Preconditioning a Parallel, Inexact Block-Jacobi Splitting of the S_N Algorithms

Jim Warsa, Kelly Thompson, Jim Morel, Jae Chang, and Kent Budge
Computer and Computational Sciences Division
Los Alamos National Laboratory
And
Michele Benzi
Department of Mathematics and Computer Sciences
Emory University

Outline

- Introduction
- Problem formulation
- TSA
- Eigenvalue spectra
- Parallel performance and scaling

Introduction

- Parallel block-Jacobi S_N
  - Spatial domain decomposition
  - Finding efficient parallel sweep schedules for unstructured grids is difficult.
  - Parallel block-Jacobi algorithm [Yavuz and Larsen, 1989 and 1992]
  - Sub-domain interface angular fluxes are iterative unknowns.
  - Convergence degrades with increasing processors (N_p iterations needed in voids).
- Krylov methods
  - Can a Krylov iterative method improve efficiency?
  - Can Transport Synthetic Acceleration (TSA) be used as an effective preconditioner?

Problem Formulation

- Time-independent S_N equations
  - Steady-state, isotropic scattering
  - Quadrature $\{\mu_m, w_m\}, m = 1, \ldots, N_A$
  - Linear discontinuous finite element spatial discretization on cells

    $j = 1, \ldots, N_c$

$$\underbrace{\left(\hat{\Omega}_m \bullet \nabla + \sigma_t(r)\right)\psi_m(r)}_{L\Psi} = \underbrace{\frac{1}{4\pi}\sigma_s(r)\phi(r)}_{MS\Phi} + \underbrace{q_m(r)}_{Q}, \quad \phi(r) = \underbrace{\sum_{m=1}^{N_A} w_m \psi_m(r)}_{D\Psi}$$

Problem Formulation

- Operator Notation
  - Define the vector $\Psi \in \mathbb{R}^{2N_c N_A}$, $\Phi \in \mathbb{R}^{2N_c}$

$$L\Psi = MS\Phi + Q$$

$$\Phi = D\Psi$$

$$\Phi = DL^{-1}MS\Phi + DL^{-1}Q$$

$$\left(I - DL^{-1}MS\right)\Phi = DL^{-1}Q$$

Problem Formulation

- Parallel block-Jacobi iteration
  - Spatially decomposed problems
  - Inverting L on every sub-domain p independently
  - Incoming boundary angular fluxes are specified from previous iteration or initial guess.

$$\Psi_p = L_p^{-1}\left(M_p S_p \Phi_p + \sum_{p'} L_{p,p'} \Psi_{p'} + Q_p \right)$$

$$\Phi_p = D_p \Psi_p$$

- L_p interior angular fluxes
- $L_{p,p'}$ couples sub-domain p to adjacent sub-domain p'.
- Scalar fluxes and boundary angular fluxes are iterative unknowns.

Problem Formulation – Parallel Implementation

- Only outgoing angular fluxes on sub-domain boundaries are transferred.
- Define $\Psi_{p,p'} = W^T_{p,p'} \Psi_p$ for sub-domain p' adjacent to p
- $W^T_{p,p'}$ extracts the outgoing angular fluxes on the boundary from Ψ_p.
- $W_{p,p'}$ maps back onto the full angular flux vector.
- Write the iteration in terms of an “augmented vector”

$$\begin{bmatrix} \Phi_p \\ \Psi_{p,p'} \end{bmatrix}$$

Problem Formulation

- Source iteration

$$\begin{bmatrix} \Phi_p \\ \Psi_{p,p'} \end{bmatrix}^{l+1} = \left(\begin{bmatrix} D_p \\ W^T_{p,p'} \end{bmatrix} L_p^{-1} \left[M_p S_p \left(-\sum_{p'} L_{p,p'} W_{p,p'} \right) \right] \right) \begin{bmatrix} \Phi_p \\ \Psi_{p,p'} \end{bmatrix}^{l} + \begin{bmatrix} D_p \\ 0_{p,p'} \end{bmatrix} L_p^{-1} Q_p$$

- Krylov iterative method

$$\underbrace{\left(I - \begin{bmatrix} D_p \\ W^T_{p,p'} \end{bmatrix} L_p^{-1} \left[M_p S_p \left(-\sum_{p'} L_{p,p'} W_{p,p'} \right) \right] \right)}_{A} \underbrace{\begin{bmatrix} \Phi_p \\ \Psi_{p,p'} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} D_p \\ 0_{p,p'} \end{bmatrix} L_p^{-1} Q_p}_{b}$$

Transport Synthetic Acceleration

- Source iteration is Richardson iteration.

$$\Phi^{l+1} = DL^{-1}MS\Phi^{l} + DL^{-1}Q$$

- Preconditioning the Richardson iteration to improve convergence. This equivalent to synthetic acceleration.

$$\Phi^{l+1} = (I + PS)DL^{-1}MS\Phi^{l} + (I + PS)DL^{-1}Q$$

$$P = URV,\ R \approx \left(L - MSD\right)^{-1}$$

- U and V are appropriate projection and interpolation operators.

Transport Synthetic Acceleration

- For Diffusion Synthetic Acceleration (DSA), R is calculated by solving a diffusion equation.
- For Transport Synthetic Acceleration (TSA), R is simply a low-order transport equation.

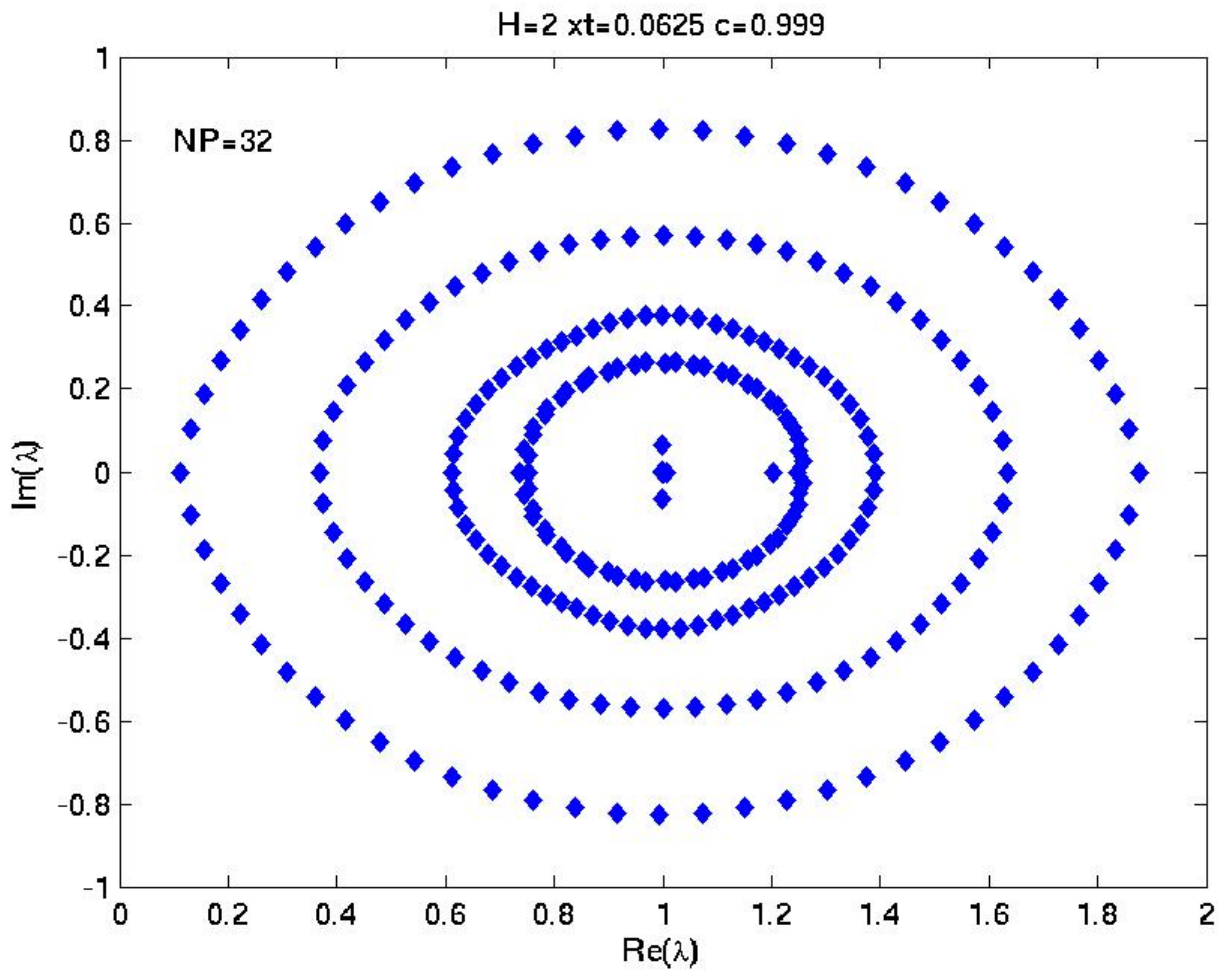
$$\left(\hat{\Omega}_m \bullet \nabla + \sigma_t'(r)\right)\psi_m'(r) = \frac{1}{4\pi}\sigma_s'(r)\phi'(r) + q_m'(r)$$

- In traditional TSA: $\sigma_s' = (1-\beta)\sigma_s$ and $\sigma_t' = (\sigma_t - \beta\sigma_s)$
  - $\beta = [0,1]$
  - L and S are modified in the low-order equations to satisfy same conservation properties as the high-order problem.
  - Typically, low-order equations use lower order quadratures and less scattering.

Transport Synthetic Acceleration

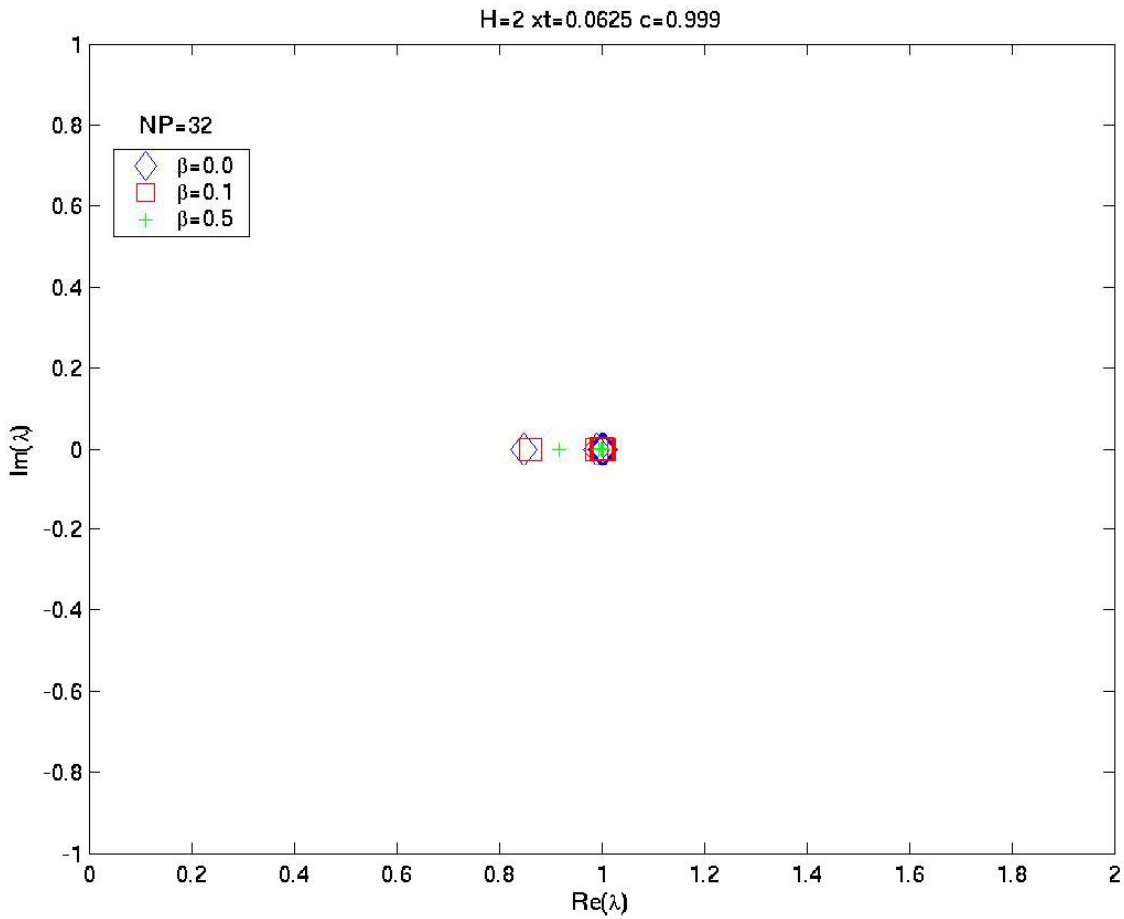
- Modified TSA: $\sigma_s' = \beta\sigma_s \quad \sigma_t' = \sigma_t$
  - σ_t remains the same and S is reduced in the low-order equations.
  - Always convergent
  - Simpler implementation within the time and energy dependent calculations
  - Can also use lower order quadratures

Eigenvalue Spectra for Source Iteration



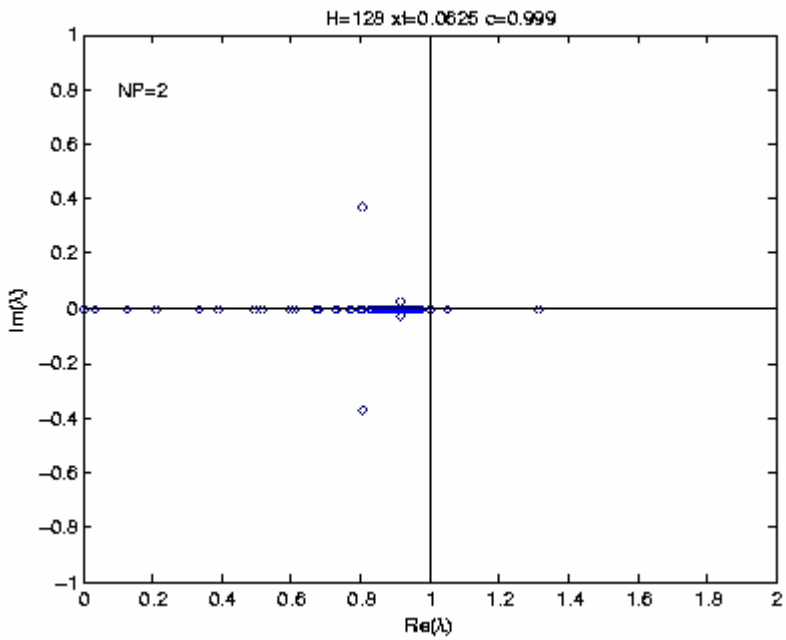
Los Alamos NATIONAL LABORATORY EST. 1943
The World's Greatest Science Protecting America
NNSA

Eigenvalue Spectra for Modified TSA

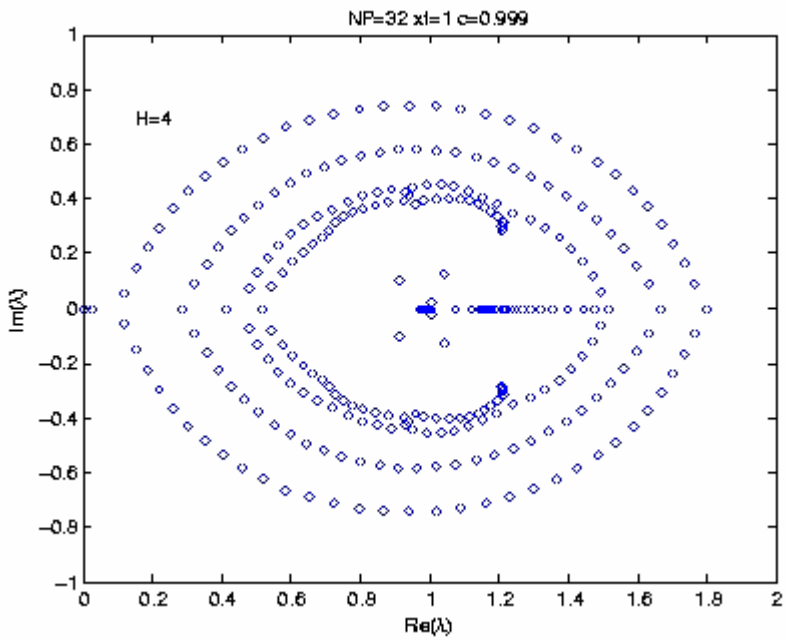


Los Alamos NATIONAL LABORATORY EST. 1943
The World's Greatest Science Protecting America
NNSA

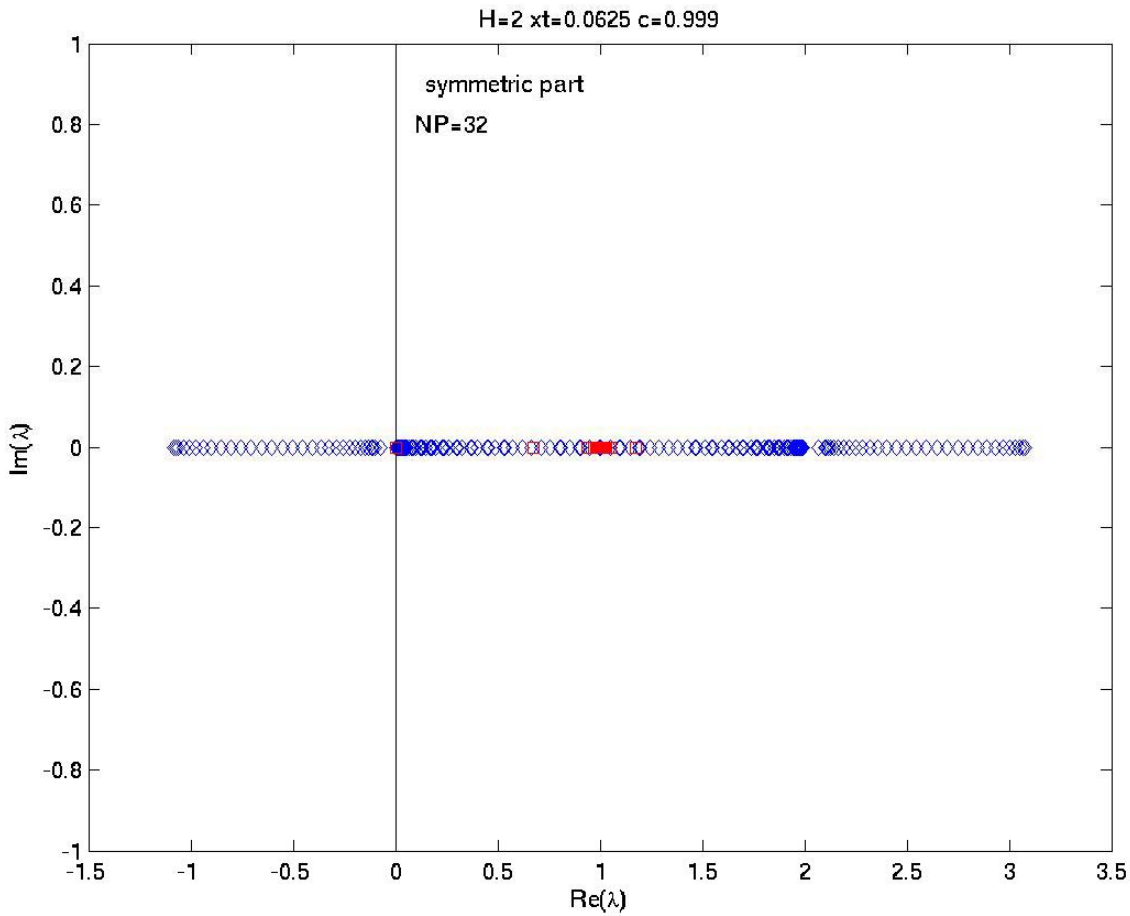
Eigenvalue Spectra for a Fixed Physical Problem (H constant) as a Function of Number of Processors



Eigenvalue Spectra for a Fixed Number of Processors as a Function of Thickness



Symmetric Part of A



Linear Algebra Observations

- A is non-normal
- For $\left(I - DL_0^{-1}L_b\right)\Phi = DL_0^{-1}Q$
  - d = degree of minimal polynomial, makespan of the subdomain graph
    = N_p for 1D

    = $2\sqrt{N_p}$ for 2D

    = $3\sqrt[3]{N_p}$ for 3D
- For GMRES(m), when $m \geq d$, it converges, otherwise, stagnation is possible since A is indefinite.

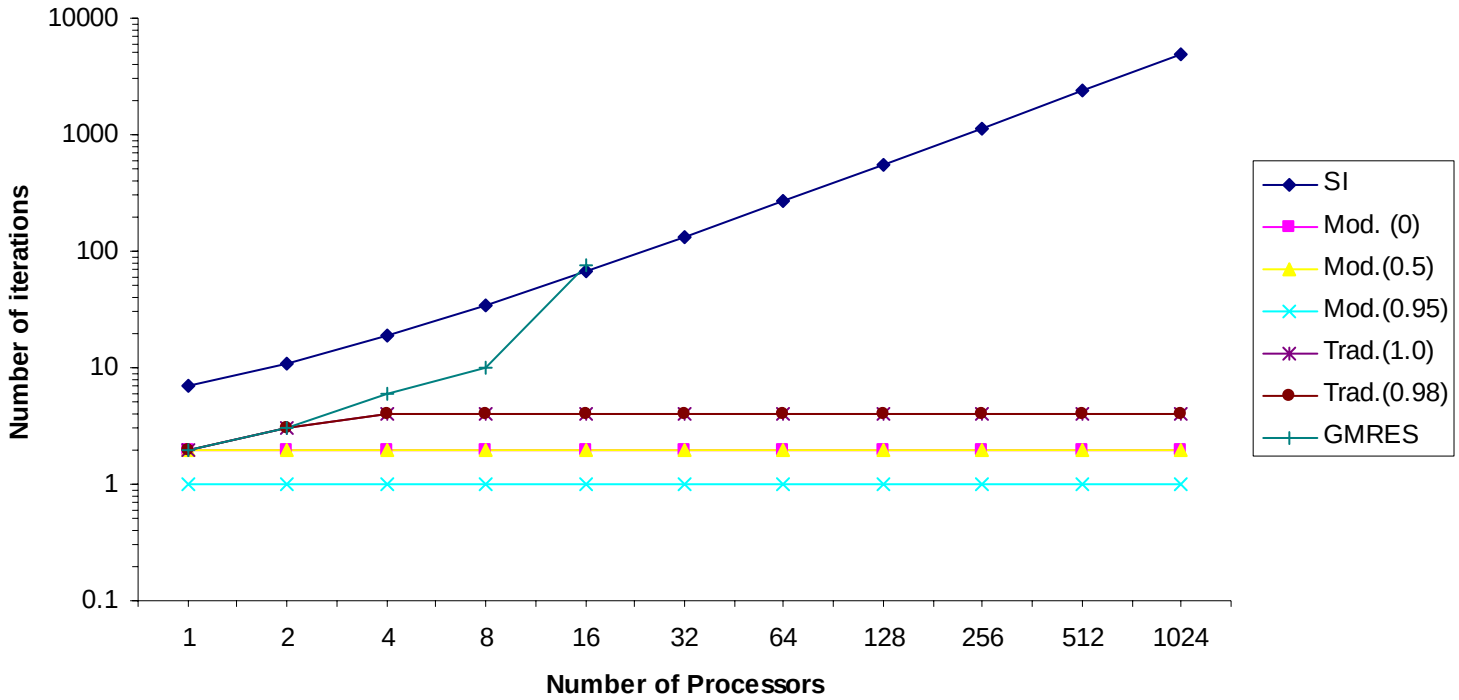
Parallel Performance and Scaling

- Problem Characteristics
  - S_8 quadrature
  - Fixed problem width H = 2 cm (equal-width mesh spacing)
  - Constant isotropic distributed source $q = 1$
  - Constant material properties $\sigma_t = 1/16 \text{ cm}^{-1}, \; c = 0.999$
  - Vacuum boundary conditions
  - Relative convergence criteria: 10^{-5} (high-order) and 10^{-7} (low-order)
  - Compare GMRES(15), source iteration, traditional TSA (β=1.0 and β=0.9), and modified TSA (β=0.0, β=0.5 and β=0.95)

Parallel Performance and Scaling

Weak Scaling

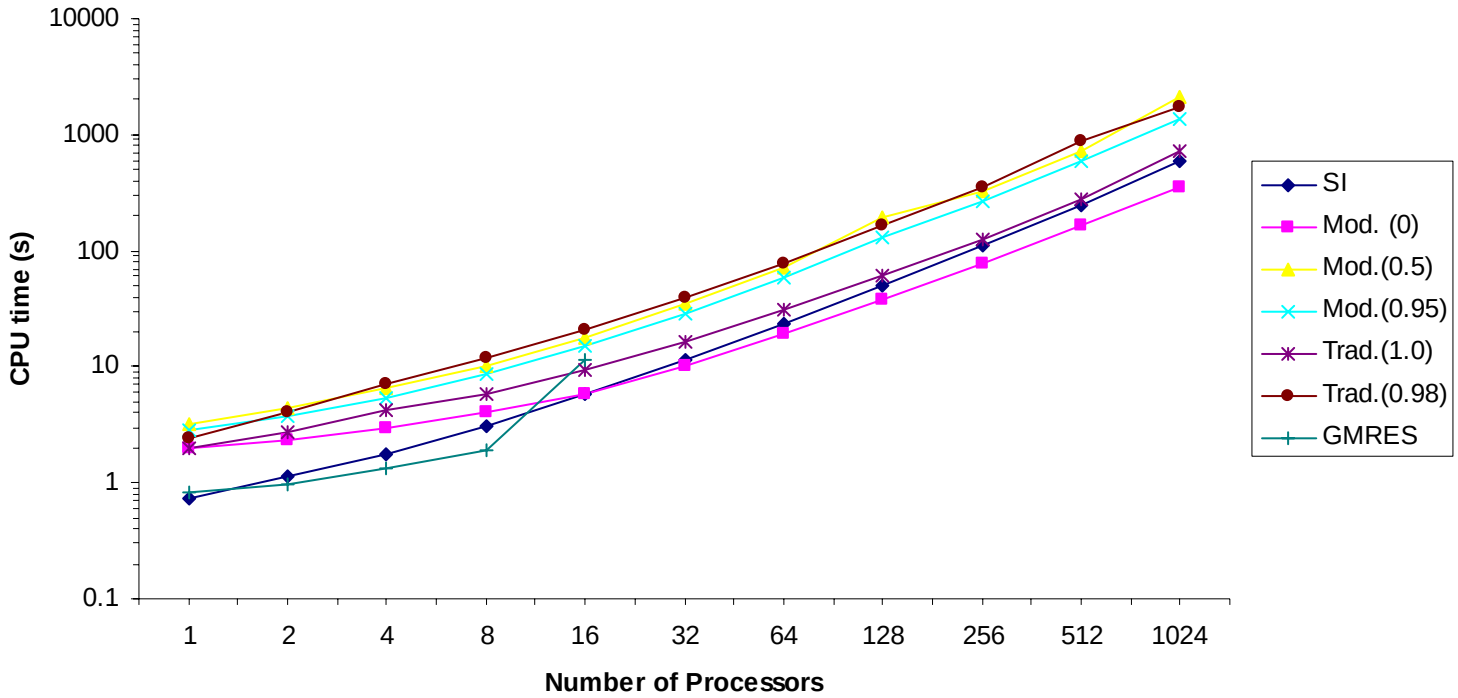
- Fixed work per processor: n = 65536 x N_p



Parallel Performance and Scaling

Weak Scaling

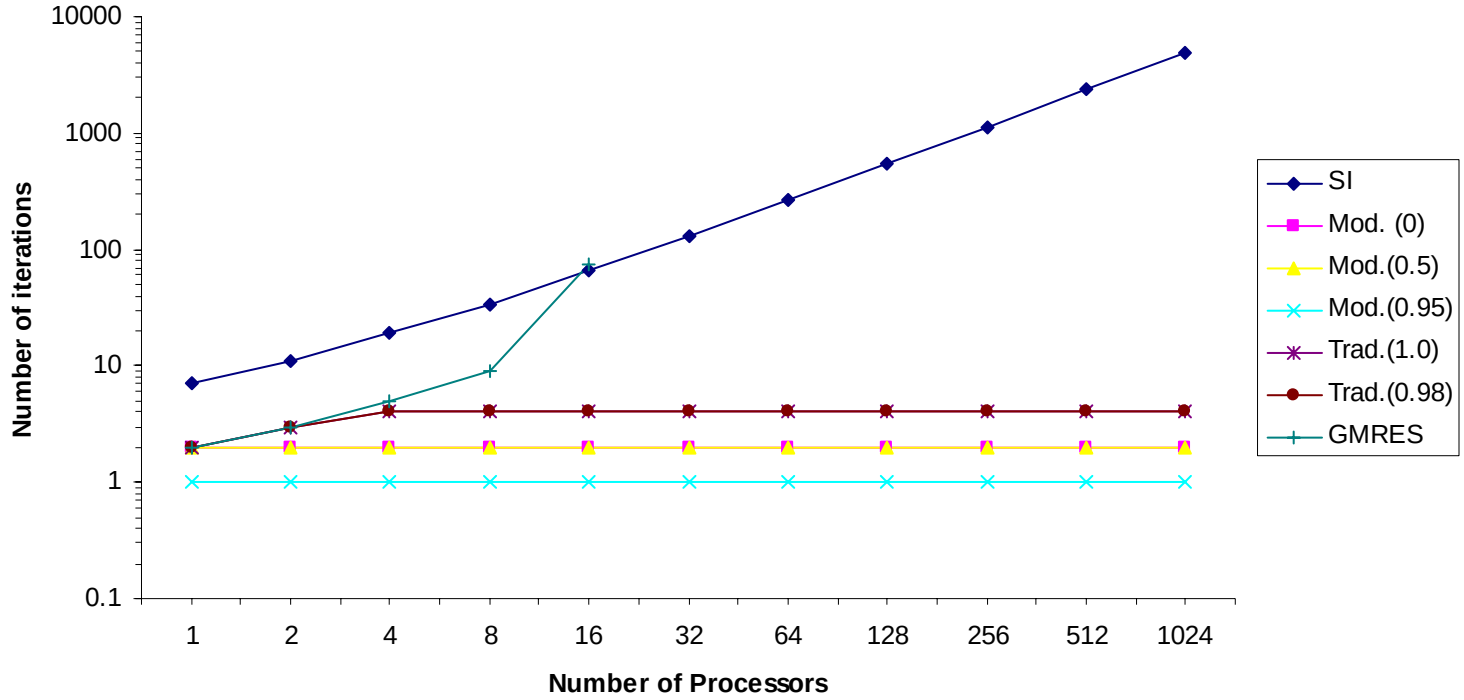
- Fixed work per processor: n = 65536 x N_p



Parallel Performance and Scaling

Strong Scaling

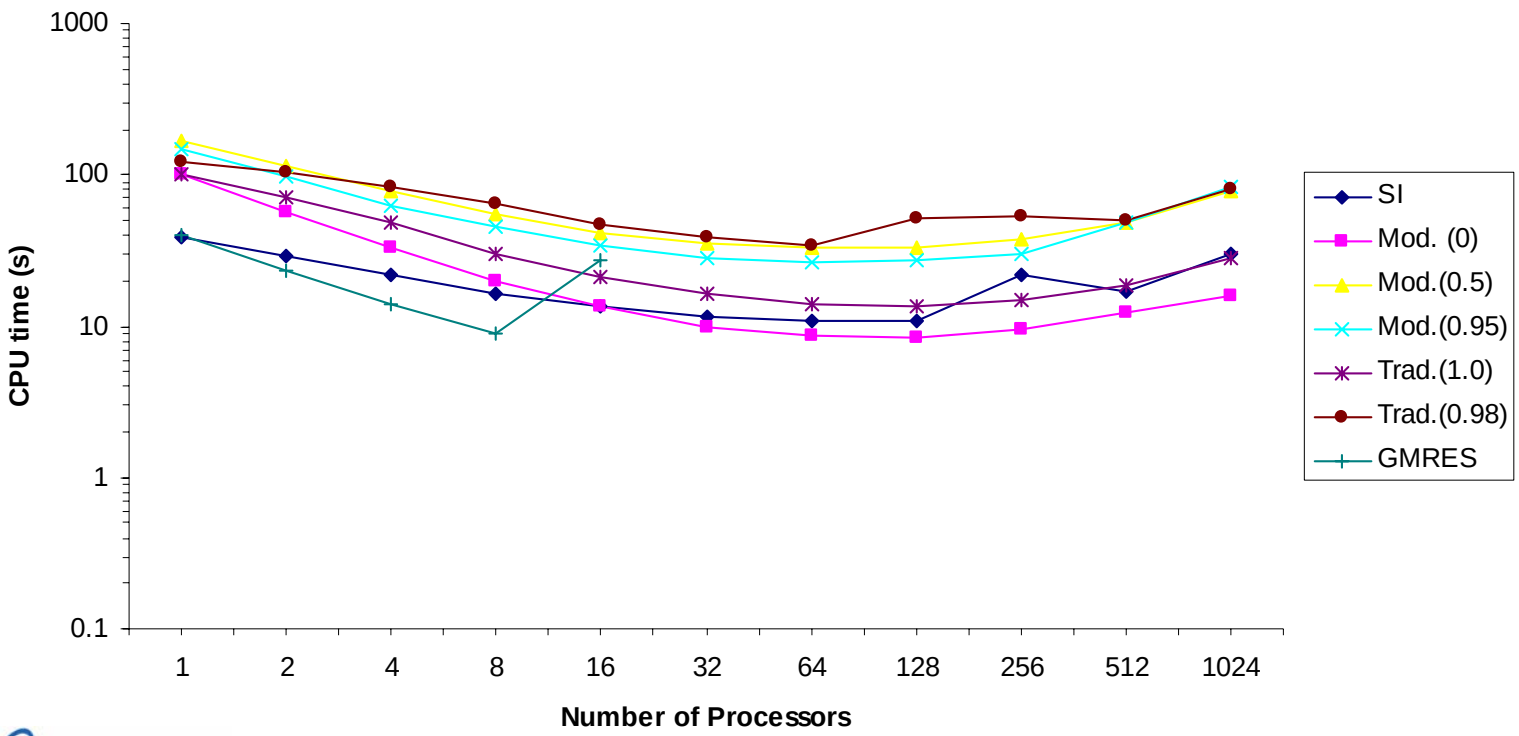
- Fixed problem size: n = 2097152



Parallel Performance and Scaling

Strong Scaling

- Fixed problem size: n = 2097152



Summary and Conclusions

- GMRES stagnates for optically thin problems when the restart value is less than the number of processors.
- Modified TSA is an effective preconditioner for GMRES when the problem is optically thin.
- Need to explore 2D problems