Metastability and Markov

(Sethna, “Entropy, Order Parameters, and Complexity”, ex. 8.21)

The hints below will help you solve parts (e) and (f) of this question, where we will numerically evaluate the slowest decaying mode and barrier crossing time by computing eigenstates of the "quantum" Hamiltonian.

In the Mathematica hints, we shall solve the differential equations directly. In the Python hints, we shall instead construct a Hamiltonian matrix by discretizing space into segments of length dx, and then finding the lowest eigenvalue of that Hamiltonian.

Construct the Hamiltonian for numerical calculations. Fill in the cubic potential and the expression for the effective quantum potential. We assume η=1 and $k_B T = 1/2$.

```
V[x_] := ...
Veff[x_, η_ : 1, kBT_ : 1 / 2] := ...
H[σ_, η_ : 1, kBT_ : 1 / 2] := -kBT / η σ''[x] + Veff[x, η, kBT] × σ[x]
```

Compute ground state eigenvalue and eigenfunction using NDEigenststem function

+/-x0 - finite boundary conditions for numerical evaluation

dx - discrete grid size

How should σ behave at the boundary?

```
eigenSystem[η_ : 1, kBT_ : 1 / 2, x0_ : 10, dx_ : 0.005, Nstates_ : 1] :=
 NDEigensystem[{H[...], DirichletCondition[σ[x] == ..., True]},
  σ[x], {x, ..., ...}, Nstates, Method → {"SpatialDiscretization" ->
     {"FiniteElement", {"MeshOptions" → {MaxCellMeasure → dx}}}}]
```

(e) For the cubic potential (eqn 7), numerically compute the eigenstate of the transformed diffusion equation with smallest eigenvalue. What does the eigenvalue predict for the lifetime? How nearly does it agree with the classic calculation of Kramers,
$\lambda_0 \approx (V_2 \tilde{V}_2)^{1/2}/(\pi\eta) \exp(-E/k_B T)$.

Compute the eigenvalue and eigenstate

```
{val, vec} = ... // Flatten
```

Plot the potential, notice where the well and barrier are located

```
Plot[..., {x, ...}, PlotRange → {...}, AxesLabel → {"x", "V(x)"}]
```

Compare escape time from eigenvalue to analytical result. We compute the percent difference between

numerical and analytical result.

```
τNumerical = 1 / (...)
τKramers = 1 / (Sqrt[... * ...] / (Pi * ...) Exp[- ... / (...)])
diff = 100 Abs[...] / τNumerical;
Print[diff, "% difference"]
```

(e) ... What does the eigenvalue predict for the lifetime? How nearly does it agree with τ from the Kramers calculation?

(f) Using the corresponding eigenstate ρ_0, plot the slowest decaying mode $\rho_0(x) = (\rho^*)^{1/2}\,\sigma_0$, normalized to one, along with the Boltzmann distribution $\rho^*(x)/Z$ and the Boltzmann probability distribution in the approximation that the well is quadratic. Explain how ρ_0 differs from the quadratic approximation, and why this is physically sensible. Explain how ρ_0 differs from ρ^*, and how it resolves an important question about how to determine the metastable probability 'in the well'.

Due to numerical errors your slowest decaying mode will blow up at one of the boundaries (the numerical eigenstate doesn't exactly cancel the blow up from the Boltzmann distribution). Choose a cutoff to restrict the normalization calculation to [-xLim, xLim]

```
boltz = Exp[...];
mode = Sqrt[...] vec;
xLim = ...;
Plot[{mode}, {x, -xLim, xLim}, PlotRange → {0, 3},
 PlotLegends → {"ρ₀"}, AxesLabel → {"x", "Density"}]
norm = NIntegrate[mode, {x, -xLim, xLim}];
```

Adjust Z manually so that the Boltzmann distribution best matches the slowest decaying mode inside the well or approximate Z by normalizing over a region near the potential well. If you choose the later option, you will want to restrict the normalization to the domain [xMin, xMax] to integrate over the well (see plot of the potential above)

```
xMin = ...;
xMax = ...;
Z = NIntegrate[..., {x, ..., ...}];
```

Compute the Boltzman distribution in the approximation that the well is quadratic

```
Vquadratic[x_] := ...
```

```
boltzQuadratic = Exp[...];
ZQuadratic = Integrate[..., {x, ...}];
```

(f) ... Plot the slowest decaying model, the Boltzmann distrubion and the Boltzman distribution in the approximation that the well is quadratic

```
In[•]:= Plot[{V[x], mode / norm, boltz / Z, boltzQuadratic / ZQuadratic}, {x, -xLim, xLim},
 PlotRange → {-1, 1}, PlotLegends → {"Potential", "ρ₀", "ρ*", "Quadratic Well"},
 FrameLabel → {"x", "Density"}, Frame → True, Axes → False]
```