Markov chain Monte Carlo methods for Path Integral Thermodynamics and energy gaps of the anharmonic oscillator

M. Barbieri * **M. Dell'Anna** *

* *University of Pisa, Largo Pontecorvo 3, I-56127 Pisa, Italy*
(email: m.barbieri20@studenti.unipi.it, m.dellanna2@studenti.unipi.it)

Abstract: The aim of this report is to study the low temperature behavior of an harmonic oscillator with x^4 perturbation, focusing in particular on its thermodynamic properties and first few energy gaps. Single site, single cluster and multi-cluster update algorithms are presented and compared.

1. INTRODUCTION

The system of interest of this report is a single quantum harmonic oscillator, with an anharmonic x^4 perturbation. We focus on thermodynamical properties and energy gaps of such system in the so called Gibbs thermal state, which is a statistical mixture of Hamiltonian eigenstates so that the density matrix $\hat{\rho}$ corresponds to a Boltzmann distribution:

$$\begin{cases} \hat{H} = \frac{1}{2m}\hat{p}^2 + \frac{m\omega^2}{2}\hat{x}^2 + \frac{g}{4}\hat{x}^4, \\ \hat{\rho} = \frac{1}{Z(\beta)} \cdot \exp\left[-\beta \hat{H}\right], \end{cases} \tag{1}$$

where $Z(\beta)$ is the partition function.

The discussion is organized as follows:

- in Section 2 we briefly summarize the procedures, choices and conventions used for the simulations;
- in Section 3 we present the statistical and fitting methods used to analyze the numerical simulations;
- in Section 4 we show and comment on the obtained results.

All used codes are available at our repository on GitHub [1].

2. METHODS OF INVESTIGATION

2.1 Units

Throughout the document, we set the Boltzmann constant $k_{\mathrm{B}} = 1$. Also, energy is measured in units of the harmonic oscillator gap, therefore $\hbar\,\omega = 1$. Finally, length are expressed in units of harmonic oscillator characteristic length, hence $\frac{\hbar}{m\omega} = 1$

2.2 Path integral and discretization

In order to investigate the wanted properties, we rely on Monte Carlo simulation to obtain the values of correlators with the Path Integral technique. Standard arguments imply that given the system's (Euclidian) Hamiltonian density $\mathcal{H}$, the respective Euclidian action S_E, and an observable $\hat{O}$ depending on the position operator only, the following relations hold:

$$S_E[x(\tau)](\beta) = \int \mathcal{H}[x(\tau)]\,\mathrm{d}\tau, \tag{2}$$

and

$$\begin{aligned} \langle O \rangle &= \mathrm{tr}[O \cdot \rho] = \\ &= \frac{1}{Z(\beta)} \cdot \int\limits_{x(0)=x(\beta)} \mathcal{D}[x(\tau)]\; O(x(\tau)) \exp[-S_E[x(\tau)]], \end{aligned} \tag{3}$$

which is the path integral representation of an expected value. It is clear how the term $\frac{1}{Z(\beta)} \exp[-S_E]$ can be interpreted as a probability density function; we can discretize the integrals introducing a time step $a = \frac{\beta}{N}$, getting:

$$P(x_0, ..., x_{N-1} = x_0) = \frac{1}{Z(\beta)} \cdot \exp\left[-a \sum_{i=0}^{N-1} \mathcal{H}[x_i]\right], \tag{4}$$

which is to be sampled for our investigation.

To this end, we build three different Markov Chains, briefly described in Section 2.3 and Section 2.4; each one will produce, at each iteration, a new path $(x_0, ..., x_{N-1})$, which will be referred to as *state* of the Markov chain. We point out that the algorithms are of the Metropolis–Hastings type, and they are only different in how the trial state is chosen.

2.3 Single site update

With this first Markov chain, the initial and trial state only differ by one point of the discretized path. Given its index $i^\star$, and the current state $v = (x_1, ..., x_{i^\star}, ..., x_{N-1})$, the trial state becomes

$$w = (x_0, ..., x_{i^\star} + \delta, ..., x_{N-1}) \tag{5}$$

were the innovation δ must be extracted form a probability density function $g(\delta \,|\, v, i^\star)$. Finally the trial state can be accepted with probability

$$P_{\mathrm{acc}} = \min\left(1, \frac{g(-\delta \,|\, w, i^\star)}{g(\delta \,|\, v, i^\star)} \cdot \frac{P(w)}{P(v)}\right) \tag{6}$$

Calling D^2 the discrete Laplacian, we observe:

$$\begin{cases} \dfrac{P(w)}{P(v)} = \exp\Big[-(\alpha_v\delta - \beta_v)^2 + \beta_v^2 - agx_{i^\star}\delta^3 - \dfrac{ag}{4}\delta^4\Big], \\ \alpha_v = \sqrt{\dfrac{a}{2}} \cdot \sqrt{\dfrac{2}{a^2} + 1 + 3gx_{i^\star}^2}, \\ \beta_v = \frac{1}{2}\, a\, \alpha_v^{-1}\big[gx_{i^\star}^3 + x_{i^\star} - D^2 x_{i^\star}\big]. \end{cases} \tag{7}$$

We point out that the two coefficients α_v, β_v defined above actually depend on initial state and the site that could be updated, but we will only index them with the former for notation clarity.
At this point, the better $g(\delta \,|\, v, i^\star)/g(-\delta \,|\, w, i^\star)$ resembles $P(v)/P(w)$, the closer the acceptance probability will be to one.

We also take into account that we need to sample $g(\delta \,|\, v)$ for every simulation update, therefore a good choice of this distribution is the gaussian in (8), which returns a decent acceptance probability and is easy to sample via the Box–Müller algorithm:

$$g(\delta \,|\, v, i^\star) = \sqrt{\frac{2}{\pi}}\, \alpha_v \cdot \exp\Big[-(\alpha_v\delta + \beta)^2\Big]. \tag{8}$$

Significant features of this choice include that the drift term $-\beta_v/\alpha_v$ is mean–reverting and the farther the initial state will be from the mean (zero), the lower the variance $1/(2\alpha_v^2)$ will be.

This choice leads to an acceptance rate of the updates of $\geq 99.5\%$.

This algorithm is very parallelizable, so we implemented it with `CUDA`. At each update, we randomly extract the parity of the firsts sites to update. So we update all sites with that parity and later the sites with the opposite one, in order to avoid data–race conditions.

2.4 Cluster update

This algorithm is a generalization of the Wolff algorithm, not based on any symmetry of the system.

Instead of modifying one site per step, we build a cluster of adjacent sites, then change every site in it of the same quantity δ. Section 2.4.1 and Section 2.4.2 give an outline of the algorithm, then all probabilities and parameters are made explicit and discussed in detail in Section 2.4.3.

Cluster building
The basic idea is to start from a randomly selected site $i^\star$, then append neighbors to the cluster with probability $P_{\text{add}}\big(\Delta x_{\text{neigh}}\big)$, which only depends on the kinetic energy contribution $\Delta x_{\text{neigh}} = x_{\text{neigh}} - x_{i^\star}$, as we show in the pseudo-code snippet below:

```
cluster[0] = i*  first cluster's site
l = 1 cluster's length;    k = 0 counter
while k < l:
    for i first neighbor of cluster[k]
        if i ∉ cluster
            append i with probability P_add(x_i − x_cluster[k])
            l ← l + 1
    k ← k + 1
```

Innovation sampling
Given the cluster $\mathcal{C} = (m, ..., M)$, of size $N_{\mathcal{C}}$, we extract an innovation δ from a probability density function $g(\delta \,|\, v, \mathcal{C})$, so that the trial state will be

$$w = (x_0, ..., x_m + \delta, ..., x_M + \delta, ..., x_{N-1}). \tag{9}$$

We remark that the entire kinetic energy change from initial to final state are due x_m and x_M updates, and so that we can write the probability of building the cluster $\mathcal{C}$ starting from v as

$$\begin{aligned} P_{\text{build}}(\mathcal{C} \,|\, v) = P_{\text{in}}(\mathcal{C}, v) \cdot \\ \cdot [1 - P_{\text{add}}(\,|x_m - x_{m-1}|\,)] \cdot \\ \cdot [1 - P_{\text{add}}(\,|x_M - x_{M+1}|)], \end{aligned} \tag{10}$$

where P_{in} only depends on the relative distances inside the cluster and it's invariant under a global shift of the cluster (and so $P_{\text{in}}(\mathcal{C}, v) = P_{\text{in}}(\mathcal{C}, w)$).

Probability tuning
As before, we aim to maximize the acceptance probability

$$P_{\text{acc}} = \min\left[1, \frac{g(-\delta \,|\, w, \mathcal{C}) \cdot P_{\text{build}}(\mathcal{C} \,|\, w) \cdot P(w)}{g(\delta \,|\, v, \mathcal{C}) \cdot P_{\text{build}}(\mathcal{C} \,|\, v) \cdot P(v)}\right] \tag{11}$$

by choosing the appropriate $g(\delta \,|\, v, \mathcal{C})$ and $P_{\text{add}}\big(\Delta x_{\text{neigh}}\big)$. One can verify that an acceptable choice for these is:

$$P_{\text{add}}\big(\Delta x_{\text{neigh}}\big) = \max\left\{1 - \gamma \exp\left[+\frac{\Delta x_{\text{neigh}}^2}{2a}\right], 0\right\}, \tag{12}$$

for a suitable choice of γ. Moreover, for the g function, we can choose

$$\begin{cases} g(\delta \,|\, v, \mathcal{C}) = \sqrt{\dfrac{2}{\pi}}\alpha_v \exp\Big[-(\alpha_v\delta + \beta_v)^2\Big], \\ \alpha_v = \sqrt{\dfrac{a}{2}N_{\mathcal{C}}} \cdot \sqrt{1 + 3g\,\widehat{x^2}}, \\ \beta_v = \dfrac{a}{2} \cdot \dfrac{N_{\mathcal{C}}}{\alpha_v} \cdot \Big[\hat{x} + g\,\widehat{x^3}\Big], \end{cases} \tag{13}$$

where:
- the parameter γ is free to vary in the range $(0, 1)$, and controls the average size of clusters, in addition to setting a maximum Δx_{neigh} that can be accepted;
- $\hat{x}, \widehat{x^2}, \widehat{x^3}$ are meant as averages within the cluster.

In comparison to the single site update, we remark that now the drift term in the innovation pdf is independent of the kinetic energy variation, as that has been taken into account by P_{add}.

In our case, we chose γ in order to have the average cluster size of about $\sqrt{N}$, with N the total number of sites in the simulation. We can reach this goal observing that Δx is approximately distributed as

$$\propto e^{-\Delta x^2/2a}, \tag{14}$$

so, roughly, $\Delta x_{\text{typ}} \sim \sqrt{2a}$; it follows than that we approximately want

$$N \cdot \left(1 - \frac{\gamma}{e}\right) \sim \sqrt{N}, \tag{15}$$

that can be used for an estimate of γ.

We implemented this algorithm both with single cluster update, completely running on CPU and written in plain `C++`, and multi–cluster update with `CUDA`, GPU parallelized.

In the latter case, we first building the clusters as in the single cluster algorithm. In order to do that, on each site, a thread compute if the next site is in the same cluster of its. Then, in order to avoid data–race conditions, in the graph that has the cluster as vertex and the interaction as edges, we color this graph and we update one color at a time.

Our parameters leads to an update acceptance ratio of 5%— 30% for the multi-cluster update and 60%— 90% for the single-cluster update.

All GPU simulations run on a `NVIDIA GTX 1650`.

As far as the random number generator is concerned, we use PCG32 [2], which is high-quality and high-speed.

3. DATA ANALYSIS

Having algorithms that sample the path distribution, we now focus on how to find energy gaps and their uncertainties. Our estimate of energy gaps fully relies on solving the following generalized eigenvalue problem (GEVP):

$$C(t+\tau)v = \lambda C(t)v, \tag{16}$$

where $C(t)$ is the connected correlator matrix at lag t:

$$C_{ij}(t) = \langle O_i(t)O_j(0)\rangle - \langle O_i\rangle\langle O_j\rangle. \tag{17}$$

We only make use of observables that depend on the position operator, since the vectors $\hat{x}^n|0\rangle$ span all the Hilbert space of the states; given a state s of the path and an operator $O(x)$, our sample is:

$$\overline{O}_s = \frac{1}{N}\sum_{i=0}^{N-1} O(x_{s,i}), \tag{18}$$

whereas our estimator for it's mean value $\langle O\rangle$, given N_s the (independent) samples number, will be:

$$\hat{O} = \frac{1}{N_s}\sum_{n=1}^{N_s} \overline{O}_n. \tag{19}$$

In particular, for each update, we computed the following observables:

- the first four powers of x, averaged along the path: $O_i = x^i;\ i = 1, ..., 4$;
- the raw correlator between those powers $O_i(t)O_j(0)$.

In Section 3.1 and Section 3.3 we briefly describe how we compute statistical uncertainties, in Section 3.2 how we use the GEVP (16) to find the first four energy gaps, and finally in Section 3.4 we explain how to reach an arbitrary target precision.

3.1 Statistical errors on correlators

Firstly, we remark that the correlators matrix $C_{ij}(t)$ is a secondary observable, for it depends on the primary observables $\langle O_i\rangle$, $\langle O_j\rangle$ and $\langle O_i(t)O_j(0)\rangle$. In order to compute the appropriate uncertainties, we rely on standard first order error propagation, using the following procedure:

1. starting from raw samples of primary observables, we block data to eliminate correlations due to the Markov Chain. Note that the number of samples reduces to N_s^{eff};
2. we compute sample covariance between the primary observables;
3. we use the first order error propagation formula (using Einstein notation):

$$\operatorname{cov}\big[f_i, f_j\big] = \frac{\partial f_i}{\partial x_l}\cdot\frac{\partial f_j}{\partial x_m}\cdot\operatorname{cov}[x_l, x_m]. \tag{20}$$

3.2 Finding energy gaps

With regard to (17), in the low temperature limit[1], the correlator takes the form:

$$\tilde{C}_{ij}(t) = \sum_{n=1}^{+\infty}\langle 0|O_i|n\rangle\big\langle n\big|O_j\big|0\big\rangle e^{-t(E_n-E_0)}. \tag{21}$$

Then in the large t limit (see below for clarification), we neglect contribution over the fourth gap, and call $\tilde{C}'_{ij}(t)$ the truncated correlator.

The solution to (16), with the $\tilde{C}'$ instead of $\tilde{C}$, is *exactly*:

$$\tilde{\lambda}'_n(\tau) = \exp[-\tau(E_n - E_0)], \tag{22}$$

which can be inverted to find gaps. These approximations introduce the following systematic errors[2]:

- finite β introduces an error on each entry of the correlators' matrix:

$$C_{ij}(t) \approx \tilde{C}_{ij} + O\big(e^{-\beta(E_1-E_0)}\big); \tag{23}$$

- finite t introduces an error on the correlators' matrix entries:

$$\tilde{C}_{ij}(t) \approx \tilde{C}'_{ij}(t)\Big[1 + O\Big(e^{-t(E_p-E_{p-2})}\Big)\Big], \tag{24}$$

  where we call E_p is the first not sampled energy level that has the same parity[3] as n;
- due to the presence of further states in the sampled C, the running of λ_n with τ is not exactly exponential. This induces a correction in the computation of the eigenvalues whose dominant component scales with

$$e^{-\tau\cdot(E_p-E_n)}. \tag{25}$$

3.3 Statistical uncertainties on energy gaps

In order to estimate statistical uncertainties on the GEVP solutions, we use Eq. (20), computing derivatives via the Hellmann–Feynman theorem. Let λ, v solve Eq. (16), and let $C_{ijk} = C_{ij}(t + k\tau)$ with $k = 0, 1$. Then the theorem states that:

$$\frac{\partial\lambda}{\partial C_{ijk}} = \frac{\Big\langle v\Big|\frac{\partial}{\partial C_{ijk}}\big[C_{ij0} - \lambda C_{ij1}\big]\Big|v\Big\rangle}{\langle v|C_{ij1}|v\rangle}. \tag{26}$$

[1] $\beta \to +\infty$ quantities will be indicated with a tilde.

[2] As discussed below, for the Hellmann–Feynman theorem, the error of the eigenvalue is proportional to the error of the matrices entries, so we will interchangeably use these notions.

[3] The symmetry $x \to -x$ induces a selection rule that make negligible contributions to the correlators of negative parity.

3.4 Parameter tuning

We set η and η_{stat} the target relative errors respectively due to systematic and statistical errors. We are particularly interested in assuring them to be reached on our fourth gap estimate, which presents the smallest signal and is the most sensitive to systematic errors by means of Eqq. (25) and (22). Firstly, we ask that both errors in Eqq. (25) and (24) are less than η and that the finite β in Eq. (23) is negligible ($\lesssim \eta/10$):

$$\begin{cases} \tau \sim t \gtrsim \dfrac{\ln(\eta)}{E_4 - E_6}, \\ \beta \gtrsim \dfrac{\ln(\eta/10)}{E_0 - E_1}. \end{cases} \tag{27}$$

Secondly, taking into account that statistical errors on λ are proportional to the ones on C, we ask the fourth gap signal to be greater than the noise, getting:

$$t \lesssim \frac{\ln(\eta_{\text{stat}})}{E_1 - E_4}. \tag{28}$$

An important remark is that we're asking:

$$\eta_{\text{stat}} \lesssim e^{-t(E_4 - E_1)} < e^{-t(E_6 - E_4)} \lesssim \eta, \tag{29}$$

hence the statistical error will be negligible with respect to the systematic one.

3.5 Discretization effects

The discreteness of the path must be taken into account as well. For each value of the coupling parameter, we perform multiple simulations varying the time step parameter a, and use a quadratic fit on the energy gaps to estimate their value in the continuum limit as shown in . For this fit, only statistical uncertainties are taken into account.

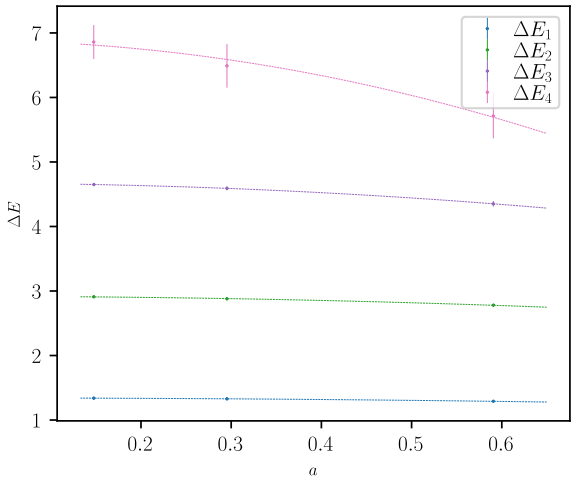

Fig. 1. Quadratic fit of the energy gaps for a simulation performed with single site update algorithm and coupling parameter $g = 7.84$

4. RESULTS

We finally present the result of our simulations. We used a relative error of 30% for the multi-cluster update algorithm, and 1% for the single site and single cluster update. As a comparison, we computed the gaps with a first order Perturbative–Variational approach, well explained in [3].

In conclusion, we see a good agreement between simulations and the Perturbative–Variational estimate of the energy gaps.

We are confident that great performance improvements can be obtained using a more suitable GPU (or higher dimensionality simulations) for the multi–cluster algorithm. However, single site update seems to stay the best option for its cleanness and overall performance.

ACKNOWLEDGEMENTS

https://github.com/mbar02/nummet-public/raw/refs/heads/master/report/Module%201/pics/2mhh3d0hx1gbl.png

REFERENCES

1. Our GitHub repository. https://github.com/mbar02/nummet-public
2. O'Neill ME (2014) PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation. Claremont, CA
3. Konishi K, Paffuti G (2015) Meccanica Quantistica: Applicazioni. Pisa University Press
4. ALPHA, Blossier B, Morte MD, et al (2009) On the generalized eigenvalue method for energies and matrix elements in lattice field theory. Journal of High Energy Physics 2009:94. https://doi.org/10.1088/1126-6708/2009/04/094
5. M. L, Wolff U (1990) How to calculate the elastic scattering matrix in two-dimensional quantum field theories by numerical simulation. Nuclear Physics B 339:222–252. https://doi.org/https://doi.org/10.1016/0550-3213(90)90540-T

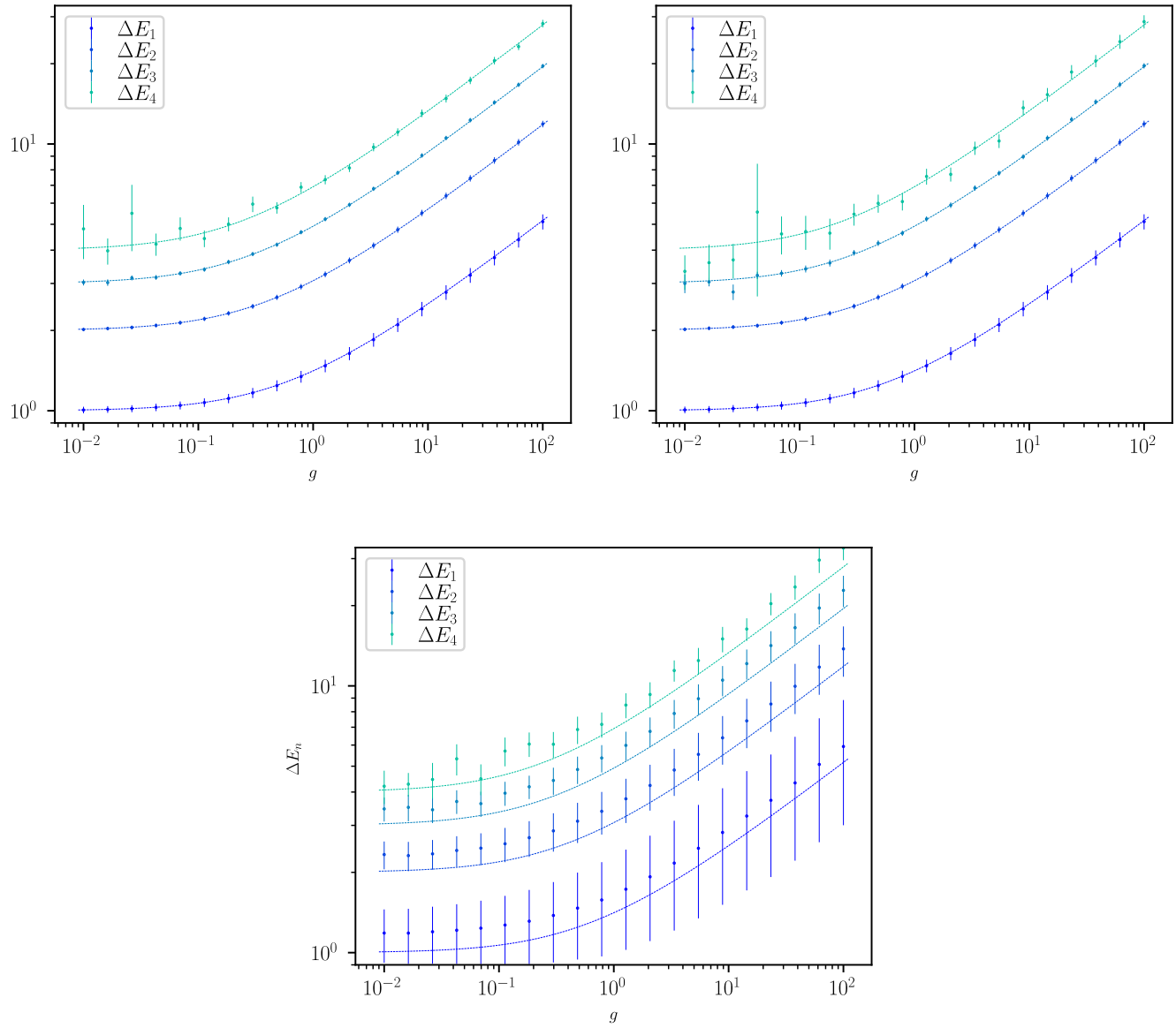

Fig. 2. Energy gaps varying the coupling parameter g, computed with all three different algorithms: single site (top left), single cluster (top right), multi-cluster (bottom). The dotted lines are the gaps computed with the Perturbative-Variational method.

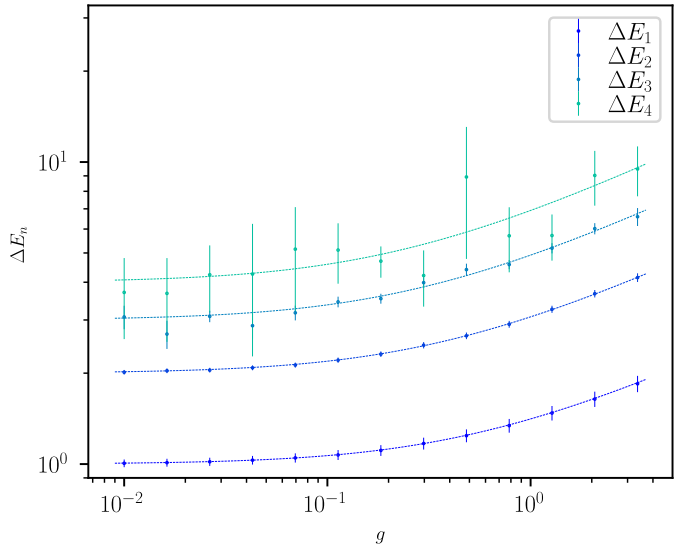

Fig. 3. Energy gaps varying the coupling parameter g, using Wolff algorithm after retuning γ.

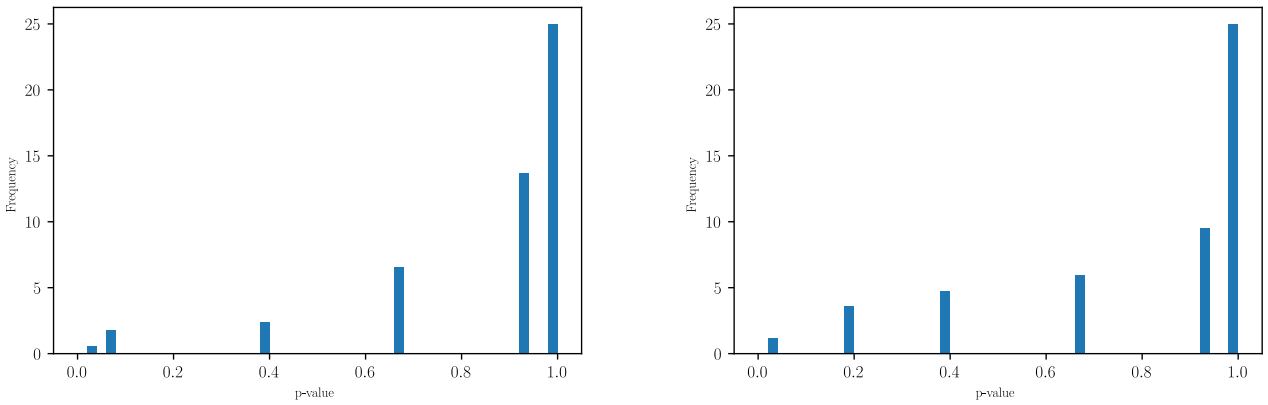

Fig. 4. Frequency of p-values from Kolmogorov-Smirnov test, comparing sample distribution (for all observables) of single cluster (left) and multi-cluster (right) to metropolis.