Environment expansion for Matrix Product States

Ian McCulloch (NTHU)
Jesse Osborne (UQ)

SQAI-NCTS Workshop
2024-03-25

國立清華大學
NATIONAL TSING HUA UNIVERSITY

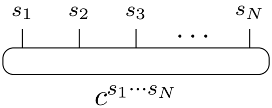
NSTC 國家科學及技術委員會
National Science and Technology Council

Overview

1. Density Matrix Renormalization Group
   - 2-site DMRG
   - Subspace expansion (3S)
2. Low-rank factorization via the Randomized SVD
3. Pre-expansion and post-expansion
   - Accelerating 2-site DMRG (pre-expansion)
   - Accelerating subspace expansion (post-expansion)
4. Benchmark example

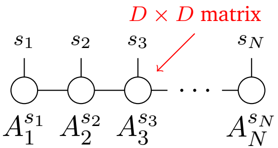
Introduction: DMRG

General many-body quantum state as a tensor product of d-dimensional local Hilbert spaces:

$|\Psi\rangle =$ 

, $\mathcal{O}(\exp N)$ DOFs.

Compress as a **matrix product state**:

$|\Psi\rangle =$ 

, $\mathcal{O}(ND^2)$ DOFs.

Good at representing locally entangled states.
Accuracy controlled by D.

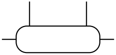
Density Matrix Renormalization Group

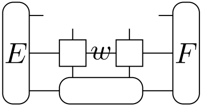
See Schollwöck, Annals of Physics 326, 96 (2011) for a review

doi:10.1016/j.aop.2010.09.012

The ‘classic’ algorithm is 2-site DMRG: update two sites of the tensor network at once.

One DMRG step: 2 sites, $Dd \times dD$ dimensional tensor

$|\Psi\rangle =$ 

$H\,|\Psi\rangle =$ 

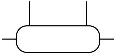
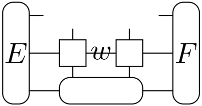
Cost of the tensor network contraction is $\mathcal{O}(wd^2D^3)$

Density Matrix Renormalization Group

See Schollwöck, Annals of Physics 326, 96 (2011) for a review
doi:10.1016/j.aop.2010.09.012

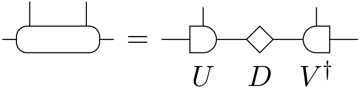
The ‘classic’ algorithm is 2-site DMRG: update two sites of the tensor network at once.

One DMRG step: 2 sites, $Dd \times dD$ dimensional tensor

$|\Psi\rangle =$  $H|\Psi\rangle =$ 

Cost of the tensor network contraction is $\mathcal{O}(wd^2D^3)$

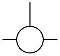
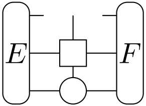
SVD to convert ψ back into MPS form:

$|\Psi\rangle =$ 

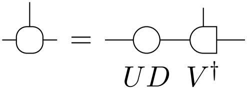
Cost of the SVD is $\mathcal{O}(d^3D^3)$

Why not update a single site?

- Much faster – $\mathcal{O}(wdD^3)$ for the matrix-vector multiply

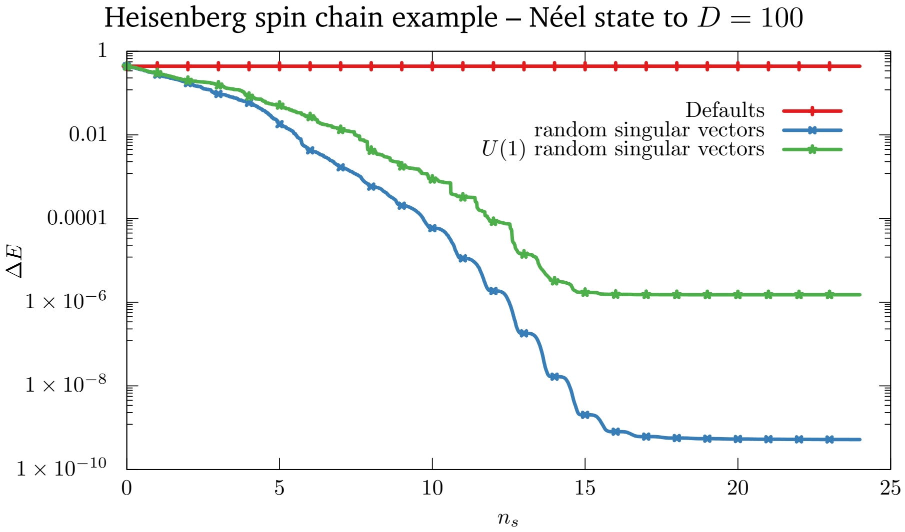
$|\Psi\rangle =$  $H\,|\Psi\rangle =$ 

- SVD is $\mathcal{O}(dD^3)$



- *Linear* scaling in the local Hilbert space dimension d
- Problem: how to increase the bond dimension? (random states?)

Single-site convergence

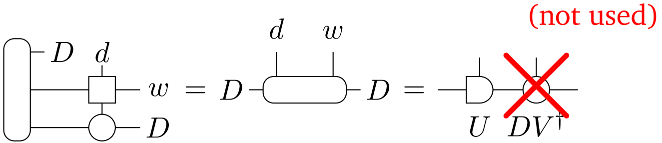


With $U(1)$ symmetry, the dimension in each symmetry sector is *frozen*

Single-site subspace expansion (3S)

C. Hubig, **IPM**, U. Schollwöck, and F. A. Wolf, Phys. Rev. B 91, 155115 (2015)
doi:10.1103/PhysRevB.91.155115

- Incorporate *new degrees of freedom* during the truncation using the environment Hamiltonian



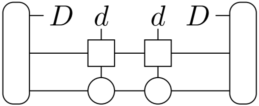
Cost of contraction is $\mathcal{O}(wdD^3)$. Cost of SVD is $\mathcal{O}(wd^2D^3)$.

Earlier method from Steven R White that uses an eigenvalue decomposition, $\mathcal{O}(d^3D^3)$ [Phys. Rev. B 72, 180403 (2005)]

3S – why does it work?

Easier to understand as a density matrix

$$\rho' = \sum_w E_w \rho E_w^\dagger = X X^\dagger =$$



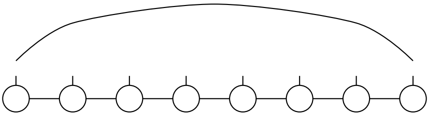
Perturb the density matrix using the block operators

- It is important to use appropriate weights
- One block operator is the identity
- The weight of the remaining components is the control parameter α
- Weight each component by the norm F_w matrix
- Omit the block Hamiltonian itself – don't need it
- The remaining set $\{E_w\}$ contains all interaction terms that are needed at the current site
  - They might not appear in the projected Hamiltonian itself!

Adding relevant degrees of freedom

- Long-range interactions

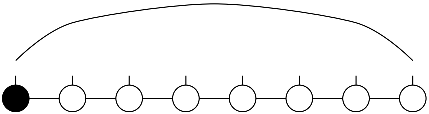
Example: periodic boundary conditions



Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

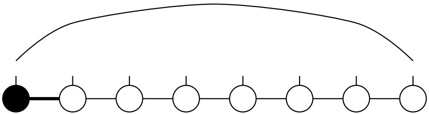


Matrix elements are zero at the active site

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

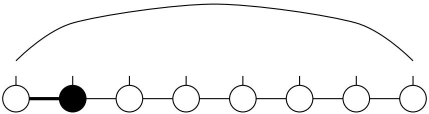


Relevant basis states at site 1 added via the mixing term

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

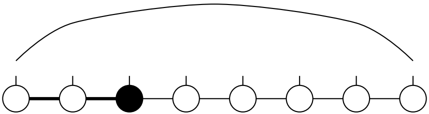


Relevant basis states are kept throughout the sweep

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

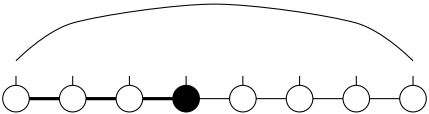


Relevant basis states are kept throughout the sweep

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

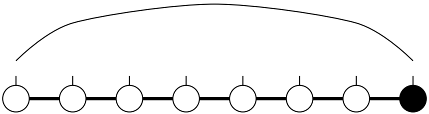


Relevant basis states are kept throughout the sweep

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

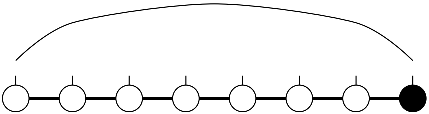


Now matrix elements are non-zero

Adding relevant degrees of freedom

- Long-range interactions

Example: periodic boundary conditions

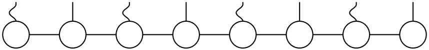


Now matrix elements are non-zero

- Inhomogeneous degrees of freedom

Example: fermions coupled to bosonic degrees of freedom

$$H = -t \sum_{<i,j>,\sigma} \left(c^\dagger_{i\sigma} c_{j\sigma} + \text{H.c.}\right) + U \sum_i n_{i\uparrow} n_{i\downarrow} + \omega \sum_i b^\dagger_i b_i + g \sum_i \left(n_{i\uparrow} + n_{i\downarrow} - 1\right) \left(b^\dagger_i + b_i\right)$$



2-site update cannot introduce new quantum number sectors

Low rank factorization: randomized SVD (RSVD)

N. Halko, P.-G. Martinsson, and J. A. Tropp, SIAM Rev. 53, 217 (2011)
doi:10.1137/090771806

- We only want a fraction $\sim 1/d$ of the singular vectors.
- Traditional algorithms calculate *all* singular values and truncate.

Low rank factorization: randomized SVD (RSVD)

N. Halko, P.-G. Martinsson, and J. A. Tropp, SIAM Rev. 53, 217 (2011)
doi:10.1137/090771806

- We only want a fraction $\sim 1/d$ of the singular vectors.
- Traditional algorithms calculate *all* singular values and truncate.

Range finding algorithm

Find (approximate) dominant k left singular values of a $m \times n$ matrix M

Construct $n \times k$ Gaussian random matrix Ω

QR decomposition:

$$QR = M\Omega$$

Q is an $n \times k$ matrix containing the dominant k (approximate) left singular vectors

- Cost is dominated by the matrix multiply $M\Omega$

Randomized SVD (RSVD)

Improving accuracy: *oversampling*

Randomized SVD

p is the *oversampling parameter* ($p = 10$ is typical)

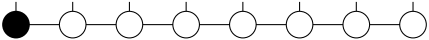
1. Construct Ω, $n \times (k + p)$ Gaussian random matrix
2. QR decomposition: $QR = M\Omega$ — Q is $n \times (k + p)$
3. Singular value decomposition $UDV^\dagger = Q^\dagger M$
4. Keep the k largest singular values — U is $(k + p) \times k$
5. QU is a good approximation to the dominant k singular values

Oversampling does not increase cost – dominated by multiplication $M\Omega$
Very good theoretical error bounds with modest oversampling
$p = 10$ works OK.

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

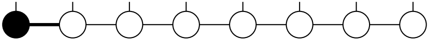


First site

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

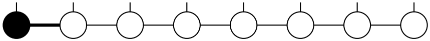


Expand the bond dimension

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

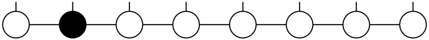


Optimize the site, then truncate

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

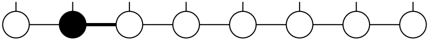


Next site

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

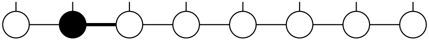


Expand the bond dimension

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

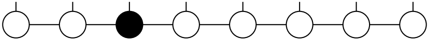


Optimize the site, then truncate

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

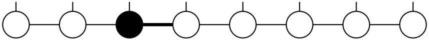


Next site

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are 'thrown away' afterwards

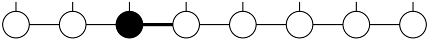


Expand the bond dimension

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

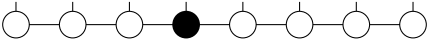


Optimize the site, then truncate

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

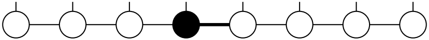


Next site

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are 'thrown away' afterwards

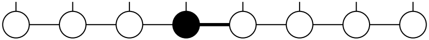


Expand the bond dimension

Pre-expansion versus post-expansion

We refer to two-site DMRG and related methods as *pre-expansion*

- Bond expansion *before* the optimization step
- Incorporate degrees of freedom from the environment
- These degrees of freedom are ‘thrown away’ afterwards

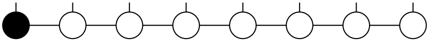


Optimize the site, then truncate

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

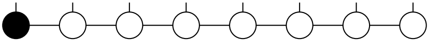


First site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

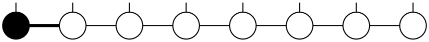


Optimize the site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

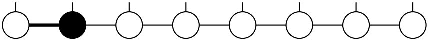


Truncate/expand the bond

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

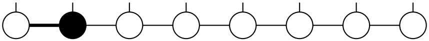


Next site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

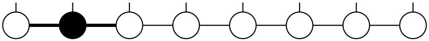


Optimize the site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

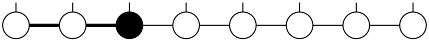


Truncate/expand the bond

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

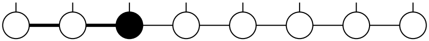


Next site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

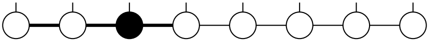


Optimize the site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

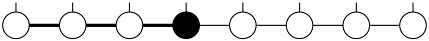


Truncate/expand the bond

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

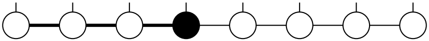


Next site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

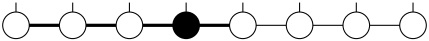


Optimize the site

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later

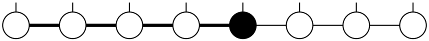


Truncate/expand the bond

Pre-expansion versus post-expansion

3S and related methods are *post-expansion*

- Bond expansion as part of the truncation step
- Introduce degrees of freedom unrelated to the optimization step
- Degrees of freedom added in the speculation that they may be useful later



Next site

Accelerating 2-site DMRG (pre-expansion)

- Kohn, Tschirsich,Keck, Plenio, Tamascelli, Mongangero, Phys. Rev. E 97, 013301 (2018) doi:10.1103/PhysRevE.97.013301
  2-site DMRG using RSVD, $\mathcal{O}(d^2 D^3)$ no longer a bottleneck for large d
- Andreas Gleis, Jheng-Wei Li, and Jan von Delft, Phys. Rev. Lett. 130, 246402 (2023) doi:10.1103/PhysRevLett.130.246402
  2-site DMRG with single site cost, overall $\mathcal{O}(dwD^3)$, but a total of 5 SVD's

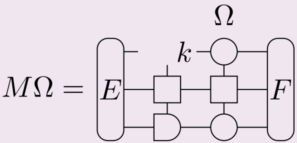
Basic idea:

- One iteration of 2-site DMRG, enlarge bond dimension by k
- Single-site DMRG in this $D + k$ dimensional basis

Overall cost for the optimization step is $\mathcal{O}(dw(D + k)^3)$

Accelerated 2-site DMRG

Simpler and faster contraction versus the method from Gleis *et al*

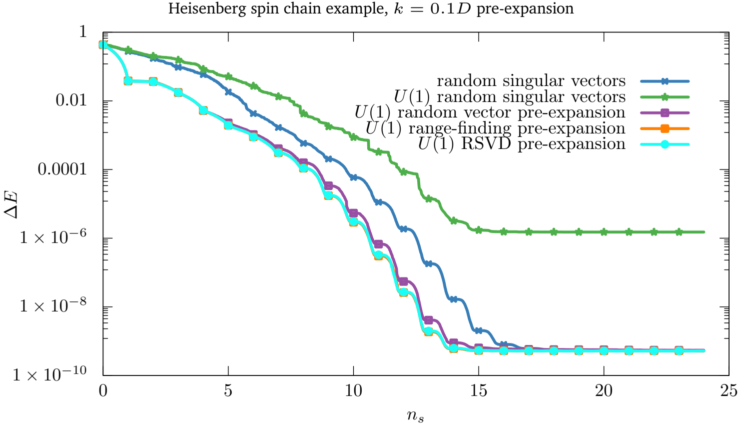


Contraction cost is $\mathcal{O}(dwkD^2)$. No $\mathcal{O}(D^3)$ operations.

1. Construct $dD \times k$ Gaussian random matrix Ω
2. Orthogonalize Ω against the states already in the environment basis
3. Insert this into the 2-site matrix-vector multiply
4. QR decomposition $QR = M\Omega$
5. Q is $dD \times k$ matrix of the expansion vectors
6. Augment the environment site and Hamiltonian

Pre-expansion variants

- Random – choose random expansion vectors
  - Same distribution of quantum number sectors as the existing states
  - Ensure at least one state in each available sector
- Range-finding – Use Q from $QR = M\Omega$ as the expansion vectors
- RSVD – Oversample the range-finding algorithm and SVD of $Q^\dagger M$

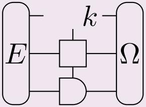


Post-expansion

Apply the RSVD ideas to the 3S algorithm

- Partition the states into two classes
  - D kept states
  - k additional *expansion vectors*
  - the MPS has $D + k$ states total
- Orthogonalize the expansion vectors against the kept states
  - only need k singular vectors of the big matrix, rather than D (or $D + k$)

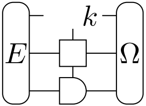
Accelerated 3S mixing

$M\Omega =$ 

Contraction cost is $\mathcal{O}(dwkD^2)$

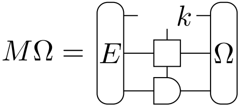
Weight the components Ω from the norm of the F matrices

Post-expansion

$M\Omega =$ 

Physical picture: we don't know (or don't trust) the F matrix elements – replace them with random numbers!

Post-expansion



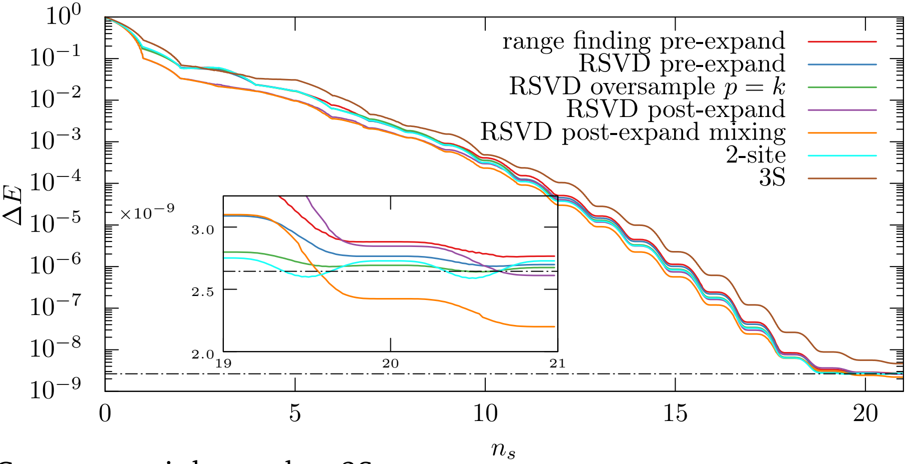
Physical picture: we don't know (or don't trust) the F matrix elements – replace them with random numbers!

Post-expansion variants

- Random – choose random expansion vectors
  - Usually not effective – similar to single-site with no singular value cutoff
- Range-finding – Use Q from $QR = M\Omega$ as the expansion vectors
  - No control over quantum number sectors
- RSVD – Oversample the range-finding algorithm and SVD of $Q^\dagger M$
- Mixing – merge the expansion vectors with a mixing factor
  - Set the mixing factor from the discarded weight of the truncation from $D + k$ to D – avoids the biggest problem of the old 3S algorithm

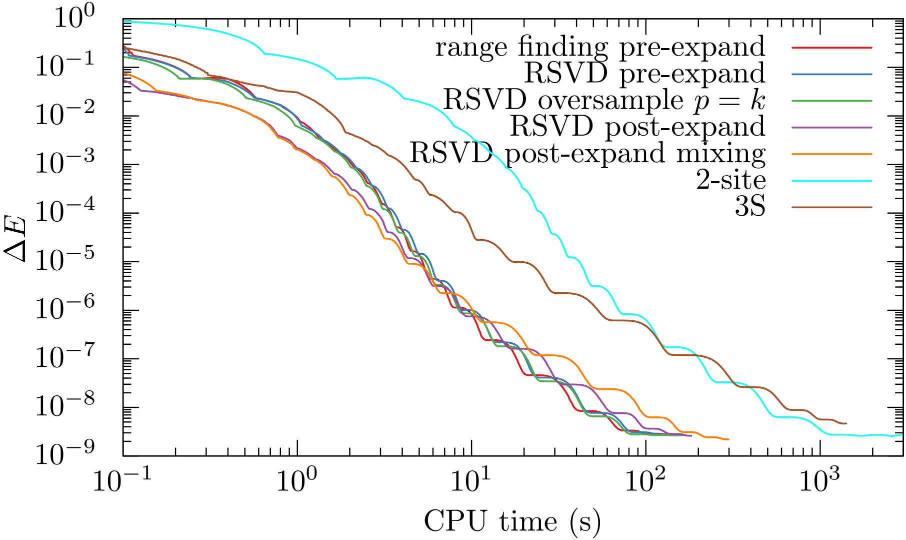
Benchmark – Hubbard-Hostein example

$$H = -t \sum_{<i,j>,\sigma} \left(c^\dagger_{i\sigma} c_{j\sigma} + \text{H.c.} \right) + U \sum_i n_{i\uparrow} n_{i\downarrow} + \omega \sum_i b^\dagger_i b_i + g \sum_i (n_{i\uparrow} + n_{i\downarrow} - 1) \left(b^\dagger_i + b_i \right)$$



Convergence is better than 3S

Hubbard-Hostein example – CPU time



Conclusion

Two bond expansion methods for DMRG

Pre-expansion:

- Similar convergence as 2-site DMRG
- Same performance as single-site DMRG; linear in d
- Bond expansion is *fast* – effectively zero cost

Post-expansion:

- Similar to 3S, good for inhomogeneous or long-range interactions
- Converges better than 3S
- Bond expansion is asymptotically faster than 3S (zero cost)

- More details at arXiv:2403.00562
- Long paper in preparation

Code available: `https://github.com/mptoolkit`
Documentation: `https://mptoolkit.qusim.net`