Parallel Domain Decomposition Preconditioning For Computational Fluid Dynamics

VecPar '98, Porto, Portugal, June 21-23, 1998

Timothy J. Barth
Information Sciences Directorate
NASA Ames Research Center
Moffett Field, California USA, 94035

Tony F. Chan
Department of Mathematics
University of California, Los Angeles
Los Angeles, California USA, 90024

Wei-Pai Tang
Department of Computer Science
University of Waterloo
Waterloo, Ontario Canada, N2L 3G1

Outline

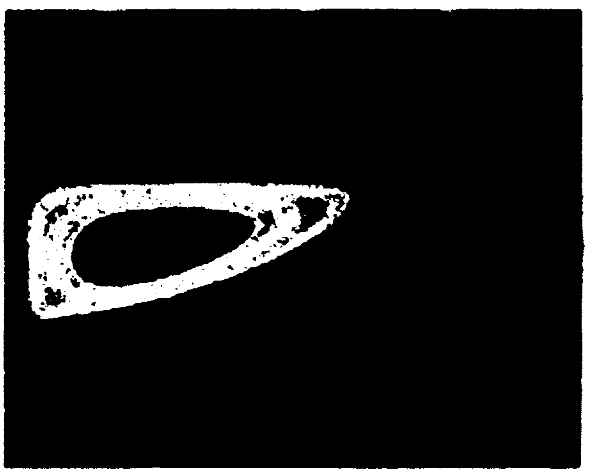
- Objectives
- Some Difficult Fluid Flow Problems
- Stabilized Spatial Discretizations
- Newton's Method for Solving the Discretized Flow Equations
- Schur Complement Domain Decomposition
  - Basic Formulation
  - Simplifying Strategies
    * Iterative Subdomain and Schur Complement Solves
    * Matrix Element Dropping
    * Localized Schur Complement Computation
    * Superspace Computations
  - Performance Evaluation
- Concluding Remarks

Objectives

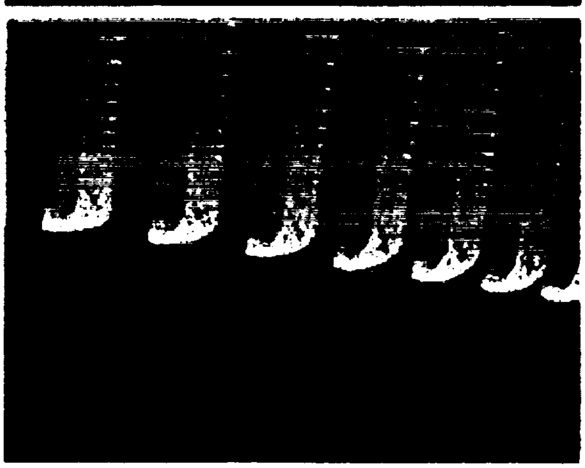
- Simulate Compressible Navier-Stokes Flow About General Geometries
  - Unstructured (Simplicial) Meshes
  - Stabilized Numerical Space Discretizations
  - Exact and Approximate Newton Iterations
- Obtain Parallel Scalability and Efficiency
  - Schur Complement Domain Decomposition
  - ILU + GMRES on Subdomain Problems
  - Emphasis on Coarse Grain Parallism
    - * SGI Origin2000 (64 processors)
    - * SGI Array (40 processors)
    - * CRAY J90 Cluster (20 processors)

Some Difficult Flow Problems

Entrance/Exit Flow



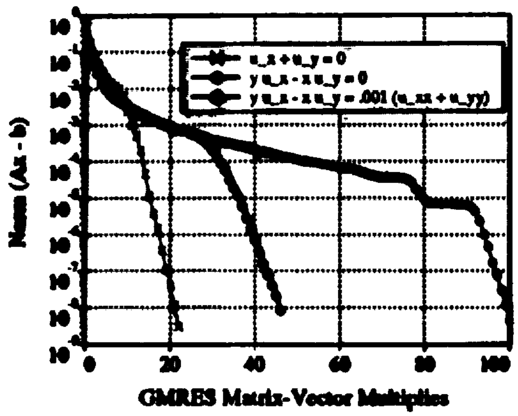
Recirculation Flow

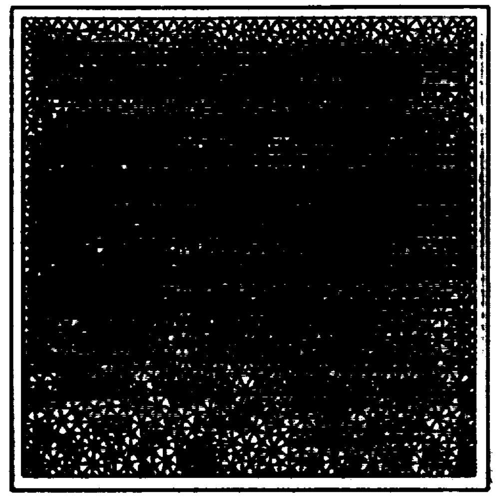


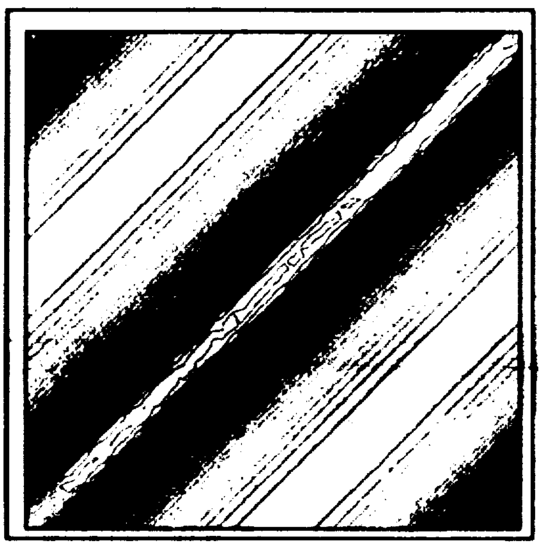
Boundary-layer Flow

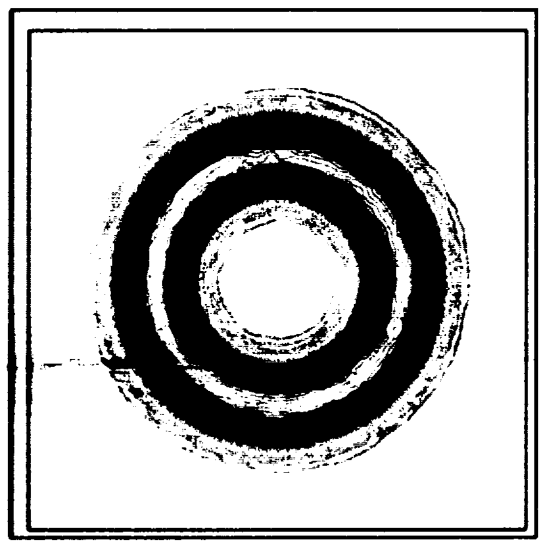
Barth, Chan, Tang (1998)

Advection Dominated Model Problems









$$u_x + u_y = 0 \qquad\qquad yu_x - xu_y = \lim_{\varepsilon \to 0} \varepsilon \Delta u$$

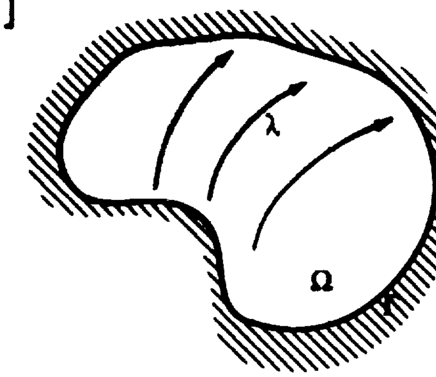
Convergence of ILU+GMRES for Cuthill-McKee ordered matrices produced from scalar SUPG spatial discretization.

Barth, Chan, Tang (1998)

Stabilized Numerical Methods
Advection Dominated Flows

- **Model Equation:**

$$u_t + \lambda \cdot \nabla u = \epsilon\, \Delta u, \;\; (x,t) \in \Omega \times [0,T]$$



with

$$u(x,0) = u_0(x), \;\; x \in \Omega$$

$$u(x,t) = g(x,t), \;\; x \in \Gamma$$

- **Stabilized F.E.M.:** (Johnson, Hughes, et. al.)

Find $u \in S_h$ such that $\forall\, w \in V_h$

$$\int_\Omega w\, u_t \; d\Omega + \int_\Omega w\, (\lambda \cdot \nabla u) d\Omega + \int_\Omega \epsilon\, (\nabla w \cdot \nabla u)\; d\Omega$$
$$+ \int_{\overline{\Omega}} (\lambda \cdot \nabla u - \epsilon \Delta u)\, \tau\, (\lambda \cdot w - \epsilon \Delta w)\; d\Omega$$
$$+ \text{ Discontinuity Capturing Operator} = 0$$

Barth, Chan, Tang (1998)

ELF

(an Element Library for Fluids)

Euler equations in divergence form:

$$u_t + f(u)_x + g(u)_y = 0, \quad f(u), g(u) : \mathbf{R}^m \mapsto \mathbf{R}^m$$

Euler equations in symmetric quasilinear form $u \mapsto v$:

$$\underbrace{A_0}_{SPD} v_t + \underbrace{AA_0}_{symm} v_x + \underbrace{BA_0}_{symm} v_y = 0, \quad A_0, A, B \in \mathbf{R}^{m \times m}$$

ELF Methodology: Galerkin Least-Squares in symmetric variables

ELF Input: Simplex geometry, Simplex dofs.

ELF Output: Jacobian matrix and rhs terms for global assembly.

Barth, Chan, Tang (1998)

Newton's Method

Fluid flow equations in semi-discrete form:

$$Du_t = R(u), \quad R(u) : \mathbf{R}^n \to \mathbf{R}^n$$

Backward Euler time integration:

$$\left[\frac{I}{\Delta t} - \left(\frac{\partial R}{\partial u}\right)^n\right](u^{n+1} - u^n) = R(u^n)$$

Let Δt vary with spatial position and $\|R(u)\|$ so that Newton's method is approached as $\|R(u)\| \to 0$.

Solving A x = b

Let

$$Ax - b = 0$$

represent a matrix problem taken from one step of Newton's method.

Solve using flexible GMRES in right preconditioned form:

$$(AM^{-1})Mx - b = 0.$$

- Allows nonstationary preconditioning (M)

Ideally

$$\kappa\left(AM^{-1}\right) = O(1).$$

Barth, Chan, Tang (1998)

Preconditioners

Some candidate preconditioners:

1. ILU factorization
   - Nonoptimal
   - Not easily parallized
2. Additive Schwarz variants of ILU on overlapping meshes
   - Nonoptimal without coarse space correction
   - 3-D tetrahedral coarsening is problematic
3. Multigrid
   - Similar difficulties as additive Schwarz
   - Agglomeration ?
4. Nonoverlapping Schur domain decomposition
   - Data locality well-suited to coarse grain parallel architectures
   - Algebraic coarse space provides global coupling

Nonoptimality of ILU Preconditioning

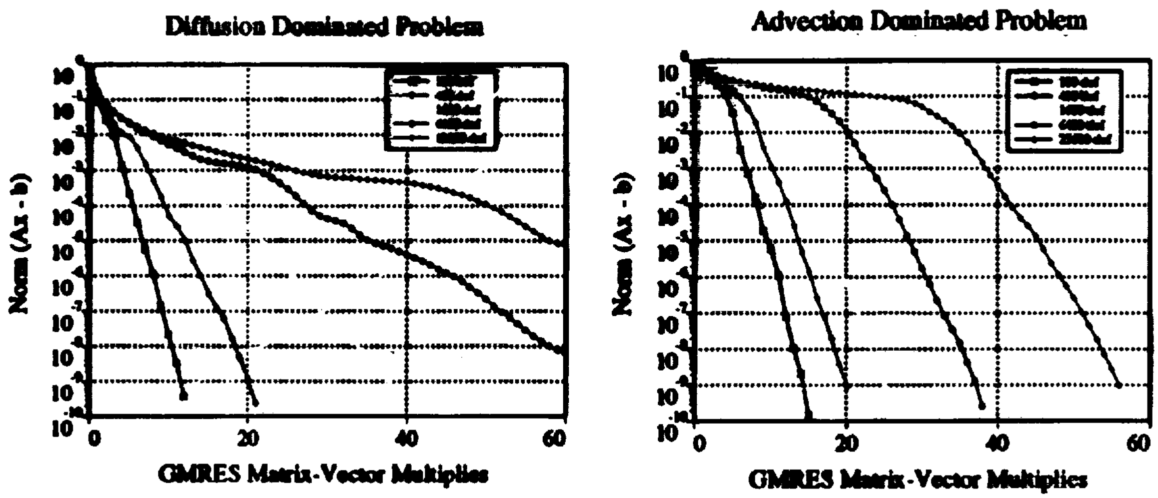
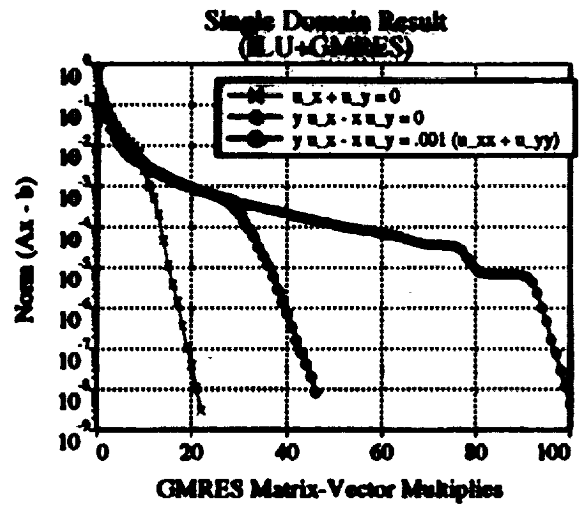


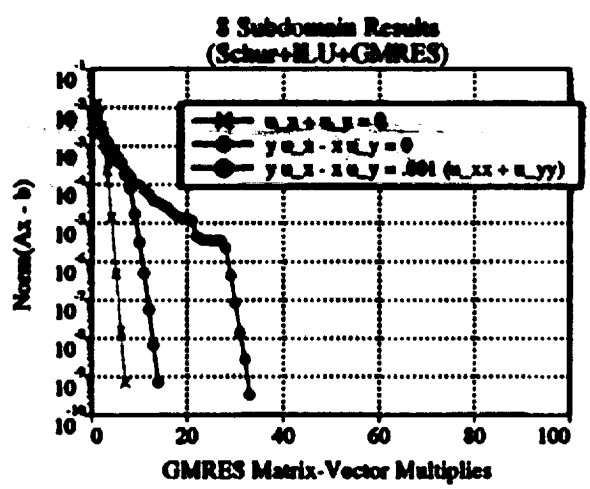
Figure 1-2. Performance of ILU for diffusion and advection dominated problems using scalar SUPG discretization and Cuthill-McKee ordering.

- ILU retains $O\left(\frac{1}{h^2}\right)$ condition number for elliptic problems.
  - Use of modified ILU variants is problematic in the nonsymmetric (advection dominated) limit.

Barth, Chan, Tang (1998)

Serial versus Parallel ILU



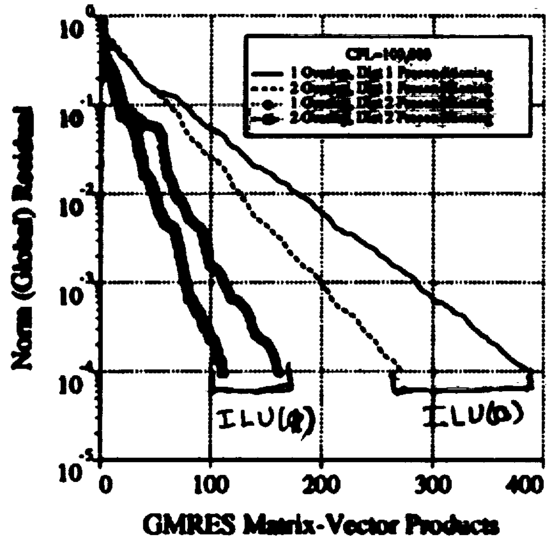


Comparison of single domain ILU preconditioning with multiple domain ILU via Schur complement.

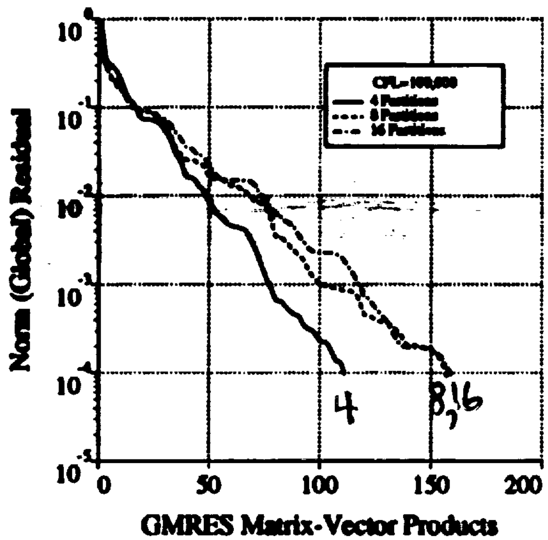
Barth, Chan, Tang (1998)

Additive Schwarz Preconditioned GMRES

(From Barth, AGARD R-807, 1995)



Effect of increased overlap and ILU fill.



Effect of increasing the number of subdomains.

- Inviscid flow about multi-element airfoil (5k vertices)
- 1 sweep overlapped ILU with no coarse space correction

Barth, Chan, Tang (1998)

Schur Complement Domain Decomposition (Some Contributors)

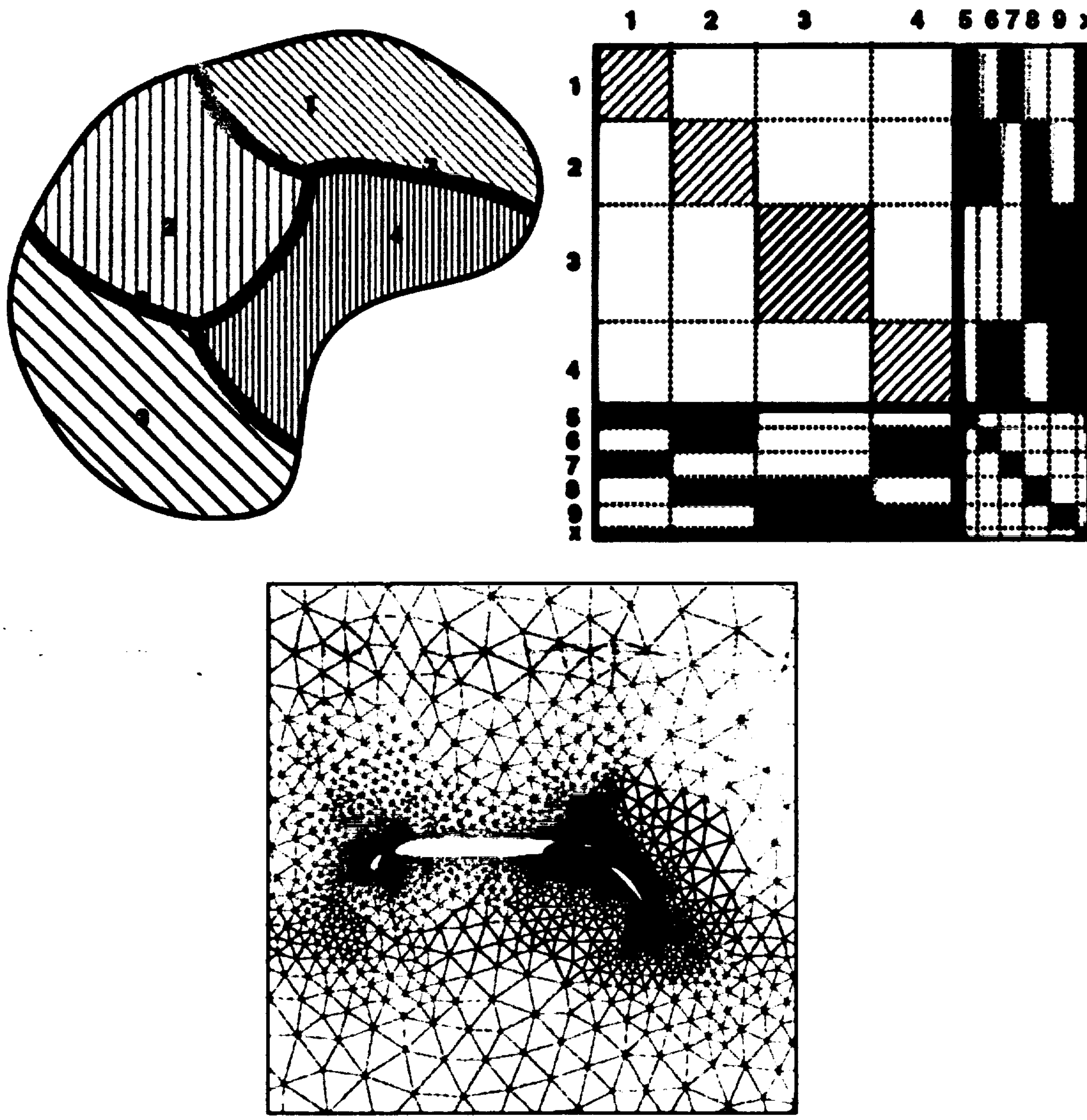
Basic Approach: Axelsson, Glowinski, Lions, Mandel, Nepomyaschikh, Périaux, Przemieniecki, Widlund, Xu

Interface Preconditioners: Bjorstad, Bramble, Chan, Keyes, Golub, Gropp, Mayers, Pasciak, Schatz, Smith

Parallel Implementation Issues: Chan, Keyes, Gropp, Smith

See Also: **Domain Decomposition: Parallel Multilevel Methods for Elliptic PDEs**, B. Smith and P. Bjorstad and W. Gropp, Cambridge Press, 1996.

Nonoverlapping Domain Decomposition via Schur Complement



Domain decomposition and induced 2×2 matrix partitioning.

Barth, Chan, Tang (1998)

Solving a 2×2 Block Matrix

- Subdomain partition as illustrated.
- Order the subdomain variables before the interface variables.
- 2×2 form of the system

$$Ax = \begin{bmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{bmatrix} \begin{pmatrix} x_I \\ x_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix},$$

- x_I, x_B denote the interior and the boundary variables.
- Block LU factorization of A:

$$A = LU = \begin{bmatrix} A_{II} & 0 \\ A_{BI} & I \end{bmatrix} \begin{bmatrix} I & A_{II}^{-1} A_{IB} \\ 0 & S \end{bmatrix}.$$

- Eliminate x_B from the reduced equations:

$$S x_B = f_B - A_{BI} A_{II}^{-1} f_I$$

- Solve for x_I by

$$x_I = A_{II}^{-1} (f_I - A_{IB} x_B),$$

- $S = A_{BB} - A_{BI} A_{II}^{-1} A_{IB}$ is the *Schur Complement* of A_{BB} in A.

Exact Schur Complement Preconditioning

(Naive)

Preprocessing Step: Calculate the Schur complement

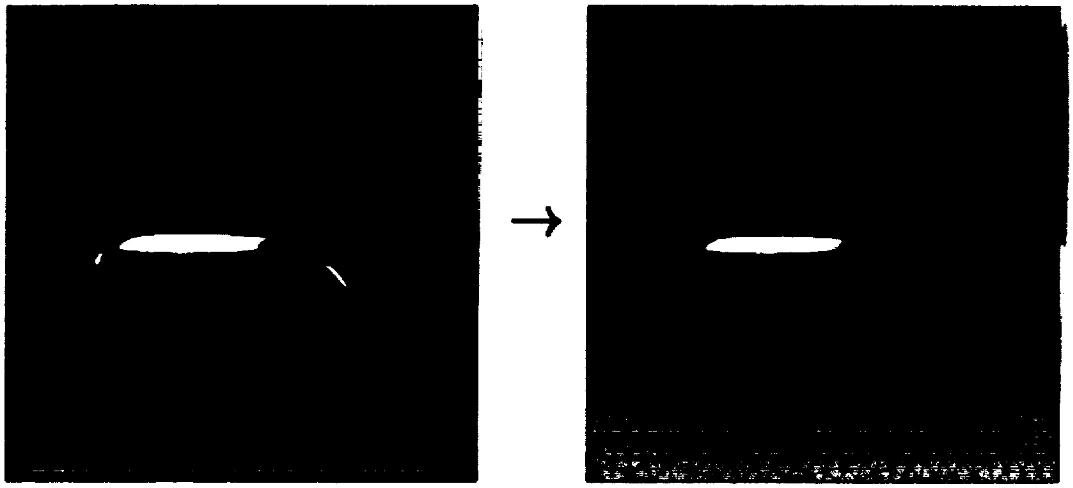
$$S = A_{BB} - \sum_i (A_{BI})_i (A_{II})_i^{-1} (A_{IB})_i .$$

Preconditioning Step: Solve $Mx = r$

Step (1)	$u_i = (A_{II})_i^{-1}\, r_i$	(parallel)
Step (2)	$v_i = (A_{BI})_i\, u_i$	(parallel)
Step (3)	$w_b = r_b - \sum v_i$	(comm)
Step (4)	$x_b = S^{-1}\, w_b$	(s-parallel,comm)
Step (5)	$y_i = (A_{IB})_i\, x_b$	(parallel)
Step (6)	$x_i = u_i - (A_{II})_i^{-1}\, y_i$	(parallel)

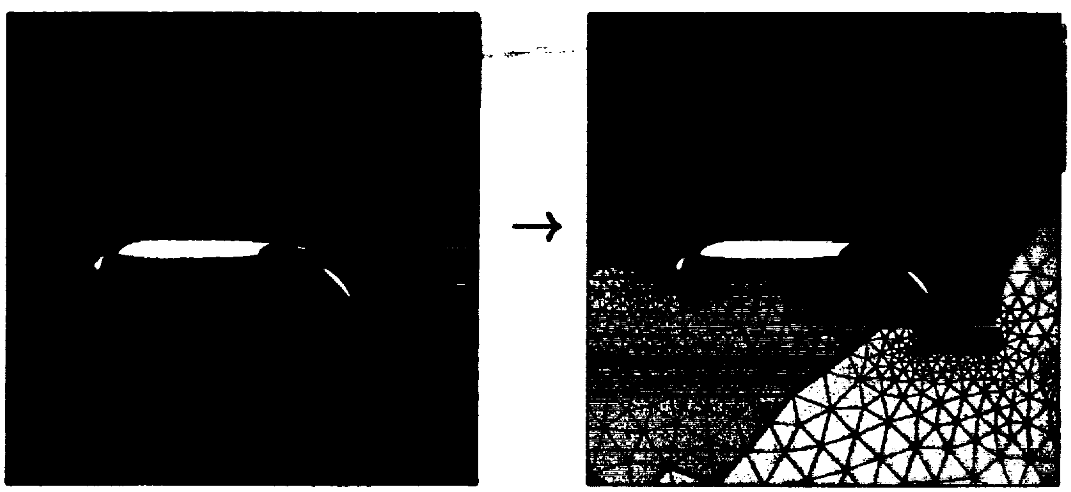
Scalability Experiments

Fixed Subdomain Size, 4×Partitions:



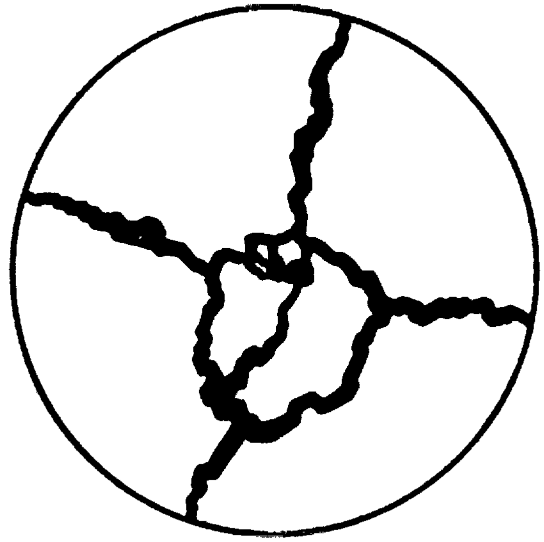
- Interior and interface dofs/partition is constant.
- Total interface size increases 4×.

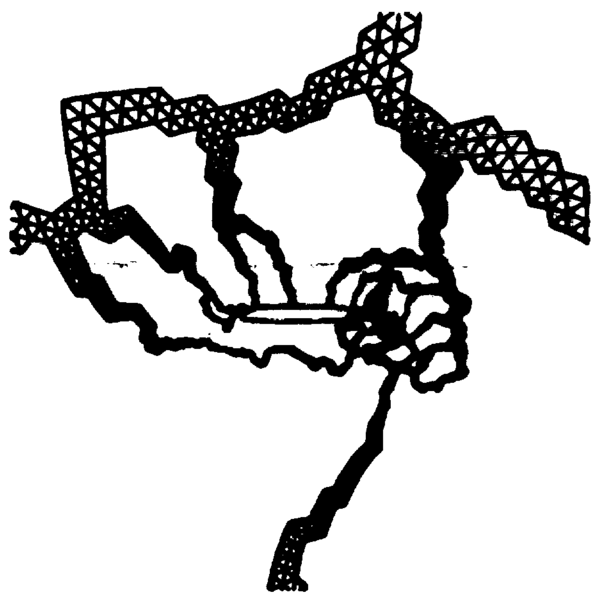
Fixed Problem Size, 4×Partitions:



- Interior and interface dofs/partition decrease 4× and 2× respectively.
- Total interface size increases 2×.

Interface Decomposition





Overview (top) and closeup (bottom) of interface decomposition obtained via partitioning of the Schur complement supernode graph (64 subdomains, 4 interfaces).

Barth, Chan, Tang (1998)

Global Preconditioning Strategy for $Ax = b$

- Partition subdomains and interfaces
- Assign a processor to each subdomain and interface component
- Precondition subdomain problems using processor-local ILU
- Precondition the Schur complement using drop-filled, processor-local ILU factorizations
- Implement flexible GMRES in a domain decomposed parallel environment
  - Parallel dot products
  - Parallel matrix-vector products

Preconditioner I: Inexact Subproblem Solves

Preprocessing Step: Calculate the approximate Schur complement

$$\tilde{S} = A_{BB} - \sum_i (A_{BI})_i (\tilde{A}_{II})_i^{-1} (A_{IB})_i.$$

1. $(A_{II})_i^{-1}(A_{IB})_i \rightarrow (\tilde{A}_{II})_i^{-1}(A_{IB})_i$ using m_1 steps of ILU+GMRES so that $S \rightarrow \tilde{S}$.
2. $(A_{II})_i^{-1} z_i \rightarrow \widehat{A}_{II}^{-1} z_i$ using m_2 steps of ILU+GMRES.
3. $\tilde{S}^{-1} w_b \rightarrow \widehat{S}^{-1} w_b$ using m_3 steps of ILU+GMRES.

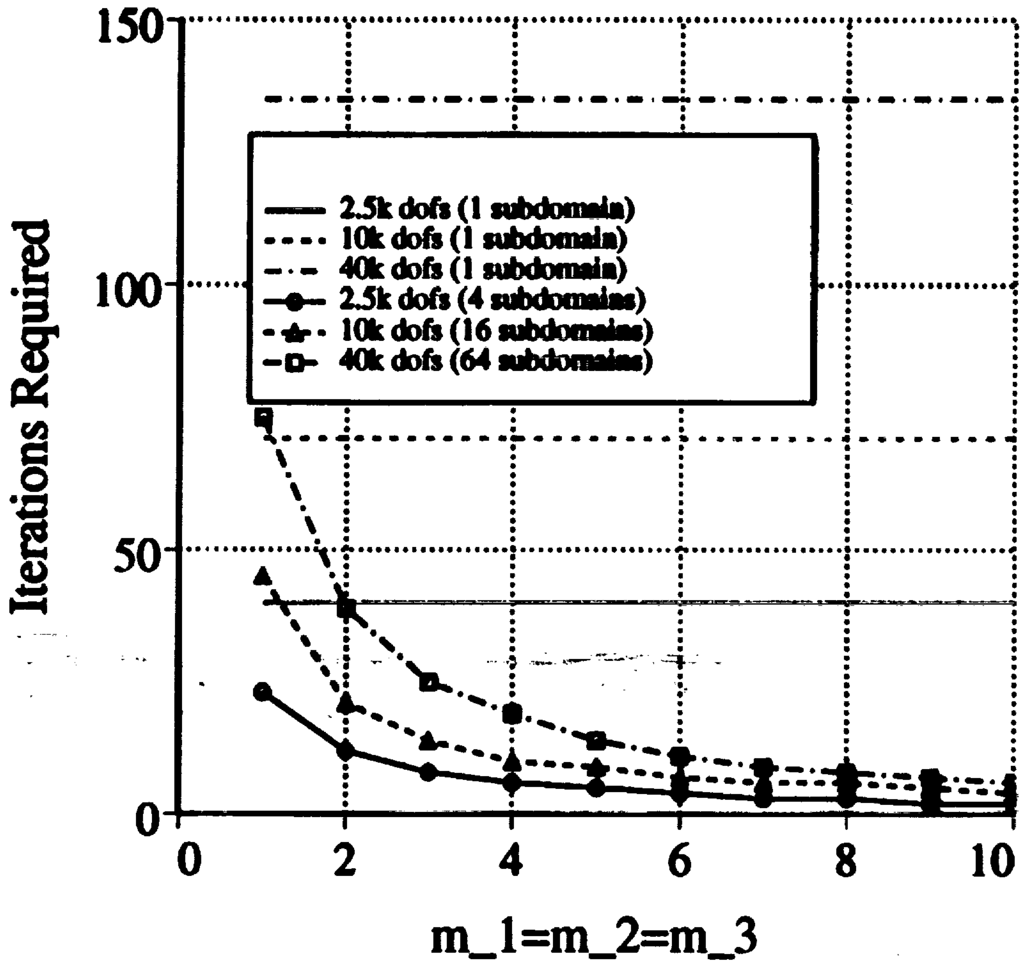
Preconditioning Step: Solve $Mx = r$

Step (1)	$u_i = (\widehat{A}_{II})_i^{-1}\, r_i$	(parallel)
Step (2)	$v_i = (A_{BI})_i\, u_i$	(parallel)
Step (3)	$w_b = r_b - \sum v_i$	(comm)
Step (4)	$x_b = \widehat{S}^{-1}\, w_b$	(s-parallel,comm)
Step (5)	$y_i = (A_{IB})_i\, x_b$	(parallel)
Step (6)	$x_i = u_i - (\widehat{A}_{II})_i^{-1}\, y_i$	(parallel)

Barth, Chan, Tang (1998)

Preconditioner I: Performance

Diffusion Dominated

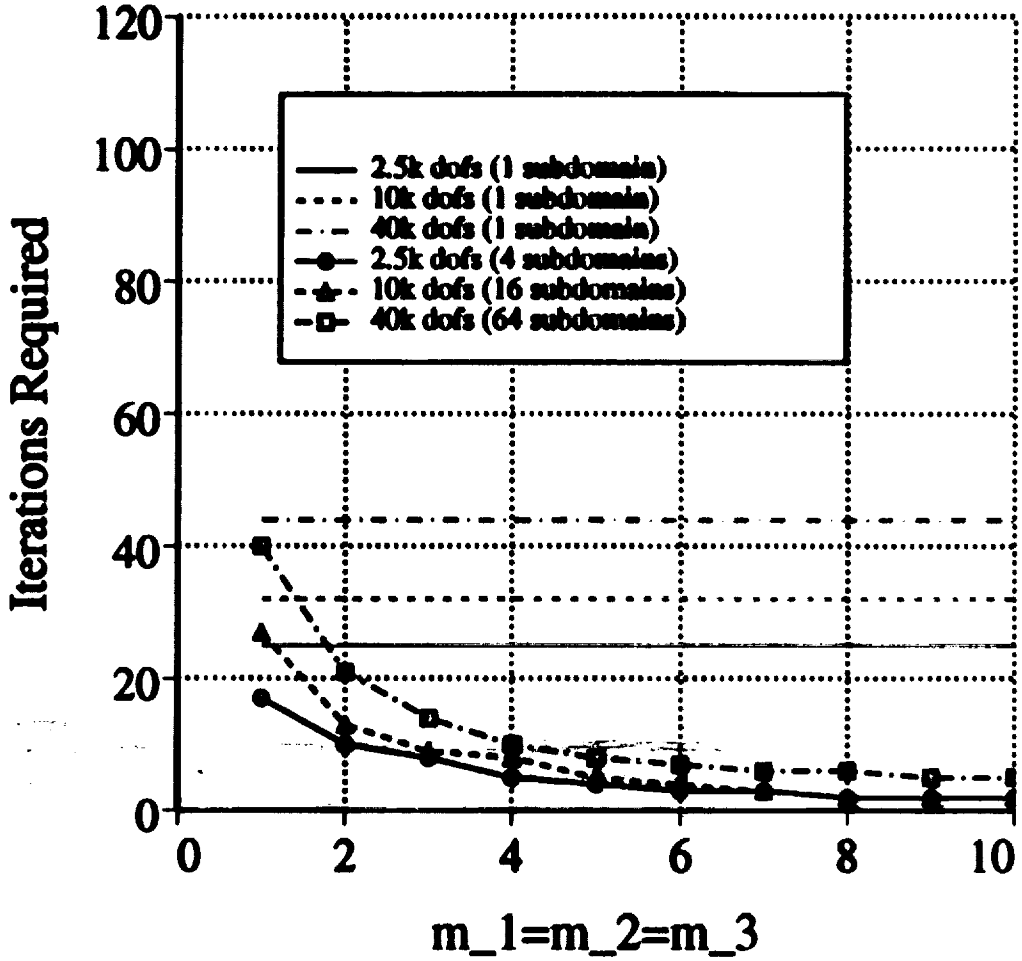


- Scalar SUPG discretization with linear elements
- Stopping criterion: $\|Ax_n - b\| \le 10^{-8}\|Ax_0 - b\|$

Barth, Chan, Tang (1998)

Preconditioner I: Performance

Advection Dominated



- **Scalar SUPG discretization with linear elements**
- **Stopping criterion:** $\|Ax_n - b\| \leq 10^{-8}\|Ax_0 - b\|$

Barth, Chan, Tang (1998)

Observations for Preconditioner I

- There exists an optimal number of subdomains for minimizing solution time.
- Replacing a single long recurrence ILU factorization with several shorter recurrences coupled via Schur complement yields an improved quality preconditioner.
- Choosing small values of the parameters m_1, m_2, and $m_3 < 3$ leads to minimum CPU times. (our experience).
- Linear growth in the time needed to form the preconditioner is avoided by further partitioning of the interface. How to precondition the parallelized interface? Processor-Local ILU.

Preconditioner II: Drop Tolerance Approximation

Drop elements of the Schur complement matrix, $\tilde{S}_{ij}$:

(1) $|S_{ij}| < tol$ (scalar matrix entries)

(2) Sparsity pattern (block matrix entries)

– For discretized elliptic problems S exhibits faster element decay than $(A_{II})_i^{-1}$ (Golub and Mayers).

Preprocessing Step: Calculate the approximate Schur complement

$$\widehat{S} = A_{BB} - \sum_i (A_{BI})_i (\tilde{A}_{II})_i^{-1} (A_{IB})_i$$

– Precondition $\widehat{S}$ with $ILU(DROP(\widehat{S}))$

Preconditioning Step: Solve $Mx = r$

Step (1)	$u_i = (\widehat{A}_{II})_i^{-1}\, r_i$	(parallel)
Step (2)	$v_i = (A_{BI})_i\, u_i$	(parallel)
Step (3)	$w_b = r_b - \sum v_i$	(comm)
Step (4)	$x_b = \widehat{S}^{-1}\, w_b$	(s-parallel,comm)
Step (5)	$y_i = (A_{IB})_i\, x_b$	(parallel)
Step (6)	$x_i = u_i - (\widehat{A}_{II})_i^{-1}\, y_i$	(parallel)

Barth, Chan, Tang (1998)

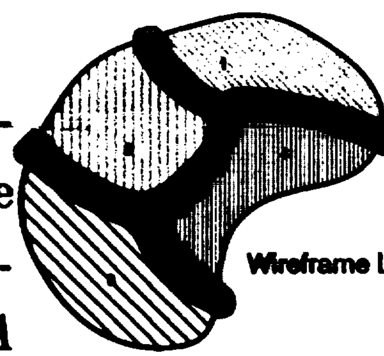
Preconditioner III: Wireframe Approximation

- Main cost in Preconditioner II is the formation of S itself: one subdomain solve is needed for *every* variable on the interface.

- Idea: Approximate S by the Schur complement of a relatively thin *wireframe* region surrounding the interface variables.

- Take a principal submatrix of A: variables within the wireframe.

- Compute the approximate Schur complement of the interface variables in this principal submatrix instead of A itself.



- From elliptic DD theory: the exact Schur complement of wireframe region is spectrally equivalent to the Schur complement of the whole domain.

Barth, Chan, Tang (1998)

Preconditioner III: Wireframe Approximation

Pure Diffusion Problem:

Mesh Size	Num Procs	Support	Iter	Time Form*	Time Apply
1600	16	2	15	.012	.11
1600	16	3	11	.017	.08
1600	16	4	6	.020	.04

Pure Advection Problem:

Mesh Size	Num Procs	Support	Iter	Time Form*	Time Apply
1600	16	2	8	.012	.09
1600	16	3	6	.017	.06
1600	16	4	4	.019	.03

– Scalar SUPG discretization with linear elements

– Stopping criterion: $\|Ax_n - b\| \leq 10^{-9}\|Ax_0 - b\|$

– No element dropping, $m_1 = 2, m_2 = 2, m_3 = 2$

* Timing based on parallelized wireframe

Barth, Chan, Tang (1998)

Preconditioner III: Wireframe Approximation

- Introduces a new adjustable parameter into the preconditioner.
- Specify the width of the wireframe strip in terms of graph distance on the mesh triangulation.
- Quality of the preconditioner improves rapidly with increasing wireframe width.
- Time taken form the Schur complement is reduced by approximately 50%.
- Further improvements in cost/performance: choosing the shape of the wireframe to better represent the PDE being solved, viz. the flow direction.

Preconditioner III: Wireframe Approximation

- **Preprocessing step: approximate Schur complement on the localized subdomains:**

$$\widehat{S} = A_{BB} - \sum_i (\tilde{A}_{BI})_i (\tilde{A}_{II})_i^{-1} (\tilde{A}_{IB})_i$$

  - $\tilde{A}_{XY}$: restrictions of A_{II}, A_{IB}, A_{BI} to the localized Schur subdomain(s).

- **The preconditioning step (solving $Mx = r$) is now:**

Step (1)	$u_i = (\widehat{A}_{II})_i^{-1} r_i$	(parallel)
Step (2)	$v_i = (A_{BI})_i u_i$	(parallel)
Step (3)	$w_b = r_b - \sum v_i$	(comm)
Step (4)	$x_b = \widehat{S}^{-1} w_b$	(s-parallel,comm)
Step (5)	$y_i = (A_{IB})_i x_b$	(parallel)
Step (6)	$x_i = u_i - (\widehat{A}_{II})_i^{-1} y_i$	(parallel)

Preconditioner IV: Supersparse Approximation

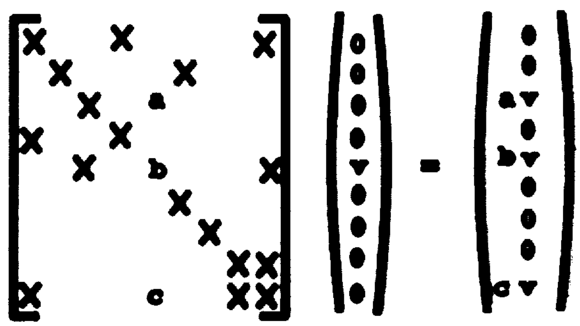
- Observations:

1. Iterative calculation of $A_{II}^{-1} A_{IB}$ is expensive.
2. Columns of A_{IB} are usually very sparse.
3. Unpreconditioned Krylov subspace methods require computation of the vector sequence $[p, Ap, A^2p, \ldots, A^m p]$, $p = col(B)$ for small m.
4. ILU preconditioning destroys sparsity of the preconditioned Krylov subspace vectors $[p, M^{-1}Ap, (M^{-1}A)^2 p, \ldots, (M^{-1}A)^m p]$.

Preconditioner IV: Supersparse Approximation

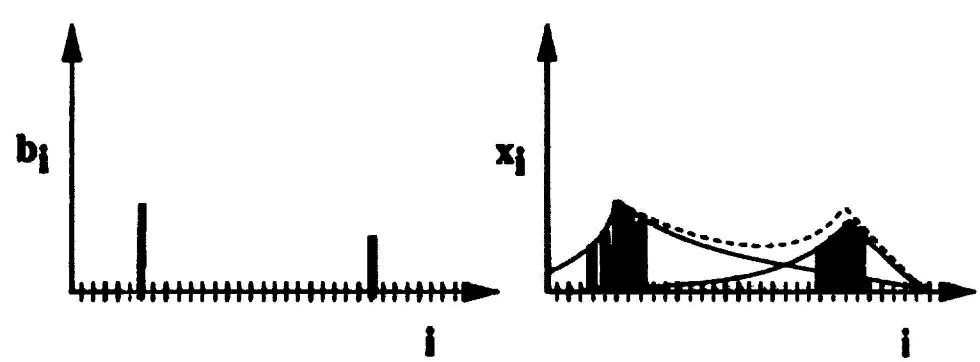
- Strategy:

  1. Eliminate wasted flop's in matrix-vector products by storing *vectors* and matrices in sparse form.



  2. Approximate the ILU backsolves based on a vector fill level strategy to maintain sparsity.

$Ax = b$



Barth, Chan, Tang (1998)

Preconditioner IV: Supersparse Approximation

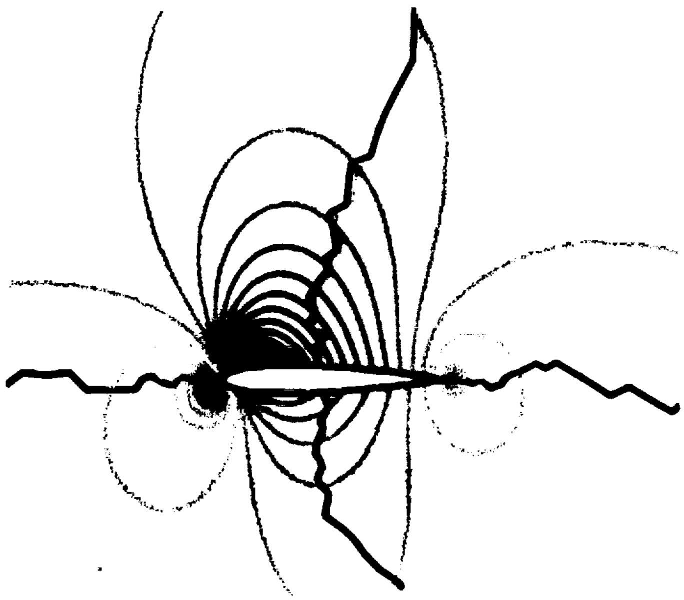
- Results indicate preconditioning performance comparable to exact backsolves.
- 60-70% reduction in cost.
- This technique can be combined with the previous wireframe strategy with combined 5-7 fold performance gains.

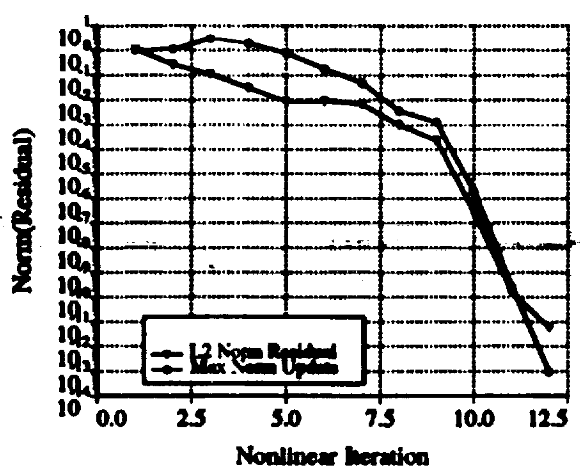
Fill Level k	Global GMRES Iter	Time(k)/Time(∞)
0	26	0.299
1	22	0.313
2	21	0.337
3	20	0.362
4	20	0.392
∞	20	1.000

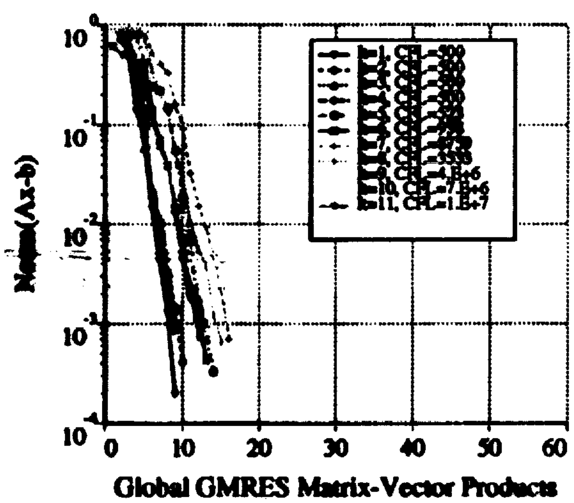
2-D test problem consisting of Euler flow past a multi-element airfoil geometry partitioned into 4 subdomains with 1600 mesh vertices in each subdomain.

Barth, Chan, Tang (1998)

Compressible Flow Calculation



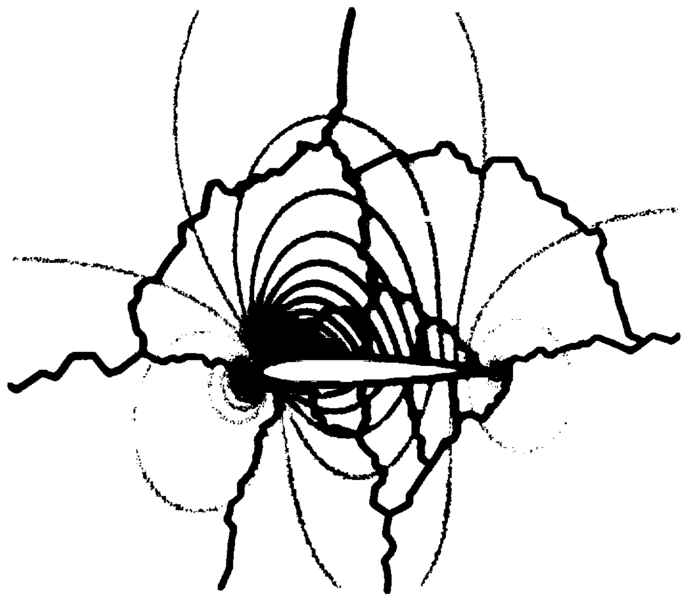


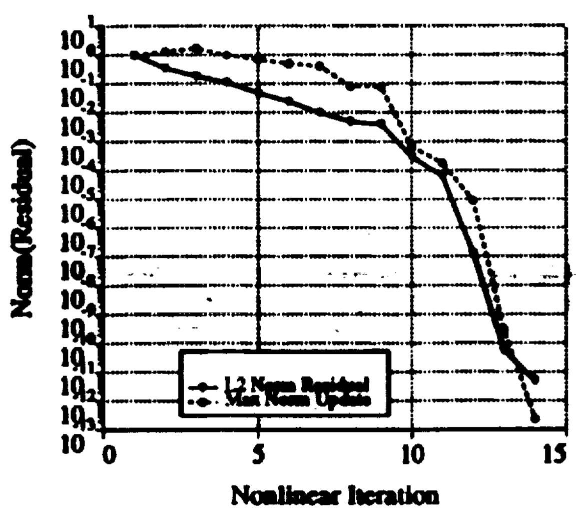


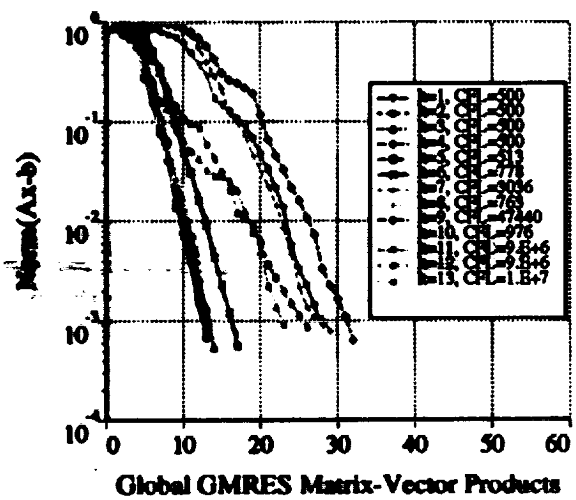
- Mach = .63, Angle of Attack = 2.0°
- ELF Galerkin Least-Squares
- Time Relaxed Newton Iteration
- $m_1 = 1$, $m_2 = m_3 = 4$, Fill Level=3
- 4 Subdomain Mesh, (5k Cells/Subdomain)

Barth, Chan, Tang (1998)

Compressible Flow Calculation



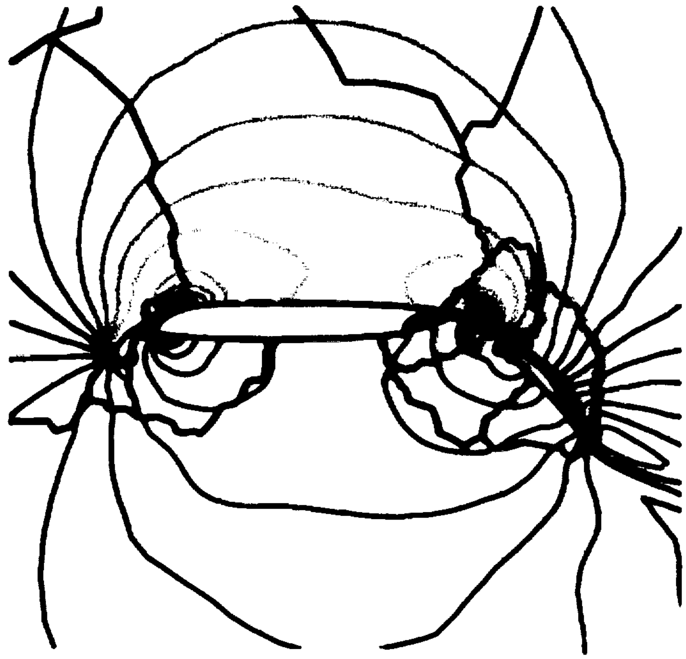


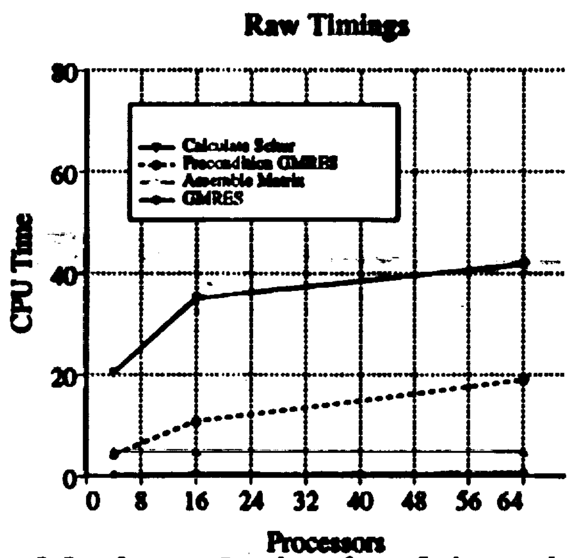


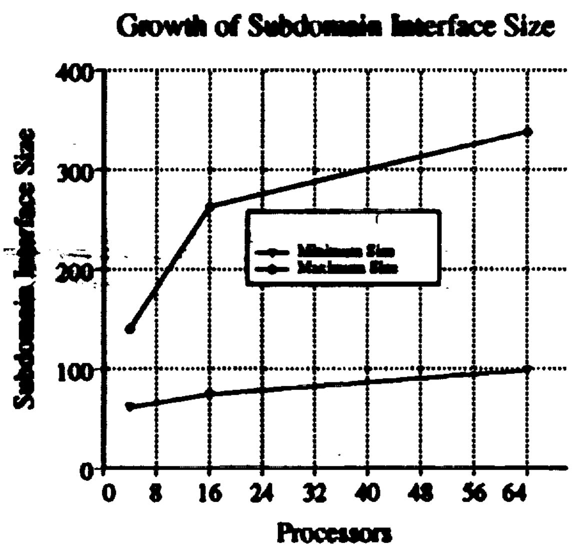
- Mach = .63, Angle of Attack = 2.0°
- ELF Galerkin Least-Squares
- Time Relaxed Newton Iteration
- $m_1 = 1$, $m_2 = m_3 = 4$, Fill Level=3
- 16 Subdomain Mesh, (5k Cells/Subdomain)

Barth, Chan, Tang (1998)

Subdomain - Interface Load Balancing



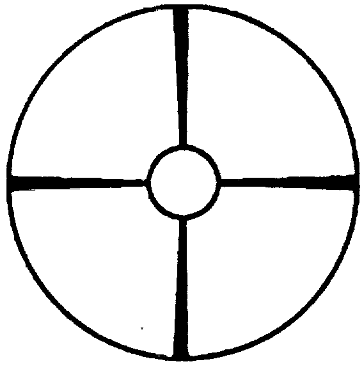




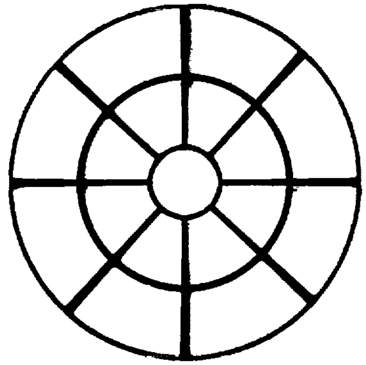
- **Mach = .2, Angle of Attack = 2.0°**
- **16 Subdomain Mesh, (5k Cells/Subdomain)**
- **IBM SP2, MPI Message Passing Protocol**
- **ELF Galerkin Least-Squares**

Barth, Chan, Tang (1998)

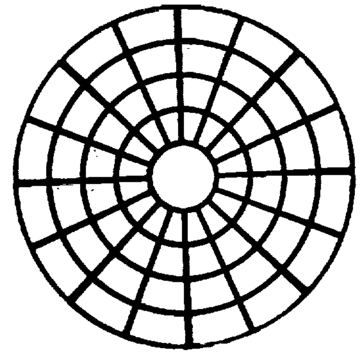
Preliminary Timings



Subdomain Interf=100



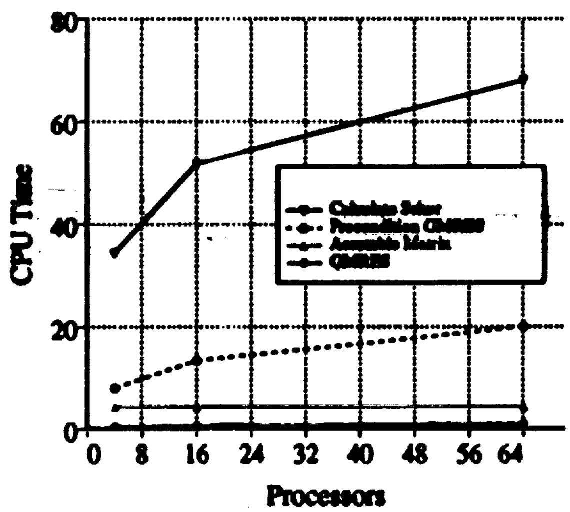
Subdomain Interf=150



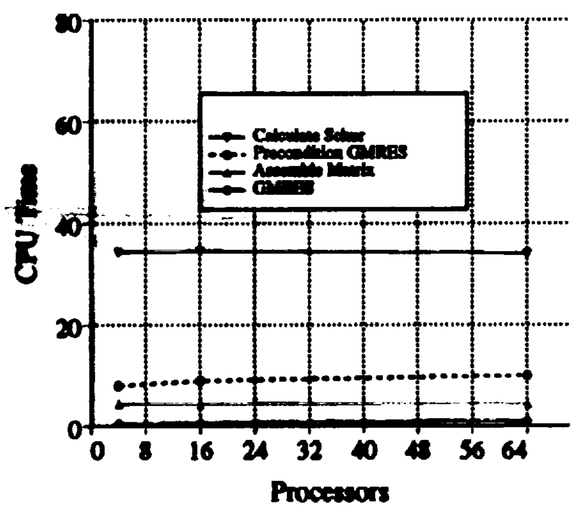
Subdomain Interf=150-200

(4 Interface Partitions)

Raw Timings



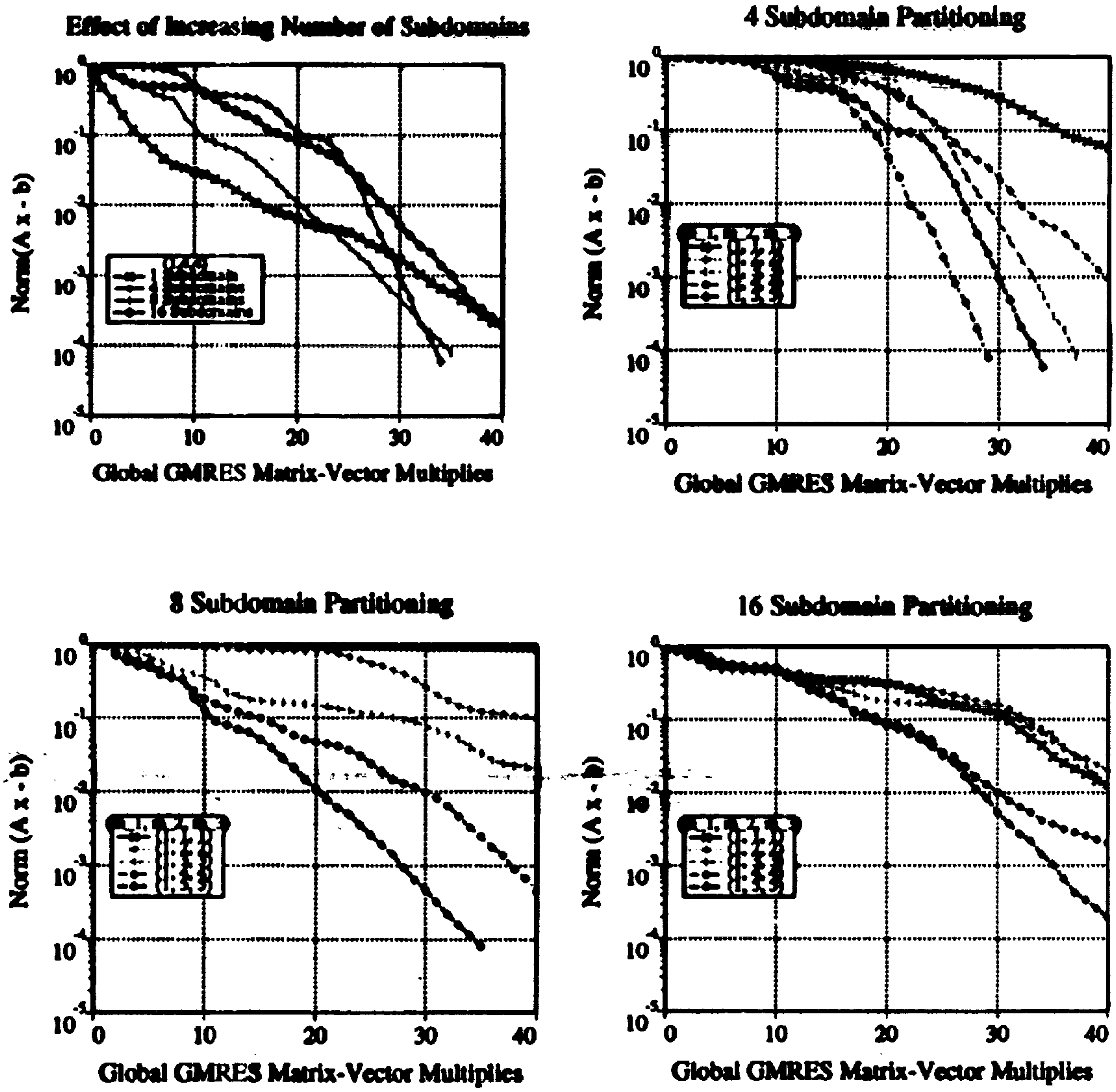
Adjusted for Interface Size



- 6 GMRES Iterations
- 5k Cells/Subdomain
- IBM SP2, MPI Message Passing Protocol
- $m_1 = 1$, $m_2 = m_3 = 4$, Fill Level=3

Barth, Chan, Tang (1998)

Performance of Schur Complement D.D. For Viscous Flow



- SUPG Discretization
- Local CFL Number 100,000
- 4, 8, 16 partitions with fixed mesh size

Barth, Chan, Tang (1998)

Concluding Remarks

- The baseline nonoverlapping scheme is very robust but relatively expensive
- Localized wireframe and supersparse computations can significantly reduce cost of the forming the Schur complement preconditioner
- Scalability: Growth of the Schur complement matrix necessitates partitioning of the interface
- Scalability: Load balancing still problematic due to size imbalance of the interface associated with each subdomain