Documentation for RCP Generator

October 7, 2009

Contents

1 Background

For information about how the algorithm works or definitions of specific variables I recommend that you read "Random Close Packing of Disks and Spheres in Confined Geometries" by K. W. Desmond & E. R. Weeks which can be found on ArXiv, document number 0903.0864.

2 Installation

Within the package downloaded there will be the source code and a makefile. To compile the code open a terminal and change to the directory containing the source code. Simply type **make** into the terminal and press enter. You may get some warnings, but you can ignore them. **Note:** In order to compile the code you will need g++ and the path must be set to so that g++ can be executed from the current directory.

After running the makefile, there should be a file called rcpgenerator.exe. This file will be the executable you will use to generate rcp configurations.

This code has been successfully compiled on the following operating systems: Mac OS X 10.4, Ubuntu, and Centos 4.5.

3 Using the code

3.1 Configuration File

In order to generate an rcp configuration the program will need to know the parameters of the system. For instance, what boundary conditions would you like, how many small particles and big particles do you want, etc? All this information must be supplied to the program via a configuration file. The configuration file can be named anything you like and is nothing more than a text file where each line of the file contains pertinent system information. A configuration file will look something like the following. (Ingore the Line statements, they are only meant to signify the line number within the text file.)

Note: The following configuration file shown below has been included with the source code. The file name is ConfigInfo.txt

Line 1:	2	# of dimensions
Line 2:	100	total number of particles
Line 3:	0.25	starting volume fraction
Line 4:	0.025	initial volume fraction step size
Line 5:	1.67	size ratio
Line 6:	0.5	percentage of small particles
Line 7:	0 1 0 1 0 1	box size
Line 8:	1 1 0	boundary conditions
Line 9:	1	flag for fixing the y in 2D or z in 3D
Line 10:	0.9999	$\epsilon_{\delta r}$
Line 11:	1e-07	ϵ_E
Line 12:	754	seed number for generating random configurations

Line 1: # of dimensions
Specifies the dimensions of the system.

Line 2: total number of particles
Specifies the total number of particle in the system. This the number of small particles plus the number of big particles.

Line 3: starting volume fraction
When the system is initialized each particle is assigned a radii such that the volume fractions is equivalent to the amount specified in this line.

Line 4: initial volume fraction step size
After all particles are assigned coordinates and radii these particles begin a process of expansion and contraction. This lines specifies the initial rate of volume fraction expansion per iteration.

Line 5: size ratio
This number specifies the ratio of the big particle radius to the small particle radius.

Line 6: percentage of small particles
This number specifies the percentage of small particles. For instance, if the total number of particles is 1000 and you set this line to be 0.7, then you will have 700 small particles and 300 big particles.

Line 7: box size
This line contains four numbers for 2D and six numbers for 3D specifying the corners of the box containing the particles. The six numbers are $x_{\min}$ $x_{\max}$ $y_{\min}$ $y_{\max}$ $z_{\min}$ $z_{\max}$, where min and max indicate the two corners of the box. For instance, if this line is set to
0 0.5 0.1 0.8 0.2 1.5, then the lower corner of the box is located at (0, 0.1, 0.2) and the upper corner is located at (0.5, 0.8, 1.5).

Line 8: boundary conditions
This line specifies if the boundary is to be periodic or not. 0 indicates a hard fixed boundary and 1 indicates a periodic boundary. This line requires 2 numbers in 2D and 3 numbers in 3D, where each number is to indicate if the boundary along either the x, y, or z direction is fixed or periodic. For instance if this line is set to 0 1 1, then the x direction has a hard fixed boundary and the y and z directions have periodic boundaries.

A circular 2D boundary condition can be implemented by setting the first number equal to -1. The diameter of the circular boundary is set by the second number in Line 7. Since the actual diameter is meaningless, I recommend that you use a diameter of 1.

Line 9: flag for fixing the y in 2D or z in 3D
This lines specifies whether or not one boundary wall can float so as to fix the normalized height h. Note h is defined to be the box width along the y direction normalized by the small particle diameter. Setting this line to 1 means that the position of the upper y boundary remains fixed. If the this line is set to 0, then the upper y boundary changes with each iterations so that h remains constant. When this line is set to 0, h will be fixed at the value $y_{\max}$ which is set in Line 7. Note that when using this feature the lower corner of the box in Line 7 should be set to (0, 0, 0). **Also h (or $y_{\max}$) must be greater than the twice the size ratio, otherwise the code will not function properly.**

Line 10: ϵ_r
This line specifies the tolerance in the minimum overlap to terminate an iteration.

Line 11: ϵ_E
This line specifies the tolerance in the minimum overlap to terminate an iteration.

Line 12: seed number for generating random configurations
This program employs the Mersenne Twister algorithm to randomly initialize the position of particles. This line specifies the seed value used to initialize the Mersenne Twister routine.

3.2 Generating an rcp state

To execute the code type the following into the terminal
./rcpgenerator -f ConfigInfo -o FolderName

Required Flags
-f: Name of configuration file
-o: Directory to output rcp configuration

Optional Flags
-v: Verbose mode. This mode will update to the screen the total iteration count and current packing fraction.

-i: If this flag is set, then each configuration after an energy minimization step will be saved to the save directory indicated by the -o flag. Note that the first line in the output file is the radii of the particles, the second line is the box size, and the following lines are the coordinates and radius type.

3.3 Output

Two files will be saved after the program has successfully finished generating an rcp configuration. The first file, labeled FinalConfig, will be the coordinates of each particle, and the second file, labeled system, will contain all the system info.

FinalConfig file format
Below is a table illustrating the data saved in the FinalConfig file.

Column 1	**Column 2**	**Column 3**	**Column 4**
x-coord	y-coord	z-coord (only for 3D)	radius type

The radius type is an integer for referencing the radius for a given particle. For a binary system there are two radii and hence there are two radius types, labeled 1 and 2. The actual radius for particles of type 1 or type 2 can be found in the system file.

system file format
The system file consist of 12 lines specifying the details of the system. Below is an explanation of each line.

Line 1: Date file was created
Line 2: Time file was created
Line 3: Number of particles
Line 4: Number of dimensions
Line 5: Size ratio of big particle radius to small particle radius
Line 6: Percentage of small particles
Line 7: Radius of type 1 particles
Line 8: Radius of type 2 particles
Line 9: Corners of box containing particles
Line 10: Boundary conditions (P for periodic and C for Confined)
Line 11: Packing fraction
Line 12: Seed used to generate packing