Generating random initial positions for blood-like suspension cells

There are two modes supported for generating random initial conditions for both and rotation of the particles.

A.) Using the built-in interface

The following line instructs the code to try to initialise the *cellFields* (which consist of RBCs and PLTs) to the given hematocrit value using the max of MacPackIter iterations.

> *randomPositionMultipleCellField3D(cellFields, hematocrit, dx, maxPackIter);*

NOTE: currently is not tested enough and might contain bugs. The preferred method is the following.

B.) Using the code as an external application

To read the initial positions from an external file, use:

> *readPositionsBloodCellField3D(cellFields, dx, particlePosFileName);*

The input file makes reproducable runs possible, and it is a simple text format which can be editied by hand.

For this use first we build the initialiser code into a stand-alone application called **packCells** with the standard *cmake* precedure:

> *cd external/dynpacking*

> *mkdir build; cd build*
>
> *cmake ..*
>
> *make*

The application **packCells** is now built in the parent directory. The argument of the program are:

- *hematocrit* - the desired hematocrit level $\in [0; 1]$
- *sX, sY, sZ* - domain size in μm
- *maxIter* - maximum number of iterations allowed
- *allowRotation* - whether or not to allow particles to have random rotation besides random positions
- *scale* - scales the bin size for the cutoff to find interaction pairs (optional, can be omitted)

The output

The program has two outputs. **cells.pos** contains the particle positions in a simple text format. The first two lines gives the number of RBCs and PLTs, and then each line describest the position and rotation of a single particle.

The other file is **ellipsoids.pov** and it is an optional file to facilitate visualisation. It can be processed by Pov-Ray to produce some basic visuals of the particle positions.