

Numerical Methods for
Hyperbolic and Parabolic Conservation Laws
Linear System Solver
Part II: GMRES, BiCG and Variants

Andreas Meister

UMBC, Department of Mathematics and Statistics

- Methods for symmetric, positive definite Matrices
 - Method of steepest descent
 - Method of conjugate directions
 - CG-scheme
- Methods for non-singular Matrices
 - GMRES
 - BiCG, CGS and BiCGSTAB
- Preconditioning
 - ILU, IC, GS, SGS, ...

Projection method & Krylov subspace approach

We consider

$$Ax = b$$

with given data $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

Splitting methods	Projection methods
Looking for approximations $x_m \in \mathbb{R}^n$	Looking for approximations $x_m \in x_0 + K_m \subset \mathbb{R}^n$ $\dim K_m = m \leq n$
Numerical algorithm $x_{m+1} = Mx_m + Nb$	Numerical algorithm (orthogonality constraint) $b - Ax_m \perp L_m \subset \mathbb{R}^n$ $\dim L_m = m \leq n$

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; B --> D[CG-Squared method (CGS)]; B --> E[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; CG["Method of conjugate gradients (CG)"] --> BiCG["Bi-conjugate gradients method (BiCG)"]; CG --> GMRES["Generalized Minimal Residual method (GMRES)"]; BiCG --> CGS["CG-Squared method (CGS)"]; BiCG --> BiCGSTAB["Bi-CG Stabilized method (BiCGSTAB)"]; style BiCG stroke-dasharray: 5 5; style CGS stroke-dasharray: 5 5; style BiCGSTAB stroke-dasharray: 5 5;
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Generalized Minimal Residual (GMRES)

Basic idea:

- Search for $x_m = \arg \min_{x \in x_0 + K_m} F(x)$
→ $K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$

- Instead of

$$F(x) = \frac{1}{2}(Ax, x) - (b, x) \quad [\text{CG} - \text{method}]$$

we introduce

$$F(x) = \|b - Ax\|_2^2$$

Properties and Consequences:

- $\|y\| \geq 0$, $\forall y \in \mathbb{R}^n$ and $\|y\| = 0 \Leftrightarrow y = 0$ yield

$$F(x) \geq 0 \text{ and } F(x) = 0 \Leftrightarrow b - Ax = 0 \Leftrightarrow x = A^{-1}b$$

- $x_m = \arg \min_{x \in x_0 + K_m} F(x) \Leftrightarrow b - Ax_m \perp AK_m$

⇒ Skew Krylov subspace method

Generalized Minimal Residual (GMRES)

Basic idea:

- Search for $x_m = \arg \min_{x \in x_0 + K_m} F(x)$
→ $K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$

- Instead of

$$F(x) = \frac{1}{2}(Ax, x) - (b, x) \quad [\text{CG - method}]$$

we introduce

$$F(x) = \|b - Ax\|_2^2$$

Properties and Consequences:

- $\|y\| \geq 0$, $\forall y \in \mathbb{R}^n$ and $\|y\| = 0 \Leftrightarrow y = 0$ yield

$$F(x) \geq 0 \text{ and } F(x) = 0 \Leftrightarrow b - Ax = 0 \Leftrightarrow x = A^{-1}b$$

- $x_m = \arg \min_{x \in x_0 + K_m} F(x) \Leftrightarrow b - Ax_m \perp AK_m$

⇒ Skew Krylov subspace method

Generalized Minimal Residual (GMRES)

Basic idea:

- Search for $x_m = \arg \min_{x \in x_0 + K_m} F(x)$
→ $K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$

- Instead of

$$F(x) = \frac{1}{2}(Ax, x) - (b, x) \quad [CG - method]$$

we introduce

$$F(x) = \|b - Ax\|_2^2$$

Properties and Consequences:

- $\|y\| \geq 0$, $\forall y \in \mathbb{R}^n$ and $\|y\| = 0 \Leftrightarrow y = 0$ yield

$$F(x) \geq 0 \text{ and } F(x) = 0 \Leftrightarrow b - Ax = 0 \Leftrightarrow x = A^{-1}b$$

- $x_m = \arg \min_{x \in x_0 + K_m} F(x) \Leftrightarrow b - Ax_m \perp AK_m$

⇒ Skew Krylov subspace method

Generalized Minimal Residual (GMRES)

Procedure:

- Calculate an **ONB** v_1, \dots, v_m of K_m
- Write $x_m \in x_0 + K_m$ in the form

$$x_m = x_0 + \sum_{j=1}^m \alpha_j v_j = x_0 + V_m \alpha^m$$

where $V_m = (v_1 \dots v_m) \in \mathbb{R}^{n \times m}$, $\alpha^m = (\alpha_1, \dots, \alpha_m)^T \in \mathbb{R}^m$

Consequence:

- Find $x_m \in x_0 + K_m \subset \mathbb{R}^n$ satisfying

$$F(x_m) \leq F(x) := \|b - Ax\|_2^2 \quad \forall x \in \mathbb{R}^n$$

\iff

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

Generalized Minimal Residual (GMRES)

Procedure:

- Calculate an ONB v_1, \dots, v_m of K_m
- Write $x_m \in x_0 + K_m$ in the form

$$x_m = x_0 + \sum_{j=1}^m \alpha_j v_j = x_0 + V_m \alpha^m$$

where $V_m = (v_1 \dots v_m) \in \mathbb{R}^{n \times m}$, $\alpha^m = (\alpha_1, \dots, \alpha_m)^T \in \mathbb{R}^m$

Consequence:

- Find $x_m \in x_0 + K_m \subset \mathbb{R}^n$ satisfying

$$F(x_m) \leq F(x) := \|b - Ax\|_2^2 \quad \forall x \in \mathbb{R}^n$$

\iff

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2^2 \quad \forall \alpha \in \mathbb{R}^m$$

Generalized Minimal Residual (GMRES)

Procedure:

- Calculate an **ONB** v_1, \dots, v_m of K_m
- Write $x_m \in x_0 + K_m$ in the form

$$x_m = x_0 + \sum_{j=1}^m \alpha_j v_j = x_0 + V_m \alpha^m$$

where $V_m = (v_1 \dots v_m) \in \mathbb{R}^{n \times m}$, $\alpha^m = (\alpha_1, \dots, \alpha_m)^T \in \mathbb{R}^m$

Consequence:

- Find $x_m \in x_0 + K_m \subset \mathbb{R}^n$ satisfying

$$F(x_m) \leq F(x) := \|b - Ax\|_2^2 \quad \forall x \in \mathbb{R}^n$$

\iff

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

Orthonormal basis (ONB) — Arnoldi-Algorithm

Sought after:

ONB v_1, \dots, v_m of Krylov subspace $K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}$

Assume: v_1, \dots, v_j represents an ONB of K_j for $j < m$

Aim: Calculation of v_{j+1}

Idea:

$$\begin{aligned} AK_j &= A \text{span}\{r_0, Ar_0, \dots, A^{j-1}r_0\} = \text{span}\{Ar_0, A^2r_0, \dots, A^j r_0\} \\ &\subset \text{span}\{r_0, Ar_0, \dots, A^j r_0\} = K_{j+1} \end{aligned}$$

and

$$AK_j = A \text{span}\{v_1, \dots, v_j\} = \text{span}\{Av_1, \dots, Av_j\}$$

Conclusion:

Use Av_j for the calculation of v_{j+1}

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Ansatz:

$$v_{j+1} = Av_j + \xi \quad \text{with } \xi \in K_j = \text{span}\{v_1, \dots, v_j\}$$

Using the formulation

$$\xi = - \sum_{i=1}^j h_{ij} v_i, \quad h_{ij} \in \mathbb{R} \quad \Longrightarrow \quad v_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i$$

Orthogonality: For $s = 1, \dots, j$:

$$0 \stackrel{!}{=} (v_s, v_{j+1}) = (v_s, Av_j) - \sum_{i=1}^j h_{ij} (v_s, v_i) \stackrel{ONB}{=} (v_s, Av_j) - h_{sj} \underbrace{(v_s, v_s)}_{=1}$$

$$\Longrightarrow h_{sj} = (v_s, Av_j), \quad s = 1, \dots, j$$

Concluding:

v_{j+1} has to be normalized

Orthonormal basis (ONB) — Arnoldi-Algorithm

Arnoldi-Algorithm

$v_1 := \frac{r_0}{\ r_0\ _2}$	
Für $j = 1, \dots, m$	
Für $i = 1, \dots, j$	
$h_{ij} := (v_i, Av_j)_2 \quad (4.3.30)$	
$w_j := Av_j - \sum_{i=1}^j h_{ij} v_i \quad (4.3.31)$	
$h_{j+1,j} := \ w_j\ _2 \quad (4.3.32)$	
Y	$h_{j+1,j} \neq 0$
$v_{j+1} := \frac{w_j}{h_{j+1,j}} \quad (4.3.33)$	$v_{j+1} := \mathbf{0}$
	STOP
	N

Orthonormal basis (ONB) — Arnoldi-Algorithm

Disadvantage: Increasing storage requirements for

$$V_m = (v_1 \dots v_m) \in \mathbb{R}^{n \times m}$$

Helpful properties:

$$(1) \quad H_m = V_m^T A V_m \quad \text{with} \quad H_m = \begin{pmatrix} h_{11} & \dots & \dots & \dots & h_{1m} \\ h_{21} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{m,m-1} & h_{mm} \end{pmatrix}$$

$$(2) \quad A V_m = V_{m+1} \bar{H}_m \quad \text{with} \quad \bar{H}_m = \begin{pmatrix} & & H_m & & \\ 0 & \dots & 0 & h_{m+1,m} & \end{pmatrix} \in \mathbb{R}^{(m+1) \times m}$$

Orthonormal basis (ONB) — Arnoldi-Algorithm

Helpful properties:

(3) Orthogonal matrix $Q_m = G_m \cdot \dots \cdot G_1 \in \mathbb{R}^{(m+1) \times (m+1)}$ (Givens) with

$$Q_m \bar{H}_m = \bar{R}_m \text{ with } \bar{R}_m = \begin{pmatrix} R_m & & \\ 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{(m+1) \times m}$$

$$R_m = \begin{pmatrix} r_{11} & \dots & \dots & r_{1m} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & r_{mm} \end{pmatrix} \in \mathbb{R}^{m \times m} \text{ non-singular}$$

(4) B consists of orthonormal columns

$$\Rightarrow \|Bx\|_2 = \|x\|_2.$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- Introducing $g = (g_1, \dots, g_{m+1})^T := Q_m(\|r_0\|_2 e_1)$, $e_1 = (1, 0, \dots, 0)^T$

$$J(\alpha) = \|b - A(x_0 + V_m \alpha)\|_2 \stackrel{r_0 = b - Ax_0}{=} \|r_0 - AV_m \alpha\|_2$$

$$\stackrel{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m \alpha\|_2 \stackrel{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2$$

$$\stackrel{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \bar{H}_m \alpha)\|_2 = \|g - Q_m \bar{H}_m \alpha\|_2$$

$$\stackrel{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ with $g = Q_m(\|r_0\|_2 e_1)$

- Optimal α :

$$\alpha^m = R_m^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

- Residual

$$J(\alpha^m) = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ with $g = Q_m(\|r_0\|_2 e_1)$

- Optimal α :

$$\alpha^m = R_m^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

- Residual

$$J(\alpha^m) = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ with $g = Q_m(\|r_0\|_2 e_1)$

- Optimal α :

$$\alpha^m = R_m^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

- Residual

$$J(\alpha^m) = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ with $g = Q_m(\|r_0\|_2 e_1)$

- Optimal α :

$$\alpha^m = R_m^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

- Residual

$$J(\alpha^m) = |g_{m+1}|$$

Generalized Minimal Residual (GMRES)

- Find $\alpha^m \in \mathbb{R}^m$ satisfying

$$J(\alpha^m) \leq J(\alpha) := \|b - A(x_0 + V_m \alpha)\|_2 \quad \forall \alpha \in \mathbb{R}^m$$

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \dots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ with $g = Q_m(\|r_0\|_2 e_1)$

- Optimal α :

$$\alpha^m = R_m^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

- Residual

$$J(\alpha^m) = |g_{m+1}|$$

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$

$x = x_0 + V_j \alpha^j$

STOP

$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$

$x = x_0 + V_m \alpha^m$

$x_0 = x, r_0 = b - Ax_0$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Algorithm – GMRES with Restart

Choose $x_0 \in \mathbb{R}^n$, calculate $r_0 = b - Ax_0$

Restart = 0

While Restart < Max. Restarts

For $j = 1, \dots, nm$

$(m \ll n)$

Extend ONB V_j

(Arnoldi)

Extend \bar{H}_{j-1} zu \bar{H}_j

(Arnoldi)

Calculate $\bar{R}_j = Q_j \bar{H}_j$

(Givens)

Calculate $(g_1, \dots, g_{j+1})^T = \|r_0\|_2 Q_j e_1$

(Givens)

If $|g_{j+1}| \leq \varepsilon$

(given tolerance)

$$\alpha^j = R_j^{-1}(g_1, \dots, g_j)^T$$

$$x = x_0 + V_j \alpha^j$$

STOP

$$\alpha^m = R_m^{-1}(g_1, \dots, g_m)^T$$

$$x = x_0 + V_m \alpha^m$$

$$x_0 = x, r_0 = b - Ax_0$$

Increase Restart by 1

Convection-Diffusion Equation

Governing Equation

$$\beta \cdot \nabla u(x, y) - \epsilon \Delta u(x, y) = 0 \quad \text{on } D = (0, 1) \times (0, 1)$$

with

$$\beta = \alpha \begin{pmatrix} \cos \frac{\pi}{4} \\ \sin \frac{\pi}{4} \end{pmatrix} \quad \alpha, \epsilon \in \mathbb{R}_0^+$$

Boundary Conditions

$$u(x, y) = x^2 + y^2 \quad \text{for } (x, y) \in \partial D$$

Mesh

$$x_i = i \cdot h \quad \text{and} \quad y_j = j \cdot h \quad \text{for } j = 0, \dots, N+1, \quad h = \frac{1}{N+1}$$

Convection-Diffusion Equation

Discretization of Laplacian (Central Difference)

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx \frac{1}{h^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j})$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) \approx \frac{1}{h^2}(u_{i,j+1} - 2u_{ij} + u_{i,j-1})$$

Discretization of convective part (Backward Difference)

$$\frac{\partial u}{\partial x}(x_i, y_j) \approx \frac{1}{h}(u_{i,j} - u_{i-1,j})$$

$$\frac{\partial u}{\partial y}(x_i, y_j) \approx \frac{1}{h}(u_{i,j} - u_{i,j-1})$$

Convection-Diffusion Equation

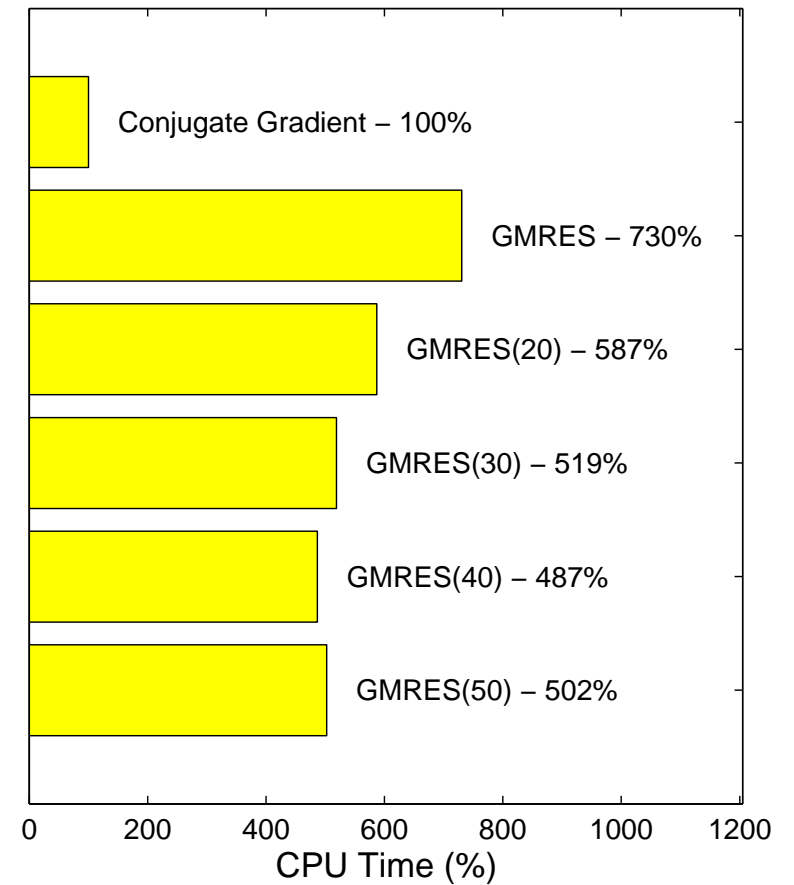
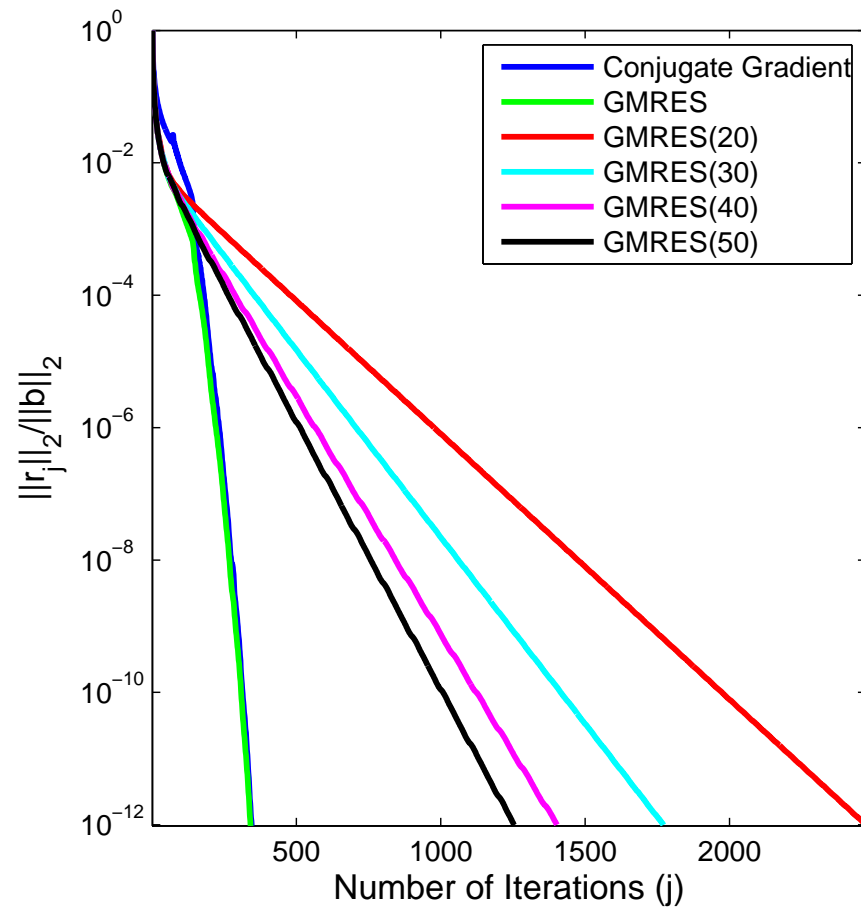
Testcases

	α	ϵ	Matrix properties
Test 1	0	1	Symmetric, positive definite
Test 2	0.1	1	Non-symmetric, non-singular
Test 3	1	0.1	Non-symmetric, non-singular

- Number of unknowns: $100 \times 100 = 10000$ ($N = 100$)
- Stopping criterion: $\|r_j\|_2 < 10^{-12} \|b\|$

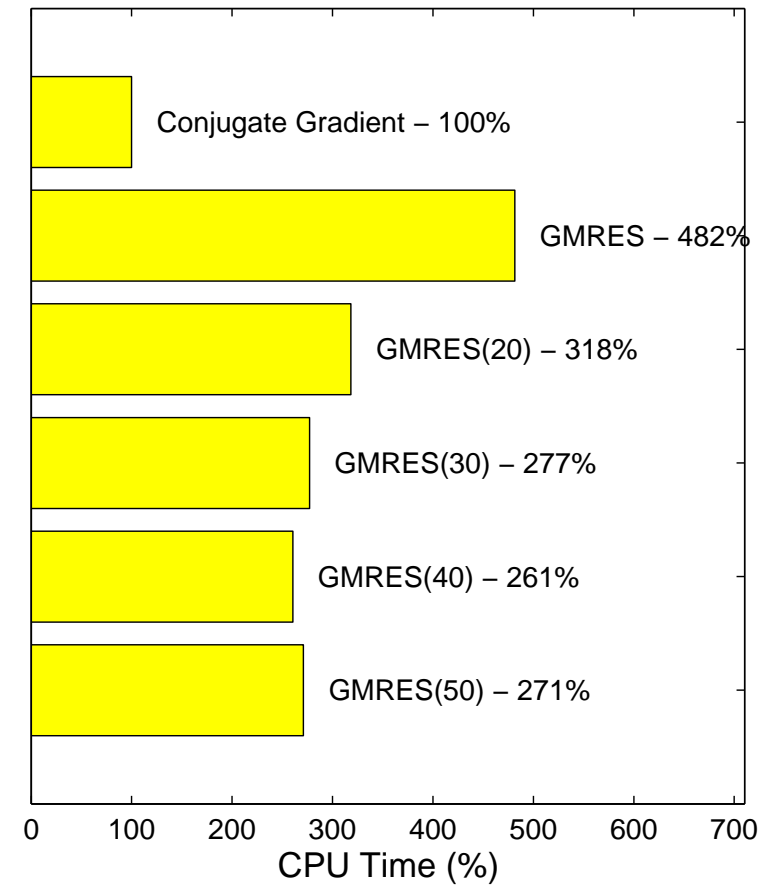
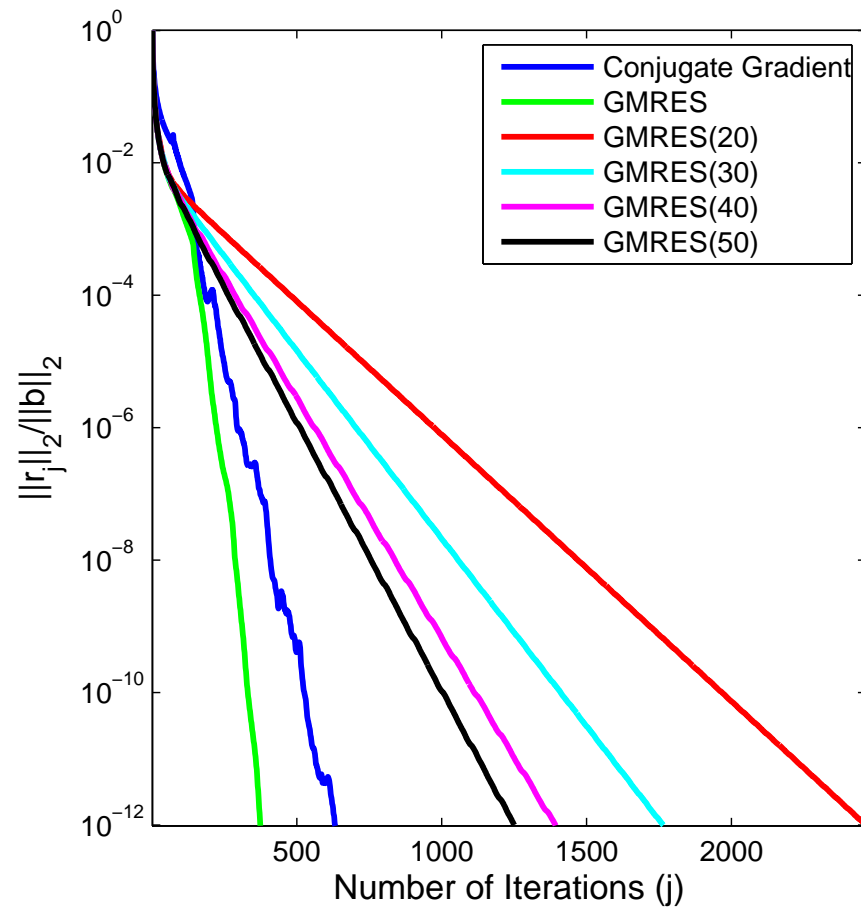
Comparison of CG, GMRES and GMRES(m)

Test 1: Pure Diffusion ($\alpha = 0, \epsilon = 1$)



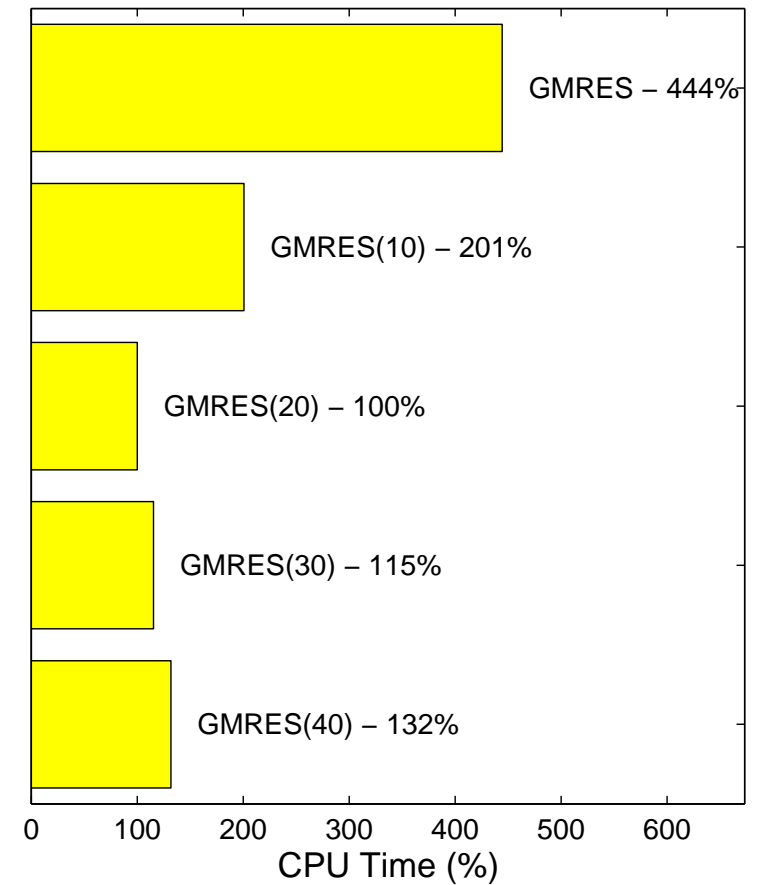
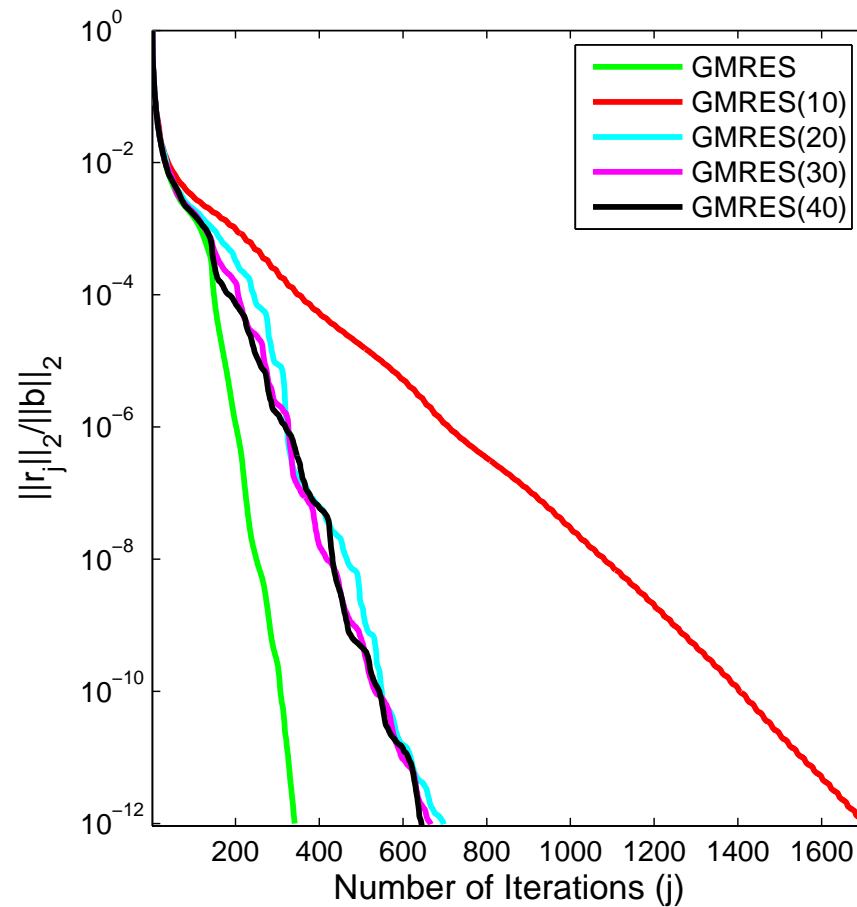
Comparison of CG, GMRES and GMRES(m)

Test 2: Weak Convection-Diffusion ($\alpha = 0.1, \epsilon = 1$)



Comparison of CG, GMRES and GMRES(m)

Test 3: Convection-Diffusion ($\alpha = 1, \epsilon = 0.1$)



Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; B --> D[CG-Squared method (CGS)]; B --> E[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

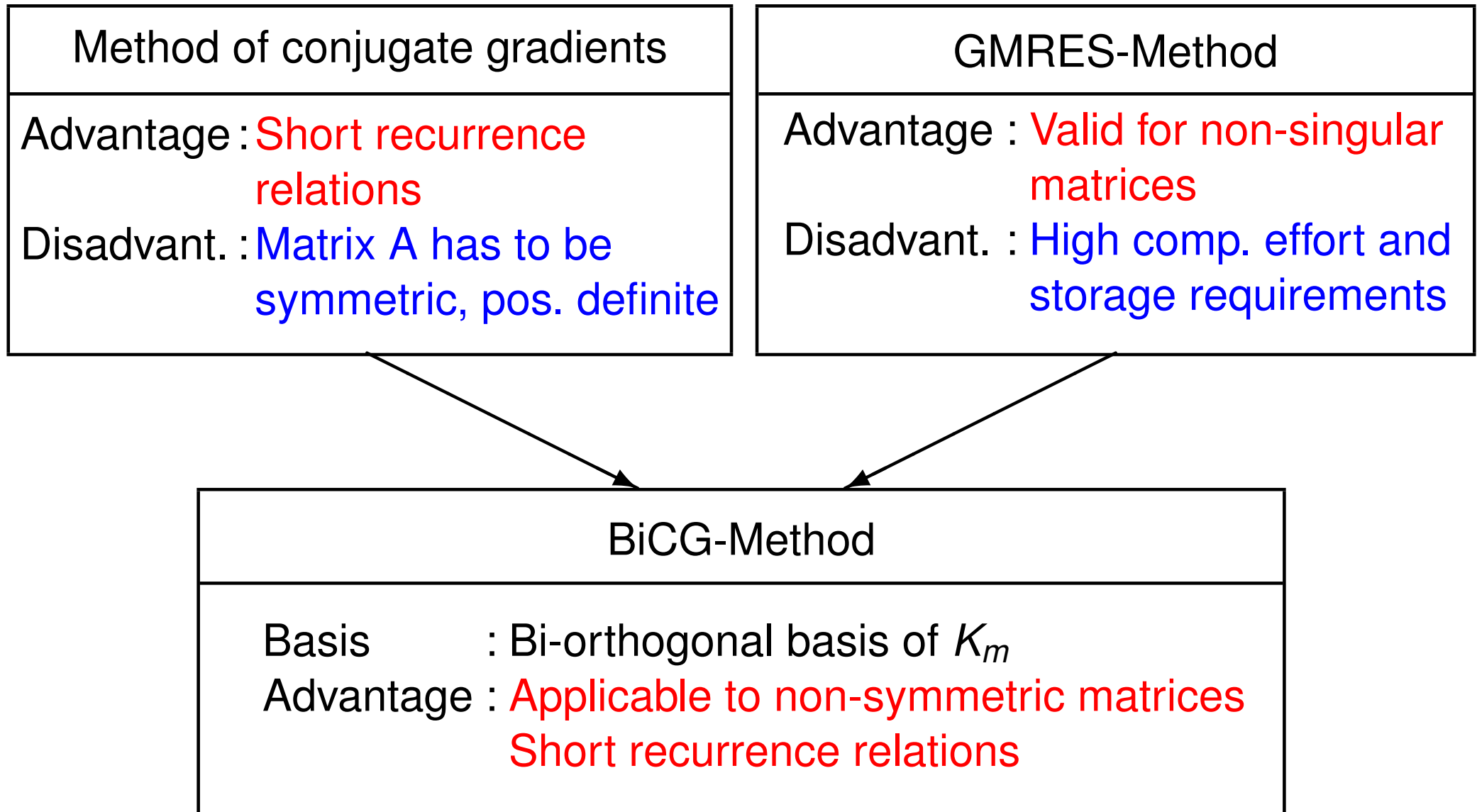
Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Bi-conjugate gradient method (BiCG)



Orthonormal basis (ONB) — Lanczos-Algorithm

- Lanczos-Algorithm = Arnoldi-Algorithm for symmetric matrices A

$$\rightarrow H_m = V_m^T A V_m = \begin{pmatrix} a_1 & c_2 & & \\ c_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_m \\ & & c_m & a_m \end{pmatrix}$$

$$\rightarrow w_j = A v_j - \underbrace{h_{j-1,j}}_{=c_j} v_{j-1} - \underbrace{h_{jj}}_{=a_j} v_j.$$

Orthonormal basis (ONB) — Lanczos-Algorithm

Lanczos-Algorithm

$$v_1 := \frac{r_0}{\|r_0\|_2}, c_1 := 0, v_0 := \mathbf{0}$$

Für $j = 1, \dots, m$

$$w_j := Av_j - c_j v_{j-1}$$

$$a_j := (w_j, v_j)_2$$

$$w_j := w_j - a_j v_j$$

$$c_{j+1} := \|w_j\|_2$$

Y $c_{j+1} \neq 0$

N

$$v_{j+1} := \frac{w_j}{c_{j+1}}$$

$$v_{j+1} := \mathbf{0}$$

STOP

Bi-orthonormal basis — BiLanczos-Algorithm

Definition: Bi-orthonormal

The vectors $v_1, \dots, v_m \in \mathbb{R}^n$ and $w_1, \dots, w_m \in \mathbb{R}^n$ are called bi-orthonormal, if

$$(v_i, w_j) = \delta_{ij}, \quad i, j = 1, \dots, m$$

holds.

Idea:

Simultaneous calculation of bi-orthonormal bases

$$v_1, \dots, v_m \quad \text{of} \quad K_m = \text{span}\{r_0, \dots, A^{m-1}r_0\}$$

$$w_1, \dots, w_m \quad \text{of} \quad K_m^T = \text{span}\{r_0, A_{r_0}^T, \dots, (A^T)^{m-1}r_0\}.$$

Bi-orthonormal basis — BiLanczos-Algorithm

Definition: Bi-orthonormal

The vectors $v_1, \dots, v_m \in \mathbb{R}^n$ and $w_1, \dots, w_m \in \mathbb{R}^n$ are called bi-orthonormal, if

$$(v_i, w_j) = \delta_{ij}, \quad i, j = 1, \dots, m$$

holds.

Idea:

Simultaneous calculation of bi-orthonormal bases

$$v_1, \dots, v_m \quad \text{of} \quad K_m = \text{span}\{r_0, \dots, A^{m-1}r_0\}$$

$$w_1, \dots, w_m \quad \text{of} \quad K_m^T = \text{span}\{r_0, A_{r_0}^T, \dots, (A^T)^{m-1}r_0\}.$$

Bi-orthonormal basis — BiLanczos-Algorithm

Idea:

Simultaneous calculation of bi-orthonormal bases

$$v_1, \dots, v_m \quad \text{of} \quad K_m = \text{span}\{r_0, \dots, A^{m-1}r_0\},$$

$$w_1, \dots, w_m \quad \text{of} \quad K_m^T = \text{span}\{r_0, A^T r_0, \dots, (A^T)^{m-1}r_0\}.$$

Effect:

- No symmetry constraint, BiLanczos = Lanczos, if A symmetric
- v_1, \dots, v_m are not orthonormal

- $W_m^T A V_m = T_m = \begin{pmatrix} \ddots & \ddots & & 0 \\ \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}$

Biorthonormal basis — BiLanczos-Algorithm

BiLanczos-Algorithm

Bi-Lanczos-Algorithmus —

$$h_{1,0} = h_{0,1} := 0$$

$$\mathbf{v}_0 = \mathbf{w}_0 := \mathbf{0}$$

$$\mathbf{v}_1 = \mathbf{w}_1 := \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|_2} \quad (4.3.48)$$

für $j = 1, \dots, m$

$$h_{jj} := (\mathbf{w}_j, \mathbf{A} \mathbf{v}_j)_2 \quad (4.3.49)$$

$$\mathbf{v}_{j+1}^* := \mathbf{A} \mathbf{v}_j - h_{jj} \mathbf{v}_j - h_{j-1,j} \mathbf{v}_{j-1} \quad (4.3.50)$$

$$\mathbf{w}_{j+1}^* := \mathbf{A}^T \mathbf{w}_j - h_{jj} \mathbf{w}_j - h_{j,j-1} \mathbf{w}_{j-1} \quad (4.3.51)$$

$$h_{j+1,j} := |(\mathbf{v}_{j+1}^*, \mathbf{w}_{j+1}^*)_2|^{1/2}$$

	$h_{j+1,j} \neq 0$	
Y		N

$h_{j,j+1} := \frac{(\mathbf{v}_{j+1}^*, \mathbf{w}_{j+1}^*)_2}{h_{j+1,j}} \quad (4.3.52)$	$h_{j,j+1} := 0$
--	------------------

$\mathbf{v}_{j+1} := \frac{\mathbf{v}_{j+1}^*}{h_{j+1,j}} \quad (4.3.53)$	$\mathbf{v}_{j+1} := \mathbf{0}$
---	----------------------------------

$\mathbf{w}_{j+1} := \frac{\mathbf{w}_{j+1}^*}{h_{j,j+1}} \quad (4.3.54)$	$\mathbf{w}_{j+1} := \mathbf{0}$
---	----------------------------------

STOP

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus	Arnoldi-Algorithmus
↓ BiCG-Methode	↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- **Ansatz:** Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

$$b - Ax_m \perp K_m^T \text{ and } x_m \in x_0 + K_m.$$

- Reason for short recurrence relations?

Bilanczos-Algorithmus ↓ BiCG-Methode	Arnoldi-Algorithmus ↓ FOM
$x_m = x_0 + V_m \alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m \alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = W_m^T (b - Ax_m)$ $= W_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - T_m \alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \dots, m$ $\Leftrightarrow 0 = V_m^T (b - Ax_m)$ $= V_m^T (r_0 - AV_m \alpha^m)$ $= \ r_0\ _2 e_1 - H_m \alpha^m$
$\alpha^m = T_m^{-1} (\ r_0\ _2 e_1)$	$\alpha^m = H_m^{-1} (\ r_0\ _2 e_1)$

BiCG-Algorithm

BiCG-Algorithmus —

Wähle $\mathbf{x}_0 \in \mathbb{R}^n$ und $\varepsilon > 0$

$$\mathbf{r}_0 = \mathbf{r}_0^* = \mathbf{p}_0 = \mathbf{p}_0^* := \mathbf{b} - \mathbf{A}\mathbf{x}_0$$

$$j := 0$$

Solange $\|\mathbf{r}_j\|_2 > \varepsilon$

$$\alpha_j := \frac{(\mathbf{r}_j, \mathbf{r}_j^*)_2}{(\mathbf{A}\mathbf{p}_j, \mathbf{p}_j^*)_2}$$

$$\mathbf{x}_{j+1} := \mathbf{x}_j + \alpha_j \mathbf{p}_j$$

$$\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j$$

$$\mathbf{r}_{j+1}^* := \mathbf{r}_j^* - \alpha_j \mathbf{A}^T \mathbf{p}_j^*$$

$$\beta_j := \frac{(\mathbf{r}_{j+1}, \mathbf{r}_{j+1}^*)_2}{(\mathbf{r}_j, \mathbf{r}_j^*)_2}$$

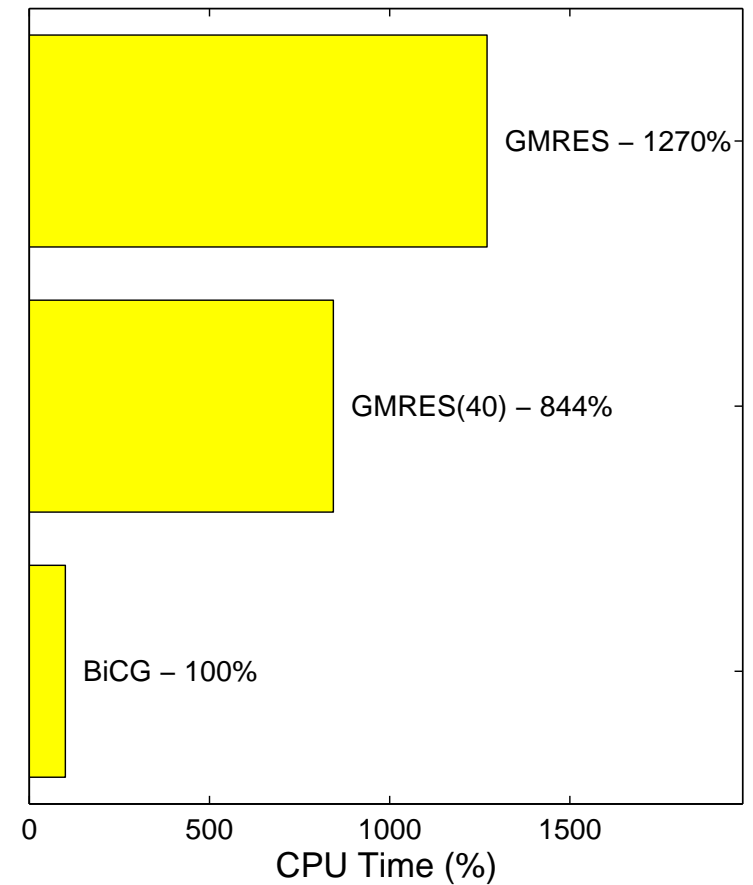
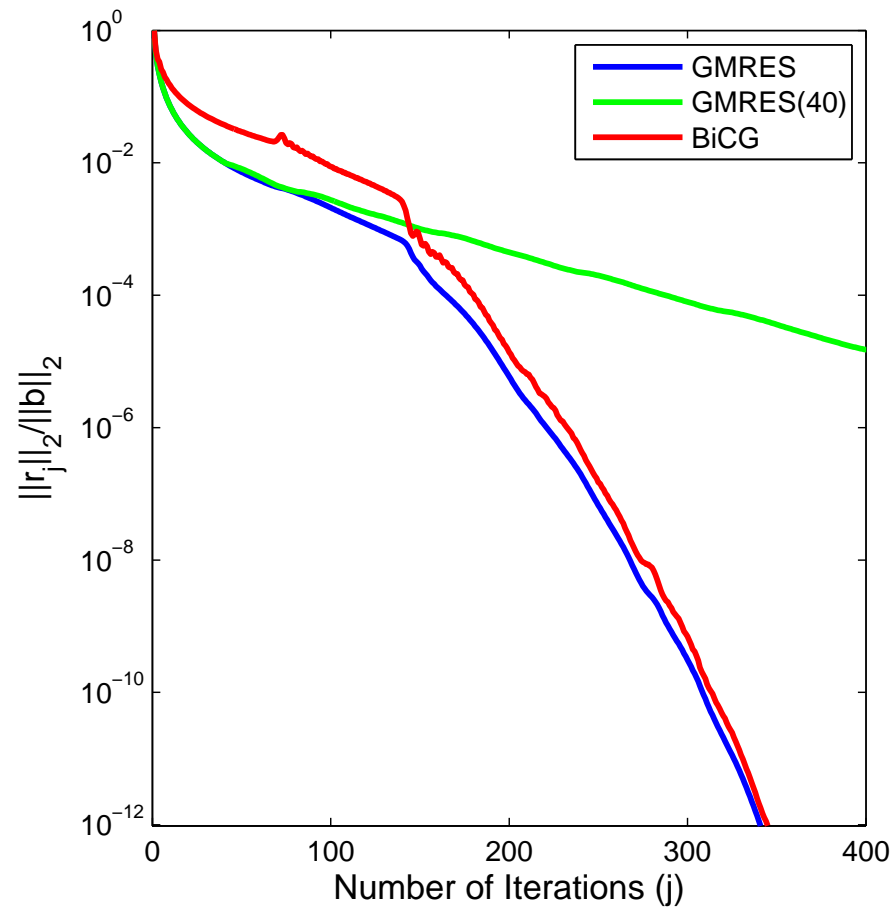
$$\mathbf{p}_{j+1} := \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j$$

$$\mathbf{p}_{j+1}^* := \mathbf{r}_{j+1}^* + \beta_j \mathbf{p}_j^*$$

$$j := j + 1$$

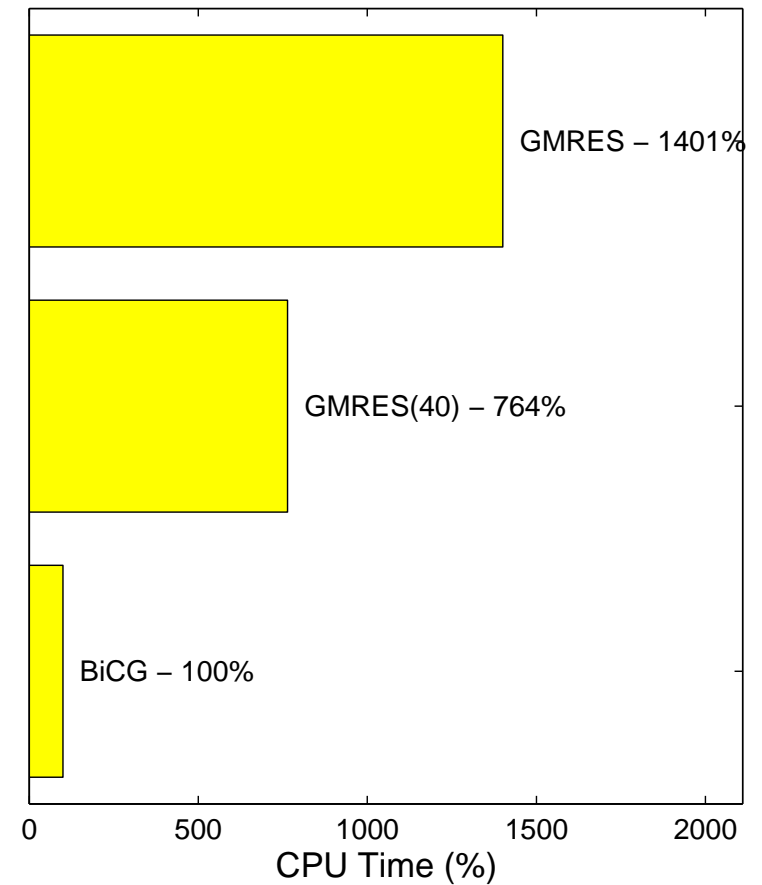
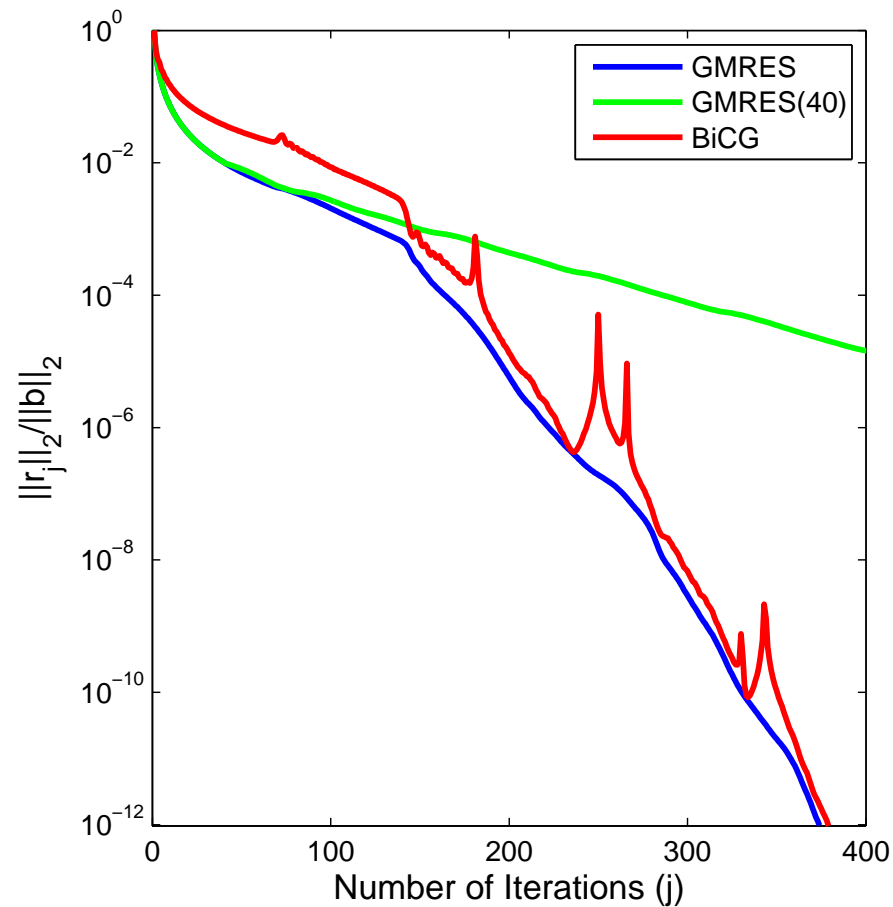
Comparison of GMRES, GMRES(m) and BiCG

Test 1: Pure Diffusion ($\alpha = 0, \epsilon = 1$)



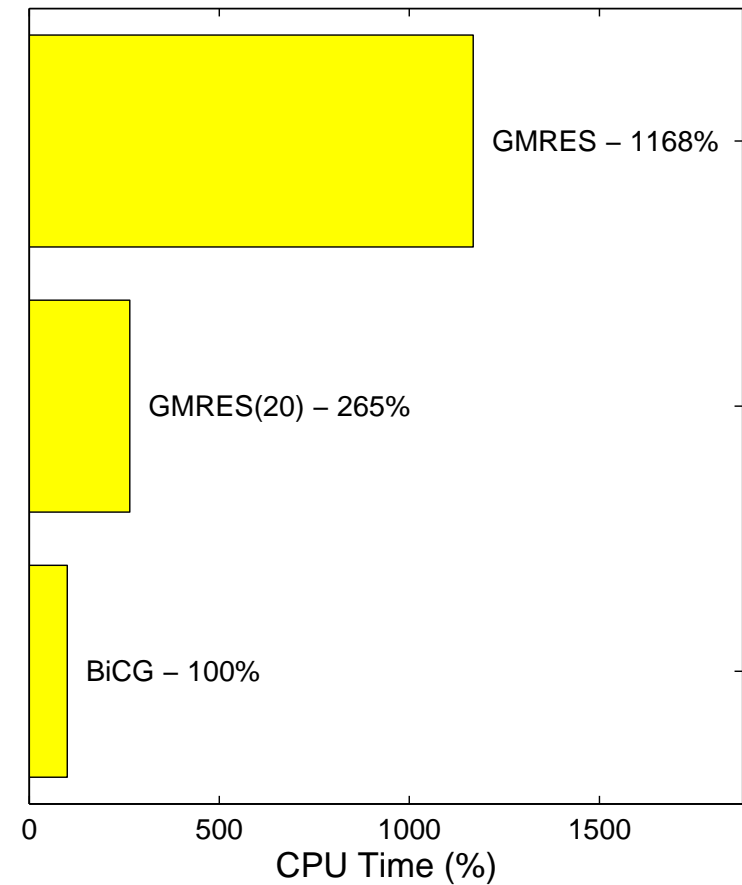
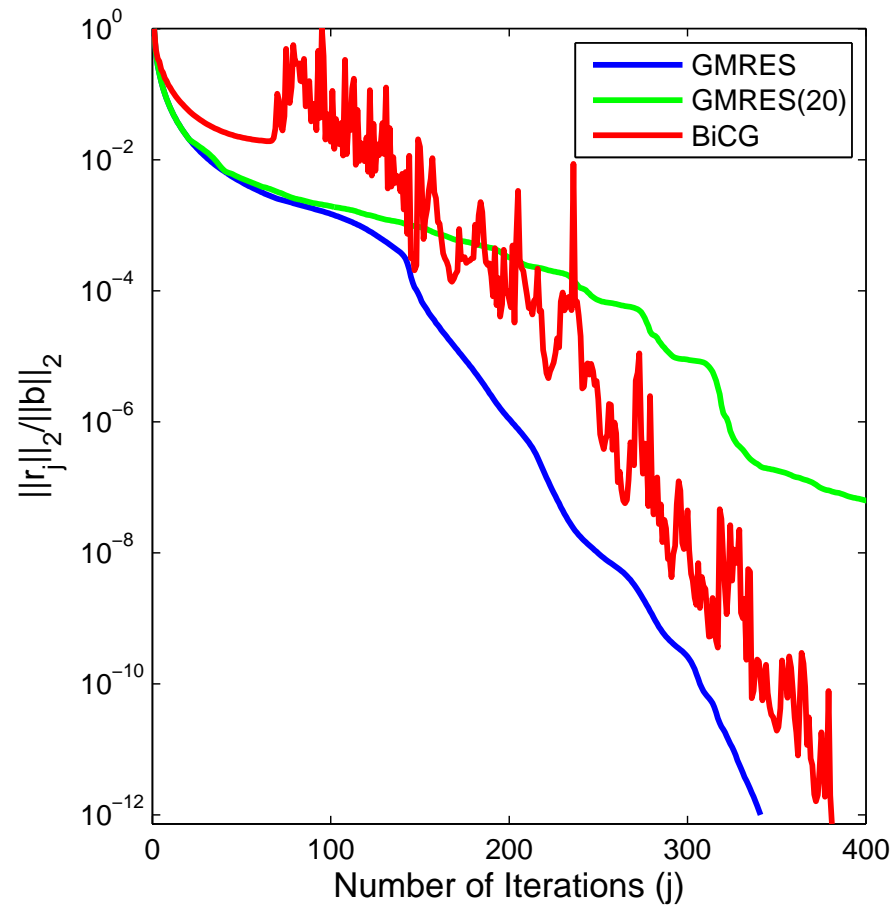
Comparison of GMRES, GMRES(m) and BiCG

Test 2: Weak Convection-Diffusion ($\alpha = 0.1, \epsilon = 1$)



Comparison of GMRES, GMRES(m) and BiCG

Test 3: Convection-Diffusion ($\alpha = 1, \epsilon = 0.1$)



BiCG-Algorithm - Summary

Derivation:

- Based on BiLanczos-Algorithm
- Skew Krylov subspace method $b - Ax_m \perp K_m^T$

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)

Disadvantages:

- Requires multiplications with A^T
- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, p_j^*)

BiCG-Algorithm - Summary

Derivation:

- Based on BiLanczos-Algorithm
- Skew Krylov subspace method $b - Ax_m \perp K_m^T$

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)

Disadvantages:

- Requires multiplications with A^T
- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, p_j^*)

BiCG-Algorithm - Summary

Derivation:

- Based on BiLanczos-Algorithm
- Skew Krylov subspace method $b - Ax_m \perp K_m^T$

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)

Disadvantages:

- Requires multiplications with A^T
- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, p_j^*)

Methods for non-singular Matrices

Method of conjugate gradients (CG)

```
graph TD; A[Method of conjugate gradients (CG)] --> B[Bi-conjugate gradients method (BiCG)]; A --> C[Generalized Minimal Residual method (GMRES)]; D[BiCG-Method] --> E[CG-Squared method (CGS)]; D --> F[Bi-CG Stabilized method (BiCGSTAB)];
```

Bi-conjugate gradients
method (BiCG)

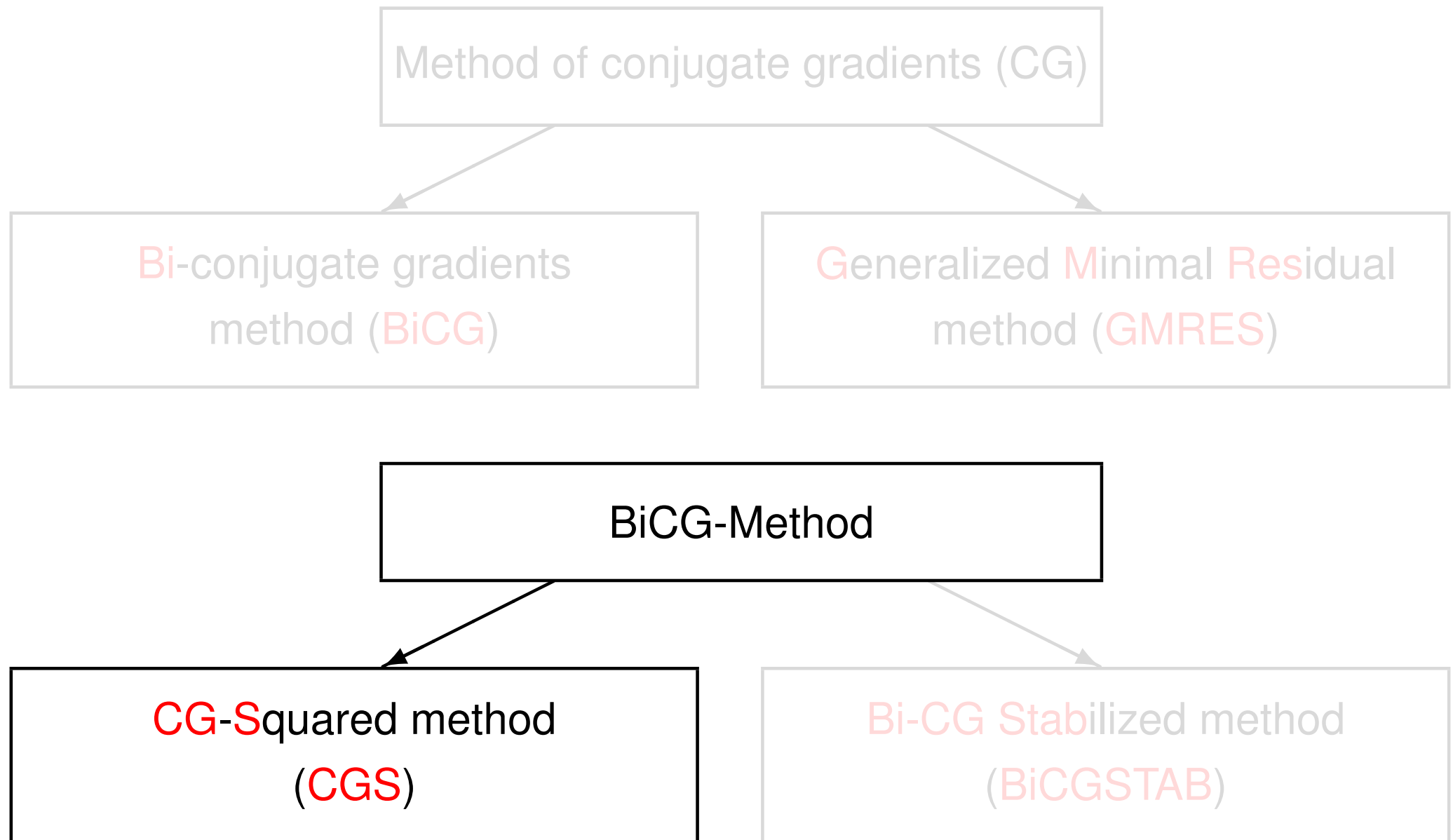
Generalized Minimal Residual
method (GMRES)

BiCG-Method

CG-Squared method
(CGS)

Bi-CG Stabilized method
(BiCGSTAB)

Methods for non-singular Matrices



CGS-Algorithm

Aim:

- Accelerate the BiCG-method
- Avoid multiplications with A^T

Monitoring:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0 \implies r_j^* = \varphi_j(A^T)r_0, \quad p_j^* = \psi_j(A^T)r_0$$

- Occurrence of r_j^* and p_j^*

Solely to calculate the scalar values α_j and β_j

$$(r_j, r_j^*) \text{ and } (Ap_j, p_j^*)$$

Basic Idea:

$$(x, A^T y) = x^T A^T y = (Ax)^T y = (Ax, y) \implies (Ax, A^T y) = (A^2 x, y)$$

CGS-Algorithm

Aim:

- Accelerate the BiCG-method
- Avoid multiplications with A^T

Monitoring:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0 \implies r_j^* = \varphi_j(A^T)r_0, \quad p_j^* = \psi_j(A^T)r_0$$

- Occurrence of r_j^* and p_j^*

Solely to calculate the scalar values α_j and β_j

$$(r_j, r_j^*) \text{ and } (Ap_j, p_j^*)$$

Basic Idea:

$$(x, A^T y) = x^T A^T y = (Ax)^T y = (Ax, y) \implies (Ax, A^T y) = (A^2 x, y)$$

CGS-Algorithm

Aim:

- Accelerate the BiCG-method
- Avoid multiplications with A^T

Monitoring:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0 \implies r_j^* = \varphi_j(A^T)r_0, \quad p_j^* = \psi_j(A^T)r_0$$

- Occurrence of r_j^* and p_j^*

Solely to calculate the scalar values α_j and β_j

$$(r_j, r_j^*) \text{ and } (Ap_j, p_j^*)$$

Basic Idea:

$$(x, A^T y) = x^T A^T y = (Ax)^T y = (Ax, y) \implies (Ax, A^T y) = (A^2 x, y)$$

CGS-Algorithm

Aim:

- Accelerate the BiCG-method
- Avoid multiplications with A^T

Monitoring:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0 \implies r_j^* = \varphi_j(A^T)r_0, \quad p_j^* = \psi_j(A^T)r_0$$

- Occurrence of r_j^* and p_j^*

Solely to calculate the scalar values α_j and β_j

$$(r_j, r_j^*) \text{ and } (Ap_j, p_j^*)$$

Basic Idea:

$$(x, A^T y) = x^T A^T y = (Ax)^T y = (Ax, y) \implies (Ax, A^T y) = (A^2 x, y)$$

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = (\underbrace{\varphi_j^2(\mathbf{A})r_0}_{=: \hat{r}_j}, r_0) = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = (\underbrace{\varphi_j^2(\mathbf{A})r_0}_{=: \hat{r}_j}, r_0) = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = \underbrace{(\varphi_j^2(\mathbf{A})r_0, r_0)}_{=: \hat{r}_j} = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = \underbrace{(\varphi_j^2(\mathbf{A})r_0, r_0)}_{=: \hat{r}_j} = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = \underbrace{(\varphi_j^2(\mathbf{A})r_0, r_0)}_{=: \hat{r}_j} = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = (\underbrace{\varphi_j^2(\mathbf{A})r_0}_{=: \hat{r}_j}, r_0) = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

Reformulation:

Using

$$r_j = \varphi_j(\mathbf{A})r_0, \quad p_j = \psi_j(\mathbf{A})r_0 \quad \text{and} \quad r_j^* = \varphi_j(\mathbf{A}^T)r_0, \quad p_j^* = \psi_j(\mathbf{A}^T)r_0$$

yields

$$(r_j, r_j^*) = (\varphi_j(\mathbf{A})r_0, \varphi_j(\mathbf{A}^T)r_0) = (\underbrace{\varphi_j^2(\mathbf{A})r_0}_{=: \hat{r}_j}, r_0) = (\hat{r}_j, r_0)$$

as well as

$$(A p_j, p_j^*) = (A \psi_j(\mathbf{A})r_0, \psi_j(\mathbf{A}^T)r_0) = (A \underbrace{\psi_j^2(\mathbf{A})r_0}_{=: \hat{p}_j}, r_0) = (A \hat{p}_j, r_0)$$

Technical Exercise:

Express the scalar values α_j and β_j by means of \hat{r}_j and \hat{p}_j

CGS-Algorithm

CGS-Algorithmus —

Wähle $\mathbf{x}_0 \in \mathbb{R}^n$ und $\varepsilon > 0$

$\mathbf{u}_0 = \mathbf{r}_0 = \mathbf{p}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $j := 0$

Solange $\|\mathbf{r}_j\|_2 > \varepsilon$

$$\mathbf{v}_j := \mathbf{A}\mathbf{p}_j, \alpha_j := \frac{(\mathbf{r}_j, \mathbf{r}_0)_2}{(\mathbf{v}_j, \mathbf{r}_0)_2}$$

$$\mathbf{q}_j := \mathbf{u}_j - \alpha_j \mathbf{v}_j$$

$$\mathbf{x}_{j+1} := \mathbf{x}_j + \alpha_j (\mathbf{u}_j + \mathbf{q}_j)$$

$$\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha_j \mathbf{A}(\mathbf{u}_j + \mathbf{q}_j)$$

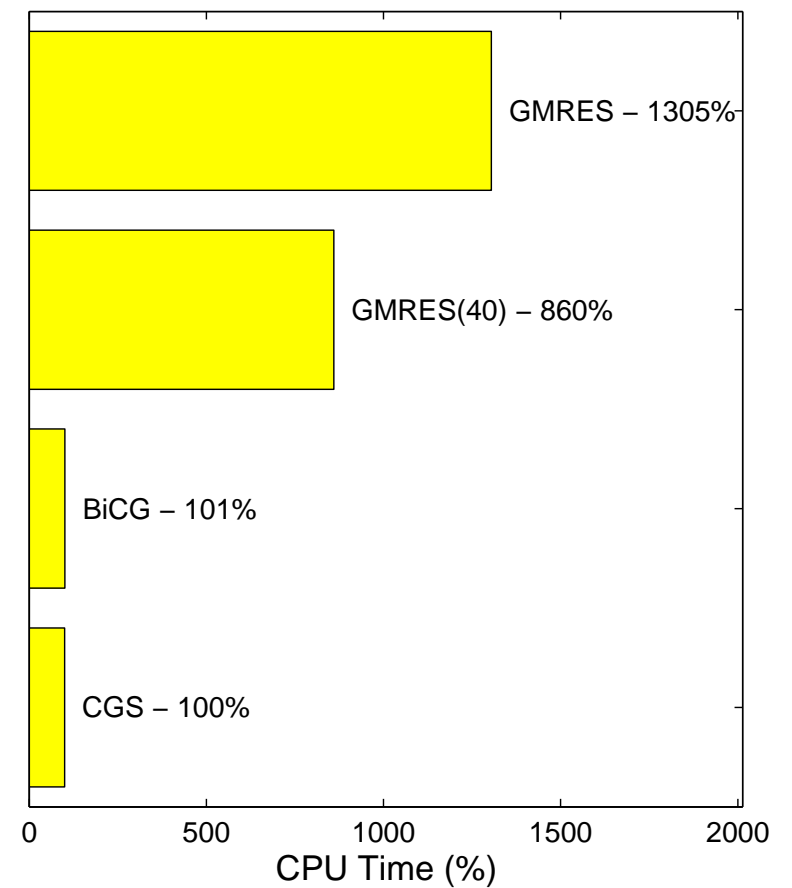
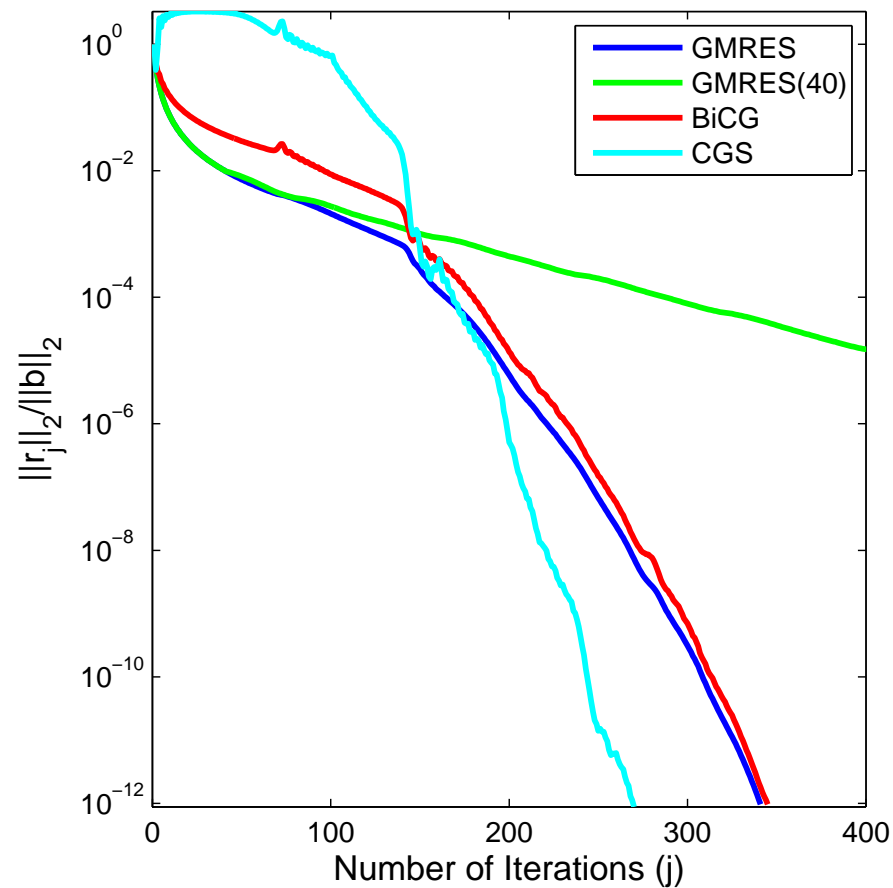
$$\beta_j := \frac{(\mathbf{r}_{j+1}, \mathbf{r}_0)_2}{(\mathbf{r}_j, \mathbf{r}_0)_2}$$

$$\mathbf{u}_{j+1} := \mathbf{r}_{j+1} + \beta_j \mathbf{q}_j$$

$$\mathbf{p}_{j+1} := \mathbf{u}_{j+1} + \beta_j (\mathbf{q}_j + \beta_j \mathbf{p}_j), j := j + 1$$

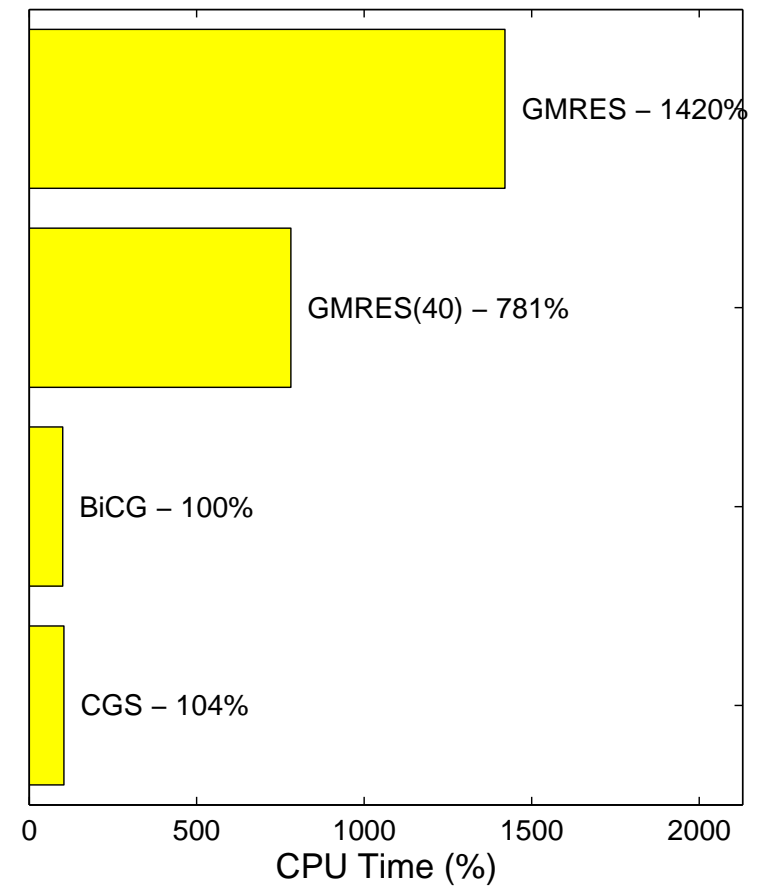
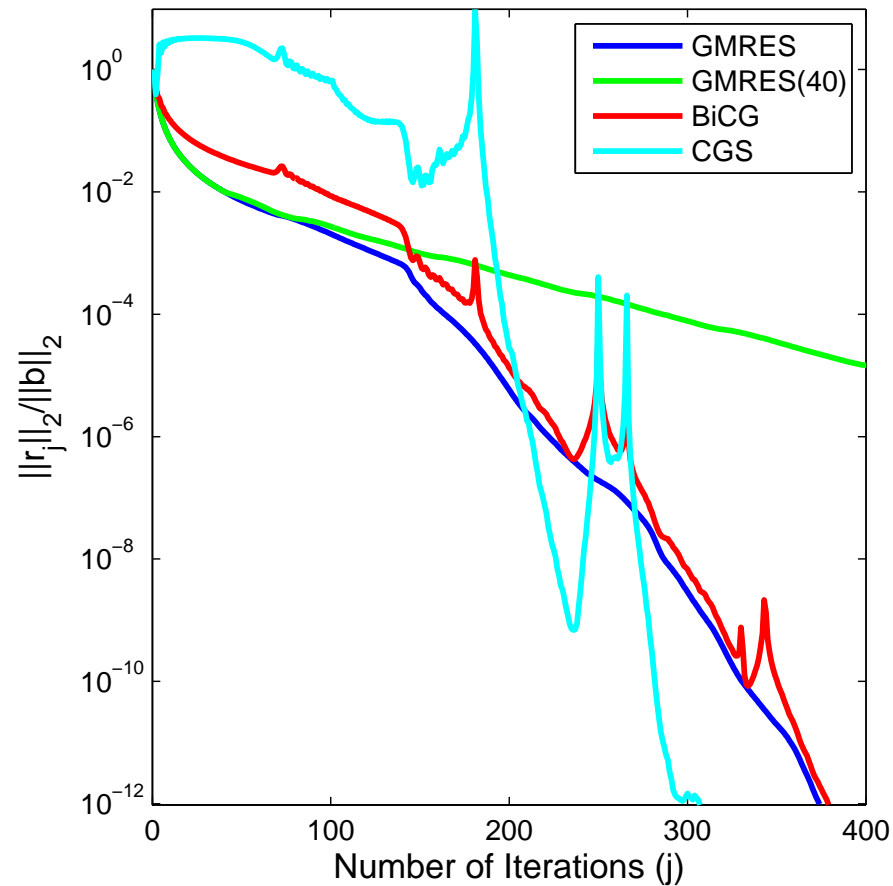
Comparison of GMRES, GMRES(m), BiCG and CGS

Test 1: Pure Diffusion ($\alpha = 0, \epsilon = 1$)



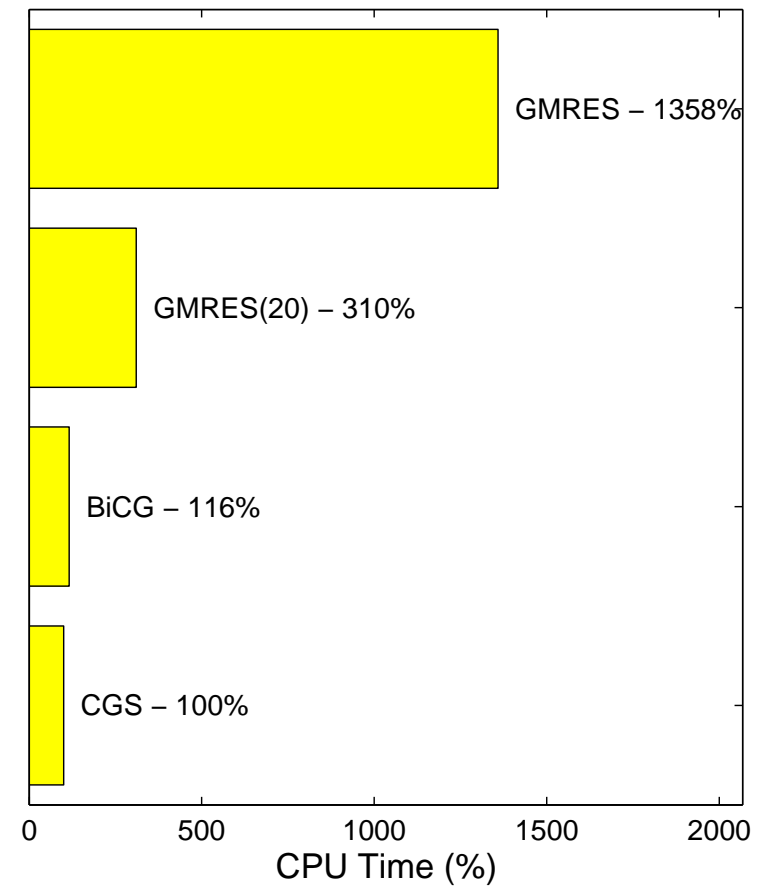
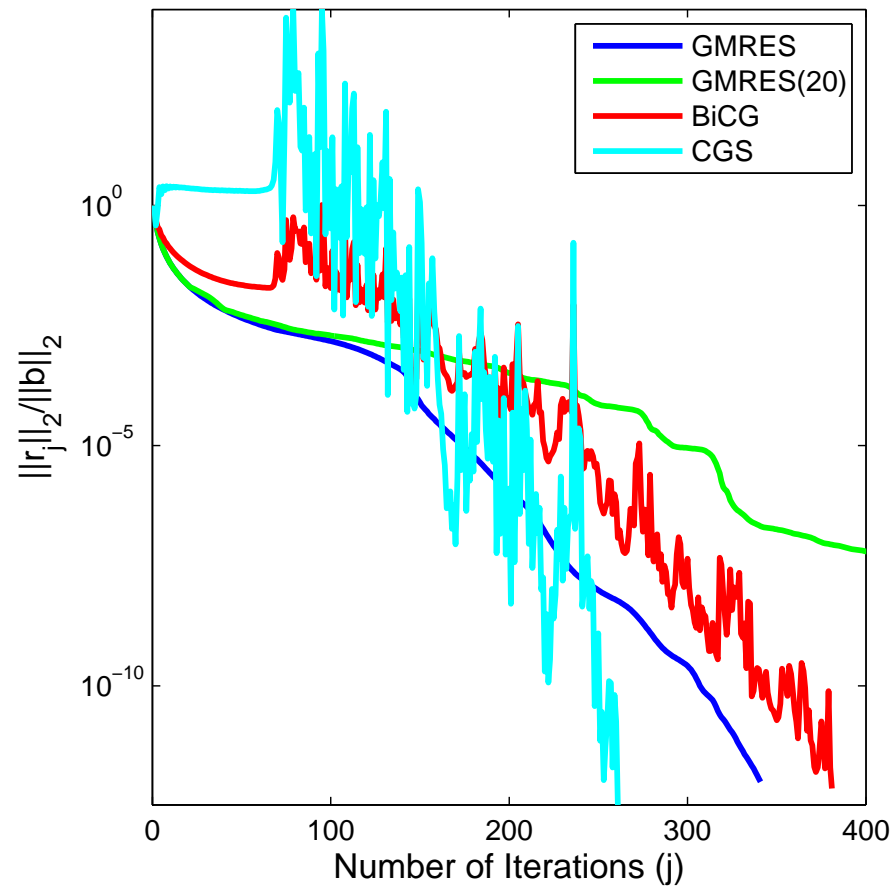
Comparison of GMRES, GMRES(m), BiCG and CGS

Test 2: Weak Convection-Diffusion ($\alpha = 0.1, \epsilon = 1$)



Comparison of GMRES, GMRES(m), BiCG and CGS

Test 3: Convection-Diffusion ($\alpha = 1, \epsilon = 0.1$)



CGS-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)

Disadvantages:

- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, r_0)

CGS-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)

Disadvantages:

- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, r_0)

CGS-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation

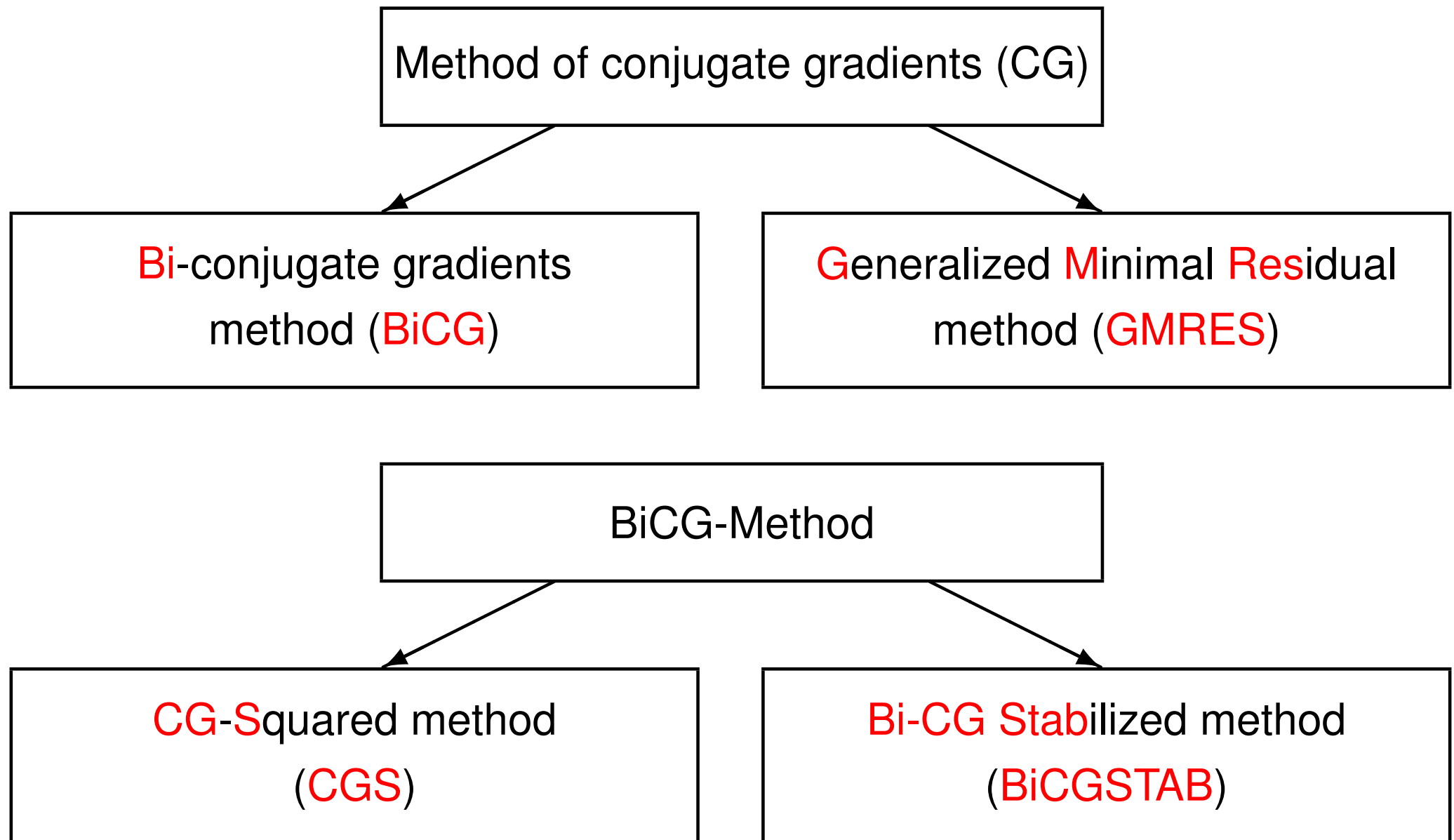
Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)

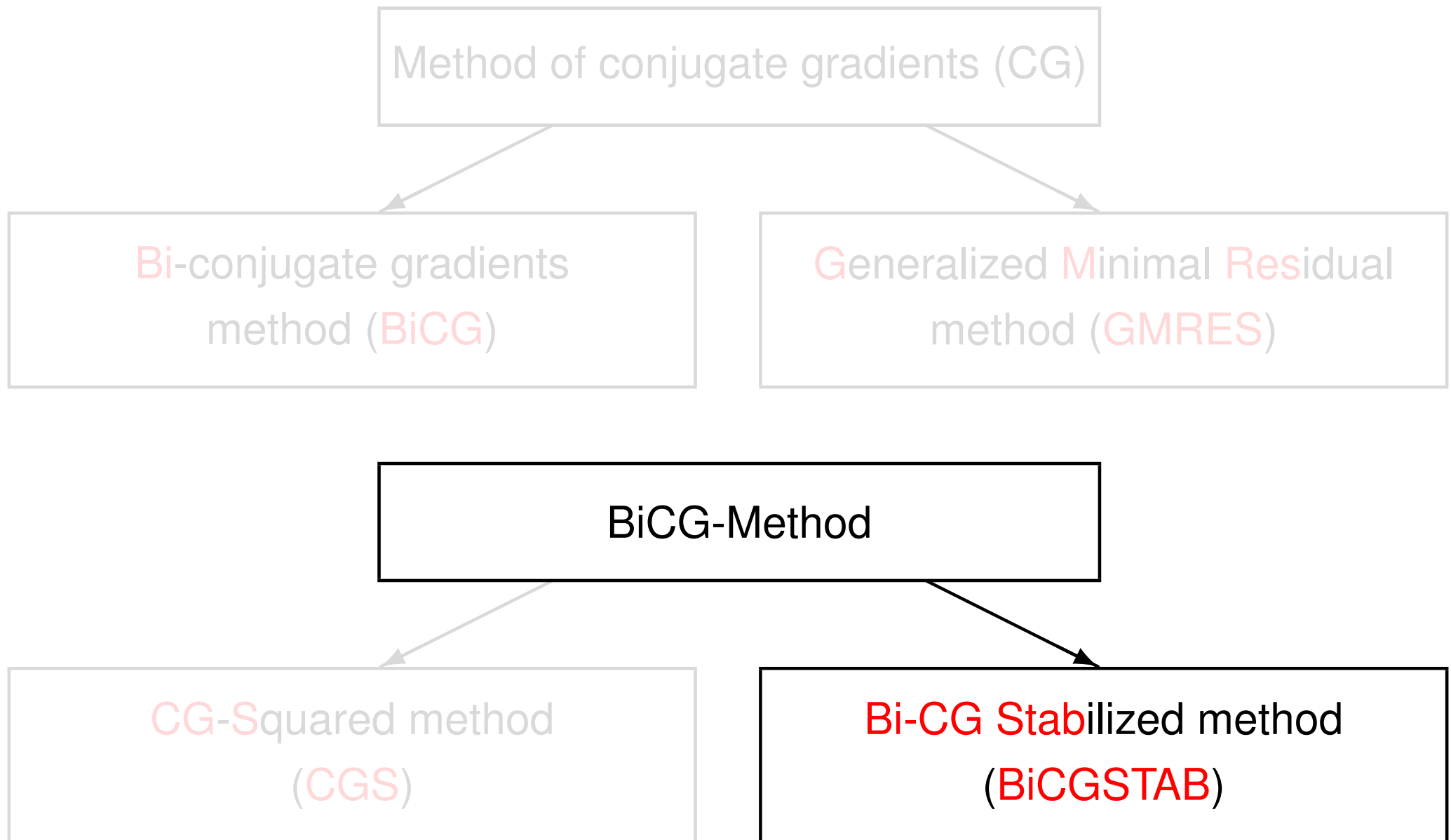
Disadvantages:

- No minimization of an underlying functional
→ Oszillations in the convergence history
- Possible break down due to division by (Ap_j, r_0)

Methods for non-singular Matrices



Methods for non-singular Matrices



BiCGSTAB-Algorithm

Aim:

- Improving the BiCG- and CGS-method
- Avoid multiplications with A^T
- Introducing a minimization of the residual

Procedure:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0$$

- Employ

$$\tilde{r}_j^* = \Phi_j(A^T)r_0, \quad \tilde{p}_j^* = \Phi_j(A^T)r_0$$

with

$$\Phi_0(A^T) = I, \quad \Phi_{j+1}(A^T) = (I - \omega_j A^T)\Phi_j(A^T)$$

BiCGSTAB-Algorithm

Aim:

- Improving the BiCG- and CGS-method
- Avoid multiplications with A^T
- Introducing a minimization of the residual

Procedure:

- Polynomial representation

$$r_j = \varphi_j(A)r_0, \quad p_j = \psi_j(A)r_0$$

- Employ

$$\tilde{r}_j^* = \Phi_j(A^T)r_0, \quad \tilde{p}_j^* = \Phi_j(A^T)r_0$$

with

$$\Phi_0(A^T) = I, \quad \Phi_{j+1}(A^T) = (I - \omega_j A^T)\Phi_j(A^T)$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \Phi_j(A^T)r_0$ and $\tilde{p}_j^* = \Phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \Phi_j(A^T)r_0) = (\Phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \Phi_j(A^T)r_0) = (\Phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$

- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

Reformulation:

Introducing $\tilde{r}_j^* = \phi_j(A^T)r_0$ and $\tilde{p}_j^* = \phi_j(A^T)r_0$ into BiCG yields

$$(r_j, \tilde{r}_j^*) = (\varphi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)\varphi_j(A)r_0, r_0)$$

and

$$(Ap_j, \tilde{p}_j^*) = (A\psi_j(A)r_0, \phi_j(A^T)r_0) = (\phi_j(A)A\psi_j(A)r_0, r_0)$$

Minimization of the residual:

- $r_{j+1} = (I - \omega_j A)s_j$
- Define $f_j(\omega) = \|(I - \omega A)s_j\|_2^2$

$$\Rightarrow f_j'(\omega) = -2(As_j, s_j) + 2\omega(As_j, As_j)$$

$$f_j''(\omega) = 2(As_j, As_j) \geq 0$$

$$\Rightarrow \omega_j = \arg \min_{\omega \in \mathbb{R}} f_j(\omega) = \frac{(As_j, s_j)}{(As_j, As_j)}$$

BiCGSTAB-Algorithm

BiCGSTAB-Algorithmus —

Wähle $\mathbf{x}_0 \in \mathbb{R}^n$ und $\varepsilon > 0$

$\mathbf{r}_0 := \mathbf{p}_0 := \mathbf{b} - \mathbf{A} \mathbf{x}_0$, $\rho_0 := (\mathbf{r}_0, \mathbf{r}_0)_2$, $j := 0$

Solange $\|\mathbf{r}_j\|_2 > \varepsilon$

$\mathbf{v}_j := \mathbf{A} \mathbf{p}_j$, $\alpha_j := \frac{\rho_j}{(\mathbf{v}_j, \mathbf{r}_0)_2}$

$\mathbf{s}_j := \mathbf{r}_j - \alpha_j \mathbf{v}_j$, $\mathbf{t}_j := \mathbf{A} \mathbf{s}_j$

$\omega_j := \frac{(\mathbf{t}_j, \mathbf{s}_j)_2}{(\mathbf{t}_j, \mathbf{t}_j)_2}$

$\mathbf{x}_{j+1} := \mathbf{x}_j + \alpha_j \mathbf{p}_j + \omega_j \mathbf{s}_j$

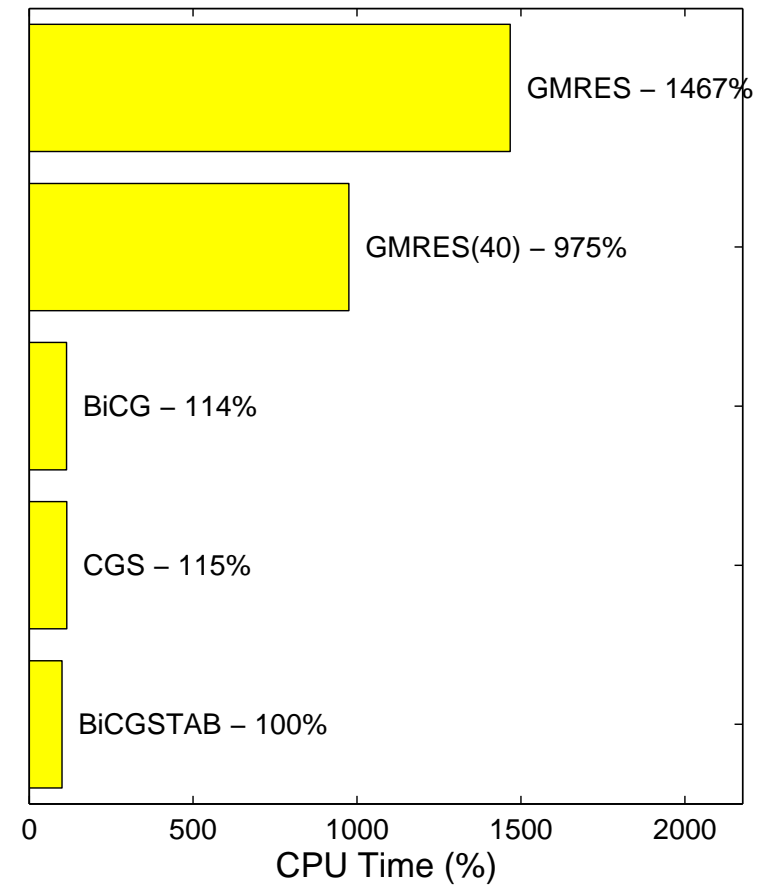
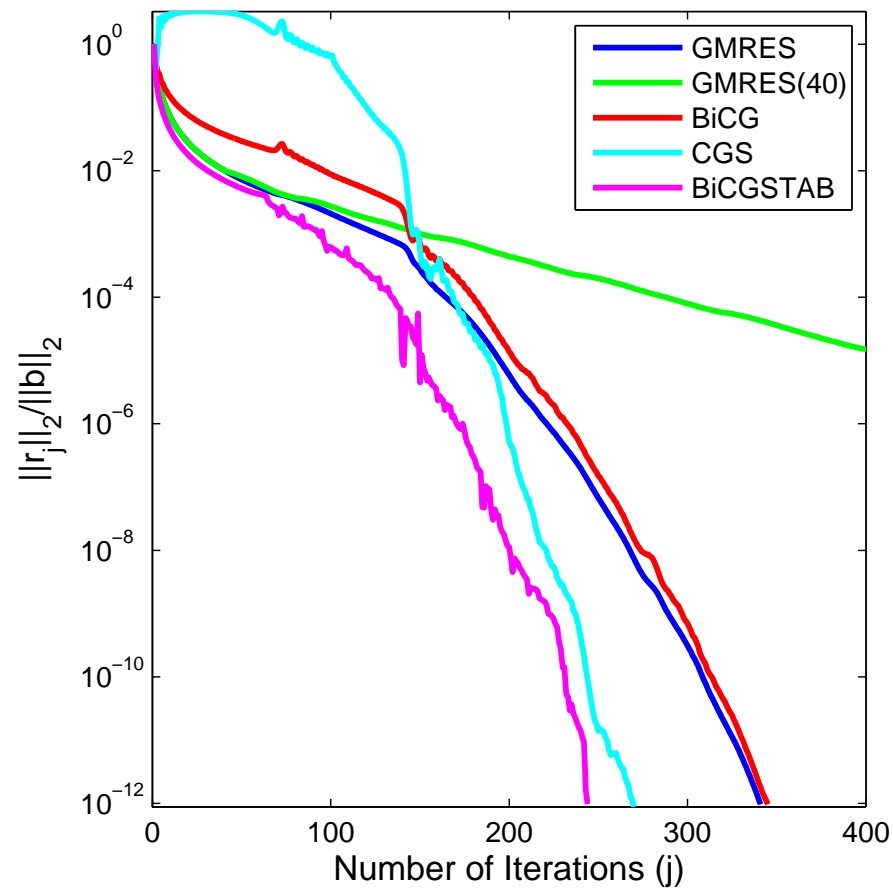
$\mathbf{r}_{j+1} := \mathbf{s}_j - \omega_j \mathbf{t}_j$

$\rho_{j+1} := (\mathbf{r}_{j+1}, \mathbf{r}_0)_2$, $\beta_j := \frac{\alpha_j \rho_{j+1}}{\omega_j \rho_j}$

$\mathbf{p}_{j+1} := \mathbf{r}_{j+1} + \beta_j (\mathbf{p}_j - \omega_j \mathbf{v}_j)$, $j := j + 1$

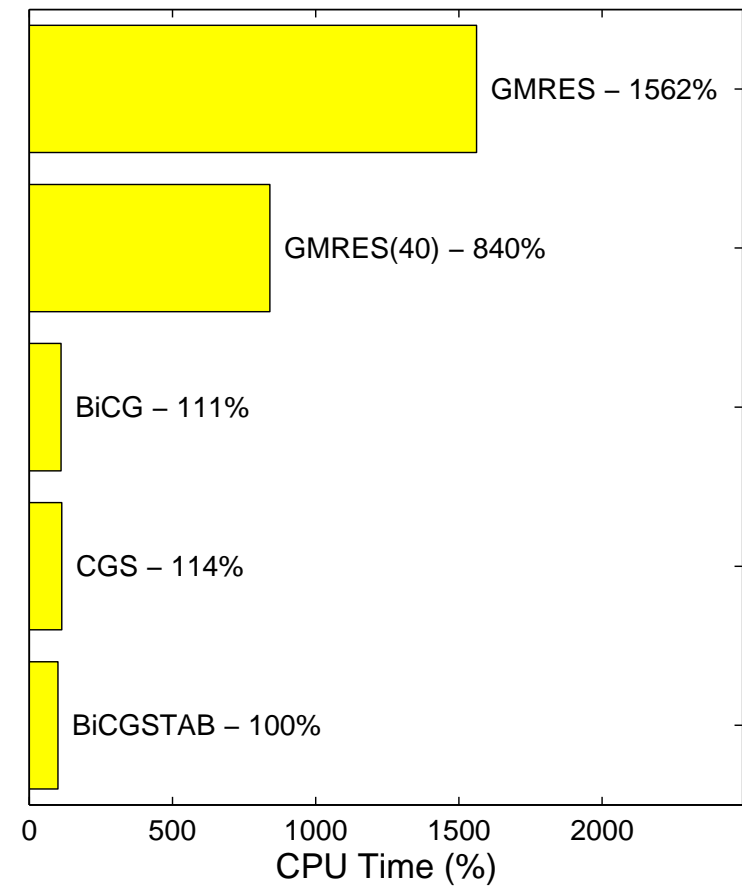
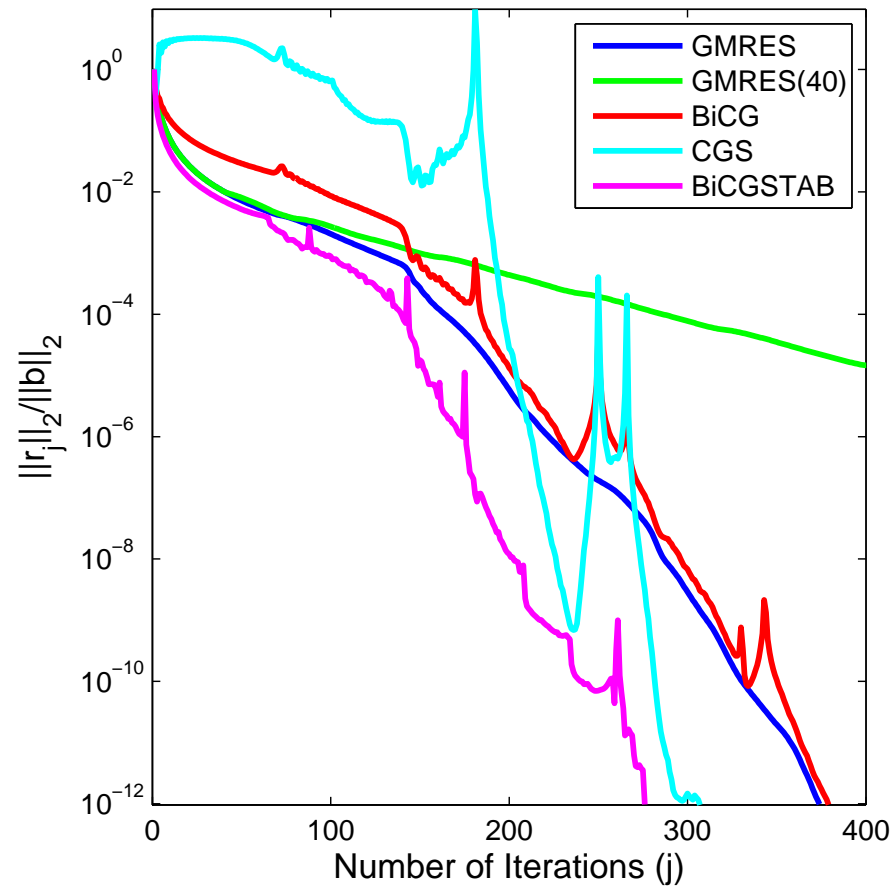
Comparison of GMRES, BiCG, CGS and BiCGSTAB

Test 1: Pure Diffusion ($\alpha = 0, \epsilon = 1$)



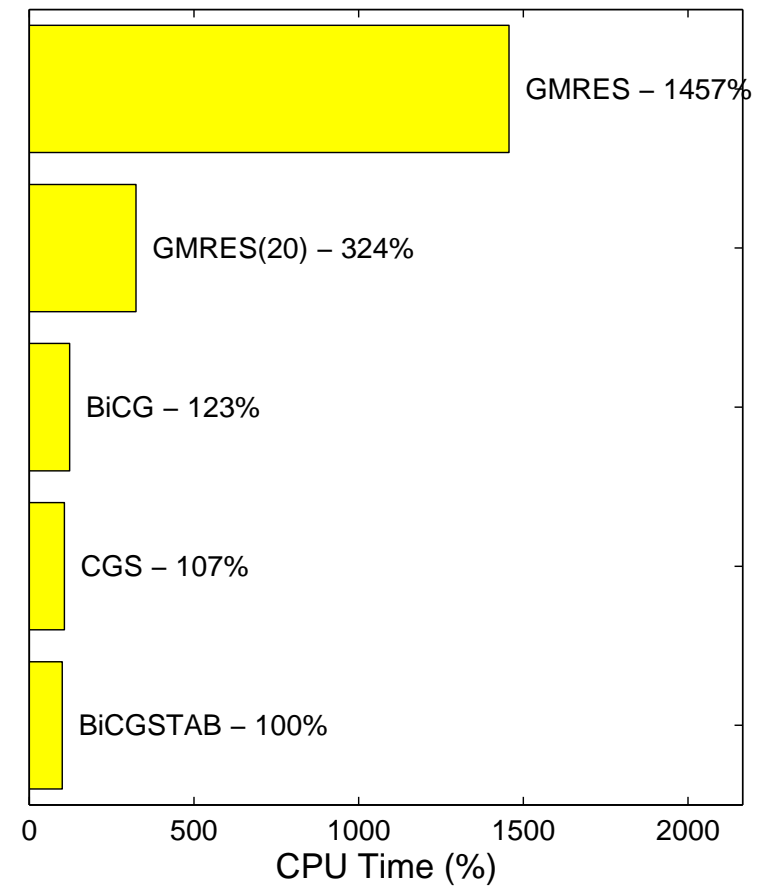
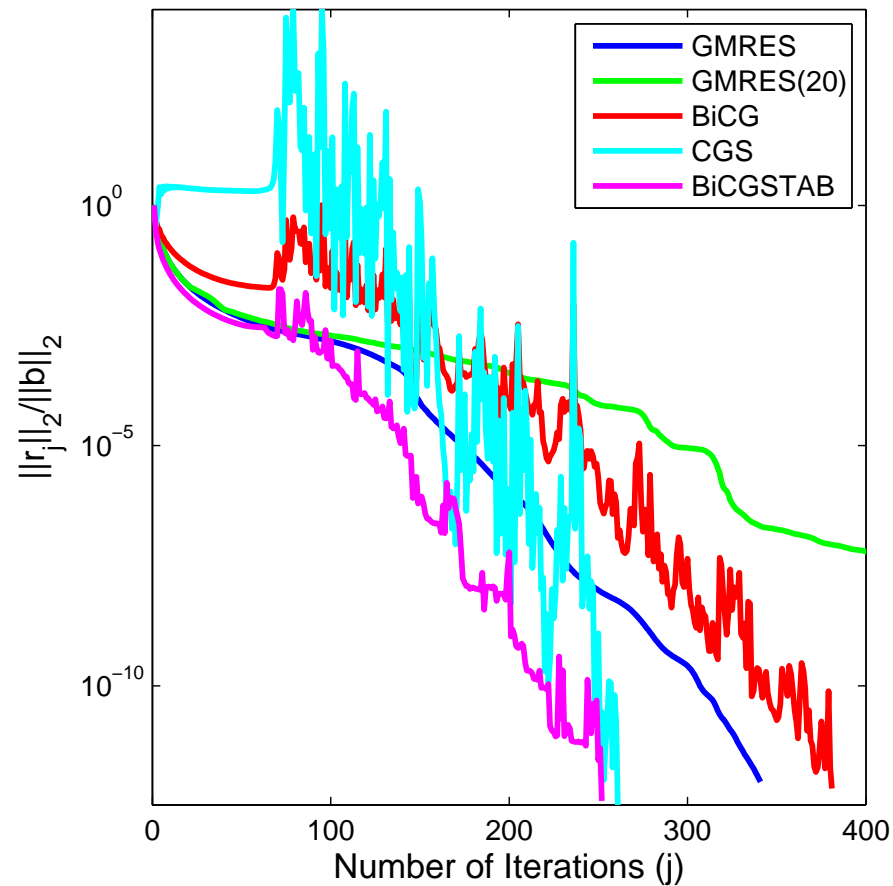
Comparison of GMRES, BiCG, CGS and BiCGSTAB

Test 2: Weak Convection-Diffusion ($\alpha = 0.1, \epsilon = 1$)



Comparison of GMRES, BiCG, CGS and BiCGSTAB

Test 3: Convection-Diffusion ($\alpha = 1, \epsilon = 0.1$)



BiCGSTAB-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation
- Residual minimization

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)
- Additional minimization technique (compared to CGS)
→ Smooth convergence history

Disadvantages:

- Possible break down due to division by $(A p_j, r_0)$

BiCGSTAB-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation
- Residual minimization

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)
- Additional minimization technique (compared to CGS)
→ Smooth convergence history

Disadvantages:

- Possible break down due to division by $(A p_j, r_0)$

BiCGSTAB-Algorithm - Summary

Derivation:

- Based on BiCG-Algorithm
- Squaring the polynomial representation
- Residual minimization

Advantages:

- Keenly less storage requirements (compared to GMRES)
- No symmetry constraint on A (compared to CG)
- Requires no multiplications with A^T (compared to BiCG)
- Additional minimization technique (compared to CGS)
→ Smooth convergence history

Disadvantages:

- Possible break down due to division by (Ap_j, r_0)