Fast Iterative Solution of Models of Incompressible Flow

Howard Elman
University of Maryland

In collaboration with:

• Victoria Howle	Sandia National Laboratories
• David Kay	University of Sussex
• Daniel Loghin	University of Birmingham
• Milan Mihajlovic	University of Manchester
• John Shadid	Sandia National Laboratories
• Robert Shuttleworth	University of Maryland
• David Silvester	University of Manchester
• Ray Tuminaro	Sandia National Laboratories
• Andy Wathen	University of Oxford

Outline

1. General approach:
   Block preconditioners for Navier-Stokes problems
2. Performance in an applied setting: MPSalsa
3. Application: Microfluidics
4. Ongoing / future research

General Statement of Problem: Incompressible Navier-Stokes Equations

$$\alpha u_t - \nu \nabla^2 u + (u \cdot \text{grad}) u + \text{grad}\, p = f$$
$$-\text{div}\, u = 0$$

α=0 $\rightarrow$ steady state problem

α=1 $\rightarrow$ evolutionary problem

Discretization and linearization $\longrightarrow$ Matrix equation

$$\begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \qquad \mathcal{A}x = b$$

Goal: Robust general solution algorithms
Easy to implement
Derived from subsidiary building blocks
Adaptible to a variety of scenarios
(steady / evolutionary / Stokes / Boussinesq)

General Approach to Preconditioning

Solving $\begin{pmatrix} F & B^T \\ B & -C \end{pmatrix}\begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \longleftrightarrow \mathcal{A}x = b$

Use preconditioner of form $\mathcal{Q} = \begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}$

Solve right-preconditioned system

$$[\mathcal{A}\mathcal{Q}^{-1}][\hat{x}] = b, \quad x = \mathcal{Q}^{-1}\hat{x}$$

using Krylov subspace method (GMRES)

$$\mathcal{A}\mathcal{Q}^{-1} = \begin{pmatrix} F & B^T \\ B & -C \end{pmatrix}\begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}^{-1} = \begin{pmatrix} FQ_F^{-1} & (FQ_F^{-1} - I)B^T Q_S^{-1} \\ BQ_F^{-1} & (BQ_F^{-1}B^T + C)Q_S^{-1} \end{pmatrix}$$

General Approach to Preconditioning

$$\mathcal{A}\mathcal{Q}^{-1} = \begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}^{-1} = \begin{pmatrix} FQ_F^{-1} & (FQ_F^{-1} - I)B^T Q_S^{-1} \\ BQ_F^{-1} & (BQ_F^{-1}B^T + C)Q_S^{-1} \end{pmatrix}$$

$$\overset{Q_F = F}{=} \begin{pmatrix} I & 0 \\ BF^{-1} & \underbrace{(BF^{-1}B^T + C)}_{S} Q_S^{-1} \end{pmatrix} \overset{Q_S = S}{=} \begin{pmatrix} I & 0 \\ BF^{-1} & I \end{pmatrix}$$

Eigenvalues $\equiv 1$ $\rightarrow$ Convergence in two steps

Seek approximation to inverses of

F ~ convection-diffusion operator

S = Schur complement matrix

Key point: Build using methods for scalar operators, use existing (multigrid) code

Two Strategies for Preconditioning *S*

$$\mathcal{Q} = \begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}$$

1. **Pressure Convection-Diffusion Preconditioner** $Q_S^{-1} \equiv M_p^{-1} F_p A_p^{-1}$

$$A_p = \text{Discrete pressure Poisson operator}$$
$$F_p = \text{Discrete convection-diffusion operator on pressure space}$$
$$M_p = \text{Pressure mass matrix}$$

2. **Least Squares Commutator**

$$Q_S^{-1} \equiv (BM_u^{-1}B^T)^{-1}(BM_u^{-1}FM_u^{-1}B^T)(BM_u^{-1}B^T)^{-1}$$

Comments:

- main cost: pressure Poisson solve
- PCD (1): requires (user) specification of auxiliary operators
- LSC (2): user independent

Derivation of these Methods

1. PCD: start with commutator of operators

$$\nabla(-\nu\nabla^2 + w\cdot\nabla)_p \approx (-\nu\nabla^2 + w\cdot\nabla)_u \nabla$$

Requires pressure convection-diffusion operator

Discrete analogue: $M_u^{-1}B^T M_p^{-1}F_p \approx M_u^{-1}F\, M_u^{-1}B^T$

$$\Rightarrow \quad BF^{-1}B^T \approx Q_s \equiv \underbrace{BM_u^{-1}B^T}_{\leftarrow A_p \rightarrow} F_p^{-1}M_p$$

2. LSC: *define* F_p to minimize

$$\left\| (M_u^{-1}F)(M_u^{-1}B^T) - (M_u^{-1}B^T)(M_u^{-1}F_p) \right\|_{M_u}$$

$$\Rightarrow Q_S^{-1} \equiv (BM_u^{-1}B^T)^{-1}(BM_u^{-1}FM_u^{-1}B^T)(BM_u^{-1}B^T)^{-1}$$

Properties of these Methods

Implementation:

To implement in GMRES: need action of $Q^{-1} = \begin{pmatrix} Q_F & B^T \\ 0 & -Q_S \end{pmatrix}^{-1}$

Convection-diffusion solve for Q_F^{-1}
Poisson solve(s) for Q_S^{-1}

Both approximated using "off-the-shelf" algebraic MG

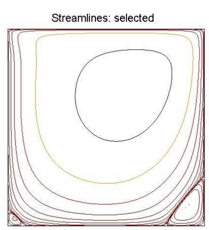
Convergence properties:

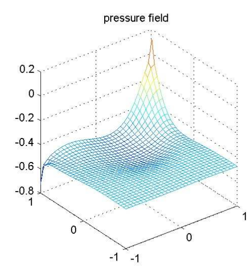
- PCD: convergence rate independent of discretization mesh size
- LSC: some dependence on mesh size, but often faster
- Both: mild dependence on Reynolds number (steady-state)
  no dependence on Re (transient)

Preliminary Performance Results

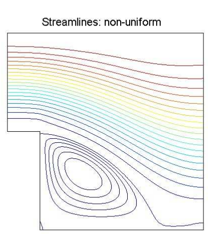
E., Silvester, & Wathen

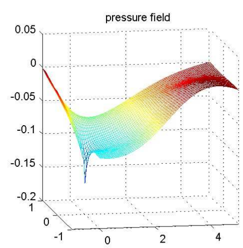
Cavity
Picard system

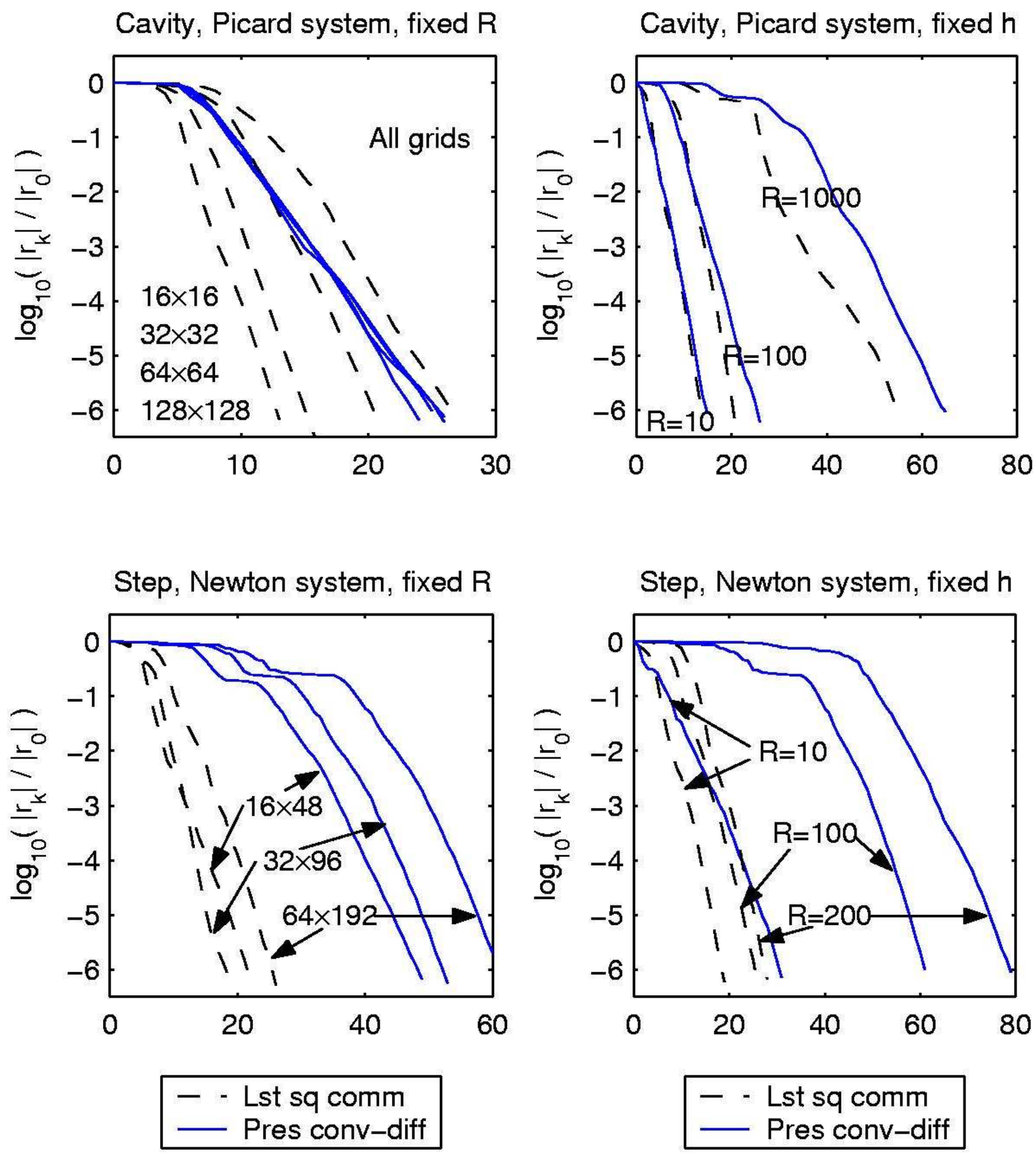




Step
Newton system







Relation to SIMPLE

Semi-Implicit Method for Pressure-Linked Equations

Patankar & Spaulding, 1972

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} = \begin{pmatrix} F & 0 \\ B & -BF^{-1}B^T \end{pmatrix} \begin{pmatrix} I & F^{-1}B^T \\ 0 & I \end{pmatrix}$$

$$\approx \begin{pmatrix} Q_F & 0 \\ B & -B\hat{F}^{-1}B^T \end{pmatrix} \begin{pmatrix} I & \hat{F}^{-1}B^T \\ 0 & I \end{pmatrix}$$

Q_F: approximate convection-diffusion solve

$\hat{F}$: diagonal part of F

N.B. Does not take convection into account

Many variants (SIMPLEC: $\hat{F} = diag(row\text{-}sum(F))$)

Benchmarking using MPSalsa

MPSalsa (Shadid, Salinger, Hennigan, Pawlowski, Smith, Wilkes,O'Rourke)

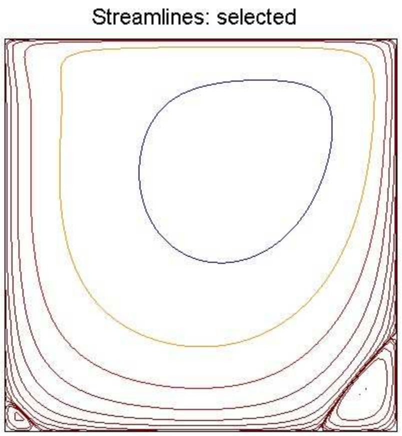
General purpose parallel code
- models low Mach number, incompressible and variable density fluid flows
- coupled with heat transport, multi-component species transport
- discretizes using biquadratic Petrov-Galerkin (Galerkin least squares) finite elements on unstructured grids
- offers Krylov subspace solvers with ILU/domain decomposition

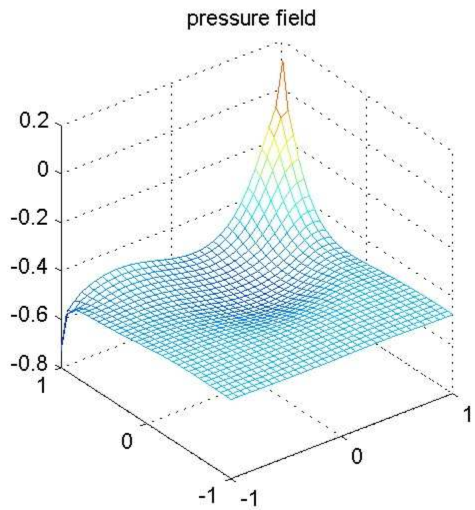
Task:
- Integrate and test block preconditioner within MPSalsa
- Build using existing Sandia software

Benchmark Problems

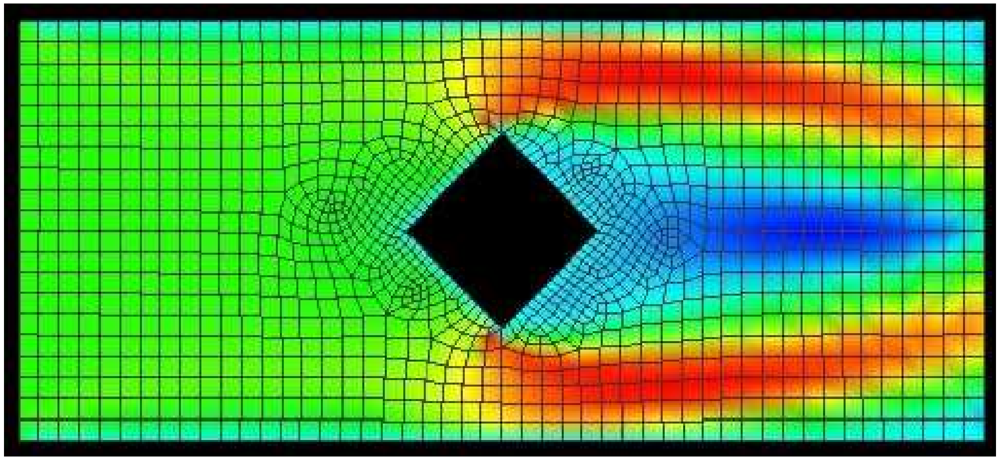
1. 2D Driven Cavity

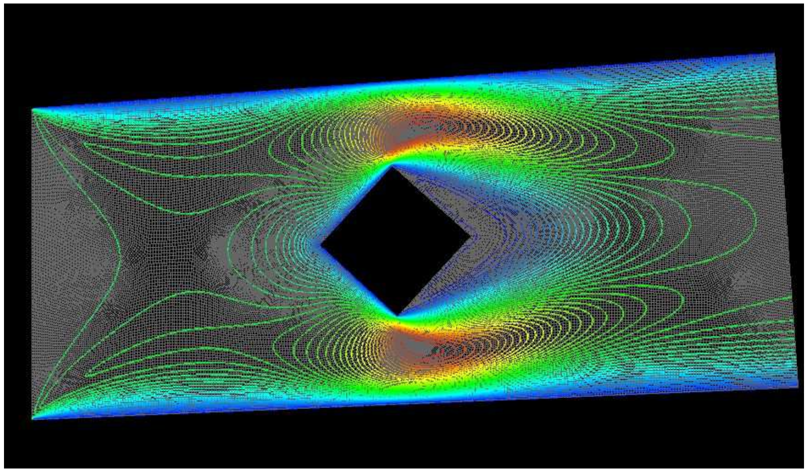
2. 3D Driven Cavity





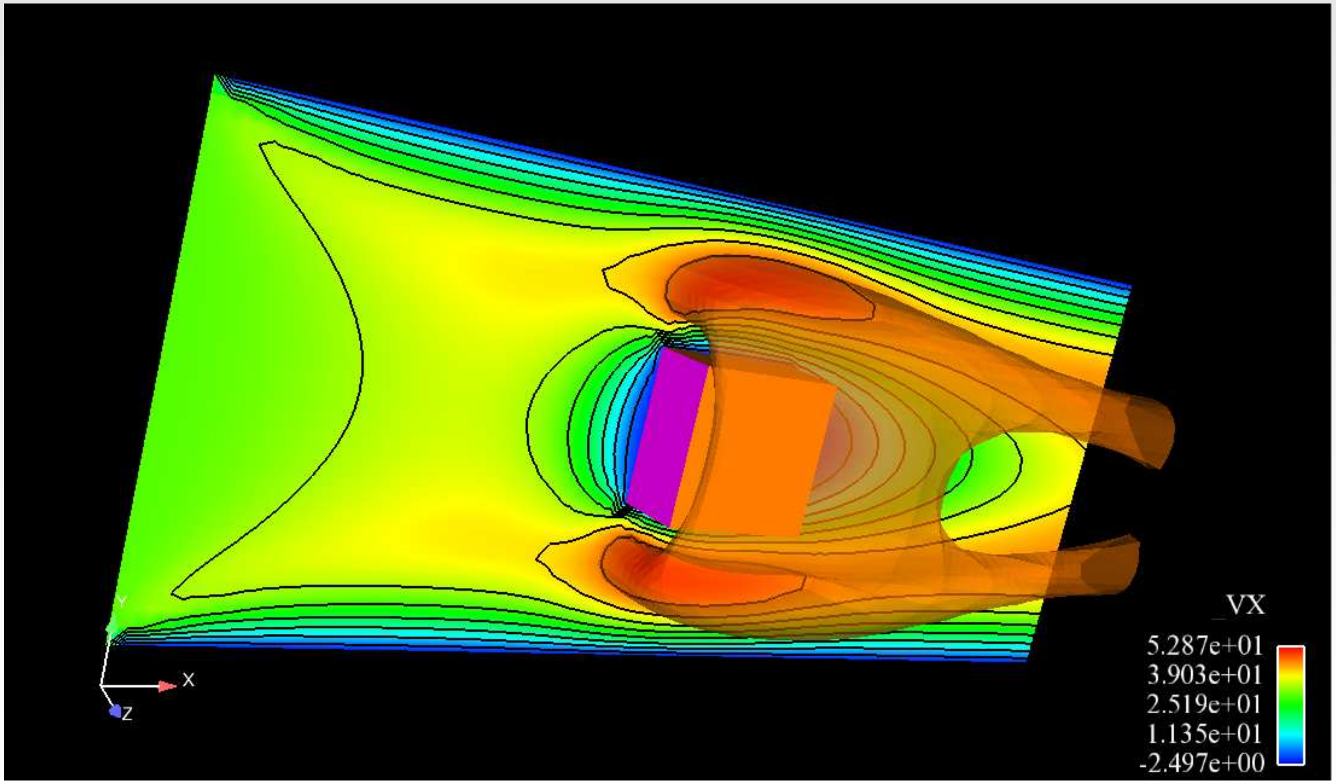
3. 2D flow over a diamond obstruction
   Inflow-outflow b.c., unstructured grid





Benchmark Problems

4. 3D flow over a cube obstruction



Criteria used in Numerical Experiments

Solving nonlinear algebraic system $\begin{pmatrix} F(u) & B^T \\ \hat{B} & -C \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$

Using Newton's method. Stop when iterate $\begin{pmatrix} u \\ p \end{pmatrix}$ satisfies

$$\Bigg\| \underbrace{\begin{pmatrix} f \\ g \end{pmatrix} - \begin{pmatrix} F(u) & B^T \\ \hat{B} & -C \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix}}_{\text{Nonlinear residual}} \Bigg\| \le 10^{-4} \left\| \begin{pmatrix} f \\ g \end{pmatrix} \right\|$$

Jacobean system: $\begin{pmatrix} F & B^T \\ \hat{B} & -C \end{pmatrix}\begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = \begin{pmatrix} r_f \\ r_g \end{pmatrix}$

Criteria used in Numerical Experiments

Solve system using Pressure Convection-Diffusion (PCD) preconditioned GMRES

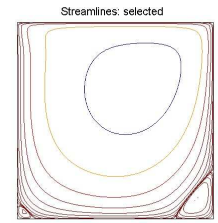
Stop GMRES iteration when

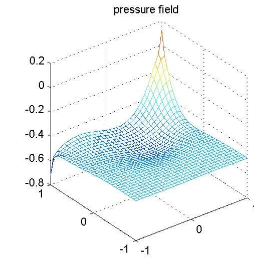
$$\left\| \begin{pmatrix} r_f \\ r_g \end{pmatrix} - \begin{pmatrix} F & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \delta u^{(k)} \\ \delta p^{(k)} \end{pmatrix} \right\| \le 10^{-5} \left\| \begin{pmatrix} r_f \\ r_g \end{pmatrix} \right\|$$

Report average $\left\{ \begin{array}{c} \text{iterations} \\ \text{CPU times} \end{array} \right\}$ over Newton run

Computations done on Sandia National Laboratories' *Institutional Computing Cluster*, with up to 64 dual Intel 3.6GHz Xenon processors with 2GB RAM each.

Results: 2D Cavity



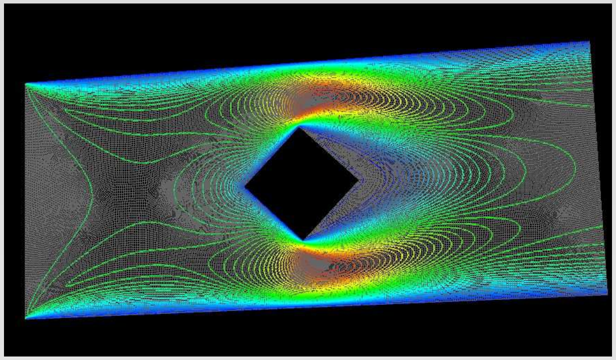


Re	Mesh size	PCD		SIMPLE		1-level DD		Procs
		Iters	Time	Iters	Time	Iters	Time	
10	64 x 64	19.4	17.2	41.8	32.9	79.4	19.4	1
	128 x 128	21.2	28.4	66.0	78.9	220.6	79.8	4
	256 x 256	23.0	69.3	104.3	229.2	467.2	619.4	16
	512 x 512	23.2	257.2	164.0	619.4	1356.8	2901.9	64
100	64 x 64	35.0	28.7	52.0	50.8	86.5	26.4	1
	128 x 128	34.9	59.5	71.8	87.9	300.3	130.2	4
	256 x 256	41.3	102.1	109.8	410.5	528.8	593.1	16
	512 x 512	41.0	345.7	169.4	941.2	NC	NC	64
1000	64 x 64	NC	NC	NC	NC	NC	NC	1
	128 x 128	126.4	570.9	142.0	1220.4	352.5	275.8	4
	256 x 256	126.6	1207.6	251.6	3494.2	839.5	2009.6	16
	512 x 512	143.2	2563.2	401.2	7598.2	NC	NC	64

Results: 3D Cavity

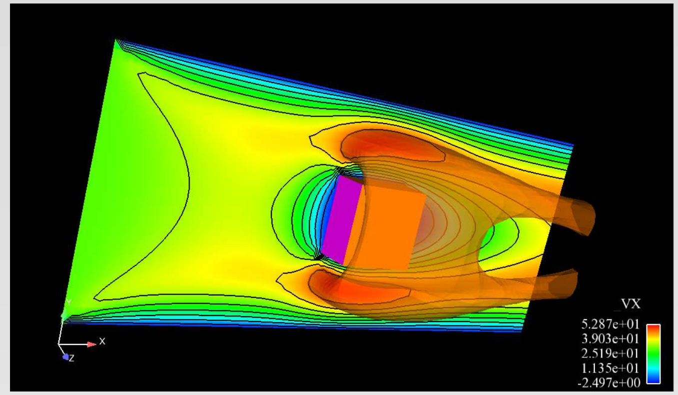
Re	Mesh size	PCD Iters	PCD Time	SIMPLE Iters	SIMPLE Time	1-level DD Iters	1-level DD Time	Procs
10	32 x 32 x 32	28.0	802.3	30.5	1205.6	67.0	634.6	1
	64 x 64 x 64	28.4	865.2	50.8	2034.1	159.8	1507.5	8
	128 x 128 x 128	31.1	1249.0	280.8	12490.5	356.2	4529.3	64
50	32 x 32 x 32	40.2	946.9	33.3	1302.6	62.2	615.5	1
	64 x 64 x 64	47.8	1061.6	52.5	2457.6	162.6	1533.2	8
	128 x 128 x 128	50.1	2101.2	291.2	14987.2	385.5	6460.9	64
100	32 x 32x 32	56.0	1232.7	40.8	1884.4	67.0	730.7	1
	64 x 64x 64	62.1	1697.8	61.6	3184.4	159.8	2131.6	8
	128 x 128 x 128	64.2	3019.2	299.1	17184.2	356.2	6953.9	64

Results: 2D Flow over Diamond Obstruction



Re	Unknowns	PCD Iters	PCD Time	SIMPLE Iters	SIMPLE Time	1-level DD Iters	1-level DD Time	Procs
10	62K	21.7	138.8	52.8	502.2	110.8	186.6	1
	256K	22.6	192.7	83.6	1203.9	282.6	1054.9	4
	1M	25.6	252.3	130.8	1845.3	890.2	6187.4	16
	4M	29.7	397.5	212.6	5834.6	NC	NC	64
25	62K	34.9	248.0	66.5	760.5	101.7	198.8	1
	256K	40.4	384.6	104.7	1920.3	273.8	1118.6	4
	1M	43.6	445.9	160.8	2985.2	864.5	6226.0	16
	4M	49.1	736.6	402.1	8241.3	NC	NC	64
40	62K	64.6	565.8	74.8	1278.7	70.4	267.2	1
	256K	68.9	975.2	113.6	2718.9	203.9	1269.3	4
	1M	72.7	1039.2	260.9	7535.0	770.0	6933.5	16
	4M	78.3	1528.6	410.1	11992.2	NC	NC	64

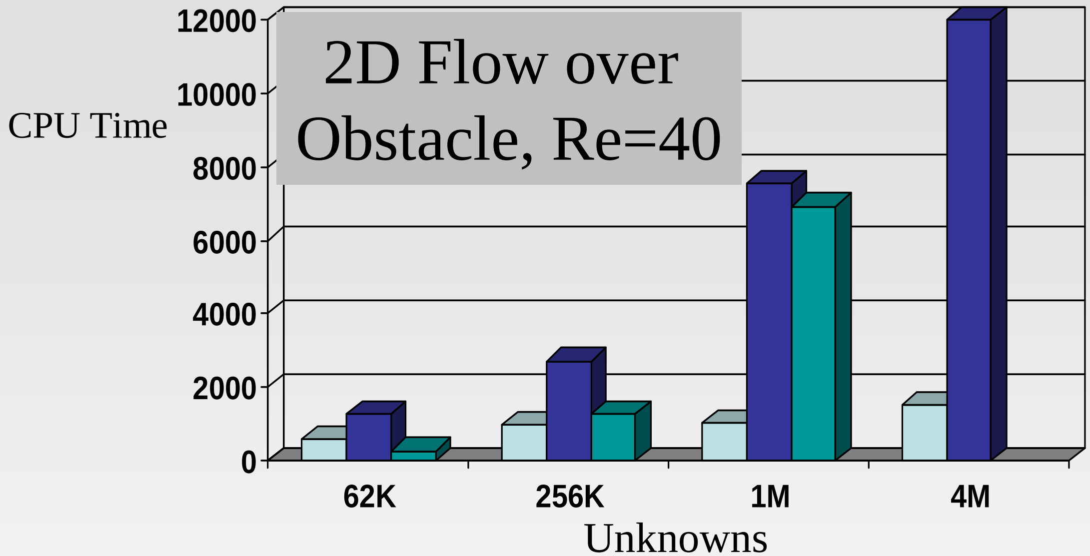
Results: 3D Flow over Cube Obstruction

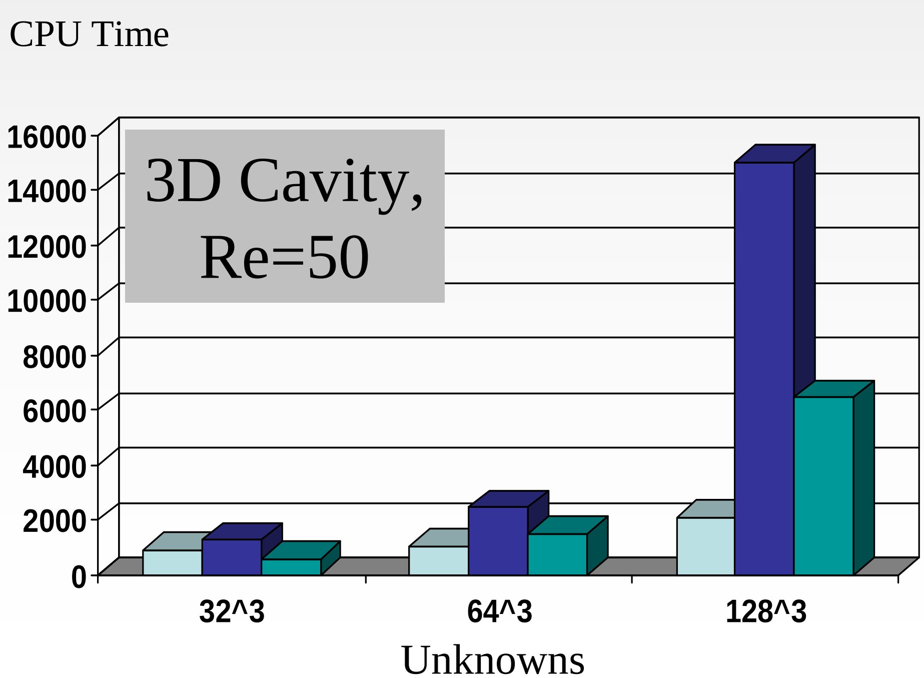


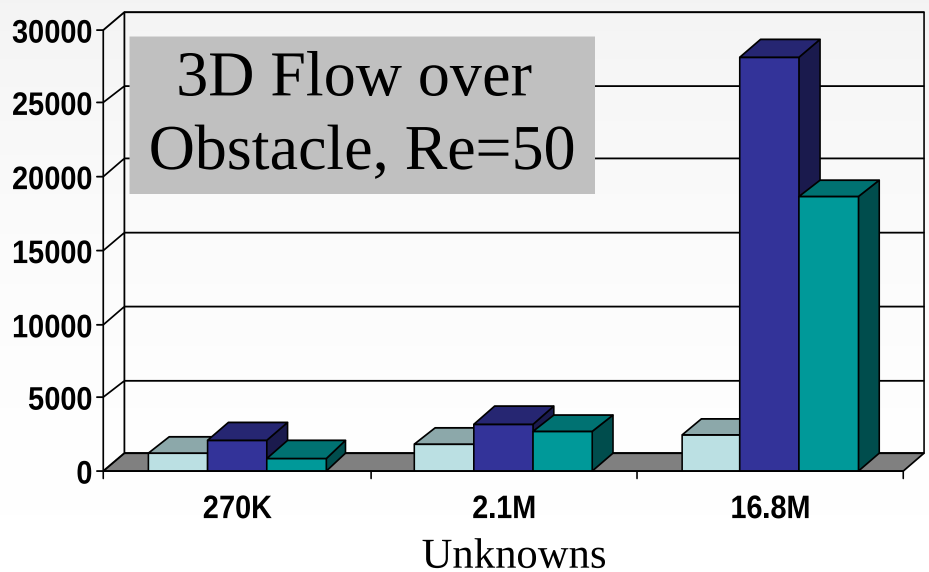
Re	Unknowns	PCD		SIMPLE		1-level DD		Procs
		Iters	Time	Iters	Time	Iters	Time	
10	270K	20.7	997.7	45.2	1897.1	67.2	859.8	1
	2.1M	21.7	1507.5	79.3	4593.2	151.2	2004.0	8
	16.8M	24.7	1997.7	118.7	19907.1	667.2	20908.0	64
50	270K	35.9	1209.7	49.2	2109.2	69.4	889.2	1
	2.1M	38.7	1797.7	84.9	3201.3	132.4	2676.1	8
	16.8M	44.7	2397.7	140.2	28156.1	637.2	18646.0	64

Graphical Depiction of these Results

- Pressure conv-diff
- Simple
- Domain decomposition







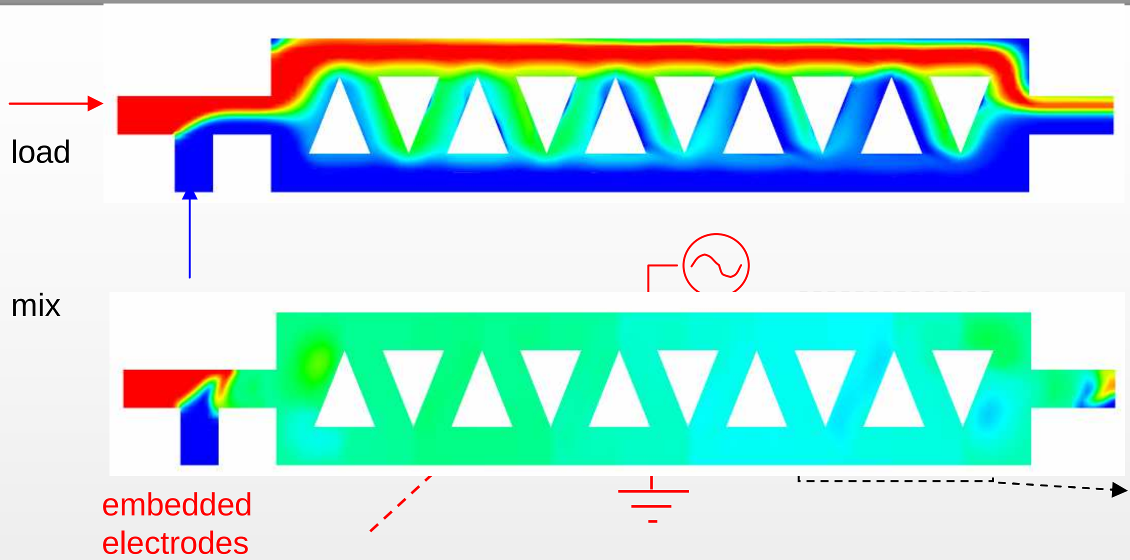
Implementation Issues

1. Solving subsidiary scalar problems (convection-diffusion and Poisson equations) using “off-the-shelf” algebraic multigrid software **ML** (smoothed aggregation).

2. Solving these systems “inexactly”.

3. Other components of the code built using Sandia tools, (Trilinos, Meros, Epetra, Aztec,CHACO, NOX), which handle nonlinear and Krylov subspace solvers and all parallelism.

Application: Topology of MicroFluidics Devices



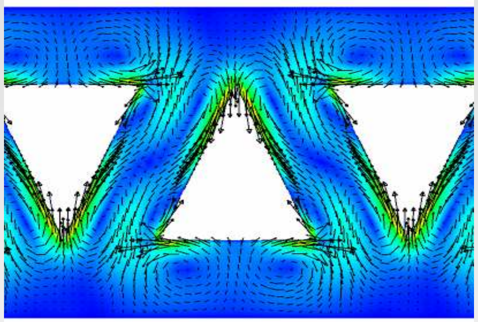
High level problem statement:

- Mix two liquids at low Re
- Flow driven by electrokinetic means: induced charge electro-osmosis (ICEO), via charge on interior obstacles
- Goal: choose shape and topology of obstructions to optimize "mixing metric"

Collaboration with SNL's Thermal/Fluid Science & Engineering Group (M. P. Kanouff, J.Templeton)

Computational Procedure

Given topology of device (38 parameters):



Electric field on obstacles obtained by solving the Laplace equation for electric potential φ, tangential component of E= ∇φ defines velocity b.c. along obstructions

Solve incompressible NS equations

Use computed velocity ***u*** to obtain mass fraction of solute

$$-D\nabla^2 m + (u \cdot \mathrm{grad}) m = 0$$

Calculate mixing metric = measure of extent of mixing

$$M = \frac{\int (m - \overline{m})^2 dV}{V}$$

Computational Procedure

Optimization loop:

Minimize *M* with respect to 38 design parameters

Optimization performed using derivative-free asynchronous parallel pattern search, via **APPSPACK** (Gray, Griffen, Hough, Kolda, Torczon)

Software environment:

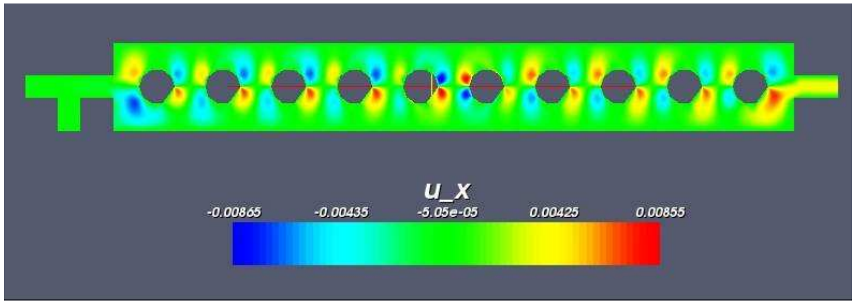
SUNDANCE (K. Long)

Results: Use PCD-Preconditioned GMRES

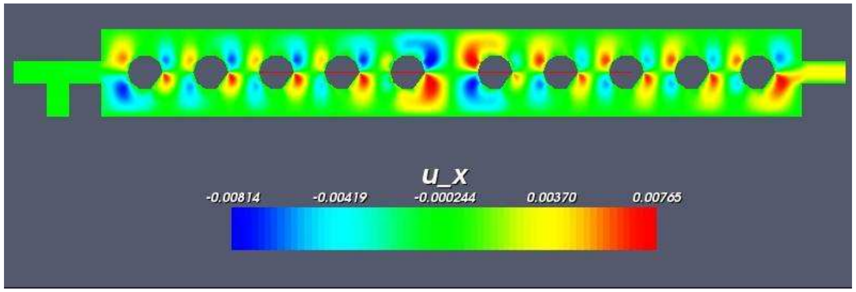
Iteration Counts	CPU time
64.0	21765.1
62.1	20831.1
67.1	21874.1
66.1	20923.9
68.2	20643.1
69.2	20173.8
60.4	20515.5
67.3	20488.9
66.3	20898.2

Examples of Flow Fields Computed

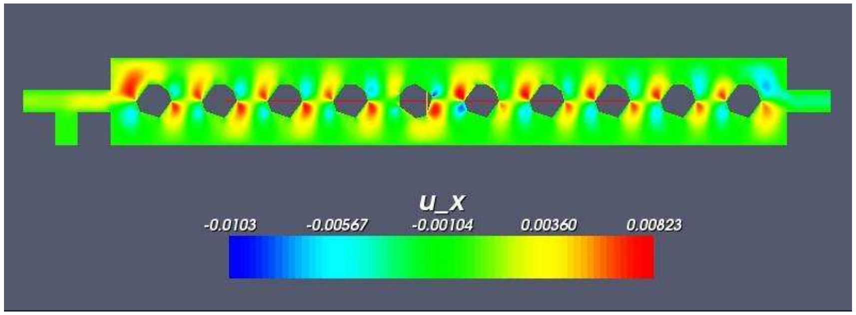
Original M = 0.0287106



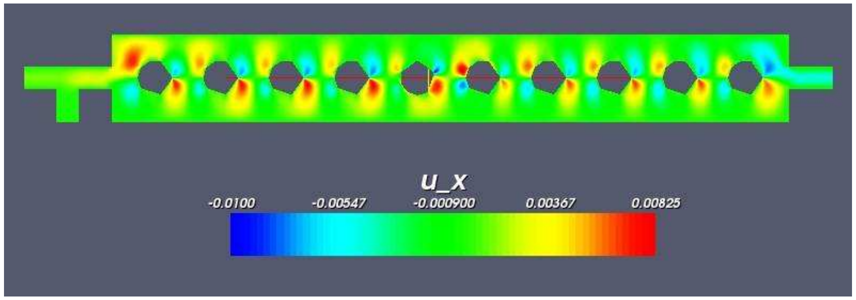
M = 0.0233216



M = 0.032451



M = 0.000811796



M = 0.000923394

Ongoing Efforts

1. Extension of these ideas to *spectral element methods*

   Build using additive Schwarz methods with *fast diagonalization methods* on subdomains

2. Use of these ideas for *stability analysis* of flows: solve

$$\begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} w \\ q \end{pmatrix} = \lambda \begin{pmatrix} M_u & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} w \\ q \end{pmatrix}$$

3. Extension of approach to handle thermal / chemical effects

   E.g. Boussinesq model$\longrightarrow$

$$\left(\begin{array}{cc:c} F_u & G & B^T \\ H & F_T & 0 \\ \hdashline B & 0 & 0 \end{array}\right) \begin{pmatrix} \delta u \\ \delta T \\ \delta p \end{pmatrix} = \begin{pmatrix} f \\ g \\ h \end{pmatrix}$$

4. Uncertainty quantification: solution algorithms for problems posed with uncertainty