Development of Single-Magnon Cross-Section/ Spin Wave Dispersion Software
Breakdown

**TASK**
- Understand properties of magnetic solids
  - Simulate interactions with magnetic lattices
    - Simple systems – very tedious
    - Complex systems – often impossible to do by hand
  - Analyze experimental data
    - Fitting

**GOAL**
- SOFTWARE
  - Improve current features of software in development
  - Expand this software to allow more functionality
Magnetism

- Used often in everyday life
  - Hard drives, credit cards, VHS
  - Transformers, generators
  - Electric Motors
  - Speakers, microphones
Atoms and Crystals

- Atoms - protons, neutrons, electrons
  - Building blocks of matter

- Crystals
  - Regular, repeating arrangement of atoms
  - Unit cell - smallest unique arrangement of atoms
  - Translations of unit cell can reconstruct entire lattice
Electrons throughout lattice interact
  - Like charges repel
    - Coulomb Repulsion
  - Intrinsic quantized angular momentum - SPIN
    - Spin vector \((S_x, S_y, S_z)\) points in direction of spin
    - Pauli Exclusion Principle
  - Moving electrons create magnetic field
    - Magnetic fields of multiple electrons interact
Ground states

- Depending on interactions, different patterns of spins on the lattice can develop
- Ground state is spin configuration with least energy
- Example:
  - Ferromagnetic
  - Antiferromagnetic
Approximate interactions by pair-wise spin interactions

\[ H = - \sum_{ij} \vec{S}_i \cdot J_{ij} \vec{S}_j - \sum_i \sum_\alpha D_i^\alpha (S_i^\alpha)^2 \]

1\textsuperscript{st} term – Interactions between two spins
2\textsuperscript{nd} term – Anisotropy

- Point spins preferentially along a particular direction in space
Simulated Annealing

- Monte Carlo Simulation
  - Time Independent
  - Minimize energy
  - Global minimum of Hamiltonian $\rightarrow$ ground state
  - Results not perfect
  - Implemented in C
Local optimization

- Simulated annealing puts us in the ball park of the global minima
  - Local optimization can then focus on the correct region of space to obtain more accurate result
- Rewrite energy as a matrix product
- Use sparse matrices to save memory
- Implemented in python and very fast!

Note: Not actual data
Starting Point

- Tom Sarvey
  - 1,000s of lines of code!
  - Open source; python & C
  - Graphical interface

- Needs:
  - Expansion – capability to calculate spin wave dispersion, cross-section, fits, optimization, etc.
  - Testing – fixing bugs, establishing testing suite
  - Maintenance – updating code
Program Demo: Lattice Creation, Bond Forming, Spin Ground State Simulation
Excited states

- *Naïve excitation*: Flip direction of one spin
  - High energy cost
- *Reality*: Spread spin reversal over many spins
  - Much lower energy cost
  - Superposition of states with one reduced spin
- Spin waves are the propagation of this misalignment of near-neighbors’ spins
**Dispersion Calculation**

- Calculate Hamiltonian
- Calculate Eigenvalues
- Eigenvalues are energies of spin wave modes
- Numeric and analytic results produced simultaneously
- Tested analytic results for simple cases

(A) Ferro: Chain; Square; Simple, Face-Centered, Body-Centered Cube
Program Demo: Dispersion Calculation
The Cross Section

- Number of neutrons scattered per second into an angle Ω with energy in [E, E+dE’]
- Measurable in scattering experiment
- One magnon cross section
- One Magnon – quantized spin wave with energy of ±ℏω
- We take Linear Approximation

Example: Simple ferromagnet quantized along z – axis

\[
\frac{d^2\sigma}{d\Omega dE} \approx \frac{(\gamma k_0)^2 k^2}{2\pi^2} \frac{(2\pi)^3}{V_0} \frac{1}{2} \left(\frac{2}{\hat{\gamma}}\right)^2 \sum_{\alpha\beta} \left( \frac{\hat{\gamma} \exp(i\hat{\omega} \tau) \exp(i\hat{\omega} q) \exp(-i\hat{\omega} \tau) \exp(-i\hat{\omega} q)}{4\pi} \right) dt
\]
What did I actually do?

TASKS
- Cross-section Calculation
- Optimization
- Fitting – Mpfit
- Testing Suite
- General code updates
- Pretty printing

COMPLICATIONS
- SymPy has little support for non-commutative algebra
  - Rewrote substitution evaluation methods in core multiplication file
  - Sent in patch for review
    - Not accepted
      - Complications with newest version of SymPy
    - Currently rewriting
Pretty printing

**LATEX GUI OUTPUT**
- Generates GUI text field containing LaTeX-ified output
- Uses Python Multiprocessing Module
- Complications
  - MainLoop() control
  - Processes Not Completely Independent

**LATEX COMPILER**
- Generates .tex, .pdf, .dvi file containing expression
- Uses Python Subprocessing Module
- Complications
  - Equation Breaking
  - Overfull Boxes
  - Requires LaTeX on machine
  - Package use
    - amsmath vs. revtex4
GUI POP-UP

COMPiled LATEX DOCUMENT

1. Dispersion Eigenvalues

\[-5.78880738293128e-5 \sin (kz) - \frac{1}{2} \sqrt{-127.9999975541152^2 \cos (kz) + 63.9999822984852^2 + 63.9999822984852^2 \cos^2 (kz)}\]

\[-5.78880738293128e-5 \sin (kz) - \frac{1}{2} \sqrt{-127.9999975541152^2 \cos (kz) + 63.9999822984852^2 + 63.9999822984852^2 \cos^2 (kz)}\]
Future

- Fitting data
- Resolution convolution
- Powder average
- Domain average
- Better non-commutative algebra system
- Comprehensive testing
- Widespread distribution
Acknowledgements

- William Ratcliff
- Tom Sarvey
- Julie Borchers
- Fellow Surfers
- NCNR Staff
- CHRNS
- NIST
- NSF