Fast sparse Fourier transformations for NMR spectroscopy

Badruddin Kamal u5429737 M.Comp@ ANU
Prof. Thomas Huber Dept. Chemistry@ ANU
Prof. Alistair Rendell CECS@ ANU
Background

• NMR data is used by scientists to understand molecular structure of proteins. It is an essential step for synthesizing new protein molecules.

• For this purpose NMR spectral data is gathered from spectral equipment.

• However taking uniform frequency data and doing calculations on it can be quite tedious. As it would take long hours to calculate this data.

• By sparsely and randomly taking data points at certain thresholds the calculation time can be decreased by several hours. [2]

• Goal is to reconstruct actual wavelets from sparse samples.
Background

First off for the process we sparsely sample NMR data at certain data points. After that we apply FFT to the Fourier transformed version of the wave. Sparse FFT is widely used in voice processing to network signal processing, where waves are sparsely sampled at intervals. The project will focus on applying this in computational chemistry.
Background

- After taking sparse sampling data in the time domain, we apply IST (it thresholds to sections of the Fourier transformed wave) and IFT on it iteratively to build the wave [2].
- At the end of our iterations we are left with the sample of the original wave. Which is used for experiments.
- Using this technique the data processing time can be reduced by several hours. Which is our main goal combined with application of other faster processing techniques (if time permits).
Approach

• Sparse Sampling quality[3]
  • Under sampling can lead to loss of information in waves.

• Time
  • It can take quite a while to process this data even after applying the methods discussed.
  • Process time is platform and application build quality dependent.

• Development cycle planned on is spiral development. With milestones.

• Software used
  • Fftw
  • Swig
  • Mpi(experimental)
Implementation

• Developed C code for the basic algorithm.
• python GUI.
• Optimized performance by implementing HPC techniques.
• Eg Loop unrolling.
• With HPC time decreased by $1/3$.

• MPI coded but still buggy as fftw3 mpi extension issues.
• And MPI shows more promise in time. However, Lack of support at NCI (fftw3 installation issues) the results were inconclusive. An aspect that can be looked into in the future.
Sample GUI application
Application
### Application Performance Comparison

<table>
<thead>
<tr>
<th>Platform</th>
<th>Number of Complex Points</th>
<th>Time(Ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUDA 4 CPU</td>
<td>1024</td>
<td>104.8</td>
</tr>
<tr>
<td>Bruker AMX 500 -128 CPU</td>
<td>1024</td>
<td>154</td>
</tr>
<tr>
<td>My Implementation Intel XEON NCI -1 CPU</td>
<td>3277</td>
<td>1550</td>
</tr>
</tbody>
</table>
Performance on 50% sparse file

![RMS vs iterations graph showing performance with different thresholds (thresh 0.0 to thresh 1.0).](image-url)
Performance on 500 iterations convergence

![Graph showing rms vs sparsity for different thresholds](image)
In conclusion

• The Data tested was generally coded and optimized for a single processor.

• It converges very well in a few 100 iterations. And at max 90% sparsity

• The algorithm works fine for NMR. But can be used for compression of wave data and Image processing. It may have some applications in those fields besides chemistry.
References


2. Application of iterative soft thresholding for fast reconstruction of NMR data non-uniformly sampled with multidimensional Poisson Gap scheduling, Sven G. Hyberts, Alexander G. Milbradt, Andreas B. Wagner, Haribabu Arthanari, Gerhard Wagner