Problem Set 7: Quantum algorithms and WKB
Graduate Quantum I
Physics 6572
James Sethna
Due Friday Oct. 10
Last correction at October 9, 2014, 1:50 pm

Potentially useful reading
Schumacher & Westmoreland section 7.2 (No-cloning theorems),
and sections 18.1 & 18.2 (quantum algorithms)
Sydney Coleman, "The Uses of Instantons", sections 1 & 2 (WKB, instantons)

7.1 **Mirror path integrals.** (Path Integrals) ③

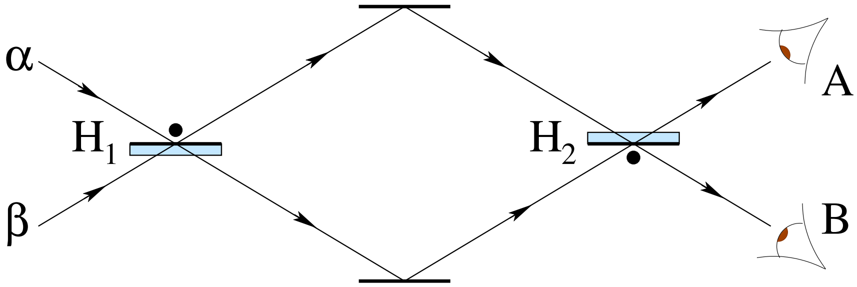


Figure 1: **Qbit weirdness**. *A photon, passing through a pair of half-silvered mirrors H_1 and H_2, undergoes quantum interference between different paths.*

One of the most compelling examples of Qbits and their weirdness is provided by the example of photons and half-silvered mirrors. Fig 1 shows a photon[1] coming from the left in a superposition $\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right)$, through a set of mirrors, to two detectors named Alice (A) and Bob (B). We work in the basis $|1\rangle = \left(\begin{smallmatrix}1\\ 0\end{smallmatrix}\right)$ representing the upper of the two beams at a given position, and $|0\rangle = \left(\begin{smallmatrix}0\\ 1\end{smallmatrix}\right)$ representing the lower of two beams. As discussed in Schumacher & Westmoreland, the half-silvered mirror H_2 approximately acts as a unitary transformation $H_2 = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right)$. The first mirror H_1, with its mirrored side on the top, changes the sign of the beam reflecting from its top, hence $\frac{1}{\sqrt{2}}\left(\begin{smallmatrix}-1 & 1\\ 1 & 1\end{smallmatrix}\right)$.

[1] We assume, as before, that the polarization of the photon lies perpendicular to the plane of the paper, so that it remains unchanged and hence unimportant to the interference.

The product $G = H_2 H_1$ is analogous to the propagator, or Green's function, for this system.[2]

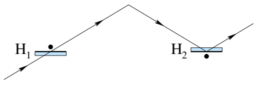
(a) *What is G? Given the impinging wave* $\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right)$ *from the left, what are the probabilities* P_A *and* P_B *that Alice and Bob will see the photon?* (Hint: Remember Bob did not see anything when the initial photon came from below.)

We can develop a kind of discrete path integral representation of the propagator G by writing

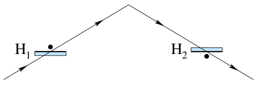
$$G = \mathbb{1} H_2 \mathbb{1} H_1 \mathbb{1} = \sum_{x_i=0}^{1} \sum_{x_m=0}^{1} \sum_{x_f=0}^{1} |x_f\rangle\langle x_f|H_2|x_m\rangle\langle x_m|H_1|x_i\rangle\langle x_i|. \tag{1}$$

Here i, m, f representing the initial states, the states in the middle, and the final (detected) states, and $\mathbb{1} = |1\rangle\langle 1| + |0\rangle\langle 0|$. If we assume the initial photon is coming from the bottom left ($x_i = 0$), there are four remaining paths in this sum.

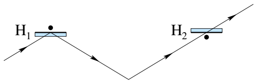
$A_1 =$



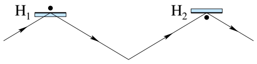
$A_2 =$



$A_3 =$



$A_4 =$



(b) *Give the four amplitudes* A_i *contributed by these four paths. Which ones contribute to* $\langle 1|G|0\rangle = G_{10}$*? Which ones contribute to* $\langle 0|G|0\rangle = G_{00}$*? Which go to Bob? Do the sum of the amplitudes going to Bob equal zero (as they should)?*

Imagine an electron traversing an electron-mirror array, impinging from below. The mirrors H_1 and H_2 have the same effect on the amplitudes as the former half-silvered ones did for the photon. Here, though, we thread a solenoid between the upper and lower paths in the middle region, enclosing a net magnetic flux Φ pointing upward out of the page. The field is zero outside the solenoid, and you may ignore the electron's spin.

(c) *As a function of* Φ*, what is the probability that an initial electron will be seen by Alice? What values of* Φ*, in multiples of the elementary flux quantum* $\Phi_0 = hc/e = 2\pi\hbar c/e$*, prevent Alice from seeing any electrons?* (Remember, the initial electron comes from below. Hints: $\oint_C \mathbf{A} \cdot \mathrm{d}\ell = \Phi$ if the path C encircles the solenoid counterclockwise once. The path-integrand amplitude for $\mathbf{x}(t)$ in a field $\mathbf{A}$ gains a phase $\zeta = \int (q/c)\mathbf{A}(x) \cdot \mathrm{d}\mathbf{x}/\hbar$. The charge on an electron is $q = -e$.)

[2] Note two confusing things in our notation. First, the photon moves from left to right (hitting H_1, then H_2), but the matrices describing the evolution propagate from right to left ($H_2 H_1$). Second, $|1\rangle = \left(\begin{smallmatrix}1\\ 0\end{smallmatrix}\right)$ is a vector whose first element is one [zeroth in Python/C], and $|0\rangle = \left(\begin{smallmatrix}0\\ 1\end{smallmatrix}\right)$ is a vector whose *second* element is one [first in Python/C].

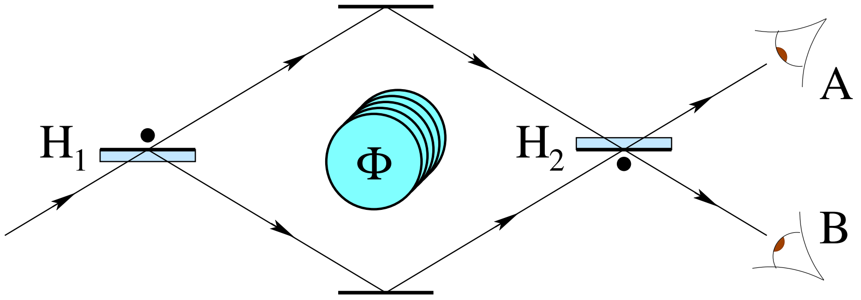


Figure 2: **Bohm-Aharonov and mirrors**.

7.2 **Quantum Algorithms.**[3] (Quantum Information Processing) ④

Are quantum computers faster than our standard classical computers?

Clearly, we need to define our terms here – since factoring 143 (the current quantum-computing world record) doesn't take long on a classical computer. The key question is how the computer time would scale for large problems. Factoring M-digit numbers on a quantum computer takes no more than $O(M^3)$ time (that is, some constant times M^3, using Shor's algorithm), while the most efficient known method for factoring on classical computers takes $O(e^{1.9M^{1/3}(\log M)^{2/3}})$. For large numbers of digits, quantum computers win (if they can be built). There are a few other problems where classical computers are known to be much slower than quantum computers: Grover's algorithm for searching an unsorted database, Simon's algorithm, . . .

Here we explore a somewhat artificial problem, solved in the quantum case using the Deutsch-Josza algorithm.[4] This will also introduce the *reversibility* of quantum computing, and the use of *gates* – unitary operators that transform Qbits to execute the quantum computer program.

The Deutsch-Josza algorithm considers functions that map n bits to one bit. Let us denote the n bits as $x_0, x_1, \cdots, x_{n-1}$, where $x_0, x_1, \cdots, x_{n-1}$ are all 0 or 1. Let $\mathbf{x} = \sum_{m=0}^{n-1} 2^m x_m$ denote the integer represented by the bit sequence $x_0, x_1, \cdots, x_{n-1}$; x_0 is the least significant bit and x_{n-1} is the most significant bit. We'll also denote $|x_0\rangle|x_1\rangle \cdots |x_{n-1}\rangle$ as $|\mathbf{x}\rangle$.

Let f be a function that maps the n bits $x_0, x_1, \cdots, x_{n-1}$ to one bit (that is, either True or False, one or zero). For example, f could be a function Prime that returns True if the integer $\mathbf{x}$ is prime, or Even that returns True if the integer is divisible by two, or Big that returns True if the integer is greater than or equal to 2^{n-1}. Our algorithm is

[3] Developed in collaboration with Bhuvanesh Sundar, based on an exercise by Paul Ginsparg.

[4] There are many discussions of the Deutsch-Josza algorithm in the literature – feel free to consult them. If you find one that is particularly pertinent, reference it properly in your writeup.

not concerned with implementing $f(\mathbf{x})$, but with testing properties of an unknown f by sampling its output. For example, testing whether f is a *constant function* (either True for all possible $\mathbf{x}$, or False for all arguments) is a challenge for classical computers. (An experiment may find a thousand Trues in a row, but to be sure that the function always returns True one must test all 2^n choices of $\mathbf{x}$.) We define a *balanced* function to be one which returns True for exactly half of the possible inputs. Thus Even and Big above are balanced, but Prime is neither balanced nor constant.

(a) *Write the four possible functions $f(x_0)$ for $n = 1$ (two possible inputs, two possible outputs). Which are constant? Which are balanced?* For larger n, most possible functions are neither balanced nor constant.

Deutsch and Josza considered the artificial problem of distinguishing between balanced and constant functions. Let us define DJ functions to be those functions guaranteed to be either balanced or constant. Given that f is a DJ function, can a quantum computer probing f distinguish between the two cases faster than a classical computer? Let us first consider how a classical computer would solve this.

(b) *Argue that in the worst case, the n-bit DJ function f would have to be called $2^{n-1}+1$ times in order to determine for certain whether it is balanced or constant.*

Our challenge is to use a quantum computer program to do this calculation with *one* operation of the operator f. How do we set this up?

A quantum computer performs unitary operations on Qbits to execute the program. Unitary operations are reversible;[5] indeed, the only irreversible step in a perfect quantum computer is the macroscopic observer reading the answer. This means that no quantum computer can perform the classical AND operation, for example – since $AND(x_0, x_1)$ is False for three different values of x_0 and x_1, it would throw out information that could not be retrieved. The workaround is to encode the answer in a final Qbit y. So if $n = 2$ and $f(x_0, x_1) = AND(x_0, x_1)$ (a function that is neither balanced nor constant), we could implement f on a quantum computer by writing a program that took $|x_0\rangle|x_1\rangle|y\rangle$ and returned $U_f(|x_0\rangle|x_1\rangle|y\rangle) = |x_0\rangle|x_1\rangle|y \oplus AND(x_0, x_1)\rangle$ where $\oplus$ is addition modulo 2. If you input $|x_0\rangle|x_1\rangle|y = 0\rangle$, the output value of $|y\rangle$ gives $f(\mathbf{x})$. If you input $|x_0\rangle|x_1\rangle|y = 1\rangle$, the output value of $|y\rangle$ gives $1 \oplus f(\mathbf{x}) = NOT(f(\mathbf{x}))$ – this feature is important to keep U_f reversible. U_f is also linear: for example, $U_f((\alpha|0\rangle + \beta|1\rangle)|0\rangle|y\rangle) = \alpha|0\rangle|0\rangle|y \oplus f(0,0)\rangle + \beta|1\rangle|0\rangle|y \oplus f(1,0)\rangle$.

(c) *Show that U_f is reversible for the case where $f = AND$ by giving an explicit method for reconstructing x_0, x_1, and y from x_0, x_1, and $y \oplus AND(x_0, x_1)$. Then show in general that U_f is its own inverse for any n-bit function f.*

We are now given a quantum computer operation that evaluates an unknown DJ function f: $U_f(|x_0\rangle|x_1\rangle \cdots |y\rangle) = |x_0\rangle|x_1\rangle \cdots |y \oplus f(\mathbf{x})\rangle)$.

Just as a classical computer can be made of AND gates, NOT gates, OR gates, etc., so a quantum computer is composed of gates that manipulate one or two Qbits by

[5] The reverse operation is $U^\dagger = U^{-1}$.

application of unitary operators. The single-Qbit gates are thus 2×2 unitary matrices.

(d) *In the basis*[6] $|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ *and* $|1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$*, write the single-Qbit gate NOT as a* 2×2 *matrix. Show that the Hadamard gate, written* $H = 1/\sqrt{2}\left(\begin{smallmatrix}1 & 1\\1 & -1\end{smallmatrix}\right)$*, is unitary.* We can implement both the NOT gate and the H gate on electron Qbits, for example, by exposing them in a magnetic field with a suitable direction and orientation.

Our strategy will be to apply U_f not to a Qbit product that corresponds to a classical bit sequence $|x_0\rangle|x_1\rangle \cdots |x_{n-1}\rangle$, but rather to a Qbit string that represents a quantum superposition of all possible classical bit sequences. Let us first consider[7] the case $n = 1$. Our strategy is to use the Hadamard gate to create a superposition of bit sequences and then apply U_f, and then re-apply the Hadamard gate to find out whether our function is constant or balanced. We shall abuse notation to use $H^{n+1}|\mathbf{x}\rangle|y\rangle$ to mean $(H|x_0\rangle)(H|x_1\rangle) \cdots (H|x_{n-1}\rangle)(H|y\rangle)$.

(e) *Starting with the case of* $n = 1$ *Qbit plus* $|y\rangle$*, initialize our two Qbits to* $|\Psi_0\rangle = |x_0\rangle|y\rangle = |0\rangle|1\rangle$*. Apply the Hadamard operation on both Qbits (exposing them both to the same magnetic field). What is the resulting superposition? Apply* U_f *for the four cases of* f *you found in part (a), and then apply the Hadamard transformation on both the Qbits again. What is the measured final value of* x_0 *for the constant functions with* $n = 1$*? For the balanced functions?*

You should have found that you could conclusively say if f were constant or balanced with just one call to U_f.

Now that you have worked out the $n = 1$ case, let us generalize to arbitrary n. The algorithm proceeds in the same way. We initialize each of the n Qbits $|x_0\rangle, |x_1\rangle, \cdots, |x_{n-1}\rangle$ to $|0\rangle$, and $|y\rangle$ to $|1\rangle$, so $|\Psi_0\rangle = |0\rangle^n|1\rangle$. We perform the Hadamard operation on all the Qbits, so $|\Psi_1\rangle = H^{n+1}|\Psi_0\rangle = (H|0\rangle)^n(H|1\rangle)$. We pass them through U_f, so $|\Psi_2\rangle = U_f|\Psi_1\rangle$. We perform another Hadamard operation on all the Qbits, $|\Psi_3\rangle = H^{n+1}|\Psi_2\rangle$. Finally, we measure the overlap with the initial state, $|\langle\Psi_0|\Psi_3\rangle|^2$, measuring the probability that $x_0 = 0$, $x_1 = 0$, $\ldots$, $x_{n-1} = 0$, $y = 1$.

Let us do this step by step. The initial state of the Qbits is $|\Psi_0\rangle = |0\rangle^n|1\rangle$. A Hadamard operation is then applied on all of them. The state of the Qbits after this operation is $|\Psi_1\rangle = H^{n+1}|\Psi_0\rangle = (H|0\rangle)^n(H|1\rangle)$.

(f) *Write* $|\Psi_1\rangle$ *as a superposition of* $|\mathbf{x}\rangle|y\rangle$ *for all possible* n*-bit binary numbers* $\mathbf{x}$ *and both values of* y*. Show that the probabilities of being in these states are all equal (but the amplitudes may have different signs).*

Now the Qbits are passed through U_f. The state of the Qbits after passing through U_f is $|\Psi_2\rangle = U_f|\Psi_1\rangle$. When f is a *constant* function, U_f changes the Qbit y in the same way for all arguments $\mathbf{x}$.

(g) *If* f *is a constant function, show that* $|\Psi_2\rangle$ *is a constant times* $|\Psi_1\rangle$*. What is this constant if* $f \equiv 0$*? If* $f \equiv 1$*? Show that the measured values of* $x_0, x_1, \cdots, x_{n-1}$ *in*

[6] I apologize for the shift in notation: I used the reverse convention in lecture, $|1\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $|0\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$.

[7] The special case $n = 1$ of the Deutsch-Josza algorithm is called the Deutsch's algorithm.

$|\Psi_3\rangle$ *after the final Hadamard operation are all* 0*, so in particular that* $|\langle\Psi_0|\Psi_3\rangle|^2 = 1$. (Hint: Do not try to apply H^{n+1} on $|\Psi_2\rangle$ written as a superposition of several terms. Instead, decompose $|\Psi_2\rangle$ as a product of a state for each Qbit ($|\Psi_2\rangle = \Pi_{0\leq i<n}|\phi_i\rangle$ where $|\phi_i\rangle$ is a superposition of $|0\rangle$ and $|1\rangle$), and use the fact that $H^2 = \mathbb{1}$).

Hence for a constant function, the result of our quantum computation always has unit probability of returning the state with all $x_i = 0$ and $y = 1$.

When f is a balanced function, the value in the Qbit y is changed differently for different arguments $\mathbf{x}$; for half of those 2^n arguments, y is left unchanged, and for half of those arguments, y is flipped (from 0 to 1 or vice versa). Let us illustrate this with an example: let us consider the function Even, which returns True if the integer $\mathbf{x}$ is divisible by two.

(h) *What is the least significant bit of an integer if it was even? If the integer was odd? Argue that whether the Qbit y is flipped by U_f or not is determined solely by the least significant bit x_0 in $|\Psi_1\rangle$.* We know that the Qbits were in a product state (a product of single Qbits) $|\Psi_1\rangle = (H|0\rangle)^n(H|1\rangle)$ before passing through U_f. *Show that the Qbits are in a product state after passing through U_f as well (i.e. $|\Psi_2\rangle = \Pi_{0\leq i<n}|\phi_i\rangle$), and write this product explicitly. Is $|\Psi_2\rangle$ different from $|\Psi_1\rangle$? What are the measured values of the Qbits $x_0, \cdots, x_{n-1}$ after the final Hadamard operation?* (Hint: Perform the Hadamard operation on each term in the product above, and use the fact that $H^2 = \mathbb{1}$.)

You should have found that $x_0, \cdots, x_{n-1}$ are measured to be something other than all zeros. The Deutsch-Josza algorithm states that for any balanced function f, the probability of measuring $x_0, \cdots, x_{n-1}$ to be all zeros is 0. We'll prove this in the following way.

(i) *Show that for an arbitrary balanced function f,*

$$|\Psi_2\rangle = U_f|\Psi_1\rangle = \frac{1}{2^{\frac{n+1}{2}}} \sum_{0\leq\mathbf{x}<2^n} (-1)^{f(\mathbf{x})}|\mathbf{x}\rangle(|0\rangle - |1\rangle) \tag{2}$$

$$= \frac{1}{2^{\frac{n+1}{2}}}\left(\sum_{\mathbf{x}:f(\mathbf{x})=0}|\mathbf{x}\rangle - \sum_{\mathbf{x}:f(\mathbf{x})=1}|\mathbf{x}\rangle\right)(|0\rangle - |1\rangle)\,.$$

After the final Hadamard operation, show that the probability of measuring $x_0 = 0$, $x_1 = 0$, $\ldots$, $x_{n-1} = 0$, $y = 1$ is zero, i.e. $\langle\Psi_0|(H^{n+1}|\Psi_2\rangle) = 0$. (Hint: Rather than calculating $\langle\Psi_0|\Psi_3\rangle$, calculate the same quantity in the form $\langle\Psi_1|\Psi_2\rangle = (\langle\Psi_0|H^{n+1})|\Psi_2\rangle = \langle\Psi_0|(H^{n+1}|\Psi_2\rangle) = \langle\Psi_0|\Psi_3\rangle$, and use eqn 2.)

Hence with one application of the function f, with 100% certainty a constant function returns the initial state and a balanced function with 100% certainty will never return the initial state.

It is amazing that we could determine whether a DJ function f is constant or balanced in just *one* evaluation of U_f. The Deutsch-Josza algorithm achieves an *exponential* speedup over its classical counterpart. The problems considered in the above (Deutsch and Deutsch-Josza) algorithms may seem far removed from applications to real world problems, but these algorithms paved the way for more complicated and powerful algorithms.

Why are we still factoring 143? The great challenge in building quantum computer is *decoherence*, the tendency of Qbits to interact with the environment and go from quantum superpositions into mixtures.

7.3 **Solving Schrödinger: WKB, instantons, and the double well.** (Computation) ③

We study the problem of quantum tunneling in a symmetric double well potential. If the barrier between the two wells is large, the ground state and first excited state are well approximated as symmetric and antisymmetric superpositions of the ground states in the two separated wells. Indeed, the low-energy physics of the double-well system can be approximated by a two-level system (TLS), another example of a Qbit. In the symmetric case, in a basis where $\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ is the state localized in the left well and and $\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ is in the right well, the Hamiltonian is

$$H_{\mathrm{TLS}} = \begin{pmatrix} 0 & -\Delta \\ -\Delta & 0 \end{pmatrix}, \tag{3}$$

where Δ is called the tunneling matrix element.

(a) *What is the energy splitting for H_{TLS}? Calculate the time evolution operator $U(t) = \exp(-\mathrm{i}H_{\mathrm{TLS}}t/\hbar)$. If we start in the left well at $t = 0$, what is the probability being in the left well after time t?*

It is traditional and useful to study the double well given by the quartic polynomial

$$V(y) = V_0 \left(\frac{y}{Q_0} - 1\right)^2 \left(\frac{y}{Q_0} + 1\right)^2. \tag{4}$$

In our notation, this potential has minima at $\pm Q_0$ separated by a barrier of height V_0; a particle of mass m has small oscillation frequency ω near the minima in each of the two wells.

(b) *Calculate the barrier height V_0 algebraically in terms of m, Q_0, and ω, that sets the small oscillation frequency near $y = Q_0$ equal to ω.*

(c) *Calculate the classical instanton action (or, equivalently, the WKB exponent) $S_0 = \int_{-Q_0}^{Q_0} \sqrt{2mV(y)}\mathrm{d}y$ for our potential. Let $Q_0 = na_0$ with $a_0 = \sqrt{\hbar/2m\omega}$ the width of the ground state in the harmonic approximation. What is $S_0/\hbar$ in terms of n? What numerical value does it have for $n = 5$? How big is $\exp(-S_0/\hbar)$, the Euclidean action's suppression to the contribution of the instanton path?* (Hint: $S_0/\hbar$ is dimensionless. As suggested by the latter part of the exercise, for our potential it should depend only on the dimensionless number n, given that V_0 is given as in part (b).)

We shall assume our particle has the mass of a hydrogen atom, the small oscillation frequency in the left and right wells is 1THz, and $Q_0 = 5a_0$. Generate a grid of length $L = 20a_0$ with $N_p = 200$ points. Let the initial wavefunction $\psi[0] = \psi_0(x + Q_0)$ be the harmonic ground state in the left well. As in the coherent-state exercise, generate the appropriate arrays $U_{\text{pot}}(\mathrm{d}t/2)$ and $U_{\text{kin}}(\mathrm{d}t)$ to evolve ψ using your BCH formula, with $\mathrm{d}t = P/20$ where P is the period of the oscillator.

(d) *Solve for the time evolution over 1000 periods. Animate the probability density $|\psi(x,t)|^2$ versus time. Does it travel between the two wells periodically, as should be predicted by your answer to part (a)?*

(e) *Plot the probability $\int_{-\infty}^{0} |\psi(x,t)|^2$ that the particle is found in the left well.* (The high-frequency wiggles are due to the small components of higher eigenstates of our initial state in the left well.) *By eye, estimate the period of oscillation, and also Δ, within 10%.*[8]

(f) *Use a nonlinear least-squares method to fit your prediction from part (a) to the probability you calculated in part (e). Include your fit in the plot for part (e). What is Δ_{Fit}?* (If your answer from part (e) is off by much more than 30%, see if you can track down the problem.) *What is the ratio of your estimated energy splitting (using the formula from part (a)), to the energy splitting between states in one of the two wells (in the harmonic approximation)?* Quantum tunneling is one of the most important sources of low-energy, long-time behavior in physics.

The instanton formula for the tunnel splitting is closely related to the WKB formula.[9] Both are of the form $\Delta = \hbar\omega_0 \exp(-S_0/\hbar)$. In the WKB formula, the prefactor $\hbar\omega_0$ is given by a matching calculation; in the instanton method it is given by a path integral incorporating small oscillations about the instanton path.[10] Gildener and Patrascioiu (*Phys. Rev. D* **16**, *423, 1977*, referred to by Coleman) did the calculation explicitly for the quartic well using instanton methods, and got

$$\Delta \sim \hbar\omega\sqrt{\frac{6S_0}{\pi\hbar}}\exp(-S_0/\hbar) \tag{5}$$

so in our notation $\omega_0/\omega = \sqrt{6S_0/\pi\hbar}$.

(g) *Calculate your numerical estimate of ω_0/ω, and compare with Gildener and Patrascioiu's estimate.*

There is an interesting story about communication between high-energy and condensed-matter physics. Sidney Coleman (high-energy Harvard theorist) and Jim Langer (condensed-matter theorist on sabbatical at Harvard) were both working on barrier

[8] Nonlinear least-squares routines typically get lost if you don't start near the best fit.

[9] The two turning points for the classical path in the harmonic well are not far enough apart for the traditional WKB formula to be accurate; a suitable generalization accounting for that proximity does agree with the instanton formula.

[10] Coleman tells us the answer from the instanton calculation is given in terms of 'Wronskians'.

crossing at the same time. Coleman was stuck for months, because his instanton calculation was a factor of two higher than the WKB estimate (the 'double well done doubly well' appendix to his Les Houches lectures, calculating quantum fluctuations causing transitions through barriers). Unbeknownst to him, Jim Langer in a nearby office had solved the problem in the statistical mechanics context (critical droplet theory, calculating thermal fluctuations causing transitions over barriers). Eventually, Coleman realized as Langer had earlier that only half the fluctuations that reach the barrier actually cross it.