A brief introduction to using SNAXS (Simulating Neutron And X-ray Scans).

D. Parshall[1]

[1]*National Institute of Standards and Technology,*
NIST Center for Neutron Research, Gaithersburg, MD 20899[*]

(Dated: March 3, 2016)

Abstract

SNAXS is a tool for condensed-matter scientists to Simulate Neutron And X-ray Scans. It is a bridge between theory and experiment, which allows the user to take a first-principles calculation as an input, and calculate the anticipated intensity as a function of momentum and energy transfer: $S(\mathbf{Q},\omega)$. SNAXS is currently a command-line interface (CLI) tool, but could be extended to use a GUI if interest warrants. While written in Matlab, most features are compliant with Octave (an open-source language almost identical to Matlab), so users and facilities may use most features without facing any licensing costs.

SNAXS can simulate $S(\mathbf{Q},\omega)$ as constant-Q scans (aka energy scans), constant-energy scans (aka Q-scans), or as colorplots seen using time-of-flight instruments. In addition, SNAXS can calculate and display dispersion relations, and the generalized density-of-states.

SNAXS makes use of several existing tools (e.g. ResLib, PhonoPy, anapert). At its core, SNAXS extracts eigenvectors from a first-principles calculation (using either phonopy or anapert), and then calculates the expected $S(\mathbf{Q},\omega)$, taking into account the instrument resolution (using either ResLib, or other tools written for SNAXS). As a related bonus, SNAXS can produce animations of the phonon polarization vectors at arbitrary $\mathbf{Q}$.

I. DISCLAIMER

This software was developed at the National Institute of Standards and Technology at the NIST Center for Neutron Research by employees of the Federal Government in the course of their official duties. Pursuant to title 17 section 105 of the United States Code this software is not subject to copyright protection and is in the public domain.

The SNAXS software package is an experimental program for calculating scattering intensity as a function of reciprocal space and energy. NIST assumes no responsibility whatsoever for its use, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. The use of certain trade names or commercial products does not imply any endorsement of a particular product, nor does it imply that the named product is necessarily the best product for the stated purpose. We would appreciate acknowledgment if the software is used.

II. QUICK TUTORIAL

You can download a virtual machine (generated using VirtualBox), which already has Octave, SNAXS, and phonopy installed. This is recommended for first-time users of phonopy. It does require installing VirtualBox (www.virtualbox.org), but this is fairly easy on most systems. Unfortunately, the file is around 8 GB, and can't be used on FAT32 filesystems (an older Windows format which has a maximum file size of 4 GB). The virtual machine file is currently stored at:

http://magnetism-school.org/static/2015/SNAXS_demo.vdi

To get the latest code, you can open a terminal within the SNAXS directory, and input:

```
git pull origin master
```

A. Startup

From within Matlab or Octave, navigate to the directory containing snaxs.m, and type the following on the command line:

```
snaxs(EXPtof)
```

SNAXS will load the EXP structure for a hypothetical time-of-flight experiment, using a sample of MgB_2. This should bring up a welcome screen and display the main menu:

```
Using hexagonal basis

------------------------------------------------------------
|  Welcome to SNAXS! (Simulating Neutron And X-ray Scans)  |
|      SNAXS is a CLI wrapper for anapert and phonopy      |
|           SNAXS was written by Dan Parshall              |
|            anapert was written by Rolf Heid              |
|            phonopy was written by Atsushi Togo           |
|----------------------------------------------------------|

Checking the EXP structure ...
    The experiment type is: tof
    NOTE in "initialize_EXP" : no value given for sample.b; using sample.a
     ... the EXP structure seems to be OK
Initializing XTAL from POSCAR calculation file
     ... the XTAL structure seems to be OK

MAIN MENU:
  1) Simulate E-scan at fixed Q
  2) Simulate Q-scan at fixed E
  3) Simulate slice of S(q,w)
  4) Plot dispersion (energy vs Q, no intensities)
  5) Calculate phonon density-of-states
 *** Perform TAS full convolution
  8) Get eigenvectors (and energies) for single Q-point
  9) Check current parameters, crystal data, etc.
  x) Exit

 Enter your choice:
```

One option is not available because this is not a triple-axis experiment. Any other option may be selected.

B. Energy scans

From the main menu, enter "1" to begin simulating energy scans. SNAXS will ask for a user input, but displays the default value to be used if no input is given. Press "Enter" to use the default value of $\mathbf{Q}$= [0, 3.85, 0]. The resulting graph is an intensity plot of a constant-Q scan (see Figure 1). The red points are the phonon energies, and the blue line is the naive intensity taking into account the sum of all phonons using a simple gaussian. This is similar to a "cut" in Mslice.

Try entering a new value for $\mathbf{Q}$, such as [0, 4, 0] (note that it is not neccessary to use commas or square brackets to input the vector). Entering that value will update the default.

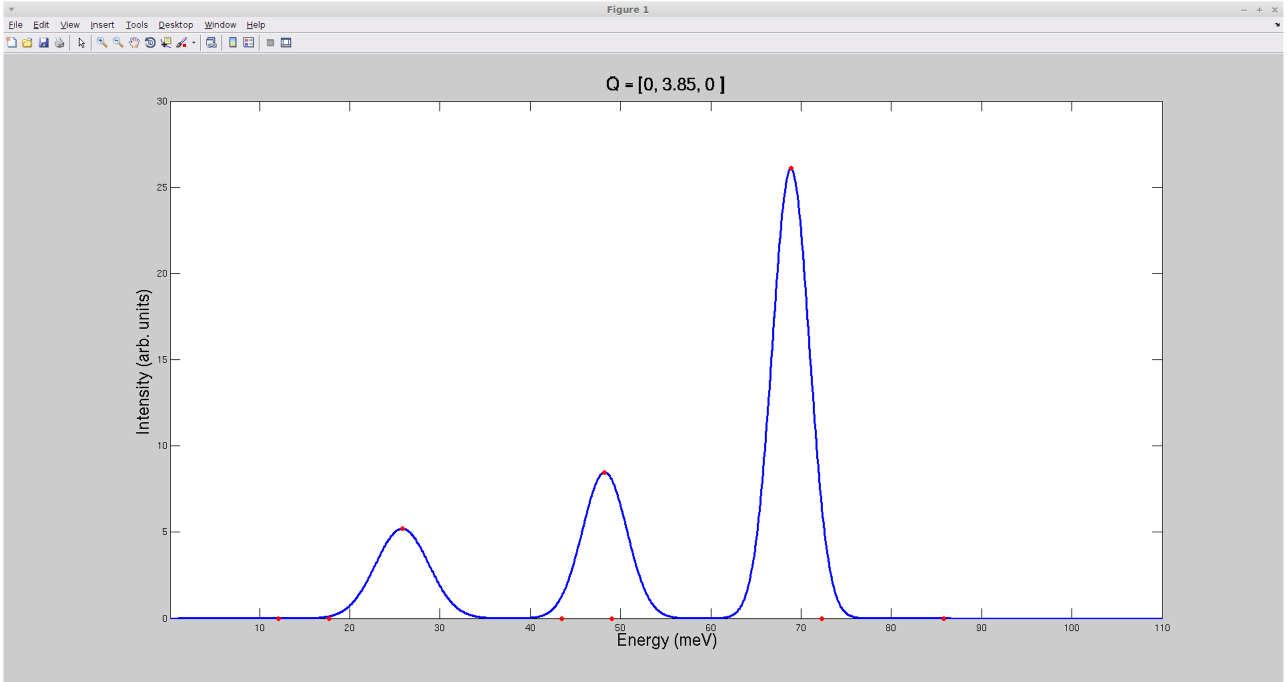

FIG. 1. Phonon intensity at constant Q = [0, 3.85, 0].

```
=== Simulating E-scan at fixed Q ===
Input Q as "H K L", or "x" to exit
 The default is [ 0 3.85  0 ].
 Enter vector (or "x" to exit): 0 4 0
```

The change in energies is fairly substantial, which can make it challenging to determine if the same modes are involved. To see the evolution of the modes across the zone, one can plot S($\mathbf{Q},\omega$), as per the following section.

C. Slices of S($\mathbf{Q},\omega$)

Type "x" for the main menu, then "3" to simulate a slice of S($\mathbf{Q},\omega$).

```
=== Simulating slice of S(q,w) ===
Input Q_min as "H K L", or "x" to exit
 The default is [ 0  0  0 ].
 Enter vector (or "x" to exit):
Input Q_max as "H K L", or "x" to exit
 The default is [ 0  4  0 ].
 Enter vector (or "x" to exit):
Input number of points (odd for symmetric scans)
 Default is 401
 Enter value (or "x" to exit):
```

```
 Calculating...
 This seems to be a time-of-flight experiment
Elapsed time is 1.385886 seconds.
 MATLAB COLORMAP
 ... finished!
```

The result is shown in Figure 2. Because of the $1/\omega$ term, the intensity of the acoustic phonons near the Bragg peak can overwhelm the optical modes. Type "l" or "L" to toggle between linear and logarithmic plotting. The current figure will automatically be replotted using the new scaling.

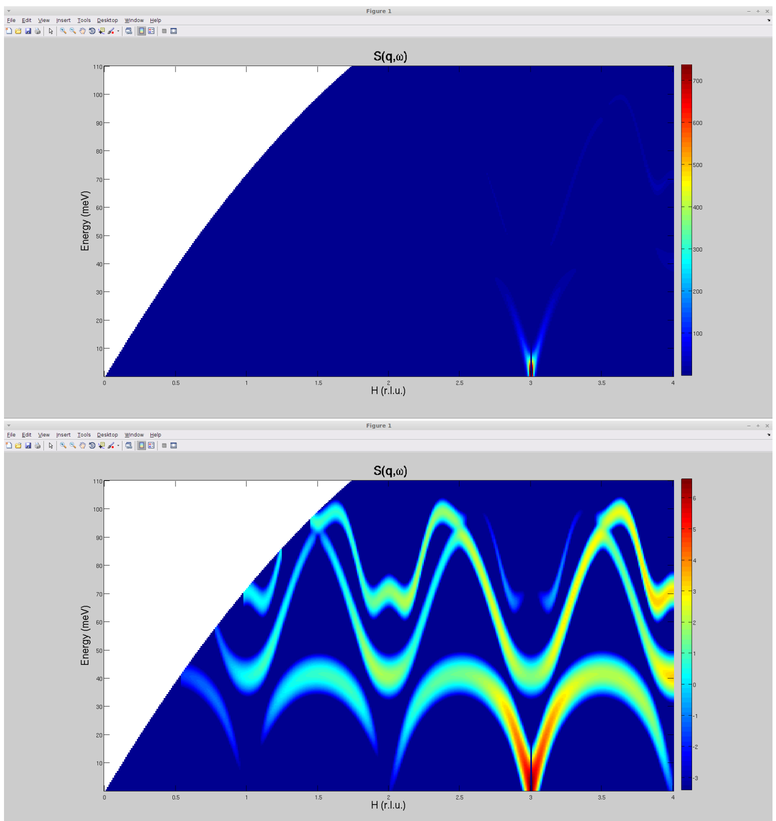

FIG. 2. Phonon intensity plotted using both linear (top panel) and logarithmic (lower panel) scaling. The dark line at H=3 is because phonons with energy zero (i.e., at the Bragg peak) are being ignored.

D. Dispersion relations

In the previous section, the visible phonons are restricted by symmetry to a subgroup of all the phonons along this vector. The complete dispersion along some direction can be plotted with Option 4 from the main menu. The default values in the dispersion menu are the same as those in the S($\mathbf{Q},\omega$) menu- a change in one will be carried to the other. So selecting Option 4 and accepting all defaults will plot the dispersion along the direction of S($\mathbf{Q},\omega$) that was just calculated. SNAXS makes no attempt to sort the modes according to symmetry, so the presence or absence of any branch crossing should be interpreted with care. It may be prudent to replot the data with a greater number of points (see Figure 3).

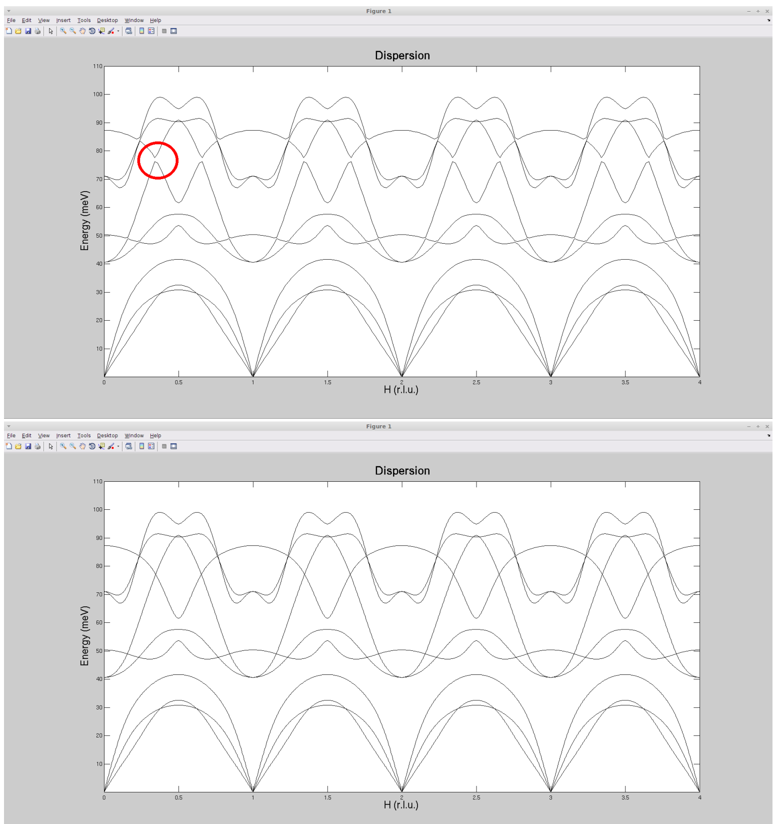

FIG. 3. Dispersion plots all phonon modes, without regard to symmetry or intensity. Use a fine step size to avoid false anticrossings.

E. Phonon density-of-states

While S($\mathbf{Q},\omega$) can only be measured with single-crystal samples, powder samples can be used to measure the isotropically-averaged phonon density-of-states S($|\mathbf{Q}|,\omega$). The intensities of the branches are weighted by the mass and scattering cross-section of the atoms involved to produce the "generalized" phonon DOS. The gDOS can be calculated by SNAXS, using Option 5 from the main menu. The user is prompted for a minimum and maximum energy (with the defaults set by the values for INFO.e_min and INFO.e_max in DEFAULTS.m). Next is a prompt for the smearing width due to the finite energy resolution (for phonopy just a single value; for anapert, the energy resolution width at the minimum and maximum of the energy range). The default values are calculated based upon the parameters in the EXP structure. Finally the user is prompted for the number of sampling points for Brillouin zone mesh (which must be three integers: N_x, N_y, N_z). More points means a denser sampling grid, but longer calculation time. The output is shown in Figure 4.

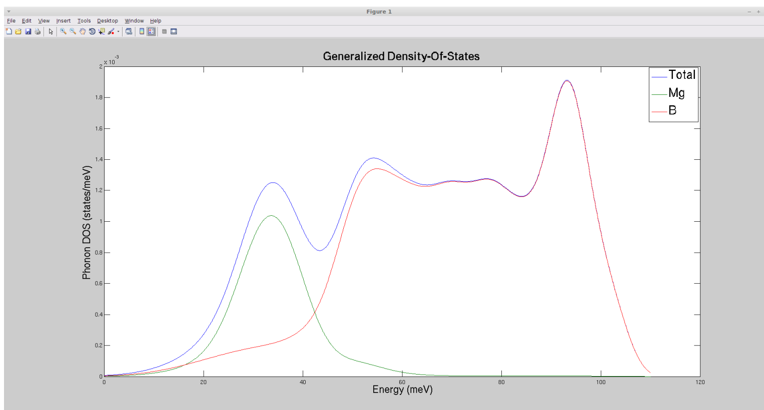

FIG. 4. Phonon Density-Of-States, including atom-specific partial DOS.

F. Eigenvectors at arbitrary Q

Understanding the eigenvector of a phonon mode (essentially, the pattern of atomic displacements) can provide insight into the deeper physics. To see an animation of the displacement, select Option 8 from the main menu (a stillshot is in Figure 5). This feature uses several graphics features only compatible with Matlab; it is not available when using Octave.

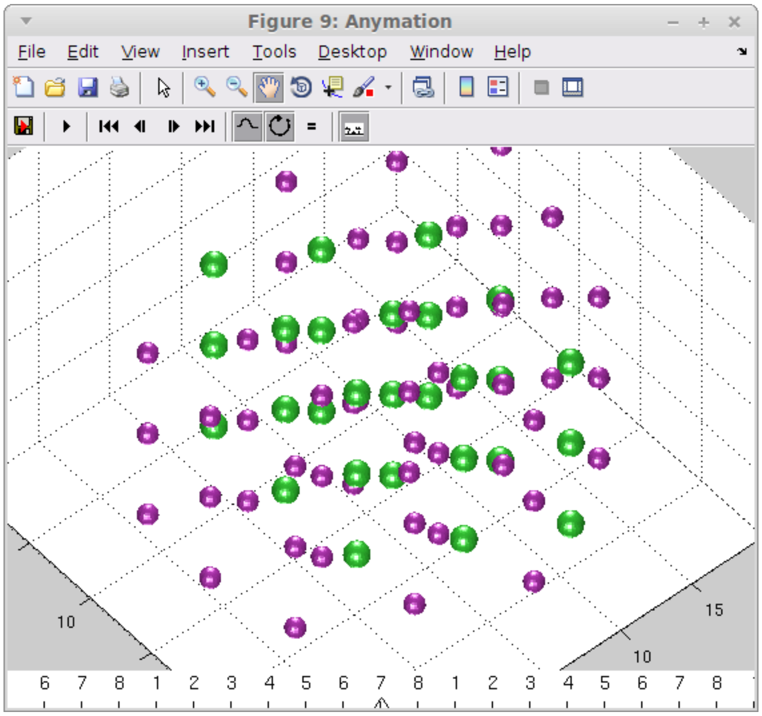

FIG. 5. Atomic displacement (proportional to eigenvectors) may be animated.

G. Changing the user basis

MgB_2 has a hexagonal structure; up to this point we have been using the hexagonal basis, in which K is equivalent to H (from the center to the side of the hexagon). The correspondance beetwen the bases may be seen in Figure 6. SNAXS allows the user to define a basis and thus operate in orthogonal coordinates (in this case the K direction is towards the point of the hexagon, and H remains unchanged). The user can set a basis transform within the EXP file (if not given, it defaults to the identity matrix). With appropriate switching logic (see EXPtas for an example), multiple bases may be used. All of bases are belong to user.

Open the files “EXPtof.m” and “EXPtas.m” using a text editor to see one way to switch between bases. To use the orthorhombic basis, as well as explore features particular to triple-axis instruments, exit SNAXS and restart, loading the triple-axis EXP file via:

```
snaxs(EXPtas)
```

After loading, switch to logarithmic scale, then try plotting $S(\mathbf{Q},\omega)$ via Option 3. In Figure fig:ortho0K0, one can immediately see that the simulation is now using fixed-E_f,

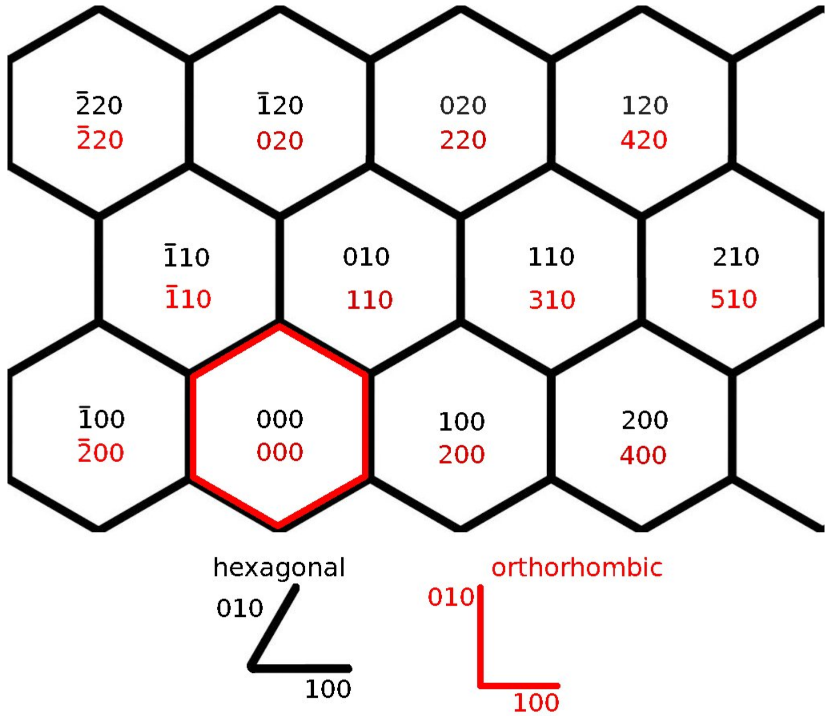

FIG. 6. Hexagonal (black) versus orthorhombic (red) coordinate systems in reciprocal space.

rather than fixed-E_i. In addition, the dispersion of the modes in the new orthorhombic basis is clearly different from the hexagonal 0K0 direction (which was shown in Figure 2).

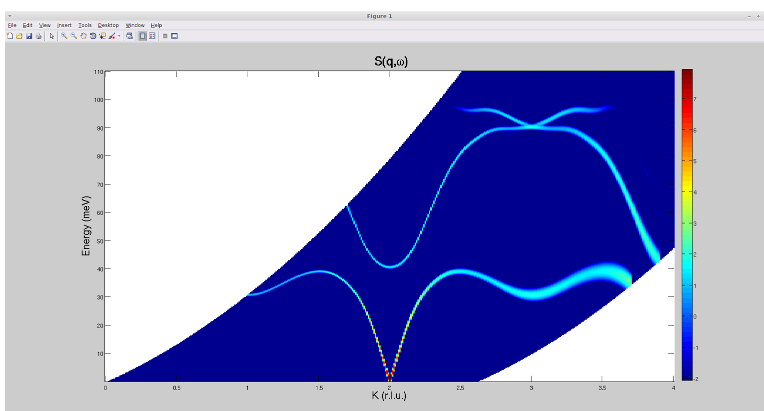

FIG. 7. S($\mathbf{Q}, \omega$) along the 0K0 direction in orthorhombic notation.

One phonon of particular interest is the E_{2g} mode, close to the Γ-point. This mode is particularly soft in MgB_2 due to the strong electron-phonon coupling (see Bohnen *et al.*, PRB 86 5771). Simulate an E-scan at $\mathbf{Q}$= [0, 3.85, 0]. The output is shown in Figure 8.

The plotting is still logarithmic, but can be toggled to linear. The size of the intensity scale

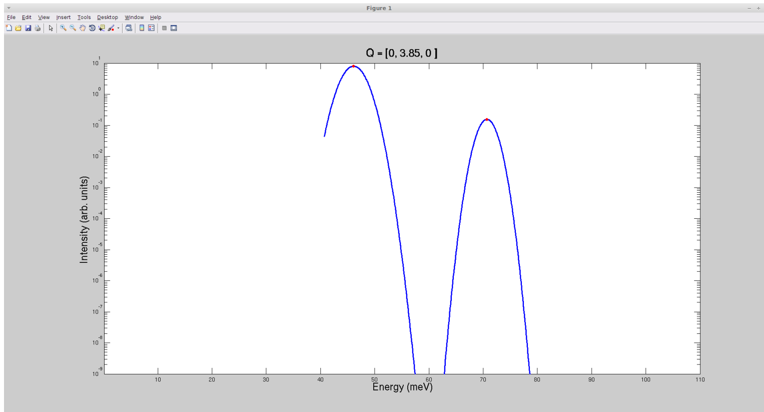

FIG. 8. Energy scan with logarithmic scaling. The range can be set using DEFAULTS.m, and is the same in both energy scans and S($\mathbf{Q},\omega$) plots.

is 10 decades, and can be set within "DEFAULTS.m" (along with many other parameters). Within the E-scan submenu, typing "i" will provide additional information about both the structure factor, and the translation between the conventional and primitive coordinates (this can be useful for confirming that EXP.basis_user is set correctly).

```
=== Simulating E-scan at fixed Q ===
Input Q as "H K L", or "x" to exit
 The default is [ 0  3.85  0 ].
 Enter vector (or "x" to exit): i
 NOTE: "user_vector.m" sent special output; may break things
 In primitive coordinates, this is : [ -1.925  3.85  0 ].
 Which translates to user basis of : [ 0  3.85  0 ].

 Mode Energy Structure_factor
  1)  10.54      0
  2)  16.03      0
  3)  23.10  66.79
  4)  42.92      0
  5)  46.11  31.62
  6)  49.37      0
  7)  67.87      0
  8)  70.69   1.59
  9)  86.15      0
```

H. Q-scans

SNAXS can simulate the intensity at a constant energy, from a starting **Q** to ending **Q**. From the main menu, select Option 2. To see the transverse acoustic phonons, select starting **Q**=[-0.2, 2, 0] and ending **Q**= [0.2, 2, 0].

```
=== Simulating Q-scan at fixed E ===
Input Q_min as "H K L", or "x" to exit
 The default is [ 0  4  0 ].
 Enter vector (or "x" to exit): -.2 2 0
Input Q_max as "H K L", or "x" to exit
 The default is [ 0  4  0 ].
 Enter vector (or "x" to exit): .2 2 0
Input number of points (odd for symmetric scans)
 Default is 401
 Enter value (or "x" to exit): 81
Input constant energy, or "x" to quit and enter a new Q-range
 Default is 6.5
 Enter value (or "x" to exit):
Calculating...
```

To make this calculation, SNAXS first simulates a slice of S($\mathbf{Q}, \omega$) as per Option 3, but then displays the intensity from just a single energy. Having calculated the slice at multiple energies already, SNAXS allows the user to examine new energies without recalculating (although changing the starting/ending points requires a recalculation). The output from such a scan is shown in Figure 9. This calculation considers only the energy width of the resolution, and ignores any contribution from momentum width. This is a reasonable approximation for time-of-flight and x-ray instruments (for which the momentum resolution is roughly constant). For triple-axis instruments, it is a useful heuristic, but not terribly accurate.

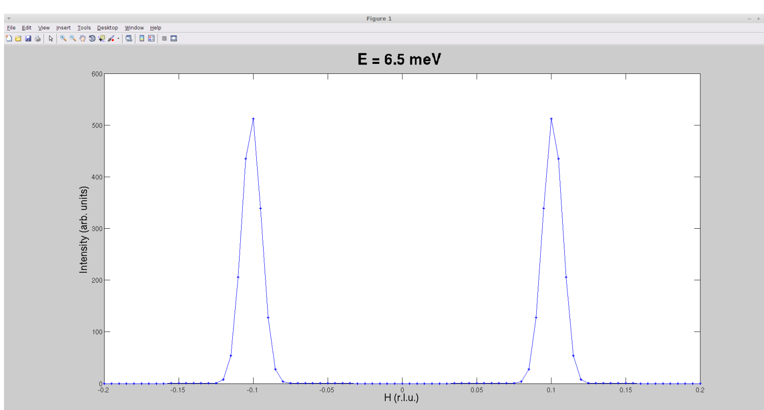

FIG. 9. Q-scan at constant energy, using naive momentum resolution.

I. Resolution convolution

The resolution function of a triple-axis spectrometer has some correlation between the energy transfer and the momentum transfer. When the slope of the resolution function lies along the same direction as the excitation dispersion, the spectrometer is "focussed". Using the constant-energy scan method described above, SNAXS treats every **Q** independently, and doesn't include the effects of resolution. To treat the resolution properly requires a convolution using the ResLib package.

To simulate the resolution-convolved transverse acoustic phonons, select Option 6 from the main menu. There is an option to simulate either a **Q**-scan or an energy scan, but for now select **Q**-scan. Use the same parameters for starting and ending **Q** as in the previous section, and set the "accuracy" parameter to 0. This will reproduce the constant-energy scan produced in the previous section. To see the convolved function, select a higher accuracy. The results of the convolution are automatically saved to file. The output is shown in Figure 10 (the left/right side are the defocussed/focussed sides).

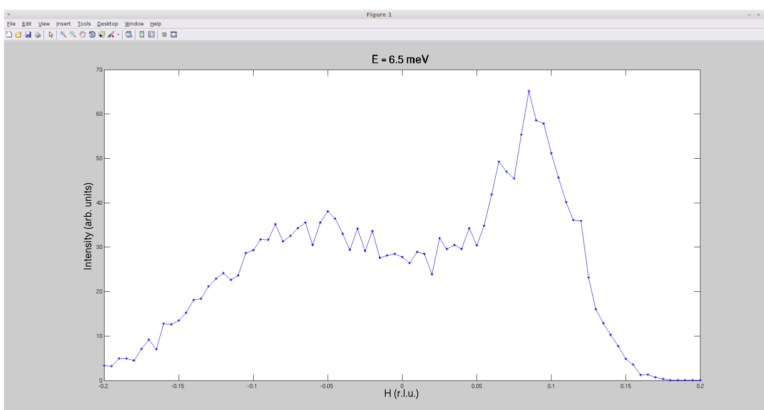

FIG. 10. Resolution-convolved Q-scan

J. Simulating x-ray experiments

The development of inelastic x-ray instruments capable of meV resolution allows phonon studies on even tiny single crystals. SNAXS can also simulate these experiments. In fact, the calculations are much faster, because there are no kinematic constraints (other than $\mathbf{Q}_{max}$, and the energy resolution is the same at all **Q** and energy transfer. The precise values for the energy resolution and maximum $\mathbf{Q}$ vary depending on choice of analyzer reflection and details of the particular instrument, but reasonable values are ≈ 2 meV and 7 Å^{-1}.

Simulating an x-ray experiment is a fast and convenient way to check dispersions, symmetry-allowed modes, etc. (with the caveat that the scattering cross-sections are different). The resolution function of IXS is approximately Lorentzian, which produces a Voigt profile after convolution with the phonon mode. The user can choose to calculate a Voigt function explicitly, or use a pseudoVoigt approxmation. This option is set in DEFAULTS.m, under INFO.voigt.

III. SNAXS IN DETAIL

SNAXS aids the experimenter by *combining* two separate factors: information about the material, and information about the instrument used for the experiment.

When conducting inelastic experiments, one measures intensity as a function of energy and momentum transfer. To simulate the result, SNAXS needs to have information both about a particular material (e.g., MgB_2), and a particular instrument configuration (e.g., BT-7). The material has properties such as dispersion, structure factor, etc. The instrument has properties such as collimation, fixed energy, etc.

As with anything, the devil is in the details.

A. Basic architecture

SNAXS is written in a fairly modular manner. The data and settings are grouped into several structures. These structures are generally combined into a meta-structure, PAR (for "parameters"), before being passed in or out of most subroutines. This makes it especially easy to hack SNAXS in new ways (example scripts are provided). The file "auto_PAR.m" generates a PAR structure strictly from the command line. That PAR structure can be passed into subroutines to calculate anything which can be done from the menu within SNAXS (as well as much more). For example, it is quite easy to write a routine which is given PAR and some Q as inputs, then updates PAR and calculates a single E-scan. That, in turn, can be used to provide a starting point for least-squares fitting of $S(\mathbf{Q}, \omega)$. It is also possible to simulate multiple slices of $S(\mathbf{Q}, \omega)$, and then integrate across those to simulate the binning performed in programs such as Horace or Mslice.

Calculating the force-constant matrix (FCM) is a standard part of *ab initio* calculations. There are multiple programs which take the FCM and calculate the phonon eigenvalues (energies) and eigenvectors (displacement vectors). SNAXS was originally written to interface with anapert (written by Rolf Heid, at the Karlsruhe Institute for Technology), and has been extended to interface with phonopy (written by Atsushi Togo, at Kyoto University). The basic process is that

1. The user requests some piece of information (e.g., an energy scan at fixed $\mathbf{Q}$).

2. SNAXS formats the request and writes it to a text file for use by the phonon calculator

3. SNAXS calls the phonon calculator, which writes the result to an output text file

4. SNAXS reads the output text file

5. SNAXS processes the data by including Bose factor, k_f/k_i weighting, etc.

6. SNAXS displays the final result to the user

For any simulation, SNAXS requires two input files: an EXP file (which provides information about the experiment configuration) and a FCM file (used by the phonon calculator program). Examples of EXP files have been included with this tutorial. In the case of anapert, the example FCM file is ANALYSIS_DATA.mgb2_q666; it contains basic information about the structure (the first 39 lines), as well as the force-constant matrices between various pairs of atoms (starting on line 10017). For phonopy, the basic structure information is stored in a POSCAR file, and the force-constant matrices in FORCE_SETS (SNAXS assumes these are stored in the same folder; pointing SNAXS to the POSCAR file is all that's required). It is possible to set a pointer to the FCM file using the parameter EXP.calculation_path.

SNAXS draws a distinction between information about the material (coming from the first-principles calculation, and stored in the XTAL structure), and information about the experiment (values set by the user, and stored in the EXP structure). Settings for plotting preferences are stored in PLOT, settings for various scan parameters in INFO, data from some set of simulations in DATA, and the eigenvectors are stored in VECS. SNAXS extends the idea of the EXP file, which was originally used in ResLib strictly for TAS experiments, to include TOF and x-ray measurements. There are demonstration files which show the required fields (EXPtof.m, EXPxray.m).

B. Using Phonopy

SNAXS is distributed with the anapert binary (either anapert.linux or anapert.exe), as well as the force-constant matrix file for MgB_2. This provides an easy way to get started, and make sure SNAXS is working on your system. However, most users will use phonopy, which is an open-source phonon calculator, written in Python.

In general, the user will need to collaborate with a theorist colleague, and have them produce a VASP-style output file (e.g., vasprun.xml). Phonopy must then be called using the "-f" option in order to generate the FORCE_SETS file. From that point, SNAXS can take over. For more information, consult the phonopy website at http://phonopy.sourceforge.net/

Currently, phonopy is only supported on *nix systems. The file phonopy.linux is a python script which is called by SNAXS. There are two EXP files which can help verify that phononpy is working on your system (EXP_phonopy_MgB2.m and EXP_phononpy_silicon.m). The MgB_2 file can be compared with the anapert results, and the silicon file can be compared with any condensed-matter textbook.

Phonopy is not officially supported in Windows. The official method for Windows users is to run Ubuntu inside a Virtual Machine. For those users, SNAXS should work perfectly. For users running Phonopy on Windows directly, they will need to modify "system_phonopy.m" and "system_cleanup.m" to handle the Windows case (and then please share that modified code).

One note: phonopy uses THz by default, so the conversion to meV is handled in 3 places: "read_phonopy_VECS", "read_PDOS", "write_phonopy". In addition, phonopy gives unit cell parameters in Å (whereas anapert uses the Bohr radius).

Phonopy uses POSCAR and FORCE_SETS as hard-coded inputs (i.e., those are the inputs that phonopy reads, without regard to which material they are for). To be sure that SNAXS is calling the files for the correct material, it automatically creates softlinks from the location of those files to the working directory. Those links are deleted each time SNAXS is started (to be sure that the correct file is being called). There is an internal check to be sure those are softlinks before deleting, but good practice is to keep those files in another directory, and point to the POSCAR file using EXP.calculation_path (SNAXS assumes that FORCE_SETS is in the same directory as POSCAR). This reduces the chance of accidentally deleting important files.

C. DEFAULTS.m

There are many parameters which need to be set, but are too trivial to bother the user with on every startup. Most of these are stored in "DEFAULTS.m". Examples are energy min/max/stepsize, and temperature. The user is encouraged to look through this file to see what options may be changed. Of particular interest is INFO.timescale, which is the time needed to make 1000 calculations of the TAS resolution function. This will vary by machine, but is used to estimate the total calculation time for convolution. One can also set the default paths for the anapert and phonopy binaries (this will override any other path which SNAXS might find).

D. Running on Windows

SNAXS should function on any platform: *nix, *nix-based Mac, or Windows (XP or later). SNAXS will write a text file (based on the user inputs) which serves as an input the phonon calculator (P_INP for anapert, QPOINTS for phonopy). SNAXS then calls anapert or phonopy using the Matlab "system" command. A consequence is that the user can examine the phonon calculator input file directly, and even call the calculator from the command line (i.e., not using SNAXS). This can aid in troubleshooting. The system is set using the built-in Matlab functions "isunix" and "ispc", and any time SNAXS makes a call to the system (calling phonopy, deleting temporary files, etc), it checks for the system, and uses the appropriate syntax. It has been tested extensively on linux Ubuntu and Mint. It has moderate testing on Mac (currently, the mac behavior is to use linux commands) with no reported issues. It has some testing on Windows XP and 7, but not as much. Windows requires administrator priviledge to make softlinks on the command line; the SNAXS workaround is to copy the FCM files, rather than link to them. This makes the SNAXS startup a few seconds slower on Windows. If you have some way of making links, you can modify "system_init" to use links instead.

E. Kitchen sink

Plotting of E-scans, Q-scans, dispersion, S($\mathbf{Q}, \omega$), resolution convolution, and density-of-states (options 1-6) work equally well in Matlab and Octave. Animation of eigenvectors (option 8) calls subroutines that are not supported in Octave.

The bulk of calculation time is spent calling anapert/phonopy. After that the primary time is calculating resolution functions (TAS in particular is slow, because the resolution function is so complicated). SNAXS tries to be moderately clever about this by using reduced coordinates where possible, only calculating modes in the user's requested range, etc.

While simulating single E-scans (Option 1 from the main menu), you can enter "i" to see a table of all the mode energies, structure factors, and intensities. Enter "q" to see the reduced $\mathbf{q}$ of the current position.

SNAXS supports non-orthogonal bases. The only routine which explicitly requires consideration of non-orthogonality is "calc_Q_sample", which calls the "GetLattice" and "star" routines from ResLib. One other routine calculates the modulus of Q "calc_Q_ang_prm", but it works with XTAL coordinates (which will generally mean the primitive basis), and converts to cartesian along the way.

It is important to keep the physical and crystal lattice constants separate- otherwise SNAXS may try to calculate the intensity of a phonon which is kinematically impossible. In that case, it will generate an imaginary component to the intensity, which make cause some subroutines to crash (but generally there are checks to prevent this). There are two places where the physical lattice constants (defined in EXP) intersect with the crystal lattice constants (defined within the calculation, stored in XTAL): "calc_Q_ang_prm", and "calc_Q_sample" (used in check_kinematics). Anapert uses atomic units, which are defined as a Bohr radius. That appears as a factor of 0.529177 A.U./Angstrom in those subroutines. In addition, ResMat uses a conversion with only 4 digits of precision- this can cause a crash with modes at the edge of kinematic accessibility. SNAXS uses the same conversion factor, but there are very occasionally rounding errors. To avoid complications due to this rounding error, all modes are checked prior to being passed to ResMat. Any modes which ResMat would calculate as inaccessible are removed.

The user basis is the transform from conventional to primitive coordinates. To keep the programming simpler and work in row vectors, it is defined as: C = P * B where C is conventional units, P is primitive units, and B is the user basis. Given three non-coplanar vectors C1, C2, C3 in the conventional basis (and their corresponding vectors in the primitive basis, P1, P2, P3), B can be calculated as:

B = inv([P1; P2; P3]) * [C1; C2; C3]

F. Troubleshooting on *nix systems

Most of the issues can be solved by making sure that anapert and phonopy are running correctly. Standard troubleshooting principles apply; isolate each step of the problem and test it. Some questions to get started:

1. Is SNAXS is generating an input file ("P_INP" for anapert, "Q_POINTS" for phonopy)? If an input file is created, but no output file results, then the problem is likely with the calculation program.

2. Does the calculation program (anapert or phonopy) run correctly from the command line?

   (a) Does the calculation run (and produce an output) from the command line if the calculation program and input files are all in the same directory?

   (b) Is calculation program is executable?

   (c) Is lapack is installed? (see below)

3. Is the path for the calculation file correct? Check location in DEFAULTS.m; be sure it is pointing to the correct calculation program (e.g., anapert.linux binary).

1. lapack

On linux systems, both anapert and phonopy require functional versions of the lapack library. Be sure that you have this installed. On Debian-based distros, just enter:

```
sudo aptitude install liblapack3
```

For anapert, you must also have a shared object "liblapack.so.3gf". To find it, try:

```
locate liblapack.so.3gf
```

If no output is returned, that means it doesn't exist with that label, so you'll need to make a link. Find the basic shared object via:

```
locate liblapack.so.3
```

Then navigate to the directory which contains "liblapack.so.3", and input:

```
ln -s liblapack.so.3 liblapack.so.3gf
```

This should solve any lapack-related issues.

2. Executable permissions

On *nix systems, a file must have appropriate permissions in order to execute. On most systems, you can navigate to the correct folder, then use the following commands (possibly prefaced by "sudo"):

```
chmod 777 anapert.linux
chmod 777 phonopy.linux
```

G. Gfortran library location on MacOS'

Mac users sometimes report an issue when Matlab doesn't know the location of the gfortran library. The symptom is that SNAXS will produce a sensible P_INP file, but the Matlab command line gives an error like:

```
dyld: lazy symbol binding failed: Symbol not found:
__gfortran_transfer_character_write
```

The solution is to tell Matlab where the appropriate library is, using something along the lines of:

```
>> setenv('DYLD_LIBRARY_PATH', '/usr/local/bin');
```

This should be input directly on the Matlab command line. You should obviously change the second input parameter to the correct path for your system.

IV. FUTURE UPGRADES

1. map of fixed energy transfer (see example script)
2. keep figures (crude implementation by setting PLOT.quiet)
3. linewidths due to lifetime
4. read in electron-phonon coupling of "read_phonopy_vecs"
5. overplot TAS resfunc on dispersion
6. correctly normalize convolutions
7. contextual help
8. UI for intensity scales
9. include sorting by irrep
10. automatic diagnostic scripts for phonopy and anapert
11. test script to be sure any changes within SNAXS propogate correctly
12.