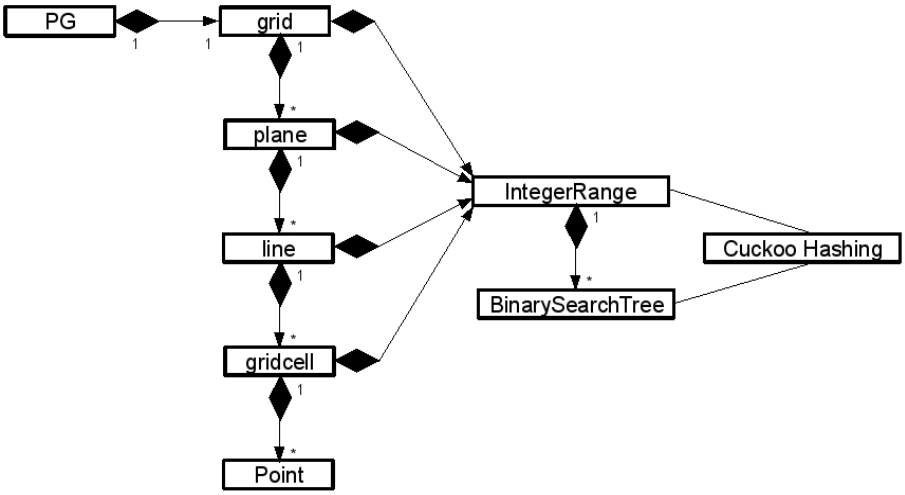
Programmers guide for the DPG library

This version of DPG is a part of the TexMol project. DPG is included as a library and the source code can be found inside the folder named DPG. The class diagram of DPG is shown here-

DPG Class Diagram



Note that augmented versions of DPG including the Surface maintenance, Born Radii computations and multiresolution features are not included into TexMol yet.

To use DPG in your code, all you need to do is the following-

- Ensure that the library libPG.so has been compiled
- Set the paths to the library and headers
- Include PG.h in your code

- Instantiate using PG(double *D*, double *xlate*, double *rmax*). *D* is the dimension of a cell, *xlate* is the minimum X, Y or Z coordinate of the atoms, *rmax* is the maximum radius of any atoms.
- Other public functions include-
  - vector<Point*> range(Point *, double delta)
  - bool pointsWithinRange(Point *q, double delta)
  - void addPoint(Point *a)
  - void addPoints(vector<Point *> *alist)
  - void removePoint(Point *a)

- The Point class contains five basic attributes the xyz coordinates, the radius r and an id.
- For specific applications, the programmer might choose to augment the 'Point' class with additional domain dependent values. For example we added Born radius attribute to the Point class when augmenting it for energteics computations.

- See TestPG for sample code.

Users guide for the DPG library

Installation

- If you have the acquired DPG as part of TexMol, then installing TexMol should also install DPG. So, follow the installation instructions in TexMol's user guide
- If you have acquired DPG as a standalone distribution, simply use the Makefile provided in the directory

Usage with TexMol

DPG is meant for a library to support/boost other applications. Hence we provide TexMol user only a simple interface via TexMol to test out the functions-

TexMol -DPGTest <path to input file>

A valid input File contains the following entries-

moleculeFile <path to a .xyzr or .xyz file>
updateFile <path to a .xform file>
queryFile <path to a .xyzr or .xyz file>
outFile <path>

Details about these files

1. This file is mandatory. DPGTest uses the moleculeFile to read a collection of atoms of a molecule and inserts them into DPG. The .xyzr file contains four numbers in each line the x, y, z coordinates and the radius of a single atom. If the radius is omitted (.xyz), then DPG assigns a fixed radius.
2. If updateFile is mentioned, then it performs the updates mentioned there. Each update is mentioned using old and new positions of an atom. If new position is omitted, then the atom is deleted.
3. If queryFile is mentioned then it performs the required range queries and stores the results in outFile.

Usage with standalone version

Execute-

TestPG <path to input file>

The details of the input file and the rest is same as for the TexMol distribution mentioned above