High Performance Computing for Neutron Tomography Reconstruction
A Parallel Approach to Filtered Backprojection (FBP)

Zongpu Li[1] Cain Gantt[2] Rick Archibald[*]

[1]Department of Physics and Materials Science
City University of Hong Kong

[2]Department of Chemistry, Astronomy, and Physics
Georgia College and State University

[*]Mentor, Oak Ridge National Laboratory (ORNL)

August 3, 2017

Outline

1. Background
   - What is Laminography?
   - Filtered Backprojection Algorithm
   - MATLAB Data
2. Objectives
3. Methods
   - Filter
   - Serial Program
   - Parallel Program
4. Future Work

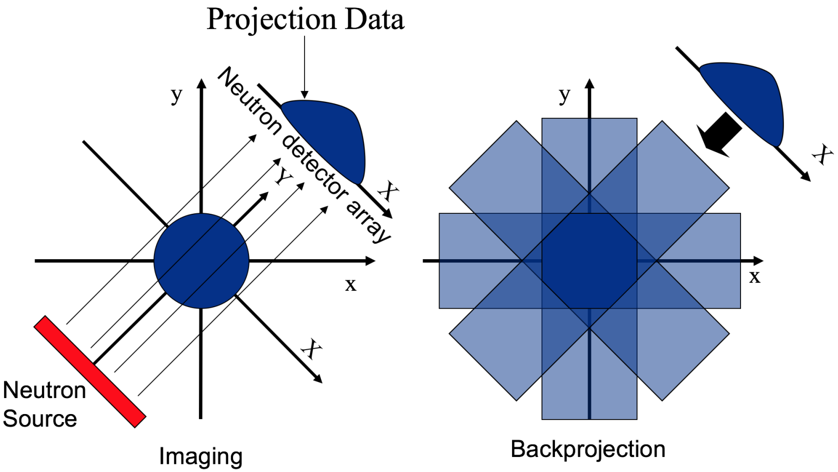
Outline

Laminography & Tomography

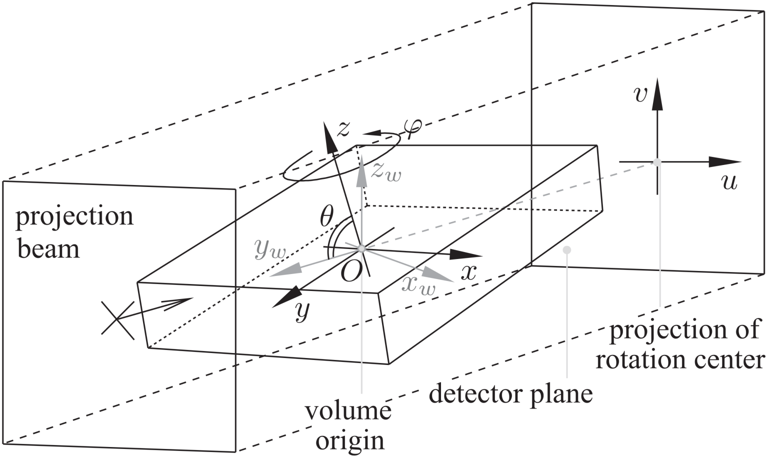
- Image Processing
- Reconstruction of 3D volume from 2D projections (sinograms)
- Fourier/harmonic analysis (specifically Radon transform)

Tomography is special case of laminography (tilt angle $= 0^\circ$)

2D Tomography



3D Laminography



Outline

Filtered Backprojection (FBP) Algorithm

Inverse Radon Transform

$$f(x, y) = \int_0^{\pi} p_f(x \cdot \cos\theta + y \cdot \sin\theta) d\theta$$

projections & orientation information $\rightarrow$ volume

For each projection...

- clean it up (filter)
- "smear" it through the volume (backproject & interpolate)

...then sum all smeared projections

Result: reconstructed 3D volume

Filtered Backprojection (FBP) Algorithm

initialize

↓

filter projections

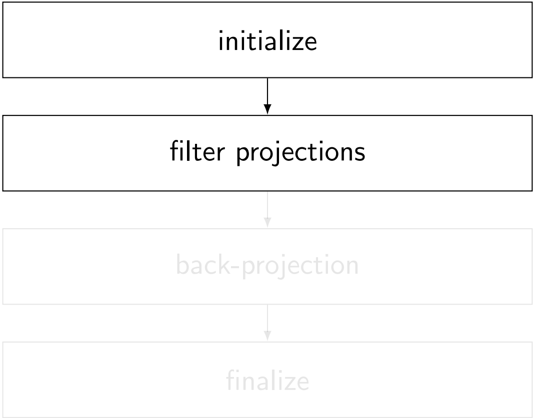
↓

back-projection

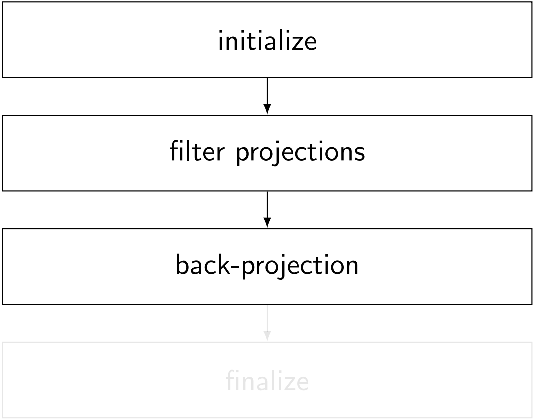
↓

finalize

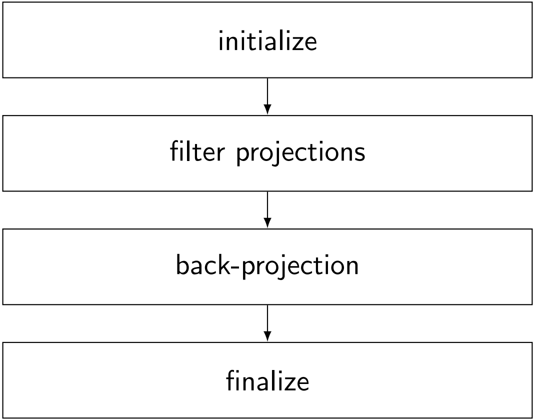
Filtered Backprojection (FBP) Algorithm



Filtered Backprojection (FBP) Algorithm

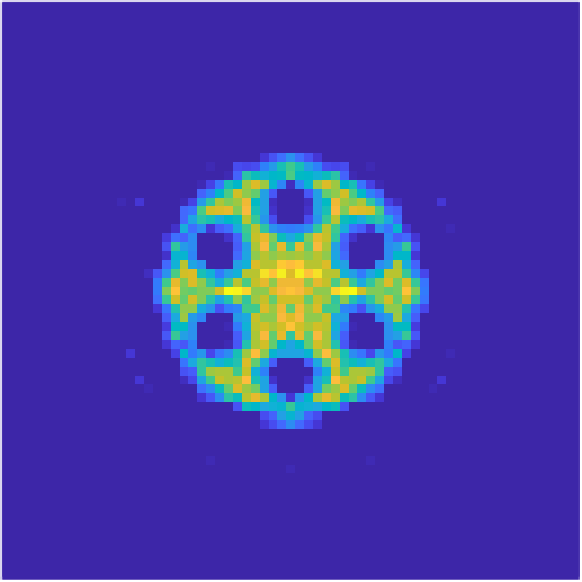


Filtered Backprojection (FBP) Algorithm



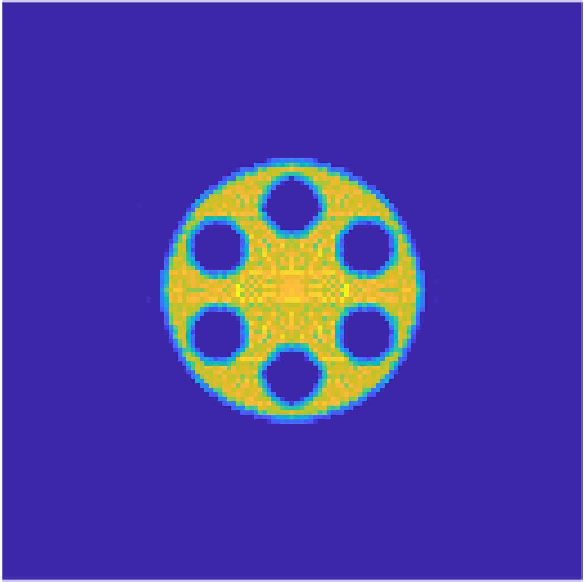
Outline

Simulation



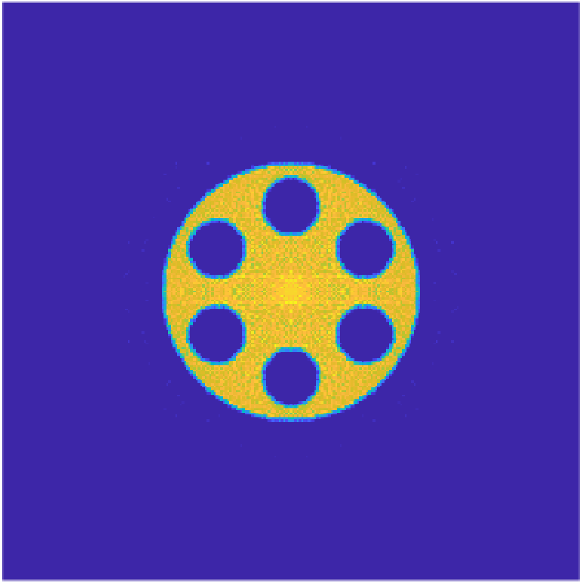
Volume: $65 \times 65 \times 65$

Simulation



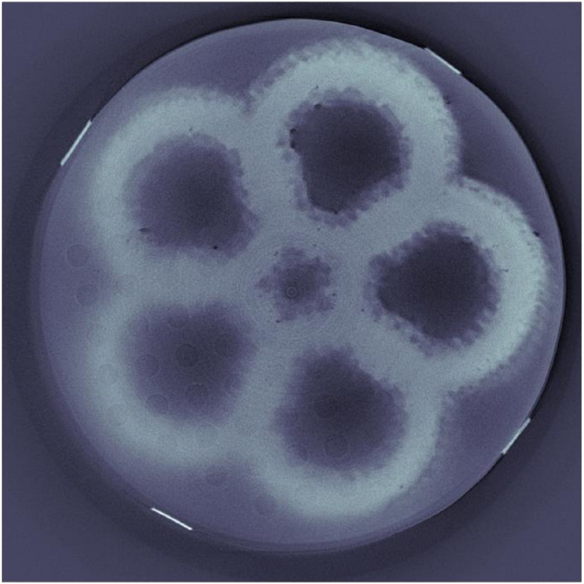
Volume: $129 \times 129 \times 129$

Simulation



Volume: $257 \times 257 \times 257$

ORNL Spallation Neutron Source (SNS) Data



“Volume”: $1501 \times 1501 \times 1$

Outline

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C
- Structure portions in parallel. . .

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C
- Structure portions in parallel...
    - Message Passing Interface (MPI)

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C
- Structure portions in parallel...
    - Message Passing Interface (MPI) → backprojection

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C
- Structure portions in parallel...
  - Message Passing Interface (MPI) $\rightarrow$ backprojection
  - Graphics Processing Unit (GPU)

Objectives

Overall reduction in code execution time and memory requirements

- Implement laminography filter
- Perform FBP serially in C
- Structure portions in parallel. . .
  - Message Passing Interface (MPI) → backprojection
  - Graphics Processing Unit (GPU) → FFT and filtering

Outline

Filter Selection

Filter necessary to reduce blurring from backprojection

- basic high-pass ramp filter (standard in tomography)
- laminographic ramp filter (scaled depending laminography angle)
- sinc filter (slightly varying frequency response)
- cutoff frequency (dependent upon experiment geometry)

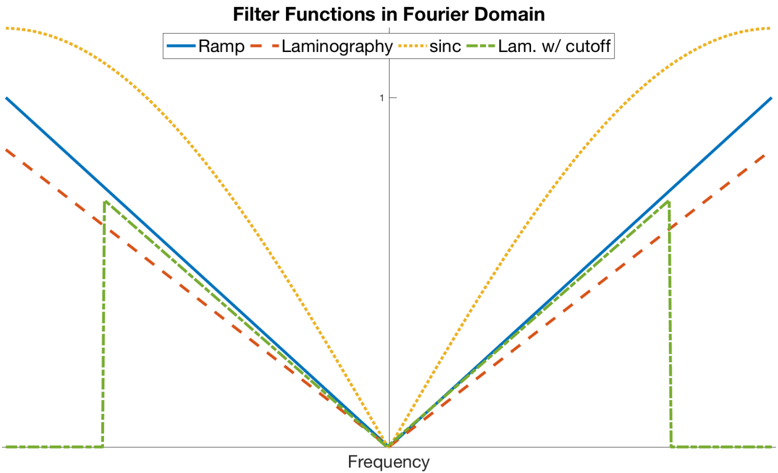
Filter Selection

Filter necessary to reduce blurring from backprojection

- basic high-pass ramp filter (standard in tomography)
- laminographic ramp filter (scaled depending laminography angle)
- sinc filter (slightly varying frequency response)
- cutoff frequency (dependent upon experiment geometry)

Using laminographic ramp filter with adjustable cutoff frequency

Filter Selection



Outline

Serial Program

Challenges

Reading data formatted for MATLAB:

- write values to binary file
- dimensions & parameters, data arrays

Serial Program

Challenges

Reading data formatted for MATLAB:

- write values to binary file
- dimensions & parameters, data arrays

Projection filtering:

- FFTW (Fastest Fourier Transform in the West)
- zero-padding on projection arrays
- forward transform, multiply filter, reverse transform

Serial Program
Challenges

Reading data formatted for MATLAB:

- write values to binary file
- dimensions & parameters, data arrays

Projection filtering:

- FFTW (Fastest Fourier Transform in the West)
- zero-padding on projection arrays
- forward transform, multiply filter, reverse transform

Backprojection:

- coordinates before & after rotation
- meshgrid?

Serial Program

Challenges

Reading data formatted for MATLAB:

- write values to binary file
- dimensions & parameters, data arrays

Projection filtering:

- FFTW (Fastest Fourier Transform in the West)
- zero-padding on projection arrays
- forward transform, multiply filter, reverse transform

Backprojection:

- coordinates before & after rotation
- single coordinate value per loop
- greatly reduced memory requirements!

Serial Program
Challenges

Reading data formatted for MATLAB:

- write values to binary file
- dimensions & parameters, data arrays

Projection filtering:

- FFTW (Fastest Fourier Transform in the West)
- zero-padding on projection arrays
- forward transform, multiply filter, reverse transform

Backprojection:

- coordinates before & after rotation
- single coordinate value per loop
- greatly reduced memory requirements!

Pointers and integers...

Outline

Parallel Program

Different Approaches

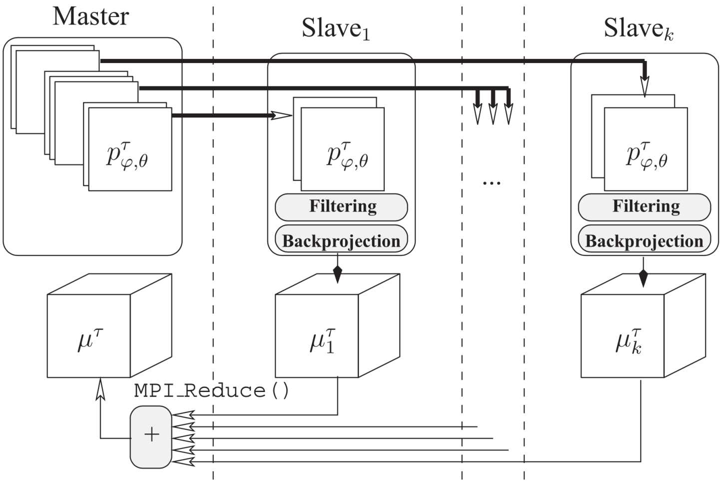
MPI (Myagotin, *et al.*):

- distribute projections (many independent projections)
- decompose reconstructed volume (single volume between all nodes)

GPU acceleration:

- FFT (many 1D transforms)
- applying filter (many multiplication operations)
- interpolation (texture mapping for hardware acceleration)

Data decomposition by projections



Intel MPI Library
Bridges at Pittsburg Supercomputing Center

Intel MPI Library
Bridges at Pittsburg Supercomputing Center

Advantages:

- shorter program execution time

Intel MPI Library
Bridges at Pittsburg Supercomputing Center

Advantages:

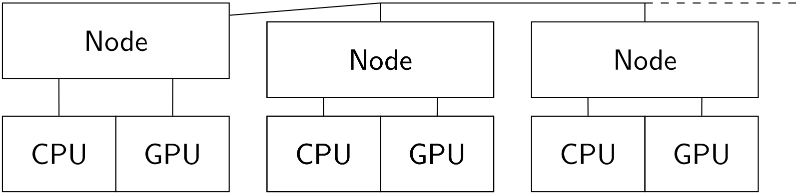
- shorter program execution time

Drawbacks:

- large memory consumption (each node has full memory in volume)
- `MPI_Reduce` operation (more operations to perform)

GPU

Hardware configuration



CUDA FFT Library (cuFFT)
CUDA Basic Linear Algebra Subroutine Library (cuBLAS)
Bridges (PSC): NVIDIA Tesla P100

CUDA FFT Library (cuFFT)
CUDA Basic Linear Algebra Subroutine Library (cuBLAS)
Bridges (PSC): NVIDIA Tesla P100

Advantages:

- functions are ready-to-go
- no need to write kernel, manually manage device memory, etc.

CUDA FFT Library (cuFFT)
CUDA Basic Linear Algebra Subroutine Library (cuBLAS)
Bridges (PSC): NVIDIA Tesla P100

Advantages:

- functions are ready-to-go
- no need to write kernel, manually manage device memory, etc.

Drawbacks:

- cuBLAS issues
- `memcopy` and device memory

CUDA FFT Library (cuFFT)
CUDA Basic Linear Algebra Subroutine Library (cuBLAS)
Bridges (PSC): NVIDIA Tesla P100

Advantages:

- functions are ready-to-go
- no need to write kernel, manually manage device memory, etc.

Drawbacks:

- cuBLAS issues
- `memcopy` and device memory

Execution time improved over serial code
Testing data sets could be too small to see major improvement?

Future Work

Much to still be done!

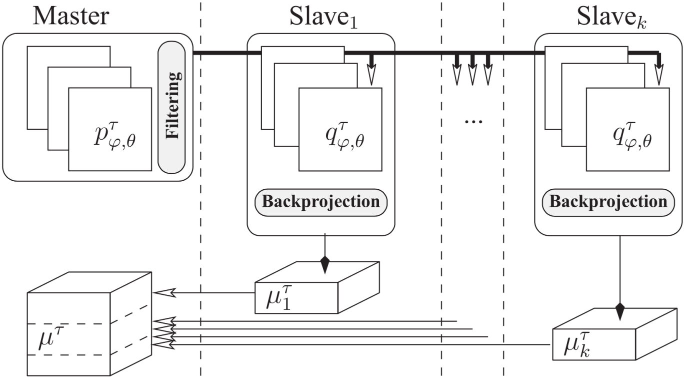
Future Work

Much to still be done!

- **bottleneck: size of volume in memory**

Future Work

Data decomposition by reconstructed volume



Future Work

Much to still be done!

- **bottleneck: size of volume in memory**
- FFT: real-valued transforms

Future Work

Much to still be done!

- **bottleneck: size of volume in memory**
- FFT: real-valued transforms
- MPI: communication improvement via "ring" method

Future Work

Much to still be done!

- **bottleneck: size of volume in memory**
- FFT: real-valued transforms
- MPI: communication improvement via “ring” method
- GPU: hardware interpolation

Acknowledgements

University of Tennessee, Knoxville (UTK)
Oak Ridge National Laboratory (ORNL)
Joint Institute for Computational Sciences (JICS)
National Science Foundation
Extreme Science and Engineering Discovery Environment (XSEDE)
Bridges system, Pittsburgh Supercomputing Center (PSC)

Dr. Rick Archibald (ORNL)
Dr. Kwai Wong (UTK, JICS)

References

Zeng, Gensheng.
Revisit of the Ramp Filter
IEEE Trans Nucl Sci., 62(1):131–136, 2015.

A. Myagotin, *et al.*
Efficient Volume Reconstruction for Parallel-Beam Computed Laminography by Filtered Backprojection on Multi-Core Clusters
IEEE Trans. Image Process., 22(12):5438–5439, 2013.