KTH ROYAL INSTITUTE
OF TECHNOLOGY

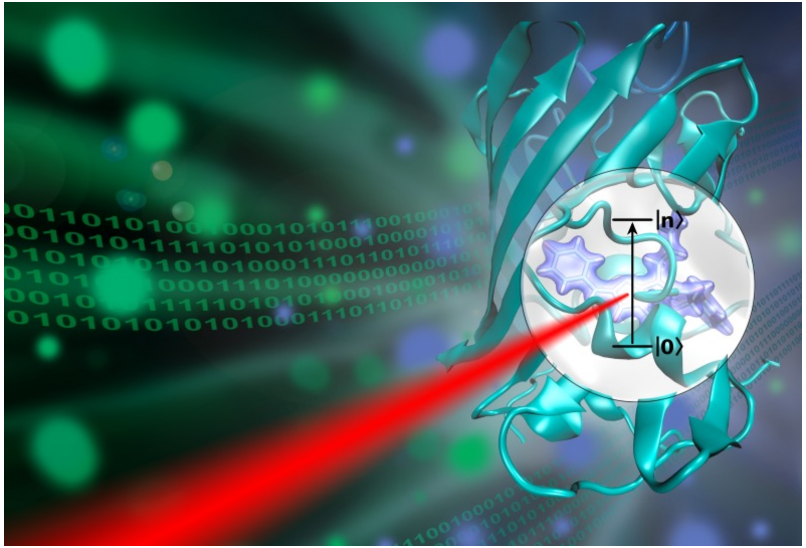
VeloxChem — Quantum Molecular Modelling in HPC Environments

Xin Li

2023-08-25

VeloxChem

- Next-generation **quantum chemistry** software for molecular properties
- Key capabilities
  - Kohn–Sham Density Functional Theory (DFT)
  - Time-dependent DFT (TDDFT)
  - Optical and X-ray absorption
  - Electronic circular dichroism (ECD)
  - Polarizabilities and dispersion coefficients
  - Complex polarization propagator (CPP)
  - Response theory for pulses



Vision and Goals

Target Science

- Biochemistry
- Nanoscience
- Spectroscopy

Science- and education-enabling software platform for quantum molecular modeling on contemporary and future HPC systems

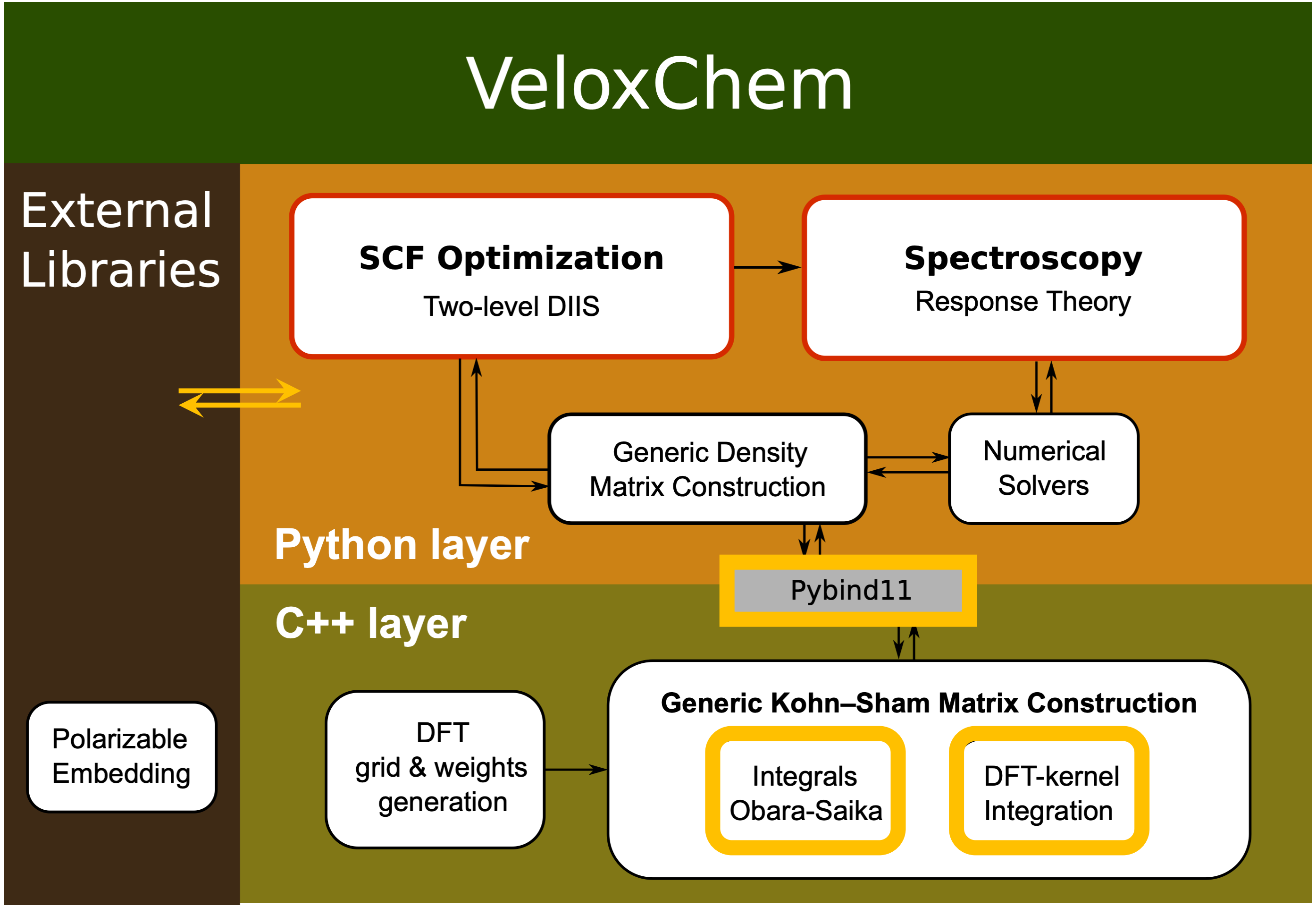
A quantum molecular software to meet the challenges of the EuroHPC project

https://eurohpc-ju.europa.eu

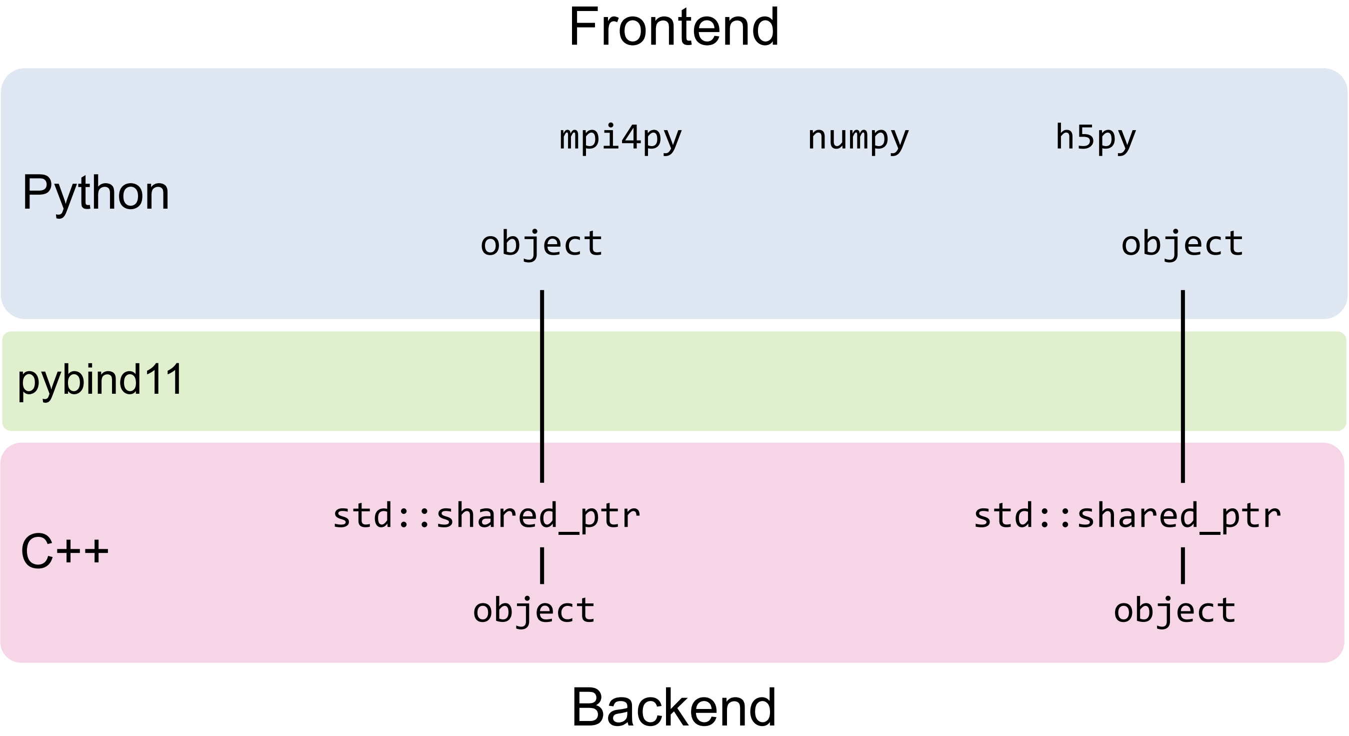
Target Hardware

- Cluster distributed memory
- CPU/GPU equipped nodes
- 2–3 Gb RAM per CPU core
- No node local spinning disk
- Fast interconnect

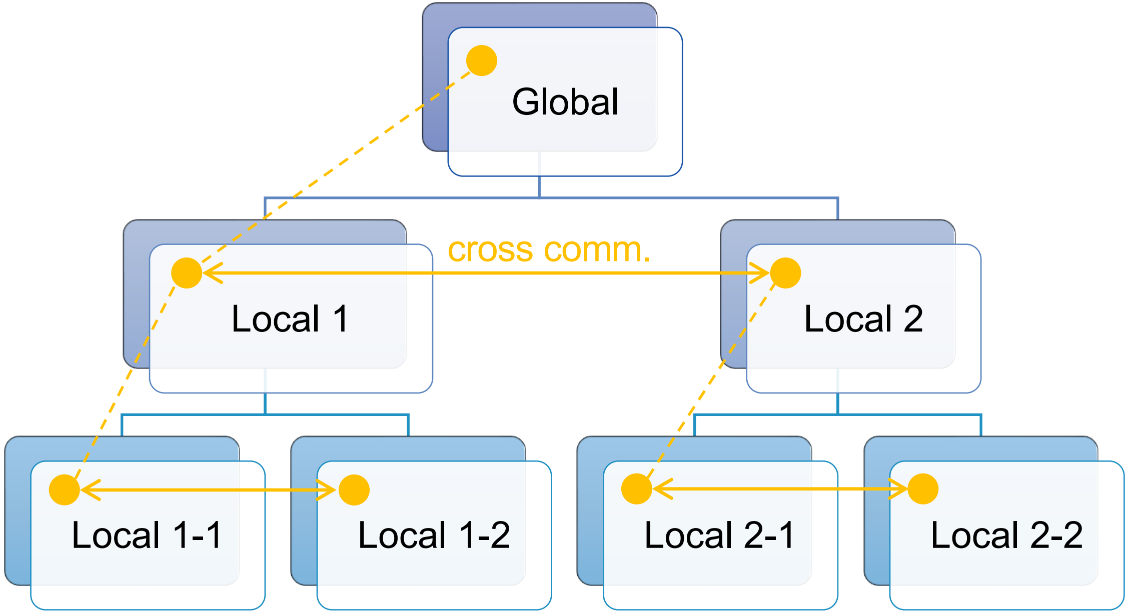
EuroHPC


VeloxChem
External Libraries
SCF Optimization
Two-level DIIS
Spectroscopy
Response Theory
Generic Density Matrix Construction
Numerical Solvers
Python layer
Pybind11
C++ layer
Polarizable Embedding
DFT grid & weights generation
Generic Kohn–Sham Matrix Construction
Integrals Obara-Saika
DFT-kernel Integration

Python/C++ interface with pybind11



Managing resources with split communicators



Program design principles

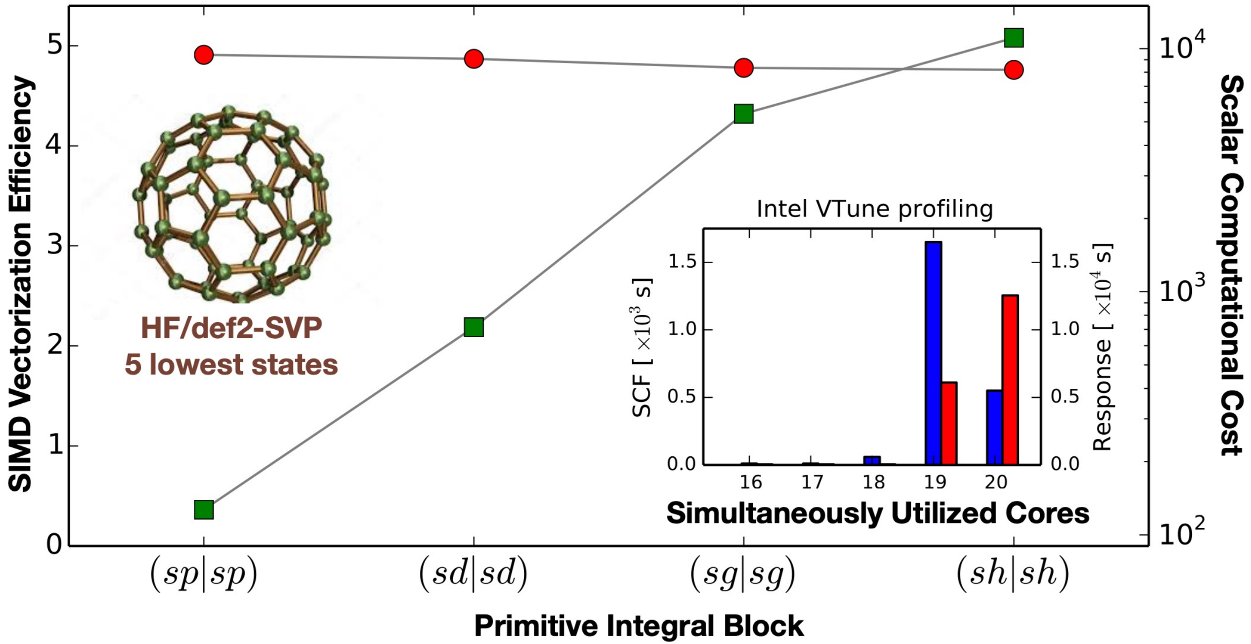
- Parallelization
  - Hybrid MPI/OpenMP parallelization
  - Flexible management of split communicators
  - Reduction and distribution with minimal communication footprint

- Vectorization
  - Autogenerated C++ code with minimized memory loads/stores on aligned data
  - Vecorization enforced by OpenMP 4.0 pragmas

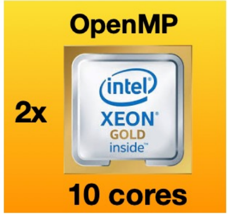
Integrals: Obara-Saika algorithm

1. Generation of basic auxiliary primitive integral $[00|00]^m$ up to m^{th} order from the family of Boys functions, where m is the sum of angular momenta on all four centers, i.e., $m=a+b+c+d$
2. Build up of primitive integrals $[00|0(c+d)]$ and $[0(a+b)|0(c+d)]$ using vertical Obara–Saika recursion
3. Generation of contracted Cartesian integrals $(0(a+b)|0(c+d))$ by contracting primitive integrals $[0(a+b)|0(c+d)]$
4. Build-up of contracted integrals $(0(a+b)|cd)$ by applying horizontal angular momentum transfer recursion to primitive integrals $(0(a+b)|0(c+d))$ on the ket-side
5. Generation of half-transformed contracted integrals $(0(a+b)|(cd)^{sp})$ from contracted integrals $(0(a+b)|cd)$ by applying Cartesian to spherical transformation on the ket-side
6. Build-up of contracted integrals $(ab|(cd)^{sp})$ by applying horizontal angular momentum transfer recursion to contracted integrals $(0(a+b)|(cd)^{sp})$ on the bra-side
7. Generation of contracted spherical integrals $((ab)^{sp}|(cd)^{sp})$ from half-transformed contracted integrals $(ab|(cd)^{sp})$ by applying Cartesian to spherical transformation on the bra-side

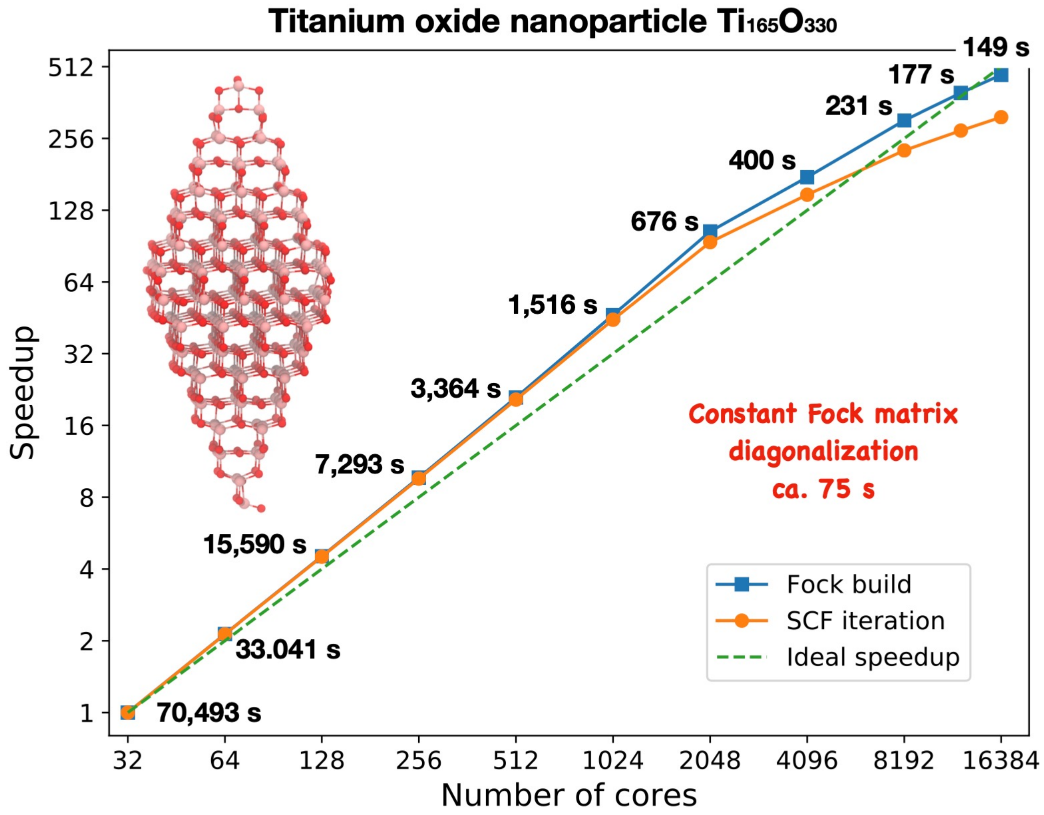
SIMD vectorization



HF/def2-SVP
5 lowest states



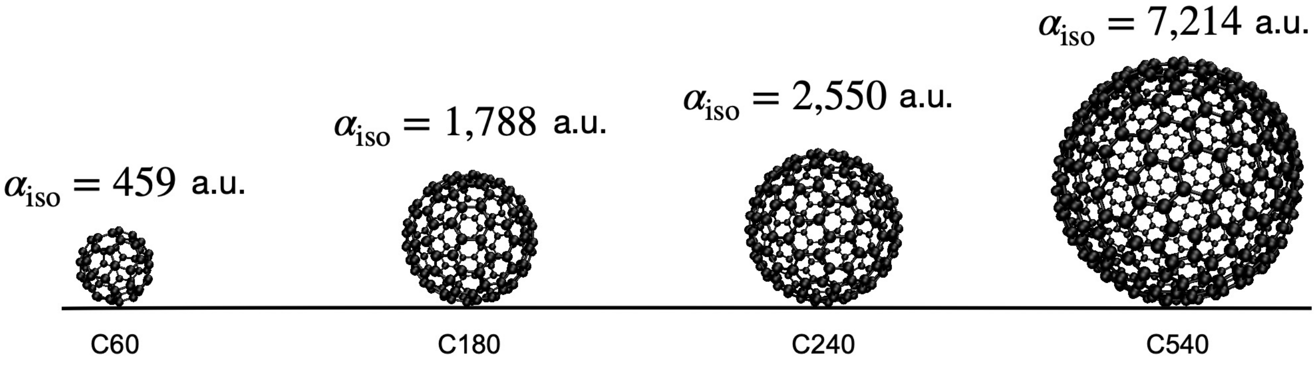
Performance (ground state)

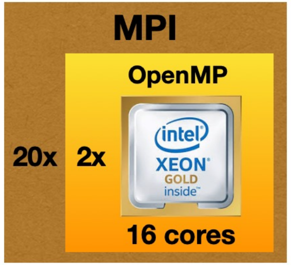


Constant Fock matrix diagonalization ca. 75 s

Beskow@PDC

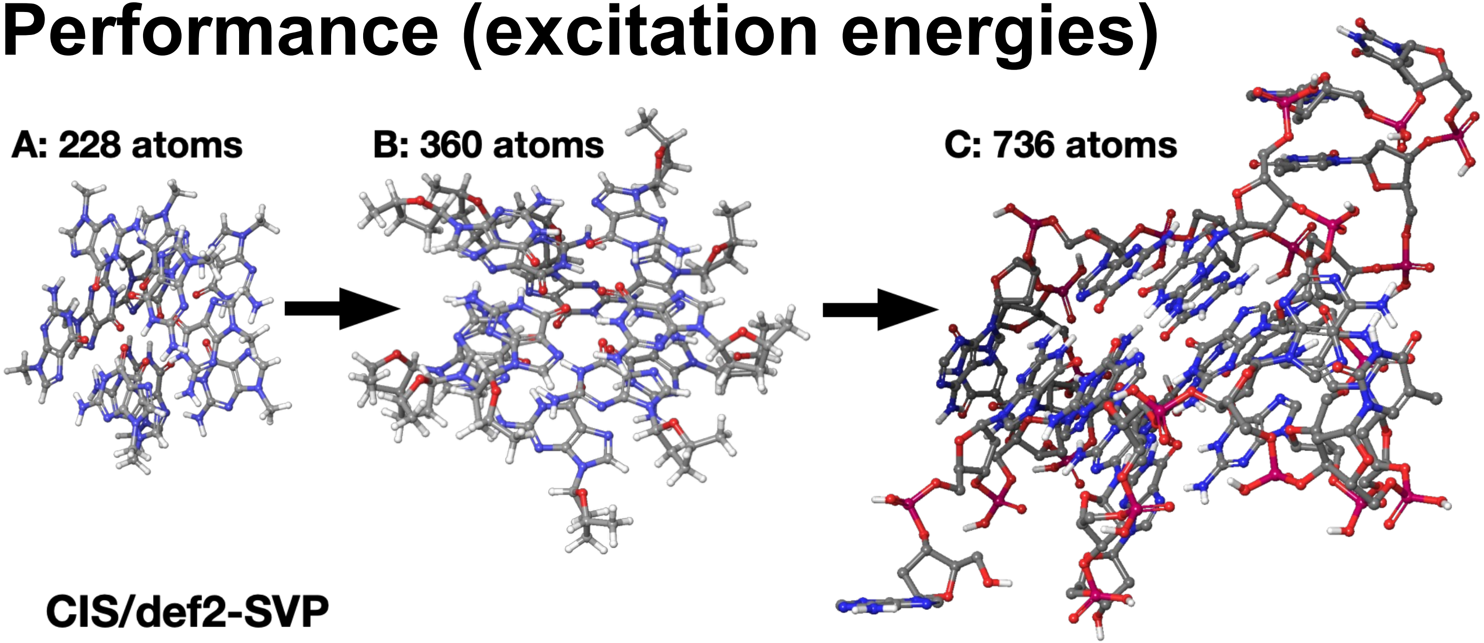
Performance (polarizability)


$\alpha_{\text{iso}} = 459$ a.u.
$\alpha_{\text{iso}} = 1{,}788$ a.u.
$\alpha_{\text{iso}} = 2{,}550$ a.u.
$\alpha_{\text{iso}} = 7{,}214$ a.u.
C60
C180
C240
C540



Fullerene	C60	C180	C240	C540
No. basis functions	840	2,520	3,360	7,560
No. iterations	14	13	12	14
Wall time, sec	**289**	**3,098**	**5,373**	**38,036**

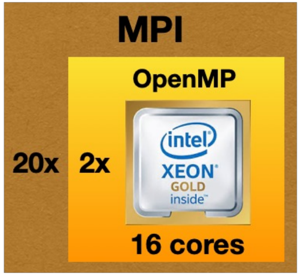
Performance (excitation energies)



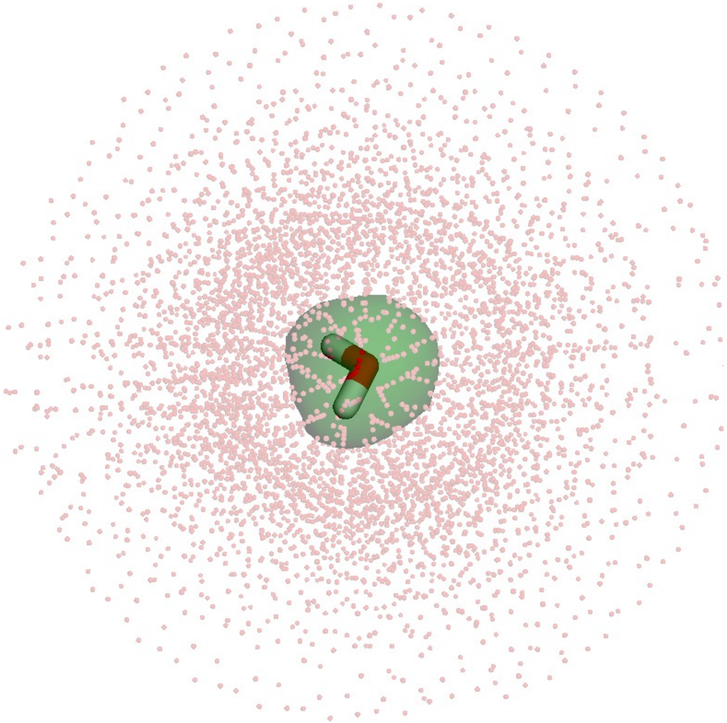
CIS/def2-SVP

5 lowest singlet excited states

Model	No. basis functions	CIS Iterations	Total Time, sec
A	2 436	32	4 901
B	3 636	35	13 517
C	7 949	45	80 494



DFT numerical integration



$$F_{\nu g} = \sum_{\mu} \rho_{\mu\nu} \chi_{\mu}(\mathbf{r}_g)$$

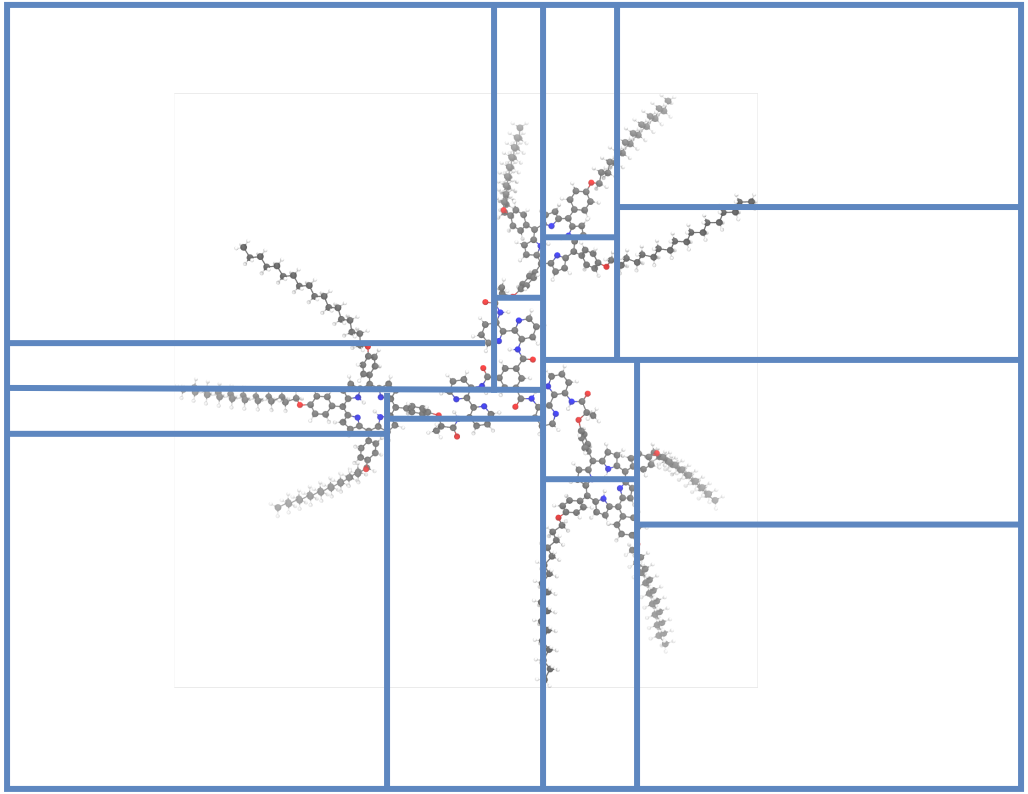
$$\rho(\mathbf{r}_g) = \sum_{\nu} F_{\nu g} \chi_{\nu}(\mathbf{r}_g)$$

$$G_{\nu g} = w_g \frac{\partial \varepsilon_{\text{xc}}}{\partial \rho}(\mathbf{r}_g) \chi_{\nu}(\mathbf{r}_g)$$

$$\mathbf{V}^{\text{XC}}_{\mu\nu} = \sum_{g} \chi_{\mu}(\mathbf{r}_g) G_{\nu g}$$

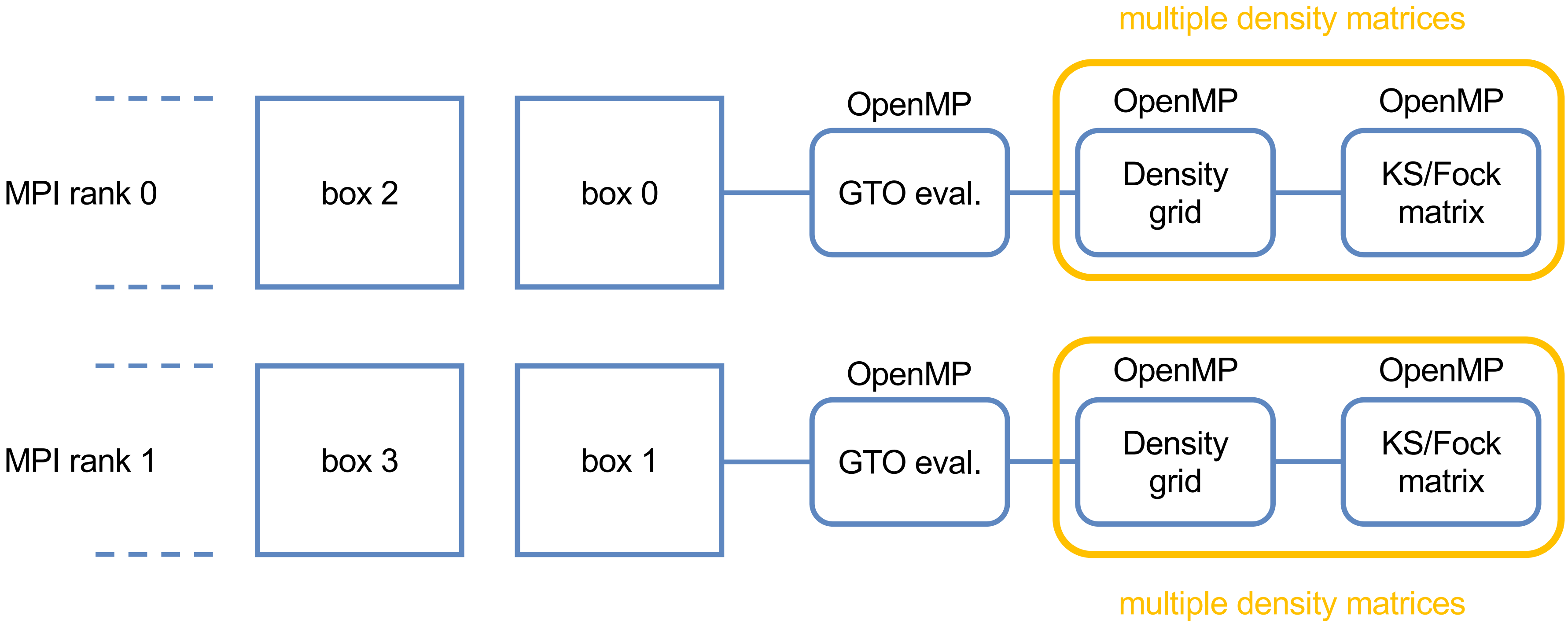
Ref: H. Laqua, T. H. Thompson, J. Kussmann, and C. Ochsenfeld, *J. Chem. Theory Comput.* 2021, 17, 1512−1521

Bisecting the grid boxes



```
 -----------------
  NPoints  NBoxes
 -----------------
    0-99   :     0
  100-199  :     0
  200-299  :     0
  300-399  :     0
  400-499  :    17
  500-599  :   497
  600-699  :    95
  700-799  :   221
  800-899  :  1343
  900-999  :  5311
 1000-1099:   2949
 -----------------
```

MPI/OpenMP parallelization

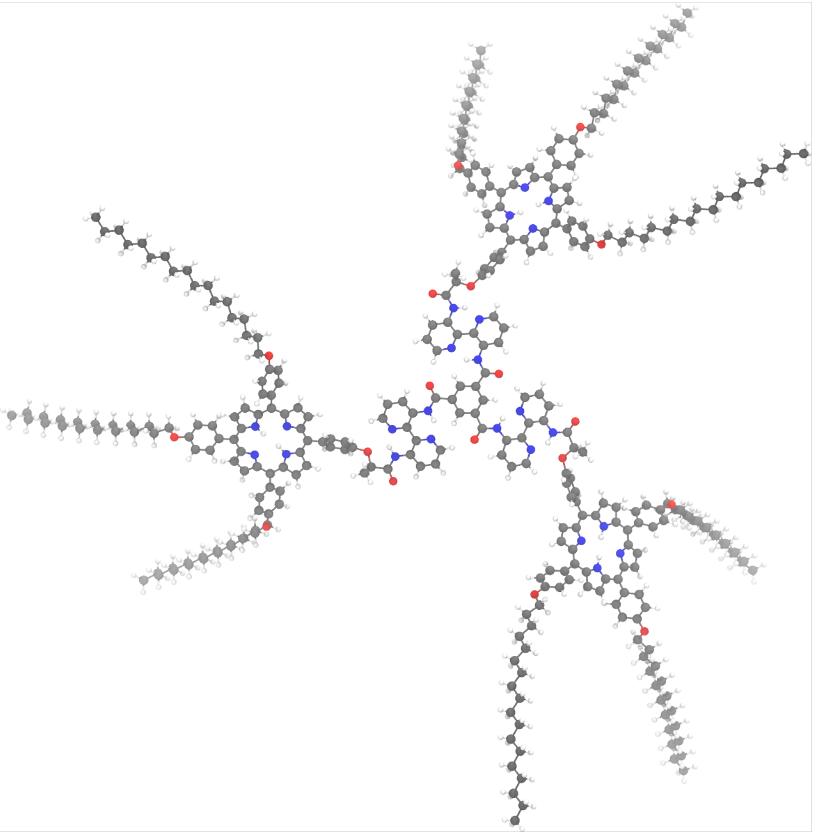


Performance (ground state)

t_{DFT} (Dardel)
8.22 sec

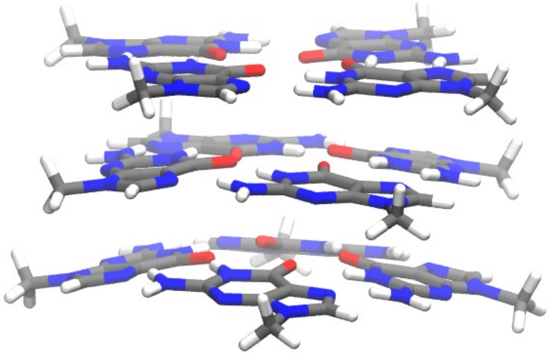
t_{DFT}: Time spent in DFT in one SCF iteration.

Functional:	B3LYP
Basis set:	def2-SVP
Number of basis functions:	7,626
Number of grid points:	9,636,124
Resource:	2 nodes



Performance (ground state)

System	# basis	t_{DFT} (Dardel)
One layer	1196	1.31 sec
Two layers	2392	4.97 sec
Three layers	3588	10.16 sec



B3LYP / def2-SVPD

2,840,663 grid points

On 2 Dardel nodes

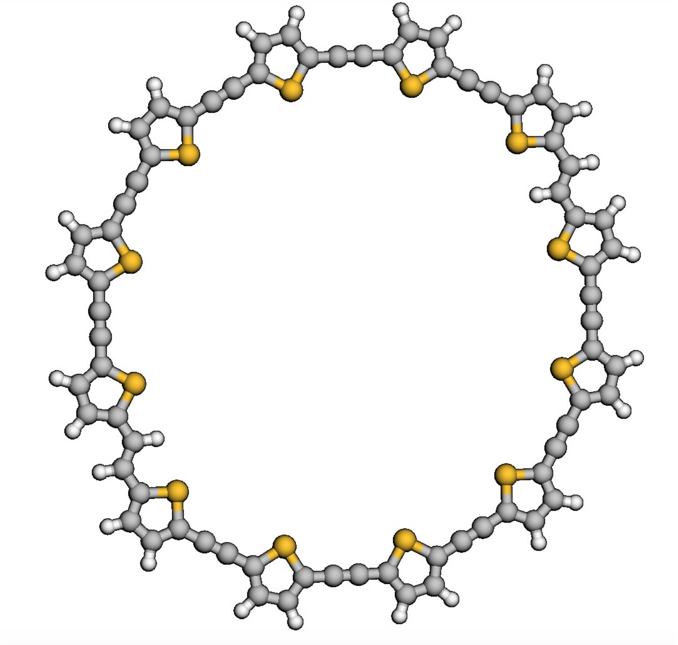
Performance (linear/nonlinear response)

# basis	# density matrices	t_{DFT} (Dardel)
1988	40 (linear rsp.)	16.62 sec
1988	270 (nonlinear rsp.)	428.50 sec

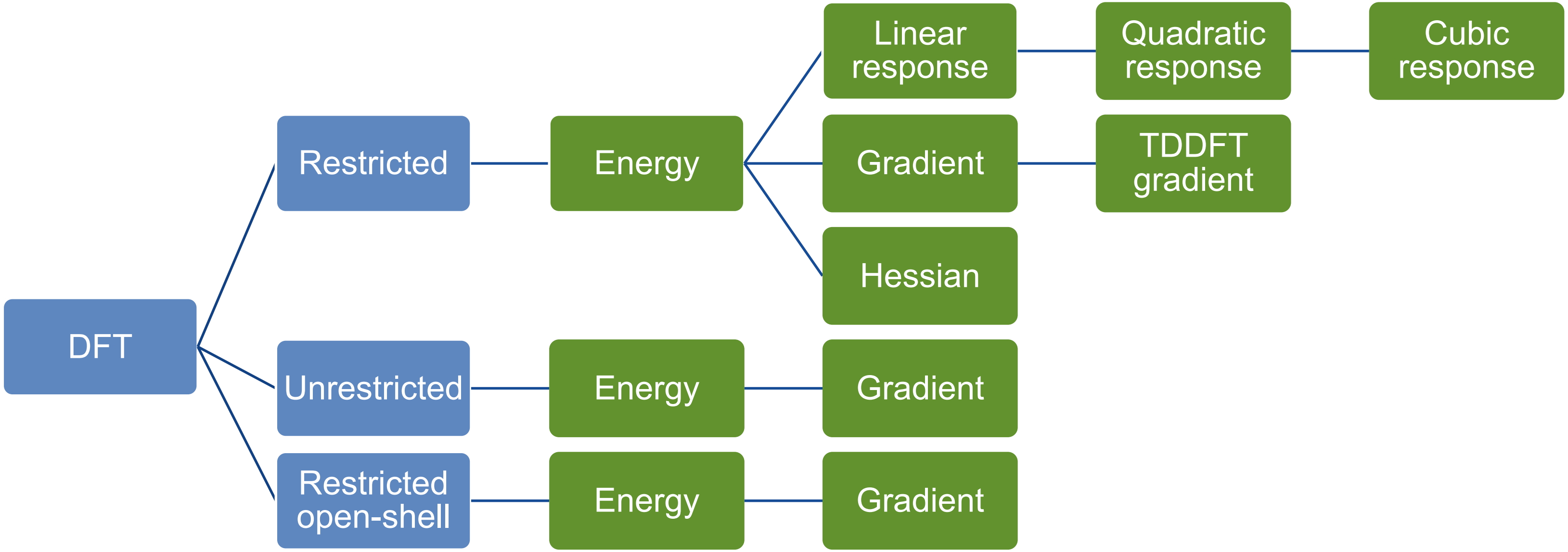
PBE0 / def2-SVPD

1,767,771 grid points

On 4 Dardel nodes



Current status of DFT in VeloxChem



DFT

Restricted

Energy

Linear response

Quadratic response

Cubic response

Gradient

TDDFT gradient

Hessian

Unrestricted

Energy

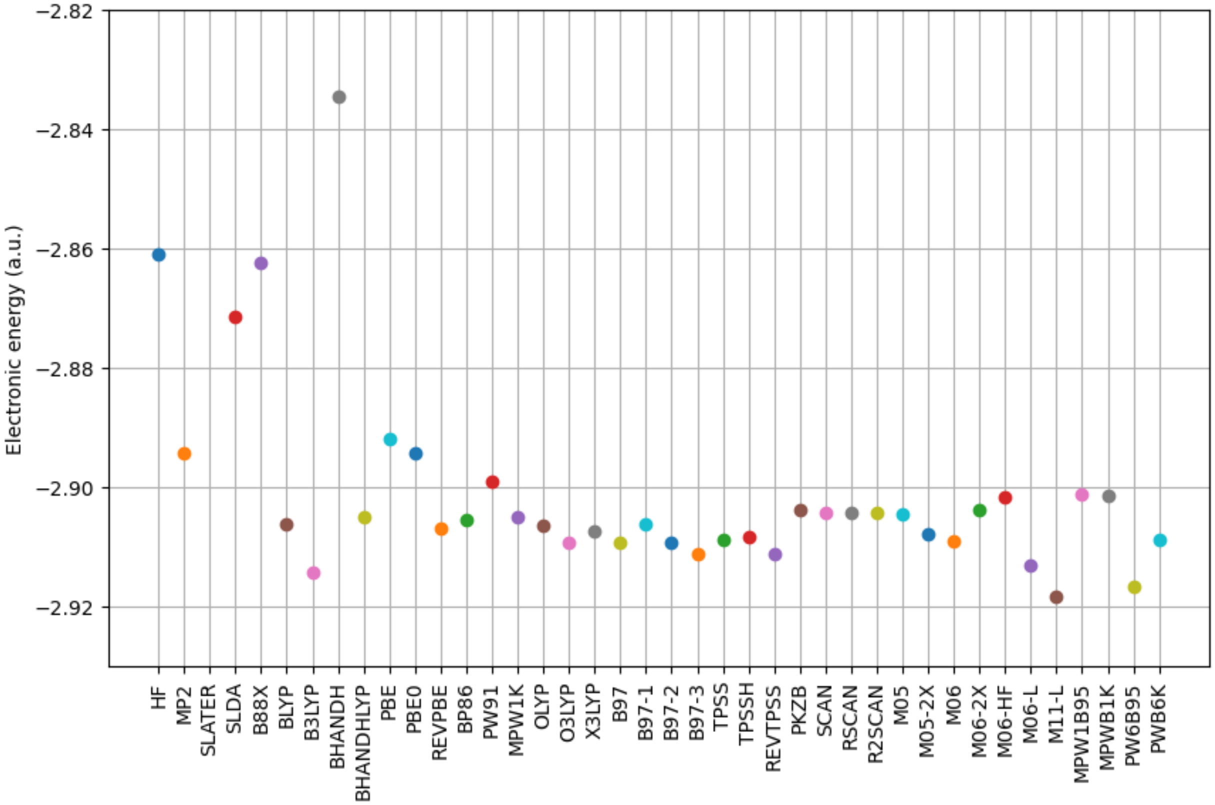
Gradient

Restricted open-shell

Energy

Gradient

Available methods: HF, MP2 and density functionals



Electronic energy of Helium atom

Basis set: cc-pVTZ

In VeloxChem, the density functionals are implemented based on libxc (https://gitlab.com/libxc/libxc)

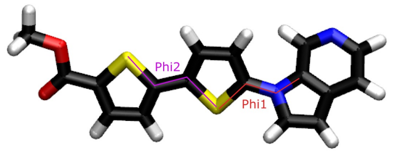
Available properties and spectra

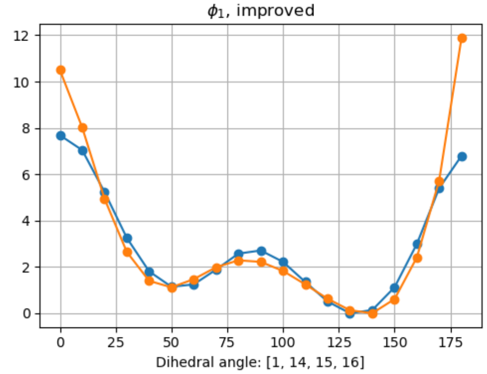
- Linear response
  - Absorption
    - > Eigensolver (supports core excitation)
    - > CPP solver
  - Electronic circular dichroism (ECD)
    - > Eigensolver (supports core excitation)
    - > CPP solver
  - C6 dispersion coefficient
  - Polarizability

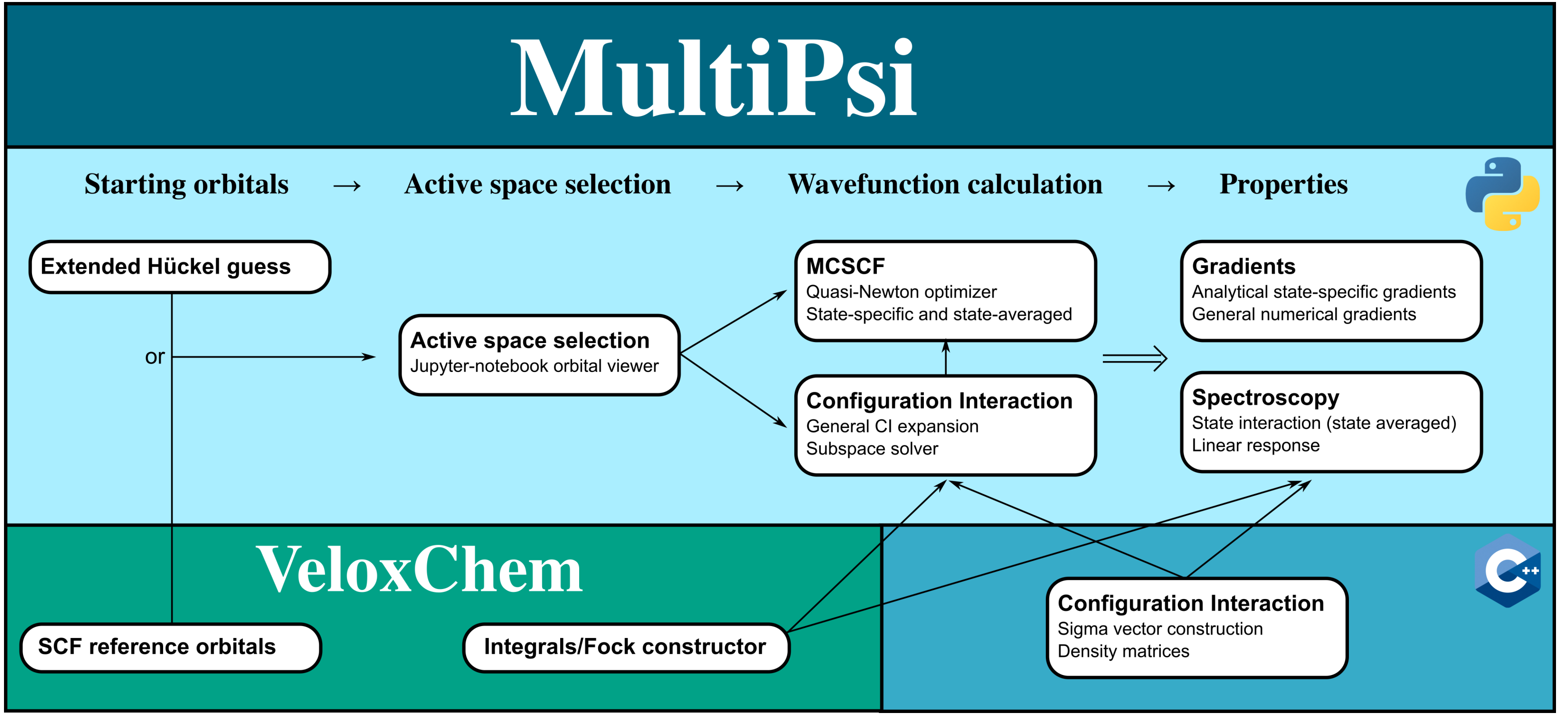
- Nonlinear response
  - Second harmonic generation (SHG)
  - Two photon absorption (TPA)

Other features

- Force field parameterization
  - Localized properties
  - RESP charges fitting
  - Dihedral angle potential fitting
- Excitonic coupling model
- Geometry optimization
  - Transition state search
  - Based on the geomeTRIC module
    - > https://github.com/leeping/geomeTRIC
- PES interpolation
- Vibrational spectroscopies






MultiPsi
Starting orbitals → Active space selection → Wavefunction calculation → Properties
Extended Hückel guess
or
Active space selection
Jupyter-notebook orbital viewer
MCSCF
Quasi-Newton optimizer
State-specific and state-averaged
Configuration Interaction
General CI expansion
Subspace solver
Gradients
Analytical state-specific gradients
General numerical gradients
Spectroscopy
State interaction (state averaged)
Linear response
VeloxChem
SCF reference orbitals
Integrals/Fock constructor
Configuration Interaction
Sigma vector construction
Density matrices

MultiPsi: a multiconfigurational extension of VeloxChem

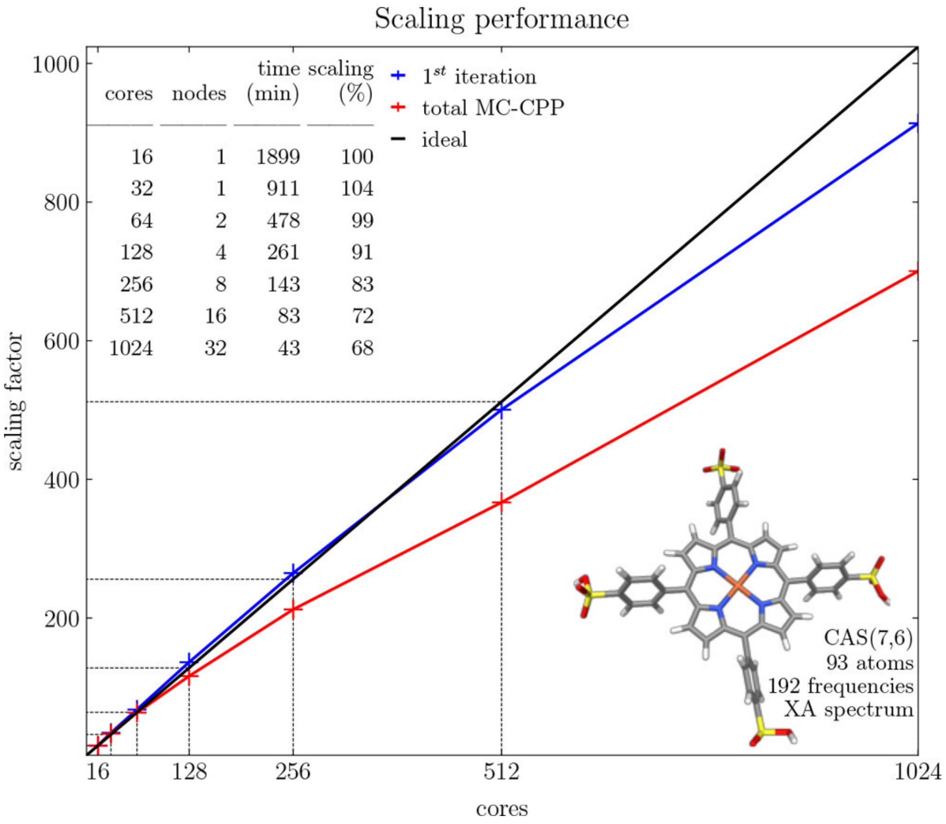
- MCSCF wavefunctions
  - Both state-specific and state-averaged
  - Quasi-Newton optimizer
  - Fock-matrix driven active AO-to-MO transformation

- Parallel general CI code
  - Record optimization of a CAS(22,22)
  - 418,151,049,316 Slater Determinants

- Linear response (standard and CPP)

Ref: https://doi.org/10.1002/wcms.1675

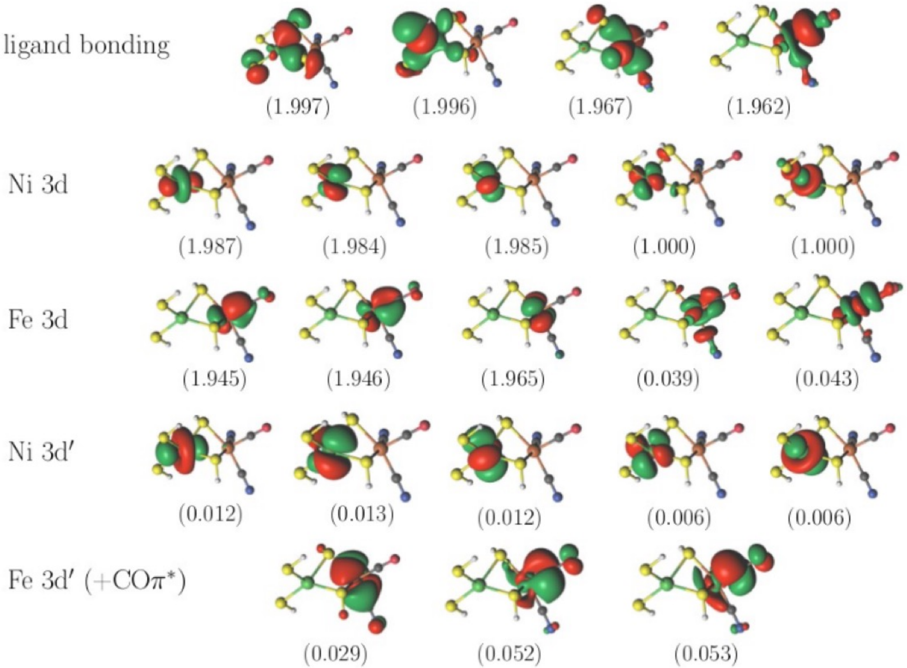
MultiPsi: benchmark



Scaling performance

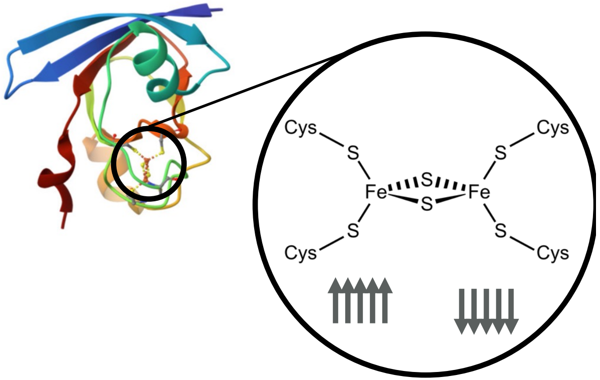
cores	nodes	time (min)	scaling (%)
16	1	1899	100
32	1	911	104
64	2	478	99
128	4	261	91
256	8	143	83
512	16	83	72
1024	32	43	68

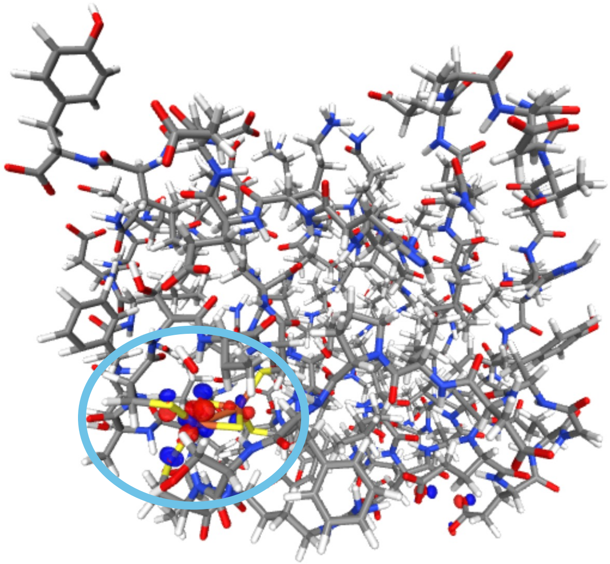
MultiPsi: $(SH)_2Ni(SH)_2Fe(CO)(CN)_2$



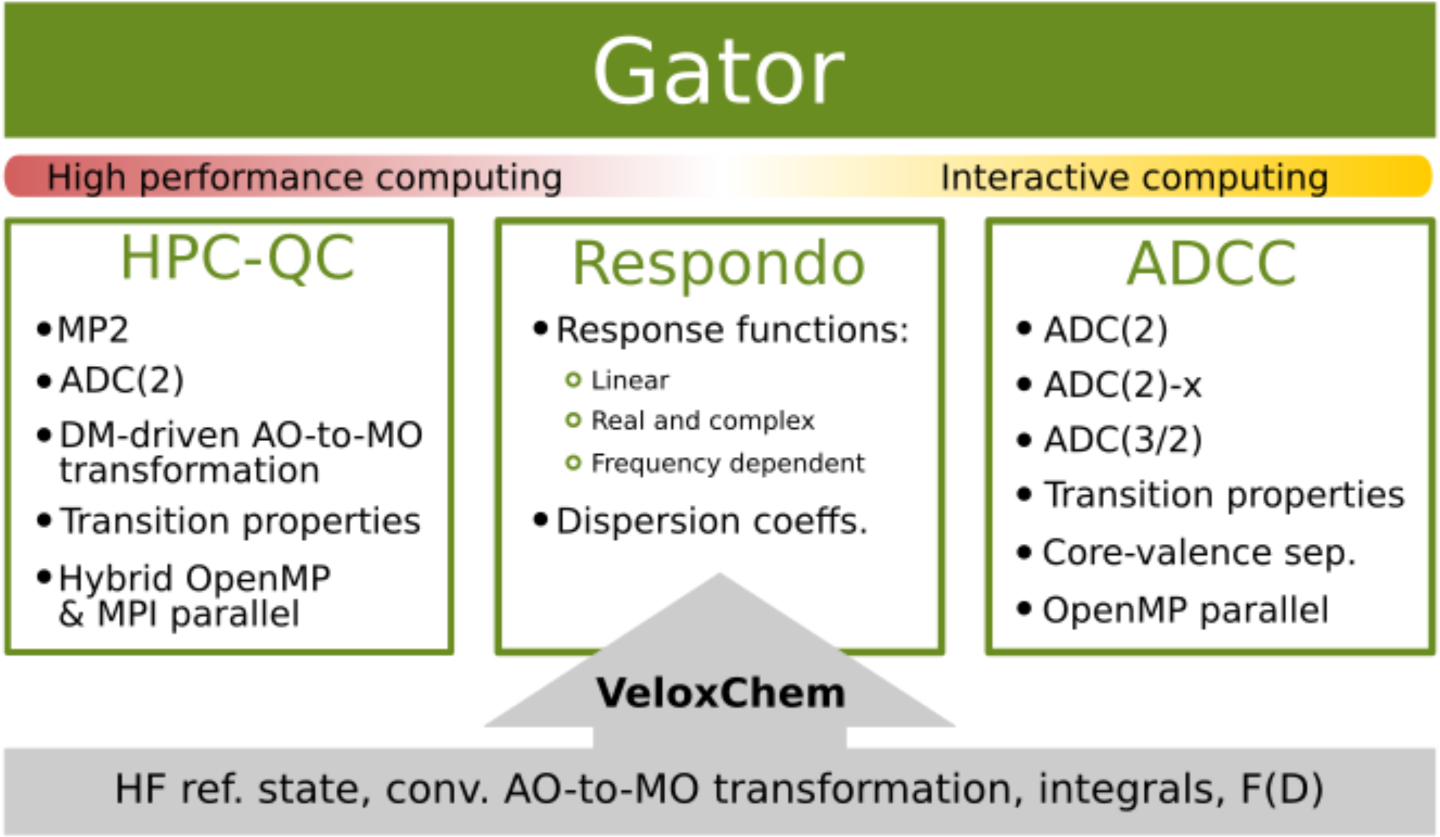
CAS(22,22) with 418,151,049,316 Slater Determinants

MultiPsi: 2Fe-2S cluster





- Variationally optimized MC-PDFT (a form of multiconfigurational DFT)
- Entire protein: 1447 atoms, 11986 basis functions, CAS(10,10)
- 1h per iteration on 8 x 32 cores (Tetralith)



Gator

High performance computing

Interactive computing

HPC-QC

- MP2
- ADC(2)
- DM-driven AO-to-MO transformation
- Transition properties
- Hybrid OpenMP & MPI parallel

Respondo

- Response functions:
  - Linear
  - Real and complex
  - Frequency dependent
- Dispersion coeffs.

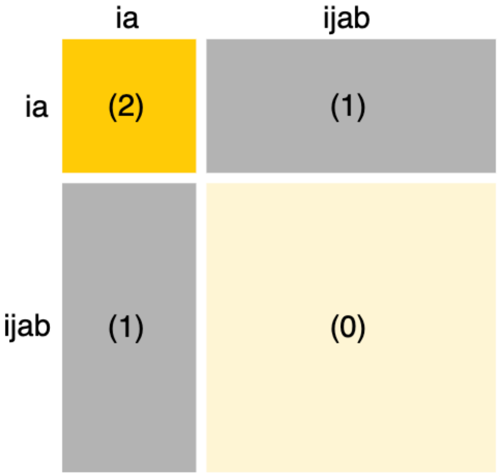
ADCC

- ADC(2)
- ADC(2)-x
- ADC(3/2)
- Transition properties
- Core-valence sep.
- OpenMP parallel

VeloxChem

HF ref. state, conv. AO-to-MO transformation, integrals, F(D)

The ADC(2) approach



	ia	ijab
ia	(2)	(1)
ijab	(1)	(0)

ADC(2)

$$\underline{\begin{pmatrix} \mathbf{M}_{\mathrm{SS}} & \mathbf{M}_{\mathrm{SD}} \\ \mathbf{M}_{\mathrm{DS}} & \enclose{circle}{\mathbf{M}_{\mathrm{DD}}} \end{pmatrix} \begin{pmatrix} \mathbf{X}_{\mathrm{S}} \\ \enclose{circle}{\mathbf{X}_{\mathrm{D}}} \end{pmatrix}} = \omega \begin{pmatrix} \mathbf{X}_{\mathrm{S}} \\ \mathbf{X}_{\mathrm{D}} \end{pmatrix}$$

Nonlinear eigensolver

σ-build

MO integrals

$$M^{(0)}_{ia,jb} = (\varepsilon_a - \varepsilon_i)\delta_{ab}\delta_{ij}$$

$$M^{(1)}_{ia,jb} = -\underline{\langle ja \| ib \rangle}$$

$$\begin{aligned} M^{(2)}_{ia,jb} = {} & \frac{1}{4}\delta_{ij} \sum_{c,k,l} \left[t_{ackl} \underline{\langle kl \| bc \rangle} + t^*_{bckl} \underline{\langle ac \| kl \rangle} \right] \\ & + \frac{1}{4}\delta_{ab} \sum_{c,d,k} \left[t_{cdik} \underline{\langle jk \| cd \rangle} + t^*_{jkcd} \underline{\langle cd \| ik \rangle} \right] \\ & - \frac{1}{2} \sum_{c,k} \left[t_{acik} \underline{\langle jk \| bc \rangle} + t^*_{jkbc} \underline{\langle ac \| ik \rangle} \right] \end{aligned}$$

$$M^{(1)}_{ia,klcd} = \underline{\langle kl \| id \rangle}\delta_{ac} - \underline{\langle kl \| ic \rangle}\delta_{ad} - \underline{\langle al \| cd \rangle}\delta_{ik} + \underline{\langle ak \| cd \rangle}\delta_{il}$$

$$M^{(1)}_{ijab,kc} = \underline{\langle kb \| ij \rangle}\delta_{ac} - \underline{\langle ka \| ij \rangle}\delta_{bc} - \underline{\langle ab \| cj \rangle}\delta_{ik} + \underline{\langle ab \| ci \rangle}\delta_{jk}$$

$$M^{(0)}_{ijab,klcd} = (\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j)\delta_{ac}\delta_{bd}\delta_{ik}\delta_{jl}$$

Jan Wenzel. *Development and Implementation of Theoretical Methods for the Description of Electronically Core-Excited States*. PhD thesis, Heidelberg University, 2016.

ADC(2) transition amplitudes

$$f^{(0)}_{jb,ai} = \delta_{ij}\delta_{ab}$$

$$f^{(1)}_{jb,ia} = -t_{ijab}$$

$$f^{(2)}_{ka,ij} = \frac{\delta_{kj}}{\varepsilon_a - \varepsilon_i}\left[\frac{1}{2}\sum_{jbc} t_{ijbc}\,\langle ja \parallel bc\rangle + \frac{1}{2}\sum_{jkb} t_{jkab}\,\langle jk \parallel ib\rangle\right]$$

$$f^{(1)}_{klab,ij} = -\delta_{ki}t_{jlab}$$

$$f^{(2)}_{ic,ab} = \frac{-\delta_{ac}}{\varepsilon_b - \varepsilon_i}\left[\frac{1}{2}\sum_{jcd} t_{ijcd}\,\langle jb \parallel cd\rangle + \frac{1}{2}\sum_{jkc} t_{jkbc}\,\langle jk \parallel ic\rangle\right]$$

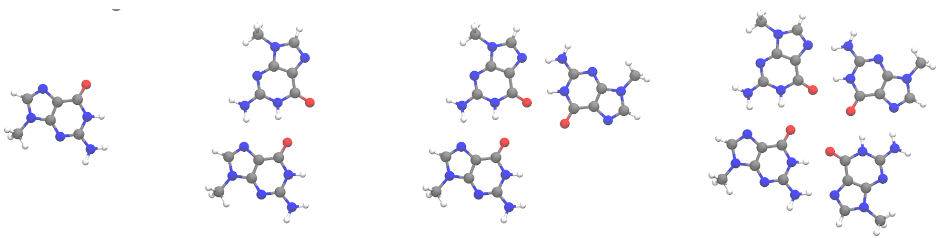
$$f^{(1)}_{ijcd,ab} = \delta_{ac}t_{ijbd}$$

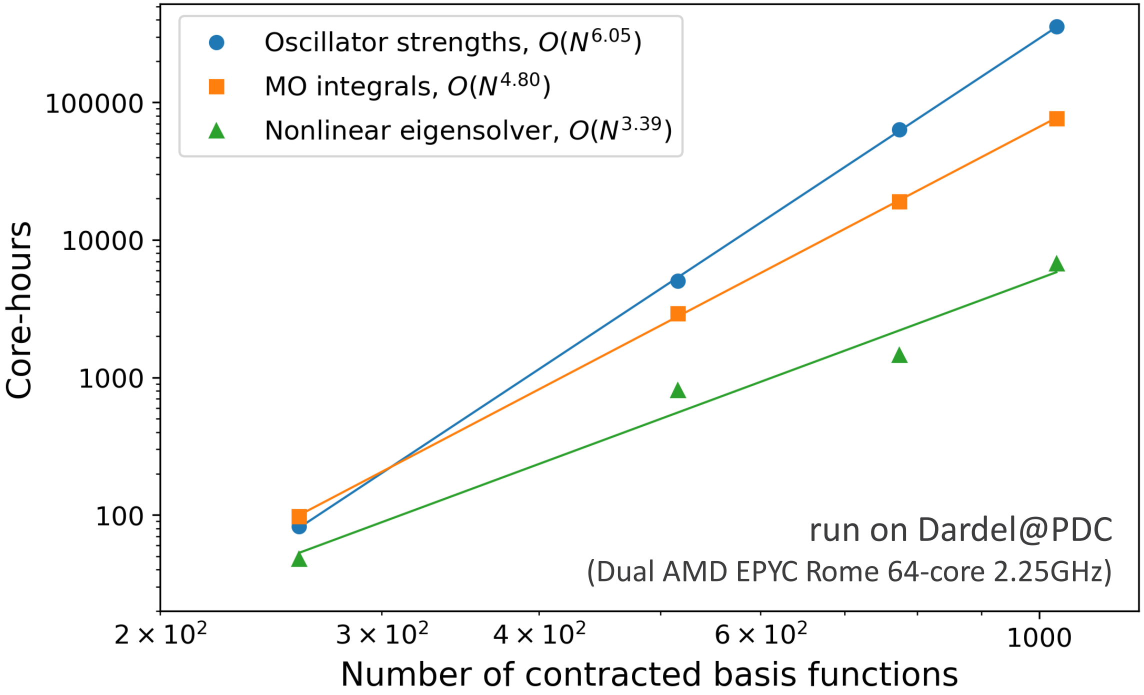
$$f^{(2)}_{jb,ai} = \frac{1}{2}\sum_{kc} t^{*}_{ikac}t_{jkbc} - \frac{1}{2}\delta_{ij}\gamma^{(2)}_{ab} + \frac{1}{2}\delta_{ab}\gamma^{(2)}_{ij}$$

$$f^{(2)}_{jb,ia} = -\frac{1}{\underline{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j}}\Big[(1-\hat{P}_{ij})(1-\hat{P}_{ab})\Big[\sum_{kc} t_{ikac}\,\langle kb \parallel jc\rangle\Big]$$

$$\underline{-\frac{1}{2}\sum_{cd} t_{ijcd}\,\langle ab \parallel cd\rangle} - \frac{1}{2}\sum_{kl} t_{klab}\,\langle kl \parallel ij\rangle\Big],$$

Benchmark





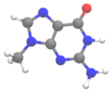
Oscillator strengths, $O(N^{6.05})$
MO integrals, $O(N^{4.80})$
Nonlinear eigensolver, $O(N^{3.39})$
Core-hours
100000
10000
1000
100
2×10^2
3×10^2
4×10^2
6×10^2
1000
Number of contracted basis functions

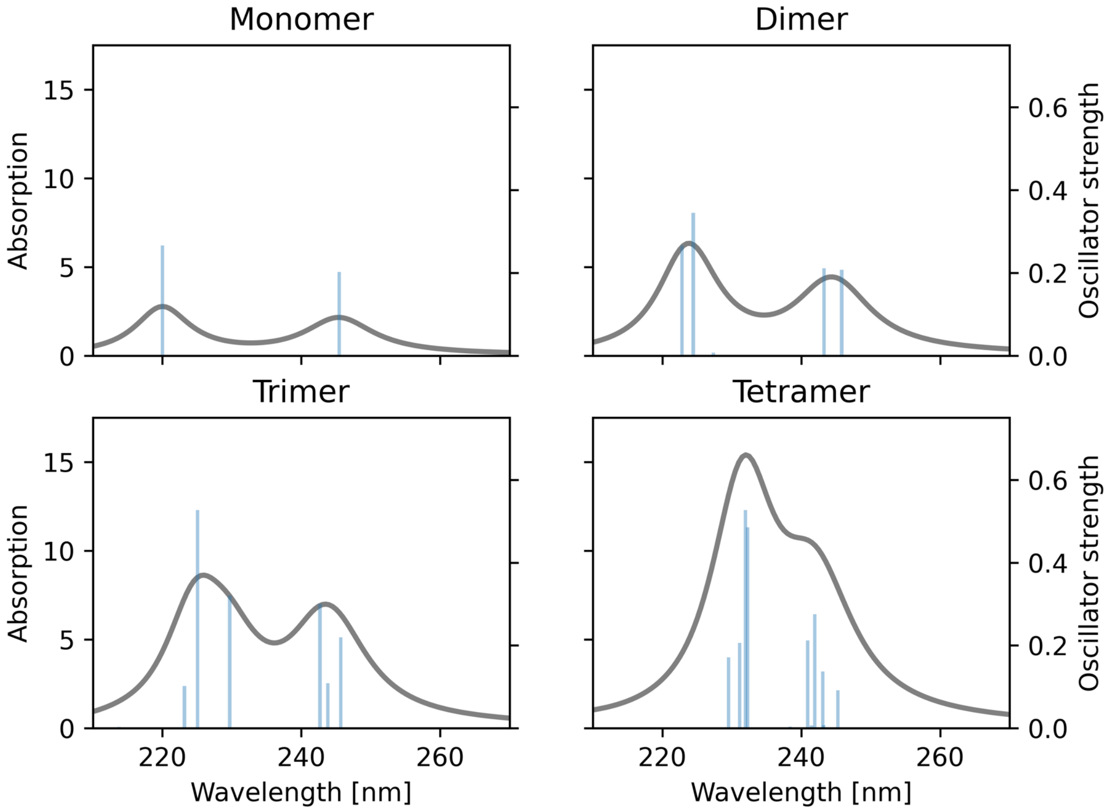
run on Dardel@PDC

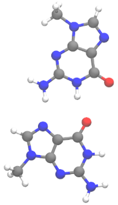
(Dual AMD EPYC Rome 64-core 2.25GHz)

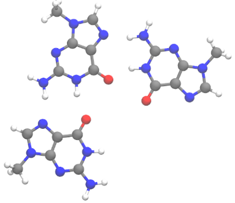
(basis set: 6-311G**)

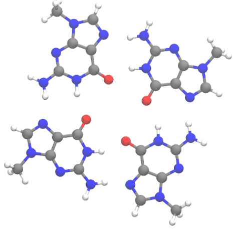
Absorption spectrum





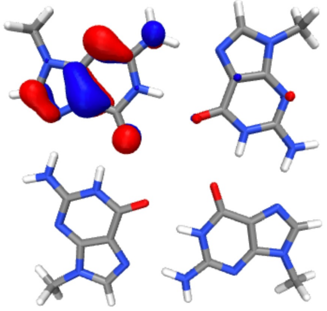






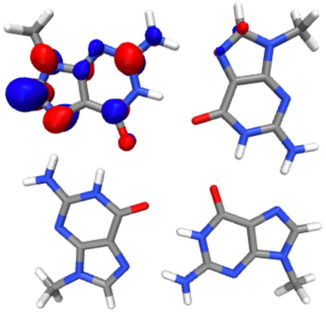
NTO analysis using single excitation vector

Tetramer
S_{10} (f = 0.5271)

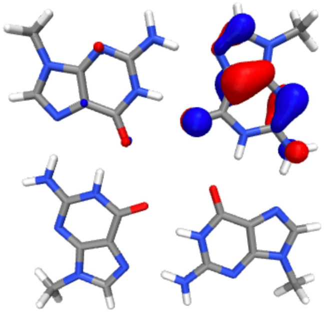


hole

λ=0.428 →

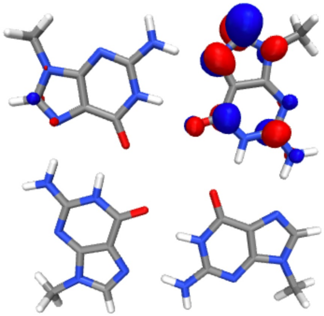


particle



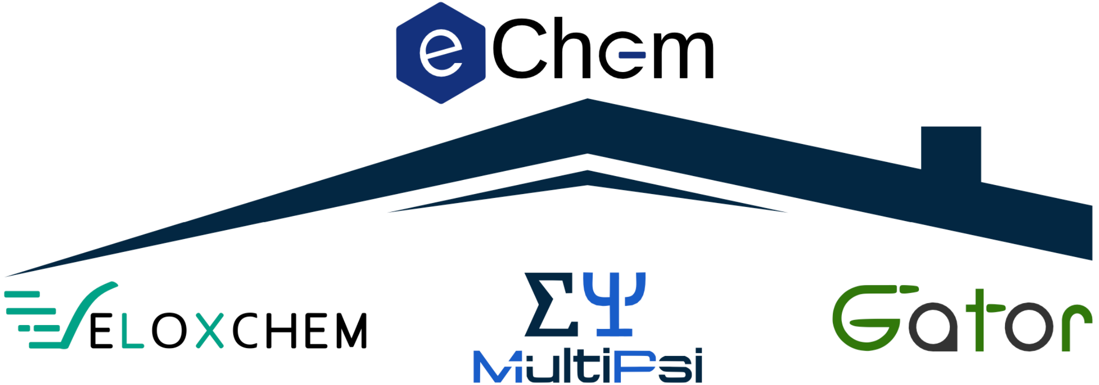
hole

λ=0.360 →



particle

The eChem project



- DFT and HF
- Response Theory
- Core Functionalities

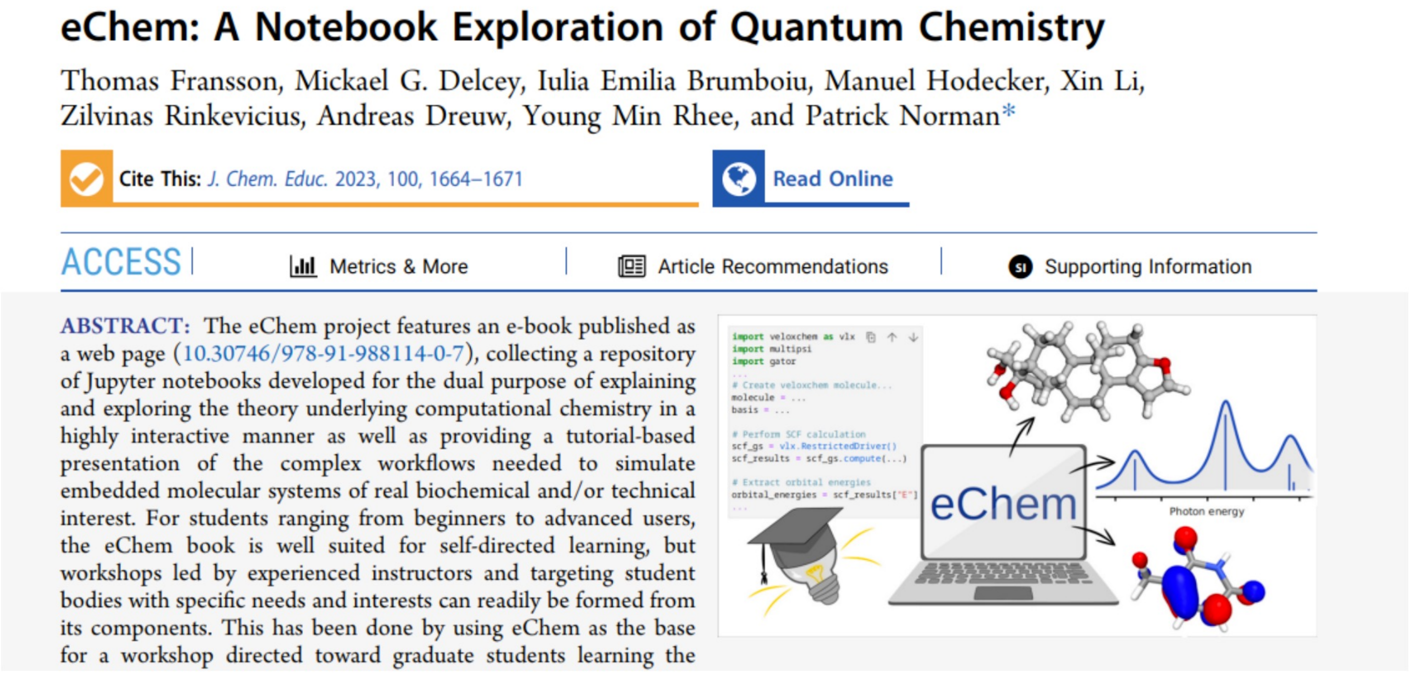
- CI
- MCSCF/MCDFT
- MRPT2

- MP2
- ADC hierarchy
- ADC–CPP

jupyter

Python-driven ***science***- and ***education***-enabling software platforms for quantum molecular modeling combining **interactive** & **HPC** features.

The eChem project



eChem: A Notebook Exploration of Quantum Chemistry

Thomas Fransson, Mickael G. Delcey, Iulia Emilia Brumboiu, Manuel Hodecker, Xin Li, Zilvinas Rinkevicius, Andreas Dreuw, Young Min Rhee, and Patrick Norman*

Cite This: *J. Chem. Educ.* 2023, 100, 1664–1671 | Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information

ABSTRACT: The eChem project features an e-book published as a web page (10.30746/978-91-988114-0-7), collecting a repository of Jupyter notebooks developed for the dual purpose of explaining and exploring the theory underlying computational chemistry in a highly interactive manner as well as providing a tutorial-based presentation of the complex workflows needed to simulate embedded molecular systems of real biochemical and/or technical interest. For students ranging from beginners to advanced users, the eChem book is well suited for self-directed learning, but workshops led by experienced instructors and targeting student bodies with specific needs and interests can readily be formed from its components. This has been done by using eChem as the base for a workshop directed toward graduate students learning the

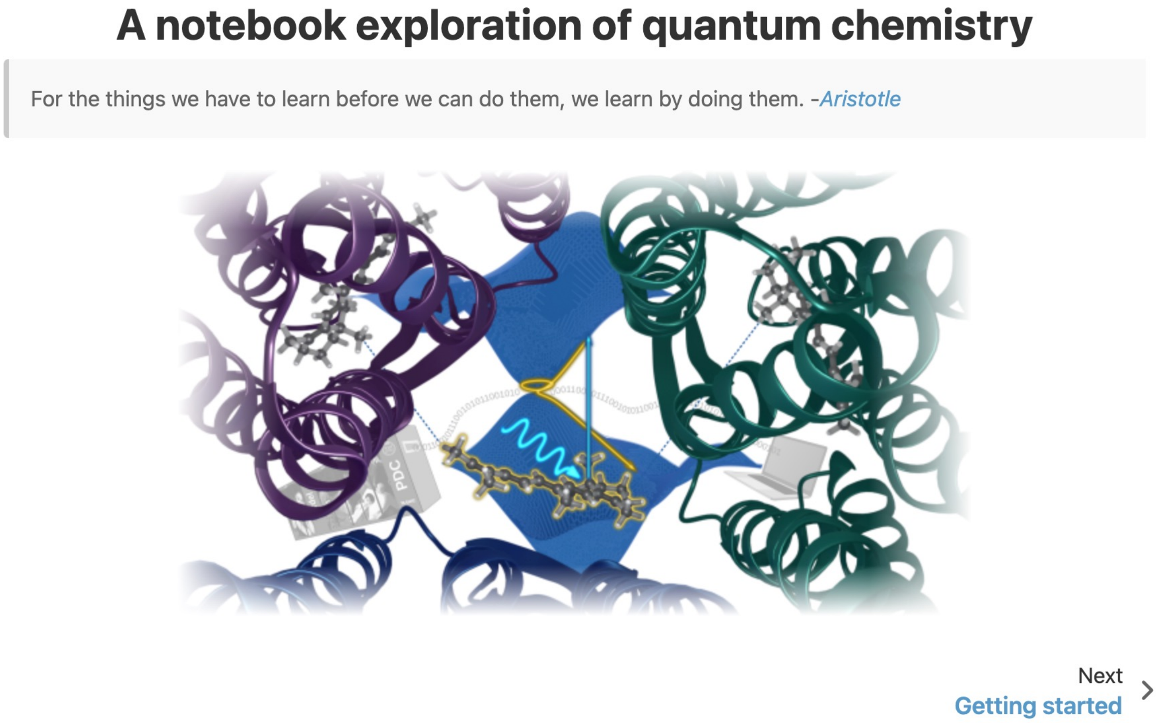
```
import veloxchem as vlx
import multipsi
import gator
...
# Create veloxchem molecule...
molecule = ...
basis = ...

# Perform SCF calculation
scf_gs = vlx.RestrictedDriver()
scf_results = scf_gs.compute(...)

# Extract orbital energies
orbital_energies = scf_results["E"]
...
```

Link: https://kthpanor.github.io/echem/ or simply http:://bit.ly/e-chem

The eChem project



A notebook exploration of quantum chemistry

> For the things we have to learn before we can do them, we learn by doing them. -*Aristotle*

Next
Getting started ›

Link: https://kthpanor.github.io/echem/ or simply http:://bit.ly/e-chem

Summary

- VeloxChem is a quantum chemistry software targeting HPC cluster resources
  - Python-driven science- and education-enabling software platform for quantum molecular modeling

- VeloxChem adopts a hybrid MPI/OpenMP and modular programming model
  - Emphasis on SIMD vectorization, multi-core scaling, low memory footprint and multi-Fock matrix handling

- VeloxChem supports a broad range of properties/spectra calculations
  - Linear/nonlinear response properties/spectra, conventional UV-Vis/X-ray spectra, geometry optimization, vibrational spectroscopies, etc.

- VeloxChem homepage: https://veloxchem.org/
- eChem homepage: https://kthpanor.github.io/echem/

Acknowledgements

Patrick Norman

Zilvinas Rinkevicius

Mickaël Delcey

Andreas Dreuw

Thank you!