FILTERING SUBSPACES

How parallelism and HPC gave new life to an old eigenvalue solver method

October 28, 2019 | E. Di Napoli |

OUTLINE

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

Summary

TOPIC

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing
- Eigencount and spectral estimates
- Shifted linear systems with multiple RHS
- A roadmap to filter optimization

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

Summary

ONCE UPON A TIME... (1957)

Eigensolvers based on subspace iteration " ... where developed in the 1960s and 1970s when the Lanczos algorithm was under the cloud." **Beresford N. Parlett (1998)**

F. L. Bauer (1924-2015)

BAUER F. L.: Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme" *(The process of staircase iteration and related procedures for the solution of the algebraic eigenvalue problem).*
Zeitschrift für angewandte Mathematik und Physik ZAMP May 1957, Volume 8, Issue 3, pp 214–235

Bi-iteration to solve $Au = \lambda u$

$$X \leftarrow AX \quad Y \leftarrow A^\top Y \qquad X = (x_1, \ldots, x_p) \quad Y = (y_1, \ldots, y_p)$$

$X \perp Y$ using linear conbinations.
Under certain conditions

$$X \to U \quad \text{and} \quad Y \to V$$

THE RISE OF SUBSPACE ITERATION (1969)

H. RUTISHAUSER

Computational Aspects of F.L. Bauer's Simultaneous Iteration Method. Numer. Math. 13 pp.4-13 (1969)

Simultaneous Iteration Method for Symmetric Matrices. Numer. Math. 16 pp. 205-223 (1970)

H. Rutishauser (1918-1970)

Key ingredients: polynomial filtering, QR factorization, search space projection, and a Ritz iteration. In addition, Rutishauser suggests:

1. initial Ritz iteration $\Rightarrow$ estimate of the bounds of the filtered out interval;
2. initial iteration with a low polynomial degree $\Rightarrow$ to avoid large error in computing the extremal eigenvalues;
3. to combine a limited number of filtering iterations before performing the orthonormalization;
4. to carefully select the initial set of vectors;
5. to limit the maximum value of the polynomial degree used in the filter to mitigate the effect of floating point errors;
6. a locking mechanism that stores converged eigenpairs.

THE GOLDEN AGE (1970-1972)

Contemporary to RUTISHAUSER

- CLINT AND JENNINGS simultaneous work of on eigenvectors and eigenvalues of real symmetric matrices by “simultaneous iteration”.[1]
- STEWART upgraded Ritz iteration, which projects the eigenproblem onto the search space using a Jacobi step, to a Rayleigh-Ritz step so as to avoid to deal with non-positive definite matrices[2].
- STEWART extension of “simultaneous iteration” to non-Hermitian eigenproblems.[3]

Quotient iteration

Stewart rigorously demonstrated the enhanced convergence of the eigenpairs when orthogonal iteration (subspace + orthonormalization) is used in conjunction with Rayleigh-Ritz.

[1] The Computer Journal, 13, p. 76 (1970) [submitted in 1968]
[2] Numer.Math. 13 pp.362-376 (1969)
[3] Numer. Math. 25, pp.125-136, (1976)

QUOTIENT ITERATION

Schematic algorithm

INPUT: Matrix A,
OUTPUT: NEV extremal eigenpairs (Λ, W).

1. Set the size of the active space $\ell >$ NEV
2. $V \longleftarrow \text{rand}(n, \ell)$

REPEAT UNTIL CONVERGENCE:

3. Compute $V \longleftarrow AV$
4. Orthogonalize the vectors $V = QR$.
5. Compute the Rayleigh quotient $G = Q^H A Q$.
6. Compute the primitive Ritz pairs (Λ, Y) by solving for $GY = Y\Lambda$.
7. Compute the Ritz pairs $(\Lambda, V \leftarrow QY)$.
8. Check which one among the Ritz pairs converged.
9. Deflate and lock the converged vectors.

END REPEAT

QUOTIENT ITERATION

Convergence properties

Backward perturbation analysis shows:

1. Ritz vector w_a converges linearly

$$\sin \angle \left(x_a, Q^{(i)} w_a^{(i)} \right) = \mathcal{O}(\theta_a^{(i)}) \quad \text{with} \quad \theta_a^{(i)} = \angle(x_a, \mathcal{R}(V^{(i)}))$$

2. Ritz value $\tilde{\lambda}_a$ converges quadratically

$$\left| \tilde{\lambda}_a^{(i)} - \lambda_a \right| = \mathcal{O}(\left[\theta_a^{(i)} \right]^2)$$

   as long as the uniform separation condition is satisfied.

3. Raleigh-Ritz minimizes the residual. If residual is small, eigenpair is an exact eigenpair of the perturbed matrix $(A + E)$.

Therefore, the standard stopping criterion for a subspace iteration is based on the computation of eigenpair residuals and not on the error.

Historically Rutishauser uses a termination criterion based on the stagnation of the angles $\theta_a^{(i)}$ and does not compute the residual before the angle is accepted.

COMPUTATIONAL FEATURES OF QUOTIENT ITERATION

Computer Memory

- Expensive
- Limited in size

Performance

Reduction of executed floating point operations (FLOPs) for speedup.

Subspace Iteration (SI) most computing and memory intensive elements

- The simultaneous iteration $V \longleftarrow AV$ has to be computed on the whole block of vectors at once. The use of polynomials requires two stack of vectors.
- Orthogonalizing the vectors $V = QR$ requires additional workspace.

SI undesirable traits

- Requires a fair amount of non-trivial memory management
- Performs a fairly large number of FLOPs to reach convergence.

Ritzit was implemented by RUTISHAUSER in Algol-60 which allowed for dynamical memory allocation.

TOPIC

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing
- Eigencount and spectral estimates
- Shifted linear systems with multiple RHS
- A roadmap to filter optimization

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

Summary

THE ENGINES BEHIND THE RESURGENCE

Density Functional Theory (DFT) and Parallelism

Electronic structure methods

- $\Phi(x_1; s_1, x_2; s_2, \ldots, x_n; s_n) \Longrightarrow \Lambda_{i,a}\phi_a(x_i; s_i)$
- $n(\mathbf{r}) = \sum_a f_a \left|\phi_a(\mathbf{r})\right|^2$
- Hohenberg-Kohn theorem

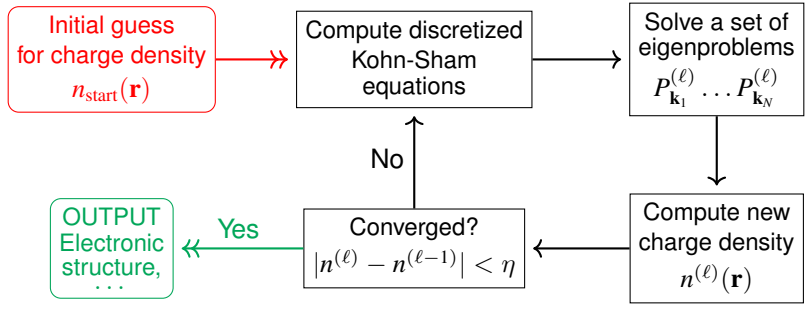
Parallelism

- Introduction of hierarchy of caches (already proposed by J. von Neumann in 1947)
- Rise of multi-core CPU
- General purpose cluster with thousands of CPUs

DFT SELF-CONSISTENT FIELD CYCLE

The Schrödinger equation for a all the dgrees of freedom of a multi-atom systems translates into a set of coupled non-linear low-dimensional self-consistent Kohn-Sham (KS) equation

$$\forall\, a \quad \text{solve} \quad \hat{H}_{\mathrm{KS}}\phi_a(\mathbf{r}) = \left(-\frac{\hbar^2}{2m}\nabla^2 + V_0(\mathbf{r})\right)\phi_a(\mathbf{r}) = \epsilon_a\phi_a(\mathbf{r})$$



ZOO OF METHODS

LDA
GGA
LDA + U
Hybrid functionals
GW-approximation

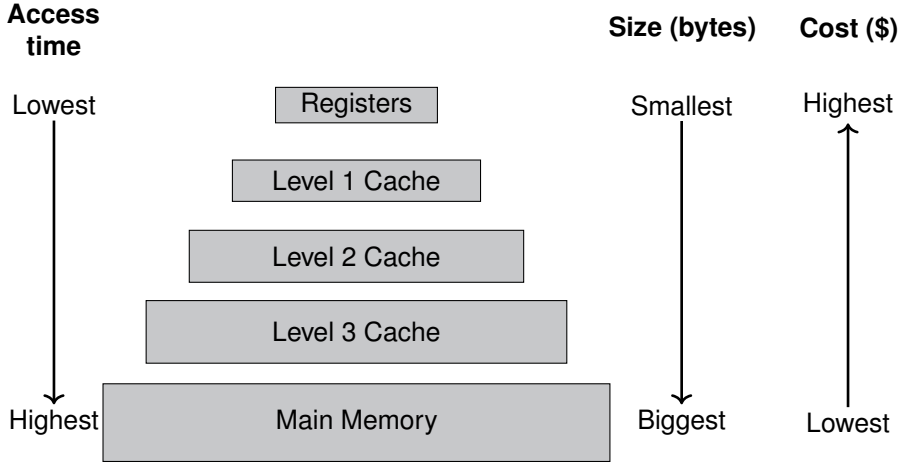
Plane waves
Localized basis set
Real space grids
Green functions

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V_0(\mathbf{r})\right)\phi_a(\mathbf{r}) = \epsilon_a\phi_a(\mathbf{r})$$

Finite differences
Non-relaticistic eqs.
Scalar-relativistic approx,
Spin-orbit coupling
Dirac equation

All-electron
Pseudo-potential
Shape approximations
Full-potential
Spin polarized calculations

A NEW HIERARCHY OF MEMORIES



MASSIVE PARALLELISM

- Multi-core CPUs
- High throughput and low latency interconnect $\Rightarrow$ large intranode networks
- General purpose computing clusters based on commodity (cheap) hardware

TWO METHODS, ONE SUBSPACE

Rational Filters (FEAST)

- E. POLIZZI, Density-matrix-based algorithm for solving eigenvalue problems. Phys. Rev. B 79, 115112 (2009)

Polynomial Filters

Y. ZHOU ET AL., Self-consistent field calculations using Chebyshev-filtered subspace iteration. J. Comp. Phys. 219, 172-184 (2006).

Features

- *Target:* $Ax = \lambda Bx$ with $A = A^H$, $B = B^H$ and B s.p.d.
- *Type:* Sparse matrices
- *Filter:* $r(A,B) = \sum_{i=1}^{n} \beta_i (A - z_i B)^{-1} B$
- *Main kernel:* Linear system solver
- *Convergence:* Solver accuracy, eigevalues count, filter accuracy.
- *Memory impact:* Very high.
- *FLOP count:* Moderately high.

- *Target:* $Ax = \lambda x$ with $A = A^H$
- *Type:* Sparse matrices
- *Filter:* Chebyshev polynomial $C_m(A)$ of degree m
- *Main kernel:* `Mat-Vec`
- *Convergence:* Polynomial degree.
- *Memory impact:* Moderately high
- *FLOP count:* Very high (depends on kernel)

THE EVOLUTION

A proliferations of codes

Rational filter eigensolves

- FEAST, IFEAST, PFEAST,etc.
- z-Pares
- PHIST
- CISS
- FITLAN
- ⋮

Polynomial Filter eigensolvers

- ChASE
- ESVL
- BEAST
- DGCheFSI
- ChebFD
- ⋮

TOPIC

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing
- Eigencount and spectral estimates
- Shifted linear systems with multiple RHS
- A roadmap to filter optimization

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

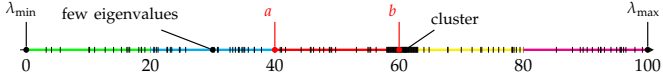
Summary

FRAMEWORK

The problem

$$Au = \lambda Bu, \quad \lambda \in [a, b], \quad A, B \in \mathbb{C}^{n \times n} \tag{1}$$

The domain



The filter

$$r(A,B) := \sum_{i}^{n} \beta_i (A - Bz_i)^{-1} B \approx \frac{1}{2\pi i} \oint_{\Gamma} (A - zB)^{-1} B \, dz \quad \equiv \sum_{\lambda_j \in [a\ b]} u_j u_j^T B$$

METHOD

REPEAT UNTIL CONVERGENCE:

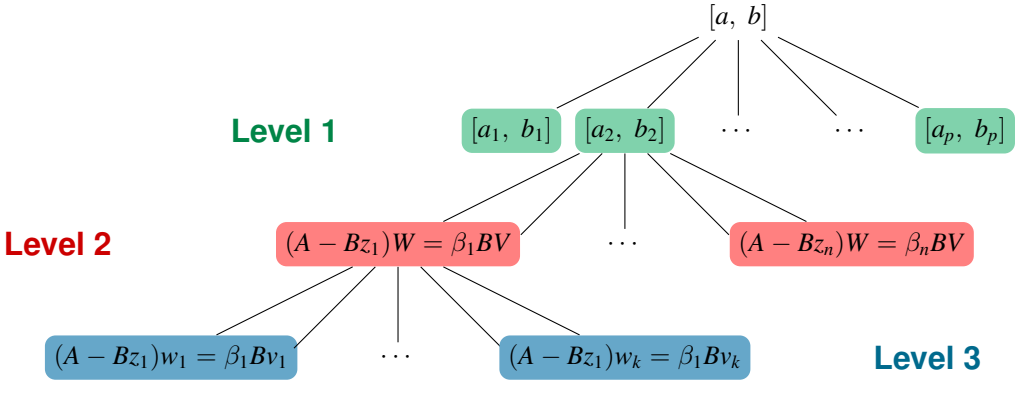
2. **Filter a block of vectors** $V \longleftarrow r(A,B)V = \sum_{i=1}^{n} \beta_i (A - Bz_i)^{-1} BV$
3. Re-orthogonalize the vectors outputted by the filter; $V = QR$.
4. Compute the Rayleigh quotient $G = Q^H \tilde{A} Q$.
5. Compute the primitive Ritz pairs (Λ, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Lambda, V \leftarrow QY)$.

END REPEAT

Core elements

1. Relies on good estimates of number $\mu_{[a\ b]}$ of eigenvalues in $[a\ b]$
2. Solve for multiple right-hand side linear systems $(A - Bz_i)W = \beta_i BV$ per complex pole z_i
3. Accuracy depends on the accuracy of the projector $r(A,B)$

PARALLELISM



LOAD BALANCING

1. **Level 1** is influenced by
   - the evenness in the distribution of number $\mu_{[a_j\ b_j]}$ of eigenvalues in each sub-interval $[a_j\ b_j]$;
   - the effectiveness of the projector $r(A, B)$ in filtering the subspace corresponding to each single interval $[a_j\ b_j]$.
2. **Level 2** is influenced by
   - the time to solution for linear systems $(A - Bz_i)W = \beta_i BV$ defined by the same matrices but distinct shifts (poles) z_i and RHS coefficients β_i;
   - the efficiency of the projector $r(A, B)$ in regulating the number of subspace iterations until convergence;
   - the number $\mu_{[a_j\ b_j]}$ of eigenvalues in $[a_j\ b_j]$ which is directly related to the size of the RHS of the linear systems.
3. **Level 3** is influenced by
   - the time to solution for linear systems $(A - Bz_i)w_k = \beta_i Bv_k$ defined by the same matrices but distinct RHS v_k;
   - the efficiency of the projector $r(A, B)$ in regulating the number of subspace iteration until convergence.

THREE IMPORTANT ELEMENTS

Eigenvalue distribution across sub-intervals

$\surd$ Kernel Polynomial Method or Lanczos DoS[a] + Stochastic estimate[b] are a good approach to address issue.

[a] L. Lin et al. DOI:10.1137/130934283
[b] EDN et al. DOI:abs/10.1002/nla.2048

Iterative linear solvers

Matching linear solver + pre-conditioner for shifted linear systems and predicting relative time to solution.

Efficiency and robustness of rational filter

$\Rightarrow$ Optimize filter using Non-linear Least Squares for best worst-case convergence[ab].

[a] J. Winkelmann, EDN DOI:10.3389/fams.2019.00005
[b] K. Kollnig, EDN To Be Submitted to SISC

ESTIMATING THE SPECTRAL EXTREMA

Estimate of λ_1 and $b_{\text{sup}} > \lambda_N$: simple repeatition of few Lanczos steps[1]

1. Compute k Lanczos steps

$$AU = UT_k + f_k e_k^\top \qquad T_k = Z^H \tilde{\Lambda}_k Z \qquad \tilde{\Lambda}_k = \text{diag}[\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k]$$

2. Compute upper bound

$$b_{\text{sup}} = \|f_k\|_2 + \max[\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k]$$

3. Estimate lower eigenvalue

$$\lambda_1 = \min[\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k]$$

$k \sim 25$ is usually sufficient

[1] Based on work by Zhou and Li (2011)

ESTIMATING THE SPECTRAL DISTRIBUTION

Additional Laczos steps are executed to build a spectral density[2] $\tilde{\phi}(t)$.

1. Compute n_{vec} times k Lanczos steps

$$AU^{[j]} = U^{[j]}T_k^{[j]} + f_k^{[j]}e_k^\top \qquad T_k^{[j]} = (Z^{[j]})^H \tilde{\Lambda}_k^{[j]} Z^{[j]} \qquad \tilde{\Lambda}_k^{[j]} = \text{diag}[\tilde{\lambda}_1^{[j]}, \ldots, \tilde{\lambda}_k^{[j]}]$$

2. Compute the spectral density

$$\tilde{\phi}(t) = \frac{1}{n_{\text{vec}}} \sum_{j=1}^{n_{\text{vec}}} \sum_{i=0}^{k} |Z_{1,i}^{[j]}|^2 g_\sigma(t - \tilde{\lambda}_i^{[j]})$$

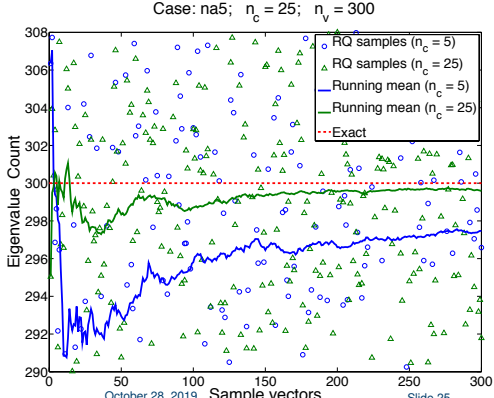
Width of the Gaussian $\sigma = 0.25 * |b_{\text{sup}} - \lambda_1|$
Number of random vectors $n_{\text{vec}} = 3 \div 5$

[2]Based on work by Lin et al. (2016)

STOCHASTIC ESTIMATES OF EIGENVALUE COUNT

Stochastic estimator

$$\mathrm{Trace}[r(A,B)] \approx \frac{n}{n_v} \sum_{j=1}^{m} \beta_j \sum_{k=1}^{n_v} v_k^\top (A - z_j B)^{-1} B v_k; \quad v_k[i] \text{ random i.i.d.}$$



THE (STILL) EXISTING BOTTLENECK

Linear systems solution

The general issue

Solving many linear system with multiple shifts and multiple right-hand-sides.

1. Algorithms favoring exchange of information for a higher degree of parallelism: MHGMRES, BGMREG-Sh, etc.
2. Algorithms favoring block operations for a better use of performant kernels: block variants of BiCGStab, QMR, GMRES in combination with middleware libraries such as OSKI, GHOST, etc.

No matter if one aims at more parallelism or more performance, it is still very difficult to balance workloads across distinct shifs and RHS.

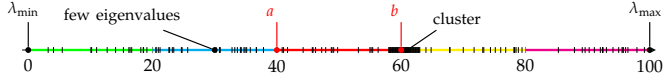
Linear systems and load balancing

(i) Must match solver + preconditioner to linear system (Anamod, Lighthouse)

(ii) Must predict relative time to solution across shifts+RHS so as to assign resources at runtime.

OPTMIZING THE FILTER

Setting up the problem



Ideal filter

$$\mathbb{1}_{(a,b)}(x) = \begin{cases} 1, & \text{if } x \in [a,b], \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

(Symmetric) Rational filter

$$r_{\beta,z}(x) := \sum_{i=1}^{m} \frac{\beta_i}{x - z_i} + \frac{\overline{\beta_i}}{x - \overline{z_i}} - \frac{\beta_i}{x + z_i} - \frac{\overline{\beta_i}}{x + \overline{z_i}}, \quad x \in \mathbb{R}, \quad \text{with } \beta \in \mathbb{C}^m, z \in (\mathbb{H}^{+R})^m \tag{3}$$

Objective function

$$f_\omega(\beta, z) := \int_{-\infty}^{\infty} \omega(x) \left(\mathbb{1}_{(a,b)}(x) - r_{\beta,z}(x)\right)^2 \mathrm{d}x, \tag{4}$$

Minimization problem

$$\underset{\beta \in \mathbb{C}^m, z \in (\mathbb{H}^{+R})^m}{\operatorname{argmin}} \; f_\omega(\beta, z). \tag{5}$$

SLISE FILTERS

Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm

- Supports only real-valued objective functions $f_\omega : \mathbb{C}^m \times \mathbb{H}^{+R} \to \mathbb{R} \;\Rightarrow\; \tilde{f}_\omega : \mathbb{R}^{4m} \to \mathbb{R}$

$$\tilde{f}\left(\begin{pmatrix}\Re(\beta^\top)\\ \Re(z^\top)\\ \Im(\beta^\top)\\ \Im(z^\top)\end{pmatrix}\right) := f(\Re(\beta) + i\Im(\beta), \Re(z) + i\Im(z)). \tag{6}$$

- Gauss-Newton method with inverse Hessian $\tilde{f}_\omega$ recursively defined as

$$H_0 := I_{4m}, \quad H_{k+1} := (I_{4m} - \frac{s_k y_k^T}{y_k^T s_k})\, H_k\, (I_{4m} - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k}, \tag{7}$$

with

$$s_k := x_{k+1} - x_k, \quad y_k := \nabla\tilde{f}(x_{k+1}) - \nabla\tilde{f}(x_k), \tag{8}$$

Very fast convergence
Starting position: existing filters (e.g. Gauss-Legendre, Elliptic)
Dependent on weight function ω

BEYOND SLISE: WORST-CASE CONVERGENCE

Conventions

In the rest of the slides we maintain the following notations

- Standard interval $[a, b] \longrightarrow [-1, 1]$,
- Active subspace size $M_0 = C \times \mu_{[a,b]}$ and $C \geq 1$,
- Gap parameter $G \in (0, 1)$ such that $G < 1 < G^{-1}$ ($-G^{-1} < -1 < -G$).

Best Worst-Case Convergence Rate (WCR)

Given a rational filter r and some fixed *gap parameter* $G \in (0, 1)$, a filtered subspace iteration converges linearly, with probability one, at a convergence rate no larger than

$$w_G(r) = \frac{\max_{x \in [-\infty, -G^{-1}] \cup [G^{-1}, \infty]} |r(x)|}{\min_{x \in [-G, G]} |r(x)|},$$

as long as no eigenvalues lie within $[-G^{-1}, -G] \cup [G, G^{-1}]$.

BEYOND SLISE: THE WISE FILTERS

New minimization problem

$$\begin{cases} \beta', z' & \leftarrow \underset{\beta,z}{\operatorname{argmin}}\, f_{\omega'}(\beta, z) \\ \omega' & \leftarrow \underset{\omega}{\operatorname{argmin}}\, w_G(r_{\beta,z}[\omega]). \end{cases}$$

- Minimize WCR instead of Subspace Iteration convergence rate.
- Nested minimization: requires thousands of SLiSe “minimizations”.
- Derivative-free minimization: Nelder-Mead algorithm.
- Eliminate parameter dependence on weight functions.

TEST ENVIRONMENT

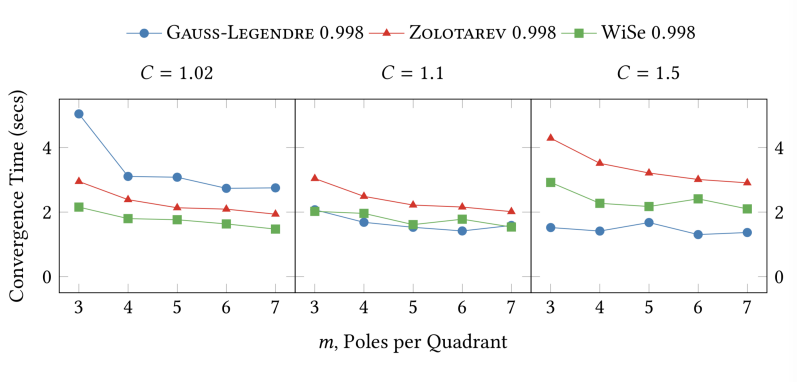
Single test

- CNT matrix, $N = 12{,}450$ with 86,808 nnz
- Interval $[a, b] = [-65.0, 4.96]$
- $M = 100$

(Large) Benchmark set

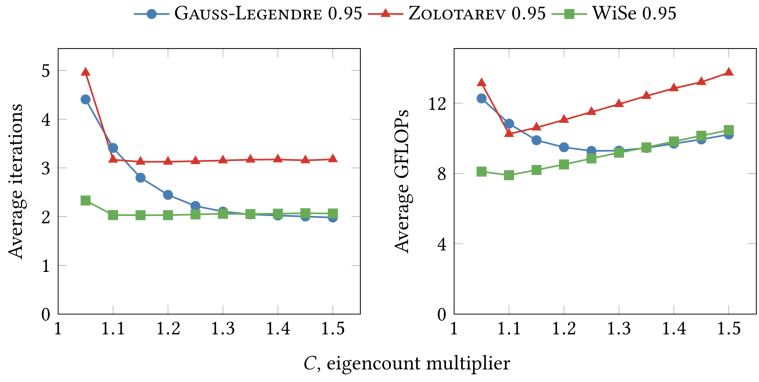
- 2116 intervals defining the corresponding interior eigenproblem,
- Each interval contains between 5 and 20 % of the total spectrum of Si_2 problem,
- Interval are selected based on “feature points”: neighborhood of an identifiable spectral feature, such as a spectral gap or a cluster.

SINGLE TEST WITH FEAST



- Best worst-case convergence of FEAST strongly correlates with WCR of filter,
- Size of the active subspace M_0 is a confounding factor: big values of C mask the correlation between WCR and τ,
- WiSe filters performance hardly depends on number of poles m.

BENCHMARK SET WITH FEAST



- Number of poles fixed to $m = 4$,
- Confirms that FEAST with WiSe filter only influenced by WCR,
- For larger active subspaces Gauss-Legendre is competitive with WiSe but costs more FLOPs.

TOPIC

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing
- Eigencount and spectral estimates
- Shifted linear systems with multiple RHS
- A roadmap to filter optimization

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

Summary

THE CORE OF THE ALGORITHM: CHEBYSHEV FILTER

The basic principle

Theorem

Let $|\gamma| > 1$ and $\mathbb{P}_m$ denote the set of polynomials of degree smaller or equal to m. Then the extremum

$$\min_{p \in \mathbb{P}_m, p(\gamma)=1} \max_{t \in [-1,1]} |p(t)|$$

is reached by

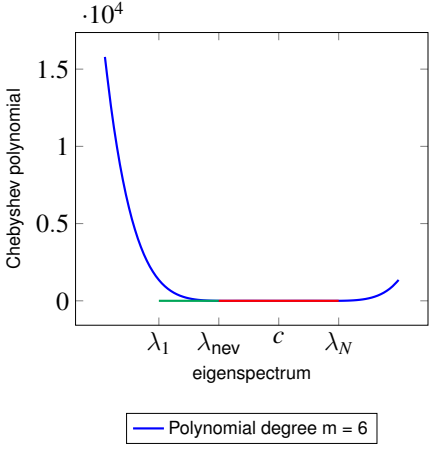
$$p_m(t) \doteq \frac{C_m(t)}{C_m(\gamma)}.$$

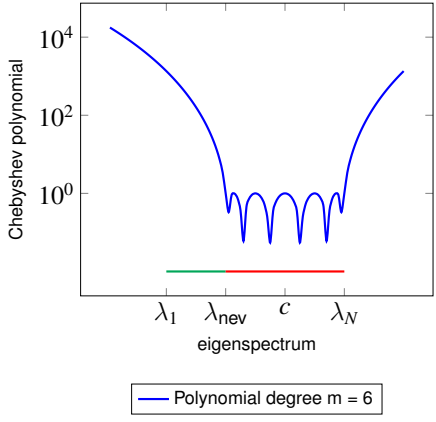
where C_m is the Chebyshev polynomial of the first kind of order m, defined as

$$C_m(t) = \begin{cases} \cos\left(m \arccos(t)\right), & t \in [-1, 1]; \\ \cosh\left(m \operatorname{arccosh}(t)\right), & |t| > 1. \end{cases}$$

DIVIDE AND CONQUER

Chebyshev polynomials





SUBSPACE ITERATION

Power Iteration: Given a generic vector $v = \sum_{i=1}^{n} s_i x_i$

$$v^m = A^m v = \sum_{i=1}^{n} s_i A^m x_i = \sum_{i=1}^{n} s_i \lambda_i^m x_i = s_1 x_1 + \sum_{i=2}^{n} s_i \left(\frac{\lambda_i}{\lambda_1}\right)^m x_i \sim \boxed{s_1 x_1}$$

Subspace iteration + Chebyshev polynomials:

$$\begin{aligned} v^m = p_m(A)v &= \sum_{i=1}^{n} s_i p_m(A) x_i = \sum_{i=1}^{n} s_i p_m(\lambda_i) x_i \\ &\approx \sum_{i=1}^{\mathsf{nev}} s_i C_m\left(\frac{\lambda_i - c}{e}\right) x_i + \sum_{j=\mathsf{nev}+1}^{n} s_j x_j \\ \text{Reorthogonalization} &+ \text{Rayleigh} - \text{Ritz} \\ &\approx \sum_{i=1}^{\mathsf{nev}} \left(s_i x_i + \sum_{j=\mathsf{nev}+1}^{n} S_j^i \frac{1}{|\rho_j|^m} x_j \right) \\ &\sim \boxed{\sum_{i=1}^{\mathsf{nev}} s_i x_i} \end{aligned}$$

THE CORE OF THE ALGORITHM: CHEBYSHEV FILTER

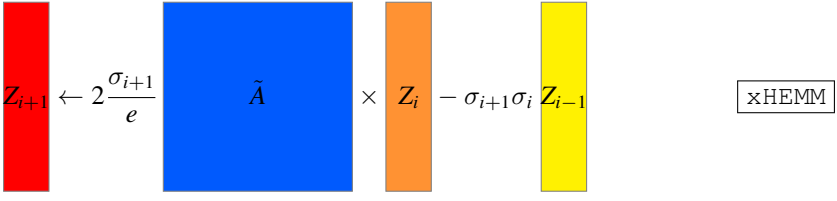
In practice

Three-terms recurrence relation

$C_{m+1}(t) = 2xC_m(t) - C_{m-1}(t)\,; \qquad m \in \mathbb{N}, \quad C_0(t) = 1, \quad C_1(t) = t$

$Z_m \doteq p_m(\tilde{A})\, Z_0$ with $\tilde{A} = A - cI_N$ and $c = \frac{b_{\sup}+\mu_{\text{nev}}}{2}$ $e = \frac{b_{\sup}-\mu_{\text{nev}}}{2}$

FOR: $i = 1 \rightarrow \text{deg} - 1$



END FOR.

CHASE LIBRARY

open source library (BSD 2.0 license)

`https://github.com/SimLabQuantumMaterials/ChASE`

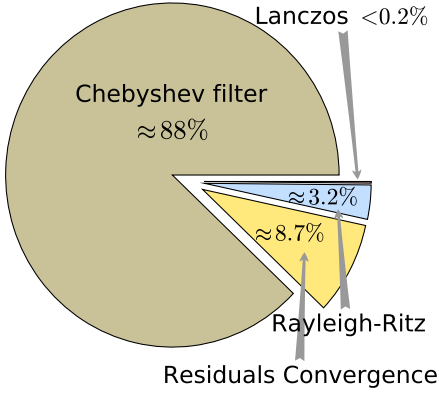
`https://doi.org/10.1145/3313828`

Highlights

- Modern C++ interface: easy-to-integrate in application codes.
- Multiple parallel implementations: performance portability.
- Excellent strong- and weak-scale performance.

WORKLOAD DISTRIBUTION

$Au_{98}Ag_{10}$ - $n = 8{,}970$ - 32 cores.

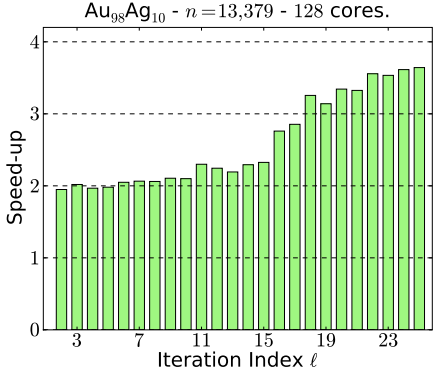


- `xGEMM` most expensive part
- Parallelizes easily over
  - MPI
  - GPUs
- Good weak scaling
- Recall: Matrix dimensions skewed

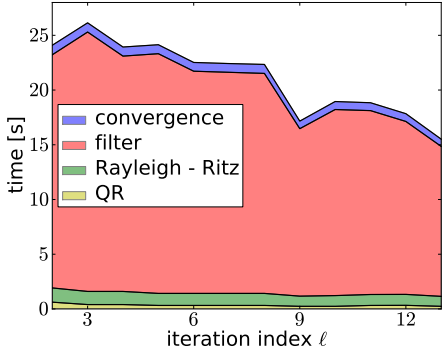
ON SEQUENCES OF EIGENPROBLEMS

As a function of DFT Self-Consistent Field iterations

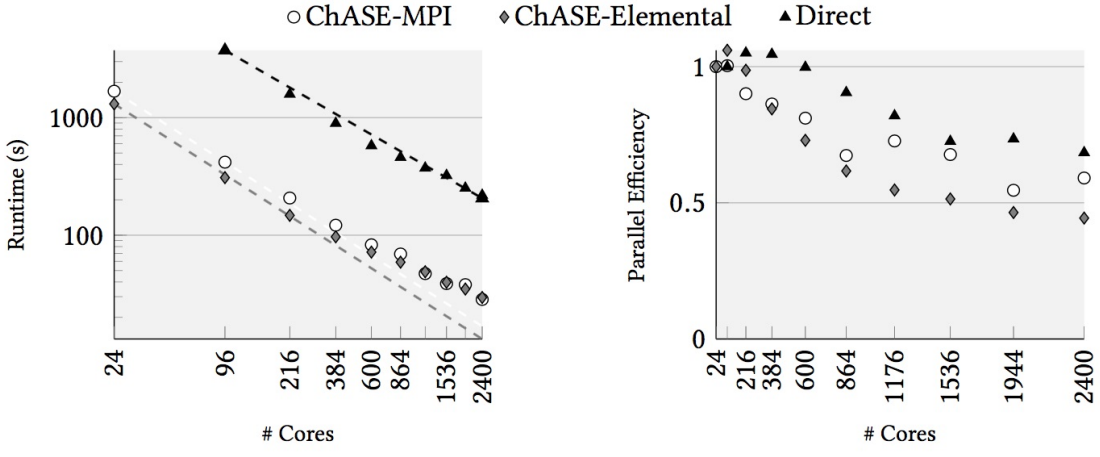
$$\text{Speed-up} = \frac{\text{CPU time (input random vectors)}}{\text{CPU time (input approximate eigenvectors)}}$$



Time spent in each stage of the algorithm as a function of the iteration index ℓ for a system of size $n = 9,273$.



STRONG SCALING



TOPIC

The early days of subspace iteration

The resurgence of “filtered” subspace iteration

Filtering by rational functions: parallelism and load balancing
- Eigencount and spectral estimates
- Shifted linear systems with multiple RHS
- A roadmap to filter optimization

Filtering by Chebyshev polynomials: low level kernels and the hierarchy of caches.

Summary

SUMMARY AND OUTLOOK

- Rational filters for Hermitian problems are mature: competitive, fast to compute, only dependence on gap parameter G, stable w.r.t. convergence rate, minimize total FLOP count.
- Estimation of spectral distribution and eigenvalue count is dependable and inexpensive.
- Solving linear systems for multiple shifts and RHS with iterative solvers have still open issues.
- Polynomial filtering is very effective in extracting FLOPS and be competitive with other solvers.
- Both rational and polynomial filtering lend itself to effective parallelization on multi-cores and many-cores platforms.

Outlook

- Integrating rational filters in the ChASE library to enable solution of interior, exterior, dense and sparse problems.
- Prediction of relative time to solution for linear systems solves,
- Optimizing filters for general complex eigenproblems.

THANK YOU

https://github.com/SimLabQuantumMaterials/

e.di.napoli@fz-juelich.de

http://www.fz-juelich.de/ias/jsc/slqm