Supplemental Information for The Role of High-Order Electron Correlation Effects in a Model System for Non-valence Correlation-bound Anions

Shiv Upadhyay, Amanda Dumi, James Shee, Kenneth D. Jordan

September 22, 2020

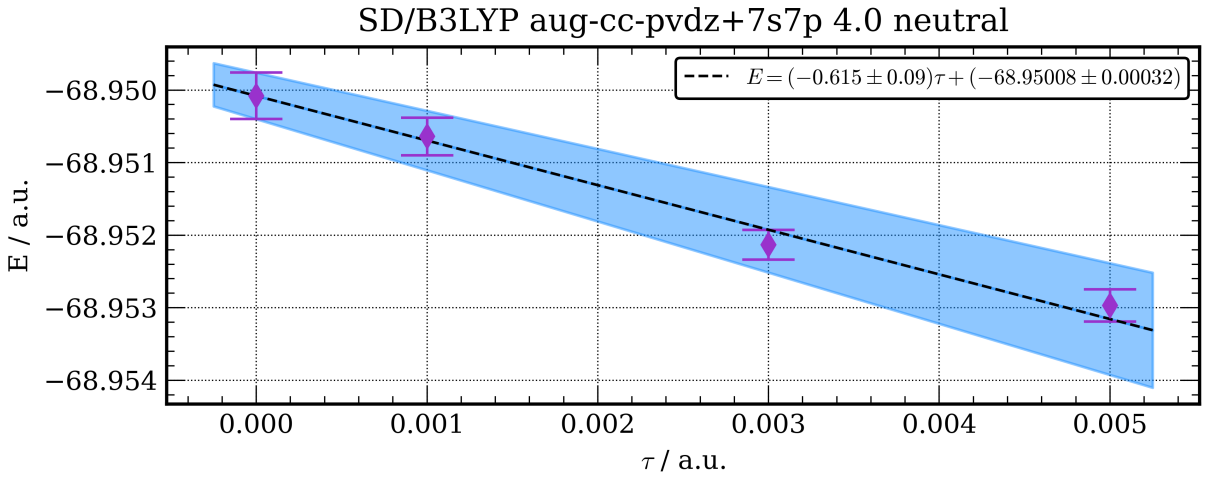
Contents

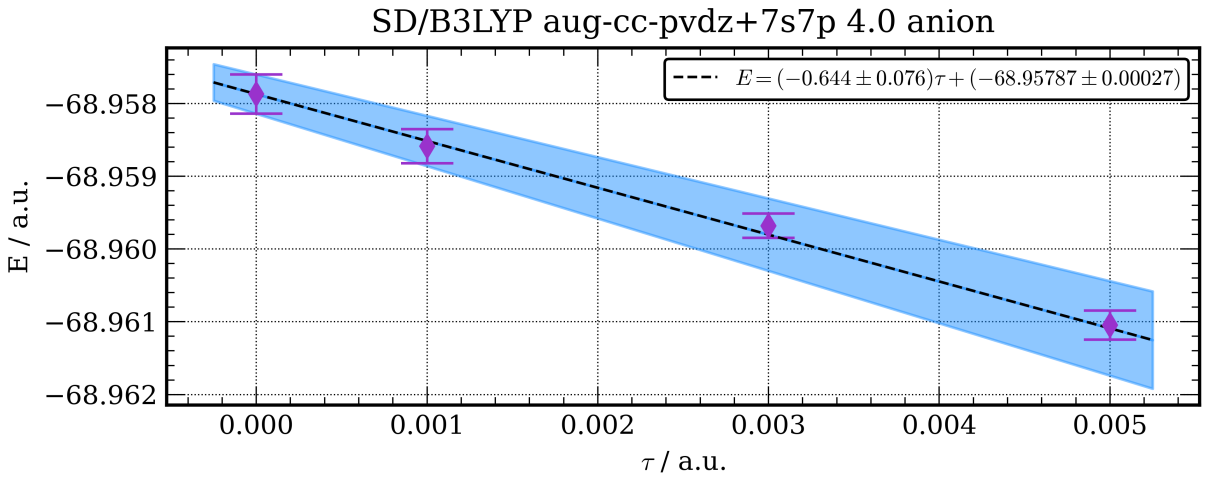
1 Supplemental Basis Functions

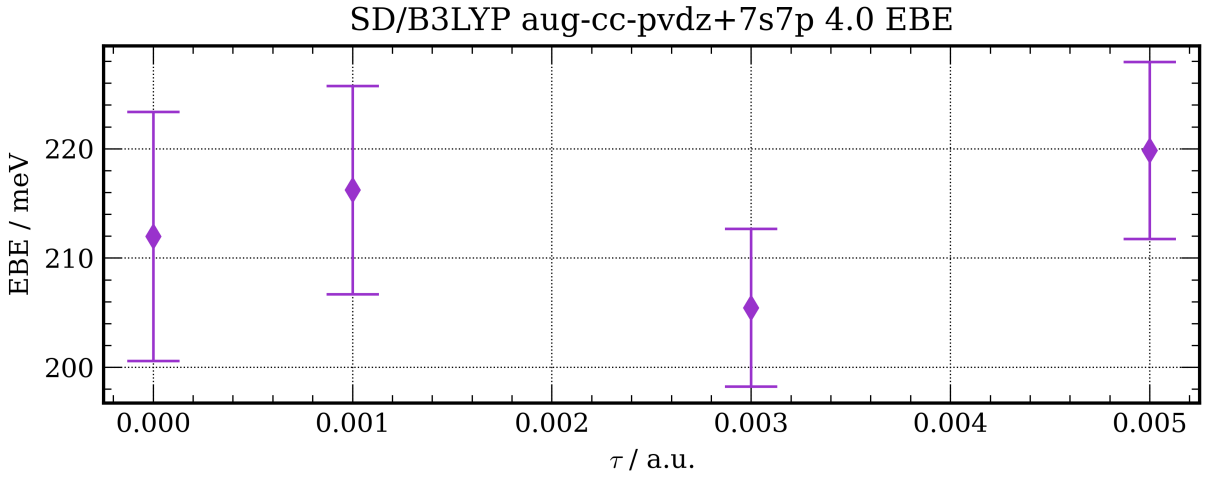
```
S   1
  1   0.02362232    1.0
S   1
  1   0.00738198    1.0
S   1
  1   0.00230687    1.0
S   1
  1   0.00072090    1.0
S   1
  1   0.00022528    1.0
S   1
  1   0.00007040    1.0
S   1
  1   0.00002200    1.0
P   1
  1   0.02362232    1.0
P   1
  1   0.00738198    1.0
P   1
  1   0.00230687    1.0
P   1
  1   0.00072090    1.0
P   1
  1   0.00022528    1.0
P   1
```

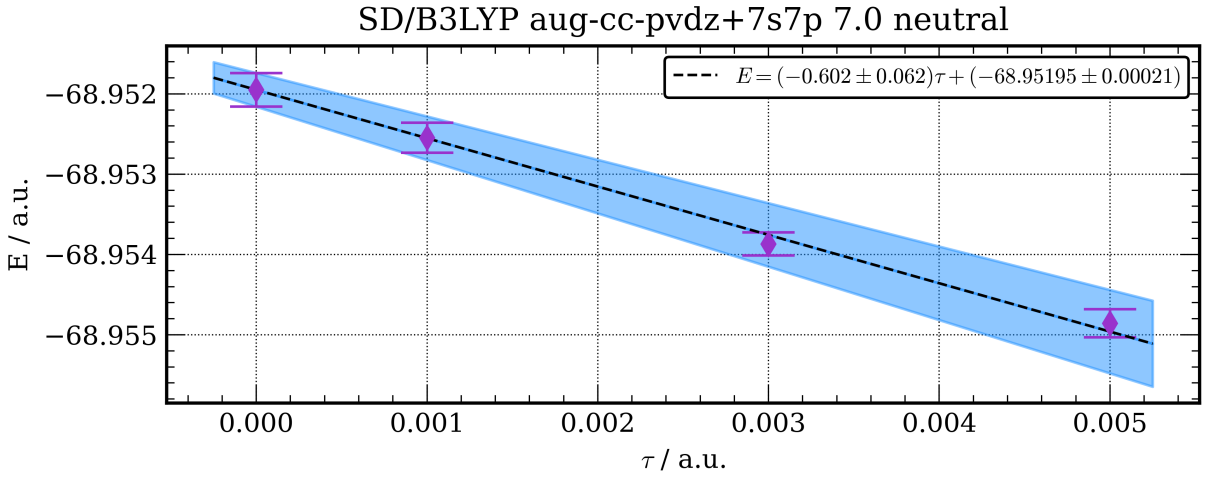
```
  1   0.00007040    1.0
P   1
  1   0.00002200    1.0

S   1
  1   0.02362232    1.0
S   1
  1   0.00738198    1.0
S   1
  1   0.00230687    1.0
P   1
  1   0.02362232    1.0
```

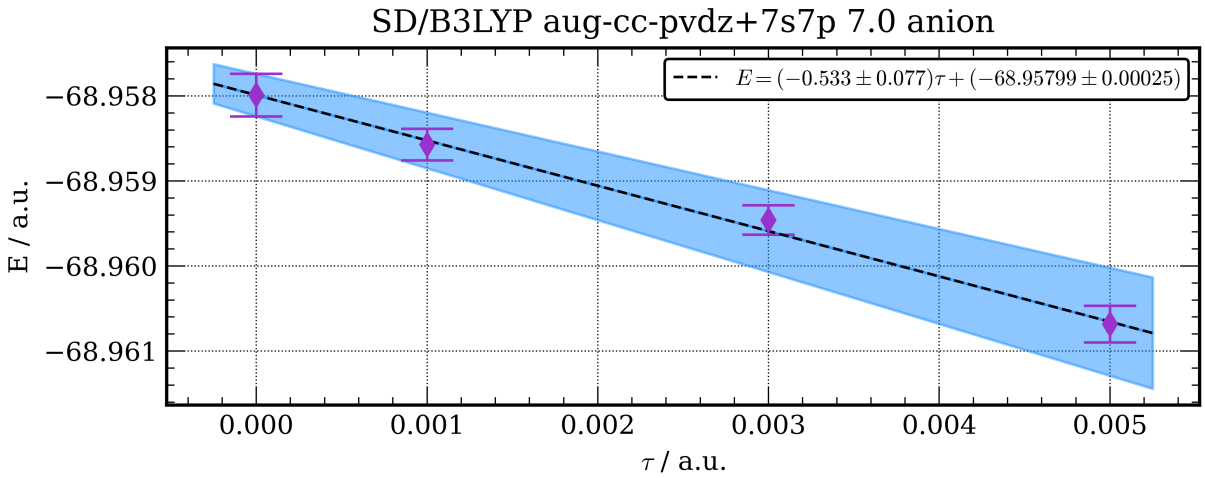
2 DMC EXTRAPOLATION

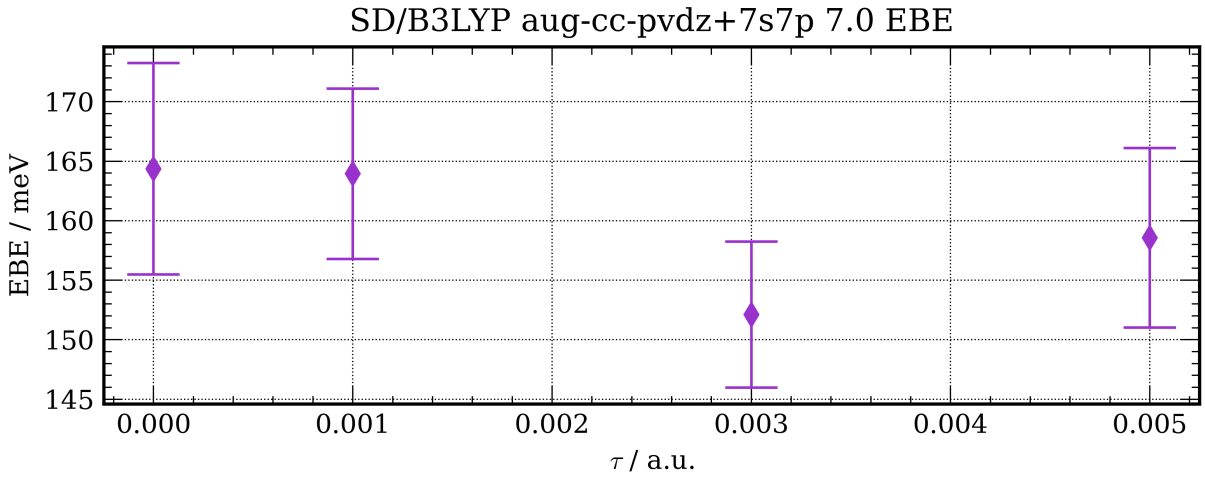


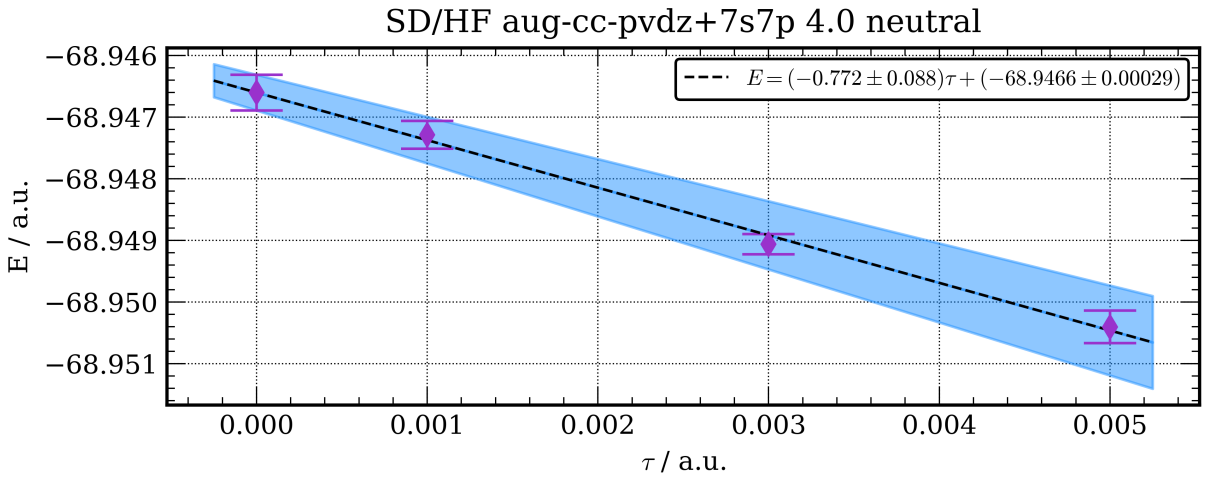


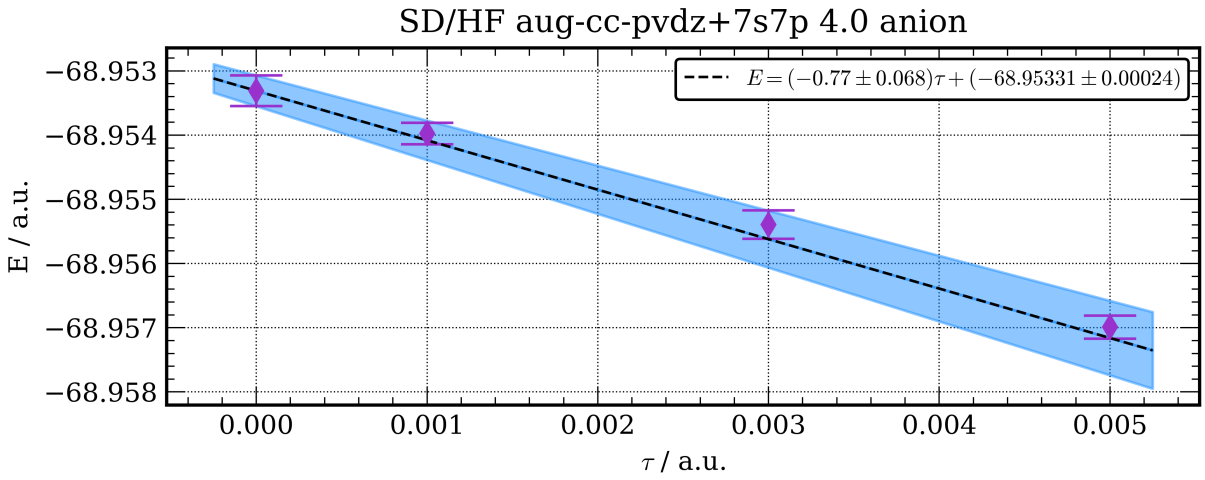


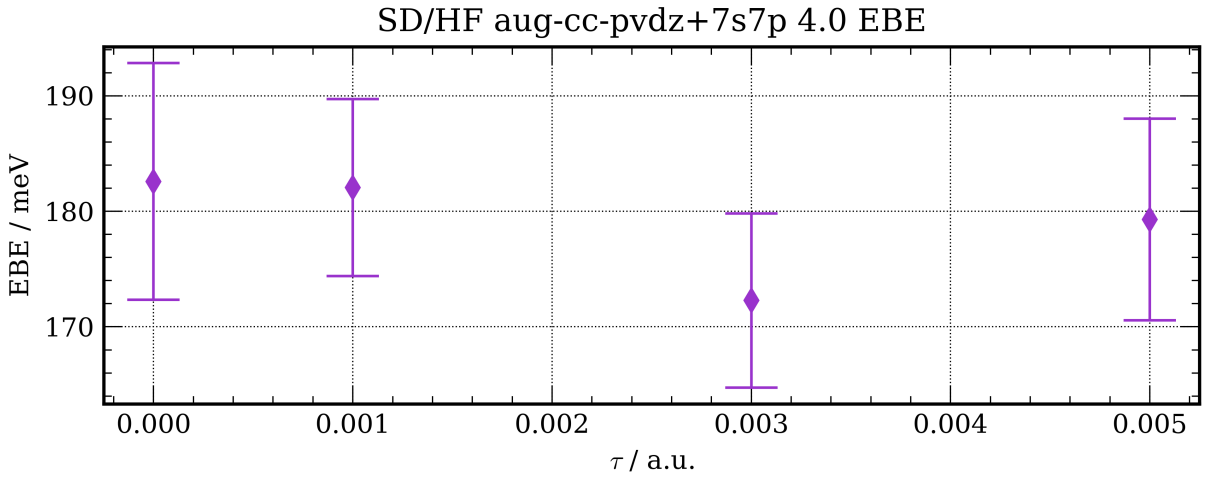

SD/B3LYP aug-cc-pvdz+7s7p 7.0 neutral
$E = (-0.602 \pm 0.062)\tau + (-68.95195 \pm 0.00021)$
E / a.u.
−68.952
−68.953
−68.954
−68.955
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

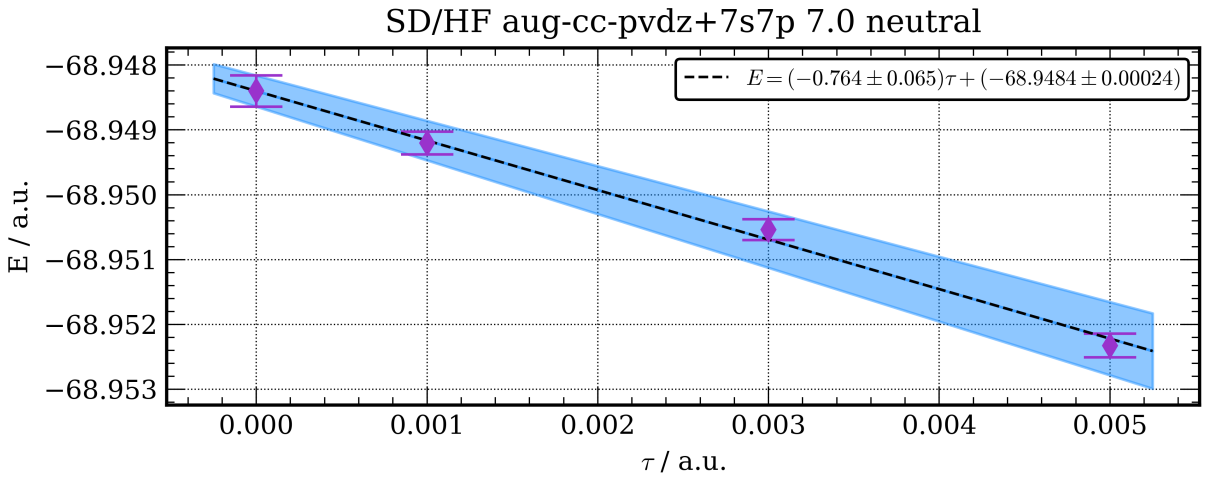

SD/B3LYP aug-cc-pvdz+7s7p 7.0 anion
$E = (-0.533 \pm 0.077)\tau + (-68.95799 \pm 0.00025)$
E / a.u.
−68.958
−68.959
−68.960
−68.961
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

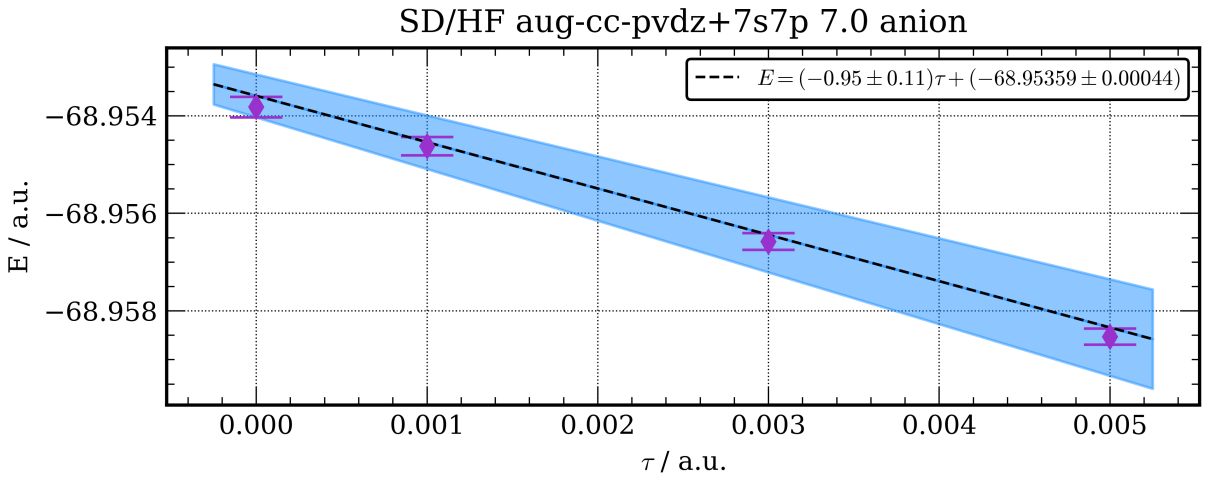

SD/B3LYP aug-cc-pvdz+7s7p 7.0 EBE
EBE / meV
170
165
160
155
150
145
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

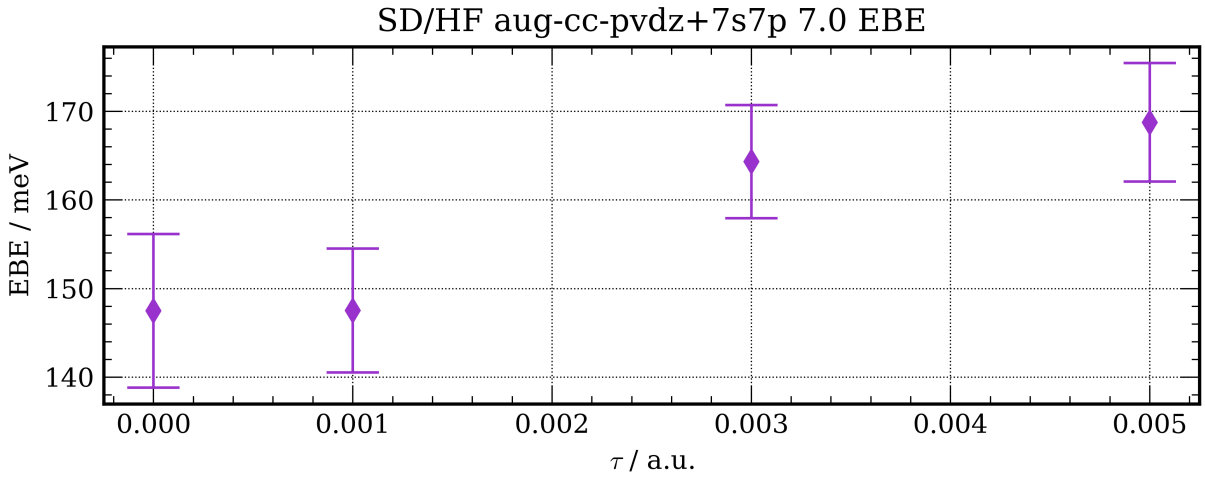

SD/HF aug-cc-pvdz+7s7p 4.0 neutral
$E = (-0.772 \pm 0.088)\tau + (-68.9466 \pm 0.00029)$
E / a.u.
−68.946
−68.947
−68.948
−68.949
−68.950
−68.951
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

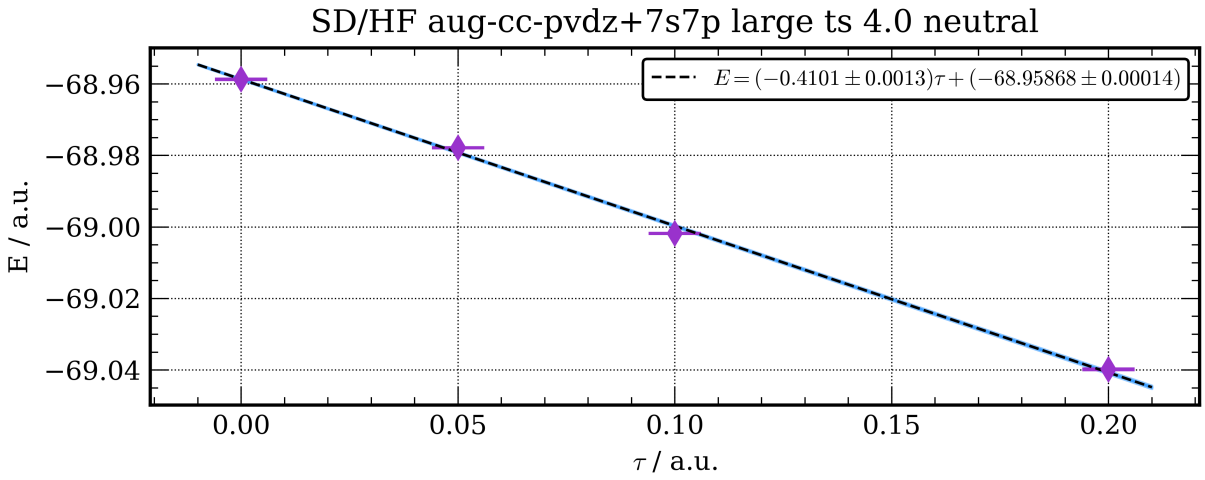

SD/HF aug-cc-pvdz+7s7p 4.0 anion
$E = (-0.77 \pm 0.068)\tau + (-68.95331 \pm 0.00024)$
E / a.u.
−68.953
−68.954
−68.955
−68.956
−68.957
−68.958
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

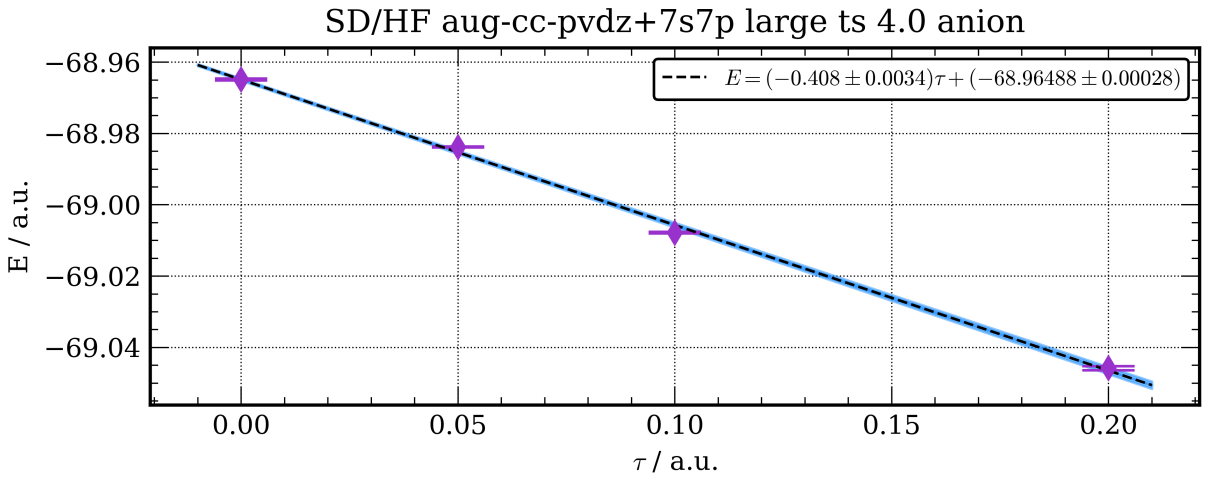

SD/HF aug-cc-pvdz+7s7p 4.0 EBE
EBE / meV
190
180
170
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

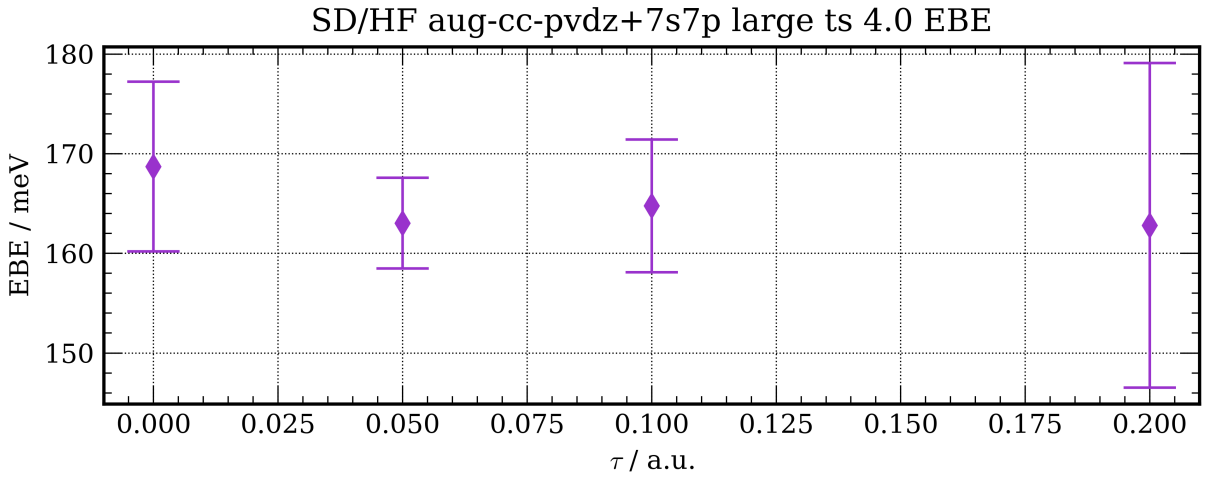

SD/HF aug-cc-pvdz+7s7p 7.0 neutral
$E = (-0.764 \pm 0.065)\tau + (-68.9484 \pm 0.00024)$
E / a.u.
−68.948
−68.949
−68.950
−68.951
−68.952
−68.953
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

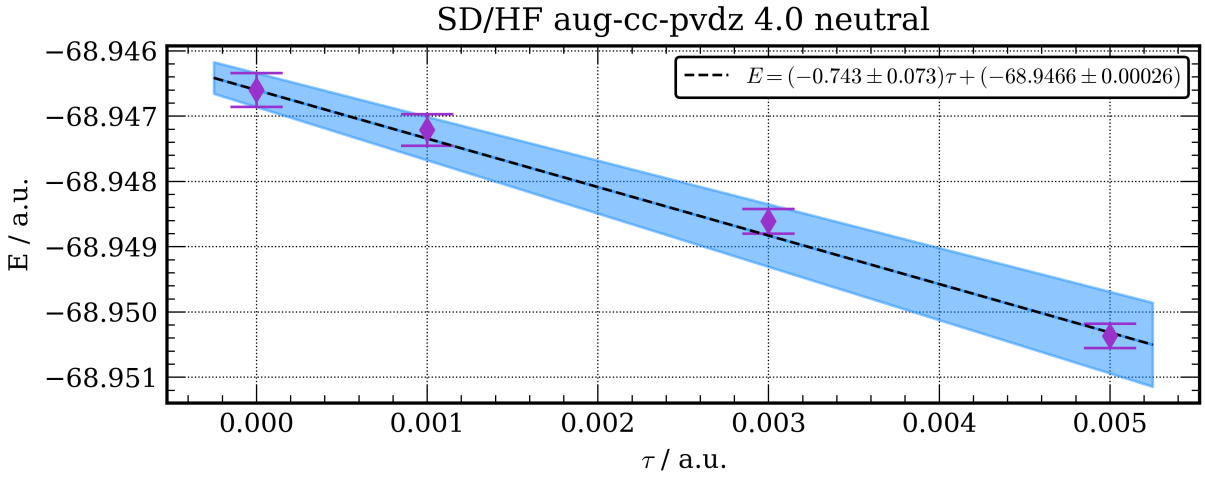

SD/HF aug-cc-pvdz+7s7p 7.0 anion
$E = (-0.95 \pm 0.11)\tau + (-68.95359 \pm 0.00044)$
E / a.u.
−68.954
−68.956
−68.958
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

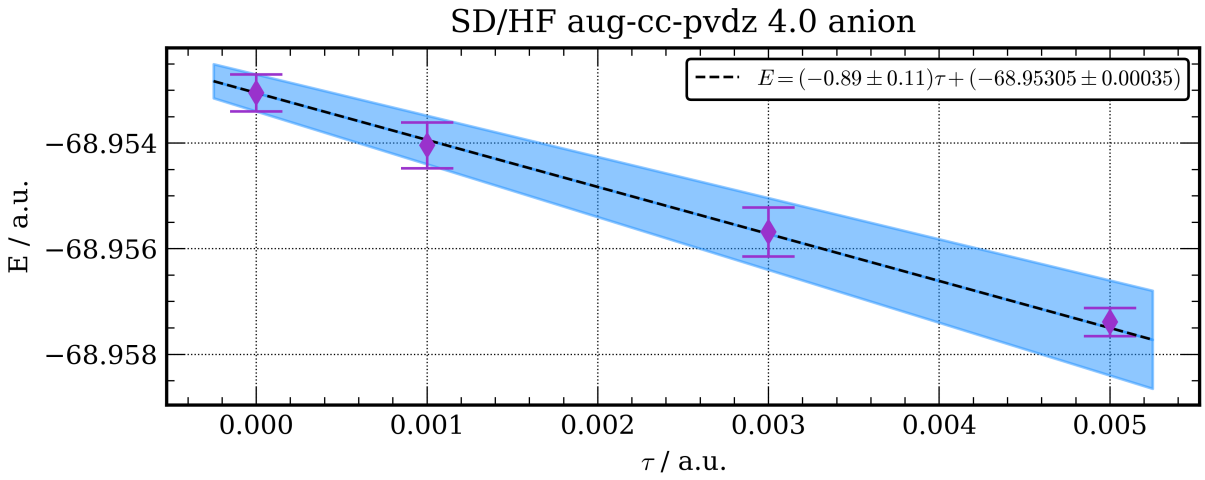

SD/HF aug-cc-pvdz+7s7p 7.0 EBE
EBE / meV
170
160
150
140
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

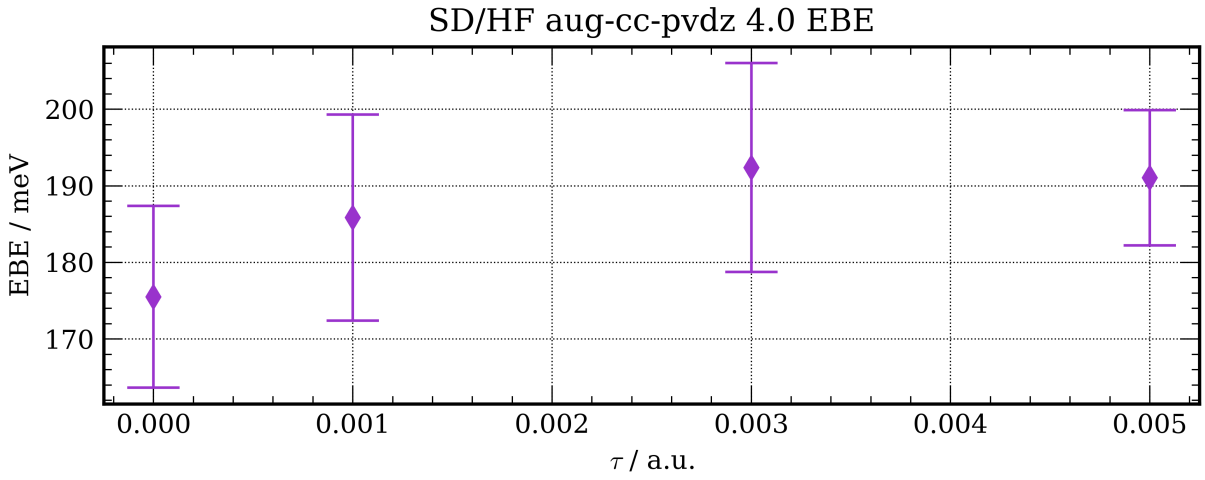

SD/HF aug-cc-pvdz+7s7p large ts 4.0 neutral
$E = (-0.4101 \pm 0.0013)\tau + (-68.95868 \pm 0.00014)$
E / a.u.
−68.96
−68.98
−69.00
−69.02
−69.04
0.00
0.05
0.10
0.15
0.20
τ / a.u.

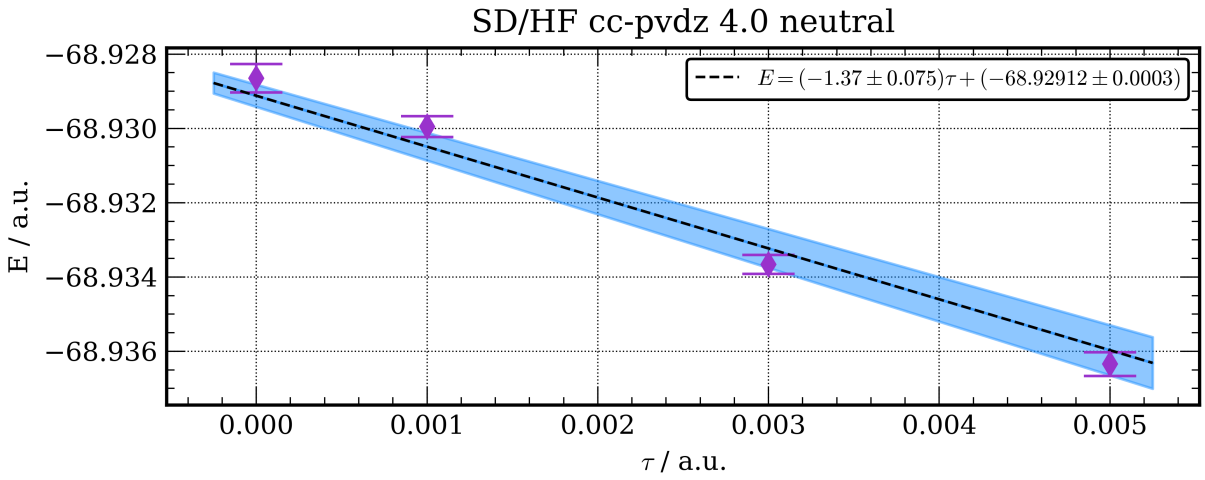

SD/HF aug-cc-pvdz+7s7p large ts 4.0 anion
$E = (-0.408 \pm 0.0034)\tau + (-68.96488 \pm 0.00028)$
E / a.u.
−68.96
−68.98
−69.00
−69.02
−69.04
0.00
0.05
0.10
0.15
0.20
τ / a.u.

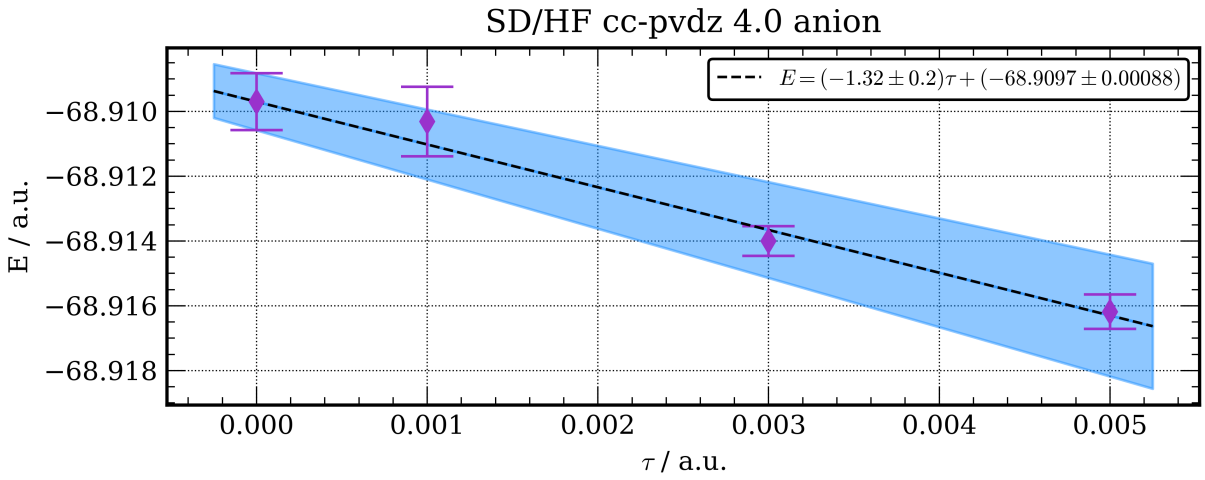

SD/HF aug-cc-pvdz+7s7p large ts 4.0 EBE
EBE / meV
180
170
160
150
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
τ / a.u.

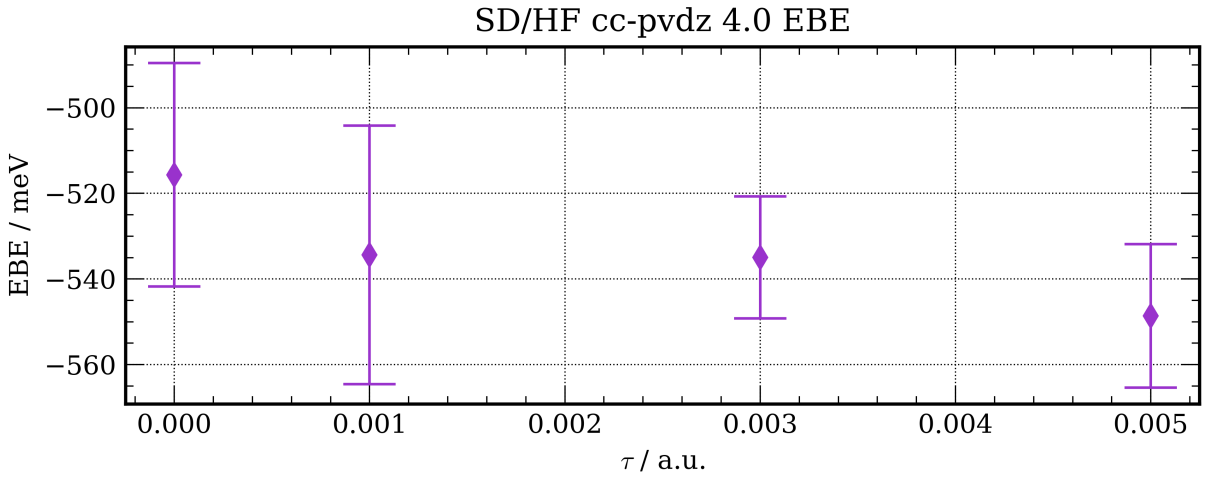

SD/HF aug-cc-pvdz 4.0 neutral
$E = (-0.743 \pm 0.073)\tau + (-68.9466 \pm 0.00026)$
E / a.u.
−68.946
−68.947
−68.948
−68.949
−68.950
−68.951
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

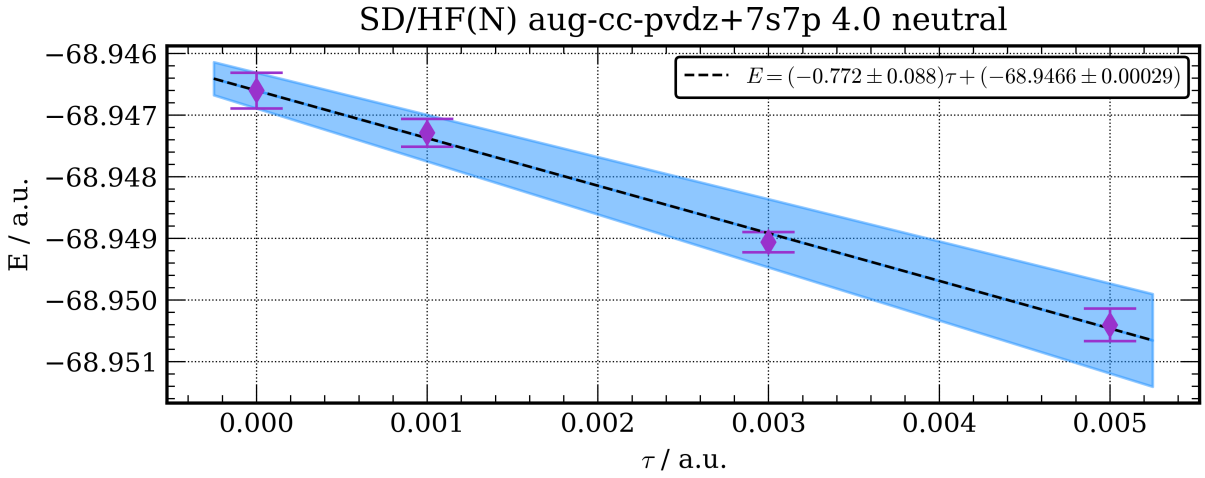

SD/HF aug-cc-pvdz 4.0 anion
$E = (-0.89 \pm 0.11)\tau + (-68.95305 \pm 0.00035)$
E / a.u.
−68.954
−68.956
−68.958
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

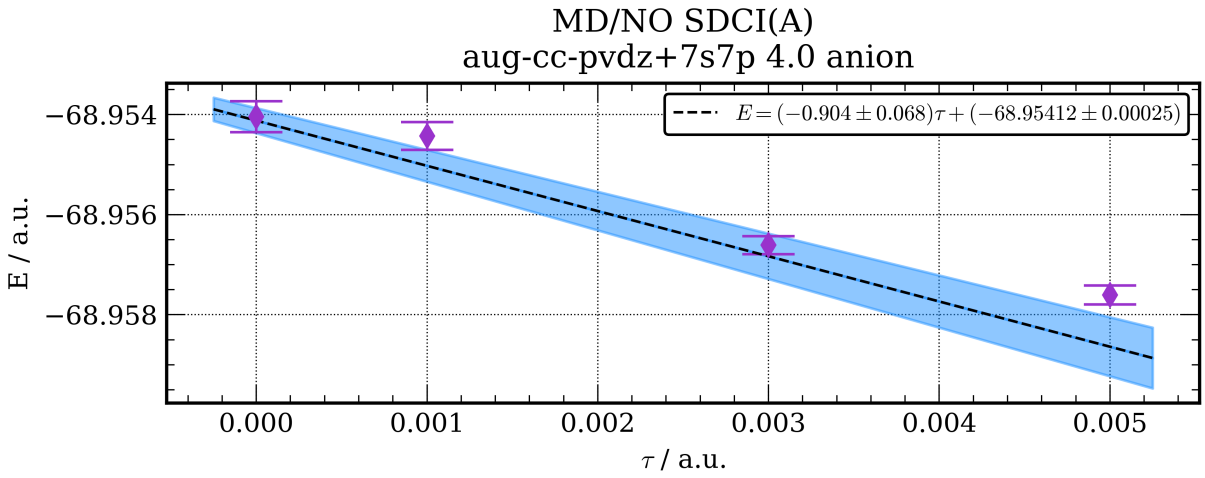

SD/HF aug-cc-pvdz 4.0 EBE
EBE / meV
200
190
180
170
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

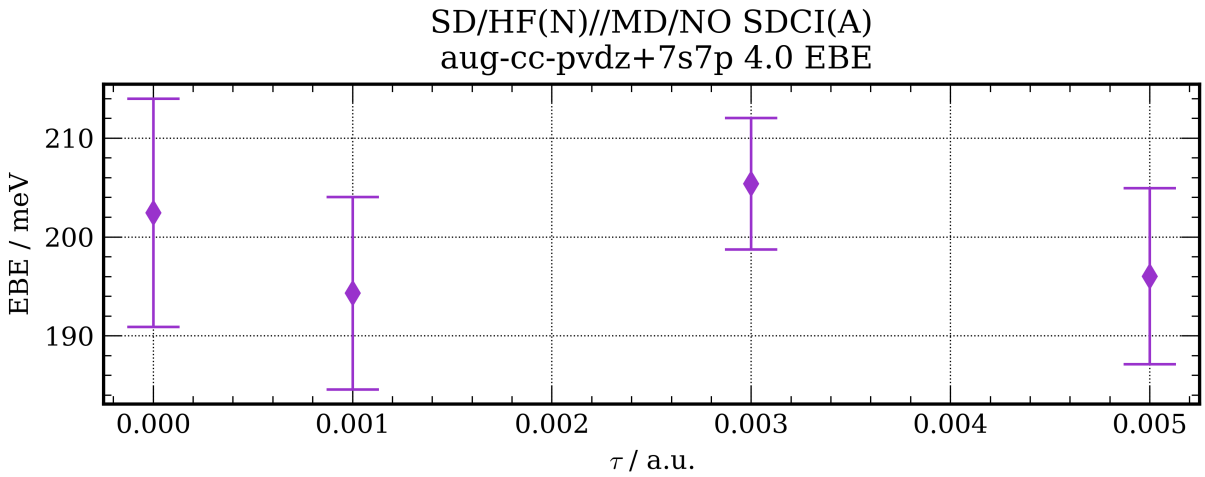

SD/HF cc-pvdz 4.0 neutral
$E = (-1.37 \pm 0.075)\tau + (-68.92912 \pm 0.0003)$
E / a.u.
−68.928
−68.930
−68.932
−68.934
−68.936
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

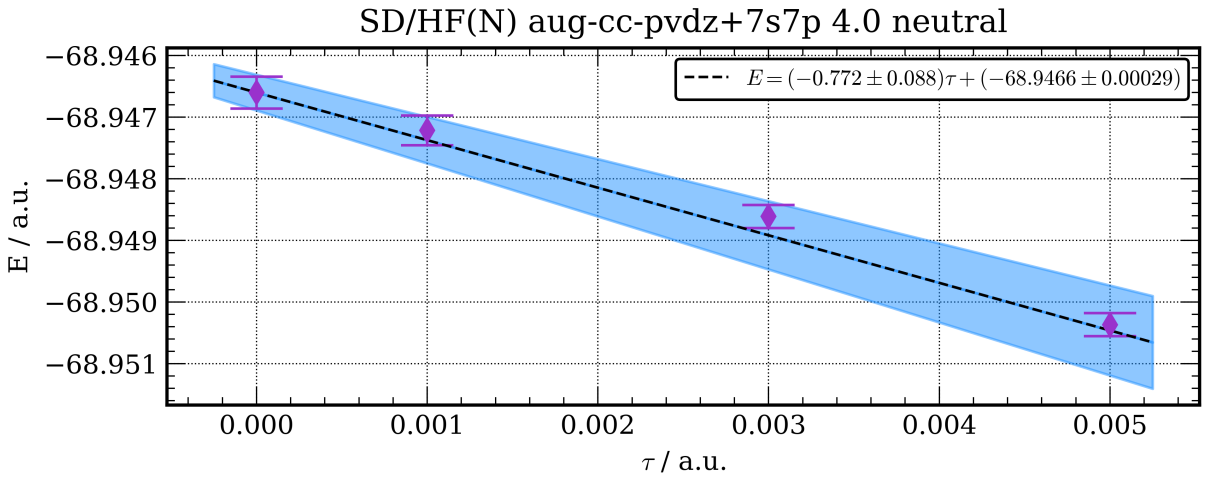

SD/HF cc-pvdz 4.0 anion
$E = (-1.32 \pm 0.2)\tau + (-68.9097 \pm 0.00088)$
E / a.u.
−68.910
−68.912
−68.914
−68.916
−68.918
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

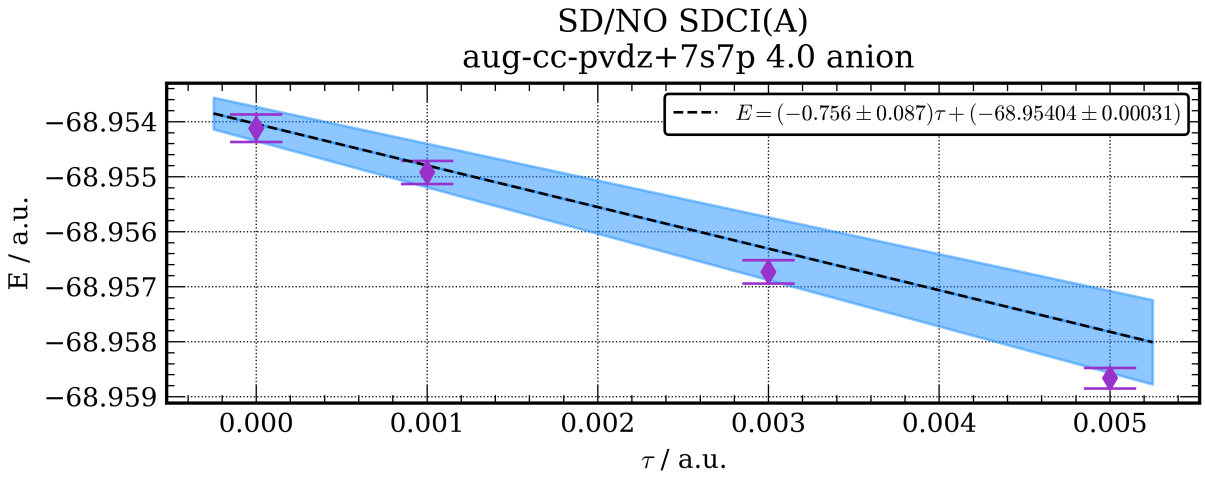

SD/HF cc-pvdz 4.0 EBE
EBE / meV
−500
−520
−540
−560
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

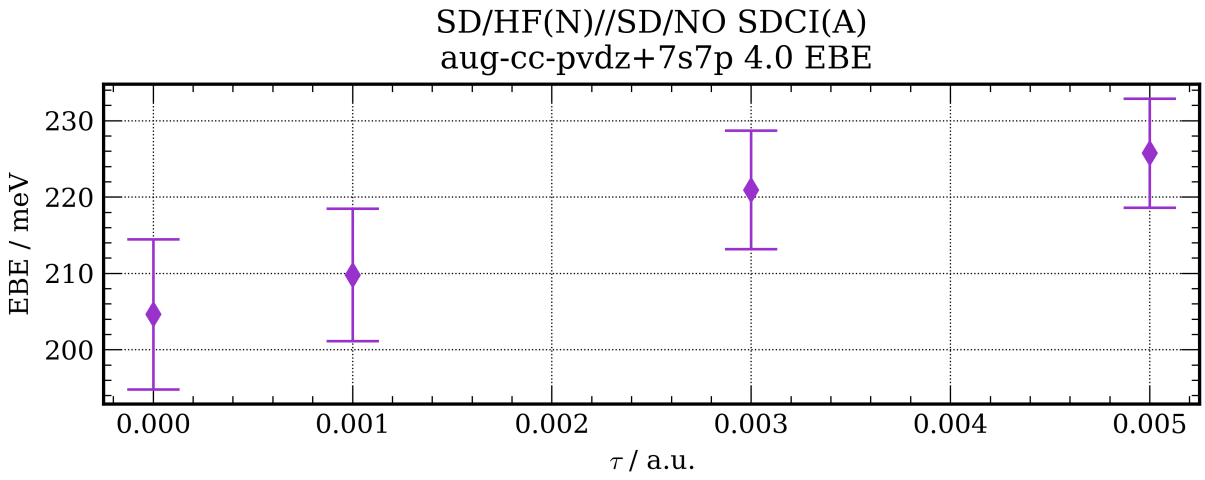

SD/HF(N) aug-cc-pvdz+7s7p 4.0 neutral
$E = (-0.772 \pm 0.088)\tau + (-68.9466 \pm 0.00029)$
E / a.u.
−68.946
−68.947
−68.948
−68.949
−68.950
−68.951
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

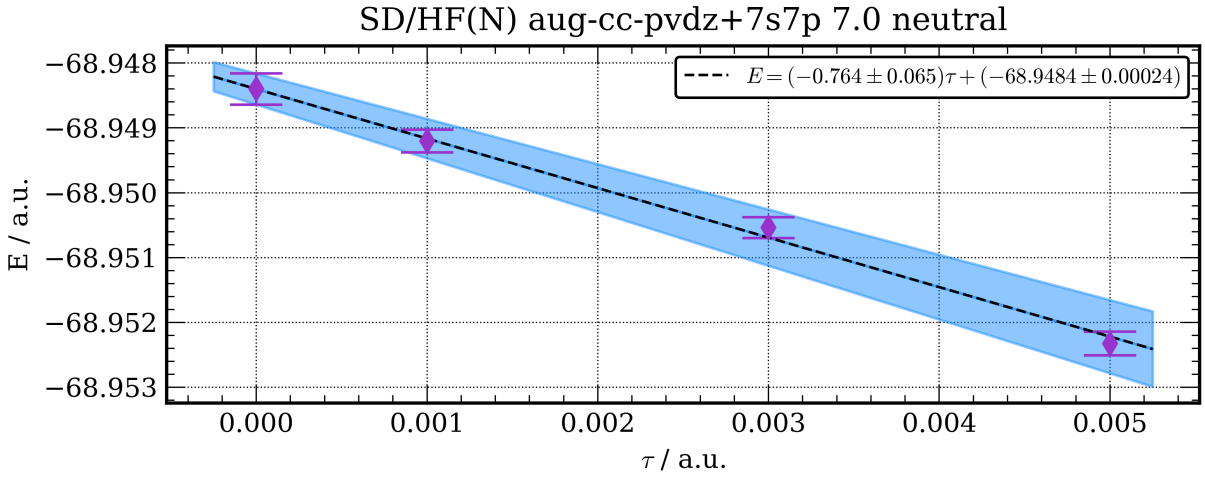

MD/NO SDCI(A)
aug-cc-pvdz+7s7p 4.0 anion
$E = (-0.904 \pm 0.068)\tau + (-68.95412 \pm 0.00025)$
E / a.u.
−68.954
−68.956
−68.958
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

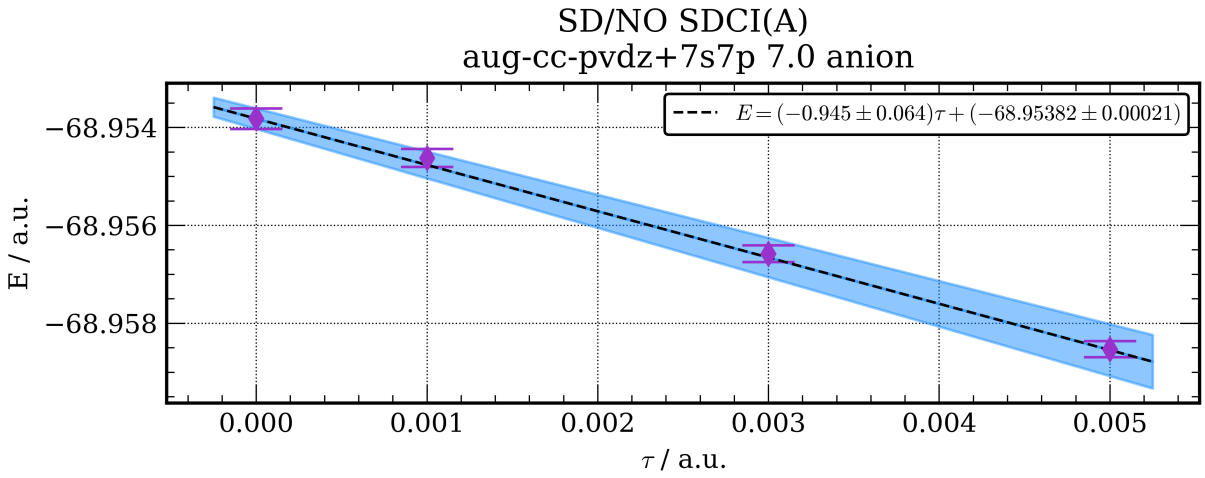

SD/HF(N)//MD/NO SDCI(A)
aug-cc-pvdz+7s7p 4.0 EBE
EBE / meV
210
200
190
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

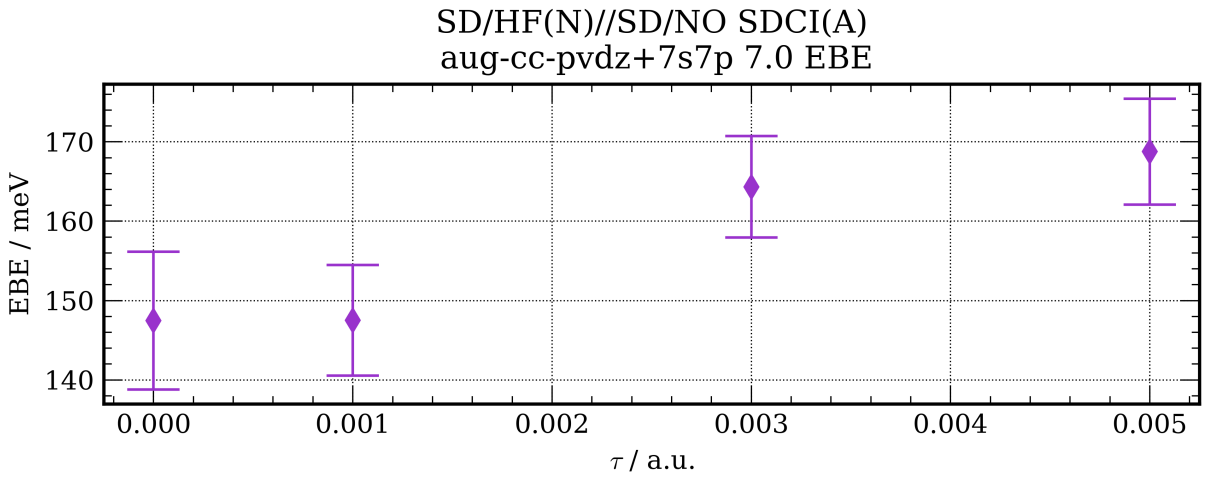

SD/HF(N) aug-cc-pvdz+7s7p 4.0 neutral
$E = (-0.772 \pm 0.088)\tau + (-68.9466 \pm 0.00029)$
E / a.u.
−68.946
−68.947
−68.948
−68.949
−68.950
−68.951
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.


SD/NO SDCI(A)
aug-cc-pvdz+7s7p 4.0 anion
$E = (-0.756 \pm 0.087)\tau + (-68.95404 \pm 0.00031)$
E / a.u.
−68.954
−68.955
−68.956
−68.957
−68.958
−68.959
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.


SD/HF(N)//SD/NO SDCI(A)
aug-cc-pvdz+7s7p 4.0 EBE
EBE / meV
230
220
210
200
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.


SD/HF(N) aug-cc-pvdz+7s7p 7.0 neutral
$E = (-0.764 \pm 0.065)\tau + (-68.9484 \pm 0.00024)$
E / a.u.
−68.948
−68.949
−68.950
−68.951
−68.952
−68.953
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.


SD/NO SDCI(A)
aug-cc-pvdz+7s7p 7.0 anion
$E = (-0.945 \pm 0.064)\tau + (-68.95382 \pm 0.00021)$
E / a.u.
−68.954
−68.956
−68.958
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.


SD/HF(N)//SD/NO SDCI(A)
aug-cc-pvdz+7s7p 7.0 EBE
EBE / meV
170
160
150
140
0.000
0.001
0.002
0.003
0.004
0.005
τ / a.u.

dependency	version
`numpy`	1.18.4
`quadpy`	0.16.2
`pyscf`	1.7.0
`cclib`	1.6.3

3 Radial orbital density plots

The radial orbital density plots are created by integrating over angular portion of the norm of the single particle wavefunction.

$$\bar{\psi}^2(r) = \int_0^{2\pi} \int_0^{\pi} \psi^2(r, \theta, \phi) r^2 \sin(\theta) dr d\theta d\phi \tag{1}$$

Discretizing this expression using a uniform radial grid and a Lebedev-Laikov quadrature for the angular components, yields a form that can be readily evaluated.

$$\bar{\psi}^2(r_i) = 4\pi r_i^2 \sum_j^{N^{ang}} w_j^{ang} \psi^2(r_i, \theta_j, \phi_j) \tag{2}$$

The function $\bar{\psi}^2(r_i)$ from equation 2 is used for plotting, with the points r_i serving as the abscissa. Since the singly occupied orbitals are normalized, the sum of the radial quadrature of this function is also checked to be close to unity.

$$\sum_i^{N^{rad}} \bar{\psi}^2(r_i) w_i^{rad} = \sum_i^{N^{rad}} \bar{\psi}^2(r_i) \Delta r \approx 1 \tag{3}$$

3.a Dependency versions

3.b Step 1: Generating a Molden file

Molden files were generated using `cclib`. The only exception is the natural orbital from the CIPSI calculations. Since QuantumPackage is not supported by `cclib`, molden files were created using the native utility in QuantumPackage 2.0. The `-g/--ghost` flag indicates the presence of a ghost atom. The `-n/--naturalorbtials` was created to allow natural orbitals

to be written instead of molecular orbitals. This flag is not yet availible in the official distribution, but a request to incorporate it in the official distribution has been opened (https://github.com/cclib/cclib/pull/948).

```
$ ccwrite molden -g "X" -n QUANTUM_CHEMISTRY_OUTPUT_FILE
```

3.c Step 2: Integrating over the angular components of the singly occupied orbital

Quadpy was used to generate the Lebedev-Laikov integration weights and points. The singly occupied molecular/natural orbital was was evaluated at these points using PySCF.

```
import numpy
import quadpy
import pyscf
import pyscf.tools

filename = "anion_allelec_res_4.0_homoshivhf_norev.molden"
r_max = 100
num_radial_pts = 1000
mo_idx = 20

mol, mo_energy, mo_coeff, mo_occ, irrep_labels, spins = pyscf.tools.molden.load(
    filename
)
singly_occ_orb = mo_coeff[:, mo_idx]

lebedev_laikov = quadpy.u3.schemes["lebedev_131"]()
angular_pts = lebedev_laikov.theta_phi
angular_weights = lebedev_laikov.weights
```

```
num_angular_pts = len(angular_pts[0])


radial_pts = numpy.linspace(r_max, 0, num_radial_pts, endpoint=False)[::-1]



def sph2cart(r, theta_phi):
    theta = theta_phi[0]
    phi = theta_phi[1]
    x = r * numpy.cos(theta) * numpy.sin(phi)
    y = r * numpy.sin(theta) * numpy.sin(phi)
    z = r * numpy.cos(phi)
    return numpy.vstack((x, y, z)).T



values = []
for r in radial_pts:
    r_pts = r * numpy.ones(num_angular_pts)
    coords = sph2cart(r_pts, angular_pts)
    ao = mol.eval_gto("GTOval_cart", coords)
    value = angular_weights @ ao @ singly_occ_orb
    values.append(4 * numpy.pi * r * r * value ** 2)


values = numpy.array(values)


numpy.savetxt("{}_values.txt".format(filename), values)
numpy.savetxt("{}_r.txt".format(filename), radial)
```