USER MANUAL – *mint* version 1.0

Simulate inelastic neutron scattering induced magnetic excitations for single-ions, molecular-magnets and finite exchange coupled spin-systems

Michael L. Baker

Department of Chemistry, The University of Manchester, Manchester, M13 9PL, UK.

The University of Manchester at Harwell, Diamond Light Source, Harwell Campus, OX110DE, UK.

michael.baker@manchester.ac.uk

mlbakerlab.co.uk

27 September 2022

Introduction

mint simulates inelastic neutron scattering (INS) induced magnetic excitations for single-ions and finite systems of exchange coupled magnetic ions. Its purpose is to aid the planning of INS experiments and for fitting experimental data to Hamiltonian models. Such models can be applied to define the electronic structure and magnetism of single-ion magnets and exchange coupled polynuclear compounds[1]. *mint* runs in MATLAB and makes use of *easyspin* syntax for the generation of a given spin Hamiltonian model. It can be run as either a stand-alone toolbox or as an add-on to *easyspin*[2]. The version distributed uses *easyspin* as a dependency. The advantage of building upon *easyspin* is that input syntax for the generation of Hamiltonian models are well documented on the *easyspin* website[3] and the same script input can be readily used to simulate electron paramagnetic resonance, direct current magnetic susceptibility and magnetisation. There are additional inputs that are not present within *easyspin* to setup INS simulations. These include defining the geometric arrangement spin carrying atoms present within the system and INS experimental parameters. In this manual *mint* is introduced, a short theoretical background is given, INS simulation inputs and options are explained, and some example calculation results are given.

Organisation

mint is built from five different subroutines that simulate:

- Inelastic neutron scattering as a function of neutron energy loss for an integrated set of neutron momentum transfers for orientation averaged samples (SE).

- Inelastic neutron scattering as a function of momentum transfer dependence for transitions between a given set of initial and final state eigenvalues for orientation averaged samples (SQ).
- Inelastic neutron scattering as a function of neutron energy loss and momentum transfer for orientation averaged samples (SQE).
- Three-dimensional neutron momentum transfer dependence for transitions between a given set of initial and final state eigenvalues for single-crystal samples (SQxyz).
- Inelastic neutron scattering as a function of neutron energy loss for an integrated set of neutron momentum transfers for single-crystal samples (SExyz).

Theoretical background

The Hamiltonian Model

The Hamiltonian model selected can include various exchange terms (Heisenberg, Ising, Biquadratic, Dzyaloshinskii–Moriya interactions etc), single-ion zero-field splitting, extended Sevens Operators with ranks 2, 4 and 6, and nuclear hyperfine interactions. The Zeeman interaction term is included for static applied magnetic fields.

Often INS experiments access only a lowest energy subset of eigenvalues of a given Hamiltonian. Therefore, to save on the computational cost of working with large Hamiltonian matrices *mint* can optionally calculate a user defined low energy subset of eigenvalues and eigenvectors[4] therefore excluding states that are not accessed experimentally. This permits INS calculations of systems with larger Hamiltonian dimensions.

The inelastic neutron scattering cross-section

The calculated inelastic neutron scattering function for magnetic scattering summed over all initial (n) and final (m) states is written as:

$$S(\boldsymbol{Q}, \hbar\omega) = \sum_{n,m} P_n I_{nm}(\boldsymbol{Q}) \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\hbar\omega - E_i + E_f\right)^2/2\sigma^2}$$

where P_n is the partition function, E_i, E_f are the initial and final neutron energies and $\hbar\omega$ is the neutron energy loss. The INS transition oscillator strength between states is given by the quantity $I_{nm}(\boldsymbol{Q})$ and is a function of the neutron momentum transfer ($\boldsymbol{Q}$). The neutron momentum transfer is defined as the difference in initial and final neutron momentum ($\boldsymbol{Q} = \boldsymbol{k_i} - \boldsymbol{k_f}$). In experiment the neutron scatting intensity as a function of energy exhibits a broadening, this can be approximated as a gaussian function, as shown above, with a width defined by σ (where the full width half maximum = $2\sqrt{2ln(2)}\sigma$). Since magnetic excitations for single-ions and finite spin systems are non-dispersive, modulations in intensity as a function of momentum transfer are considerably broader than instrument resolution, consequently a convolution with instrument momentum transfer resolution is not required. Subroutines for the simulation of polycrystalline and powder samples (spectrum types: SE, SQ and SQE) make use of an analytical method for orientation averaging[5].

Scattering due to spin only

Here we consider the scattering due to spin-only for a finite magnetic system, where unpaired electrons are localised at coordinates $(\boldsymbol{r}_k)$. Scattering due to spin only is a suitable approximation when the orbital angular momentum is zero or is quenched by the presence of a ligand field at the scattering atomic sites (k). The intensity quantity $I_{nm}(\boldsymbol{Q})$ is given by

$$I_{nm}(\boldsymbol{Q}) = \sum_{kk'} F_k(Q)F_{k'}(Q)e^{i\boldsymbol{Q}\cdot(r_k-r_{k'})}\sum_{\alpha\beta}\left(\delta_{\alpha\beta}-\frac{Q_\alpha Q_\beta}{Q^2}\right)\langle n|\hat{s}_{k\alpha}|m\rangle\langle m|\hat{s}_{k'\beta}|n\rangle$$

Where $\hat{s}_{k\alpha}$ and $\hat{s}_{k'\beta}$ are spin x, y and z operators of spin sites k and k' respectively. The magnetic form factor $(F_k(Q))$ for neutrons scattered by ion k is given by

$$\langle j_0(s)\rangle = Ae^{-as^2} + Be^{-bs^2} + Ce^{-cs^2} + D$$

Where $\langle j_0(s)\rangle$ represent the radial integration of unpaired electrons, $s = Q/4\pi$ and coefficients A, B, C, D, a, b and c are constants fitted from Hartree-Fock wave-functions[6].

Scattering due to spin and orbital angular momentum

The ions of many f-block elements have both spin and unquenched orbital angular momentum. The intensity quantity is modified such that the $\hat{s}_{k\alpha}$ and $\hat{s}_{k'\beta}$ operators are interchanged with the total angular momentum operators $(\hat{J}_{k\alpha}, \hat{J}_{k'\beta})$ and the magnetic formfactor $F(Q)$ is replaced by

$$\frac{g_j}{2}F(Q) = \langle j_0(s)\rangle + (1-2/g_J)\langle j_2(s)\rangle$$

Where g_J is the Landé splitting factor and $\langle j_n(s)\rangle$ represent the radial integration of unpaired electrons. These integrals are obtained from spherical Bessel functions of order n. Coefficients of an analytical approximation to the $\langle j_n(s)\rangle$ integrals have the form

$$\langle j_0(s)\rangle = Ae^{-as^2} + Be^{-bs^2} + Ce^{-cs^2} + D$$
$$\langle j_2(s)\rangle = \left(Ae^{-as^2} + Be^{-bs^2} + Ce^{-cs^2} + D\right)s^2$$

Magnetic form-factor coefficients for transition metal-ions and lanthanides in common oxidation states are included within *mint* in the spin-only approximation $(F(Q) = \langle j_0(s)\rangle)$. If the required element, oxidation-state or if this level of precision is not appropriate, parameters for calculating magnetic formfactors with spin and orbital angular momentum should be defined by the user as a 1x15 array per ion of order [j0A j0a j0B j0b j0C j0c j0D j2A j2a j2B j2b j2C j2c j2D gJ].

Input syntax for the simulation of INS

mint follows the same input syntax structure as *easyspin*. These include Sys, Exp and Opt structures with fields that relate to the sample system, experiment and calculation respectively.

Neutron scatting specific input syntax relating the studied sample

Sys.FormFactor – defines the magnetic formfactor for spin carrying metal ions. The magnetic formfactor for relevant transition metal ions of oxidation states from 0 to 4+ and lanthanides with oxidation states 2+ and 3+ are recognised with a string input that includes the element followed by a number representing the oxidation state. For instance, the magnetic formfactor for terbium 3+ is given as:

```
        Sys.FormFactor = 'Tb3'.
```

For multinuclear spin systems the magnetic formfactor for each ion can be given as a cell array. For instance, the magnetic formfactor of a dimer composed of terbium 3+ and copper 2+ is given as:

```
        Sys.FormFactor = {'Tb3','Cu2'}
```

Where the first ion defined within the spin system is terbium 3+ with a J = 6 ground term and the second is copper 2+ with a S = ½ (`Sys.S = [6 1/2]`). There is also the option for a user defined magnetic formfactor that is given as a 1x15 array of form [j0A j0a j0B j0b j0C j0c j0D j2A j2a j2B j2b j2C j2c j2D gJ] which would replace a single string in the Sys.FormFactor cell array.

Sys.Coords – defines the relative position of spin carrying metal-ions. Cartesian coordinates in ångströms must be defined for exchange coupled systems. The coordinate system is the same as the Hamiltonian. The x, y and z coordinates for multiple ions are given, for example, for a three-metal-ion system as follows:

```
        Sys.Coords = [x1 y1 z1; x2 y2 z2; x3 y3 z3].
```

Neutron scatting specific input syntax relating the experiment geometry and sample environment

Exp.Energy – defines the energy array for which the INS spectrum will be simulated. The energy array units should be given in meV.

Exp.lwfwhm – defines the line width broadening of INS transition full width half maxima given in meV. Linewidth broadening is gaussian and is constant as a function of energy. If there is a need for linewidth that varies as a function of energy or non-gaussian type broadening this could be implemented in the future.

Exp.Q – defines the neutron momentum transfer magnitude array in inverse ångströms for SQ and SQE type calculations.

Exp.Qx / Exp.Qy / Exp.Qz – defines neutron momentum transfer arrays in cartesian x, y and z directions in inverse ångströms for SQxyz and SExyz calculations. Selecting arrays requires consideration of how to visualise the result. For SQxyz calculations it is often clearer and most efficient to simulate multiple 2D slices of 3D momentum transfer dependence by giving a single value for one of the three Exp.Qx, Exp.Qy or Exp.Qz arrays. Multiple 2D slices then can be constructed and plotted together using the plotSQxyz function. Alternatively, if the three-dimensional INS cross-sesction is requested, the result can be viewed as an iso-surface, however such calculations are intensive to evaluate for large portions of momentum transfer space. For SExyz calculations Exp.Qx, Exp.Qy, Exp.Qz arrays must be of the same length.

Exp.Temperature – the sample temperature given in Kelvin. This can be either a single temperature or an array of temperatures that are evaluated and returned for plotting. Only the first temperature value given is read for SQE or SQxyz calculations.

Exp.B – the applied magnetic field vector at the sample given in milli-Tesla as a three-element array, specifying direction and magnitude [x y z] in cartesian coordinates. Applied magnetic fields can only be included for single crystal INS calculations.

Exp.SpectrumType – type of simulation requested this can be either:

- `Exp.SpectrumType = 'SE';` neutron scattering as a function of neutron energy loss for an integrated set of neutron momentum transfers for orientation averaged samples.
- `Exp.SpectrumType = 'SQ';` neutron momentum transfer dependence for transitions between a given set of initial and final state eigenvalues for orientation averaged samples.
- `Exp.SpectrumType = 'SQE';` neutron scattering as a function of neutron energy loss and momentum transfer for orientation averaged samples.
- `Exp.SpectrumType = 'SQxyz';` three-dimensional neutron momentum transfer dependence for transitions between a given set of initial and final state eigenvalues for single crystal samples.
- `Exp.SpectrumType = 'SExyz';` neutron scattering as a function of neutron energy loss for an integrated set of neutron momentum transfers for single crystal samples.

Exp.InitialStates – is an array defining the eigenvalue indexes considered as initial states for which transitions are calculated. If Exp.InitialStates is not given then Exp.InitialStates is automatically limited to transitions that have a Boltzmann population greater than 0.1%.

Exp.FinalStates – is an array defining the eigenvalue indexes considered as final states for which transitions are calculated. If Exp.FinalStates is not given all possible transitions with the requested energy spectrum range are evaluated.

Neutron scatting specific input syntax relating to computation

Opt.NumEigs - is an integer that caps the number of eigenvalues considered in the INS simulation. This is relevant when computing large spin systems with Hamiltonian matrix dimensions that are prohibitively large to evaluate in their entirety. It is useful when only the lowest energy eigenvalues are accessed experimentally. If Opt.NumEigs is not specified the INS calculation includes all eigenvalues by default.

NB! It is worth confirming that the use of Opt.NumEigs has not affected the simulated INS result. To verify this, it is recommended that a second calculation be conducted where Opt.NumEigs is increased slightly to ensure that there is no impact on simulated INS spectrum result.

Running simulations and analysing results

Simulations are conducted with the "mint" command, *mint* takes the Sys, Exp and Opt structures as defined above. It returns the neutron scattering cross-section as the first output and optionally returns the eigenvalues and eigenvectors of the system Hamiltonian:

```
cross_sect = mint(Sys,Exp);
[cross_sect,Eigs,Vecs] = mint(Sys,Exp,Opt);
```

Orientation averaged simulation results can be plotted using standard MATLAB functions, for instance plotting the result of an SE calculation can be given as:

```
figure
plot(Exp.Energy,cross_sect);
xlabel('Energy (meV)')
ylabel('Intensity (arb. units)')
```

An SQ calculation can be given as:

```
figure
plot(Exp.Q,cross_sect); hold on
xlabel('Q ($\AA^{-1}$)','interpreter','latex')
ylabel('Intensity (arb. units)')
```

An SQE calculation can be plotted using the Matlab surf function as follows:

```
figure
surf(Exp.Q,Exp.Energy,cross_sect)
view(0,90); shading interp; colorbar; axis tight
xlabel('Q ($\AA^{-1}$)','interpreter','latex')
ylabel('Energy (meV)')
```

A two-dimensional slice in three dimensional neutron momentum transfer can be plotted with the mint-function plotSQxyz as follows:

```
figure
plotSQxyz(Exp.Qx, Exp.Qy, Exp.Qz,cross_sect);
```

Multiple two dimensional slices of three dimensional momentum transfer dependence can be plotted together as follows:

```
figure
plotSQxyz(Exp.Qx1,Exp.Qy1,Exp.Qz1,cross1); hold on; caxis([0 18]);
plotSQxyz(Exp.Qx2,Exp.Qy2,Exp.Qz2,cross2); hold on; caxis([0 18]);
plotSQxyz(Exp.Qx3,Exp.Qy3,Exp.Qz3,cross3); hold on; caxis([0 18]);
```

Note - that for each two-dimensional plane the colour axis range (caxis) given must be equivalent.

Licence

This software is provided free for use for non-commercial applications. If mint is used in research that results in scientific publication please cite “mint: simulate inelastic neutron scattering induced magnetic excitations for single-ions, molecular-magnets and finite exchange coupled spin-systems, Michael. L. Baker, 2022”, a journal paper for citation will follow soon.

Installation

1. *mint* requires the installation of easyspin[2] as a dependency. Visit the easyspin website[3] for downloading the latest version.

2. Visit the Baker group website[7] to download the latest version of mint.

3. Unpack the downloaded zip package to a folder.

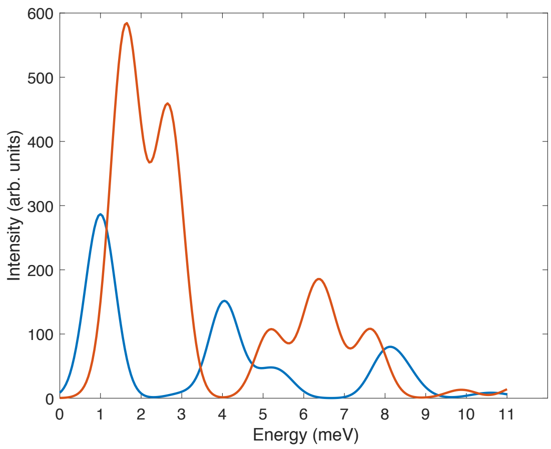
4. Launch MATLAB and go to Home → Set Path. Add the mint folder to the matlab search path by clicking on "Add Folder...", selecting the mint subfolder directory, and clicking on “Save”.

Example input scripts and resultant figures

Example 1: Crystal field excitations of lanthanide single ion magnets[8]

```
% Simulation reproducing Figure 4b of
% Chem. Commun., 2016,52, 2091-2094
clear all
% some constants
rcm = 29979.2458;   %rcm    to MHz
meV = rcm*8.065;    %meV    to MHz

% INS specific inputs
Exp.SpectrumType = 'SE';
Exp.Temperature = 30;
Exp.Energy = 0:0.05:11; Exp.lwfwhm = 0.87;
Exp.Q = 1.15:0.05:1.65;
Sys.FormFactor = 'Tb3';
Sys.Coords = [0 0 0];

% Tb3+ J=6
Sys.S = 6;

%Tb-b compound crystal field parameters
Sys.B2 = [0 0 0.712 0 0.0729].*meV;
Sys.B4 = [0 0 0 0 0 0 0 0 0].*meV;

% run simulation and plot
[cross,Eigsb] = mint(Sys,Exp);
figure
plot(Exp.Energy,cross.*3); hold on

%Tb-a compound crystal field parameters
Sys.B2 = [0 0 0.705 0 0.0250].*meV;
```

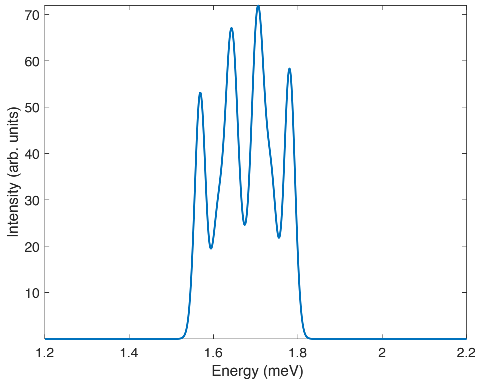
```
      Sys.B4 = [0 0 0 0 0 0 0 0 -8.5E-4].*meV;

      % run simulation and plot over previous simulation in a 3 to 7 scaled
      ratio to match figure 4b of Chem. Commun., 2016,52, 2091-2094
      [cross,Eigsa] = mint(Sys,Exp);
      plot(Exp.Energy,cross.*7);

      % format figure axes
      set(gca,'xtick',[0:1:11])
      xlabel('Energy (meV)');
      ylabel('Intensity (arb. units)');
```



Example 2: an exchange coupled Tb – Cu single molecule magnet including anisotropic spin exchange and nuclear hyperfine coupling.[9]

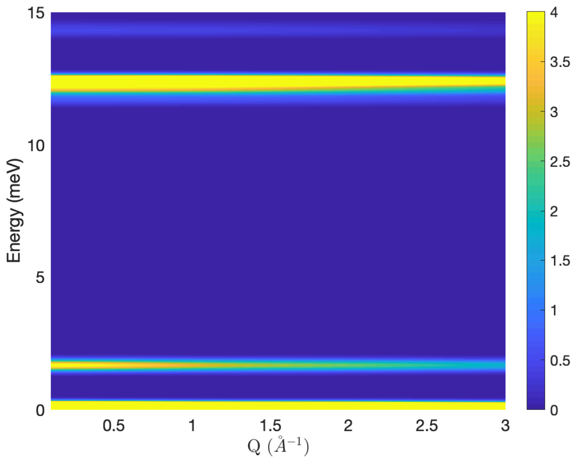
```
%% Example 2
% PHYSICAL REVIEW B 88, 064405 (2013)
clear all
% some constants
rcm = 29979.2458;    % reciprocal cm to MHz
meV = rcm*8.065;     % meV           to MHz

% All parameters come from PHYSICAL REVIEW B 88, 064405 (2013)

%Setup of J=6 for Tb3+ and S=1/2 for Cu
Sys.S = [6 1/2];

% Exchange coupling
Jxx = -0.54;
Jyy = -0.54;
Jzz = -0.298;
Sys.ee = [Jxx Jyy Jzz].*meV;
Sys.Coords = [0 0 0; 3.48 0 0]; % distance between Tb and Cu

%hyperfine term
Sys.Nucs = '159Tb,63Cu';
Sys.A = [6.2e-3 6.2e-3 6.2e-3 0 0 0 ; 0 0 0 0.043 0.043 0.043].*meV;

Sys.FormFactor = {'Tb3','Cu2'}; %elements included
Sys.Coords = [0 0 0; 3.48 0 0]; % distance between Tb and Cu

%Stevens parameters
```

```
B20 = -0.197;
B22 = 0.066;
B40 = 1.5e-4;
B44 = 5e-4;
B60 = -1.25e-5;
B64 = 1.17e-4;
Sys.B2 = [B22 0 B20 0 0 ; zeros(1,5)].*meV;        % B(k=2,q) with q =
+2,+1,0,-1,-2
Sys.B4 = [B44 0 0 0 B40 0 0 0 0 ; zeros(1,9)].*meV;  % B(k=4,q) with q =
+4,+3,+2,+1,0,-1,-2,-3,-4
Sys.B6 = [0 0 B64 0 0 0 B60 0 0 0 0 0 0 ; zeros(1,13)].*meV;  % B(k=6,q)
with q = +4,+3,+2,+1,0,-1,-2,-3,-4

% SE calculation
Exp.Energy = [0:0.01:16]; % spectrum energy range in meV
Exp.lwfwhm = .3; % gaussin broadening FWHM in meV
Exp.Q = [0.1,0.5,1,1.5,2]; % Q range in A^-1
Exp.SpectrumType = 'SE'; % S(E) only
Exp.Temperature = 3.5; % Kelvin

[cross_sect,Eigs]=mint(Sys,Exp);

% SE calculation
Exp.Energy = [1.2:0.001:2.2]; % spectrum energy range in meV
Exp.lwfwhm = .03; % gaussin broadening FWHM in meV
Exp.Q = [0.3:0.1:1]; % Q range in A^-1
Exp.SpectrumType = 'SE'; % S(E) only
Exp.Temperature = 3.5; % Kelvin
[cross_sect,Eigs]=mint(Sys,Exp);

%make figure 5a
figure
plot(Exp.Energy,cross_sect); hold on
xlabel('Energy (meV)')
ylabel('Intensity (arb. units)')
set(gca,'xtick',[1.2:0.2:2.2])
axis tight
```

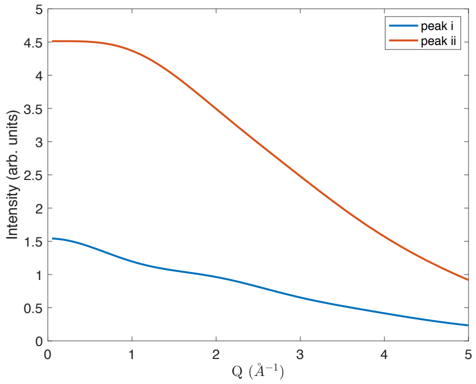


```
% SQE map calculation, simulation of figure 2
Exp.Energy = [0:0.05:15]; % spectrum energy range in meV
Exp.Q = [0.1:0.05:3]; %Q range in A^-1
Exp.lwfwhm = .3; % gaussin broadening FWHM in meV
Exp.SpectrumType = 'SQE' ; % SQE map calculation
[cross_sect,Eigs]=mint(Sys,Exp);

%make figure
```

```
figure
surf(Exp.Q,Exp.Energy,cross_sect)
view(0,90); shading interp; colorbar; axis tight
caxis([0,4]);
xlabel('Q ($\AA^{-1}$)','interpreter','latex')
ylabel('Energy (meV)')
```

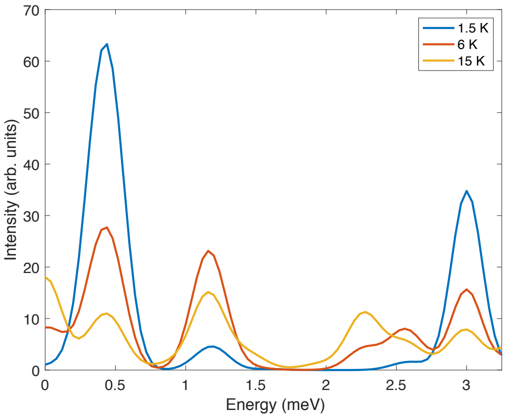


```
%make SQ plot
Exp.SpectrumType = 'SQ';
Exp.Q = [0:0.05:5]; %Q range in A^-1
Exp.InitialStates = [1:32]; % ground doublet
Exp.FinalStates = [33:64]; % to exchange split excited states
[cross_sect1,Eigs]=mint(Sys,Exp);

Exp.InitialStates = [1:32]; % ground doublet
Exp.FinalStates = [65:160]; % to CF and exchange split excited states
[cross_sect2,Eigs]=mint(Sys,Exp);

figure
plot(Exp.Q,cross_sect1,Exp.Q,cross_sect2); hold on
xlabel('Q ($\AA^{-1}$)','interpreter','latex')
ylabel('Intensity (arb. units)')
legend('peak i','peak ii')
```



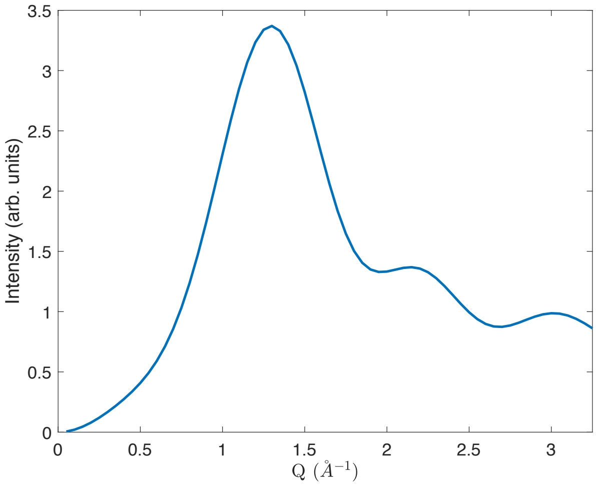
Example 3: a Cr_6 spin-chain with antiferromagnetic Heisenberg exchange coupling and single ion anisotropy.[10]

```
% Reproduces Fig 6 of Dalton Trans., 2011, 40, 2725-2734.
clear all;
% some constants
rcm = 29979.2458;   %rcm    to MHz
meV = rcm*8.065;    %meV    to MHz

% setup of the Hamiltonian in the usual easyspin way
D = -0.041.*meV; % convert from meV to MHz
E = 0.007.*meV; % convert from meV to MHz


Sys.S = [3/2 3/2 3/2 3/2 3/2 3/2];
Sys.D = [D E;D E;D E;D E;D E;D E];
Sys.J = [1.46 0 0 0 0 1.46 0 0 0 1.46 0 0 1.46 0 1.46].*meV; %
convert from meV to MHz
Sys.FormFactor = 'Cr3';
% Cr ion coords in angstroms
Sys.Coords =   [      0          0          0 ;
                -3.3807     0.0840    -0.2765 ;
                -5.7290    -2.3170     0.1103 ;
                -5.5960    -5.6470    -0.2956 ;
                -3.1148    -7.8270     0.2920 ;
                 0.2482    -7.7410    -0.1803];

Opt.NumEigs = 100;

Exp.SpectrumType = 'SE';
Exp.Temperature = [1.5, 6, 15];
Exp.Energy = -1:0.04:4;
Exp.lwfwhm = 0.25;
Exp.Q = [0.1,0.5,1,1.5,2];
 [cross,Eigs] = mint(Sys,Exp,Opt);
figure
plot(Exp.Energy,cross)
xlim([0 3.25])
xlabel('Energy (meV)')
ylabel('Intensity (arb. units)')
legend('1.5 K','6 K','15 K')
```



```
%make Fig S14 of Dalton Trans., 2011, 40, 2725-2734.
Exp.SpectrumType = 'SQ';
Exp.Q = [0:0.05:5]; %Q range in A^-1
Exp.InitialStates = 2:4;
Exp.FinalStates = 5:9;
```

```
      Exp.Temperature = 6;
      [cross_sectQ,Eigs]=mint(Sys,Exp,Opt);

      figure
      plot(Exp.Q,cross_sectQ); hold on
      xlabel('Q ($\AA^{-1}$)','interpreter','latex')
      ylabel('Intensity (arb. units)')
      xlim([0 3.25])
```

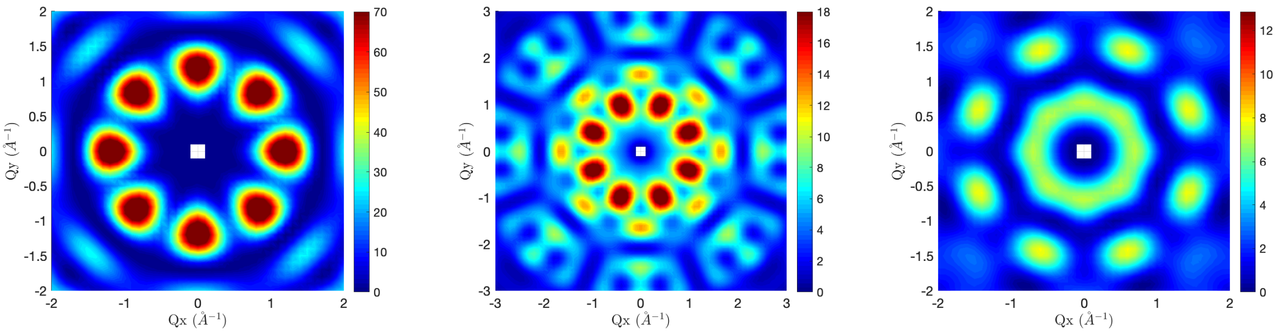


Example 4: Single crystal momentum transfer dependence of a Cr8 molecular nanomagnet.[11]

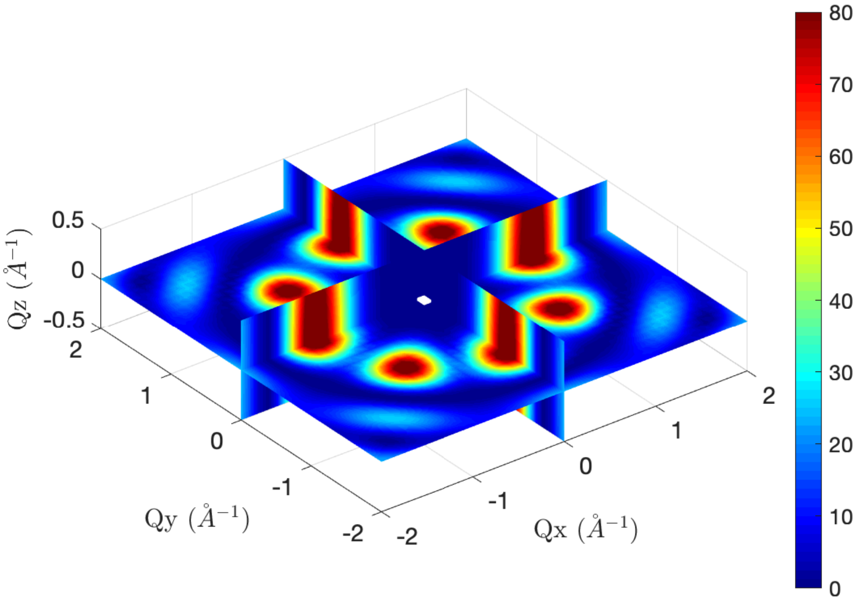
```
% reproducing Nat Phys, 2012, 8, 906 Figure 3 parts d e and f
clear all;
% some constants
rcm = 29979.2458;   %rcm    to MHz
meV = rcm*8.065;    %meV    to MHz

% setup of the Hamiltonian in the usual easyspin way
D = -0.041.*meV; % convert from meV to MHz
E = 0.007.*meV; % convert from meV to MHz

Sys.S = [3/2 3/2 3/2 3/2 3/2 3/2 3/2 3/2];
Sys.D = [D E;D E;D E;D E;D E;D E;D E;D E];
Sys.J = [1.46 0 0 0 0 0 1.46 1.46 0 0 0 0 0 1.46 0 0 0 0 1.46 0 0 0 1.46 0
0 1.46 0 1.47].*meV; % convert from meV to MHz
Sys.FormFactor = 'Cr3';

% include eigenvalues up to 6 meV only
Opt.NumEigs = 55;

Exp.SpectrumType = 'SQxyz';
Exp.Temperature = 1.5;

% coordinates for molecule 1 in the unit cell
Sys.Coords =    [0.000 0.000  0.000;
2.971 1.621 -0.154;
3.947 4.860  0.000;
2.325 7.832 -0.154;
-0.914 8.807  0.000;
-3.885 7.186 -0.154;
-4.861 3.947  0.000;
-3.239 0.975 -0.154];
```

```
% figure 3d molecule 1
Exp.InitialStates = 1;
Exp.FinalStates = 2:4;
Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross1_mol1,Eigs] = mint(Sys,Exp,Opt);

% figure 3e molecule 1
Exp.InitialStates = 1;
Exp.FinalStates = 10:21;
Exp.Qx = -3:0.1:3; Exp.Qy = -3:0.1:3; Exp.Qz = 0;
[cross2_mol1,Eigs] = mint(Sys,Exp,Opt);

% figure 3f molecule 1
Exp.InitialStates = 1;
Exp.FinalStates = 29:55;
Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross3_mol1,Eigs] = mint(Sys,Exp,Opt);

% coordinates for molecule 2 in the unit cell
Sys.Coords =    [0.000 0.000  0.000
3.239 0.976 0.154;
4.860 3.947 0.000;
3.885 7.186 0.154;
0.913 8.808 0.000;
-2.326 7.832 0.154;
-3.947 4.861 0.000;
-2.972 1.622 0.154];

% figure 3d molecule 2
Exp.InitialStates = 1;
Exp.FinalStates = 2:4;
Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross1_mol2,Eigs] = mint(Sys,Exp,Opt);

% figure 3e molecule 2
Exp.InitialStates = 1;
Exp.FinalStates = 10:21;
Exp.Qx = -3:0.1:3; Exp.Qy = -3:0.1:3; Exp.Qz = 0;
[cross2_mol2,Eigs] = mint(Sys,Exp,Opt);

% figure 3f molecule 2
Exp.InitialStates = 1;
Exp.FinalStates = 29:55;
Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross3_mol2,Eigs] = mint(Sys,Exp,Opt);

figure
subplot(1,3,1)
plotSQxyz(-2:0.1:2,-2:0.1:2,0,cross1_mol1+cross1_mol2); hold on; caxis([0
18]);
xlabel('Qx ($\AA^{-1}$)','interpreter','latex')
ylabel('Qy ($\AA^{-1}$)','interpreter','latex')
view(0,90)
caxis([0 70]);
colormap jet
caxis;
axis tight
subplot(1,3,2)
plotSQxyz(-3:0.1:3,-3:0.1:3,0,cross2_mol1+cross2_mol2); hold on; caxis([0
18]);
```

```
xlabel('Qx ($\AA^{-1}$)','interpreter','latex')
ylabel('Qy ($\AA^{-1}$)','interpreter','latex')
view(0,90)
caxis([0 18]);
colormap jet
caxis;
axis tight
subplot(1,3,3)
plotSQxyz(-2:0.1:2,-2:0.1:2,0,cross3_mol1+cross3_mol2); hold on; caxis([0
18]);
xlabel('Qx ($\AA^{-1}$)','interpreter','latex')
ylabel('Qy ($\AA^{-1}$)','interpreter','latex')
view(0,90)
caxis([0 12.85]);
colormap jet
caxis;
axis tight
```



```
% make a 3D slice simulating Figure 2 from Nat Phys, 2012, 8, 906
Exp.InitialStates = 1;
Exp.FinalStates = 2:4;
Opt.NumEigs = 4;

% coordinates for molecule 1 in the unit cell
Sys.Coords =    [0.000 0.000  0.000;
2.971 1.621 -0.154;
3.947 4.860  0.000;
2.325 7.832 -0.154;
-0.914 8.807  0.000;
-3.885 7.186 -0.154;
-4.861 3.947  0.000;
-3.239 0.975 -0.154];

Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross4_mol1,Eigs] = mint(Sys,Exp,Opt);

Exp.Qx = -2:0.1:2; Exp.Qy = 0; Exp.Qz = -0.5:0.05:0.5;
[cross5_mol1,Eigs] = mint(Sys,Exp,Opt);

Exp.Qx = 0; Exp.Qy = -2:0.1:2; Exp.Qz = -0.5:0.05:0.5;
[cross6_mol1,Eigs] = mint(Sys,Exp,Opt);

% coordinates for molecule 2 in the unit cell
Sys.Coords =    [0.000 0.000  0.000
3.239 0.976 0.154;
4.860 3.947 0.000;
3.885 7.186 0.154;
0.913 8.808 0.000;
-2.326 7.832 0.154;
-3.947 4.861 0.000;
-2.972 1.622 0.154];

Exp.Qx = -2:0.1:2; Exp.Qy = -2:0.1:2; Exp.Qz = 0;
[cross4_mol2,Eigs] = mint(Sys,Exp,Opt);

Exp.Qx = -2:0.1:2; Exp.Qy = 0; Exp.Qz = -0.5:0.05:0.5;
[cross5_mol2,Eigs] = mint(Sys,Exp,Opt);

Exp.Qx = 0; Exp.Qy = -2:0.1:2; Exp.Qz = -0.5:0.05:0.5;
[cross6_mol2,Eigs] = mint(Sys,Exp,Opt);

figure
plotSQxyz(-2:0.1:2,-2:0.1:2,0,cross4_mol1+cross4_mol2); hold on; caxis([0
80]);
plotSQxyz(-2:0.1:2,0,-0.5:0.05:0.5,cross5_mol1+cross5_mol2); hold on;
caxis([0 80]);
plotSQxyz(0,-2:0.1:2,-0.5:0.05:0.5,cross6_mol1+cross6_mol2); hold on;
caxis([0 80]);
xlabel('Qx ($\AA^{-1}$)','interpreter','latex')
ylabel('Qy ($\AA^{-1}$)','interpreter','latex')
colormap jet
caxis;
axis equal
```



References

1 M. L. Baker, S. J. Blundell, N. Domingo and S. Hill, ed. S. Gao, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 231–291.

2 S. Stoll and A. Schweiger, *J. Magn. Reson.*, 2006, **178**, 42–55.

3 easyspin, https://easyspin.org/.

4 D. Kressner, *Numer. Math.*, 2006, **103**, 461–483.

5 R. Caciuffo, T. Guidi, G. Amoretti, S. Carretta, E. Liviotti, P. Santini, C. Mondelli, G. Timco, C. A. Muryn and R. E. P. Winpenny, *Phys. Rev. B*, 2005, **71**, 174407.

6 D. Albert-José and L. Gerry, Eds., *NEUTRON DATA BOOKLET*, OCP Science, 2nd edn., 2003.

7 mint, https://www.mlbakerlab.co.uk.

8 M. Vonci, M. J. Giansiracusa, R. W. Gable, W. den Heuvel, K. Latham, B. Moubaraki, K. S. Murray, D. Yu, R. A. Mole, A. Soncini and C. Boskovic, *Chem. Commun.*, 2016, **52**, 2091–2094.

9 M. Kofu, O. Yamamuro, T. Kajiwara, Y. Yoshimura, M. Nakano, K. Nakajima, S. Ohira-Kawamura, T. Kikuchi and Y. Inamura, *Phys. Rev. B*, 2013, **88**, 64405.

10 M. L. Baker, A. Bianchi, S. Carretta, D. Collison, R. J. Docherty, E. J. L. McInnes, A. McRobbie, C. A. Muryn, H. Mutka, S. Piligkos, M. Rancan, P. Santini, G. A. Timco, P. L. W. Tregenna-Piggott, F. Tuna, H. U. Güdel and R. E. P. Winpenny, *Dalt. Trans.*, 2011, **40**, 2725–2734.

11 M. L. Baker, T. Guidi, S. Carretta, J. Ollivier, H. Mutka, H. U. Güdel, G. A. Timco, E. J. L. McInnes, G. Amoretti, R. E. P. Winpenny and P. Santini, *Nat Phys*, 2012, **8**, 906.