MRPT2 computation

Hybrid stochastic-deterministic approach

Yann Garniron, Anthony Scemama, Pierre-Francois Loos, Michel Caffarel

GMO 16/03/2017

Outline

Iterative selection of determinants

- start with n determinants

$$\Psi = \sum_{I}^{n} c_I |D_I\rangle$$

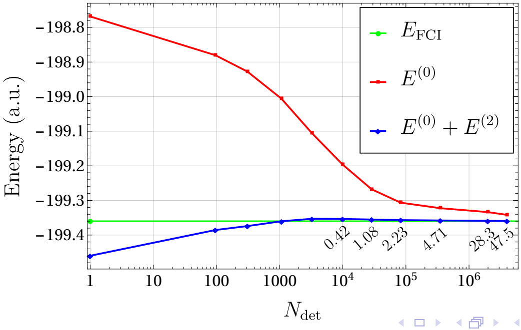
- Compute the perturbative contribution of external determinants

$$\mu_\alpha = \frac{\langle \Psi | H | \alpha \rangle^2}{\Delta E_\alpha}$$

- if $|\mu_\alpha| > t$, add α to Ψ
- diagonalize H
- ... repeat with Ψ now of size $N > n$

MRPT2 estimates full-CI energy

- $\Psi = \sum_I c_I |I\rangle$
- $E^{(2)} = \sum_\alpha \frac{|\langle\Psi|H|\alpha\rangle|^2}{\Delta E_\alpha}$
- ΔE_α depends on MR flavor. In our case, Epstein-Nesbet PT : $\Delta E_\alpha = E^{(0)} - \langle\alpha|H|\alpha\rangle$



Sensible computation of $E^{(2)}$

- rewrite $E^{(2)}$ as

$$\sum_{I,J} c_I c_J \frac{\langle I|H|\alpha\rangle\langle\alpha|H|J\rangle}{\Delta E_\alpha}$$

- Compare $|I\rangle$ and $|J\rangle$ to generate all $|\alpha\rangle$ interacting with both

MRPT2 is a by-product of CIPSI

- In both cases, all $\langle I|H|\alpha\rangle$ are computed
- MRPT2 : only a sum is required. It doesn't matter in what order they are computed
- CIPSI : we need the intermediate sums $\sum_I c_I \langle I|H|\alpha\rangle$ to compute

$$\mu_\alpha = \frac{\left(\sum_I c_I \langle I|H|\alpha\rangle\right)^2}{\Delta E_\alpha}$$

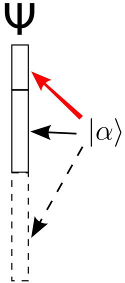
- MRPT2 can be computed as a by-product of CIPSI

$$E^{(2)} = \sum_\alpha \mu_\alpha$$

...but it isn't necessarily the best way to do it.

CIPSI allows for more approximation

- MRPT2 needs to account for a huge number of small contributions.
- CIPSI is only interested in the biggest μ_α. CPU time spent computing smaller ones is wasted.
  - Smaller c_I are ignored ; Ψ is truncated
  - α has to interact with at least one larger c_I

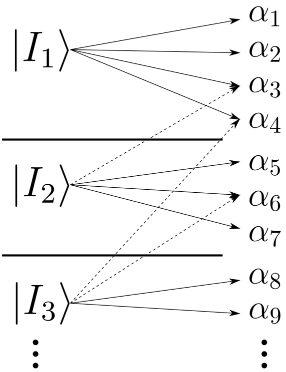


How we used to do

- CIPSI yield approximate MRPT2 for the PREVIOUS iteration.
- An extra iteration is required, with a larger number of determinants.
- The approximations need to be lowered
- ...expensive, hence the idea of making it stochastic!

α generation

- $|I\rangle$ generates all $|\alpha\rangle$ interacting with it.
- Those $|\alpha\rangle$ which have been previously generated are not generated again



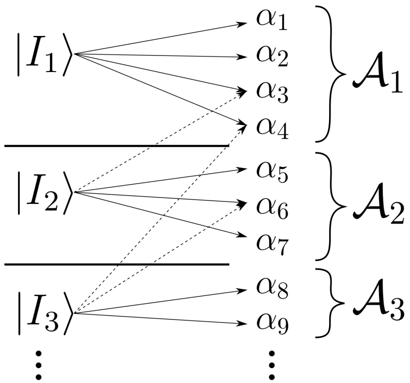
α are grouped in N_{det} batches $\mathcal{A}$

- The elementary contributions will be

$$e_I = \sum_{\alpha \in \mathcal{A}_I} \mu_\alpha$$

-

$$E^{(2)} = \sum_I e_I$$

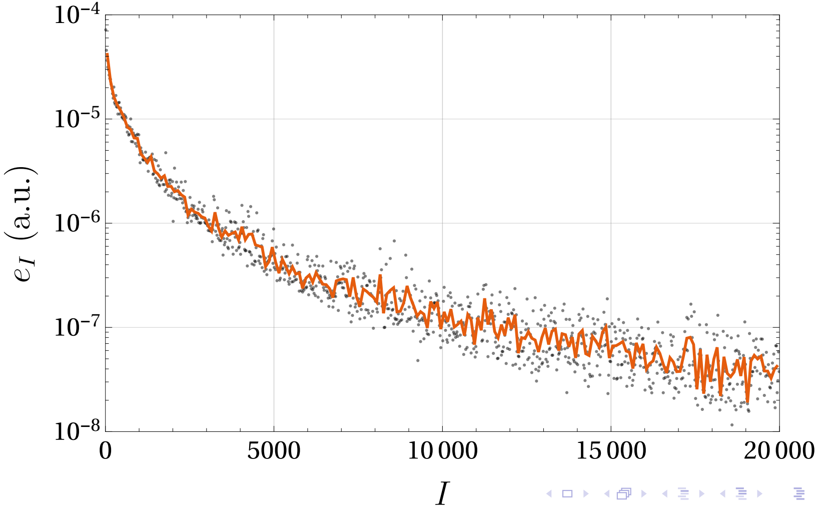


Monte-Carlo in a nutshell

- We want to compute $\sum_I e_I$ but it's too expensive
- We randomly draw some e_I and assume they have the same average value as the whole set
- Such a computation can be made much faster using an estimator p_I giving a probability to draw e_I
- p_I should be proportional to e_I as much as possible.

e_I decreases rapidly

F_2 with 1M det in cc-pVDZ



There are several reasons for the decrease of e_I

- ΔE_α increases
- The number of associated $|\alpha\rangle$ decreases, since more and more have already been generated
- Associated $|\alpha\rangle$ are by construction disconnected from previous determinants.

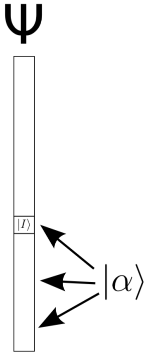
Estimator for e_I

- e_I is estimated by the norm of the sub-wavefunction it may connect to

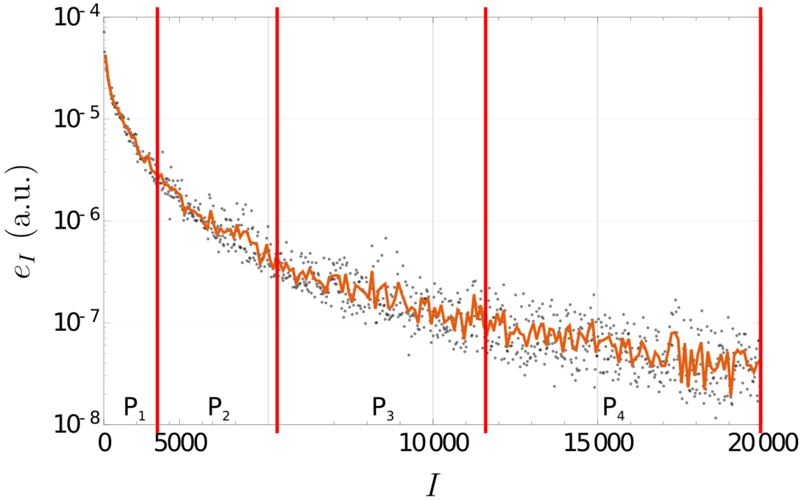
$$\sum_{J \geq I} c_J^2$$

- In practice we use the largest term c_I^2.
- Our sampling function is

$$p_I = c_I^2$$



Reducing variance by partitioning



Sampling sets of e_I

- p is divided in M equiprobable subspaces from P_1 to P_M
- Let J be a random set of M samples taken among all possible I, K a random tuple of M samples such that $K_i \in P_i$

$$var\left(\sum_{i=1}^{M} e_{J_i}\right) \geq var\left(\sum_{i=1}^{M} e_{K_i}\right)$$

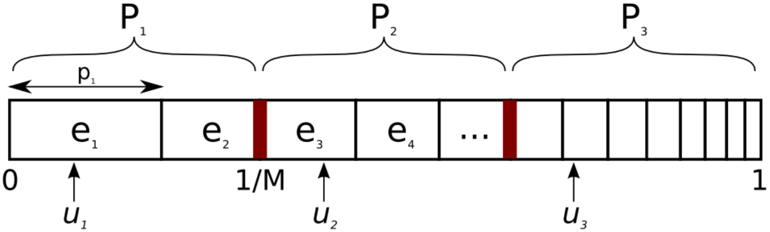
- We will actually be drawing/sampling sets of e_I

Sets are built as "combs" of e_I

- A set of e_I is associated with a random value u ranging from 0 to $\frac{1}{M}$
- The set is built by picking all e at "positions"

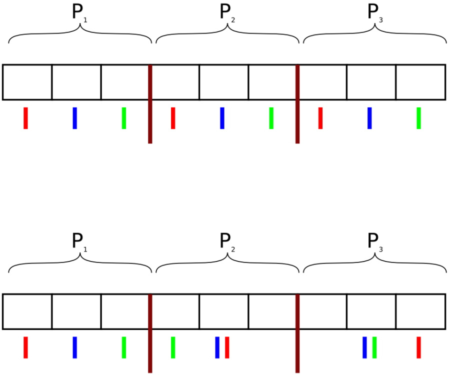
$$u_k = \frac{u + k - 1}{M}; k = 1, ..., M$$

- The sample value U associated with u is $e_1 + e_3 + e_7$

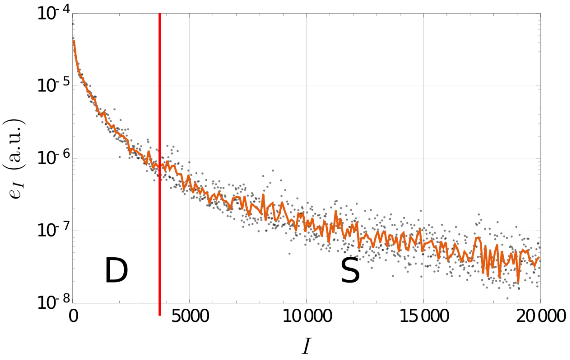


Sets are built as "combs" of e_I

- Combs furthermore reduces variance by correlating e_I

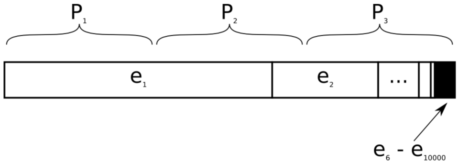


The head of the wavefunction is made deterministic



- $E^{(2)} = E_D^{(2)} + E_S^{(2)}$

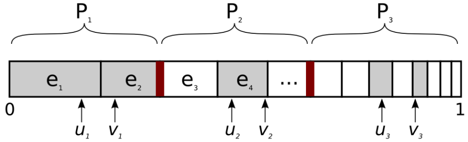
Initial deterministic part



- Usually, most determinants are crushed into a tiny probability
- The first determinants are moved to the deterministic part until a better balance is obtained

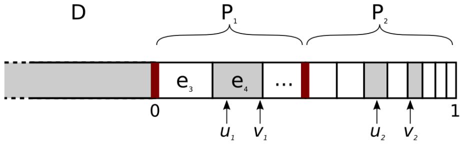
Increasing the deterministic part

- When a subspace P_i is fully computed, its contribution is known with no error.



Increasing the deterministic part

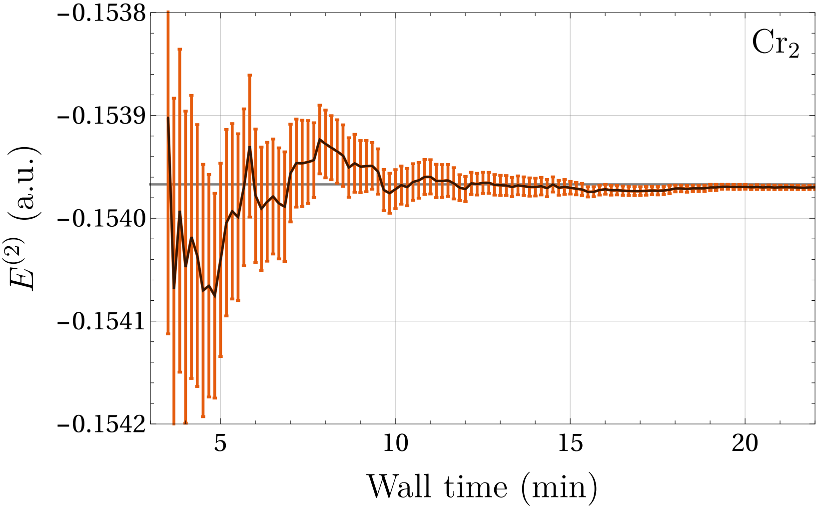
- e_I belonging to this subspace are removed form U and V the values associated with combs u and v, and added to $E_D^{(2)}$
- $\Delta E_D^{(2)} = e_1 + e_2$
- $\Delta U = -e_1$
- $\Delta V = -e_2$



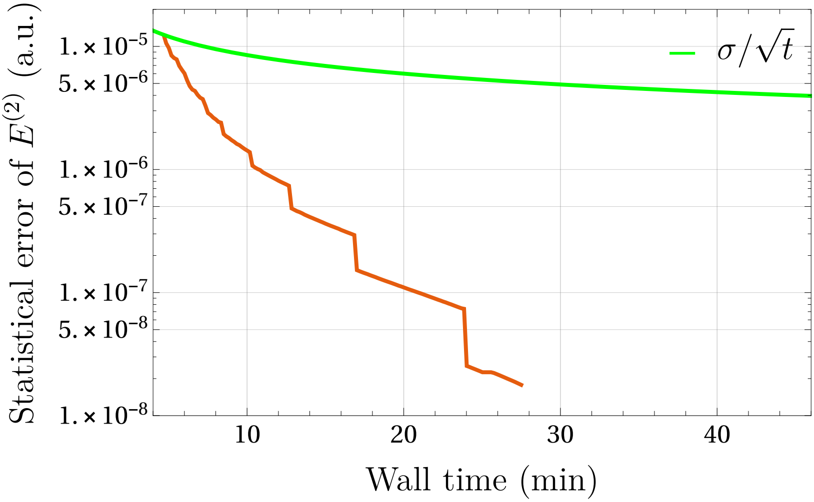
... until you get the exact value?

- e_I are stored so they are only computed once
- As the computation goes, fewer and fewer are left to be computed, so combs are drawn faster and faster
- At some point, all e_I are computed and the exact result is obtained with almost no extra computation time

Cr_2, 5M det, cc-pVDZ, (28e,76o)



Hybrid VS purely stochastic



Test runs with Cr_2

Basis	$E^{(2)}$	Wall time
cc-pVDZ		50 nodes (800 cores)
	$-0.153\,9(2)$	4 min
	$-0.153\,94(2)$	10 min
	$-0.153\,970(3)$	20 min
	$-0.153\,967\,0(2)$	30 min
	$-0.153\,967\,027$ (exact)	2 hr
cc-pVTZ		50 nodes (800 cores)
	$-0.222\,3(6)$	18 min
	$-0.222\,5(1)$	33 min
	$-0.222\,48(3)$	1 hr
	$-0.222\,513(4)$	2 hr
	—	$\sim$ 15 hr (estimated)
cc-pVQZ		250 nodes (4000 cores)
	$-0.225\,2(2)$	2 hr
	$-0.225\,24(5)$	3 hr
	$-0.225\,236(7)$	5 hr
	$-0.225\,236\,7(3)$	6 hr
	—	$\sim$ 24 hr (estimated)

Summary

- $E^{(2)}$ is computed with a small error bar, smaller than the accuracy of MRPT2 vs full-CI energy
- Can't be longer than the full deterministic computation
- 90% parallel efficiency