IDR(s)

A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations

Harrachov 2007

Peter Sonneveld en Martin van Gijzen

August 24, 2007

TU Delft

Delft University of Technology

Outline

- Introduction
- The Induced Dimension Reduction Theorem
- The IDR(s) algorithm
- Numerical experiments
- Conclusions

TU Delft

Product Bi-CG methods

Bi-CG solves nonsymmetric linear systems using (short) CG recursions but needs extra matvec with $\boldsymbol{A}^H$.

Idea of Sonneveld: use 'wasted' matvec in a more useful way.
Result: transpose-free methods:

- CGS (Sonneveld, 1989)
- Bi-CGSTAB (Van der Vorst, 1992)
- BiCGSTAB2 (Gutknecht, 1993)
- TFQMR (Freund, 1993)
- BiCGstab(ℓ) (Sleijpen and Fokkema, 1994)
- Many other variants

Historical remarks

Sonneveld first developed IDR (1980).

Analysis showed that IDR was Bi-CG combined with linear minimal residual steps.

The fact that IDR is transpose free, combined with the relation with Bi-CG led to the development of a now famous algorithm: CGS.

Later Van der Vorst proposed another famous method: Bi-CGSTAB, which is mathematicaly equivalent with IDR.

As a result of these developments, the basic IDR idea was abandoned for the Bi-CG approach. IDR is now forgotten.

The IDR idea

The IDR-idea is to generate a sequence of subspaces $\mathcal{G}_0 \cdots \mathcal{G}_j$ with the following operations:

- Intersect $\mathcal{G}_{j-1}$ with fixed subspace $\mathcal{S}$,
- Compute $\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$.

The subspaces $\mathcal{G}_0 \cdots \mathcal{G}_j$ are nested and of shrinking dimension.

The IDR theorem

Theorem 1 (IDR) *Let $\boldsymbol{A}$ be any matrix in $\mathbb{C}^{N\times N}$, let $\boldsymbol{v}_0$ be any nonzero vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the complete Krylov space $\mathcal{K}^N(\boldsymbol{A}, \boldsymbol{v}_0)$. Let $\mathcal{S}$ denote any (proper) subspace of $\mathbb{C}^N$ such that $\mathcal{S}$ and $\mathcal{G}_0$ do not share a nontrivial invariant subspace of $\boldsymbol{A}$, and define the sequence $\mathcal{G}_j$, $j = 1, 2, \ldots$ as*

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where the ω_j's are nonzero scalars. Then

(i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$ for all $j > 0$.

(ii) $\mathcal{G}_j = \{\boldsymbol{0}\}$ for some $j \leq N$.

Making an IDR algorithm

The IDR theorem can be used to construct solution algorithms.

This is done by constructing residuals $\boldsymbol{r}_n \in \mathcal{G}_j$.

According to the IDR theorem ultimately $\boldsymbol{r}_n \in \mathcal{G}_M = \{\mathbf{0}\}$.

In order to generate residuals and corresponding solution approximations we first look at the basic recursions.

TU Delft

Krylov methods: basic recursion (1)

A Krylov-type solver produces iterates $\boldsymbol{x}_n$, for which the residuals $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$ are in the Krylov spaces

$$\mathcal{K}^n(\boldsymbol{A}, \boldsymbol{r}_0) = \boldsymbol{r}_0 \oplus \boldsymbol{A}\boldsymbol{r}_0 \oplus \boldsymbol{A}^2\boldsymbol{r}_0 \oplus \cdots \oplus \boldsymbol{A}^n\boldsymbol{r}_0 \ ,$$

The next residual $\boldsymbol{r}_{n+1}$ can be generated by

$$\boldsymbol{r}_{n+1} = -\alpha \boldsymbol{A}\boldsymbol{r}_n + \sum_{j=0}^{\widehat{j}} \beta_j \boldsymbol{r}_{n-j} \ .$$

The parameters α, β_j determine the specific method and must be such that $\boldsymbol{x}_{n+1}$ can be computed.

TUDelft

Krylov methods: basic recursion (2)

By using the difference vector

$$\Delta \boldsymbol{r}_k = \boldsymbol{r}_{k+1} - \boldsymbol{r}_k = -\boldsymbol{A}(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n),$$

an explicit way to satisfy this requirement is

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \alpha \boldsymbol{A} \boldsymbol{r}_n - \sum_{j=1}^{\widehat{j}} \gamma_j \Delta \boldsymbol{r}_{n-j} \ ,$$

which leads to the following update for the x estimate:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha \boldsymbol{r}_n - \sum_{j=1}^{\widehat{j}} \gamma_j \Delta \boldsymbol{x}_{n-j} \ ,$$

Computation of a new residual (1)

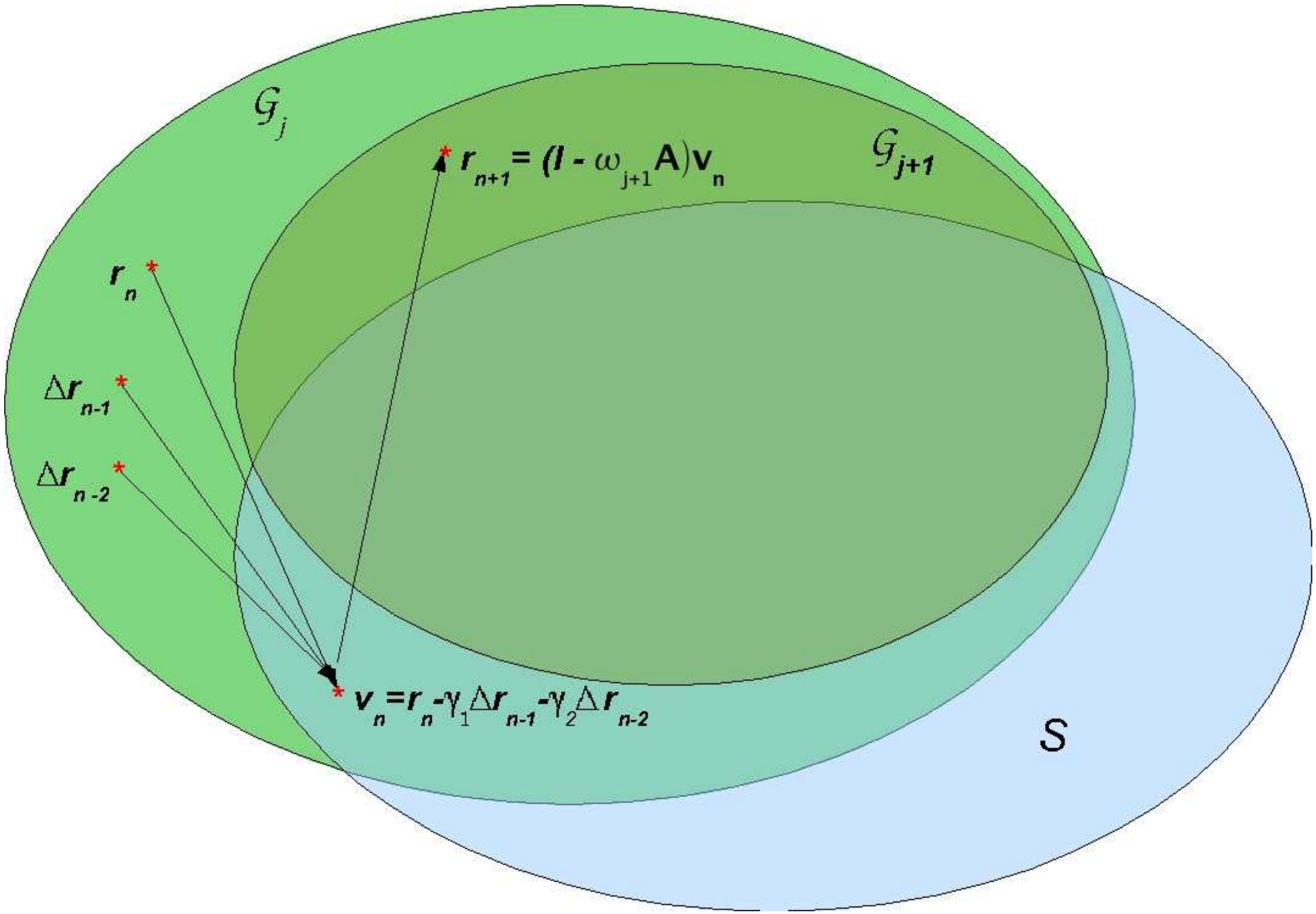
Residuals are computed that are forced to be in the subspaces $\mathcal{G}_j$, by application of the IDR-theorem.

The residual $\boldsymbol{r}_{n+1}$ is in $\mathcal{G}_{j+1}$ if

$$\boldsymbol{r}_{n+1} = (\boldsymbol{I} - \omega_{j+1}\boldsymbol{A})\boldsymbol{v} \quad \text{with } \boldsymbol{v} \in \mathcal{G}_j \cap \mathcal{S} \; .$$

The main problem is to find $\boldsymbol{v}$.

Computation of a new residual (2)



Computation of a new residual (3)

The vector $\boldsymbol{v}$ is a combination of the residuals $\boldsymbol{r}_l$ in $\mathcal{G}_j$.

$$\boldsymbol{v} = \boldsymbol{r}_n - \sum_{j=1}^{\widehat{j}} \gamma_j \Delta \boldsymbol{r}_{n-j} \ .$$

Let the space $\mathcal{S}$ be the left null space of some $N \times s$ matrix $\boldsymbol{P}$:

$$\boldsymbol{P} = (\boldsymbol{p}_1 \ \boldsymbol{p}_2 \ \ldots \ \boldsymbol{p}_s), \quad \mathcal{S} = \mathcal{N}(\boldsymbol{P}^H) \, .$$

Since $\boldsymbol{v}$ is also in $\mathcal{S} = \mathcal{N}(\boldsymbol{P}^H)$, it must satisfy

$$\boldsymbol{P}^H \boldsymbol{v} = \boldsymbol{0} \, .$$

Combining these two yields an $s \times \widehat{j}$ linear system for the coefficients γ_j that (normally) is uniquely solvable if $\widehat{j} = s$.

Computation of a new residual (4)

Hence with the residual $\boldsymbol{r}_n$, and a matrix $\Delta \boldsymbol{R}$ consisting of the last s residual differences:

$$\Delta \boldsymbol{R} = (\Delta \boldsymbol{r}_{n-1} \; \Delta \boldsymbol{r}_{n-2} \; \ldots \; \Delta \boldsymbol{r}_{n-s})$$

a suitable $\boldsymbol{v}$ can be found by

Solve $s \times s$ system $\qquad (\boldsymbol{P}^H \Delta \boldsymbol{R})\boldsymbol{c} = \boldsymbol{P}^H \boldsymbol{r}_n$

Calculate $\qquad \boldsymbol{v} = \boldsymbol{r}_n - \Delta \boldsymbol{R}\boldsymbol{c}$

Building $\mathcal{G}_{j+1}$ (1)

Assume $\boldsymbol{r}_n$ and all columns of $\Delta\boldsymbol{R}$ are in $\mathcal{G}_j$, and let $\boldsymbol{r}_{n+1}$ be calculated as

$$\boldsymbol{r}_{n+1} = \boldsymbol{v} - \omega_{j+1}\boldsymbol{A}\boldsymbol{v}$$

Then $\boldsymbol{r}_{n+1} \in \mathcal{G}_{j+1}$.
Since $\mathcal{G}_{j+1} \subset \mathcal{G}_j$ (theorem) we automatically have

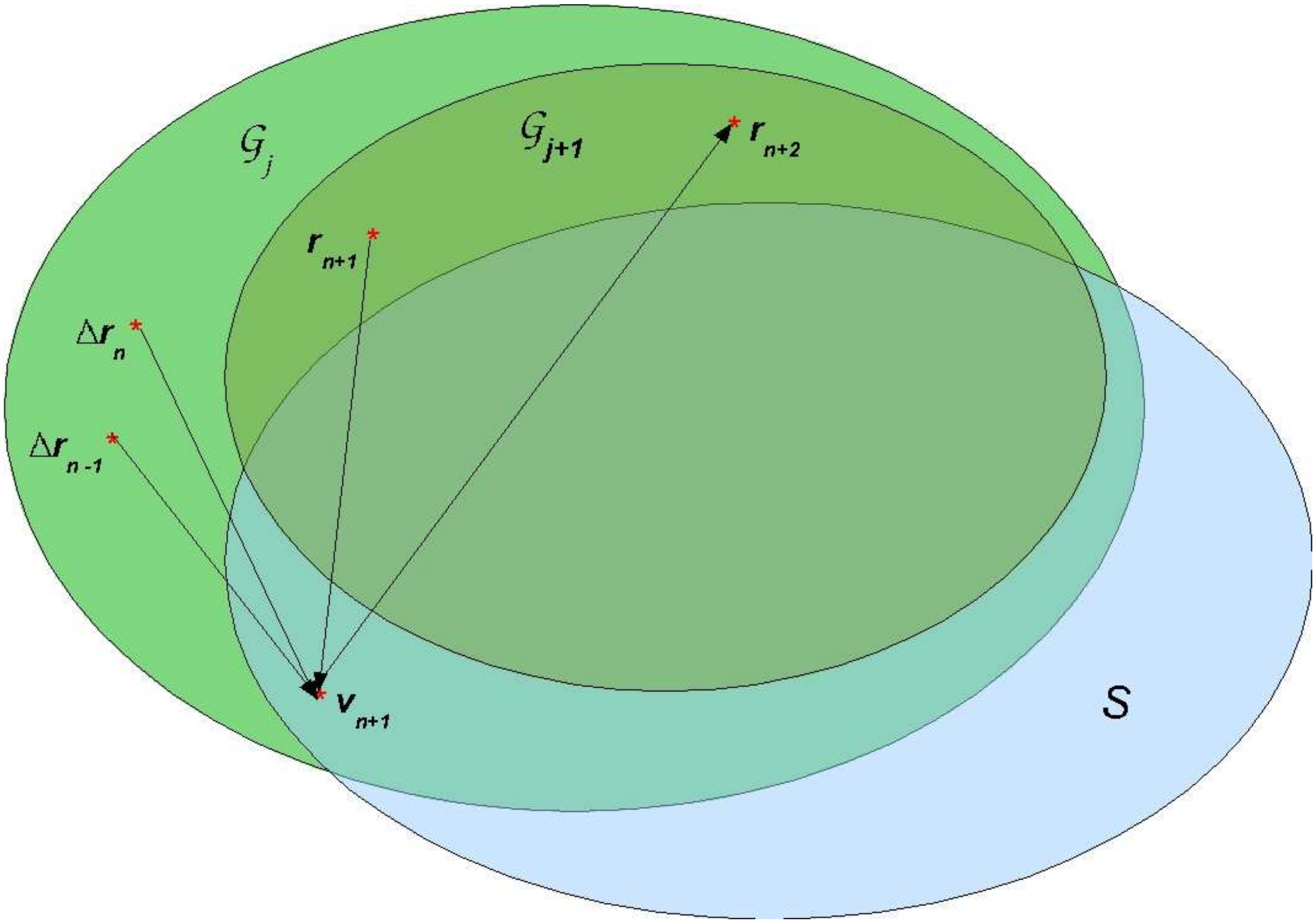
$$\boldsymbol{r}_{n+1} \in \mathcal{G}_j$$

Now the next $\Delta\boldsymbol{R}$ is made by repeating the calculations.

In this way we find $s+1$ residuals in $\mathcal{G}_{j+1}$

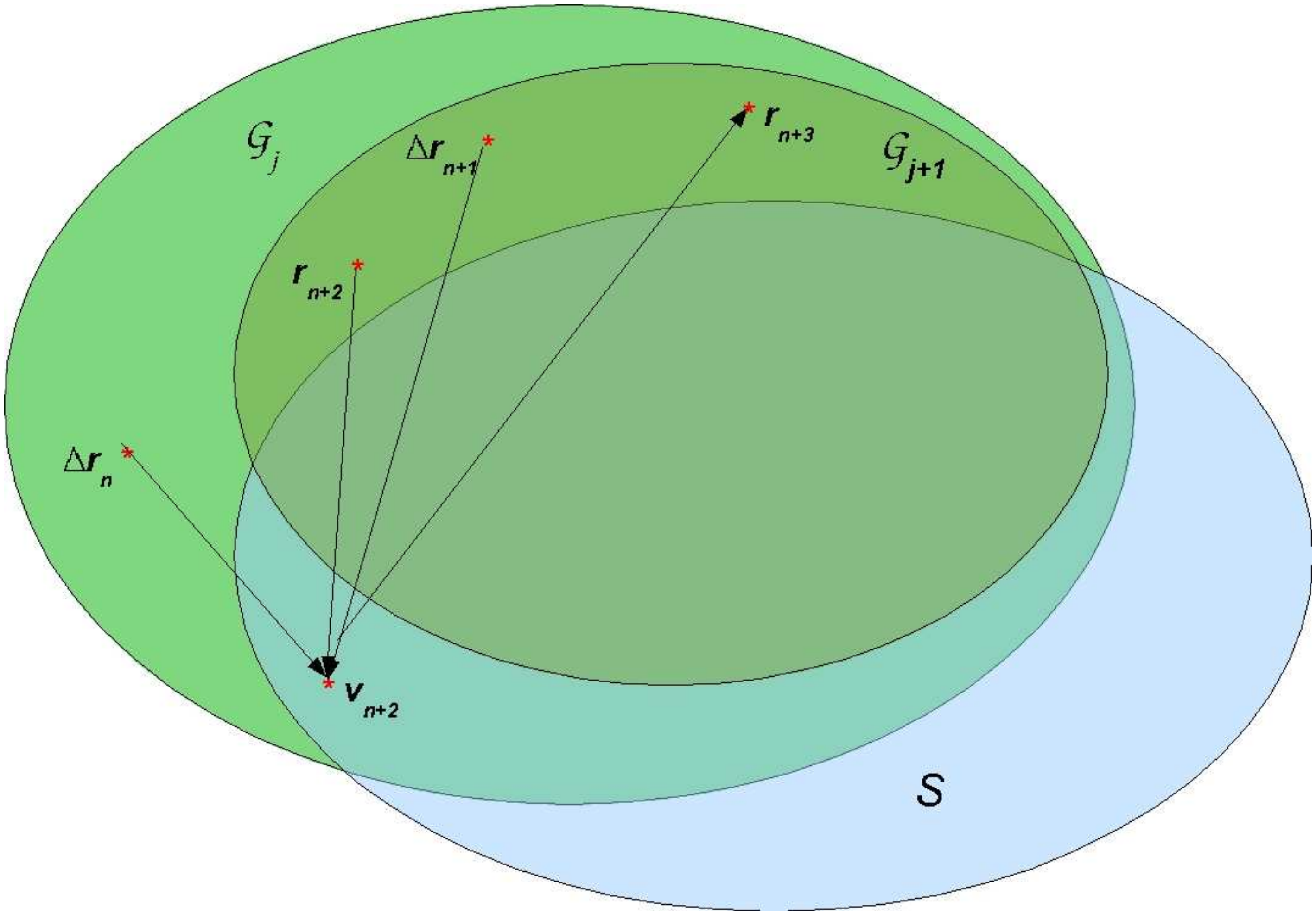
TUDelft

Building $\mathcal{G}_{j+1}$ (2)

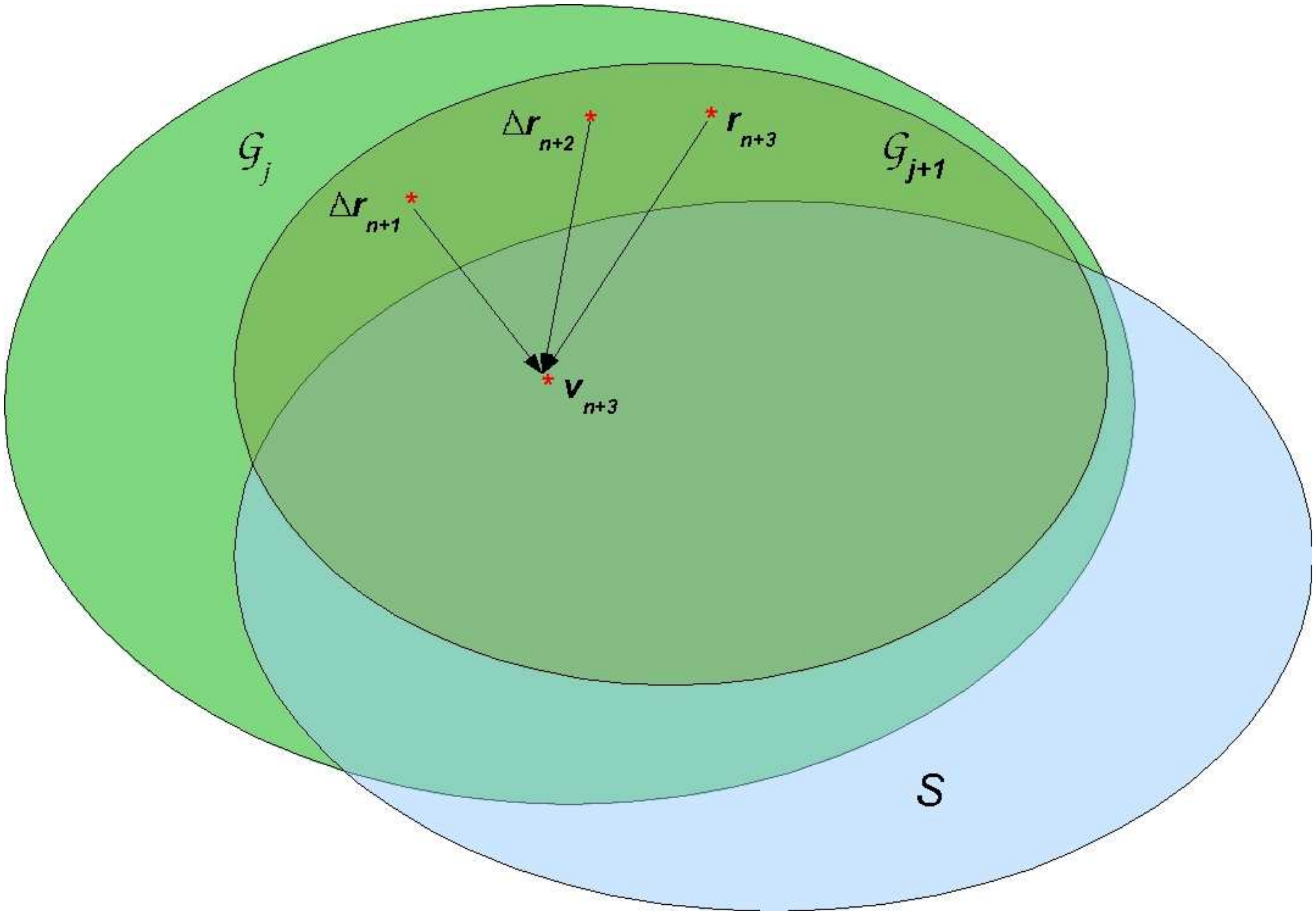


TU Delft

Building $\mathcal{G}_{j+1}$ (3)



Building $\mathcal{G}_{j+1}$ (4)



TUDelft

A few details

1. The first $s+1$ residuals, starting with $\boldsymbol{r}_0$ can be constructed by any Krylov-based iteration, such as a local minimum residual method.
2. In our actual implementation, all steps are identical. However, in calculating the first residual in $\mathcal{G}_{j+1}$, a new value ω_{j+1} may be chosen. We choose ω_{j+1} such that $\|\boldsymbol{v} - \omega_{j+1}\boldsymbol{A}\boldsymbol{v}\|$ is minimal.

TU Delft

Basic IDR(s) algorithm.

while $\|\boldsymbol{r}_n\| > TOL$ **or** $n < MAXIT$ **do**

  for $k = 0$ to s **do**

    Solve $\boldsymbol{c}$ from $\boldsymbol{P}^H d\boldsymbol{R}_n \boldsymbol{c} = \boldsymbol{P}^H \boldsymbol{r}_n$

    $\boldsymbol{v} = \boldsymbol{r}_n - d\boldsymbol{R}_n \boldsymbol{c}$; $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{v}$;

    if $k = 0$ **then**

      $\omega = (\boldsymbol{t}^H \boldsymbol{v})/(\boldsymbol{t}^H \boldsymbol{t})$;

    end if

    $d\boldsymbol{r}_n = -d\boldsymbol{R}_n \boldsymbol{c} - \omega \boldsymbol{t}$; $d\boldsymbol{x}_n = -d\boldsymbol{X}_n \boldsymbol{c} + \omega \boldsymbol{v}$;

    $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_n$; $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_n$;

    $n = n + 1$;

    $d\boldsymbol{R}_n = (d\boldsymbol{r}_{n-1} \cdots d\boldsymbol{r}_{n-s})$; $d\boldsymbol{X}_n = (d\boldsymbol{x}_{n-1} \cdots d\boldsymbol{x}_{n-s})$;

  end for

end while

Vector operations per MATVEC

Method	DOT	AXPY	Memory Requirements
IDR(1)	2	4	8
IDR(2)	$2\frac{2}{3}$	$5\frac{5}{6}$	11
IDR(4)	$4\frac{2}{5}$	$9\frac{7}{10}$	17
IDR(6)	$6\frac{2}{7}$	$13\frac{9}{14}$	23
Full GMRES	$\frac{n+1}{2}$	$\frac{n+1}{2}$	$n+2$
BiCGSTAB	2	3	7

TU Delft

Relation with other methods

Although the approach is different, IDR(s) is closely related to some Bi-CGSTAB methods:

- IDR(1) and Bi-CGSTAB yield the same residuals at the even steps.
- ML(k)BiCGSTAB (Yeung and Chen, 1999) seems closely related to IDR(s), BUT
  - IDR(s) is MUCH simpler (both conceptually and its implementation)
  - Other, more natural extensions are possible, e.g. to avoid breakdown.

Performance of IDR(s)

The IDR theorem states that

- it is possible to generate a sequence of nested subspace $\mathcal{G}_j$ of shrinking dimension,
- but does not say how fast the dimension shrinks

It can be proven that the dimension reduction is (normally) s,
So $dim(\mathcal{G}_{j+1}) = dim(\mathcal{G}_j) - s$.
IDR(s) requires at at most $N + \frac{N}{s}$ matrix-vector multiplications to compute the exact solution.

TU Delft

Numerical experiments

We will present two typical numerical examples

- A 2D Ocean Circulation Problem
- A 3D Helmholtz Problem

A 2D Ocean Circulation Problem

We compare IDR(s) with Full GMRES, restarted GMRES and Bi-CGSTAB.

This ocean example is representative for a wide class of CFD problems.

We will compare:

- Rate of convergence
- Stagnation level (of the true residual norm)

TU Delft

Stommel's model for ocean circulation

Balance between bottom friction, wind stress and Coriolis force.

$$-r\,\Delta\psi - \beta\,\frac{\partial\psi}{\partial x} = (\nabla \times \mathbf{F})_z$$

plus circulation condition around islands k

$$\oint_{\Gamma_k} r\,\frac{\partial\psi}{\partial n}\,ds = -\oint_{\Gamma_k} \mathbf{F}\cdot\mathbf{s}\,ds.$$

- ψ: streamfunction
- r: bottom friction parameter
- β: Coriolis parameter
- $\mathbf{F}$: Wind stress

TUDelft

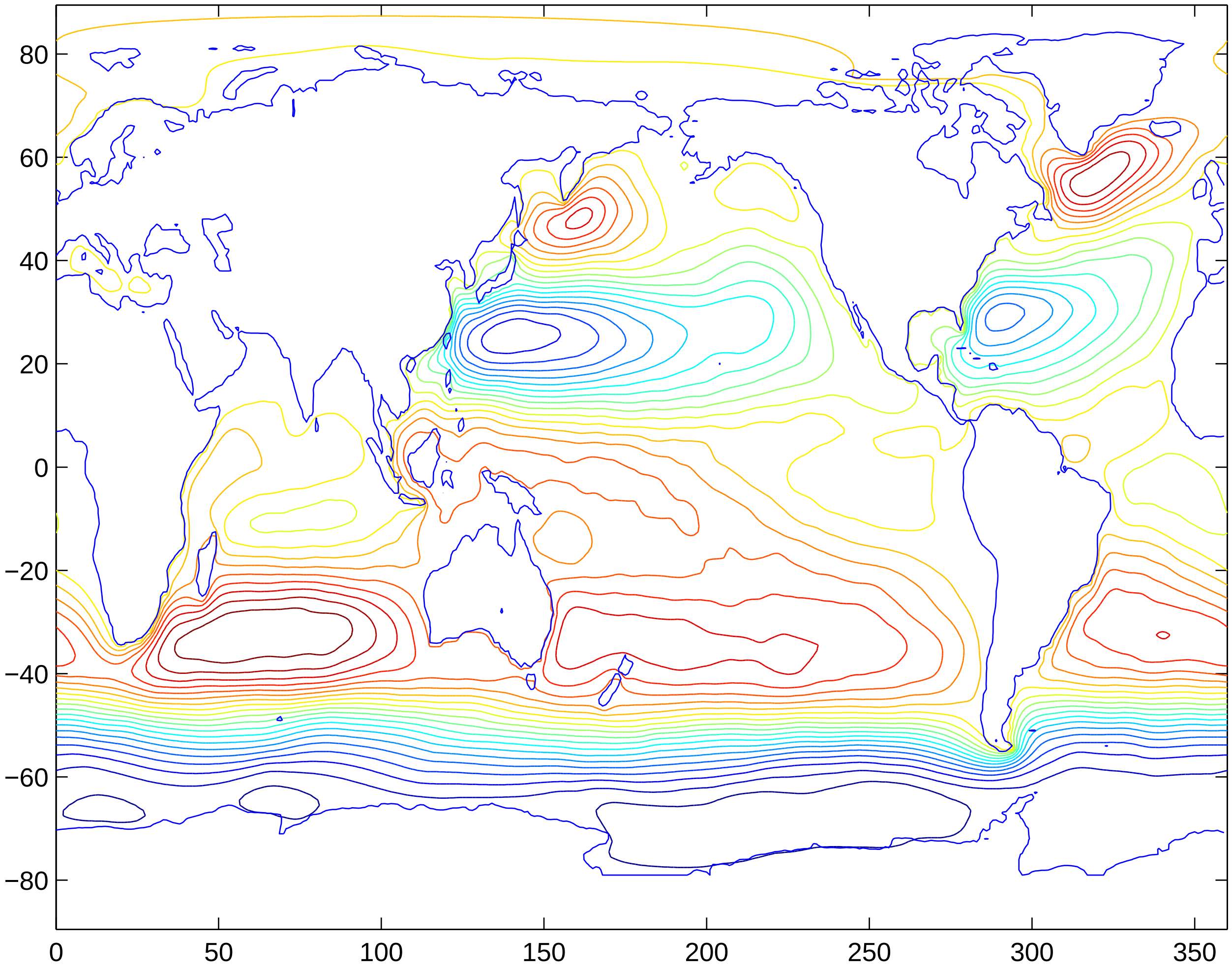
Discretization of the ocean problem

- Discretization with linear finite elements
- Results in nonsymmetric system of 42248
- Eigenvalues are (almost) real

Solution parameters:

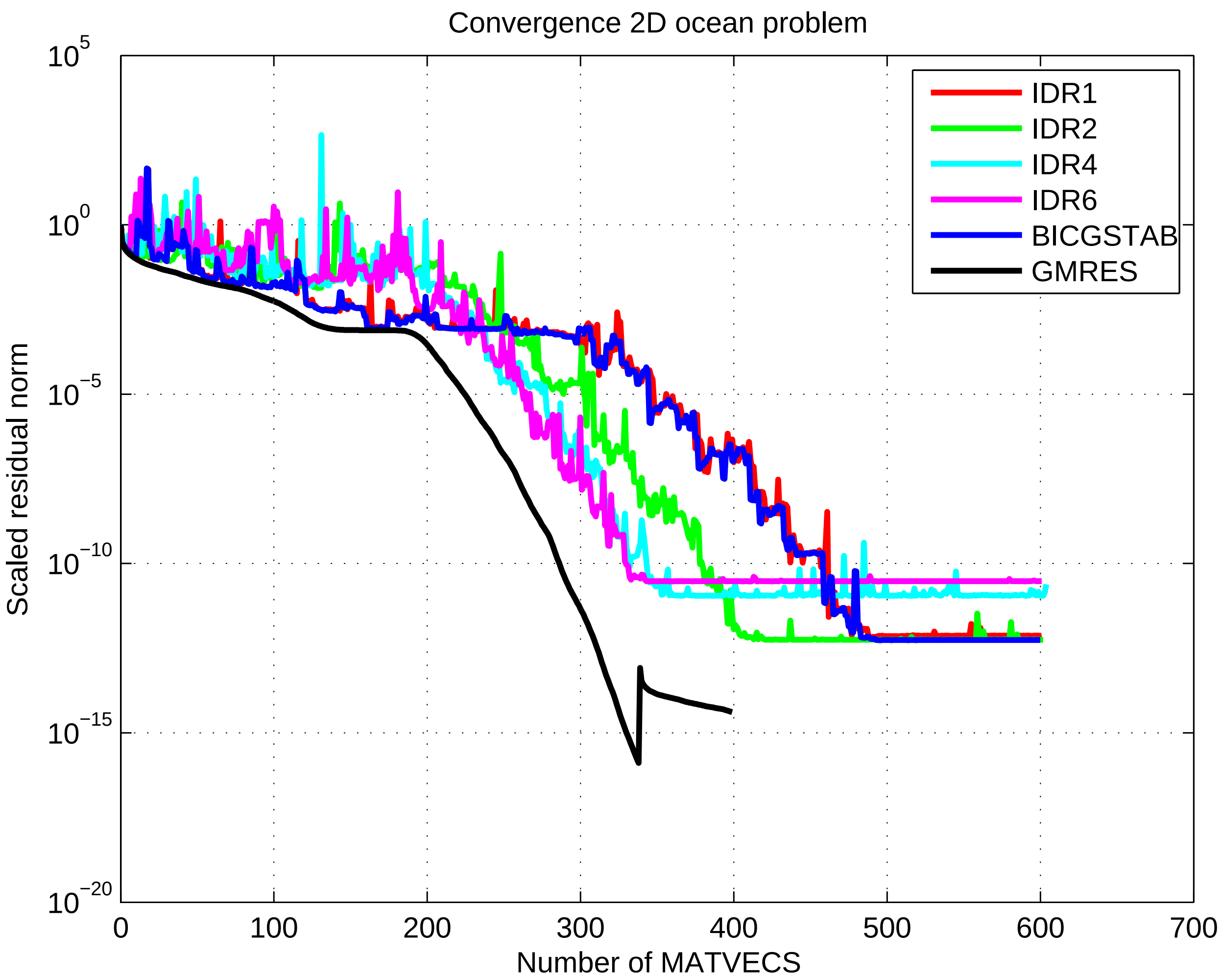
- ILU(0) preconditioning
- $\boldsymbol{P}$: $s-1$ random vectors plus $\boldsymbol{r}_0$ (for comparison with Bi-CGSTAB)

Solution of the ocean problem



TU Delft

Convergence for the ocean problem



TU Delft

Some observations

- Required number of MATVECS decreases if s is increased. IDR(4) and IDR(6) are close to the optimal convergence curve of full GMRES.
- Convergence curves of IDR(1) and Bi-CGSTAB coincide.
- Stagnation levels of IDR(s) comparable with Bi-CGSTAB.

Required number of MATVECS

Method	Number of MATVECS	Vectors
Full GMRES	265	268
GMRES(20)	> 10000	23
GMRES(50)	4671	53
Bi-CGSTAB	411	7
IDR(1)	420	8
IDR(2)	339	11
IDR(4)	315	17
IDR(6)	307	23

Tolerance: $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n\| < 10^{-8}\|\boldsymbol{b}\|$

TU Delft

A 3D Helmholtz Problem

Example models sound propagation in a room of $4 \times 4 \times 4m^3$.

A harmonic sound source gives acoustic pressure field

$$\boldsymbol{p}(\boldsymbol{x}, t) = \widehat{p}(\boldsymbol{x})e^{2\pi i f t}.$$

The pressure function $\widehat{p}$ can be determined from

$$\frac{-(2\pi f)^2}{c^2}\widehat{p} - \Delta\widehat{p} = \delta(\boldsymbol{x} - \boldsymbol{x}_s) \quad \text{in } \Omega.$$

in which

- c: the sound speed (340 m/s)
- $\delta(\boldsymbol{x} - \boldsymbol{x}_s)$: the harmonic point source, in the center of the room.

Boundary conditions

Five of the walls are reflecting, modeled by

$$\frac{\partial \widehat{p}}{\partial n} = 0 \; ,$$

and the remaining wall is sound absorbing,

$$\frac{\partial \widehat{p}}{\partial n} = -\frac{2\pi i f}{c}\widehat{p} \; \text{on } \Gamma_3 .$$

Discretization

Discretization with FEM yields linear system

$$[-(2\pi f)^2 \boldsymbol{M} + 2\pi i f \boldsymbol{C} + \boldsymbol{K}]\boldsymbol{p} = \boldsymbol{b}$$

- Frequency $f = 100Hz$.
- System matrix complex, symmetric (but not Hermitian) and indefinite: difficult for iterative methods
- gridsize $h = 8\ cm$: 132651 equations

Solution parameters:

- ILU(0) preconditioning
- $\boldsymbol{P}$: Initial residual plus $s-1$ real random vectors
- Only comparison with BiCGstab(ℓ)

Results Helmholtz Problem

Method	Number of MATVECS	Elapsed time [s]
IDR(1)	1500	3322
IDR(2)	598	1329
IDR(4)	353	783
IDR(6)	310	698
BiCGstab(1)	1828	3712
BiCGstab(2)	1008	2045
BiCGstab(4)	656	1362
BiCGstab(8)	608	1337

Tolerance: $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n\| < 10^{-8}\|\boldsymbol{b}\|$

Conclusions

- The IDR-theorem offers a new approach for the development of iterative solution algorithms
- The IDR(s) algorithm presented here is quite promising and seems to outperform state-of-the-art Bi-CG-type methods for important classes of problems.

More information:

http://ta.twi.tudelft.nl/nw/users/gijzen/software.html

- Report: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, submitted
- Matlab code, (includes preconditioning and deflation)