linux in 2004

publications related to CONUSS:

W. Sturhahn and E. Gerdau, Phys. Rev. B 49 (1994)

W. Sturhahn, Hyperfine Interact 125 (2000)

CONUSS now supports:

- all Mössbauer isotopes
- forward scattering and Bragg/Laue reflections
- no limitations by sample structure
- combined hyperfine interactions
- distributions of hyperfine fields
- textures
- relaxation effects
- full polarization and directional dependences
- thickness effects
- time spectra (SMS) and energy spectra (trad. Mössbauer spectr.)
- sample combinations
- time, energy, and angle averaging
- sample thickness distributions
- comparison to experimental data including fitting
- flexible assignment and grouping of fit parameters

More on CONUSS:

- has been used for data evaluation in numerous publications
- distributed under GPL, source code public, evaluations traceable
- can be obtained per e-mail from Wolfgang Sturhahn, no charge
- a major upgrade, CONUSS-2.0.0, was released in 2010

- CONUSS-2.0.0
 - ☆ has a simple installation procedure for Unix and Mac OS X
 - ☆ offers all previous capabilities of CONUSS
 - ☆ has enhanced fit capabilities
 - ☆ features run-time graphics
 - ☆ has a new Monte Carlo approach to find start-values, explore the parameter space, and smart parameter optimization

- possible future development
 - ☆ Graphical input file editor
 - ☆ support of grazing incidence geometry

KMCO app screen shot:

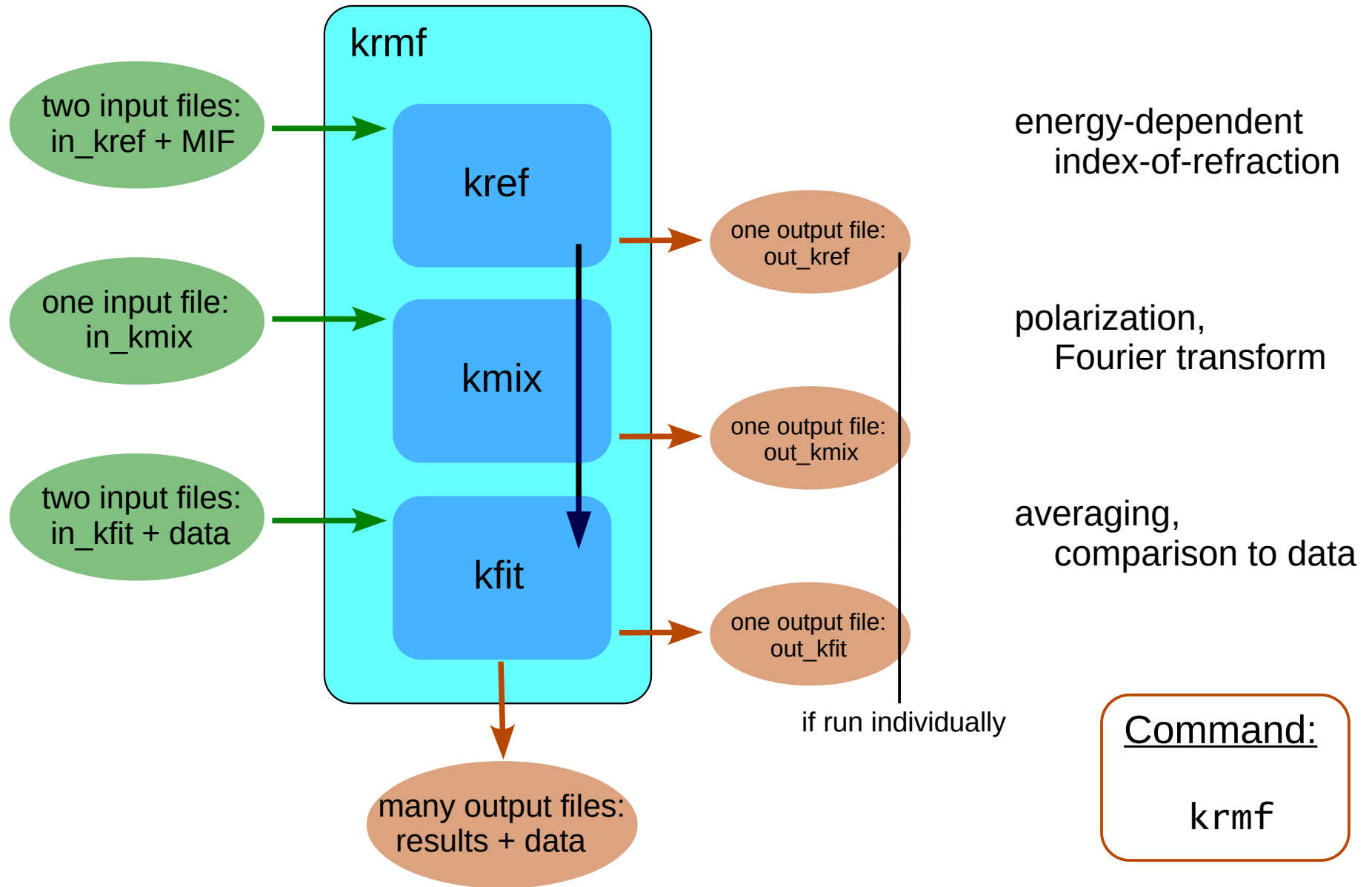
The screenshot displays the KMCO application interface on a Mac OS desktop. The desktop background is a scenic view of a lake and trees. The application windows are as follows:

- Terminal Window (Title: Terminal - xmgrace - 115x54):** Shows the KMCO execution output, including copyright information, estimated success probabilities, and best samplings.
- Plot Window (Title: Grace: Untitled (modified)):** Displays a plot titled "Sampling 1721 :: Chi^2 = 3.683". The y-axis is labeled "counts" and ranges from 10^2 to 10^4 . The x-axis is labeled "time (ns)" and ranges from 0 to 140. The plot shows a noisy signal with a red fit line.
- File Browser Window (Title: /Users/wsturhah/CONUSS-2.0.1/example...):** Shows a list of files and folders, including "fepv_in", "fepv_in.mco", "in_kfit", "in_kmco", "in_kmix", "in_kmix.mco", "in_kref", "in_kref.mco", "mcochi2.hst", "mcoparm.hst", "out_kmco", "parmv2.3D", "parmv2v3.3D", "Perovskite_30GPa.dat10", "parm01.dat", "parm02.dat", "parm03.dat", "parm04.dat", "parm05.dat", "parm06.dat", "parm07.dat", "parm08.dat", and "parm09.dat".
- Emacs Editor Window (Title: emacs: fepv_in):** Shows the fit parameters for the "fepv_in" file. The parameters are listed as follows:

```
* fit parameters
* =====
* the first number gives the center of parameter search box|
* the second number gives the size of the parameter search box
*
% @ &wght21 := 0.25 0.2
% @ &quad21 := 2.48 0.5
@ &iso21 := 0
@ &dist21 := 0
*
% @ &wght22 := 0.25 0.2
% @ &quad22 := 1.53 0.5
@ &iso22 := 0
@ &dist22 := 0.3 0.2
*
% @ &wght3 := 0.5 0.3
% @ &quad3 := 0.6 0.3
% @ &iso3 := -0.7 0.3
% @ &dist3 := 0.45 0.2
*
-----1-----2-----3-----4-----5-----6-----7-->
* MR isfname and MR transition
ISOS----XEmacs: fepv_in (Fundamental PerDel)----L15--C61--4%-----
(No changes need to be saved)
```
- Emacs Editor Window (Title: emacs: in_kmco):** Shows the input and output files for the "in_kmco" file. The files are listed as follows:

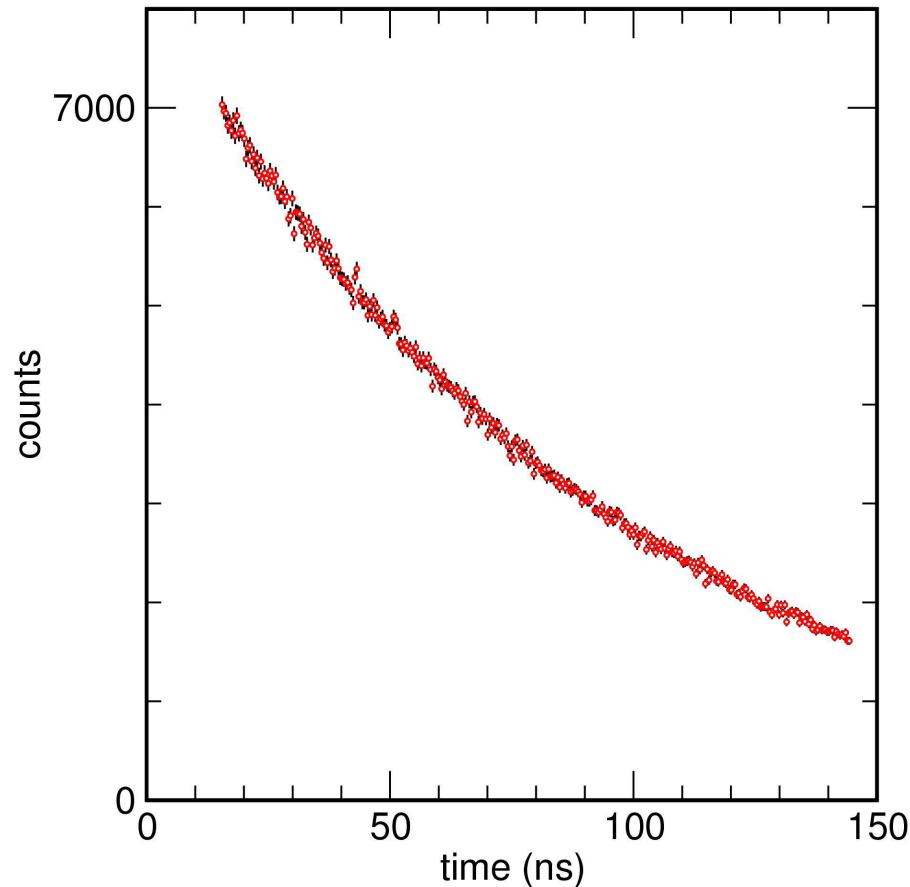
```
* input and output files
* =====
(1) input file for module KREF :: in_kref
(2) input file for module KMIX :: in_kmix
(3) input file for module KFIT :: in_kfit
*
* sampling|
* =====
(4) number of levels :: 10
(5) random samples per level :: 200
(6) search box level-reduction factor :: 0.6
(7) factor for acceptable chi^2 minima :: 3
*
ISOS----XEmacs: in_kmco (Fundamental PerDel)----L37--C10--57%-----
```

Module configuration, theory and simple fit:



SMS example 1.1:

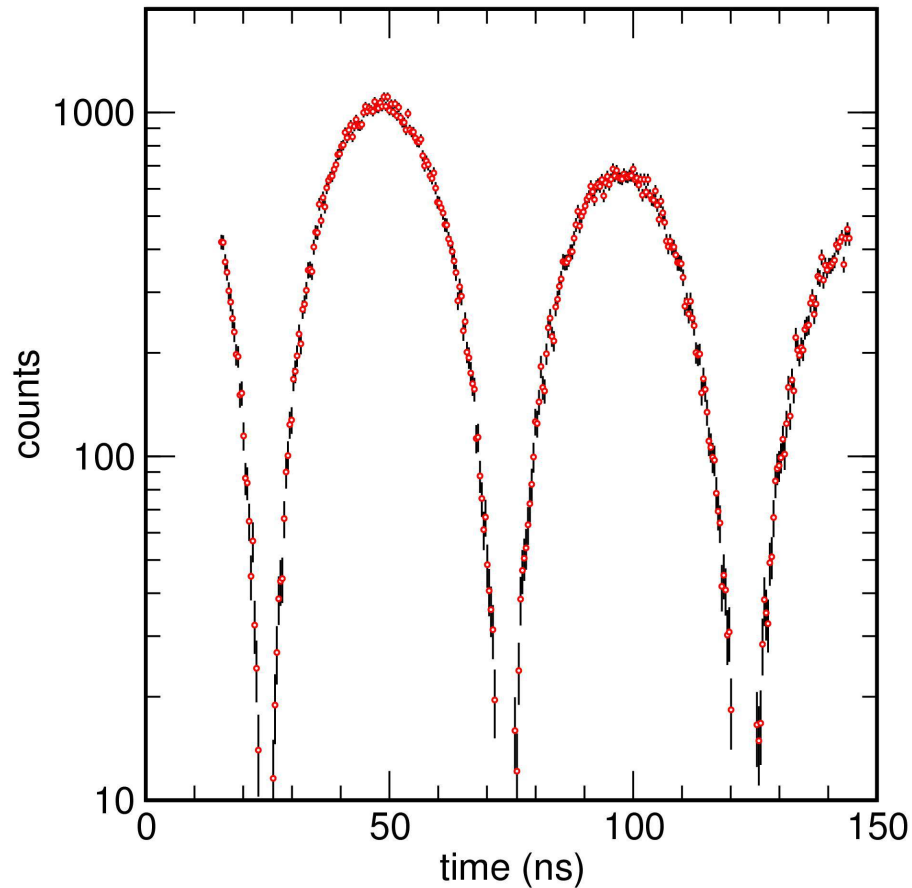
➤ simulate the following SMS spectrum



- ☆ construct the input files
in_kref, in_kmix, in_kfit, ex1.1_in
- ☆ observe the effect of isomer shift,
thickness, quadrupole splitting
- ☆ Tips: watch correlations

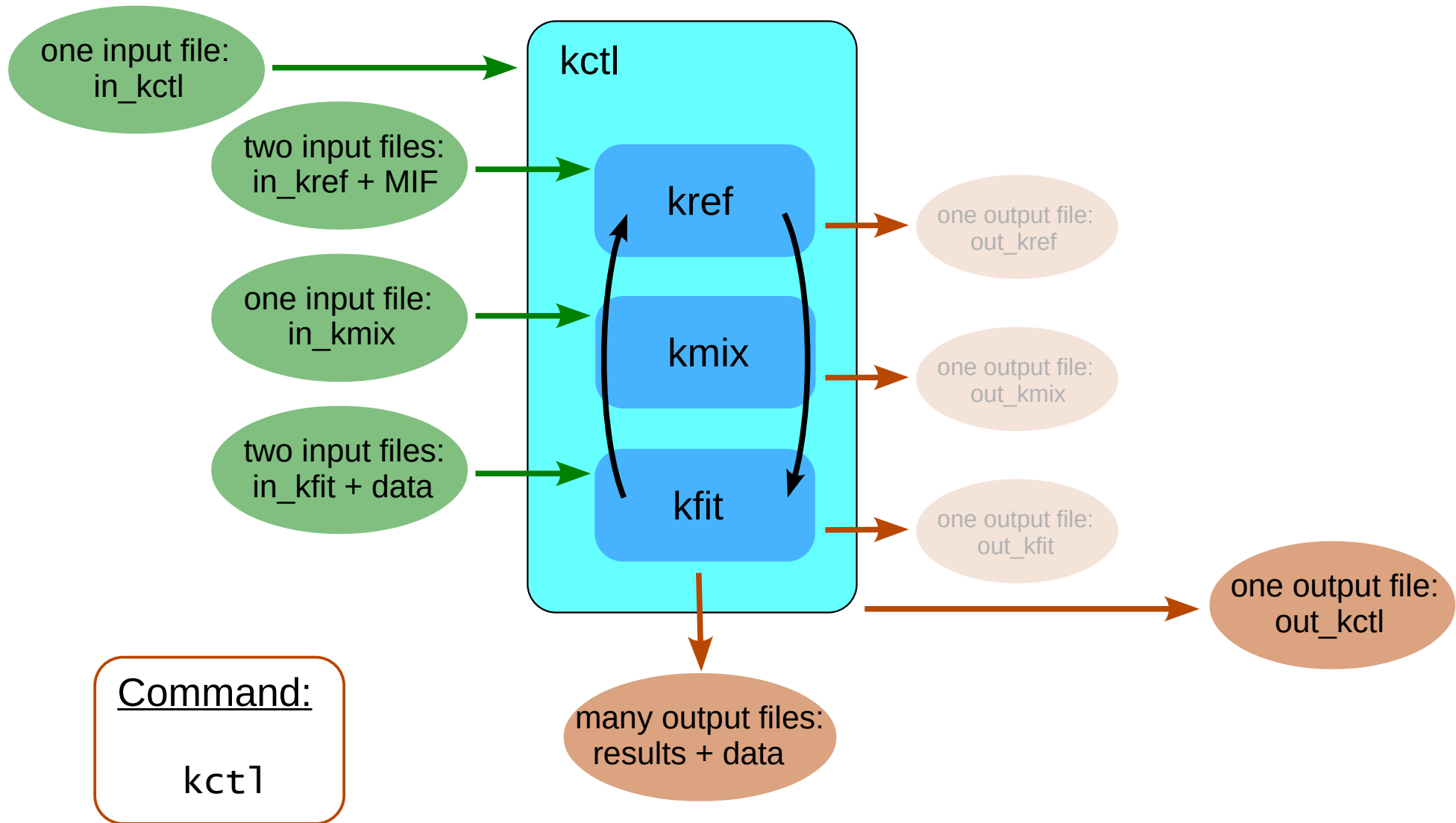
SMS example 2.1:

➤ simulate the following SMS spectrum



- ☆ construct the input files
in_kref, in_kmix, in_kfit, ex2.1_in
- ☆ observe the effect of thickness,
quadrupole splitting
- ☆ Tips: watch correlations

Module configuration, general fitting:



Fitting of SMS spectra:

➤ strategy

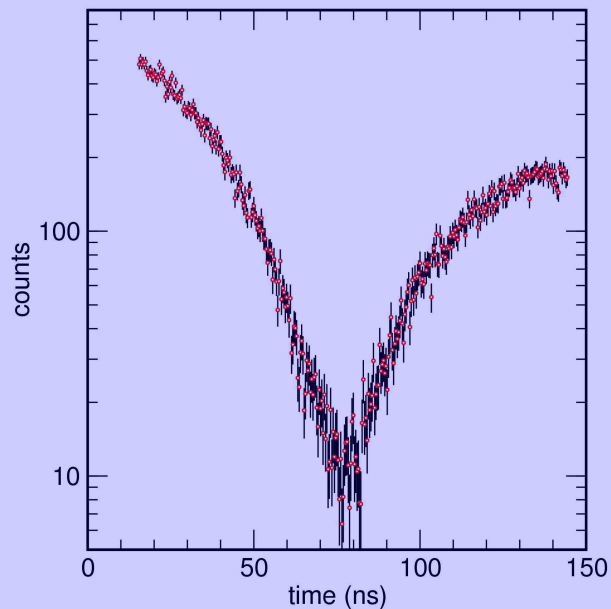
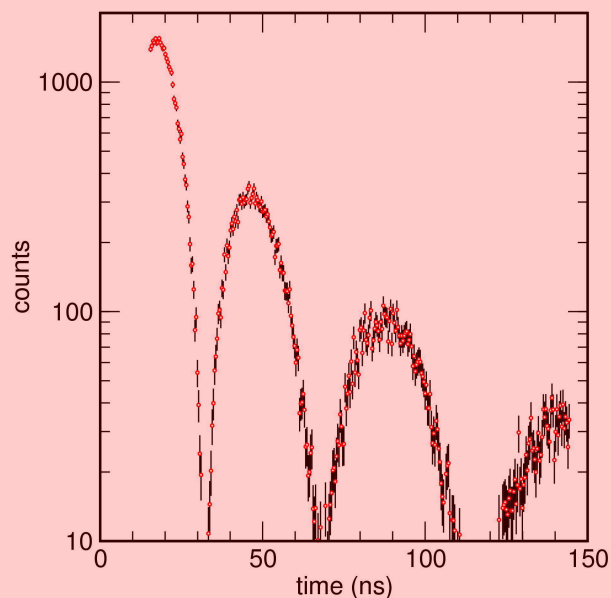
- ☆ identify relevant parameters
- ☆ find start values using command `krmf`
- ☆ optimize parameter values using `kctl`

➤ examples 1.2-4, 2.1-3, and 3.1-3

- ☆ construct the input files `in_kref`, `in_kmix`, `in_kfit`, `ex_in`, `in_kctl`
- ☆ focus on isomer shift, thickness, quadrupole splitting

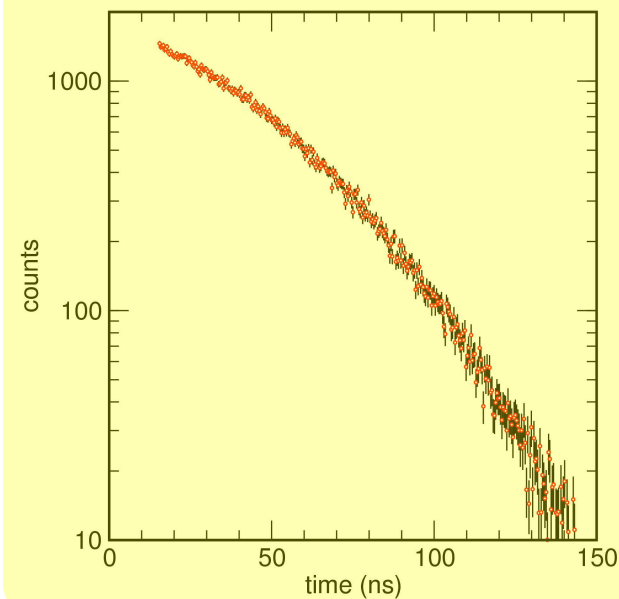
SMS examples:

- example 1.2
focus on thickness



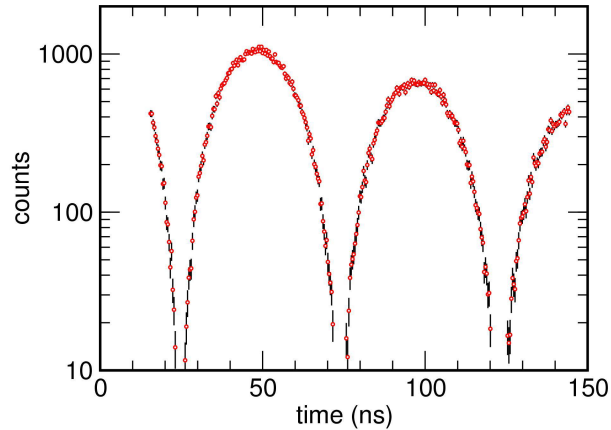
- example 1.3
two sites; isomer shift;
thickness $0.1\mu\text{m}$

- example 1.4
IS distribution;
thickness $0.1\mu\text{m}$

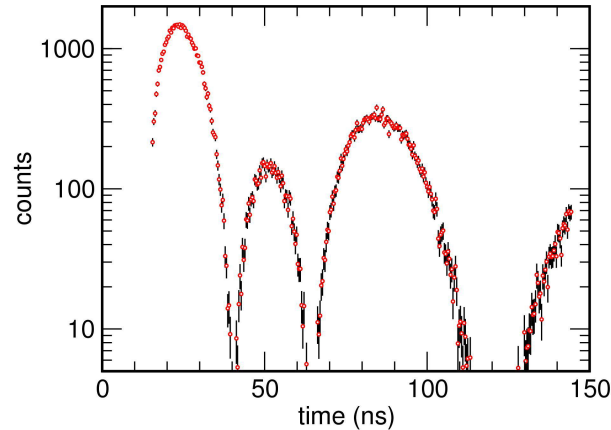


SMS examples, quadrupole splitting, isomer shift:

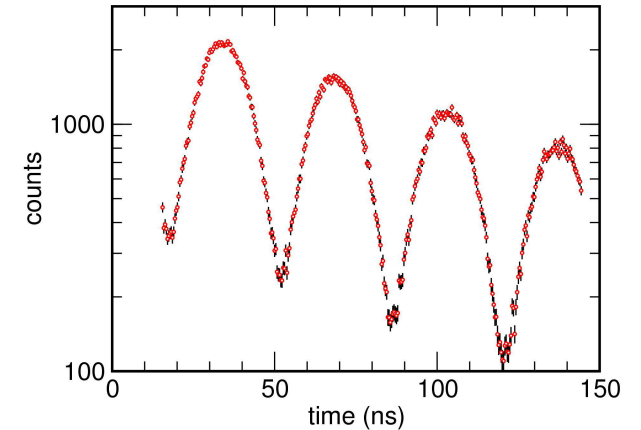
➤ example 2.1
thickness $0.1\mu\text{m}$



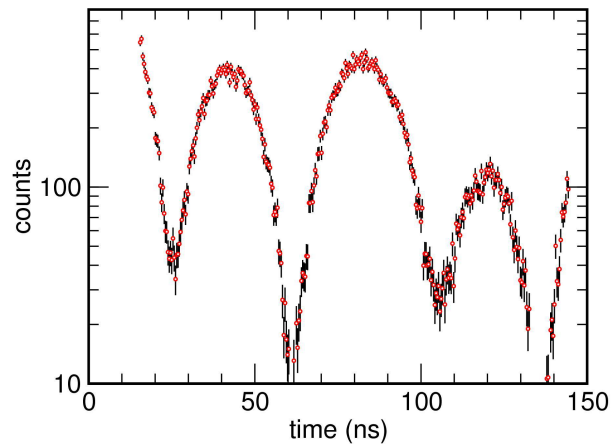
➤ example 2.2



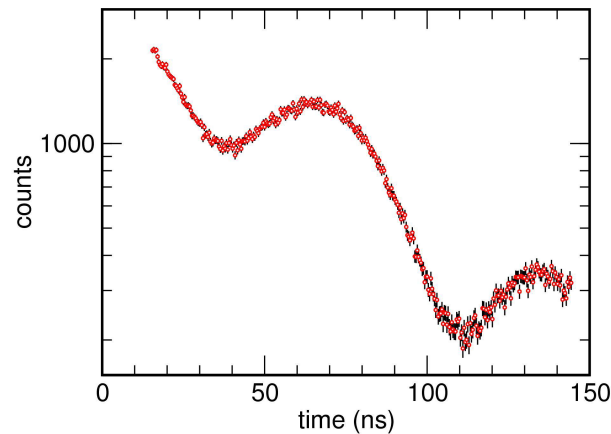
➤ example 2.3
thickness $0.1\mu\text{m}$; texture



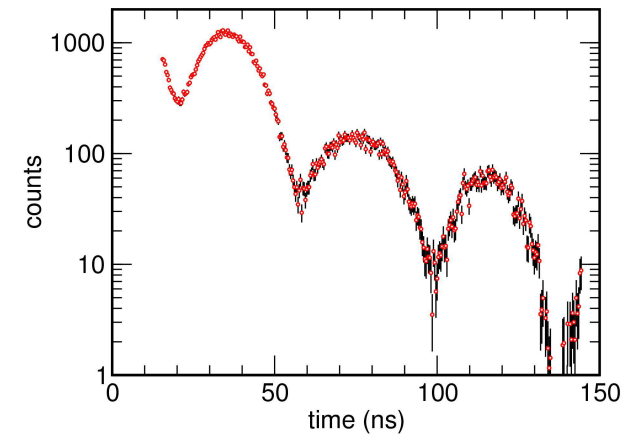
➤ example 3.1
 $0.1\mu\text{m}$; two sites



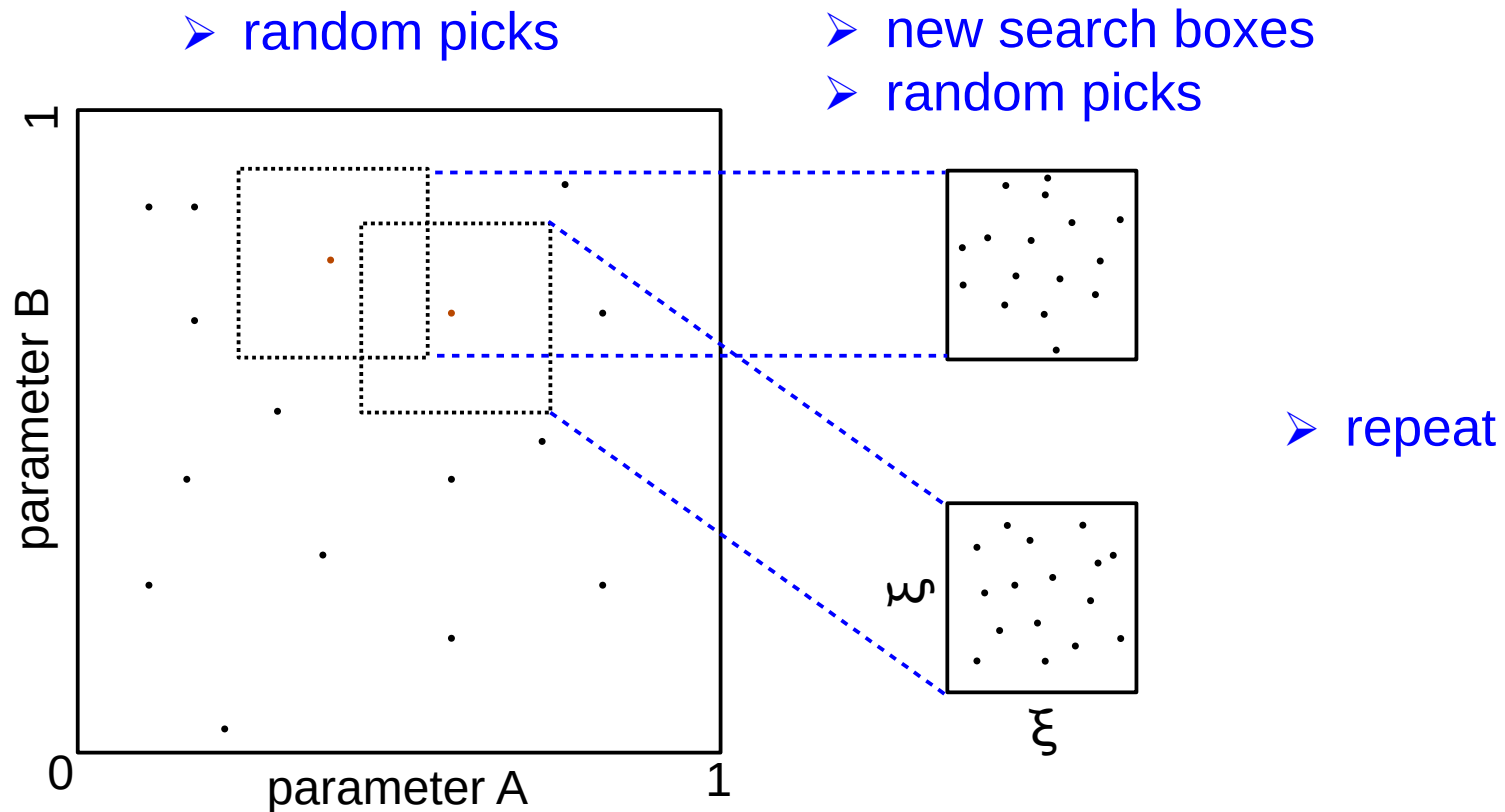
➤ example 3.2
 $0.1\mu\text{m}$; two sites



➤ example 3.3
 $0.1\mu\text{m}$; two sites; distr.

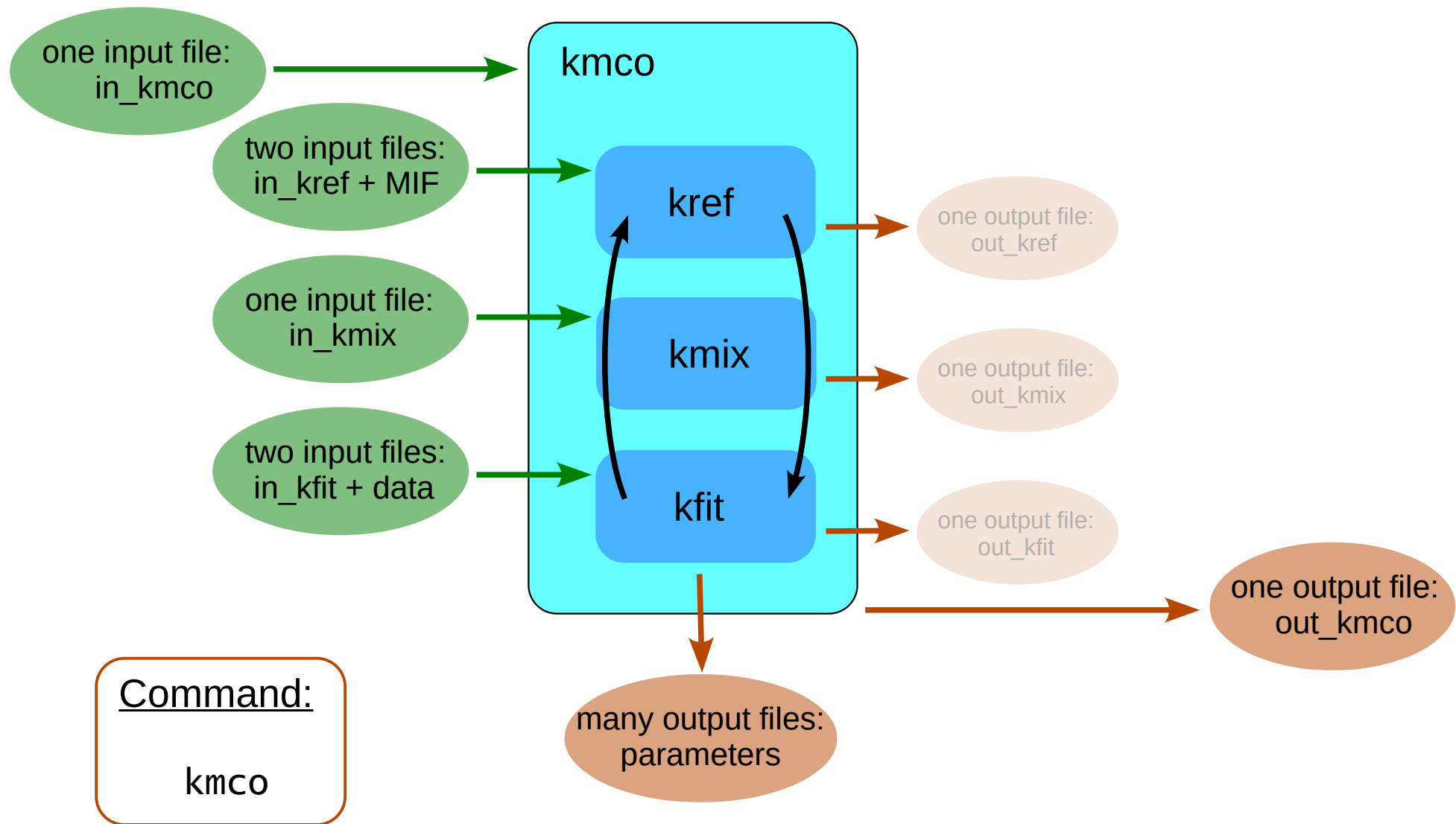


Randomized search:



☆ in each step the N-dimensional search space shrinks by ξ^N

Module configuration, Monte Carlo gamble:



Shot gun approach to fitting of SMS spectra:

➤ strategy

- ☆ identify relevant parameters
- ☆ explore parameter space using command `kmco`
- ☆ optimize parameter values using `kctl`

➤ redo examples that you found to be most difficult to fit

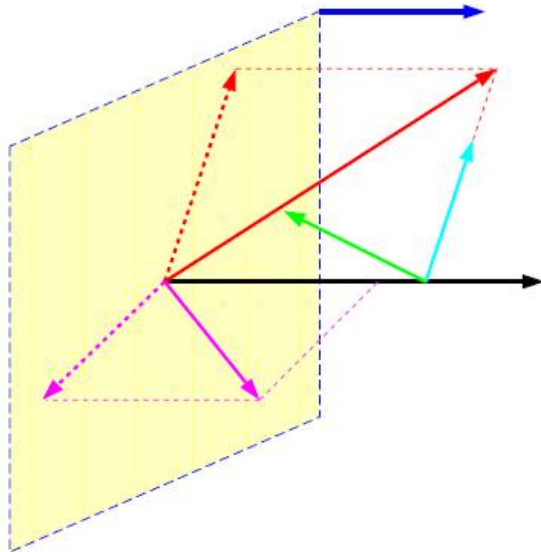
- ☆ construct the input files `in_kref`, `in_kmix`, `in_kfit`, `ex_in`, `in_kctl`
- ☆ focus on isomer shift, thickness, quadrupole splitting

END OF SATURDAY'S CLASS.

TOMORROW: MAGNETIC FIELDS

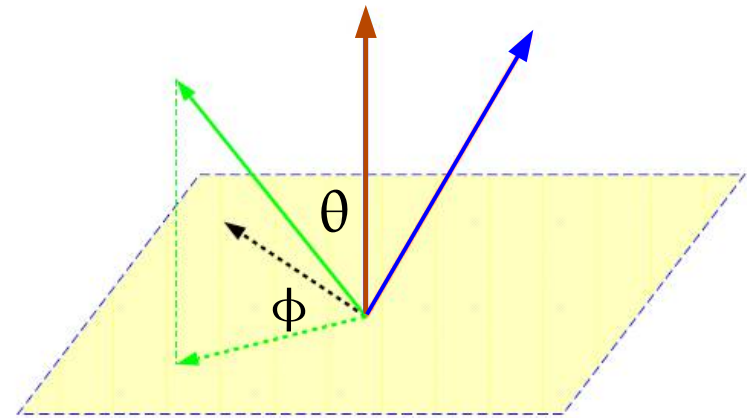
Polarization and field directions:

- polarization, defined by B_{ext}



- (→) direction of incident radiation
- (→) surface normal lines (8), (9), (10)
- (→) direction of external magnetic field
 $\angle(\cdot, \cdot) = \text{line (12)}$ $\angle(\cdot, \cdot) = \text{line (13)}$
- (→) = Sigma polarization
- (→) = Pi polarization
- (→) lattice base vector

- magnetic hyperfine field



- (→) surface normal
- (→) direction of external magnetic field
- (→) cross product (→) X (→)
- (→) direction of the magnetic hyperfine field
 $\angle(\cdot, \cdot) = \text{line (26.#.9)}$ $\angle(\cdot, \cdot) = \text{line (26.#.10)}$

Magnetic SMS spectra:

➤ strategy

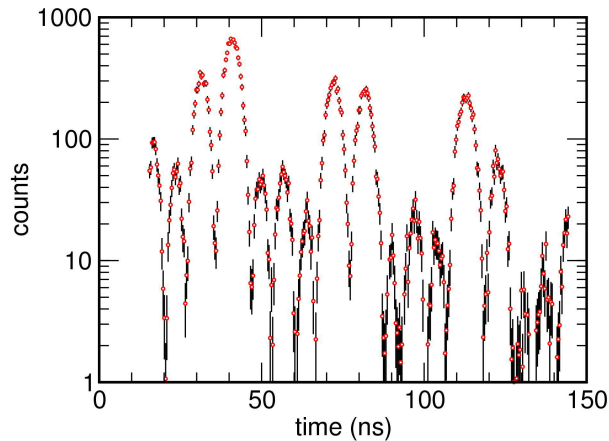
- ☆ identify relevant parameters
- ☆ use your choice approach...

➤ examples 4.1-3 and 5.1-3

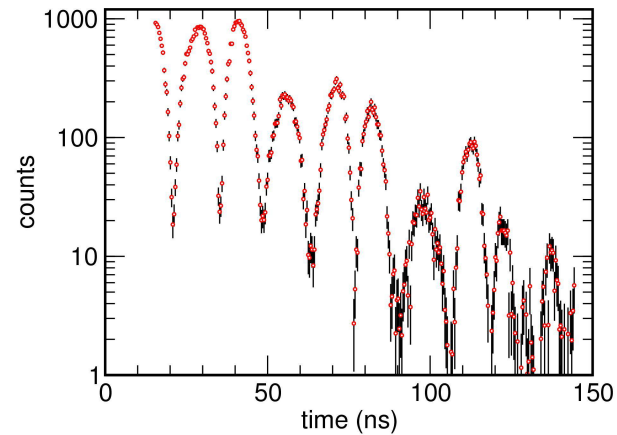
- ☆ construct the input files `in_kref`, `in_kmix`, `in_kfit`, `ex_in`, `in_kctl`
- ☆ focus on magnetic fields: magnitude, direction, and distribution

SMS examples, magnetic fields:

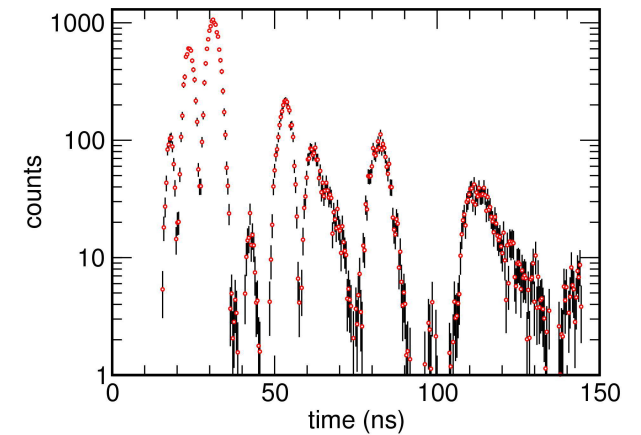
➤ example 4.1
no texture



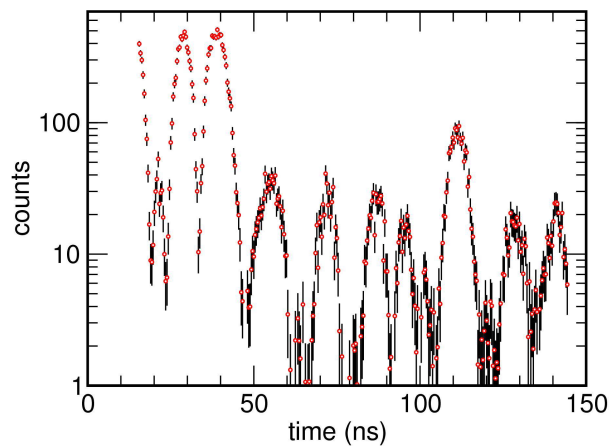
➤ example 4.2
texture



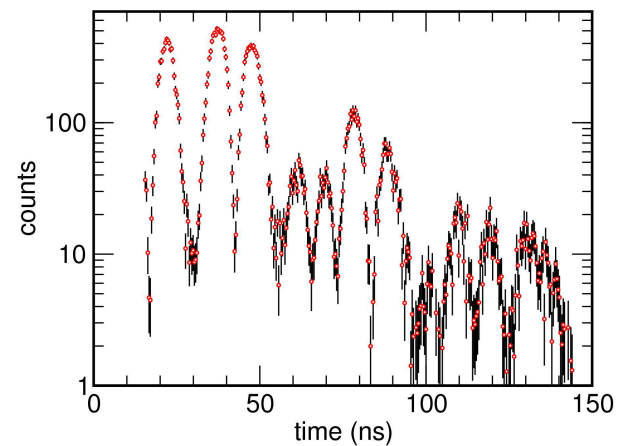
➤ example 4.3
no texture; distribution



➤ example 5.1
no texture



➤ example 5.2



➤ example 5.3

