Update on Fourier-Banded Preconditioning

Carl Sovinec

Department of Engineering Physics
University of Wisconsin-Madison

Online NIMROD Team Meeting
January 16-17, 2023

Supported by US DOE grants DE-SC0018642 and DE-SC0018001

Outline

I. Description of Fourier-banded preconditioning
II. Implementation
    A. Matrix computations
    B. Parallelization
III. Performance
IV. Porting to NIMROD

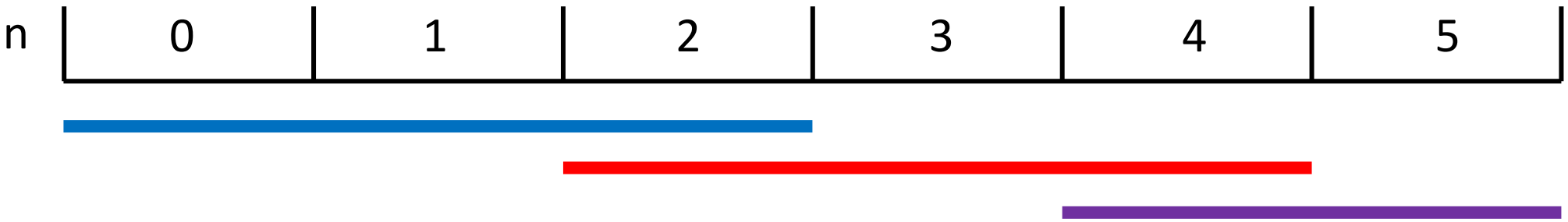
Many approaches to improve toroidal preconditioning have been tested in NIMROD.

1. Partial Gauss-Seidel-like updates with limited coupling within process layers helped in some cases.
2. Flexible GMRES (FGMRES) alternating between block Fourier diagonal (bFd) and block toroidal-plane diagonal is not better than bFd alone.
3. Kyle Bunkers tried Fourier multigrid and found that coarse-representation solves did not help the full solve.
4. 1D solves over toroidal angle showed little synergistic benefit with bFd with additive, multiplicative, or FGMRES approaches.
5. Fully coupled static condensation to reduce the system size reduces iteration count by ~ 50%. (Aug. 2019 team meeting)

Direct solves over multiple Fourier components and the full poloidal plane has been implemented in NIMSTELL.

Modifications to the ***V***- and *T*-matrix routines produce systems for bands of Fourier components.

- Bands may overlap.
- Systems use real algebra for separated real/imaginary components.



Example set of matrices with *stride* = 2 and *overlap* = 1.

Modifications are at integrand, block, and finite-element levels.

The finite-element (FE) level has a banded_matrix_create routine.

- Bands are allocated to layers, similar to Fourier components, and a layer may have >1 band.
- Integration routines maintain existing layer decomposition of blocks and are called from the FE level.
- Initialization creates arrays that record which integrand contributions are needed from each processor layer.
- Asynchronous point-to-point communication collects integrands for all blocks within poloidal-decomposition group.

A new rblock matrix routine is just an element gather operation.

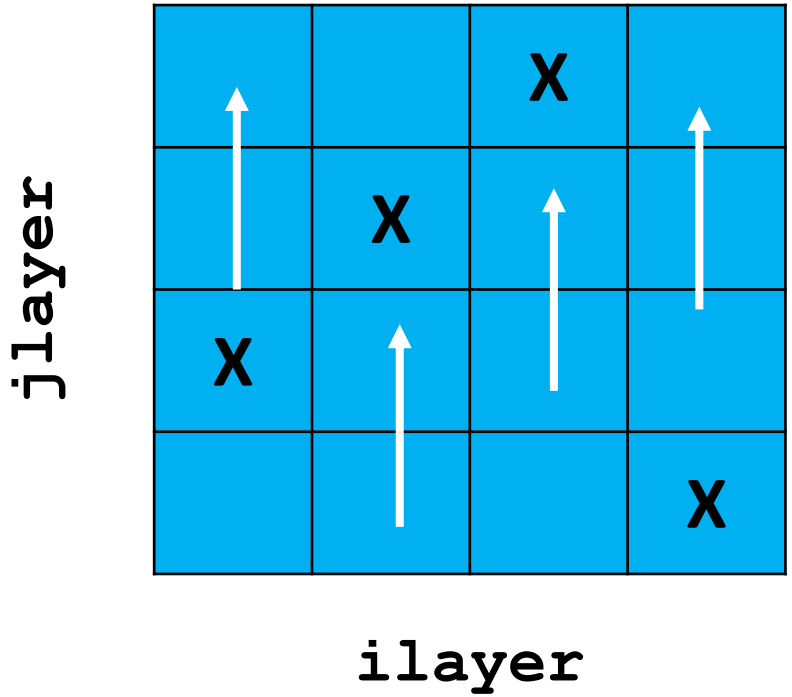
Layer-decomposed sections of the integrand array for each band are passed to the rblock_gather_real_matrix routine.

- It is told what range of elements is in the passed integrand array.
- It performs the gather operation.
- This is similar to standard block routines but does not have the integrand call so that layer communication isn't repeated for each block.

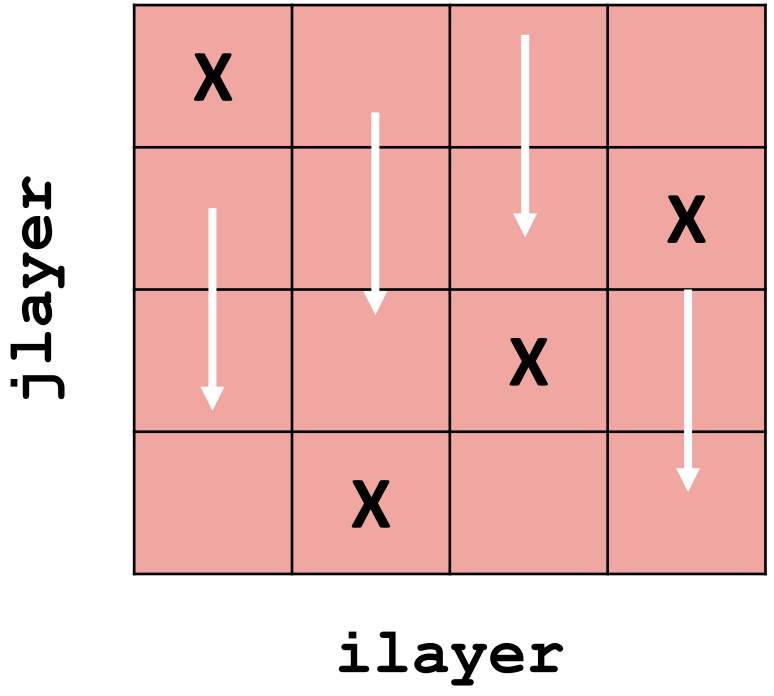
Loop in the FE banded_matrix_create handles the layer parallelism.

```
Loop over layer count
   Identify recv, send (j)layers
   Post receive if not self-layer
   Loop over bands for jlayer & compute integrands
   Post send if not self-layer
   Wait for recv if not self-layer
   Element gather for recv'd bands
   Wait for send completion (could be delayed?)
End loop
```

Layer loop in the FE banded_matrix_create coordinates communication and computation.



Integrand & send layer orderings.
Arrows show jlayer looping.



Receive layer orderings.
Arrows show jlayer looping.

Matrix integrand routines are formulated like “dot” routines.

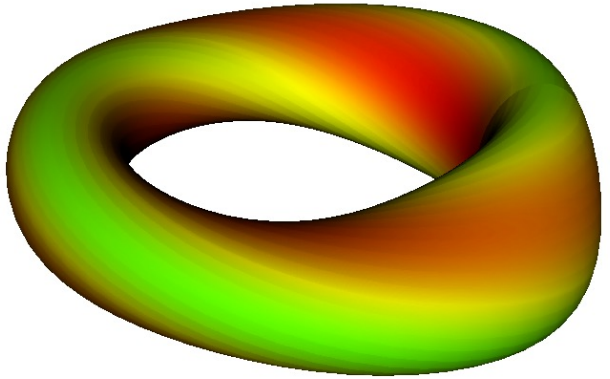
- Trigonometric basis functions are $2\cos(2\pi i_\zeta n/N_\zeta)$ and $-2\sin(2\pi i_\zeta n/N_\zeta)$ for real and imaginary parts.
- Range of n differs for each matrix band.
- Loops are over column-FE basis, column-trig basis, row-FE basis, row-trig basis, and toroidal grid (for explicit integration).
- Column basis functions are manipulated like the direction vectors in ”dot” integrand routines.
- Integrand array has cell index last; trig and vector indices are packed together.

The threed preconditioning routines require communication.

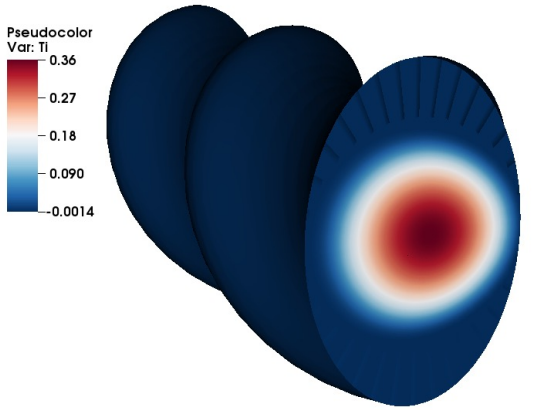
- Layer decomposition of Fourier components does not necessarily match the band decomposition.
  - Also, bands can have overlapped Fourier components.
- There is asynchronous layer communication for residuals.
- Complex data is packed into real-data structures.
- The banded preconditioner is applied.
- Real data is unpacked.
- Asynchronous communication sends data back to standard layers.

Computations on anisotropic thermal conduction compare performance when varying stride and overlap.

- Elements are bicubic. Fourier representation is $0 \leq n \leq 11$.
- Outer part of the mesh is a ring of 32x32 elements; 1216 total elements.
- Geometry is toroidal with a twisting elliptical cross section, $\mathrm{R/a} = 4$.
- Magnetic field is generated by a vector-potential solve before the timestep loop; toroidal current density is uniform.
- Computations have varied stride (“st”) and overlap (“ov”).
- Computations were run on 40 cores (4 layers) or 60 cores (6 layers) of Cori KNL. Some needed 2 nodes for memory.



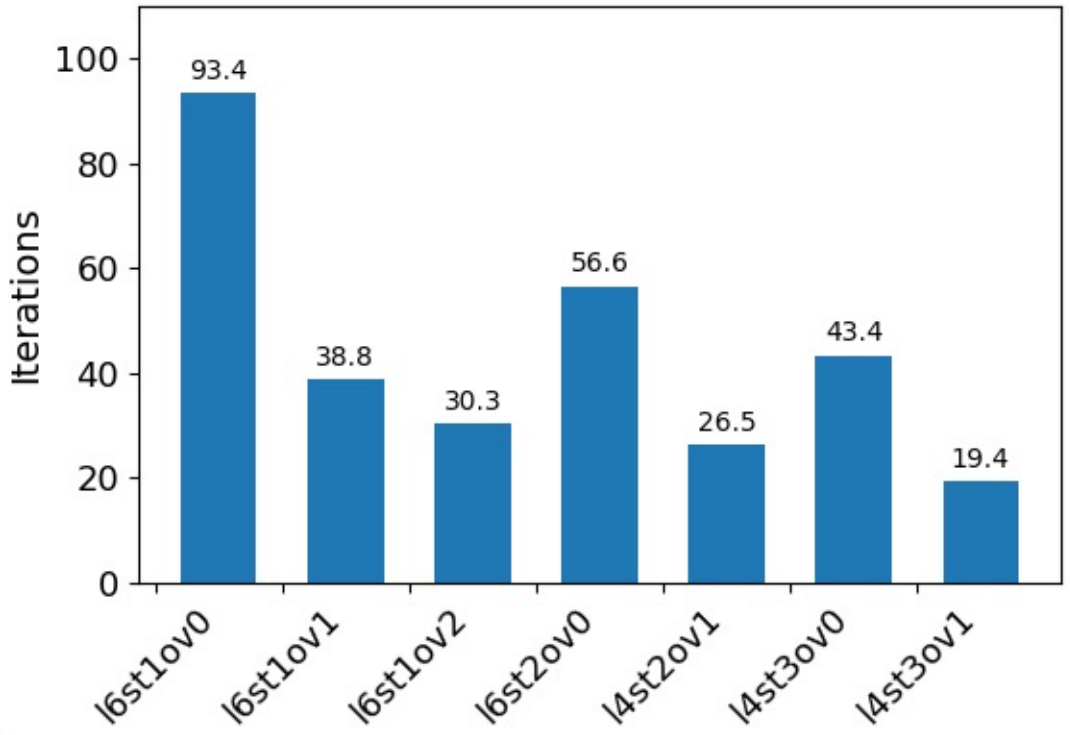
$|B_{pol}|$ highlights surface shaping.

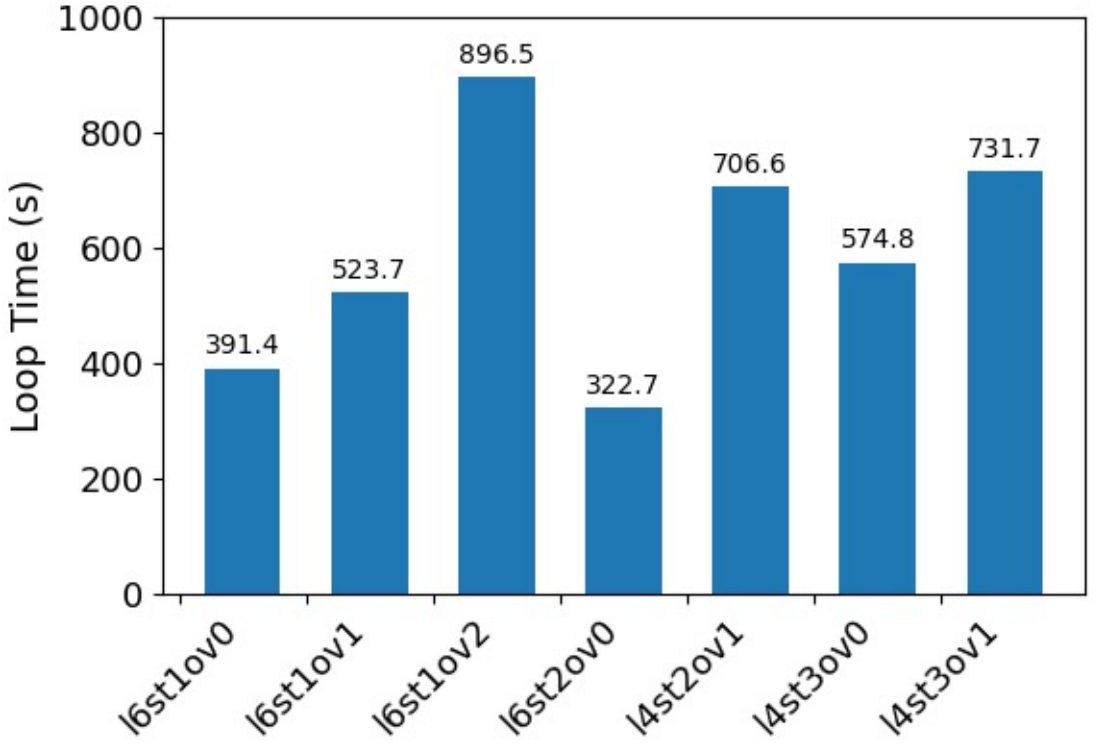


Temperature from thermal conduction with $\kappa_{\parallel}/\kappa_{\perp} = 10^6$ shows flux surfaces.

Iteration count depends on both stride and overlap.

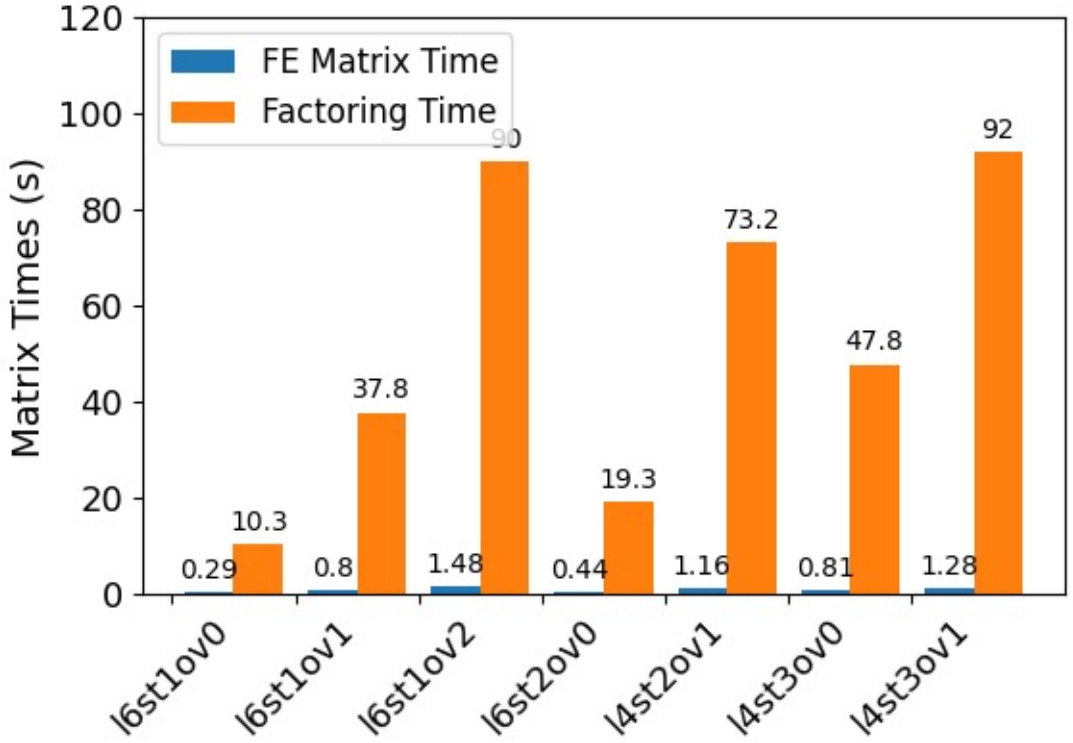
- Computations are run 50 steps with matrices being computed 4 times.

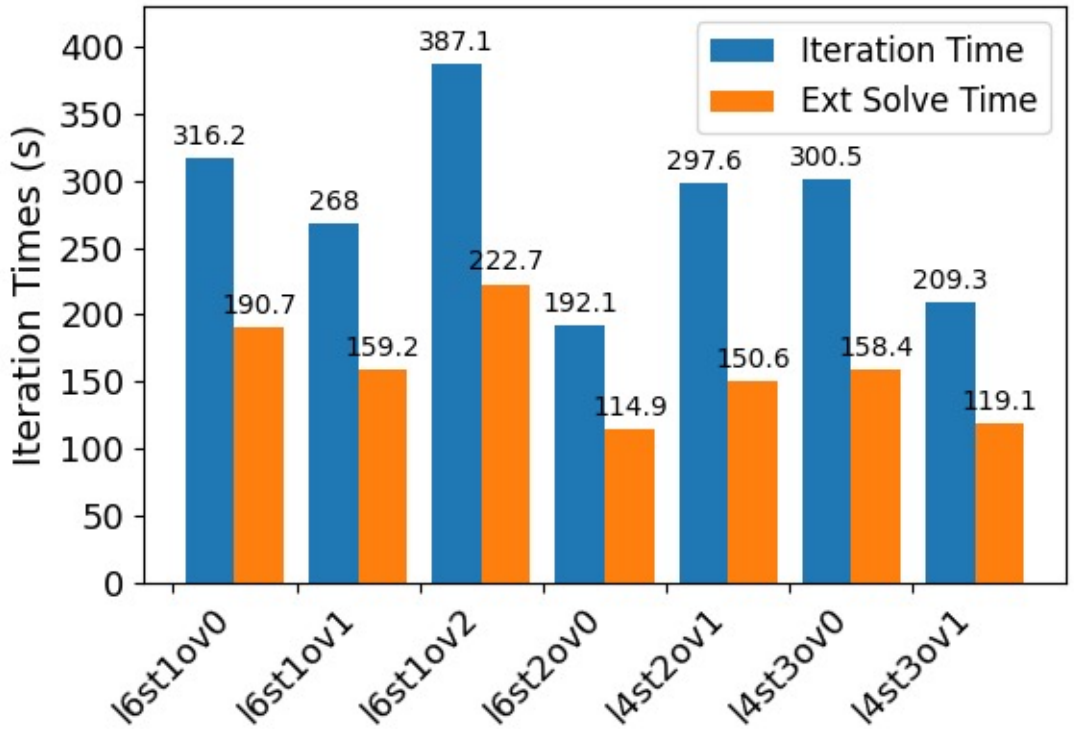




- Reducing iteration count does not equate to reduced run time.

Solve times improve with good choices of stride and overlap.





- The matrix-creation timing is not correct.
- Communication for preconditioning is not problematic.

Status

- The programming for the vector potential is nearing completion.
  - Packing Fourier components and different edge fields got complicated.
- A stride=2, overlap=2 computation produced a memory corruption error on Cori.
- Some aspects will be easier in NIMROD, but numerical summing over phi-grid will be necessary.
  - Computational machinery will be ported.
  - It will require new matrix integrand routines.

Conclusions and Discussion

- Direct solves over bands of Fourier components is possible.
  - Full matrix computation and factoring is tractable for bands of Fourier components.
  - Layer parallelization is functioning.
- Multi-harmonic bands reduces iteration, significantly in some cases.

- Bands could be composed of non-sequentially spaced harmonics.
- 3D static condensation with block-Fourier-diagonal is also worth trying.