



# Evaluating Feynman Path Integrals Through Numerical Solution

Archie Headland, Department of Physics, Durham University, 14/12/2025

## Theory

The Feynman path integral is built on the idea that a particle follows **every** possible path between two points, each weighted with an **action**,  $S$ . In one-dimensional space, the **propagator**,  $G$ , represents the probability amplitude for a particle to evolve from  $x_i$  to  $x_f$  from time  $t_i$  to  $t_f$ , it is computed by using a path integral:

$$G(x_f, x_i) = \int \mathcal{D}x(t) e^{iS[x]/\hbar} \quad (1)$$

This path integral is numerically hard to compute as the integrand is **complex** and exhibits **oscillating** behaviour. This can be avoided by converting real time to **imaginary** time:  $t = -it$ . This is used to derive the **Euclidean** path integral:

$$G(x_f, x_i) = \int \mathcal{D}x(t) e^{-S[x]/\hbar} \quad (2)$$

Each path in the sum is weighted by  $e^{-S[x]/\hbar}$ , meaning paths with large action are exponentially **suppressed**. The path of **minimum** action corresponds to the trajectory predicted by **Newton's** laws. Therefore, **classical** mechanics emerges as the limiting case of **quantum** mechanics. For **macroscopic** objects, the propagator is dominated by paths very close to the path of **minimum** action, this is because their action for more distant paths much larger than  $\hbar$  and so contributions from distant paths are near zero. As a result, quantum effects become **negligible** at macroscopic scales and are primarily observable at **microscopic** scales.

## Path Generation

Solving a Euclidean path integral is difficult, as a particle can follow an infinite number of infinitely complex paths. To simplify this problem a computer program was used to generate a **huge** number of **discretised** paths. Each path allows the particle to jump between space in discrete time steps as shown in Fig. 1 where the total number of steps is  $N$ , and the time spacing between steps is set as  $a$ .

Every path should have the same start and end value,  $x_0 = x_N$ . This is a boundary condition for the ground-state wave function of a harmonic oscillator. These simplifications reduce the problem to an approximate expression of the quantum propagator:

$$G(x_f, x_i) \approx A \int_{-\infty}^{\infty} dx_1 \dots dx_{N-1} e^{-S[x]} \quad (3)$$

## Metropolis Algorithm

The Metropolis algorithm samples paths with probability proportional to  $e^{-S[x]/\hbar}$ , preferentially selecting paths with **lower** action. It starts with an arbitrary path and modifies it by **randomising** every point  $x_j$  according to the following steps:

1. Generate a random number  $\zeta$  from a uniform probability distribution between  $-\epsilon$  and  $\epsilon$  where  $\epsilon$  is a chosen constant.
2. Update  $x_j$  to  $x_j + \zeta$  and compute the change in action  $\Delta S$  caused by this replacement.
3. If  $\Delta S < 0$  keep the new value for  $x_j$  and visit the next point.
4. If  $\Delta S > 0$  generate a random number  $\eta$  uniformly distributed between 0 and 1. Keep the new  $x_j$  if  $e^{-\Delta S} > \eta$ , else restore the old value and visit the next point.

This process is known as one sweep or **update**, we aim to choose  $\epsilon$  such that 40-60% of the  $x_j$ 's are changed on each sweep.

To start, the path is updated  $10 N_{cor}$  times to **thermalise** the path (to remove the bias from the initial chosen path). The path is then updated  $N_{cor}$  times and the position at the **midpoint** of the path is recorded, this is then repeated  $N_{cf}$  times until sufficient paths have been recorded. In the limit of **large** imaginary time the distribution of midpoint positions sampled by the metropolis algorithm approaches the probability density  $|\psi_0(x)|^2$ .

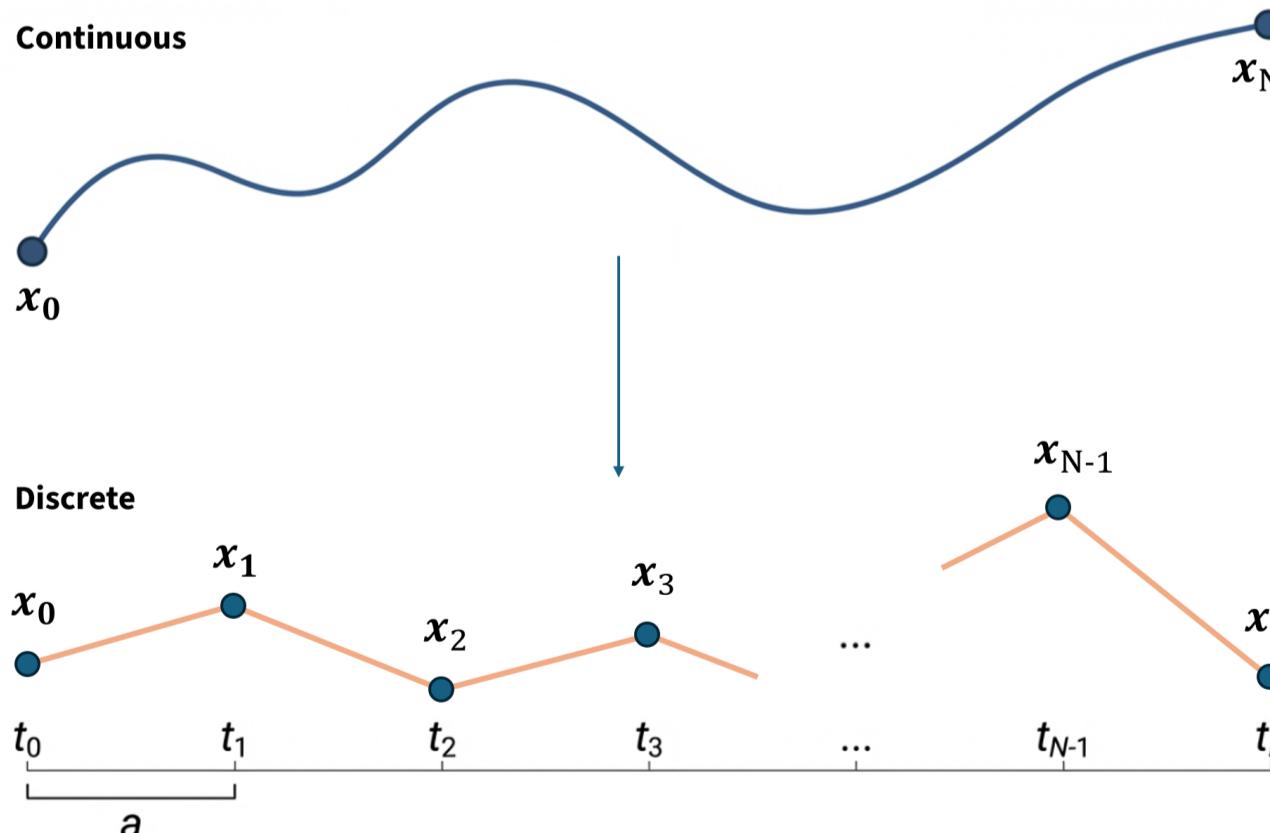


Fig.1. Diagram showing the time discretisation of a continuous complex path with time spacing  $a$ .

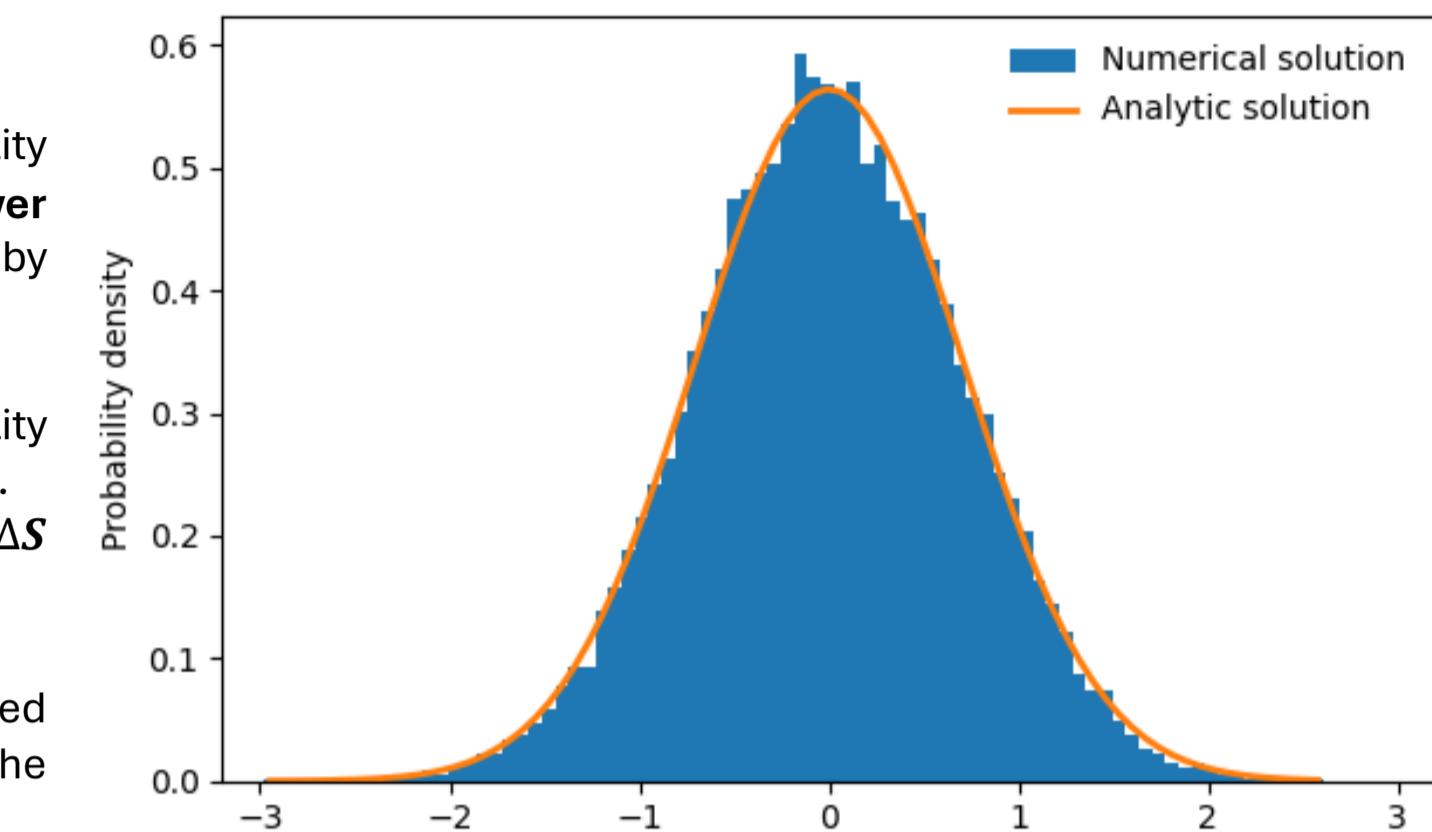


Fig.2. Analytic against approximate ground-state probability density for a harmonic oscillator found with the metropolis algorithm with  $\epsilon = 1.4$ ,  $a = 0.5$ ,  $N_{cor} = 200$ ,  $N_{cf} = 20000$ ,  $N = 100$ .

## Results

The numerical approximation appears to reproduce the expected **gaussian** shape over the range  $-3 \leq x \leq 3$ . Chi-squared analysis was used to assess **quantitative** agreement where each bin of the histogram was compared to the analytic solution. For 5 runs of the program the **average** chi-squared value was  $\chi^2 = 77.9$  and the reduced chi-squared value was  $\chi^2_v = 1.08$ , this shows **strong** agreement between the numerical and analytical fit. Since  $\chi^2 \approx 1$  this suggests that discrepancies in the numerical fit are caused by a **finite** number of trials rather than any systematic errors with our metropolis algorithm

## Conclusion

The metropolis algorithm is an effective way to approximate ground-state probability distributions for a harmonic oscillator. The accuracy of the numerical fit could be further improved by using a **larger** number of trials  $N_{cf}$ , or a larger  $N_{cor}$  to reduce correlation between samples. A future extension of this work could be the extraction of **excited-state** energies in anharmonic potentials, where the absence of exact solutions makes numerical methods essential.

## References

- [1] R. P. Feynman, "Space-Time Approach to Non-Relativistic Quantum Mechanics." Rev. Mod. Phys. 20, 367–387 (1948). doi: 10.1103/RevModPhys.20.367
- [2] G. P. Lepage, "Lattice QCD for Novices." (2005). doi: 10.48550/arXiv.hep-lat/0506036