Chapter 1

算法设计文档

1.1 数学概念的包装

以下的 dim 都是模板参数维数.

1.1.1 方程求解器, class equationsolver<dim>

这个类是方程求解的接口,

属性:

- 离散点的数量. int n[dim];
- 离散点各个维度上的距离. Real h[dim];
- 存储每个网格点上目标函数 v 的值. Real* value;
- 存储每个网格点上右端函数 f 的值. Real* fvalue;
- 存储每个网格点上离散方程的残差值. Real* rvalue;
- 将高维坐标转化为在一维存储数组坐标的操作类指针 coordinate* pcoord
- 各个环节采用不同的方式的记录值, 如松弛迭代时取用 Jacobi 迭代或者 Gauss-Seidel 迭代. boundary_type btype = Diri; restrict_operator restrictop = full; interpolat_operator interpolatop = linear; cycle_type ctype = V_cycle; stop_criteria stopcondi = maxstep; relax_type rtype = GaussSeidel;

操作:

- 初始化, 必须输入网格数量 n, 间距 h 和边界条件和右端项在每个网格上的值 fvalue. 初始值和边界 v 是可选项 (默认为 0). equationsolver(n ,h, fvalue, value).
- 输出存储的近似值 value 和残差 rvalue. pair<Real*,Real*> getresult() const;

虚操作: 需要在子类中重新定义

- 选取可选项. bool chooseoption(name, type);
- 更新残差. void updatervalue();

- 限制算子. equationsolver* Restrict();
- 插值算子. eqautionsolver* Interpolate();
- 松弛算子. void Relax();
- 求解函数. void solver(v, stopcondition);

1.1.2 泊松方程求解器,class possionsolver<dim> : public equationsolver<dim>

泊松方程的求解器, 继承求解器的接口.
没有额外的固有属性

操作:

- 实现 equationsolver 中的所有虚操作.
- 各种松弛算子. void RB_GaussSeidel_Relax(); void Relax(Rv); void Jacobirelax(v,f,result);
- 分别实现全权重限制算子和插入限制算子. void Fullrestrict(v, result);
- 插值算子. equationsolver* Linearinterpolate();

1.1.3 多重网格循环方式, class cycle<dim>

这是个操作类, 不含有属性. class Vcycle,class FMGcycle 分别从中继承实现接口

虚操作:

- 对求解器 psolver 进行一次循环运算, 循环参数为 v. void operator(psolver, v);

1.2 UML 类图

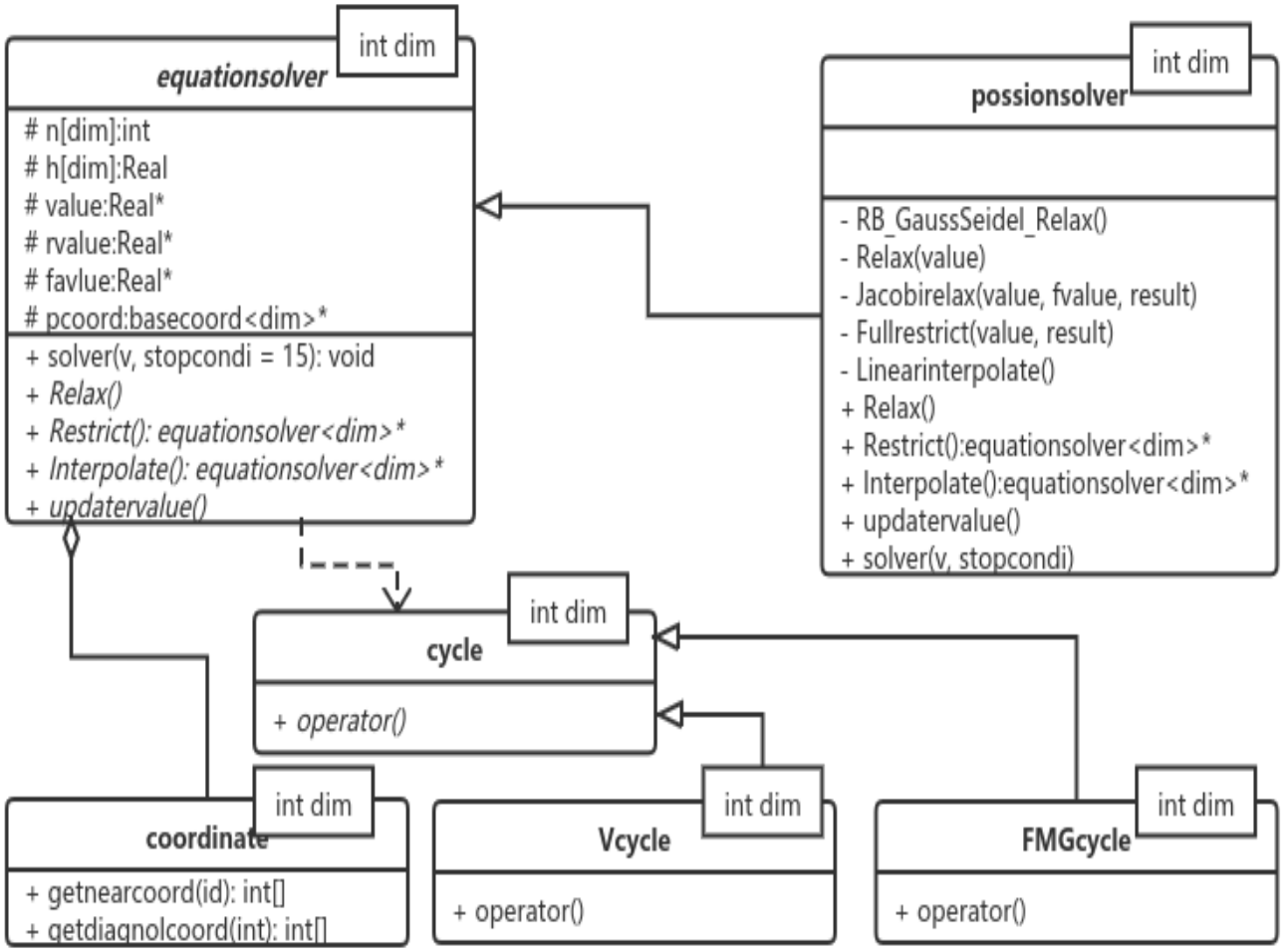

图 1.1: 多重网格程序 UML 类图