# Supporting information - Efficient Hartree-Fock in real-space with projection operators

Nicholas M. Boffi, Manish Jain and Amir Natan\*

E-mail: amirnatan@post.tau.ac.il

## Effect of projection on $C_2H_6^+$ virtual states

We show here the effect of projection on the  $\alpha$  and  $\beta$  channels of  $C_2H_6^+$ . For both channels the error is usually below 0.001eV. This error is greater than the error shown in the manuscript as we have used a grid spacing of h=0.4a.u. compared to h=0.3a.u. in the manuscript main text. In Fig. 1 for the  $\alpha$  channel we show that when we do not project onto the virtual subspace the error in the first virtual eigenvalue is 0.4eV. Projecting on four states gives for the four virtual states an error that is below  $10^{-4}$ eV, and one can see that the error jumps in the fifth virtual state. When we project onto 40 virtual states we see very low error in all states. The same behavior is shown in Fig. 2 for the  $\beta$  channel. Here we can see that when we do not project onto the virtual subspace, the error in the first virtual states.



Figure 1: Eigenstate error (compared to the full operator) for the  $C_2H_6^+ \alpha$  channel. Blue circles show the error for projection onto zero virtual states, black diamonds show the error for projection onto four virtual states, and red larger circles show error for projection onto forty virtual states. The vertical dashed line shows the highest occupied state.



Figure 2: Eigenstate error (compared to the full operator) for the  $C_2H_6^+$   $\beta$  channel. Blue circles show the error for projection onto zero virtual states, black diamonds show the error for projection onto four virtual states, and red larger circles show the error for projection onto forty virtual states. The vertical dashed line shows the highest occupied state.

## Effect of projection on PBE0 $C_{60}$ virtual states

We show here the effect of the projection on a hybrid PBE0 calculation for the virtual states of  $C_{60}$ . The coordinates are the ones that were used in the main text of the manuscript. We compare calculations for projection onto zero, five, ten, and twenty virtual states with a 6-311G<sup>\*\*</sup> basis set calculation that was performed with the NWCHEM code. We demonstrate that one can progress to higher virtual states in the DOS by increasing the number of projected states. Zero projected states clearly does not capture any of the peaks. Five projected states captures only the first peak, while ten projected states captures the first two peaks. Finally, twenty projected states describes all three negative peaks in the DOS for the virtual states.



Figure 3:  $\rm C_{60}$  PBE0 virtual states Density of States (DOS) with different number of projected virtual states

#### Matrix-Vector operations scaling with different solvers

In Appendix B of the main text we assumed that we have  $P_0$  Matrix-Vector operations  $(\hat{H}\psi)$ per eigenvalue. In this section we show the behavior for both the ARPACK and Chebyshev-Davidson solvers. In figure 4 we show the number of operations to get all the eigenvalues and it is easy to see an almost linear behavior.



Figure 4: Number of  $\hat{H}\psi$  operations to get all eigenvalues as a number of the calculated eigenvalues. Blue asterisks show Chebyshev-Davidson data, red circles show ARPACK data.

We can estimate  $P_0$  by plotting the number of operations per eigenstate. This is shown

in the next figure where we show the number of operation per eigenstate which is exactly the  $P_0$  that we used in the main text.



Figure 5: Number of  $\hat{H}\psi$  operations per eigenstate. Blue asterisks show Chebyshev-Davidson data, red circles show ARPACK data.

It can be seen from Figure 5 that for large molecules the value of  $P_0$  approaches 15 for the ARPACK solver, the Chebyshev-Davidson solver is slightly lower but can have oscillations. This leads to more matrix-vector operations with the ARPACK solver and in large molecules we have 10-20 percent longer run times. In Small molecules the difference is dramatic and the Chebychev-Davidson can be almost 10 times faster than ARPACK.

### Additional performance analysis

We can take the data of Figure 6 of the main text and combine it with the number of matrixvector operations to get the values of  $T_{poisson}$  and  $T_{Hproj}$  from Appendix B. To get  $T_{poisson}$ we note that if we have N occupied states and we use additional M virtual states, then the total time for Fock preparation stage,  $T_{prep_{-}fock}$ , is:

$$T_{prep-fock} = \left(\frac{N(N+1)}{2} + N \cdot M\right) T_{poisson} \tag{1}$$

We can also use the assumption  $N \times T_{Poisson} \gg T_{HHartree}$  to write the time of a single diagonalization of the full operator,  $T_{full\_single\_diag}$  as:

$$T_{full\_single\_diag} \sim P_0 \times N \times N \times T_{poisson} \tag{2}$$

With the combination of Eq. 1 and Eq. 2 we can estimate the full operator diagonalization time direction from the Fock preparation time. This is shown in Figure:



Figure 6: Additional timing information, diagonalization time for the full opearator (solid blue line with asterisks) and the projected operator (dashed red line with circles). Fock prepation time is shown with cyan diagonalization time (Eq. 2 with  $P_0 = 15$ ) is shown with black squares and a dashed line.  $T_{poisson}$  is shown with cyan stars and a dashed line.  $T_{Hproj}$  is shown with red x symbols and a dashed line.

## Performance of the $\hat{K} \cdot \hat{P}$ representation

In this section we show the performance of the  $\hat{K} \cdot \hat{P}_M$  representation. With ARPACK we did not manage to get convergence - instead we get oscillations around the correct value (We show this later for H<sub>2</sub>), this makes sense as regular usage of ARPACK requires a Hermitian operator and this representation is non-hermitian. With Chebyshev-Davidson we manage to get convergence with this representation but it requires more orbital SCF iterations and

hence more Fock preparation sycles. Although a single diagonalization is slightly faster, there are also more diagonalizations that are required. Overall we get a roughly 1.5-2 times slower calculation as a result of those factors.



Figure 7:  $\hat{K} \cdot \hat{P}_M$  comparison for the number of orbital SCF cycles. All calculations were performed with the Chebyshev-Davidson eigensolver.

We demonstrate the lack of convergence of the ARPACK solver (for this non-hermitian representation) in Figure 8.



Figure 8: Comparison of the convergence of  $H_2$  HOMO for the  $\hat{K} \cdot \hat{P}$  representation with ARPACK (red circles) and Chebyshev-Davidson (blue asterisks). We use freely the term "SCF cycle", one can see that the Chebyshev-Davidson solver completed 4 outer SCF cycles that converge, while ARPACK does not converge at all.