Exact Density Matrix Purification for Parallel Computations

Edmond Chow[*] and Jie Chen[†]
[*]School of Computational Science and Engineering, Georgia Institute of Technology
[†]IBM Research

Solving or Circumventing Eigenvalue Problems in Electronic Structure Theory
MolSSI Workshop and ELSI Conference
Richmond, VA, August 15–17, 2018

Density matrix purification

Given H and the chemical potential/Fermi level μ, the density matrix is

$$D = U \Lambda_D U^T$$

where $H = U \Lambda_H U^T$ and

$$(\Lambda_D)_{ii} = \begin{cases} 1 & \text{if } (\Lambda_H)_{ii} < \mu \\ 0 & \text{otherwise.} \end{cases}$$

Purification techniques compute D via the Heaviside function h,

$$D = h(\mu I - H)$$

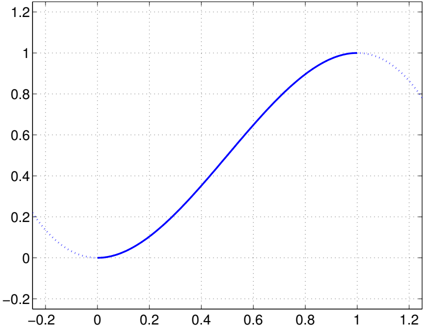
using (McWeeny) iterations of the form

$$D_{k+1} = 3D_k^2 - 2D_k^3, \quad D_0 = \text{scaled and shifted } H$$

such that D_0 has spectrum in $[0, 1]$ and μ is mapped to 0.5.

Density matrix purification

Since the iterations preserve the eigenvectors, we only need to examine what happens to the eigenvalues of D_k.



$$f(x) = 3x^2 - 2x^3$$

If $x > 0.5$, then $f(x) > x$
If $x < 0.5$, then $f(x) < x$

Stop the iterations when $\|D_k - D_k^2\|$ is small.

Motivation

Purification is based on matrix multiplication, which has more parallelism than eigendecomposition-based methods.

Purification with truncation is often used in linear scaling electronic structure methods.

For Hartree-Fock for molecules on massively parallel computers, we advocate using density matrix purification *without truncation.*

Note: convergence will depend on the size of the bandgap; also, purification requires more flops than eigendecomposition.

This talk: quantum chemistry context: focus on small/moderate sized problems and high accuracy.

Unknown chemical potential

Canonical purification (Palser-Manolopoulos 1998)
Trace-correcting purification (Niklasson 2002)

Polynomial varies from step to step to preserve the trace of D (number of occupied orbitals).

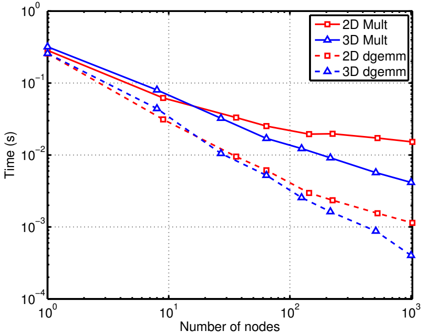
Canonical purification

```
Set $D_0$ appropriately, depending on number of occupied orbitals
for $k = 0, 1, \ldots$ until convergence do
    $c_k = \text{trace}(D_k^2 - D_k^3) / \text{trace}(D_k - D_k^2)$
    if $c_k \leq 1/2$ then
        $D_{k+1} = \left((1 - 2c_k)D_k + (1 + c_k)D_k^2 - D_k^3\right) / (1 - c_k)$
    else
        $D_{k+1} = \left((1 + c_k)D_k^2 - D_k^3\right) / c_k$
    end
end
```

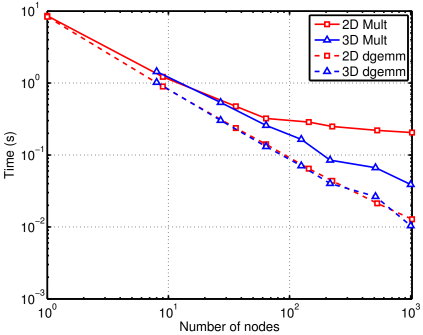
Distributed 2D and 3D matrix multiplication

SUMMA (van de Geijn and Watts 1997)

3D algorithm (Agarwal, Balle, Gustavson, Joshi, and Palkar 1995)



3555 basis functions

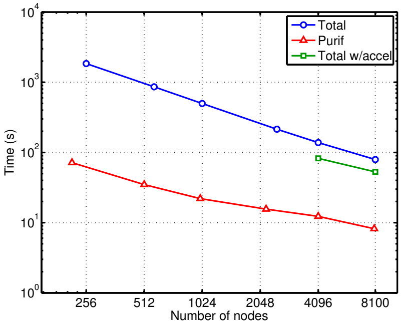


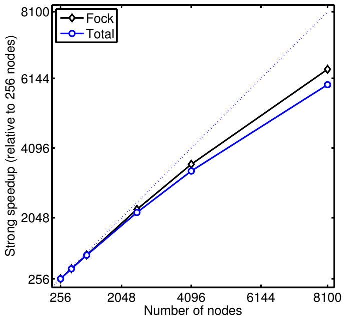
11163 basis functions

Bottleneck is communication for large numbers of nodes. 3D algorithm requires less communication.

(Chow, Liu, Smelyanskiy, Hammond, JCP, 2015)

SCF timings and speedup for 1hsg_180 problem on Tianhe-2





Protein-ligand system with cc-pVDZ

2938 atoms, 27394 basis functions

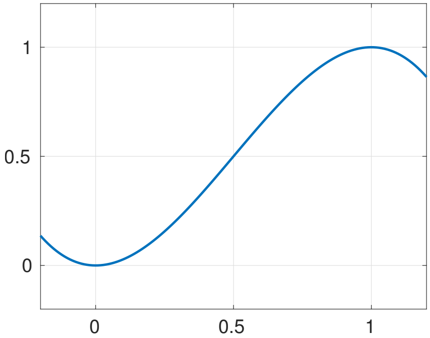
Tianhe-2: 24 CPU cores per node

Relative parallel efficiency at 8100 nodes: 73.5%

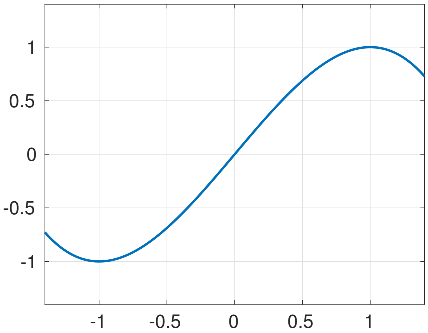
GTFock library: `https://github.com/gtfock-chem`

(Chow, Liu, Misra, et al., IJHPCA, 2015)

McWeeny purification is related to the Newton-Schulz method for computing the matrix sign function



McWeeny polynomial, $3x^2 - 2x^3$



Newton-Schulz polynomial, $\frac{1}{2}x(3 - x^2)$

$$X_{k+1} = \tfrac{1}{2} X_k (3I - X_k^2), \quad X_0 = \text{scaled and shifted } H$$

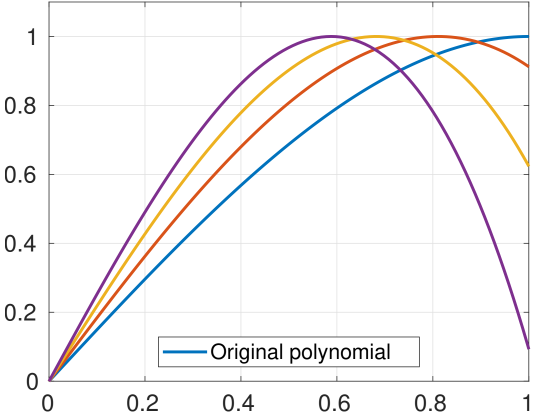
such that X_0 has spectrum in $[-1, 1]$ and μ is mapped to 0.

Stop the iterations when $\|I - X_k^2\|$ is small.

Non-monotonic purification polynomials

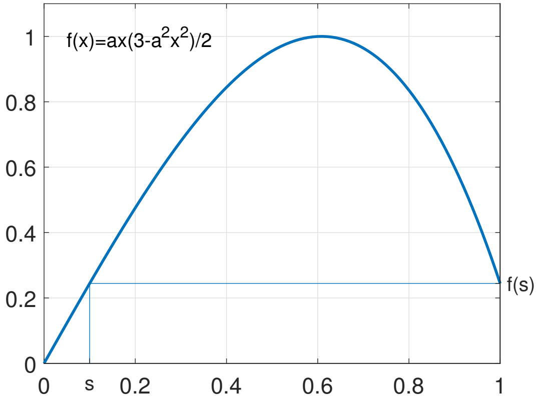
Pick a polynomial that is steeper for small magnitude eigenvalues.

- Rubensson 2011
- Suryanarayana 2013



Non-monotonic purification polynomials

The polynomial adapts as iterations progress so that the smallest eigenvalue is always mapped to a larger value *t* and no eigenvalue is mapped to a value smaller than *t*.



To find scale factor *a*, choose $f(1) = f(s)$ where *s* is a bound on the magnitude of the smallest magnitude eigenvalue of X_k.

The nonmonotonic purification methods are unstable!

Ref.: Nakatsukasa-Higham 2012 (for iterations for the sign function)

First define the backward error. This can be done using the polar decomposition.

The *polar decomposition* for a nonsingular A is

$$A = XS$$

where X is orthogonal and S is symmetric positive definite. If X is known, then $S = X^T A$. There exists a *Newton-Schulz* algorithm for computing X:

$$X_{k+1} = \tfrac{1}{2} X_k (3I - X_k^T X_k), \quad X_0 = A.$$

The backward error is

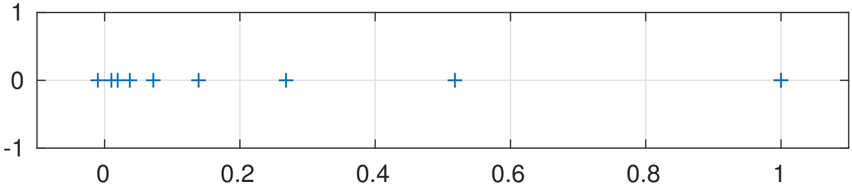
$$R = A - \hat{X}\hat{S}, \quad \hat{S} = \tfrac{1}{2}(\hat{X}^T A + (\hat{X}^T A)^T).$$

$\|R\| / \|A\|$ should be small when the algorithm converges if the algorithm is backward stable.

Test problems

Test problems have max eigenvalue at 1, and eigenvalues straddling zero at $-\delta$ and δ.

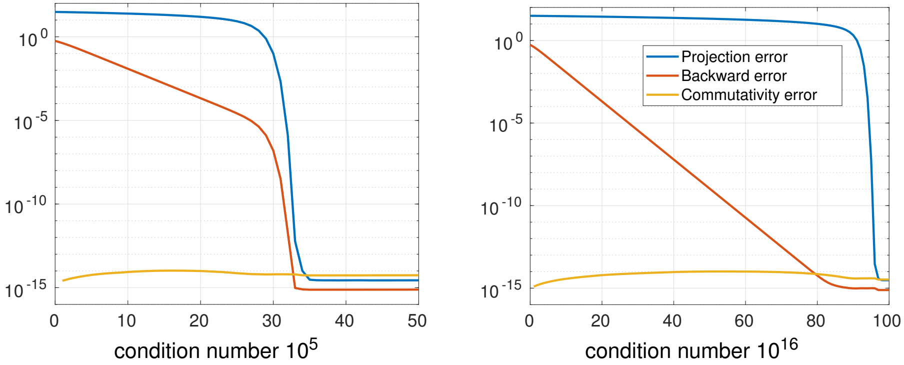
Example: $n = 10$, condition number $1/\delta = 100$.



Test matrix $A = Q\Lambda Q^T$, where Q is from the QR decomposition of a random matrix.

McWeeny purification (stable)

Convergence history for two test problems $n = 1000$



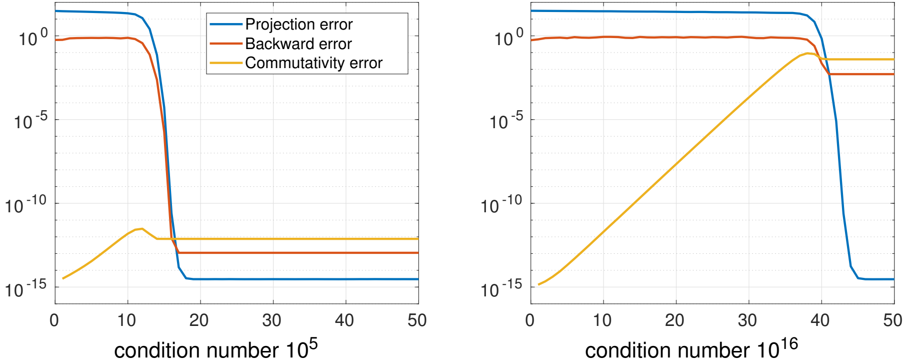
Projection error: $\|I - X^2\|_F$

Backward error: $\|A - \hat{X}\hat{S}\|_F, \quad \hat{S} = \frac{1}{2}(\hat{X}^T A + (\hat{X}^T A)^T)$

Commutativity error: $\|AX - XA\|_F$

Instability with nonmonotonic polynomials

Convergence history for two test problems $n = 1000$



condition number 10^5

condition number 10^{16}

Iteration count is approximately halved.

Backward error is large in the case of condition number 10^{16}.

Source of instability

Instability arises from large decreases when an eigenvalue is mapped by f (Nakatsukasa-Higham 2012). If f satisfies

$$\frac{f(\lambda_i)}{\|\hat{X}_k\|_2} \geq \frac{1}{d}\left(\frac{\lambda_i}{\|\hat{X}_k\|_2}\right), \quad \lambda_i \text{ is an eigenvalue of } \hat{X}_k$$

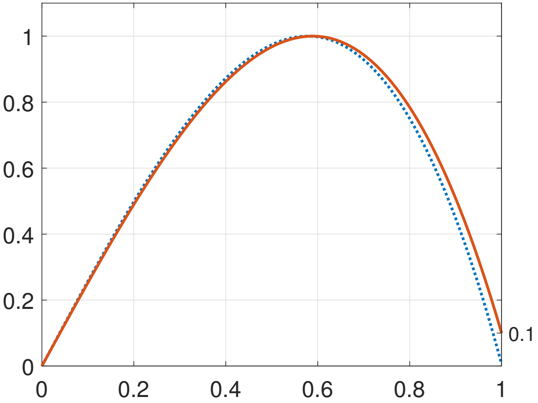
then

$$\hat{X}\hat{S} = A + d\varepsilon\|A\|_2, \quad \hat{S} = \tfrac{1}{2}(\hat{X}^T A + (\hat{X}^T A)^T).$$

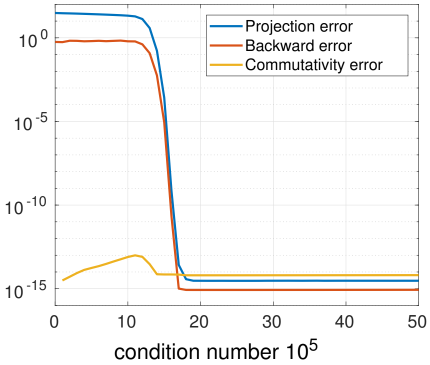
Instability does not depend on matrix size but depends on the eigenvectors, e.g., no instability if the eigenvectors are unit vectors.

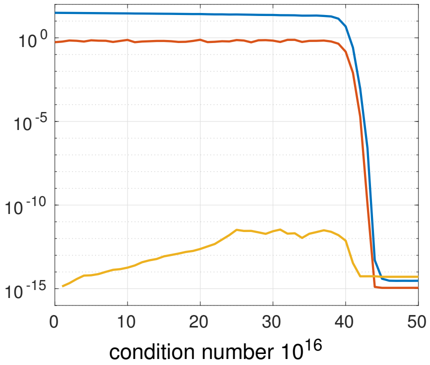
The instability can be ameliorated

Avoid large decreases by limiting the minimum value of $f(1)$ using a *threshold*, e.g., $f(1) = 0.1$.



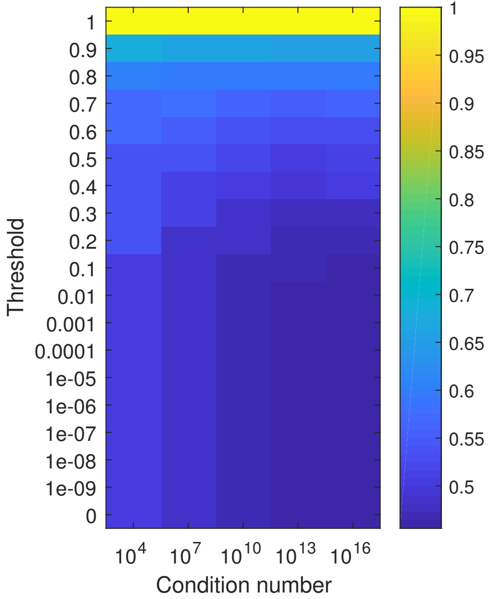
Stabilized nonmonotonic purification (threshold 0.1)



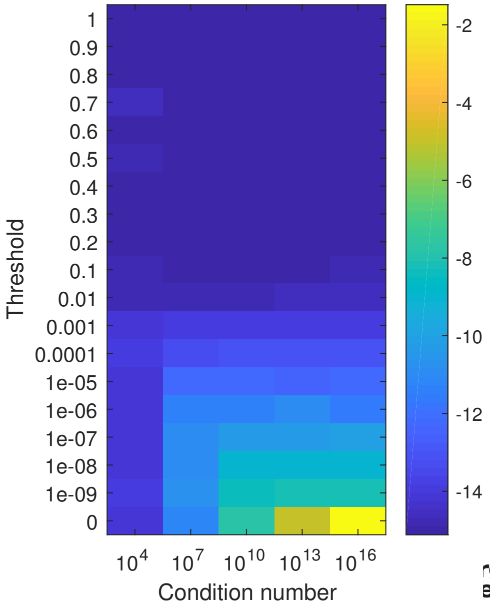


Convergence rate is minimally impacted.

Effect of different thresholds



Scaled iteration count



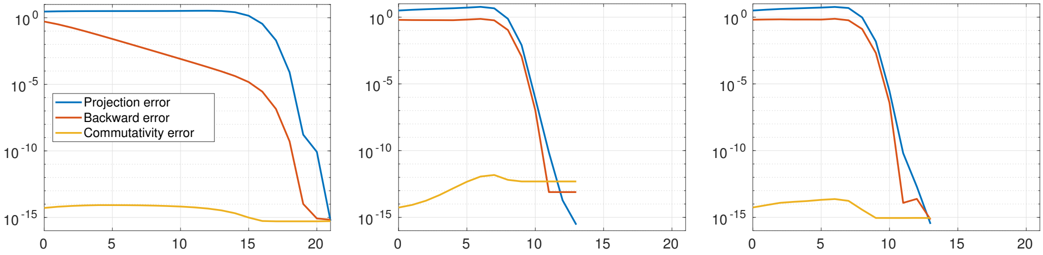
Log10 backward error

Iteration counts for different thresholds

	Condition number				
Threshold	1e4	1e7	1e10	1e13	1e16
1.0	28	45	62	79	96
0.9	19	30	41	52	63
0.8	17	27	37	47	57
0.7	16	26	35	44	54
0.6	16	25	33	42	51
0.5	15	24	32	40	49
0.4	15	23	31	39	48
0.3	15	23	30	38	46
0.2	15	22	30	37	45
0.1	14	22	29	37	44
0.01	14	22	29	36	44
0.001	14	22	29	36	44
0.0001	14	22	29	36	44
1e-5	14	22	29	36	44
1e-6	14	22	29	36	44
1e-7	14	22	29	36	44
1e-8	14	22	29	36	44
1e-9	14	22	29	36	44
0	14	22	29	36	44

Results for trace-correcting density matrix purification

Test example with condition number 10^5 and $n = 1000$



Standard

Unstable

Stabilized

Conclusions

- Density matrix purification without truncation is advocated for modestly-sized problems on highly parallel computers.
- Nonmonotonic density matrix purification is not stable, but instability is only evident in ill-conditioned cases and, in any case, is easy to ameliorate.
- Open question: effect of slight noncommutativity on outer SCF iterations?

Funding

NSF ACI grant

Intel Parallel Computing Center grant