Iterative Solvers for Large Linear Systems

Part Va: GMRES and BiCG

Andreas Meister

University of Kassel, Department of Analysis and Applied Mathematics

Outline

- Basics of Iterative Methods
- Splitting-schemes
  - Jacobi- u. Gauß-Seidel-scheme
  - Relaxation methods
- Methods for symmetric, positive definite Matrices
  - Method of steepest descent
  - Method of conjugate directions
  - CG-scheme

Outline

- Multigrid Method
  - Smoother, Prolongation, Restriction
  - Twogrid Method and Extension
- Methods for non-singular Matrices
  - GMRES
  - BiCG, CGS and BiCGSTAB
- Preconditioning
  - ILU, IC, GS, SGS, ...

Projection method & Krylov subspace approach

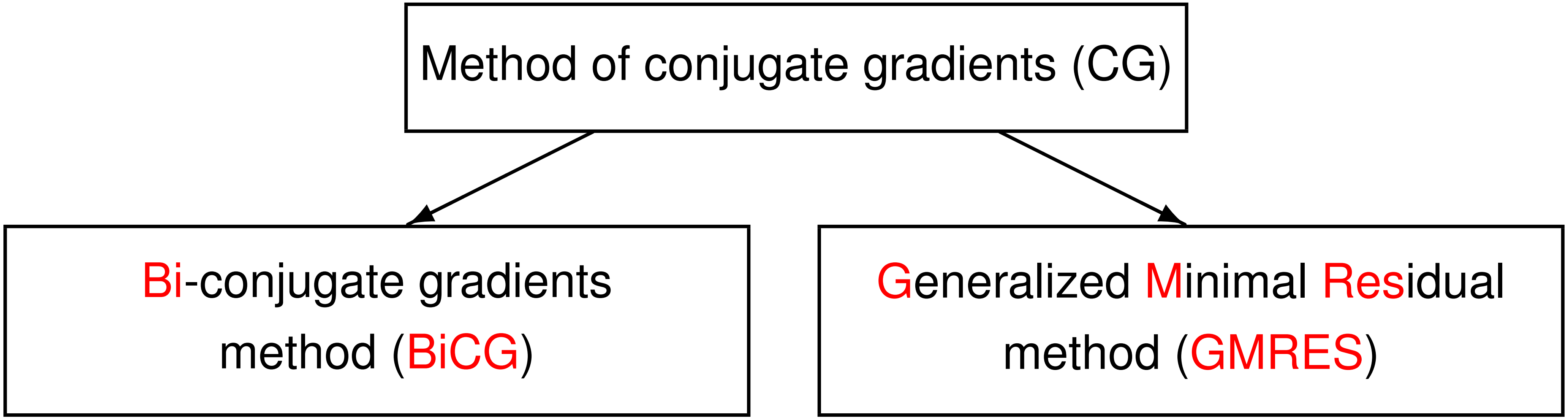
We consider

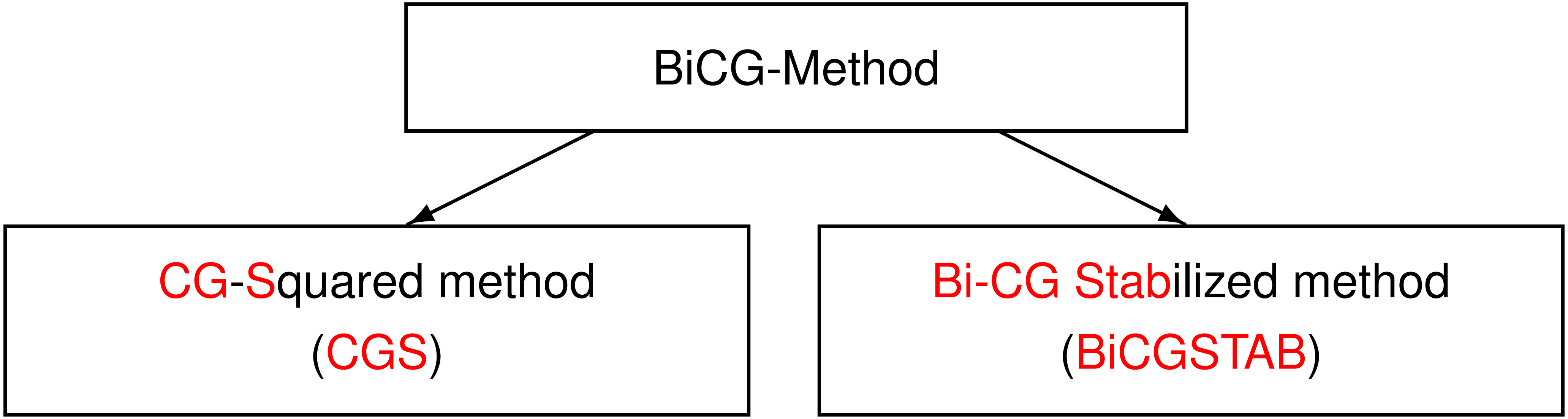
$$Ax = b$$

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

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





Generalized Minimal Residual (GMRES)

Basic idea:

- Search for $x_m = \arg \min\limits_{x \in x_0 + K_m} F(x)$

  $\longrightarrow K_m = \text{span}\{r_0, Ar_0, \ldots, A^{m-1} r_0\}$
- Instead of

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

  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

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

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

  $\Longrightarrow$ Skew Krylov subspace method

Generalized Minimal Residual (GMRES)

Procedure:

- Calculate an ONB $v_1, \ldots, v_m$ 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 $V_m = (v_1 \ldots v_m) \in \mathbb{R}^{n \times m}, \ \alpha^m = (\alpha_1, \ldots, \alpha_m)^T \in \mathbb{R}^m$

Consequence:

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

$$F(x_m) \leq F(x) := \|b - Ax\|_2^2 \quad \forall x \in x_0 + K_m$$

$\Longleftrightarrow$

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

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

Orthonomal basis (ONB) — Arnoldi-Algorithm

Sought after:

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

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

Aim: Calculation of v_{j+1}

Idea:

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

and

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

Conclusion:

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

Orthonomal basis (ONB) — Arnoldi-Algorithm

Ansatz:

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

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 $s = 1, \dots, j$:

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

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

Concluding:

v_{j+1} has to be normalized

Arnoldi-Algorithm

$$v_1 := \frac{r_0}{\|r_0\|_2}$$

Für $j = 1, \ldots, m$

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

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

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

Orthonomal basis (ONB) — Arnoldi-Algorithm

Helpful properties:

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

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

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

(4) B consists of orthonormal columns

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

Generalized Minimal Residual (GMRES)

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

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

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

$$\begin{aligned}
J(\alpha) &= \|b - A(x_0 + V_m\alpha)\|_2 \overset{r_0 = b - Ax_0}{=} \|r_0 - AV_m\alpha\|_2 \\
&\overset{v_1 = \frac{r_0}{\|r_0\|_2}}{=} \|\|r_0\|_2 v_1 - AV_m\alpha\|_2 \overset{(2)}{=} \|V_{m+1}(\|r_0\|_2 e_1 - \overline{H}_m\alpha)\|_2 \\
&\overset{(4)}{=} \|Q_m(\|r_0\|_2 e_1 - \overline{H}_m\alpha)\|_2 = \|g - Q_m\overline{H}_m\alpha\|_2 \\
&\overset{(3)}{=} \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \ldots 0 \end{pmatrix} \alpha \right\|_2 \ge \left\| \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{m+1} \end{pmatrix} \right\|_2 = |g_{m+1}|
\end{aligned}$$

Generalized Minimal Residual (GMRES)

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

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

- $J(\alpha) = \left\| \begin{pmatrix} g_1 \\ \vdots \\ g_{m+1} \end{pmatrix} - \begin{pmatrix} R_m \\ 0 \ldots 0 \end{pmatrix} \alpha \right\|_2 \geq |g_{m+1}|$ 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 $j = 1, \ldots, nm$ $\quad (m << n)$

Extend ONB V_j (Arnoldi)

Extend $\overline{H}_{j-1}$ zu $\overline{H}_j$ (Arnoldi)

Calculate $\overline{R}_j = Q_j \overline{H}_j$ (Givens)

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

If $|g_{j+1}| \leq \varepsilon$ (given tolerance)

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

$x = x_0 + V_j \alpha^j$

STOP

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

$x = x_0 + V_m \alpha^m$

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

Increase Restart by 1

Convection-Diffusion Equation

Governing Equation

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

with

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

Boundary Conditions

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

Mesh

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

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

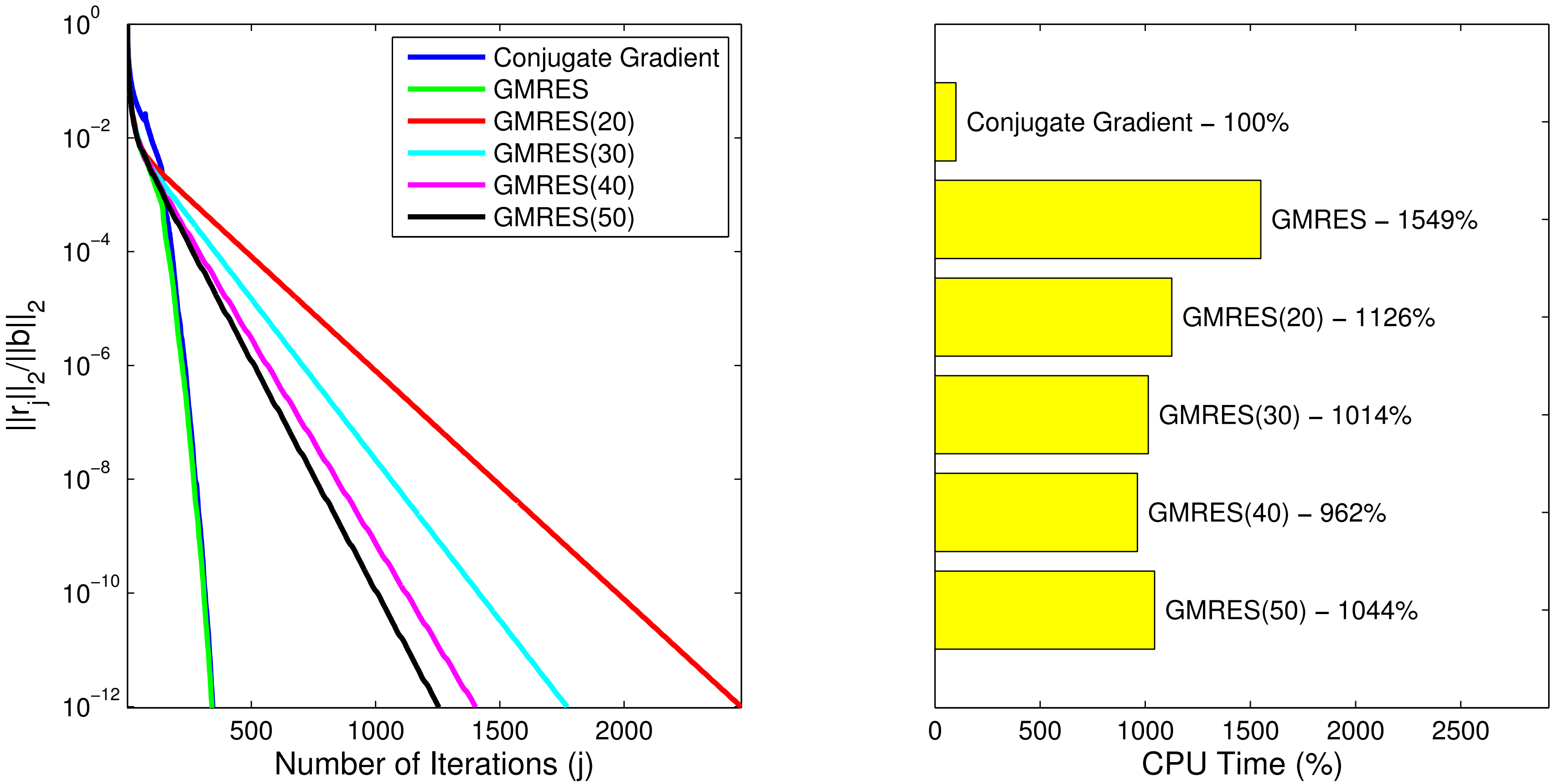
Testcases

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

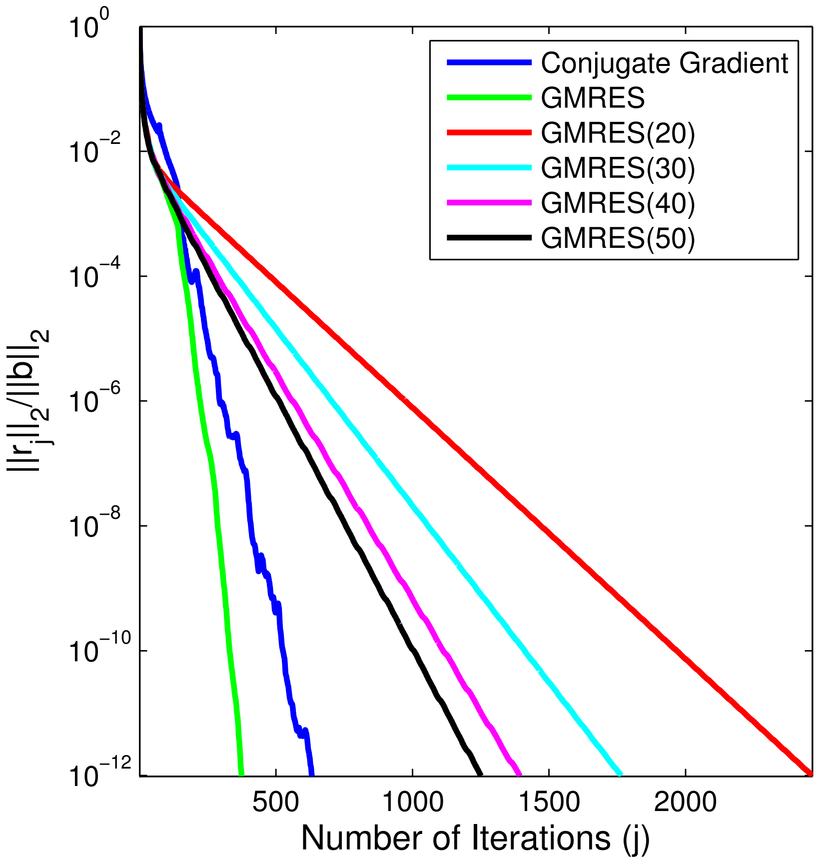
Comparison of CG, GMRES and GMRES(m)

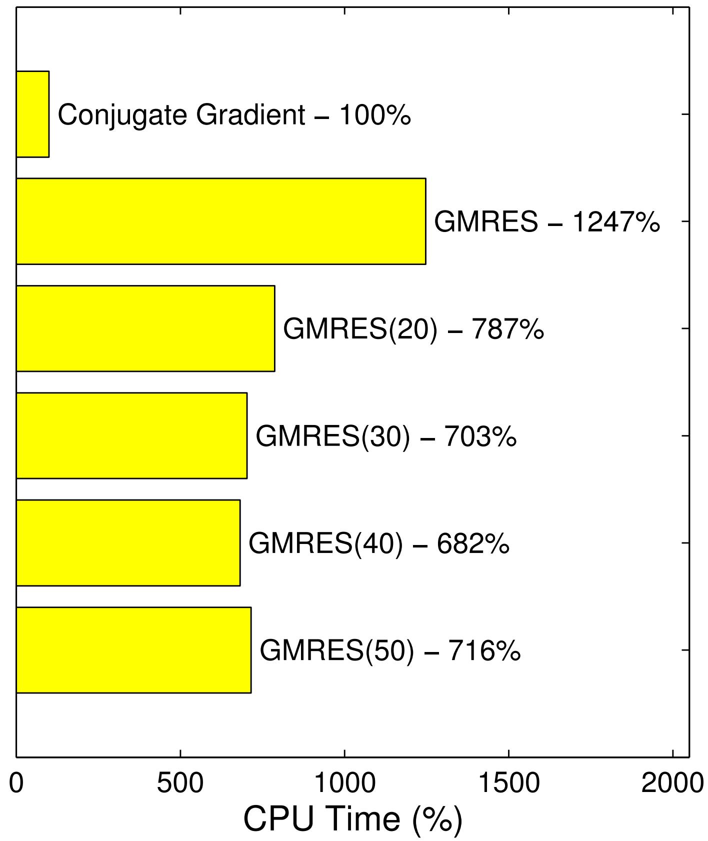
Test 1: Pure Diffusion ($\alpha = 0$, $\epsilon = 1$)



Comparison of CG, GMRES and GMRES(m)

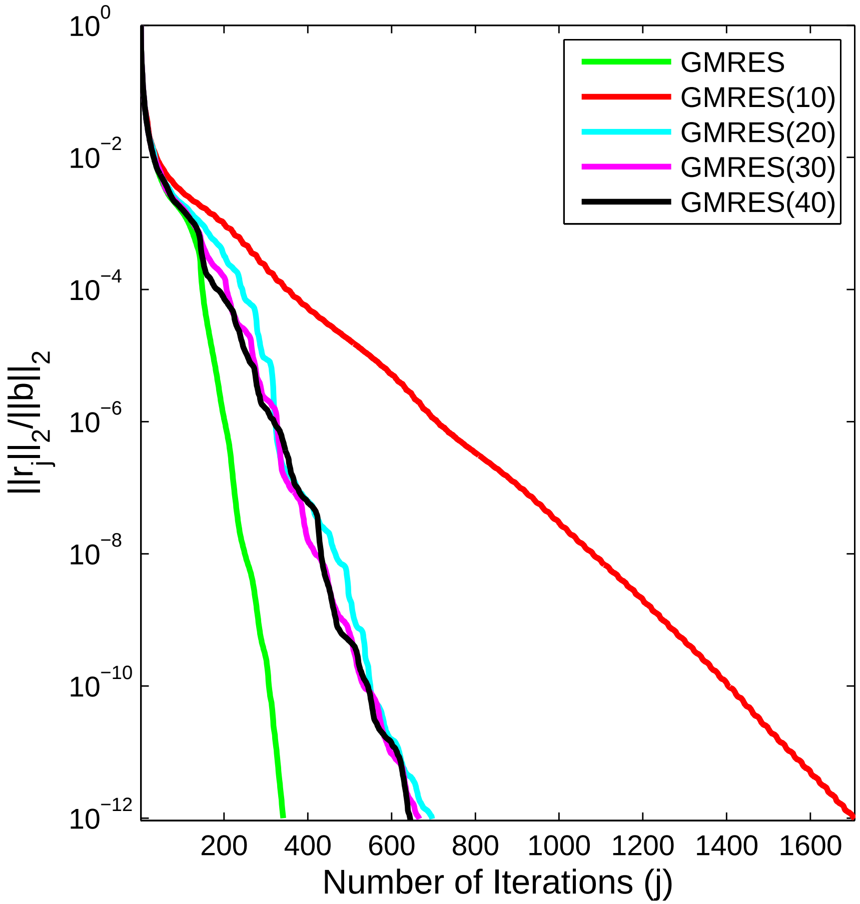
Test 2: Weak Convection-Diffusion ($\alpha = 0.1$, $\epsilon = 1$)

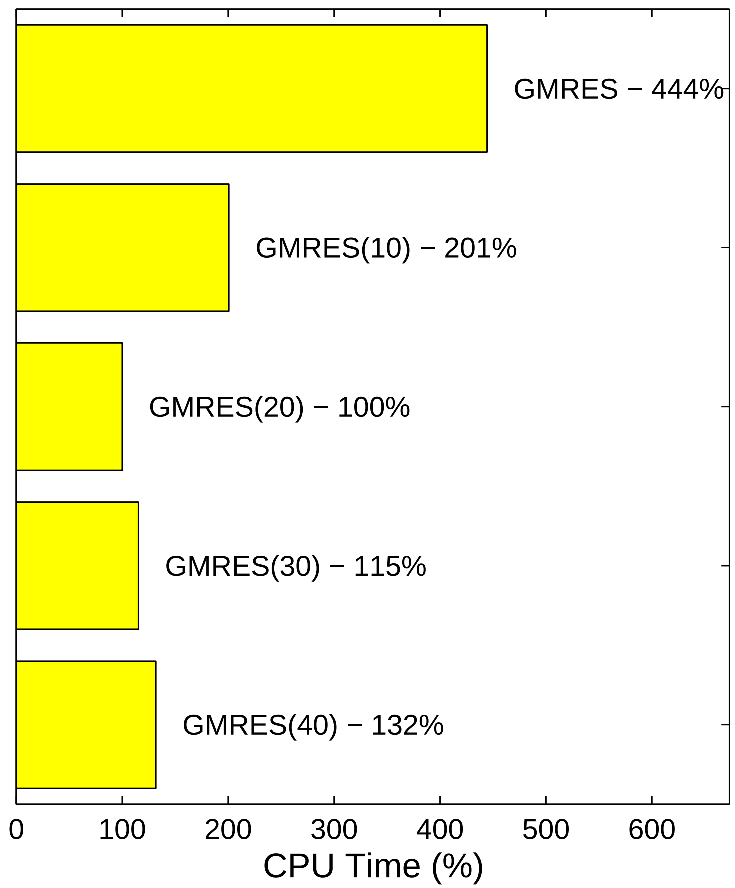




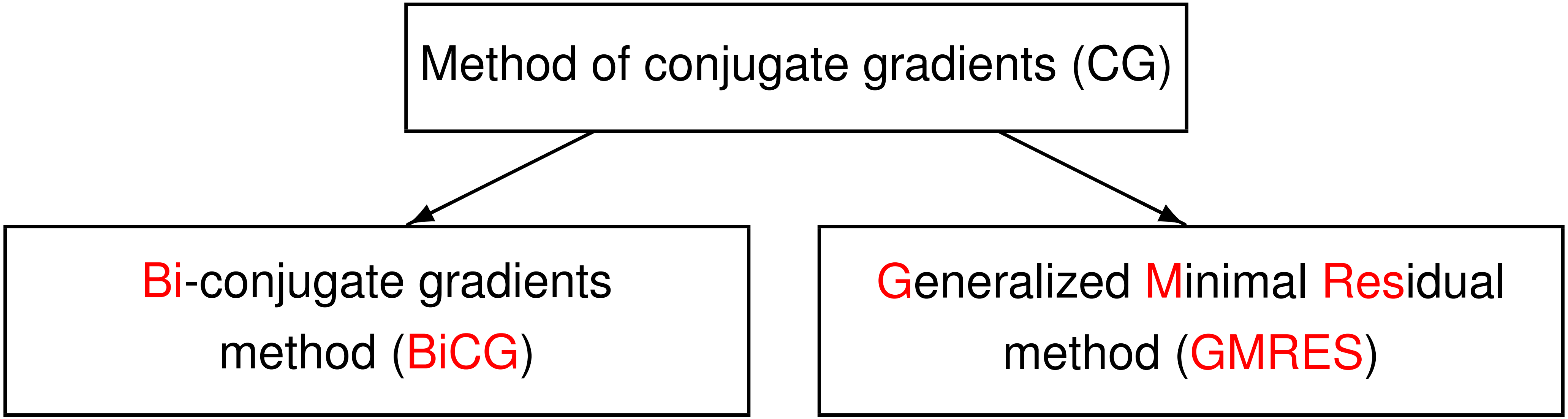
Comparison of CG, GMRES and GMRES(m)

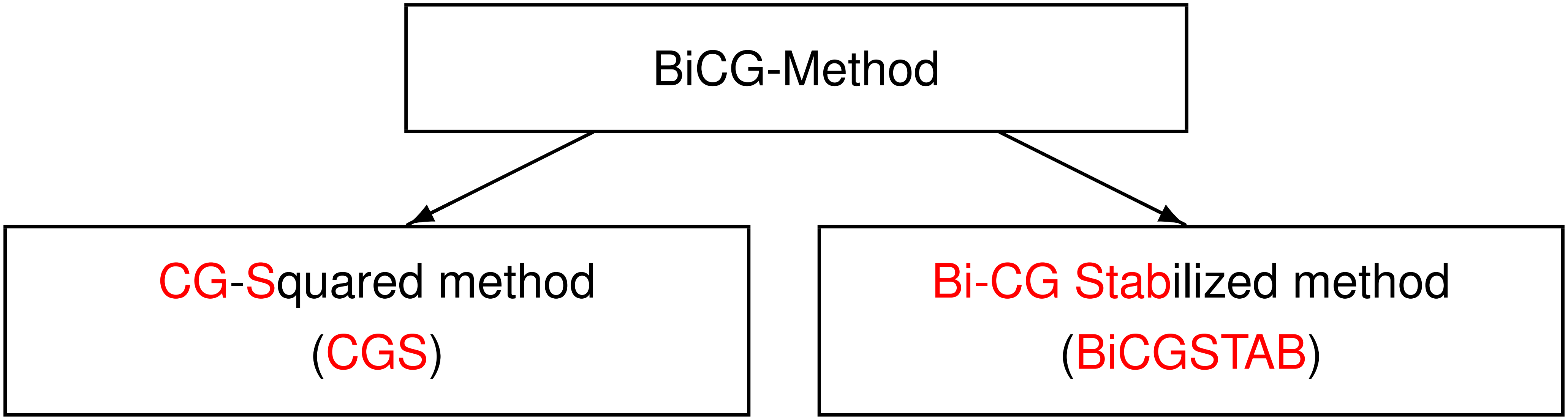
Test 3: Convection-Diffusion ($\alpha = 1$, $\epsilon = 0.1$)





Methods for non-singular Matrices





Bi-conjugate gradient method (BiCG)

Method of conjugate gradients

Advantage : Short recurrence relations

Disadvant. : Matrix A has to be symmetric, pos. definite

GMRES-Method

Advantage : Valid for non-singular matrices

Disadvant. : High comp. effort and storage requirements

BiCG-Method

Basis : Bi-orthogonal basis of K_m

Advantage : Applicable to non-symmetric matrices
Short recurrence relations

Orthonomal basis (ONB) — Lanczos-Algorithm

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

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

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

Orthonomal basis (ONB) — Lanczos-Algorithm

Lanczos-Algorithm

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

Für $j = 1, \ldots, m$

- $w_j := \boldsymbol{A} v_j - c_j v_{j-1}$
- $a_j := (w_j, v_j)_2$
- $w_j := w_j - a_j v_j$
- $c_{j+1} := \|w_j\|_2$
- $c_{j+1} \neq 0$
  - Y: $v_{j+1} := \dfrac{w_j}{c_{j+1}}$
  - N: $v_{j+1} := \mathbf{0}$
    - STOP

Bi-orthonormal basis — BiLanczos-Algorithm

Definition: Bi-orthonormal

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

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

holds.

Idea:

Simultaneous calculation of bi-orthonormal bases

$$\begin{aligned} v_1, \ldots, v_m \quad &\text{of} \quad K_m = \mathrm{span}\{r_0, \ldots, A^{m-1} r_0\} \\ w_1, \ldots, w_m \quad &\text{of} \quad K_m^T = \mathrm{span}\{r_0, A^T r_0, \ldots, (A^T)^{m-1} r_0\}\,. \end{aligned}$$

Bi-orthonormal basis — BiLanczos-Algorithm

Idea:

Simultaneous calculation of bi-orthonormal bases

$$
\begin{aligned}
v_1, \ldots, v_m \quad &\text{of} \quad K_m = \text{span}\{r_0, \ldots, A^{m-1} r_o\}, \\
w_1, \ldots, w_m \quad &\text{of} \quad K_m^T = \text{span}\{r_0, A^T r_0, \ldots, (A^T)^{m-1} r_0\}.
\end{aligned}
$$

Effect:

- No symmetry constraint, BiLanczos = Lanczos, if A symmetric
- $v_1, \ldots, v_m$ are not orthonormal
- $W_m^T A V_m = T_m = \begin{pmatrix} \ddots & \ddots & & 0 \\ \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix}$

Biorthonomal basis — BiLanczos-Algorithm

BiLanczos-Algorithm

Bi-Lanczos-Algorithmus —

- $h_{1,0} = h_{0,1} := 0$
- $\boldsymbol{v}_0 = \boldsymbol{w}_0 := \boldsymbol{0}$
- $\boldsymbol{v}_1 = \boldsymbol{w}_1 := \dfrac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2}$ (4.3.48)
- für $j = 1, \ldots, m$
  - $h_{jj} := (\boldsymbol{w}_j, \boldsymbol{A}\,\boldsymbol{v}_j)_2$ (4.3.49)
  - $\boldsymbol{v}^*_{j+1} := \boldsymbol{A}\,\boldsymbol{v}_j - h_{jj}\boldsymbol{v}_j - h_{j-1,j}\boldsymbol{v}_{j-1}$ (4.3.50)
  - $\boldsymbol{w}^*_{j+1} := \boldsymbol{A}^T\boldsymbol{w}_j - h_{jj}\boldsymbol{w}_j - h_{j,j-1}\boldsymbol{w}_{j-1}$ (4.3.51)
  - $h_{j+1,j} := |(\boldsymbol{v}^*_{j+1}, \boldsymbol{w}^*_{j+1})_2|^{1/2}$
  - $h_{j+1,j} \neq 0$
    - Y:
      - $h_{j,j+1} := \dfrac{(\boldsymbol{v}^*_{j+1}, \boldsymbol{w}^*_{j+1})_2}{h_{j+1,j}}$ (4.3.52)
      - $\boldsymbol{v}_{j+1} := \dfrac{\boldsymbol{v}^*_{j+1}}{h_{j+1,j}}$ (4.3.53)
      - $\boldsymbol{w}_{j+1} := \dfrac{\boldsymbol{w}^*_{j+1}}{h_{j,j+1}}$ (4.3.54)
    - N:
      - $h_{j,j+1} := 0$
      - $\boldsymbol{v}_{j+1} := \boldsymbol{0}$
      - $\boldsymbol{w}_{j+1} := \boldsymbol{0}$
      - STOP

BiCG-Algorithm

- Ansatz: Skew Krylov subspace method based on

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

- Reason for short recurrence relations?

Bilanczos-Algorithmus $\downarrow$ BiCG-Methode	Arnoldi-Algorithmus $\downarrow$ FOM
$x_m = x_0 + V_m\alpha^m$ $b - Ax_m \perp K_m^T$	$x_m = x_0 + V_m\alpha^m$ $b - Ax_m = \perp K_m$
$(w_j, b - Ax_m) = 0, \quad j = 1, \ldots, m$ $\Leftrightarrow 0 = W_m^T(b - Ax_m)$ $= W_m^T(r_0 - AV_m\alpha^m)$ $= \lVert r_0 \rVert_2 e_1 - T_m\alpha^m$	$(v_j, b - Ax_m) = 0, \quad j = 1, \ldots, m$ $\Leftrightarrow 0 = V_m^T(b - Ax_m)$ $= V_m^T(r_0 - AV_m\alpha^m)$ $= \lVert r_0 \rVert_2 e_1 - H_m\alpha^m$
$\alpha^m = T_m^{-1}(\lVert r_0 \rVert_2 e_1)$	$\alpha^m = H_m^{-1}(\lVert r_0 \rVert_2 e_1)$

BiCG-Algorithm

BiCG-Algorithmus —

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

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

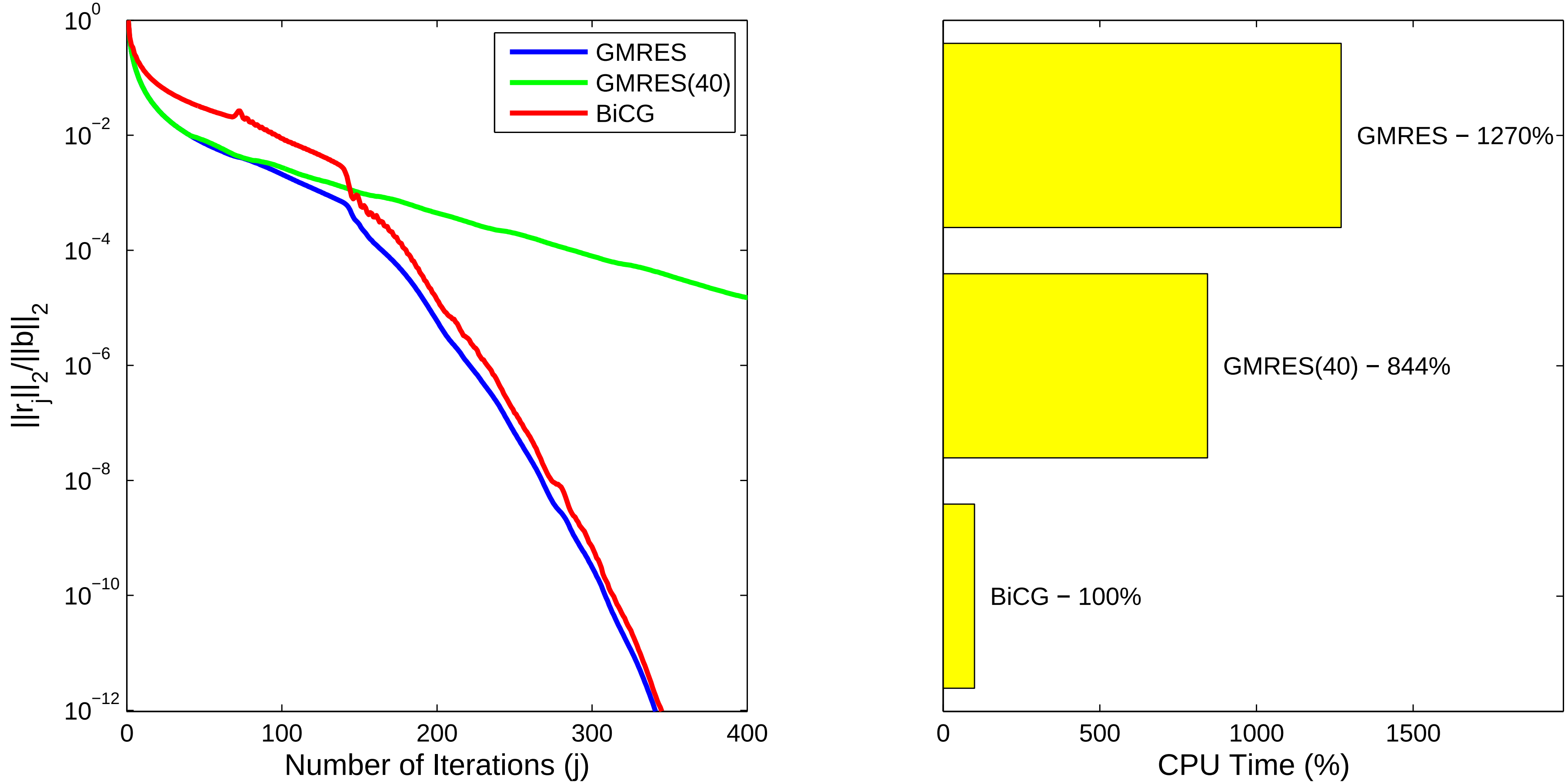
$j := 0$

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

- $\alpha_j := \dfrac{(\boldsymbol{r}_j, \boldsymbol{r}_j^*)_2}{(\boldsymbol{A}\,\boldsymbol{p}_j, \boldsymbol{p}_j^*)_2}$
- $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j + \alpha_j \boldsymbol{p}_j$
- $\boldsymbol{r}_{j+1} := \boldsymbol{r}_j - \alpha_j \boldsymbol{A}\,\boldsymbol{p}_j$
- $\boldsymbol{r}_{j+1}^* := \boldsymbol{r}_j^* - \alpha_j \boldsymbol{A}^T \boldsymbol{p}_j^*$
- $\beta_j := \dfrac{(\boldsymbol{r}_{j+1}, \boldsymbol{r}_{j+1}^*)_2}{(\boldsymbol{r}_j, \boldsymbol{r}_j^*)_2}$
- $\boldsymbol{p}_{j+1} := \boldsymbol{r}_{j+1} + \beta_j \boldsymbol{p}_j$
- $\boldsymbol{p}_{j+1}^* := \boldsymbol{r}_{j+1}^* + \beta_j \boldsymbol{p}_j^*$
- $j := j + 1$

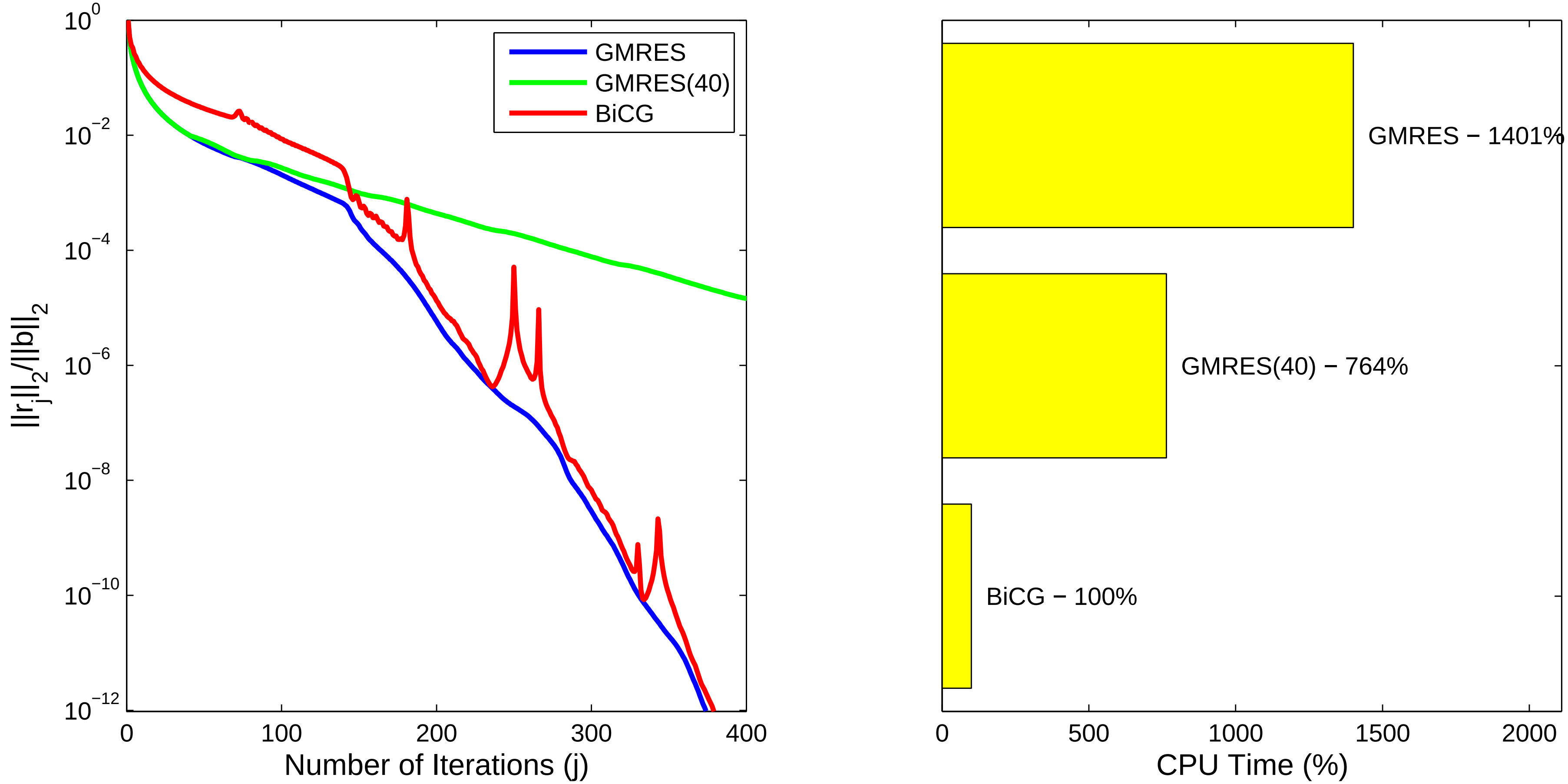
Comparison of GMRES, GMRES(m) and BiCG

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



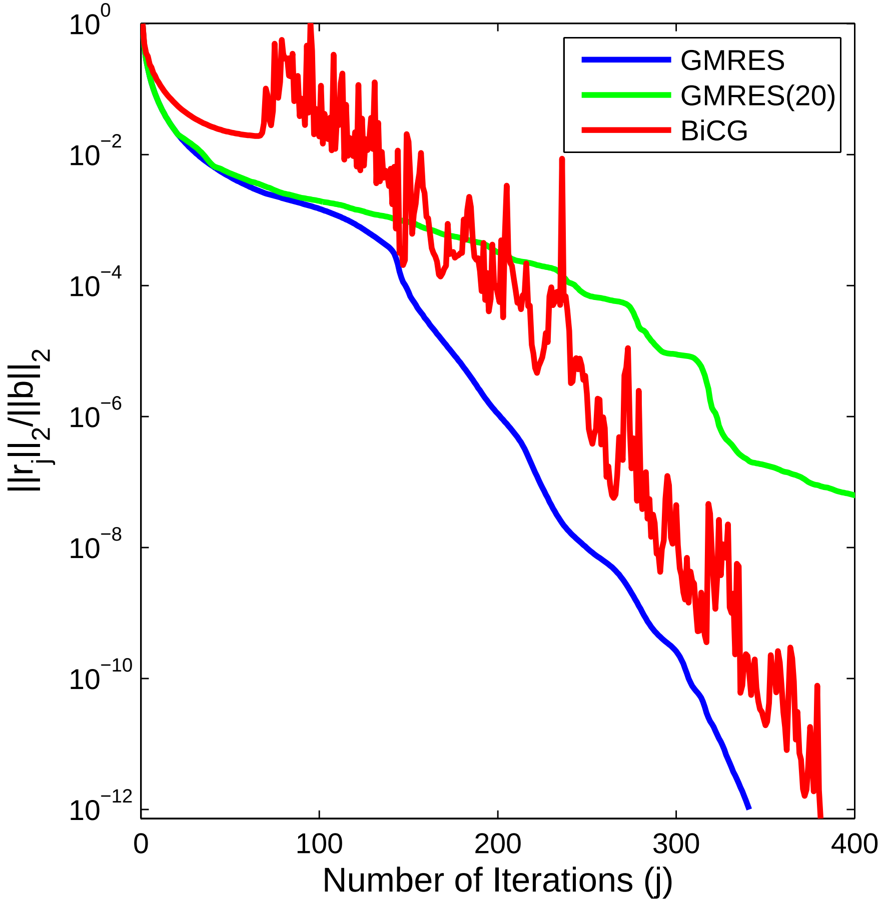
Comparison of GMRES, GMRES(m) and BiCG

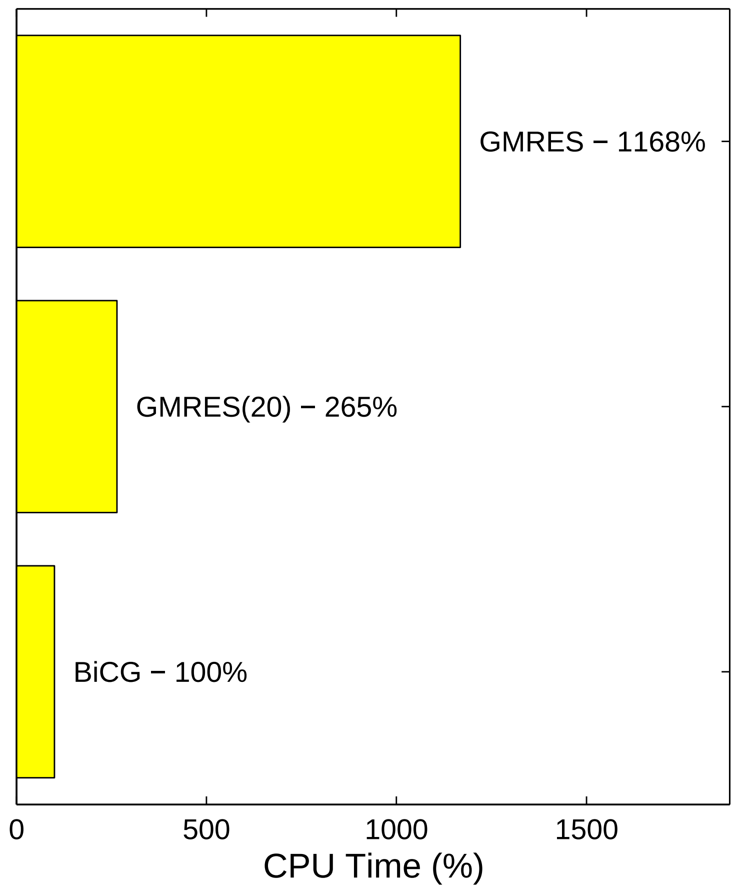
Test 2: Weak Convection-Diffusion ($\alpha = 0.1$, $\epsilon = 1$)



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
  $\rightarrow$ Oszillations in the convergence history
- Possible break down due to division by $(A p_j, p_j^*)$