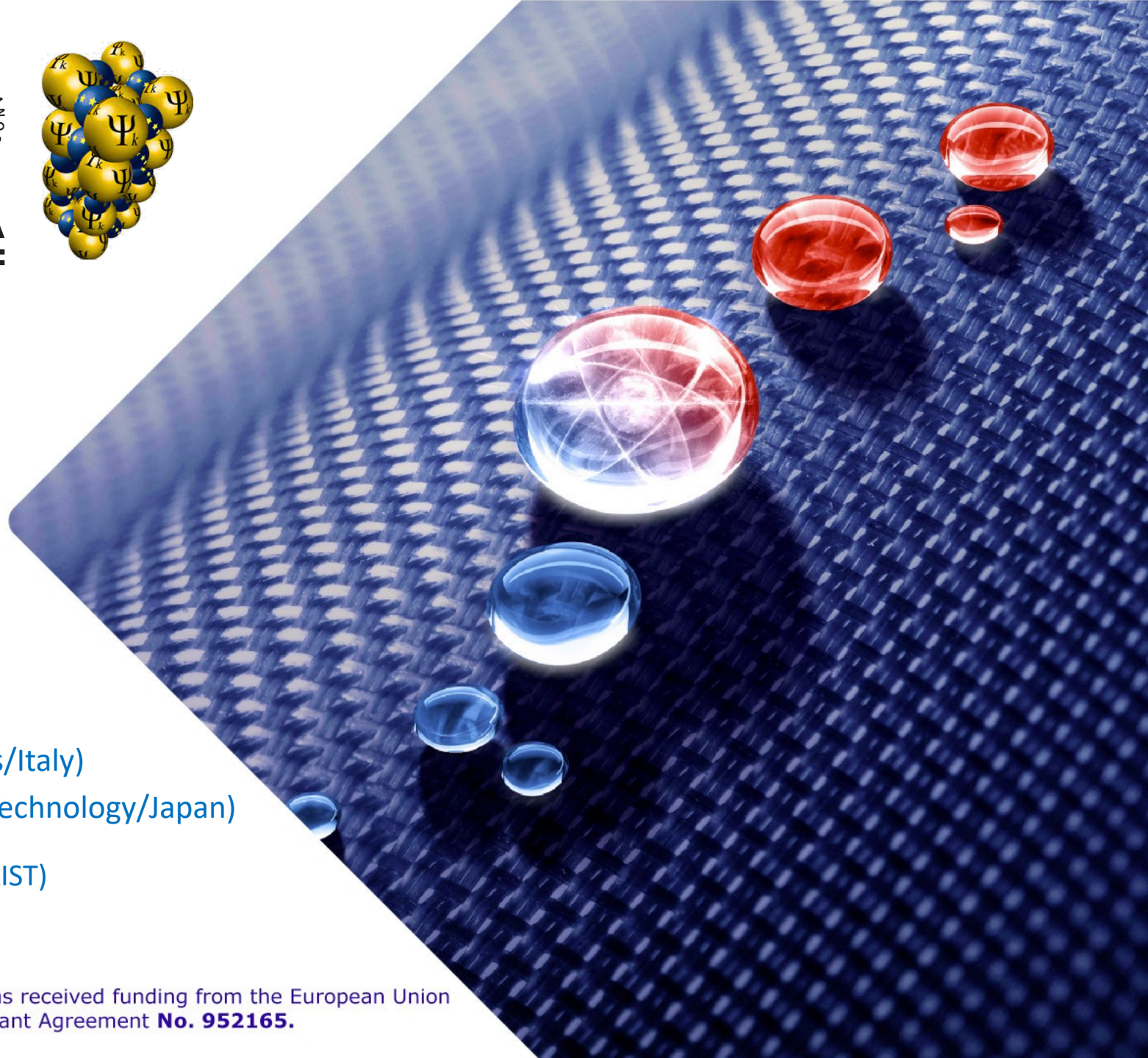
Targeting Real chemical accuracy at the EXascale

SISSA



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.**

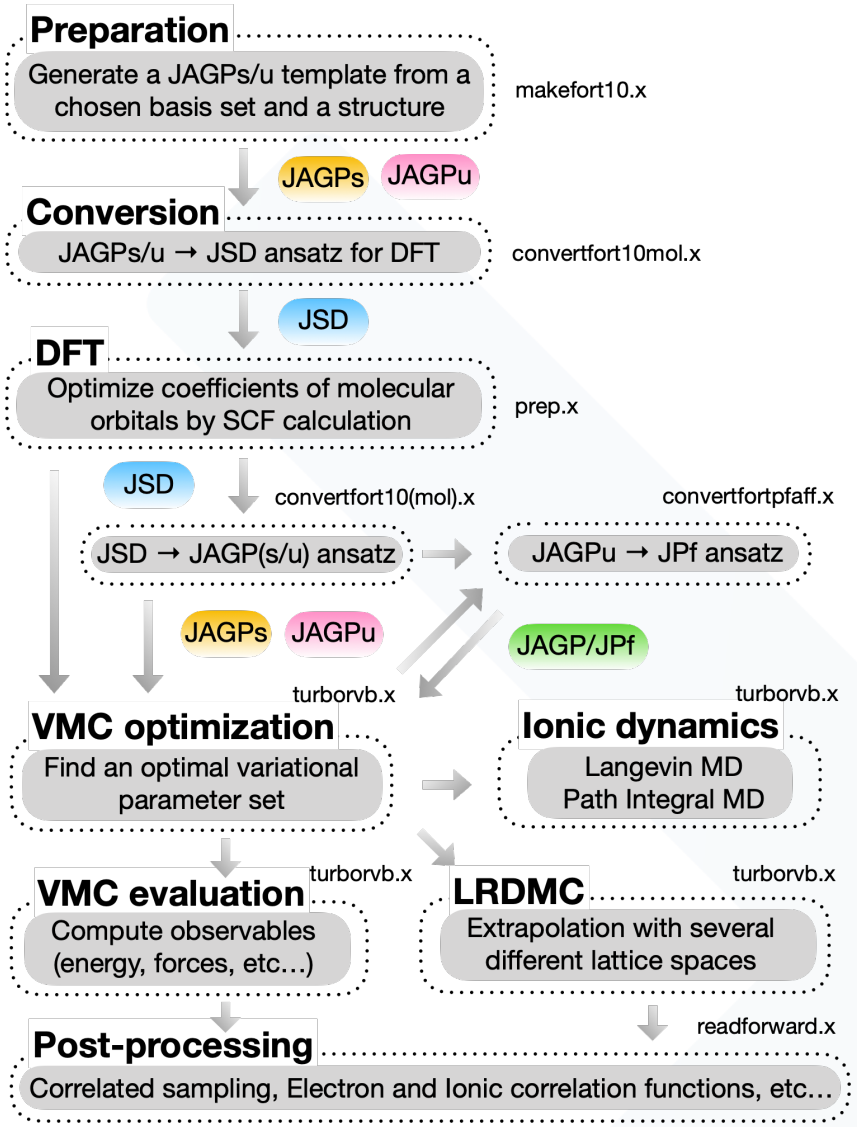
Targeting Real chemical accuracy at the EXascale

SISSA

Day4:
- LRDMC (Lattice regularized DMC)

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.**

A typical workflow in TurboRVB (in detail)



= 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** **Day2/3**

4. VMC Do a VMC run. **turborvb.x** **Day3**

5. LRDMC LRDMC with the optimized WF. **turborvb.x** **Day4**

```
&simulation
    itestr4=-6
    ngen=24100
    iopt=1
    maxtime=10800
/

&pseudo
/

&dmclrdmc
    tbra=0.1
    etry=-5.50
    Klrdmc=0.0
    alat=-0.40
    !iesrandoma=.true.
    !alat2=0.0
    gamma=0.0
    parcutg=1
/

&readio
    !iread=2
/

&parameters
    !ieskin=1
/
```

Important parameters:

Itestr4 = -6: LRDMC

ngen: The number of iterations (branchings)

tbra: Projection time

etry: Energy shift for the projection

alat: Coarser grid size (Bohr).

alat2: Denser grid size (Bohr).

Lattice regularized DMC (LRDMC)

The projection technique to filer out the ground state from a trial wave function (typically, optimized by VMC).

$$|\Upsilon_0\rangle \propto \lim_{M\to\infty} \left(\mathbf{\Lambda} - \hat{\mathcal{H}}\right)^M |\Psi_{\mathrm{T}}\rangle$$

$$= \lim_{M\to\infty} (\lambda - E_0)^M \left[a_0 |\Upsilon_0\rangle + \sum_{n\neq 0} \left(\frac{\lambda - E_n}{\lambda - E_0}\right)^M a_n |\Upsilon_n\rangle \right],$$

Since $\frac{\lambda - E_n}{\lambda - E_0} < 1$ the projection filters out the ground state WF from a given trial WF

In TurboRVB, “etry” is the corresponding parameter. $\lambda = -2 \times \mathrm{etry}$

e.g., one can use a VMC energy for etry.

To apply the GFMC method for continuous systems.

M. Casula et al., ***Phys. Rev. Lett.*** 95, 100201 (2005)

$$\Delta_i f(x_i, y_i, z_i) \approx \Delta_i^a f(x_i, y_i, z_i)$$
$$\equiv \frac{1}{a^2}\{[f(x_i + a) - f(x_i)] + [f(x_i - a) - f(x_i)]\}$$
$$\leftrightarrow y_i \leftrightarrow z_i,$$

$$V^a(\mathbf{x}) = V(\mathbf{x}) + \frac{1}{2}\left[\frac{\sum_i (\Delta_i^a - \Delta_i)\Psi_G(\mathbf{x})}{\Psi_G(\mathbf{x})}\right].$$

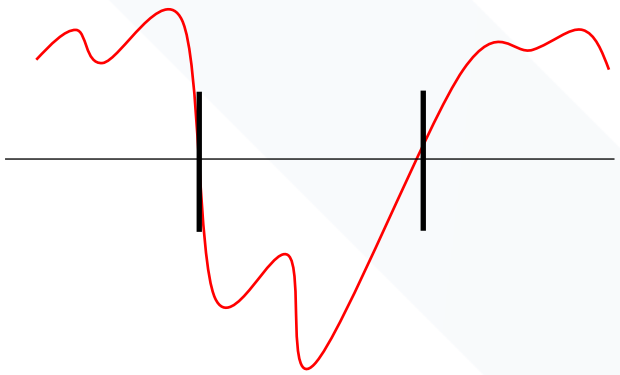
In TurboRVB, "**alat**" is the corresponding parameter. The unit is bohr.

Since the Trotter-Suzuki decomposition is not needed in the LRDMC framework, the "time-step" error does not exist in LRDMC unlike DMC, but this "lattice-space" error exists instead. We need extrapolation for alat. (later)

Fixed-node approximation

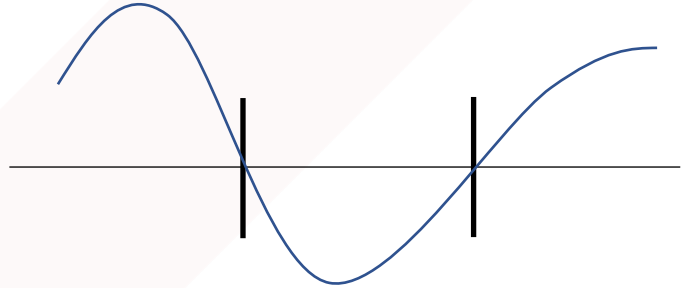
The Green's function cannot be made strictly positive for fermions; therefore, the fixed-node (FN) approximation has to be introduced in order to avoid the sign problem.

Trial WF



Projection.

Fixed-node WF

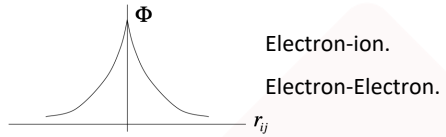


The nodal surface never changes during the simulation! i.e., Only the amplitude is relaxed.

The most difficult part in practice

3. VMC Optimize wavefunctions and VMC run.

A wavefunction reads $\Psi_{AS}(\vec{R}) \times \exp(J(\vec{R}))$ ⟶



Anti-symmetric part. Jastrow factor.

To satisfy the cusp.

Jastrow factor = $\exp(J(\vec{R}))$ No effect on the nodal surface!! $\Psi(\vec{R}) \equiv \Psi(\vec{r_1}, \vec{r_2}, \cdots, \vec{r_N}) = 0$

Anti-symmetric part = $\Psi_{AS}(\vec{R})$ Determines the nodal surface. Its initial guess is taken from a DFT calculation!!

TurboRVB:

Both parts are optimized.

Stable nodal surface optimization

Independent of an initial DFT guess!?

Other QMC codes:

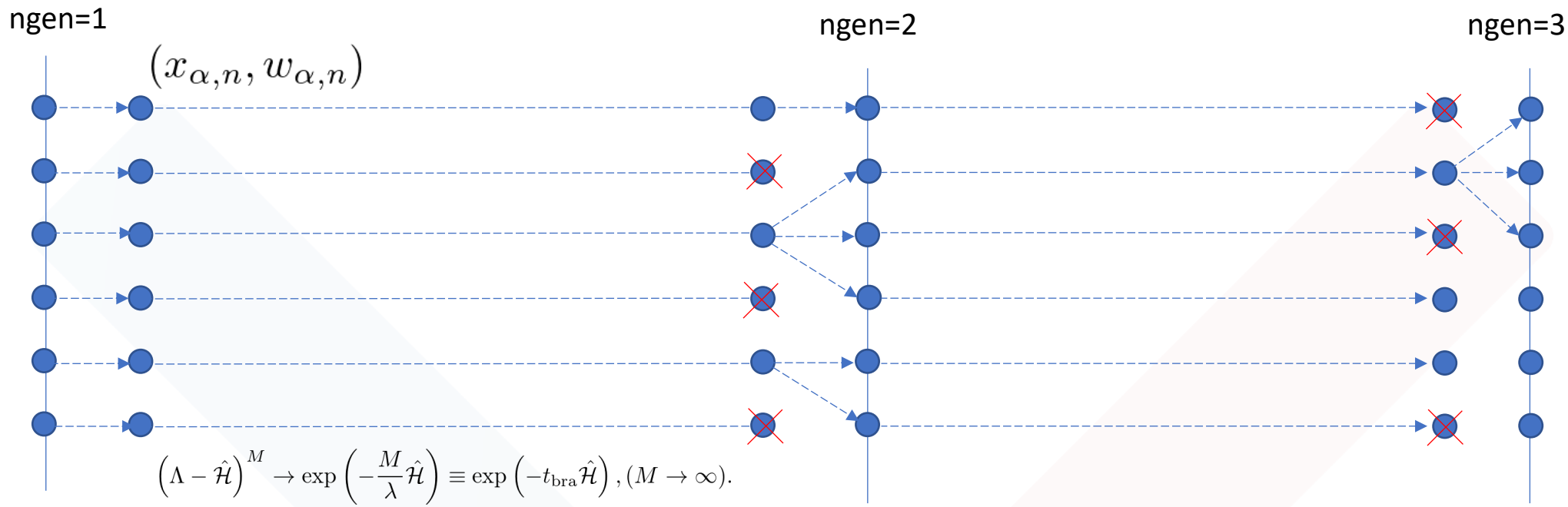
Only the Jastrow part is optimized.

Unstable nodal surface optimization

Dependent on an initial DFT guess!!

e.g., however, the optimization sometimes is stuck in a local minimum if the initialization was wrong etc…

Branching and related parameters



weights and positions are updated.

Branching!!

1. Set the new weights $w'_{\alpha,n} = \bar{w} \equiv \frac{1}{N_w}\sum_\beta w_{\beta,n}.$
2. Select the new walkers $p_{\alpha,n} = w_{\alpha,n}/\sum_\beta w_{\beta,n},$

- “ngen” is the number of branchings! (ngen).
- The branching is done every τ_{nbra} steps. (tbra).
  - too small tbra -> The weights are not update.
  - too large tbra -> Only few walkers survive.

Check your output! Av. num. of survived walkers/ # walkers in the branching 0.99 > 0.90!

Reblocking

forcefn.sh “bin”, “corr”, “init”, “pulay”, or

turbo-genius.sh –j lrdmc -post -reb “bin”, -eq “init” -col “corr”

pip0_fn.d=energy

```
% cat pip0_fn.d
number of bins read =        1201
Energy =  -11.0854289356563       1.239503202184784E-004
Variance square =  0.126708380716482       1.148750765092961E-003
Est. energy error bar =  1.234807072779590E-004  2.503947626011507E-006
Est. corr. time  =   1.85075908836029       7.596952532743223E-002
Energy (ave) = -11.0854159959592 1.144905833254917E-004
```

forcefn.dat=forces

```
Force component 1
Force    =  6.004763869201490E-003  4.997922374161991E-005
6.273565633363322E-007
Der Eloc =  6.927675852724724E-003  4.999242839793062E-005
<OH> =  0.557134685159244       7.437283601136703E-005
<O><H> = -0.557596141151006       7.447559481785158E-005
2*(<OH> - <O><H>) = -9.229119835232336E-004  2.922997214772288E-006
```

“bin”: the length of reblocking (binning) size

“corr”: correcting factor

“init”: the length of equilibration steps (init * bin)

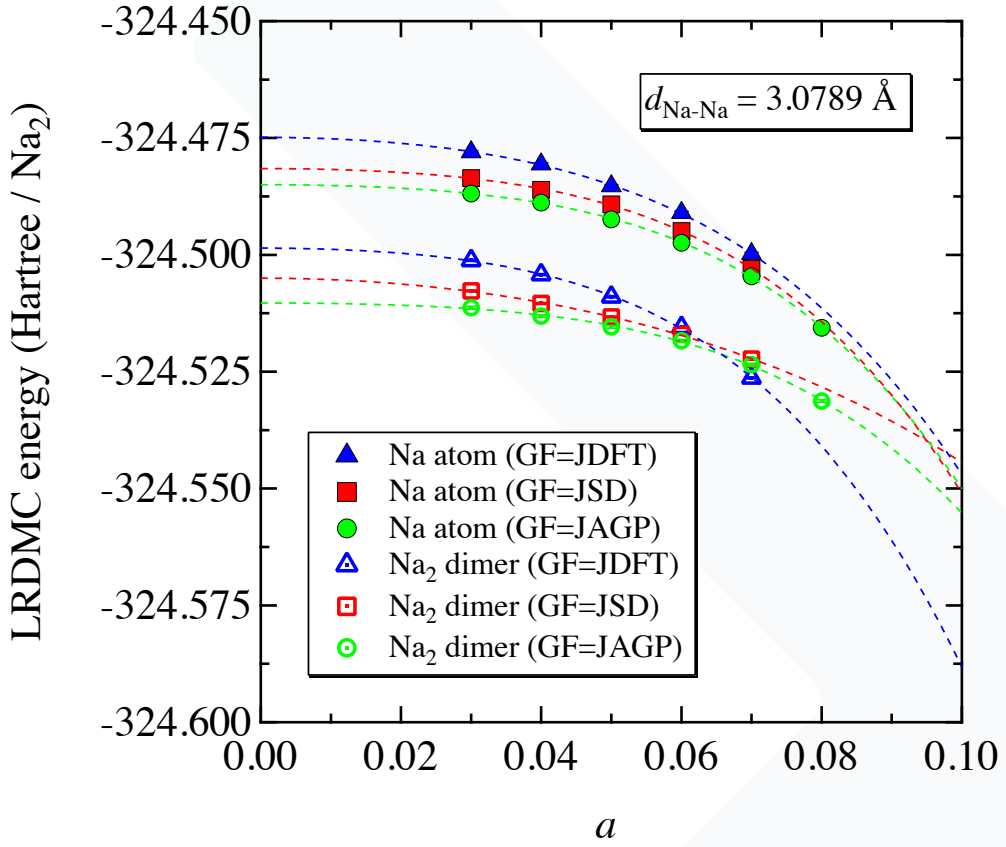
“pulay”: the ratio of the pulay term (1 is OK)

“corr”: correcting factor (p)

The average weights are stored and are set to one for all walkers after each branching.

$$E_0 \approx \frac{\sum_n \mathcal{G}_n^p e_L(x_n)}{\sum_n \mathcal{G}_n^p} \quad \mathcal{G}_n^p = \prod_{j=1}^{p} \bar{w}_{n-j},$$

$$\Delta_i f\left(x_i, y_i, z_i\right) \approx \Delta_i^a f\left(x_i, y_i, z_i\right)$$
$$\equiv \frac{1}{a^2}\left\{\left[f\left(x_i + a\right) - f\left(x_i\right)\right] + \left[f\left(x_i - a\right) - f\left(x_i\right)\right]\right\}$$



alat extrapolation with funvsa.x

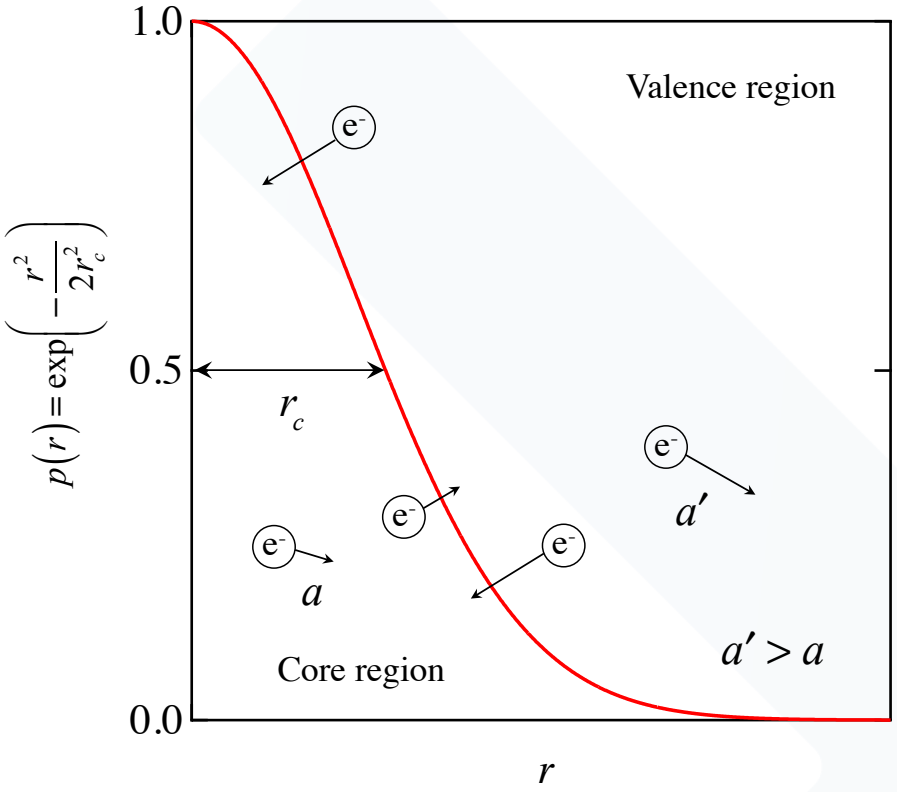
quartic: $E(a) = E(0) + k_1 \cdot a^2 + k_2 \cdot a^4$

quadratic: $E(a) = E(0) + k_1 \cdot a^2$

```
# See. Readme of funvsa.x in detail.
# 2=(up to a^4) number of data 4 1
2  5  4  1
0.10 -11.0850188375511 1.250592379643920E-004
0.20 -11.0854289356563 1.239503202184784E-004
0.30 -11.0855955871707 1.334024389855123E-004
0.40 -11.0860656088368 1.279739901272860E-004
0.50 -11.0868942724581 1.340429878094154E-004
```

The extrapolation behaves well unlike the standard DMC!!

Double grid algorithm



Double grid algorithm

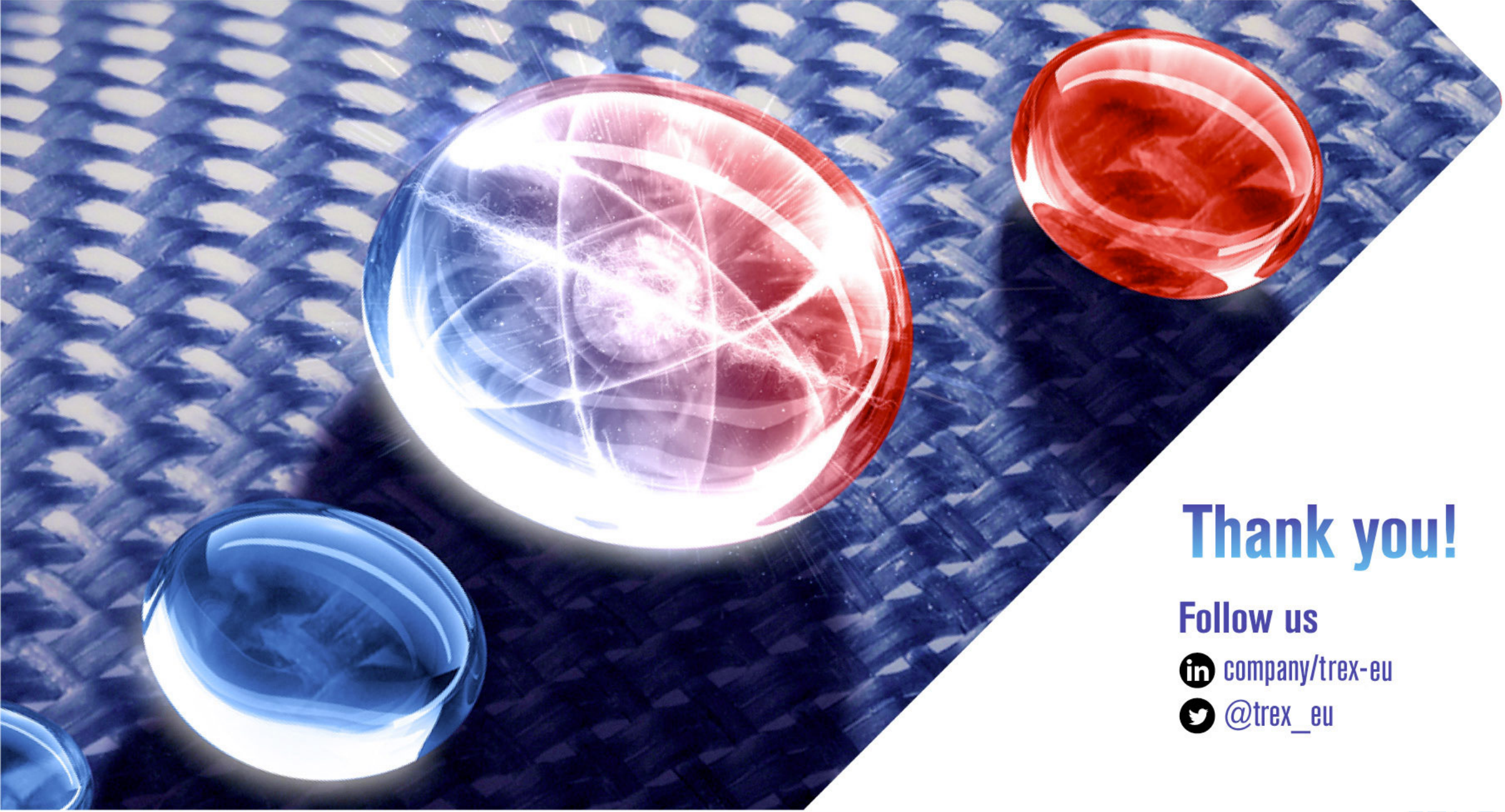
Key parameters

r_c	determines the core length	(plat)
a'	step for the valence region	(alat2)
a	step for the core region	(alat)

When you put a positive value for **alat**, TurboRVB automatically sets proper parameters for plat and alat2.

K. Nakano et al., Phys. Rev. B 101, 155106 (2020)

		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			
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									03-01 DFT Day1					04-02 Conv. Day3			
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.**

TREX
Targeting Real chemical accuracy at the EXascale