

# Krylov Space Solvers

Martin H. Gutknecht

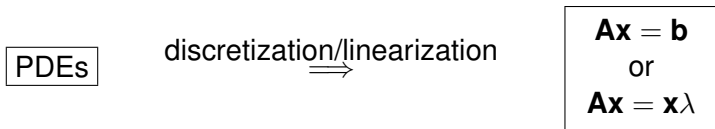
Seminar for Applied Mathematics  
ETH Zurich

International Symposium on Frontiers of Computational Science  
Nagoya, 12/13 Dec. 2005

# Sparse Matrices

**Large sparse linear systems of equations** or **large sparse matrix eigenvalue problems** appear in most applications of scientific computing.

In particular, discretization of PDEs with the **finite element method (FEM)** or with the **finite difference method (FDM)** leads to such problems:



Here,  $\mathbf{A}$  is  $N \times N$ , nonsingular, large, and **sparse**.

*large*:    say,  $500 \leq N \leq 100'000'000$

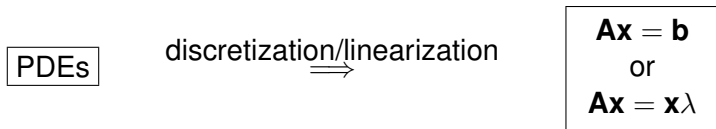
*sparse*:    most elements are zero;  
              say, 5 to 50 nonzero elements per row

Sparse matrices are stored in appropriate data formats.

# Sparse Matrices

**Large sparse linear systems of equations** or **large sparse matrix eigenvalue problems** appear in most applications of scientific computing.

In particular, discretization of PDEs with the **finite element method (FEM)** or with the **finite difference method (FDM)** leads to such problems:



Here,  $\mathbf{A}$  is  $N \times N$ , nonsingular, large, and **sparse**.

*large*:    say,  $500 \leq N \leq 100'000'000$

*sparse*:    most elements are zero;

          say, 5 to 50 nonzero elements per row

Sparse matrices are stored in appropriate data formats.

Often, when PDEs are solved, most computer time is spent for repeatedly solving the linear system or the eigenvalue problem.

In iterative methods  $\mathbf{A}$  is only needed to compute  $\mathbf{A}\mathbf{y}$  for any  $\mathbf{y} \in \mathbb{R}^N$ . Thus,  $\mathbf{A}$  may be given as a procedure/function

$$\mathbf{A} : \mathbf{y} \mapsto \mathbf{A}\mathbf{y}.$$

We will refer to this operation as a **matrix-vector product (MV)** although in practice the required computation may be much more complicated than multiplying a sparse matrix with a vector.

Of course, variations of the two problems  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and  $\mathbf{A}\mathbf{x} = \mathbf{x}\lambda$  appear also.

We concentrate here on iterative methods for linear systems of equations  $\mathbf{A}\mathbf{x} = \mathbf{b}$ .

Often, when PDEs are solved, most computer time is spent for repeatedly solving the linear system or the eigenvalue problem.

In iterative methods  $\mathbf{A}$  is only needed to compute  $\mathbf{A}\mathbf{y}$  for any  $\mathbf{y} \in \mathbb{R}^N$ . Thus,  $\mathbf{A}$  may be given as a procedure/function

$$\mathbf{A} : \mathbf{y} \mapsto \mathbf{A}\mathbf{y}.$$

We will refer to this operation as a **matrix-vector product (MV)** although in practice the required computation may be much more complicated than multiplying a sparse matrix with a vector.

Of course, variations of the two problems  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and  $\mathbf{A}\mathbf{x} = \mathbf{x}\lambda$  appear also.

We concentrate here on iterative methods for linear systems of equations  $\mathbf{A}\mathbf{x} = \mathbf{b}$ .

Often, when PDEs are solved, most computer time is spent for repeatedly solving the linear system or the eigenvalue problem.

In iterative methods  $\mathbf{A}$  is only needed to compute  $\mathbf{A}\mathbf{y}$  for any  $\mathbf{y} \in \mathbb{R}^N$ . Thus,  $\mathbf{A}$  may be given as a procedure/function

$$\mathbf{A} : \mathbf{y} \mapsto \mathbf{A}\mathbf{y}.$$

We will refer to this operation as a **matrix-vector product (MV)** although in practice the required computation may be much more complicated than multiplying a sparse matrix with a vector.

Of course, variations of the two problems  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and  $\mathbf{A}\mathbf{x} = \mathbf{x}\lambda$  appear also.

We concentrate here on iterative methods for linear systems of equations  $\mathbf{A}\mathbf{x} = \mathbf{b}$ .

# Direct vs. iterative methods

Alternative to iterative methods for linear systems:

**sparse direct solvers**, which are ingenious modifications of Gaussian elimination ( $\rightsquigarrow$  **sparse LU decomposition**)

There are very effective, hardware-dependent implementations.

*Rule of thumb: direct for 1D and 2D — iterative for 3D.*

Results by Stefan Röllin ['05<sub>Diss</sub>] on *semiconductor device simulation*:

Direct solver *PARDISO*, iterative solvers *Slip90* and *ILS* (all from ISS/ETH Zurich [Prof. Wolfgang Fichtner]).

Results shown are on a sequential computer, although both *PARDISO* and *ILS* are best on shared memory multiprocessor computers (OPENMP).

The new package *ILS* applies

- a **nonsymmetric permutation**; see Duff/Koster ['00<sub>SIMAX</sub>]
- a **symmetric permutation**, e.g. *nested dissection (ND)*, *reverse Cuthill-McKee*, or *multiple minimum degree (MMD)*
- **ILUT preconditioning**
- in **iterative method**, preferably **BICGSTAB**

Results by Stefan Röllin ['05<sub>Diss</sub>] on *semiconductor device simulation*:

Direct solver *PARDISO*, iterative solvers *Slip90* and *ILS* (all from ISS/ETH Zurich [Prof. Wolfgang Fichtner]).

Results shown are on a sequential computer, although both *PARDISO* and *ILS* are best on shared memory multiprocessor computers (OPENMP).

The new package *ILS* applies

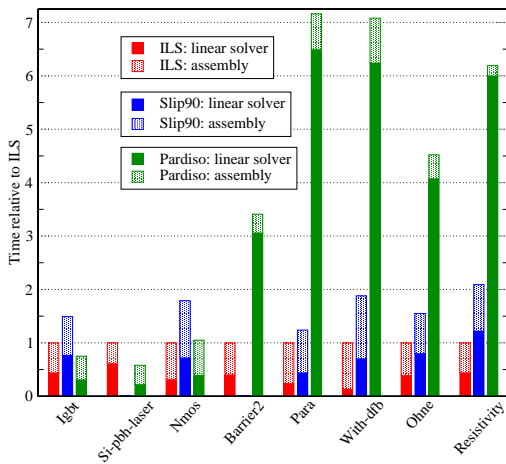
- a **nonsymmetric permutation**; see Duff/Koster ['00<sub>SIMAX</sub>]
- a **symmetric permutation**, e.g. *nested dissection (ND)*, *reverse Cuthill-McKee*, or *multiple minimum degree (MMD)*
- **ILUT preconditioning**
- in **iterative method**, preferably **BICGSTAB**

*Matrices used in the numerical experiments of Röllin*

| name of problem | unknowns | structural nonzeros | dim. |
|-----------------|----------|---------------------|------|
| Igbt-10         | 11'010   | 234'984             | 2D   |
| Si-pbh-laser-21 | 14'086   | 511'484             | 2D   |
| Nmos-10         | 18'627   | 387'457             | 2D   |
| Barrier2-7      | 115'625  | 6'372'663           | 3D   |
| Para-7          | 155'924  | 8'374'204           | 3D   |
| With-dfb-36     | 174'272  | 8'625'700           | 3D   |
| Ohne-9          | 183'038  | 11'170'886          | 3D   |
| Resistivity-9   | 318'026  | 19'455'650          | 3D   |

Nonsymmetric permutations may reduce the condition number and increase the diagonal dominance:

| Matrix     | Original |          |          | Scaled & permuted with MPS |          |          |
|------------|----------|----------|----------|----------------------------|----------|----------|
|            | Condest  | d.d.rows | d.d.cols | Condest                    | d.d.rows | d.d.cols |
| Igbt-10    | 4.73e+19 | 2'421    | 132      | 1.55e+08                   | 2'397    | 5'032    |
| Si-pbh...  | 7.11e+23 | 1'530    | 54       | 5.34e+08                   | 1'488    | 3'456    |
| Nmos-10    | 9.28e+20 | 2'951    | 57       | 6.09e+06                   | 2'951    | 6'862    |
| Barrier... | 2.99e+19 | 30'073   | 5'486    | 1.15e+19                   | 24'956   | 53'930   |
| Para-7     | 1.48e+19 | 41'144   | 5'768    | 2.74e+19                   | 39'386   | 76'920   |
| With-...   | 1.25e+20 | 33'424   | 3'837    | 9.75e+06                   | 32'582   | 75'299   |
| Ohne-9     | 7.48e+19 | 45'567   | 3'975    | 1.11e+20                   | 43'799   | 91'609   |
| Resist...  | failed   | 105'980  | 768      | 1.08e+09                   | 105'850  | 109'581  |



*Overall runtime for different solvers and simulations. Scaled to ILS. Dark bars show time spent for the solution of the linear systems. Some runs for PARDISO were done on faster computers with more memory.*

The simplest iterative method is **Jacobi iteration**.  
It is the same as diagonally preconditioned **fixed point iteration** or **Picard iteration**:

If  $\mathbf{D}$  is the diagonal of  $\mathbf{A}$ , and if  $\mathbf{D}$  is nonsingular, we transform  $\mathbf{Ax} = \mathbf{b}$  into

$$\boxed{\mathbf{x} = \widehat{\mathbf{B}}\mathbf{x} + \widehat{\mathbf{b}}} \quad \text{with} \quad \boxed{\widehat{\mathbf{B}} := \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}, \quad \widehat{\mathbf{b}} := \mathbf{D}^{-1}\mathbf{b}} \quad (1)$$

and apply the fixed point iteration  $\mathbf{x}_{n+1} := \widehat{\mathbf{B}}\mathbf{x}_n + \widehat{\mathbf{b}}$ .

## THEOREM

For the Jacobi iteration holds

$$\mathbf{x}_n \rightarrow \mathbf{x}_* \text{ for any } \mathbf{x}_0 \iff \rho(\hat{\mathbf{B}}) < 1, \quad (2)$$

where  $\mathbf{x}_* := \mathbf{A}^{-1}\mathbf{b}$  and  $\rho(\hat{\mathbf{B}}) := \max\{|\lambda| \mid \lambda \text{ eigenvalue of } \hat{\mathbf{B}}\}$  is the **spectral radius** of  $\hat{\mathbf{B}}$ .

## EXAMPLE 1.

Simplest example of a boundary value problem:

$$\begin{aligned} u'' &= f && \text{on } (0, 1), \\ u(0) &= u(1) = 0. \end{aligned} \tag{3}$$

Possible interpretation: steady state distribution of heat in a rod.

$$\mathbf{A} := \mathbf{T} := \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix},$$

$$\hat{\mathbf{B}} := \mathbf{I} - \frac{1}{2}\mathbf{T}, \quad \rho(\hat{\mathbf{B}}) = \rho(\mathbf{I} - \frac{1}{2}\mathbf{T}) = \cos \frac{\pi}{N+1}.$$

If  $N = 100$ ,  $\rho(\hat{\mathbf{B}}) = 0.99951628\dots$

**In order to reduce the error by a factor of 10, we need 4760 iterations!**

Since we cannot compute the  $n$ th **error (vector)**

$$\mathbf{d}_n := \mathbf{x}_n - \mathbf{x}_*, \quad (4)$$

for checking the convergence we use the  $n$ th **residual (vector)**

$$\mathbf{r}_n := \mathbf{b} - \mathbf{A}\mathbf{x}_n. \quad (5)$$

Note that

$$\mathbf{r}_n = -\mathbf{A}(\mathbf{x}_n - \mathbf{x}_*) = -\mathbf{A}\mathbf{d}_n. \quad (6)$$

Assuming  $\mathbf{D} = \mathbf{I}$  and letting  $\mathbf{B} := \mathbf{I} - \mathbf{A}$  we have

$$\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n = \mathbf{B}\mathbf{x}_n + \mathbf{b} - \mathbf{x}_n = \mathbf{x}_{n+1} - \mathbf{x}_n,$$

so we can rewrite the Jacobi iteration as

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{r}_n. \quad (7)$$

Multiplying this by  $-\mathbf{A}$ , we obtain a recursion for the residual:

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \mathbf{A}\mathbf{r}_n = \mathbf{B}\mathbf{r}_n. \quad (8)$$

Since we cannot compute the  $n$ th **error (vector)**

$$\mathbf{d}_n := \mathbf{x}_n - \mathbf{x}_*, \quad (4)$$

for checking the convergence we use the  $n$ th **residual (vector)**

$$\mathbf{r}_n := \mathbf{b} - \mathbf{A}\mathbf{x}_n. \quad (5)$$

Note that

$$\mathbf{r}_n = -\mathbf{A}(\mathbf{x}_n - \mathbf{x}_*) = -\mathbf{A}\mathbf{d}_n. \quad (6)$$

Assuming  $\mathbf{D} = \mathbf{I}$  and letting  $\mathbf{B} := \mathbf{I} - \mathbf{A}$  we have

$$\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n = \mathbf{B}\mathbf{x}_n + \mathbf{b} - \mathbf{x}_n = \mathbf{x}_{n+1} - \mathbf{x}_n,$$

so we can rewrite the Jacobi iteration as

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{r}_n. \quad (7)$$

Multiplying this by  $-\mathbf{A}$ , we obtain a recursion for the residual:

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \mathbf{A}\mathbf{r}_n = \mathbf{B}\mathbf{r}_n. \quad (8)$$

From (8) it follows by induction that

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \text{span} \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \quad (9)$$

where  $p_n$  is a polynomial of exact degree  $n$  and  $\mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)$  is the  $(n+1)$ th **Krylov (sub)space generated by  $\mathbf{A}$  from  $\mathbf{r}_0$** . Here, for Jacobi iteration,  $p_n(\zeta) = (1 - \zeta)^n$ .

From (7) we conclude that

$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{r}_0 + \dots + \mathbf{r}_{n-1} = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (10)$$

with a polynomial  $q_{n-1}$  of exact degree  $n-1$ .

$q_{n-1}(\mathbf{A})$  and  $p_n(\mathbf{A})$  require a total of  $n+1$  matrix-vector multiplications (**MVs**); this is the main work.

*Is there a better choice for  $\mathbf{x}_n$  in the same affine space?*

From (8) it follows by induction that

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \text{span} \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \quad (9)$$

where  $p_n$  is a polynomial of exact degree  $n$  and  $\mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0)$  is the  $(n+1)$ th **Krylov (sub)space generated by  $\mathbf{A}$  from  $\mathbf{r}_0$** . Here, for Jacobi iteration,  $p_n(\zeta) = (1 - \zeta)^n$ .

From (7) we conclude that

$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{r}_0 + \dots + \mathbf{r}_{n-1} = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \quad (10)$$

with a polynomial  $q_{n-1}$  of exact degree  $n - 1$ .

$q_{n-1}(\mathbf{A})$  and  $p_n(\mathbf{A})$  require a total of  $n + 1$  matrix-vector multiplications (**MVs**); this is the main work.

*Is there a better choice for  $\mathbf{x}_n$  in the same affine space?*

**DEFINITION.** Given a nonsingular  $\mathbf{A} \in \mathbb{C}^{N \times N}$  and  $\mathbf{y} \neq \mathbf{o} \in \mathbb{C}^N$ , the  $n$ th **Krylov (sub)space**  $\mathcal{K}_n(\mathbf{A}, \mathbf{y})$  generated by  $\mathbf{A}$  from  $\mathbf{y}$  is

$$\mathcal{K}_n \equiv \mathcal{K}_n(\mathbf{A}, \mathbf{y}) \equiv \text{span}(\mathbf{y}, \mathbf{A}\mathbf{y}, \dots, \mathbf{A}^{n-1}\mathbf{y}). \quad (11)$$



Clearly,

$$\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \dots$$

*When does the equal sign hold?*

**DEFINITION.** Given a nonsingular  $\mathbf{A} \in \mathbb{C}^{N \times N}$  and  $\mathbf{y} \neq \mathbf{o} \in \mathbb{C}^N$ , the  $n$ th **Krylov (sub)space**  $\mathcal{K}_n(\mathbf{A}, \mathbf{y})$  generated by  $\mathbf{A}$  from  $\mathbf{y}$  is

$$\mathcal{K}_n \equiv \mathcal{K}_n(\mathbf{A}, \mathbf{y}) \equiv \text{span}(\mathbf{y}, \mathbf{A}\mathbf{y}, \dots, \mathbf{A}^{n-1}\mathbf{y}). \quad (11)$$



Clearly,

$$\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \dots$$

*When does the equal sign hold?*

The following lemma answers this question.

**LEMMA**

There is a positive integer  $\bar{\nu} \equiv \bar{\nu}(\mathbf{y}, \mathbf{A})$  such that

$$\dim \mathcal{K}_n(\mathbf{A}, \mathbf{y}) = \begin{cases} n & \text{if } n \leq \bar{\nu}, \\ \bar{\nu} & \text{if } n \geq \bar{\nu}. \end{cases}$$

**DEFINITION.** The positive integer  $\bar{\nu} \equiv \bar{\nu}(\mathbf{y}, \mathbf{A})$  of Lemma 2 is called **grade of  $\mathbf{y}$  with respect to  $\mathbf{A}$** . ▲

## LEMMA

The nonnegative integer  $\bar{\nu}$  of Lemma 2 satisfies

$$\bar{\nu} = \min \left\{ n \mid \mathbf{A}^{-1}\mathbf{y} \in \mathcal{K}_n(\mathbf{A}, \mathbf{y}) \right\} \leq \partial \hat{\chi}_{\mathbf{A}},$$

where  $\partial \hat{\chi}_{\mathbf{A}}$  denotes the degree of the minimal polynomial of  $\mathbf{A}$ .

## COROLLARY

Let  $\mathbf{x}_*$  be the solution of  $\mathbf{Ax} = \mathbf{b}$  and let  $\mathbf{x}_0$  be any initial approximation of it and  $\mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0$  the corresponding residual. Moreover, let  $\bar{\nu} := \bar{\nu}(\mathbf{r}_0, \mathbf{A})$ . Then

$$\mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_{\bar{\nu}}(\mathbf{A}, \mathbf{r}_0).$$

## LEMMA

The nonnegative integer  $\bar{\nu}$  of Lemma 2 satisfies

$$\bar{\nu} = \min \left\{ n \mid \mathbf{A}^{-1} \mathbf{y} \in \mathcal{K}_n(\mathbf{A}, \mathbf{y}) \right\} \leq \partial \hat{\chi}_{\mathbf{A}},$$

where  $\partial \hat{\chi}_{\mathbf{A}}$  denotes the degree of the minimal polynomial of  $\mathbf{A}$ .

## COROLLARY

Let  $\mathbf{x}_*$  be the solution of  $\mathbf{A}\mathbf{x} = \mathbf{b}$  and let  $\mathbf{x}_0$  be any initial approximation of it and  $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$  the corresponding residual. Moreover, let  $\bar{\nu} := \bar{\nu}(\mathbf{r}_0, \mathbf{A})$ . Then

$$\mathbf{x}_* \in \mathbf{x}_0 + \mathcal{K}_{\bar{\nu}}(\mathbf{A}, \mathbf{r}_0).$$

**DEFINITION.** A **(standard) Krylov space method for solving a linear system  $\mathbf{Ax} = \mathbf{b}$**  or, briefly, a **(standard) Krylov space solver** is an iterative method starting from some initial approximation  $\mathbf{x}_0$  and the corresponding residual  $\mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0$  and generating for all, or at least most  $n$ , iterates  $\mathbf{x}_n$  such that

$$\boxed{\mathbf{x}_n - \mathbf{x}_0 = q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)} \quad (12)$$

with a polynomial  $q_{n-1}$  of exact degree  $n - 1$ . ▲

## LEMMA

The residuals of a Krylov space solver satisfy

$$\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \subseteq \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \quad (13)$$

where  $p_n$  is a polynomial of degree  $n$ , which is related to the polynomial  $q_{n-1}$  of (12) by

$$p_n(\zeta) = 1 - \zeta q_{n-1}(\zeta). \quad (14)$$

In particular,

$$p_n(0) = 1. \quad (15)$$

**DEFINITION.**  $p_n \in \mathcal{P}_n$  is the  $n$ th **residual polynomial**.  
Condition (15) is its **consistency condition**. ▲

**REMARK.** For some Krylov space solvers (e.g., BICG) there may exist exceptional situations, where for some  $n$  the iterate  $\mathbf{x}_n$  and the residual  $\mathbf{r}_n$  are not defined.

There are also **nonstandard Krylov space methods** where the search space for  $\mathbf{x}_n - \mathbf{x}_0$  is still a Krylov space, but one that differs from  $\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ . ▼

**REMARK.** With respect to the “influence on the development and practice of science and engineering in the 20th century”, Krylov space methods are considered as one of the ten most important classes of numerical methods. ▼

**REMARK.** For some Krylov space solvers (e.g., BICG) there may exist exceptional situations, where for some  $n$  the iterate  $\mathbf{x}_n$  and the residual  $\mathbf{r}_n$  are not defined.

There are also **nonstandard Krylov space methods** where the search space for  $\mathbf{x}_n - \mathbf{x}_0$  is still a Krylov space, but one that differs from  $\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ . ▼

**REMARK.** With respect to the “influence on the development and practice of science and engineering in the 20th century”, Krylov space methods are considered as one of the ten most important classes of numerical methods. ▼

**REMARK.** For some Krylov space solvers (e.g., BICG) there may exist exceptional situations, where for some  $n$  the iterate  $\mathbf{x}_n$  and the residual  $\mathbf{r}_n$  are not defined.

There are also **nonstandard Krylov space methods** where the search space for  $\mathbf{x}_n - \mathbf{x}_0$  is still a Krylov space, but one that differs from  $\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ . ▼

**REMARK.** With respect to the “influence on the development and practice of science and engineering in the 20th century”, Krylov space methods are considered as one of the ten most important classes of numerical methods. ▼

## Alternative names for Krylov (sub)space solvers/methods:

- **gradient methods**,
- **semi-iterative methods**,
- **polynomial acceleration methods**,
- **polynomial preconditioners**,
- **Krylov subspace iterations**.

# Preconditioning

When applied to large real-world problems Krylov space solvers often converge very slowly — if at all. In practice, Krylov space solvers are therefore nearly always applied with **preconditioning**:  $\mathbf{Ax} = \mathbf{b}$  is replaced by

$$\underbrace{\mathbf{CA}}_{\hat{\mathbf{A}}} \mathbf{x} = \underbrace{\mathbf{Cb}}_{\hat{\mathbf{b}}} \quad \text{left preconditioner} \quad (16)$$

or

$$\underbrace{\mathbf{AC}}_{\hat{\mathbf{A}}} \underbrace{\mathbf{C}^{-1}\mathbf{x}}_{\hat{\mathbf{x}}} = \mathbf{b} \quad \text{right preconditioner} \quad (17)$$

or

$$\underbrace{\mathbf{C}_L \mathbf{A} \mathbf{C}_R}_{\hat{\mathbf{A}}} \underbrace{\mathbf{C}_R^{-1} \mathbf{x}}_{\hat{\mathbf{x}}} = \underbrace{\mathbf{C}_L \mathbf{b}}_{\hat{\mathbf{b}}} \quad \text{split preconditioner} \quad (18)$$

# Energy norm minimization

Stable states are characterized by minimum energy.

Discretization leads to the minimization of a quadratic function:

$$\Psi(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + \gamma \quad (19)$$

with an *spd matrix*  $\mathbf{A}$ . (We assume real data now.)

$\Psi$  is convex and has a unique minimum. Its gradient is

$$\nabla \Psi(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} = -\mathbf{r}, \quad (20)$$

where  $\mathbf{r}$  is the residual corresponding to  $\mathbf{x}$ . Hence,

$$\mathbf{x} \text{ minimizer of } \Psi \iff \nabla \Psi(\mathbf{x}) = \mathbf{0} \iff \mathbf{A} \mathbf{x} = \mathbf{b}. \quad (21)$$

If  $\mathbf{x}_*$  denotes the minimizer and  $\mathbf{d} := \mathbf{x} - \mathbf{x}_*$  the error vector, and if we choose  $\gamma := \frac{1}{2} \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}$ , it is easily seen that

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{A} \mathbf{x} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

# Energy norm minimization

Stable states are characterized by minimum energy.

Discretization leads to the minimization of a quadratic function:

$$\Psi(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + \gamma \quad (19)$$

with an *spd matrix*  $\mathbf{A}$ . (We assume real data now.)

$\Psi$  is convex and has a unique minimum. Its gradient is

$$\nabla \Psi(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} = -\mathbf{r}, \quad (20)$$

where  $\mathbf{r}$  is the residual corresponding to  $\mathbf{x}$ . Hence,

$$\mathbf{x} \text{ minimizer of } \Psi \iff \nabla \Psi(\mathbf{x}) = \mathbf{0} \iff \mathbf{A} \mathbf{x} = \mathbf{b}. \quad (21)$$

If  $\mathbf{x}_*$  denotes the minimizer and  $\mathbf{d} := \mathbf{x} - \mathbf{x}_*$  the error vector, and if we choose  $\gamma := \frac{1}{2} \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}$ , it is easily seen that

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{A} \mathbf{x} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

# Energy norm minimization

Stable states are characterized by minimum energy.

Discretization leads to the minimization of a quadratic function:

$$\Psi(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} + \gamma \quad (19)$$

with an *spd matrix*  $\mathbf{A}$ . (We assume real data now.)

$\Psi$  is convex and has a unique minimum. Its gradient is

$$\nabla \Psi(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} = -\mathbf{r}, \quad (20)$$

where  $\mathbf{r}$  is the residual corresponding to  $\mathbf{x}$ . Hence,

$$\mathbf{x} \text{ minimizer of } \Psi \iff \nabla \Psi(\mathbf{x}) = \mathbf{0} \iff \mathbf{A} \mathbf{x} = \mathbf{b}. \quad (21)$$

If  $\mathbf{x}_*$  denotes the minimizer and  $\mathbf{d} := \mathbf{x} - \mathbf{x}_*$  the error vector, and if we choose  $\gamma := \frac{1}{2} \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}$ , it is easily seen that

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{A} \mathbf{x} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

Again:

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{Ax} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

Here

$$\|\mathbf{d}\|_{\mathbf{A}} = \sqrt{\mathbf{d}^T \mathbf{A} \mathbf{d}} \quad (23)$$

is the **A-norm** or **energy norm** of the error vector.

In summary: **If  $\mathbf{A}$  is spd, to minimize the quadratic function  $\Psi$  means to minimize the energy norm of the error vector of the linear system  $\mathbf{Ax} = \mathbf{b}$ .**

**The minimizer  $\mathbf{x}_*$  is the solution of  $\mathbf{Ax} = \mathbf{b}$ .**

Since  $\mathbf{A}$  is spd, the level curves  $\Psi(\mathbf{x}) = \text{const}$  are ellipses if  $N = 2$  and ellipsoids if  $N = 3$ .

Again:

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{Ax} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

Here

$$\|\mathbf{d}\|_{\mathbf{A}} = \sqrt{\mathbf{d}^T \mathbf{A} \mathbf{d}} \quad (23)$$

is the **A-norm** or **energy norm** of the error vector.

In summary: **If  $\mathbf{A}$  is spd, to minimize the quadratic function  $\Psi$  means to minimize the energy norm of the error vector of the linear system  $\mathbf{Ax} = \mathbf{b}$ .**

**The minimizer  $\mathbf{x}_*$  is the solution of  $\mathbf{Ax} = \mathbf{b}$ .**

Since  $\mathbf{A}$  is spd, the level curves  $\Psi(\mathbf{x}) = \text{const}$  are ellipses if  $N = 2$  and ellipsoids if  $N = 3$ .

Again:

$$\|\mathbf{d}\|_{\mathbf{A}}^2 = \|\mathbf{x} - \mathbf{x}_*\|_{\mathbf{A}}^2 = \|\mathbf{Ax} - \mathbf{b}\|_{\mathbf{A}^{-1}}^2 = \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2 = 2 \Psi(\mathbf{x}). \quad (22)$$

Here

$$\|\mathbf{d}\|_{\mathbf{A}} = \sqrt{\mathbf{d}^T \mathbf{A} \mathbf{d}} \quad (23)$$

is the **A-norm** or **energy norm** of the error vector.

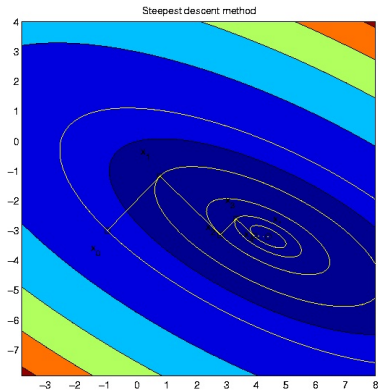
In summary: **If  $\mathbf{A}$  is spd, to minimize the quadratic function  $\Psi$  means to minimize the energy norm of the error vector of the linear system  $\mathbf{Ax} = \mathbf{b}$ .**

**The minimizer  $\mathbf{x}_*$  is the solution of  $\mathbf{Ax} = \mathbf{b}$ .**

Since  $\mathbf{A}$  is spd, the level curves  $\Psi(\mathbf{x}) = \text{const}$  are ellipses if  $N = 2$  and ellipsoids if  $N = 3$ .

# The method of steepest descent

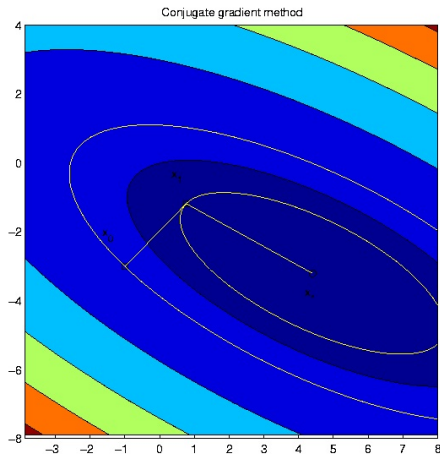
The above suggest to find the minimizer of  $\Psi$  by moving down the surface representing  $\Psi$  in the direction of steepest descent. In each step we go to the lowest point in this direction.



*Even for a  $2 \times 2$  system many steps are needed to get close to the solution.*

# Conjugate direction methods

We can do much better: we choose the second direction **conjugate** or **A-orthogonal** to the first one:  $\mathbf{v}_1^T \mathbf{A} \mathbf{v}_0 = 0$ .



*Now two steps are enough!*

*How does this generalize to  $N$  dimensions?*

We choose **search directions** or **direction vectors**  $\mathbf{v}_n$  that are conjugate ( $\mathbf{A}$ -orthogonal) to each other:

$$\mathbf{v}_n^T \mathbf{A} \mathbf{v}_k = 0, \quad k = 0, \dots, n-1, \quad (24)$$

and define

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{v}_n \omega_n, \quad (25)$$

so that

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A} \mathbf{v}_n \omega_n. \quad (26)$$

$\omega_n$  is chosen such that the  $\mathbf{A}$ -norm of the error is minimized on the line

$$\omega \mapsto \mathbf{x}_n + \mathbf{v}_n \omega. \quad (27)$$

This leads to

$$\omega_n := \frac{\langle \mathbf{r}_n, \mathbf{v}_n \rangle}{\langle \mathbf{v}_n, \mathbf{A} \mathbf{v}_n \rangle}. \quad (28)$$

*How does this generalize to  $N$  dimensions?*

We choose **search directions** or **direction vectors**  $\mathbf{v}_n$  that are conjugate ( $\mathbf{A}$ -orthogonal) to each other:

$$\mathbf{v}_n^T \mathbf{A} \mathbf{v}_k = 0, \quad k = 0, \dots, n-1, \quad (24)$$

and define

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{v}_n \omega_n, \quad (25)$$

so that

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A} \mathbf{v}_n \omega_n. \quad (26)$$

$\omega_n$  is chosen such that the  $\mathbf{A}$ -norm of the error is minimized on the line

$$\omega \mapsto \mathbf{x}_n + \mathbf{v}_n \omega. \quad (27)$$

This leads to

$$\omega_n := \frac{\langle \mathbf{r}_n, \mathbf{v}_n \rangle}{\langle \mathbf{v}_n, \mathbf{A} \mathbf{v}_n \rangle}. \quad (28)$$

**DEFINITION.** Any iterative method satisfying (24), (25), and (28) is called a **conjugate direction (CD) method**. ▲

By definition, such a method chooses the step length  $\omega_n$  so that  $\mathbf{x}_{n+1}$  is locally optimal on the search line.

But does it also yield the best

$$\mathbf{x}_{n+1} \in \mathbf{x}_0 + \text{span} \{ \mathbf{v}_0, \dots, \mathbf{v}_n \} \quad (29)$$

with respect to the  $\mathbf{A}$ -norm of the error?

Yes!

**DEFINITION.** Any iterative method satisfying (24), (25), and (28) is called a **conjugate direction (CD) method**. ▲

By definition, such a method chooses the step length  $\omega_n$  so that  $\mathbf{x}_{n+1}$  is locally optimal on the search line.

But does it also yield the best

$$\mathbf{x}_{n+1} \in \mathbf{x}_0 + \text{span} \{ \mathbf{v}_0, \dots, \mathbf{v}_n \} \quad (29)$$

with respect to the  $\mathbf{A}$ -norm of the error?

Yes!

**DEFINITION.** Any iterative method satisfying (24), (25), and (28) is called a **conjugate direction (CD) method**. ▲

By definition, such a method chooses the step length  $\omega_n$  so that  $\mathbf{x}_{n+1}$  is locally optimal on the search line.

But does it also yield the best

$$\mathbf{x}_{n+1} \in \mathbf{x}_0 + \text{span} \{ \mathbf{v}_0, \dots, \mathbf{v}_n \} \quad (29)$$

with respect to the  $\mathbf{A}$ -norm of the error?

**Yes!**

## THEOREM

*For a conjugate direction method the problem of minimizing the energy norm of the error of an approximate solution of the form (29) decouples into  $n + 1$  one-dimensional minimization problems on the lines  $\omega \mapsto \mathbf{x}_k + \mathbf{v}_k\omega$ ,  $k = 0, 1, \dots, n$ . A conjugate direction method yields after  $n + 1$  steps the approximate solution of the form (29) that minimizes the energy norm of the error in this affine space.*

PROOF. One shows that

$$\Psi(\mathbf{x}_{n+1}) = \Psi(\mathbf{x}_n) - \omega_n \mathbf{v}_n^T \mathbf{r}_0 + \frac{1}{2} \omega_n^2 \mathbf{v}_n^T \mathbf{A} \mathbf{v}_n.$$

□

Conjugate direction (CD) methods as well as the special case of the conjugate gradient (CG) method treated next are due to [Hestenes and Stiefel \(1952\)](#).

## THEOREM

*For a conjugate direction method the problem of minimizing the energy norm of the error of an approximate solution of the form (29) decouples into  $n + 1$  one-dimensional minimization problems on the lines  $\omega \mapsto \mathbf{x}_k + \mathbf{v}_k\omega$ ,  $k = 0, 1, \dots, n$ . A conjugate direction method yields after  $n + 1$  steps the approximate solution of the form (29) that minimizes the energy norm of the error in this affine space.*

PROOF. One shows that

$$\Psi(\mathbf{x}_{n+1}) = \Psi(\mathbf{x}_n) - \omega_n \mathbf{v}_n^T \mathbf{r}_0 + \frac{1}{2} \omega_n^2 \mathbf{v}_n^T \mathbf{A} \mathbf{v}_n.$$

□

Conjugate direction (CD) methods as well as the special case of the conjugate gradient (CG) method treated next are due to [Hestenes and Stiefel \(1952\)](#).

# The conjugate gradient (CG) method

In general, conjugate direction methods are not Krylov space solvers, but with suitably chosen search directions they are.

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{v}_0\omega_0 + \cdots + \mathbf{v}_n\omega_n \in \mathbf{x}_0 + \text{span} \{ \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n \} \quad (30)$$

shows that we need

$$\text{span} \{ \mathbf{v}_0, \dots, \mathbf{v}_n \} = \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \quad n = 0, 1, 2, \dots \quad (31)$$

**DEFINITION.** The **conjugate gradient (CG) method** is the conjugate direction method with the choice (31). ▲

The previous theorem leads to the main result on CG:

## THEOREM

*The CG method yields approx. solutions  $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$  that are optimal in the sense that they minimize the energy norm ( $\mathbf{A}$ -norm) of the error (i.e., the  $\mathbf{A}^{-1}$ -norm of the residual) for  $\mathbf{x}_n$  from this affine space.*



# The conjugate gradient (CG) method

In general, conjugate direction methods are not Krylov space solvers, but with suitably chosen search directions they are.

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{v}_0\omega_0 + \cdots + \mathbf{v}_n\omega_n \in \mathbf{x}_0 + \text{span} \{ \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n \} \quad (30)$$

shows that we need

$$\text{span} \{ \mathbf{v}_0, \dots, \mathbf{v}_n \} = \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0), \quad n = 0, 1, 2, \dots \quad (31)$$

**DEFINITION.** The **conjugate gradient (CG) method** is the conjugate direction method with the choice (31). ▲

The previous theorem leads to the main result on CG:

## THEOREM

*The CG method yields approx. solutions  $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$  that are optimal in the sense that they minimize the energy norm ( $\mathbf{A}$ -norm) of the error (i.e., the  $\mathbf{A}^{-1}$ -norm of the residual) for  $\mathbf{x}_n$  from this affine space.*

**Properties of the CG method (Ass.:  $\mathbf{A}$  spd or Hpd):**

$$\mathbf{x}_n - \mathbf{x}_0 \in \mathcal{K}_n, \quad \mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n \subseteq \mathcal{K}_{n+1}, \quad \mathbf{v}_n \in \mathcal{K}_{n+1},$$

The residuals  $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$  form an *orthogonal basis* of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{r}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$  form a *conjugate basis* of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{v}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

$\|\mathbf{x}_n - \mathbf{x}_\star\|_{\mathbf{A}} = \|\mathbf{r}_n\|_{\mathbf{A}^{-1}}$  is *shortest possible*.

Associated with this minimality is the **Galerkin condition**

$$\mathcal{K}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1} \tag{32}$$

**Properties of the CG method (Ass.:  $\mathbf{A}$  spd or Hpd):**

$$\mathbf{x}_n - \mathbf{x}_0 \in \mathcal{K}_n, \quad \mathbf{r}_n \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n \subseteq \mathcal{K}_{n+1}, \quad \mathbf{v}_n \in \mathcal{K}_{n+1},$$

The residuals  $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$  form an *orthogonal basis* of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{r}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$  form a *conjugate basis* of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{v}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

$\|\mathbf{x}_n - \mathbf{x}_\star\|_{\mathbf{A}} = \|\mathbf{r}_n\|_{\mathbf{A}^{-1}}$  is *shortest possible*.

Associated with this minimality is the **Galerkin condition**

$$\mathcal{K}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1} \tag{32}$$

## Algorithm

For solving  $\mathbf{Ax} = \mathbf{b}$  choose an initial approximation  $\mathbf{x}_0$ , and let  $\mathbf{v}_0 := \mathbf{r}_0 := \mathbf{b} - \mathbf{Ax}_0$  and  $\delta_0 := \|\mathbf{r}_0\|^2$ . Then, for  $n = 0, 1, 2, \dots$ , compute

$$\delta'_n := \|\mathbf{v}_n\|_{\mathbf{A}}^2, \quad (33a)$$

$$\omega_n := \delta_n / \delta'_n, \quad (33b)$$

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \mathbf{v}_n \omega_n, \quad (33c)$$

$$\mathbf{r}_{n+1} := \mathbf{r}_n - \mathbf{Av}_n \omega_n, \quad (33d)$$

$$\delta_{n+1} := \|\mathbf{r}_{n+1}\|^2, \quad (33e)$$

$$\psi_n := -\delta_{n+1} / \delta_n, \quad (33f)$$

$$\mathbf{v}_{n+1} := \mathbf{r}_{n+1} - \mathbf{v}_n \psi_n. \quad (33g)$$

If  $\|\mathbf{r}_{n+1}\| \leq \text{tol}$ , the algorithm terminates and  $\mathbf{x}_{n+1}$  is a sufficiently accurate approximation of the solution.

# The conjugate residual (CR) method

The **conjugate residual (CR) method** is fully analogous to the CG method, but the 2-norm is replaced by the  $\mathbf{A}$ -norm.

## Properties of the CR method (Ass.: $\mathbf{A}$ Herm.):

The residuals  $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$  form an  **$\mathbf{A}$ -orthogonal basis** of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{r}_m, \mathbf{A}\mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$  form a  **$\mathbf{A}^2$ -orthogonal basis** of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{v}_m, \mathbf{A}^2\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

$\|\mathbf{x}_n - \mathbf{x}_*\|_{\mathbf{A}^2} = \|\mathbf{r}_n\|_2$  is shortest possible.

Associated with this minimality is the **Galerkin condition**

$$\mathbf{A}\mathcal{K}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1}.$$

(34)

# The conjugate residual (CR) method

The **conjugate residual (CR) method** is fully analogous to the CG method, but the 2-norm is replaced by the  $\mathbf{A}$ -norm.

## Properties of the CR method (Ass.: $\mathbf{A}$ Herm.):

The residuals  $\{\mathbf{r}_n\}_{n=0}^{\bar{\nu}-1}$  form an  **$\mathbf{A}$ -orthogonal basis** of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{r}_m, \mathbf{A}\mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^{\bar{\nu}-1}$  form a  **$\mathbf{A}^2$ -orthogonal basis** of  $\mathcal{K}_{\bar{\nu}}$ :

$$\langle \mathbf{v}_m, \mathbf{A}^2\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

$\|\mathbf{x}_n - \mathbf{x}_*\|_{\mathbf{A}^2} = \|\mathbf{r}_n\|_2$  is shortest possible.

Associated with this minimality is the **Galerkin condition**

$$\mathbf{A}\mathcal{K}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1}. \quad (34)$$

# Nonsymmetric systems

Solving nonsymmetric (or non-Hermitian) linear systems iteratively with Krylov space solvers is considerably more difficult and costly than symmetric (or Hermitian) systems.

There are two fundamentally different ways to generalize CG:

- Maintain the orthogonality of the projection and the related minimality of the error by constructing either orthogonal residuals  $\mathbf{r}_n$  ( $\rightsquigarrow$  **generalized CG (GCG)**) or  $\mathbf{A}^T \mathbf{A}$ -orthogonal search directions  $\mathbf{v}_n$  ( $\rightsquigarrow$  **generalized CR (GCR)**).

*The recursions involve all previously constructed residuals or search directions and all previously constructed iterates.*

- Maintain short recurrence formulas for residuals, direction vectors and iterates  $\rightsquigarrow$  **biconjugate gradient (BiCG) method and to Lanczos-type product methods (LTPM)**.

*At best oblique projection method; no minimality of error vectors or residuals.*

# Nonsymmetric systems

Solving nonsymmetric (or non-Hermitian) linear systems iteratively with Krylov space solvers is considerably more difficult and costly than symmetric (or Hermitian) systems.

There are two fundamentally different ways to generalize CG:

- Maintain the orthogonality of the projection and the related minimality of the error by constructing either orthogonal residuals  $\mathbf{r}_n$  ( $\rightsquigarrow$  **generalized CG (GCG)**) or  $\mathbf{A}^T \mathbf{A}$ -orthogonal search directions  $\mathbf{v}_n$  ( $\rightsquigarrow$  **generalized CR (GCR)**).

*The recursions involve all previously constructed residuals or search directions and all previously constructed iterates.*

- Maintain short recurrence formulas for residuals, direction vectors and iterates  $\rightsquigarrow$  **biconjugate gradient (BiCG) method and to Lanczos-type product methods (LTPM)**.  
*At best oblique projection method; no minimality of error vectors or residuals.*

# Nonsymmetric systems

Solving nonsymmetric (or non-Hermitian) linear systems iteratively with Krylov space solvers is considerably more difficult and costly than symmetric (or Hermitian) systems.

There are two fundamentally different ways to generalize CG:

- Maintain the orthogonality of the projection and the related minimality of the error by constructing either orthogonal residuals  $\mathbf{r}_n$  ( $\rightsquigarrow$  **generalized CG (GCG)**) or  $\mathbf{A}^T \mathbf{A}$ -orthogonal search directions  $\mathbf{v}_n$  ( $\rightsquigarrow$  **generalized CR (GCR)**).

*The recursions involve all previously constructed residuals or search directions and all previously constructed iterates.*

- Maintain short recurrence formulas for residuals, direction vectors and iterates  $\rightsquigarrow$  **biconjugate gradient (BiCG) method and to Lanczos-type product methods (LTPM)**.  
*At best oblique projection method; no minimality of error vectors or residuals.*

# The biconjugate gradient (BICG) method

While CG (for spd  $\mathbf{A}$ ) has mutually *orthogonal residuals*  $\mathbf{r}_n$  with

$$\mathbf{r}_n = \mathbf{p}_n(\mathbf{A})\mathbf{r}_0 \in \text{span} \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0),$$

BICG construct in the same space residuals *orthogonal to a dual Krylov space spanned by “shadow residuals”*

$$\tilde{\mathbf{r}}_n = \tilde{\mathbf{p}}_n(\mathbf{A}^T)\tilde{\mathbf{r}}_0 \in \text{span} \{\tilde{\mathbf{r}}_0, \mathbf{A}^T\tilde{\mathbf{r}}_0, \dots, (\mathbf{A}^T)^n\tilde{\mathbf{r}}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}^T, \tilde{\mathbf{r}}_0) \equiv: \tilde{\mathcal{K}}_{n+1}$$

$\tilde{\mathbf{r}}_0$  can be chosen freely.

There are two Galerkin conditions

$$\tilde{\mathcal{K}}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1}, \quad \mathcal{K}_n \perp \tilde{\mathbf{r}}_n \in \tilde{\mathcal{K}}_{n+1},$$

but only the first one is relevant for determining  $\mathbf{x}_n$ .

# The biconjugate gradient (BICG) method

While CG (for spd  $\mathbf{A}$ ) has mutually *orthogonal residuals*  $\mathbf{r}_n$  with

$$\mathbf{r}_n = \tilde{p}_n(\mathbf{A})\mathbf{r}_0 \in \text{span} \{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^n\mathbf{r}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}, \mathbf{r}_0),$$

BICG construct in the same space residuals *orthogonal to a dual Krylov space spanned by “shadow residuals”*

$$\tilde{\mathbf{r}}_n = \tilde{p}_n(\mathbf{A}^T)\tilde{\mathbf{r}}_0 \in \text{span} \{\tilde{\mathbf{r}}_0, \mathbf{A}^T\tilde{\mathbf{r}}_0, \dots, (\mathbf{A}^T)^n\tilde{\mathbf{r}}_0\} \equiv: \mathcal{K}_{n+1}(\mathbf{A}^T, \tilde{\mathbf{r}}_0) \equiv: \tilde{\mathcal{K}}_{n+1}$$

$\tilde{\mathbf{r}}_0$  can be chosen freely.

There are two Galerkin conditions

$$\tilde{\mathcal{K}}_n \perp \mathbf{r}_n \in \mathcal{K}_{n+1},$$

$$\mathcal{K}_n \perp \tilde{\mathbf{r}}_n \in \tilde{\mathcal{K}}_{n+1},$$

but only the first one is relevant for determining  $\mathbf{x}_n$ .

The residuals  $\{\mathbf{r}_n\}_{n=0}^m$  and the shadow residuals  $\{\tilde{\mathbf{r}}_n\}_{n=0}^m$  form *biorthogonal* or *dual bases* of  $\mathcal{K}_{m+1}$  and  $\tilde{\mathcal{K}}_{m+1}$ :

$$\langle \tilde{\mathbf{r}}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^m$  and the **“shadow search directions”**  $\{\tilde{\mathbf{v}}_n\}_{n=0}^m$  form *biconjugate bases* of  $\mathcal{K}_{m+1}$  and  $\mathcal{K}_{m+1}$ :

$$\langle \tilde{\mathbf{v}}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

BICG goes back to [Lanczos \(1952\)](#) and [Fletcher \(1976\)](#).

Each step requires two MVs to extend  $\mathcal{K}_n$  and  $\tilde{\mathcal{K}}_n$ : one multiplication by  $\mathbf{A}$  and one by  $\mathbf{A}^T$ .

The residuals  $\{\mathbf{r}_n\}_{n=0}^m$  and the shadow residuals  $\{\tilde{\mathbf{r}}_n\}_{n=0}^m$  form *biorthogonal* or *dual bases* of  $\mathcal{K}_{m+1}$  and  $\tilde{\mathcal{K}}_{m+1}$ :

$$\langle \tilde{\mathbf{r}}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^m$  and the **“shadow search directions”**  $\{\tilde{\mathbf{v}}_n\}_{n=0}^m$  form *biconjugate bases* of  $\mathcal{K}_{m+1}$  and  $\mathcal{K}_{m+1}$ :

$$\langle \tilde{\mathbf{v}}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

BICG goes back to [Lanczos \(1952\)](#) and [Fletcher \(1976\)](#).

Each step requires two MVs to extend  $\mathcal{K}_n$  and  $\tilde{\mathcal{K}}_n$ : one multiplication by  $\mathbf{A}$  and one by  $\mathbf{A}^T$ .

The residuals  $\{\mathbf{r}_n\}_{n=0}^m$  and the shadow residuals  $\{\tilde{\mathbf{r}}_n\}_{n=0}^m$  form *biorthogonal* or *dual bases* of  $\mathcal{K}_{m+1}$  and  $\tilde{\mathcal{K}}_{m+1}$ :

$$\langle \tilde{\mathbf{r}}_m, \mathbf{r}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta_n \neq 0 & \text{if } m = n. \end{cases}$$

The search directions  $\{\mathbf{v}_n\}_{n=0}^m$  and the **“shadow search directions”**  $\{\tilde{\mathbf{v}}_n\}_{n=0}^m$  form *biconjugate bases* of  $\mathcal{K}_{m+1}$  and  $\mathcal{K}_{m+1}$ :

$$\langle \tilde{\mathbf{v}}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0 & \text{if } m \neq n, \\ \delta'_n \neq 0 & \text{if } m = n. \end{cases}$$

BICG goes back to [Lanczos \(1952\)](#) and [Fletcher \(1976\)](#).

Each step requires two MVs to extend  $\mathcal{K}_n$  and  $\tilde{\mathcal{K}}_n$ : one multiplication by  $\mathbf{A}$  and one by  $\mathbf{A}^T$ .

# Lanczos-type product methods (LTPMs)

Sonneveld (1989) found with the **(bi)conjugate gradient squared method (BICGS)** a way to replace the multiplication with  $\mathbf{A}^T$  by a second one with  $\mathbf{A}$ .

The  $n$ th residual polynomial is  $p_n^2$ , where  $p_n$  is the  $n$ th BICG residual polynomial, which satisfies a three-term recursion.

In each step the dimension of the Krylov space and the search space increases by 2. Convergence is nearly twice as fast, but often somewhat erratic.

**BICGSTAB** (Van der Vorst, 1992) includes some local optimization and smoothing.

The  $n$ th residual polynomial is  $p_n t_n$ , where

$$t_{n+1}(\zeta) = (1 - \chi_{n+1}\zeta)t_n(\zeta).$$

Van der Vorst's paper is the most often cited one in mathematics.

# Lanczos-type product methods (LTPMs)

Sonneveld (1989) found with the **(bi)conjugate gradient squared method (BICGS)** a way to replace the multiplication with  $\mathbf{A}^T$  by a second one with  $\mathbf{A}$ .

The  $n$ th residual polynomial is  $p_n^2$ , where  $p_n$  is the  $n$ th BICG residual polynomial, which satisfies a three-term recursion.

In each step the dimension of the Krylov space and the search space increases by 2. Convergence is nearly twice as fast, but often somewhat erratic.

**BICGSTAB** (Van der Vorst, 1992) includes some local optimization and smoothing.

The  $n$ th residual polynomial is  $p_n t_n$ , where

$$t_{n+1}(\zeta) = (1 - \chi_{n+1}\zeta)t_n(\zeta).$$

Van der Vorst's paper is the most often cited one in mathematics.

In BICGSTAB all zeros of  $t_n$  are real (if  $\mathbf{A}$ ,  $\mathbf{b}$  are real).  
It is better to choose two possibly complex new zeros in every other iteration:  $\rightsquigarrow$  **BICGSTAB2** (G., 1993)

Further generalizations of BICGSTAB include:

BICGSTAB( $\ell$ ) (Sleijpen/Fokkema, 1993; Sleijpen/VdV/F, 1994)  
GPBI-CG (Zhang, 1997), etc.

**These LTPMs are often the most efficient solvers.**

They do not require  $\mathbf{A}^T$ , and they are typically about twice as fast as BICG.

The memory needed does not increase with the iteration index  $n$  (unlike in GMRES).

In BICGSTAB all zeros of  $t_n$  are real (if  $\mathbf{A}$ ,  $\mathbf{b}$  are real).  
It is better to choose two possibly complex new zeros in every other iteration:  $\rightsquigarrow$  **BICGSTAB2** (G., 1993)

Further generalizations of BICGSTAB include:

**BICGSTAB( $\ell$ )** (Sleijpen/Fokkema, 1993; Sleijpen/VdV/F, 1994)  
**GPBI-CG** (Zhang, 1997), etc.

**These LTPMs are often the most efficient solvers.**

They do not require  $\mathbf{A}^T$ , and they are typically about twice as fast as BICG.

The memory needed does not increase with the iteration index  $n$  (unlike in GMRES).

In BICGSTAB all zeros of  $t_n$  are real (if  $\mathbf{A}$ ,  $\mathbf{b}$  are real).  
It is better to choose two possibly complex new zeros in every other iteration:  $\rightsquigarrow$  **BICGSTAB2** (G., 1993)

Further generalizations of BICGSTAB include:

**BICGSTAB( $\ell$ )** (Sleijpen/Fokkema, 1993; Sleijpen/VdV/F, 1994)  
**GPBI-CG** (Zhang, 1997), etc.

**These LTPMs are often the most efficient solvers.**

They do not require  $\mathbf{A}^T$ , and they are typically about twice as fast as BICG.

The memory needed does not increase with the iteration index  $n$  (unlike in GMRES).

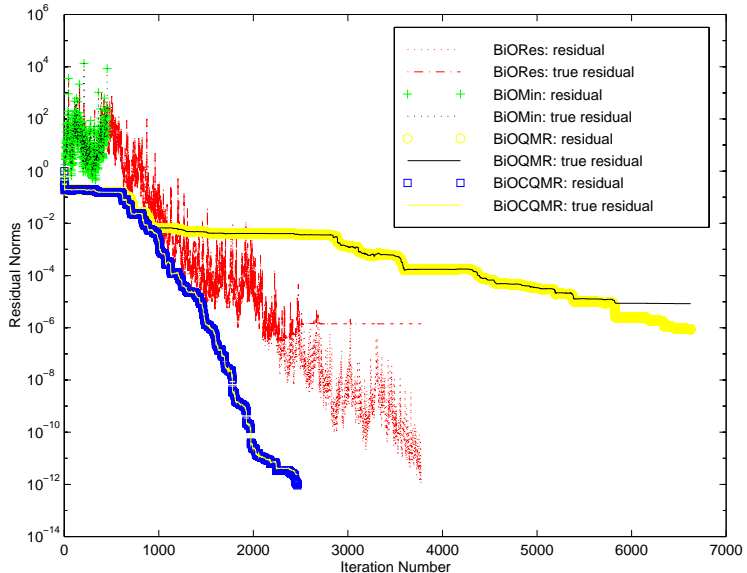
# Solving the system in coordinate space

There is yet another class of Krylov space solvers, which includes well-known methods like **MINRES**, **SYMMLQ**, **GMRES**, and **QMR**. It was pioneered by **Paige and Saunders (1975)**.

We successively construct a basis of the Krylov space by combining the extension of the space with Gram-Schmidt orthogonalization (or biorthogonalization), and at each iteration we solve  $\mathbf{Ax} = \mathbf{b}$  approximately in coordinate space.

- **symmetric Lanczos process**  $\rightsquigarrow$  **MINRES**, **SYMMLQ** (Paige/Saunders, 1975)
- **nonsymmetric Lanczos process**  $\rightsquigarrow$  **QMR** (Freund/Nachtigal, 1991)
- **Arnoldi process**  $\rightsquigarrow$  **GMRES** (Saad/Schultz, 1985)

# Breakdowns and roundoff



- Krylov (sub)space solvers are very effective tools.
- There are a large number of methods of this class.
- Preconditioning is most important.
- Breakdowns and roundoff may be a problem.

Thanks for listening and come to ...

## 6th International Congress on Industrial and Applied Mathematics




[www.iciam07.ch](http://www.iciam07.ch)



Zurich, Switzerland  
16 - 20 July 2007



-  K. Röllin (2005), Parallel iterative solvers in computational electronics, PhD thesis, Diss. No. 15859, ETH Zurich, Zurich, Switzerland.