

RKMAG

USER'S MANUAL

John B. Ketterson

K.Rivkin

Introduction

This program is intended to offer an alternative to traditional approaches, like using Runge-Kutta integration of Landau-Lifshitz equation, when it comes to calculating resonant spin wave modes in arbitrary shaped bodies.

The idea behind the method is to formulate the problem of finding the resonant frequencies and resonant modes as an eigenvalue problem. Further on, upon calculating the resonant modes, one can use these modes to calculate the absorption properties of the body in question.

We would to offer our appreciation to Mr. Alexander Heifetz and Dr. Paul Sievert for their contribution to this project.

We also would like to mention that while this method have been developed by us independently, we came to know that some of it's elements where previously published in works of P. Politi, M. G. Pini, A. Rettori, F. C. Nortemann, R. L. Stamps, R. E. Campley and their collaborators.

The program is intended for non-commercial use.

Introduction	2
Installation	6
a. Optional commands	7
SET_GYRO	7
SET_BETA	7
SET_CURRENT	7
SET_HRF	8
SET_VRML	9
SET_SCALE	9
b. Describing the object	9
LOAD_MAT	10
MASQUE_LOAD	10
MASQUE_SAVE	11
SET_CELL	11
STUDY_SHAPE	11
MASQUE_CREATE	12
ADD_SHAPE	12
MASQUE_INFO	14
TENSOR_CREATE	14
ADD_ANISOTROPY	15
ADD_ANISOTROPY_FIELD	15
TENSOR_SAVE	16
TENSOR_LOAD	16
TENSOR_MACRO	16
c. Magnetization dynamics and other related procedures	17
M_LOAD	17
M_SAVE	17
INIT_M	17
ADD_EXT_FIELD	18
RELAX	18
H_SAVE	19
H_LOAD	20
H_AVERAGE	20
SET_TIMESTEP	20
EXEC	20
d. Solving for eigenvalues and eigenvectors	21
SOLVE_MATRIX	21
SOLVE_MATRIX_AR	21
EIGEN_SAVE_TEXT	22
OMEGA_PRINT	22
EIGEN_LOAD	22
CALC_C	22
PRINT_C	23
PRINT_ABS	23
VEC3	23
e. Displaying the eigenmodes	24
PRINT_PHASES	25
DRAW_MODES	25
DRAW_MODES_P	26
DRAW_MODES_E	27
END	27

Final Notes.....29

This program uses Gaussian units, with an exception of the units for time and frequency, which are being defined by the choice of gyroscopic coefficient γ - see the description of SET_GYRO command below.

The method have been previously described in a number of publications; for more information please read the following work:

K. Rivkin and J. B. Ketterson, "Micromagnetic simulations of absorption spectra", to be published in Journal of Magnetism and Magnetic Materials.

<http://www.rkmag.com/articles/abs2.pdf>

Installation.

Currently the program is intended to work under Linux/FreeBSD OS, but can be relatively easily recompiled for other operational systems. To install the program you need to download the appropriate executable file from <http://www.rkmag.com/page2.html>

While the new version of the software is supposed to be stand-alone executable, it may occur that on certain system the program require certain libraries to be present in the system; These libraries are available at www.rkmag.com/page2.html

All one needs to do is to place them into one of the following directories: /lib, /usr/lib, usr/local/lib, or other directory, that is included in ld search path. After doing so ldconfig or /etc/ldconfig should be executed.

It's also highly recommended that ulimit –s command shall be used before the program's execution in order to set the stack size to a much higher than default value (500-5000 Kilobytes should suffice for most of the cases – “ulimit –s 5000000” would be a safe choice). Otherwise, the segmentation fault can occur.

Unfortunately current version does not contain check the input.dat file for errors. Incorrect input data will most likely result in a segmentation fault.

The program is intended for non-commercial use, however we do express our desire that all the papers written with the help of RKMAG software packages should acknowledge such fact. We also ask you to be so kind as to submit the references to such papers to us by e-mailing us at rivkin@rkmag.com .

It is also important to mention that the maximum number of individual cells (dipoles) that one can effectively use with eigenvalue method is 6000-11,000. If one wants to have a better discretization with more dipoles, one needs to use one of approximate methods described below, for example the best results for thin films can be obtained by using sparce algorithms.

The program is executed by running the executable file (for example rkmag). In the same directory there should be present the input data file, input.dat. Input.dat should contain commands that define the actions that will be performed by the program.

The format of each command is:

COMMAND

Parameter1

Parameter2

.....

Where each new parameter should appear on a new line. In this manual, parameters that should be entered only for one specific value of other parameters are shown in square brackets.

Empty lines are thrown away during the input.

a. Optional commands.

In this section we will describe the commands that can be used to override the default values of various parameters.

SET_GYRO

Gyro

Assigns the value Gyro (double) to the gyroscopic coefficient γ . Affects the units in which frequencies and times are being presented. For example, Gyro=1.75D-2 makes all times to be in nanoseconds, and all frequencies – in GHz. Gyro=2.785D-3 corresponds to using GHz/2 π for the frequencies.

The value by default is 1.

SET_BETA

Beta

Sets the value of damping parameter β to Beta (double). The value by default is 0.

SET_CURRENT

Current

Cx

Cy

Cz

Describes the presence of a spin polarized current in the system. **Current** (double) - current's strength in Oe (1000 Oe approximately corresponds to 10^8 A/cm²). **Cx, Cy, Cz** (double) – x, y and z components of spin polarized current's polarization.

By default the spin polarized current has a zero magnitude (there is no current in the system).

SET_HRF**Type****[real(Cx)****imag(Cx)****real(Cy)****imag(Cy)****real(Cz)****imag(Cz)]****[real(Cx)****imag(Cx)****real(Cy)****imag(Cy)****real(Cz)****imag(Cz)****i]****[i2****Filename]****[icur****cx****cy****cz]**

Type (integer) is equal to 1 for a uniform r.f. field, 2 for a linear field, 3 for a field from a file, 4 for a field due to the r.f. spin polarized current.

The uniform field (Type=1) is defined by three complex components Cx, Cy, Cz (complex) that are entered as two real numbers each – real and imaginary part respectively. The rf field itself is then defined as:

$$\mathbf{h}^{(rf)} = \text{Re}\left(\left(\hat{x}C_x + \hat{y}C_y + \hat{z}C_z\right)e^{-i\omega t}\right)$$

Linear field (Type=2) is defined by the same set of components and in addition to them also by an integer i, which defines the coordinate upon which the r.f. field linearly depends: 1 for x, 2 for y, 3 for z. As an example, the field that linearly depends on x (i=1) will look like:

$$\mathbf{h}^{(rf)} = \text{Re}\left(\left(\hat{x}C_x + \hat{y}C_y + \hat{z}C_z\right)(ax + b)e^{-i\omega t}\right)$$

where a and b are selected in such a way that the field is 0 in the middle of the sample, positive on one side and negative on the other side.

If a file is being read (Type=3), then *i2* (integer) defines the file's type (currently only *i2*=1 – text file is being supported) and *Filename* (character) is the file's name. Text file has a format:

x y z real(cx) imag(cx) real(cy) imag(cy) real(cz) imag(cz)

where *x,y,z* are integers and the rest are real numbers corresponding to real and imaginary parts of parameters *cx, cy, cz* .

R.f. spin polarized current (Type=4) can be defined by its magnitude *icur* (double) and the coordinate components of its polarization – *cx, cy* and *cz* (double).

SET_VRML

Scvx

Scvy

Scvz

Vtype

Sets the parameters of VRML output: *Scvx, Scvy, Scvz* (integers) define over how many elements along *x, y* and *z* coordinates respectively the averaging should be performed, with the result being used in making a vrml file. *Vtype* (integer) defines whether the elements should be presented “as they are” (*Vtype*=1) or as slices alongside the *z* direction (*Vtype*=1). By default *Scvx=Scvy=Scvz=Vtype=1*.

SET_SCALE

Sx

Sy

Sz

Sets the spatial averaging in the matrix creation (as defined in my thesis) to that among *Sx, Sy* and *Sz* (integers) cells along *x, y* and *z* axis respectively. The use of this command is recommended for only very advanced users. The values by default are *Sx=Sy=Sz*.

b. Describing the object.

In our case we do it by using a square lattice composed of cells (or discrete dipoles), and a *masque* which defines which lattice sites are occupied and which are not. It also tells which material occupies the site. Materials are characterized by two parameters – exchange stiffness and saturation magnetization.

First of all, we need to input such materials. It's done by invoking the command

LOAD_MAT

N

 A_1 M_{s1} A_2 M_{s2}

....

Here N(integer) is the total number of materials, M_s and A (double) are saturation magnetization and exchange stiffness for each of the materials.

If we already have a masque created by previous calculations, we can load it by using:

MASQUE_LOAD**Type****Filename**

Type(integer) tells whether the file filename(character) is a binary (Type=2), text (Type=1) or modified text (Type=3) file.

Text file format is as follows:

```
I Xmax Ymax Zmax
x y z masque(x,y,z)
```

where I is a total number of elements in the system, X_{max} , Y_{max} , Z_{max} (integers) are maximum values of x,y and z coordinates respectively, in number of cells, x,y,z(integers) are coordinates, masque(x,y,z) (integer) is a material occupying the site with such coordinates – 0 corresponds to an empty cell, positive integers correspond to materials, numbered in the same order they are in the LOAD_MAT statement.

Modified text file (Type=3) is a text file with the following format:

First 4 integers describe the total number of elements (zero and non-zero) in the masque, maximum values of x, y and z coordinates in the lattice.

Following them there are depictions of the masque for each of the z coordinates, for example:

z=0

0001000
 0201002
 0002000

in this example we have empty sites denoted by 0, and those occupied by the first and second materials (as given in LOAD_MAT) – by numbers 1 and 2.

One can save the masque by using:

MASQUE_SAVE

Type

Filename

Where the parameters have the same meaning as before: Type(integer) tells whether the file filename(character) is a binary (Type=2), text (Type=1) or a modified text (Type=3) file, only now the file is being saved rather than read. For the descriptions of the file formats please see the section on MASQUE_SAVE command.

If one needs to calculate the demagnetization tensor (see later), or to use any of the shape-defining commands, one first needs to use:

SET_CELL

Cellx

Celly

Cellz

Which tells the program the cell's dimensions used – cellx, celly, cellz (double), in centimeters.

If one just needs to study a simple body (currently rectangular prisms, rings, ellipses, discs and spheres are being supported), the following command can be used:

STUDY_SHAPE

Type

[R_{ext}

R_{int}

H]
[R_{ext}]
[D_x
D_y
D_z]
[
R_{ext}
R_{ext2}
H]

Where Type(integer) defines the shape of the object – 1 corresponds to a disc/ring, 2 to a sphere, 3 to a slab, 4 to an ellipse.

If Type=1: R_{ext}, R_{int}, H (double) define external, internal diameters and the height, in centimeters. The object lies in x-y plane.

If Type=2 R_{ext} defines the sphere's diameter in centimeters.

If Type=3, D_x, D_y, D_z (double), in centimeters, define the length of the slab along x,y and z axis.

If Type=4 R_{ext} is the diameter along the semimajor axis, R_{ext2} – diameter along the semiminor axis, H is the thickness, in centimeters.

Another way to create a desired object is to use the following two commands:

MASQUE_CREATE

X_{max}
Y_{max}
Z_{max}

Where , X_{max}, Y_{max}, Z_{max} (double) are maximum values of x,y and z coordinates respectively, in centimeters.

Then one can add shapes by using:

ADD_SHAPE

Type
[R_{ext}
R_{int}

H
Cx
Cy
Cz
Mat]
[R_{ext}
Cx
Cy
Cz
Mat]
[x1
x2
y1
y2
z1
z2
Mat]
[R_{ext}
R_{ext2}
H
Cx
Cy
Cz
Mat]

This command can be issued multiple times in order to define a sample consisting of many objects.

Type(integer) defines the object's shape – 1 corresponds to a disc/ring, 2 to a sphere, 3 to a prism, 4 to an ellipse. All values are given in centimeters. In all cases Mat (integer) is the material for the object. Mat=0 corresponds to the empty space, and can be used to introduce holes into objects.

Type=1:

R_{ext}, R_{int}, H (double) are external, internal diameters and the thickness of the disk/ring.

Cx,Cy,Cz (double) are coordinates of the ring's or disc's center.

Type=2:

R_{ext} (double) is sphere's diameter, C_x, C_y, C_z (double) are coordinates of the sphere's center.

Type=3:

Slab is defined as spanning all the space between coordinates x_1 and x_2 (all coordinates are real numbers with double precision by default) on x axis, y_1 and y_2 on y axis, z_1 and z_2 on z axis.

Type=4:

R_{ext} , R_{ext2} , H (double) are semimajor, semiminor diameters and the thickness of the disk/ring.

C_x, C_y, C_z (double) are coordinates of the ring's or disc's center.

MASQUE_INFO

This command prints the total number of occupied cells in the system, followed by the number of cells along x, y and z directions.

After the object's creation one has to calculate the demagnetization tensor. This is done by issuing

Calculation of the demagnetization tensor is performed by issuing a command:

TENSOR_CREATE

Nx

[Ny

Nz

Rcut]

If N_x (integer) equals -1 the demagnetization tensor is calculated by using Newell's formulas for uniformly magnetized prisms. Individual cells in this case are uniformly magnetized prisms and the magnetic fields are macroscopic. If $N_x > 0$ the demagnetization tensor is calculated by assuming that each individual cells is composed from magnetic dipoles, total number $N_x * N_y * N_z$, where N_x , N_y and N_z (integers) define how many dipoles are along x,y and z axis. The field is microscopic (Lorentz local field), but all physical characteristics – resonant frequencies, switching times etc. should be the same. When macroscopic fields are being used,

the total energy of the system includes certain self-energy terms, therefore the total energy can be positive.

Second method is usually faster and in certain cases can be slightly more precise than using Newell's formulas. We recommend using something like a dipole per nm for N_x , N_y and N_z (so that 5x5x25nm cell will have $N_x=5$ $N_y=5$ $N_z=25$), with no less than 5 and no more than 30 for each of N_s .

R_{cut} (double) is a cut-off diameter in centimeters – above this value instead of calculating the average field per cell, only the field at a central point is calculated. This diameter should usually be slightly bigger than twice the lattice (or discretization) constant. The Newell's method on the other hand is more straightforward and easier to use.

In order to add the uniaxial anisotropy into the discussion, one must use:

ADD_ANISOTROPY

Mat

Cx

Cy

Cz

K

Mat (integer) is the material with the anisotropy, **Cx**, **Cy**, **Cz** (double) are components of a unit vector that defines the anisotropy's direction. **K** is the anisotropy coefficient.

This corresponds to adding the effective anisotropy field, defined by $H_A^{eff} = \frac{K(\mathbf{m} \cdot \hat{\mathbf{H}}_A)}{M_s^2} \hat{\mathbf{H}}_A$,

$\hat{\mathbf{H}}_A$ is a unit vector, defined by **Cx**, **Cy**, **Cz**.

Alternatively,

ADD_ANISOTROPY_FIELD

Mat

Cx

Cy

Cz

K

can be used. In this case K (double) gives not the anisotropy coefficient, but the value of effective anisotropy field itself – in Oe.

One can save the calculated tensor by using:

TENSOR_SAVE

Mat

Type

Filename

Separate materials should be saved/loaded to/from different files. Mat (integer) defines which material's tensor must be saved, Type (integer) defines whether the text (Type=1) or binary (Type=2) should be used. Text mode should not be used for permanent storage due to the precision loss, but rather only to check the correctness of the tensor. Filename (character) is the name of the file being used for saving into.

To load the tensor, one must use:

TENSOR_LOAD

Mat

Type

Filename

Everything is the same as in the case of TENSOR_SAVE, but now the tensor is being loaded from a file.

TENSOR_MACRO

This command is somewhat unusual – it calculates and saves (text version into macro.txt, binary version into macro.dat) the tensor corresponding to the whole object, i.e. the table that shows the average field inside the occupied sites in the object as function of the object's magnetization. In order to avoid huge self-fields due to the exchange interaction, exchange stiffness in LOAD_MAT preferably should be zero.

This routine can be useful when a macrospin approximation is being desired for a complex system – in this case it produces the demagnetization tensor for such a macrospin.

c. Magnetization dynamics and other related procedures.

First we need to start with choosing an initial configuration for the magnetization. It can be done by loading it from a file:

M_LOAD

Type

Filename

Type (integer) is 1 for the text, 2 for binary mode, Filename (character) is the file's name. Again, operating with text files involves certain loss of precision.

At any time similar command can be used to save the magnetization distribution:

M_SAVE

Type

Filename

[i]

This command saves the magnetic moments into the file Filename. The only difference from M_LOAD is that Type=4 leads to saving of magnetic moments into VRML file, Type=1 – text file, Type=2 – binary file. VRML scene is created according to the parameters given by SET_VRML.

The distribution of the magnetization can also be initialized by using:

INIT_M

Type

[Cx

Cy

Cz]

Type (integer) defines whether the initial configuration is a saturated state (Type=2), when all dipoles are pointing in the same direction, in this case Cx, Cy, Cz (double) define such direction (the vector with coordinates Cx, Cy, Cz does not need to be a unit vector). If Type is equal to 1,

no other parameters need to be entered – the initial configuration is defined as a vortex lying in x-y plane, with the center coinciding with that of the object itself. If Type is equal to 3, the initial configuration is random.

External d.c. magnetic field is set by using:

ADD_EXT_FIELD

```
i
[Cx
Cy
Cz]
[j
Filename]
```

If i (integer) is equal to 1, C_x , C_y and C_z give the vector components of a uniform external d.c. field. If i is equal to 2, the field is read from the file `Filename` (character), with j defining the file's type (1 for text, 2 for binary).

It is important to mention, that this command can be issued many times, with the resulting field being a sum over the contributions created by each of such commands – for example some external file can be loaded, and then, an additional field added by issuing `ADD_EXT_FIELD` for the second time.

`INIT_M` or `M_LOAD` should always precede `ADD_EXT_FIELD`. By default (if the command is omitted) external field is assumed to be 0 everywhere.

In order to find an equilibrium configuration, one can use so called relaxation, or modified Powell method; another option is to use Runge-Kutta to integrate Landau-Lifshitz equation in the presence of damping.

Both are done by using:

RELAX

```
I
Maxerror
Koeff
```

This command finds the equilibrium configuration. *I* (integer) is a number of configurations that are going to be used. If *I* is equal to 1, only the configuration that was previously defined by `INIT_M` or `M_LOAD` is going to be used. If *I* is more than 1, for the first configuration

I (integer) defines how many times the calculation is going to be performed. If *I* is equal to 1, the relaxation is performed only once. If *I* is greater than 1, the relaxation is performed *I* times, with the lowest energy configuration being selected from the results. Initial configuration is the same as defined by `M_LOAD` or `INIT_M` for the first run, and random for the other runs.

For each of the runs, error and energy are being displayed at the standard output (computer screen). When error is below `maxerror` (double, we recommend the value $1D-5$) the process is stopped. Magnetization of the sample is written into the file `Im.dat` where *I* is the run number, starting with 1. Magnetic field of the final configuration is written into `Ih.dat`, and `I.wrl` contains the VRML picture of the configuration obtained.

Energies of the runs are written into the file `fort.101`.

Finally the lowest energy configuration is selected among the runs and used as a final equilibrium configuration, together with the corresponding magnetic field.

`Koeff` (double) is a coefficient controlling convergence of the method. It can have values in between of 0 and 1, with bigger values being recommended (0.8 to 1.0).

`FFT` is used automatically for large samples.

If `koeff` is less than zero, instead of the relaxation method, simple Runge-Kutta integration is being performed using the timestep set by `SET_TIMESTEP` and the damping coefficient defined in `SET_BETA`. Runge-Kutta integration is terminated when the difference in energies between two consecutive runs is less than `maxerror`. This method can be more effective when there are strong d.c. fields present in the system.

H_SAVE

Type

Filename

Type (integer) is 1 for the text, 2 for binary mode, 3 for VRML, Filename (character) is the file's name. Using text files involves certain loss of precision.

Analogously magnetic fields can be loaded using:

H_LOAD**Type****Filename**

Type (integer) is 1 for the text, 2 for the binary mode, Filename (character) is the file's name.

H_AVERAGE**Cx1****Cy1****Cz1****Cx2****Cy2****Cz2**

Averages the field in occupied sites in a rectangular prism with the lower left corner with coordinates Cx1, Cy1, Cz1 (double) and the upper right corner with coordinates Cx2, Cy2, Cz2 (double). The resulting field is displayed on the screen.

SET_TIMESTEP**Timestep**

If one wishes to use Runge-Kutta integration (in particular the next command, EXEC), one first has to set the timestep. This commands can set the timestep explicitly to the value given by Timestep (double), in the units defined by SET_GYRO, or, if timestep<0 it will run a program that finds the optimal value of timestep automatically. The method used to calculate the optimal value is overly conservative; in most cases it returns somewhat smaller timestep value than the ones traditionally used.

EXEC**Endtime****Vtime****Filename****[omega]**

Proceeds with Runge-Kutta integration for parameters set by SET_CURRENT, SET_HRF etc., for the time period Endtime (double). If Vtime (double) is greater than zero, every Vtime interval the magnetization distribution is being saved into VRML file, therefore the evolution of the magnetization can be observed. The files are numbered, starting with 1 (1.wrl). Besides that, the program outputs into the file Filename (character) time, average magnetization along x, y and z axis and the system's total energy. If r.f. field is present in the system, omega (double) indicates the r.f. field's frequency.

d. Solving for eigenvalues and eigenvectors.

In order to use the procedures described in this section one needs to have:

1. Local magnetic moments and magnetic fields.
2. Masque.
3. Tensor.
4. Materials' parameters (LOAD_MAT).

SOLVE_MATRIX

Nmodes

Upon completion, this command will produce all the eigenvalues and eigenvectors. Only the Nmodes (integer) lowest frequency modes are stored; the rest are discarded. This is usable since most of the high-frequency modes typically do not affect experiments. If Nmodes equals -1 all of the modes are stored. Right and left eigenvectors, eigenvalues and scaling factors (as set by SET_SCALE or SET_SCALE_R2) are saved in files eigen.dat and eigenl.dat.

SOLVE_MATRIX_AR

Type

Nmodes

Ncv

[Rcut]

Uses Arnoldi algorithm to calculate selected low-frequency eigenvectors and eigenvalues. Only the right eigenvectors are being computed, with an assumption that they are more or less orthogonal to each other. While for the sake of calculating the absorption spectra it is usually a good assumption (5% error in the absorption peaks' magnitude is usually tolerable), it is unwise to use the results of SOLVE_MATRIX_AR in VEC3 analysis. SOLVE_MATRIX should be used instead.

If **Type** (integer) equals 1 simple Arnoldi algorithm is being used. For the effective performance, the maximum number of dipoles should be kept around 10,000 or so maximum. **Nmodes** is a desirable number of modes. **Ncv** (integer) is a performance regulating parameter, connected with a number of Krylov vectors used in the simulation. Usually the number should be

at least 50, but no larger than $2.5 \cdot N_{\text{modes}}$, with an exception of the case when more than 5000 dipoles are being used. In this case, slightly smaller numbers, starting with 30 and reaching maximum $2 \cdot N_{\text{modes}}$ should be used. Typically, the number of modes effectively participating in the absorption of uniform r.f. field does not exceed 25 or so low frequency modes.

If **Type** (integer) equals 2, sparse algorithms will be used with an assumption that the dipole-dipole interaction between dipoles separated from each other by the distance larger than **Rcut** (double). This is a very effective method to calculate the spectra of very large thin film particles. **Rcut** typically can be set to be around 200nm (200D-7), but one needs to check to make sure that the calculations are well converged.

EIGEN_SAVE_TEXT

Type
Filename

Filename (character) is the file's name. Both the eigenvectors and eigenvalues are going to be saved in a text file Filename. Type (integer) defines whether right (Type=1) or left (Type=2) eigenvectors are going to be saved.

In order to print only the eigenfrequencies:

OMEGA_PRINT

Filename

Prints all of the resonant frequencies to the file Filename.

EIGEN_LOAD

Reads the eigenmodes from files eigen.dat and eigenl.dat.

CALC_C

Calculates the oscillation strengths.

PRINT_C**Filename**

Prints the oscillation strengths to the file `Filename` (character). Format is:

Mode number (from 1 to N, where N is a number of macrocells), real part of oscillator strength, imaginary part of oscillator strength, real part of eigenfrequency, imaginary part of eigenfrequency. Imaginary part of oscillator strength should be positive and corresponds to the absorption. In case if imaginary part is negative, it is most likely due to discretization or some numerical errors – unless the initial configuration is not an equilibrium.

PRINT_ABS**Filename****Omegastep**

Prints the absorption graph to the file `Filename` (character), `omegastep` (double) is the step size for the frequency axis, in the units defined by `SET_GYRO`. The frequencies chosen for the graph lie mostly in between of the lowest and highest excited modes. Absorption is printed in arbitrary units (the object's volume is not being taken into account), however r.f. different fields acting on the same object will produce the absorption that are going to be in the same units, and therefore can be compared to one another.

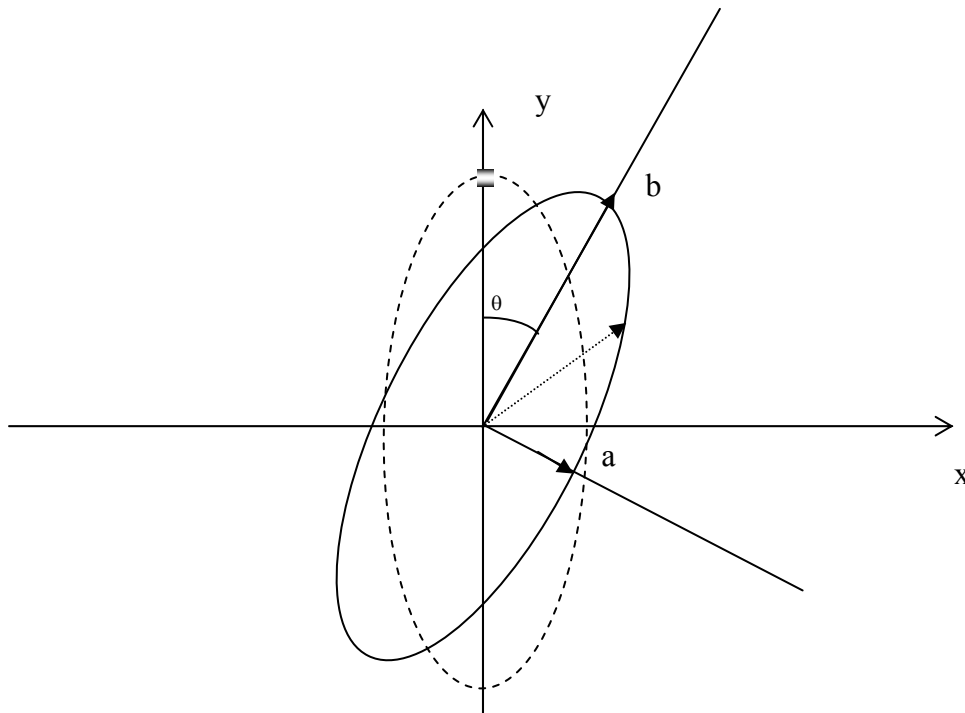
VEC3**Nmodes****Endtime****Vtime****Fname****[omega]**

Similar to **EXEC** with one difference – obtained results are expanding onto the basis of the eigenmodes (should be either calculated via `SOLVE_MATRIX` or loaded via `EIGEN_LOAD`). Proceeds with Runge-Kutta integration for parameters set by `SET_CURRENT`, `SET_HRF` etc., for the time period `Endtime` (double). Besides that, the program outputs into the file `Filename` (character) time, average magnetization along x, y and z axis and the system's total energy. If r.f. field is present in the system, `omega` (double) indicates the r.f. field's frequency. Unlike **EXEC**, the results are projected onto the basis of first `Nmodes` eigenvectors every `Vtime` (double) time. `Modes.dat` and `phases.dat` files contain absolute values and phases of such projections. More powerfully excited modes are than presented in `modej.dat` files, where j is a

mode number (from low to high frequencies), and collectively in modes01.dat and modes03.dat (those with maximum excitation higher than 10 and 30% of the maximum excitation observed).

e. Displaying the eigenmodes.

As a result of solving the eigenvalue equation we have a set of modes, each of them represented by 2 complex numbers per lattice site. These 4 numbers uniquely define the oscillation of magnetic moments in the plane, perpendicular to the direction of equilibrium magnetization. In general such oscillation orbits are ellipses:



Here you can see that the ellipse can be uniquely identified by another 4 parameters:

- a. Semimajor axis b .
- b. Semiminor axis a .
- c. The angle in between of semimajor axis and y axis.
- d. The initial phase or initial position of an oscillating magnetic moment.

There are a few routines that allow one to make plots and drawings of the modes:

PRINT_PHASES

I

I2

I and I2 are integers. This command prints the information about the modes with numbs in between of I and I2 (modes are arranged by numbers from low to high frequencies, starting with 1).

The print contains:

X, y and z coordinates of a cell (in number of cells, for example a cell can be identified as 1st cell along x axis, 2nd cell along z axis and 0st cell along z axis), angle formed by an initial position of a magnetic dipole with local y axis, angle formed by semimajor axis with local y axis, magnitude of semiminor and semimajor radiuses (normalized to 1).

DRAW_MODES

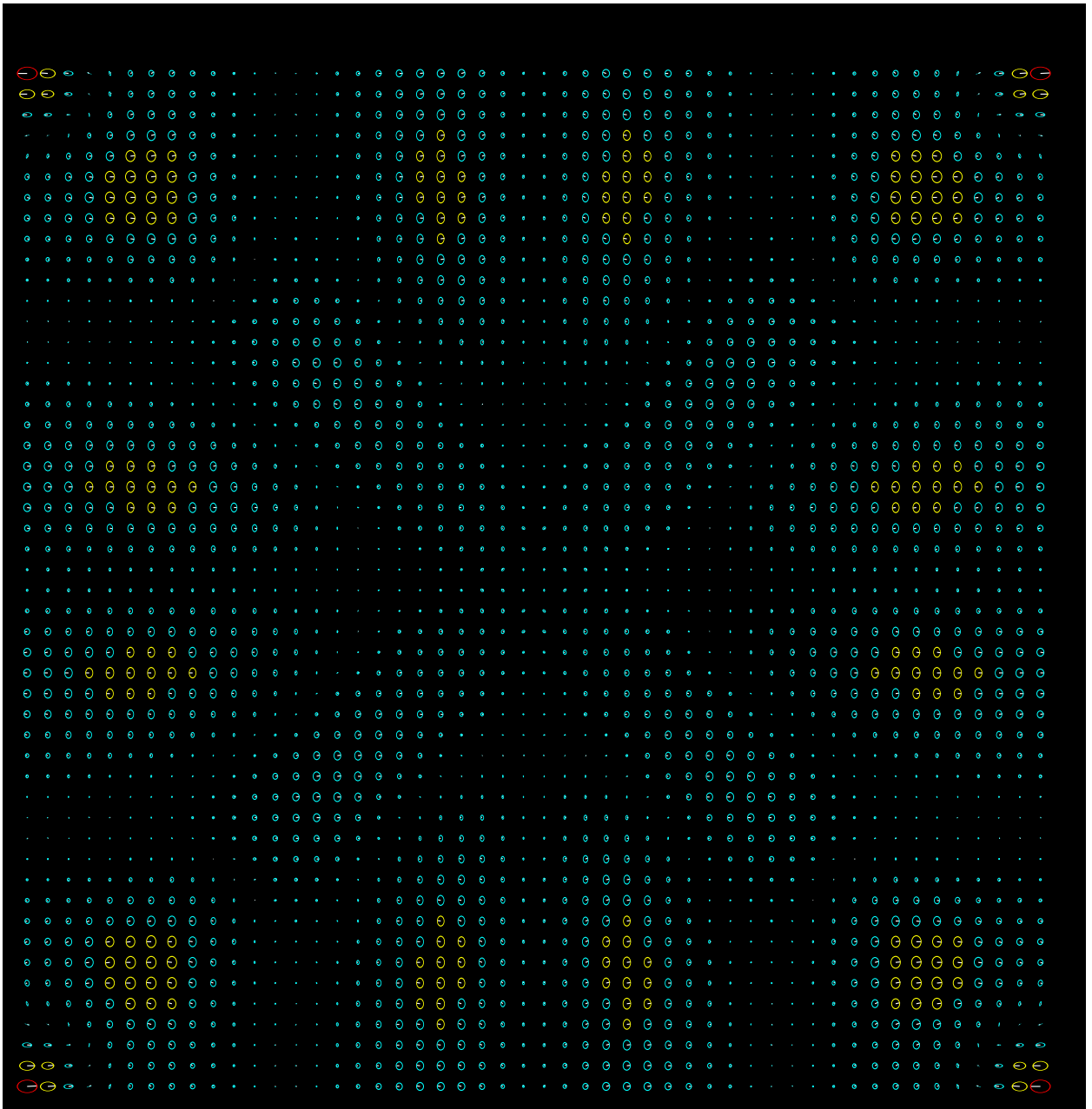
I

I2

I and I2 have the same meaning as in PRINT_PHASES, but this time the modes are represented as a two-dimensional Encapsulated PostScript (EPS) graph. File's name consists of a 0, followed by dash, followed by the mode number (from low to high frequencies, same number as given by OMEGA_PRINT, PRINT_C or PRINT_PHASES). Planes with different z coordinate are represented as different slices.

One has to take in mind that the precessional orbits lie in different individual planes, each of them is perpendicular to the local equilibrium magnetic moment. In the graphs, the ellipses depict the orbits of each of the spins, with the principal radii proportional to the precession amplitude. The color also signifies the amplitude of precession – red for the maximum amplitude, yellow for the amplitudes above 50% of this maximum amplitude, and blue for the lower amplitudes. The white lines emanating from the orbit centers identify the relative phases of the oscillations; the orbits shown here are actually located in different coordinate systems defined by the local direction of magnetization, as noted above.

Example:



DRAW_MODES_P

P

F

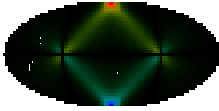
I

I2

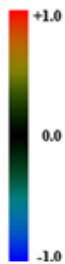
Instead of printing modes in local coordinates, this routine prints projections of the modes with numbers between I and I2 (integers) onto a global Cartesian coordinate system. P (integer) defines which projection is going to be printed – 1 for x, 2 for y, 3 for z coordinates. F (double) is initial phase. The best value to start with is probably 0, but if one suspects traveling spin

waves to be present in the system, one can try other values (from 0 to 2π) to see the modes' evolution.

Example:



Here you can see the modes in an ellipse. Colors denote sign and magnitude of excitations:



DRAW_MODES_E

Magnitude

Can only be executed after the `CALC_C` command.

Prints global and local coordinate representations for all the modes that have absorption higher than $\text{Magnitude}(\text{double}) \times \text{maximum absorption}$. I.e. $\text{Magnitude}=0.05$ means that all of the modes that have absorption strength of at least 5% of maximum will be plotted. Different coordinates are distinguished by numbers in the files' name: 0 – for local coordinates, from 1 to 3 – x, y and z coordinates.

We had a very positive experience using programs like Adobe PhotoShop CS to work with the graphs created by RKMAG

END

Finishes the program's execution. It's appearance at the program's end is mandatory.

Final Notes.

Can be found at <http://www.rkmag.com>

If you have a question please feel to contact us at rivkin@rkmag.com

If you experience a problem while executing our program, please include your input.dat file, your CPU model, and all the output produced by the program (this can be obtained by executing `rkmag &>out.txt`).

We greatly appreciate your input !