# C++部分类

Ball类

该类为程序主体类，主要是为建立弹性球模型而创建的。该类用到<Vect.h> <stdafx.h> <Wall.h>这三个相关头文件。主要实现对气体分子球型模型的运动行为模拟，以及完成在过程中对气体分子球型模型的一系列操作，其中也包含着球体的基本属性。

该类的主要公共成员包括：

setVel—球体的速度设定操作

setLoc—球体的位置设定操作

reverseVelo—球体的反转速度

stepForward—球体运动前进操作

stepBack—球体反向运动操作

isCollision—碰撞检测：

1. 球球碰撞检测(此处包含前进一步操作)
2. 球墙碰撞检测（此处包含前进一步操作）

先进行球对6面板整体碰撞检测，再对球与单面板碰撞检测做分析。

1. 球初始位置查重（notstep）
2. 球板碰撞检测

colWith—碰撞操作

1. 球球碰撞操作
2. 球墙碰撞操作
3. 球板碰撞操作

该类的主要私有成员包括：

球的各种基本属性：

radius—半径

mass—半径

velo—速度

loc—位置

声明如下：

#pragma once

#include "Vect.h"

#include "stdafx.h"

#include "Wall.h"

class Ball

{

public:

static double DT;

Ball();

Ball(Vect &l, Vect &v);

Ball(double m,double r,Vect l, Vect v);

virtual ~Ball();

Ball(const Ball& other);

inline void setVelo(Vect &v) { velo = v; }

inline void setLoc(Vect &l) { loc = l; }

void reverseVelo() { /\*std::cout<<"Yes!"<<std::endl;velo.outp();\*/velo = velo\*(-1);/\*velo.outp(); \*/}

inline const Vect v\_()const { return velo; }

inline const Vect l\_()const { return loc; }

inline const double r\_()const { return radius; }

inline const double m\_()const { return mass; }

void stepForward(double t=DT);

void stepBack(double t = DT);

bool isCollision\_1(Ball a); //与小球碰撞的检测

bool isCollision\_2(Wall &w);

bool isCollision\_2\_notStep(Wall &w);

bool isCollision\_3(Plank &p);

void colWith\_1(Ball &a,double t=DT); //与小球碰撞

void colWith\_2(Wall &w);

void colWith\_3(Plank &p);

protected:

private:

Vect velo, loc;

double mass=1, radius=1;

};

Stdafx类

该类是为标准系统内或是经常使用但不常更改的特定于项目的包含文件的包含文件而创建的，主要功能是为了对基本常用头文件打包，简化程序，提高效率。

其中主要包括以下几个头文件<stdio.h> <tchar.h> <iostream> <vector>

并于此定义了时间小量dT和范围Range。

#pragma once

#include <stdio.h>

#include <tchar.h>

#include <iostream>

#include <vector>

#define dT 0.1

#define RANGE 20

// TODO: 在此处引用程序需要的其他头文件

Vect类

Vect类是一个矢量类，由于Codeblocks以及Qt中系统自带<vector>矢量类是以数组形式操作的，并不方便此程序编写运行，故依照几何直观进行矢量类构造。该类主要是实现矢量自身的属性：长度，方向，以及矢量和矢量的关系，比如运算：加、减、数乘和点乘，并且同时进行了运算符重载，将运算符拓展到矢量范围里，从而更好地服务于整个程序。

该类的主要公共成员：

length—矢量长度

dis—距离

arg—夹角

dotProduct&crossProduct—点乘，叉乘运算

standardLize—单位化

operator+,-,\*—运算符重载

该类的主要私有成员：

三维空间坐标x，y，z—矢量的基础属性

#pragma once

class Vect

{

//friend class Ball;

public:

Vect() = default;

Vect(const Vect&);

Vect(double, double, double);

virtual ~Vect();

double length() const;

Vect dirc();

double dis(const Vect&) const;

double arg(const Vect&) const;

double dotProduct(const Vect &) const;

Vect crossProduct(const Vect &);

void standardLize();

void outp() const;

Vect operator+(const Vect& t)const;

Vect operator-(const Vect& t)const;

Vect operator\*(double t);

inline friend Vect operator\*(double t, Vect &p);

inline friend Vect operator-(Vect &p) { return p\*(-1); }

inline double x\_() const { return x; }

inline double y\_() const { return y; }

inline double z\_() const { return z; }

protected:

private:

double x, y, z;

};

Wall类

该类是一个对于墙板plank的集合，它其中功能的具体实现是要依靠对plank操作，此外，它与小球碰撞的实现主要是依靠plank和ball相互作用，其成员函数也都是和plank类密切相关的。在此课题中，我们的气体分子是在空间六面体中运动，因此墙面存在着6个主要坐标，左右两侧，上下两面，前后两面的3组坐标。

该类的主要公共成员：

Size—墙面大小

x1,x2,y1,y2,z1,z2—墙的6大坐标

addPlank—添加墙板操作

该类的主要私有成员：

n—墙面数量，添加墙板时会发生变化

vector<Plank> pVec—储存6个plank的vector，随时调用

和墙类的碰撞主要分为两类，以x板为例，一种是从正到负碰x板，另一种是从负到正碰x板。

声明：

#pragma once

#include "stdafx.h"

#include "Plank.h"

using std::vector;

class Wall

{

friend class Ball;

public:

Wall()=default;

Wall(int size);

Wall(int x1, int x2, int y1, int y2, int z1, int z2);

virtual ~Wall();

//void initialize();

void addPlank(Plank &p); //添加墙面

protected:

private:

int n = 0; //墙面数目，在加隔板时会发生变化

vector<Plank> pVec; //直接操作墙板会更好，整个墙体只需包含每一块墙板即可，具体实现细节交给墙板，可以不一定是矩形，甚至可以是其他性状的墙面

};

Plank类

该类即是为构建墙板而创造的类，是上述Wall类的具体操作实现对象。主要用于实现与小球碰撞的相互作用，以及界定边界区域。

该类的主要公共成员：

Plank(int f, int s)—带有表征面类型参数和面特征参数的板面

该类的主要私有成员：

style—表征面类型（可以进行调整和更换）

k—面特征

#pragma once

#include "stdafx.h"

class Plank

{

friend class Ball;

public:

Plank();

Plank(int f, int s);

virtual ~Plank();

protected:

private:

//表征面的方程，为方便处理，只分为六类，分别平行于xy,xz,yz平面，这样每个面只需一个参数即可

int style; //表征面类型

int k; //面特征

};

Gas类

该类是基于Ball类的弹性球体模型下对气体分子构建的类。该类主要对气体运动整体过程进行了检测和操作，包括对气体位置的检测，增加气体球体的操作，随时间变化气体状态分子运动的操作，能量的传递和变化以及翻转问题。是一个综合全面的类。是Ball类作用的体现。

该类的主要公共成员：

is\_Overlap—判断是否越界

addBall—加入小球操作

timePass—时间流转操作

EK—小球动能属性

Reverse—参量反转操作

该类的私有成员：

vector<Ball> bVec—一个包含大量小球的矢量Vector

#pragma once

#include "stdafx.h"

#include "Ball.h"

using std::vector;

class Gas

{

public:

Gas() = default;

virtual ~Gas();

bool is\_Overlap(const Ball &temp) const;

void addBall(Ball &temp);

void timePass(Wall &w, double t=dT);

void outp();

const vector<Ball>& output() { return bVec; }

double EK() const;

void reverse();

protected:

private:

vector<Ball> bVec;

};