Supplementary Information for
Path integral Monte Carlo method for the quantum anharmonic oscillator

Shikhar Mittal

The Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom

Marise J. E. Westbroek

The Blackett Laboratory, Imperial College London,
London SW7 2AZ, United Kingdom and
Department of Earth Science and Engineering,
Imperial College London, London SW7 2BP, United Kingdom

Peter R. King

Department of Earth Science and Engineering,
Imperial College London, London SW7 2BP, United Kingdom

Dimitri D. Vvedensky

The Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom

Supplementary information for the article "Path integral Monte Carlo method for the quantum anharmonic oscillator", which includes (i) a derivation of the correlation functions used, (ii) a discussion of the Markov chain Monte Carlo method, and (iii) a summary of spline fits.

I. THE PARTITION FUNCTION

The partition function is

$$Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right) . \tag{1}$$

Inserting the completeness relation for the orthonormal eigenkets of $\hat{H}$, $1 = \sum_{n=0}^{\infty} |n\rangle\langle n|$, into the partition function yields

$$\begin{aligned} Z &= \mathrm{Tr}\left[e^{-\beta\hat{H}}\left(\sum_{n=0}^{\infty}|n\rangle\langle n|\right)\right] = \mathrm{Tr}\left(\sum_{n=0}^{\infty}|n\rangle e^{-\beta E_n}\langle n|\right) \\ &= \sum_{n=0}^{\infty} e^{-\beta E_n}\langle n|n\rangle = \sum_{n=0}^{\infty} e^{-\beta E_n} , \end{aligned} \tag{2}$$

which is the more familiar form of the partition function.

II. CORRELATION FUNCTIONS

Consider the $2n$-point correlation function:

$$G_{2n}(\tau) = \langle \hat{x}^{\,n}(\tau)\hat{x}^{\,n}(0)\rangle - \langle \hat{x}^{\,n}(\tau)\rangle\langle \hat{x}^{\,n}(0)\rangle . \tag{3}$$

The first term is defined as

$$\begin{aligned} \langle \hat{x}^{\,n}(\tau)\hat{x}^{\,n}(0)\rangle &= \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}\hat{x}^{\,n}(\tau)\hat{x}^{\,n}(0)\right] \\ &= \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}e^{\hat{H}\tau/\hbar}\hat{x}^{\,n}(0)e^{-\hat{H}\tau/\hbar}\hat{x}^{\,n}(0)\right] . \end{aligned} \tag{4}$$

Using the completeness relation for the orthonormal eigenkets of $\hat{H}$ twice in the right-hand side,

$$\begin{aligned} \langle \hat{x}^{\,n}(\tau)\hat{x}^{\,n}(0)\rangle &= \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}e^{\hat{H}\tau/\hbar}\left(\sum_{p=0}^{\infty}|p\rangle\langle p|\right)\hat{x}^{\,n}(0)e^{-\hat{H}\tau/\hbar}\left(\sum_{q=0}^{\infty}|q\rangle\langle q|\right)\hat{x}^{\,n}(0)\right] \\ &= \frac{1}{Z}\mathrm{Tr}\left[\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}|p\rangle e^{-\beta E_p}e^{E_p\tau/\hbar}\langle p|\hat{x}^{\,n}(0)|q\rangle e^{-E_q\tau/\hbar}\langle q|\hat{x}^{\,n}(0)\right] \\ &= \frac{1}{Z}\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}e^{-\beta E_p}e^{E_p\tau/\hbar}\langle p|\hat{x}^{\,n}(0)|q\rangle e^{-E_q\tau/\hbar}\langle q|\hat{x}^{\,n}(0)|p\rangle \\ &= \frac{1}{Z}\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}e^{-\beta E_p}e^{-(E_q-E_p)\tau/\hbar}\left|\langle p|\hat{x}^{n}(0)|q\rangle\right|^2 . \end{aligned} \tag{5}$$

We also have

$$\langle \hat{x}^{\,n}(\tau)\rangle = \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}\hat{x}^{\,n}(\tau)\right] = \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}e^{\hat{H}\tau/\hbar}\hat{x}^{\,n}(0)e^{-\hat{H}\tau/\hbar}\right] . \tag{6}$$

Again using the completeness relation twice in the right-hand side,

$$\begin{aligned}
\langle \hat{x}^{\,n}(\tau)\rangle &= \frac{1}{Z}\mathrm{Tr}\left[e^{-\beta\hat{H}}e^{\hat{H}\tau/\hbar}\left(\sum_{p=0}^{\infty}|p\rangle\langle p|\right)\hat{x}^{\,n}(0)e^{-\hat{H}\tau/\hbar}\left(\sum_{q=0}^{\infty}|q\rangle\langle q|\right)\right] \\
&= \frac{1}{Z}\mathrm{Tr}\left[\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}|p\rangle\, e^{-\beta E_p}e^{E_p\tau/\hbar}\langle p|\,\hat{x}^{\,n}(0)\,|q\rangle\, e^{-E_q\tau/\hbar}\langle q|\right] \\
&= \frac{1}{Z}\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}e^{-\beta E_p}e^{E_p\tau/\hbar}\langle p|\,\hat{x}^{\,n}(0)\,|q\rangle\, e^{-E_q\tau/\hbar}\langle q|p\rangle \\
&= \frac{1}{Z}\sum_{p=0}^{\infty}e^{-\beta E_p}\langle p|\,\hat{x}^{\,n}(0)\,|p\rangle\ .
\end{aligned} \tag{7}$$

Since the expression for $\langle \hat{x}^{\,n}(\tau)\rangle$ does not depend on τ (which is expected)

$$\langle \hat{x}^{n}(0)\rangle = \frac{1}{Z}\sum_{p=0}^{\infty}e^{-\beta E_p}\langle p|\,\hat{x}^{\,n}(0)\,|p\rangle\ , \tag{8}$$

which is the same as (7). Thus, by combining (3), (5), (7) and (8), we obtain

$$\begin{aligned}
G_{2n}(\tau) = \frac{1}{Z}\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}e^{-\beta E_p}e^{-(E_q-E_p)\tau/\hbar}\left|\langle p|\,\hat{x}^{n}(0)\,|q\rangle\right|^2\ldots \\
-\left[\frac{1}{Z}\sum_{p=0}^{\infty}e^{-\beta E_p}\langle p|\,\hat{x}^{\,n}(0)\,|p\rangle\right]\left[\frac{1}{Z}\sum_{p=0}^{\infty}e^{-\beta E_p}\langle p|\,\hat{x}^{n}(0)\,|p\rangle\right] .
\end{aligned} \tag{9}$$

To take the low-temperature limit of G_{2n}, we first have that[1]

$$Z = \sum_{n=0}^{\infty}e^{-\beta E_n}\,. \tag{10}$$

Then, for the terms in (9), the limit $\beta\to\infty$ yields

$$\begin{aligned}
\lim_{\beta\to\infty}\left(\frac{1}{Z}\sum_{p=0}^{\infty}e^{-\beta E_p}\langle p|\,\hat{x}^{n}(0)\,|p\rangle\right) &= \lim_{\beta\to\infty}\left(\frac{e^{-\beta E_0}\langle 0|\,\hat{x}^{n}(0)\,|0\rangle + e^{-\beta E_1}\langle 1|\,\hat{x}^{n}(0)\,|1\rangle+\cdots}{e^{-\beta E_0}+e^{-\beta E_1}+\cdots}\right) \\
&= \lim_{\beta\to\infty}\left(\frac{\langle 0|\,\hat{x}^{n}(0)\,|0\rangle + e^{-\beta(E_1-E_0)}\langle 1|\,\hat{x}^{n}(0)\,|1\rangle+\cdots}{1+e^{-\beta(E_1-E_0)}+\cdots}\right) \\
&= \langle 0|\,\hat{x}^{n}(0)\,|0\rangle\ .
\end{aligned} \tag{11}$$

Similarly,

$$\begin{aligned}
&\lim_{\beta\to\infty}\left(\frac{1}{Z}\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}e^{-\beta E_p}e^{-(E_q-E_p)\tau/\hbar}\left|\langle p|\,\hat{x}^n(0)\,|q\rangle\right|^2\right)\\
&=\lim_{\beta\to\infty}\left(\frac{\sum_{q=0}^{\infty}e^{-\beta E_0}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^n(0)\,|q\rangle\right|^2+e^{-\beta E_1}e^{-(E_q-E_1)\tau/\hbar}\left|\langle 1|\,\hat{x}^n(0)\,|q\rangle\right|^2+\cdots}{e^{-\beta E_0}+e^{-\beta E_1}+\cdots}\right)\\
&=\lim_{\beta\to\infty}\left(\frac{\sum_{q=0}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^n(0)\,|q\rangle\right|^2+e^{-\beta(E_1-E_0)}e^{-(E_q-E_1)\tau/\hbar}\left|\langle 1|\,\hat{x}^n(0)\,|q\rangle\right|^2+\cdots}{1+e^{-\beta(E_1-E_0)}+\cdots}\right)\\
&=\sum_{q=0}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^n(0)\,|q\rangle\right|^2. \qquad (12)
\end{aligned}$$

Hence,

$$G_{2n}(\tau)=\sum_{q=0}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^n(0)\,|q\rangle\right|^2-\left|\langle 0|\,\hat{x}^n(0)\,|0\rangle\right|^2. \qquad (13)$$

For $n=1$,

$$\begin{aligned}
G_2(\tau)&=\sum_{q=0}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}(0)\,|q\rangle\right|^2-\left|\langle 0|\,\hat{x}(0)\,|0\rangle\right|^2\\
&=\sum_{q=1}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}(0)\,|q\rangle\right|^2. \qquad (14)
\end{aligned}$$

For $n=2$,

$$\begin{aligned}
G_4(\tau)&=\sum_{q=0}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^2(0)\,|q\rangle\right|^2-\left|\langle 0|\,\hat{x}^2(0)\,|0\rangle\right|^2\\
&=\sum_{q=2}^{\infty}e^{-(E_q-E_0)\tau/\hbar}\left|\langle 0|\,\hat{x}^2(0)\,|q\rangle\right|^2. \qquad (15)
\end{aligned}$$

Consider the matrix element $\langle 0|\hat{x}^2(0)|q\rangle$. For the harmonic oscillator, the ground state has even parity, the first excited state has odd parity, and parity alternates between even and odd for all higher lying states. We expect the same pattern for the anharmonic oscillator. Therefore, as the operator $\hat{x}^2$ and the ground-state wave function are both even functions of x, the matrix element $\langle 0|\hat{x}^2(0)|q\rangle$ vanishes if the wave function of the qth excited state is an odd function of x. In particular $\langle 0|\,\hat{x}^2(0)\,|1\rangle=0$. Hence, the sum in (15) begins with $q=2$.

III. MONTE CARLO MARKOV CHAINS

A. Background

The sum over paths in the evaluation of the path integral is carried out with the Markov chain Monte Carlo (MCMC) method. This method was developed by Metropolis *et al.*,[2] who were studying the equilibrium properties of interacting particle systems. Instead of simulating the actual dynamical relaxation of such a system to equilibrium, the authors made the key observation that, for the calculation of equilibrium properties, a Markov chain that attains the same equilibrium distribution by any sequence of intermediate states is sufficient. Hastings[3] viewed the Metropolis method mainly as a way of sampling high-dimensional probability distributions, which is, in fact, the modern use of this algorithm. Hastings' article was written in a statistical style that abstracted the Metropolis method into a transition operator on a Markov chain whose target distribution is the invariant distribution of the chain, rather than the equilibrium distribution of Metropolis method. Simulations that follow this approach are said to use the *Metropolis–Hastings* algorithm.

B. Basic Concepts and Notation

A Markov chain is a sequence of events in which a given position in the chain depends only on the immediately preceding position, rather than on the history of the chain. A Markov chain is defined by three attributes:

1. A *state space* which defines the values that can be taken by the chain.

2. A *transition operator* that defines the probability of one state progressing to another state.

3. An *initial condition* that specifies the state from which the progression of the chain begins.

Our goal is to calculate the expectation value $\langle\hat{\mathcal{O}}\rangle$ of some operator $\mathcal{O}$ over a finite time interval,

$$\langle\hat{\mathcal{O}}\rangle = \sum_{k=1}^{N} \mathcal{O}(\boldsymbol{x}_k) P_{\mathrm{eq}}(\boldsymbol{x}_k)\,, \tag{16}$$

where $\boldsymbol{x}_k = \left(x_1^{(k)}, x_2^{(k)}, \ldots, x_N^{(k)}\right)$ is a configuration of the system, which also represents a path, and $P_{\rm eq}(\boldsymbol{x}_k)$ is the probability distribution in the canonical ensemble. The MCMC solution to this problem to the calculation of (16) is to construct a Markov chain on the state space $\boldsymbol{X}$ that has $P_{\rm eq}(\boldsymbol{x})$ as the stationary distribution. In other words, given transition probabilities $W(\boldsymbol{x}, \boldsymbol{x}')$ for chains $\boldsymbol{x}$ and $\boldsymbol{x}'$ in state space, we have

$$\int P_{\rm eq}(\boldsymbol{x}) W(\boldsymbol{x}, \boldsymbol{x}')\, d\boldsymbol{x} = P_{\rm eq}(\boldsymbol{x}')\,. \tag{17}$$

The discretized path integral representation of the canonical partition function in (10),

$$\begin{aligned} Z &= \int \prod_{k=1}^{N_\tau} dx_k \left(\frac{m}{2\pi\hbar\,\delta\tau}\right) \exp\left\{-\frac{\delta\tau}{\hbar}\sum_{i=1}^{N_\tau}\left[\frac{m}{2}\left(\frac{x_{i+1}-x_i}{\delta\tau}\right)^2 + V(x_i)\right]\right\} \\ &\equiv \int [D\boldsymbol{x}(\tau)] e^{-S(\boldsymbol{x})/\hbar}\,, \end{aligned} \tag{18}$$

identifies the equilibrium probability of the configuration $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{N_\tau}\}$ as

$$P_{\rm eq}(\boldsymbol{x}) = \frac{e^{-S(\boldsymbol{x})/\hbar}}{Z}\,, \tag{19}$$

and averages of an operator $\mathcal{O}(\boldsymbol{x})$ are calculated as

$$\langle \mathcal{O} \rangle = \int_X [D\boldsymbol{x}(\tau)] \mathcal{O}(\boldsymbol{x}) P_{\rm eq}(\boldsymbol{x})\,, \tag{20}$$

in which $[D\boldsymbol{x}(\tau)]$ signifies that the integral is over all paths in the state space X.

The Monte Carlo method introduced by Metropolis is based on the idea of "importance sampling", whereby the phase-space points $\boldsymbol{x}$ in (20) are not selected completely at random, which would be impractical for higher-dimensional spaces, but are chosen to be more densely distributed in region(s) of phase space providing the dominant contributions to the integral. The ergodicity of the Markov chains used here enables us to calculate averages as arithmetic averages over Markov chains:

$$\int_X [D\boldsymbol{x}(\tau)] \mathcal{O}(\boldsymbol{x}) P_{\rm eq}(\boldsymbol{x}) = \lim_{n\to\infty}\left[\frac{1}{n}\sum_{i=1}^{n} \mathcal{O}(\boldsymbol{x}_i)\right]\,, \tag{21}$$

where n is the number of states generated by the Markov chain. Of course, in actual calculations, only a finite number of states are used to estimate averages, as determined by the operator and the system. The next section provides the mathematical foundation for (21).

C. Properties of Markov Chains

If the state space is finite or countable, then the elements of the Markov chain can be represented as vectors and the transition probability as a matrix that operates on these vectors. However, most applications of the MCMC method have uncountable state spaces in which the initial state is an unconditional probability distribution and the transitions are expressed in terms of a conditional probability distribution. Finite state spaces are simpler to present, so we will use this setting in what follows.

For a Markov chain that consists of n states, the transition operator is an $n \times n$ matrix P whose entries p_{ij} signify the probability of moving from state s_i to state s_j in a single step. Similarly, the probability to move from state s_i to s_j in 2 steps is $\sum_{k=1}^{n} p_{ik}p_{kj}$, which is the (i,j)th element of P^2. The generalization to m steps is the (i,j)th element of P^m.

If a transition operator does not change across transitions, the Markov chain is called (time) homogeneous. An important consequence of homogeneity is that, as $t \to \infty$, the Markov chain will reach an equilibrium called the stationary distribution of the chain. The stationary distribution of a Markov chain is important for sampling from probability distributions, which is at the heart of MCMC methods. In more formal terms, suppose that the current state of the Markov chain is represented by the probability vector $\boldsymbol{u}$, which means that u_i is the probability of being in state s_i. Now we want to know the probability v_j of finding the chain in state s_j after m-steps. In matrix form, this is given by

$$\boldsymbol{v}^{\mathsf{T}} = \boldsymbol{u}^{\mathsf{T}}\mathsf{P}^m \,. \tag{22}$$

If there is a vector such that $\boldsymbol{w}^{\mathsf{T}} = \boldsymbol{w}^{\mathsf{T}}\mathsf{P}$, which also implies $\boldsymbol{w}^{\mathsf{T}} = \boldsymbol{w}^{\mathsf{T}}\mathsf{P}^m$, then the Markov chain is said to be in *equilibrium* and $\boldsymbol{w}$ is called a *stationary* vector. Computationally, to find such a vector, one calculates the left eigenstate of the matrix P corresponding to unit eigenvalue.

The continuous state analogue of p_{ij} is the transition density $p(x,y)$. The continuous quantity corresponding to the m-step transition probability P^m is denoted as $p^{(m)}(x,y)$, which is defined by the Chapman–Kolmogorov recursion relation,

$$p^{(n)}(x,y) = \int_S p^{(n-1)}(x,z)p(z,y)\,dz \,, \tag{23}$$

for $n \geq 2$.

The most important theorem behind the MCMC method is:[4,5]

Fundamental limit theorem. An irreducible, aperiodic Markov chain with transition matrix P has a stationary distribution $\boldsymbol{w}$ satisfying $w_j > 0$, $\sum_j w_j = 1$, and $\boldsymbol{w}^{\mathsf{T}} = \boldsymbol{w}^{\mathsf{T}}\mathsf{P}$, if and only if all its states are positively recurrent, and w is unique and identical to the limiting distribution

$$w_j = \lim_{n\to\infty} (\mathsf{P}^n)_{ij} \tag{24}$$

with the following definitions:

- An *irreducible* Markov chain is the one in which the probability to go from any state s_i to any other state s_j in a finite number of steps is non-zero. Thus, there exists a finite n such that

$$\sum_{k_1}\sum_{k_2}\cdots\sum_{k_{n-1}} p_{ik_1}p_{k_1k_2}\cdots p_{k_{n-1}j} \neq 0\,. \tag{25}$$

- The states of a Markov chain are called *positive recurrent* if the probability to return to the same state is unity after a finite number of evolution steps.

- The states are *periodic* if the transition $s_i \to s_i$ is only possible in steps which are integral multiples of the period $d(i)$. The period is defined as the highest common factor for all m for which $(\mathsf{P}^m)_{ii} > 0$. For an aperiodic state s_j, $d(j) = 1$.

Note that the stationary distribution obtained in (24) is independent of initial state s_i. This theorem gives us the freedom to start in any *ergodic* (aperiodic, irreducible and positive recurrent) Markov chain and we are guaranteed to end up in an equilibrium state. The process of evolution of the chain into a stationary state is called *thermalization*. Although stated in terms of discrete states, with suitable modifications, this theorem is also valid for continuous states. We refer the reader to Ref. 5 for details.

IV. SPLINES

There are several methods for fitting curves to a set of given data points that enable the prediction of values between these points (interpolation) and the estimate of values outside the range of the points (extrapolation). In spline interpolation, piecewise functions pass through

a set of data points, often referred to as *knots*, such that the function is smooth at these data points and satisfies some specified conditions at or near the first and last points. Thus, if there are n points (x_i, y_i), for $i = 1, 2, \ldots, n$, a total of $n-1$ functions is used, one for each interval between the ith and $(i+1)$st data points, for $i = 1, 2, \ldots, n-1$. Cubic polynomials are most commonly employed to strike a balance between smoothness and avoiding Runge's phenomenon, which is the appearance of oscillations at the edges of an interval that occurs when using high-order polynomials for a set of equidistant interpolation points. The cubic polynomials are written as

$$P_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i\,, \tag{26}$$

where $i = 1, 2, \ldots, n-1$ for $x \in [x_i, x_{i+1}]$. Each polynomial has four unknown coefficients (a_i, b_i, c_i, d_i), so the complete spline has $4(n-1)$ unknowns. These unknowns must be chosen according to conditions that the spline must satisfy:

1. $P_i(x_i) = y_i$ and $P_{n-1}(x_n) = y_n$ for $i = 1, 2, \ldots, n-1$. These conditions guarantee that the spline interpolates the data points.

2. $P_i(x_i) = P_{i+1}(x_i)$ for $i = 1, 2, \ldots, n-1$. These conditions require that the values of adjacent polynomials are the same at the points where they meet, which ensure that the interpolating spline is *continuous*.

3. $\left.\dfrac{dP_{i-1}}{dx}\right|_{x=x_i} = \left.\dfrac{dP_i}{dx}\right|_{x=x_i}$ for $i = 2, \ldots, n-1$ or, in abbreviated notation, $P'_{i-1}(x_i) = P'_i(x_i)$. These conditions supplement the continuity of the spline by requiring that the slopes of adjacent polynomials are the same at the point where they meet. Thus, the spline is *differentiable* on (x_1, x_n).

4. $\left.\dfrac{d^2P_{i-1}}{dx^2}\right|_{x=x_i} = \left.\dfrac{d^2P_i}{dx^2}\right|_{x=x_i}$ for $i = 2, \ldots, n-1$ or, in abbreviated notation, $P''_{i-1}(x_i) = P''_i(x_i)$. This condition supplements the continuity and differentiability of the spline by requiring that the second derivatives of adjacent polynomials are the same at the point where they meet. Thus, the spline has a *second derivative* at every point on (x_1, x_n).

The four conditions on the spline provide

$$(n-1) + (n-1) + (n-2) + (n-2) = 4n - 6 \tag{27}$$

constraints on the $4(n-1)$ coefficients of the spline. That leaves $4n-4-(4n-6)=2$ conditions left to uniquely determine the spline. These are imposed at or near the two end points of the data set, effectively providing boundary conditions for the spline function. There are various ways of specifying these boundary conditions:

1. The *natural spline* boundary conditions are

$$\left.\frac{d^2P_1}{dx^2}\right|_{x=x_1} = \left.\frac{d^2P_{n-1}}{dx^2}\right|_{x=x_n} = 0, \tag{28}$$

that is, the second derivatives vanishes at the end points. These boundary conditions seldom used since they does not provide a sufficiently accurate approximation near the end points of the data set.

2. The *clamped spline* boundary conditions set the first derivatives at the end points to a particular value, which may be known, or specified at the user's discretion:

$$\left.\frac{dP_1}{dx}\right|_{x=x_1} = f_1'\,, \qquad \left.\frac{dP_{n-1}}{dx}\right|_{x=x_n} = f_n'\,. \tag{29}$$

The equations determining the coefficients for cubic splines for either natural or clamped boundary conditions can be expressed as tridiagonal matrices.[7]

3. The *not-a-knot* boundary conditions, where no extra conditions are imposed at the end points. At each point (i.e. knot), the spline changes from the cubic polynomial in the preceding interval changes smoothly to the polynomial in the next interval. The basic idea of the not-a-knot boundary conditions is to not change the cubic polynomials across the second and penultimate points, which are first two interior points. Thus, these two points are not knots, which results in the first two intervals having the same spline function and the last two intervals having the same two spline functions. The mathematical expression of these boundary conditions are

$$\left.\frac{d^3P_1}{dx^3}\right|_{x=x_2} = \left.\frac{d^3P_2}{dx^3}\right|_{x=x_2}, \qquad \left.\frac{d^3P_{n-2}}{dx^3}\right|_{x=x_{n-1}} = \left.\frac{d^3P_{n-1}}{dx^3}\right|_{x=x_{n-1}}. \tag{30}$$

The default in-built MATLAB function performs a cubic not-a-knot spline fit.

V. COMPUTER PROGRAMS

Two MATLAB codes were used to produce the results in our research. The first one is numerical evaluation of the path integral using MCMC. Within this code we have used another

short MATLAB code `UWerr.m`, which is freely available on the internet.[6] The purpose of `UWerr.m` is to calculate the mean of a sample which may exhibit autocorrelations. The second code is the direct integration of the Schrödinger equation. The following are the codes, created on MATLAB R2018a Update 3 (9.4.0.885841).

Listing 1. Path Integral ().

```
N_sweep=20000;
ep=input('Lattice spacing: ');
h=input('Enter the h for [-h,h] ');
L=input('The anharmonicity parameter: ');
N_t=250/ep;

X2=zeros(1,N_sweep); %To store <x^2> from each path
X4=zeros(1,N_sweep); %To store <x^4> from each path
u=zeros(1,N_t+1);    %Initialise a cold start
U=[];

for j=1:N_sweep
    %accrate=0;     %Uncomment for acceptance rate
    order=randi([2,N_t],1,N_t-1);%Site visiting order

    for i=1:N_t-1   %This loop consitutes 1 sweep
        k=order(i);
        del=2*h*(rand-0.5);
        uc=u(k)+del;
        DS=ep*del*((1+ep^2/2)*(uc+u(k))-(u(k-1)+u(k+1)))+...
            L*ep^5*(uc^4-u(k)^4);

        if(rand<exp(-DS))
            u(k)=uc;
            %accrate=accrate+1/(N_t-1);
        end
    end
```

```
    X2(j)=mean(u.^2);
    X4(j)=mean(u.^4);
    if mod(j,100)==0 %100 sweeps skipped for ep>0.2 otherwise
       use 2*tauint
        U=[U ep*u];
    end
end
[UWX2,UerrX2,ererX2,tauX2,dtauX2]=UWerr(X2',[],[],0,1);
[UWX4,UerrX4,ererX4,tauX4,dtauX4]=UWerr(X4',[],[],0,1);

H=histogram(U,50,'Normalization','pdf');  %Path histogram
fprintf('Bin width of histogram %f\n', H.BinWidth);
xlabel('$x$','Interpreter','latex','Fontsize',20)
ylabel('$|\psi_0|^2$','Interpreter','latex','Fontsize',20)

E_0=ep^2*UWX2+3*L*ep^4*UWX4;    %Virial theorem
ErrE_0=ep^2*UerrX2+3*L*ep^4*UerrX4; %Associated error
fprintf('\n E_0 = %f\n', E_0);
fprintf('Error in E_0 = %f\n', ErrE_0);
```

Listing 2. Schrödinger Equation ().

```
function SE
global n a
a=input('Anharmonicity = ');
n=input('Enter the n: ');

n1=n;
solinit=bvpinit(linspace(0,10,100),@init,n1);
sol=bvp4c(@sch,@bc,solinit);
u=linspace(0,10,1000); %Solve the BVP on this interval
```

```
uo=linspace(-10,10,1999);
Su=deval(sol,u);

SU=Su(1,:);                 %Only the wavefunction is needed!
pks=length(abs(findpeaks(SU( abs(SU.^2)>0.01 ).^2)));
if SU(1)==1
    count=2*pks+1;
elseif SU(1)==0
    count=2*pks;
end

for q=1:10
    if count~=n+1
        n1=(n1+sol.parameters);
        solinit=bvpinit(linspace(0,10,100),@init,n1);
        sol=bvp4c(@sch,@bc,solinit);
        Su=deval(sol,u);
        SU=Su(1,:);
        pks=length(abs(findpeaks(SU(abs(SU.^2)>0.01).^2)));
        if SU(1)==1
            count=2*pks+1;
        elseif SU(1)==0
            count=2*pks;
        end
    else
        break;
    end
end

if mod(n,2)==0              %For even function
    psi=[flip(SU(2:end)) SU(1) SU(2:end)]; %Even extension
else                        %For odd function
```

```
    psi=[-flip(SU(2:end)) SU(1) SU(2:end)]; %Odd extension
end
N=simp(psi.^2); %Normalisation constant

plot(uo,1/N*psi.^2,'r','LineWidth',1.5)
xlabel('$x$','Interpreter','latex','Fontsize',20)
ylabel('$|\psi_0|^2$','Interpreter','latex','fontsize',20)

u2=simp(uo.^2.*psi.^2)/N; u4=simp(uo.^4.*psi.^2)/N;
fprintf('<u^2>= %0.4f\n',u2)
fprintf('<u^4>= %0.4f\n',u4)
fprintf('\nThe eigen-energy is %0.6f\n',sol.parameters);

%-------------------------------------------------------------
function eqns=sch(x,y,E)
global a
eqns=[y(2),(x^2+2*a*x^4-2*E)*y(1)];
%SE is 2nd order; convert it into 2 Ist order eq.

%-------------------------------------------------------------
function res = bc(ya,yb,E)
global n
if mod(n,2)~=0
    res=[ya(1),yb(1),ya(2)-1];
else
    res=[ya(1)-1,yb(1),ya(2)];
end
%-------------------------------------------------------------
%This function creates an initial guess for the function.
%An obvious choice for the guess would be the known
%solutions of harmonic oscillator.
function psi = init(x)
```

```
global n
if n~=0
    psi=[hermiteH(n,x)*exp(-x^2/2),(2*n*hermiteH(n-1,x)...
        -x*hermiteH(n,x))*exp(-x^2/2)];
else
    psi=[hermiteH(0,x)*exp(-x^2/2),-x*hermiteH(0,x)*...
        exp(-x^2/2)];
end
%psi(1)= wavefunction, psi(2) = its derivative
%------------------------------------------------------------
%Simpson's 1/3 rule for numerical integration
function I = simp(y)
I=1/(300)*(y(1)+2*sum(y(3:2:1997))+4*sum(y(2:2:1998))...
    +y(end));
```

---

[1] F. Reif, **Fundamentals of Statistical and Thermal Physics** (McGraw–Hill, New York, 1985).

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of state calculations by fast computing machines," J. Chem. Phys. **21**, 1087–1092 (1953).

[3] W. A. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," Biometrica **57**, 97–190 (1970).

[4] C. Morningstar, "The Monte Carlo method in quantum field theory," `arXiv:hep-lat/0702020v1` (2007).

[5] G. O. Roberts and J. S. Rosenthal, "General state space Markov chains and MCMC algorithms," Probab. Surveys **1**, 20–71 (2004).

[6] U. Wolff, "Monte Carlo errors with less errors," Comput. Phys. Commun. **156**, 143–153 (2004).

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing* 3rd edn (Cambridge University Press, Cambridge, UK, 2007), pp. 120–124.