中国科学院大学
夏季强化课程
2022

# Fast Solvers for Large Algebraic Systems

## Lecture 3. Methods for non-symmetric problems

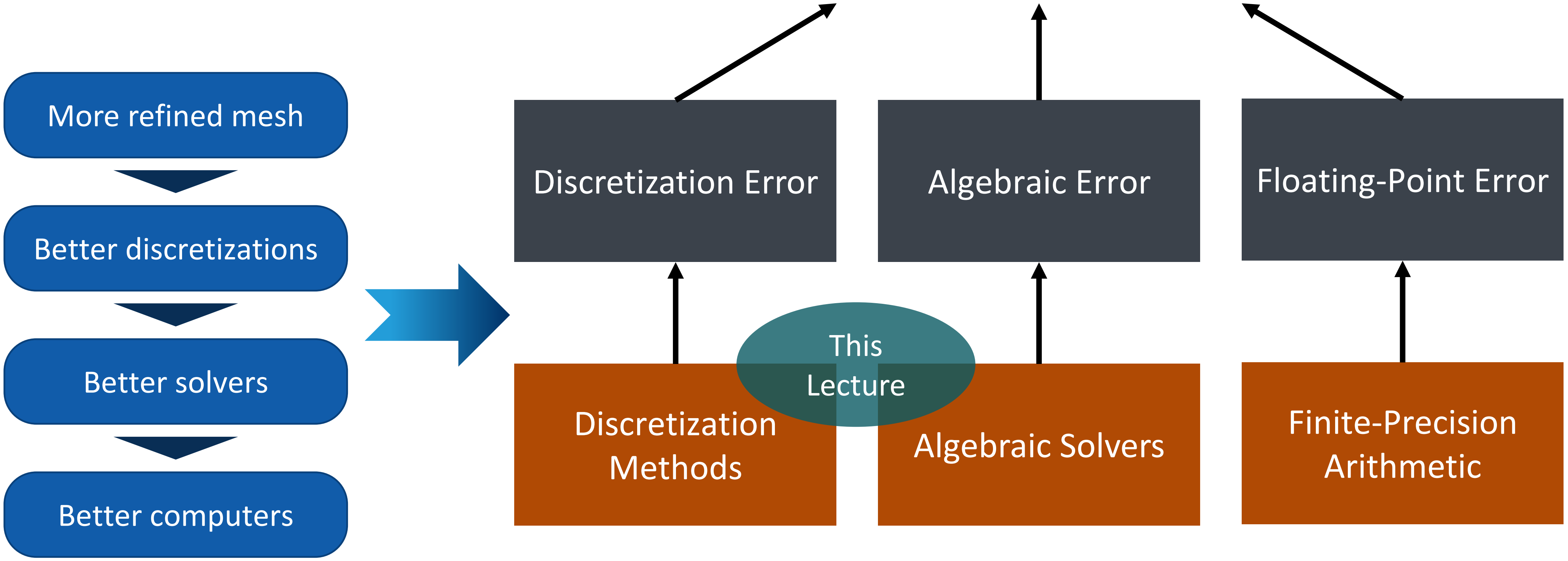
Chensong Zhang, AMSS
http://lsec.cc.ac.cn/~zhangcs

3
非对称问题求解方法

# Table of Contents

# Sources of Error in Simulation

Approximation: $u(x) = U_h(x) + \mathcal{E}_{\text{dis}} + \mathcal{E}_{\text{alg}} + \mathcal{E}_{\text{fp}}$

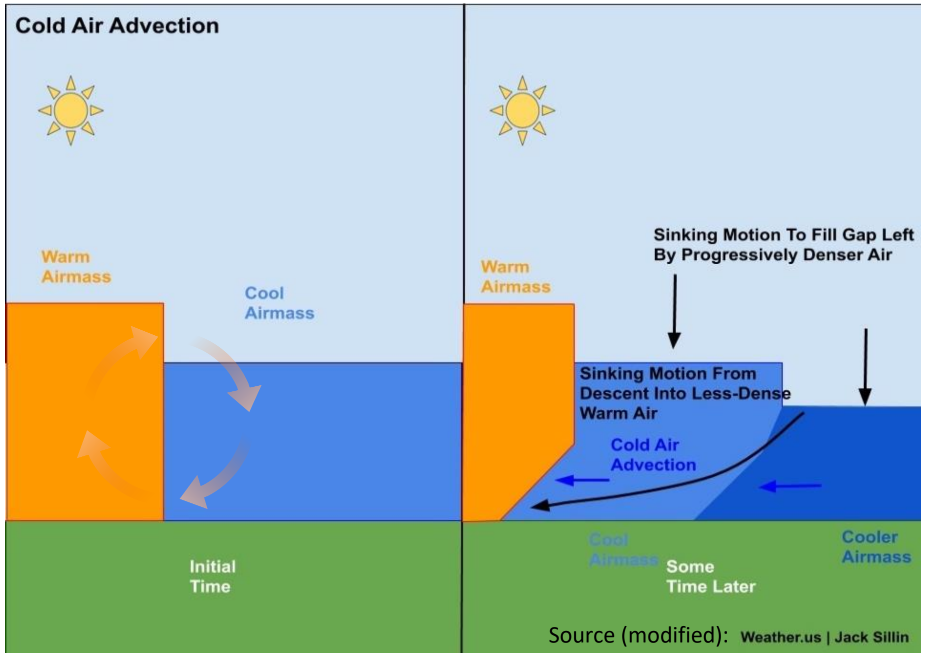


# Introduction

A non-symmetric model problem and its discretizations

/01

# A "Simple" Model Problem




Mass conservation

$$\frac{d}{dt}\int_{\omega}\rho(x,t)dx = -\int_{\partial\omega}\mathbf{F}(x,t)\cdot n\, dS$$

where

$$\mathbf{F}(x,t) = \rho(x,t)\,\mathbf{v}$$

Physical background

- $v(x) \equiv \mathbf{v}$: Velocity field of the air or water
- ☞ $\rho(x,t)$: Density of certain component
- $\mathbf{F}(x,t)$: Mass flux at point $x$ and time $t$
- $f(x,t)$: Source/sink at point $x$ and time $t$

Advection

Convection

Differences?

Model equations

- Transport equation: $D_t\rho(x,t) := \rho_t(x,t) + \mathbf{v}\cdot\nabla\rho(x,t) = 0$ (Material derivative)
- Transport equation with source/sink: $\rho_t(x,t) + \mathbf{v}\cdot\nabla\rho(x,t) = f(x,t)$
- Advection-diffusion equation: $\rho_t(x,t) + \mathbf{v}\cdot\nabla\rho(x,t) - \Delta\rho(x,t) = f(x,t)$
- Poisson's equation: $-\Delta\rho(x,t) = f(x,t)$ (Steady-state, no advection)

- Mass (energy) advection: Transfer of a substance due to bulk motion
- Heat convection: Combined effect of bulk motion and diffusion

# Advection-Diffusion Equation

- Advection-diffusion (convection-diffusion)

$$\rho_t(x,t) + \mathbf{v} \cdot \nabla \rho(x,t) - \Delta \rho(x,t) = f(x,t)$$

$$u_t(x,t) + b \cdot \nabla u(x,t) - \Delta u(x,t) = f(x,t)$$

- Discretize the PDE with implicit discretization methods
  - If discretize in a classical way, we will get non-symmetric linear systems
  - Q: Is it still possible to get symmetric problems?
  - Yes. We can discretize the material derivative (like ELM in Lecture 2)
- How to solve the non-symmetric linear systems?
  - Direct solvers still work very well: $LL^T$, $LDL^T$ → LU
  - Iterative solvers need to be modified: CG → GMRES, BiCGstab, …
  - Preconditioners most likely need to be modified: IC → ILU, CAMG → ???

# Steady-State Advection-Diffusion Equation

- Steady-state advection-diffusion equation (ADE)

$$-\nabla\cdot(\mu\nabla u)+b\cdot\nabla u=f,\quad \Omega \qquad\qquad \nabla\cdot b=0$$
$$u=0,\quad \partial\Omega$$

- For this model problem, the weak form of advection is skew self-adjoint:

$$c(u,v):=(b\cdot\nabla u,v)=-(u,\nabla\cdot(vb))=-(u,v\nabla\cdot b+b\cdot\nabla v)=-(u,b\cdot\nabla v)$$

- ADE is bounded and coercive ➔ Lax-Milgram Theorem ➔ Well-posed
- Challenging to solve: boundary layers, convection-dominant, …
- Difficult to solve if the Péclet number is large (convection dominates)

Dimensionless

$$\text{Péclet}:=\frac{|b|\,L}{\mu},\quad L \text{ is the characteristic length of } \Omega$$

There's a discrete version!
Mesh/discrete Péclet number

# General FE Discretization

- Some operators:

$$\hat{\mathcal{A}}u := \mathcal{A}u + \mathcal{N}u, \quad H_0^1(\Omega) \longrightarrow H^{-1}(\Omega)$$

$$\mathcal{A}u := -\nabla \cdot (\mu \nabla u), \quad H_0^1(\Omega) \longrightarrow H^{-1}(\Omega) \qquad \mathcal{N}u := (b \cdot \nabla)\, u$$

- Define bilinear forms and the finite element approximation

$$a(u,v) := \int_\Omega \mu \nabla u \cdot \nabla v$$

Finite element approximation

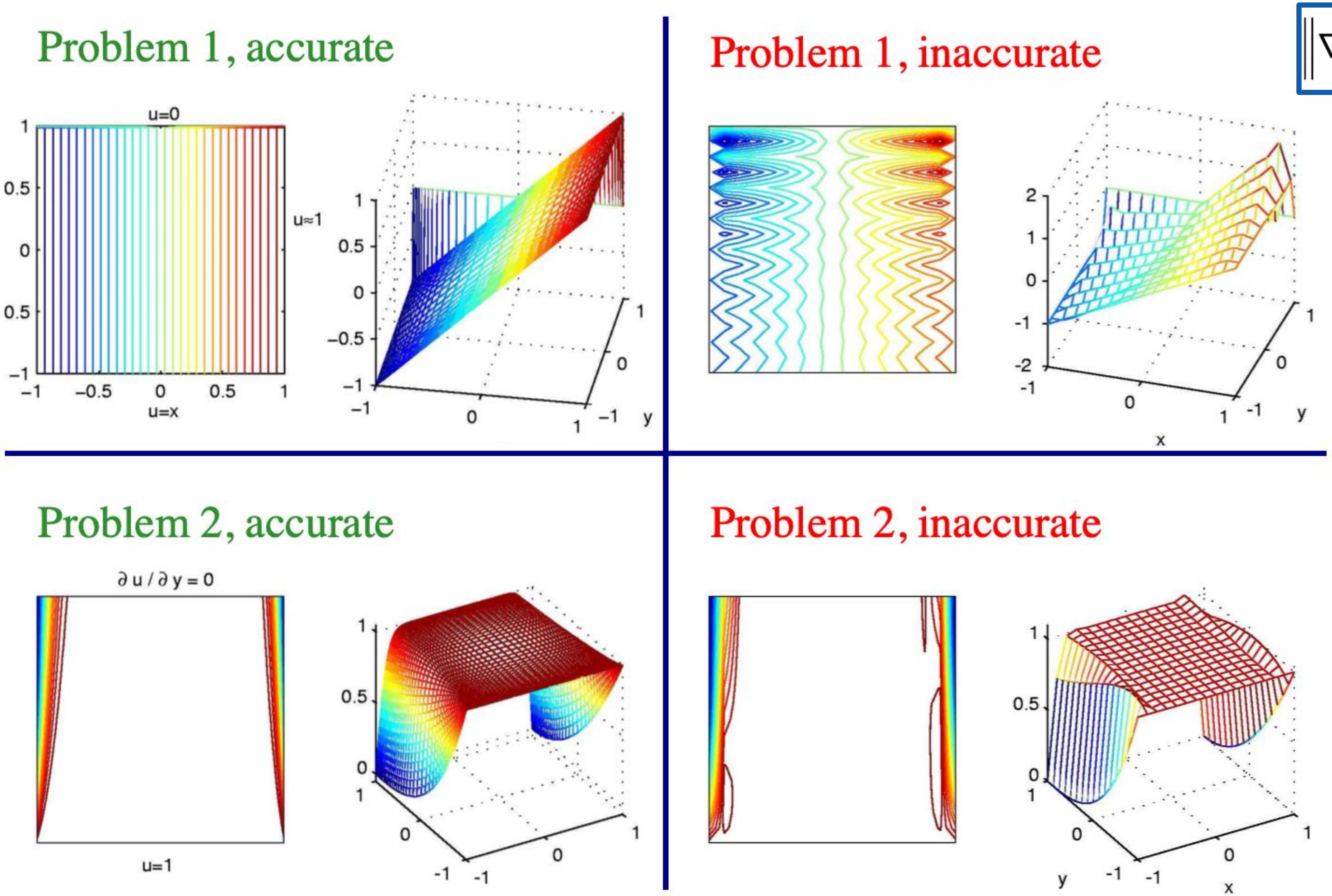
$$\hat{a}(u,v) := a(u,v) + c(u,v) \Longrightarrow \text{Find } u_h \in V_h : \ \hat{a}(u_h, v_h) = (f, v_h), \ \ \forall v_h \in V_h$$

$$c(u,v) := \int_\Omega (b \cdot \nabla) u\, v$$

Solving the resulting algebraic system is not easy.

Ref: Jinchao Xu. “Two-grid Discretization Techniques for Linear and Nonlinear PDEs”. SIAM J. Numer. Anal. 33, 5, 1759–1777, 1996

# Difficulties in Numerical Simulation




$$\left\|\nabla(u-u_h)\right\| \le \frac{1+\text{Péclet}}{\mu}\inf_{v_h}\left\|\nabla(u-v_h)\right\|$$

- Quasi-optimal estimate, could have large constant
- Sharp boundary layers might cause non-physical oscillations if the mesh Péclet number is large

Source: Howard Elman. “Discretization and solution of convection-diffusion problems”. Lecture Notes, 2006

# EAFE Method

NCMIS

- 1D model: $-(\mathcal{F}(u))' = f \iff -(u' + bu)' = f \qquad u(0) = u(1) = 0$
- Change variable: $u = e^{-bx} w \implies u' + bu = e^{-bx} w' \implies -\left(e^{-bx} w'\right)' = f$
- Flux term: $\mathcal{F} = e^{-bx}\left(e^{bx} u\right)' \implies \left(e^{bx} u\right)' = \mathcal{F} e^{bx}$
- Numerical flux: $e^{-bx}\left(e^{bx} u\right)' = F \qquad F_i = \frac{b}{e^{bx_{i+1}} - e^{bx_i}}\left(e^{bx_{i+1}} u_{i+1} - e^{bx_i} u_i\right)$
- P1 finite element: $\int_{x_i}^{x_{i+1}} \mathcal{F} \phi_i' \approx F_i \int_{x_i}^{x_{i+1}} \phi_i' = F_i h \cdot (-1/h) = -F_i$
- EAFE: $\frac{1}{h}\Big( - B(bh) u_{i-1} + \big(B(bh) + B(-bh)\big) u_i - B(-bh) u_{i+1} \Big) = f_i$

Using p.w. constant flux approximations and integrating in each interval

Bernouilli function

Sounds Good?

Ref: Xu and Zikatanov, 1999

# A Two-Level FE Discretization

- Introduce a new finite element space (usually smaller space)

$$\text{Find } u_H \in V_H: \ \hat{a}(u_H, v_H) = (f, v_H), \ \ \forall v_H \in V_H$$

$$\text{Find } u_{h,H} \in V_h: \ a(u_{h,H}, v_h) = (f, v_h) - c(u_H, v_h), \ \ \forall v_h \in V_h$$

- Convergence results:

$$\|u_h - u_{h,H}\|_1 \lesssim H^{r+1}\|u\|_{r+1} \quad \text{and} \quad \|u - u_{h,H}\|_1 \lesssim \left(h^r + H^{r+1}\right)\|u\|_{r+1}$$

- Improved the two-grid discretization

$$\text{Find } e_H \in V_H: \ \hat{a}(e_H, v_H) = c(u_H - u_{h,H}, v_H), \ \ \forall v_H \in V_H$$

$$\text{Update } u^*_{h,H} = u_{h,H} + e_H$$

Ref: Xu, Jinchao. "An introduction to multilevel methods." Wavelets, multilevel methods and elliptic PDEs, 1997

# A Two-Level Preconditioner

- Assume that there exists a subspace $V_c \subset V$ such that

$$\text{Find } u_c \in V_c : \ (\hat{A}u_c, v_c) = (\hat{A}u, v_c), \ \ \forall v_c \in V_c$$

has a unique solution, defined by:

$$\text{Define } P_c : V \to V_c, \ P_c u = u_c$$

Need to solve the coarse problem here

- Define coarse-level problem and L$^2$-projection

$$(\hat{\mathcal{A}}_c u_c, v_c) = (\hat{\mathcal{A}} u, v_c), \ \ \forall u_c, v_c \in V_c$$

$$(Q_c u, v_c) = (u, v_c), \ \ \forall u \in V, v_c \in V_c$$

- Construct a new preconditioner:

Preconditioner for the symmetric part

$$\hat{\mathcal{B}} := \beta\mathcal{B} + \hat{\mathcal{A}}_c^{-1} Q_c \qquad \hat{\mathcal{A}}_c P_c = Q_c \hat{\mathcal{A}}$$

"Solver" for the non-symmetric problem

$$\hat{\mathcal{B}}\hat{\mathcal{A}} := \beta\mathcal{B}\hat{\mathcal{A}} + P_c$$

Ref: J. Xu and X.-C. Cai. "A preconditioned GMRES method for nonsymmetric or indefinite problems". Math. Comp. 59, 311-319, 1992

# The Finite Element Circus

Former circus ringmasters D. Arnold and R. Falk and the current co-ring master S. Brenner at the 90th birthday of Ivo Babuška (2016)

**HISTORY OF THE FINITE ELEMENT CIRCUS**

COMPILED BY RICHARD S. FALK, SEPTEMBER 9, 2014

Source: https://sites.google.com/view/fecircus

The Finite Element Circus is a regular conference devoted to the theory and applications of the finite element method and related areas of numerical analysis and partial differential equations. The Circus was conceived by Ivo Babŭska, Bruce Kellogg, and Jim Bramble over beer and pizza at the Beltway Plaza shopping center in Hyattsville, Maryland in 1970, and the first circus was held at the University of Maryland, College Park later that year. Serious mathematical study of the finite element method was just getting underway and the Circus provided an important opportunity for those in the field to share current research.

- R. Scott, High order methods for fluid flow, Finite Element Circus, Spring 1996
- H. Wang, An ELLAM scheme for advection diffusion equations, Finite Element Circus, Spring 1996
- J. Xu, EAFE scheme for convection-diffusion equations and conservation laws, Finite Element Circus, Fall 1997

# Krylov Subspace Methods

Krylov subspace methods for non-symmetric problems

/02

# Krylov Matrices

- By the Cayley–Hamilton theorem, there exists a polynomial $q_{n-1}(\lambda) \in \mathcal{P}_{n-1}$ , such that

$$A^{-1} = q_{n-1}(A)$$

- The Krylov matrix can be defined as $K_n := [r, Ar, A^2 r, \ldots, A^{n-1} r]$

- Get a similar transformation:

Upper Hessenberg ↓

$$AK_n = [Ar, A^2 r, \ldots, A^n r] = K_n\, [e_2, e_3, \ldots, e_n, K_n^{-1} A^n r] =: K_n C_n$$

$$\Longrightarrow \quad C_n = K_n^{-1} A K_n$$

- Apply the QR factorization $K_n = Q_n R_n$

$$R_n^{-1} Q_n^* A Q_n R_n = C_n$$

$$Q_n^* A Q_n = R_n C_n R_n^{-1} =: H_n$$

$$\Longrightarrow \quad Q_n^* A Q_n = H_n$$

# Krylov Subspace Methods

- The above Krylov matrix approach is not useful in practice:
  - Nearly-singular ➔ It's a power sequence
  - Expensive to compute the full QR factorization
- The Krylov subspace

$$\mathcal{K}_m(A, r) := \text{span}\{r, Ar, A^2 r, \ldots, A^{m-1} r\}$$

Nested Subspaces

- Examples: CG, MinRes, GMRES, BiCGstab, FOM, GCR, ORTHOMIN, ... (see Y. Saad 2003)
- We will focus on the generalized minimum residual (GMRES) method:

$$\min_{e \in \mathcal{K}_m(A,r)} \|r - Ae\|$$

➔ Line 4 of Algorithm 1

Simoncini, Valeria, and Daniel B. Szyld. "Recent computational developments in Krylov subspace methods for linear systems." Numerical Linear Algebra with Applications 14.1, 1-59, 2007

# Basic Idea of GMRES

Saad and Schultz 1986

- Suppose that we have an **orthonormal basis** of the Krylov subspace

$$\mathcal{K}_m := \mathcal{K}_m(A, r) = \mathrm{span}\{q_1, q_2, \ldots, q_m\}$$ 1

- Solve a least squares (LSQ) problem to find an "optimal" solution in the Krylov subspace

$$\text{Find } e_m \in \mathcal{K}_m \quad \Longrightarrow \quad e_m = Q_m y, \quad y \in \mathbb{C}^m$$

$$\min_{e_m \in \mathcal{K}_m} \left\| r - A e_m \right\| \longrightarrow \min_{y \in \mathbb{C}^m} \left\| r - A Q_m y \right\| \longrightarrow \min_{y \in \mathbb{C}^m} \left\| r - Q_{m+1} \bar{H}_m \, y \right\|$$

$$AQ_m = Q_{m+1} \bar{H}_m$$ 2

$$\longrightarrow \min_{y \in \mathbb{C}^m} \left\| Q_{m+1}^* r - \bar{H}_m \, y \right\| \longrightarrow \min_{y \in \mathbb{C}^m} \left\| \beta e_1 - \bar{H}_m \, y \right\|, \quad \beta := \|r\|$$

Multiply unitary matrix does not change 2-norm

3 $$q_1 := \frac{r}{\|r\|} \longrightarrow q_1^* r = \|r\|, \quad q_j^* r = 0, \quad j = 2, 3, \ldots$$

Only need to solve a smaller LSQ problem

# Eigenvalue Equations

- Assume that we have the Hessenberg decomposition of $A$

$$A = QHQ^*, \quad Q: \text{orthogonal}, \quad H: \text{upper Hessenberg}$$

- We want part of this decomposition

$$A[q_1, \dots, q_m, q_{m+1}, \dots, q_n] = [q_1, \dots, q_m, q_{m+1}, \dots, q_n]H$$

$$= [q_1, \dots, q_m, q_{m+1}, \dots, q_n] \cdot \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1m} & \cdots & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2m} & \cdots & \cdots & h_{2n} \\ 0 & h_{32} & \ddots & \cdots & h_{3m} & \cdots & \cdots & h_{3n} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \cdots & \ddots & h_{m,m-1} & h_{mm} & \cdots & \vdots & \vdots \\ \vdots & \cdots & \cdots & 0 & h_{m+1,m} & \ddots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & h_{n,n-1} & h_{nn} \end{bmatrix}$$

$\bar{H}_m$

# Toward A Practical Iterative Procedure

- Part of the eigenvalue equation:

$$A[q_1, \ldots, q_m] = [q_1, \ldots, q_m, q_{m+1}]\, H(1:m+1, 1:m)$$

$$\boxed{AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^T}$$

- How to develop an iterative procedure?

$$\boxed{AQ_m = Q_{m+1}\bar{H}_m, \quad A \in \mathbb{C}^{n\times n}, Q_m \in \mathbb{C}^{n\times m}, \bar{H}_m \in \mathbb{C}^{(m+1)\times m}}$$

2

$$\Downarrow$$

$$Aq_m = h_{1m}q_1 + h_{2m}q_2 + \cdots + h_{mm}q_m + h_{m+1,m}q_{m+1}$$

$$\Downarrow$$

$$q_{m+1} = \frac{Aq_m - \sum_{i=1}^{m} h_{im}q_i}{h_{m+1,m}}$$

There should be an iterative procedure to construct q-vectors!

# Arnoldi Iteration

- Assume that we start from the residual vector

$$q_1 := \text{normalized initial residual} = \frac{r}{\|r\|}$$

3

- Gram-Schmidt orthogonalization

$$q_1^* q_2 = 0$$

$$q_2 = \frac{Aq_1 - h_{11}q_1}{h_{21}} \quad \Rightarrow \quad \boxed{h_{11} = q_1^* A q_1}$$

$$w_2 := Aq_1 - h_{11}q_1$$

$$q_2 = \frac{w_2}{h_{21}}, \quad \boxed{h_{21} = \|w_2\|} \quad \Rightarrow \quad Q,\ H$$

# Practical Implementation of Arnoldi

Algorithm 4: Arnoldi algorithm with Gram-Schmidt

```
%% Given an initial residual vector r ∈ ℝ^n;
for j = 1, 2, ..., m
    h_{i,j} ← (Aq_j, q_i),  i = 1 : j;
    w_j ← Aq_j − \sum_{i=1}^{j} h_{i,j} q_i;
    h_{j+1,j} ← ‖w_j‖;
    if h_{j+1,j} == 0, break; %% lucky breakdown
    q_{j+1} ← w_j / h_{j+1,j};
end
```

← Standard Gram-Schmidt, not stable!

➔ Modified Gram-Schmidt ➔ Reorthogonalization

- Arnoldi algorithm breaks down at step $j$, if and only if the minimal polynomial of the vector $q_1$ (i.e. $p(A)q_1 = 0$) is of degree $j$
- See Proposition 6.6 in Y. Saad, "Iterative Methods for Sparse Linear Systems" ($2^{nd}$ Edition), 2003

# Classical Gram-Schmidt Method

- Orthogonalization

$$\{a_1, a_2, \ldots, a_{j-1}\} \Rightarrow \{q_1, q_2, \ldots, q_{j-1}\} \quad \text{Orthonormal}$$

1

$$\text{Add a new vector } a_j \Rightarrow q_j \ ?$$

- Classical G-S:

$$w_j = a_j$$

$$\left.\begin{aligned} h_{ij} &= q_i^* a_j \\ w_j &= w_j - h_{ij} q_i \end{aligned}\right\} \quad i = 1 : j-1 \quad \Rightarrow \quad \left\{\begin{aligned} & w_j = a_j - \sum_{i=1}^{j-1} h_{ij} q_i \\ & q_1^* w_j = q_1^* a_j - h_{1j} = 0 \end{aligned}\right.$$

$$h_{jj} = \|w_j\|$$

$$q_j = \frac{w_j}{h_{jj}}$$

# Modified Gram-Schmidt Method

- Orthogonalization

$$\{a_1, a_2, \ldots, a_{j-1}\} \longrightarrow \{q_1, q_2, \ldots, q_{j-1}\} \quad \text{Orthonormal}$$

1

$$\text{Add a new vector } a_j \longrightarrow q_j \ ?$$

- Modified G-S:

$$w_j = a_j$$

$$\left.\begin{aligned} h_{ij} &= q_i^* w_j \\ w_j &= w_j - h_{ij} q_i \end{aligned}\right\} \; i = 1 : j-1 \longrightarrow \left\{\begin{aligned} w_j &= a_j - \sum_{i=1}^{j-1} h_{ij} q_i \\ q_1^* w_j &= q_1^* a_j - h_{1j} = 0 \end{aligned}\right.$$

$$h_{jj} = \|w_j\|$$

$$q_j = \frac{w_j}{h_{jj}}$$

# Generalized Minimum Residual Method

Algorithm 5: Generalized minimum residual method

%% Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and an initial guess $x \in \mathbb{R}^n$;
$r \leftarrow b - Ax,\ \beta \leftarrow \|r\|,\ q_1 \leftarrow r/\beta;$
**for** $j = 1, 2, \ldots, m$
  $w_j \leftarrow Aq_j;$
  $h_{ij} \leftarrow (w_j, q_i),\ w_j \leftarrow w_j - h_{ij}q_i, \quad i = 1, 2, \ldots, j;$
  $h_{j+1,j} \leftarrow \|w_j\|;$
  **if** $h_{j+1,j} == 0$
    $m \leftarrow j$, **break**;
  **end**
  $q_{j+1} \leftarrow w_j / h_{j+1,j};$
**end**
$\bar{H}_m \leftarrow \{h_{ij}\}_{1\leqslant i\leqslant m+1, 1\leqslant j\leqslant m};$
$y_m \leftarrow \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|;$
**Update:** $x \leftarrow x + Q_m y_m;$

Lucky breakdown: If and only if the iterative solution is exact!

- GMRES: the MGS variant
- GMRES: the Householder variant
- GMRES with restarting (stagnation)
- GMRES with deflated restarting
- GMRES with variable restarting

# Stability of GMRES

- Backward error analysis for GMRES with MGS in finite-precision arithmetic

$$\frac{\|b - Ax_m\|}{\|A\|\,\|x_m\| + \|b\|} \leq O(m^{2.5})\,\varepsilon$$

- Review the forward error for direct methods discussed in Lecture 2
- Finite-precision arithmetic (floating-point calculation) will be discussed in Lecture 5
- Cannot store too many iterations for large linear systems!
- Cannot maintain orthogonality and numerical stability!
- We have to restart the iteration ➔ GMRES($m$)

Ref: Christopher C. Paige, Miroslav Rozložník, and Zdeněk Strakoš. "Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES". SIAM Journal on Matrix Analysis and Applications, 2006; 28:264–284

# Convergence of GMRES

- General results: If $A$ is diagonalizable, i.e. $A = Z\,\Lambda\,Z^{-1}$, then

$$r^{(k)} = b - Ax^{(k)} = r^{(0)} - Aq_{k-1}(A)r^{(0)} = p_k(A)r^{(0)}$$

$$\Rightarrow \quad \|r^{(k)}\| \le \max_{i=1,\ldots,n} |p_k(\lambda_i)|\ \kappa(Z)\ \|r^{(0)}\|$$

- If $A$ is normal, then $Z$ is unitary ➔ $\|Z\| = \|Z^{-1}\| = 1$ ➔ Only need to analyze $p_k(\lambda_i)$

- Classical estimate [Elman 1982; Saad, Schultz 1986]: If $A = Z\Lambda Z^{-1}$ is diagonalizable and $\Lambda := \text{diag}(\lambda_1, \ldots, \lambda_n)$, then

$$\|r^{(k)}\| \le \|r^{(0)}\| \cdot \kappa(Z) \min_{p_k \in \mathcal{P}_k} \max_{i=1,\ldots,n} |p_k(\lambda_i)|.$$

- If $A$ is real and positive definite, then

$$\|r^{(k)}\| \le \|r^{(0)}\| \left(1 - \frac{a}{b}\right)^{\frac{k}{2}}, \quad a = \lambda_{\min}\left(\frac{A + A^T}{2}\right)^2, \quad b = \lambda_{\max}(A^T A).$$

# Convergence Behavior of GMRES

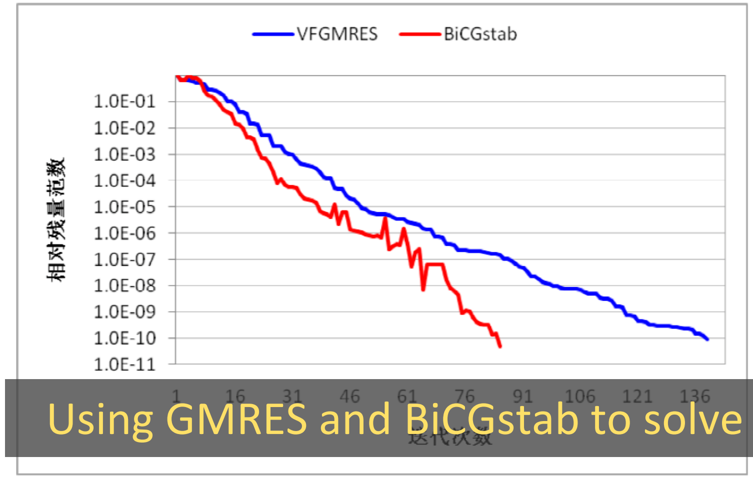
SIAM J. MATRIX ANAL. APPL.
Vol. 17, No. 3, pp. 465–469, July 1996

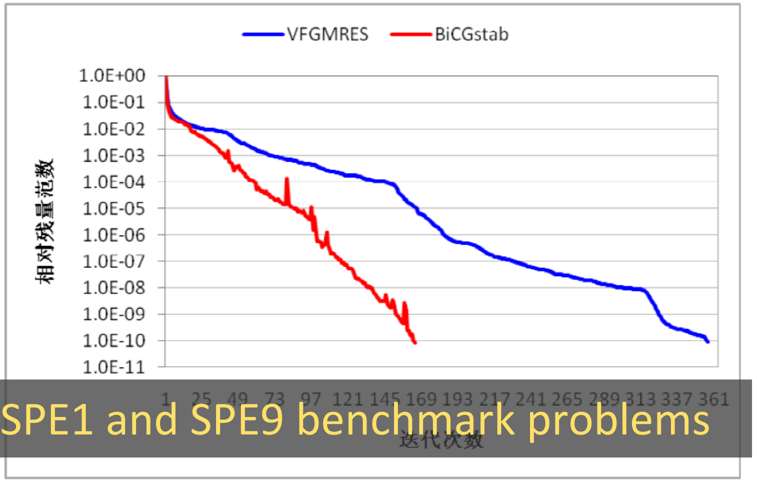

## ANY NONINCREASING CONVERGENCE CURVE IS POSSIBLE FOR GMRES*

ANNE GREENBAUM†, VLASTIMIL PTÁK‡, AND ZDENĚK STRAKOŠ‡

**Abstract.** Given a nonincreasing positive sequence $f(0) \geq f(1) \geq \cdots \geq f(n-1) > 0$, it is shown that there exists an $n$ by $n$ matrix $A$ and a vector $r^0$ with $\|r^0\| = f(0)$ such that $f(k) = \|r^k\|$, $k = 1, \ldots, n-1$, where $r^k$ is the residual at step $k$ of the GMRES algorithm applied to the linear system $Ax = b$, with initial residual $r^0 = b - Ax^0$. Moreover, the matrix $A$ can be chosen to have any desired eigenvalues.

- Need good preconditioners
- If $m$ gets too large, we cannot store everything; need to do truncation or restarting
- Restarts may kill convergence!
- Reuse the previous iterations after restarts? Do not forget everything when restarted
- Different preconditioners at each iteration?




Using GMRES and BiCGstab to solve SPE1 and SPE9 benchmark problems

# Preconditioned GMRES Method

Algorithm 6: GMRES method with right preconditioner

1 `%% Given a nonsingular matrix` $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, `and an initial guess` $x \in \mathbb{R}^n$;
2 $r \leftarrow b - Ax$, $\beta \leftarrow \|r\|$, $q_1 \leftarrow r/\beta$;
3 `for` $j = 1, 2, \ldots, m$
4     $w_j \leftarrow AM^{-1}q_j$;
5     $h_{ij} \leftarrow (w_j, q_i)$, $w_j \leftarrow w_j - h_{ij}q_i$, $\quad i = 1, 2, \ldots, j$;
6     $h_{j+1,j} \leftarrow \|w_j\|$;
7     `if` $h_{j+1,j} == 0$
8         $m \leftarrow j$, `break`;
9     `end`
10     $q_{j+1} \leftarrow w_j / h_{j+1,j}$;
11 `end`
12 $\bar{H}_m \leftarrow \{h_{ij}\}_{1\leqslant i \leqslant m+1, 1 \leqslant j \leqslant m}$;
13 $y_m \leftarrow \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|$;
14 `Update:` $x \leftarrow x + M^{-1}Q_m y_m$;

We do not save the preconditioned vectors because their sizes; instead, we just apply the preconditioner at Line 4 and Line 14. This way, one more precondition step, but less memory needed!

$$\mathcal{K}_m(AM^{-1}, b) := \operatorname{span}\{b, AM^{-1}b, (AM^{-1})^2 b, \ldots, (AM^{-1})^{m-1} b\}$$

# Flexible GMRES Method

Algorithm 7: Flexible GMRES method with right preconditioner

%% Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, and an initial guess $x \in \mathbb{R}^n$;
$r \leftarrow b - Ax, \ \beta \leftarrow \|r\|, \ q_1 \leftarrow r/\beta$;
**for** $j = 1, 2, \ldots, m$
    $z_j \leftarrow M_j^{-1} q_j$;
    $w_j \leftarrow A z_j$;
    $h_{ij} \leftarrow (w_j, q_i), \ w_j \leftarrow w_j - h_{ij} q_i, \quad i = 1, 2, \ldots, j$;
    $h_{j+1,j} \leftarrow \|w_j\|$;
    $q_{j+1} \leftarrow w_j / h_{j+1,j}$;
**end**
$\bar{H}_m \leftarrow \{h_{ij}\}_{1\leqslant i \leqslant m+1, 1 \leqslant j \leqslant m}$;
$y_m \leftarrow \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|$;
Update: $x \leftarrow x + Z_m y_m$;

In order to allow different preconditioners at different steps, we have adjusted the PGMRES method and store the preconditioned vectors in FGMRES.

- FGMRES is useful in practice and will be used later over and over again
- But pay attention: Lucky breakdown no more!

# Finding Eigenvalues using Krylov Methods

- Consider the eigenvalue problem

$$Ax = \lambda x$$

- Approximate the original space using the Krylov subspace

$$x = Q_k y_k + r \approx Q_k y_k$$

- Approximate eigenvalue problem in the Krylov subspace

$$A\,Q_k y_k \approx \lambda\,Q_k y_k$$

$$H_k y_k = Q_k^* A Q_k y_k \approx \lambda y_k$$

$$H_k y = \mu y$$

- This gives a way to approximate the eigenvalues of the original problem

# Take Care of Small Eigenvalues

- Small eigenvalues can cause troubles for Krylov subspace methods
- Add approximate eigenvectors targeting the "smallest eigenvalues" to the Krylov subspace

$$\mathrm{span}\{r, Ar, A^2 r, \ldots, A^{m-1} r, \, v_1, \ldots, v_k\}$$

- Q: How to obtain these eigenvectors?
- Approximate eigen information using Krylov subspaces (e.g. Paige, Parlett, van der Vorst 1995)

$H_m + h_{m+1,m}^2 H_m^{-T} e_m e_m^T$ → Harmonic Ritz values $(\tilde{\theta}_i, \tilde{g}_i)$ → $\tilde{y}_i := Q_m \tilde{g}_i$

Find smallest eigenpairs

Good approximations of $\{v_1, \ldots, v_k\}$

→ $\mathrm{span}\{r, Ar, A^2 r, \ldots, A^{m-1} r, \, \tilde{y}_1, \ldots, \tilde{y}_k\}$ → GMRES-E (Morgan 1995)

# GMRES with Deflated Restarting

- Step 1. Apply the standard GMRES(m)
- Step 2. Find smallest harmonic Ritz values $(\tilde{\theta}_i, \tilde{g}_i)$
- Step 3. Orthonormalize $\{\tilde{g}_i\}_{i=1,\dots,k} \Longrightarrow P_k$ , and append a zero row to it: $(k+1) \times m$
- Step 4. Orthonormalize $c := \beta e_1 - \bar{H}_m y$ against $\bar{P}_k \Longrightarrow P_{k+1}$
- Step 5. Form the deflated subspace for restrarting

$$\bar{H}_k^{\mathrm{new}} := P_{k+1}^T \bar{H}_m P_k, \quad Q_{k+1}^{\mathrm{new}} := Q_{m+1} P_{k+1}$$

- Step 6. Re-orthogonalize and continue with a new Arnoldi iteration (GMRES-DR)

Ref: Morgan, Ronald. “GMRES with Deflated Restarting.” SIAM J. Sci. Comput. 24 (2002): 20-37.

# Performance of GMRES-DR

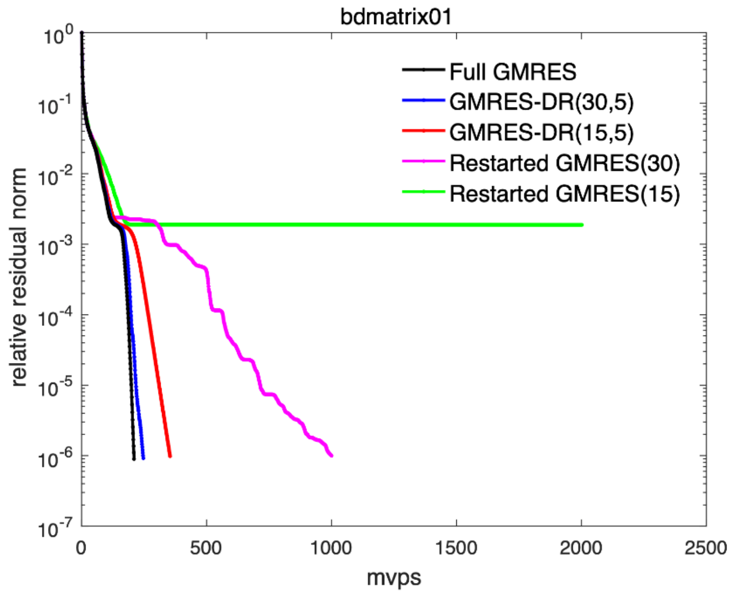
$$A = \begin{pmatrix} 0.1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & \ddots & 0 \\ 0 & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \dots & 999 \end{pmatrix}$$

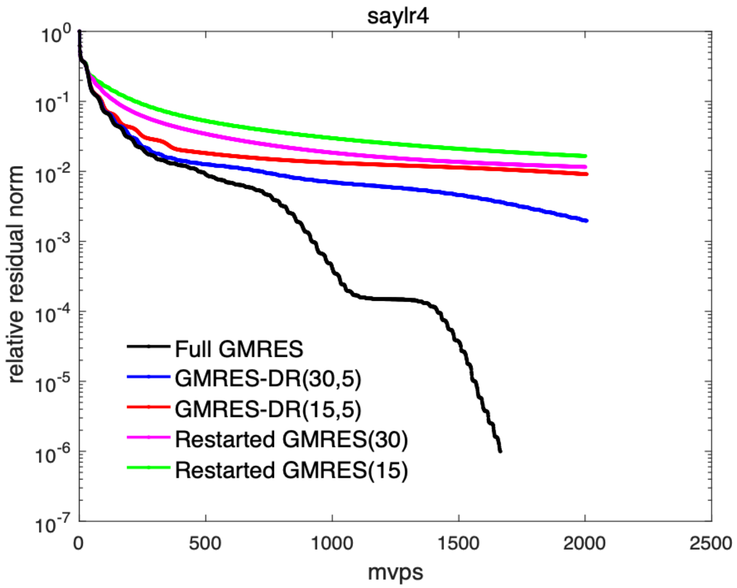
$Ax = b$

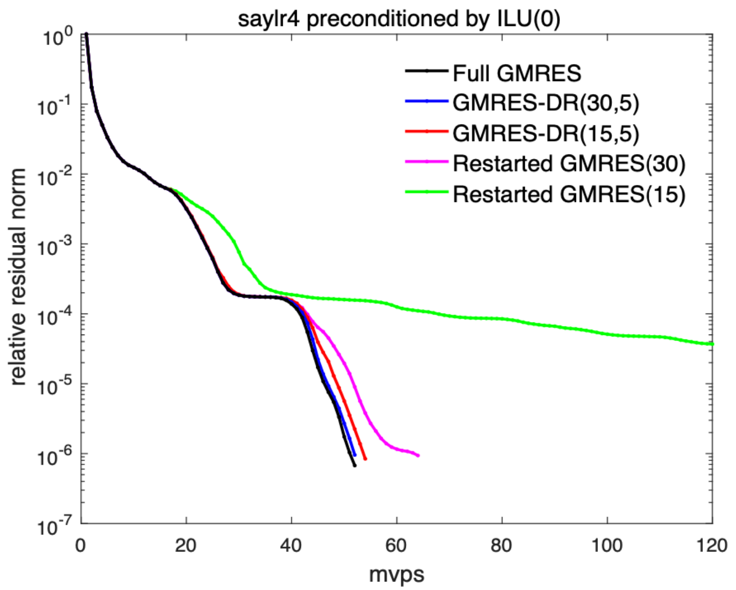
$b$ : generated with normal distribution

- Full GMRES converges
- Restarting may destroy convergence
- Increase restart number may help
- GMRES-DR can improve convergence
- Variable restarting might help?

Source: 电子科技大学荆燕飞《应用数学中的某些前沿问题》学科前沿课程笔记，2022



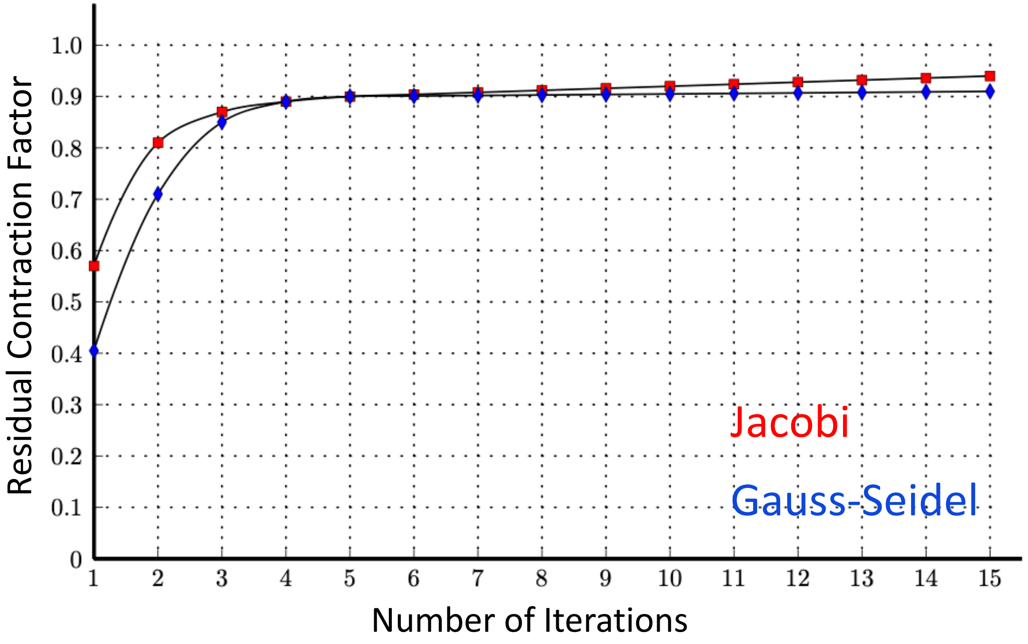



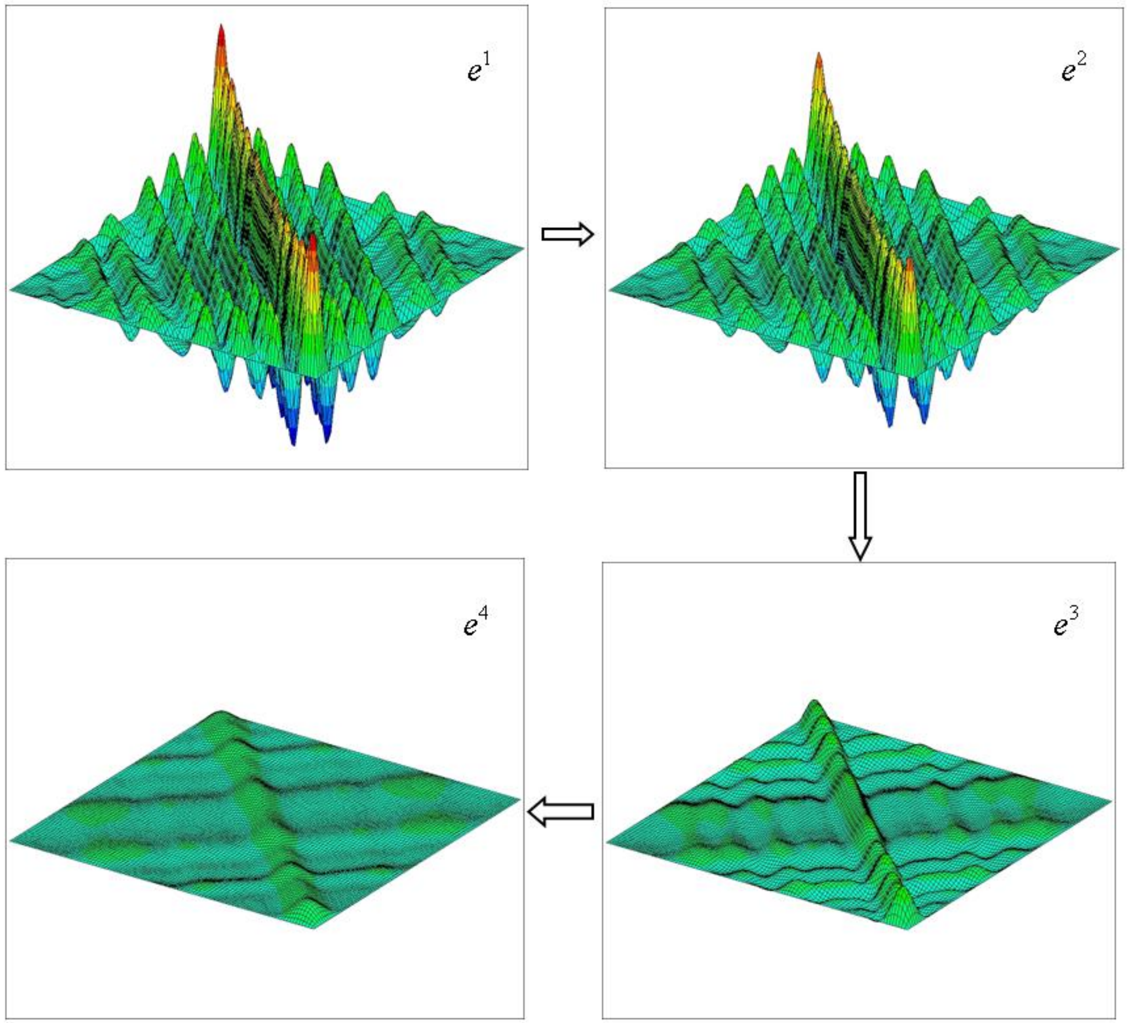
# Multigrid Preconditioners

Multigrid designed for non-symmetric problems

/03

# Simple Iterative Methods




- Simple iterative methods usually slow down after a few iterations (relaxation stage)
- Local relaxation methods have smoothing properties but cannot deal with global error

# Two-Grid Methods

- Step 1. Presmoothing with a simple iterative method (smoother / relaxation)

$$x \leftarrow x + M^{-1}(b - Ax)$$

- Step 2. Coarse grid correction (CGC)

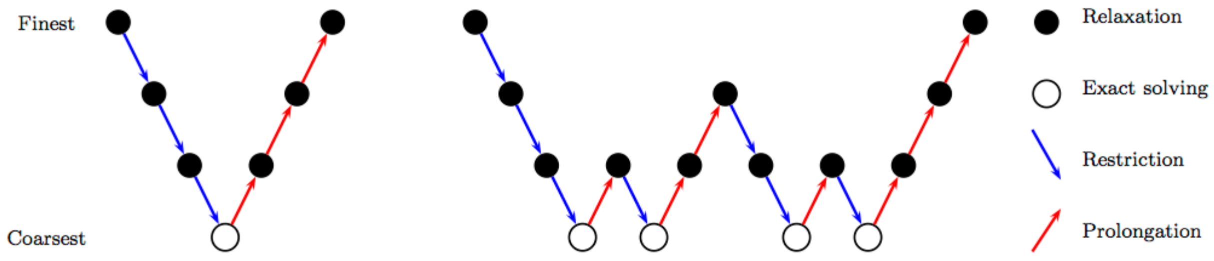
$$x \leftarrow x + PA_c^{-1}P^T(b - Ax) \quad \Longleftarrow \quad A_c e = r$$

- Step 3. Postsmoothing (optional)

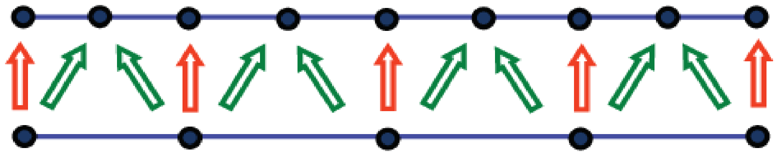
$$x \leftarrow x + M^{-T}(b - Ax)$$

- Does such a simple method work?
- Why does this method work?
- What are the key components that make this method work?
- How to improve this TG method?

# Construction of Coarse Problems




Multilevel cycles (left: V-cycle, right: W-cycle)




$$I_{2h}^{h} = \begin{pmatrix} 1 & & & & \\ 1/2 & 1/2 & & & \\ & 1 & & & \\ & 1/2 & 1/2 & & \\ & & 1 & & \\ & & 1/2 & 1/2 & \\ & & & 1 & \\ & & & 1/2 & 1/2 \\ & & & & 1 \end{pmatrix}$$

Piecewise linear interpolation

Constructing coarse problems:

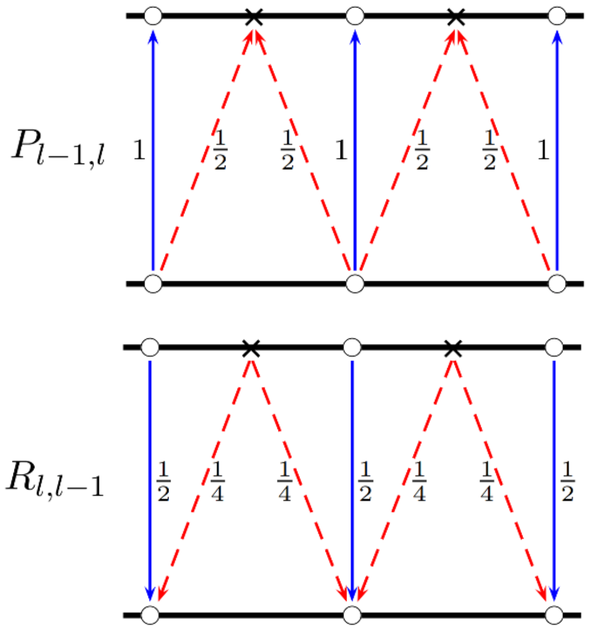
- Coarse problems can be directly defined on the coarse mesh in some cases
- Coarse problems can also be constructed using transfer operators and the following relation

$$A_c := RAP = P^T AP$$

Galerkin relation for SPD problems

Question: Why geometric multigrid methods don't usually work this way?

# Simple Numerical Tests on GMG




Prolongation

Restriction

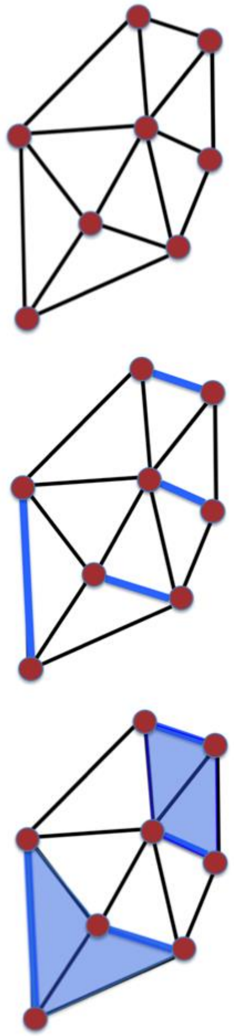
| #Levels | #DOF | #Iter | Contract factor |
|---|---|---|---|
| 5 | 31 | 4 | 0.0257 |
| 6 | 63 | 4 | 0.0259 |
| 7 | 127 | 4 | 0.0260 |
| 8 | 255 | 4 | 0.0260 |
| 9 | 511 | 4 | 0.0261 |
| 10 | 1023 | 4 | 0.0262 |

Table: Convergence behavior of 1D geometric multigrid method.

- GMG usually avoids storing stiffness matrices and other operators
- GMG often can achieve the so-called “textbook multigrid performance”
- GMG converges uniformly and has optimal complexity
- But GMG methods are usually constructed based on hierarchical meshes

It is certainly a Schwarz method, but on multiple levels!

# Algebraic Multigrid



- If a hierarchical mesh is available, then we can construct coarse levels easily
- If only the stiffness matrix is available, then we must construct the hierarchy in an algebraic manner

- Tentative prolongation

$$P_{ij} := \begin{cases} 1, & i \in \mathcal{A}_j \\ 0, & i \notin \mathcal{A}_j \end{cases} \qquad P := \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \qquad R := P^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- Coarser matrix

$$A_c := RAP = P^T AP$$

# Numerical Experiments and Comparisons

Direct method vs AMG methods vs AMG Preconditioned GMRES methods

| EAFE Method | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mu$ | N | MUMPS | CAMG | | CAMG+GMRES | | AIR2 | | AIR2+GMRES | |
| | | Time (s) | NumIt | Time (s) | NumIt | Time (s) | NumIt | Time (s) | NumIt | Time (s) |
| $10^0$ | 256X256 | 0.93 | 23 | 0.18 | 11 | 0.13 | 11 | 0.21 | 8 | 0.24 |
| | 512X512 | 4.33 | 23 | 0.79 | 10 | 0.61 | 10 | 0.94 | 8 | 1.09 |
| | 1024X1024 | 25.28 | 23 | 3.68 | 10 | 2.85 | 11 | 4.79 | 8 | 5.23 |
| | 2048X2048 | 157.77 | 22 | 16.23 | 10 | 14.15 | 10 | 21.52 | 8 | 23.61 |
| $10^{-2}$ | 256X256 | 0.93 | 16 | 0.13 | 10 | 0.13 | 8 | 0.19 | 6 | 0.23 |
| | 512X512 | 4.41 | 16 | 0.64 | 10 | 0.6 | 8 | 0.88 | 6 | 1.03 |
| | 1024X1024 | 26.08 | 16 | 3.03 | 10 | 2.86 | 8 | 4.28 | 6 | 4.87 |
| | 2048X2048 | 158.03 | 16 | 15.02 | 9 | 12.57 | 7 | 19.38 | 6 | 22.2 |
| $10^{-4}$ | 256X256 | 0.98 | x | x | 14 | 0.26 | 7 | 0.39 | 5 | 0.52 |
| | 512X512 | 5.02 | 65 | 2.69 | 15 | 1.17 | 9 | 2.55 | 5 | 3.43 |
| | 1024X1024 | 29.66 | 13 | 3.06 | 13 | 3.52 | 11 | 11.91 | 6 | 14.88 |
| | 2048X2048 | 205.16 | 8 | 11.74 | 9 | 13.45 | 23 | 48.48 | 5 | 40.26 |
| $10^{-6}$ | 256X256 | 0.91 | x | x | 16 | 0.28 | 12 | 0.24 | 5 | 0.22 |
| | 512X512 | 4.32 | x | x | 16 | 1.27 | 8 | 0.83 | 5 | 0.89 |
| | 1024X1024 | 23.27 | x | x | 15 | 5.53 | 9 | 3.97 | 5 | 4.11 |
| | 2048X2048 | 143.19 | x | x | 15 | 25.49 | 13 | 21.52 | 7 | 20.96 |

# Reading and Thinking

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 4, pp. 962–984


## A TECHNIQUE FOR ACCELERATING THE CONVERGENCE OF RESTARTED GMRES*

A. H. BAKER†, E. R. JESSUP‡, AND T. MANTEUFFEL§

**Abstract.** We have observed that the residual vectors at the end of each restart cycle of restarted GMRES often alternate direction in a cyclic fashion, thereby slowing convergence. We present a new technique for accelerating the convergence of restarted GMRES by disrupting this alternating pattern. The new algorithm resembles a full conjugate gradient method with polynomial preconditioning, and its implementation requires minimal changes to the standard restarted GMRES algorithm.

**Key words.** GMRES, iterative methods, Krylov subspace, restart, nonsymmetric linear systems

**AMS subject classification.** 65F10

**DOI.** 10.1137/S0895479803422014

- Do you need to solve non-symmetric problems in your applications?
- What linear solvers do you use for solving these problems?
- Have you used some of the Krylov methods to solve non-symmetric problems?
- How do they perform? How do they scale?
- If not yet, try …

# Contact Me

- Office hours: Mon 14:00—15:00
- Walk-in or online with appointment
- zhangcs@lsec.cc.ac.cn
- http://lsec.cc.ac.cn/~zhangcs

My sincere gratitude to:

Liuqiang Zhong, Yanfei Jing, Xiaoqiang Yue, Bin Dai

# Review

- Why the Krylov matrix approach is rarely used in practice:
    - Nearly singular (It's a power sequence)
    - Expensive to compute the full QR factorization
- The reorthogonalized Arnoldi iteration is often used to construct the GMRES method:
    - The modified Gram-Schmidt method is numerically stable
    - Terminate at any time → No need to compute the full QR factorization
    - $m$ is usually much smaller than $n$ → tall skinny matrices
    - For symmetric problems, $H_m$ reduces to a symmetric tridiagonal matrix → Lanczos methods
- Multigrid methods for nonsymmetric problems
    - Better with relaxation methods for nonsymmetric problems (GMG)
    - Use symmetrized problem or other available information to construct preconditioner

中国科学院大学
夏季强化课程
2022

# Fast Solvers for Large Algebraic Systems

THANKS

Chensong Zhang, AMSS
http://lsec.cc.ac.cn/~zhangcs

Release version 2022.07.12