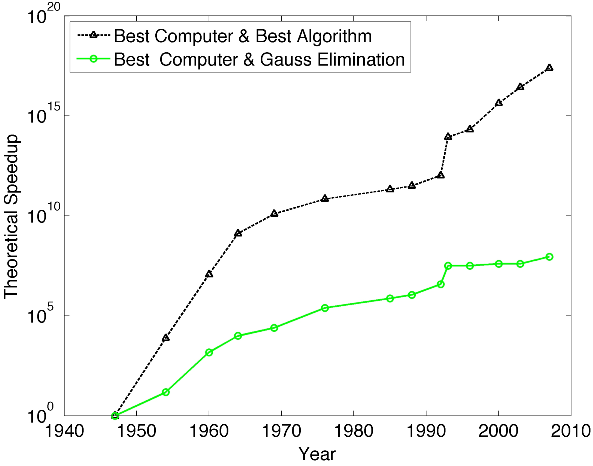
Fast Solvers for Models of Fluid Flow with Spectral Elements

P. Aaron Lott
National Institute of Standards and Technology

In collaboration with

Howard Elman
University of Maryland, College Park

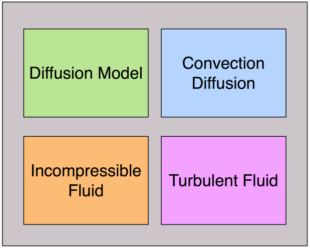
Evolution of Algorithms and Machines



Idealized speedup of solvers for 3D Poisson Equation 64^3 grid

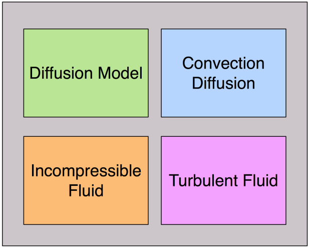
Adapted from Deville, Fischer & Mund

What is scalable?

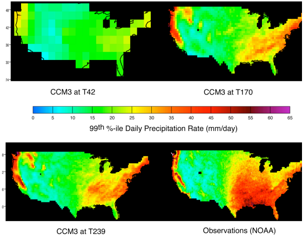


Solve more difficult problems

What is scalable?



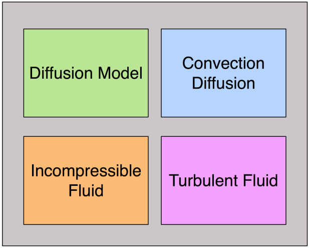
Solve more difficult problems



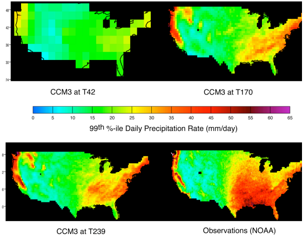
At increased resolution

Image by Duffy

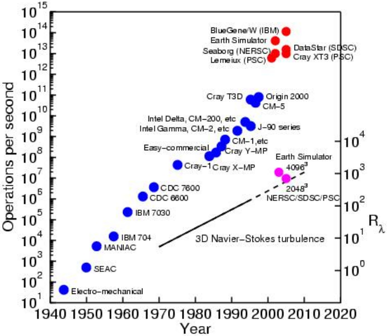
What is scalable?



Solve more difficult problems



At increased resolution
Image by Duffy



Efficiently as the number of processors increase
Image by Donzis

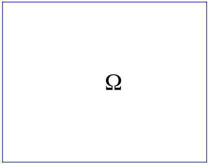
Fluid Model

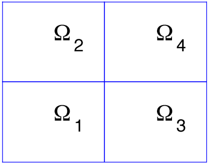
Steady Incompressible Navier-Stokes Equations

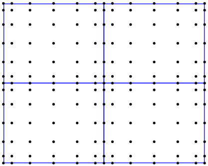
$$\left.\begin{array}{rcl} -\frac{1}{Re}\nabla^2\vec{u} + (\vec{u}\cdot\nabla)\vec{u} + \nabla p & = & f \\ \nabla\cdot\vec{u} & = & 0 \end{array}\right. \quad \text{in} \quad \Omega$$

$$\vec{u} = \vec{u}_D \quad \text{on} \quad \partial\Omega_D, \qquad \nu\frac{\partial\vec{u}}{\partial n} - \vec{n}p = 0 \quad \text{on} \quad \partial\Omega_N.$$

Spectral Element Method



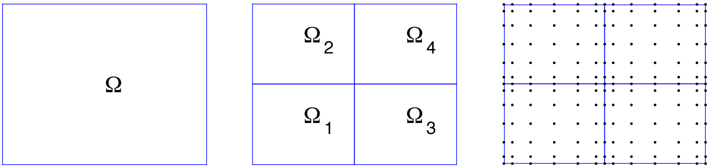




The solution is expressed via a nodal basis on each element

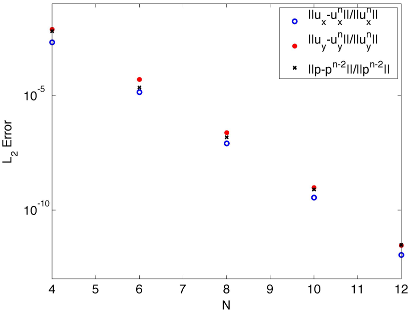
$$u_e^N(x, y) = \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} u_{ij}\pi_i(x)\pi_j(y).$$

Spectral Element Method



The solution is expressed via a nodal basis on each element

$$u_e^N(x,y) = \sum_{i=1}^{N+1}\sum_{j=1}^{N+1} u_{ij}\pi_i(x)\pi_j(y).$$



Discrete Nonlinear Block System

$$\begin{bmatrix} N(u) & -D^T \\ -D & 0 \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} Mf \\ 0 \end{pmatrix}$$

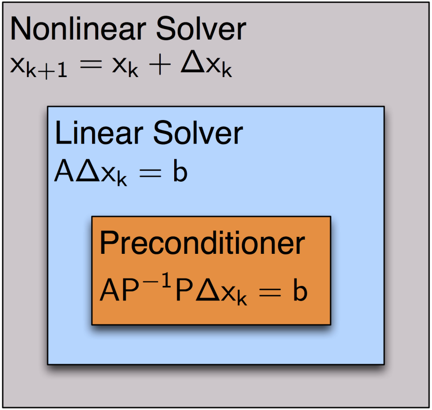
$N(u)$ - Nonlinear Convection-Diffusion

D^T - Gradient

D - Divergence

M - Identity Matrix

Overview


Nonlinear Solver
$x_{k+1} = x_k + \Delta x_k$
Linear Solver
$A\Delta x_k = b$
Preconditioner
$AP^{-1}P\Delta x_k = b$

Block Preconditioner [4]

Picard Linearization & use upper block of LU factorization

$$\underbrace{\begin{bmatrix} F & -D^T \\ -D & 0 \end{bmatrix}}_{A} = \begin{bmatrix} I & 0 \\ -DF^{-1} & I \end{bmatrix} \begin{bmatrix} F & -D^T \\ 0 & -S \end{bmatrix}$$

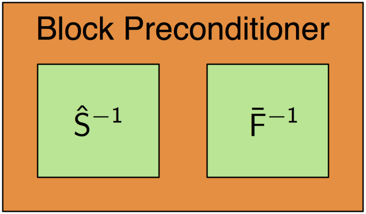
Block Preconditioner [4]

Picard Linearization & use upper block of LU factorization

$$\underbrace{\begin{bmatrix} F & -D^T \\ -D & 0 \end{bmatrix}}_{A} = \begin{bmatrix} I & 0 \\ -DF^{-1} & I \end{bmatrix} \begin{bmatrix} F & -D^T \\ 0 & -S \end{bmatrix}$$

Approximate F and S to make computationally efficient [2]

$$\underbrace{\begin{bmatrix} \bar{F} & -D^T \\ 0 & -\hat{S} \end{bmatrix}}_{P}$$



$\hat{S}$: Least-Squares Commutator [1]

Idea: Form a commutator $\epsilon = L_v \nabla - \nabla L_p$

$$\epsilon_h = (M^{-1}F)(M^{-1}D^T) - (M^{-1}D^T)(M_p^{-1}F_p)$$

$\hat{S}$: Least-Squares Commutator [1]

Idea: Form a commutator $\epsilon = L_v \nabla - \nabla L_p$

$$\epsilon_h = (M^{-1}F)(M^{-1}D^T) - (M^{-1}D^T)(M_p^{-1}F_p)$$

If ϵ_h is small, and we pre-multipy by $DF^{-1}M$ and post-multiply by $F_p^{-1}M_p$, then

$$S = DF^{-1}D^T = (DF^{-1}M)(M^{-1}F)(M^{-1}D^T) \approx DM^{-1}D^T F_p M_p.$$

$\hat{S}$: Least-Squares Commutator [1]

Idea: Form a commutator $\epsilon = L_v \nabla - \nabla L_p$

$$\epsilon_h = (M^{-1}F)(M^{-1}D^T) - (M^{-1}D^T)(M_p^{-1}F_p)$$

If ϵ_h is small, and we pre-multipy by $DF^{-1}M$ and post-multiply by $F_p^{-1}M_p$, then

$$S = DF^{-1}D^T = (DF^{-1}M)(M^{-1}F)(M^{-1}D^T) \approx DM^{-1}D^T F_p M_p.$$

Approximate F_p using Least-Squares to minimize ϵ_h

$\hat{S}$: Least-Squares Commutator [1]

Idea: Form a commutator $\epsilon = L_v \nabla - \nabla L_p$

$$\epsilon_h = (M^{-1}F)(M^{-1}D^T) - (M^{-1}D^T)(M_p^{-1}F_p)$$

If ϵ_h is small, and we pre-multipy by $DF^{-1}M$ and post-multiply by $F_p^{-1}M_p$, then

$$S = DF^{-1}D^T = (DF^{-1}M)(M^{-1}F)(M^{-1}D^T) \approx DM^{-1}D^T F_p M_p.$$

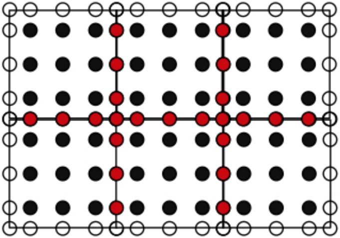
Approximate F_p using Least-Squares to minimize ϵ_h

$$\hat{S} := (DM^{-1}D^T)(DM^{-1}FM^{-1}D^T)^{-1}(DM^{-1}D^T)$$

$$\hat{S}^{-1} = \underbrace{(DM^{-1}D^T)^{-1}}_{\textit{Poisson Solve}}(DM^{-1}FM^{-1}D^T)\underbrace{(DM^{-1}D^T)^{-1}}_{\textit{Poisson Solve}}$$

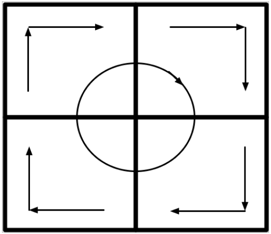
Structure of F block

$$
\begin{aligned}
F_x^e &= \epsilon(\hat{M} \otimes \hat{A}) + W_x^e(\hat{M} \otimes \hat{C}) \\
F_y^e &= \epsilon(\hat{A} \otimes \hat{M}) + W_y^e(\hat{C} \otimes \hat{M}) \\
F &= F_x + F_y
\end{aligned}
$$



$$
\begin{bmatrix}
F_{II}^1 & 0 & \dots & 0 & F_{I\Gamma}^1 \\
0 & F_{II}^2 & 0 & \dots & F_{I\Gamma}^2 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \dots & F_{II}^E & F_{I\Gamma}^E \\
0 & 0 & \dots & 0 & F_S
\end{bmatrix}
\begin{pmatrix}
u_{I^1} \\ u_{I^2} \\ \vdots \\ u_{I^E} \\ u_\Gamma
\end{pmatrix}
=
\begin{pmatrix}
\hat{b}_{I^1} \\ \hat{b}_{I^2} \\ \vdots \\ \hat{b}_{I^E} \\ g_\Gamma
\end{pmatrix}
$$

$\bar{F}$: Exploit Tensor Product Structure [3]

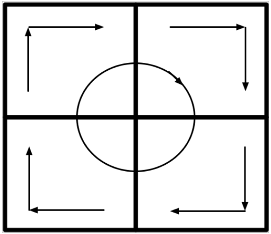


$$\begin{bmatrix} \bar{F}^1_{II} & 0 & \dots & 0 & \bar{F}^1_{I\Gamma} \\ 0 & \bar{F}^2_{II} & 0 & \dots & \bar{F}^2_{I\Gamma} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{F}^E_{II} & \bar{F}^E_{I\Gamma} \\ 0 & 0 & \dots & 0 & \bar{F}_S \end{bmatrix}$$

$$\bar{F}_S = \sum_{e=1}^{E} (\bar{F}^e_{\Gamma\Gamma} - \bar{F}^e_{\Gamma I} \, {\bar{F}^e_{II}}^{-1} \, \bar{F}^e_{I\Gamma})$$

$$\bar{F}^e_{II} = \hat{M} \otimes \hat{F}_x + \hat{F}_y \otimes \hat{M}$$

$\bar{F}$: Exploit Tensor Product Structure [3]



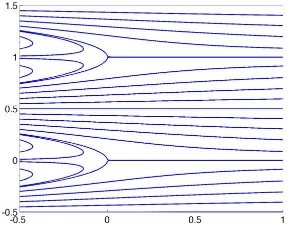
$$\begin{bmatrix} \bar{F}^1_{II} & 0 & \dots & 0 & \bar{F}^1_{I\Gamma} \\ 0 & \bar{F}^2_{II} & 0 & \dots & \bar{F}^2_{I\Gamma} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{F}^E_{II} & \bar{F}^E_{I\Gamma} \\ 0 & 0 & \dots & 0 & \bar{F}_S \end{bmatrix}$$

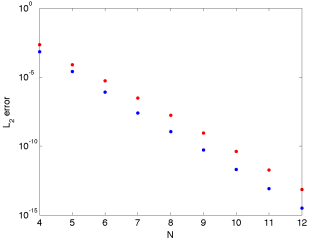
$$\bar{F}_S = \sum_{e=1}^{E} (\bar{F}^e_{\Gamma\Gamma} - \bar{F}^e_{\Gamma I} \bar{F}^{e\;-1}_{II} \bar{F}^e_{I\Gamma})$$

$$\bar{F}^e_{II} = \hat{M} \otimes \hat{F}_x + \hat{F}_y \otimes \hat{M}$$

$$\bar{F}^{e\;-1}_{II} = \underbrace{\tilde{M}(V_y \otimes V_x)(\Lambda_y \otimes I + I \otimes \Lambda_x)^{-1}(V_y^{-1} \otimes V_x^{-1})\tilde{M}}_{\text{Diagonalized via 1D operators!}}$$

Nonlinear Solver/Polynomial Degree case study: Kovasznay Flow



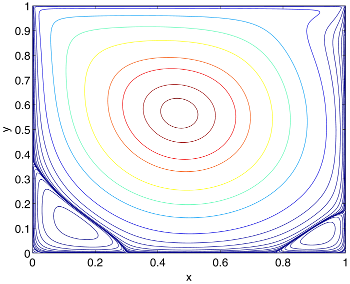


N	Picard steps	Linear steps	$\bar{F}^{-1}$ steps	$\hat{S}^{-1}$ steps
4	18	18	6	35
8	21	21	9	109
12	30	21	12	256

N	Newton steps	Linear steps	$\bar{F}^{-1}$ steps	$\hat{S}^{-1}$ steps
4	5	23	6	45
8	5	35	9	146
12	6	45	12	242

E=12, Re=40, $\bar{F}$ and $\hat{S}$ inexact $\tau = 10^{-4}$

Element & Reynolds number dependence case study: Lid Driven Cavity



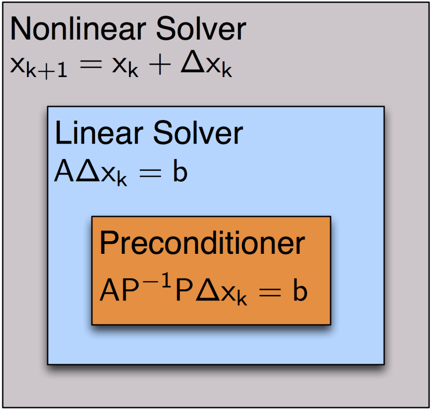
E	Newton steps	Linear steps	$\bar{F}^{-1}$ steps	$\hat{S}^{-1}$ steps
16	5	28	17	47
64	4	24	42	106
256	5	25	115	248
1024	5	31	316	472

N=4, Re=100, $\bar{F}$ and $\hat{S}$ inexact $\tau = 10^{-4}$

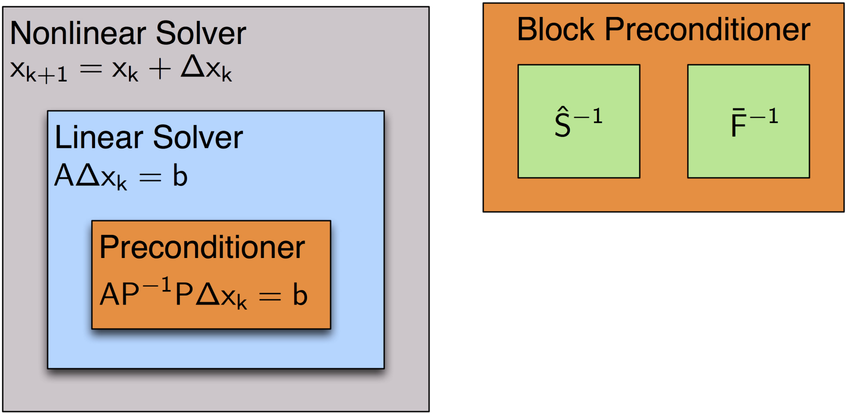
Re	Newton steps	Linear steps	$\bar{F}^{-1}$ steps	$\hat{S}^{-1}$ steps
10	4	13	113	200
100	5	25	117	200
1000	6	91	129	200

N=4, E=256, $\bar{F}$ and $\hat{S}$ inexact $\tau = 10^{-4}$

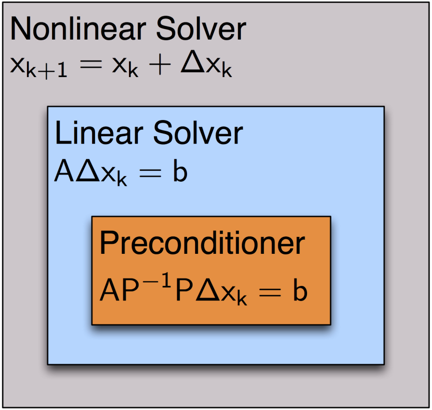
Review

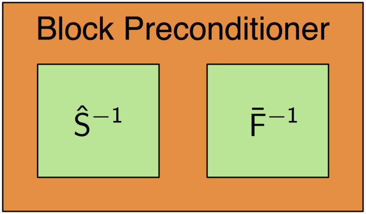

Nonlinear Solver
$x_{k+1} = x_k + \Delta x_k$
Linear Solver
$A \Delta x_k = b$
Preconditioner
$AP^{-1} P \Delta x_k = b$

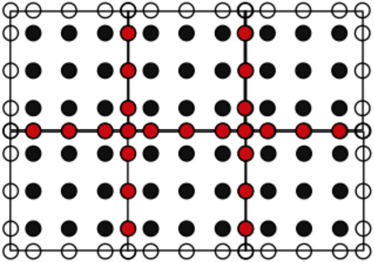
Review


Nonlinear Solver
$x_{k+1} = x_k + \Delta x_k$
Linear Solver
$A \Delta x_k = b$
Preconditioner
$A P^{-1} P \Delta x_k = b$
Block Preconditioner
$\hat{S}^{-1}$
$\bar{F}^{-1}$

Review


Nonlinear Solver
$x_{k+1} = x_k + \Delta x_k$
Linear Solver
$A \Delta x_k = b$
Preconditioner
$AP^{-1}P\Delta x_k = b$


Block Preconditioner
$\hat{S}^{-1}$
$\bar{F}^{-1}$



Conclusions

Summary

- Development of fast steady flow solver for SEM
- Based on block preconditioner using LSC & DD
- Displays mild dependence on mesh size and Re

Conclusions

Summary

- Development of fast steady flow solver for SEM
- Based on block preconditioner using LSC & DD
- Displays mild dependence on mesh size and Re

Current Work

- Accelerating $\hat{S}$ Poisson Solves with DD
- Incorporating Coarse Grid Preconditioner for $\bar{F}$
- Block Fast Diagonalization for Newton Linearization
- Thermosolutal Convection with Jeff McFadden @ NIST
- Global Climate Models with Kate Evans @ ORNL

References

H. C. ELMAN, V. E. HOWLE, J. SHADID, D. SILVESTER, AND R. TUMINARO, *Least squares preconditioners for stabilized discretizations of the Navier-Stokes equations*, SIAM Journal on Scientific Computing, 30 (2007), pp. 290–311.

H. C. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

P. A. LOTT AND H. C. ELMAN, *Fast iterative solvers for convection-diffusion systems with spectral elements*, Numerical Methods in Partial Differential Equations, (To Appear), p. DOI: 10.1002/num.20518.

———, *Fast iterative solvers for incompressible Navier-Stokes equations with spectral elements*, In preparation.