Problem 3

The first step in designing the grid was to find an appropriate integration range. The integral of interest is

$$Z = \int_{\mathbb{R}^2} e^{-\beta U(x,y)} dxdy \,.$$

The exponent will blow up away from the origin, so we need to find a range such that the integrand becomes essentially zero. The smallest β that we will be plotting is $\beta = 0.01$. If we do a contour plot of the integrand for this smallest β we can find a safe range to perform our numerical integration for all β. Figure 1 indicates that a safe domain of integration would be $-5 \leq x \leq 3$ and $-3 \leq y \leq 5$.

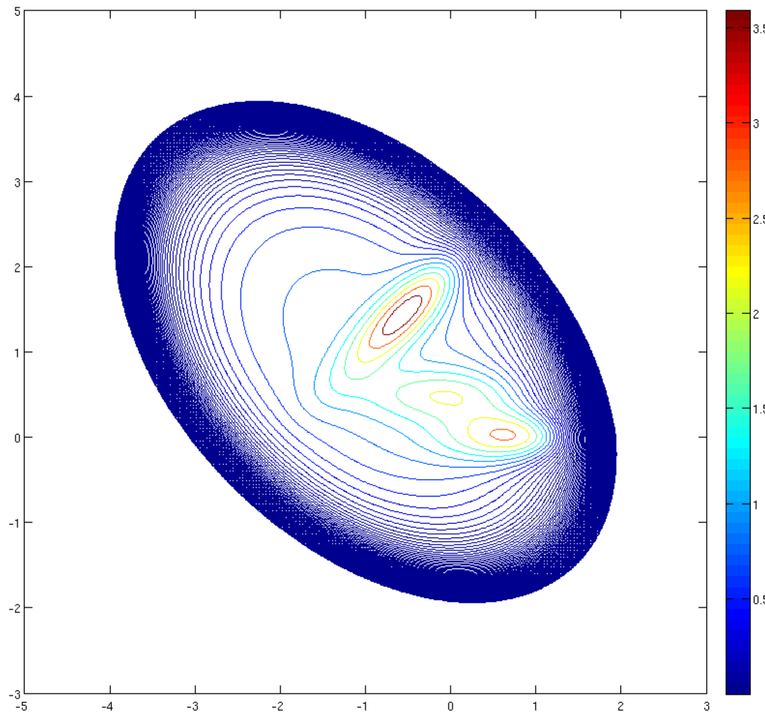

Figure 1: Contour plot of $\exp(-\beta U(x,y))$ for $\beta = 0.01$

The next task is to choose an appropriate numerical quadrature method. Ideally, we would like to concentrate our efforts on the part of the function that is changing more rapidly, but I decided to go with a uniform refinement Riemann integration for its simplicity and predictable convergence properties. I discovered that the interesting range of β was $0.01 \leq \beta \leq 1$. Running a couple of resolution tests at the extremes: for $\beta = 0.01$ and a 50×50 grid $\frac{-1}{\beta}\log(Z) = -236.0067$. At double the resolution, I got -238.0375, so the first two digits are accurate as desired. For $\beta = 1$ the results at 50×50 and 100×100 were -145.8771 and -146.1914, respectively. So we can conclude that a 50×50 grid is accurate enough.

We encounter another numerical difficulty when attempting to evaluate our integral. The Muller potential has a minimum value of approximately -145, so when we evaluate $e^{-\beta(-145)} = e^{145\beta}$, for moderate β, our integrand blows up and approaches the machine overflow, losing a lot of accuracy in the process. We can perform a little numerical trick as follows

$$\begin{aligned}
F &= \frac{-1}{\beta}\log\left(\int e^{-\beta U(x,y)}dxdy\right) \\
&= \frac{-1}{\beta}\log\left(\int e^{-\beta U_{min}}e^{-\beta(U(x,y)-U_{min})}\right) \\
&= \frac{-1}{\beta}\log\left(e^{-\beta U_{min}}\int e^{-\beta(U(x,y)-U_{min})}\right) \\
&= U_{min} + \frac{-1}{\beta}\log\left(\int e^{-\beta(U(x,y)-U_{min})}\right),
\end{aligned}$$

where U_{min} is some offset on the order of the minimum potential which makes the integral easier to evaluate numerically. For our purposes, we use $U_{min} = -145$, which appears to produce good results for a variety of β.

The calculation time is, of course going to depend heavily on the chosen mesh resolution and number of plot points. With our 50×50 grid, each integration takes an average of 0.05 seconds. With 50 plot points, this comes out to a total runtime of approximately 2.5 seconds.

Figure 2 plots the free energy $F = -\beta^{-1}\log(Z)$ as a function of β.

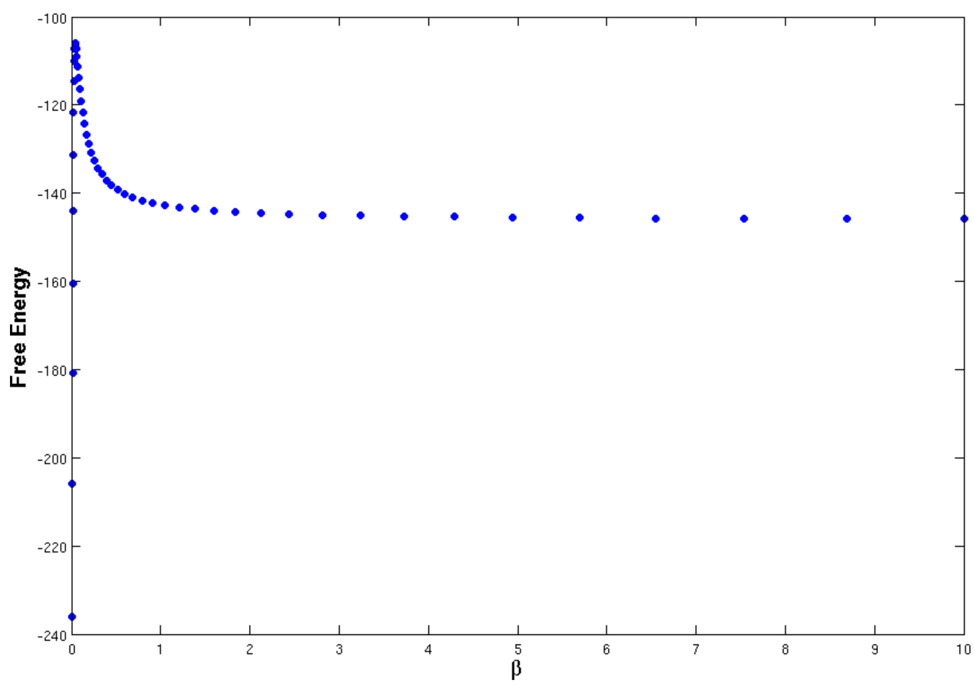

Figure 2: Free energy versus β

PlotMuller.m

```
% Written by Truman Ellis
clear all; close all; clc;

% Number of beta's to plot
nBeta = 50;
beta = logspace(-2, 1, nBeta);
% Number of Integration Points
nx = 50;
ny = 50;
% Integration Domain
xMin = -5;
xMax = 3;
yMin = -3;
yMax = 5;
% Calculate grid size
hx = (xMax - xMin)/nx;
hy = (yMax - yMin)/ny;
dA = hx*hy;
% Construct mesh
x = linspace(-5, 3, nx);
y = linspace(-3, 5, ny);

% Define offset to make integration easier
Umin = -145;

% Start quadrature sum
dU = zeros(1,nBeta);
% Loop over different beta's
for nb = 1:nBeta
    % Integrate over the domain
    for i = 1:nx
        for j=1:ny
            % Sum differential volume
           dU(nb) = dU(nb) + exp(-beta(nb)*(muller(x(i), y(j))-Umin))*dA;
        end
    end
end
% Calculate free energy: F = -1/beta*log(Z)
F = Umin + -1./beta.*log(dU);

% Plot results
plot(beta,F,'o','MarkerFaceColor','blue','MarkerSize',5)
xlabel('\beta','FontSize',14)
ylabel('Free Energy','FontSize',14,'FontWeight','demi')
```