

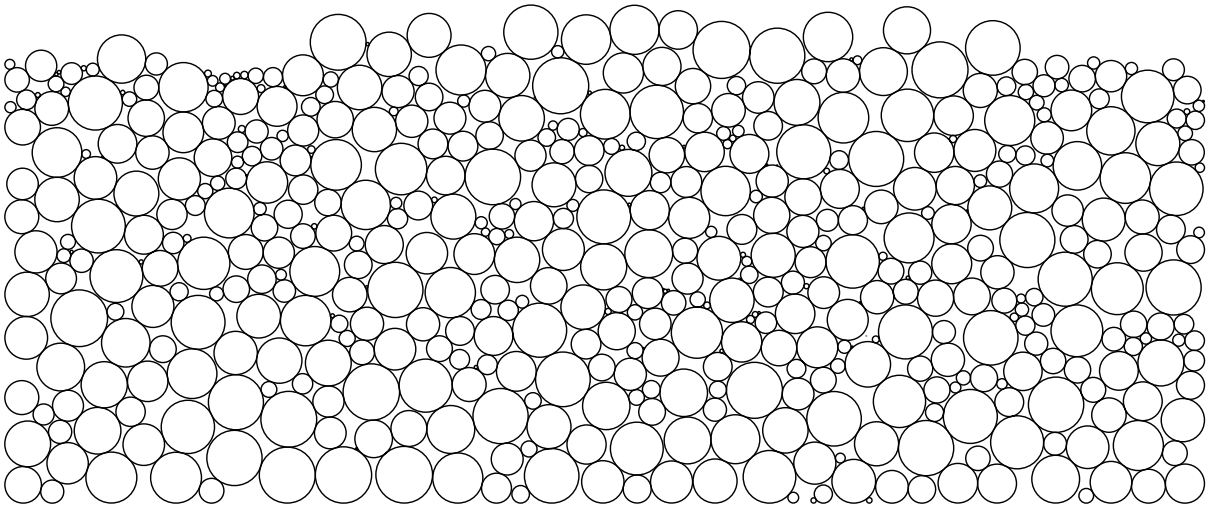
## Physics 420/580: Assignment 5

Due: Friday, December 2, 2011

Packing is a good example of a difficult optimization problem. Here we consider  $N$  discs in a plane and ask what configuration *with no discs overlapping* minimizes the gravitational energy

$$E = \sum_{i=1}^N m_i y_i = \sum_{i=1}^N R_i^2 y_i.$$

Each disc is centred on the coordinate pair  $(x_i, y_i)$  and has a radius  $R_i$ , drawn independently from some distribution, and a mass  $m_i = R_i^2$ . It's hard to find the best packing arrangement because the phase space  $\mathbb{R}^{2N}$  grows exponentially in the number of discs, and there are many low-lying configurations that are close in energy to the global minimum.



We'll assume that the radii are randomly distributed according to

$$p(R) = \frac{1}{D} R^{-1+1/D} \theta(1-R) \theta(R).$$

A powerlaw form is what you might find for the distribution of particle sizes in gravel or soil. The disorder strength  $D \geq 0$  controls the average values of the radius and mass:  $\bar{R} = \int dR p(R) R = 1/(D+1)$ , and  $\bar{m} = \int dR p(R) R^2 = 1/(2D+1)$ . In the strong-disorder limit ( $D \rightarrow \infty$ ), the distribution approaches  $p(R) \sim 1/R$ . In the uniform limit ( $D \rightarrow 0$ ), the radii are all the same:  $p(R) \rightarrow \delta(R-1)$ . To select  $R_i$  values, we construct the cumulative probability distribution

$$\mathcal{P}(R) = \int_0^R dR' p(R') = R^{1/D} = \xi,$$

and associate it with a random number  $\xi \in [0, 1]$ . Inverting this equation gives  $R := \xi^D$ .

We're going to try to optimize the packing with simulated annealing. The strategy will be to attempt to move each disc along an arbitrary direction  $\hat{\mathbf{n}} = (\cos \theta, \sin \theta)$  some distance up to the next point of collision. It's convenient to parameterize the updated disc position as  $\mathbf{r}_i(\lambda) = \mathbf{r}_i + \lambda \hat{\mathbf{n}}$ . Thus, if we're moving disc  $i$ , then we need to determine for each  $j \neq i$  the  $\lambda$  values that lead to the collision  $|\mathbf{r}_i(\lambda) - \mathbf{r}_j| = R_i + R_j$ . Convince yourself that the solution is one of

$$\lambda_{1,2} = -\hat{\mathbf{n}} \cdot (\mathbf{r}_i - \mathbf{r}_j) \pm \sqrt{[\hat{\mathbf{n}} \cdot (\mathbf{r}_i - \mathbf{r}_j)]^2 + (R_i + R_j)^2 - (\mathbf{r}_i - \mathbf{r}_j)^2}.$$

In a similar way, we can also keep track of the  $\lambda$  values for intersection with container walls. If we take  $\lambda^{(+)}$  to be the smallest of all the positive collision values and  $\lambda^{(-)}$  to be the largest of all the negative ones, then the range  $\lambda^{(-)} < \lambda < \lambda^{(+)}$  will correspond to the full range of valid new disc positions. We suppose that the system is held in contact with a heat bath at temperature  $T$ . Each unbiased move should be accepted with probability  $\min(1, e^{-\Delta E/T})$ , where  $\Delta E$  is the gravitational energy gained or lost from the change  $\Delta y = \lambda \sin \theta$  in the disc's vertical position. Alternatively, we can accept with probability 1 moves drawn from the distribution  $\min(1, e^{-\Delta E/T})$ .

1. (10 points) Unpack the `Assignment5.tar.gz` archive, `cd` into the `Assignment5` directory, and compile the provided code by invoking `make`. The programs `packing` and `packing_openGL` take two arguments, corresponding to the width  $L$  of the system and the disorder  $D$ , and an optional argument that allows you to seed the random number generator.

```
$ make
$ ./packing_openGL
usage: packing_openGL (linear size) (distributional disorder) [seed]
$ ./packing_openGL 41 0.25
```

In the graphical version of the program, you can click and drag the mouse in the main window to change the temperature of the simulation. Try repeatedly heating and cooling the system. Observe what happens when you cool slowly versus quickly.

You'll notice that even though we've input  $D = 0.25$  at the command line, all the discs are the same size. Modify the function `initialize` in the file `update.cpp` so that each disc radius is drawn from the correct distribution. Be sure that the energy `E` and the total mass `totalM` are properly accumulated.

It is also the case that the discs are confined only by a floor at  $y = 0$ . Make appropriate changes to the `update` function so that the discs also experience hard walls at  $x = 0$  and  $x = L$ . The width is provided in the code as `double L`.

2. (10 points) Organize the (nongraphical) code in `packing.cpp` to implement simulated annealing and devise an effective cooling schedule. Report the lowest energy you manage to find for the following initial parameters.

```
$ ./packing 41 0.75 100
$ evince config.eps    ## Linux
$ open config.eps      ## Mac
```

Include the corresponding (automatically-generated) `config.eps` file along with your assignment submission.

3. (10 points) Make a plot of the optimal energy averaged over many disc realizations. You should compute the energies both per disc and per unit mass and over a range of disorder values up to  $D = 2$ . You might want to use the `gather` program you wrote in the last assignment to compute the error bars.