Escape rate for a Brownian particle in a radial cubic spline trap

Marc Meléndez

December 8, 2020

Brownian motion has many applications in mathematics, physics and finance, but it was originally conceived as the description of small micron-scale particles suspended in water, which vibrate due to thermal fluctuations. Here, we are interested in a Brownian particle that moves around in two dimensions and feels the force due to a particular type of potential energy trap. We aim to describe the time it takes for such a particle to escape from the trap.

1 Cubic spline trap

Let r represent the distance between the position of a Brownian particle and the centre of a cubic spline trap, and let $V(r)$ represent the potential energy of such a particle (see Fig. 1). We choose $V(r)$ in such a way that the depth of the trap equals E_0, its radius is R and, that it lies flat at the origin and at $r = R$ (so $V'(0) = V'(R) = 0$).

$$V(r) = \begin{cases} E_0 \left(-2\left(\frac{r}{R}\right)^3 + 3\left(\frac{r}{R}\right)^2 - 1\right), & \text{for } r < R, \\ 0, & \text{for } r \geq R. \end{cases} \tag{1}$$

A trapped particle, then, feels a force

$$\mathbf{F}(\mathbf{r}) = -\nabla V = \frac{6}{R^2}\left(\frac{r}{R} - 1\right)\ \frac{\mathbf{r}}{R}. \tag{2}$$

We wish to determine the rate at which a trapped Brownian particle escapes from the trap. Dimensional analysis reveals that the rate μ equals

$$\mu = \frac{D}{R^2}\ f\left(\frac{E_0}{k_B T}\right), \tag{3}$$

where D stands for the diffusion coefficient, $k_B T$ the thermal energy, and f for an as yet undetermined function.

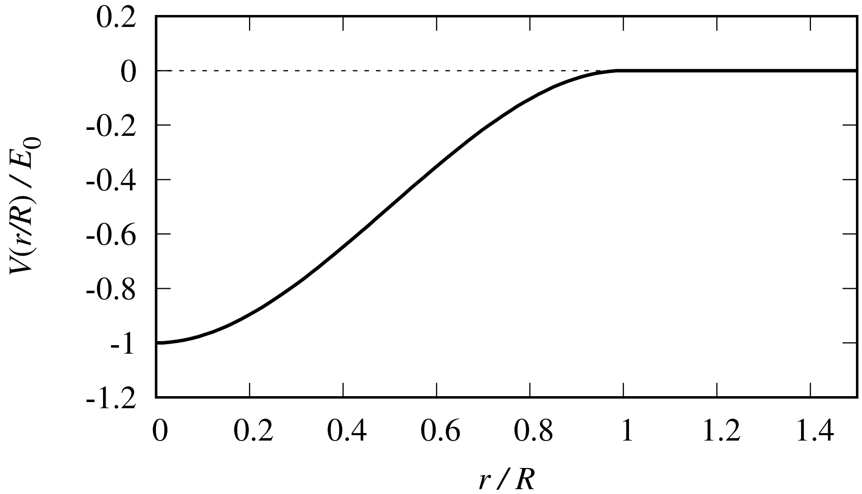

Figure 1: Radially symmetrical cubic spline trap of depth E_0.

2 Numerical simulation

We can check our predictions by setting up a numerical simulation of the process process. A minimal C program could accept E_0, R, D and $k_B T$ as parameters.

```
/* Usage message */
if(argc < 6) {
  fprintf(stderr, "Usage: %s <trap_depth> "
                  "<trap_radius> "
                  "<diffusion_coefficient> "
                  "<thermal_energy>"
                  "<dim>\n",
          argv[0]);
  return 0;
}
```

Following the usage message, we declare the parameters and variables needed for the Brownian simulation. The jagged trajectory of Brownian motion results from integrating the (Itō) stochastic differential equation

$$d\mathbf{r} = M\mathbf{F}(\mathbf{r})\;dt + \sqrt{2D}\;d\mathbf{W}, \tag{4}$$

where $M = D/k_B T$ stands for the particle mobility, $-\nabla V$ is the force due to our cubic spline trap and $d\mathbf{W}$ is the two-dimensional Wiener process.

```
/* Simulation parameters */
```

```c
double E0 = atof(argv[1]); /* Trap depth */
double R = atof(argv[2]);  /* Trap radius */
double D = atof(argv[3]);  /* Diffusion coefficient */
double kT = atof(argv[4]); /* Thermal energy */
int dim = atoi(argv[5]);   /* Dimensionality */
double M = D/kT;           /* Mobility */
double dt = 0.0001;        /* Time step */

/* Variable declarations */
int nruns = 10000;     /* Number of runs */
int run;               /* Current run */
int i;                 /* Coordinate index */
double q[dim];         /* Particle position */
double F[dim], Fmod;   /* Force and modulus of force*/
double r;              /* Distance to origin */
double r2;             /* Distance squared */
long int tstep;        /* Current step */
long int sumtime = 0;  /* Sum of times */
long int sumt2 = 0;    /* Sum of times squared */
```

The `nruns` realisations of the stochastic process proceed by setting the Brownian particle at the origin of coordinates and then letting it diffuse until it reaches the edge of the trap at $r = R$. An Euler-Maruyama scheme integrates the equations of motion. `Gaussian(0,1)` produces a random number from a normal distribution with null mean and unit standard deviation.

```c
/* nruns realisations of the stochastic process */
for(run = 0; run < nruns; run++) {
  /* Reset position */
  for(i = 0; i < dim; ++i) q[i] = 0.0;

  /* Run Brownian dynamics until particle escapes */
  for(tstep = 0; 1; tstep++) {
    /* Particle position */
    r2 = 0; for(i = 0; i < dim; ++i) r2 += q[i]*q[i];
    r = sqrt(r2);

    /* Break when particle leaves the trap */
    if(r >= R) break;

    /* Force vector */
    Fmod = 6.0*E0*(r/R - 1)/(R*R*R);
    for(i = 0; i < dim; ++i) F[i] = Fmod*q[i];

    /* Euler-Maruyama scheme */
```

```
    for(i = 0; i < dim; ++i)
      q[i] += M*F[i]*dt + Gaussian(0,1)*sqrt(2*D*dt);
  }

  sumtime += tstep;
  sumt2 += tstep*tstep;
}
```

The code ends by outputting the results, calculating the inverse of the mean time it took the Brownian motion to escape from the trap.

```
float meantime = (sumtime*dt)/((float) nruns);
float mu = 1.0/meantime;

printf("# E0 \t\t R \t\t D \t\t kT \t\t "
       "mu \t\t error(mu)\n");
printf("%f\t%f\t%f\t%f\t%f\t%f\n", E0, R, D, kT, mu,
        mu*mu*sqrt((sumt2*dt*dt)/((float) nruns)
                - meantime*meantime)/sqrt(nruns));
```

3 Numerical results

The first very obvious test of the code verifies that μ is indeed proportional to D and inversely proportional to R^2, as stated by Eq. (3). Fig. 2 shows that the simulations agree with this prediction, although smaller values of R depart from the curve, probably due to issues with the integration time step.

By setting D/R equal to one and plotting μ for different values of $E_0/(k_BT)$, we are in effect tracing the shape of function f in Eq. (3). Deep traps seem to satisfy

$$\mu \propto \left(\frac{E_0}{k_BT}\right)^2 e^{-\frac{E_0}{k_BT}} \tag{5}$$

(see Fig. 3).

4 Theoretical considerations

The final step in this small project investigates the shape of function f in Eq. (3) following the classical approach by Kramers [1]. First we imagine a one-dimensional setting with a potential symmetrical about the origin $r = 0$, so that $V(-r) = V(r)$. Simply applying Kramer's formula for the escape rate

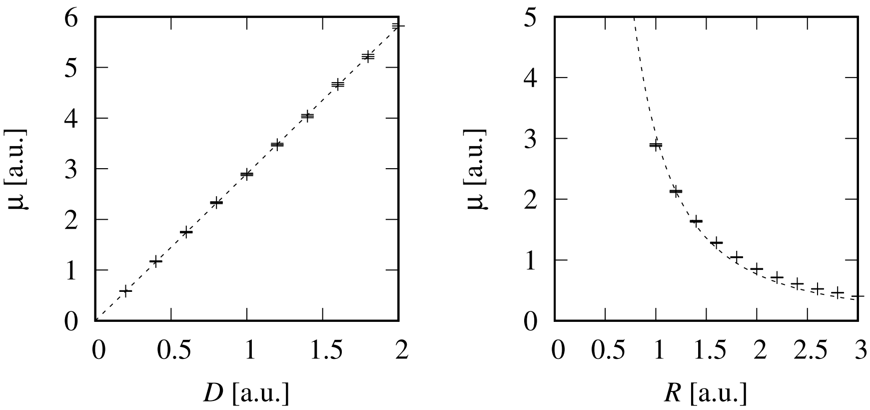

Figure 2: Escape rate mu versus diffusion coefficient D (*left*) and trap radius R (*right*) confirming the relation predicted by Eq. (3). The dotted lines follow fits with the theoretical tendencies.

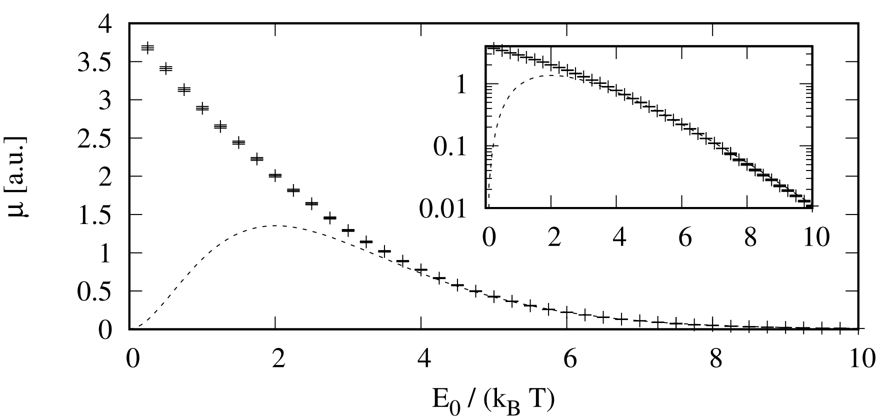

Figure 3: Escape rate versus $\frac{E_0}{k_B T}$ for $\frac{D}{R^2} = 1$. Points represent simulation results. The dotted line follows $\frac{5}{2}\left(\frac{E_0}{k_B T}\right)^2 e^{-\frac{E_0}{k_B T}}$. The inset plots the same data points, but with a logarithmic scale on the vertical axis.

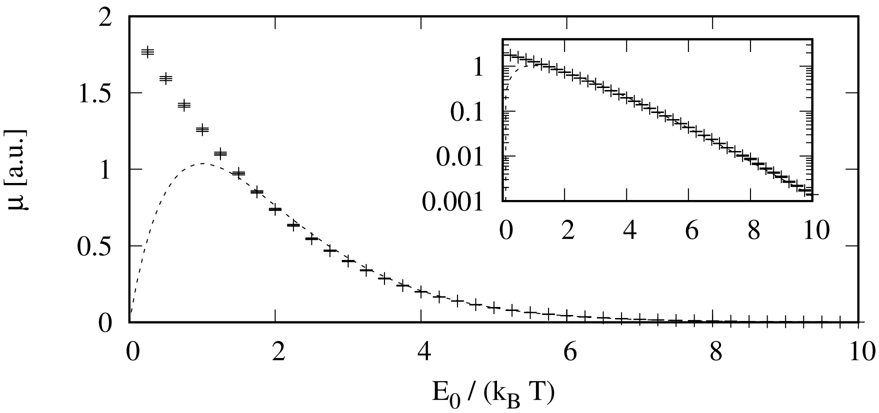

Figure 4: Escape rate versus $\frac{E_0}{k_B T}$ for $\frac{D}{R^2} = 1$ in the one-dimensional version of the problem. Simulation results are drawn with points. The dotted line follows $2.82\ \left(\frac{E_0}{k_B T}\right) e^{-\frac{E_0}{k_B T}}$. The inset shows the points on a logarithmic scale on the vertical axis.

(taken from [2] without much reflection) gives

$$\begin{aligned}\mu &= 2\frac{D}{2\pi k_B T}\sqrt{V''(0)\ |V''(R)|}e^{-\frac{E_0}{k_B T}} \\ &= \frac{6DE_0}{\pi R^2 k_B T}e^{-\frac{E_0}{k_B T}}.\end{aligned} \tag{6}$$

The extra factor of two in front comes from the two exits from the trap (at $r = R$ and $r = -R$). Simulations suggest this factor should in fact be three instead. Perhaps this emerges from a discontinuity of V'' at $r = R$? Whatever the reason, the figure confirms a decay proportional to $\frac{E_0}{k_B T} \exp\left(-\frac{E_0}{k_B T}\right)$ (Fig. 4).

References

[1] H. A. Kramers, *Brownian motion in a field of force and the diffusion model of chemical reactions*, Physica **7**, 4, 284–304 (1940).

[2] `https://home.icts.res.in/~abhi/notes/kram.pdf`.