A Set of Fortran 90 and Python Routines for Solving Linear Equations with IDR(s)

R. Astudillo,* M. Baumann,@* M. B. van Gijzen*

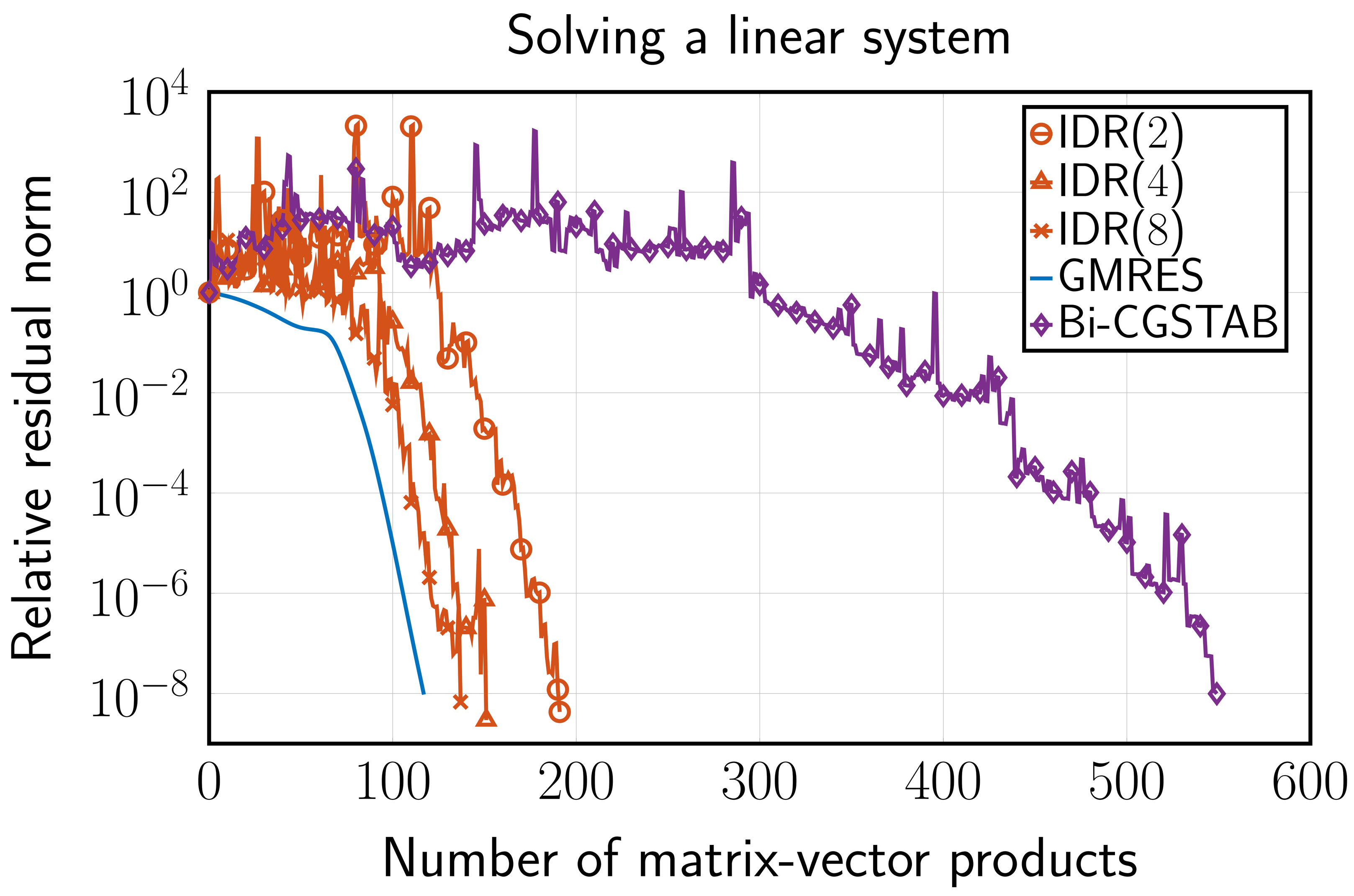
@M.M.Baumann@tudelft.nl, *Delft Institute of Applied Mathematics, TU Delft

Induced Dimension Reduction (IDR) method

IDR(s) is a Krylov subspace method originally proposed to solve system of linear equations,

$$A\mathbf{x} = \mathbf{b},$$

where the coefficient matrix A is large, sparse, and non-symmetric. IDR(s) is a short-recurrence method which has obtained attention for its rapid convergence and computational efficiency.



[1] P. Sonneveld and M. B. van Gijzen. *IDR(s): A Family of Simple and Fast Algorithms for Solving Large Nonsymmetric Systems of Linear Equations*, SIAM J. Sci. Comput., 31(2), 1035–1062, 2008

IDR(s) for linear matrix equations

Generalization of the IDR theorem. Let $\mathcal{A}$ be a linear operator over a finite dimensional subspace $\mathcal{D}$ and $\mathcal{I}$ be the identity operator over the same subspace. Let $\mathcal{S}$ be any (proper) subspace of $\mathcal{D}$. Define $\mathcal{G}_0 \equiv \mathcal{D}$. If $\mathcal{S}$ and $\mathcal{G}_0$ do not share a nontrivial invariant subspace of the operator $\mathcal{A}$, then the sequence of subspaces $\mathcal{G}_j$, defined as

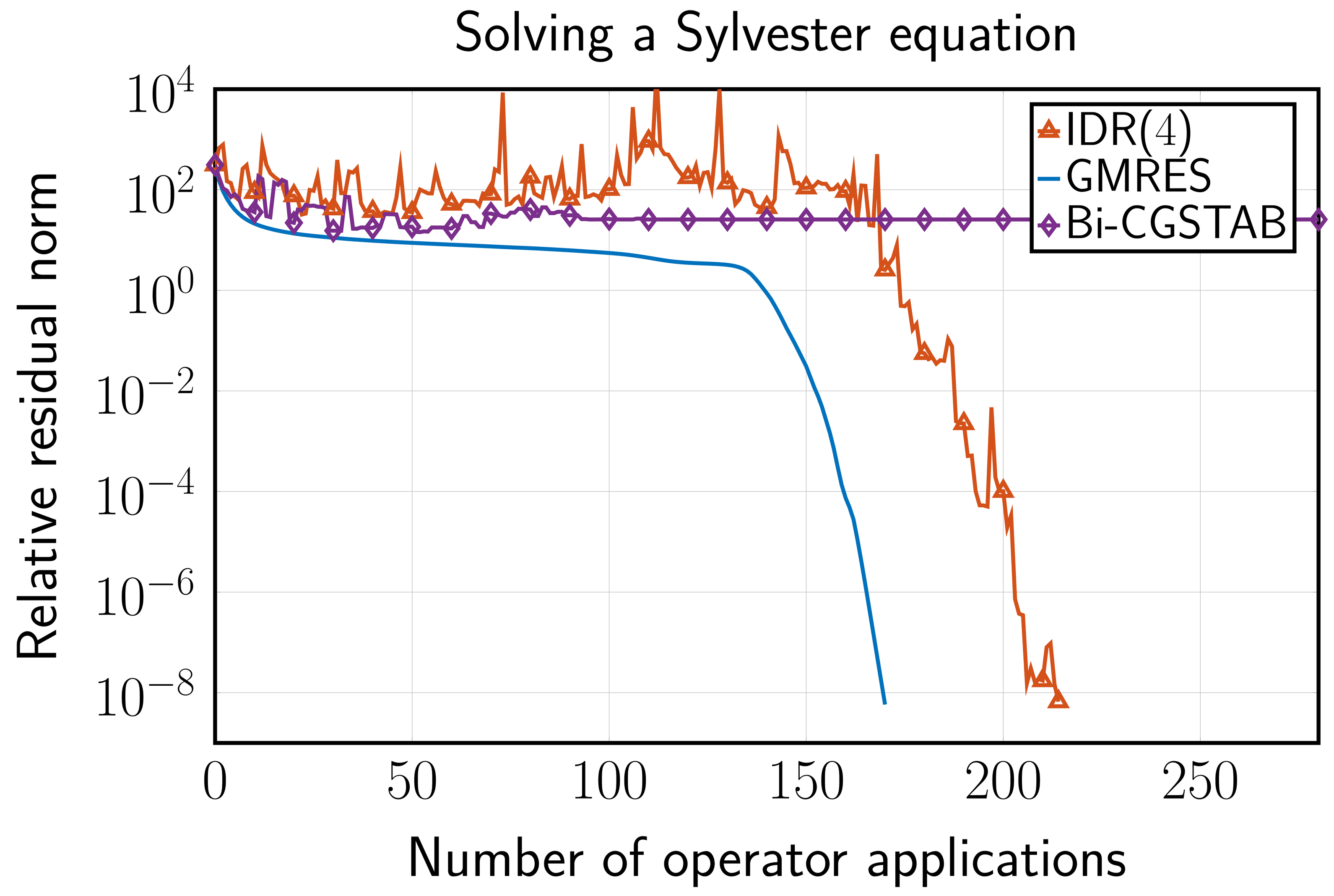
$$\mathcal{G}_j \equiv (\mathcal{I} - \omega_j \mathcal{A})(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2 \ldots, \quad \omega_j \neq 0,$$

has the following properties for $j \geq 0$:
(1) $\mathcal{G}_{j+1} \subset \mathcal{G}_j$, and (2) $\dim(\mathcal{G}_{j+1}) < \dim(\mathcal{G}_j)$ unless $\mathcal{G}_j = \{\mathbf{0}\}$.

Using this generalization, we develop an IDR-algorithm for solving linear matrix equations of the form,

$$\sum_{j=1}^{k} A_j \mathbf{X} B_j^{\mathsf{T}} = C.$$

[2] R. Astudillo and M. B. van Gijzen. *Induced Dimension Reduction method for solving linear matrix equations*, Delft University of Technology, TR-05, 2015



IDR(s) for shifted linear systems

For the efficient solution of shifted linear systems,

$$(A - \sigma_k I)\mathbf{x}_k = \mathbf{b}, \quad k = 1, 2, ...,$$

two IDR variants have recently been developed:

- Multi-shift `QMRIDR(s)` relies on a generalized Hessenberg decomposition, cf. [4],
- `MSIDR(s)` generates collinear residuals such that $\mathbf{r}_j^{(\sigma_k)} \in \mathcal{G}_j, \forall k$, cf. [3].

The latter can be used as a preconditioner in a nested algorithm.

[3] M. Baumann and M. B. van Gijzen. *Nested Krylov methods for shifted linear systems*, SIAM J. Sci. Comput. Copper Mountain Special Issue, 2014 [in press]

[4] M. B. van Gijzen, G. L. G. Sleijpen, and J.-P. M. Zemke. *Flexible and multi-shift induced dimension reduction algorithms for solving large sparse linear systems*, Numer. Linear Algebra Appl. 22(1), 2015, 1-25

Software features

IDR implementation developed at TU Delft:

- Standalone implementation in Fortran 90 and Python
- Flexible user interface via `types`
- Advanced features, e.g. subspace recycling, Ritz values
- Solving matrix equations and multiple right-hand sides:

```
>> A  = numpy.random.rand(n,n)
>> S  = numpy.diag(sigma)
>> B  = numpy.random.rand(n,n)
>> op = lambda X: A*X - X*S.T
>> s  = 4
>> X  = idrs(op,B,s)
```