Randomized Implicitly Restarted Arnoldi algorithm for the unsymmetric eigenvalue problem

Jean-Guillaume de Damas
under the supervision of Laura Grigori

INRIA Paris

June 17, 2024

Extreme-scale
Mathematically-based
Computational
Chemistry

Krylov iterative methods

One of the most time consuming step in a simulation : solving of a large and sparse eigenvalue problem:

$$Ax = \lambda x,$$

$$A \in \mathbb{R}^{n\times n}, \quad x \in \mathbb{R}^{n}, \quad \lambda \in \mathbb{C}, \quad n \in \mathbb{N} \quad \text{large, say } n \approx 10^6$$

Krylov iterative methods

One of the most time consuming step in a simulation : solving of a large and sparse eigenvalue problem:

$$Ax = \lambda x,$$
$$A \in \mathbb{R}^{n\times n}, \quad x \in \mathbb{R}^n, \quad \lambda \in \mathbb{C}, \quad n \in \mathbb{N} \quad \text{large, say } n \approx 10^6$$

Tool : Krylov subspace methods. These are iterative methods that benefit from system sparsity by using matrix-vector products. They build upon a starting vector v_1 usually taken at random and the ill-conditioned subspace

$$\mathcal{K}_k(A, v_1) \equiv \text{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\} \tag{1}$$

Krylov iterative methods

Tool : Krylov subspace methods. These are iterative methods that benefit from system sparsity by using matrix-vector products. They build upon a starting vector v_1 usually taken at random and the ill-conditioned subspace

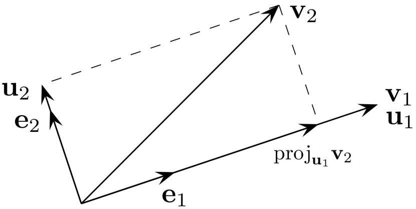
$$\mathcal{K}_k(A, v_1) \equiv \text{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\}$$

Krylov iterative methods

Tool : Krylov subspace methods. These are iterative methods that benefit from system sparsity by using matrix-vector products. They build upon a starting vector v_1 usually taken at random and the ill-conditioned subspace

$$\mathcal{K}_k(A, v_1) \equiv \mathrm{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\}$$

They rely on orthonormalization process, most famous orthonormalization process : Gram-Schmidt process. Costs $O(nk^2)$ where k is the number of iterations.



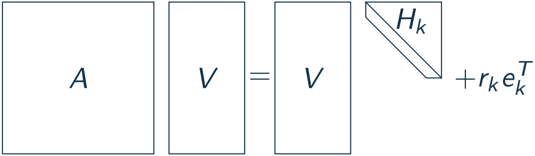
The Arnoldi factorization

When Gram-Schmidt is applied to the set $\{v_1, Av_1, \ldots, A^{k-1}v_1\}$, this gives the Arnoldi factorization:

Definition (Arnoldi factorization)

$$AV = VH_k + r_k e_k^T \tag{2}$$

with orthogonal $V \in \mathbb{R}^{n \times k}$ and upper Hessenberg $H_k \in \mathbb{R}^{k \times k}$.



Restarting an Arnoldi factorization

Extract eigenpairs using Rayleigh-Ritz procedure: If $AV = VH_k + r_k e_k^T$, then

$$\text{Compute exact pairs} \quad H_k y = \tilde{\lambda} y \tag{3}$$

$$\text{Obtain Ritz pairs} \quad A\tilde{u} = \tilde{\lambda}\tilde{u} + r_k e_k^T y \tag{4}$$

with $\tilde{u} = Vy$ giving $A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K}_k$ (Galerkin condition). $(\tilde{\lambda}, \tilde{u})$ is a Ritz pair.

Restarting an Arnoldi factorization

Extract eigenpairs using Rayleigh-Ritz procedure: If $AV = VH_k + r_k e_k^T$, then

$$\text{Compute exact pairs} \quad H_k y = \tilde{\lambda} y \tag{3}$$

$$\text{Obtain Ritz pairs} \quad A\tilde{u} = \tilde{\lambda}\tilde{u} + r_k e_k^T y \tag{4}$$

with $\tilde{u} = Vy$ giving $A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K}_k$ (Galerkin condition). $(\tilde{\lambda}, \tilde{u})$ is a Ritz pair.

Issues : Cost of $O(nk^2)$ for the orthogonalization process when k increases up to $k \approx 10^2 - 10^3$. Storage of k vectors of size n for V.

Restarting an Arnoldi factorization

Extract eigenpairs using Rayleigh-Ritz procedure: If $AV = VH_k + r_k e_k^T$, then

$$\text{Compute exact pairs} \quad H_k y = \tilde{\lambda} y \tag{3}$$

$$\text{Obtain Ritz pairs} \quad A\tilde{u} = \tilde{\lambda}\tilde{u} + r_k e_k^T y \tag{4}$$

with $\tilde{u} = Vy$ giving $A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K}_k$ (Galerkin condition). $(\tilde{\lambda}, \tilde{u})$ is a Ritz pair.

Issues : Cost of $O(nk^2)$ for the orthogonalization process when k increases up to $k \approx 10^2 - 10^3$. Storage of k vectors of size n for V.

Solution : one can restart the Arnoldi factorization. Fix a dimension k, compute new v_1^+ from the current factorization. How to compute a relevant v_1^+ ?

Implicitly Restarted Arnoldi method

Idea from Sorensen in [Sor92]: apply polynomial filtering

$$v_1^+ = \psi_p(A) v_1 \tag{5}$$

Discard unwanted directions with $\psi_p(A) = \prod_{i=1}^{p}(A - \tilde{\lambda}_i I)$.

Implicitly Restarted Arnoldi method

Idea from Sorensen in [Sor92]: apply polynomial filtering

$$v_1^+ = \psi_p(A)v_1 \tag{5}$$

Discard unwanted directions with $\psi_p(A) = \prod_{i=1}^{p}(A - \tilde{\lambda}_i I)$.
Apply the shifted-QR algorithm to H_{k+p}:

$$(H_{k+p} - \tilde{\lambda}_i I) = QR \text{ then } H_{k+p}^+ = Q^T H_{k+p} Q \tag{6}$$

Implicitly Restarted Arnoldi method

Idea from Sorensen in [Sor92]: apply polynomial filtering

$$v_1^+ = \psi_p(A) v_1 \tag{5}$$

Discard unwanted directions with $\psi_p(A) = \prod_{i=1}^{p}(A - \tilde{\lambda}_i I)$.
Apply the shifted-QR algorithm to H_{k+p}:

$$(H_{k+p} - \tilde{\lambda}_i I) = QR \text{ then } H_{k+p}^+ = Q^T H_{k+p} Q \tag{6}$$

Brings v_1^+ closer to the span of a partial Schur factorization of A

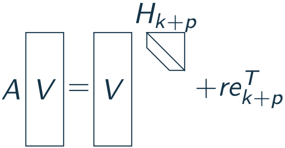
$$AZ = ZT, \text{ orthogonal } Z \in \mathbb{R}^{n\times k} \tag{7}$$

Given $AV = VH_k + r_k e_k^T$, it holds [Sor92, Theorem 2.8]:

$$r_k = 0 \iff v_1 = Zy \tag{8}$$

Scheme of IRA

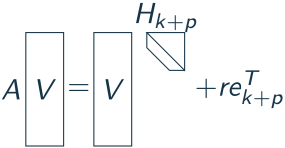
From a length k: $AV = VH_k + r_k e_k^T$, extend:



$A\,V = V\,H_{k+p} + re_{k+p}^T$

Scheme of IRA

From a length k: $AV = VH_k + r_k e_k^T$, extend:

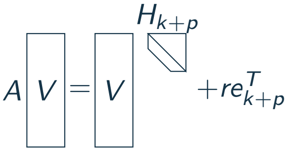


$$A \boxed{V} = \boxed{V} H_{k+p} + r e_{k+p}^T$$

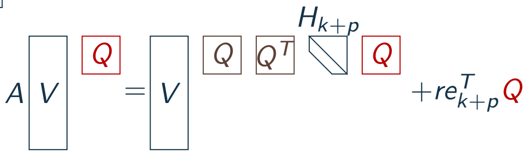
Apply p shifted QR steps to obtain $H_{k+p}^{+} = QR = Q^T H_{k+p} Q$

Scheme of IRA

From a length k: $AV = VH_k + r_k e_k^T$, extend:

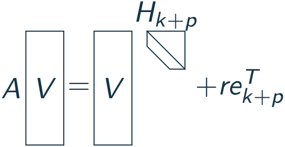


Apply p shifted QR steps to obtain
$H_{k+p}^+ = QR = Q^T H_{k+p} Q$

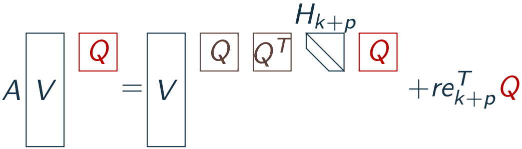


Scheme of IRA

From a length k: $AV = VH_k + r_k e_k^T$, extend:



Apply p shifted QR steps to obtain
$H_{k+p}^+ = QR = Q^T H_{k+p} Q$



$e_{k+p}^T Q = [0 \ldots 0 \; \eta_k \ldots \eta_{k+p}]$: equate first k columns to continue with
$A(VQ)[e_1, \ldots, e_k] = (VQ)[e_1, \ldots, e_k]\tilde{H}_k + \tilde{r}_k e_k^T$.

Sketching

IRA requires a sequence of contractation and expansion steps. Expansion is expensive due to orthogonalization.

Sketching

IRA requires a sequence of contractation and expansion steps. Expansion is expensive due to orthogonalization.

We turn to sketching:

Definition (ε-embedding from [Woo14])

We say that $\Omega \in \mathbb{R}^{d \times n}$ is an ε-embedding for some k-dimensional subspace $\mathcal{K}_k \subset \mathbb{R}^n$ if

$$\forall\, x, y \in \mathcal{K}_k,\ |\langle \Omega x, \Omega y \rangle - \langle x, y \rangle| \leq \varepsilon \|x\|\,\|y\|\,. \tag{9}$$

$$\text{i.e. } \forall\, x \in \mathcal{K}_k,\ (1-\varepsilon)\|x\|^2 \leq \|\Omega x\|^2 \leq (1+\varepsilon)\|x\|^2. \tag{10}$$

Usually $d = \frac{k}{\varepsilon^2} \approx 4k$ s.t. $k \leq d << n$: computational cost gained.

Orthogonalization process

From **INPUT** $W \in \mathbb{R}^{n \times k}$, general algorithm to **OUTPUT** $Q \in \mathbb{R}^{n \times k}$ Ω-orthonormal: $(\Omega Q)^T(\Omega Q) = I_k$, $\text{span}\{Q\} = \text{span}\{W\}$ and $S = \Omega Q$:

```
for $j = 1, \dots, k$ do
    Initialize $q_j = W(:, j) \in \mathbb{R}^n$
    Sketch $s_j = \Omega q_j \in \mathbb{R}^d$
    Solve with a given method $R_j = \arg\min_{y \in \mathbb{R}^{j-1}} \|S_{j-1} y - s_j\| \in \mathbb{R}^{j-1}$
    Update $q_j = q_j - Q_{j-1} R_j$
    Re-sketch $s_j = \Omega q_j$
    Store $q_j / \|s_j\|$ and $s_j / \|s_j\|$
end for
```

Orthogonalization process

Solve with a given method

$$R_j = \arg\min_{y \in \mathbb{R}^{j-1}} \|S_{j-1} y - s_j\| \in \mathbb{R}^{j-1}$$

Case 1 : $R_j = S_{j-1} \setminus s_j$. RGS from [BG22].

Orthogonalization process

Solve with a given method

$$R_j = \arg\min_{y \in \mathbb{R}^{j-1}} \|S_{j-1}y - s_j\| \in \mathbb{R}^{j-1}$$

Case 1 : $R_j = S_{j-1} \setminus s_j$. RGS from [BG22].
Case 2 : We propose $R_j = S_{j-1}^T s_j$. Called rCGS.

Orthogonalization process

Solve with a given method

$$R_j = \arg \min_{y \in \mathbb{R}^{j-1}} \|S_{j-1} y - s_j\| \in \mathbb{R}^{j-1}$$

Case 1 : $R_j = S_{j-1} \setminus s_j$. RGS from [BG22].
Case 2 : We propose $R_j = S_{j-1}^T s_j$. Called rCGS.
Often requires reorthogonalization:

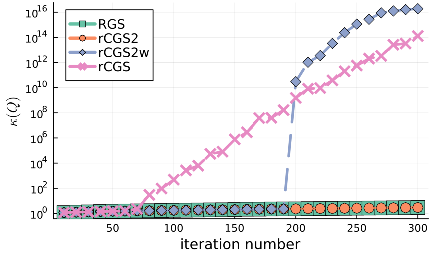
$$R_j = S_{j-1}^T s_j \rightarrow q_j = q_j - V_j R_j \rightarrow s_j = \Omega q_j \quad \text{2 times} \tag{11}$$

Can be accelerated as noted in [BG22]:

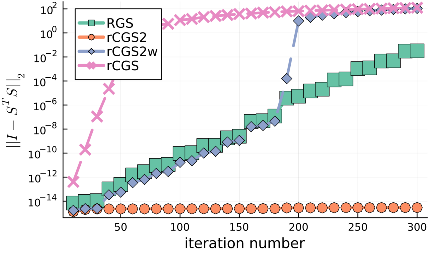
$$R_j = S_{j-1}^T s_j \rightarrow s_j = s_j - S_{j-1} R_j \quad \text{2 times} \tag{12}$$

A mix of the two to stabilize it is possible. Called rCGS2.

Orthogonalization process



$\kappa(Q)$ for $Q \in \mathbb{R}^{n\times(\#\text{ of iteration})}$



$\left\|I - S^TS\right\|$ over the iterations

$W \in \mathbb{R}^{10^5\times 300}$ numerically singular. rCGS2 costlier but more stable.

randomized Arnoldi

RGS or variant applied to $\{v_1, Av_1, \ldots, A^{k-1}v_1\}$ gives

Definition (randomized Arnoldi factorization)

$$AV = VH_k + r_k e_k^T \tag{13}$$

with $V \in \mathbb{R}^{n\times k}$ Ω-orthogonal: $(\Omega V)^T(\Omega V) = I_k$ and $\kappa(V) \leq \frac{\sqrt{1+\varepsilon}}{\sqrt{1-\varepsilon}} = \sqrt{3}$

randomized Arnoldi

RGS or variant applied to $\{v_1, Av_1, \ldots, A^{k-1}v_1\}$ gives

Definition (randomized Arnoldi factorization)

$$AV = VH_k + r_k e_k^T \tag{13}$$

with $V \in \mathbb{R}^{n \times k}$ Ω-orthogonal: $(\Omega V)^T(\Omega V) = I_k$ and $\kappa(V) \leq \frac{\sqrt{1+\varepsilon}}{\sqrt{1-\varepsilon}} = \sqrt{3}$

Theorem (Jean-Guillaume de Damas, Laura Grigori)

If H_k is unreduced in (13), then $r_k = 0$ if and only if $v_1 = Zy$, where $AZ = ZT$ is a k partial Ω-orthonormal Schur factorization of A with $Z \in \mathbb{R}^{n \times k}$ and for some $y \in \mathbb{R}^k$.

rIRA

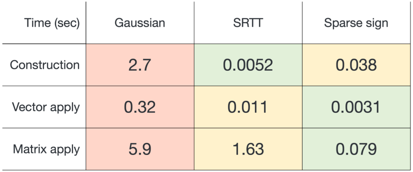
```
$k$ steps randomized Arnoldi: $AV = VH_k + r_k e_k^T$, $S = (\Omega V) \in \mathbb{R}^{d \times k}$
while convergence not declared do
    Extend $AV = VH_{k+p} + r_{k+p} e_{k+p}^T$, $S = (\Omega V) \in \mathbb{R}^{d \times (k+p)}$
    Compute eigenvalues $(\tilde{\lambda}_1, \dots, \tilde{\lambda}_{k+p})$ of $H_{k+p}$. Monitor residual norms on wanted pairs and define $p$ shifts from unwanted.
    Get $H_{k+p}^{+} = Q^T H_{k+p} Q$ from $p$ QR shifted steps
    Set $\tilde{V} = VQ$ and $\tilde{S} = SQ$
    Set $\tilde{r}_k = \alpha \tilde{V}(:, k+1) + \beta r_{k+p}$
    Set $\tilde{s}_k = \alpha \tilde{S}(:, k+1) + \beta s_{k+p}$
    Truncate $\tilde{H}_k = H_{k+p}^{+}(1:k, 1:k)$, $\tilde{V} = \tilde{V}(:, 1:k)$ and $\tilde{S} = \tilde{S}(:, 1:k)$
    Continue with factorization $A\tilde{V} = \tilde{V}\tilde{H}_k + \tilde{r}_k e_k^T$, $\tilde{S} = (\Omega \tilde{V}) \in \mathbb{R}^{d \times k}$.
end while
```

Sketching matrix

We take Ω as a Sparse Sign matrix of parameter $\zeta = 8$:

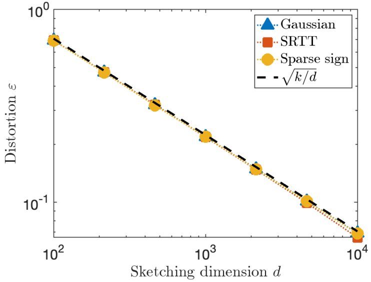
$$\Omega = \frac{1}{\sqrt{\zeta}}[s_1 \dots s_n], \tag{14}$$

where each s_i is a sparse column with exactly ζ random signs ± 1 draw with probability $1/2$. Experiments from Ethan Epperly's blog:



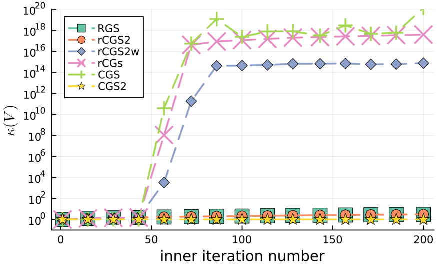
Time (sec)	Gaussian	SRTT	Sparse sign
Construction	2.7	0.0052	0.038
Vector apply	0.32	0.011	0.0031
Matrix apply	5.9	1.63	0.079

From *"Which sketch should I use ?"*
$n = 10^6, d = 400, k = 200$



$\kappa(V)$ for various orthogonalization and Ritz values

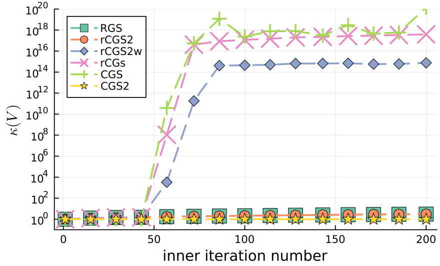
$A \in \mathbb{R}^{33{,}833 \times 33{,}833}$ named poli4, $k = 100$ and $k + p = m = 200$.



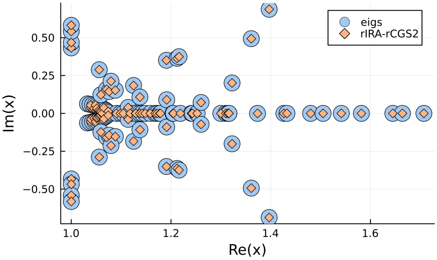
$\kappa(V)$ over inner iterations

$\kappa(V)$ for various orthogonalization and Ritz values

$A \in \mathbb{R}^{33,833 \times 33,833}$ named poli4, $k = 100$ and $k + p = m = 200$.



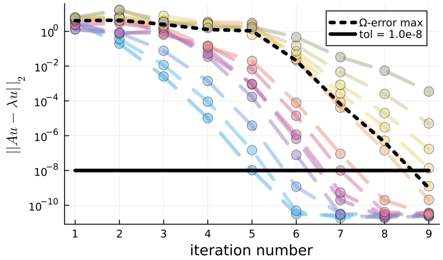
$\kappa(V)$ over inner iterations



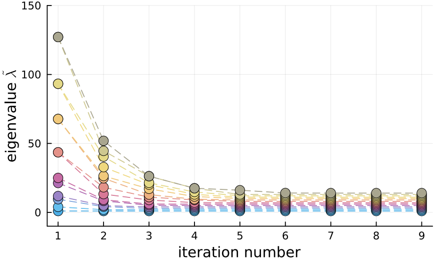
Eigenvalues using rCGS2

Eigenvalues obtained in 2 outer iterations.

$A \in \mathbb{R}^{800\times 800}$ with $\Lambda_A = \{1, 2, 3, \dots, 800\}$



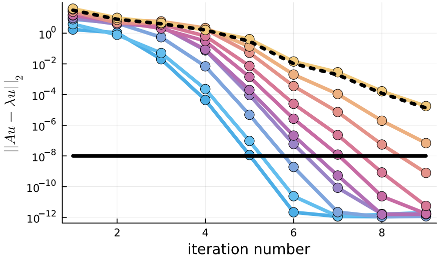
Residual norms $\|A\tilde{u}_i - \tilde{\lambda}_i \tilde{u}_i\|$



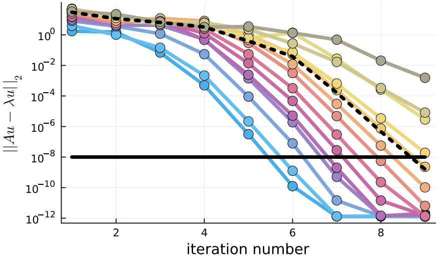
Ritz values

rIRA to compute $k = 10$ Ritz pairs of smallest modulus (SM) for the non-symmetric 800×800 toy matrix A with spectrum $\Lambda_A = \{1, 2, 3, \dots, 800\}$ and $k + p = 50$

$A \in \mathbb{R}^{800\times 800}$ with $\Lambda_A = \{1, 2, 3, \dots, 800\}$

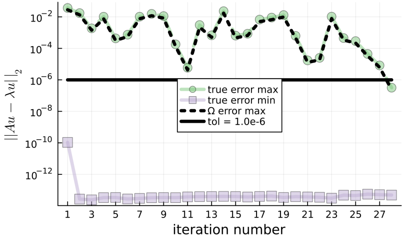


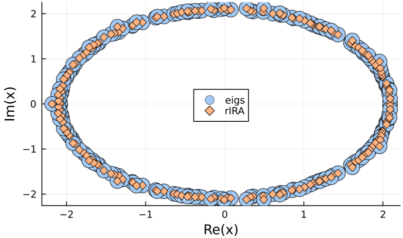
$k = 10, k + p = 50$



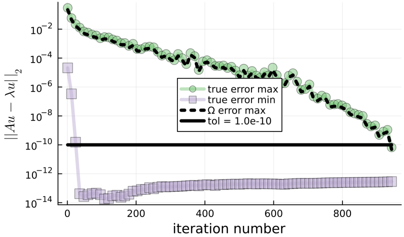
$k = 14, k + p = 50$

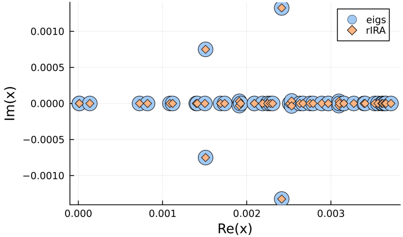
Hamrle3: Circuit Simulation, $n = 1.4 \times 10^6$, $k = 200$ LM, $m = 500$.





Vas_stokes_1M: Semiconductor Process, $n = 10^6$, $k = 50$ SM, $m = 100$.





Comparison eigs / RGS-rIRA. Tolerance $\eta = 10^{-10}$

k	m	Which	Nit	Time (s)
tmt_unsym (Electromagnetic Problem) of size $n = 9 \times 10^5$ with 4.5×10^6 nonzeros.				
20	200	LM	98 / 102	1417 / **1296**
20	200	SM	74 / 69	1149 / **1054**
Vas_stokes_1M (Semiconductor Process Problem) of size $n = 1.1 \times 10^6$ with 3.5×10^7 nonzeros.				
50	200	LM	7 / 7	175 / **161**
50	200	SM	336 / 575	7727 / **12860**
atmosmodl (Computational Fluid Dynamics) of size $n = 1.5 \times 10^6$ with 1.0×10^7 nonzeros.				
50	200	LM	30 / 32	725 / **638**
50	200	SM	32 / 34	740 / **687**
ML_Geer (Strucural Problem) of size $n = 1.5 \times 10^6$ with 1.1×10^8 nonzeros.				
50	200	LM	3 / 3	142 / **134**
50	200	SM	maxit (1000)	no convergence

Randomized Arnoldi analysis

Theorem (Optimal property)

Suppose $AV = VH_k + r_k e_k^T$ is a randomized Arnoldi factorization. Then H_k is the representation in $\mathbb{R}^k$ with respect to the basis V of the randomized projection $P^{\Omega}_{\mathcal{K}_k} A P^{\Omega}_{\mathcal{K}_k}$ restricted to $\mathcal{K}_k$, that is

$$P^{\Omega}_{\mathcal{K}_k} A P^{\Omega}_{\mathcal{K}_k} x = VH_k y \quad \text{when } x = Vy. \tag{15}$$

Moreover, its characteristic polynomial $\hat{p}_k$ minimizes $\|\Omega p(A) v_1\|$ over the set $\mathcal{PM}_k$ of all monic polynomial p of degree k, i.e.

$$\hat{p}_k = \arg \min_{p \in \mathcal{PM}_k} \|\Omega p(A) v_1\| \tag{16}$$

Randomized Arnoldi analysis

Theorem (Residual as a function of v_1)

Let $AV_j = V_jH_j + r_je_j^T$ be a sequence of j successive inner randomized Arnoldi steps with $1 \leq j \leq k$ such that $V_j \in \mathbb{R}^{n\times j}$, $H_j \in \mathbb{R}^{j\times j}$, H_k is unreduced and Ω is an ε-embedding for $\operatorname{span}\{V_k, r_k\}$. Note by $\hat{p}_j$ the characteristic polynomial of H_j. Then:

$$r_j = \frac{\hat{p}_j(A)v_1}{\|\Omega\hat{p}_{j-1}(A)v_1\|}. \tag{17}$$

Randomized Arnoldi analysis

Theorem (Residual as a function of v_1)

Let $AV_j = V_j H_j + r_j e_j^T$ be a sequence of j successive inner randomized Arnoldi steps with $1 \leq j \leq k$ such that $V_j \in \mathbb{R}^{n\times j}$, $H_j \in \mathbb{R}^{j\times j}$, H_k is unreduced and Ω is an ε-embedding for $\mathrm{span}\{V_k, r_k\}$. Note by $\hat{p}_j$ the characteristic polynomial of H_j. Then:

$$r_j = \frac{\hat{p}_j(A)v_1}{\|\Omega \hat{p}_{j-1}(A)v_1\|}. \tag{17}$$

Theorem (Randomized Implicit Q theorem)

A randomized Arnoldi $AV = VH_k + r_k e_k^T$ is uniquely defined by v_1.

rIRA analysis

Theorem (Restarting subspace)

rIRA does not add error when restarting from Ritz vectors $\tilde{u}_1, \dots, \tilde{u}_k$*:*

$$\operatorname{span}\{VQ[e_1, \dots, e_k]\} = \operatorname{span}\{\tilde{u}_1, \dots, \tilde{u}_k\} \tag{18}$$

rIRA analysis

Theorem (Restarting subspace)

rIRA does not add error when restarting from Ritz vectors $\tilde{u}_1, \ldots, \tilde{u}_k$:

$$\operatorname{span}\{VQ[e_1, \ldots, e_k]\} = \operatorname{span}\{\tilde{u}_1, \ldots, \tilde{u}_k\} \tag{18}$$

Theorem (Fixed shifts convergence)

Let π_i the product of the subdiagonal elements of $H_k^{(i)}$. Then there exists a constant K and a positive integer I such that for $i > I$:

$$0 \leq \pi_i \leq \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\gamma^i K. \tag{19}$$

Summary

- Explored randomization for the Arnoldi process, proposed rCGS2
- Derived a restarting algorithm rIRA using randomized Arnoldi
- Maintained precision, obtained speedup

Thank you for your attention.
Reach me at jean-guillaume.de-damas@inria.fr!

Appendix

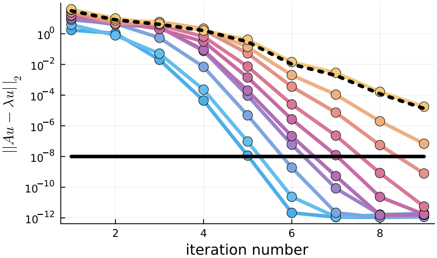
- W is defined as

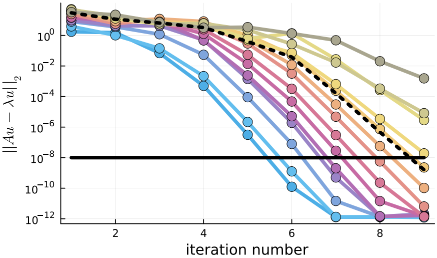
$$W_{i,j} = \frac{\sin(10(\mu_j + x_i))}{\cos(100(\mu_j - x - i)) + 1.1} \tag{20}$$

  where x_i and μ_j range 0 to 1 with equally distanced points, given $n = 10^5$ and $k = 300$.

- Comparison eigs / rIRA done on a node with 2x Cascade Lake Intel Xeon 5218 16 cores, 2.4GHz processor and 192GB of RAM.
- Sparse Sign matrix of parameter $\zeta = 8$: $\Omega = \frac{1}{\sqrt{\zeta}}[s_1 \dots s_n]$, where each s_i is a sparse column with exactly ζ random signs ±1 draw with probability $1/2$.

Appendix


10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
$||Au - \lambda u||_2$
2
4
6
8
iteration number


10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
$||Au - \lambda u||_2$
2
4
6
8
iteration number