Refinement of a Methyl Torsional Potential against Quasi-elastic Neutron Scattering Experiments

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

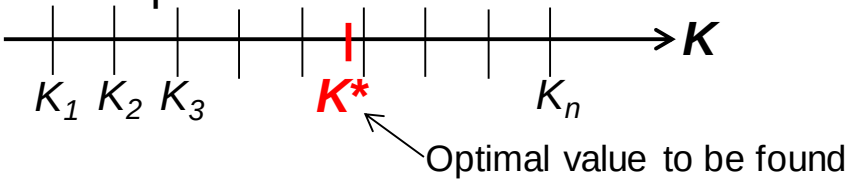
Refinement Strategy

We aim to refine *a single parameter* in a molecular dynamics force-field by comparison against Quasi Elastic Neutron Scattering (QENS) experiments

Typical scenarios when single parameter refinement is warranted:

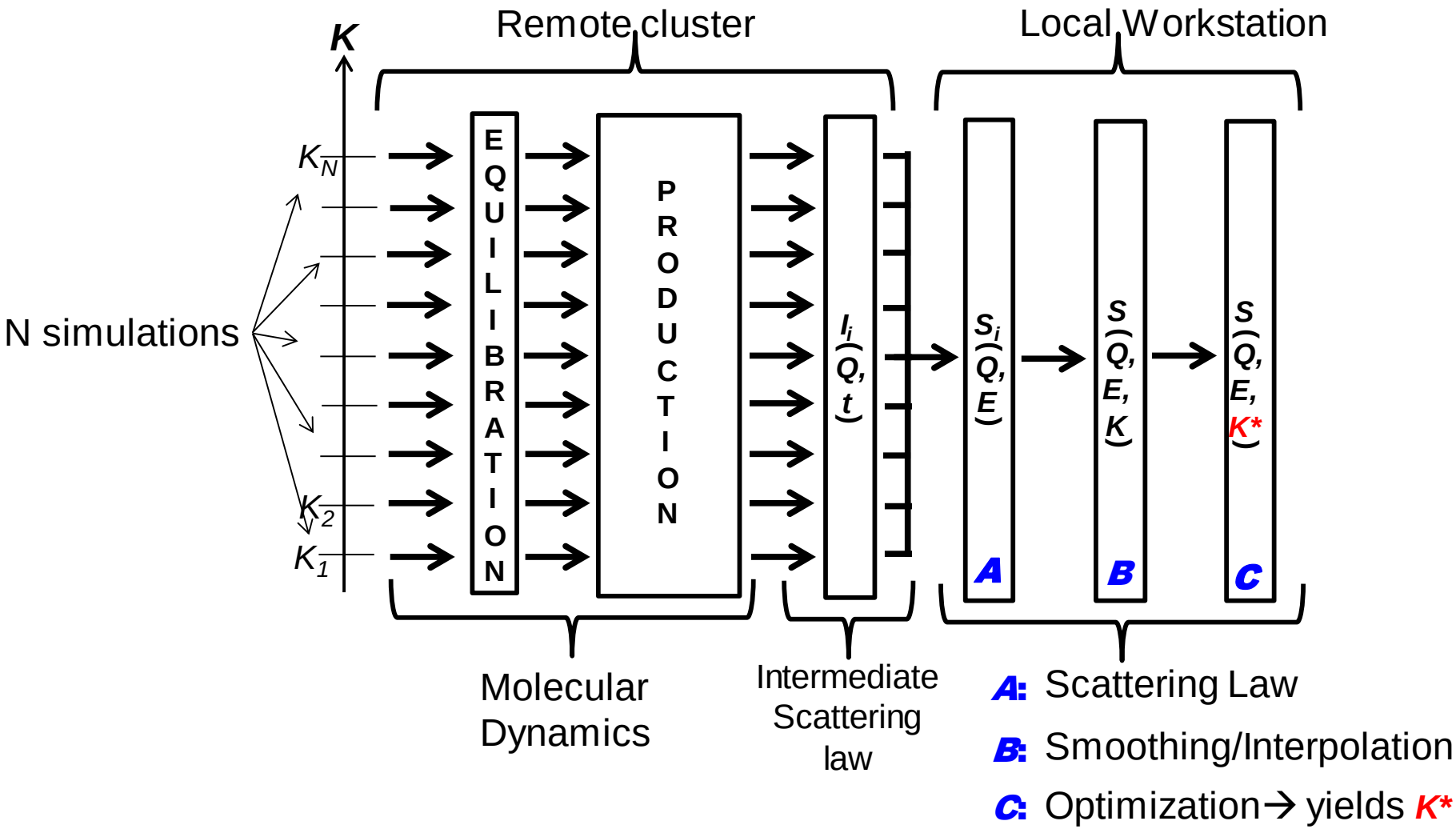
- Diffusion highly sensitive to a single parameter (QENS signal determined by H's)
- Coarse-grained simulations (few parameters)

Simple strategy: Perform a series of simulations in a range of parameter values, each with a fixed value of force-field parameter *K*.



Main advantage: simulations are run in parallel (take less than a day)

Refinement Workflow



The Sampling Problem

Observation: Molecular Dynamics (MD) simulations are finite in time and size.

→ Errors in the correlation functions

$$\left\langle b_k^2 e^{-iQ[r_k(t_0)-r_k(t_0+t)]} \right\rangle_{\Omega_Q, t_0, k} \rightarrow S(Q,E)$$

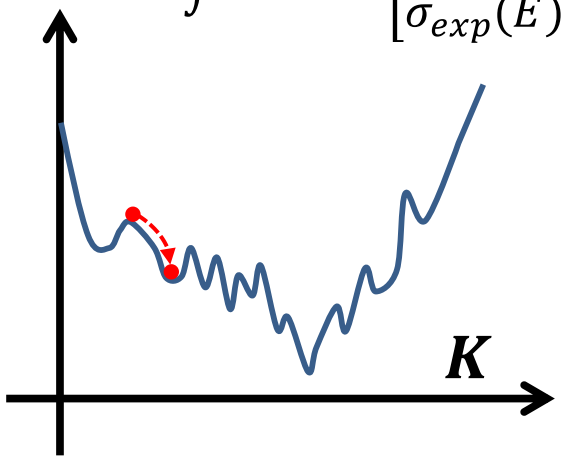
Why do we care?

Errors produce a rugged landscape in χ^2

$$\chi^2(K) = \int \frac{\left[S_{exp}(E) - S_{model}(E,K)\right]^2}{\left[\sigma_{exp}(E)\right]^2} dE$$

→ $\partial_K \chi^2(K)$ is ill defined

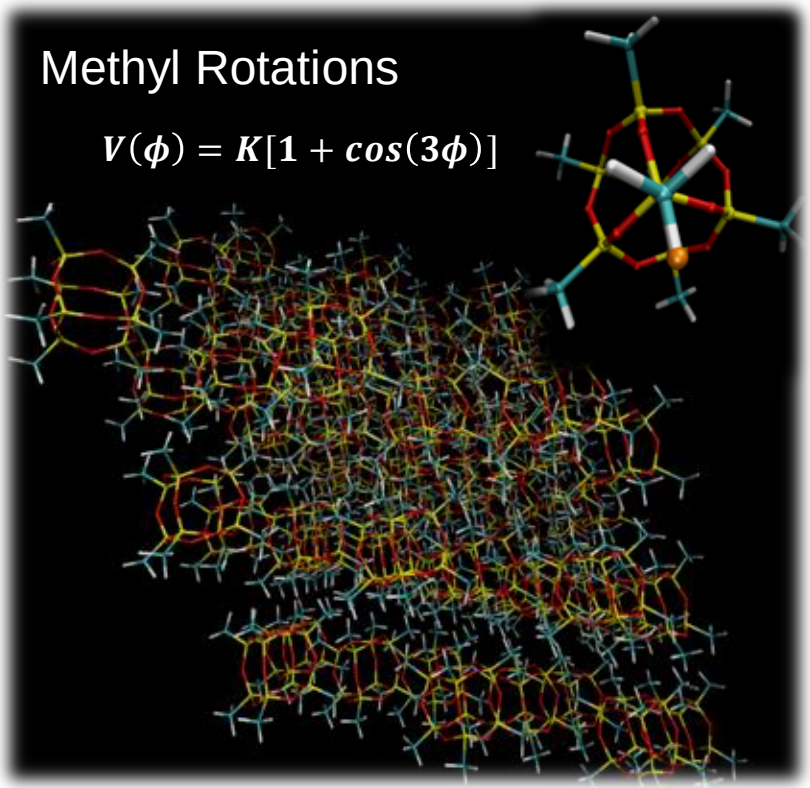
→ Minimization of $\chi^2(K)$ will get stuck in a (false) local minimum.

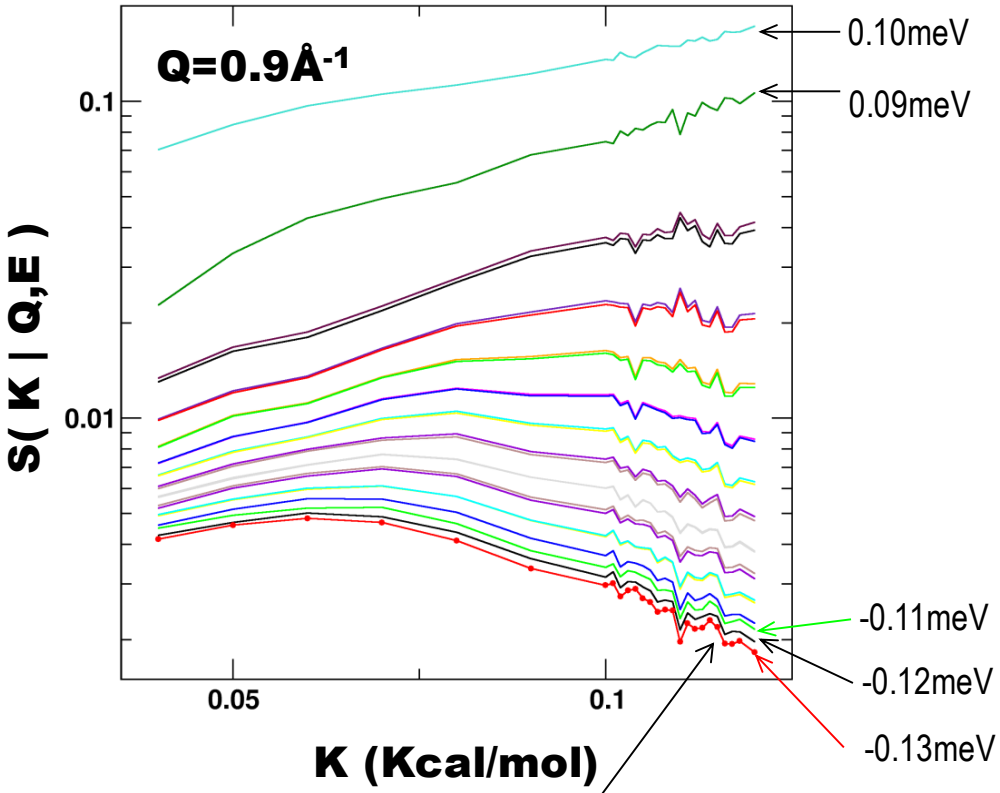


We must smooth and interpolate the set of scattering laws $\{S_i(Q,E) = S(Q,E,K_i)\}$ prior to optimization.

The Sampling Problem: Example

- Methyl rotations in Octa-Methyl Silsesquioxanes

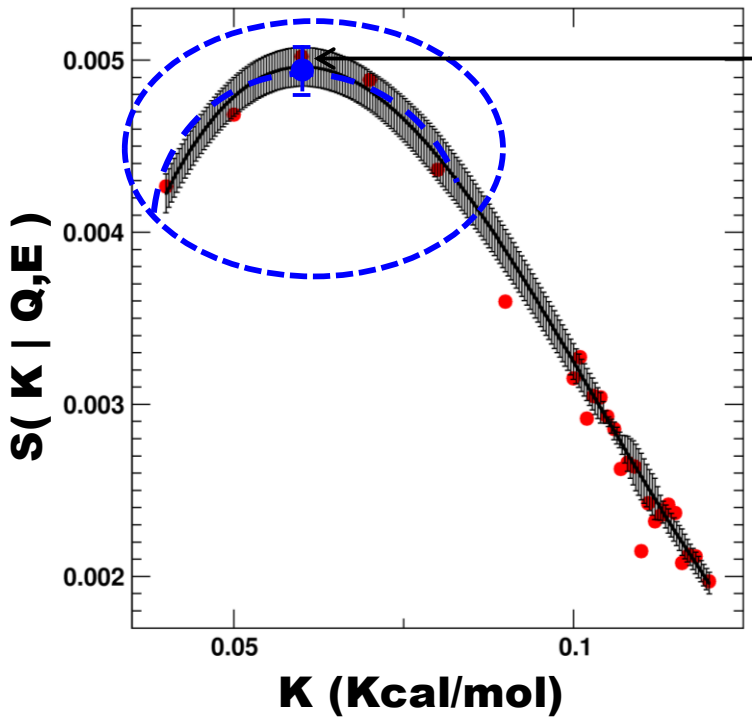




Simulation-derived scattering laws are not smooth

Smoothing the Scattering Law

1. Start with the set of simulation-derived scattering laws { $S(K_i)$ }
2. Go through every point K_i and perform a local regression



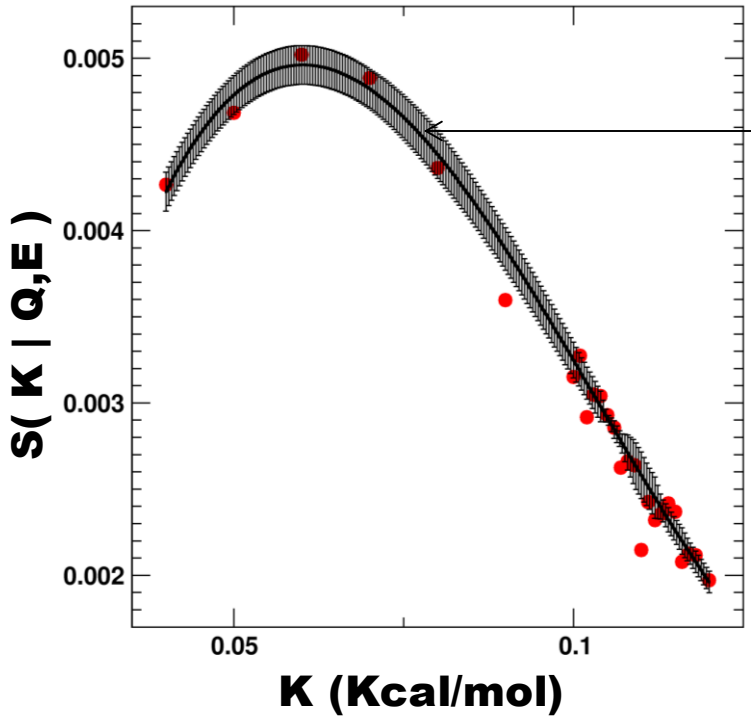
After this local regression, the blue point substitutes the red point.

After the local regression step, we end up with a new set of scattering laws, and associated errors

$$\{ S(K_i) \} \rightarrow \{ S'(K_i) \pm \sigma(K_i) \}$$

Smoothing the Scattering Law

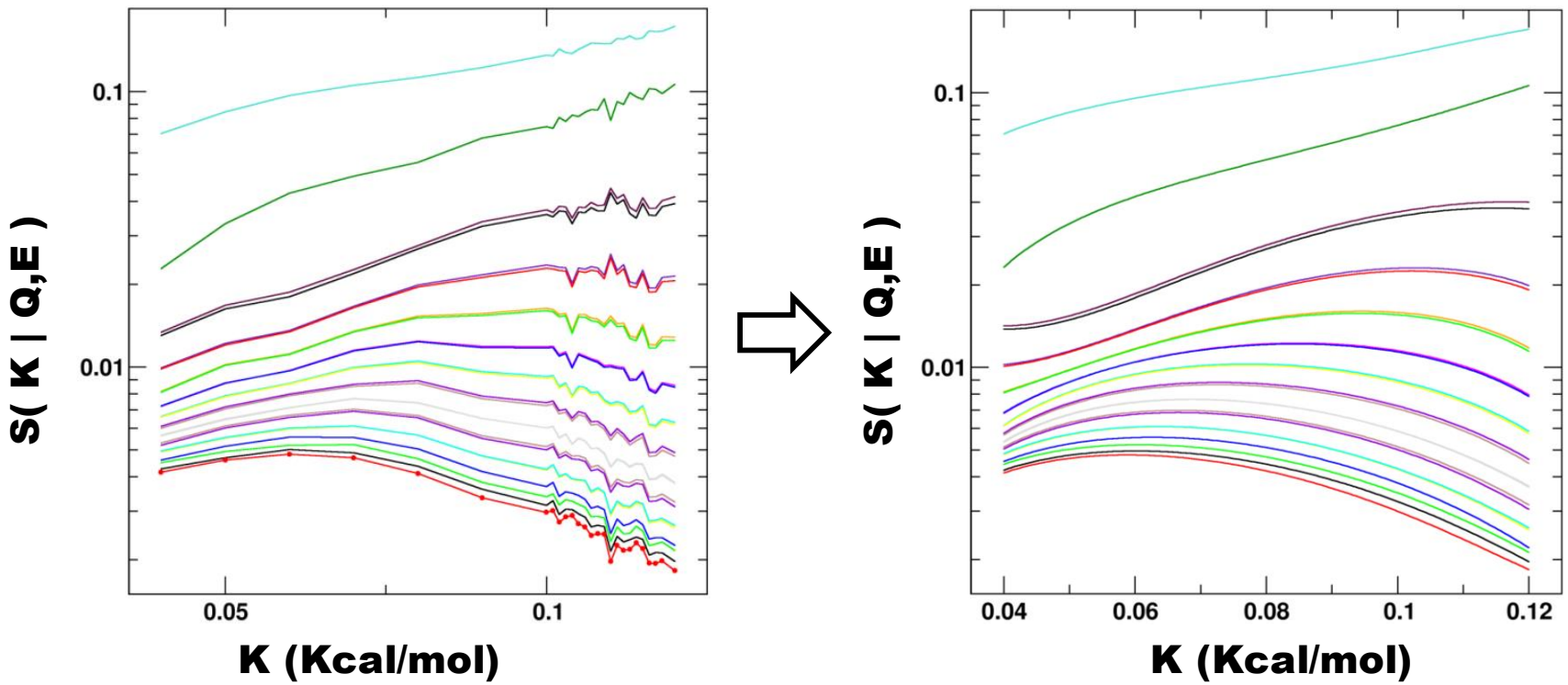
3. Construct spline: { $S'(K_i) \pm \sigma(K_i)$ } $\rightarrow S(K) \pm \sigma(K)$



Spline is smooth, suitable for optimization

Spline does not necessary pass through the original (K_i ,$S(K_i)$) points

Smoothing the Scattering Law

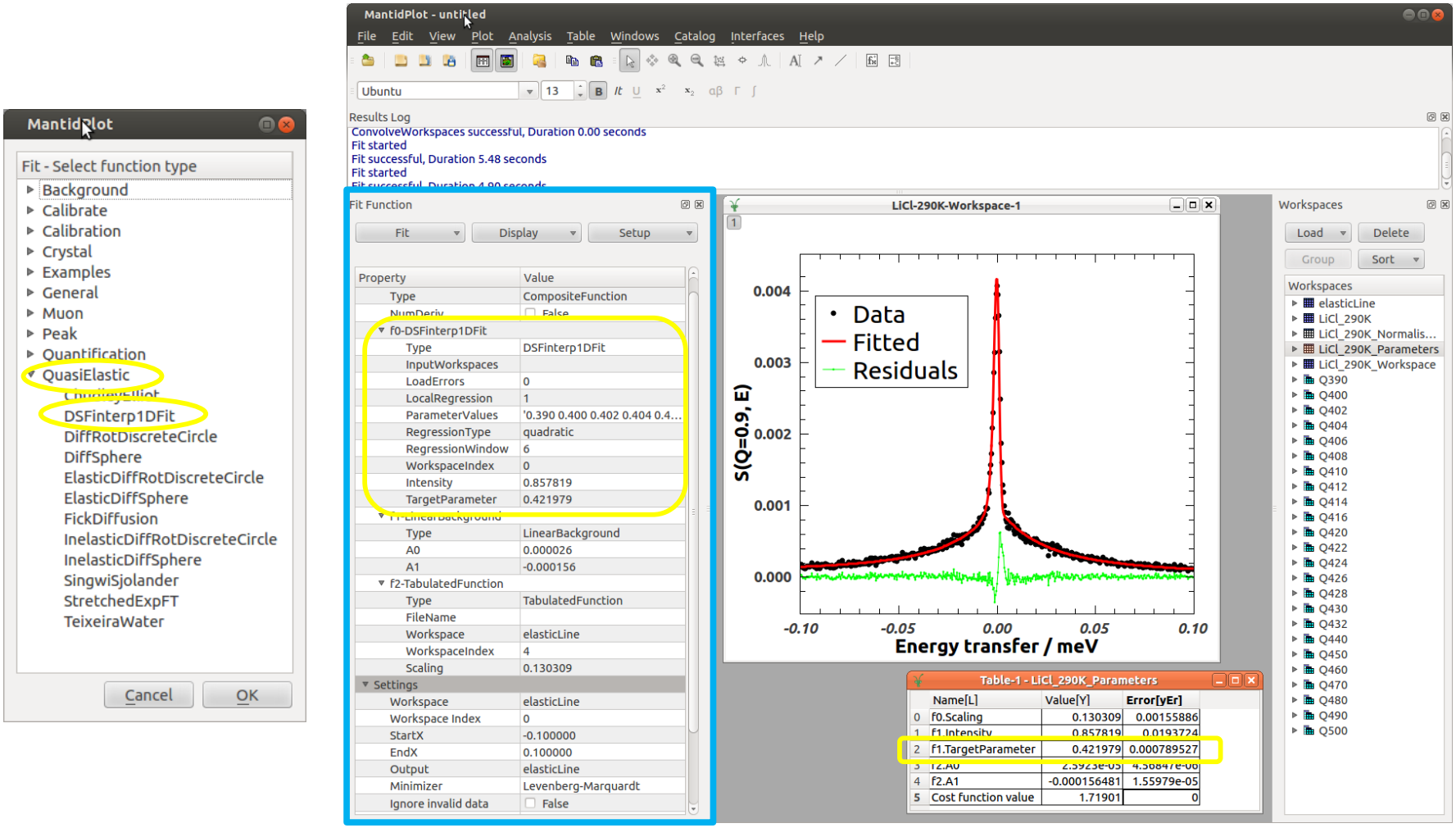


- *S(K,Q,E)* is smooth and derivable, like an analytical model
- *σ(K,Q,E)* error estimation for the simulations

$$\chi^2(K) = \int \frac{\left[S_{exp}(E) - S_{sim}(E,K)\right]^2 + \left[\sigma_{sim}(E,K)\right]^2}{\left[\sigma_{exp}(E)\right]^2} dE$$

Implementation in Mantid: DSFinterp1DFit

- http://www.mantidproject.org/DSFinterp1DFit



Fit Using the Script Window

$$S_{model}(Q,E) = Res(Q,E) \otimes [a_0(Q) \cdot \delta(E) + a_1(Q) \cdot S_{sim}(Q,E)] + L.B.$$

Load experimental S(Q,E) and resolution

Load simulated I(Q,t,K_i)

Fourier Transform and convolve with resolution

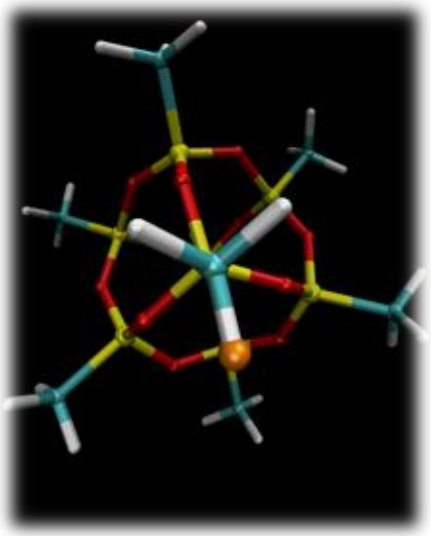
Define the fit function a_0*R(Q,E)+a_1*S(Q,E)+LB

File Edit Execute Window

showcase2.py

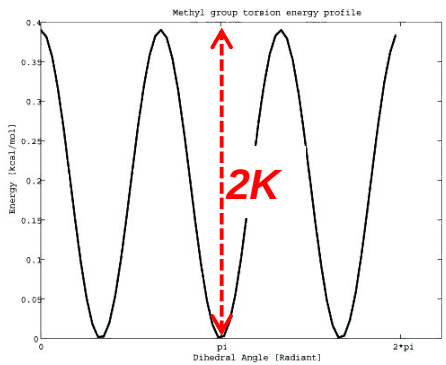
```
rootd='/projects/resea /POSS/me8t8.v2/expdata'
LoadNexus(Filename='{0}/elastic.nxs'.format(rootd),                    c')
LoadNexus(Filename='{0}/exp200K.nxs'.format(rootd),                    K')

rootd='/projects/research/POSS/me8t8.v2/SingleMolecu                   0_Ksearch
for K in '0.040  0.050  0.060  0.070  0.080  0.090                    00'.split():
    LoadSassena(Filename='{0}/K{1}/fqt_inc_run0_rms2first.h5'.format(rootd,K), Time                    {0}'.format(K))
K='0.110'
LoadSassena(Filename='{0}/K{1}/fqt_inc_run1_rms2first.h5'.format(rootd,K), TimeUnit=                    .format(K))

for K in '0.040  0.050  0.060  0.070  0.080  0.090  0.100 0.110'.split():
    SassenaFFT(InputWorkspace='incSMK{0}'.format(K), FFTonlyRealpart=1, DetailedBalance=                    mp=200)
    Rebin(InputWorkspace='incSMK{0}_sqw'.format(K), Params=[-0.2,0.0004,0.2], Output                    orkspace='incSMK{0}_sqw'.format(K))
    ConvolveWorkspaces(Workspace1='elastic', Workspace2='incSMK{0}_sqw'.format(K), OutputWorkspace='simSMK{0}'.format(K))
    Scale(InputWorkspace='simSMK{0}'.format(K), Factor=1.0e-05,Operation='Multiply', OutputWorkspace='simSMK{0}'.format(K))

parametervalues='0.040  0.050  0.060  0.070  0.080  0.080 0.100  0.110'
workspaces=''
for K in parametervalues.split(): workspaces += ' simSMK{0}'.format(K)
fit_string = 'name=TabulatedFunction,Workspace=elastic,WorkspaceIndex={0},Scaling=1,constr
    'name=DSFinterp1DFit,InputWorkspaces="{0}",ParameterValues="{1}",'.format(workspace
    'LoadErrors=0,LocalRegression=1,RegressionType=quadratic,RegressionWindow=6,' +\
    'WorkspaceIndex=8,Intensity=1.0,TargetParameter=0.11,' +\
    'constraints=(0.001<Intensity,0.04<TargetParameter<0.12);'.format(fmin,fmax) +\
    'name=LinearBackground,A0=0.0,A1=0.0'
    Fit(fit_string, InputWorkspace='exp200K', WorkspaceIndex=7, StartX=-0.13, EndX=0.1, CreateOutput = 1 )
```

Test: Octa-Methyl Silsesquioxane



Methyl Rotations

$$V(\phi) = K[1 + cos(3\phi)]$$



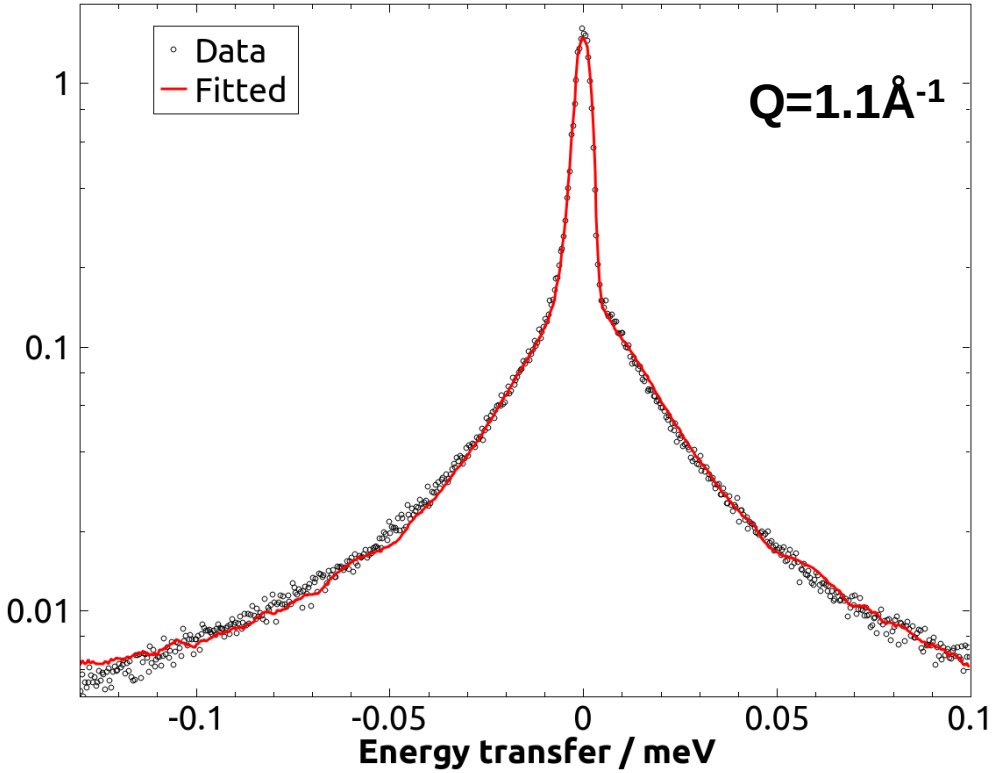
- Experiments at BASIS@SNS: E_a=*1.22 Kcal/mol* → ***K_{exp}=0.068 Kcal/mol***
- Initial guess K=0.11 Kcal/mol (off by 62%)

Eight NVT simulations of a single molecule, each 10ns long, at T=200K

K={0.04, 0.05, … , 0.11}

Test: Octa-Methyl Silsesquioxane

$$S_{model}(Q,E) = Res(Q,E) \otimes [a_0(Q) \cdot \delta(E) + a_1(Q) \cdot S_{sim}(Q,E)] + L.B.$$



Q	L	K*	X^2
0.3	20.9	**0.0635**4	1.83
0.5	12.6	**0.0635**0	2.44
0.7	9.0	**0.0631**3	2.20
0.9	7.0	**0.0639**8	2.69
1.1	5.7	**0.0647**2	2.43
1.3	4.8	**0.0649**5	2.54
1.5	4.2	**0.0650**8	2.89
1.7	3.7	**0.0651**8	3.55
1.9	3.3	**0.0649**3	4.13

$\langle K^* \rangle_Q$= **0.0643** Kcal/mol
(off by 5%)

- Derivation of ***K**** from a single temperature (200K)