1 Coding (Ranking of the pages).

(a) Solve a PageRank problem with Monte Carlo and make a performance plot.

(b) Solve a PageRank problem with Richardson iteration $\boldsymbol{x} = \alpha \boldsymbol{P}\boldsymbol{x} + (1-\alpha)\boldsymbol{v}$ and make a performance plot.

2 Coding (Diffusing around).

Use Monte Carlo to solve an elliptic boundary value problem like the one presented in class. You do not need to solve the problem globally, just approximate the solution as well as you can at a single interior point.

3 Coding (How long can you avoid yourself?).

(a) Simulate self-avoiding walks by using any method. How long can you go? What is your best estimate for the number of self-avoiding walks?

(b) Simulate self-avoiding walks by using Rob's favorite method: the Rosenbluth & Rosenbluth method with resampling at every 10th time step. Now how long can you go? What is your best estimate for the number of self-avoiding walks?

4 Choose-your-own-adventure (Lemmings revisited).

What is the probability that a lemming, starting at rung 1 of a 50 rung ladder, will make it to the top before falling to the ground? Recall that a lemming has a half chance of climbing to a higher run and a half chance of falling to the ground (rung 0), and these probabilities remain the same for all the rungs. Write down a clever Monte Carlo scheme to find the answer, and potentially implement it.

5 Math (Counting SAWs)

.
Recall that Z_t is the number of self-avoiding walks (SAWs) of length t. Prove that $\lim_{t\to\infty} \log Z_t/t$ exists, and prove the best error bounds you can obtain.

6 Math (Matrix inversion error bounds).

In class (and in the notes) we derived a general variance bound $\mathbb{E}\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2 \le \|\boldsymbol{b}\|_1^2\, \mathbb{E}\tau^2$ for the Monte Carlo matrix inversion method. Let's refine this bound for particular problems.

(a) What is the best variance bound for Monte Carlo applied to the elliptic boundary value problem? Your answer should only depend on the norm $\|f\|_\infty$ for the boundary data and the number of particles N.

(b) What is the best variance bound for Monte Carlo applied to the PageRank problem?

7 Math (Optimal rare event sampling).

Consider the forward flux sampling scheme described in class and in the notes. We want to calculate the probability that a standard Brownian Motion $(X_t)_{t\geq 0}$ starting from $X_0 = 0$ hits a high value $X_t = K$ before hitting $X_t = -1$. We introduce a sequence of thresholds $0 = \lambda_0 < \lambda_1 < \cdot < \lambda_L = K$. We run an ensemble of N independent Brownian motions starting from each threshold λ_i until hitting the next threshold λ_{i+1} or hitting -1, and let N_{i+1} denote the number of successes. Last, we estimate $\mathbb{P}\{X_t = N \text{ before } X_t = -1\}$ using the product $\hat{p} = N_1 N_2 \cdots N_L / N^L$.

Write down the variance for this method and compute the optimal thresholds (Hint: use the fact that a Brownian motion starting from $X_t = 0$ hits $+A$ before hitting $-B$ with probability $B/(A+B)$). What is the best variance as we take $L \to \infty$?

Partial solution: Using the hint, the solution to the problem is

$$p = \mathbb{P}\{X_t = K \text{ before } X_t = -1 \,|\, X_0 = 0\} = \frac{1}{K+1}.$$

Next, we define the probability of making it from threshold λ_i to threshold λ_{i+1} before falling to the level -1:

$$p_i = \mathbb{P}\{X_t = \lambda_{i+1} \text{ before } X_t = -1 \,|\, X_0 = \lambda_i\} = \frac{\lambda_i + 1}{\lambda_{i+1} + 1}.$$

We observe that each N_i is binomially distributed with mean Np_i and variance $Np_i(1-p_i)$. Hence, we compute

$$\mathbb{E}\,\hat{p} = \prod_{i=1}^{L} p_i = p, \qquad \mathbb{E}\,|\hat{p}|^2 = \prod_{i=1}^{L} p_i^2 \left(1 + \frac{1/p_i - 1}{N}\right) = p^2 \prod_{i=1}^{L} \left(1 + \frac{1/p_i - 1}{N}\right).$$

The variance for the method is

$$p^2 \prod_{i=1}^{L} \left(1 + \frac{1/p_i - 1}{N}\right) - p^2.$$

To optimize the thresholds, we set $\nu_i = \log(1/p_i)$ for $1 \leq i \leq L$. We need to find ν_i values that minimize

$$\prod_{i=1}^{L} \left(1 + \frac{e^{\nu_i} - 1}{N}\right)$$

or equivalently minimize the function

$$f(\nu_1, \ldots, \nu_L) = \sum_{i=1}^{L} \log\left(1 + \frac{e^{\nu_i} - 1}{N}\right) = \sum_{i=1}^{L} \left(\log(N - 1 + e^{\nu_i}) - \log N\right).$$

We also need to enforce the constraints that each $\nu_i \geq 0$ and $\sum_{i=1}^{L} \nu_i = \log(K+1)$. We explicitly calculate

$$\partial^2_{\nu_i \nu_i} f(\nu_1, \ldots, \nu_L) = \partial^2_{\nu_i \nu_i} \log(N - 1 + e^{\nu_i}) = \partial_{\nu_i} \left(\frac{e^{\nu_i}}{N - 1 + e^{\nu_i}}\right) = \frac{(N-1)e^{\nu_i}}{(N - 1 + e^{\nu_i})^2} > 0$$

and $\partial^2_{\nu_i \nu_j} f(\nu_1, \ldots, \nu_L) = 0$ for $i \neq j$. Therefore, the second derivative matrix $\nabla^2 f(\nu_1, \ldots, \nu_L)$ is positive definite, so f is convex. We are minimizing a convex function, symmetric in all of its inputs, subject to linear constraints. The solution is to set all the inputs equal to one another. Consequently, we find that each $p_i = (K+1)^{-1/L}$ and

$$\lambda_i = (K+1)^{i/L} - 1, \qquad i = 0, 1, \ldots, L.$$

The resulting variance of forward flux sampling is

$$p^2 \left[\left(1 + \frac{(K+1)^{1/L} - 1}{N} \right)^L - 1 \right].$$

Taking $L \to \infty$, we find that

$$p^2 \left[\left(1 + \frac{(K+1)^{1/L} - 1}{N} \right)^L - 1 \right] \downarrow p^2 \left[(K+1)^{1/N} - 1 \right] \geq p^2 \frac{\log(K+1)}{N}.$$

We can never escape from the fundamental $\mathcal{O}(1/N)$ Monte Carlo variance scaling. Yet, we can do much better than the direct Monte Carlo simulation strategy, which gives variance

$$\frac{p(1-p)}{N} = p^2 \frac{1/(K+1) - 1}{N}.$$