

Chemical Spectroscopy Software

Yongqiang (YQ) Cheng

Computational Instrument Scientist

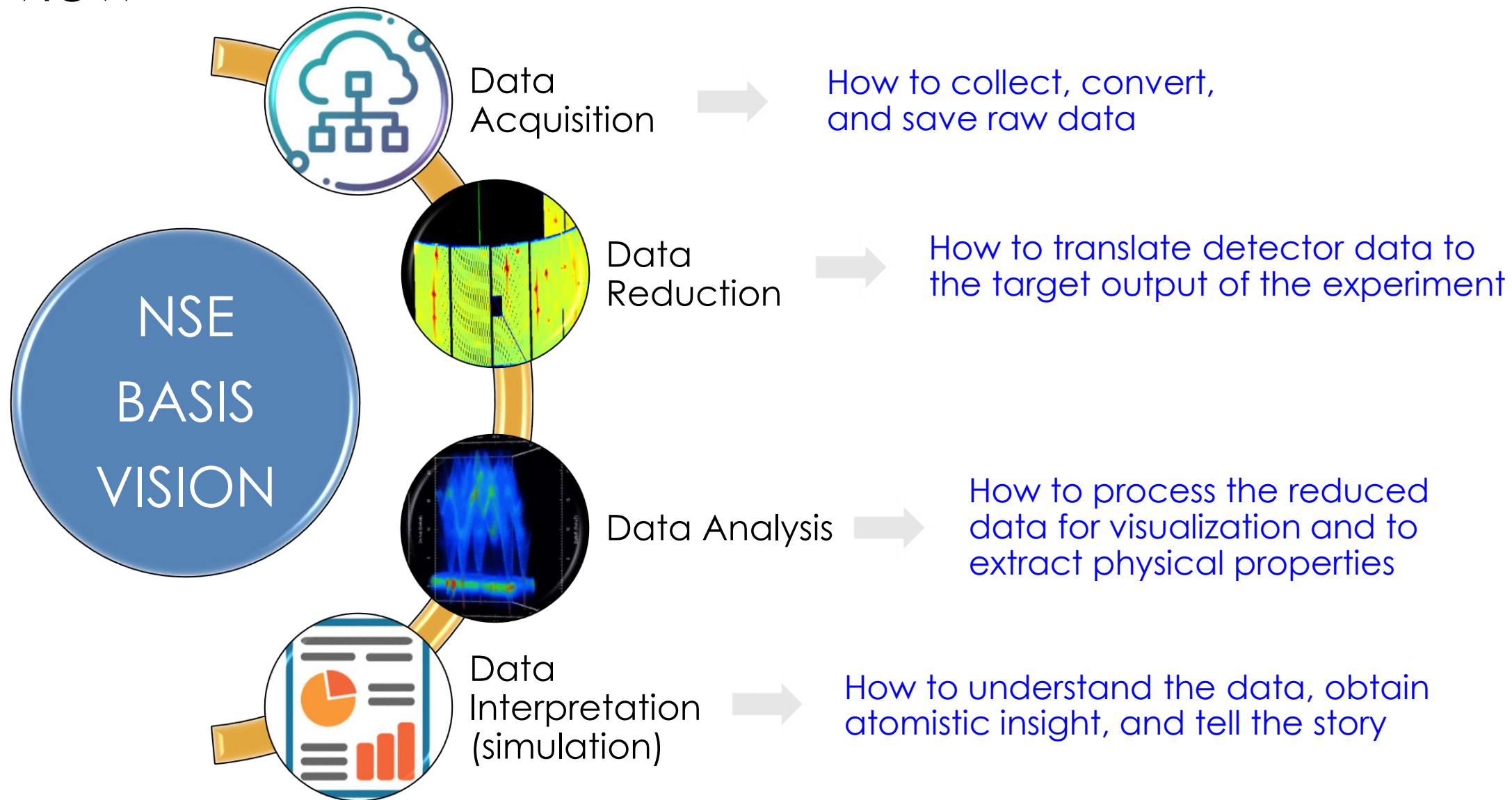
Spectroscopy Group

Neutron Scattering Division

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

2020 Review of the Instrument Suite for Spectroscopy
September 17-18, 2020

Overview



A brief summary of current status

- What is working
- What has been added recently
- What is needed in the near future



BL-15 (NSE)

- Data Acquisition
 - Jülich system
 - Transitioning to EPICS
- Data Reduction
 - DrSPINE (Data Reduction for SPIN Echo experiments)
- Data Analysis
 - Model fitting
- Data Interpretation
 - Molecular dynamics (MD) simulation



Efficient data extraction from neutron time-of-flight spin-echo raw data

P. A. Zolnierczuk,^a O. Holderer,^b S. Pasini,^b T. Koziellewski,^c L. R. Stingaciu^d and M. Monkenbusch^{e*}

^aForschungszentrum Jülich GmbH, JCNS Outstation, Oak Ridge, Tennessee, USA, ^bForschungszentrum Jülich GmbH, JCNS MLZ, Garching, Germany, ^cForschungszentrum Jülich GmbH, JCNS-1, Jülich, Germany, and ^dNSC, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA, *Correspondence e-mail: m.monkenbusch@fz-juelich.de

Received 15 May 2019
Accepted 2 August 2019

Edited by Th. Proffen, Oak Ridge National Laboratory, USA

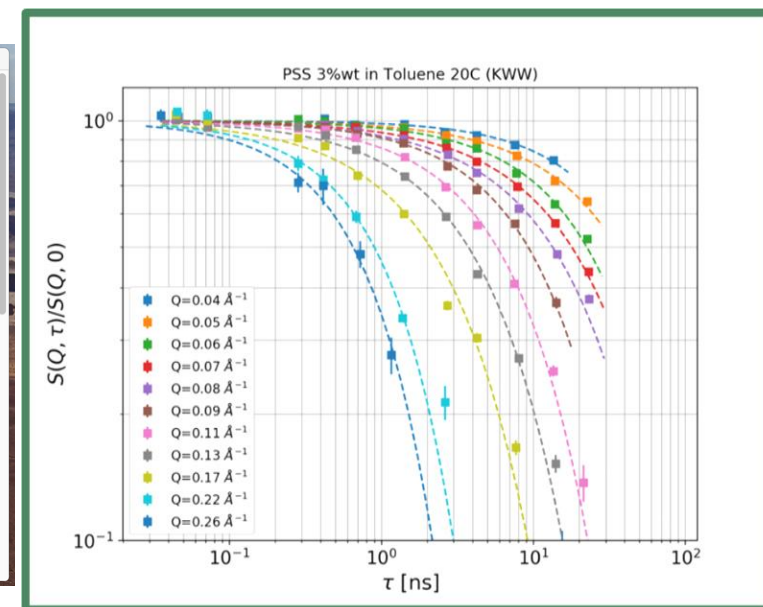
Keywords: neutron spin echo; NSE; spallation neutron sources; data reduction.

Supporting information: this article has supporting information at journals.iucr.org

Neutron spin-echo spectrometers with a position-sensitive detector and operating with extended time-of-flight-tagged wavelength frames are able to collect a comprehensive set of data covering a large range of wavevector and Fourier time space with only a few instrumental settings in a quasi-continuous way. Extracting all the information contained in the raw data and mapping them to a suitable physical space in the most efficient way is a challenge. This article reports algorithms employed in dedicated software, *DrSpine* (data reduction for spin echo), that achieves this goal and yields reliable representations of the intermediate scattering function $S(Q, t)$ independent of the selected 'binning'.

Zolnierczuk *et al.* J. Appl. Cryst. (2019). **52**,1022-1034 (DOI: 10.1107/S1600576719010847)

```
dr_macro_acns (~/Wrk/SpinEcho/ipts-19149) - VIM
macro
set a 1e-10
set ns 1e-09
bins tof custom 4 12 24 38
! resolution
read s8572.echo s8574.echo s8575.echo as res ! 8A
! buffer
read s8580.echo s8581.echo s8582.echo as bgr ! 8A
! PSS3 / Toluene 20C
read s8645.echo s8646.echo s8647.echo as sam tfac 0.8645 ! 8A
match all
fit all
histo q custom 0.04/a 0.05/a 0.06/a 0.07/a 0.08/a 0.09/a 0.12/a 0.14/a 0.20/a 0.25/a 0.35/a
histo tau nbins 20 min 0.001*ns max 100*ns log
collect bgr 1 volfrac 1.0
plot sqt model kww summary
```



BL-2 (BASIS)

- Data Acquisition
 - EPICS completed March 2019
- Data Reduction
 - Mantid (automatic and manual)
- Data Analysis
 - DAVE (PAN) or QCLIMAX
- Data Interpretation
 - MD simulation

BASISReduction input dialog

Multiple-file BASIS reduction for its two reflections.

RunNumbers: 72579

DoIndividual

NoMonitorNorm

GroupDetectors: None

NormalizeToFirst

Reflection Selector

ReflectionType: silicon111

EnergyBins: -120,0.4,120

MomentumTransferBins: 0.3,0.2,1.9

MaskFile: /SNS/BSS/shared/autoreduce...

Normalization by Vanadium

DivideByVanadium

NormalizationType: by Q slice

NormRunNumbers: 70739

NormWavelengthRange: 6.24,6.3

? Keep Open

MD simulations for NSE and BASIS

- Examples are few and the connections are indirect
- A bottleneck for instrument productivity and the impact of the publications
- Case-by-case (large-scale) MD models, challenges for automation and general user support
- Lack of proper tools to convert large-scale MD trajectories into neutron scattering functions (Sassena, nMOLDYN, LiquidLib)
- Software development priority for NSE and BASIS

BL-16B (VISION)

- Data Acquisition
 - EPICS (since 2015)
- Data Reduction
 - Mantid (automatic)
- Data Analysis
 - Mantid scripts and algorithms
- Data Interpretation
 - Lattice dynamics (LD) or MD
 - OCLIMAX

```
9 #=====control block=====
10 file_structure = 'Flat' # IPTS: IPTS subfolders. Flat: flat structure, all data under the root dir
11 weight = 1 # 0: from raw file. 1: from reduced file. 2: from log file. 3: from error
12 save_flag = 0 # 0: output not saved. 1: processed spectra saved in nxs and ascii
13 root_dir = 'C:/Users/yyc/Documents/Data/VISION' # root directory to the data files
14 save_dir = 'C:/Users/yyc/Documents/Data/Mantid/data/ascii' # needed only for save_flag=1
15 #-----
16 raw_subdir = 'nexus' # needed only for file_structure='IPTS' and weight=0
17 red_subdir = 'shared/manualreduce' # needed only for file_structure='IPTS'
18 log_file = 'E:/Data/VISION/VIS_logbook.csv' # needed only for weight=2
19 #=====end of control block=====
20 #=====global parameters=====
21 scalex = 1.0; scaley = 1.0 # scaling parameters for x (energy transfer) and y (intensity)
22 shiftx = 0.0; shifty = 0.0 # shift along x and y (after scaling)
23 bin = '-2,0.025,5,-0.005,1000' # binning parameters (meV)
24 smooth = 0 # smooth parameter NPoints (0 for no smoothing)
25 align = '0,0,0,0' # alignment params (meV): el_low,el_high,ins_low,ins_high (e.g. '-0.5,0.5,50,200)
26 unit = 'meV' # unit for output workspaces/files, 'cm-1' for wavenumber
27 bose = False # False: no bose correction even temperature is given
28 #=====end of global parameters=====
29 #=====data block=====
30 # the global parameters can be overwritten in the data block (redefined for a particular dataset)
31 # d.append( DataSet('16502-', 'b', temperature, bin=' ', smooth=, align=, scalex=, scaley=, shiftx=, shifty=
32
33 # Cu-faujasite_2nd_blank_Al_conflat_can_slits_20x40_at_5K
34 d.append( DataSet('48292-', 'b', 5) )
```

Script (for batch processing with consistency and reproducibility)

GUI as Mantid algorithms (for ad hoc manual processing)

VisionLoadMultipleFiles input dialog

This algorithm loads multiple reduced nxs files and merge them.

Run_numbers	34964-
Weighting_method	pchg from reduced file
File_structure	Flat
Root_dir	C:/Users/yyc/Documents/Dat...
Raw_subdir	nexus
Reduced_subdir	shared/manualreduce
Logbook	/SNS/VIS/shared/VIS_team/V...
Rebin_data	<input checked="" type="checkbox"/>
Rebin_param	-2,0.025,5,-0.005,1000
Smooth_data	<input type="checkbox"/>
Smooth_param	5
Unit	cm-1
RenameOutputWorkspace	<input type="checkbox"/>
Output_name	out

? Keep Open Run Close

Recent Development

- DrSPINE (Piotr Zolnierczuk)
- QCLIMAX (Timmy Ramirez-Cuesta)
- OCLIMAX (YQ Cheng)



Traditional methods

- Minimization, least squares algorithms
 - Sequentially fitting QENS functions to data
 - Parameters collected
 - Plot parameters vs Q
 - Fit parameters to functionality
-
- Laborious, time consuming, error-prone
 - Difficult to track the history of the data

QCLIMAX



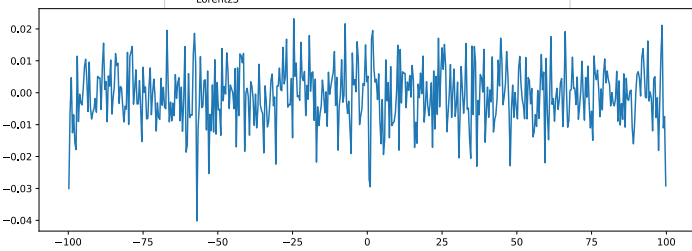
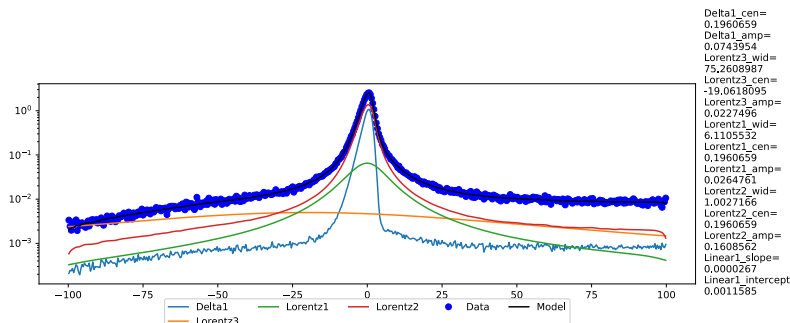
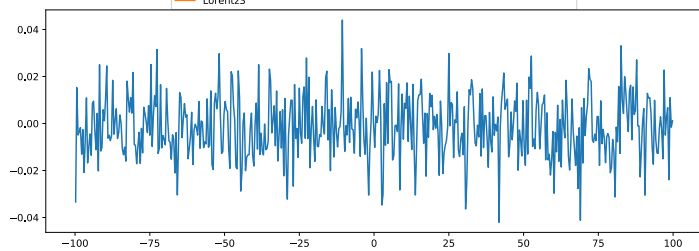
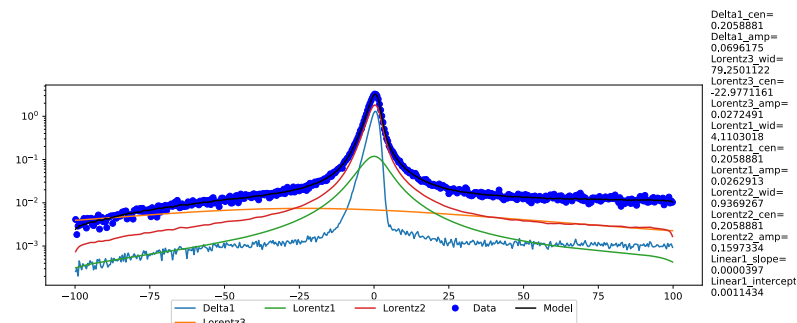
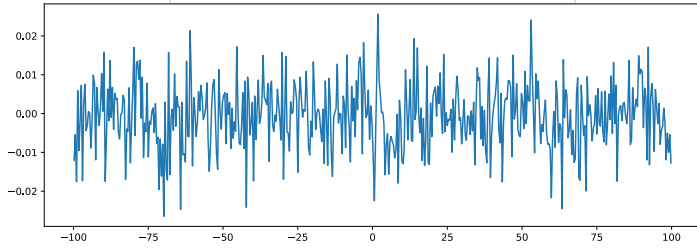
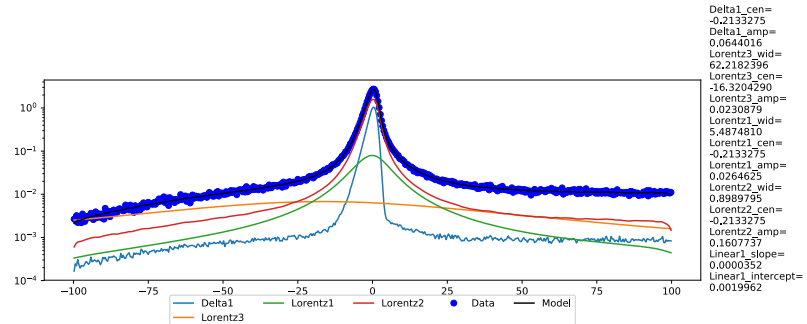
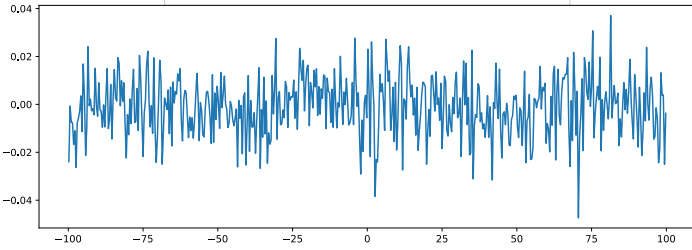
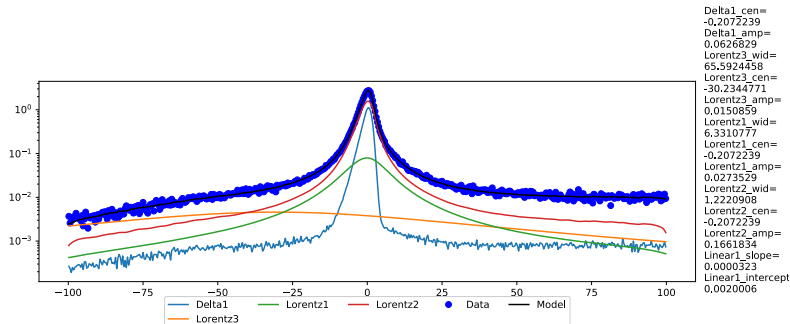
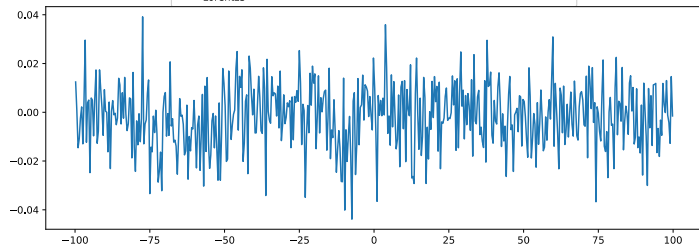
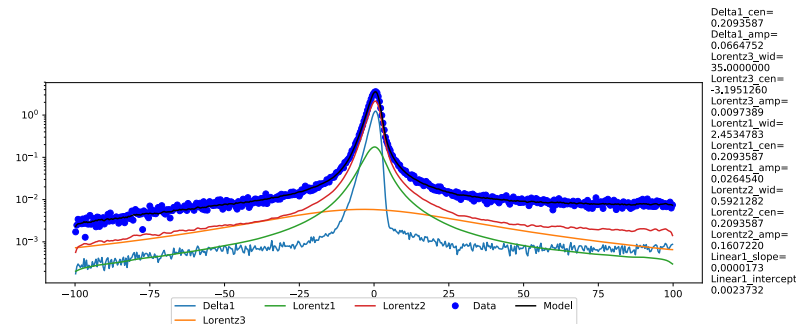
**Non-Linear Least-Squares Minimization
and Curve-Fitting for Python**
Release 0.9.6

Matthew Newville, Till Stensitzki, and others

Mamontov E., Smith R.W., Billings J.J., Ramirez-Cuesta A.J., "Simple analytical model for fitting QENS data from liquids", Physica B: Condensed Matter, 566, 50-54 (2019).

Courtesy: Timmy Ramirez-Cuesta

QCLIMAX example



Global fitting of the data

Fitting data using parameter while imposing constraints overt this parameter:

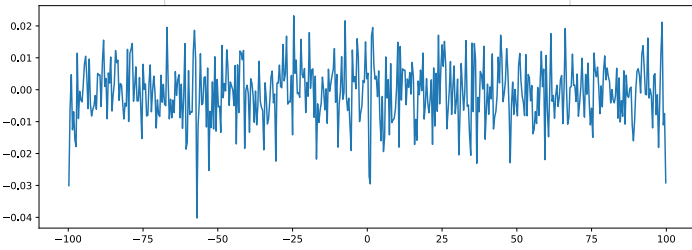
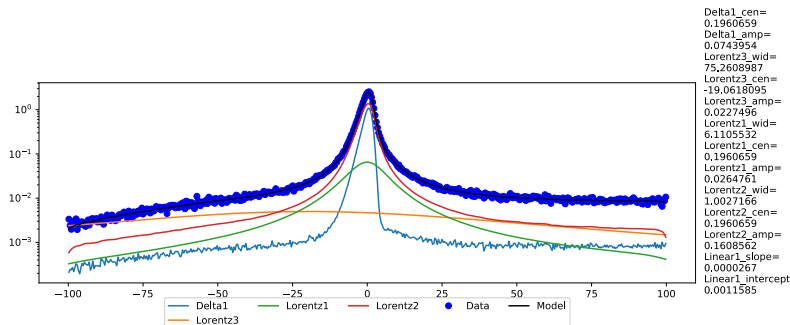
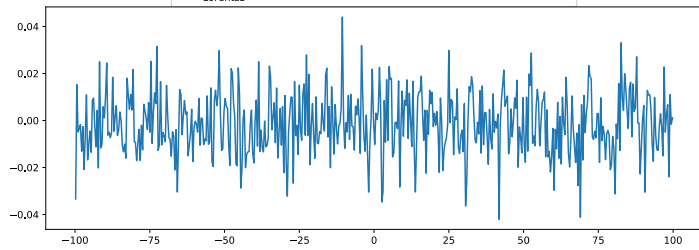
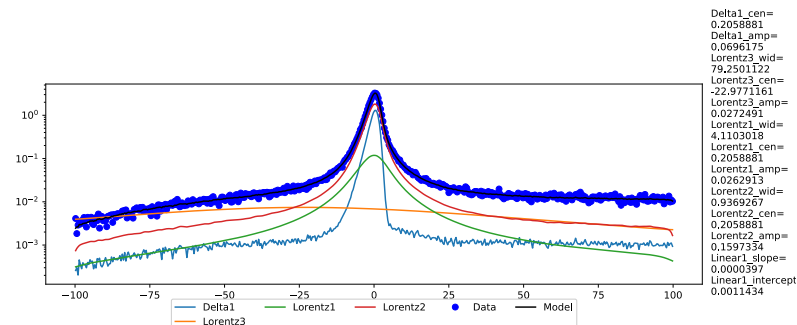
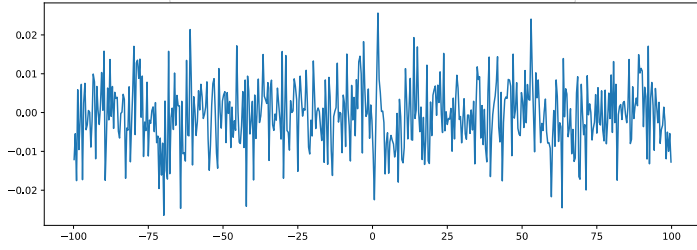
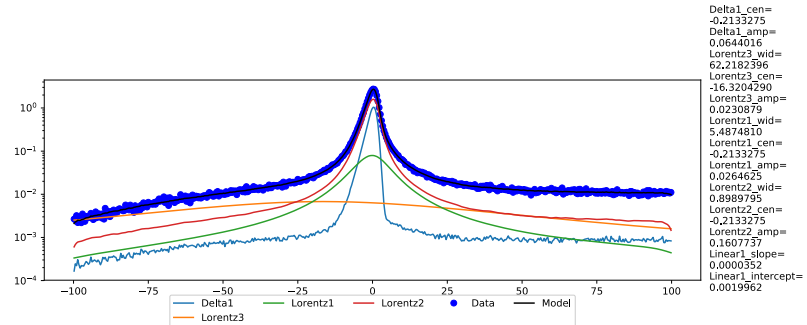
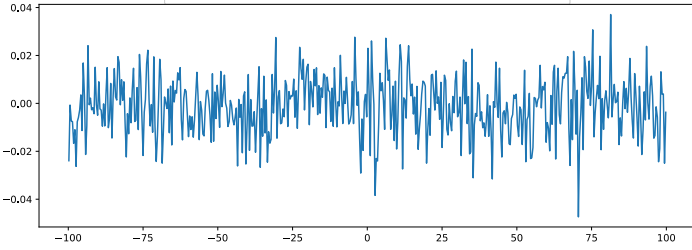
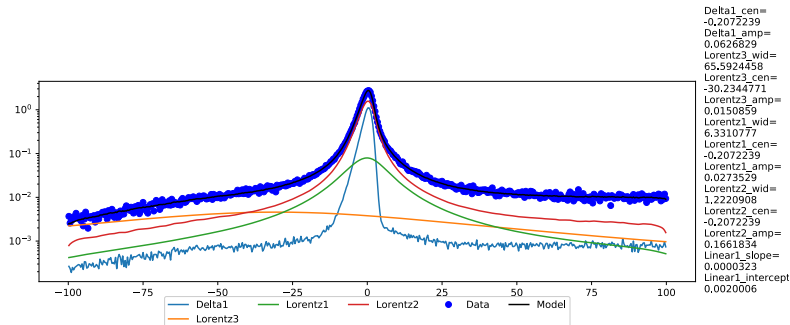
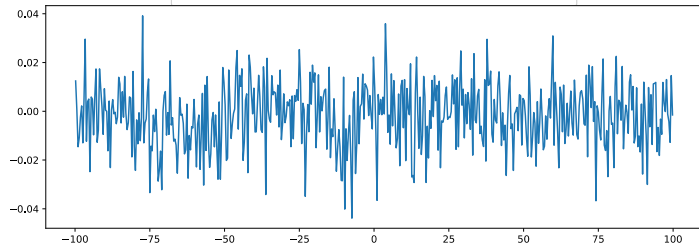
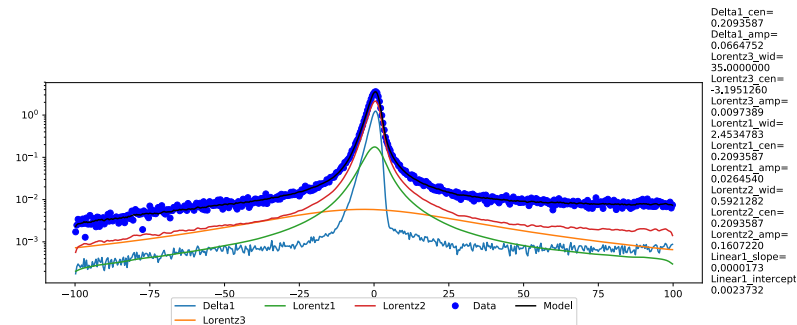
$$FWHM = f(\alpha, \beta, \gamma, \dots, \omega, \mathbf{Q})$$

Returning best fit values for $\alpha, \beta, \gamma \dots$ and so forth

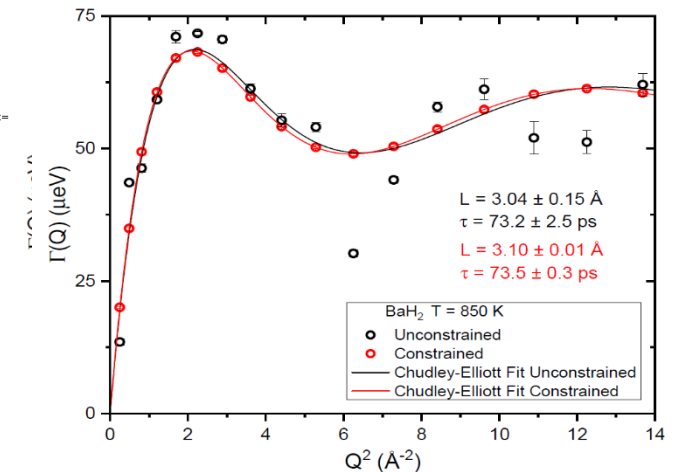
Chudley-Elliot $HWHM(Q) = \frac{\hbar}{\tau} \cdot \left(1 - \frac{\sin(Ql)}{Ql} \right)$

Courtesy: Timmy Ramirez-Cuesta

QCLIMAX example

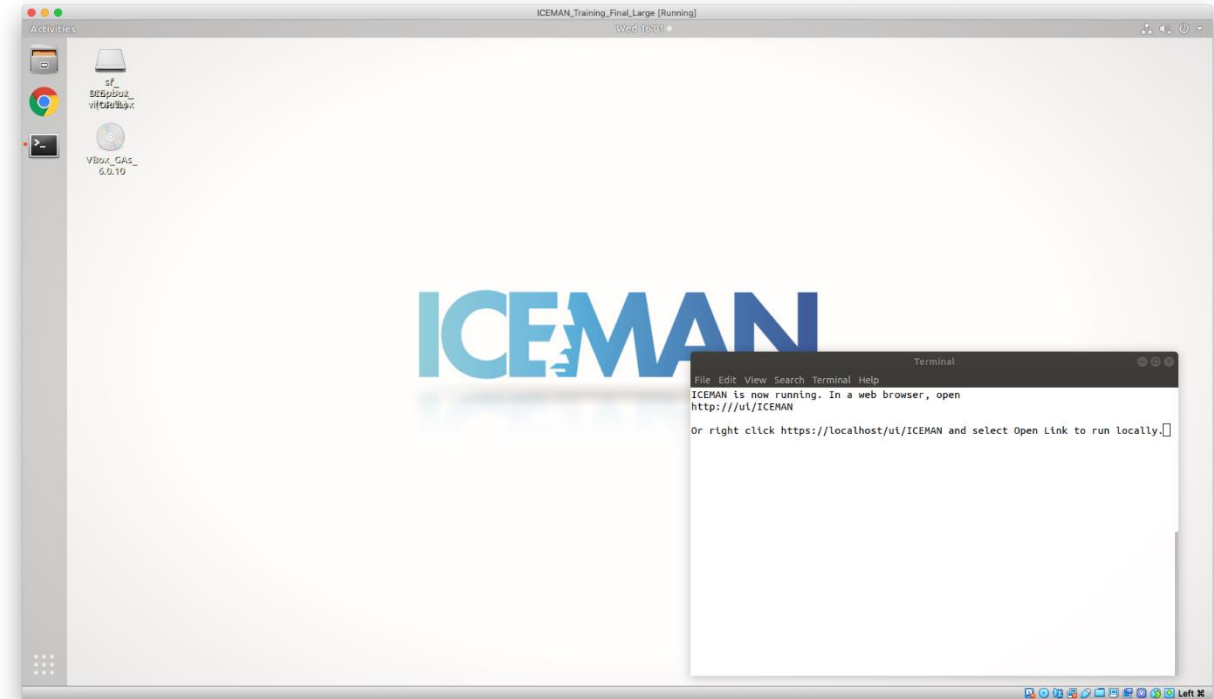
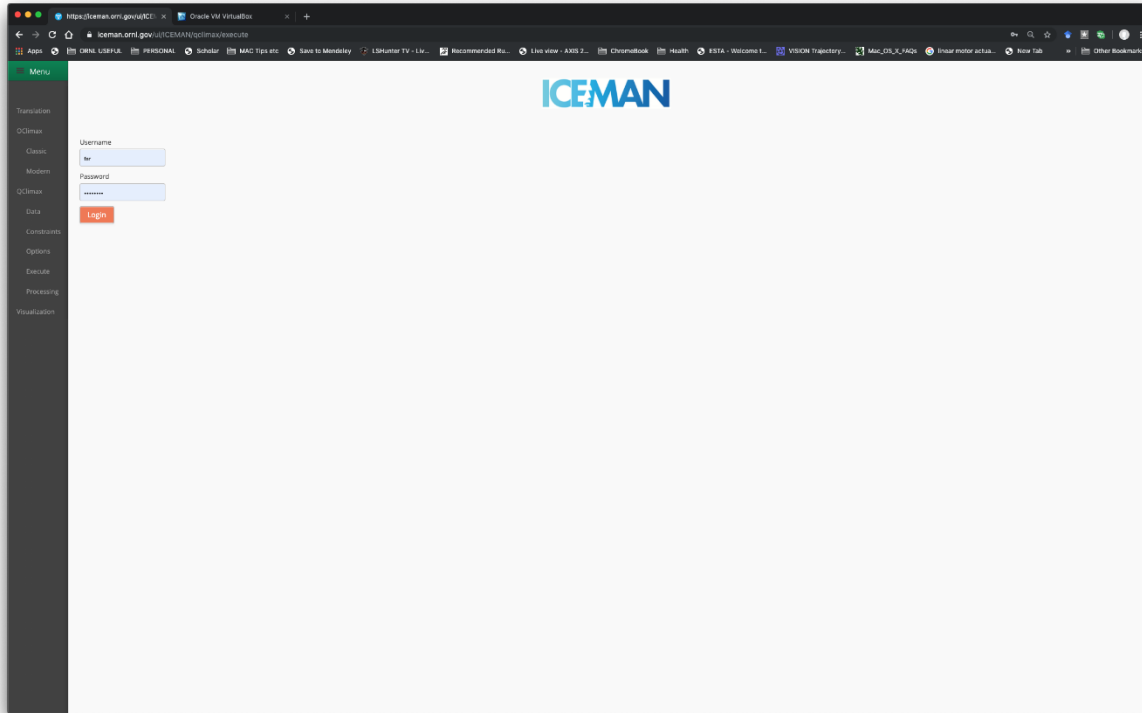
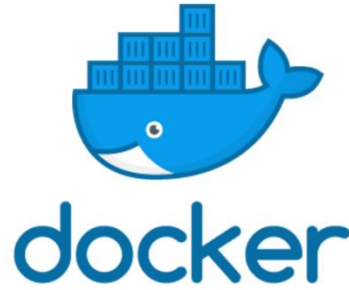


Example:
Chudley-Elliott fit to hydrogen in metal hydrides



How to use QCLIMAX: many flavors

- Docker container
- Virtual machine
- Web access (hosted at ORNL, CADES)



Simple interface

The screenshot shows the ICEMAN web interface. The main content area is titled 'BASIS' and contains two sections: 'Resolution File' and 'Data Files'. The 'Resolution File' section shows a file named 'BaH2_300K_111.dat' with buttons for 'Select File', 'Select Function', and 'Graph'. Below it, there is a checkbox section for 'Select which Q values to use in the fitting' with options Q0(0.3), Q1(0.5), Q2(0.7), and Q3(1.0). The 'Data Files' section has a 'Select Data' button, a 'Specify temperatures' checkbox, and a 'Graph' button. Below this is a table of data files:

File
BaH2_710K_111.dat
BaH2_730K_111.dat
BaH2_750K_111.dat
BaH2_690K_111.dat

The 'Model Functions' panel on the right contains three model configurations, each with a 'Graph' button and a close button:

- Linear1** (Linear):

Name	Starting Value	Min	Max
intercept	1.0E-8		
slope	1.0E-8		
- Delta1** (Delta):

Name	Starting Value	Min	Max
amp	1.0	0.0	
cen	0.0	-3900.0	3900.0
- Lorentz1** (Lorentz):

Name	Starting Value	Min	Max
amp	1.0	0.0	
cen	1.0E-4	-100.0	100.0
wid	0.5	0.025	
- Lorentz2** (Lorentz):

Name	Starting Value	Min	Max
amp	1.0	0.0	
cen	1.0E-4	-100.0	100.0
wid	0.5	0.025	
- Lorentz3** (Lorentz):

Name	Starting Value	Min	Max
amp	1.0	0.0	
cen	1.0E-4	-100.0	100.0
wid	75.0	5.0	140.0

Use your constraints

The screenshot shows the ICEMAN web interface in a browser window. The URL is <https://iceman.ornl.gov/ui/ICEMAN/qclimax/constraints>. The page title is "ICEMAN" and the user is logged in as "Ramirez Cuesta, Anibal".

The interface displays a list of constraints for a simulation run named "Run1". The constraints are organized into a table with columns for the constraint name, a formula, and various control options.

Constraint Name	Formula	Set Constraint	Relax constraint	Rattle value	Relaxation %	Deactivated
Linear1_intercept		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Linear1_slope		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Delta1_amp		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Delta1_cen	P[Lorentz2_cen]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz1_amp		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz1_cen	P[Lorentz2_cen]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz1_wid	ChudleyElliott(3.6,Lorentz1widtau,Q_)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz2_amp	P[Lorentz1_amp]*C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz2_cen		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz2_wid	ChudleyElliott(4.2,Lorentz2widtau,Q_)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz3_amp		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz3_cen		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
Lorentz3_wid		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>

Constraints not there? No problem, write them in Python!

The screenshot shows the ICEMAN web interface. On the left is a sidebar menu with options like Translation, OClimax, Classic, Modern, QClimax, Data, Constraints, Options, Execute, Processing, and Visualization. The main area displays a table of parameters:

Name	Starting Value
Lorentz1widtau	350.0
Lorentz2widtau	450.0
C	0.5

Below the table is a text input field for writing an expression for the constraint, with the example: `ChudleyElliott(3.6,Lorentz1widtau,Q_VALUE)`. A central modal window shows a Python code editor with the following code:

```
1 # Write a Python function here, describing the constraint you want to use.
2 # The first line consists of the function name and its parameters. Replace Constraint_Name with the name you want for your
3 # Inside the parentheses are the arguments. The first argument will always be symtable. This is a Python data structure co
4 # the fitting as well as the other functions, whether built in or written by you
5 def ChudleyElliott(symtable, L,tau,Q_VALUE):
6     """Custom constraint. Equivalent to the command argument1 + argument2 * sin(Delta1_amp) """
7     # Write your custom constraint below
8     # You can use the symtable to access functions or parameters by writing symtable[name] as seen below
9     # In order to use this function in your constraint, you must use it as you would a normal function but leaving out the
10    # For example Constraint_Name(1,2)
11    # Units are:
12    # 10E-10 m^2/sec, then 10E-10 m^2/sec * 1/Angstrom =
13    # 10E-10 m^2/sec * 1/1E-20 m^2 = 10E10 1/sec
14    # Reduced Plank's constant = 6.58211951E-16 eV*sec
15    # therefore
16    # 10E10 1/sec * 6.58211951E-16 eV*sec = 6.58211951E-06 eV = 6.58211951 micro eV
17    # in this case D = 10E-10 m^2/sec
18    # Units of tau:
19    # tau * D * Q^2 = 1
20    # tau = 1/(D * Q^2) = 1/(1E-10 m^2/sec * 1/(1E-20 m^2)) = 1E-10
21    # mutiplying tau *1E2 gives tau in 1E-12 sec = 1 ps
22
23    x=Q_VALUE
```

Buttons for 'Set Constraint', 'Relax constraint', 'Rattle value', 'Relaxation %', and 'Deactivated' are visible for each parameter row. The modal window also has 'OK', 'Cancel', and 'Download' buttons.

Visualize, edit results, restart, download etc.

The screenshot shows a web browser window displaying the ICEMAN interface. The browser's address bar shows the URL `https://iceman.ornl.gov/ui/ICEMAN/visualization`. The page features the ICEMAN logo at the top center and a user profile dropdown for 'Ramirez Cuesta, Anibal' at the top right. On the left, a dark sidebar contains a 'Menu' section with options: Translation, OClimax, Classic, Modern, QClimax, Data, Constraints, Options, Execute, Processing, and Visualization. The main content area displays a file list under the heading 'qclimax'. The list includes various files such as `BaH2_300K_111_1L_CE-2019-10-16_09:01:36`, `BaH2_690K_111_0_csv`, `BaH2_690K_111_0_Params_by_Q.csv`, `BaH2_690K_111Parameters.pdf`, `BaH2_710K_111_0_csv`, `BaH2_710K_111_0_Params_by_Q.csv`, `BaH2_710K_111Parameters.pdf`, `BaH2_730K_111_0_csv`, `BaH2_730K_111_0_Params_by_Q.csv`, `BaH2_730K_111Parameters.pdf`, `BaH2_750K_111_0_csv`, `BaH2_750K_111_0_Params_by_Q.csv`, `BaH2_750K_111Parameters.pdf`, and `BaH2_750K_111_docker-qclimax.ini_0_pre-error-calculation_report.txt..`. A 'Download' button is located at the bottom left of the file list.

Recent Development

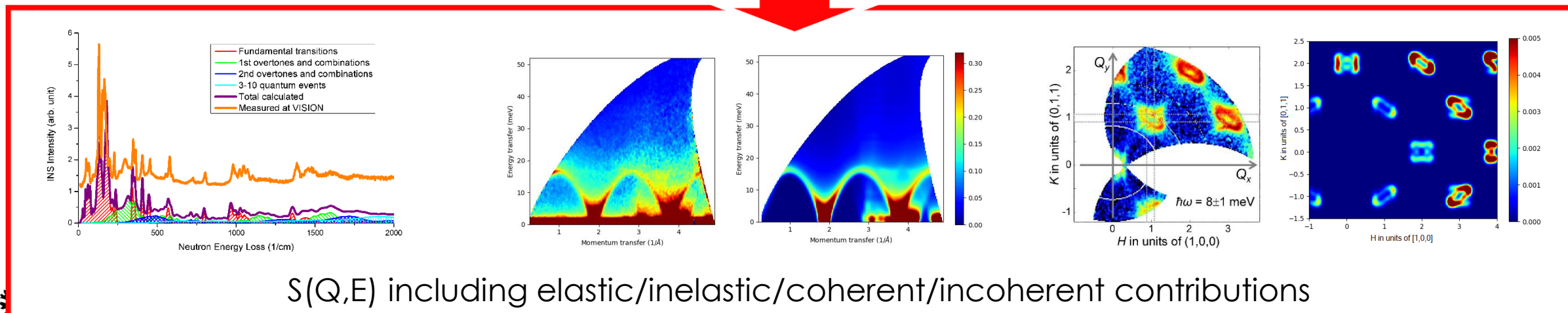
- DrSPINE (Piotr Zolnierczuk)
- QCLIMAX (Timmy Ramirez-Cuesta)
- OCLIMAX (YQ Cheng)



OCLIMAX bridges theory and INS experiments



Normal modes or phonons → **OCLIMAX** ← Molecular dynamics trajectories



S(Q,E) including elastic/inelastic/coherent/incoherent contributions

Development of OCLIMAX

- Started 2016
- First version released 2017
- Paper published 2019
- Used to analyze data from VISION and multiple other neutron spectrometers

Features:

- ❖ Full scattering (coherent, incoherent, elastic, inelastic)
- ❖ Powders and single crystals
- ❖ Temperature effects
- ❖ Multiphonon excitations
- ❖ Arbitrary instrument geometry and resolution
- ❖ Arbitrary cuts in 4-dimensional Q-E space
- ❖ Interface with atomistic modeling tools (e.g. DFT codes)
- ❖ Interface with INS data analysis tools (e.g. DAVE and Mantid)
- ❖ User-friendly (multiple platform, easy to use, fast on PCs)
- ❖ Parameter-free thermal neutron scattering cross-sections
- ❖ Molecular dynamics trajectories to INS spectra

JCTC

Journal of Chemical Theory and Computation

Article

Cite This: *J. Chem. Theory Comput.* 2019, 15, 1974–1982

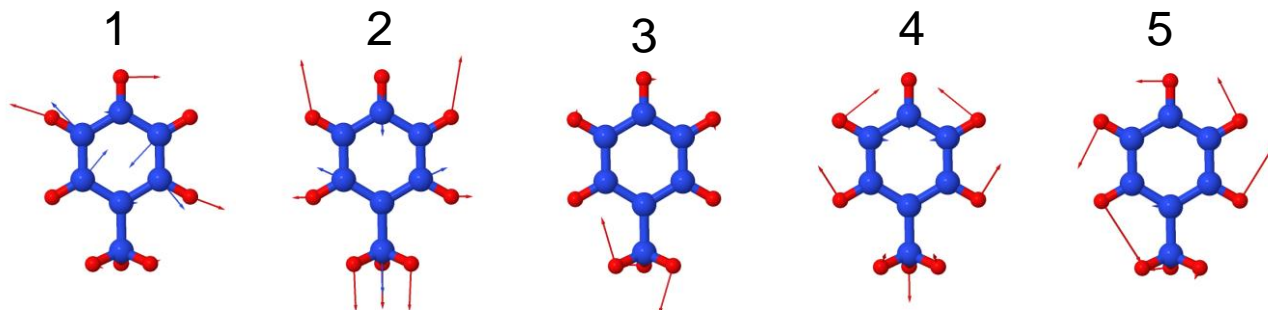
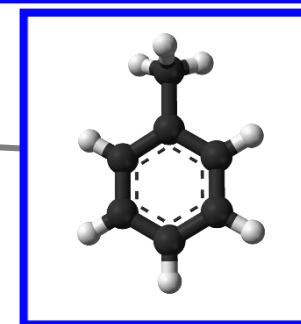
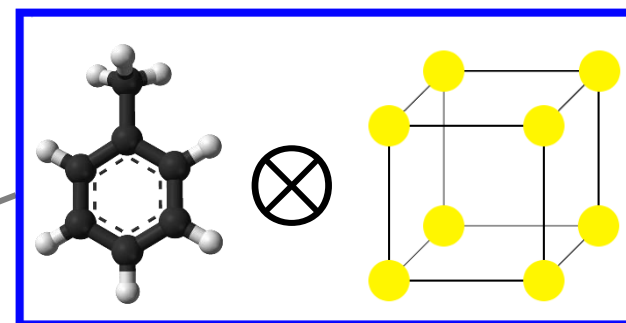
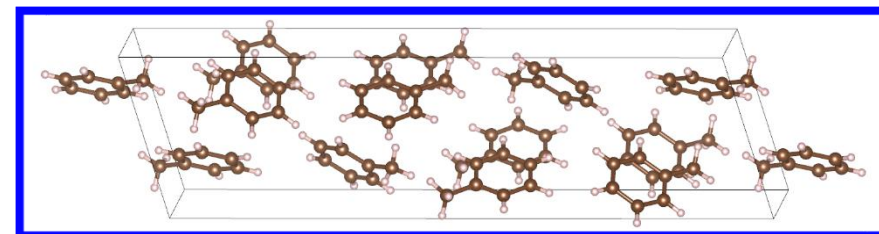
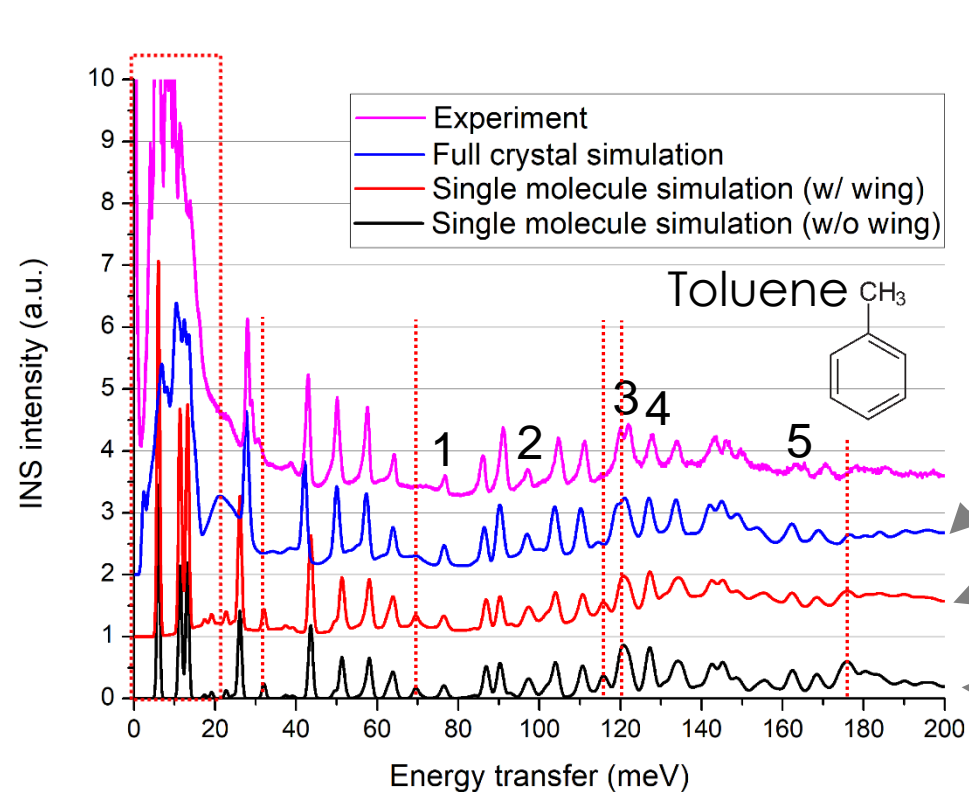
pubs.acs.org/JCTC

Simulation of Inelastic Neutron Scattering Spectra Using OCLIMAX

Y. Q. Cheng,^{*✉} L. L. Daemen, A. I. Kolesnikov,[✉] and A. J. Ramirez-Cuesta^{*}

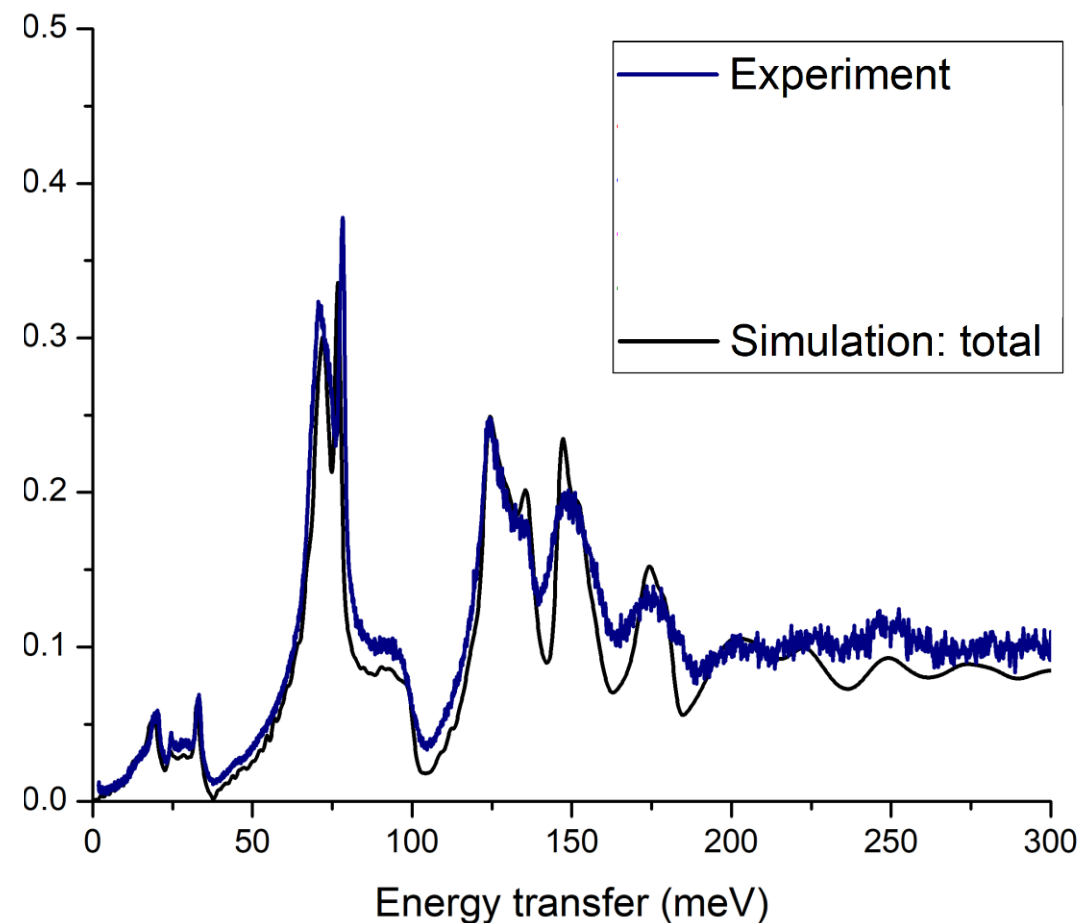
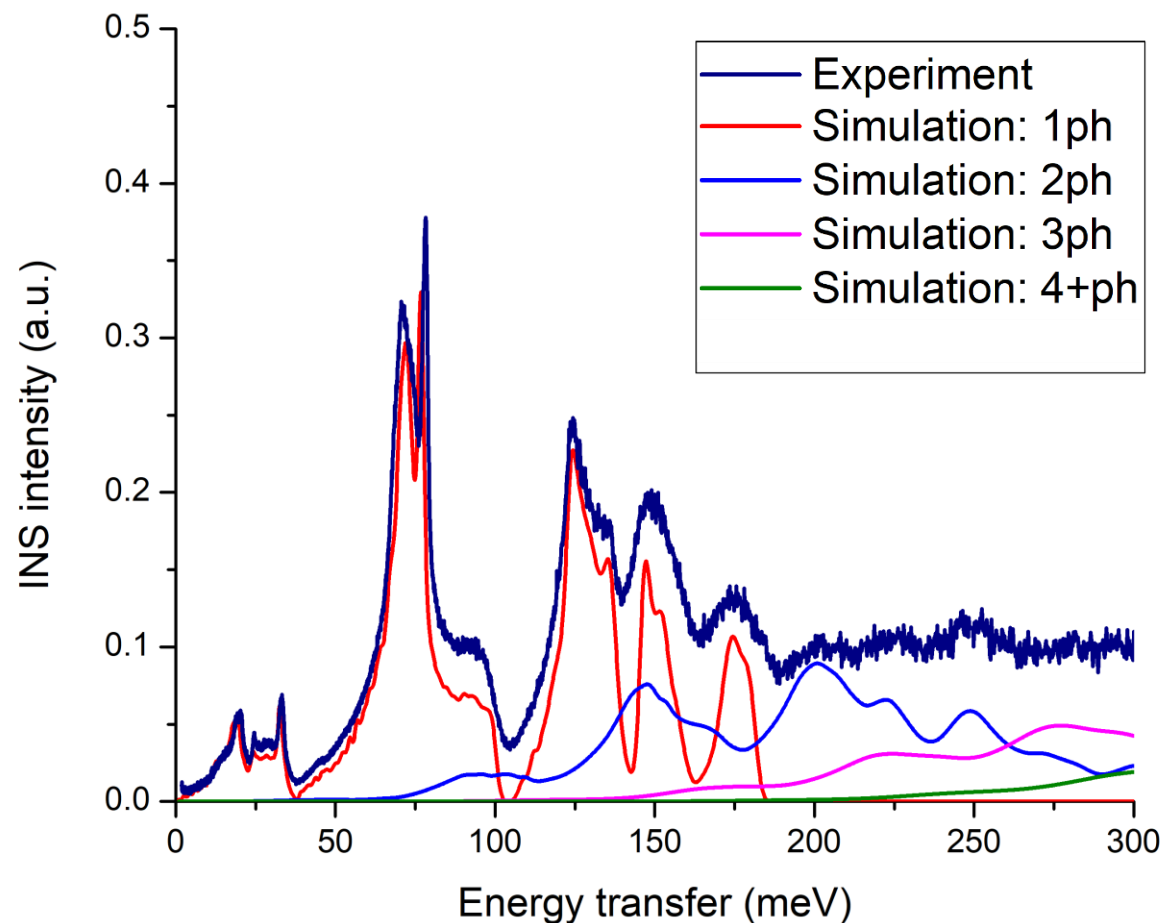
Neutron Scattering Division, Oak Ridge National Laboratory, Oak Ridge Tennessee 37831, United States

OCLIMAX example: From single molecule to solid



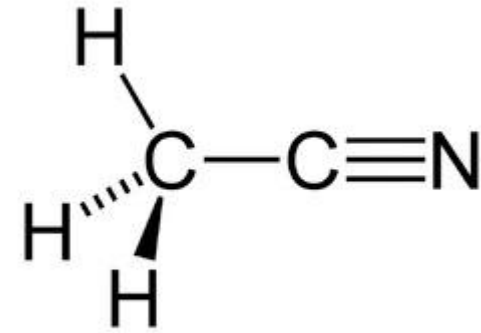
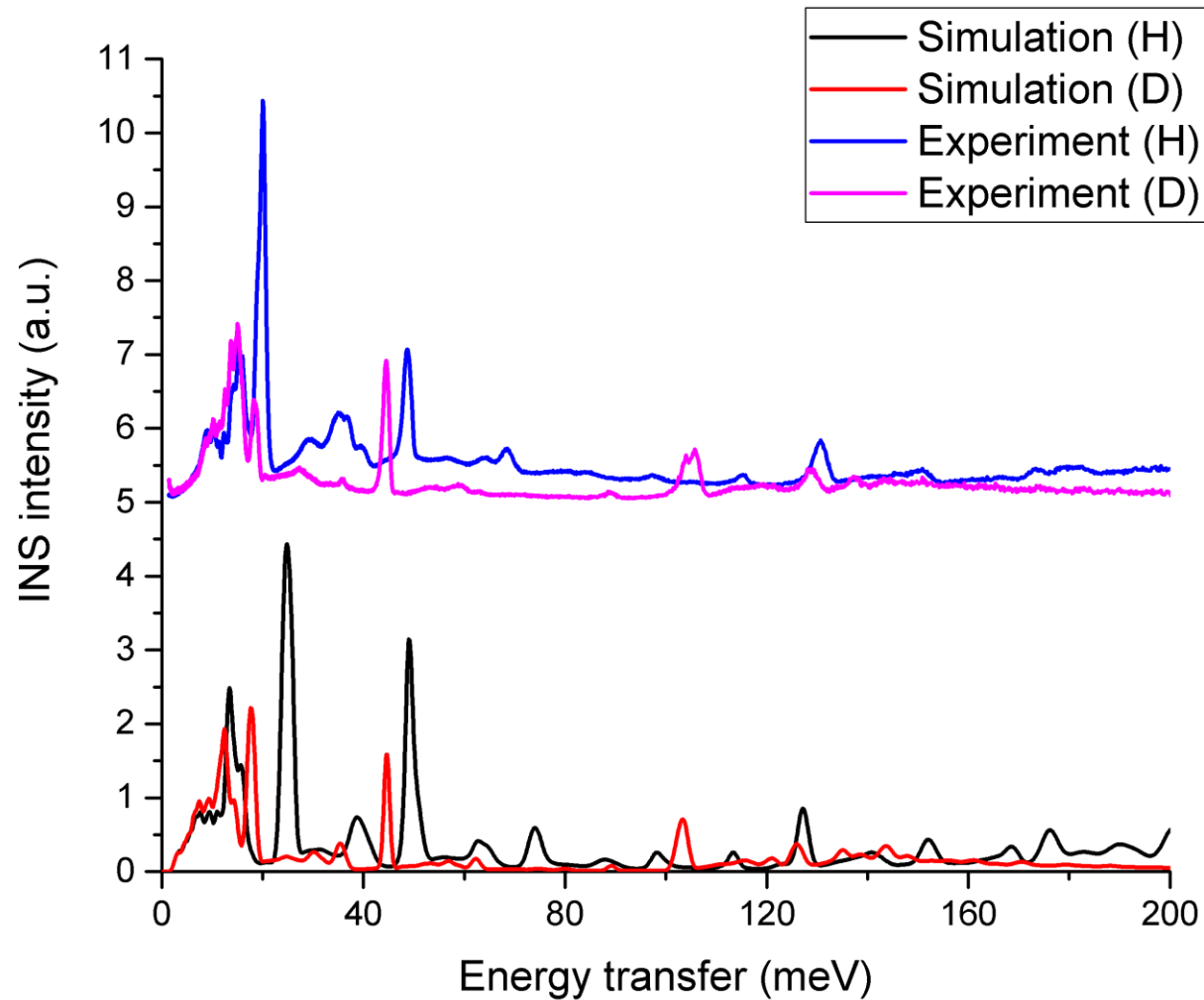
✓ Understanding intermolecular interactions (van der Waals forces, hydrogen bonding, charge transfer)

OCLIMAX example: Multiphonon excitations

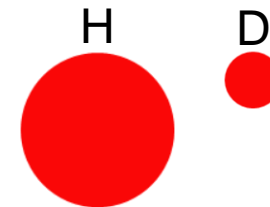


- ✓ Solving phonon density of states
- ✓ Understanding anharmonicity and potential energy landscape

OCLIMAX example: Isotope substitution



$$\omega = \sqrt{\frac{k}{m}}$$



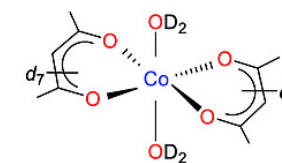
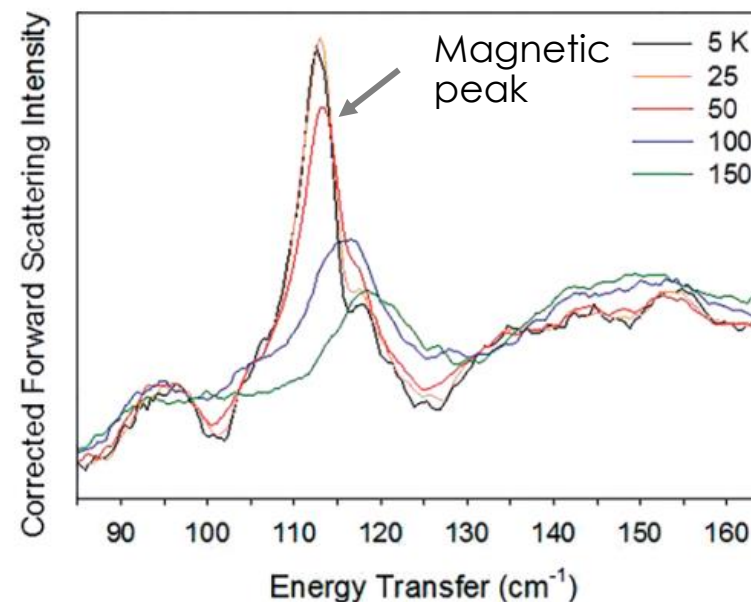
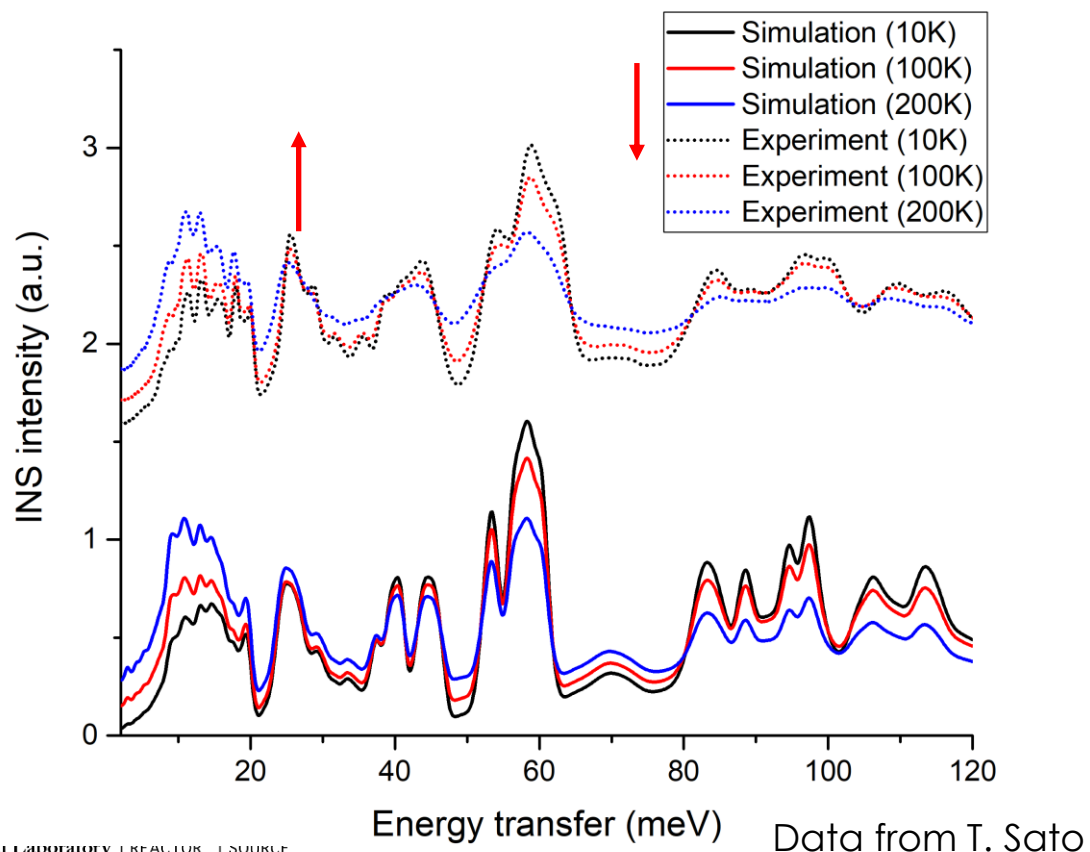
- ✓ Virtual experiment for doping effects and isotope labeling
- ✓ Breaking down the total intensity into partial contributions from individual species or atoms

OCLIMAX example: Temperature effects

- Phonon population
- Debye-Waller factor

$$W_d = \frac{\hbar}{4m_d N_q} \sum_s \frac{(\mathbf{Q} \cdot \mathbf{e}_{ds})^2}{\omega_s} (2n_s + 1)$$

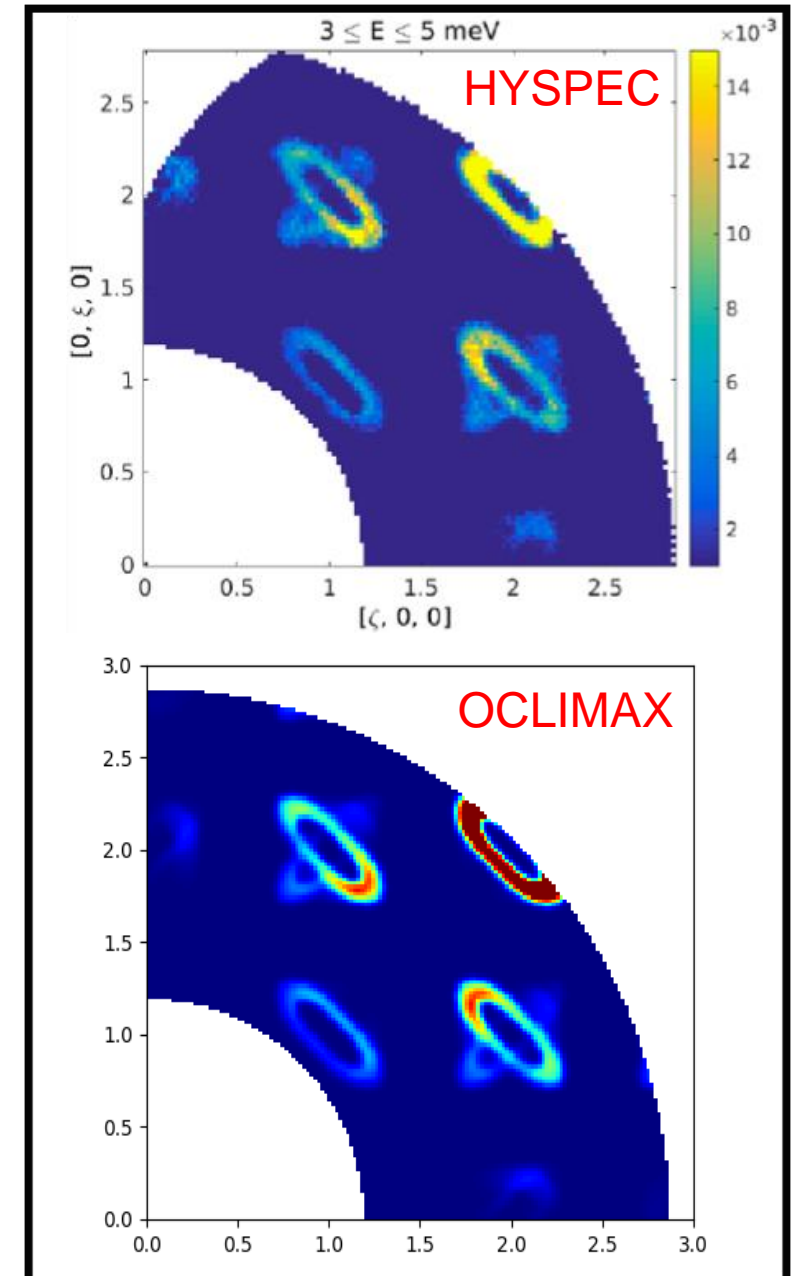
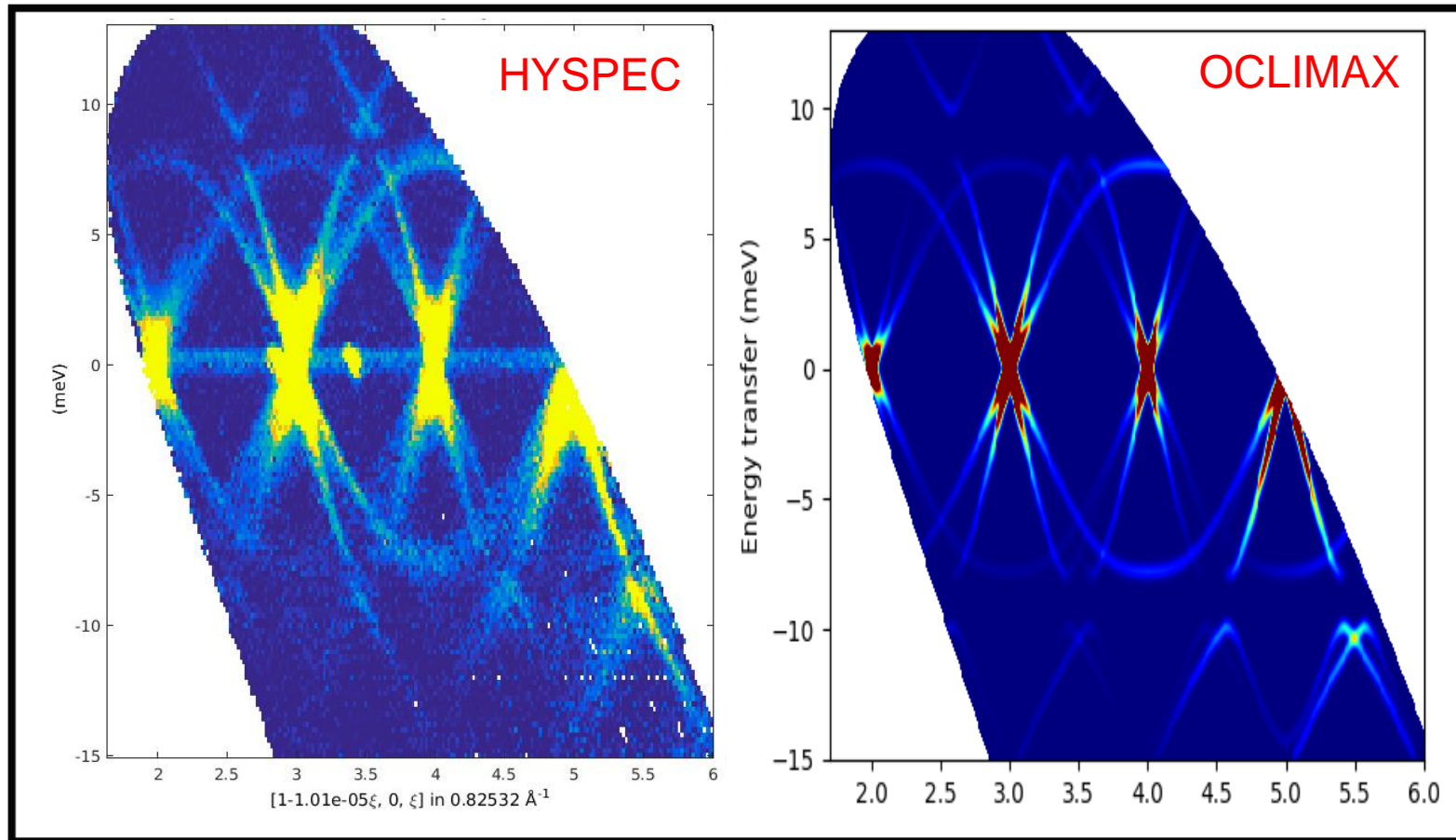
$$S_{inc\pm 1}(\mathbf{Q}, \omega) = \sum_d \frac{1}{2m_d} \left\{ \bar{b}_d^2 - (\bar{b}_d)^2 \right\} \exp(-2W_d) \sum_s \frac{|\mathbf{Q} \cdot \mathbf{e}_{ds}|^2}{\omega_s} \left\langle n_s + \frac{1}{2} \pm \frac{1}{2} \right\rangle \delta(\omega \mp \omega_s)$$



Stavretis S.E., Cheng Y.Q., Daemen L.L., Brown C.M., Moseley D.H., Bill E., Atanasov M., Ramirez-Cuesta A.J., Neese F., Xue Z., European Journal of Inorganic Chemistry, 2019, 8, 1119-1127 (2019).

- ✓ Separating different excitations based on temperature dependence
- ✓ Benchmarking atomic displacement parameters

OCLIMAX example: Single crystal cuts in 4-D Q-E space

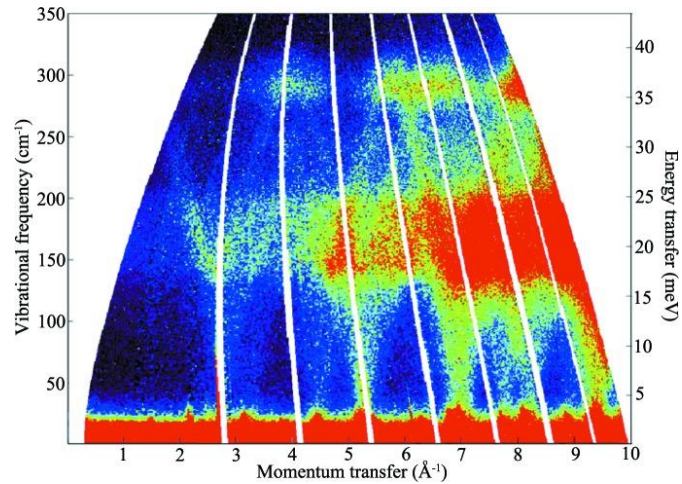


- ✓ Validating phonon frequencies, polarization vectors, and force constants
- ✓ Understanding phonon anomalies.

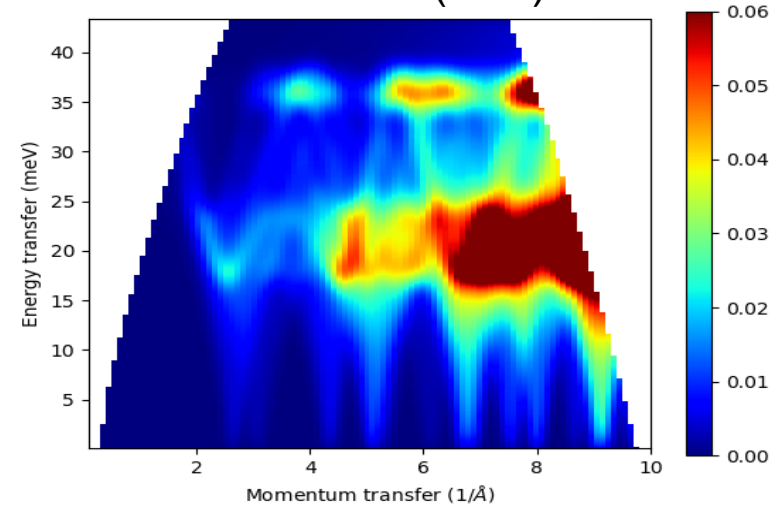
Unpublished INS data from R. Hermann. DFT simulations by L. Lindsay and T. Pandey @ORNL

OCLIMAX example: Coherent powder INS (aluminum)

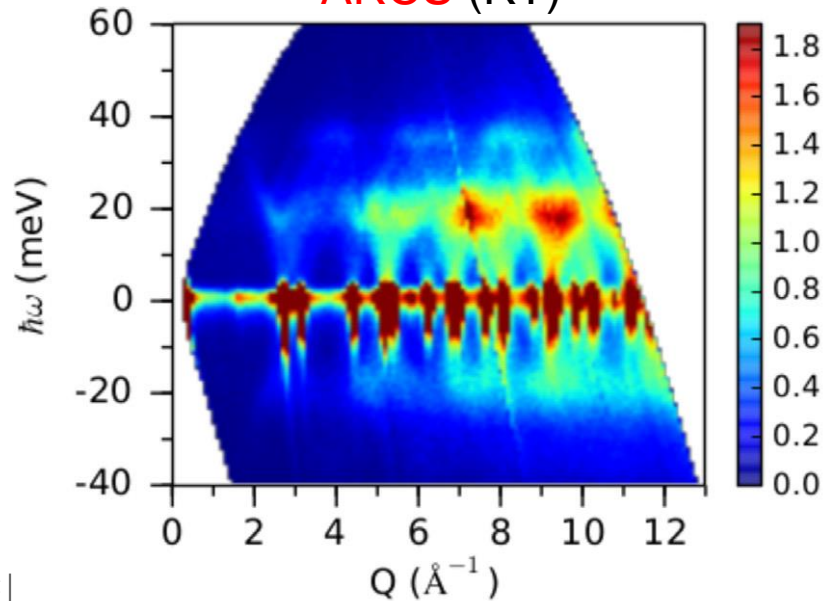
MARI (10K)



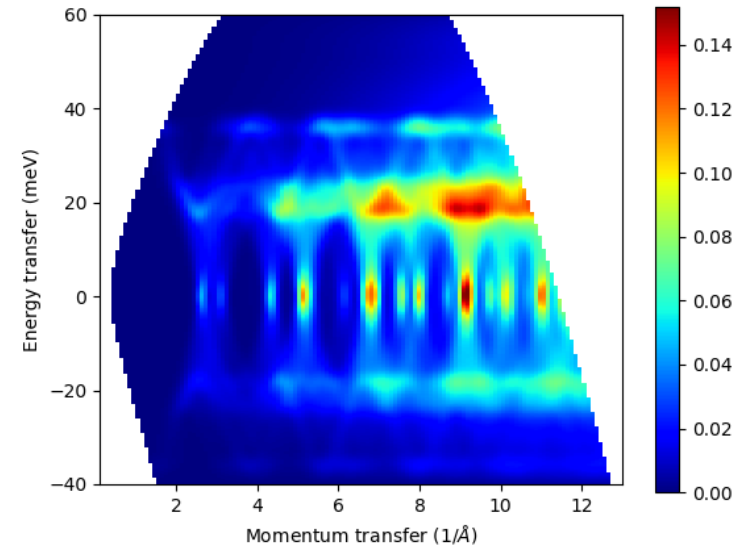
OCLIMAX (10K)



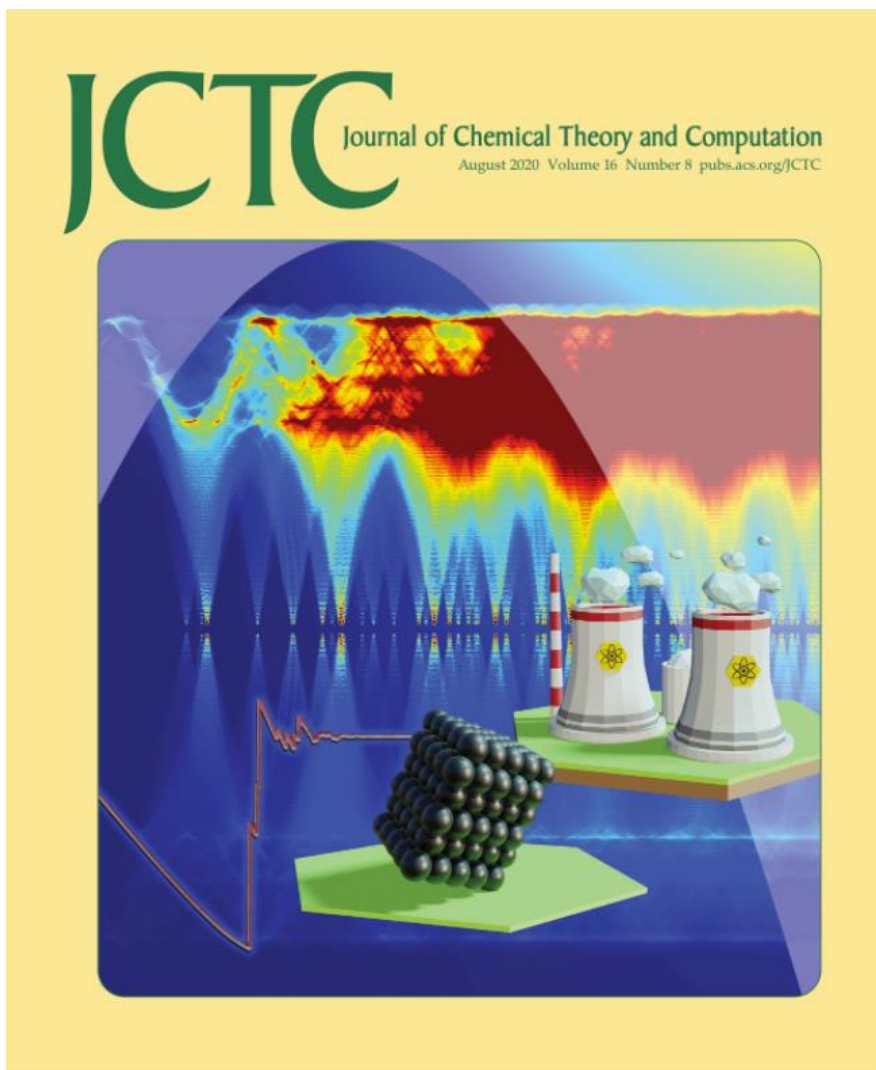
ARCS (RT)



OCLIMAX (RT)



Ab initio thermal neutron scattering cross-section of solids



JCTC

Journal of Chemical Theory and Computation

August 2020 Volume 16 Number 8 pubs.acs.org/JCTC

JCTC
Journal of Chemical Theory and Computation

pubs.acs.org/JCTC

Article

Calculation of the Thermal Neutron Scattering Cross-Section of Solids Using OCLIMAX

Y. Q. Cheng* and A. J. Ramirez-Cuesta



Cite This: *J. Chem. Theory Comput.* 2020, 16, 5212–5217



Read Online

ACCESS |

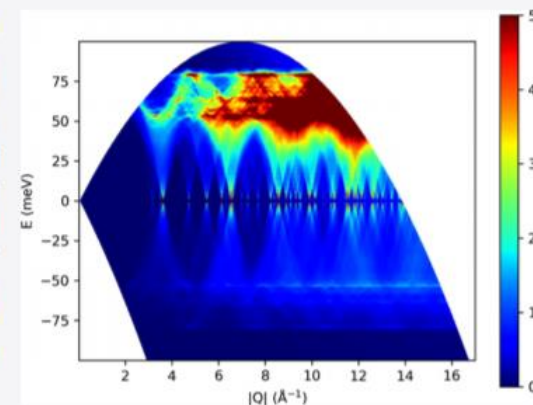


Metrics & More



Article Recommendations

ABSTRACT: The thermal neutron scattering cross-section of a solid depends on the energy (or wavelength) of the incident neutrons. Devising a method to calculate the energy dependence from first principles, without the approximations built in the scattering theory, has been a major undertaking in nuclear engineering. Here, we demonstrate such a calculation method using the program OCLIMAX. Our approach eliminates various approximations and limitations involved in a regular calculation with the LEAPR module of NJOY code, and the results are compared with available experimental and theoretical data. It is also demonstrated how additional insight can be obtained from the calculated full dynamical structure factor. The results reported here show the great potential and excellent platform provided by OCLIMAX for future development in the study of neutron thermalization in solid materials for different applications.

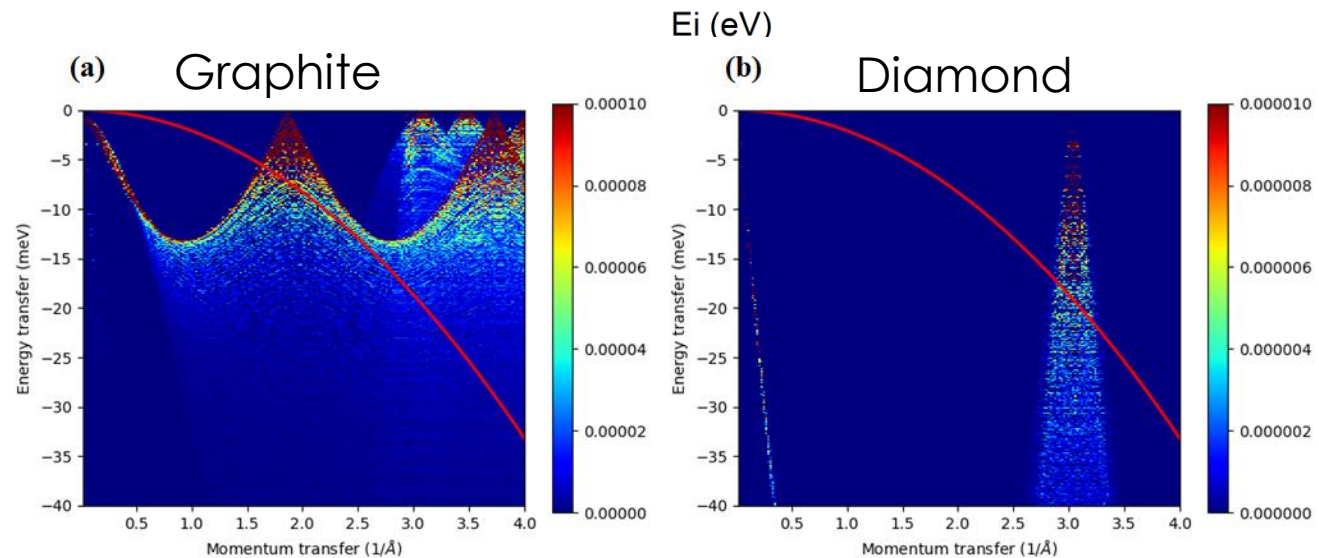
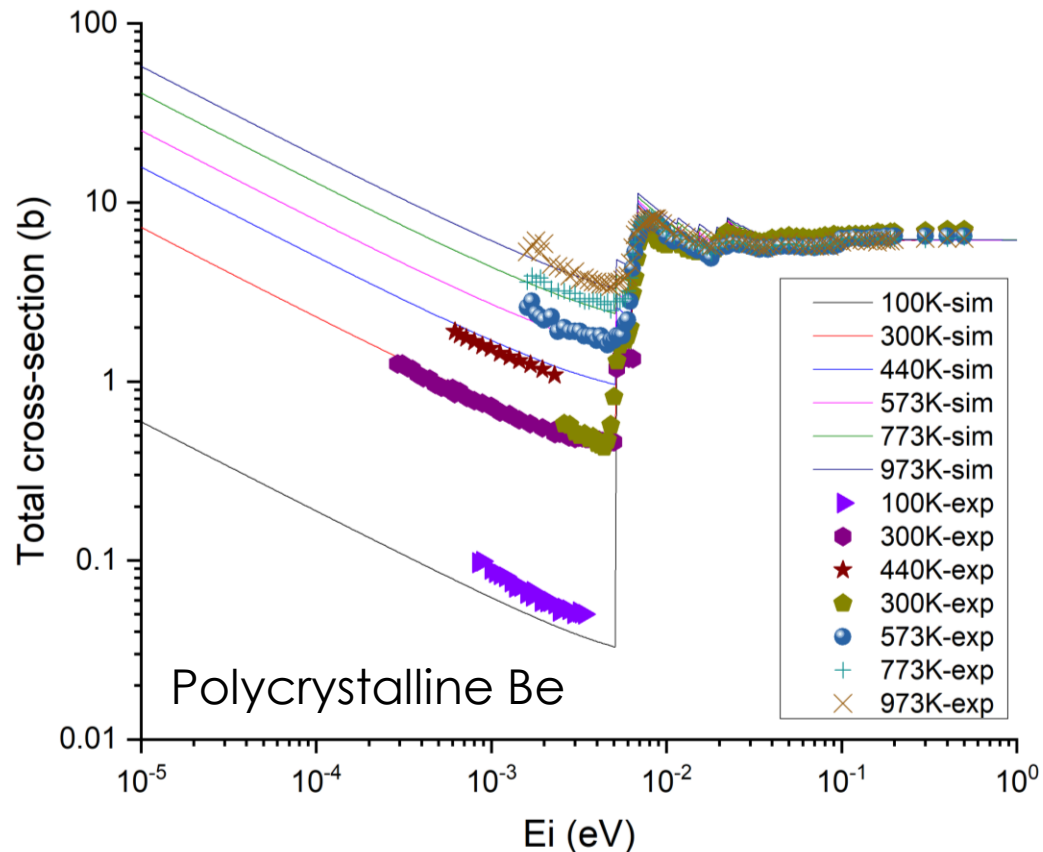
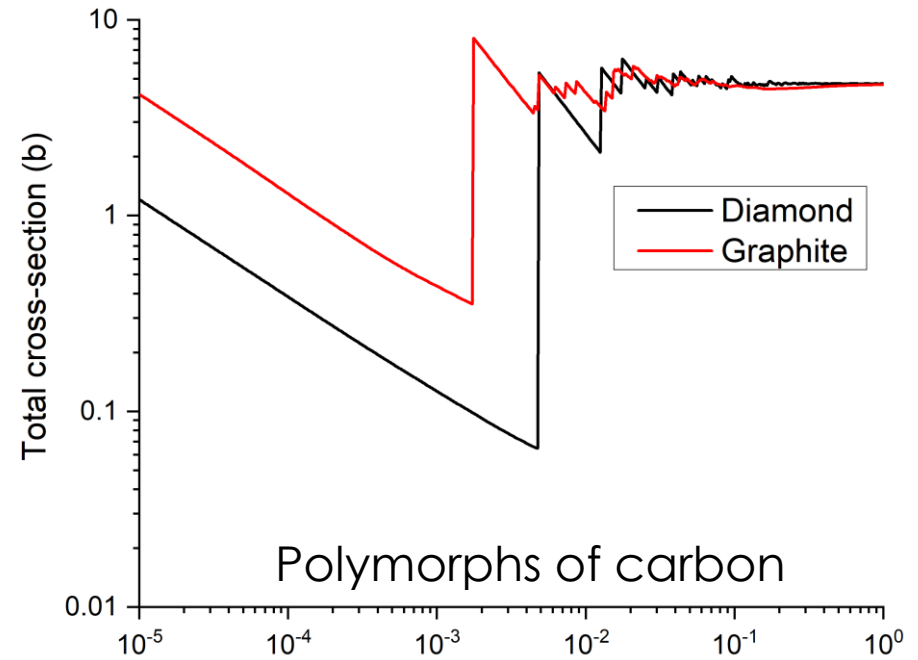


ACS Publications
Most Trusted. Most Cited. Most Read.

www.acs.org

Ab initio thermal neutron scattering cross-section of solids

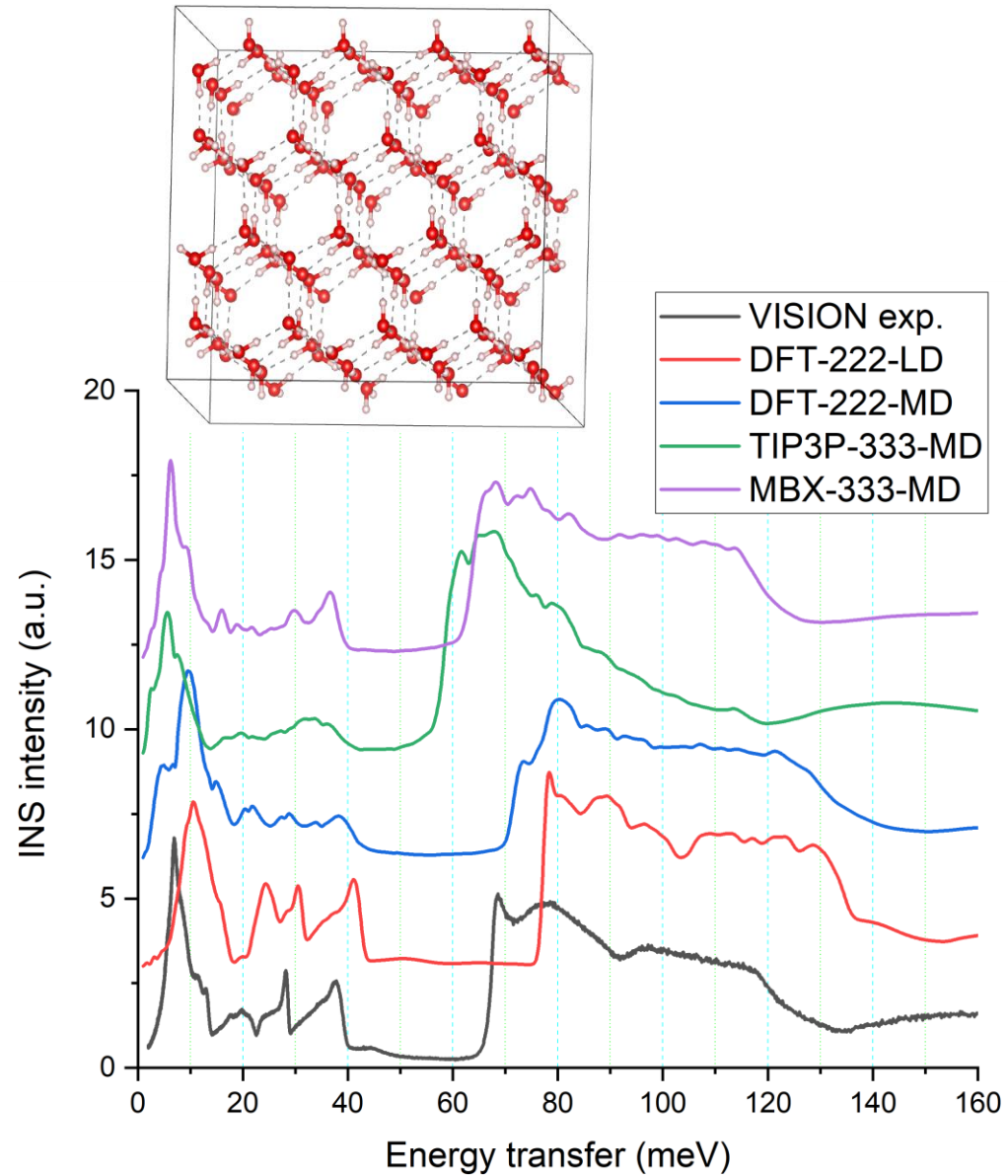
- ✓ DFT → XS, no intermediate model/parameter
- ✓ Incoherent approximation lifted
- ✓ Computational screening and nuclear material database
- ✓ Insight on dynamical mechanism



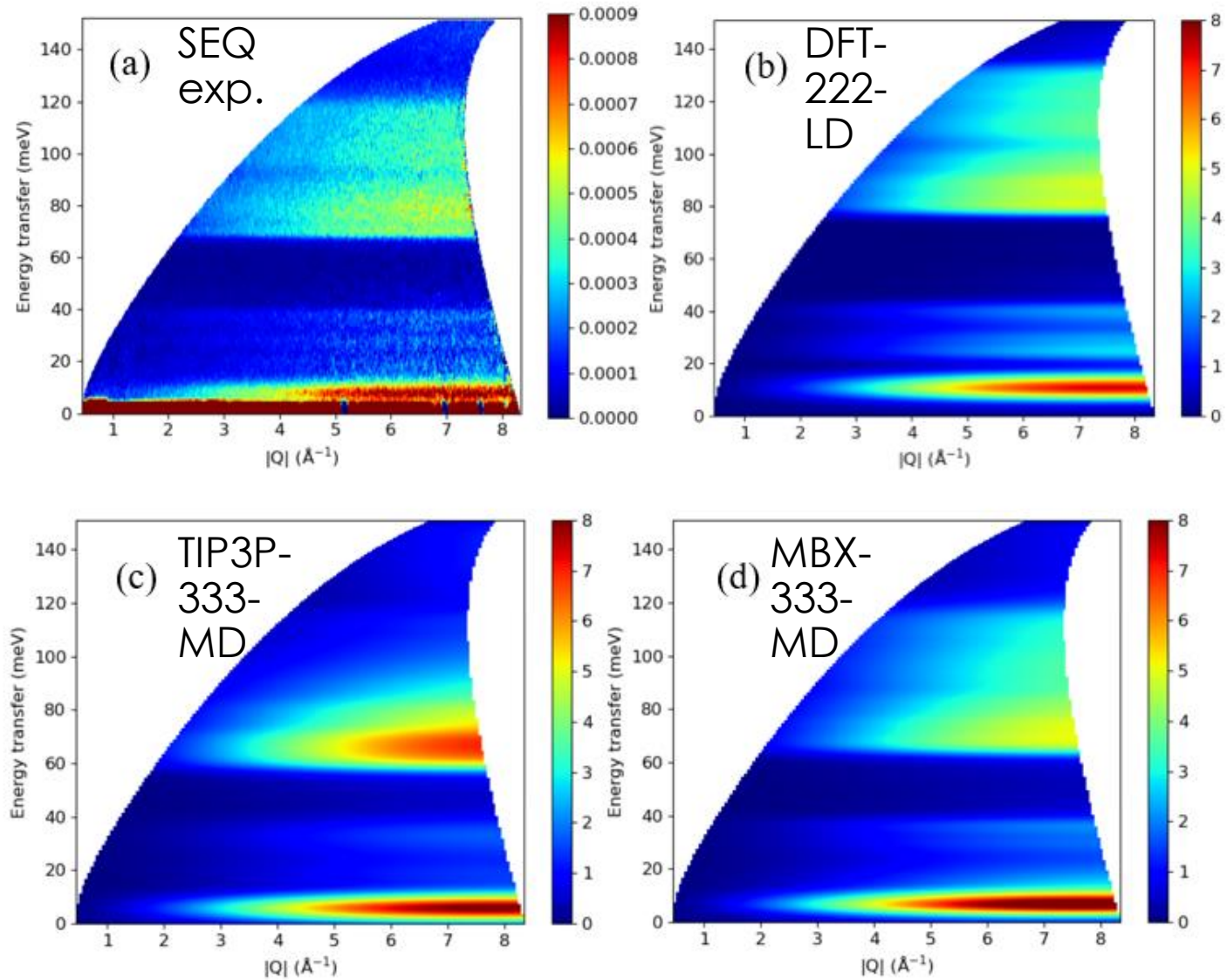
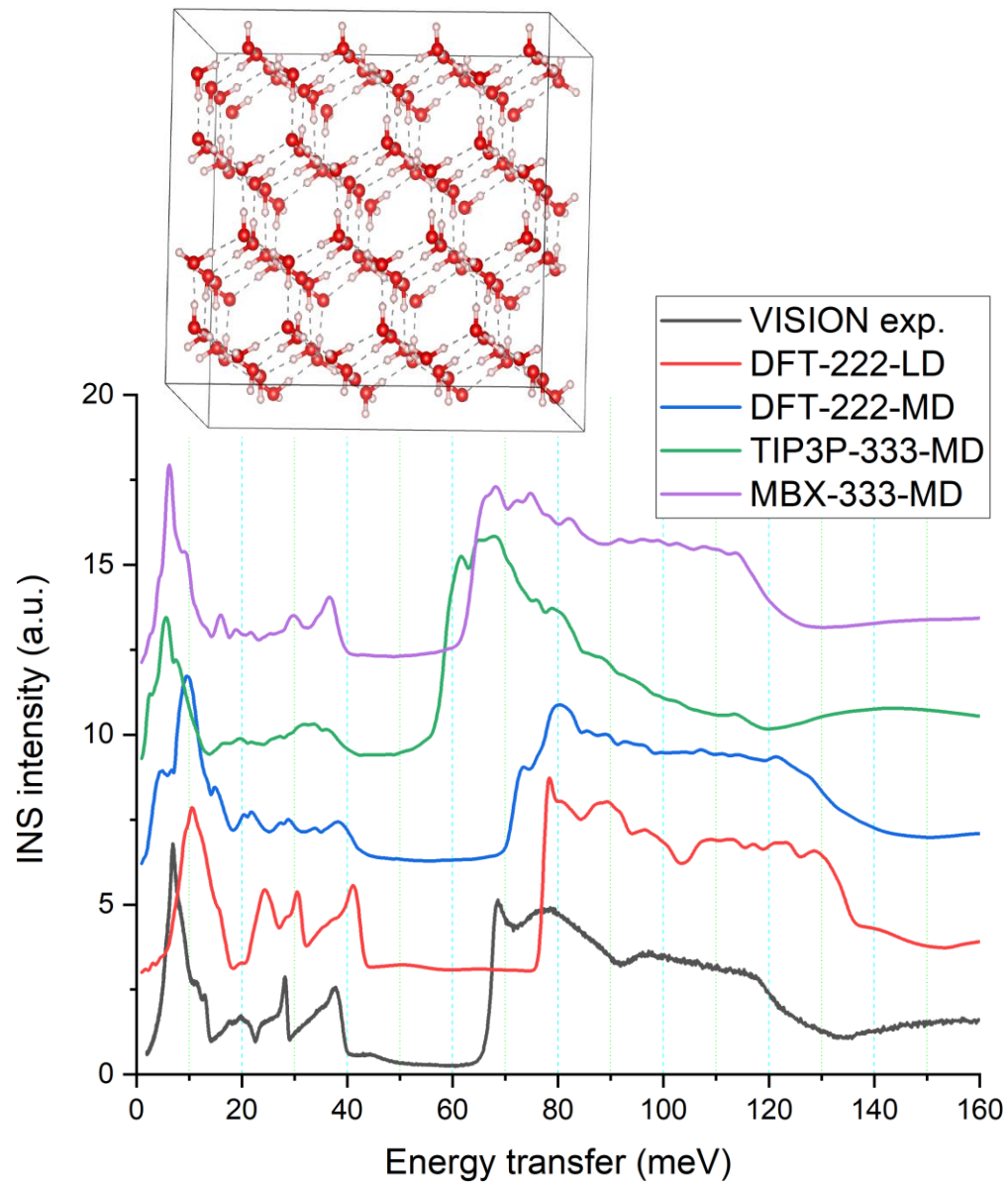
MD trajectories to INS: Motivations

- MD simulations are more routinely and efficiently done than LD simulations, especially for large systems (defective, disordered, amorphous, polymeric, biological, or simply complex).
- An incomplete LD simulation is useless, but an “incomplete” MD simulation can still provide useful information.
- MD simulation requires much less accuracy (in terms of convergence criteria for energy and force evaluation) to achieve comparable quality of INS spectra.
- MD simulation naturally includes temperature-dependent anharmonicity

MD trajectories to INS: Example (ice Ih)



MD trajectories to INS: Example (ice Ih)



Summary and future work



Summary and future work

- Data acquisition and reduction
 - NSE: Recently developed DrSPINE addressed previous issues
 - NSE: Upcoming transition to EPICS
- Data analysis
 - BASIS: QCLIMAX is a major new development with many desired features such as global fitting and constraints.
- Data interpretation (simulation)
 - NSE/BASIS: MD simulation is the most relevant modeling technique; significant development is needed in order to better integrate MD simulations with NSE/BASIS experiments.
 - VISION: OCLIMAX has evolved to become a powerful tool for general vibrational INS simulations.

Questions?

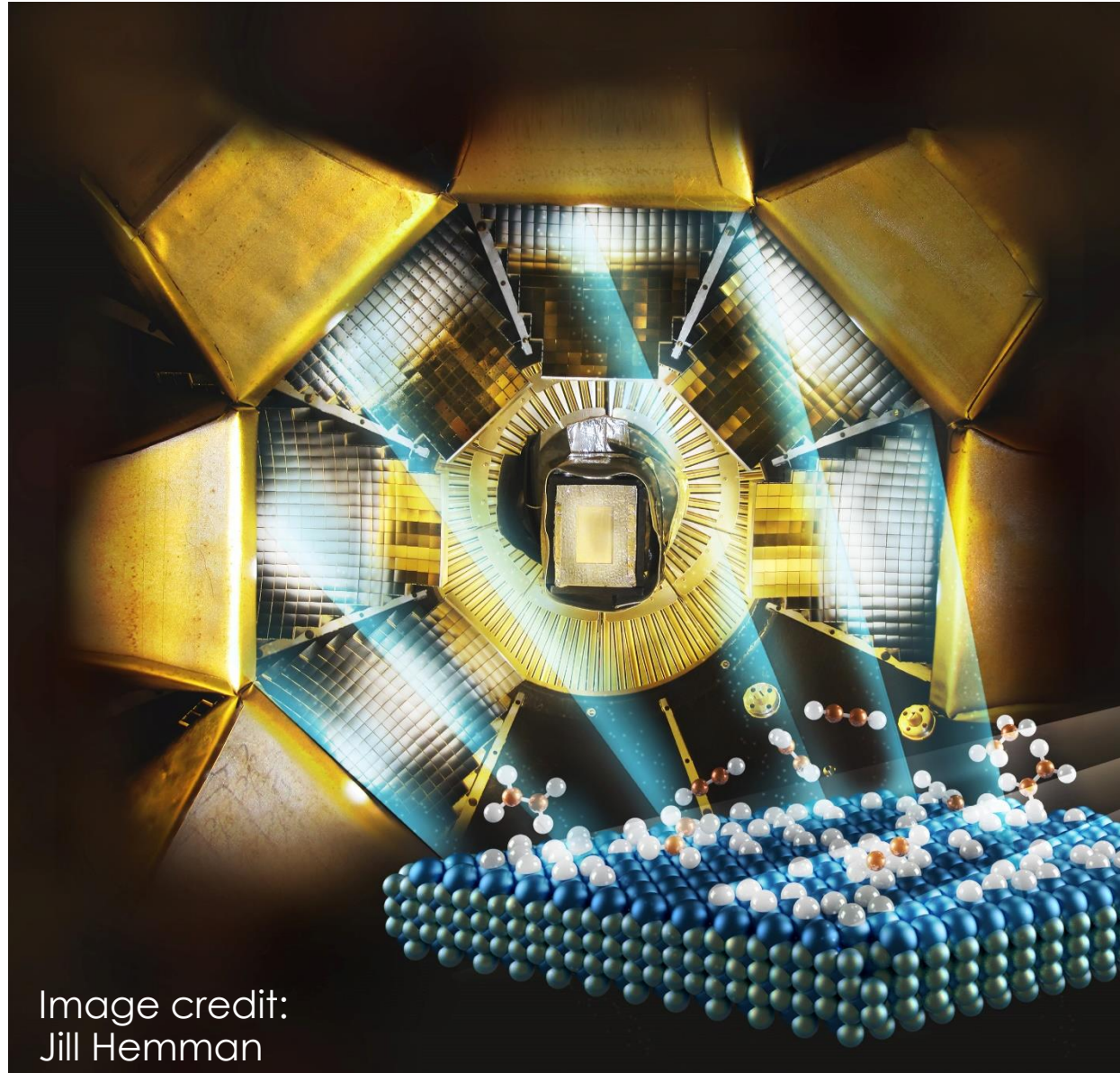


Image credit:
Jill Hemman

How to obtain OCLIMAX

- Install Docker (<https://www.docker.com/>)
- Install OCLIMAX

- For Linux/Mac (or Virtual Box on Windows)

Open a terminal, run:

```
$ curl -sL https://sites.google.com/site/ornliceman/getoclimax | bash  
$ oclimax pull
```

- For Windows (Native Windows 10)

Visit <https://sites.google.com/site/ornliceman/download>

Download oclimax.bat to your working directory

Open the cmd prompt or Powershell prompt, go to the working directory, run:

```
$ oclimax.bat pull
```

- For more information

Download the user manual at

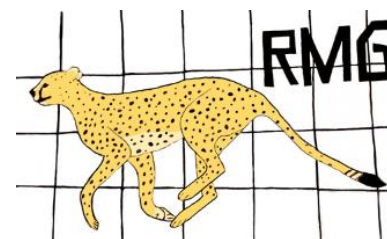
<https://sites.google.com/site/ornliceman/download>

Convert your files to OCLIMAX input file

- Automatically extract phonon frequencies and polarization vectors (or MD trajectories) from your DFT output files and generate the input file for OCLIMAX
- Currently support



NWCHEM
HIGH-PERFORMANCE COMPUTATIONAL
CHEMISTRY SOFTWARE



CASTEP, VASP, Phonopy, CP2K, Quantum Espresso, Crystal, Gaussian, ORCA, NWChem, DMol3, RMG, LAMMPS

e.g., **\$ oclimax convert -c yourfile.phonon**

How to run OCLIMAX

- Input file options for OCLIMAX:

*.oclimax (full crystal information) *.aclimax (xyz, can be molecules) *.tclimax (MD trajectories)
*.params (parameters that define the calculation)

- By default, OCLIMAX calculates VISION spectra with standard parameters. To do this, run:

```
$ oclimax run yourfile.oclimax
```

- The output files of this run include:

yourfile*.csv: The simulated INS spectra or other results
yourfile.params: The (default) parameters used for this calculation

- To run the simulation with different parameters, you may edit the parameter file, and then run

```
$ oclimax run yourfile.oclimax yourfile.params
```

Parameters for OCLIMAX calculation

```
## General parameters
TASK = 0 # 0:inc approx. 1:coh+inc. 2:s-xtal Q-E. 3:s-xtal Q-Q
INSTR = 0 # 0:VISION 1:indirect 2:direct 3:S(Q,E) or S(Qx,Qy) map
TEMP = 0.00 # Temperature [K]
E_UNIT = 0 # Energy unit [eu] (0:cm-1,1:meV,2:THz)
OUTPUT = 0 # 0:standard, 1:restart, 2:SPE, 3:full, 4:DOS, 5:modes, 6:S(Q)
```

```
## Additional general parameters
MAXO = 10 # Maximum order of excitation
CONV = 2 # Start convolution from order=CONV (2 or 3)
PHASE = 0 # Phase factor of polarization vector (0 or 1)
MASK = 0 # Set 1 to apply mask on S(Q,E)/S(Qx,Qy) map (INSTR=3)
ELASTIC = -0.10E+01 -0.10E+01 # e1>0:add el in S(Q,E),e2>0:S(Q) res
```

```
## E parameters
MINE = 8.00 # Energy range (minimum) to calculate [eu]
MAXE = 5000.00 # Energy range (maximum) to calculate [eu]
dE = 1.000 # Energy bin size [eu]
ECUT = 8.000 # Exclude modes below this cutoff energy [eu]
ERES = 0.25E+01 0.50E-02 0.10E-06 # E resolution coeff
```

```
## Q parameters
MINQ = 0.50 # Q range (minimum) to calculate
MAXQ = 20.00 # Q range (maximum) to calculate
dQ = 0.50 # Q bin size
QRES = -0.10E+01 # Q resolution coeff
```

```
## Instrument parameters
THETA = 135.0 45.0 # List of scattering angles [degree]
Ef = 32.00 # Final energy [eu] (INSTR=1)
Ei = 5000.00 # Incident energy [eu] (INSTR=2)
L1 = 11.60 # L1 [m] for DGS (INSTR=2 or 3, ERES=0)
L2 = 2.00 # L2 [m] for DGS (INSTR=2 or 3, ERES=0)
L3 = 3.00 # L3 [m] for DGS (INSTR=2 or 3, ERES=0)
dt_m = 3.91 # dt_m [us] for DGS (INSTR=2 or 3, ERES=0)
dt_ch = 5.95 # dt_ch [us] for DGS (INSTR=2 or 3, ERES=0)
dL3 = 3.50 # dL3 [cm] for DGS (INSTR=2 or 3, ERES=0)
```

```
## Single crystal parameters
HKL = 0.0 0.0 0.0 # HKL (TASK=2 or 3)
Q_vec = 0.0 0.0 1.0 # Q vector dir (TASK=2 or 3)
Q_vec_y = 1.0 0.0 0.0 # Q vector dir y-axis (TASK=3)
MINQ_y = 1.00 # Q range (minimum) y-axis (TASK=3)
MAXQ_y = 2.00 # Q range (maximum) y-axis (TASK=3)
dQ_y = 0.02 # Q bin size y-axis (TASK=3)
```

```
## Wing parameters
WING = 0 # Wing calculation (0:no wing,1:isotropic,2:ST tensor)
A_ISO = 0.0350 # Isotropic A_external for wing calculation
W_WIDTH = 150.0 # Energy width [eu] of initial wing
```

- Powder and single crystal
- Coherent and incoherent
- Temperature effect
- Wing calculation for single molecules
- Instrument geometry and resolution
- Various output formats

Ab initio thermal neutron scattering cross-section of solids

$$\sigma_{total}(E_i) = \sigma_{sc}(E_i) + \sigma_{abs}(E_i)$$

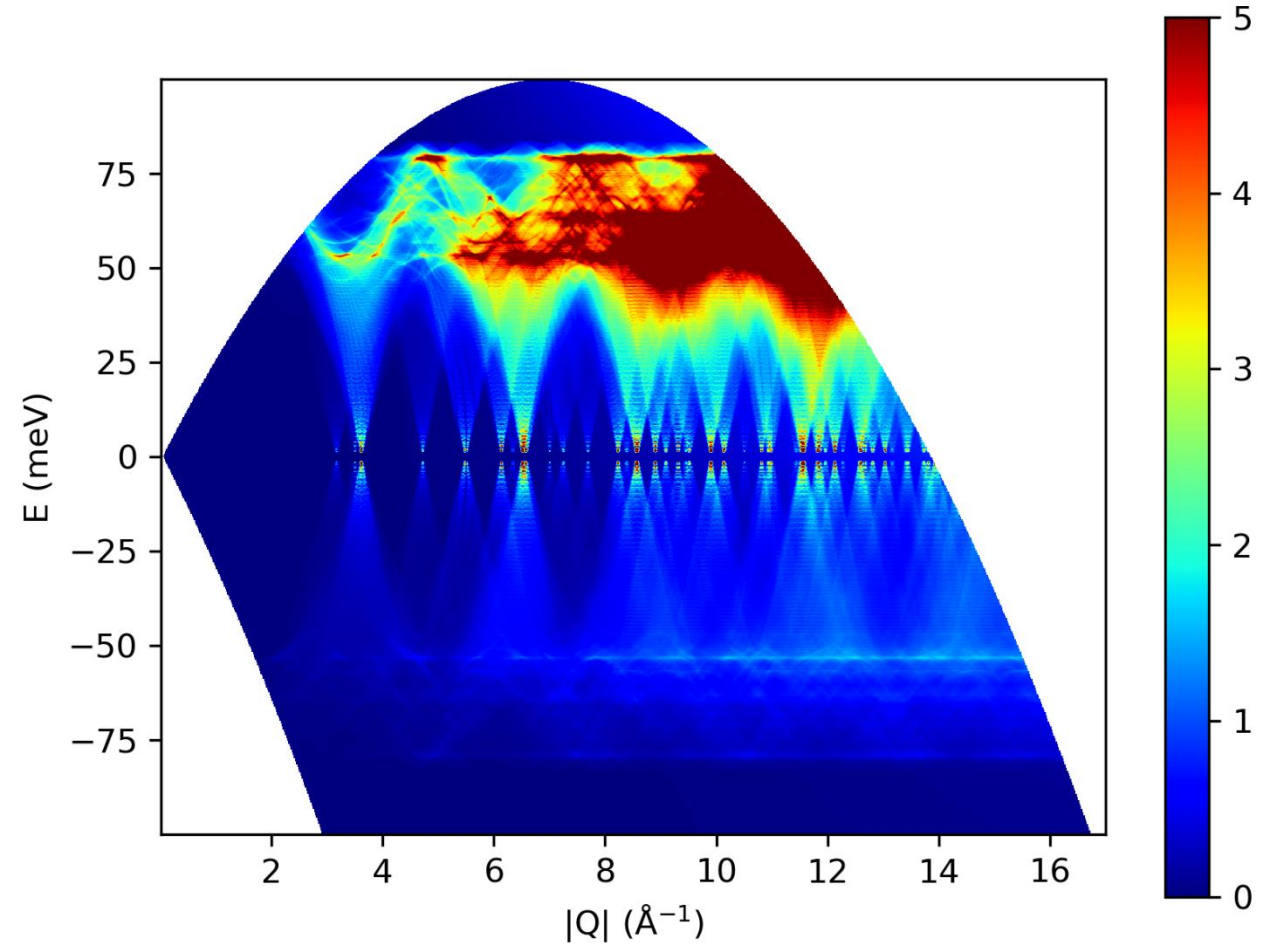
$$\sigma_{abs}(E_i) = \sigma_{abs}(25.30 \text{ meV}) \sqrt{\frac{25.30 \text{ meV}}{E_i}}$$

$$\sigma_{sc}(E_i) = \sigma_{coh}^{el}(E_i) + \sigma_{inc}^{el}(E_i) + \sigma_{coh}^{inel}(E_i) + \sigma_{inc}^{inel}(E_i)$$

$$\sigma_{sc}(E_i) = \frac{N}{2k_0^2} \int_0^\infty Q dQ \int_{\omega_{min}}^{\omega_{max}} S(Q, \omega) d\omega$$

$$\omega_{max}(Q, k_0) = \frac{\hbar k_0 Q}{m} \left(1 + \frac{Q}{2k_0}\right)$$

$$\omega_{min}(Q, k_0) = \frac{\hbar k_0 Q}{m} \left(1 - \frac{Q}{2k_0}\right)$$



MD trajectories to INS: More examples

