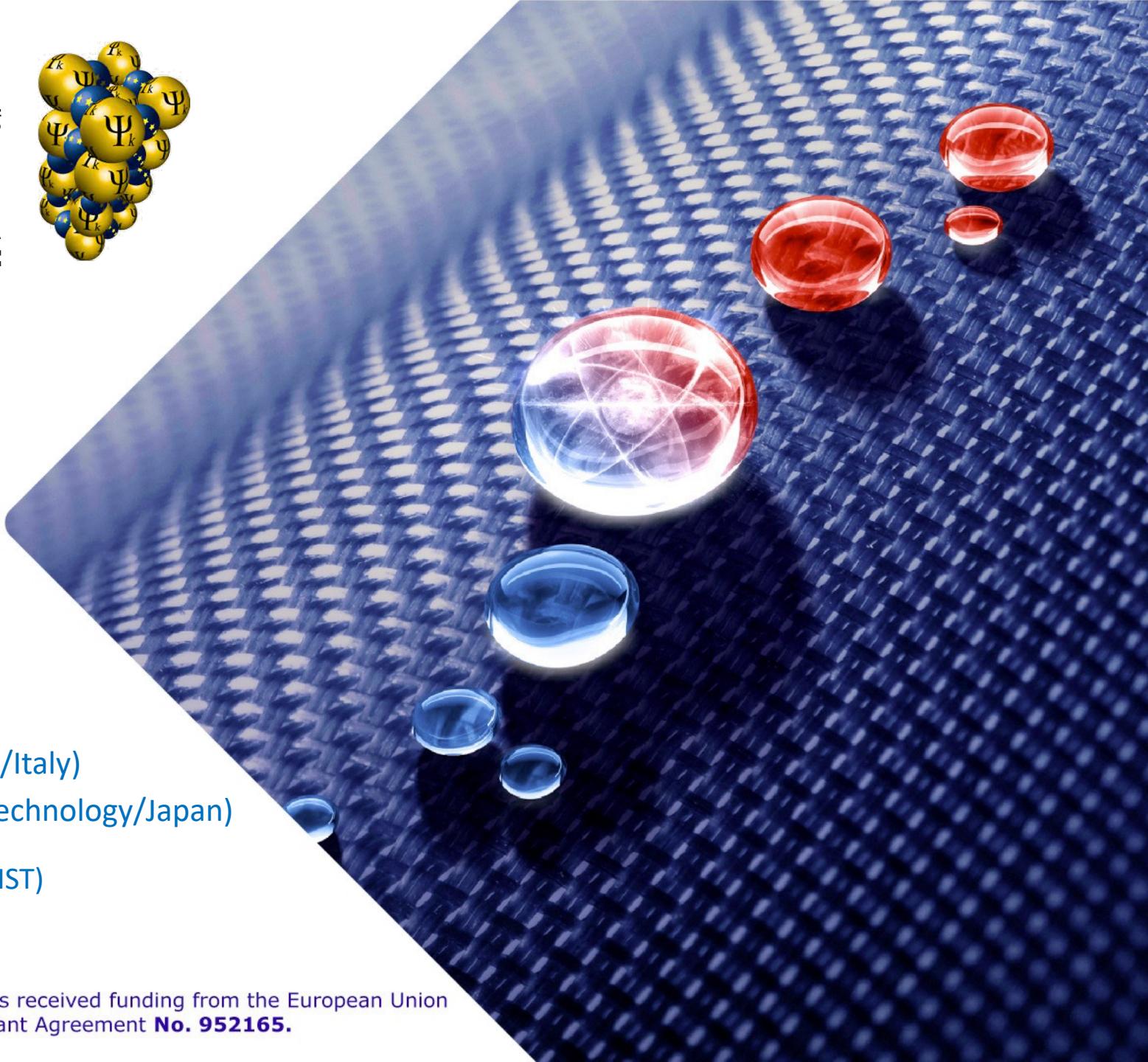


TurboRVB and Turbo-Genius: Overview and Workflow

Kosuke Nakano

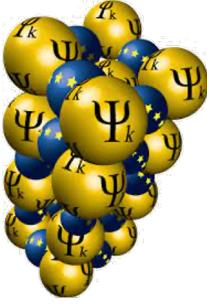
- SISSA (International School for Advanced Studies/Italy)
 - JAIST (Japan Advanced Institute of Science and Technology/Japan)
- (Prof. Sorella group/SISSA) (Prof. Maezono group/JAIST)



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.



Targeting Real chemical accuracy at the EXascale

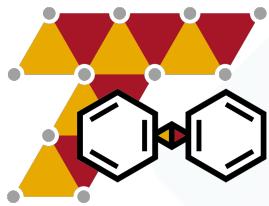


Day1:

- General Introduction
- Wavefunction representation
- Built-in DFT code



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.



TurboRVB

Quantum Monte Carlo Package SISSA

QMC engines (DFT, VMC-optimization, VMC, LRDMC)

K. Nakano, C. Attaccalite, M. Barborini, L. Capriotti, M. Casula, E. Coccia, M. Dagrada, Y. Luo, G. Mazzola, A. Zen, and S. Sorella, *J. Chem. Phys.* 152, 204121 (2020)



Turbo-Genius

User friendly python wrappers.

K. Nakano et al., *in preparation* (2021)

TurboRVB: A many-body toolkit for *ab initio* electronic simulations by quantum Monte Carlo

Cite as: J. Chem. Phys. 152, 204121 (2020); <https://doi.org/10.1063/5.0005037>

Submitted: 19 February 2020 . Accepted: 20 March 2020 . Published Online: 29 May 2020

Kousuke Nakano , Claudio Attaccalite , Matteo Barborini , Luca Capriotti , Michele Casula , Emanuele Coccia , Mario Dagrada, Claudio Genovese , Ye Luo , Guglielmo Mazzola , Andrea Zen , and Sandro Sorella

COLLECTIONS

Paper published as part of the special topic on [Collection](#)

Note: This article is part of the JCP Special Topic on Electronic Structure Software.



[Home](#)

[View page source](#)

[Home](#)

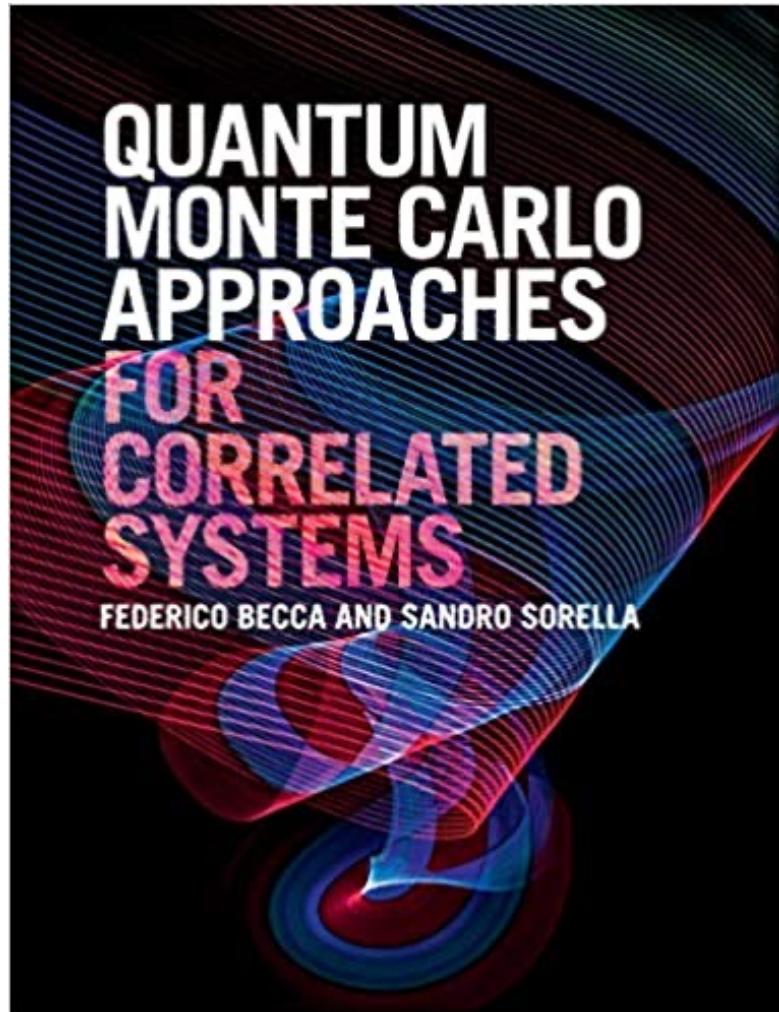


TurboRVB
Quantum Monte Carlo Package 

News

- K. Nakano et al. have published a paper in [Phys. Rev. B 103, L121110 \(2021\)](#).
This paper has been selected as an [Editors' Suggestion](#).
- Our TurboRVB workshop will be held on 12-16 July 2021 at SISSA (Italy)!
Please have a look at [Summer School on Quantum Monte Carlo methods 2021](#).
[Online registration](#) can be done from the TREX website.
- C. Genovese and S. Sorella have published a paper in [J. Chem. Phys. 153, 164301 \(2020\)](#).
- C. Genovese et al. have published a paper in [J. Chem. Theory Comput. 16 6114-6131 \(2020\)](#).
- We have published a TurboRVB review paper in [J. Chem. Phys. 152, 204121 \(2020\)](#).

K. Nakano, C. Attaccalite, M. Barborini, L. Capriotti, M. Casula, E. Coccia, M. Dagrada, Y. Luo, G. Mazzola, A. Zen, and S. Sorella, *J. Chem. Phys.* **152**, 204121 (2020)



Federico Becca, and Sandro Sorella

Publisher : Cambridge University Press; 1st edition
(November 1, 2017)

Language : English

Hardcover : 286 pages

ISBN-10 : 9781107129931

Please visit our website :-) All the papers and Ph.D theses using TurboRVB are listed here.

[TurboRVB website](#)
Updated on 03/06/2021

Search docs

CONTENTS:

- News
- Developers
- Source code
- Workshops
- Positions

↳ Publications

2021
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009

↳ The nature of the chemical bond in the dicarbon molecule

» Publications [View page source](#)

Publications



Quantum Monte Carlo Package / SISSA

2021

- Atomic forces by quantum Monte Carlo: Application to phonon dispersion calculations, K. Nakano, T. Morresi, M. Casula, R. Maezono, and S. Sorella, *Phys. Rev. B* **103**, L121110 (2021).
Selected as an Editors' Suggestion

2020

- Ground-state properties of the hydrogen chain: insulator-to-metal transition, dimerization, and magnetic phases, M. Motta, C. Genovese, F. Ma, Z. Cui, R. Sawaya, G.K. Chan, N. Chepiga, P. Helms, C. Jiménez-Hoyos, A.J. Millis, U. Ray, E. Ronca, H. Shi, S. Sorella, E.M. Stoudenmire, S.R. White, and S. Zhang (Simons Collaboration on the Many-Electron Problem), *Phys. Rev. X* **10**, 031058 (2020).

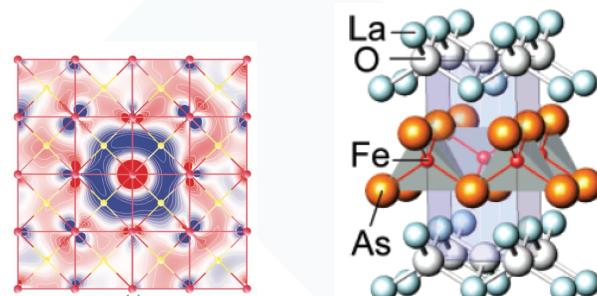
↳ The nature of the chemical bond in the dicarbon molecule

» PhD thesis [View page source](#)

PhD thesis

- Dr. Claudio Genovese, 2020:
Geminal Power in QMC, [pdf](#)
- Dr. Félix Mouhat, 2018:
Fully quantum dynamics of protonated water clusters, [pdf](#)
- Dr. Nicolas Dupuy, 2016:
Corrélations électroniques des acènes vers la limite de longue taille : Étude par Monte Carlo quantique (Electronic correlations in the acenes to ards the long-size limit: a Monte Carlo study), [pdf](#)
- Dr. Henri Hay, 2016:
Étude de la structure et des propriétés des polymorphes de SiO₂ et B₂O₃ par méthodes ab initio (Structural properties of SiO₂ and B₂O₃ polymorphs by ab initio methods), [pdf](#)
- Dr. Mario Dagdaga, 2016:
Improved quantum Monte Carlo simulations: from open to extended systems, [pdf](#)
- Dr. Nicolas Dévaux, 2015:
Étude par Monte Carlo quantique de la transition α - γ du Cérium (Quantum Monte Carlo study of the α - γ transition in Cerium), [pdf](#)
- Dr. Guglielmo Mazzola, 2014:
Metallization and dissociation in high pressure liquid hydrogen by an efficient molecular dynamics with quantum Monte Carlo, [pdf](#)
- Dr. Ye Luo, 2014:
Ab initio molecular dynamics of water by quantum Monte Carlo, [pdf](#)

-Strongly correlated materials

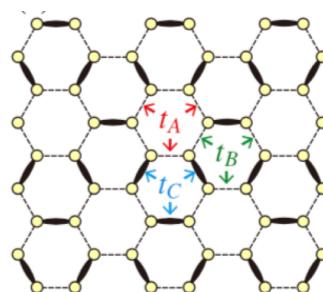


Y. Kamihara, et al.: JACS 130 (2008) 3296.

Unconventional superconductors

M. Casulla et al., Phys. Rev. B 88, 155125 (2013)

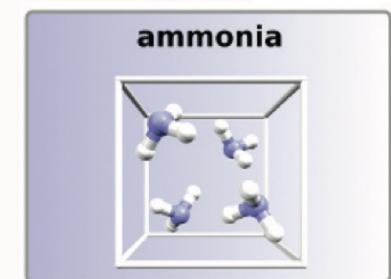
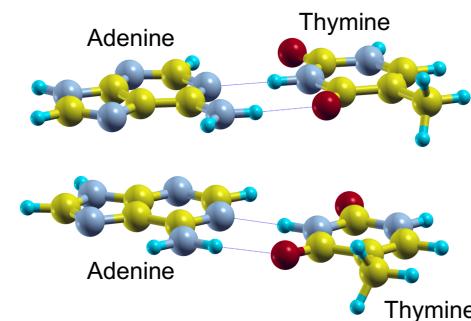
-Van der Waals-related



Two-dimensional compounds
(graphene, stacked DNA)

S. Sorella et al., Phys. Rev. Lett. 121, 066402 (2018)

-Materials in an extreme condition



Molecular crystals

A. Zen et al., PNAS, 115, 1724 (2018)

Others:

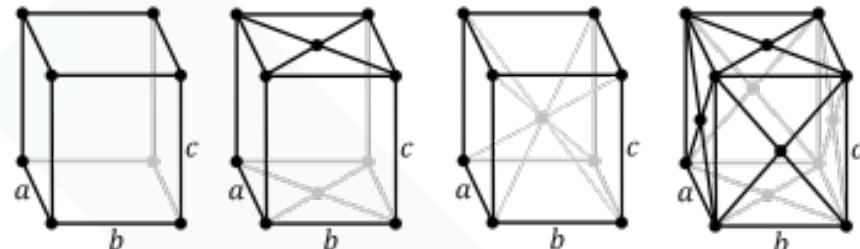
H-He mixture's phase diagram in Jupiter

G. Mazzola, Phys. Rev. Lett. 120, 025701 (2018)

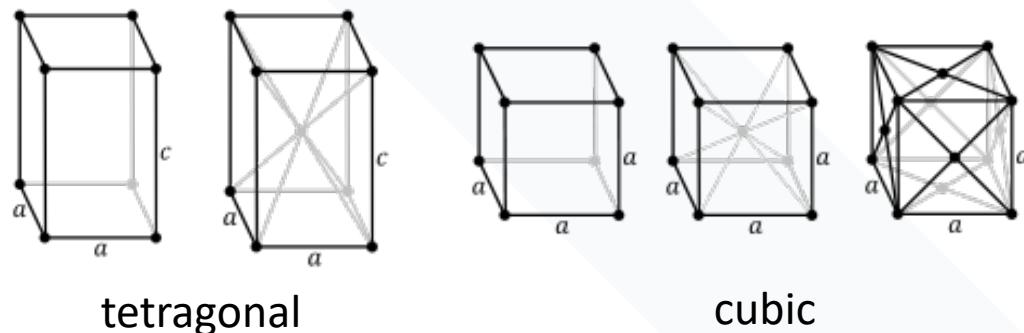
- (i) It employs the Resonating Valence Bond (RVB) WF that includes static and dynamical correlation effects beyond the commonly used Slater determinant, while keeping the computational cost at the single-determinant level.
- (ii) The code implements a VMC algorithm based on localized orbitals (e.g., Gaussians) and state-of-the-art optimization methods, such as the stochastic reconfiguration. Therefore, at the VMC level, one can optimize not only the amplitude of the WF (i.e., the Jastrow factor), but also the nodal surfaces (e.g., the Slater determinant). This leads to a better variational energy in general, and also improves the corresponding FN-DMC energy.
- (iii) The energy derivatives (e.g., atomic forces) are calculated very efficiently thanks to an implementation based on the Adjoint Algorithmic Differentiation (AAD). As a consequence, one can perform structural optimizations and Langevin molecular dynamics.
- (iv) The code implements the newly developed Lattice Regularized Diffusion Monte Carlo (LRDMC), a stable DMC algorithm that, very recently, has shown to have a better scaling with the atomic number Z , compared with standard DMC.

K. Nakano, C. Attaccalite, M. Barborini, L. Capriotti, M. Casula, E. Coccia, M. Dagrada, Y. Luo, G. Mazzola, A. Zen, and S. Sorella, *J. Chem. Phys.* 152, 204121 (2020)

Orthogonal bravais (-2020)



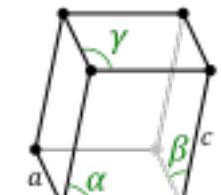
Orthorhombic



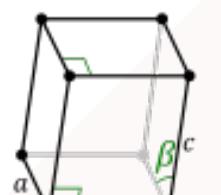
tetragonal

cubic

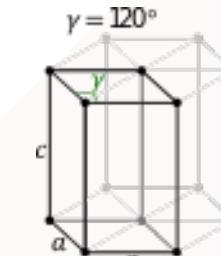
Also non-orthogonal bravais (2021-)



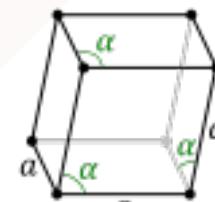
triclinic



monoclinic



hexagonal



Rhombohedral

Reference: Wikipedia images

This will be published in the near future.



Very flexible ansatzs have been implemented!!

$$\Psi = \left| \phi_i(\vec{r}_j) \right|$$

$\phi_i(\vec{r}_j)$ = One-particle

Slater determinant
Determinant

$$\Psi = \left| f(\vec{r}_i, \vec{r}_j) \right|$$

$f(\vec{r}_i, \vec{r}_j) = \sum_{a,b} c_{\{a,i\}, \{b,j\}} \phi_a(\vec{r}_i) \phi_b(\vec{r}_j)$

Parings of up-dn electrons.

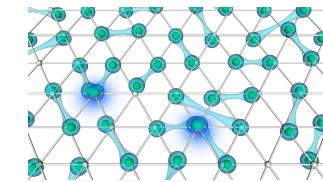
Called “Geminal”

$$\Psi = \text{Pf} \left[f(\vec{r}_i, \vec{r}_j) \right]$$

up-up/dn-dn pairing also.

Pfaffian

More complex



Resonating Valence Bond (RVB)

c.f. P.W. Anderson

Very flexible ansatzs (the larger number of variational parameters), satisfying the anti-symmetry!

C.Genovese et al., J. Chem. Theory Comput. 16, 6114 (2020)

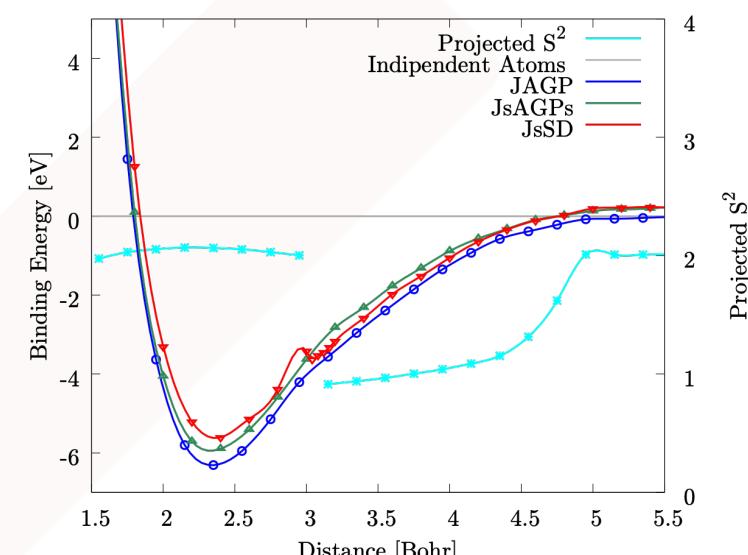
Double-bond?? Quadruple-bond??, spin-singlet.

LRDMC results

S. Shaik, et al. Nat. Chem. 4 195–200 (2012)

More complex.
↓

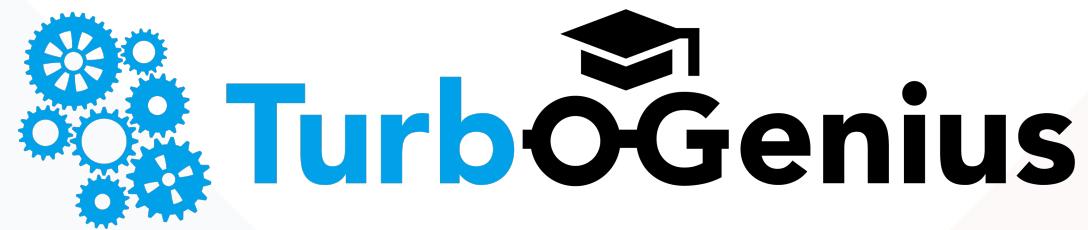
Wavefunction	C atom (Ha)	C ₂ molecule (Ha)	Binding (eV)
Jastrow Slater	-37.82966(4)	-75.8672(1)	5.656(3)
Jastrow Geminal (Singlet)	-37.8364(1)	-75.8938(2)	6.01(1)
Jastrow Geminal (Singlet + broken sym.)	-37.8364(1)	-75.8935(2)	6.00(1)
Jastrow Geminal (All-pairing, Pfaffian)	-37.8363(1)	-75.9045(2)	6.31(1)
Estimated exact	-37.8450	-75.9045(2)	6.44(2) (Exp.)



C.Genovese et al., J. Chem. Theory Comput. 16, 6114 (2020)

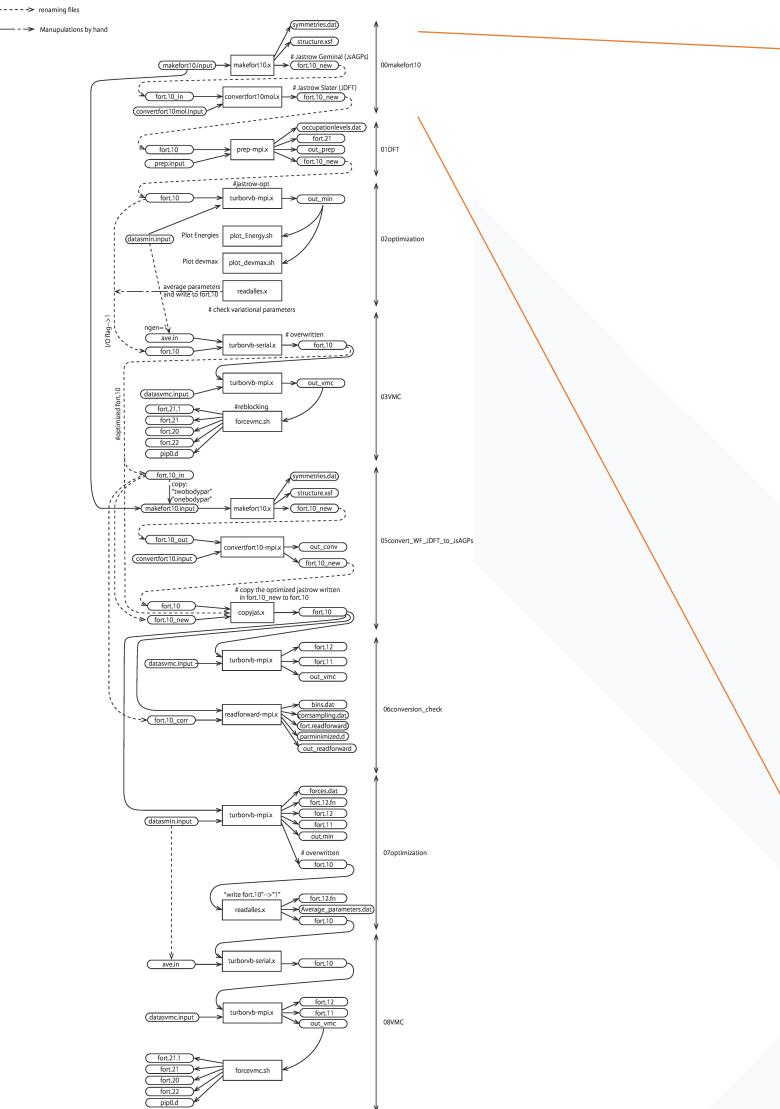
CCSD(T) with the V5Z basis = 6.24 eV

The LRDMC gives a more accurate result than CCSD(T) for the challenging molecule!



K. Nakano et al., in preparation (2021)

Any contributor is welcome!!!



```
% vi makefort10.input. (edit input files manually)
```

...

ATOM_1. (Define the cc_pVTZ basis)

```
&shells
nshelldet=7
nshelljas=6
/
1 1 16
1 5.09500000000000
1 1 16
1 1.159000000000
...
```

Parameters atomic Jastrow wf

```
1 1 16
1 0.83560000000
```

...

```
% makefort10.x < makefort10.input > out_make
```

```
% mv fort.10_new fort.10_in
```

```
% convertfort10mol.x < convertfort10mol.input > out_mol
```

```
% mv fort.10_new fort.10
```



Turbo-Genius makes it much easier!!!

```
% turbo-genius.py -j makefort10 -g -str H2_dimer.xyz -basis cc_pVTZ
```

```
% turbo-genius.py -r -post
```

Where can we download TurboRVB and Turbo-Genius from?

The screenshot shows the TurboRVB project page on the SISSA GitLab server. The page includes a sidebar with various icons, a header with navigation links like 'Projects', 'Groups', and 'More', and a search bar. The main content area displays project statistics (4,600 Commits, 13 Branches, 1 Tag, 350.4 MB Files, 350.5 MB Storage), a commit history with a recent revision by Sandro Sorella, and a file browser showing directories AD, DFT, and bin.

From SISSA-gitlab server.

<https://git-scm.sissa.it>

A request: to sorella@sissa.it

For time being, turborvb and turbo-genius are inhouse codes, so DO NOT distribute to the public.

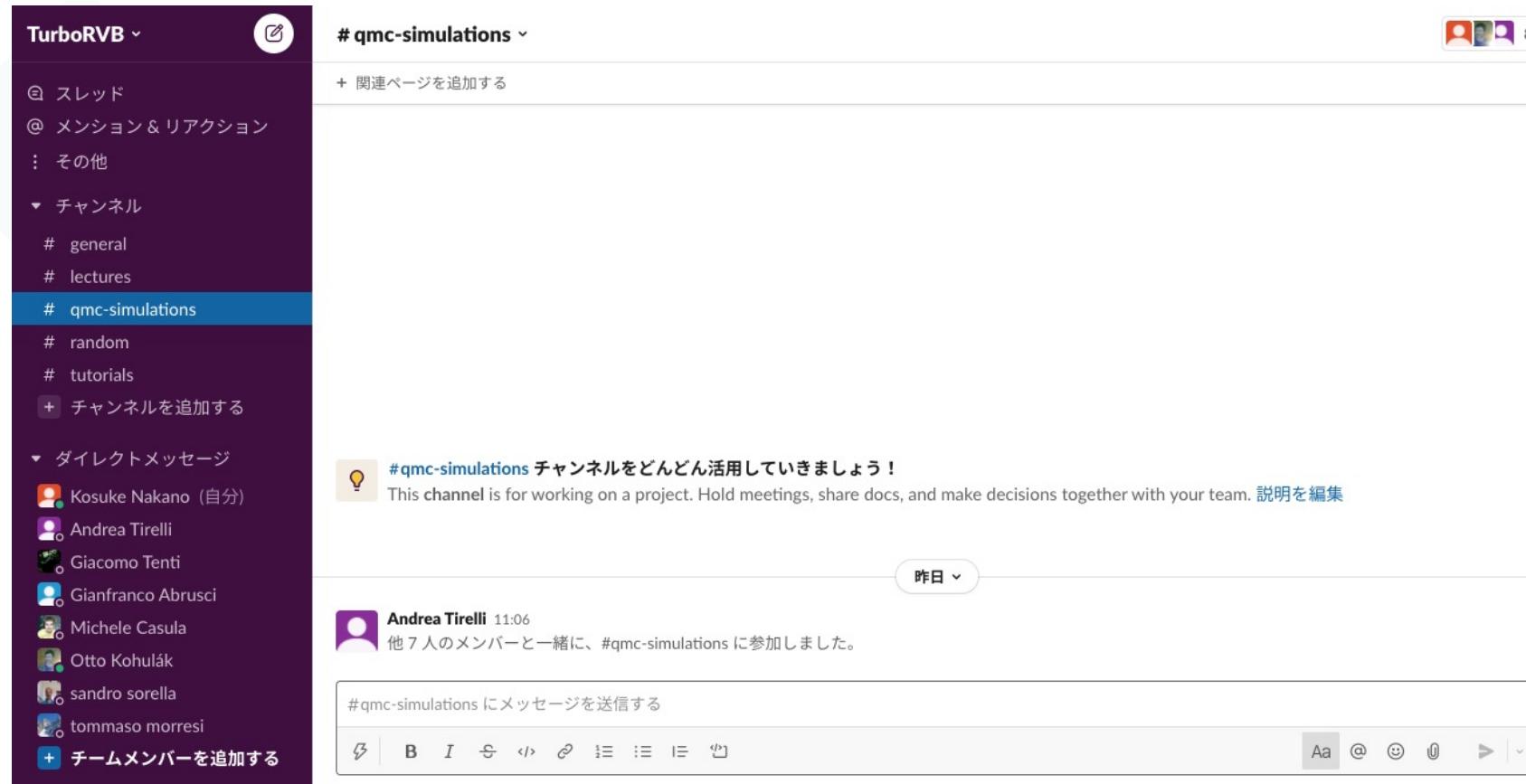
Within a couple of years, all the codes will be public under an appropriate license (maybe BSD) :-)

TurboRVB gitlab repository

```
└── AD  
└── DFT  
└── bin  
└── config  
└── config_cmake  
└── pseudo  
└── src  
└── test  
└── tool  
└── turbo_genius_dist
```

TurboRVB AAD source files
TurboRVB DFT (prep) source files
Installation directory of executables
Machine-specific include files (for make)
Machine-specific include files (for CMake)
Pseudopotential libraries
TurboRVB main source files
This directory is intended to contain jobs to test a build
TurboRVB tools source files
Turbo-Genius source files

How can I ask something to the developers? via slack !!



If you find some bugs or you have some (open) question, please post to here !!



TurboRVB/Turbo-Genius manual and tutorials

The screenshot shows a sidebar on the left with a dark background and white text. It includes a header "TurboRVB userguides" with a house icon, a timestamp "Updated on 05/07/2021", and a search bar labeled "Search docs". Below these are three categories: "User Manuals", "Tutorials", and "Developer Manuals". The main content area on the right has a blue header "TurboRVB user manuals" with a house icon and a "View page source" link. The main title "TurboRVB user manuals" is bolded. Below it is a navigation tree:

- User Manuals
 - TurboRVB Manual
 - Installation
 - TurboRVB in a nutshell
 - Wavefunction
 - makefort10
 - DFT driver
 - QMC driver
 - Tools and input examples
 - Workflow
 - Turbo-Genius Manual
 - Installation
 - Turbo-Genius in a nutshell
- Tutorials
 - TurboRVB tutorials
 - 00 TurboRVB install
 - 01_01Hydrogen_dimer
 - 01_02Hydrogen_dimer
 - 02_01Lithium_dimer
 - 98 Wavefunction optimization

Any contributor is welcome!!!

From SISSA-gitlab server.

<https://git-scm.sissa.it>

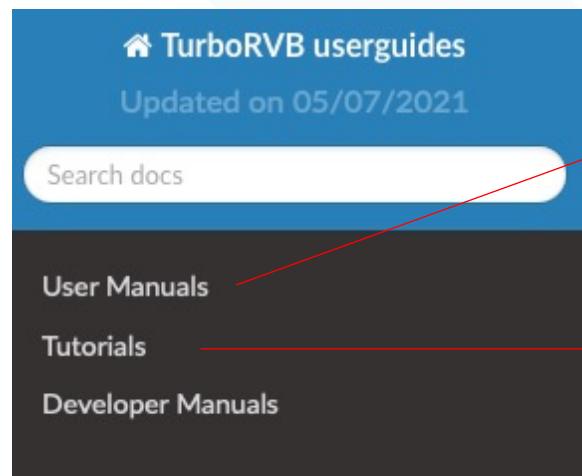
They are composed by sphinx. All the tutorial in this school is also included here.

If you want to see the userguide, please let us know. We will give you the permission.

git clone git@git-scm.sissa.it:sorella/turborvb_userguides.git

Any contributors is welcome!!!

Open /turborvb_userguides -> build -> html -> index.html



- User Manuals.
- TurboRVB tutorials.
- Turbo-Genius tutorials (for the hands-on session).

on your local comp. (for the TREX summer school).

```
rsync -avz xxxx@login.m100.cineca.it:/m100_work/tra21_trex/TurboRVB_userguides/build/html .  
open ./html/index.html
```

We strongly recommend Intel, IBM, and Fujitsu Fortran compilers. (Not gfortran).

1. Legacy make:

You do not have to do this for the hands-on session!!!

Copy a config file: config/make_XXX.inc make.inc

Copy a make.txt file: src/make.txt_standard src/make.txt

Compile TurboRVB: ./makeall (serial) or ./makeall-mpi

If you want to clean it: make cleanall

Fugaku, Hokusai (RIKEN)

Marconi, Marconi100 (CINECA)

SISSA-cluster (SISSA)

Kagayaki (JAIST)

2. Modern CMake: (Sorry, under refactoring)

```
mkdir build
```

```
cd build
```

```
cmake -DCONFIG=gnu-mac-serial/intel-galileo-serial/etc ..
```

```
make      # compile all serial/parallel programs.
```

```
make install # copy generated binaries to ./bin directory.
```

Whenever compiling TurboRVB on a new machine, please check if the compilation is correct.

Go to test/ directory. Run ./jobcheck-serial.sh

```
% ./jobcheck-serial.sh  
...  
Warning: This debugging tool rounds off < 10**-6.  
Warning: You can set ROUND_OFF in this script..
```

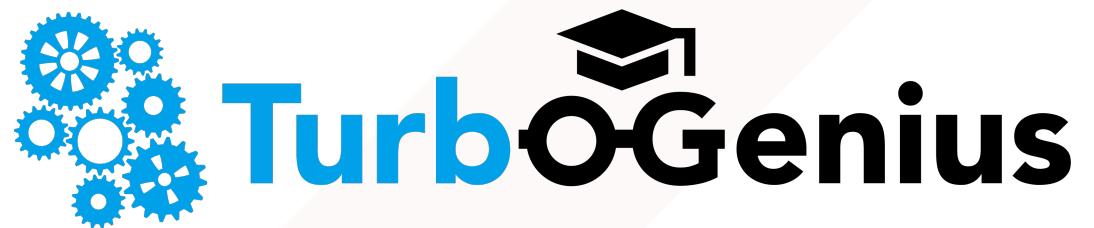
You do not have to do this for the hands-on session!!!

```
Debugging starts:  
#####
DFT-OPEN TEST  
dir=dft_open  
-DFT energy = -1.161081  
-DFT energy (ref) = -1.161081  
-The diff = 0  
-If the diff is finite, there is something wrong.  
#####
...  
All test has finished. Enjoy Turbo :-)
```

This checks various type of calculations. You see outputs if they are inconsistent with the references.

Turbo-Genius relies on various python packages (pymatgen, ASE, etc...), so we recommend that you install it via pip. Also, we recommend that you used a virtual environment to keep it your environment clean.

1. Go to turbo_genius_dist/ directory.
2. Run pip install -e .

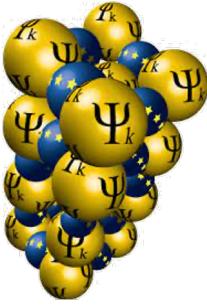


Anaconda virtual environment!! Go to [<https://www.anaconda.com>]

You do not have to do this for the hands-on session!!!



Targeting Real chemical accuracy at the EXascale



Day1:

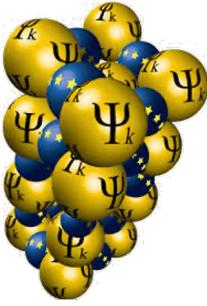
- General Introduction
- Wavefunction representation
- Built-in DFT code



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.



Targeting Real chemical accuracy at the EXascale

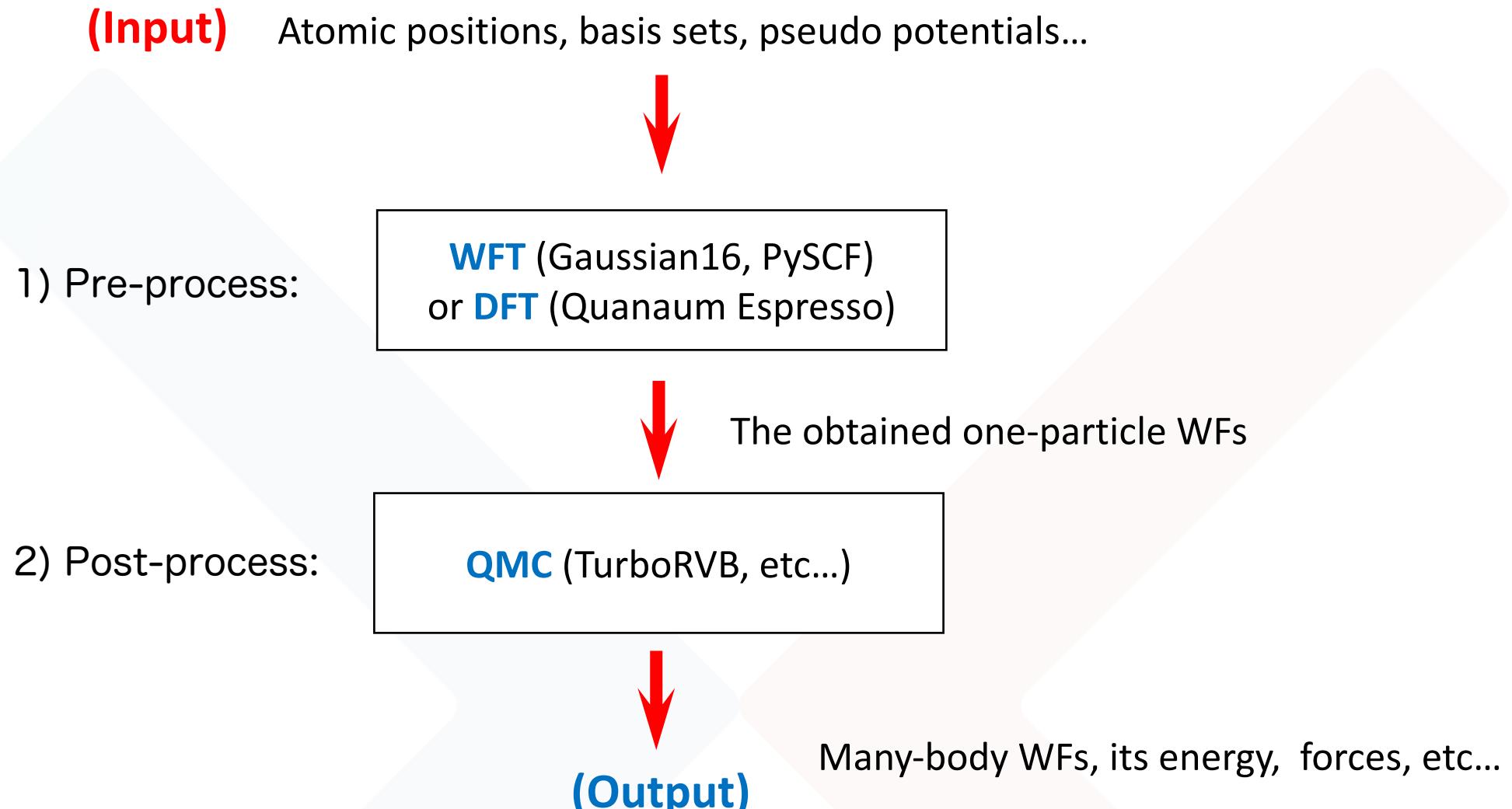


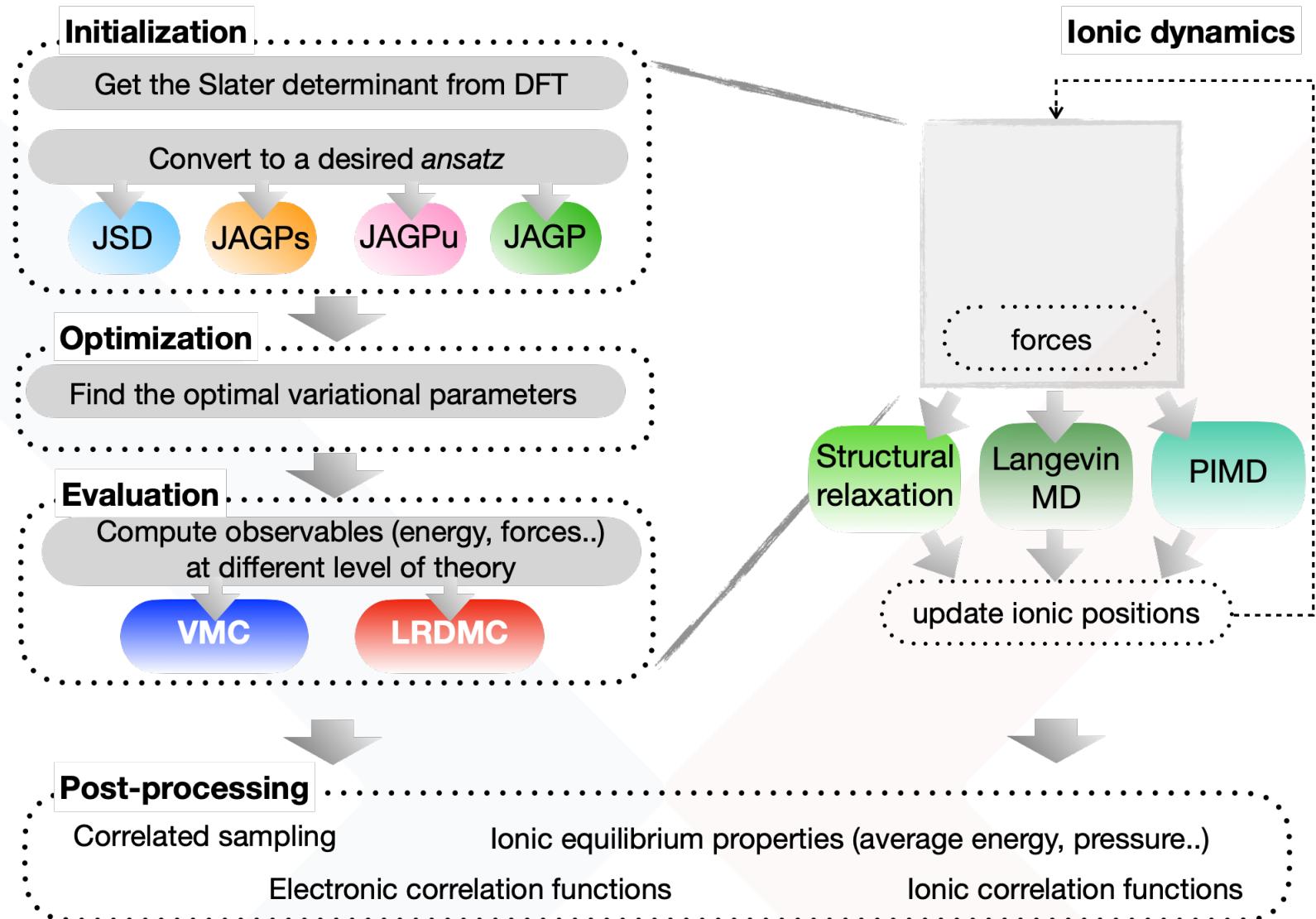
Day1:

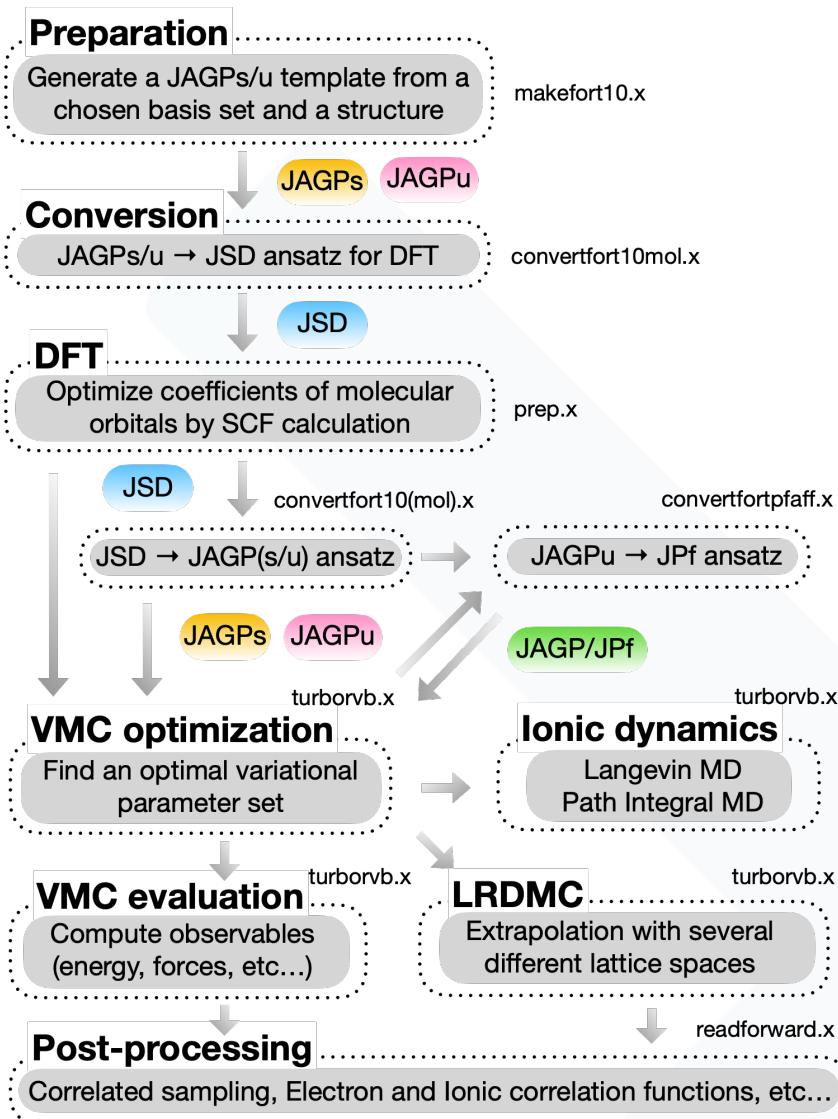
- General Introduction
- Wavefunction representation
- Built-in DFT code



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.







= Workflow =

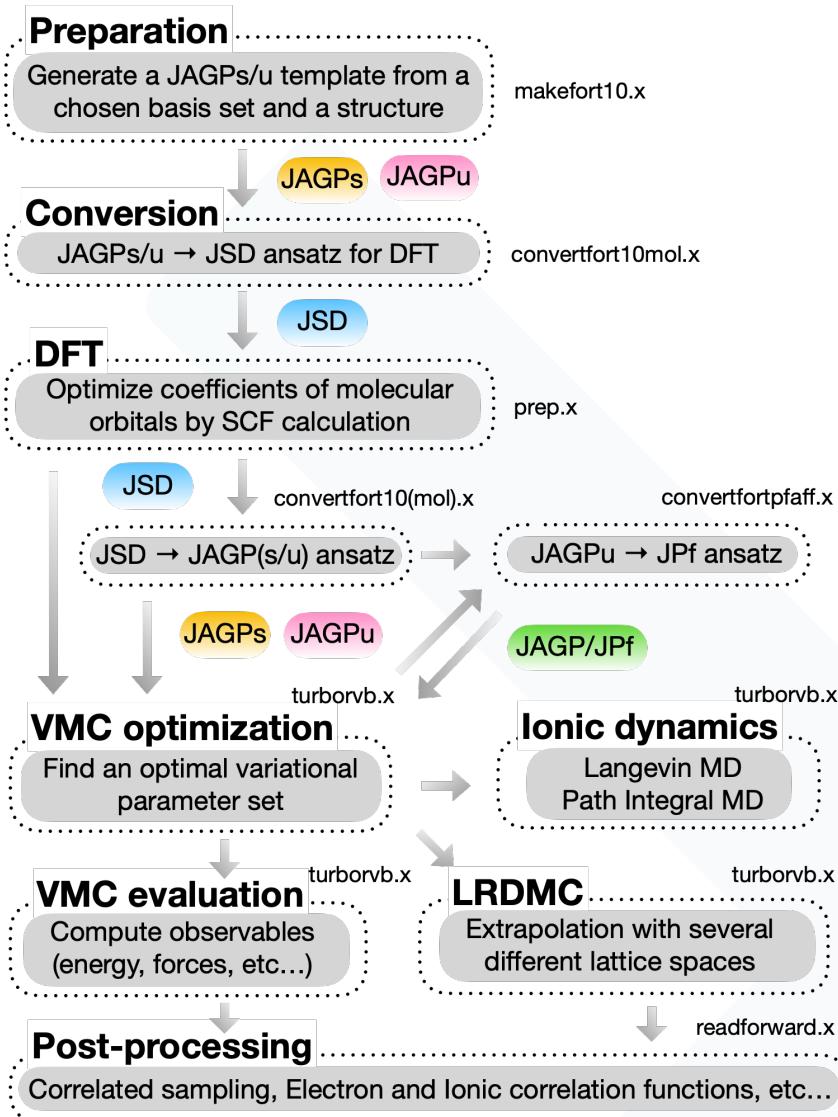
1. Prepare a structure and basis set **makefort10.x**

2. DFT Construct a reasonable initial WF! **prep.x**

3. VMC-opt Optimize the wavefunction **turborvb.x**

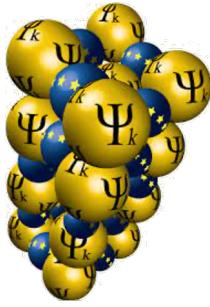
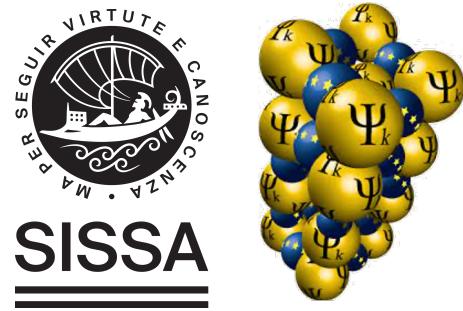
4. VMC Do a VMC run. **turborvb.x**

5. LRDMC LRDMC with the optimized WF. **turborvb.x**



= Workflow =

1. Prepare a structure and basis set **makefort10.x** **Day1**
2. DFT Construct a reasonable initial WF! **prep.x** **Day1**
3. VMC-opt Optimize the wavefunction **turborvb.x** **Day2**
4. VMC Do a VMC run. **turborvb.x** **Day3**
5. LRDMC LRDMC with the optimized WF. **turborvb.x** **Day4**

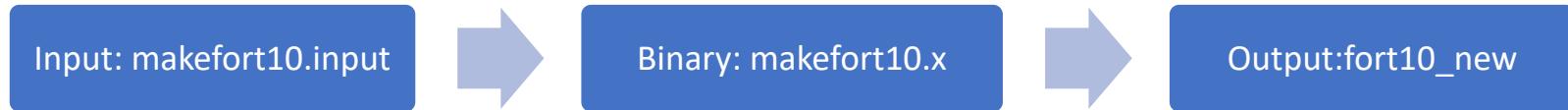


Day1:

- General Introduction
- **Wavefunction representation**
- Built-in DFT code



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.



`makefort10.x` is a tool for generating JAGP WF(fort.10) from `makefort10.input`.

```
# Ion coordinates
N1 Z1           x1     y1     z1
N2 Z2           x2     y2     z2
  ..
Nn Zn           xn     yn     zn
```

Structural information.

```
#      Parameters atomic wf
        1          4      300
        1    2.0   1.0  3.231  7.54
```

Basis-set information.

```

posunits='crystal'
natoms=2
ntyp=1
complexfort10=.false.
pbcfort10=.true.
!yes_pffaff=.true.
celldm(1)=4.648726266579395
celldm(2)=1.0
celldm(3)=4.065040650406504
celldm(4)=1.5707963267948966
celldm(5)=1.5707963267948966
celldm(6)=2.0943951023931953
yes_tilted=.true.
nxyz(1)=3
nxyz(2)=3
nxyz(3)=1
phase(1)=0.0
phase(2)=0.0
phase(3)=0.0
phasedo(1)=0.0
phasedo(2)=0.0
phasedo(3)=0.0

```

makefort10.input file

Wavefunction file (fort.10)

All the information (except for pseudo potential info.) is included in a single file, named “fort.10”

```
# fort.10 of the C2-dimer (the Pfaffian ansatz with the Filippi pseudo potential.)
# Nelup #Nel # Ion
    4      -8      2
# Shell Det. # Shell Jas.
    50      43
# Jas 2body # Det # 3 body atomic par.
   -22     1482     42
# Det mat. =/0 # Jas mat. =/0
    120     8370
# Eq. Det atomic par. # Eq. 3 body atomic. par.
    741      21
# unconstrained iesfree,iessw,ieskinr,I/O flag
    8370     120      6      0
# Ion coordinates
4.00000000000000      6.00000000000000      0.00000000000000E+000
0.00000000000000E+000  -1.1499954166875
4.00000000000000      6.00000000000000      0.00000000000000E+000
0.00000000000000E+000  1.1499954166875
# Constraints for forces: ion - coordinate
    1      1      1
    1      1      2
    1      1      3
    1      2      1
    1      2      2
    1      2      3
# Parameters Jastrow two body
-1  0.342214663461764
...
```

“fort.10” can be generated by
“makefort10.x” (see later).

Header:

```
# Nelup #Nel # Ion
      2        4       1
# Shell Det. # Shell Jas.
      3        3
# Jas 2body # Det # 3 body atomic par.
     -8       16       8
# Det mat. =/0 # Jas mat. =/0
      6        6
# Eq. Det atomic par. # Eq. 3 body atomic. par.
     13       8
# unconstrained iesfree, iessw, ieskinr, I/O flag
      4        4       0       0
```

The total number of Jastrow variational param.

The total number of determinant variational param.

The number of atomic forces.

Nelup: The number of spin up electrons in the system.

Nel: The total number of electrons in the system.

Ion: The number of nuclei in the system.

Jas 2body: Onebody and Twobody Jastrow types

Coordinates:

```
# Ion coordinates
N1 Z1      x1      y1      z1
N2 Z2      x2      y2      z2
  ..
Nn Zn      xn      yn      zn
```

- N: Atomic number
- Z: The number of valence electrons
- xn, yn, zn : atomic positions (Bohr)

Pseudo potential case $N \neq Z$

If you want to use a H-pseudo potential,
please put $N=1.0$, $Z=1.00001$ (dummy).

TurboRVB employs a many-body WF ansatz that can be written as the product of two terms:

$$\Psi = \Phi_{\text{AS}} \times \exp J \quad J = J_1 + J_2 + J_{3/4}$$

- **AS (Anti-symmetric part)** e.g.,

$$\Phi_{\text{AGPs}} = \det(G_S) \quad \text{where} \quad g_s(\mathbf{i}, \mathbf{j}) = f_S(\mathbf{r}_i, \mathbf{r}_j) \frac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}},$$

- **J (Jastrow part)**

$$J_1^h(\mathbf{r}_1, \dots, \mathbf{r}_N) = \sum_{i=1}^N \sum_{a=1}^{N_{\text{at}}} \left(-(2Z_a)^{3/4} u_a \left((2Z_a)^{1/4} |\mathbf{r}_i - \mathbf{R}_a| \right) \right) \quad J_1^{inh}(\mathbf{r}_1\sigma_1, \dots, \mathbf{r}_N\sigma_N) = \sum_{i=1}^N \sum_{a=1}^{N_{\text{at}}} \left(\sum_l M_{a,l}^{\sigma_i} \chi_{a,l}(\mathbf{r}_i) \right)$$

electron-ion cusp

$$J_2(\mathbf{r}_1\sigma_1, \dots, \mathbf{r}_N\sigma_N) = \sum_{i < j} v_{\sigma_i, \sigma_j}(|\mathbf{r}_i - \mathbf{r}_j|) \quad J_{3/4}(\mathbf{r}_1\sigma_1, \dots, \mathbf{r}_N\sigma_N) = \sum_{i < j} \left(\sum_{a,l} \sum_{b,m} M_{a,l,b,m}^{\sigma_i, \sigma_j} \chi_{a,l}(\mathbf{r}_i) \chi_{b,m}(\mathbf{r}_j) \right),$$

electron-electron cusp

Why one-particle orbitals (basis sets) are needed for the many-body WF theory??

- AS (Anti-symmetric part)

Parametrization!

$$g_s(\mathbf{i}, \mathbf{j}) = f_S(\mathbf{r}_i, \mathbf{r}_j) \frac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}}, \quad f_S(\mathbf{r}_i, \mathbf{r}_j) = \sum_{l,m,a,b} A_{\{a,l\},\{b,m\}} \psi_{a,l}(\mathbf{r}_i) \psi_{b,m}(\mathbf{r}_j).$$

Paring functions

- Inhomogeneous and three-body Jastrow parts

$$J_1^{inh}(\mathbf{r}_1\sigma_1, \dots, \mathbf{r}_N\sigma_N) = \sum_{i=1}^N \sum_{a=1}^{N_{at}} \left(\sum_l M_{a,l}^{\sigma_i} \chi_{a,l}(\mathbf{r}_i) \right)$$
$$J_{3/4}(\mathbf{r}_1\sigma_1, \dots, \mathbf{r}_N\sigma_N) = \sum_{i < j} \left(\sum_{a,l} \sum_{b,m} M_{a,l,b,m}^{\sigma_i, \sigma_j} \chi_{a,l}(\mathbf{r}_i) \chi_{b,m}(\mathbf{r}_j) \right),$$

Basis set for the determinant part:

```
#          Parameters atomic wf
Shell_Multiplicity      Number of par.
Ion index                [par (1, NUMBER OF PAR.)]

#          Parameters atomic wf
1           1
1  0.5000000000000000
3           1
1  1.000000000000000
1           1
2  0.3000000000000000
1           1
3  0.3000000000000000
1           1
4  0.3000000000000000
1           1
5  0.3000000000000000
```

```
#          Parameters atomic wf
1           4      300
1  2.0   1.0  3.231  7.54
```

$$\phi(r) = 3.231 \cdot \exp(-2.0 \cdot r^2) + 7.54 \cdot \exp(-1.0 \cdot r^2)$$

Shell codes: 16 -> s orbital
36 -> p orbital
68 -> d orbital
48 -> f orbital
51 -> g orbital
72 -> h orbital
73 -> i orbital

$$\phi(r) \sim \exp(-Zr^2)$$

-Open, pseudo potential:

One can use a provided basis set as it is.

e.g., cc_pVDZ, cc_pVTZ, etc...

e.g., [<http://burkatzki.com/pseudos/index.2.html>]

e.g., [<https://pseudopotentiallibrary.org>]

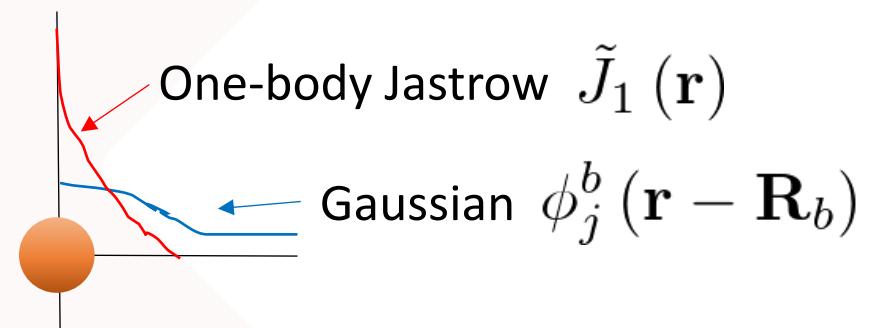
-Open, all-electron:

e.g., basis-set exchange: [<https://www.basissetexchange.org>]

One can use a provided basis set, but it is better to cut several largest exponents.

The WF in the vicinity of nuclei is compensated by the one-body Jastrow part in TurboRVB.

$$\tilde{\phi}_j^b(\mathbf{r} - \mathbf{R}_b) = \phi_j^b(\mathbf{r} - \mathbf{R}_b) \tilde{J}_1(\mathbf{r})$$



TurboRVB employs the CRYSTAL periodic basis for PBC calculations:

$$\psi_{l,m,I}^{\text{PBC}}(\mathbf{r}; \zeta) = \sum_{\mathbf{T}_s} \psi_{l,m,I}(\mathbf{r} + \mathbf{T}_s; \zeta) e^{-i\mathbf{k}_s \cdot \mathbf{T}_s}$$

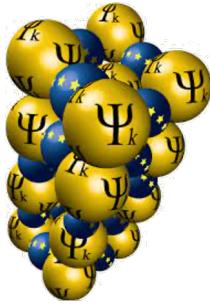
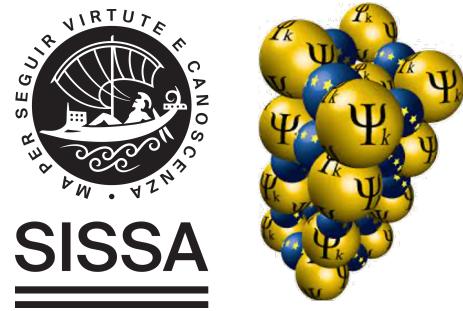
-PBC, pseudo potential:

Unfortunately, provided basis sets for open systems are redundant for periodic cases, so we recommend that one should cut several smaller exponents, typically, smaller than 0.10.

-PBC, all-electron:

The same for all-electron cases. Basis sets provided for open systems such as Basis set exchange [<https://www.basissetexchange.org>] are usually redundant for a periodic case, so we recommend that one should cut several smaller exponents, typically, smaller than 0.10.

One can also use all-electron basis sets optimized for periodic systems such as ones provided in the CRYSTAL DFT code [<https://www.crystal.unito.it/basis-sets.php>].



Day1:

- General Introduction
- Wavefunction representation
- Built-in DFT code



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.

Input: prep.input, fort.10



Binary: prep-(serial, mpi).x



Output: fort10_new

prep.x is a built-in DFT code!!

Why built-in?

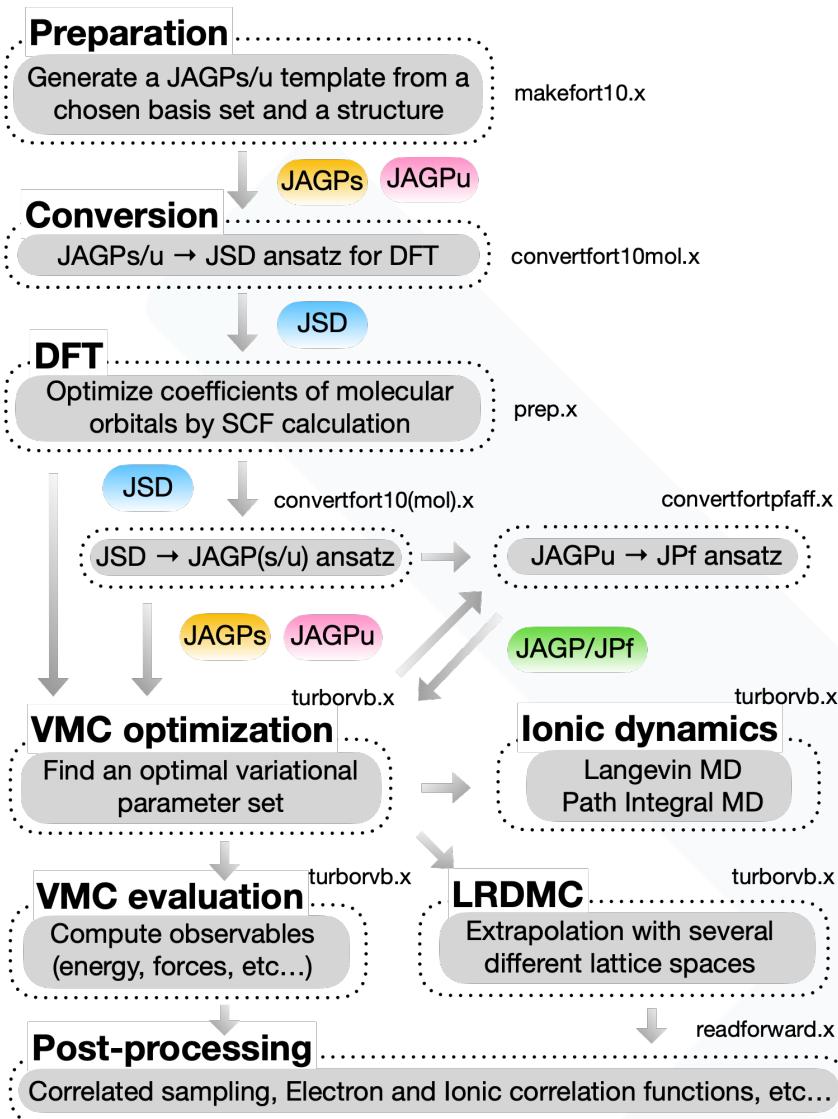
$$\tilde{\phi}_j^b(\mathbf{r} - \mathbf{R}_b) = \phi_j^b(\mathbf{r} - \mathbf{R}_b) \tilde{J}_1(\mathbf{r})$$

As mentioned before, the modified gaussian orbital is used.

So, we cannot exploit the analytical integration even though we employ the Gaussian primitive orbitals.

The CRYSTAL basis + cusp. for PBC cases.

We are also implementing converters for several QC codes (e.g., Gaussian) via TREX-IO.



= Workflow =

1. Prepare a structure and basis set **makefort10.x** **Day1**
2. DFT Construct a reasonable initial WF! **prep.x** **Day1**
3. VMC-opt Optimize the wavefunction **turborvb.x** **Day2**
4. VMC Do a VMC run. **turborvb.x** **Day3**
5. LRDMC LRDMC with the optimized WF. **turborvb.x** **Day4**

Input: convertfort10mol.input,
fort.10_in

Binary: convertfort10mol.x

Output: fort10_new

convertfort10mol.x is a tool for adding molecular orbitals to fort.10_in.

This is used for converting a JAGP WF to a JSD WF.

JAGPs

JSD

JAGPs

$$f(\mathbf{r}_i, \mathbf{r}_j) = \sum_{l,m,a,b} A_{\{a,l\},\{b,m\}} \psi_{a,l}(\mathbf{r}_i) \psi_{b,m}(\mathbf{r}_j)$$

JSD Slater Determinant

$$f(\mathbf{r}_i, \mathbf{r}_j) = \sum_{k=1}^{M=N_{\text{el}}/2} \lambda_k \tilde{\Phi}_k(\mathbf{r}_i) \tilde{\Phi}_k(\mathbf{r}_j)$$

with $\tilde{\Phi}_k = \sum_{i=1}^{N_{\text{basis}}} c_{i,k} \cdot \phi_i(\mathbf{r})$

DFT (prep.x) works only with molecular orbitals!! So, one should convert a WF from the JsAGPs to JSD.

Molecular orbitals (100000): In fort.10, 1000000 indicates a molecular orbital.

```
#always 1,    the number of components, 100000
#index of basis [1,2,...]
#coefficients for basis [1,2,...]
      180      1000000
      1       2       3       4       5
      6       7       8       9      10      11
     12      13      14      15      16      17
     18      19      20      21      22      23
     24      25      26      27      28      29
     30      31      32      33      34      35
     36      37      38      39      40      41
     42      43      44      45      46      47
     48      49      50      51      52      53
     54      55      56      57      58      59
     60      61      62      63      64      65
     66      67      68      69      70      71
     72      73      74      75      76      77
     78      79      80      81      82      83
     84      85      86      87      88      89
   90  0.438271164894104      -4.608166217803955E-002
 0.189550578594208      7.299757003784180E-002 -0.129178702831268
-0.241831779479980     -7.793867588043213E-002 -0.143670558929443
-0.181271851062775     -0.265352427959442      0.374841809272766
 5.072158575057082E-002  0.386640286745070      0.421764402088586
```

$$\Phi_k = \sum_{i=1}^{N_{\text{basis}}} c_{i,k} \cdot \phi_i(\mathbf{r})$$

Molecular orbitals can be added by “convertfort10mol.x”. DFT works only with molecular orbitals.

```
# input file for prep(-mpi).x  
# minimal input
```

```
&simulation  
itestr4=-4  
iopt=1  
/  
&pseudo  
/  
&vmc  
/  
&optimization  
molopt=1  
/  
&readio  
/  
&parameters  
/  
&molecul
```

```
ax=0.2  
ay=0.2  
az=0.2  
nx=50  
ny=50  
nz=130
```

} DFT grids (next page)

```
/  
&dft  
maxit=50  
epsdft=1d-3  
mixing=0.75d0  
typedft=1  
optocc=0  
nelocc=3  
/  
2 2 2
```

DFT parameters:

DFT parameters:

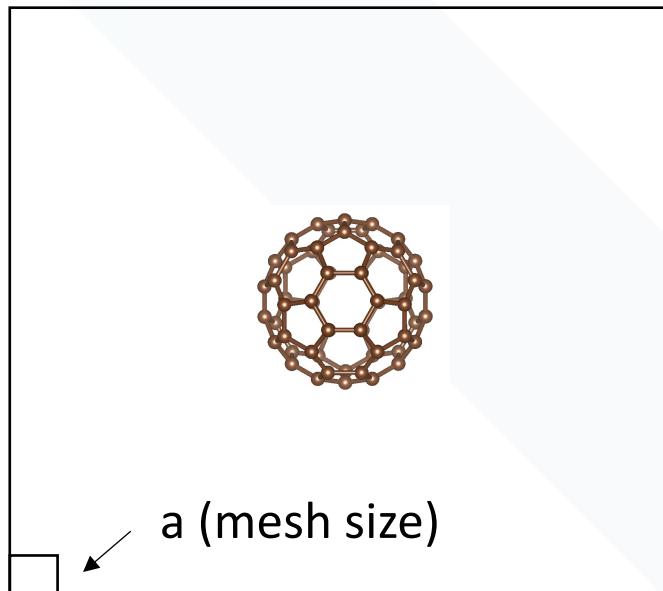
- maxit: Maximum number of iterations in the self-consistent cycle.
- epsdft: Tolerance in the convergence of total energy.
- mixing: Linear simple mixing parameter in the density.
- typedft: 1:LDA (PZ 1981) / 4:Standard LSDA.
- optocc: optocc = 0 (default) Fixed occupation. /
optocc = 1 Use a smeared Fermi distribution
- nelocc: It is the number of occupations that are read.
(only when optocc=0)
/ Fixed occupations

Input: prep.input, fort.10

Binary: prep-(serial, mpi).x

Output:fort10_new

Box and mesh sizes are so important for obtaining converged results in practice !!



$$L \text{ (box size)} = a * n$$

$$H = \hat{T} + V_{\text{ele-ion}}(\vec{r}) + V_{\text{ele-ele}}(\vec{r}) + V_{\text{XC}}(\vec{r})$$

For a calculation with PPs, $a \sim 0.10$ bohr is small enough.

For an all-electron calculation, $a < 0.05$ bohr is needed.
The double-grid algorithm should also be helpful.

If you have enough memories, we recommend
 $L \sim 20$ Bohr for the safety.

L_z = cell length for a periodic system. Automatically set.

Input: prep.input, fort.10



Binary: prep-(serial, mpi).x



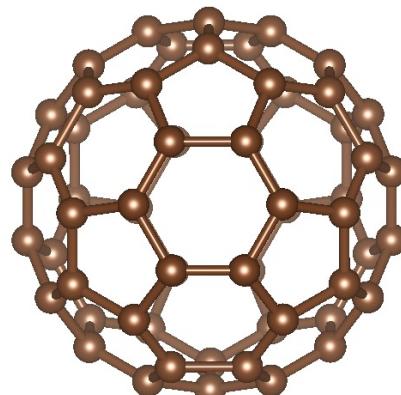
Output:fort10_new

TurboRVB implements only LDA and LSDA functionals.

e.g, typedft = 1 (LDA), The Slater exchange + Perdew81, the corresponding XC in Gaussian is SPL.

e.g, typedft = 4 (LSDA).

Turbo-prep and Gaussian show consistent energies as far as all parameters are the same, i.e., basis-set, XC, box and grid sizes are enough...



C60, Fullerene

Gaussian = -339.7215 (Ha)

Turboprep = -339.7222 (Ha)

Basis=uncontracted cc-pVTZ

PP=ccECP

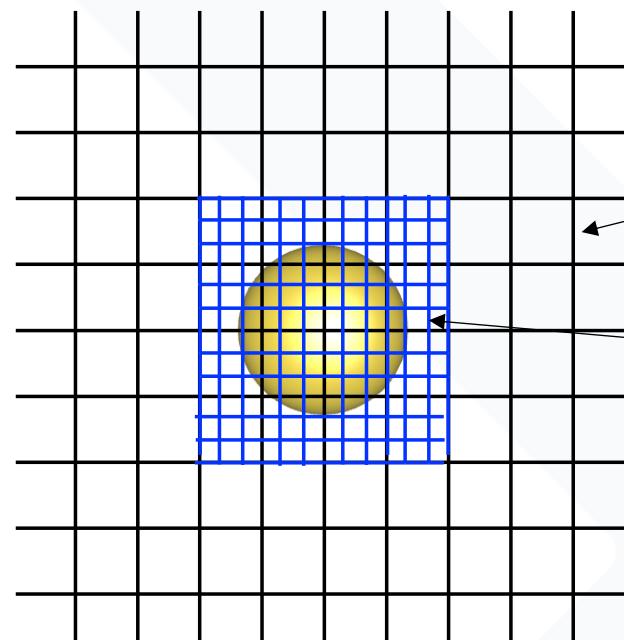
XC=SPL

Input: prep.input, fort.10

Binary: prep-(serial, mpi).x

Output:fort10_new

Double-grid DFT



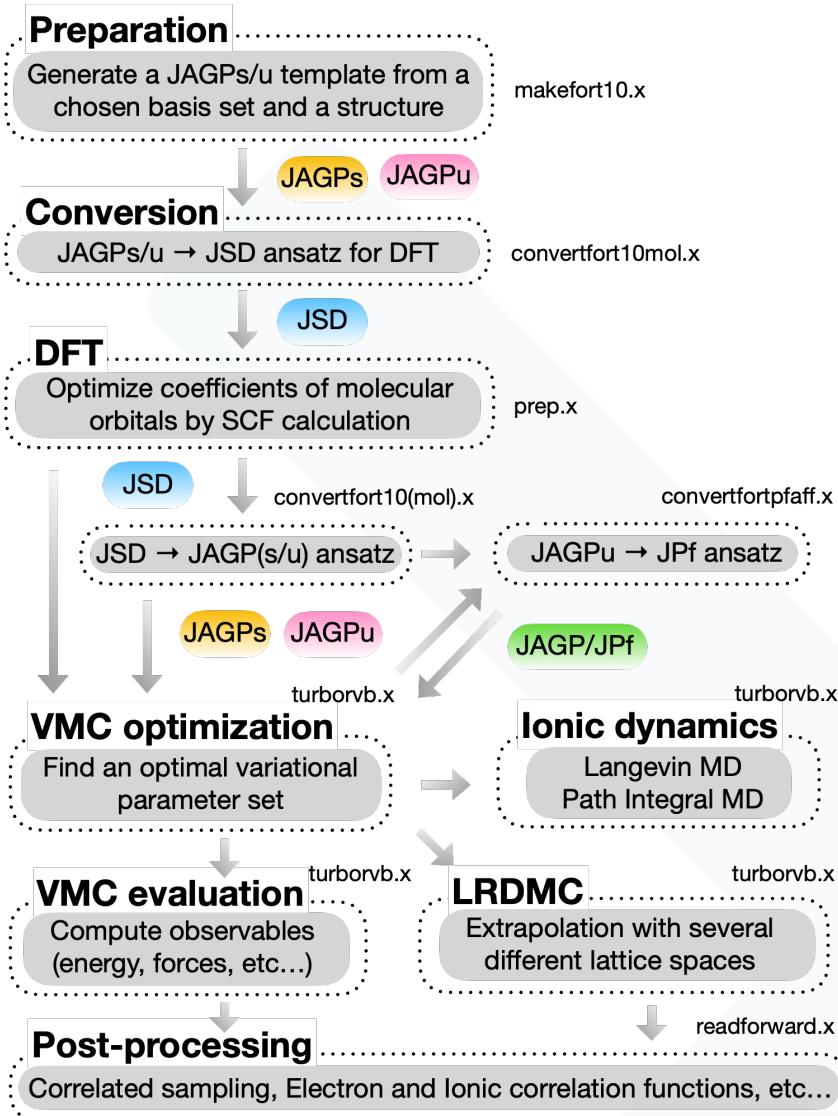
$$H = \hat{T} + V_{\text{ele-ion}}(\vec{r}) + V_{\text{ele-ele}}(\vec{r}) + V_{\text{XC}}(\vec{r})$$

Evaluation on the coarse grid.

Interpolation on the dense grid in the vicinities of nuclei.

Turn on “doublegrid” option in the &simulation section.

K. Nakano, et al., *J. Chem. Theory Comput.* 15, 4044-4055 (2019).

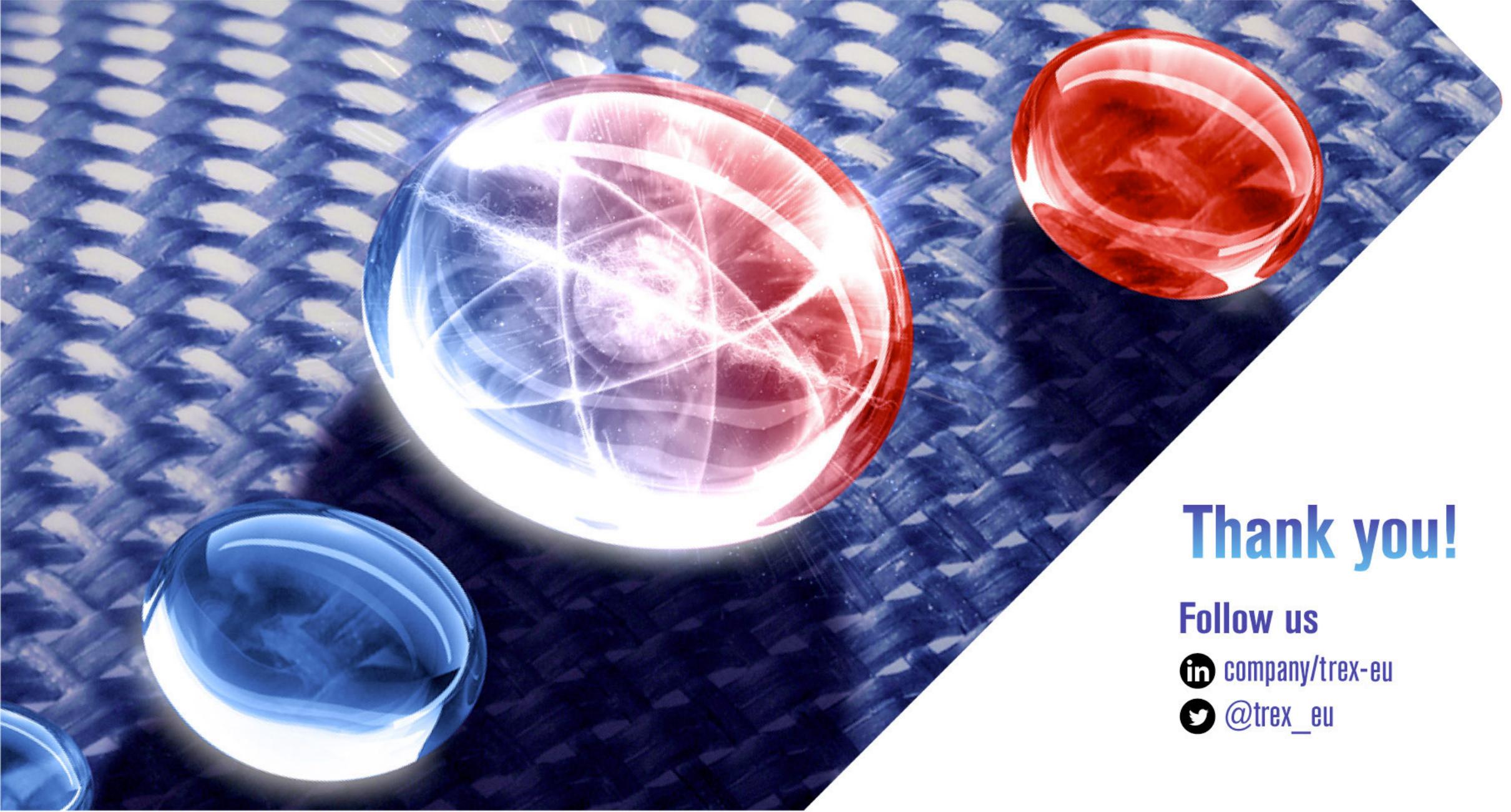


= Workflow =

1. Prepare a structure and basis set **makefort10.x** **Day1**
2. DFT Construct a reasonable initial WF! **prep.x** **Day1**
3. VMC-opt Optimize the wavefunction **turborvb.x** **Day2**
4. VMC Do a VMC run. **turborvb.x** **Day3**
5. LRDMC LRDMC with the optimized WF. **turborvb.x** **Day4**

Ansatz Level

		JDFT								JsAGPs				JAGPu				JAGP(JPf)					
1. H ₂ -dimer	H ₂	01 DFT Day1	02 VMCo Day2,3	03 VMC Day3	04 DMC Day4	06 Conv. Day2	08 VMCo Day2,3	09 VMC Day3	10 DMC Day4	NA				NA				NA					
2. C ₂ -dimer	C ₂	01-01 01-05 DFT Day1	01-06 VMCo Day2,3	01-07 VMC Day3	01-08 DMC Day4	02-01 Conv. Day2	02-02 VMCo Day2,3	02-03 VMC Day3	02-04 DMC Day4	03-01 03-03 DFT Day1	03-04 Conv. Day2	03-05 VMCo Day2,3	03-06 VMC Day3	03-06 DMC Day4	04-01 Conv. Day3	04-03 VMCo Day4	04-03 VMC Day4	04-03 DMC Day4					
	C	Day1	Day2,3	Day3	Day4	Day2	Day2,3	Day3	Day4	03-01 DFT Day1													
3. H ₂ on graphene	H ₂ + Graphene	01 DFT Day1	02 VMCo Day2,3	03 VMC Day3	04 DMC Day4	NA				NA				NA									
	Graphene	05 DFT Day1	06 VMCo Day2,3	07 VMC Day3	08 DMC Day4																		
	H ₂	09 DFT Day1	10 VMCo Day2, 3	11 VMCo Day3	12 DMC Day4																		



Thank you!

Follow us

 [company/trex-eu](#)
 [@trex_eu](#)



Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union
Horizon 2020 research and innovation programme under Grant Agreement **No. 952165**.