Preconditioning and Scalability of Implicit Extended MHD Plasma Simulation by FETI-DP Domain Substructuring

Alan H. Glasser

Los Alamos
NATIONAL LABORATORY

Collaborators: Slava Lukin, Barry Smith

Center for Extended Macroscopic Modeling
Orlando, Florida, November 11, 2007

Scalability By Domain Decomposition

- 3D extended MHD modeling of magnetically confined fusion plasmas requires petascale computing: 1 petaflop = 10^{15} flops ~10^{5} procs.
- Efficient petascale computing requires scalable linear systems: solution time independent of grid size, number of processors.
- Current method: static condensation; not scalable.
- Domain decomposition is a promising approach to scalability.
  - Schwarz overlapping methods.
  - Non-overlapping methods, domain substructuring, *e.g.* FETI-DP.
- Analytical proofs of scalability for simple systems: Poisson, linear elasticity, Navier-Stokes.
- Empirical studies: using existing 2D SEL code for extended MHD.

Condition Number and Scalability

Condition Number

$$\mathbf{L}\mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad \kappa\left(\mathbf{L}\right) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Scalability

A matrix is scalable is its condition number, and hence the number of Krylov iterations to convergence, is independent of the number of subdomains.

Theorem

Scalability of dual matrix $\mathbf{F}$ has been proven analytically for a limited range of elliptic problems: Poisson, linear elasticity, Navier-Stokes. $\kappa(\mathbf{F}) = C\left[1 + \ln(H/h)\right]^{\gamma}$, $\gamma = 2$ or 3.

Conjecture

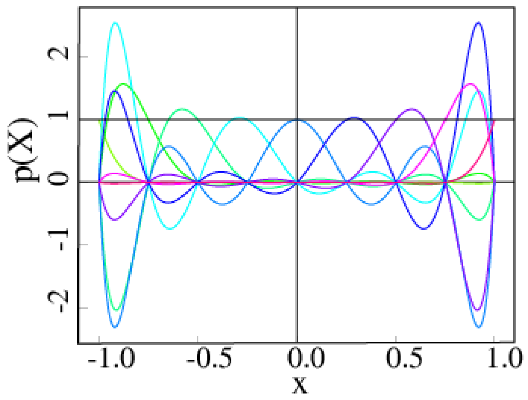
$\mathbf{F}$ is scalable for a broader range of problems, to be determined empirically.

Development Platform
SEL 2D Spectral Element Code

- Flux-source form: simple, general problem setup.
- Spatial discretization:
  - High-order C^0 spectral elements, modal basis
  - Harmonic grid generation, adaptation, alignment
- Time step: fully implicit, 2^{nd}-order accurate,
  - θ-scheme
  - BDF2
- Static condensation, Schur complement.
  - Small local direct solves for grid cell interiors.
  - Preconditioned GMRES for Schur complement.
- Distributed parallel operation with MPI and PETSc.

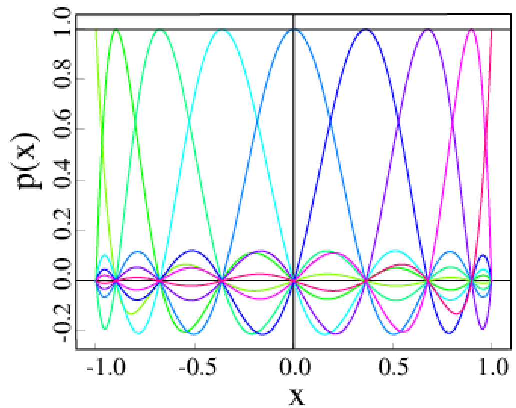
Alternative Polynomial Bases

Uniform Nodal Basis



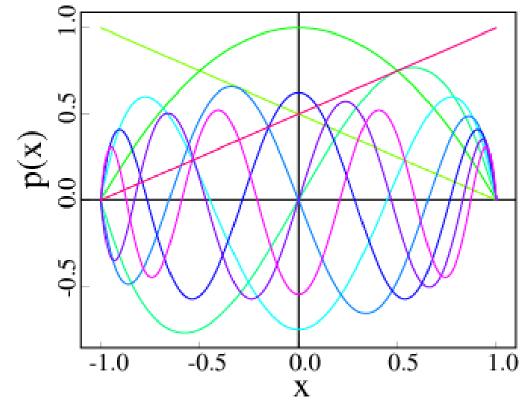
- Lagrange interpolatory polynomials
- Uniformly-spaced nodes
- Diagonally subdominant

Jacobi Nodal Basis



- Lagrange interpolatory polynomials
- Nodes at roots of *$(1-x^2)\, P_n^{(0,0)}(x)$*
- Diagonally dominant

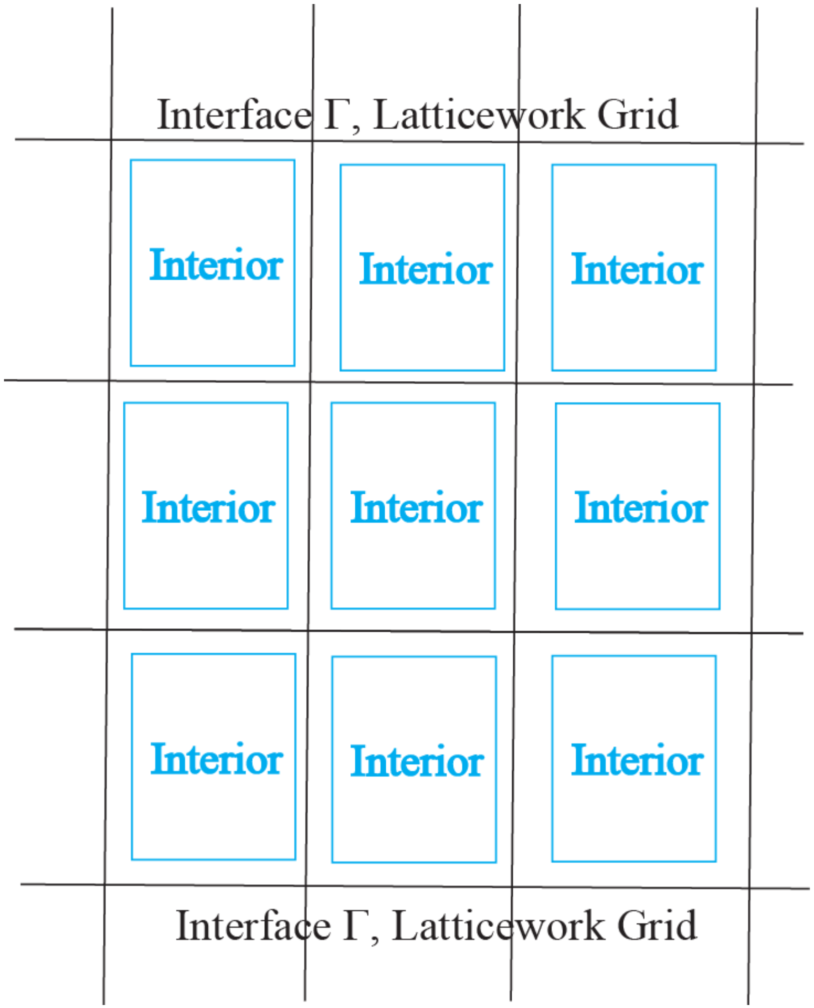
Spectral (Modal) Basis



- Jacobi polynomials *$(1+x)/2$, $(1-x)/2$, $(1-x^2)\, P_n^{(1,1)}(x)$*
- Nearly orthogonal
- Manifest exponential convergence

Static Condensation

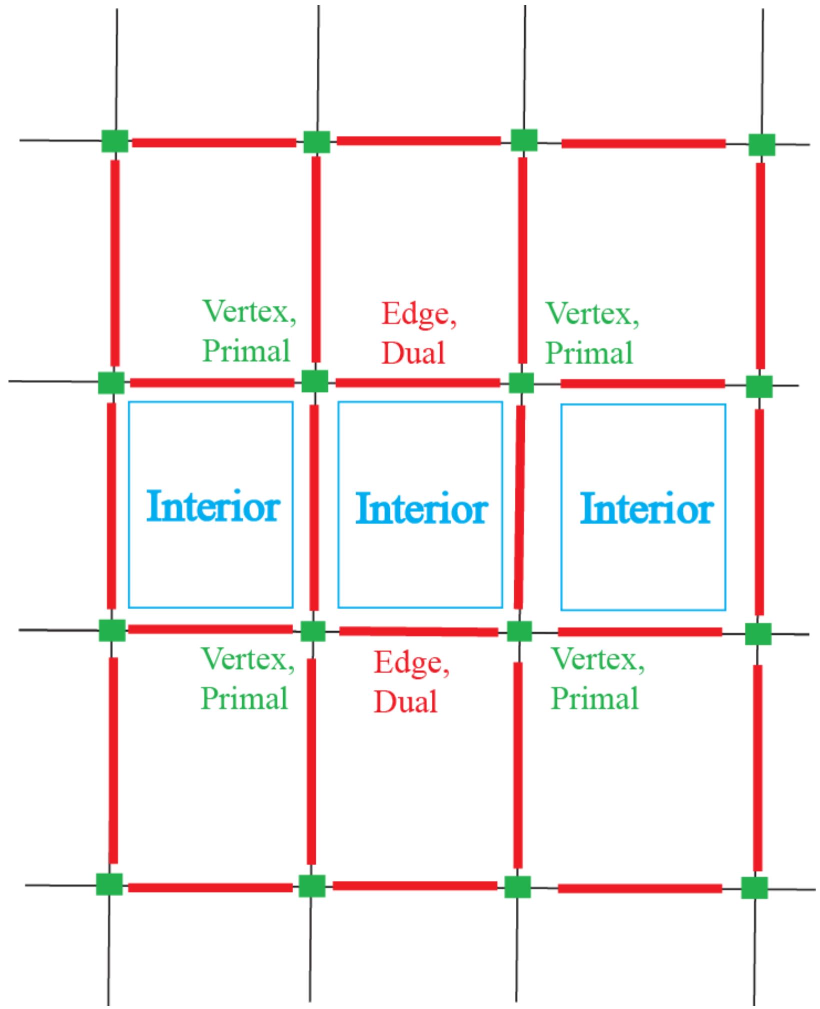
- Implicit time step requires linear system solution: **L u = r**.
- Direct solution time grows as n^3.
- Break up large matrix into smaller pieces: Interiors + Interface.
- Small direct solves for interior.
- Advanced parallel iterative solves for interface.
- Substantially reduces solution time.
- Not scalable to petaflop parallel computers; solution time grows with problem size.



FETI-DP

Finite Element Tearing and Interconnecting, Dual-Primal

- Break up large matrix into three pieces: interior + dual + primal.
- Small direct solves for interior.
- Parallel direct solve for primal points.
- Matrix-free preconditioned GMRES for dual points.
- Primal solve provides information to dual problem about coarse global conditions, providing scalability.
- Interior preconditioner accelerates convergence of dual solve.
- The iterative, parallel, dual problem is scalable!



FETI-DP

Finite Element Tearing and Interconnecting, Dual-Primal
Domain decomposition, non-overlapping, Schur complement

Axel Klawonn and Olof B. Widlund,
"Dual-Primal FETI Methods for Linear Elasticity,"
Comm. Pure Appl. Math. **59**, 1523-1572 (2006).

Partition

- I: Interior points, inside each subdomain (grid cell) Ω_i.
- Δ: Dual interface points, continuity imposed by Lagrange multipliers.
- Π: Primal interface points, continuity imposed directly.

Initial Block Matrix Form

$$\mathsf{L}\mathbf{u} = \mathbf{r}, \quad \mathsf{L} = \begin{pmatrix} \mathsf{L}_{II} & \mathsf{L}_{I\Delta} & \mathsf{L}_{I\Pi} \\ \mathsf{L}_{\Delta I} & \mathsf{L}_{\Delta\Delta} & \mathsf{L}_{\Delta\Pi} \\ \mathsf{L}_{\Pi I} & \mathsf{L}_{\Pi\Delta} & \mathsf{L}_{\Pi\Pi} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Delta \\ \mathbf{u}_\Pi \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} \mathbf{r}_I \\ \mathbf{r}_\Delta \\ \mathbf{r}_\Pi \end{pmatrix}$$

Algebraic Reorganization

Local Block Matrices: $\mathbf{I} + \Delta$

$$\mathsf{L}_{BB} = \begin{pmatrix} \mathsf{L}_{II} & \mathsf{L}_{I\Delta} \\ \mathsf{L}_{\Delta I} & \mathsf{L}_{\Delta\Delta} \end{pmatrix}, \quad \mathbf{u}_B = \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Delta \end{pmatrix}, \quad \mathbf{r}_B = \begin{pmatrix} \mathbf{r}_I \\ \mathbf{r}_\Delta \end{pmatrix}$$

Dual Continuity: Lagrange Multipliers

λ is a vector of Lagrange multipliers used to impose continuity on the dual dependent variables $\mathbf{u}_\Delta$.

$$\mathsf{B} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{B}_\Delta \end{pmatrix}, \quad \mathsf{B}_\Delta \mathbf{u}_\Delta = 0, \quad \mathsf{L}_{BB}\mathbf{u}_B + \mathsf{L}_{B\Pi}\mathbf{u}_\Pi + \mathsf{B}^T\lambda = \mathbf{r}_B$$

Final Block Matrix Form

$$\mathsf{L} = \begin{pmatrix} \mathsf{L}_{BB} & \mathsf{L}_{B\Pi} & \mathsf{B}^T \\ \mathsf{L}_{\Pi B} & \mathsf{L}_{\Pi\Pi} & \mathbf{0} \\ \mathsf{B} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_B \\ \mathbf{u}_\Pi \\ \lambda \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} \mathbf{r}_B \\ \mathbf{r}_\Pi \\ \mathbf{0} \end{pmatrix}$$

Solution and Reduction

Solutions for $\mathbf{u}_B$ and $\mathbf{u}_\Pi$

$$\mathbf{u}_B = \mathsf{L}_{BB}^{-1}\left(\mathbf{r}_B - \mathsf{L}_{B\Pi}\mathbf{u}_\Pi - \mathsf{B}^T\lambda\right)$$

$$\mathsf{S}_{\Pi\Pi} \equiv \mathsf{L}_{\Pi\Pi} - \mathsf{L}_{\Pi B}\mathsf{L}_{BB}^{-1}\mathsf{L}_{B\Pi}$$

$$\mathbf{u}_\Pi = \mathsf{S}_{\Pi\Pi}^{-1}\left[\mathbf{r}_\Pi - \mathsf{L}_{\Pi B}\mathsf{L}_{BB}^{-1}\left(\mathbf{r}_B - \mathsf{B}^T\lambda\right)\right]$$

Global Schur Complement Equation for λ

$$\mathsf{F}\lambda = \mathbf{d}$$

$$\mathsf{F} = \mathsf{B}\left(\mathsf{L}_{BB}^{-1} + \mathsf{L}_{BB}^{-1}\mathsf{L}_{B\Pi}\mathsf{S}_{\Pi\Pi}^{-1}\mathsf{L}_{\Pi B}\mathsf{L}_{BB}^{-1}\right)\mathsf{B}^T$$

$$\mathbf{d} = \mathsf{B}\mathsf{L}_{BB}^{-1}\left[\mathbf{r}_B - \mathsf{L}_{B\Pi}\mathsf{S}_{\Pi\Pi}^{-1}\left(\mathbf{r}_\Pi - \mathsf{L}_{\Pi B}\mathsf{L}_{BB}^{-1}\mathbf{r}_B\right)\right]$$

Algorithms: Example

Schur Complement Matrix-Vector Product $\mathsf{F}\lambda$, Eqs. (8) and (9)

1. Compute $\mathbf{u}_B = \mathbf{B}^T\lambda$, using restriction $\mathbf{R}_D$ and sign σ.
2. Compute $\mathbf{u}_B = \mathbf{L}_{BB}^{-1}\mathbf{u}_B$, using LAPACK.
3. Copy $\mathbf{v}_B = \mathbf{u}_B$.
4. Compute $\mathbf{u}_{\Pi l} = \mathbf{L}_{\Pi B}\mathbf{u}_B$.
5. Assemble $\mathbf{u}_{\Pi l}$ into $\mathbf{u}_{\Pi g}$, using extension $\mathbf{R}_{\Pi}^T$.
6. Compute $\mathbf{u}_{\Pi g} = \mathbf{S}_{\Pi\Pi}^{-1}\mathbf{u}_{\Pi g}$, using SuperLU_DIST.
7. Disassemble $\mathbf{u}_{\Pi g}$ into $\mathbf{u}_{\Pi l}$, using restriction $\mathbf{R}_{\Pi}$.
8. Compute $\mathbf{u}_B = \mathbf{L}_{B\Pi}\mathbf{u}_{\Pi l}$.
9. Compute $\mathbf{u}_B = \mathbf{L}_{BB}^{-1}\mathbf{u}_B$, using LAPACK.
10. Add $\mathbf{u}_B = \mathbf{u}_B + \mathbf{v}_B$.
11. Compute $\mathsf{F}\lambda = \mathbf{B}\mathbf{u}_B$, using extension $\mathbf{R}_D^T$ and sign σ.

Cell-Wise Preconditioning

Definitions For Each Subdomain Ω_i

$$\mathbf{B}_{D,\Delta}^{(i)} \equiv \text{ scaled jump matrix}$$

$$\mathbf{R}_{\Gamma\Delta}^{(i)} \equiv \text{ restriction matrix from full interface to dual variables}$$

$$\mathbf{S}_{\varepsilon}^{(i)} \equiv \text{ Schur complement obtained by eliminating interior variables}$$

Preconditioner

$$\mathbf{M}^{-1} = \sum_{i=1}^{n} \mathbf{B}_{D,\Delta}^{(i)} \mathbf{R}_{\Gamma\Delta}^{(i)} \mathbf{S}_{\varepsilon}^{(i)} \mathbf{R}_{\Gamma\Delta}^{(i)T} \mathbf{B}_{D,\Delta}^{(i)T}, \quad \mathbf{M}^{-1}\mathbf{F}\lambda = \mathbf{M}^{-1}\mathbf{d}$$

Solution Strategy

- Small dense block matrices of $\mathbf{L}_{BB}$ solved locally by LAPACK.
- Sparse global, primal matrix $\mathbf{S}_{\Pi\Pi}$ solved by SuperLU.
  - Short-term: redundant on all processors.
  - Medium-term: distributed SuperLU.
  - Long-term: ILU*n*-preconditioned GMRES.
- Global Schur complement matrix $\mathbf{F}$ solved by matrix-free parallel preconditioned GMRES.
- Choose primal interface constraints to provide coarse global problem, ensure scalability. 2D: vertices. 3D: more complicated.
- The scalability of $\mathbf{F}$ is accomplished by the coarse, primal solver. The quality of the preconditioner determines the rate of convergence but not the scalability.

Research Program

- Use existing 2D SEL spectral element code as test bed.
- Implement FETI-DP as a modification of existing static condensation routines.
- Study a progression of extended MHD systems as *nx*, *ny*, and nproc are increased to determine constancy of:
  - condition number
  - Krylov iterations to convergence
  - cpu time per processor
- Extend spectral element code to 3D.
- Investigate optimal choice of primal constraints for scalability.

Test Equation
Anisotropic Convection-Diffusion

Scalar PDE

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \nabla \cdot (\mathbf{D} \cdot \nabla u) + S$$

Velocity Vector and Diffusion Tensor

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}$$

$$D_{11} = d_0, \quad D_{22} = 1/d_0, \quad D_{12} = d_1 + d_2, D_{21} = d_1 - d_2$$

$$\det \mathbf{D} = 1 + d_2^2 - d_1^2$$

Source Term

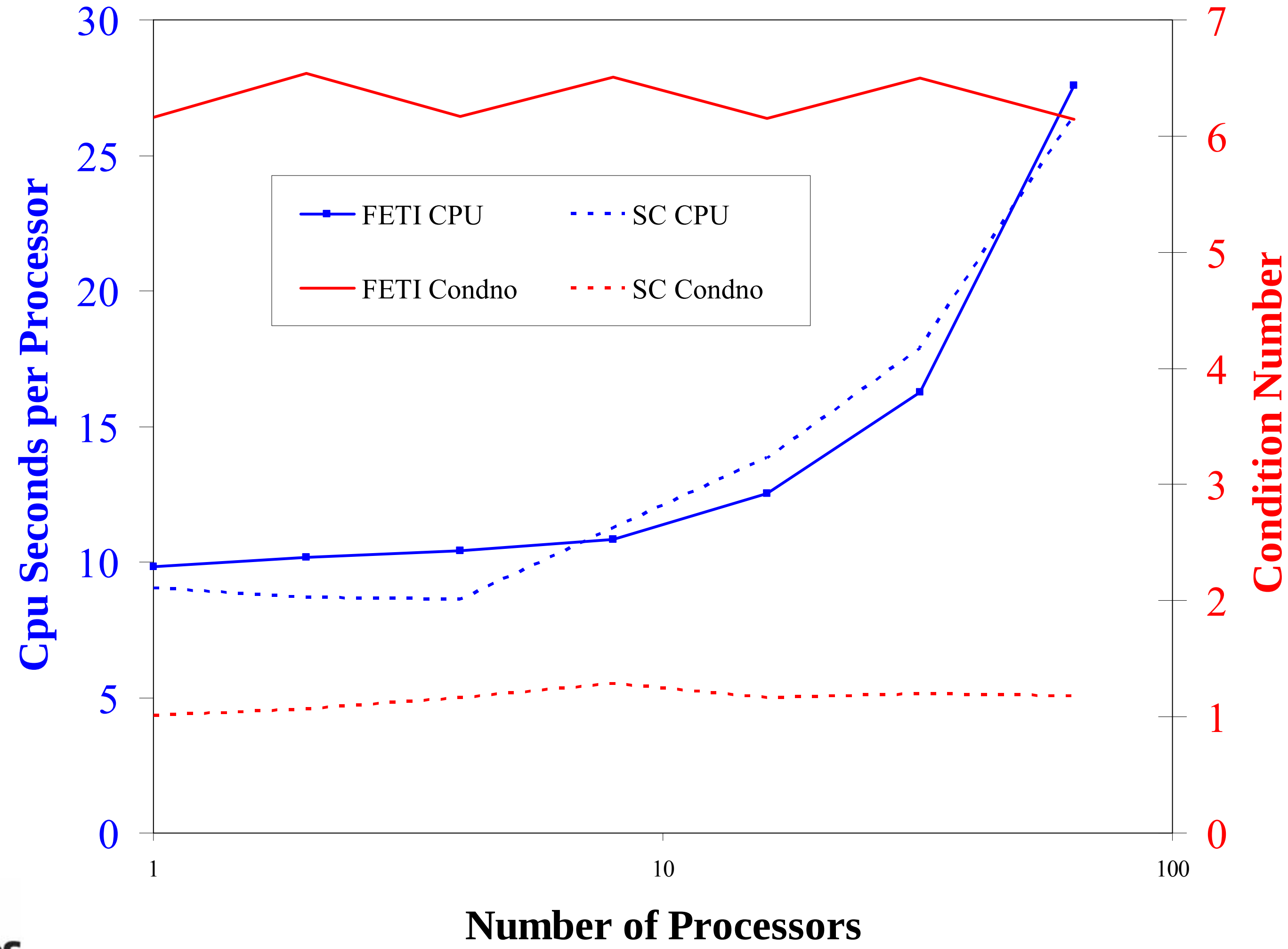
$$S(x, y) = \sin(m\pi x) \sin(n\pi y)$$

Initial and Boundary Conditions

$$u(x, y, t) = 0 \text{ at } x = 0, 1; \quad y = 0, 1; \quad t = 0$$

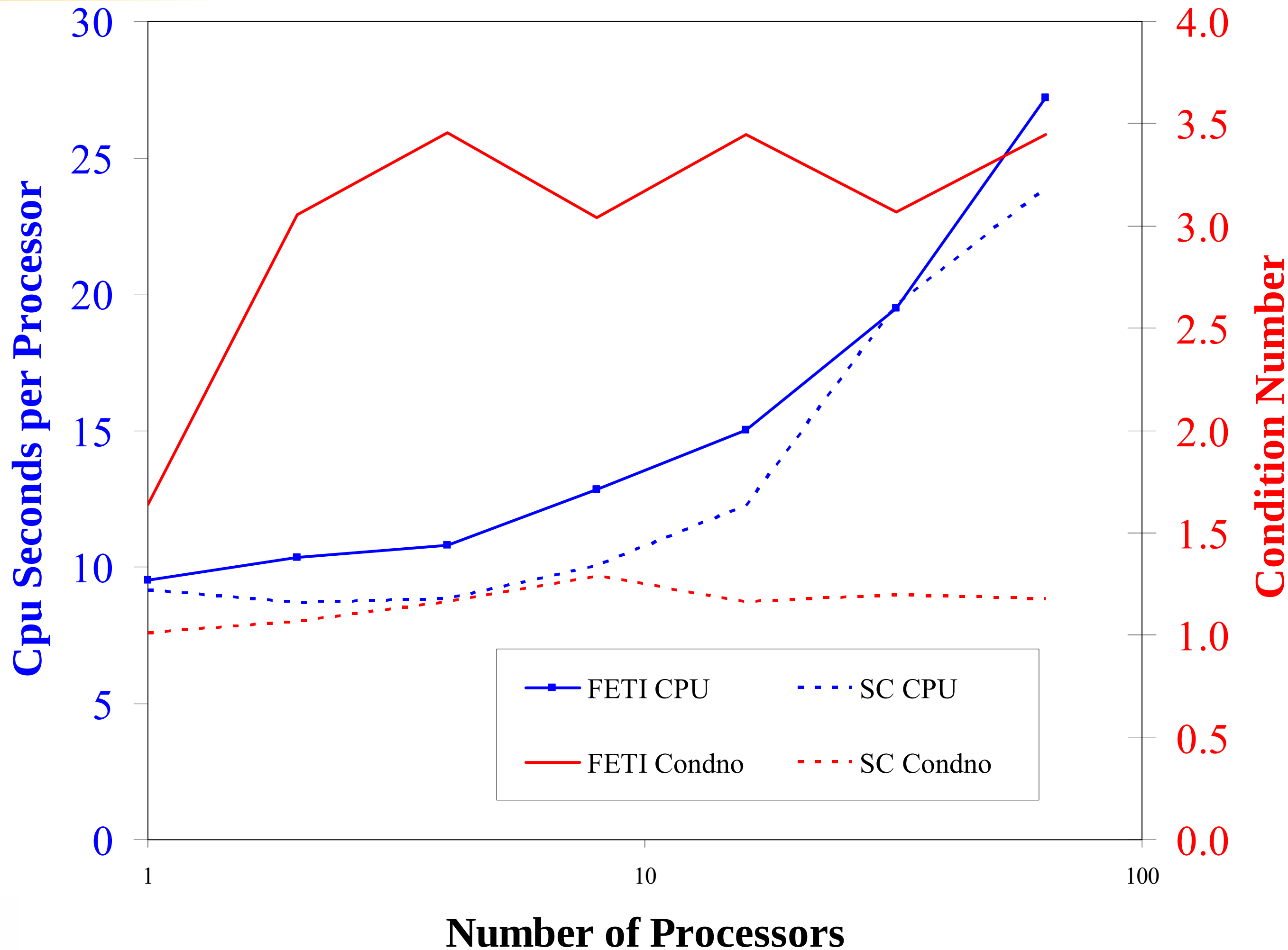
Parallel Scaling Tests

Isotropic Heat Equation



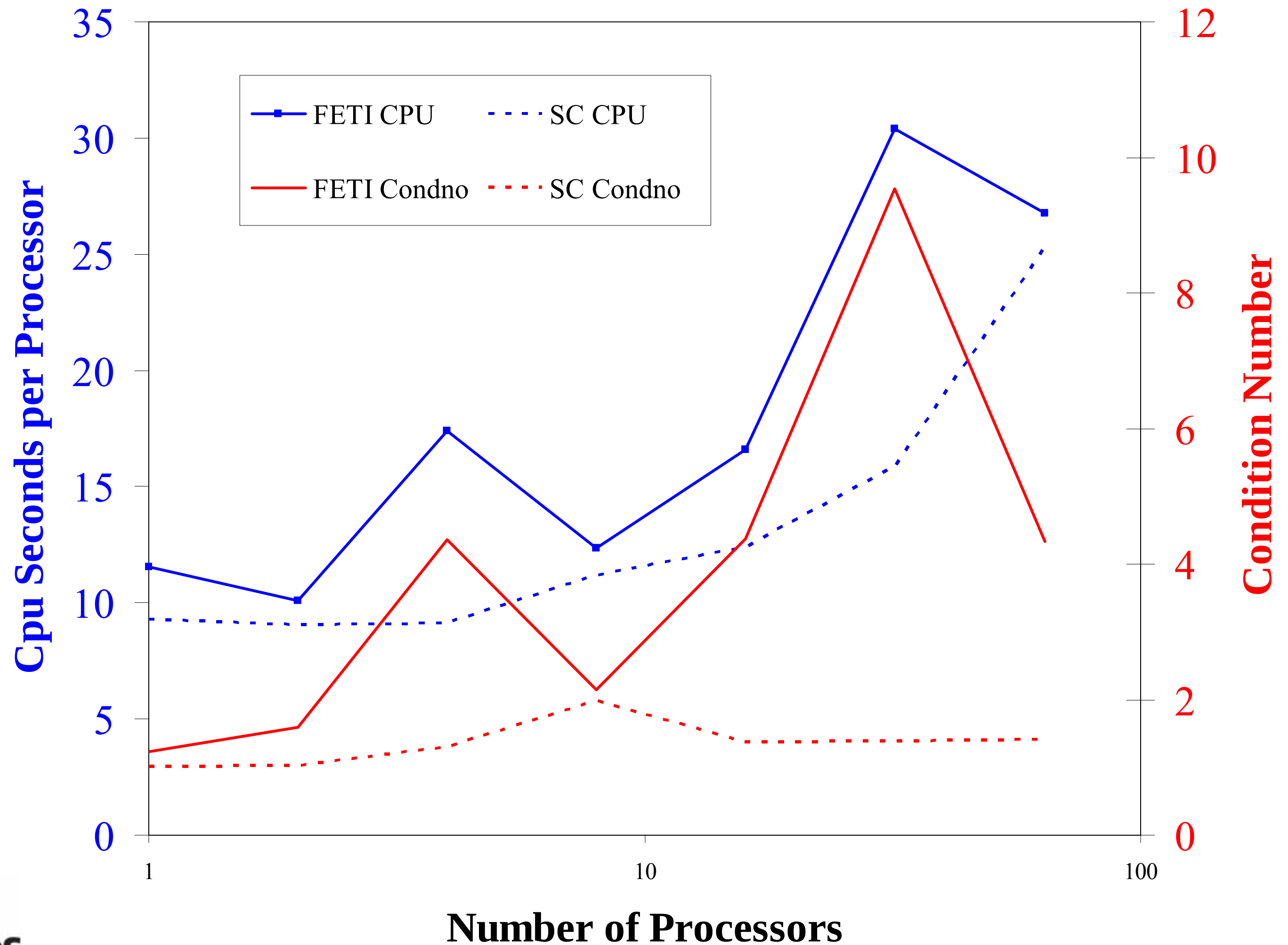
Parallel Scaling Tests

Anisotropic Heat Equation



Parallel Scaling Tests

Anisotropic Convection-Diffusion Equation



Status of Development of FETI-DP: Past Work

- Literature study, equations derived, algorithms formulated.
- Code written, compiled, debugged: ~2500 new lines of Fortran 95.
- Extensive testing and verification of method on single real processor.
- Operation on parallel Linux clusters; bassi.nersc.gov.
- Scaling tests, comparison to older method of static condensation.
- Profiling and optimization: understand where time is going, tune algorithm.
- Scalability appears to extend to asymmetric problem, convection-diffusion equation, condition number doesn't scale increase with problem size.
- Problem: waves, hyperbolic problems.

Status of Development of FETI-DP: Future Work

- Treatment of hyperbolic problems.
- Physics-based preconditioning applied at the level of vectors and matrices. Preserve flux-source form, adapt FETI-DP implementation. Luis Chacon.
- Apply to extended MHD, 2D reconnection.
- Extension to C^1 triangles, incorporation into M3D.
- Extension to 3D; optimal choice of primal constraints.