Targeting Real chemical accuracy at the EXascale

# TREX e-school with TurboRVB

## Lecturer: Sandro Sorella

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.**

# Lecture III : QMC Wave function Optimization

- Reminding Variational quantum Monte Carlo
- Recent progress in stochastic optimization
- Correlated sampling
- Direct evaluation of energy derivatives
- Stochastic reconfiguration or Natural gradients
- Examples

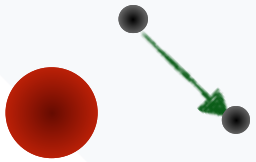
# Jastrow |Mean Field> : known facts

- 1D. The Jastrow- Fermi gas is the exact ground state for the Tomonaga-Luttinger model (B. Tayo and S.Sorella Phys. Rev.B, 2008)
- D>1 Spin-Wave theory ground state is a Jastrow |MF=Neel>
- Good description of Mott insulators ( M. Capello et al. PRL 2005)
- Spin liquid, the ground state of the Kitaev model can be written as a Gutzwiller projected pairing function with triplet correlations, again J( Gutzwiller-projection)-MeanField=Pfaffian function discussed before.
- Superconductivity in strongly correlated models (e.g. Hubbard) without the electron-phonon mechanism.
- Laughlin's WF FQHE is a (complex) Jastrow- MeanField (Jain PRB' 90)
- Recently Valence bond Solids (Kekule' ) within a single determinant ansatz, taking into account several configurations.

$$\pi(x) = \frac{|\Psi(x)|^2}{\int dx' |\Psi(x')|^2} \rightarrow \{x_i\}_\Psi \qquad x = \{\mathbf{r}_1\sigma_1, \mathbf{r}_2\sigma_2, \ldots, \mathbf{r}_N\sigma_N\}$$

- **Sampling the electron moves with Markov chains**

$$x \rightarrow x'$$



$$P(x \rightarrow x') = \min\left(1, \frac{|\langle x'|\Psi\rangle|^2}{|\langle x|\Psi\rangle|^2}\right)$$

- **The variational principle**

$$\frac{\langle\Psi|H|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \int dx \pi(x) e_L(x) \approx \frac{1}{M} \sum_{\text{M samples}} e_L(x_{\text{sample}})$$

$$e_L(x) = \frac{\langle x|H|\Psi\rangle}{\langle x|\Psi\rangle}$$ ← The local energy

You believe in human learning and define a wave function $|\Psi_\alpha\rangle = J|\Psi_{\mathrm{MF}}\rangle$ where “α” indicates a set of variational parameters and J is a correlation term.

For instance J = Gutzwiller projection, Jastrow correlation, three-body ….

The expectation value of the Hamiltonian, due to J, can be evaluated statistically:

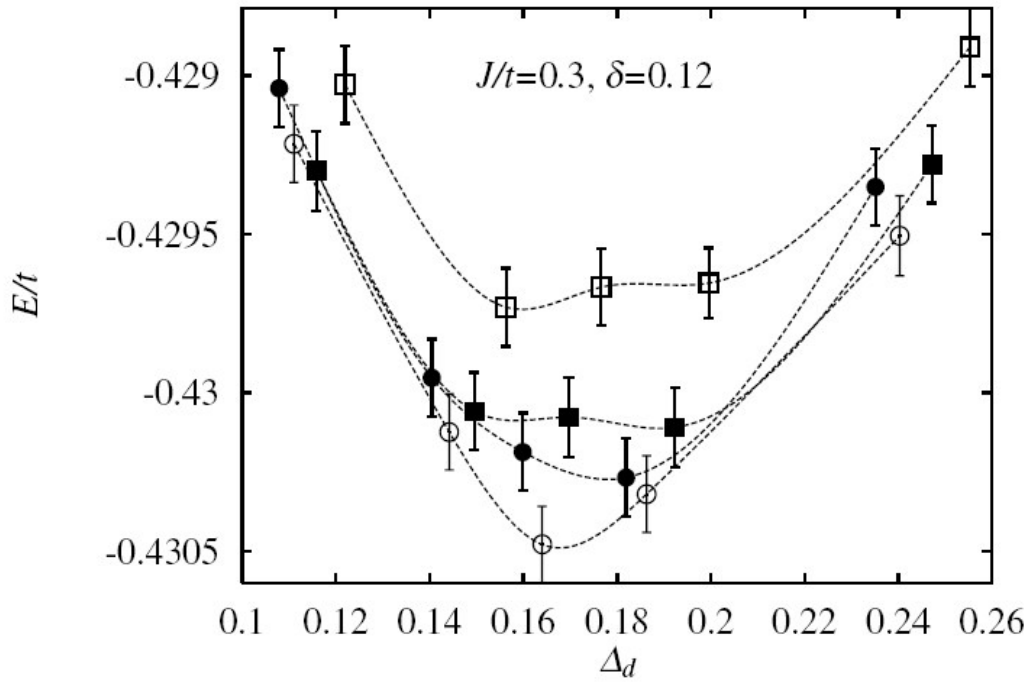
$$E_{VMC}(\alpha) = \frac{\langle\Psi_\alpha|H|\Psi_\alpha\rangle}{\langle\Psi_\alpha|\Psi_\alpha\rangle}$$ by QMC sampling

In the last decade ML and VMC were able to handle several parameters even when the target energy is known only statistically.

# The "game" started in '65 by McMillan

$He_4$ was studied with 2-parameters : $J = \prod_{i<j} \exp(-a_1/r_{ij}^{a_2})$

But also "recently" (M. Ogata et al. PRL 85 (2000) J= ~~$\prod \exp(-a_1/r_{ij})$~~ :




The standard: parameters were optimized with several independent energy runs:

1) at most two variational parameters
2) large computational resources for a single optimization

# Several variational parameters problem

Many parameters → Mean-field self consistency

e.g. Slater determinant in a lattice contains
L (number of sites) x N (number of electrons)
variational parameters. No problem with
Hartree, Hartree-Fock, LDA, CGA,HSE…

In a many-body wf, self-consistent approaches are not available (what are the sc-fields?) and
→ Many parameters optimization problem
→ Moreover the energy evaluation is noisy

# Correlated sampling, Umrigar 1988

We want to compute the energy $E(\alpha') = \frac{\langle \Psi_{\alpha'} | H | \Psi_{\alpha'} \rangle}{\langle \Psi_{\alpha'} | \Psi_{\alpha'} \rangle}$

Suppose to employ a Markov chain with the original weight

$$W_\alpha(x) = |\Psi_\alpha(x)|^2 \neq |\Psi_{\alpha'}(x)|^2, \text{ i.e. } \pi(x) = \frac{W_\alpha(x)}{\sum_{x'} W_\alpha(x')}$$

Then we can compute: $R(x) = \frac{W_{\alpha'}(x)}{W_\alpha(x)} = \frac{|\Psi_{\alpha'}(x)|^2}{|\Psi_\alpha(x)|^2}$

$$E_{\alpha'} = \frac{\sum_x \pi(x) R(x) e_L^{\alpha'}(x)}{\sum_x R(x)\pi(x)} = \frac{\sum_{\text{M samples}} e_L^{\alpha'}(x_{\text{sample}}) R(x_{\text{sample}})}{\sum_{\text{M samples}} R(x_{\text{sample}})}$$

Go→

Essentially variance R(x) ~ exp ( # electrons)

Some improvement was to consider the minimization of the variance but the problem is only alleviated not solved.

Thus the approach is limited to very few electrons N<= 10

# Second progress, calculation of energy derivatives

$$E(\alpha) = \frac{\int dx^{3N} \Psi_\alpha(x) H \Psi_\alpha(x)}{\int dx^{3N} \Psi_\alpha^2(x)}$$

$$\begin{aligned}
\partial_\alpha E(\alpha) &= 2\mathrm{cov}(e_L(x), O(x)) \\
\pi_(x) &\propto \Psi_\alpha^2(x) \\
O(x) &= \partial_\alpha \ln |\Psi_\alpha(x)| \\
e_L(x) &= \frac{[H\Psi_\alpha](x)}{\Psi_\alpha(x)}
\end{aligned}$$

Go→

Next attempt ~ 2000: try to compute derivative and apply steepest descent:

$$\alpha'_k - \alpha_k = \delta\alpha_k = -\Delta \frac{\partial E}{\partial \alpha_k} = \Delta f_k$$

For small enough Δ the energy should go down. So one should optimize at each step by determining the optimal Δ:

$$\mathrm{Min}_\Delta E(\vec{\alpha} + \Delta \vec{f})$$

Go (explain SD)→

# H2: simplest Jastrow+simplest geminal

$$J(\vec{x}) = J_1(\vec{x})J_2(\vec{x})$$

$$J_1(\vec{x}) = \exp\left[\sum_{I=1}^{N} u_{ei}(\vec{r}_i)\right]$$

$$J_2(\vec{x}) = \exp\left[\sum_{i<j} u_{ee}(|\vec{r}_i - \vec{r}_j|)\right]$$

$$u_{ei}(\vec{r}) = -\sum_I Z_I \frac{|\vec{r} - \vec{R}_I|}{1 + a|\vec{r} - \vec{R}_I|}$$

$$u_{ee}(r) = \frac{r}{2(1+br)}$$

$$\text{Basis} : \phi_{1,a}(\vec{r}) = \exp(-Z|\vec{r} - \vec{R}_a|^2) \; a = 1, 2$$

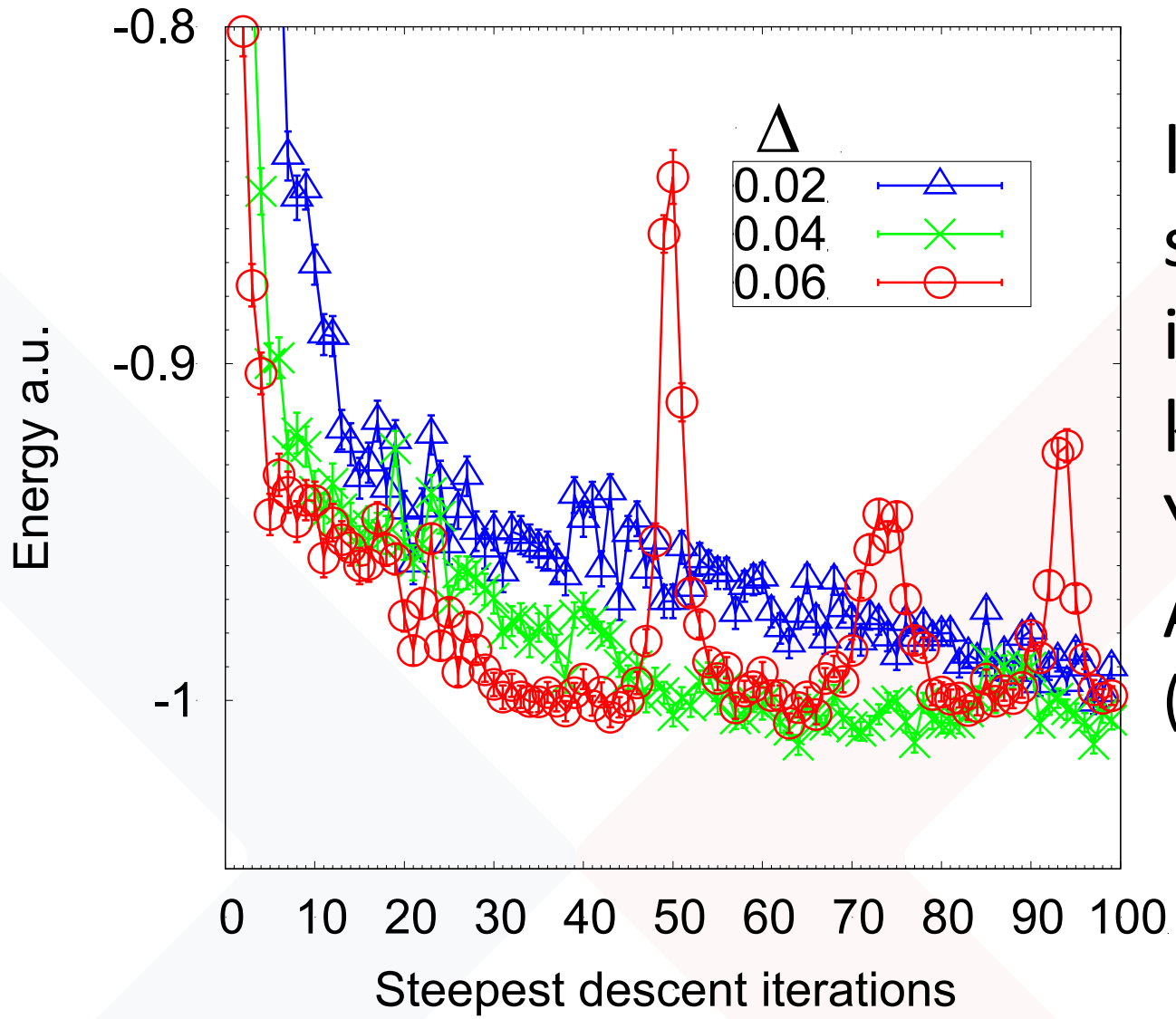
$$g(\vec{r}, \vec{r}') = \sum_{ia,jb} \lambda_{ia,jb}\phi_{ia}(\vec{r})\phi_{jb}(\vec{r}')$$

$$\lambda_{11,11} = \lambda_{12.12} = 1$$

$$\lambda_{11,12} = \lambda_{12.11} = \lambda$$

By symmetry only two parameters λ & Z in g

# In practice one chooses Δ (tpar in TurboRVB) once for all




If you want to try steepest descent in TurboRVB:
kl=-2
You can also use ADAMS (from ML) (yes_adams=.true.)

## Setting constant Δ is equivalent to optimize:

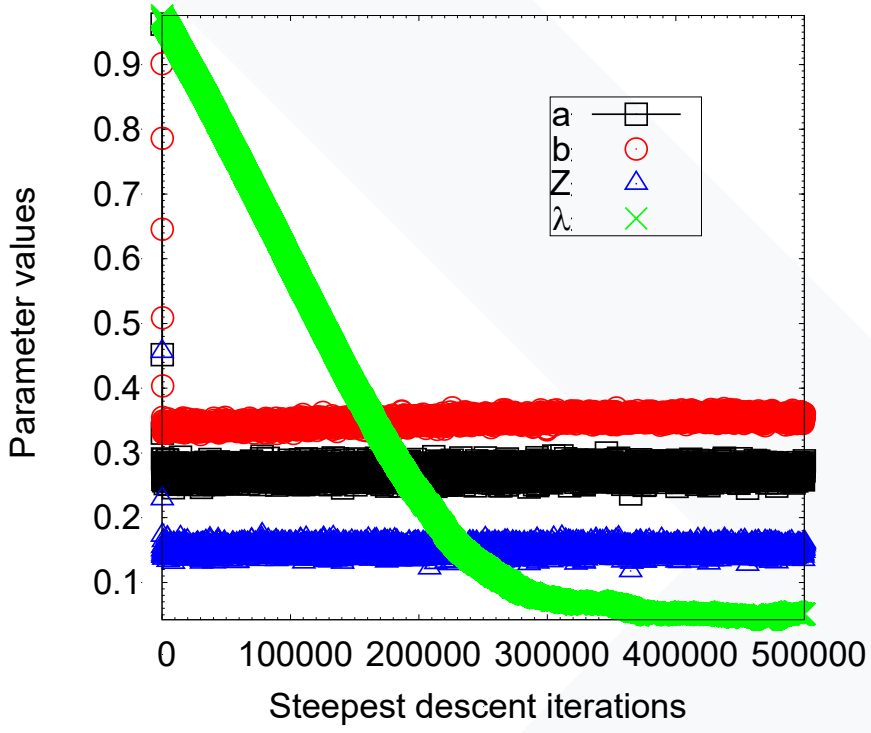
$$\mathrm{Cost}(\delta\alpha) = E(\vec{\alpha}) + \sum_k \left( \delta\alpha_k f_k + \frac{1}{2\Delta} \delta\alpha_k^2 \right)$$

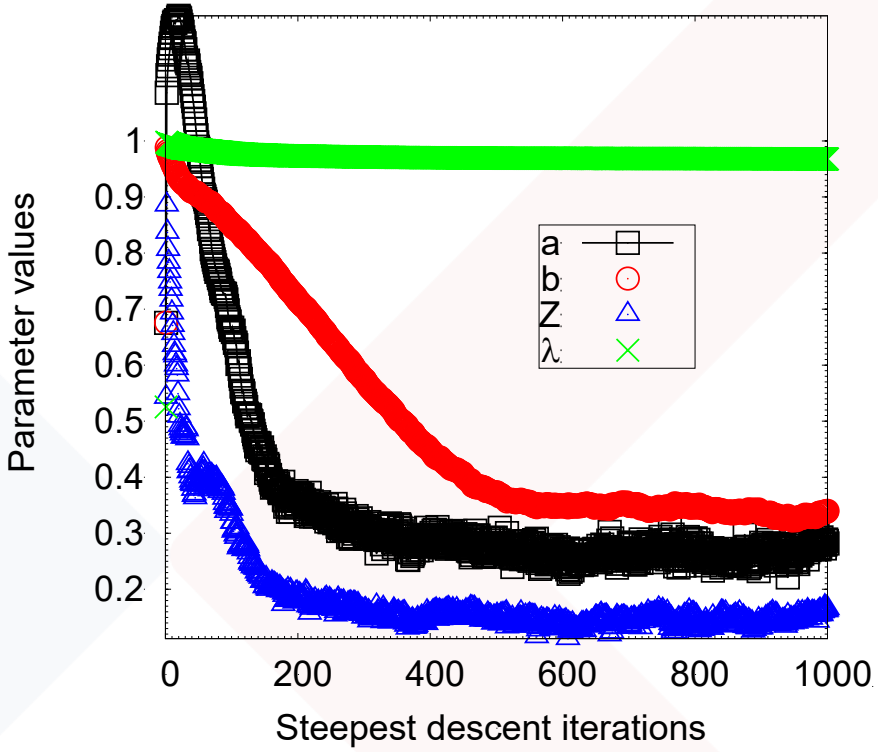
We implicitly optimize a Cost function at each step that penalizes big changes of variational parameters.

Is this the optimal thing to do?

There are directions that are extremely difficult to optimize, usually the most physical ones…

$H_2$ molecule (see tutorial) R=0.8 a.u. 4 parameters optimization



For each variational parameter $\alpha_k$ of a vector

$$\vec{\alpha} = \{\alpha_1, \alpha_2, \cdots \alpha_p\}$$

We can define an operator $O_k$:

$$O_k(x) = \frac{\partial_{\alpha_k} \Psi_{\vec{\alpha}}(x)}{\Psi_{\vec{\alpha}}(x)} \text{ for } \Psi_{\vec{\alpha}}(x) \neq 0$$

$$\langle x | O_k | x' \rangle = \delta_{x,x'} O_k(x)$$

$$|\Psi_{\vec{\alpha}+\delta\vec{\alpha}}\rangle = \left[1 + \sum_{k=1}^{p} \delta\alpha_k O_k\right] |\Psi_{\vec{\alpha}}\rangle$$

Go→

But we are interested to normalized wf

$$|\psi_{\vec{\alpha}}\rangle = \frac{|\Psi_{\vec{\alpha}}\rangle}{||\Psi_{\vec{\alpha}}||}$$

Thus:

$$\begin{aligned} |\psi_{\vec{\alpha}+\delta\vec{\alpha}}\rangle &= \left(1 + \sum_{k=1}^{p} \delta\alpha_k (O_k - \bar{O}_k)|\psi_{\vec{\alpha}}\rangle\right) \\ \bar{O}_k &= \langle\psi_{\vec{\alpha}}|O_k|\psi_{\vec{\alpha}}\rangle = \langle\langle O_k(x)\rangle\rangle \end{aligned}$$

But now we can ask how much we actually change the wf. when we change $\alpha'_k \rightarrow \alpha_k + \delta\alpha_k$

$$ds^2 = || \frac{|\Psi_{\vec{\alpha}'}\rangle}{||\Psi_{\vec{\alpha}'}||} - \frac{|\Psi_{\vec{\alpha}}\rangle}{||\Psi_{\vec{\alpha}}||} ||^2 = \sum_{k,k'} \delta\alpha_k \delta\alpha_{k'} S_{k,k'}$$

$$S_{k,k'} = \mathrm{cov}(O_k, O_{k'})$$

This matrix is also known as the Fisher information metric F=4 S of the probability $p_{\vec{\alpha}}(x) \propto \Psi_{\vec{\alpha}}(x)^2$

Go→

Thus instead of using the Euclidean metric we can use the more appropriate Fisher information metric to define our cost function:

$$\mathrm{Cost}(\delta\vec{\alpha}) = E(\vec{\alpha}) - \sum_k f_k \delta\alpha_k + \frac{ds^2}{2\Delta}$$

$$ds^2 = \sum_{k,k'} \delta\alpha_k \delta\alpha_{k'} S_{k,k'}$$

$$\mathrm{Min}_{\delta\vec{\alpha}}\mathrm{Cost}(\delta\vec{\alpha}) \rightarrow \delta\alpha_k = \Delta \sum_{k'} S^{-1}_{k,k'} f_{k'}$$

# The final updates for small Δ

Steepest descent: $$\delta\alpha_k = \Delta f_k$$

Newton-Raphson: $$\delta\alpha_k = \sum_{k'} H^{-1}_{k,k'} f_{k'}$$

SR or natural gradients: $$\delta\alpha_k = \Delta \sum_{k'} S^{-1}_{k,k'} f_{k'}$$

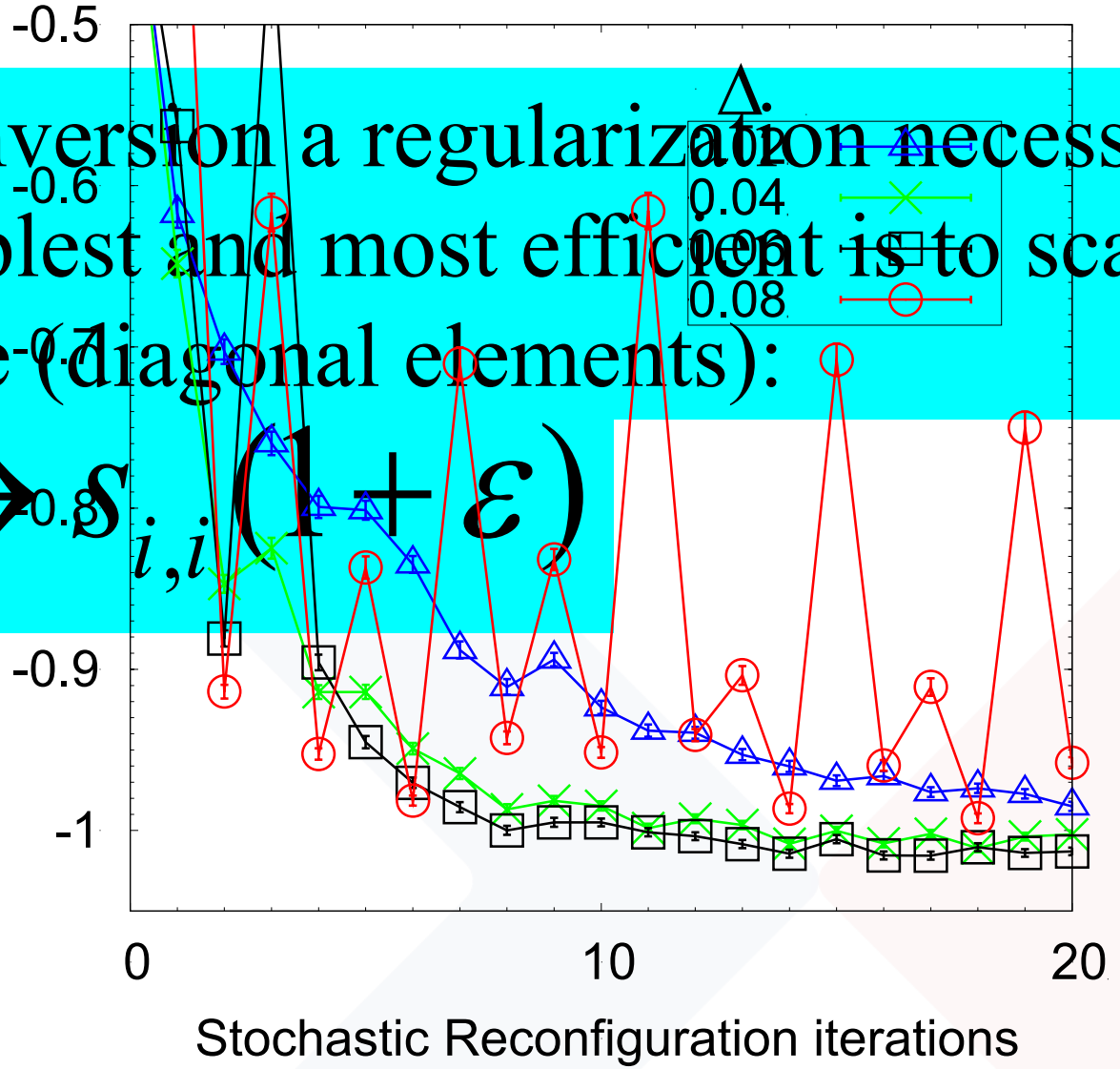
See: S. S. PRL **80**, 4558 (1998), PRB **64**, 240512 (2001)

Natural gradients: S.I. Amari *Neural Computation* **10, 251–276** (1998)

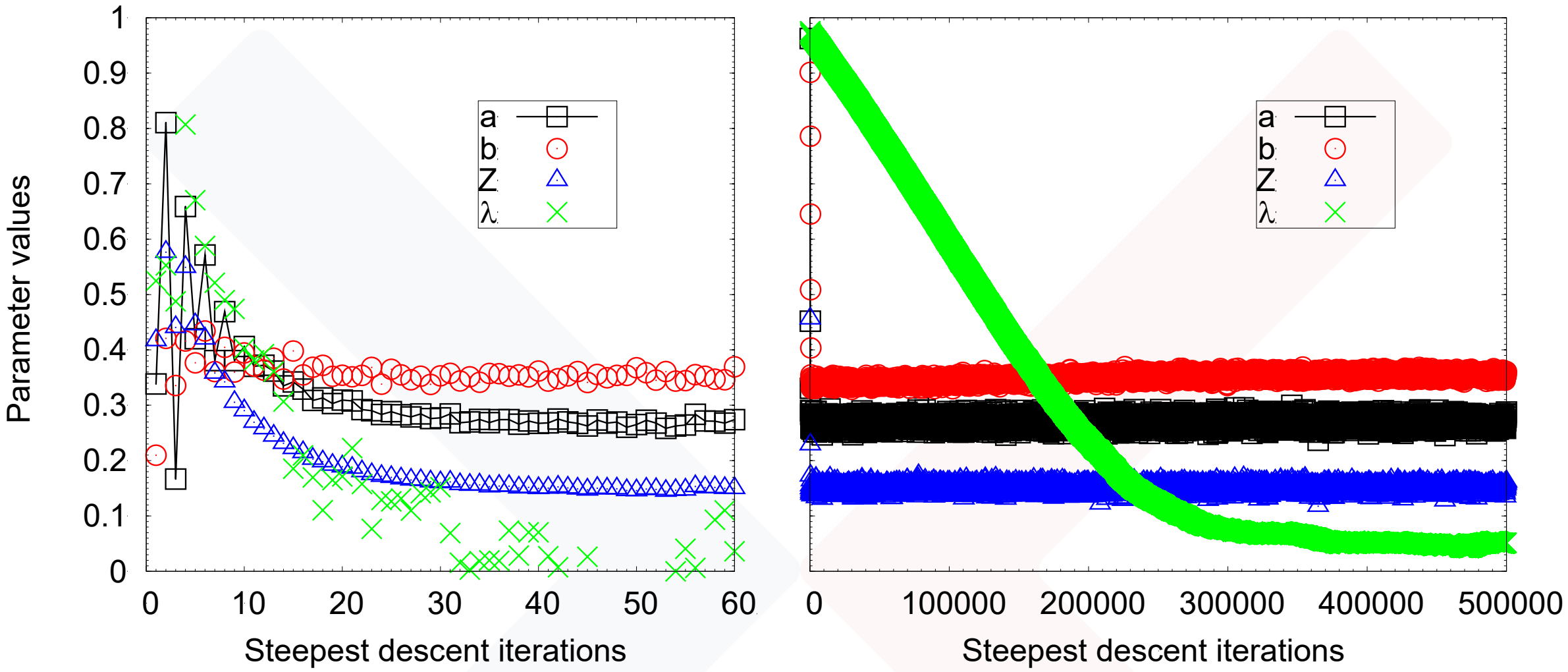
# Test case $H_2$ 4 variational parameters ε=0.01

In the inversion a regularization necessary: the simplest and most efficient is to scale the variance (diagonal elements):

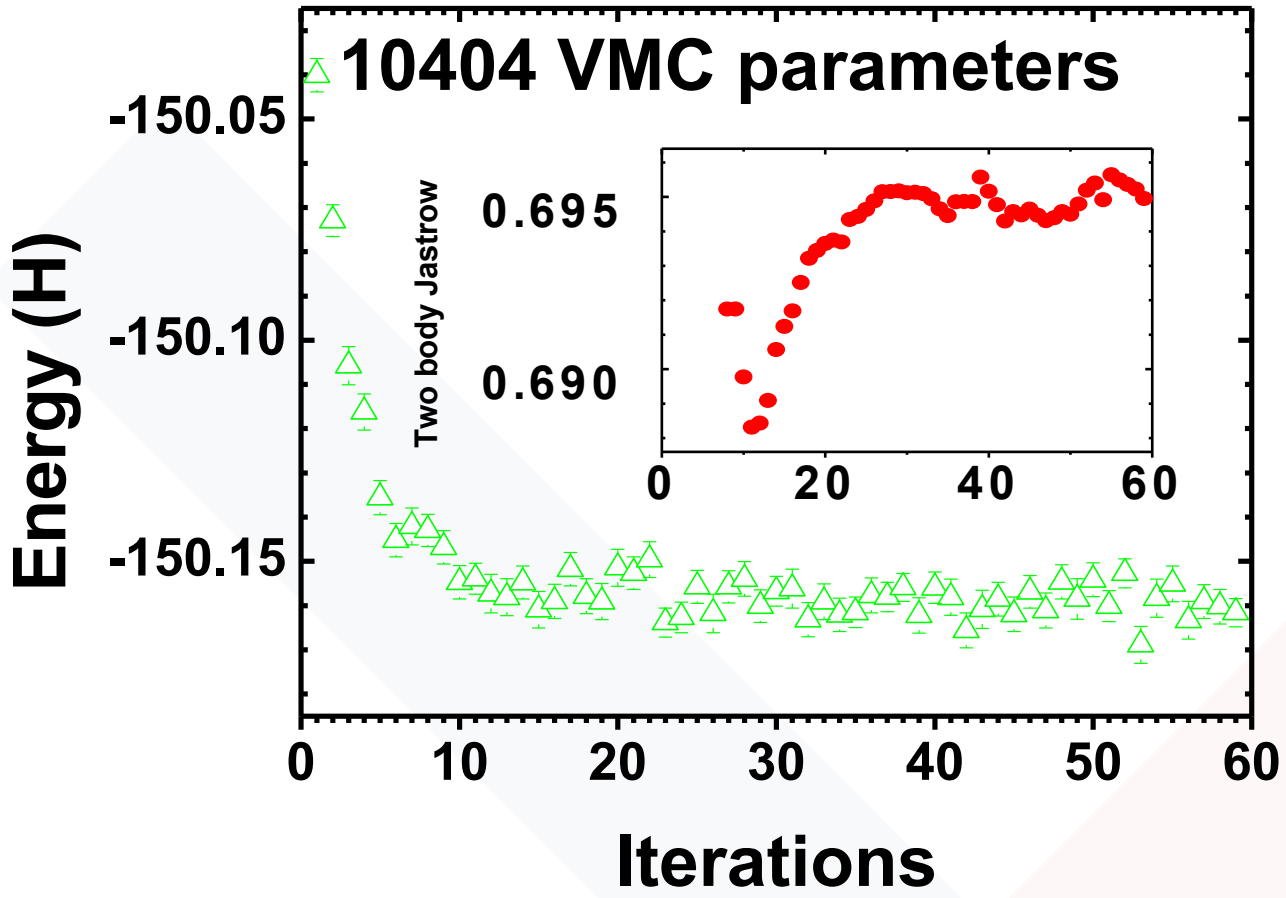
$$S_{i,i} \rightarrow S_{i,i}(1+\varepsilon)$$



# Removing most of the slowness of steepest



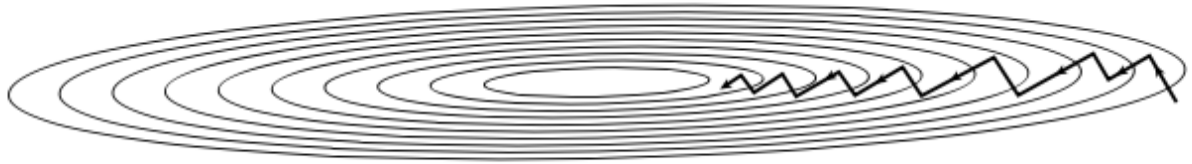
# A more difficult optimization




$$\varepsilon = 0.0001$$

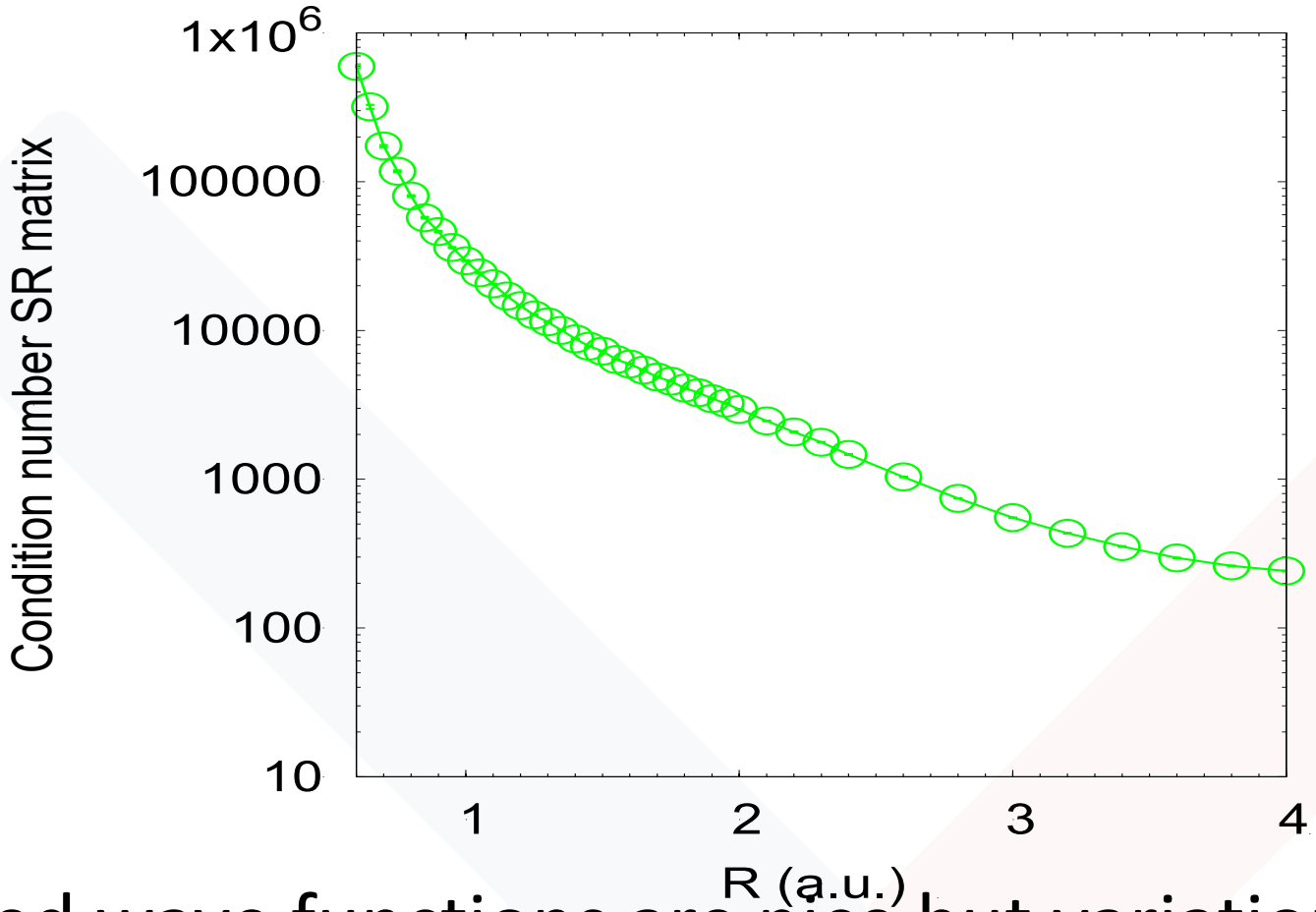
Benzene dimer: S.S., M. Casula and D. Rocca JCP **127**, 014105 (2007)

It is well known that the steepest is slow for Hessian matrices with large condition number r: Namely #Iterations >~ r



(a)

# Here is the condition matrix of SR in $H_2$




Correlated wave functions are nice but variational parameters entangled→ SR ill conditioned soon
That's why for R=0.8 we need 400000 iteration

# Concluding remarks

- Optimization of several parameters is nowdays possible within QMC thanks to SR or natural gradients
- Hessian method is in principle faster and a further improvement is possible by using partial information of it→ the linear method
- More popular optimized known in ML (e.g.  ADAMS) are much less accurate and useless for our accuracy target.

At variance of normal parameters α the atomic coordinates R appear also in the Hamiltonian $H_R$ and we need to compute R,α derivatives of:

$$E_{R,\alpha} = \frac{\langle \Psi_{\alpha,R} | H_R | \Psi_{\alpha,R} \rangle}{\langle \Psi_{\alpha,R} | \Psi_{\alpha,R} \rangle} = \sum_x \pi(x) e_L^{\alpha,R}(x)$$

where now: $$e_L^{\alpha,R}(x) = \frac{\langle x | H | \Psi_{\alpha,R} \rangle}{\langle x | \Psi_{\alpha,R} \rangle}$$

is also dependent upon both α and R

$$E_{R+dR,\alpha} = \frac{\sum_{x \in \text{samples}} |\frac{\Psi_{R+dR,\alpha}}{\Psi_{R,\alpha}}|^2(x) e_L^{R+dR,\alpha}(x)}{\sum_{x \in \text{samples}} |\frac{\Psi_{R+dR,\alpha}}{\Psi_{R,\alpha}}|^2(x)}$$

Thus by differentiating with respect to dR we obtain:

Go→

# Helmann-Feynmann and Pulay contributions

$$F_R = -\frac{dE_{R,\alpha}}{dR} = F_{\mathrm{Helmann-Feynman}} + F_{\mathrm{Pulay}}$$

$$F_{HF} = -\langle \partial_R e_L^{R,\alpha} \rangle = -\frac{1}{M}\sum_{i=1}^{M} \partial_R e_L^{R,\alpha}(x_i)$$

$$F_P = 2\mathrm{Cov}(O_R, e_L^{\alpha,R}) = -\frac{2}{M}\sum_{i=1}^{M} O_R(x_i) \times (e_L^{\alpha,R}(x_i) - E_{VMC})$$

$$O_R(x_i) = \partial_R \log|\Psi_{\alpha,R}(x_i)|$$

The variational parameters α(R) depend on R because we should expect correction o(dR) if we reoptimize the wf at R+dR

Thus we should add: $F = F_{HF} + F_P - \sum_{\alpha} (\partial_R \alpha)(\partial_\alpha E_{\alpha,R})$

But if we have done a good optimization and are at an energy minimum:

$\partial_\alpha E_{\alpha,R} = 0 \quad \forall \alpha$ because they are just the Euler's conditions of minimum energy (not necessarily the absolute one)

Assume $n = \Psi(x)$ defines a variable indicating the distance from the so called nodal surface n=0.

Then for n→0 $-\partial_R e_L^{\alpha,R}$ or $O_R e_L^{\alpha,R} \propto \frac{1}{n^2}$ $\pi(n) \propto n^2$

The mean of the above random variable is finite but their variance:

$$\int dn(n^2)(1/n^4) \to \infty$$

$$\pi(x) = \frac{|\Psi^{\alpha,R}(x)|^2}{R(x)} \qquad R(x) \propto n^2 \text{ for } n \to 0$$

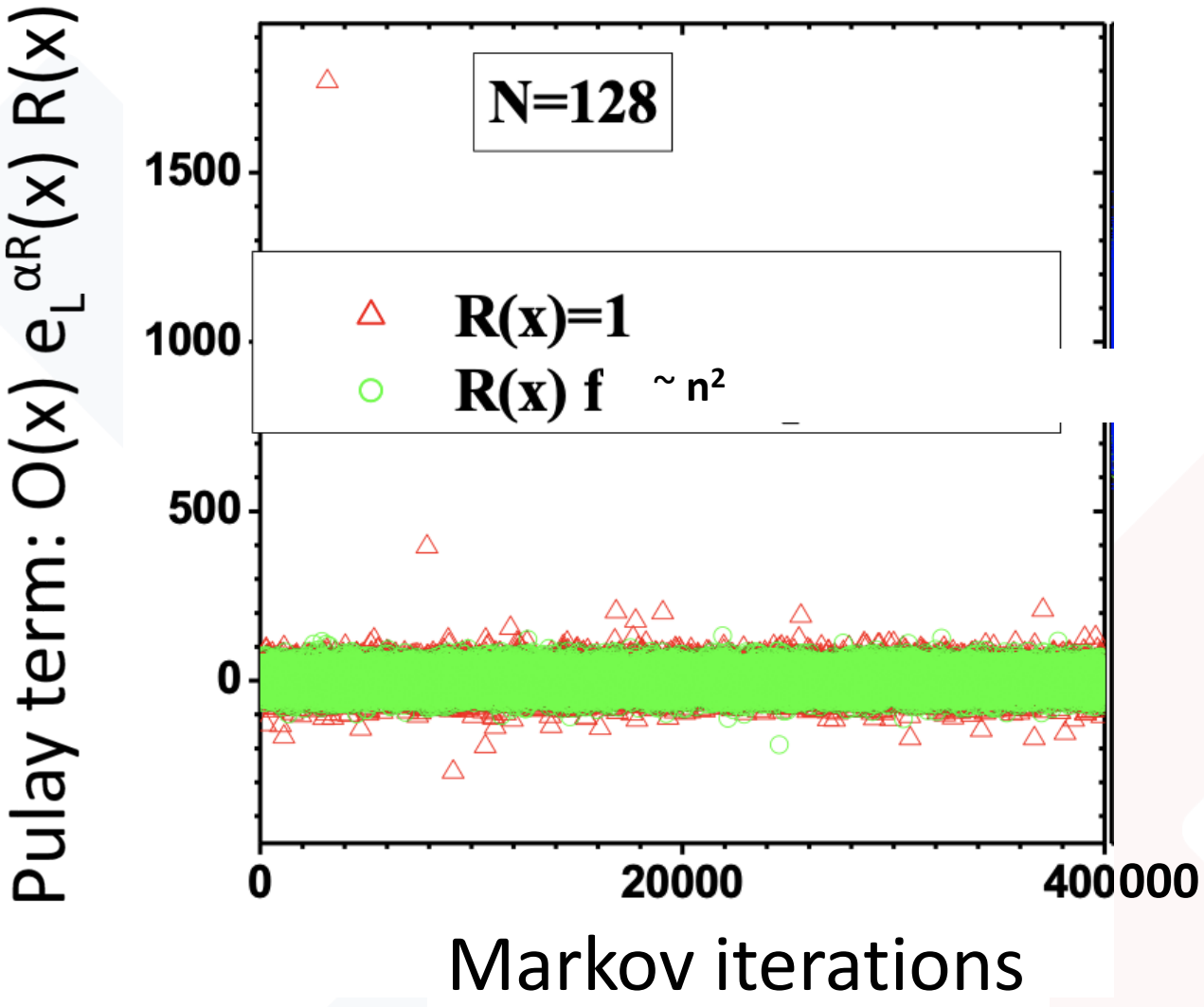
Then we have to compute average of finite random variables:

$$-\partial e_L^{\alpha,R} \times R(x) \text{ and } -O_R e_L^{R,\alpha} R(x) \to \text{FINITE for } n \to 0$$

Since n vanishes as our determinant/Pfaffian of a matrix M.
Each element of $M^{-1}$ blows up as 1/n and therefore the choice:

$$R(x) = \frac{1}{1 + \sum_{ij} |\epsilon M_{ij}^{-1}|^2} \propto n^2$$

Satisfies the requirements and make the variance FINITE




Sometimes you can definitively solve a boring problem in QMC

# But how to compute all these messy derivatives?

X,R and WF parameters ⟶ eloc,logpsi

TASK1: x,R → dist(I,i) I=1,2,…,#ion (nion) ,i=1,2,…,#el (nel) electron-ion distances.
TASK2: x,R,dist→ kel,vpseudolocal,prefactor,ivic,tmu (all electron coordinates in the pseudo mesh and LRDMC mesh ivic, with matrix element coefficients: prefactor for pseudo mesh, and tmu for LRDMC laplacian discretization)
TASK3: R → iond(I,J) (ion-ion distance)
TASK4: x,R Z's (exponents basis)→ winv(1:nelorb,0:indt+4,1:nel),winvj(nelorbj,0:indt+4,1:nel) (basis array)
TASK5: winv,winvj,jasmat,detmat,detmat_c,mu_c→ winvbar,winvjbar (Geminal Jastrow matrices contracted with basis)
TASK6: iond→ vpot (Classical Coulomb ion potential)
TASK7: vpseudolocal → vpot (add Ewald if PBC and pseudo contribution if present)
TASK8: winv,winvbar → logpsidetln, ainv (Inverse Geminal matrix for fast updates)
TASK9: x,R,winvj,winvjbar, Z's→ jastrowall_ee(1:nel,1:nel,0:indt+4),jastrowall_ei(1:nion,1:nel),tabpip (for fast updates Jastrow)
→ logpsi
TASK10: x,R,ainv,winvbar→ winvup,winvdo
TASK11: tabpip,winvup,winvdo→ eloc

The subroutine name is compute_eloc_logpsi and is badly written as many input are explicitly passed (huge call) and several others are passed via f90 modules. A refactoring of this routine could help for future developments.

TASK(INPUT→OUTPUT)

$$\mathrm{TASK_b}(\overline{\mathrm{OUTPUT}} \rightarrow \overline{\mathrm{INPUT}})$$

$$\overline{\mathrm{INPUT}}^i = \overline{\mathrm{INPUT}}^i + \frac{\partial \sum_k \overline{\mathrm{OUTPUT}}^k \mathrm{OUTPUT}_k}{\partial \mathrm{INPUT}_i}$$

# AAD of compute_eloc_logpsi, the new adjoint of the input

TASK11_b
TASK10_b
…

$$\bar{e}_L = 1 \quad \rightarrow \quad \begin{cases} \bar{\vec{r}}_i & = & \partial_{\vec{r}_i} e_L \\ \bar{\vec{R}}_i & = & \partial_{\vec{R}_i} e_L \end{cases}$$

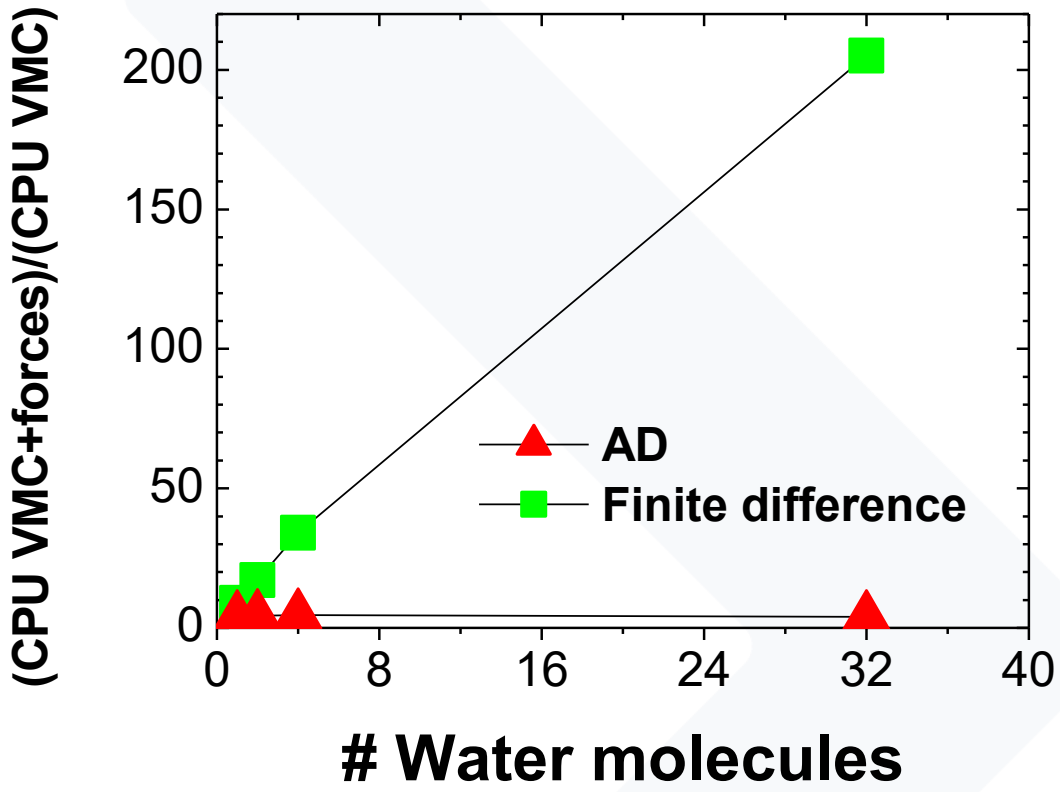
TASK1_b

$$\ln(\bar{\Psi}) = 1 \quad \rightarrow \quad \begin{cases} \bar{\vec{r}}_i & = & \partial_{\vec{r}_i} \ln(\Psi) \\ \bar{\vec{R}}_i & = & \partial_{\vec{R}_i} \ln(\Psi) \end{cases}$$

Thus we can compute all necessary for ion forces and/or optimization

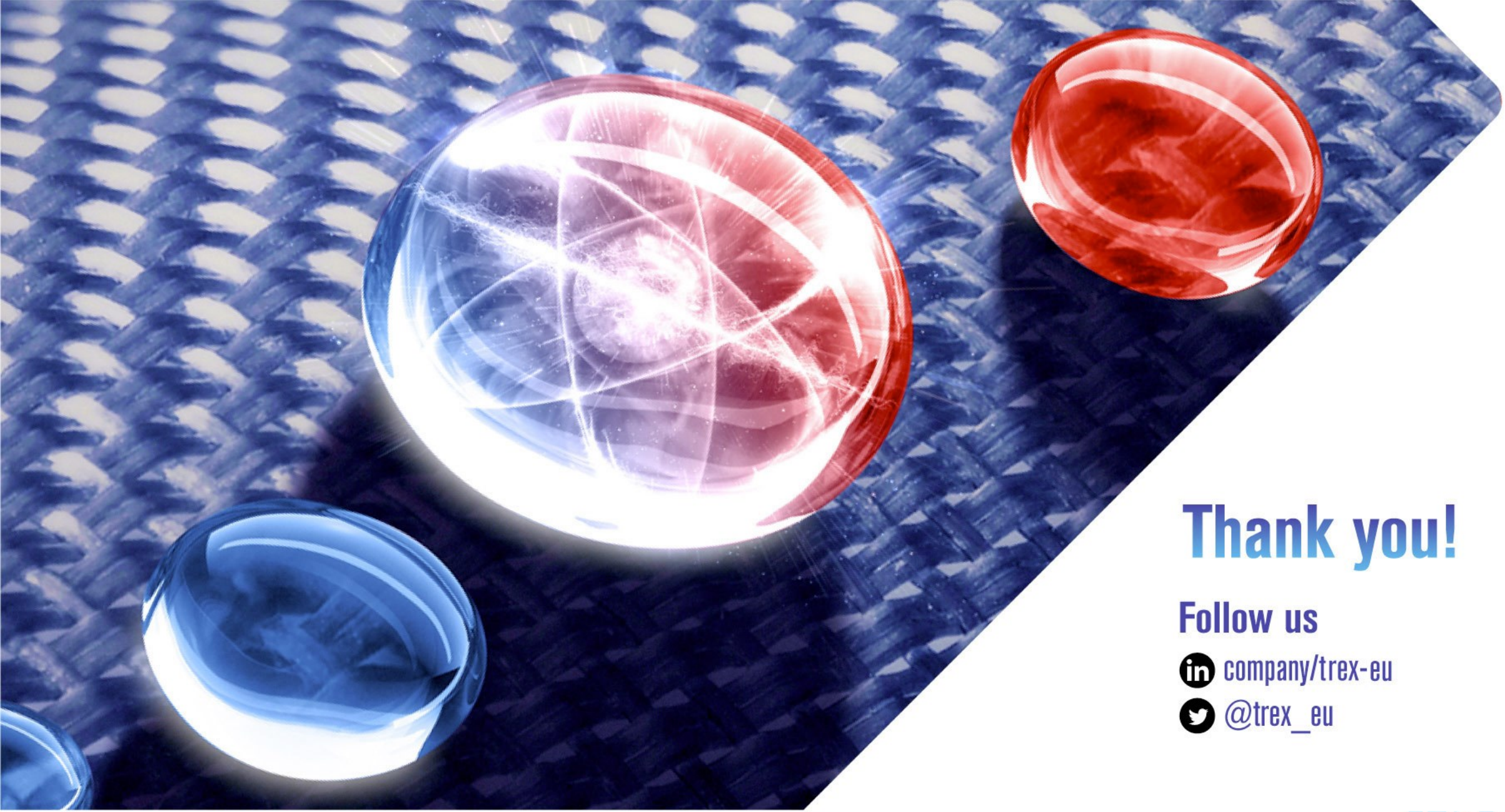
GPU implementation is trivial, each task BLAS3 offload and that's it.

Cpu time referenced to simple VMC (only energy)
for computing all 3M force components in water.




It is a theorem it should be at most
4 times more expensive
Use of pseudopotentials &
Jastrow → 11 instances

For DMC the overhead to
Compute all forces is ~0
but there is approximation



# 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