

EMULATING FLEXIBLE PRECISION WITH OZAKI-II IN ITERATIVE METHODS: THE CASE OF THE CHASE LIBRARY

1st of September, 2026 | E. Di Napoli, C. Richefort, X.wu | Jülich Supercomputing Centre | ADAC 19th |

I. BRIEF OVERVIEW OF CHASE

ChASE
CHEBYSHEV ACCELERATED SUBSPACE EIGENSOLVER

- Multi-GPU Iterative Dense Eigensolver
- Targets Hermitian & Pseudo-Hermitian matrices
- Based on multiple sequences of GEMM operations

Already tested on a broad range of platforms, including :



JUWELS BOOSTER



FUGAKU



FRONTIER



JUPITER

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: NEV wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

JÜLICH Forschungszentrum

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: NEV wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `NEV` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `NEV` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `NEV` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.
5. Compute the primitive Ritz pairs (Ω, Y) by solving for $GY = Y\Lambda$.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `NEV` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.
5. Compute the primitive Ritz pairs (Ω, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Omega, X \leftarrow QY)$.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: NEV wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.
5. Compute the primitive Ritz pairs (Ω, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Omega, X \leftarrow QY)$.
7. Compute the residuals of the Ritz vectors $\| AX - X\Omega \|$.

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `NEV` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

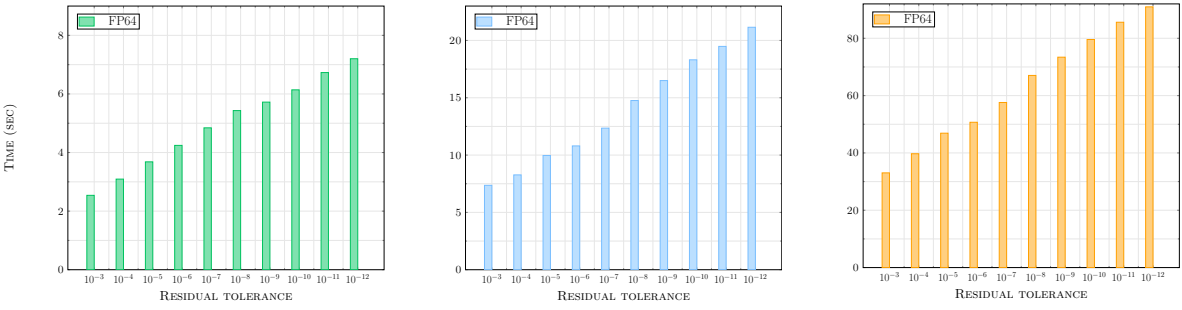
REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.
5. Compute the primitive Ritz pairs (Ω, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Omega, X \leftarrow QY)$.
7. Compute the residuals of the Ritz vectors $\| AX - X\Omega \|$.
8. Deflate and lock the converged vectors.

END REPEAT

I. BRIEF OVERVIEW OF CHASE

FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k

Medium Matrix - N = 62k

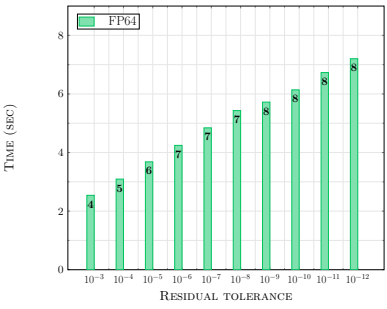
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

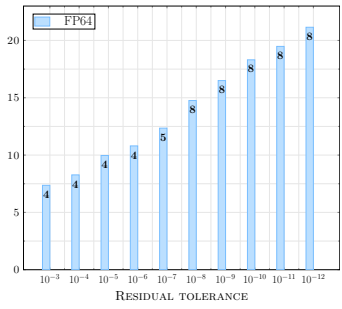
JÜLICH Forschungszentrum

I. BRIEF OVERVIEW OF CHASE

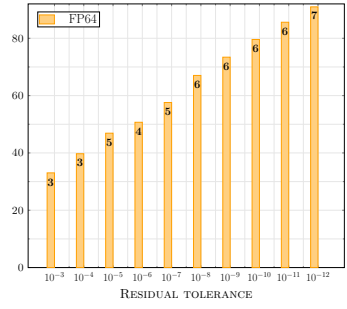
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

JÜLICH Forschungszentrum

I. BRIEF OVERVIEW OF CHASE

INPUT: Hermitian matrix A, `tol`,`deg` OUTPUT: `nev` wanted eigenpairs (Ω, X).

1. Identify the parameters $\{\mu_1, \mu_{\text{nevex}}, \mu_n\}$ with Lanczos

REPEAT UNTIL CONVERGENCE:

2. Filter a block of vectors $X \longleftarrow p(A)X$ with optimal degree.
3. Re-orthogonalize the vectors outputted by the filter; $X = QR$.
4. Compute the Rayleigh quotient $G = Q^* AQ$.
5. Compute the primitive Ritz pairs (Ω, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Omega, X \leftarrow QY)$.
7. Compute the residuals of the Ritz vectors $\| AX - X\Omega \|$.
8. Deflate and lock the converged vectors.

END REPEAT

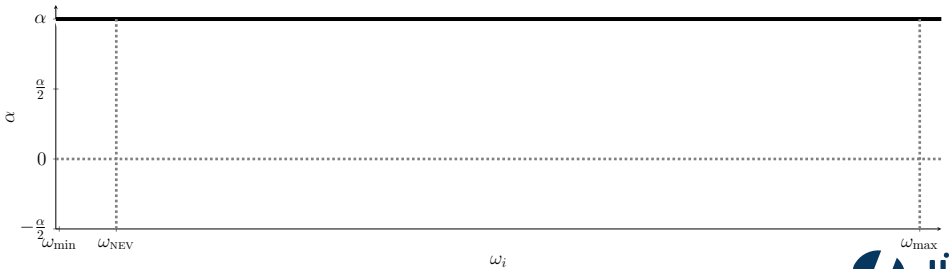
I. BRIEF OVERVIEW OF CHASE

Let a random vector $\boldsymbol{r}$ be decomposed into the eigen basis of A. Therefore,

$$\boldsymbol{r} = \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i \qquad \text{with} \qquad A\boldsymbol{x}_i = \omega_i \boldsymbol{x}_i.$$

Since $\boldsymbol{r}$ is random, let assume $\alpha_i = \alpha$. Hence, applying a polynomial of A to $\boldsymbol{r}$ gives

$$A\boldsymbol{r} = \sum_{i=1}^{m} \alpha\lambda_i \boldsymbol{x}_i \quad , \quad A^2\boldsymbol{r} = \sum_{i=1}^{m} \alpha\lambda_i^2 \boldsymbol{x}_i \quad , \quad p(A)\boldsymbol{r} = \sum_{i=1}^{m} \alpha p(\lambda_i) \boldsymbol{x}_i$$



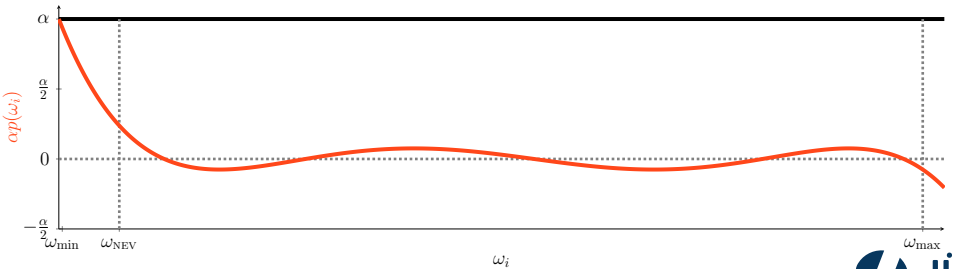
I. BRIEF OVERVIEW OF CHASE

Let a random vector $\boldsymbol{r}$ be decomposed into the eigen basis of A. Therefore,

$$\boldsymbol{r} = \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i \qquad \text{with} \qquad A\boldsymbol{x}_i = \omega_i \boldsymbol{x}_i.$$

Since $\boldsymbol{r}$ is random, let assume $\alpha_i = \alpha$. Hence, applying a polynomial of A to $\boldsymbol{r}$ gives

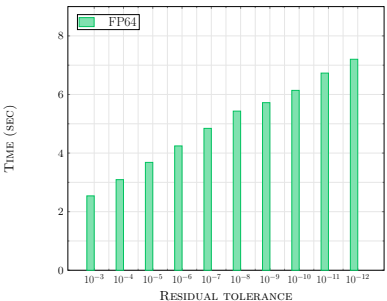
$$A\boldsymbol{r} = \sum_{i=1}^{m} \alpha\lambda_i \boldsymbol{x}_i \quad , \quad A^2\boldsymbol{r} = \sum_{i=1}^{m} \alpha\lambda_i^2 \boldsymbol{x}_i \quad , \quad p(A)\boldsymbol{r} = \sum_{i=1}^{m} \alpha p(\lambda_i)\boldsymbol{x}_i$$



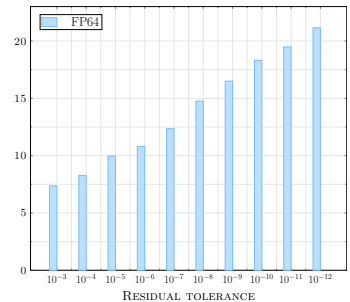
JÜLICH Forschungszentrum

II. THE FLEXIBLE RESIDUAL TOLERANCE EXPERIMENT

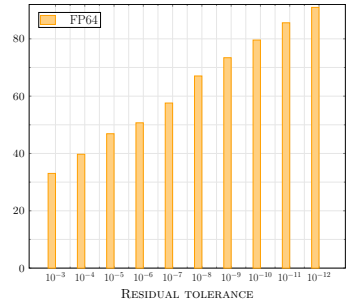
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



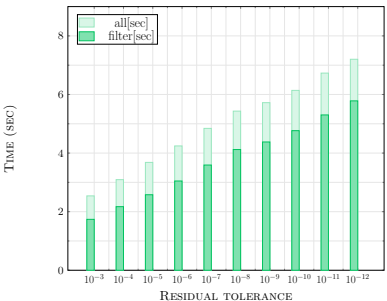
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

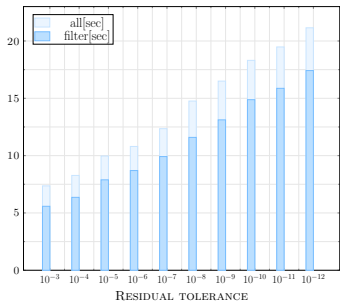
JÜLICH Forschungszentrum

II. THE FLEXIBLE RESIDUAL TOLERANCE EXPERIMENT

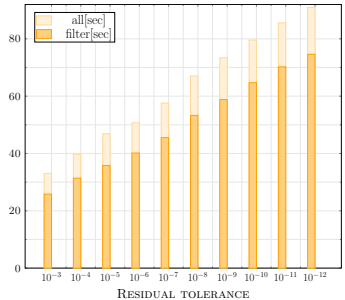
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



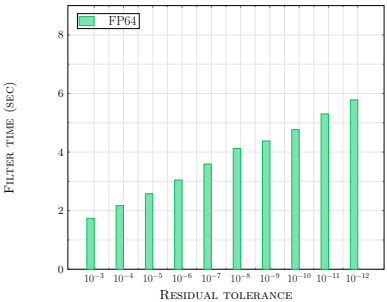
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

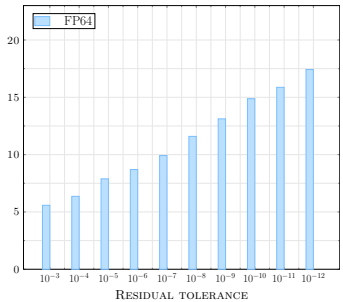
JÜLICH Forschungszentrum

II. THE FLEXIBLE RESIDUAL TOLERANCE EXPERIMENT

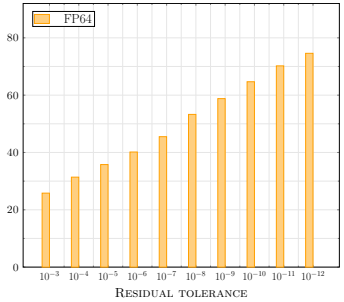
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Instead of computing the polynomial itself, we leverage on the three-term recurrence relation

$$V_{i+1} = \alpha_i A V_i + \beta_i V_{i-1} \quad \text{with} \quad V_1 = A V_0 \quad \text{and} \quad V_0 = \text{RANDOM}$$

Therefore, the implementation of the filter looks like :

```
1 V = random(N,nevex)
2 GEMM('N','N',alpha,A,V,0,W)  // W = alpha AV
3 for d <= degree:
4   alpha, beta = GetCoefficients(iter,lambda,lowerb,upperb)
5   GEMM('N','N',alpha,A,W,beta,V)  // V = alpha AW + beta V
6   V,W = W,V //swap
```


JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Instead of computing the polynomial itself, we leverage on the three-term recurrence relation

$$V_{i+1} = \alpha_i A V_i + \beta_i V_{i-1} \quad \text{with} \quad V_1 = A V_0 \quad \text{and} \quad V_0 = \text{RANDOM}$$

Therefore, the implementation of the filter looks like :

```
1 V = random(N,nevex)
2 GEMM('N','N',alpha,A,V,0,W)  // W = alpha AV
3 for d <= degree:
4   alpha, beta = GetCoefficients(iter,lambda,lowerb,upperb)
5   GEMM('N','N',alpha,A,W,beta,V)  // V = alpha AW + beta V
6   V,W = W,V //swap
```

GEMM operations ($C \leftarrow \alpha A \times B + \beta C$) represent around 80% of the FLOPS !

III. OZAKI-II SCHEME INTO CHASE

Instead of computing the polynomial itself, we leverage on the three-term recurrence relation

$$V_{i+1} = \alpha_i A V_i + \beta_i V_{i-1} \quad \text{with} \quad V_1 = AV_0 \quad \text{and} \quad V_0 = \text{RANDOM}$$

Therefore, the implementation of the filter looks like :

```
1 V = random(N,nevex)
2 OZAKI_GEMM('N','N',alpha,A,V,0,W,s)  // W = alpha AV
3 for d <= degree:
4   alpha, beta = GetCoefficients(iter,lambda,lowerb,upperb)
5   OZAKI_GEMM('N','N',alpha,A,W,beta,V,s)  // V = alpha AW + beta V
6   V,W = W,V //swap
```


JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

1. Scale and Truncate :

$$A' = \mathrm{TRUNC}(2^{\mathrm{DIAG}(\boldsymbol{u})} \times A) \qquad \text{and} \qquad B' = \mathrm{TRUNC}(B \times 2^{\mathrm{DIAG}(\boldsymbol{v})})$$

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

1. Scale and Truncate :

$$A' = \text{TRUNC}(2^{\text{DIAG}(\boldsymbol{u})} \times A) \qquad \text{and} \qquad B' = \text{TRUNC}(B \times 2^{\text{DIAG}(\boldsymbol{v})})$$

2. Generate the INT8 matrices

$$A_i = A' \bmod p_i \qquad \text{and} \qquad B_i = B' \bmod p_i, \quad \text{with} \quad i \leq s.$$

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

1. Scale and Truncate :

$$A' = \text{TRUNC}(2^{\text{DIAG}(\boldsymbol{u})} \times A) \qquad \text{and} \qquad B' = \text{TRUNC}(B \times 2^{\text{DIAG}(\boldsymbol{v})})$$

2. Generate the INT8 matrices

$$A_i = A' \bmod p_i \qquad \text{and} \qquad B_i = B' \bmod p_i, \quad \text{with} \quad i \leq s.$$

3. Compute the INT8 output matrices

$$C_i = A_i B_i \bmod p_i, \quad \text{with} \quad i \leq s.$$

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

1. Scale and Truncate :

$$A' = \text{TRUNC}(2^{\text{DIAG}(\boldsymbol{u})} \times A) \qquad \text{and} \qquad B' = \text{TRUNC}(B \times 2^{\text{DIAG}(\boldsymbol{v})})$$

2. Generate the INT8 matrices

$$A_i = A' \bmod p_i \qquad \text{and} \qquad B_i = B' \bmod p_i, \quad \text{with} \quad i \leq s.$$

3. Compute the INT8 output matrices

$$C_i = A_i B_i \bmod p_i, \quad \text{with} \quad i \leq s.$$

4. Accumulate the results

$$C' = \sum_{i=1}^{s} C_i \left(\frac{P}{p_i} \bmod p_i \right)^{-1} \frac{P}{p_i}, \quad \text{with} \quad P = \prod_{i=1}^{s} p_i.$$

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. Given a set $\{p_1, p_2, \ldots, p_s\}$ of s coprime modulis, the steps are

1. Scale and Truncate :

$$A' = \mathrm{TRUNC}(2^{\mathrm{DIAG}(\boldsymbol{u})} \times A) \qquad \text{and} \qquad B' = \mathrm{TRUNC}(B \times 2^{\mathrm{DIAG}(\boldsymbol{v})})$$

2. Generate the INT8 matrices

$$A_i = A' \bmod p_i \qquad \text{and} \qquad B_i = B' \bmod p_i, \quad \text{with} \quad i \le s.$$

3. Compute the INT8 output matrices

$$C_i = A_i B_i \bmod p_i, \quad \text{with} \quad i \le s.$$

4. Accumulate the results

$$C' = \sum_{i=1}^{s} C_i \left(\frac{P}{p_i} \bmod p_i \right)^{-1} \frac{P}{p_i}, \quad \text{with} \quad P = \prod_{i=1}^{s} p_i.$$

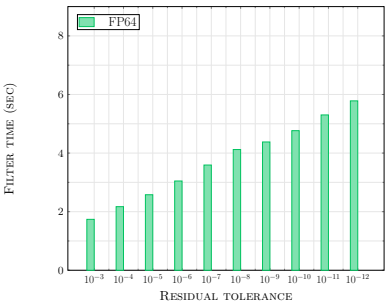
5. Unscale the accumulation

$$C = AB \approx 2^{\mathrm{DIAG}(-\boldsymbol{u})} C' 2^{\mathrm{DIAG}(-\boldsymbol{v})}.$$

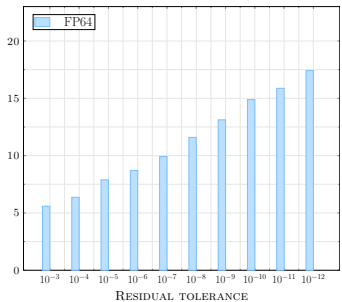
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

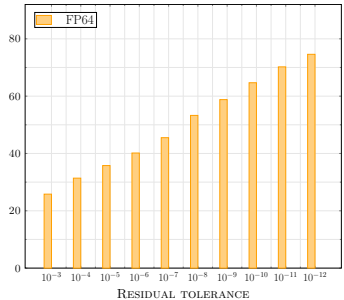
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



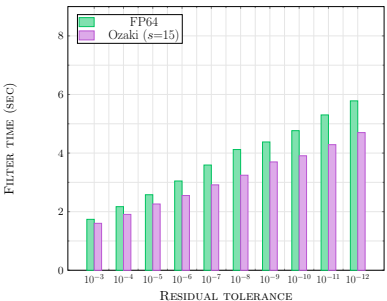
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

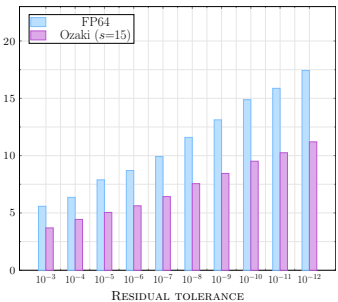
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

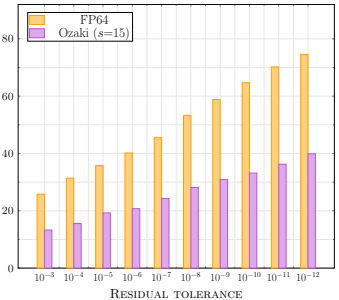
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k

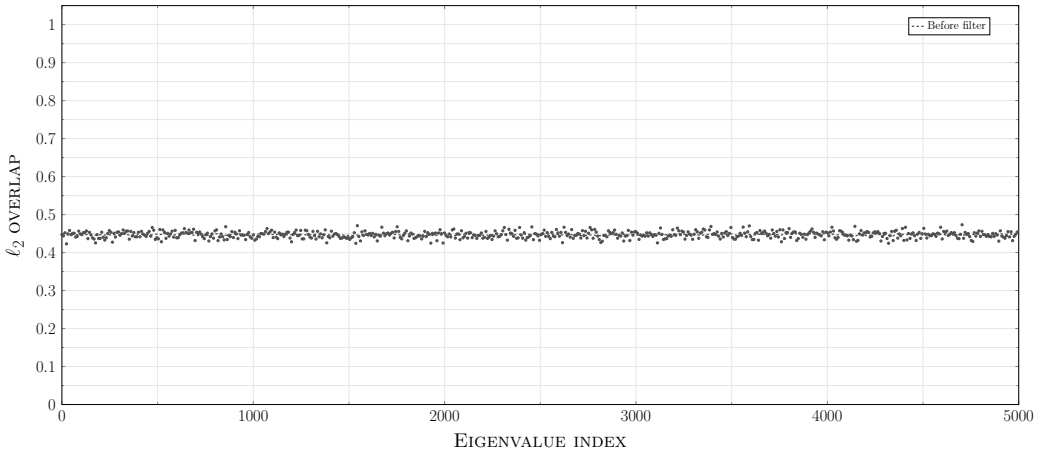


Large Matrix - N = 115k

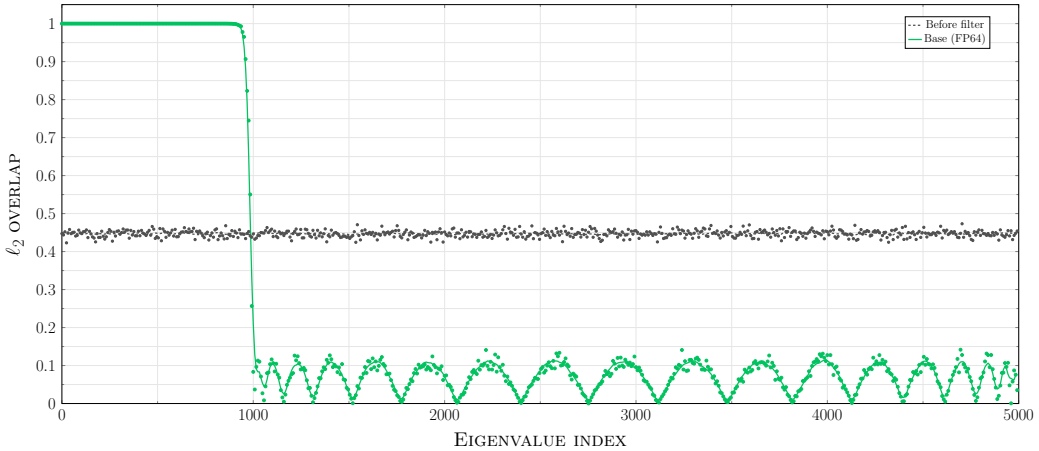
Figure : *Filter Time* vs. *Residual Tolerance*

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

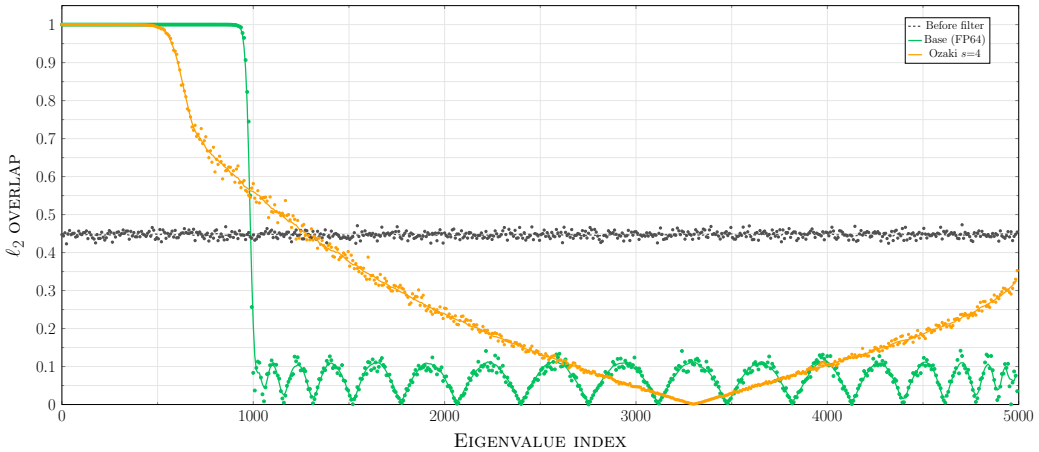


III. OZAKI-II SCHEME INTO CHASE



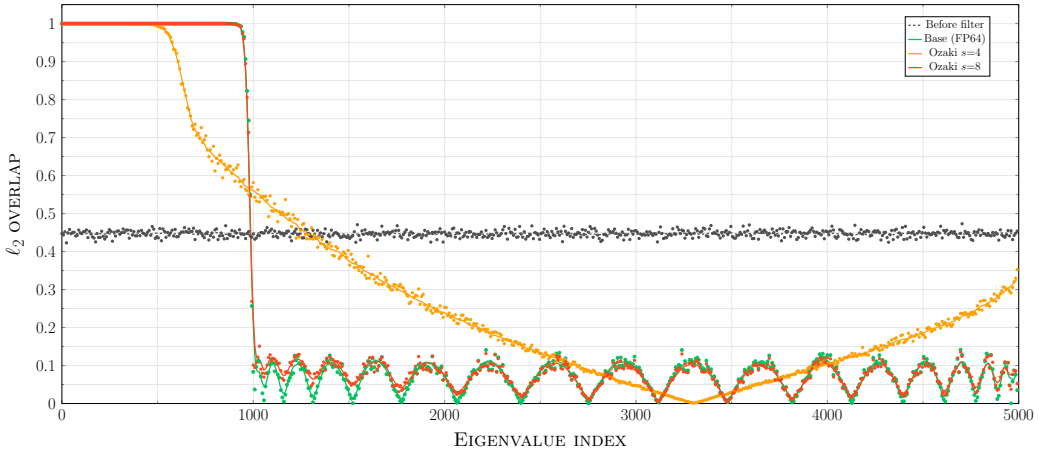
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE



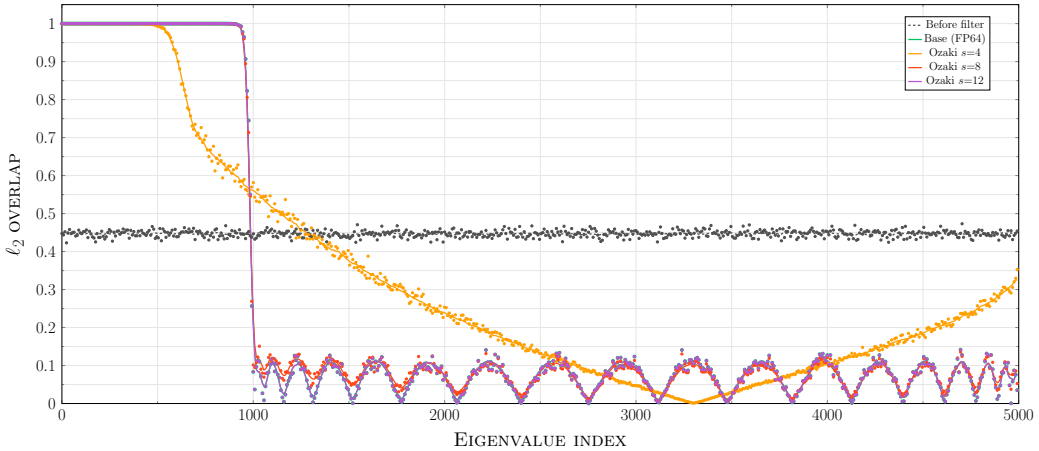
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE



JÜLICH Forschungszentrum

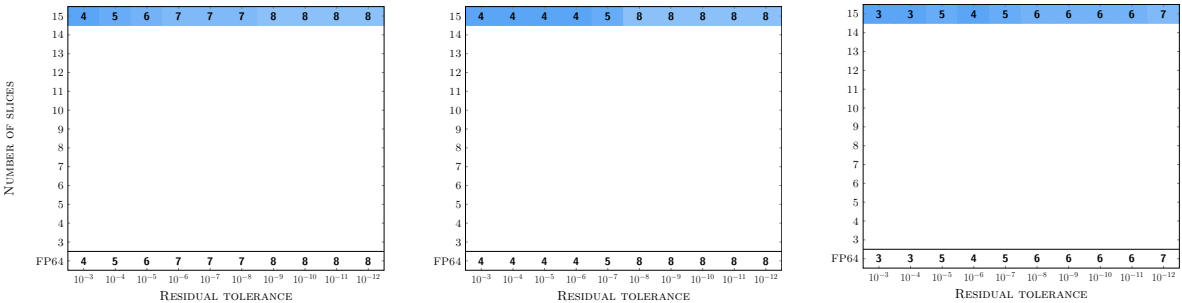
III. OZAKI-II SCHEME INTO CHASE



JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k

Medium Matrix - N = 62k

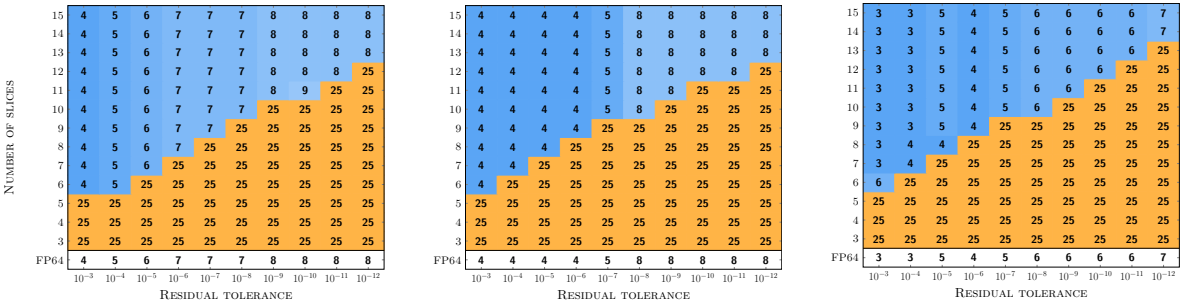
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k

Medium Matrix - N = 62k

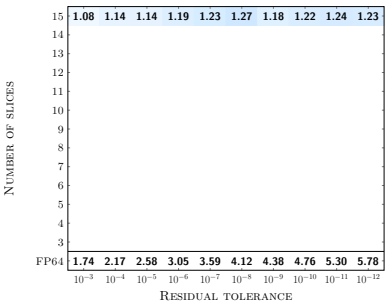
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

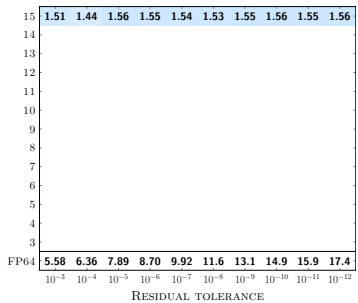
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

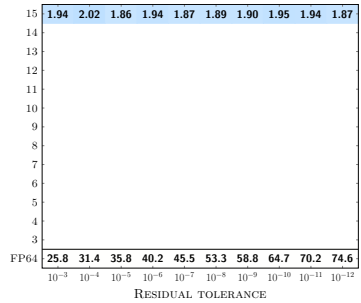
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



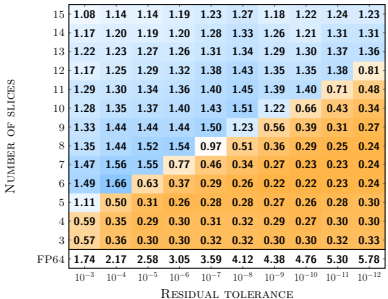
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

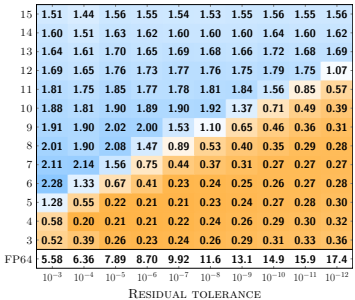
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

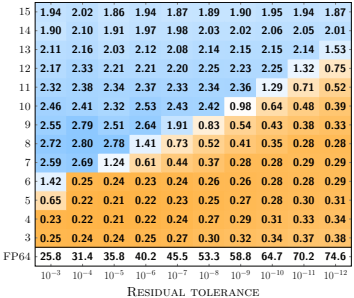
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



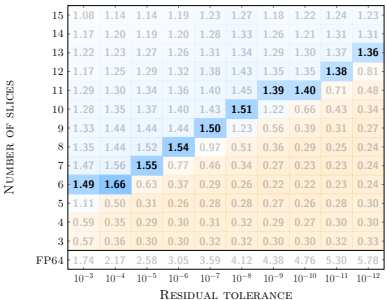
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

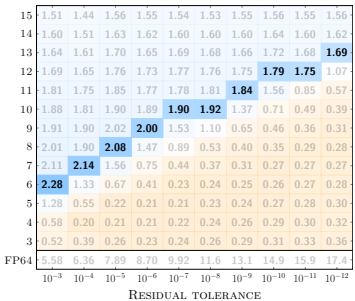
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

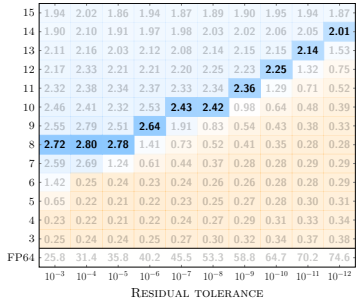
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



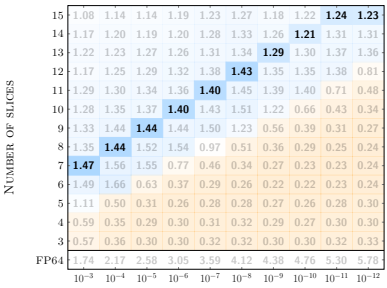
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

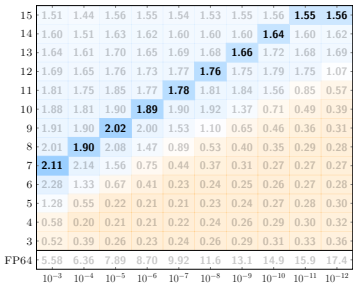
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

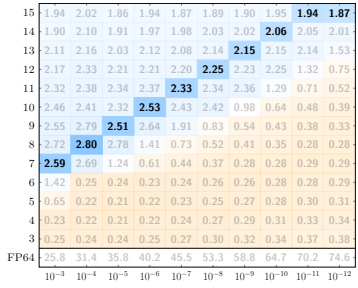
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



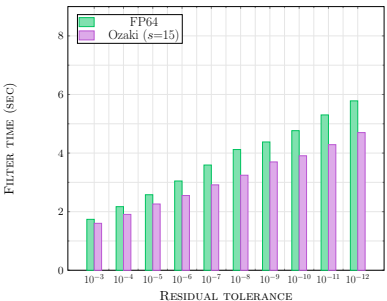
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

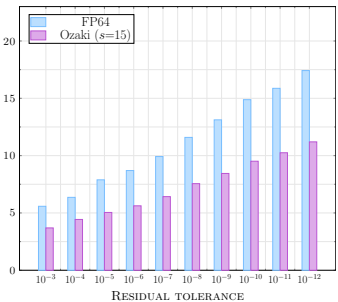
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

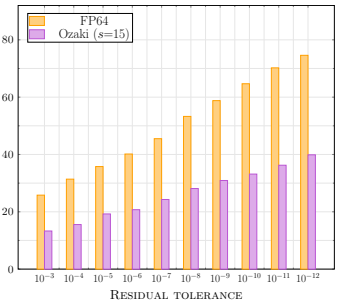
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k



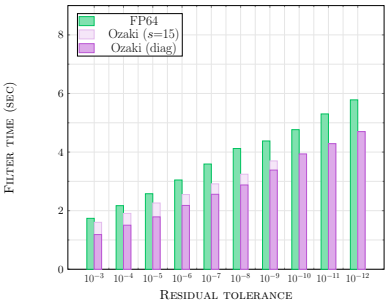
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

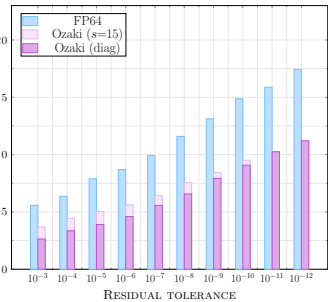
JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

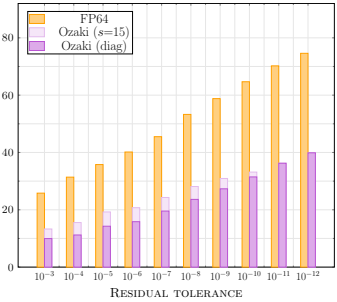
FP64 Experiments with 16 GH200 GPUs on JUPITER – NEV = 3% N, NEX = NEV/2



Small Matrix - N = 39k



Medium Matrix - N = 62k

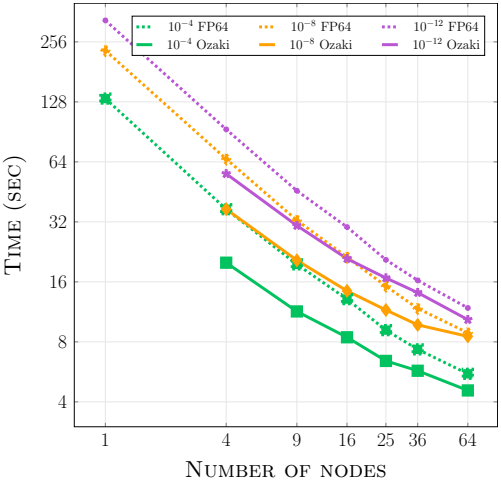


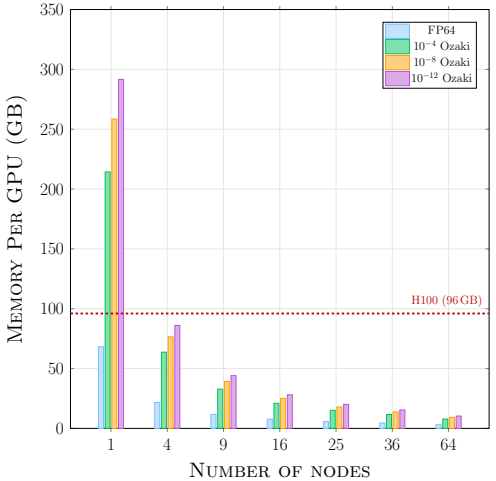
Large Matrix - N = 115k

Figure : *Filter Time* vs. *Residual Tolerance*

JÜLICH Forschungszentrum

III. OZAKI-II SCHEME INTO CHASE

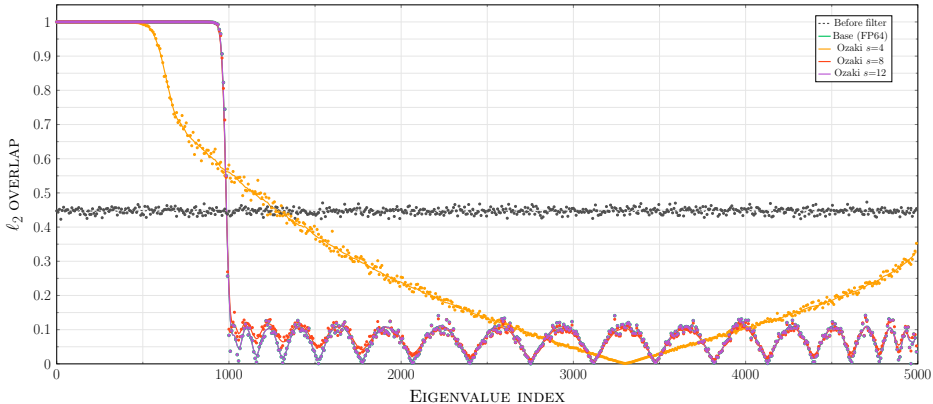




IV. BUILDING A THEORETICAL FRAMEWORK

Matrices are generated following

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad \text{DEGREE} = 20, \quad \phi = 0.0.$$

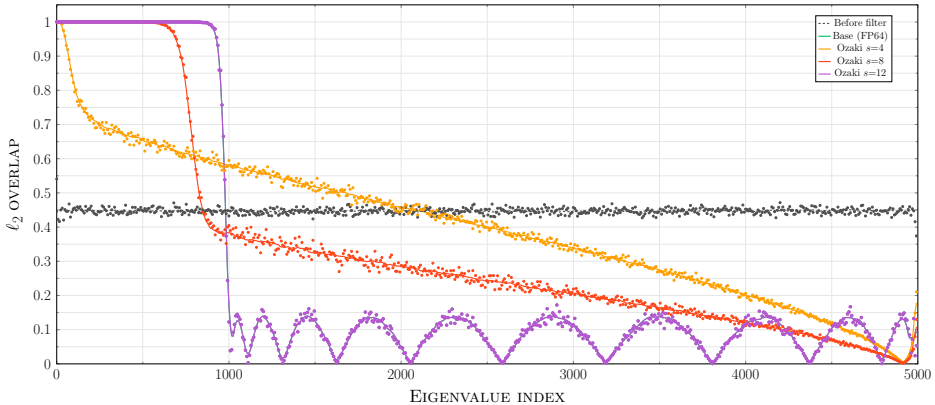


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Matrices are generated following

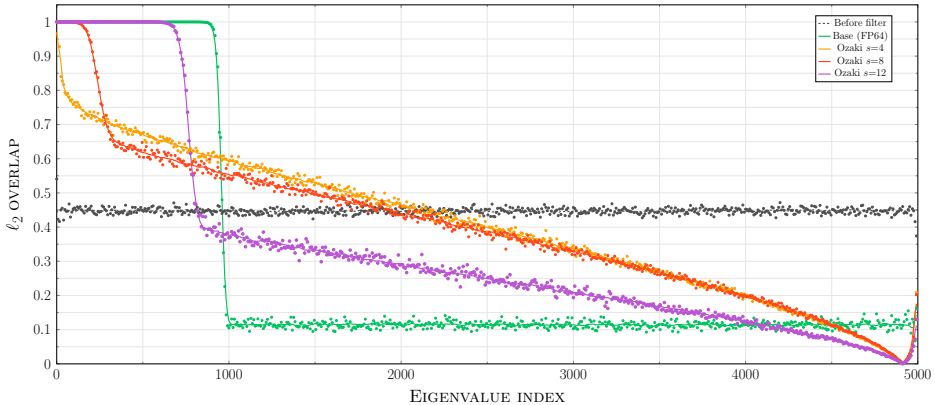
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad \text{DEGREE} = 20, \quad \phi = 1.5.$$



IV. BUILDING A THEORETICAL FRAMEWORK

Matrices are generated following

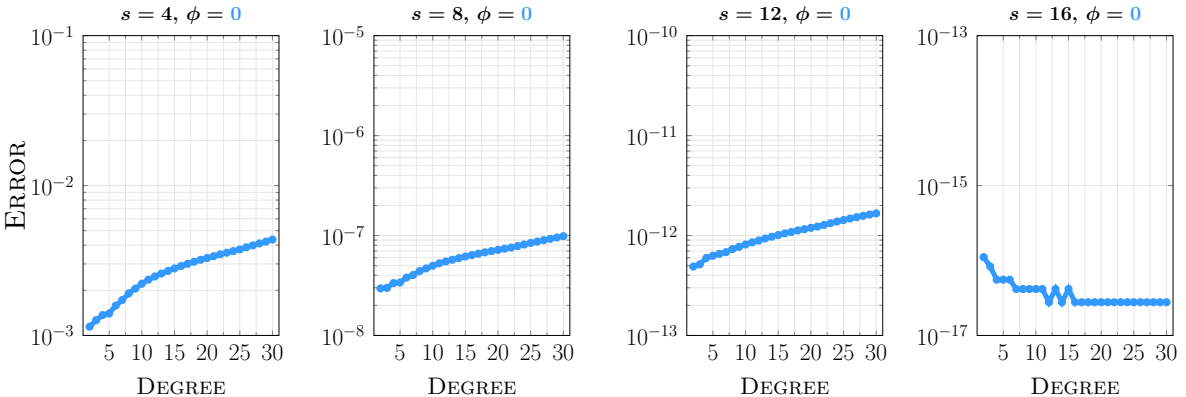
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad \text{DEGREE} = 30, \quad \phi = 1.5.$$



IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad E_{\text{deg}} = \text{FP128}(V_{\text{deg}}) - \text{Oz-II}(V_{\text{deg}}).$$

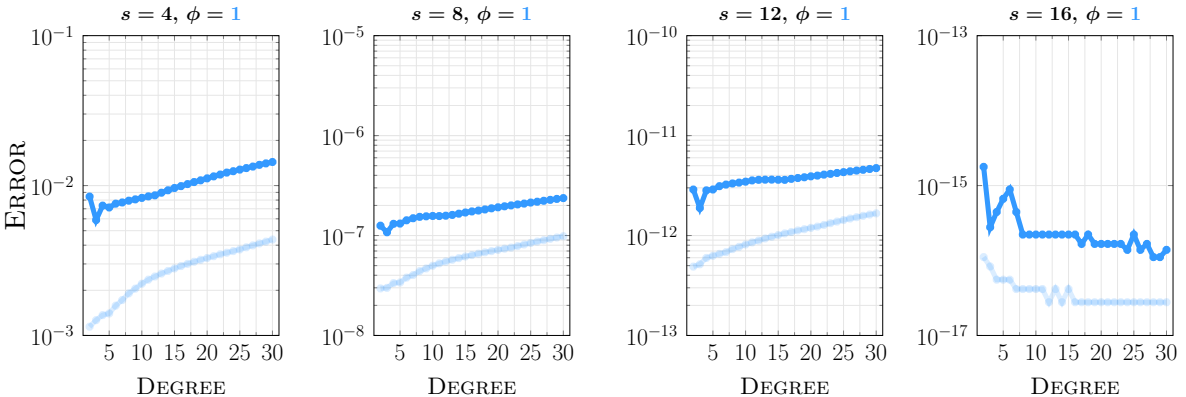


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad E_{\text{deg}} = \text{FP128}(V_{\text{deg}}) - \text{Oz-II}(V_{\text{deg}}).$$

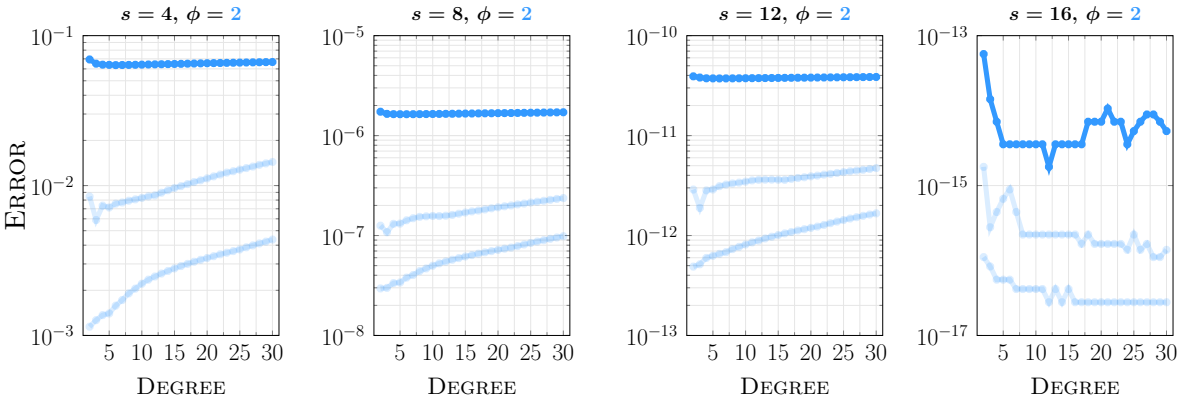


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad \text{with} \quad \alpha_{ij} \sim \mathcal{U}(0,1), \quad \beta_{ij} \sim \mathcal{N}(0,1) \quad | \quad E_{\text{deg}} = \text{FP128}(V_{\text{deg}}) - \text{Oz-II}(V_{\text{deg}}).$$



JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

In ChASE, the filtered-space is given by the three-terms recurrence relation

$$V_{i+1} = 2\sigma_{i+1}\bar{A}V_i - \sigma_{i+1}\sigma_i V_{i-1} \quad \text{with} \quad V_1 = \sigma_1 \bar{A} V_0 \quad \text{and} \quad V_0 = \text{RANDOM}(N, nev + nex)$$

JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

In ChASE, the filtered-space is given by the three-terms recurrence relation

$$V_{i+1} = 2\sigma_{i+1}\bar{A}V_i - \sigma_{i+1}\sigma_i V_{i-1} \quad \text{with} \quad V_1 = \sigma_1 \bar{A}V_0 \quad \text{and} \quad V_0 = \text{RANDOM}(N, nev + nex)$$

In practice, each GEMM operation introduces a floating-point error $E_j^f := \bar{A}\hat{V}_j - [\bar{A}\hat{V}_j]$, such that

$$\hat{V}_{i+1} = 2\sigma_{i+1}[\bar{A}\hat{V}_i] - \sigma_{i+1}\sigma_i \hat{V}_{i-1} \quad \text{with} \quad \hat{V}_1 = \sigma_1[\bar{A}V_0].$$

JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

In ChASE, the filtered-space is given by the three-terms recurrence relation

$$V_{i+1} = 2\sigma_{i+1}\bar{A}V_i - \sigma_{i+1}\sigma_i V_{i-1} \quad \text{with} \quad V_1 = \sigma_1 \bar{A}V_0 \quad \text{and} \quad V_0 = \texttt{RANDOM}(N, nev + nex)$$

In practice, each GEMM operation introduces a floating-point error $E_j^f := \bar{A}\hat{V}_j - [\bar{A}\hat{V}_j]$, such that

$$\hat{V}_{i+1} = 2\sigma_{i+1}[\bar{A}\hat{V}_i] - \sigma_{i+1}\sigma_i \hat{V}_{i-1} \quad \text{with} \quad \hat{V}_1 = \sigma_1[\bar{A}V_0].$$

This error propagates as the following non-homogeneous three-terms recurrence relation

$$\begin{aligned} E_{i+1}^p = V_{i+1} - \hat{V}_{i+1} &= 2\sigma_{i+1}(\bar{A}V_i - [\bar{A}\hat{V}_i]) - \sigma_{i+1}\sigma_i(V_{i-1} - \hat{V}_{i-1}) \\ &= 2\sigma_{i+1}(\bar{A}E_i^p + E_i^f) - \sigma_{i+1}\sigma_i E_{i-1}^p \\ &= 2\sigma_{i+1}\bar{A}E_i^p - \sigma_{i+1}\sigma_i E_{i-1}^p + 2\sigma_{i+1}E_i^f. \end{aligned}$$

JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

In ChASE, the filtered-space is given by the three-terms recurrence relation

$$V_{i+1} = 2\sigma_{i+1}\bar{A}V_i - \sigma_{i+1}\sigma_i V_{i-1} \quad \text{with} \quad V_1 = \sigma_1 \bar{A}V_0 \quad \text{and} \quad V_0 = \texttt{RANDOM}(N, nev + nex)$$

In practice, each GEMM operation introduces a floating-point error $E_j^f := \bar{A}\hat{V}_j - [\bar{A}\hat{V}_j]$, such that

$$\hat{V}_{i+1} = 2\sigma_{i+1}[\bar{A}\hat{V}_i] - \sigma_{i+1}\sigma_i \hat{V}_{i-1} \quad \text{with} \quad \hat{V}_1 = \sigma_1[\bar{A}V_0].$$

This error propagates as the following non-homogeneous three-terms recurrence relation

$$\begin{aligned} E_{i+1}^p = V_{i+1} - \hat{V}_{i+1} &= 2\sigma_{i+1}(\bar{A}V_i - [\bar{A}\hat{V}_i]) - \sigma_{i+1}\sigma_i(V_{i-1} - \hat{V}_{i-1}) \\ &= 2\sigma_{i+1}(\bar{A}E_i^p + E_i^f) - \sigma_{i+1}\sigma_i E_{i-1}^p \\ &= 2\sigma_{i+1}\bar{A}E_i^p - \sigma_{i+1}\sigma_i E_{i-1}^p + 2\sigma_{i+1}E_i^f. \end{aligned}$$

Let U_j be a Chebyshev Polynomial of the 2$^{\text{ND}}$ and of degree j. At degree DEG, the propagated error is given by

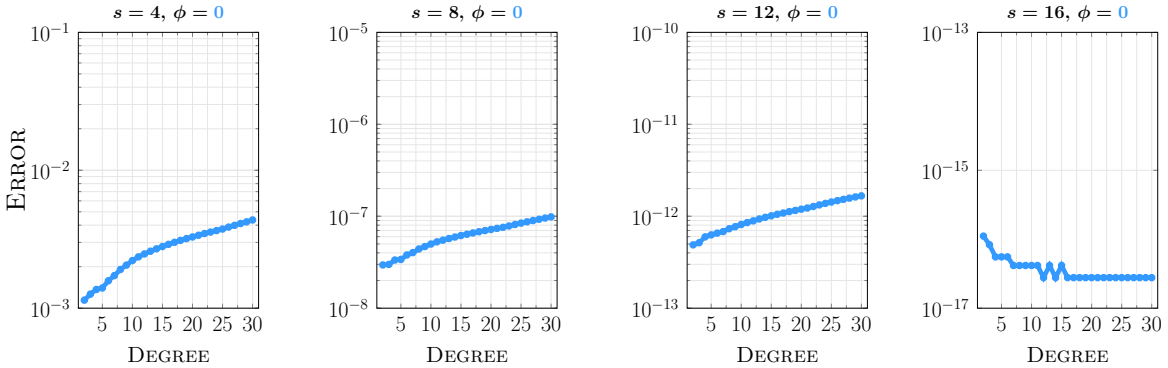
$$E_{\text{DEG}}^p = 2\sum_{j=0}^{\text{DEG}-1} \frac{\eta_j}{\eta_{\text{DEG}}} U_{\text{DEG}-1-j}(\bar{A})\, E_j^f$$

JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

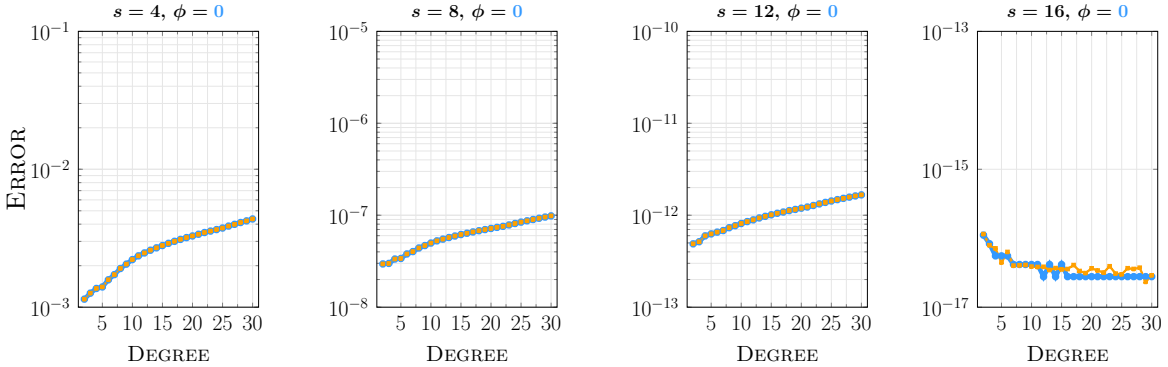
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad E_{\text{deg}} = \text{FP128}(V_{\text{deg}}) - \text{Oz-II}(V_{\text{deg}}).$$



IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

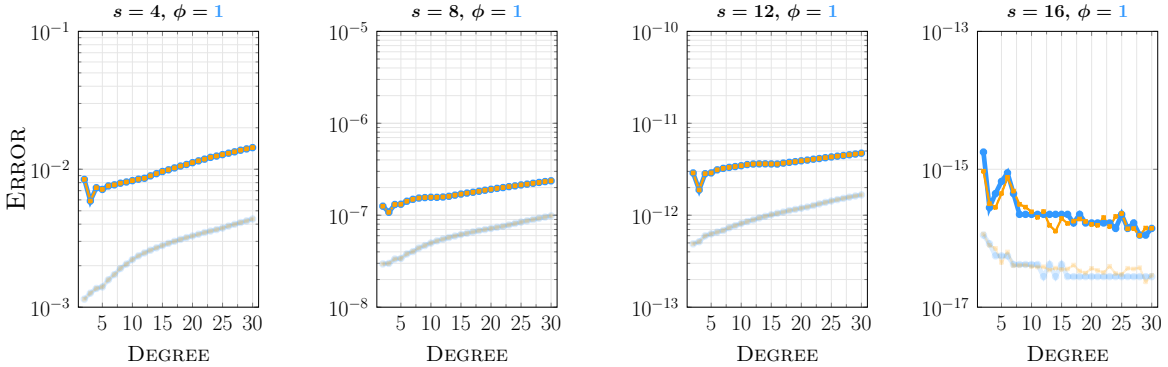
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad E^p_{\text{DEG}} = 2 \sum_{j=0}^{\text{DEG}-1} \frac{\eta_j}{\eta_{\text{DEG}}} U_{\text{DEG}-1-j}(\bar{A})\, E^f_j.$$



IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

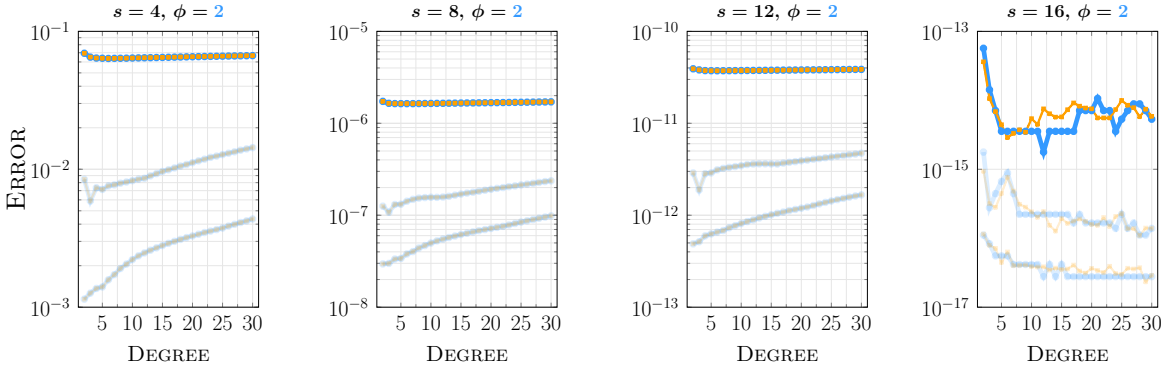
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad E^p_{\text{DEG}} = 2 \sum_{j=0}^{\text{DEG}-1} \frac{\eta_j}{\eta_{\text{DEG}}} U_{\text{DEG}-1-j}(\bar{A})\, E^f_j.$$



IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

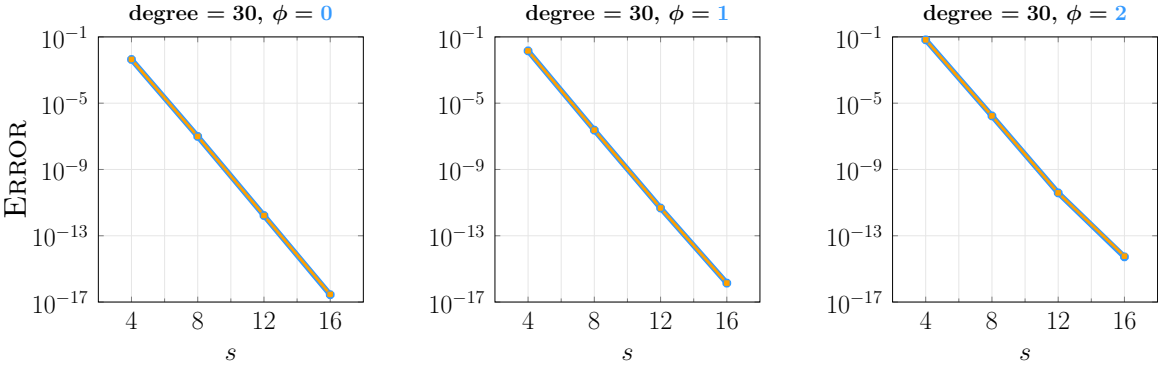
$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad E^p_{\text{DEG}} = 2 \sum_{j=0}^{\text{DEG}-1} \frac{\eta_j}{\eta_{\text{DEG}}} U_{\text{DEG}-1-j}(\bar{A})\, E^f_j.$$



IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad E^p_{\text{DEG}} = 2 \sum_{j=0}^{\text{DEG}-1} \frac{\eta_j}{\eta_{\text{DEG}}} U_{\text{DEG}-1-j}(\bar{A})\, E^f_j.$$

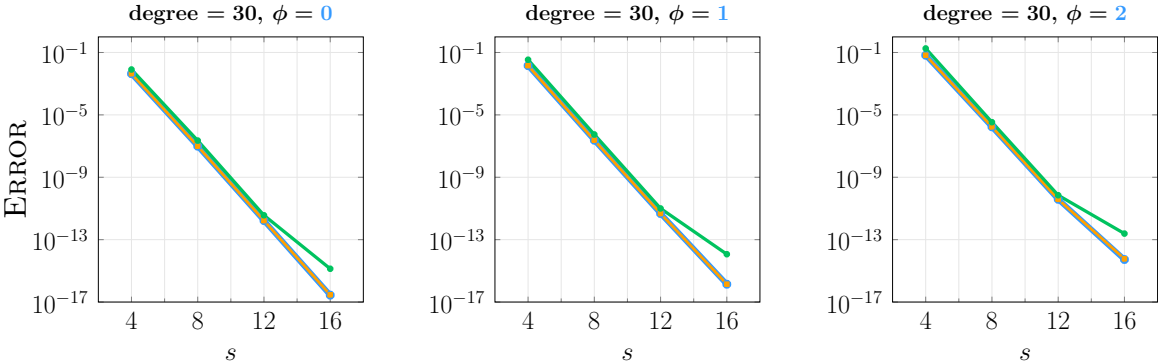


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad \left|E^p_{\text{DEG}}\right| \leq 2 \sum_{j=0}^{\text{DEG}-1} \left| \frac{\eta_j \nu_{\text{DEG}-j}}{\eta_{\text{DEG}}} E^f_j \right|.$$

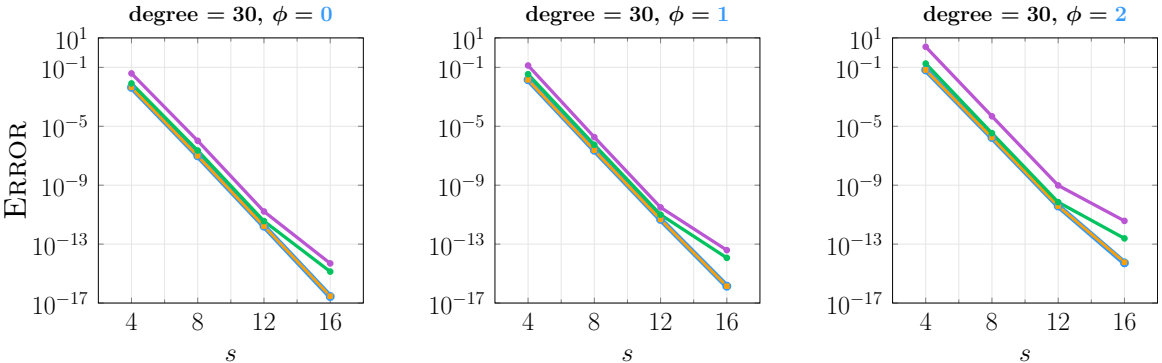


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad |E^p_{\text{DEG}}| \leq 2\left|E^f_0\right| \sum_{j=0}^{\text{DEG}-1} \left|\frac{\eta_j \nu_{\text{DEG}-j}}{\eta_{\text{DEG}}}\right|.$$

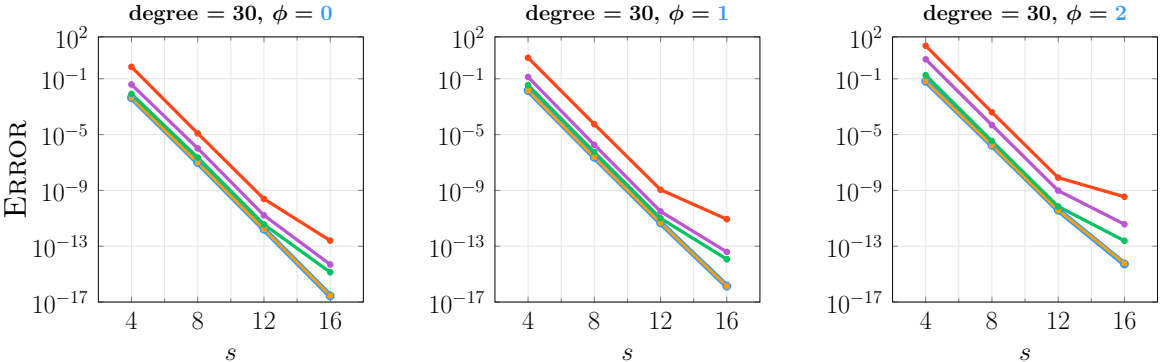


JÜLICH Forschungszentrum

IV. BUILDING A THEORETICAL FRAMEWORK

Let A be the input hermitian matrix and V the initial subspace, with $m = n = 1000$ and $nev + nex = 100$, such that

$$(A_{ij}|V_{ij}) = (\alpha_{ij} - 0.5) \times e^{\phi\beta_{ij}} \quad | \quad \left|E^{p}_{\mathrm{DEG}}\right| \leq 2\left(t|\bar{A}|\boldsymbol{v}\boldsymbol{b}^T + t\boldsymbol{a}\boldsymbol{v}^T|V| + (n + r_b)t^2\boldsymbol{a}\boldsymbol{b}^T\right)\sum_{j=0}^{\mathrm{DEG}-1}\left|\frac{\eta_j\nu_{\mathrm{DEG}-j}}{\eta_{\mathrm{DEG}}}\right| \text{ (derived from [1]).}$$



[1] : *Error Analysis of Matrix Multiplication Emulation Using Ozaki-II Scheme, Yuki Uchino, Katsuhisa Ozaki, Toshiyuki Imamura, 2026*

JÜLICH Forschungszentrum

CONCLUSION

Interconnecting Application Codes with NLA Libraries is essential for :

- Selecting a solver based on hardware configuration

JÜLICH Forschungszentrum

CONCLUSION

Interconnecting Application Codes with NLA Libraries is essential for :

- Selecting a solver based on hardware configuration
- Abstracting data redistribution when necessary

CONCLUSION

Interconnecting Application Codes with NLA Libraries is essential for :

- Selecting a solver based on hardware configuration
- Abstracting data redistribution when necessary
- Connecting the application error tolerance with compute precision

CONCLUSION

Interconnecting Application Codes with NLA Libraries is essential for :

- Selecting a solver based on hardware configuration
- Abstracting data redistribution when necessary
- Connecting the application error tolerance with compute precision

Building this bridge is one of the motivation of the ExaNLA initiative



First ExaNLA Gathering - Feb. 2025

