A brief introduction to coupled cluster methods

Alejandro Gallo

TECHNISCHE UNIVERSITÄT WIEN

ABINIT School 2024
Thursday, 1st of February

Outline

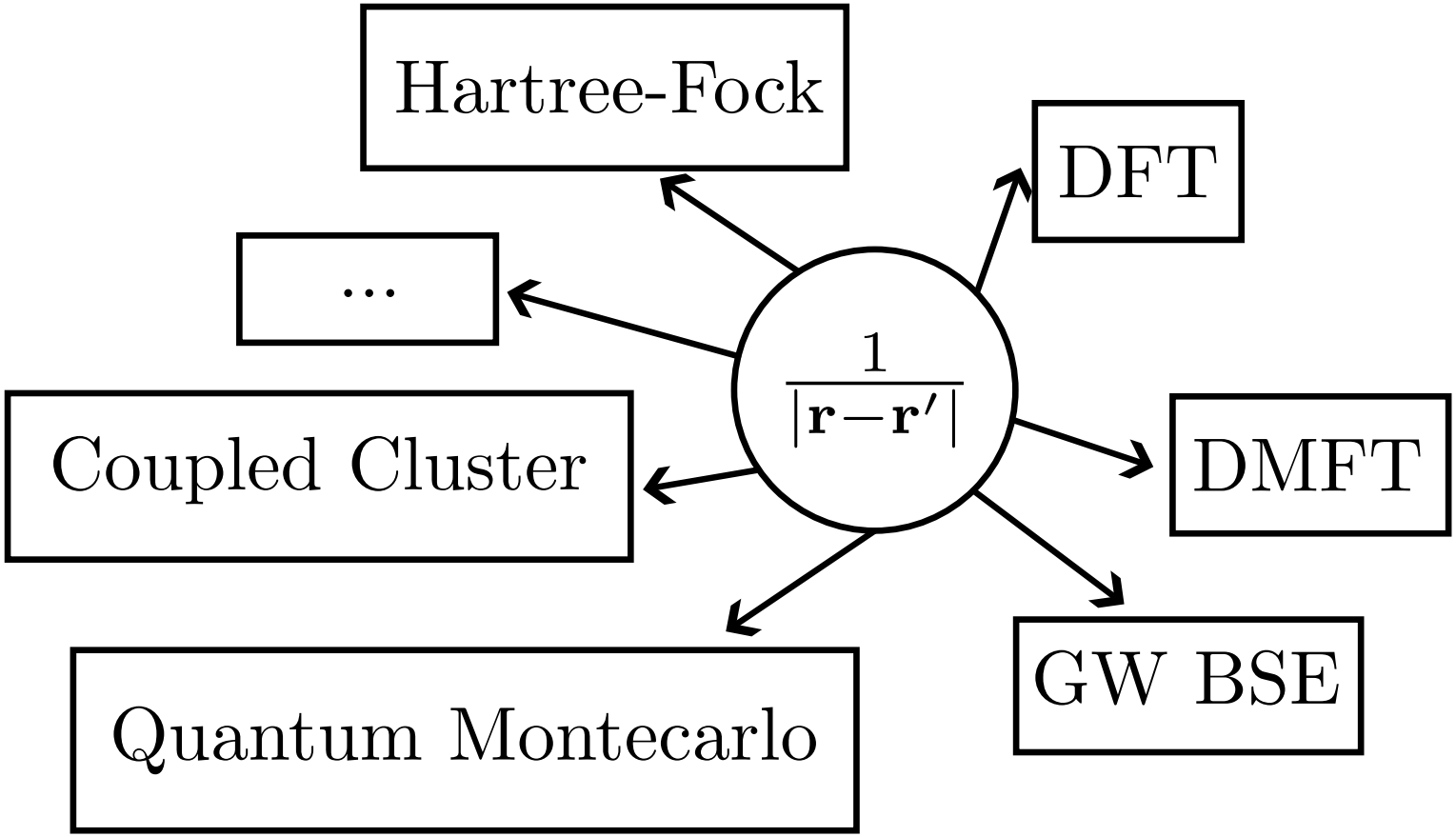
- Wavefunction basics
- The coupled cluster *Ansatz*
- Motivation: atomization energy errors with CCSD(T)
- Our code and its interfaces
- This afternoon: more information about the interface with `ABINIT`

What unites us all here?

What unites us all here?

$$\frac{1}{|\mathbf{r} - \mathbf{r}'|}$$

What unites us all here?



Hartree-Fock I

- Orbital basis $\{\varphi_1, \ldots, \varphi_N\}$
- Take one slater determinant

$$|\Psi\rangle = |\varphi_1 \cdots \varphi_N\rangle$$

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \frac{1}{\sqrt{N!}} \det \begin{pmatrix} \varphi_1(\mathbf{r}_1) & \cdots & \varphi_1(\mathbf{r}_N) \\ \vdots & \ldots & \vdots \\ \varphi_N(\mathbf{r}_1) & \cdots & \varphi_N(\mathbf{r}_N) \end{pmatrix}$$

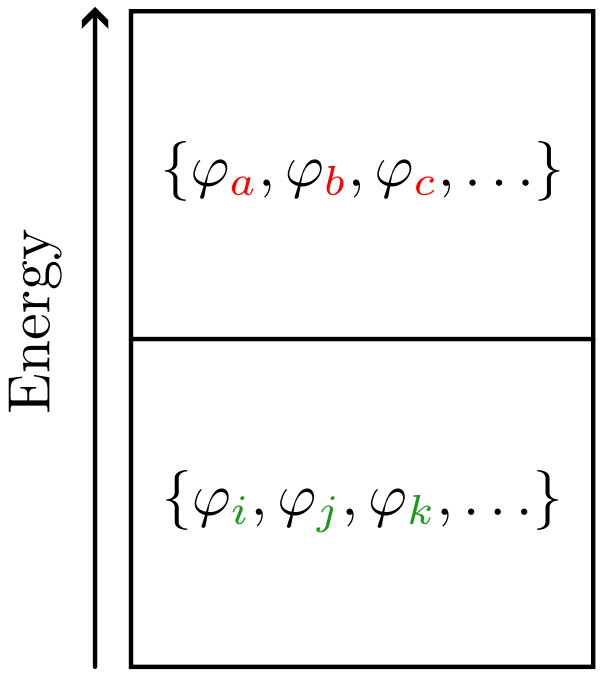
- Compute the energy of this determinant

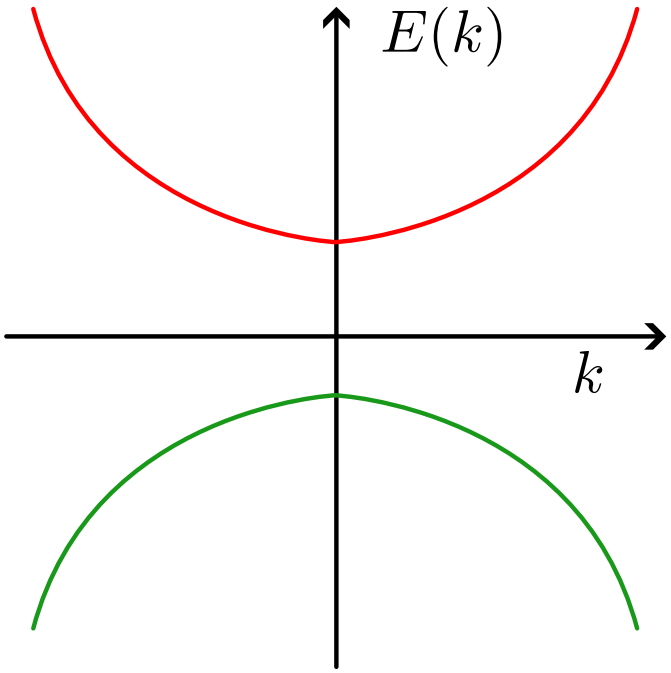
$$\begin{aligned} E &= \langle \Psi | \hat{H} | \Psi \rangle \\ &= \int \Psi^*(\mathbf{r}_1 \ldots) \hat{H}(\mathbf{r}_1 \ldots) \Psi(\mathbf{r}_1 \ldots) \, \mathrm{d}\mathbf{r}_1 \ldots \mathrm{d}\mathbf{r}_N \end{aligned}$$

- Minimize the energy in terms of the orbitals

$$E_{\mathrm{HF}} = \min_{\varphi_i} E$$

Hartree-Fock II





- Solving the self-consistent equations leads to
  occupied orbitals/bands $\{\varphi_i, \varphi_j, \varphi_k, \ldots\}$
  virtual orbitals/bands $\{\varphi_a, \varphi_b, \varphi_c, \ldots\}$

Hartree-Fock III

- A problem with HF: lack of correlation.
  Two electrons up and down spins

$$\text{orbitals} \rightarrow \{\varphi(\mathbf{r})\,|\alpha\rangle\,, \varphi(\mathbf{r})\,|\beta\rangle\}$$

- Slater determinant

$$\begin{aligned}\Psi(\mathbf{r}_1, \mathbf{r}_2) &= \frac{1}{\sqrt{2}} \det \begin{pmatrix} \varphi(\mathbf{r}_1)\,|\alpha\rangle & \varphi(\mathbf{r}_2)\,|\alpha\rangle \\ \varphi(\mathbf{r}_1)\,|\beta\rangle & \varphi(\mathbf{r}_2)\,|\beta\rangle \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \varphi(\mathbf{r}_1)\varphi(\mathbf{r}_2)(|\alpha\rangle\,|\beta\rangle - |\beta\rangle\,|\alpha\rangle)\end{aligned}$$

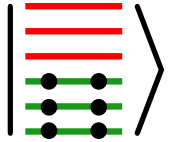
- Probability density is separable

$$|\Psi(\mathbf{r}_1, \mathbf{r}_2)|^2 = |\varphi(\mathbf{r}_1)|^2 \times |\varphi(\mathbf{r}_2)|^2$$

Adding a bit more correlation I

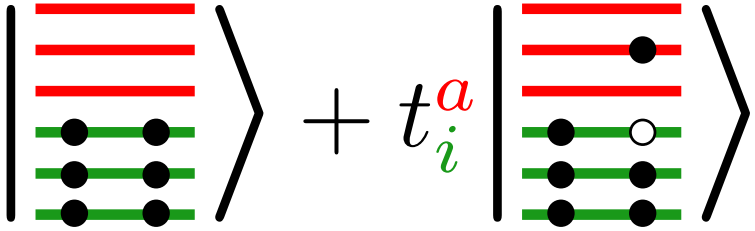
- One determinant is not enough, everything beyond having one determinant is called *Correlation Energy*
- Hartree-Fock determinant

$$|\mathrm{HF}\rangle = |\varphi_i \varphi_j \varphi_k \cdots\rangle =$$



- Add one singles excitation:

$$|\Psi_i^a\rangle = (1 + t_i^a \hat{c}_a^\dagger \hat{c}_i)\,|\mathrm{HF}\rangle$$



$$|\mathrm{HF}\rangle \quad + t_i^a \hat{c}_a^\dagger \hat{c}_i\,|\mathrm{HF}\rangle$$

- $|\Psi_i^a\rangle$ adds some correlation depending on a parameter t_i^a.

Adding a bit more correlation II

- Add all one singles excitations one after the other

$$
\begin{aligned}
|\Psi_{\mathrm{S}}\rangle &= (1 + t_i^a \hat{c}_a^\dagger \hat{c}_i)(1 + t_j^b \hat{c}_b^\dagger \hat{c}_j) \cdots |\mathrm{HF}\rangle \\
&= \prod_{\substack{i \in \text{Occupied} \\ a \in \text{Virtual}}} (1 + t_i^a \hat{c}_a^\dagger \hat{c}_i) \, |\mathrm{HF}\rangle \\
&= |\mathrm{HF}\rangle + \sum_{ia} t_i^a \hat{c}_a^\dagger \hat{c}_i \, |\mathrm{HF}\rangle + \sum_{ijab} t_i^a t_j^b \hat{c}_a^\dagger \hat{c}_i \hat{c}_b^\dagger \hat{c}_j \, |\mathrm{HF}\rangle + \cdots
\end{aligned}
$$

- But

$$
\hat{c}_a^\dagger \hat{c}_i \hat{c}_a^\dagger \hat{c}_i = 0 \quad \Longrightarrow \quad 1 + \hat{c}_a^\dagger \hat{c}_i = 1 + \hat{c}_a^\dagger \hat{c}_i + \frac{1}{2!} \left(\hat{c}_a^\dagger \hat{c}_i \right)^2 + \cdots = e^{\hat{c}_a^\dagger \hat{c}_i}
$$

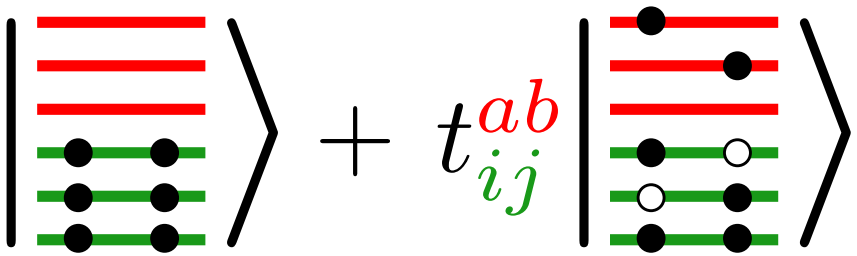
- Therefore

$$
|\Psi_{\mathrm{S}}\rangle = e^{\sum_{ai} t_i^a \hat{c}_a^\dagger \hat{c}_i} \, |\mathrm{HF}\rangle
$$

Adding a bit more correlation III

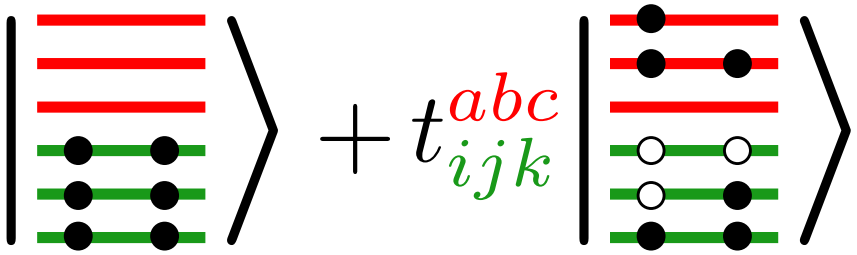
- One can also add pure two body excitations

$$|\Psi_{ij}^{ab}\rangle = (1 + t_{ij}^{ab}\hat{c}_a^\dagger\hat{c}_b^\dagger\hat{c}_j\hat{c}_i)\,|\mathrm{HF}\rangle$$



- Three body excitations

$$|\Psi_{ijk}^{abc}\rangle = (1 + t_{ijk}^{abc}\hat{c}_a^\dagger\hat{c}_b^\dagger\hat{c}_c^\dagger\hat{c}_k\hat{c}_j\hat{c}_i)\,|\mathrm{HF}\rangle$$



Coupled Cluster *Ansatz*

- Following the same procedure

$$|\Psi_{\mathrm{CC}}\rangle = \prod_{\substack{i \in \text{Occupied} \\ a \in \text{Virtual}}} (1 + t_i^a \hat{c}_a^\dagger \hat{c}_i) \prod_{a<b, i<j} (1 + t_{ij}^{ab} \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i) \cdots |\mathrm{HF}\rangle$$

$$= e^{\sum_{ai} t_i^a \hat{c}_a^\dagger \hat{c}_i + \sum_{abij} \frac{1}{4} t_{ij}^{ab} \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i + \cdots} |\mathrm{HF}\rangle$$

$$= e^{\hat{T}} |\mathrm{HF}\rangle$$

- The cluster operator

$$\hat{T} = \sum_{ai} t_i^a \hat{c}_a^\dagger \hat{c}_i + \sum_{abij} \frac{1}{2!^2} t_{ij}^{ab} \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i + \sum_{abcijk} \frac{1}{3!^2} t_{ijk}^{abc} \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_c^\dagger \hat{c}_k \hat{c}_j \hat{c}_i + \cdots$$

N_{el}	t_{ij}^{ab} (GB)	t_{ijk}^{abc} (TB)
50	0.7	5
100	10.5	383
150	53.0	4370

Wave function correlation vs DFT correlation I

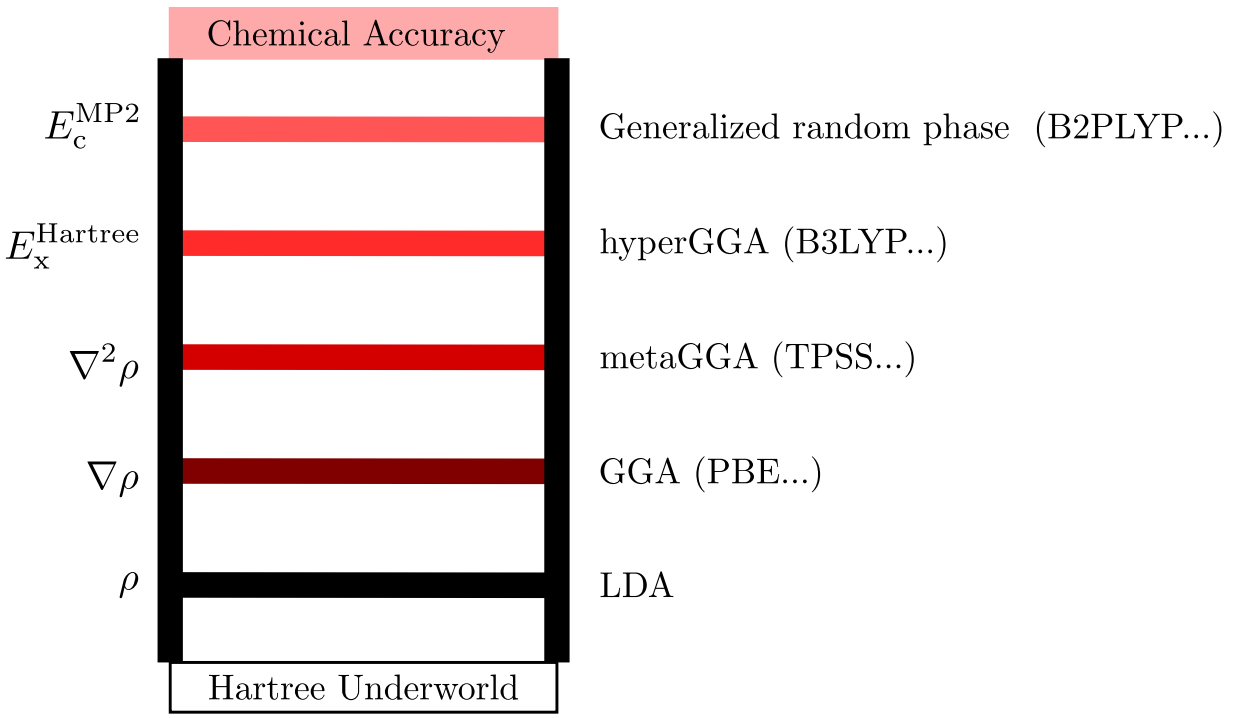
$$\left\{-\nabla^2 + v_{\text{ext}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}\mathbf{r}' + v_{\text{xc}}[n(\mathbf{r})]\right\}\varphi_n(\mathbf{r}) = \epsilon_n \varphi_n(\mathbf{r})$$

versus

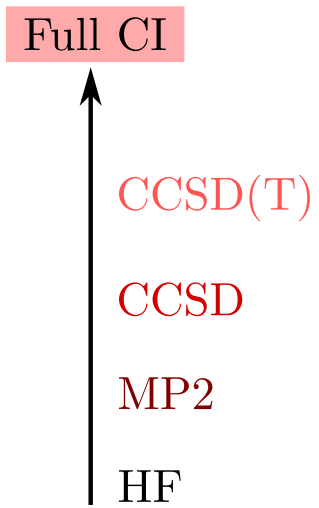
$$|\Psi\rangle = e^{\hat{T}}\,|\text{HF}\rangle = (1+\hat{C})\,|\text{HF}\rangle$$

Wave function correlation vs DFT correlation II

- ρ



- $|\Psi\rangle$



The Electronic Hamiltonian

- First quantization

$$\hat{H} = \underbrace{\sum_i \frac{1}{2}\hat{\mathbf{p}}_i^2}_{\text{El. Kinetic Energy}} - \underbrace{\sum_{i,J} \frac{Z_J}{|\mathbf{r}_i - \mathbf{r}_J|}}_{\text{Nuclei interaction}} + \underbrace{\frac{1}{2}\sum_{i \neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{Electron-electron}}$$

- Second quantization

$$\hat{H} = \sum_{p,q} h_q^p \hat{c}_p^\dagger \hat{c}_q + \sum_{p,q,r,s} \frac{1}{4} V_{rs}^{pq} \hat{c}_p^\dagger \hat{c}_q^\dagger \hat{c}_s \hat{c}_r$$

where p, q, r, s can be either occupied $(i, j, k, \ldots)$ or virtual $(a, b, c, \ldots)$

- Coulomb integrals V_{rs}^{pq}

$$V_{rs}^{pq} = \int \frac{\varphi_p^*(\mathbf{r}_1)\varphi_r(\mathbf{r}_1)\varphi_q^*(\mathbf{r}_2)\varphi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2$$

Projection method for coupled cluster I

- insert ansatz into stationary Schrödinger equation

$$\hat{H} e^{\hat{T}} |\mathrm{HF}\rangle = E\, e^{\hat{T}} |\mathrm{HF}\rangle$$
$$e^{-\hat{T}} \hat{H} e^{\hat{T}} |\mathrm{HF}\rangle = e^{-\hat{T}} E\, e^{\hat{T}} |\mathrm{HF}\rangle = E\, |\mathrm{HF}\rangle$$

- project onto GS Slater determinant for correlated energy

$$\langle \mathrm{HF}| e^{-\hat{T}} \hat{H}\, e^{\hat{T}} |\mathrm{HF}\rangle = \langle \mathrm{HF}| E\, |\mathrm{HF}\rangle = E$$

- Orbital basis

  occupied $\{\varphi_i, \varphi_j, \ldots\}$
  virtual $\{\varphi_a, \varphi_b, \ldots\}$

- Determinant basis

  singles $\left\{ |{}^{a}_{i}\rangle = \hat{c}_a^\dagger \hat{c}_i\, |\mathrm{HF}\rangle \right\}$
  doubles $\left\{ |{}^{ab}_{ij}\rangle = \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i\, |\mathrm{HF}\rangle \right\}$

Projection method for coupled cluster II

$$e^{-\hat{T}}\hat{H}e^{\hat{T}}\left|\mathrm{HF}\right\rangle = e^{-\hat{T}}E\,e^{\hat{T}}\left|\mathrm{HF}\right\rangle = E\left|\mathrm{HF}\right\rangle$$

- project onto determinant basis elements for coefficient equations

$$\left\langle{}^{a}_{i}\right|e^{-\hat{T}}\hat{H}\,e^{\hat{T}}\left|\mathrm{HF}\right\rangle = \left\langle{}^{a}_{i}\right|E\left|\mathrm{HF}\right\rangle = 0$$
$$\left\langle{}^{ab}_{ij}\right|e^{-\hat{T}}\hat{H}\,e^{\hat{T}}\left|\mathrm{HF}\right\rangle = \left\langle{}^{ab}_{ij}\right|E\left|\mathrm{HF}\right\rangle = 0$$

Coupled cluster equations (CCSD)

- The Schrödinger equation is non-linear now

$$\hat{H}e^{\hat{T}}|\mathrm{HF}\rangle = E_{\mathrm{CC}}e^{\hat{T}}|\mathrm{HF}\rangle \quad \Rightarrow \quad \underbrace{e^{-\hat{T}}\hat{H}e^{\hat{T}}}_{\bar{H}}|\mathrm{HF}\rangle = E_{\mathrm{CC}}|\mathrm{HF}\rangle$$

- Solve it projectively

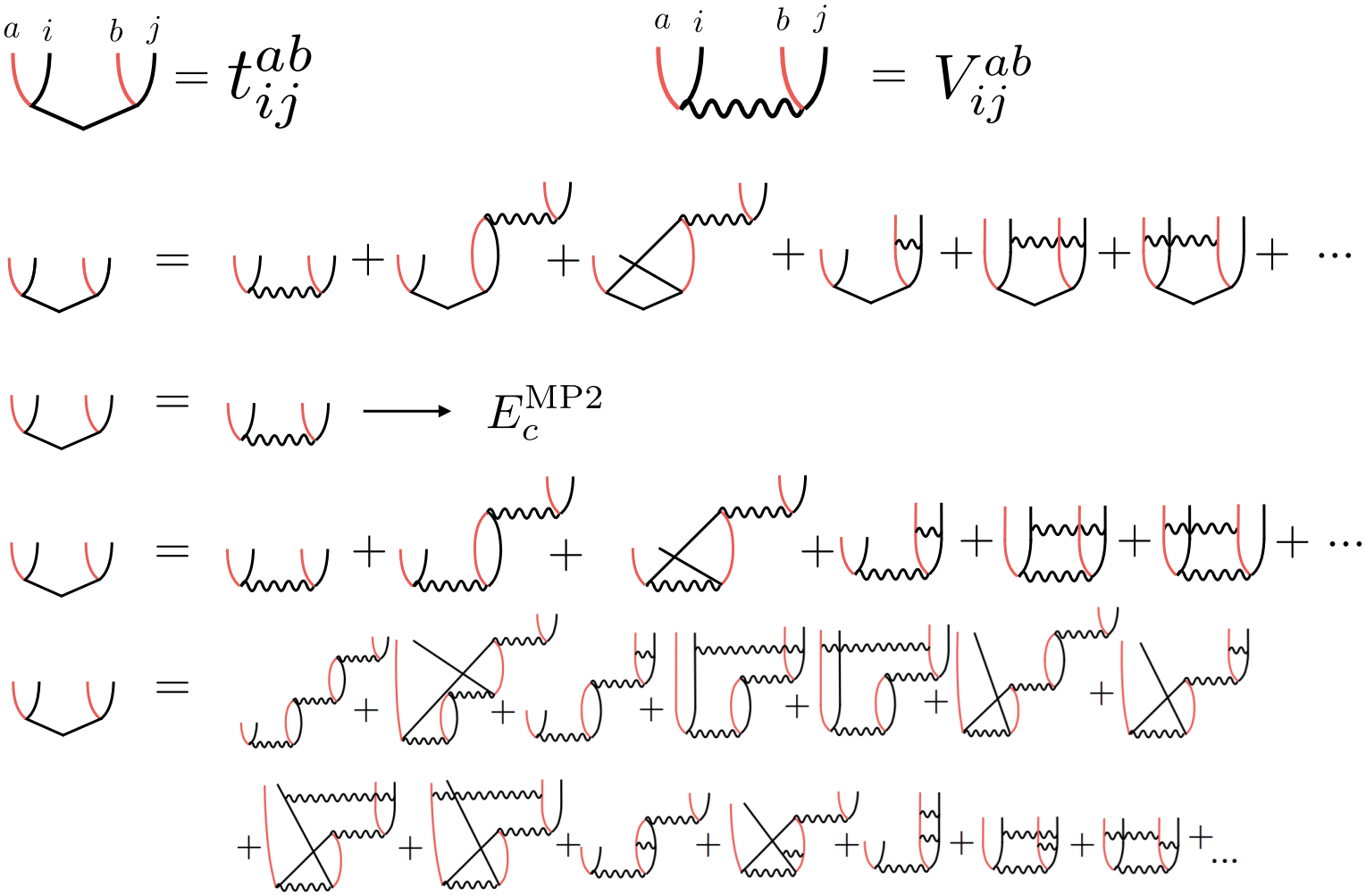
$$\begin{aligned}
E_{\mathrm{CC}} &= \langle \mathrm{HF}|\bar{H}|\mathrm{HF}\rangle &&= \textstyle\sum_{i,j,a,b} \frac{1}{4} V_{ab}^{ij} T_{ij}^{ab} + \dots \\
0 &= \langle {}_{i}^{a}|\, \bar{H}|\mathrm{HF}\rangle &&= \textstyle\sum_{b} f_{b}^{a} t_{i}^{b} - \sum_{j} f_{i}^{j} t_{j}^{a} + \dots \\
0 &= \langle {}_{ij}^{ab}|\, \bar{H}|\mathrm{HF}\rangle &&= \textstyle\sum_{c} f_{c}^{b} t_{ij}^{ac} + \sum_{ab} V_{cd}^{ab} t_{ij}^{cd} + \dots
\end{aligned}$$

- Start with an intial guess of $t_{ij}^{ab} = V_{ij}^{ab}$ and iterate (and hope for the best)
- Scaling $\mathcal{O}(N^6)$

Example of t_{ij}^{cd} equations (sum over repeated indices)

$$
\begin{aligned}
t_{ij}^{cd} = {} & V_{ij}^{cd} - t_m^c V_{ij}^{md} + t_m^d V_{ij}^{mc} + t_j^e V_{ie}^{cd} - t_i^e V_{je}^{cd} - f_i^m t_{mj}^{cd} \\
& + f_j^m t_{mi}^{cd} - f_e^d t_{ij}^{ec} + f_e^c t_{ij}^{ed} + \frac{1}{2} t_{mn}^{cd} V_{ij}^{mn} + t_{nj}^{ec} V_{ie}^{nd} - t_{nj}^{ed} V_{ie}^{nc} - t_{ni}^{ec} V_{je}^{nd} + t_{ni}^{ed} V_{je}^{nc} + \frac{1}{2} t_{ij}^{ef} V_{ef}^{cd} \\
& + t_m^c t_n^d V_{ij}^{mn} - t_j^e t_n^c V_{ie}^{nd} + t_j^e t_n^d V_{ie}^{nc} + t_i^e t_n^c V_{je}^{nd} \\
& - t_i^e t_n^d V_{je}^{nc} + t_i^e t_j^f V_{ef}^{cd} - f_f^m t_{mj}^{cd} t_i^f + f_f^m t_{mi}^{cd} t_j^f + f_f^m t_{ij}^{fc} t_m^d - f_f^m t_{ij}^{fd} t_m^c \\
& + \frac{1}{2} t_{mn}^{cd} t_j^g V_{ig}^{mn} - \frac{1}{2} t_{mn}^{cd} t_i^g V_{jg}^{mn} - t_{nj}^{ec} t_o^d V_{ie}^{no} + t_{nj}^{ed} t_o^c V_{ie}^{no} + t_{ni}^{ec} t_o^d V_{je}^{no} - t_{ni}^{ed} t_o^c V_{je}^{no} \\
& - t_{mj}^{cd} t_o^f V_{if}^{mo} + t_{mi}^{cd} t_o^f V_{jf}^{mo} - t_{nj}^{ec} t_i^g V_{eg}^{nd} + t_{nj}^{ed} t_i^g V_{eg}^{nc} + t_{ni}^{ec} t_j^g V_{eg}^{nd} - t_{ni}^{ed} t_j^g V_{eg}^{nc} - \frac{1}{2} t_{ij}^{ef} t_o^c V_{ef}^{od} \\
& + \frac{1}{2} t_{ij}^{ef} t_o^d V_{ef}^{oc} + t_{ij}^{ec} t_o^f V_{ef}^{od} - t_{ij}^{ed} t_o^f V_{ef}^{oc} + \frac{1}{2} t_{ij}^{ed} t_{op}^{fc} V_{ef}^{op} \\
& - \frac{1}{2} t_{ij}^{ec} t_{op}^{fd} V_{ef}^{op} + \frac{1}{4} t_{ij}^{ef} t_{op}^{cd} V_{ef}^{op} - \frac{1}{2} t_{mi}^{cd} t_{pj}^{fg} V_{fg}^{mp} + \frac{1}{2} t_{mj}^{cd} t_{pi}^{fg} V_{fg}^{mp} - t_{ni}^{ed} t_{pj}^{gc} V_{eg}^{np} \\
& + t_{ni}^{ec} t_{pj}^{gd} V_{eg}^{np} + t_j^e t_n^c t_o^d V_{ie}^{no} - t_i^e t_n^c t_o^d V_{je}^{no} - t_i^e t_j^f t_o^c V_{ef}^{od} \\
& + t_i^e t_j^f t_o^d V_{ef}^{oc} + \frac{1}{2} t_i^e t_j^f t_{op}^{cd} V_{ef}^{op} + t_i^e t_n^d t_{pj}^{gc} V_{eg}^{np} - t_i^e t_n^c t_{pj}^{gd} V_{eg}^{np} \\
& - t_j^e t_n^d t_{pi}^{gc} V_{eg}^{np} + t_j^e t_n^c t_{pi}^{gd} V_{eg}^{np} + t_i^e t_o^f t_{pj}^{cd} V_{ef}^{op} - t_j^e t_o^f t_{pi}^{cd} V_{ef}^{op} \\
& + \frac{1}{2} t_m^c t_n^d t_{ij}^{gh} V_{gh}^{mn} - t_m^d t_o^f t_{ij}^{hc} V_{fh}^{mo} + t_m^c t_o^f t_{ij}^{hd} V_{fh}^{mo} + t_i^e t_j^f t_o^c t_p^d V_{ef}^{op}
\end{aligned}
$$

Diagramatic fractal structure



Amplitudes with k-points

- Bloch orbitals $\varphi_i \to \varphi_{i\mathbf{k}_i}$
- Translational symmetry reduces memory constraints for integrals (block sparsity)

$$V_{r\mathbf{k}_r\ s\mathbf{k}_s}^{p\mathbf{k}_p\ q\mathbf{k}_q}\delta(\mathbf{k}_p + \mathbf{k}_q = \mathbf{k}_r + \mathbf{k}_s + \mathbf{G})$$

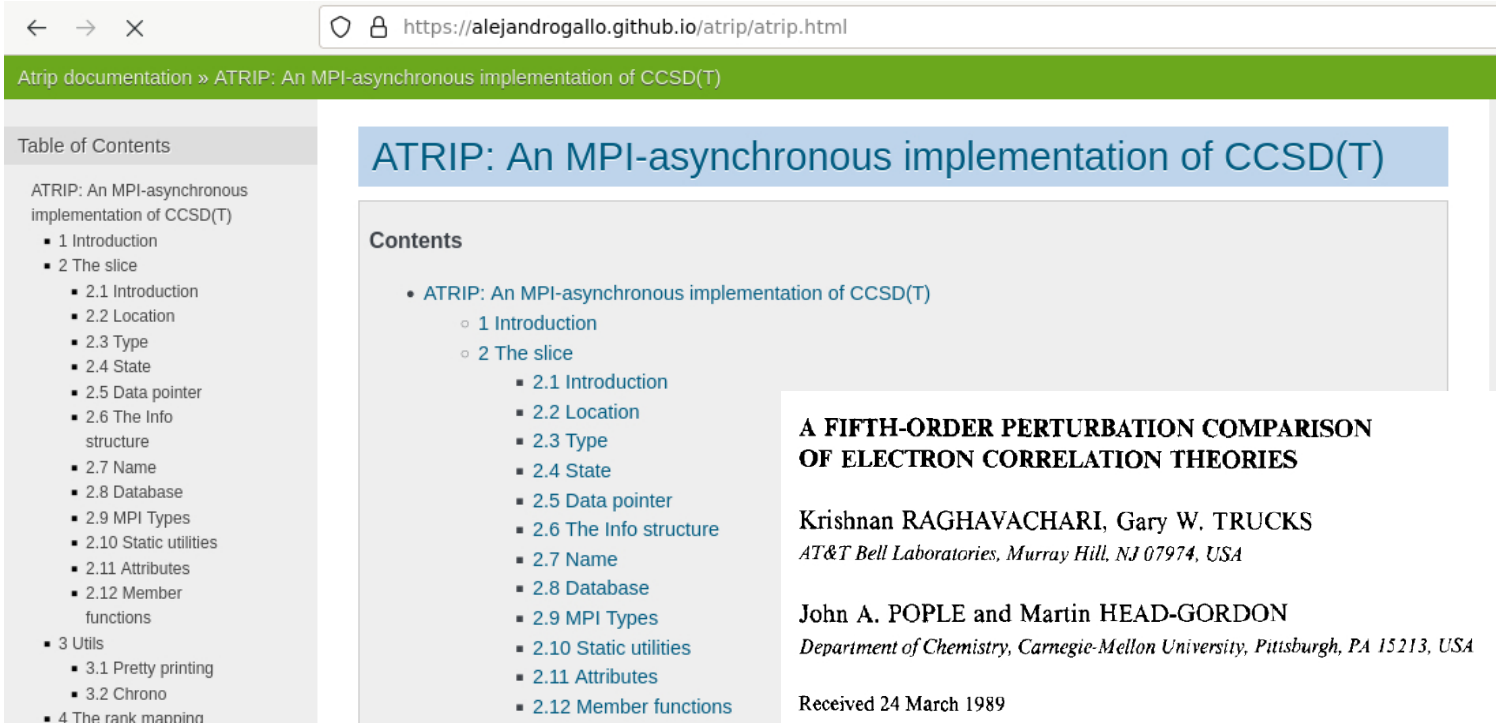
  with $\mathbf{G}$ a reciprocal lattice vector
- $t_{i\mathbf{k}_i\ j\mathbf{k}_j}^{a\mathbf{k}_a\ b\mathbf{k}_b}$ inherits block sparse structure
- Adapt equations to obey momentum conservation
- Adapt underlying tensor libraries to efficiently deal with block sparsity

CCSD(T)

- CCSDT too expensive for most electron sizes ($\mathcal{O}(N^8)$)

N_{el}	t_{ij}^{ab} (GB)	t_{ijk}^{abc} (TB)
50	0.7	5
100	10.5	383
150	53.0	4370

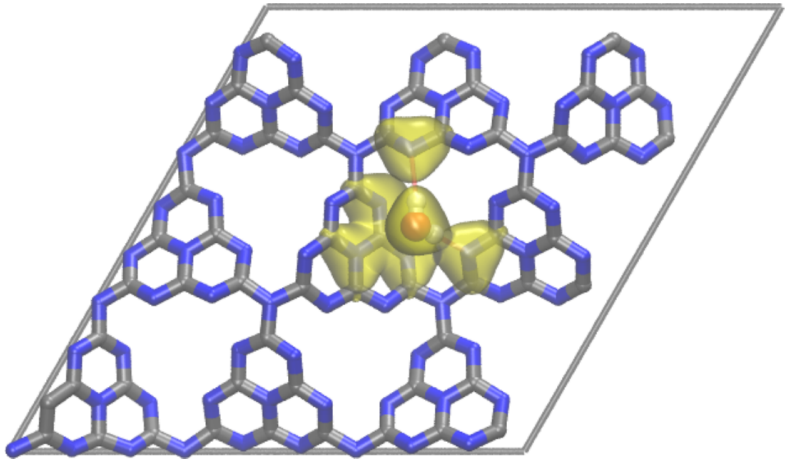
- Add triples correlation in a perturbative, *one-shot* fashion
- Exascale (GPU & CPU) implementation `ATRIP` $\mathcal{O}(N^7)$



CCSD(T): atomization energy errors

$$H\Psi(\mathbf{r}_1, ..., \mathbf{r}_N) = E\Psi(\mathbf{r}_1, ...\mathbf{r}_N)$$

CC4S ATRIP



Atomization energy errors

ΔE	PBE	SCAN	CCSD(T)
N2	-65.04	35.41	0.47
H2	20.51	7.79	0.00
F2	-58.41	-1.46	2.11
O2	-96.94	-28.38	2.39
CO	-39.38	26.48	-0.06
C2H2	-40.12	16.69	-0.80
CCH	-46.58	-0.03	2.84
CH2	-15.49	-25.29	0.12
CO2	-109.58	-4.93	-0.38
H2O2	-55.08	13.15	0.58
H2O	-5.03	19.28	0.02
HCO	-67.33	-9.42	0.41
HF	-1.59	14.94	0.05
NO	-82.85	6.60	1.53
OH	-11.08	-4.56	0.35
HNO	-74.63	25.14	1.09
CN	-69.55	13.10	6.57
HCN	-55.09	24.25	-0.17
CF	-51.94	-10.92	0.40
NH2	-25.49	-9.97	0.54
NH	-22.63	-11.15	0.51
rms	56.5	17.5	1.8

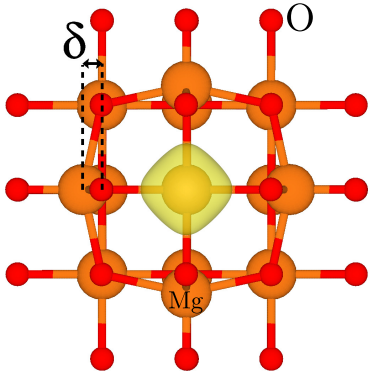
Room temp. 2.5 kJ/mol

Bonus: excited states for F-centers in MgO, SrO, CaO[1]

- Equation of motion formalism, (diagonalize $e^{-\hat{T}}\hat{H}e^{\hat{T}}$)

$$|R_n\rangle = \hat{R}_n e^{\hat{T}} |\mathrm{HF}\rangle \quad \hat{H}|R_n\rangle = E_n |R_n\rangle$$

- Absorption and emission energies



System	Method	Absorption (eV)	Emission (eV)
CaO	EOM-CCSD@HF	3.13	1.93
	Experiment	3.02	1.93
	Experiment	3.1	2.05 – 2.01
	TDDFT@B3LYP	3.52	2.1
	G_0W_0@PBE	3.20	
SrO	EOM-CCSD@HF	2.34	1.2
	Experiment	2.4	
MgO	EOM-CCSD@HF	5.28	3.66
	Experiment	5.0	2.4
	QMC	5.0(1)	3.8(1)
	CASPT2	5.44	4.09
	G_0W_0@LDA0-BSE	4.95	3.4

[1] A. Gallo et al. JCP, 154, 234103, 2021

./thank-you\!

TECHNISCHE UNIVERSITÄT WIEN

European Research Council
Established by the European Commission

NOMAD
NOVEL MATERIALS DISCOVERY

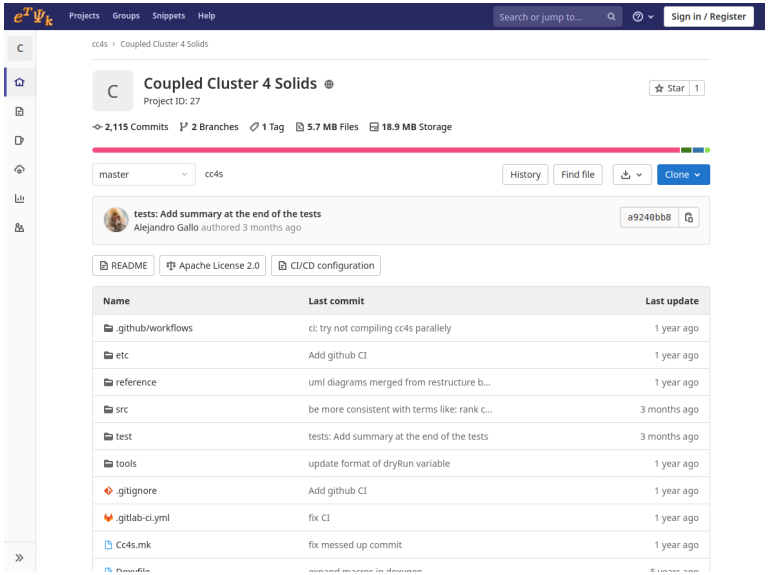
Hands-on session with the `ABINIT` interface

- We will calculate the atomization energy of `Be` as a test to show the workflow

  ABINIT DFT
  CC4S CCSD
  CC4S CCSD(T)

./CC4S/

- Massively parallel `C++` code for coupled cluster for solids.
- Interfaces with other codes exist, more on this later
- © `APACHEv2` license
- `github:cc4s/cc4s`



./CC4S/docs

- Comprehensive documentation
- https://manuals.cc4s.org/user-manual

Cc4s | Home | Sitemap | Algorithms | Objects

Table of Contents

1. Brief description
2. Algorithm call
3. Algorithm input
   3.1. method
   3.2. linearzied
   3.3. integralsSliceSize
   3.4. maxIterations
   3.5. energyConvergence
   3.6. amplitudesConvergence
   3.7. initialAmplitudes
   3.8. mixer
      3.8.1. DiisMixer
      3.8.2. LinearMixer
4. Algorithm output
   4.1. Sample `stdout`
5. Sample `yaml` output
6. Computational complexity
7. Theory
   7.1. Coupled-Cluster Singles Doubles (`Ccsd`)
   7.2. Direct-Ring Couple-Cluster Doubles (`Drccd`)
8. Literature

CoupledCluster

1. Brief description

This algorithm solves the CCSD amplitude equations using the Coulomb integrals and Hartree–Fock eigenenergies.

2. Algorithm call

A typical input file snippet to call the `CoupledCluster` algorithm is given below.

```
- name: CoupledCluster
  in:
    method: Ccsd
    integralsSliceSize: 100
    slicedEigenEnergies: EigenEnergies
    coulombIntegrals: CoulombIntegrals
    slicedCoulombVertex: CoulombVertex
    maxIterations: 20
    energyConvergence: 1.0E-4
    amplitudesConvergence: 1.0E-4
    mixer:
      type: DiisMixer
      maxResidua: 5
    #or for instance a linear mixer
    #mixer:
    #  type: LinearMixer
    #  ratio: 1.0
  out:
    amplitudes: Amplitudes
```

3. Algorithm input

Table 1: Input keywords

Keyword	Value
`method`	Type of coupled cluster algorithm
`linearized`	Use linearized or fully non-linear coupled cluster
`integralsSliceSize`	Integral slice size

./CC4S/libs/CTF

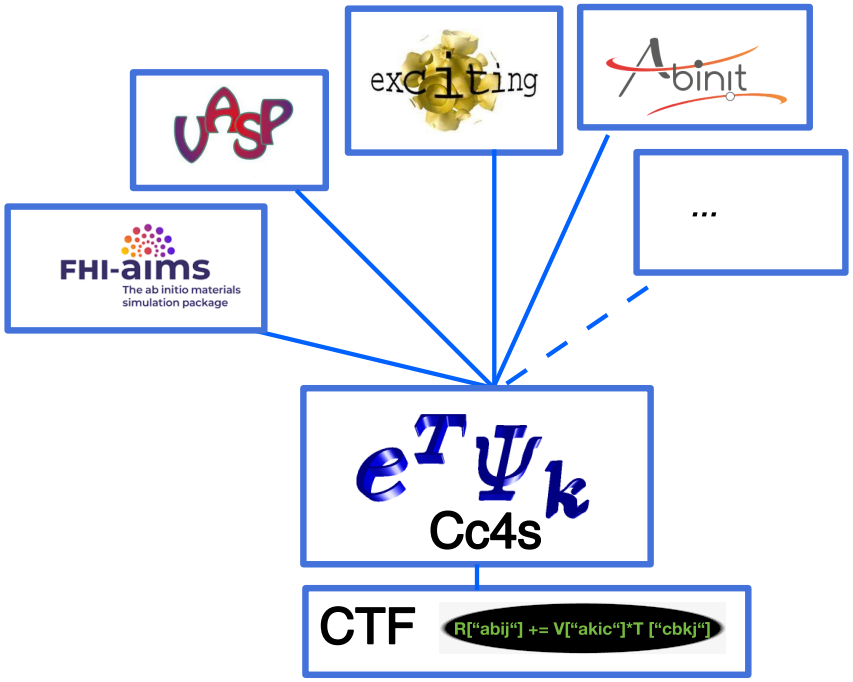
- Cyclops tensor framework
  `https://github.com/cyclops-community/ctf`
- Easily write tensor contractions massive parallelly

$$r_{ij}^{ab} = \sum_{ckdl} t_{ik}^{ac} V_{cd}^{lk} t_{lj}^{db}$$

turns into MPI C++ as

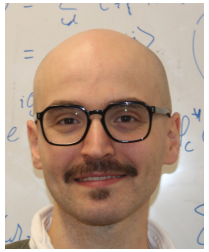
```
Tensor<double> r(4, ....), ...;

r["abij"] = t["acik"]
          * V["lkcd"]
          * t["dblj"];
```

./interfaces/


VASP
exciting
Abinit
...
FHI-aims
The ab initio materials simulation package
Cc4s
CTF
R["abij"] += V["akic"]*T ["cbkj"]

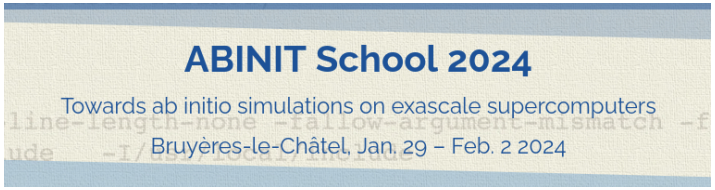
./interfaces/abinit

M. Giantomassi



A. Gallo

```
##########################################
# Dataset 2: CC4S interface
##########################################
optdriver2  6          # Activate GWR code
gwr_task2 "CC4S"       # Enter Abinit -> CC4S interface
getden2     1
nband2     100         # Number of (occ + empty) bands
ecuteps2   4
```



- Decemeber 2023 visit organized by X. Gonze in Louvain-la-Neuve
- Hands-on session in ABINIT school 2024

./interfaces/exciting

exciting

a full-potential all-electron package implementing linearized augmented planewave methods

DOWNLOAD FORUM

E. Moerman

A. Gulans B. Maurer

F. Hummel

K. Kacars

./interfaces/fhi-aims

FHI-aims
The ab initio materials simulation package

JOSS
The Journal of Open Source Software

Interface to high-performance periodic coupled-cluster theory calculations with atom-centered, localized basis functions

Evgeny Moerman[1], **Felix Hummel**[2], **Andreas Grüneis**[2], **Andreas Irmler**[2], **and Matthias Scheffler**[1]

1 The NOMAD Laboratory at the Fritz Haber Institute of the Max Plank Society, Berlin, Germany **2** Institute for Theoretical Physics, TU Wien, Vienna, Austria

DOI: 10.21105/joss.04040

Ingredients coming from ABINIT: Coulomb Vertex

- CC4S uses a *cholesky decomposition* of the Coulomb Integrals:

$$
\begin{aligned}
V_{rs}^{pq} &= \int \frac{\varphi_p^*(\mathbf{r}_1)\varphi_r(\mathbf{r}_1)\varphi_q^*(\mathbf{r}_2)\varphi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \\
&= \sum_G \bar{\Gamma}_r^{pG}\ \Gamma_s^{qG}
\end{aligned}
$$

where

$$
\Gamma_r^{pG} = \int_\Omega \mathrm{d}\mathbf{r} \sqrt{\frac{4\pi}{\mathbf{G}^2}}\, e^{i\mathbf{r}\cdot\mathbf{G}}\, \psi^*{}_p(\mathbf{r})\, \psi_r(\mathbf{r})
$$

- The Γ tensor is given by the files `CoulombVertex.yaml` and `CoulombVertex.elements`.
- Every tensor has a description in the `yaml` file, and data in the `.elements` file.

Ingredients coming from ABINIT: Eigenenergies ϵ_i

- At the moment KS eigenenergies supported, future work will enhance the Fock matrix routines in ABINIT in order to improve the hybrids and the CC4S interface.

ABINIT input file: SCF run

```
# Number of datasets: 2-step calculation
ndtset 2

ngkpt     1 1 1              # Number of grid points for k-point generation
nshiftk   1
shiftk    0 0 0              # Mesh centered on gamma
istwfk    *1                 # Required (for the time being)

# Definition of the plane-wave basis set
ecut 42.0                     # This should be used to reach convergence.

####################
# Dataset 1: SCF run
####################
nband1  48                   # Number of bands
tolvrs1  1.0d-8              # Tolerance to stop SCF cycles
```

ABINIT input file: CC4S

- It uses the GWR subsystem
- $N_{\mathrm{v}} + N_{\mathrm{o}} = 100$

```
#########################################
# Dataset 2: CC4S interface
#########################################
optdriver2  6          # Activate GWR code
gwr_task2 "CC4S"       # Enter Abinit -> CC4S interface
# gwr_task2 "CC4S_FULL" # Full DIAGO total number of states
getden2     1
nband2     100        # Number of (occ + empty) bands
ecuteps2   4
```

./CC4S/cc4s.in

- It uses also `yaml` as the input file format
- `CC4S` is quite verbose, like an *assembly* language
- Simple format

```
- name: NameOfTheAlgorithm
  in:
    # input parameters
    nelec: 42              #  Fake examples
    encut: 424242
  out:
    # output parameters
    density: Rho
```

./CC4S/Read

- The `Read` step reads in tensors
- Read Γ_r^{Gp}

```
- name: Read
  in:
    fileName: "CoulombVertex.yaml"
  out:
    destination: CoulombVertex
```

- Read eigenspektrum

```
- name: Read
  in:
    fileName: "EigenEnergies.yaml"
  out:
    destination: EigenEnergies
```

./CC4S/ holes and particles

- turn $\epsilon_p \rightarrow \{\epsilon_i, \epsilon_a\}$

```
- name: DefineHolesAndParticles
  in:
    eigenEnergies: EigenEnergies
  out:
    slicedEigenEnergies: EigenEnergies
```

- turn $\Gamma_{pGq} \rightarrow \{\Gamma_{pGq}, \Gamma_{aGb}, \Gamma_{aGi}, \Gamma_{iGj}\}$

```
- name: SliceOperator
  in:
    slicedEigenEnergies: EigenEnergies
    operator: CoulombVertex
  out:
    slicedOperator: CoulombVertex
```

./CC4S/coulomb-integrals

- Compute *lazy* Coulomb integrals $V_{rs}^{pq}(\Gamma)$

```
- name: VertexCoulombIntegrals
  in:
    slicedCoulombVertex: CoulombVertex
  out:
    coulombIntegrals: CoulombIntegrals
```

./CC4S/CCSD

```
- name: CoupledCluster
  in:
    method: Ccsd
    slicedEigenEnergies: EigenEnergies
    coulombIntegrals: CoulombIntegrals
    slicedCoulombVertex: CoulombVertex
    integralsSliceSize: 100
    maxIterations: 100
    energyConvergence: 1.0E-5
    amplitudesConvergence: 1.0E-5
    mixer:
      type: DiisMixer
      maxResidua: 12
  out:
    energy: CcsdEnergy
    amplitudes: Amplitudes
```

- Store the t_i^a, t_{ij}^{ab} in the tensor `Amplitudes`, you could then write out this tensor for inspection or postprocessing

./CC4S/CCSD(T)

- Use the CCSD amplitudes of previous step

```
- name: PerturbativeTriples
  in:
    slicedEigenEnergies: EigenEnergies
    amplitudes: Amplitudes
    coulombIntegrals: CoulombIntegrals
```

./CC4S/compile

```
git clone https://github.com/cc4s/cc4s
cd cc4s
```

or use `Makefile` configurations

- `./etc/config/gcc-oblas-ompi.mk`
- `./etc/config/icc-mkl-impi.mk`
- `./etc/config/icc-mkl-ompi.mk`
- You can edit this example files and do

```
# make external libraries
make CONFIG=gcc-oblas-ompi extern
make CONFIG=gcc-oblas-ompi
```

./CC4S/make-configuration/gcc-openBLAS-openmpi

```
include Cc4s.mk
include Extern.mk
include etc/make/blas.mk

# compiler and linker
CXX = mpicxx

# general and language options (for preprocessing, compiling and linking)
CXXFLAGS += \
-fopenmp -std=c++11 \
-Wall -pedantic --all-warnings -fmax-errors=3 \
-Wno-vla \
-Wno-int-in-bool-context

# optimization options (only for compiling and linking)
CXXFLAGS += -Ofast -march=native -fno-lto

CTF_CONFIG_FLAGS = CXX=${CXX} \
                   AR=gcc-ar \
                   CXXFLAGS="-Ofast -std=c++11 -march=native -fno-lto" \
                   LIBS="-L$(BLAS_PATH)" \
                   --no-dynamic \
                   --without-scalapack

LDFLAGS += \
-Wl,-Bstatic \
${STATIC_LIBS} \
-lgfortran -lquadmath \
-Wl,-Bdynamic \
${DYNAMIC_LIBS} \
```

LIVE DEMO
I ALSO LIKE TO LIVE DANGEROUSLY
memegenerator.net