Newton-Krylov-Schwarz for Coupled Multi-physics Problems

Xiao-Chuan Cai

Department of Computer Science
University of Colorado Boulder
cai@cs.colorado.edu

Joint work with A. Barker, C. Yang, R. Chen, Y. Wu, X. Li, T. Zhao, and D. Keyes

Outline

- Background and motivations

- Newton-Krylov and (linear/nonlinear) Schwarz algorithms

- Examples

  - Compressible Euler in 2D (climate modeling)

  - Incompressible Navier-Stokes in 3D (flow around a complete wind turbine)

  - Fluid-structure interaction in 3D (blood flow in artery)

- Final remarks

Background and motivations

- Steady state nonlinear PDE: One "has" to solve a nonlinear problem

- Unsteady nonlinear PDE:

  One may not have to solve any nonlinear problems. Popular approaches for unsteady nonlinear problems

  - explicit methods
  - decoupled methods (operator splitting)
  - linearly implicit methods

  But with nonlinearly implicit methods, one may

  - accurately capture nonlinear coupling between components
  - conserve more physical quantities
  - obtain long(er) time integration

  But you have to solve nonlinear systems $\longrightarrow$ this talk

Nonlinearly implicit methods

Two desirable properties of a nonlinear solver for solving $F(x) = 0$

- Parallelism and scalability: domain decomposition provides parallelism, multilevel provides "processor-scalabilities"

- Robustness (for multi-physics): The algorithm converges under undesirable conditions
  - F (or $F^{'}$) often has some of the following features: large dimension, sparse, highly nonlinear, non-elliptic, non-symmetric
  - Very often, when the solution is physically interesting the function F is mathematically bad $-$ locally nonsmooth (sharp gradient, boundary or internal layers, etc)

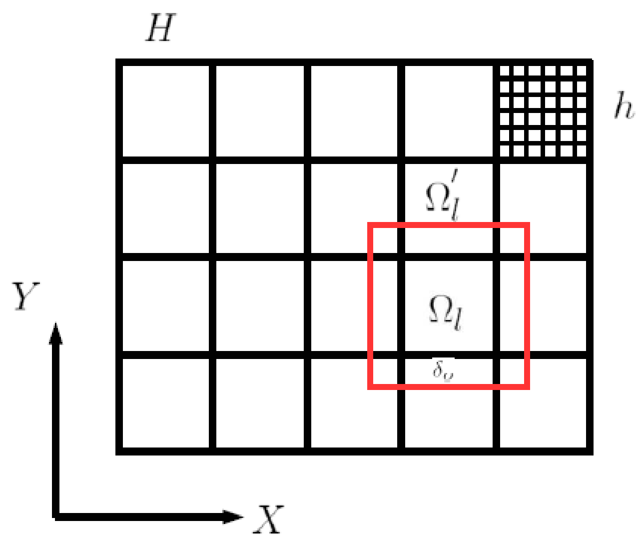
Existing nonlinear solvers

- Nonlinear iterative methods
  - All (?) linear iterative methods can be modified for non-linear problems (Richardson, J, GS, ..., CG, GMRES, ... , Multigrid, Domain Decomposition)
  - Newton type methods for $F(x) = 0$ (inexact Newton, Jacobian-Free Newton, Semi-smooth Newton, ...)
  - Preconditioned Newton (left-preconditioned Newton $G(F(x)) = 0$, right-preconditioned Newton $F(G(x)) = 0$)

- Most of the nonlinear solvers are well-studied for the class of monotone elliptic problems. The applicability ranges of the solvers go far beyond the class of elliptic equations, but their numerical/mathematical behavior are largely unknown

Focus for rest of talk – overlapping Schwarz

- Schwarz as a nonlinear solver
- Schwarz as a linear preconditioner
- Schwarz as a nonlinear preconditioner



Schwarz as a nonlinear solver

- The original Schwarz alternating method is a linear solver, a nonlinear solver, a discretization scheme, an optimization problem solver, a multiphysics coupling method, ...

- Theory available for elliptic problems (P. L. Lions 87, 88, and others)

- Additive version (Cai and Dryja 94, Dryja and Hackbusch 97)

- In general, Schwarz is not too good an iterative solver in practice (lack of robustness, slow convergence)

Schwarz as a linear preconditioner in a nonlinear solver

- Newton-Krylov-Schwarz (Cai, Gropp, Keyes, Tidriri, 94): NKS

- The default nonlinear solver in PETSc

- Newton: Inexact Newton, semi-smooth Newton, with line-search, trust region, different stopping conditions

- Krylov: GMRES, rGMRES, CG, ...

- Schwarz: Additive, multiplicative, restricted, hybrid, multi-level versions (V-, W-, F-cycle, ...)

- Tested for several classes of applications
  - Navier-Stokes, Euler, potential flows (low Mach and transonic)
  - MHD (resistive Hall, reconnection)
  - Radiation difffusion (black body radiation in optically thick medium)
  - Fluid-structure interaction (blood in artery)
  - PDE constrained optimization problems (fluid control)
  - Inverse problems (source recover)
  - Nonlinear eigenvalue problems (with Jacobi-Davidson-Schwarz)
  - ...

Some references

- X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, *Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246-265

- M. Paraschivoiu, X.-C. Cai, M. Sarkis, D. P. Young, and D. Keyes, *Multi-domain multi-model formulation for compressible flows: conservative interface coupling and parallel implicit solvers for 3D unstructured meshes*, AIAA Paper 99-0784, 1999

- X.-C. Cai and D. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183-200

- S. Ovtchinnikov and X.-C. Cai, *One-level Newton-Krylov-Schwarz algorithm for unsteady nonlinear radiation diffusion problem*, Numer. Lin. Alg. Applic., (2004), pp. 867-881

- F.-N. Hwang and X.-C. Cai, *A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations*, J. Comput. Phys., 204 (2005), pp. 666-691

- S. Ovtchinnikov, F. Dobrian, X.-C. Cai, and D. Keyes, *Additive Schwarz-based fully coupled implicit methods for resistive Hall magnetohydrodynamic problems*, J. Comput. Phys., 225 (2007), pp. 1919-1936

- A. Barker and X.-C. Cai, *Two-level Newton and hybrid Schwarz preconditioners for fluid-structure interaction*, SIAM J. Sci. Comput., 32 (2010), pp. 2395-2417

- X.-C. Cai and X. Li, *Inexact Newton methods with nonlinear restricted additive Schwarz preconditioning for problems with high local nonlinearity*, SIAM J. Sci. Comput., 33 (2011), pp. 746-762

Schwarz as a nonlinear preconditioner

- Two types of nonlinear problems
  - Problems with global nonlinearity (nonlinear elliptic equations, for example)
  - Problems with global nonlinearity plus some local, often high, nonlinearities (boundary layers, shock waves, etc)
- Global NKS is good at removing the global nonlinearity
- Nonlinear Schwarz, if used wisely, can remove the local nonlinearities (nonlinear elimination)
- A good strategy is to combine "Nonlinear Schwarz" with "Global NKS" (one serves as a preconditioner for the other)

Newton and subspace Newton

Suppose $x^{(k)}$ is the current approximate solution, a new approximate solution is defined as

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} s^{(k)}$$

- $\lambda^{(k)}$ is the step length computed using a linesearch
- $s^{(k)}$ is a good search direction if a non-zero $\lambda^{(k)}$ can be found such that
$$\frac{1}{2}\|F(x^{(k+1)}\|^2 \leq \frac{1}{2}\|F(x^{(k)}\|^2 - \alpha\lambda^{(k)}(s^{(k)})^T JF(x^{(k)})$$

The method is called a Newton's method if $s^{(k)}$ is computed, somehow, from the (preconditioned) Jacobian equation

$$M^{-1}F^{'}(x^{(k)})s^{(k)} = M^{-1}F(x^{(k)})$$

Subspace Newton

- Very often in a large nonlinear system, only a small subset of the functions are highly nonlinear (Example: A boundary layer or a shock wave in a large computational domain, the number of equations associated with the shock could be less than 1% of the total number of equations)

- For example

$$\begin{cases} F_1(x_1, x_2) = 0 \\ F_2(x_1, x_2) = 0 \end{cases}$$

  Eliminate (implicit function theorem) the small bad component x_2

$$x_2 = G(x_1)$$

  The left-over is a slightly smaller and easier to solver nonlinear system

$$F_1(x_1, G(x_1)) = 0$$

- It is not easy to identify the bad component, and the method doesn't work too well in practice when both local and global nonlinearities present in the problem

Nonlinearly preconditioned Newton methods

- References: Cai and Keyes, SISC 2002, Cai, Keyes, and Marcinkowski, IJNMF 2002, Hwang and Cai, JCP 2005, Cai and Li, SISC 2011

- Consider a nonlinear system (x_1 and x_2 may have overlapping components)

$$\begin{cases} F_1(x_1, x_2) = 0 \\ F_2(x_1, x_2) = 0 \end{cases}$$

- Nonlinear Schwarz preconditioning: Let T_1 and T_2 be the solutions of

$$F_1(x_1 - T_1, x_2) = 0 \text{ and } F_2(x_1, x_2 - T_2) = 0$$

- A nonlinearly preconditioned system

$$T_1(x_1, x_2) + T_2(x_1, x_2) = 0,$$

  which is solved by a global Newton's method (ASPIN)

- The two systems have the same solution, and the second one often has better conditioning

Nonlinearly preconditioned Newton methods

- In ASPIN, both local and global Newton are used to remove local and global nonlinearities

- This can be regarded as a left preconditioned Newton since the nonlinear function is modified by the local Newton

- The solution is not modified (proof exists for elliptic problems)

- One- and two-level additive versions available; other versions are yet to be studied

Some Examples Using NKS

SWE in Curvilinear Coordinate on Cubed-sphere

$$\frac{\partial Q}{\partial t} + \frac{1}{\Lambda}\frac{\partial(\Lambda F)}{\partial x} + \frac{1}{\Lambda}\frac{\partial(\Lambda G)}{\partial y} + S = 0, \quad (x, y) \in [-\pi/4, \pi/4]^2 \text{ (SWE)}$$

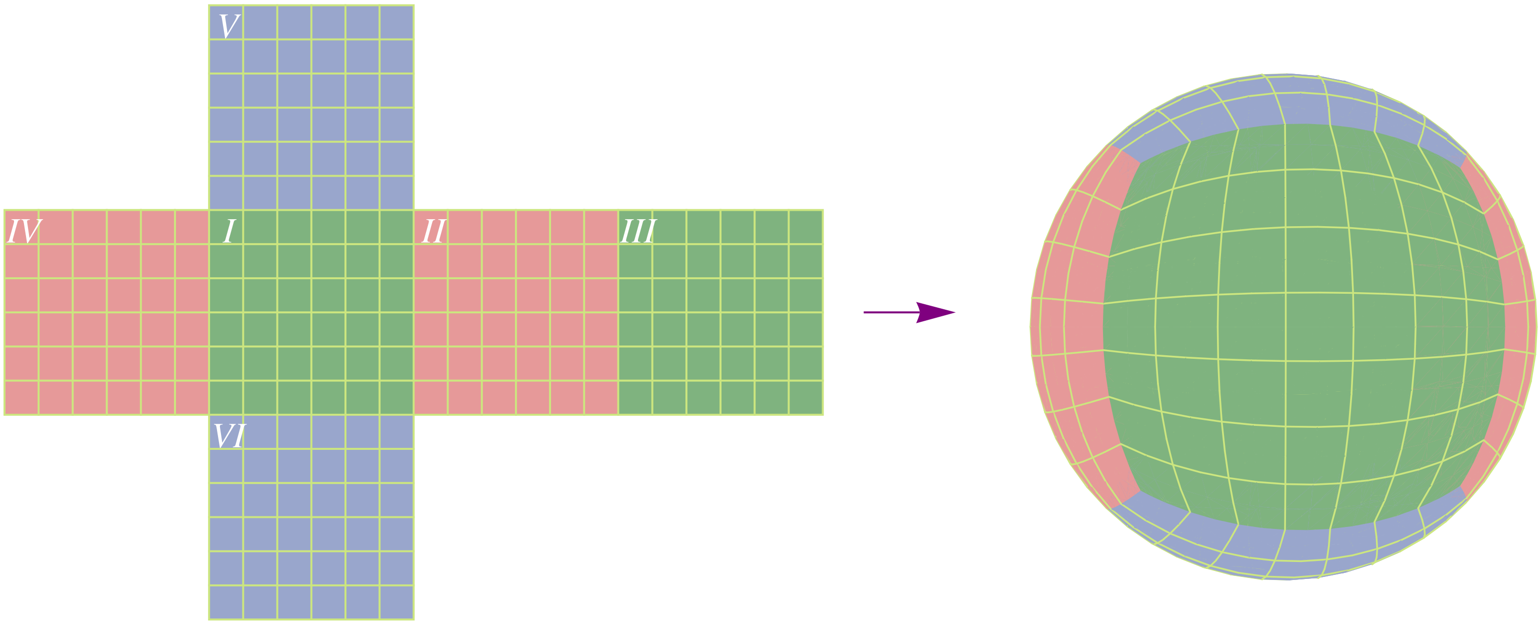
$$Q = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \; F = \begin{pmatrix} hu \\ huu + \frac{1}{2}gg^{11}h^2 \\ huv + \frac{1}{2}gg^{12}h^2 \end{pmatrix}, \; G = \begin{pmatrix} hv \\ huv + \frac{1}{2}gg^{12}h^2 \\ hvv + \frac{1}{2}gg^{22}h^2 \end{pmatrix}, \; S = \begin{pmatrix} 0 \\ S_1 \\ S_2 \end{pmatrix}$$

with source terms

$$S_1 = \Gamma^1_{11}(huu) + 2\Gamma^1_{12}(huv) + f\Lambda\left(g^{12}hu - g^{11}hv\right) + gh\left(g^{11}\frac{\partial b}{\partial x} + g^{12}\frac{\partial b}{\partial y}\right)$$

$$S_2 = 2\Gamma^2_{12}(huv) + \Gamma^2_{22}(hvv) + f\Lambda\left(g^{22}hu - g^{12}hv\right) + gh\left(g^{12}\frac{\partial b}{\partial x} + g^{22}\frac{\partial b}{\partial y}\right)$$

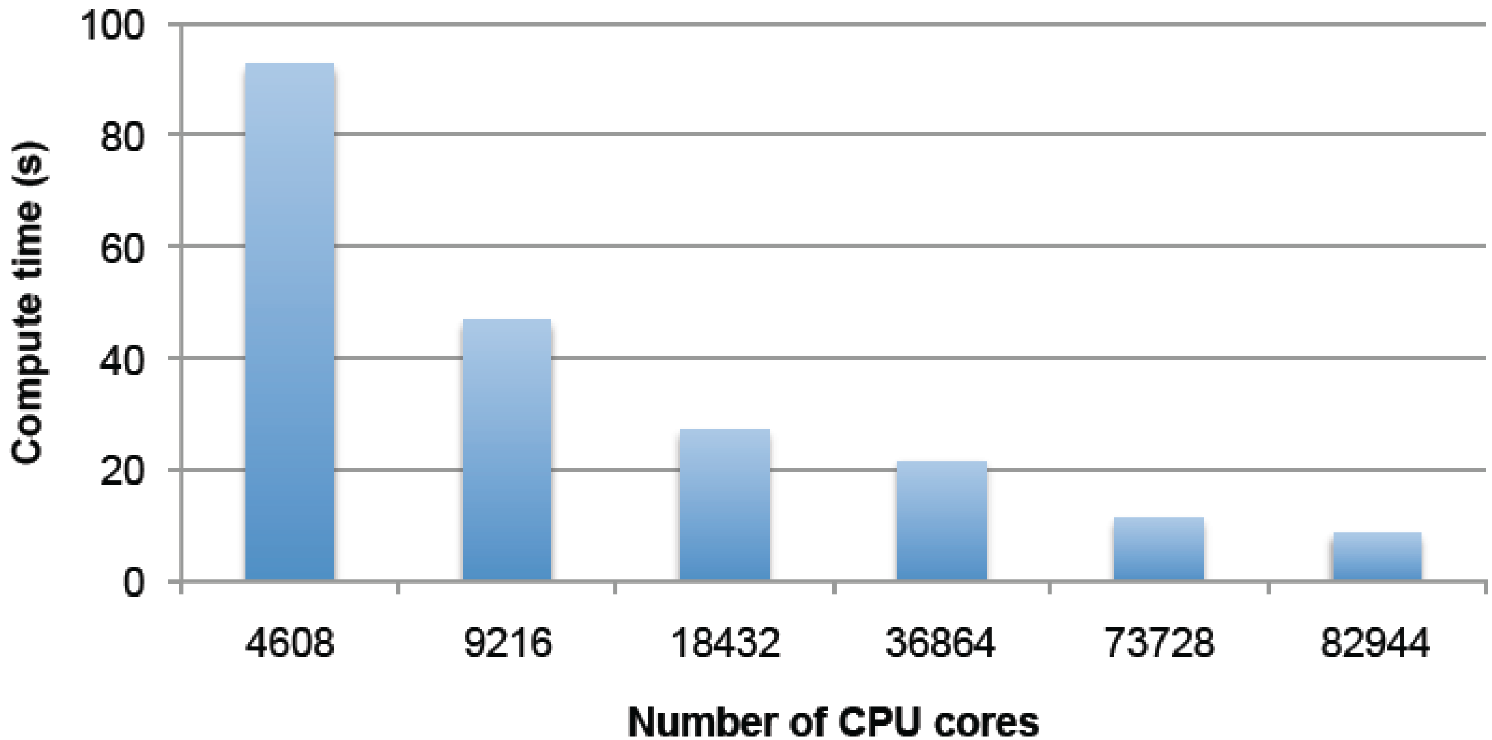
The Cubed-sphere



- Sadourny, MWR 1972
- Ronchi, Iacono, and Paolucci, JCP 1996
- Rancic, Purser, and Mesinger, QJRMS 1996
- Nair et al 2005, Rossmanith et al 2004, Putman et al 2007...

Strong-scaling tests: Compute Time

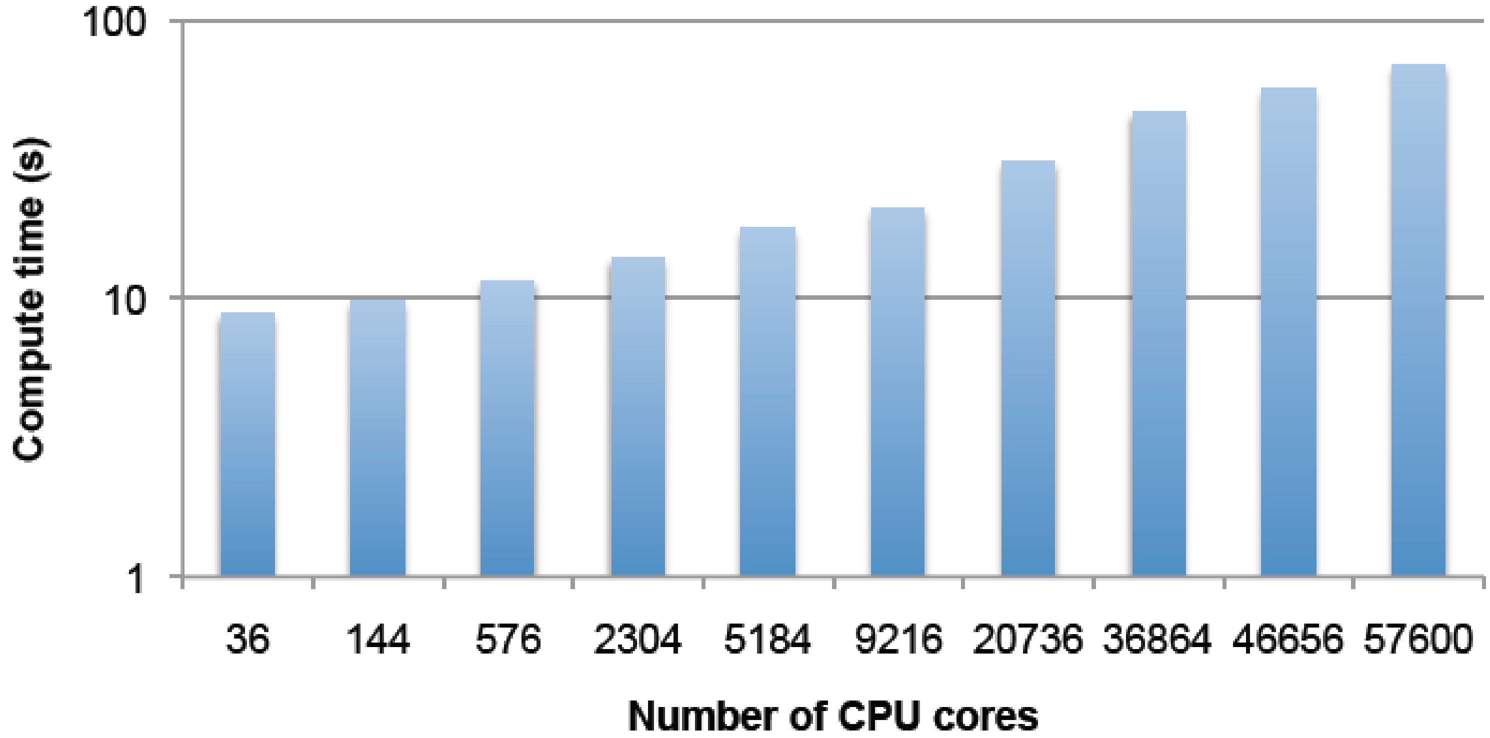
Fixed mesh: $6144 \times 6144 \times 6$ (DOF=0.68B)



From 4608 to 82944 cores, parallel eff. = 60%

Weak-scaling Tests: Compute Time

Max mesh: $10240 \times 10240 \times 6$ (DOF=1.8B)



The compute time increases 6.3X as the # of cores increases 1600X

2D compressible Euler model

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial z} + S = 0,$$

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho w \\ \rho\theta \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u w \\ \rho u \theta \end{pmatrix}, G = \begin{pmatrix} \rho w \\ \rho w u \\ \rho w^2 + p \\ \rho w \theta \end{pmatrix}, S = \begin{pmatrix} 0 \\ 0 \\ \rho g \\ 0 \end{pmatrix}$$

- Simplified from 3-D Euler: restrict to $x - z$ plane, omit Coriolis force
- Prognostic variables: density ρ, velocity (u, w), potential temperature θ
- State equation: $p = p_{00} \left(\frac{\rho R \theta}{p_{00}}\right)^{\gamma}$
- Constants: $p_{00} = 1013.25\,\text{hPa}$, $g = 9.8 m/s^2$, $c_p = 1004.67\,\text{J}/(\text{kg}\cdot\text{K})$, $c_v = 717.63\,\text{J}/(\text{kg}\cdot\text{K})$, $R = c_p - c_v = 287.04\,\text{J}/(\text{kg}\cdot\text{K})$, $\gamma = c_p/c_v \approx 1.4$
- Physical dissipation: only for momentum/potential temperature eqs

$$\frac{\partial \rho\phi}{\partial t} + ... - \nabla \cdot (\nu\rho\nabla\phi) = 0, \quad \text{for}\,\phi = u, w, \theta$$

- Reference: C. Yang and X.-C. Cai, *A scalable fully implicit compressible Euler solver for mesoscale nonhydrostatic simulation of atmospheric flows*, SIAM J. Sci, Comput., 2014 (to appear)

2D compressible Euler model: shifted system

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial z} + S = 0,$$

- The Euler model is a nonhydrostatic model at mesoscale
- Shift to recover the hydrostatic state (a special solution)

$$Q = \begin{pmatrix} \rho' \\ \rho u \\ \rho w \\ (\rho\theta)' \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p' \\ \rho u w \\ \rho u \theta \end{pmatrix}, G = \begin{pmatrix} \rho w \\ \rho w u \\ \rho w^2 + p' \\ \rho w \theta \end{pmatrix}, S = \begin{pmatrix} 0 \\ 0 \\ \rho' g \\ 0 \end{pmatrix}$$

where

$$\rho' = \rho - \bar{\rho}, \quad p' = p - \bar{p}, \quad (\rho\theta)' = \rho\theta - \bar{\rho}\bar{\theta}$$

and 'bar' indicates hydrostatic state: $\frac{\partial \bar{p}}{\partial z} = -\bar{\rho} g$

- The flux Jacobians remain unchanged, e.g.,

$$J_1 = \frac{\partial F}{\partial Q} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -u^2 & 2u & 0 & c^2/\theta \\ -uw & w & u & 0 \\ -u\theta & \theta & 0 & u \end{pmatrix}, \quad c = \sqrt{\gamma p/\rho}.$$

Fully implicit method

- Secord-order ESDIRK with adaptive time stepping, and second-order cell-centered finite volume

- Jacobian-free NKS for the nonlinear algebraic system with first-order cell-centered finite volume based Schwarz preconditioner

- Pointwise ordering of the unknowns and equations
$$X = \{(q_1, q_2, q_3, q_4)_{1,1}, (q_1, q_2, q_3, q_4)_{2,1}, \ldots\}$$

- Each entry of the Jacobian matrix is a 4×4 block

$$J = \left(\begin{array}{ccc|ccc|ccc}
\times & \times & & \times & & & & & \\
\times & \times & \times & & \times & & & & \\
& \times & \times & & & \times & & & \\
\hline
\times & & & \times & \times & & \times & & \\
& \times & & \times & \times & \times & & \times & \\
& & \times & & \times & \times & & & \times \\
\hline
& & & \times & & & \times & \times & \\
& & & & \times & & \times & \times & \times \\
& & & & & \times & & \times & \times
\end{array}\right)_{n \times n (block)} \qquad \times = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$

- Incomplete factorizations in the preconditioner solve are based on this point-block structure (the 4×4 blocks are solved exactly with an analytic inverse)

Newton-Krylov-Schwarz

- The nonlinear system $\mathcal{F}(X) = 0$, at time step n
- Suppose X_ℓ is the current approximate solution

$$X_{\ell+1} = X_\ell + \lambda_\ell S_\ell, \quad \ell = 0, 1, ...$$

- Initial guess $X_0 = X^{(n-1)}$: solution of the previous time step
- λ_ℓ is the step length computed using a linesearch
- S_ℓ is the Newton correction vector, by approximately solving

$$J_\ell(X_\ell)M^{-1}(MS_\ell) = -\mathcal{F}(X_\ell)$$

  using restarted (f)GMRES with relative tolerance η
- J_ℓ is the Jacobian, it is never generated (we use Jacobian-free method)
- M^{-1} is a Schwarz preconditioner

- $\Omega = \cup_k \Omega_k$ (non-overlapping). $\Omega_k \to \Omega_s^\delta$: extend Ω_k with δ layers (overlapping).
- One-level restricted additive Schwarz (RAS) preconditioner

$$M_{one}^{-1} = \sum_{k=1}^{np} (R_k^0)^T B_k^{-1} R_k^\delta.$$

- Two-level Schwarz (coarse first, cascade)

  x: current approximate solution, $r = b - Jx$: residual.

$$x^{(i)} := x + M_c^{-1} r, \quad x^{(ii)} := x^{(i)} + M_{one}^{-1} r^{(i)}$$

  - GMRES is switched to fGMRES for the outer iteration
  - One-level RAS can be used on both fine and coarse level, with independent adjustable parameters
- The subdomain matrices in the Schwarz preconditioners are generated based on a first-order spatial discretization (AUSM+up with 1-st order reconstruction)

A test case: Schär mountain

- From: Schär et al 2002 MWR
- Physical domain: $[-25\,\mathrm{km}, 25\,\mathrm{km}] \times [h(x), 21\mathrm{km}]$
- Initial condition
  - Constant mean flow of $\bar{u} = 10\mathrm{m/s}$
  - Uniformly stratified state with ground temperature $\bar{\theta}_0 = 280\mathrm{K}$ and Brunt-Vaisala frequency $\mathcal{N} = 0.01/\mathrm{s}$
  - Mountain profile (five peaks)

  $$h(x) = h_m e^{-(\frac{x}{r})^2} \cos^2\left(\frac{\pi x}{\lambda_c}\right)$$

  where $h_m = 250\,\mathrm{m}$, $r = 5\,\mathrm{km}$ and $\lambda_c = 4\,\mathrm{km}$
- Non-reflecting boundary conditions: sponge layers outside the domain of interest $[-10\,\mathrm{km}, 10\,\mathrm{km}] \times [0, 10\,\mathrm{km}]$

Strong scaling (# of unknowns 75 millions)

Strong scaling results for solving the Schär mountain problem on a 6144×3072 mesh to $t = 100\,\mathrm{s}$. For the explicit SSP RK-2 method, the time step size is $\Delta t = 0.01\,\mathrm{s}$; for the fully implicit method, the time step size is $\Delta t = 10\,\mathrm{s}$ and the overlap is $\delta = 2$.

	Newton/Step		GMRES/Newton		Compute time (s)	
np	ILU(1)	LU	ILU(1)	LU	ILU(1)	LU
576	2.1	2.1	69.6	34.6	222.6	299.8
1152	2.1	2.1	70.9	37.3	108.3	140.2
2304	2.1	2.1	74.5	41.9	57.9	72.3
4608	2.1	2.1	75.7	44.8	29.1	37.8
9216	2.1	2.1	82.1	52.0	16.1	20.2
18432	2.1	2.1	84.5	55.5	8.3	10.4

Note: The parallel speedup is nearly linear

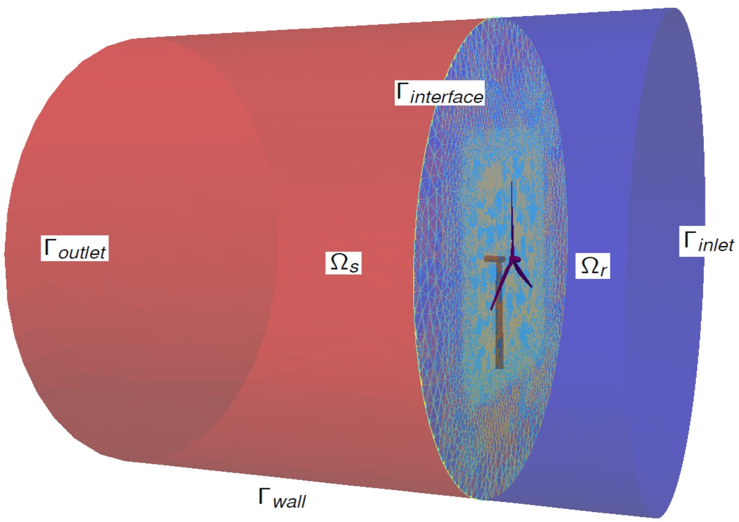
Weak scaling

Weak scaling results of the fully implicit method and the explicit SSP RK-2 method.

np	72	288	1152
Mesh size (in x)	576	1152	2304
Mesh size (in z)	288	576	1152
Implicit time steps	38	66	89
Total Newton	135	187	241
Total GMRES	14035	28332	54537
Total compute time (s)	115.7	254.7	502.9
Explicit time steps	360000	720000	1440000
Total compute time (s)	1863.7	3716.6	7456.3

Flow passing a complete wind turbine with tower/rotor

The flow is modeled by the 3D incompressible Navier-Stokes equations defined on a partial moving domain



Spatial discretization

We use a $P1 - P1$ stabilized finite element method in the spatial domain. The semi-discrete stabilized finite element formulation reads as:

$$\mathbf{B}_s(\mathbf{u}_s, p_s; \Phi_s, \psi_s) + \mathbf{B}_r(\mathbf{u}_r, p_r; \Phi_r, \psi_r) - \mathbf{F}_s(\Phi_s, \psi_s) - \mathbf{F}_r(\Phi_r, \psi_r) = \mathbf{0}$$

where

$$\begin{aligned}\mathbf{B}_s(\mathbf{u}_s, p_s; \Phi_s, \psi_s) = \rho \int_{\Omega_t^s} \frac{\partial \mathbf{u}_s^h}{\partial t} \cdot \Phi_s^h d\Omega_t^s + \mu \int_{\Omega_t^s} \nabla \mathbf{u}_s^h : \nabla \Phi_s^h d\Omega_t^s \\ + \rho \int_{\Omega_t^s} ((\mathbf{u}_s^h \cdot \nabla)\mathbf{u}_s^h \cdot \Phi_s^h d\Omega_t^s - \int_{\Omega_t^s} p_s^h \nabla \cdot \Phi_s^h d\Omega_t^s \\ + \int_{\Omega_t^s} (\nabla \cdot \mathbf{u}_s^h)\varphi_s^h d\Omega_t^s + \sum_{K \in \mathcal{T}^h} \left(\nabla \cdot \mathbf{u}_s^h,\ \tau_c \nabla \cdot \Phi_s^h\right)_K \\ + \sum_{K \in \mathcal{T}^h} \left(\frac{\partial \mathbf{u}_s^h}{\partial t} + (\mathbf{u}_s^h \cdot \nabla)\mathbf{u}_s^h + \nabla p_s^h,\ \tau_m(\mathbf{u}_s^h \cdot \nabla \Phi_s^h + \nabla \varphi_s^h)\right)_K\end{aligned}$$

$$\mathbf{F}_s(\Phi_s, \psi_s) = \int_{\Omega_t^s} \mathbf{f} \cdot \Phi_s^h d\Omega_t^s + \sum_{K \in \mathcal{T}^h} \left(\mathbf{f},\ \tau_m(\mathbf{u}_s^h \cdot \nabla \Phi_s^h + \nabla \varphi_s^h)\right)_K$$

Temporal discretization

An implicit backward Euler finite difference method is used in the temporal domain, that is, for a given semi-discretized system

$$\frac{d\mathbf{U}}{dt} = L(\mathbf{U}),$$

the backward Euler scheme

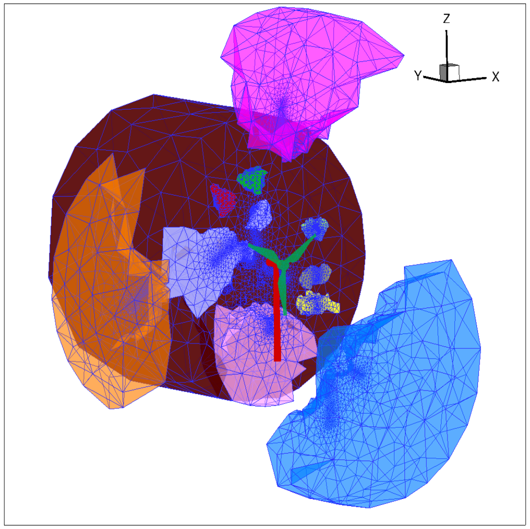
$$\frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} = L(\mathbf{U}^n)$$

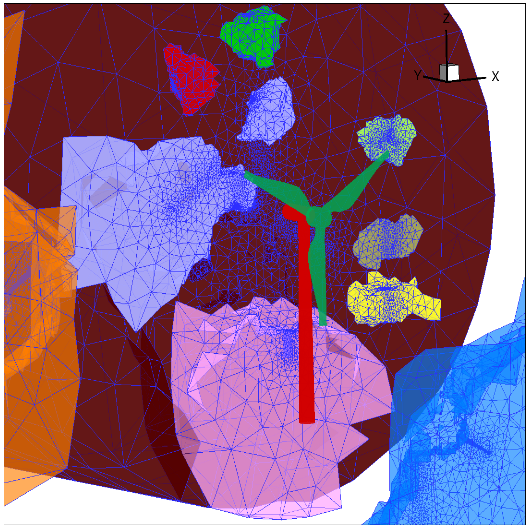
for the time integration. In each time step, we need to solve a nonlinear system:

$$\mathbf{F}(\mathbf{U}^n) = \mathbf{0}.$$

Additive Schwarz preconditioner

Partition the grid into subdomains





One-level restricted additive Schwarz preconditioner

$$\mathbf{M}^{-1} = \sum_{\ell=1}^{N_p} (R_\ell^0)^T \, \mathbf{J}_\ell^{-1} \, R_\ell^\delta$$

- R_ℓ^0 and R_ℓ^δ are restrictions to the degrees of freedom in the nonoverlapping subdomain Ω_ℓ and the overlapping subdomain Ω_ℓ^δ
- $\mathbf{J}_\ell = R_\ell^\delta \mathbf{J} \left(R_\ell^\delta\right)^T$ is a restriction of the Jacobian matrix
- Point-block incomplate LU factorization (ILU) is used to obtain the inverse of the subdomain Jacobian $\mathbf{J}_\ell^{-1}$
- The point-block ILU means that we group all physical components associated with a mesh point as a block and always perform an exact LU factorization for this small block in the ILU factorization

Wind turbine model

A three blade wind turbine with SERI 5807 root region airfoil and SERI 5806 tip region airfoil

Blade length = 63m
Tower height = 90m
Cylinder: D = 378m, H = 493m
Wind velocity = 15 m/s
Rotor velocity = 22 rpm
Fluid viscosity $\mu = 1.831 \times 10^{-5} kg/(ms)$
Number of elements = 1.1×10^{7}
$Re = 1.0 \times 10^{8}$

Parallel Performance

Jacobian matrix: Analytic. Subdomain solve: point-block ILU(1). Overlap 1

np	Newton	GMRES	Time (s)
512	3.0	51.72	127.3
1024	3.0	52.77	77.7
1536	3.1	53.94	67.5
2048	3.0	57.42	53.0

$DOF = 9.0 \times 10^6$

A fluid-structure interaction problem

- The linear elasticity equation for the wall structure

$$\rho_s \frac{\partial^2 \mathbf{x}_s}{\partial t^2} + \alpha \frac{\partial \mathbf{x}_s}{\partial t} - \nabla \cdot \sigma_s = \mathbf{f}_s \quad \text{in } \Omega_s$$

- The incompressible Navier-Stokes equations for the fluid in the arbitrary Lagrangian-Eulerian (ALE) framework

$$\rho_f \left.\frac{\partial \mathbf{u}_f}{\partial t}\right|_Y + \rho_f[(\mathbf{u}_f - \omega_g) \cdot \nabla]\mathbf{u}_f - \nabla \cdot \sigma_f = 0 \quad \text{in } \Omega_f(t)$$

$$\nabla \cdot \mathbf{u}_f = 0 \quad \text{in } \Omega_f(t)$$

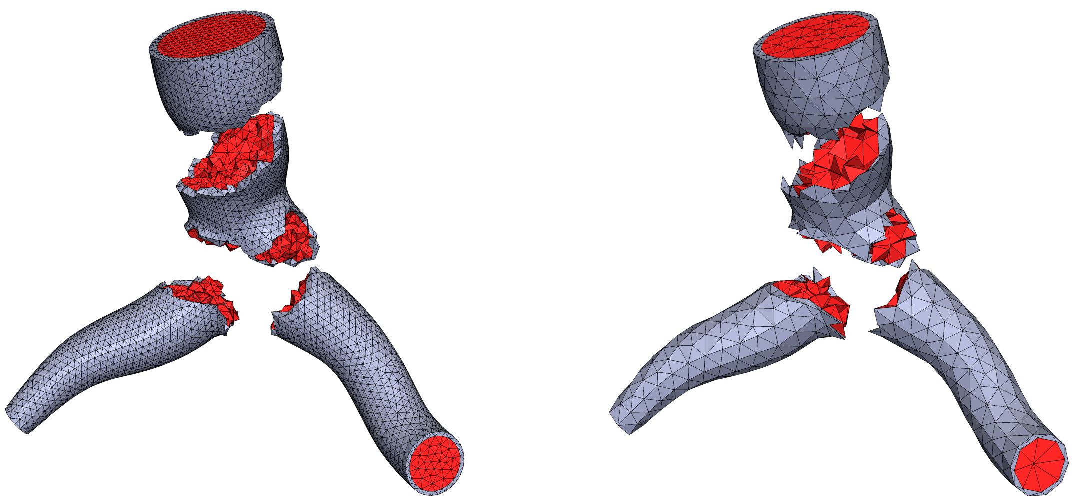
- The Laplace equation for the fluid domain movement

$$\Delta \mathbf{x}_f = 0 \quad \text{in } \Omega_0$$

- Coupling conditions on the fluid-structure interface

$$\sigma_s \cdot \mathbf{n}_s = -\sigma_f \cdot \mathbf{n}_f, \ \ \mathbf{u_f} = \frac{\partial \mathbf{x_s}}{\partial t}, \ \ \mathbf{x}_f = \mathbf{x}_s$$

A partitioned fluid-structure domain



Example partition of a fine and a coarse mesh into 4 subdomains (red=fluid, gray=artery)

Implicit finite element discretization of the fully coupled system

Find $x_s \in X_h$, $\dot{x}_s \in X_h$, $u_f \in V_h$, $p_f \in P_h$ and $x_f \in Z_h$ such that $\forall \phi_s \in X_h$, $\forall \varphi_s \in X_h$, $\forall \phi_f \in V_{h,0}$, $\forall \psi_f \in P_h$, and $\forall \xi \in Z_{h,0}$,

$$B_s(\{x_s, \dot{x}_s\}, \{\phi_s, \varphi_s\}; \sigma_f) + B(\{u_f, p_f\}, \{\phi_f, \psi_f\}; x_f) + B_m(x_f, \xi) = 0$$

Time discretization: We use a fully-implicit method for the time domain; e.g BDF2

- At the n^{th} time step, we obtain the solution x^n by solving a sparse, nonlinear algebraic system

$$\mathcal{F}_n(x^n) = 0,$$

where $x^n = (u_f^n \;\; p_f^n \;\; x_f^n \;\; x_s^n \;\; \dot{x_s}^n)^T$

About $\mathcal{F}$

- All terms of the equations are treated implicitly
- The discretized system $\mathcal{F}$ is highly nonlinear
  - The nonlinearity comes through the convective term, the stabilization terms and dependency of the moving mesh
- The system $\mathcal{F}$ is of mixed type
  - The fluid equations are nonlinear parabolic
  - The structure equations are linear hyperbolic
  - The moving mesh equations are elliptic
- The coupling conditions on the interface are enforced implicitly as part of the system $\mathcal{F}$
- When the resistive boundary condition is used, all equations on the outlet boundary are coupled

Monolithic Newton-Krylov-Schwarz

At each time step we solve the nonlinear system $\mathcal{F}(x) = 0$ with an inexact Newton method with cubic linesearch

- **Point-block ordering:** All physical components associated with a mesh point is grouped together as a block. Point-block versions of LU and ILU are used to obtain the inverse or approximate inverse of the subdomain Jacobian

- Solve a preconditioned linear Jacobian system to find the Newton correction $s^{(k)}$, by using a Krylov subspace method

  $$J(x^{(k)})M_k^{-1}M_k s^{(k)} = -\mathcal{F}(x^{(k)})$$

  Here J is a full Jacobian matrix computed analytically Update the approximation $x^{(k+1)} = x^{(k)} + \theta^{(k)} s^{(k)}$, where $\theta^{(k)} \in (0, 1]$ is the step length parameter

- $M_k^{-1}$is a restricted additive Schwarz (RAS) preconditioner, with point-block subdomain solve

Parallel performance

		One-level			Two-level		
DOF	np	Newton	fGMRES	time	Newton	fGMRES	time
$1.24 \cdot 10^6$	256	2.0	76.50	21.92	2.0	10.30	18.32
	512	2.0	102.30	9.58	2.0	11.40	8.27
	1024	2.0	129.45	6.30	2.0	12.45	4.42
$4.61 \cdot 10^6$	512	2.0	121.50	104.25	2.0	15.45	87.67
	1024	2.0	146.90	44.94	2.0	14.40	35.64
	2048	2.0	193.55	20.11	2.0	18.50	15.77
	3072	2.0	219.25	18.71	2.0	20.70	7.91

Inexact Jacobi-Davidson-Krylov-Schwarz for a nonlinear eigenvalue problem in 3D

The 3D Schrödinger equation is discretized with a cell-centered finite volume method on an uniform mesh in Cartesian coordinates The matrix polynomial eigenvalue problem takes the form

$$(\lambda^5 A_5 + \lambda^4 A_4 + \lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0)u = 0$$

Computational challenges:

- nonlinearity
- the eigenvalues of interest are located in the interior of the spectrum
- high accuracy is requied, resulting in large dimension of coefficient matrix

Input: A_i for $i = 0, \cdots, m$, and the maximum number of iterations k

1. Let $V = [v]$. v is an initial guess eigenvector with $\|v\|_2 = 1$

For $n = 0, \cdots, k$

2. Compute $W_i = A_i V$ and $M_i = V^H W_i$ for $i = 0, \cdots, m$

3. Select the desired eienpair (ϕ, s) with $\|s\|_2 = 1$ from the projected polynomial eigenproblem $\left(\sum_{i=0}^{m} \phi^i M_i\right) s = 0$

4. Compute $u = Vs$, and $r = \mathcal{A}_\phi u$

5. If the stopping criteria is satisfied, then stop

6. Compute $p_n = \mathcal{A}'_{\phi_n} u_n = \left(\sum_{i=1}^{m} i\phi_n^{i-1} A_i\right) u_n$

7. Solve approximately the correction equation with Schwarz preconditioning

$$\left\| \left(I - \frac{pu^*}{u^*p}\right) \mathcal{A}_\phi (I - uu^*)t + r \right\|_2 \leq \varepsilon_n \|r\|_2, \qquad t \perp u$$

8. Orthogonalize t against V, set $v = v/\|t\|_2$, then $V \leftarrow [V, v]$

End for

Results for the first two eigenpairs

- Matrix size: $161,101,649 \times 161,101,649$

	One-level Schwarz			Two-level Schwarz		
np	JD	FGMRES	Time(s)	JD	FGMRES	Time(s)
5120	4	185.25	42.48	4	38.75	7.48
7168	4	185.25	29.20	4	39.50	6.36
9216	4	186.00	23.23	4	38.75	5.34
10240	4	186.00	22.46	4	39.75	4.84

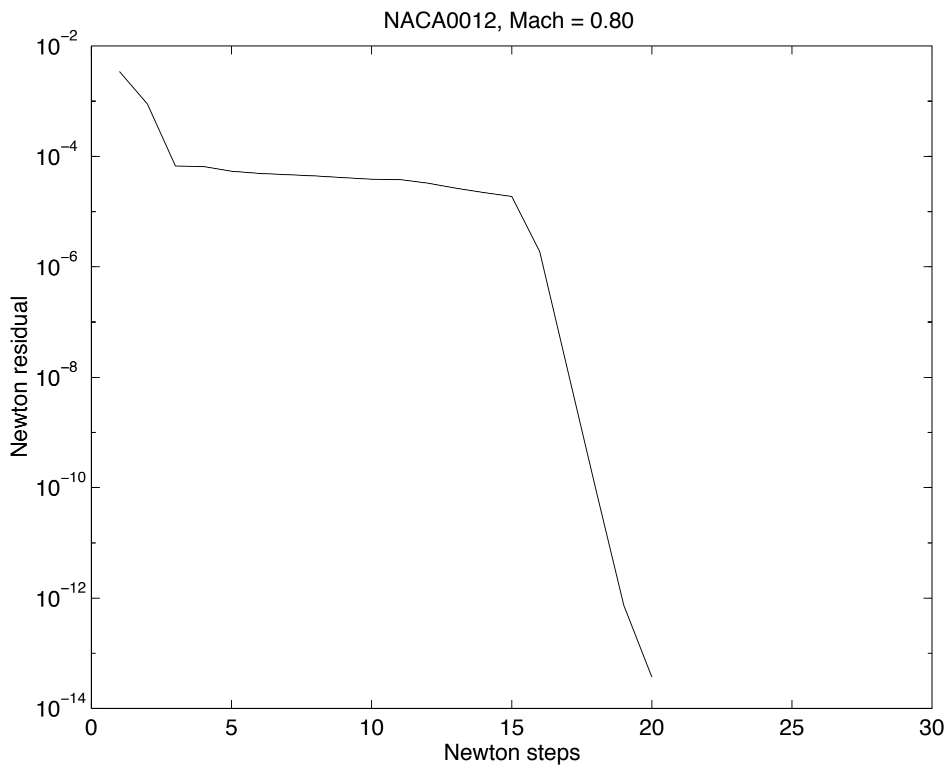
	One-level Schwarz			Two-level Schwarz		
np	JD	FGMRES	Time(s)	JD	FGMRES	Time(s)
5120	4	251.75	71.10	4	34.75	9.47
7168	4	255.75	47.96	4	34.25	6.20
9216	4	254.50	39.29	4	34.50	5.64
10240	4	256.50	37.37	4	34.00	5.29

Residual histories of the first 6 eigenpairs

it	e_0	e_1	e_2	e_3	e_4	e_5
0	$2.59e{+}00$	$4.25e{+}00$	$4.25e{+}00$	$9.43e{+}00$	$5.46e{+}00$	$7.34e{+}00$
1	$7.62e{-}02$	$2.23e{-}01$	$2.23e{-}01$	$4.05e{+}00$	$5.96e{-}01$	$1.46e{+}00$
2	$1.92e{-}04$	$3.45e{-}04$	$3.38e{-}04$	$8.93e{-}01$	$6.65e{-}03$	$8.27e{-}02$
3	$2.15e{-}08$	$7.00e{-}08$	$4.44e{-}08$	$8.20e{-}02$	$7.10e{-}07$	$8.51e{-}05$
4	$3.02e{-}12$	$7.00e{-}12$	$1.46e{-}11$	$7.35e{-}04$	$4.57e{-}11$	$6.59e{-}08$
5				$1.07e{-}07$		$1.89e{-}11$
6				$3.63e{-}11$		

Note: quadratic convergence!

Sometimes, Newton-Krylov doesn't converge or converges for a while and then stagnate



What do I do ?

Schwarz as a nonlinear preconditioner for Newton for problems with high local nonlinearities

- Additive Schwarz as a left preconditioner is not hard to program if there is a NKS based code available, but not too easy since $F(\cdot)$ needs to be replaced. On the other hand, if additive Schwarz is used as a right preconditioner, $F(\cdot)$ doesn't have to be touched at all

- Basic idea: Replace $F(x) = 0$ by $F(G(y)) = 0$. $G(y)$ is some locally corrected x

- The preconditioner $G(\cdot)$: Nonlinear elimination or nonlinear Schwarz, which can be considered as "multiple nonlinear eliminations"

- Or, we can consider $G(\cdot)$ as a way to improve the initial guess before every Newton iteration (BTW, Newton is a one-step method; i.e., every solution is simply an initial guess for the next Newton iteration)

- X.-C. Cai and D. E. Keyes, SIAM J. Sci. Comput., (2002)
  X.-C. Cai and X. Li, SIAM J. Sci. Comput., (2011)

Nonlinear elimination – peak removing

Consider a nonlinear problem $F(x) = 0$ defined on Ω with the current approximate solution x_c

- Identify the worst region. A peak of F is a region $\omega \in \Omega$ such that $\|F(x_c)\|_{2(\omega)}$ is large

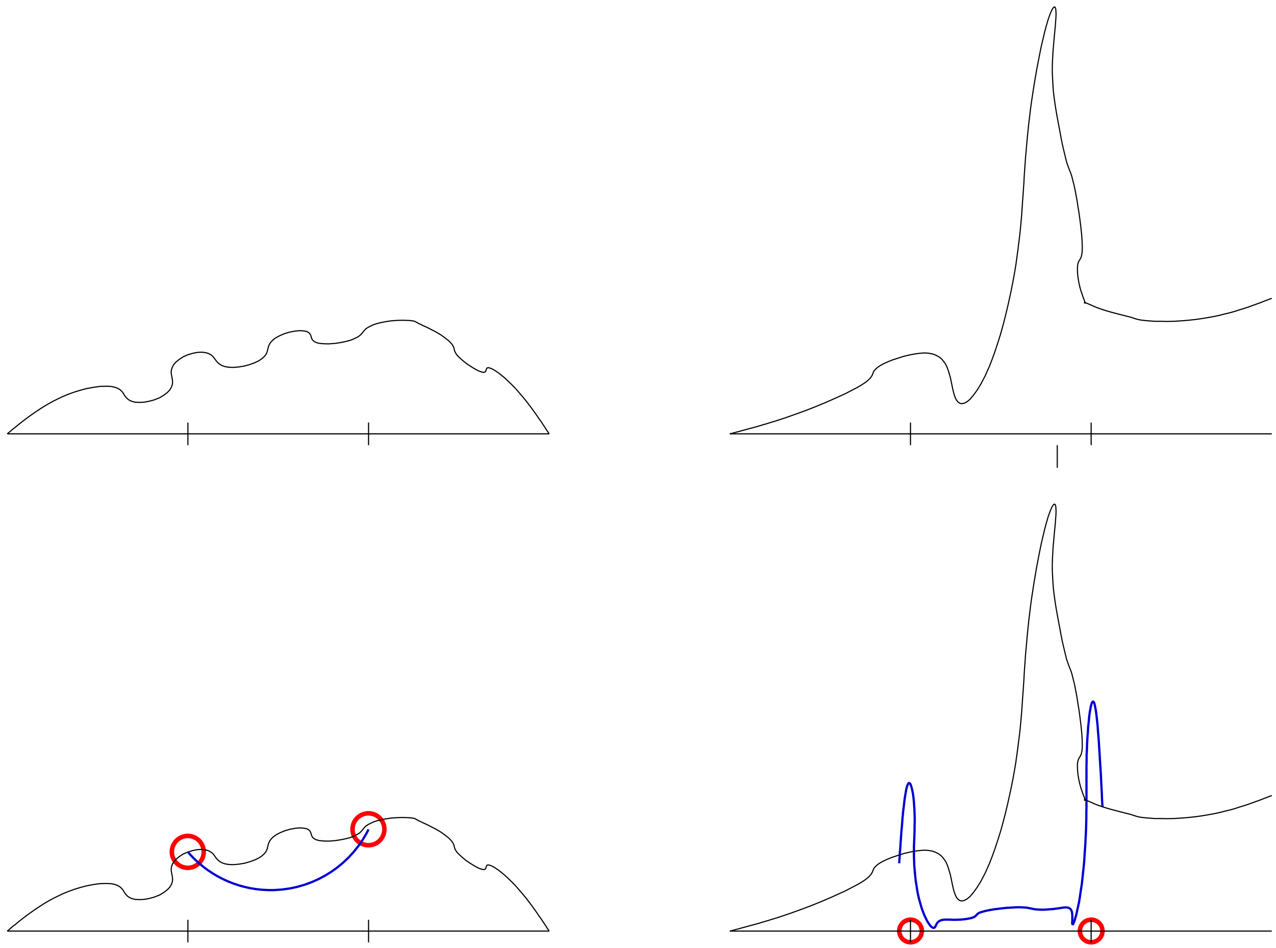
- Solve a local nonlinear problem

$$F|_\omega(x_\omega) = 0, \text{ with boundary condition } x_\omega|_{\partial\omega} = x_c|_{\partial\omega}$$

- Locally correct the solution

$$x_{new} = \begin{cases} x_\omega & \text{in } \omega \\ x_c & \text{in } \Omega \setminus \omega \end{cases}$$

Pros and Cons of nonlinear elimination



current solution current residual function

Schwarz preconditioners

- R_i^δ, R_i^0 restriction with and without overlap. δ is the overlapping size

- Additive Schwarz

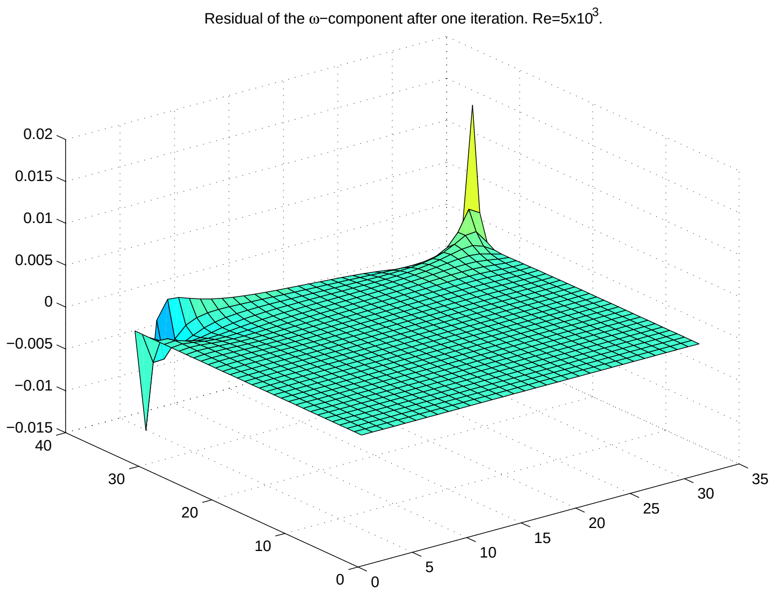
$$\text{Linear}: \sum_{i=1}^{N} (R_i^\delta)^T A_i^{-1} R_i^\delta \qquad \text{Nonlinear}: \sum_{i=1}^{N} (R_i^\delta)^T F_i^{-1} (R_i^\delta x)$$

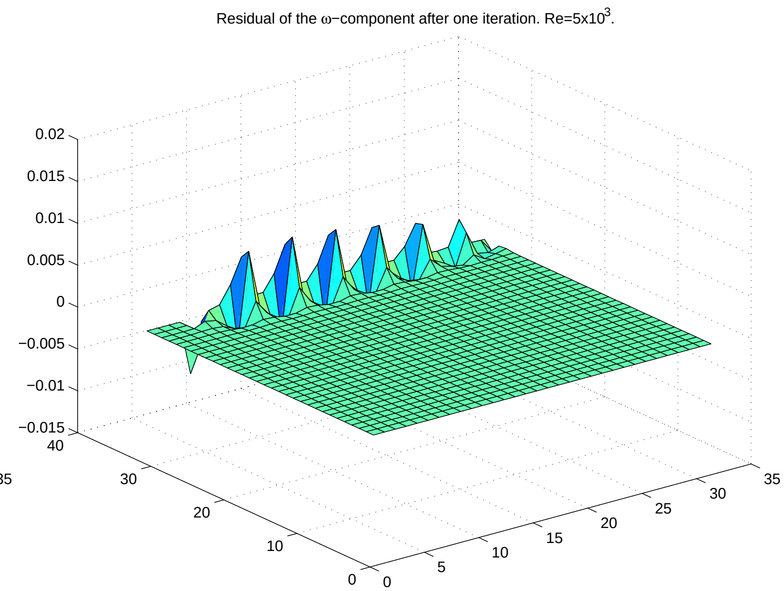
- Restricted additive Schwarz

$$\text{Linear}: \sum_{i=1}^{N} (R_i^0)^T A_i^{-1} R_i^\delta \qquad \text{Nonlinear}: \sum_{i=1}^{N} (R_i^0)^T F_i^{-1} (R_i^\delta x)$$

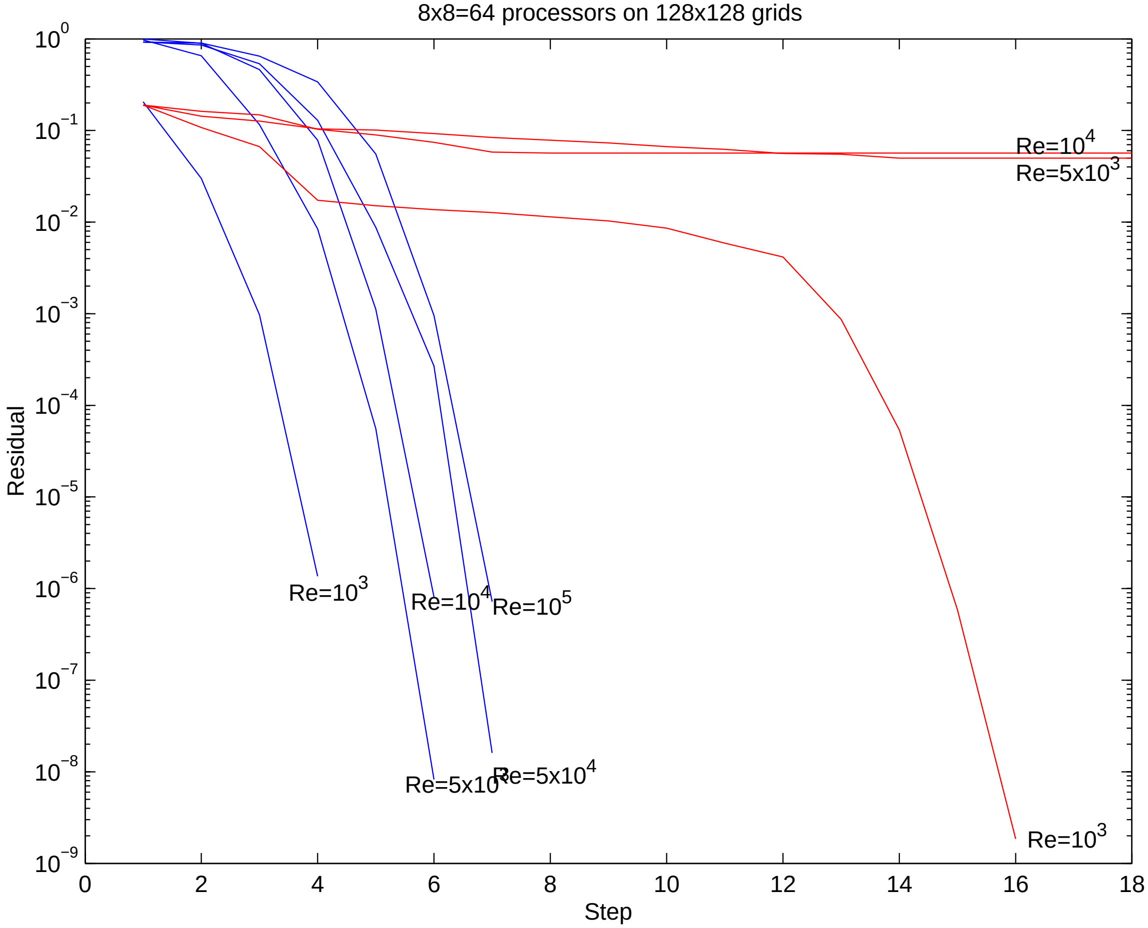
Effect of nonlinear RAS preconditioning

Two-dimensional driven cavity flow problem with high Reynolds number





Comparing NKS and RAS-NKS



Driven cavity flow (128×128 mesh, tol=10^{-6})

Global Newton iterations					
# of processors	$Re = 10^3$	$Re = 5 \times 10^3$	$Re = 10^4$	$Re = 5 \times 10^4$	$Re = 10^5$
$8 \times 8 = 64$	4	6	6	7	7
$8 \times 16 = 128$	4	5	6	7	7
$16 \times 16 = 256$	4	5	6	7	8
Average GMRES iterations					
$8 \times 8 = 64$	64	59	55	57	57
$8 \times 16 = 128$	84	68	63	58	57
$16 \times 16 = 256$	114	102	101	90	86
Ranges of the subdomain Newton iterations					
$8 \times 8 = 64$	$0 \sim 5$	$0 \sim 7$	$0 \sim 7$	$0 \sim 9$	$0 \sim 10$
$8 \times 16 = 128$	$0 \sim 4$	$0 \sim 5$	$0 \sim 5$	$0 \sim 6$	$0 \sim 7$
$16 \times 16 = 256$	$0 \sim 4$	$0 \sim 5$	$0 \sim 6$	$0 \sim 8$	$0 \sim 9$
Computing times (sec)					
$8 \times 8 = 64$	2.581	2.703	6.469	7.653	8.095
$8 \times 16 = 128$	1.131	1.196	1.233	1.894	2.010
$16 \times 16 = 256$	0.6272	1.481	0.7853	1.619	1.970

Some final remarks

- For problems with global nonlinearity, NKS is a good general purpose parallel solver. Multilevel maybe necessary if the number of processors is large

- For problems with global and local nonlinearities, a combination of full space Newton and subspace Newton offers a good strategy

- For problems with only local nonlinearities, subspace Newton is often sufficient

- It is often difficult to tell what types of nonlinearities a problem may have

- The norm of the residual function, $\|F(x)\|_2$, is often not a good monitor, unfortunately all existing nonlinear theory are based on $\|F(x)\|_2$

- Many parameters (stopping conditions)

- Some papers can be found at

`www.colorado.edu/cs/users/cai`