三维 CAD 建模

1. 定义：在计算机内有效表示、构建、处理产品设计中的三维实体的一门学科。
2. 核心：**三维形状建模**
3. . 三维形状建模发展历程：

a. 工程制图：pros 功能强，通用；cons 不直观、制作成本高、制作周期长、修改存储传输困难

b. 计算机绘图（60 年代），数字化：二维 CAD 系统

解决了修改存储传输困难的问题

c. 实体造型（包括自由曲线曲面） 70 年代。

包含三维实体的完整集合表示；运行设计人员在三维空间直接进行建模

d. 参数化特征建模（80 年代） 自动化、集成化。

引入约束、特征等；支持语义建模

e. 逆向工程

通过扫描数据来进行重建

Ivan Edward Sutherland，计算机图形学之父，Sketchpad：A Man-Machine Graphical Communication System

3. 课程内容：

1）实体建模及其表示方法：完备性、有效性、唯一性、无二义性

线框表示：组成边表、边点、方程，简单、处理容易、高效。

B-rep/CSG 树/分解模型/参数化模型和特征模型

2）建模操作

基本形素生操作/局部操作/布尔操作/特征编辑/变动操作设计

3）三维 CAD 建模系统设计。

表示法选择/求交功能设计/系统鲁棒性保证/ACIS 简介

实体模型

1. 计算机模型产生过程

客观对象空间→数学抽象空间（点集拓扑）→计算机表示→形式化/抽象表示

Q：三维实体在数学里面对应什么？A：欧氏空间->点集 【点集】反例：单点、边

2. 体的点集模型(三维空间中)

定义 1：给定$A \in E^3$(空间)，记 $r(A) = C(i(A))$（A 的所有内点的闭包），若 r(A)≡ A，则 A 为正则集。 C 表示闭包，i 表示内点集(interior)

闭包：2.2 开集、内部；闭集、导集；闭包 基本关系 - 知乎 (zhihu.com)

<以正方体为例> 直观上此式的含义是删去依附于三维点集 A 上的所有悬挂面、孤立边和点

内点定义：其中任意一点，存在邻域，邻域内所有的点都在集合内

悬挂边 -> 单独有条边，或者单独有条面

推论：

- 孤立的点、边、面，不是正则集；带悬挂边、悬挂面，也不是正则集
- E^3本身是正则集，$r(A)$本身是正则集（*排除作为实体模型，因为需要有界*）
- 三维实体沿点或边的并集是正则集；两个正则集的布尔运算**不能**保证是正则集，举例：两个物体在顶点粘到一起——类似焊接

加*号表示求正则集， $A \cap^* B = r(A \cap B)$（正则集合运算）

定理：三维实体是E^3中有界正则点集。

3. 体的曲面模型

定义：A 为 E2 中的一个紧致的拓扑空间，如果 A 中的每一个点都存在一个邻域与 E2 中的开圆盘同胚，则称 A 为二维流形(2-manifold)。

同胚：同胚是一个数学概念，指两个拓扑空间之间存在一个连续的双射和其逆映射都连续的映射

紧致：该流形体是**有限**的且**闭合**的

性质：

- 二维流形是封闭曲面。

二维流形不一定对应一个三维物体（Klein 瓶）-> 不可定向（例：莫比乌斯环）

正则集的边界不一定是二维流形（共点或共边拼接而成的正则体）

定理：三维流形体的充要条件：其边界为可定向的二维流形。

4. 二维流形的平面图模型

定义： 一个有向平面图，带有有限个顶点$N = \{n_1, n_2, \dots, n_i\}$，有限条边$A = \{a_1, a_2, \dots, a_j\}$，以及有限个由 N,A 中的点、边围成的有一定走向的多边形$R = \{r_1, r_2, \dots, r_k\}$。

定理 一个平面图为流形体平面图的充要条件：

（1）对每一条边，**有且仅有**一条边与其对等。

（2）对每一个顶点，图中与顶点相连的多边形**只形成一个循环**。

（3）平面图是可定向的，即平面图中多边形的走向能与它在立体形的走向相反。如，都约定了用右手定则判断方向，指向体外）

5. 实体表示方法应具备的特性

1）有效性：对每个实体表示，都存在一个三维物体与其对应。

2）唯一性：对每一个三维物体，有且仅有一个实体表示与其对应。

3）无二义性：对一个有效的实体表示，有且仅有一个三维物体与其对应。

4）完备性：不需要添加任何其他元素，这个对象也可称为完备的或完全的

(B-rep)边界表示法及其数据结构

1. B-rep 的组成：拓扑结构、几何信息。

Q：什么决定了一个物体/形状？　A：拓扑结构：平面图模型；点线面信息及其相互之间的连接关系

如果网格的每个边最多被两个面片共用，那么这个网格就是流形网络，否则称为非流形网络

（1）拓扑结构：拓扑元素+拓扑关系(所有边界元素+拓扑关系)

拓扑元素种类：面(Face)(有界)、边(Edge)、点(Vertex)、体、环：内环走向与外环走向相反

拓扑关系（9 种）：

点为中心：V{V},V{E},V{F}；

边为中心：E{V},E{E},E{F}；　（非相互独立）

面为中心：F{V},F{E},F{F}；

（2）边界的几何信息：描述物体的大小、位置、尺寸等信息。(与拓扑对象映射)

	拓扑	几何
点：坐标	Vertex →	Point
边：边的方程	Edge →	Curve
面：面的方程	Face →	Surface

2.翼边数据结构（Winged-edge structure）

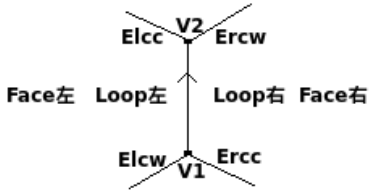
由 Baumgart at Stanford 提出；时间换空间的思想；寻找一组基 → 没有简单的重复信息 → 但是可以通过推导出来；e.g. V{V}+E{V} -> V{V}

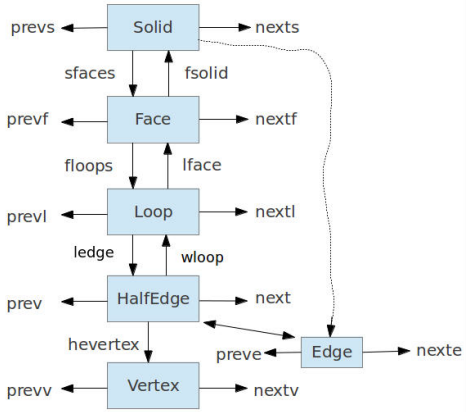
Loop(环)：边构成封闭的曲面 （外环、内环）；一组首尾相连封闭二维区域形成一个环(Loop)

核心思想：**以边为核心**组织数据结构；

引入环，$\{v_1, v_2, loop_l, loop_r, E_{lcc}, E_{lcw}, E_{rcc}, E_{rcw}\}$共八个指针表示一条边

翼边数据结构与半边结构在计算机中的数据结构表示





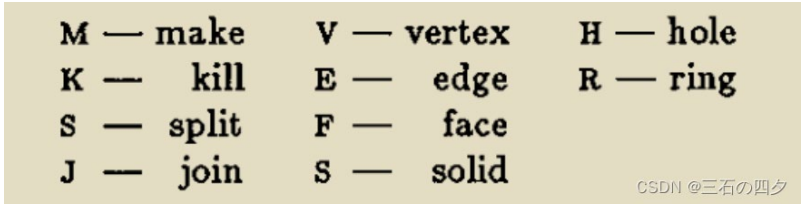
对流形体而言，每一条边相邻的点 2 个，相邻的边 4 条，相邻的面

拓扑关系选择：E(V)、E(E)、E(F)、V(E)

通过判定 Loop 在边的左边还是右边，判断绕向来找到 Loop 上的所有边。

```
all_edges(lp){
 te = lp -> eo; //取一条开始的edge
 do {
  if(te -> loop_r == lp) //不能确保环上每条边的方向都是通顺的
   te = te-> E_rcw;
   else
     te = te-> E_lcw;
 } while(te != lp->eo)
}
```

这种边的结构是不满足二维流形的可定向条件。在这个基础上对翼边的结构进行改进，提出了半边结构的思想，满足了可定向性。



虚线代表可以生成线框模型：{V},{E},E{V}

带空的环称为(rings)

核心思想：边在三维模型的表达中是一个重要元素，以边为重点来组织数据结构

基于 B-rep 的建模操作

欧拉操作：旨在有效可靠地构建 B-rep 中拓扑结构（通用、有效）

1. 欧拉公式：$V-E+F=2$

$$V-E+F=2\left(S(体)-H(柄)\right)+R(内环) \qquad (*)$$

2. 欧拉操作的设计思想：通用，有效

   基于欧拉公式来保证其有效性；

   提供一组通用、完备的 B-rep 拓扑生成操作；

我的理解是欧拉操作给模型的拓扑生成提供了完备的理论依据。

3. 欧拉操作的选择 （*）为 6 维空面的 5 维超平面

   我们需要五组基来表示：

v	e	f	h	r	s	Operator
1	1	0	0	0	0	mev
0	1	1	0	0	0	mef
1	0	1	0	0	1	mvfs
0	-1	0	0	1	0	kemr
0	0	-1	1	1	0	kfmrh

- 欧拉操作的选择：维护扩展欧拉公式的等号成立

S 实体 Sold 的数目，H 空洞/柄 hole/handle 的数目；R 内环 ring 的数目

```
struct solid{
  solid* prevs;
  solid* nexts;
  face* sfaces;
};

struct face{
  face* prevf;
  face* nextf;
  solid* fsolid;
  loop* floops;
};

struct loop{
  loop* prevl;
  loop* nextl;
  face* lface;
  halfedge* lhe;
  edge* ledg;
};

struct halfedge{
  halfedge* prv;
  halfedge* nxt;
  loop* heloop;
  vertex* hev;
  edge* hee;
};

// optional?
struct edge{
  edge* preve;
  edge* nexte;
  solid* es; // ?
  halfedge* ehe;
};

struct vertex{
  vertex* prevv;
  vertex* nextv;
  halfedge* vhe;
};
```

欧拉操作	含义
`mev(v1 ,v2 ,e)`: make edge +vertex	生成一个新的点 e2，连接该点到已有点 v1，与给定点的边，如果环已经有边，新加边与给定边的的四个半边要能形成一个循环；如果环还没有边，新边的两个半边要能循环
`mef(v1, v2, f1, f2, e)`: make edge + face	连接面 f1 上的两个点 v1，v2，生成一条新边 e，并产生一个新面，如已经有了三条边，连接上度数为 1 的两个点，生成一个矩形面
`mvfs(v, f)`: make vertex + face + solid	生成含有一个点的面，并且构成一个新的体，**一般是从无到有的第一步**
`kemr(e)`: kill edge & make ring	删除一条边 e，生成该边某一邻面上的新的内环，如一个矩形面里面有一个三角形环，三角形环与矩形面有一条边相连，kemr 的作用就是消去连接的这条边，使得三角形环成为内环
`kfmrh(f1, f2)`: kill face & make ring + hole	删除与面 f1 相接触的一个面 f2，生成面 f1 上的一个内环，并形成体上的一个通孔/柄，或者将两个物体合并为一个物体
`semv(e1, v, e2)`: split edge & make vertex	将 e1 分割成两段，生成一个新的点 v 和一条新的边 e2(辅助操作)

example构造同轴空心立方体

```
mvfs()
mev() * 3
mef() // 顶面构造完成
mev() * 4 // 构造四条侧边
mef() * 4 // 除了底面都构造好了
mev() * 4
mef() // 底面的内环好了并有一条边与底部四条边连接，底面现在有一个环和一个空心面
mev() * 4 // 空洞立方体的四条侧边
mef() * 4 // 空洞立方体的四个侧面
kfmrh() // done
```

> 构建一个简单的立方体，构建四个侧面的时候就要选取一个面进行构建。也就是说，一开始构造的顶面，有两个法向量，一个向上一个向下，选取法向量向下的面继续构造即可。
>
> 最后每个面都要选取法向量，朝外即可。侧边的两个半边，分别可以用来给相邻的两个面形成向外的法向量。

4. 欧拉操作的结论

可以证明：欧拉操作是有效的；用欧拉操作对形体操作结果在物理上是可实现的，欧拉操作是完备的，任何形体可在有限步的欧拉操作中构造出来。

（1）所有流形体的 B-rep 欧拉操作都能构造；

（2）由欧拉操作构建的 B-rep 在拓扑结构上一定有效；

（3）将其正确的嵌入 3 维欧式空间E^3，结果一定是流形体。

扫成操作（sweeping）

扫成体：由一个**平面二维区域**（由直线段、圆弧和自由曲线围成的二维曲线）沿一条既定轨迹线，扫描产生的三维空间点集。 2.5d

（1）平移扫掠（translational sweeping）：轨迹线为一段**直线**

1. 几何构造：

每一个已知点构造一个新的几何点，新点 = 已知点+$d \cdot \vec{v}$，由此构造一条新的边

每一个已知边构造一条新的几何边，根据侧边或者轨迹线构造一个新的几何面

每一个已知面，根据轨迹线构造一个新的几何面

2. 拓扑构造（见右图）

（2）旋转扫掠

轨迹线是圆或圆弧；平面区域位于轴线的一侧；算法实现与平移扫掠相似，对封闭情况做特殊处理

（3）广义扫成

轨迹线为任意空间轨迹；平面（二维）区域在扫掠中可变化。

Solidworks 方法：引入多条轮廓控制线，要求平面区域的轮廓线在垂采样点处与所有控制线接触

拓扑信息构造：与平移相似

几何构建：如何精确表示扫掠曲面

理论方法：s 表示弧长参数，$\vec{t}$表示切向量，$\vec{n}$和$\vec{b}$分别表示 x 和 y 方向的法向量？

平面表示为$C(u) = (C_1(u), C_2(u))$分别表示 x 和 y 方向

```
sweep(S: Solid, F: Face, V: Vector, d: REAL)
// suffix of up: a vertex on the bottom (assuming sweeping downwards)
begin
 L = loop of F;
 firstv = first vertex of L;
 firstup = firstv + d*v;
 mev(firstv, firstup, newe); // new edge
 prevup = firstup;
 nextv = next vertex of L;
 while(nextv != firstv)
 begin:
   up = nextv + d*v;
   mev(nextv, up, newe);
   // side face
   mef(prevup, up, F, newf, newe);
   // continue looping
   prevup = up;
   nextv = next vertex of L;
 end
 // bottom face
 mef(prevup, firstup, F, newf, newe);
end
```

沿着轨迹曲线的可以表示为

$\Gamma_{(s,u)} = \vec{r}(s) + C_1(u)\vec{n}(s) + C_2(u)\vec{b}(s)$

实用方法：

平面的边界和空间轨迹线均采用 NURBS 曲线表示，扫掠曲面均采用 NURBS 曲面逼近表示

曲面逼近表示：由轨迹线采样点处的平面区域的 NURBS 曲线表示的控制顶点组合而成

（4）使用限制：只能构造 2.5 维物体。

基于 B-rep 的布尔运算

1.算法分析：

问题：输入　两个B-rep，b(A)，b(B)

输出　b(A <op> B)　　（` <op>`：并、交、差）

b(A∪B) = b(A) out B + b(B) out A + b(A) ∩ b(B) 同向部分

b(A∩B) = b(A) in B + b(B) in A 同向部分

b(A-B) = b(A) out B + b(B) in A + b(A) on b(B) 异向部分

2.算法步骤：

（1）.求两物体各表面之间的交线段

- ✧ 先bbox包围盒粗判：剔除无交面

  (aabb)（axis align bounding box）

  （obb）(紧致包围盒，沿着方向轴）
- ✧ 两曲面方程求交，求出交线方程；
- ✧ 交线与每一面的边界求交，求出位于曲面内部的部分；
- ✧ 求两曲面内部交线段的公共部分。
  - ■ 怎么有效判别两个自由曲面的交叉部分是难题之一
  - ■ 这个环节很耗时，布尔操作很耗时的原因之一

（2）将所有有交表面相对于另一物体进行面体分类

- ✧ 对每一有交的面，搜索其所有的交线段，构成完整交线；
- ✧ 对剖分出的面进行面体分类，点体分类

（3）建立结果物体的B-rep，建立新实体的数据结构

原则：尽可能重用已有数据结构

a). 建立交线的数据结构，相应地修改交面的数据结构；

b). 重建结果物体的面表，根据布尔运算符来选择面

c). 对于A-*B，需要对拼合体中属于B的面作反向操作（应该是为了方便and）

3.求交效率的提高

（1）.连续求交法：首先求出一个初始交点，然后采用与当前交点相关的两个特定的相交面进行求交，求出交线段及下一个交点，如此循环直至回到初始交点。

（2）.包围盒粗制　bounding box　排除无交面对

局部操作：直接对三维物体的几何信息或拓扑结构作局部修改的操作

1.负操作 negative

功能：将实体所有边界面的法向改为相反方向

算法实现：1.改变每个 Face 面的法向；2.改变所有半边的走向：3.改变半边的起点；4.改变下一条边的指针

2.对称映射

功能：给定一个物体，构成一个对称物体；通过对给定实体关于一个平面作反射变换得到一个新的实体，或者改变给定实体的形状

算法实现：（1）拓扑上：复制原实体+负操作

（2）几何上：对相应元素作对称变换

（3）如果镜面为体上的一个面，处理重合面；对于以实体的一个面作对称面的情况，对重合元素作合并处理

3.倒角

功能：将物体上的一个点（或一条边）用一个小平面替代，消除一个尖角或尖边

算法实现：

几何上，通过给定的参数计算出新的顶点的坐标，面的方程；

拓扑上，用semv()、mef()生成新的点边面，再用kef()、kev()删除不要的点边面。

4.圆角

功能：将物体上的一个点或者一条边（一组边）用曲面替代，且保证新曲面与原点、边之间的面三角光滑拼接。

光滑拼接：一阶导连续。

算法实现：

理论方法：球的包络面；

使用方法：分别构造多张曲面拼接而成（如立方体的圆角，连带着边也圆角了，角就变成了构造三个四边曲面）

难点：难在如何保证光滑性

5. 粘合操作

功能：将两个在表面贴合的实体“粘合”成一个新的实体；

算法实现：修改贴合面的表示

几何上，处理贴合面　(A-*B)U(B-*A) ，确定(A-*B)、(B-*A)的边界

拓扑上，将(A-*B)、(B-*A)的边界，分别作为两个面表示

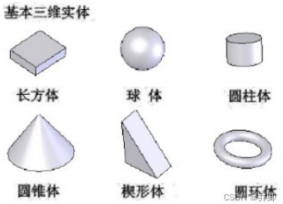
处理重合边的两个面表示

建立新实体的面表

6. 扭变 / 直接建模操作 / 同构建模

部分边面做操作，不改变拓扑结构

直接对实体模型中的一个或一组边界元素进行**平移**或**旋转**变换来生成、变动实体模型



CSG 树表示 Constructive solid geometry

边界表示（Boundary Representation, B-reps），即用一组曲面（含平面）来描 述物体，这些曲面将物体分为内部和外部。边界表示具体又包括多边形表面模型和扫 描表示两种。

构造实体几何表示（Constructive Solid Geometry，CSG），它将实体表示成立方体、长方体、圆柱体、圆锥体等基本体素的组合，可以采用并、交、差等运算构造新的形体。

空间分割表示（Space-Partitioning），用来描述物体的内部性质，将包含一物体的空间区域划分成一组小的、非重叠的、连续实体（通常是立方体）。

（1）.基本体素的半空间表示

半空间：给定一张解析曲面，$f(x,y,z)=0$，则$H_(f)=p\in E3|f(x,y,z)\leq 0.$ 与 $H+(f)=p\in E3|f(x,y,z)>0$

$$H\_(f)\cup H_+(f)=E^3$$

基本体素：（1）用半空间表示：任一凸的三维实体可以表示为其边界曲面半空间的交集，不能为自由曲面

基本体素的种类：长方体、圆柱体、球、正棱柱、正棱锥、环面

（2）.CSG 树：用于表示三维实体的一棵二叉树，其终端结点表示基本体素，中间结点表示正则集合运算。

（3）.CSG 树表示的数据结构

树结点的表示：左右二叉树的指针、类型码（非零）、坐标变换、叶节点（为值 0）、基本体素指针（叶节点非空；方程+定形参数）、其他节点（值）（1 并； 2 交；3 差，为不同类型布尔运算赋 enum）

拼合运算。（就是一个二叉树图）

完整性：都能转换为 Brep，因此可以完整地表示一个实体

唯一性：不成立，一个实体可以有多个 CSG 树，容易构造反例

有效性：一定有效，因为构造的过程都是有效的

（4）“分治”算法

```
F(tree, data){
if (tree is a leaf)
then primitive,F(tree, data)
else combine(F(left_subtree, data), F(right_subtree, data), node(tree))
}
```

对于 CSG 应用分治(divide-and-conquer)算法。所谓分治，就是对于实际应用中遇到的问题，倘若它比较复杂，不能直接求解，就将这个问题分成若干个小问题分别求解；然后再将求得的小问题的解合成整个问题的解。显而易见，这是一种**递归**的方法：倘若分得的小问题仍不易直接求解，可将小问题再细分下去，一直到可以求解为止。

（5）边界元素分类算法

点体分类、边体分类、面体分类

点体分类：以$S=A\bigcap^* B$ 为例，给定点 P

	in A	on A	out A
in B	in S	on S	out S
on B	on S	on S / out S	out S
out B	out S	out S	out S

边体分类：基于半空间做，以$S = A \bigcap^* B$为例，给定面 X

同向部分

$$X_{inS} = X_{inA} \bigcap^* X_{inB}$$

$$X_{onS} = (X_{onA} \bigcap^* X_{inB}) \bigcup^* (X_{inA} \bigcap^* X_{onB}) \bigcup^* (X_{onA} \bigcap X_{onB})$$

$$X_{outS} = X - (X_{inS} \bigcup^* X_{onS})$$

线框生成：对所有的面求交，生成交线；对每一个交线，求 $curv_{onS}$;加上面的边界

Brep 生成：对每一个基本体素的面 surf，求$surf_{onS}$，即为边界面

分解模型（类似于栅格化）Decomposition Model

B-rep 和 CSG 的共同点：精确表示。

所以要离散表示，但会增加存储。

分解模型：将三维空间分割成一组体元，对三维物体利用其在空间所占据的所有体元对其进行的逼近表示。

1. Voxel 表示：体元是同大小、同方向的立方体。（与坐标轴平行）

数据结构：三维整数数组 c(ijk)

拼合运算：通过比较相应单元的数值即可得出结果。

构造：基于 B-rep/CSG 树表示转化、直接扫描转换（基于硬件）

可以自适应分割。

八叉树 Octree-CSDN 博客

2. 八叉树表示：体元是同方向但不同大小的立方体。

八叉树表示的构造：

自适应分割

a. 定义包含所有实体的宇宙空间，设为$2^n * 2^n * 2^n$，生成一个根结点。

b.（递归）用当前结点的**三个中分面**对其进行分割，可将其分割为八个子空间，构成当前结点的八个子节点，并依次建立每个子节点与实体的占据关系：**Full、Empty、Partial**

c. 对于处于 Partial 的子节点判别其是否满足精度要求，不满足则重复（b）的过程，直至满足精度要求。

八叉树节点的编码方式：

a. 兄弟之间的编号：

含左下角点的子节点编号为 0；

沿 x 轴正向相邻的兄弟编号增加 1；

沿 y 轴正向相邻的兄弟编号增加 2；

沿 z 轴正向相邻的兄弟编号增加 4；

b. 任一节点的编号：$N1, N2, \ldots, Nm.\ 0 \le Ni \le 7$，m 为该节点在八叉树中的层次。

节点的几何信息表示：

边长为 $2^{(n-m)}$

设$N_i = e_i 2^2 + d_i 2^1 + c_i 2^0$，左下角点的坐标为：

$$X = c_1 2^{n-1} + \cdots + c_m 2^{n-m}$$

$$Y = d_1 2^{n-1} + \cdots + d_m 2^{n-m}$$

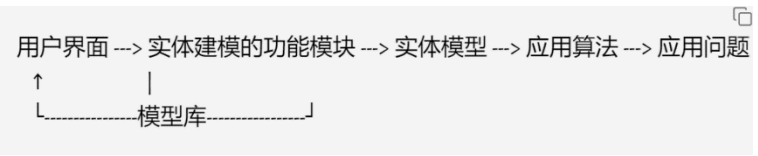
$$Z = e_1 2^{n-1} + \cdots + e_m 2^{n-m}$$

分析：

1. 表示简单，处理简单、统一、可靠；
2. 可以表示物体内部特性；
3. 便于并行处理；
4. 精度问题（缺陷）；
5. 存储问题（缺陷）；
6. 表示与物体位置的关系（缺陷）；

实体造型系统

一、 系统结构



模型库的模型一般存为中性文件，比如 stp 格式

二、系统设计

	Brep	CSG	DM
优点	显式且精确的表示，支持对它们进行交互访问和修改； 可以支持局部操作； 可以支持几何约束表示 可以支持表达很多产品的非几何信息，如公差；	表示简洁，记录了建模的过程（支持undo和redo）；	表示了表面内部的信息，可以支持体积计算等操作；
缺点	表示与相关算法复杂，记录了建模的结果； 没有记录建模过程，不能redo和undo	不支持交互访问和修改； 表示不唯一； 不支持局部操作； 不支持几何约束表示； 不支持表达非几何信息；	逼近表示，不精确； 空间效率低，存储不友好；

1. 选择实体表示法

B-rep：显示表示实体的所有边界信息、支持局部操作；

支持几何约束表示及基于几何约束的设计；

支持并几何工艺信息的表示，如**公差**（实际参数值的允许变动量，最大最小之间）；

表示能力极强。

缺点：表示与操作复杂，点操作极差；

只表示结果，不表示过程，无法支持对基于过程的建模设计。

CSG 树：与 B-rep 基本相反，表示不唯一。

（分解模型）DM 表示：辅助表示、物性计算、拼合运算。

综上，理想方法：多种并存 问题：高度复杂，难以保证一致性

实用方法：一种为主（B-rep），其它为辅（CSG、DM）

- 路线 1：GUI 以 CSG 表示 -> 修正的 CSG 表示 -> 从 CSG 转化为 Brep 表示
- 路线 2：GUI 以 CSG 表示 -> 局部操作，直接建模得到 Brep 表示
- 何为修正的 CSG：对 CSG 的基本体素用 Brep 表示
- 仅采用 Brep 表示的算法，不采用 CSG 表示的算法

自由曲面求交算法

曲面种类多样；存在精度、可靠性问题

求交算法设计

1. 统一求交算法：统一成平面多边形求交；NURBS 求交（同样有精度等问题）
2. 自由曲面求交方程：理论方法：用 NURBS 很复杂；曲面分别为$S_1(u,v)$，$S_2(u,v)$，S_1 隐式化得到 $f(x,y,z)$，带入 S_2 有交线 $f\big(x_2(u,v),y_2(u,v)\big)=0$

实用方法：

离散法：（1）.将曲面离散为满足误差要求的**三角形组**；

（2）.三角形之间求交，形成交线；（找到一个鲁棒的三角形求交算法很难）

特点：一般性，简单，全局性

质量问题（不光顺）、效率问题。

数值法（跟踪法）：跟踪法

确定初始交点；确定跟踪方向，两个切平面法向的叉积；确定跟踪步长；连续跟踪下一个交点，直至曲面边界或回到初始交点；初始交点精化方法

确定初始交点：采样 uv 线，分析整体的几何拓扑结构

（1）.确定初始近似交点

（2）.在当前交点处沿跟踪方向(两个切平面法向，向前跨一个步长，得出下一个交点近似值，并对其用牛顿迭代法求算，得出准确交点。

特点：质量好，精度高，具有一般性，可能不收敛，依赖于初始点的选取。

求交的情况：

与边界有交点；与边界无交点；与边界无交点切有环；只交于一个点（切点）

交成一个 8 字，怎么确定跟踪方向？

裁剪曲面(被相交曲面裁剪出的曲面)的表示与处理（Trimmed Surface）

定义：一张完整的曲面经过与实体求交后被裁剪出的部分曲面

特点：是不完整的曲面，参数域复杂，包含交线作为其边界线，必须到参数域上处理。

表示：曲面参数方程+曲面边界的空间表示及其参数域上的表示。

参数域以及空间边界线即为显式表示

对于两个相交面，S_1上的交线投影到空间的曲线应该和S_2上投影到空间的曲线一致，这对于逼近表示来说是困难的，that is，not watertight（水密）

一个解决方案，对相交区域重新三角面离散化，但是也要确保交线处的精度

系统的可靠性

1. 几何造型操作不可靠的原因分析

主要是精度可能会造成问题

根本原因：计算机的**浮点表示**；（数据表示有误差）

直接原因：几何关系的判别是相对于容差的，相对于给定误差是相互关联的；

浮点数系下判别出的相关几何关系可能发生冲突。

解决方案的基本思想：

（1）.判别方法尽可能的证明相关的几何关系，判别具有一致性；

（2）.能发现不一致几何关系，进行调整，不成功宣布不能做。

2.一种提高几何关系判别准确性的方法（基于点面容差的方法，面为平面多边形）

（1）.基本思想

（2）.初步确定重合关系的方法

重合关系：**点面**关系重合 PF、**点边 PE**、**点点 PP**、**边面 EF**、**边边 EE**、**面面 FF** 等。（6种）

所有都统一到判别 PF ，然后其它关系可以通过点面之间的关系来判别

点面重合关系的判断方法：给定一个容差∋.，点 P 和面$F = u_1, u_2, \dots, u_n$，

设 F 的方程为 $ax + by + cz = d$，用线性方程方法求解

定义容差 $t_p = 2 * max_{i=1}^{n}|d_i|$，用来确保计算面 F 的点都会被判定在面上

则如果$|dist(P,F)| \le t_p$+∋， 认为 P 与 F 重合

基于点面关系来判别其他的重合关系：

通过 PF 判定 FF，PE 和 EF；通过 PE 判定 PP 和 EE

基于几何元素局部、动态容差表示提高几何关系判别准确性

一个问题，假设两条无限长的直线要判断重合

交点附近采样可能会判定相邻的两个点重合

继续采样，一定范围内可能相邻的点都会被判定重合

判定采样序列的第一个点和最后一个点，**很有可能显然并不符合重合的判定**

（1）几何元素的容差表示

由几何元素和三个容差域（ϵ, δ, Δ）共同表示，因为浮点数表示，不能进行精确判断，需要容差域

$$\epsilon = \tau - \gamma,\ \delta = \tau + \gamma,\ \gamma \ll \tau$$

（2）几何关系的判定与容差表示的动态调整

重合、包含、相交、分离

a.分离关系：两个元素的δ域不变

调整： Δ i=d/2,d 为两者之间的距离。

b.重合关系：

判别条件：0_1 和 0_2 两几何元素在它们的ϵ域的交集内存在一个它们共同的逼近表示，则两者具有重合关系。

调整:将两者合并为统一表示,基于逼近产生一个新的容差表示(元素+三个容差域)，ϵ域和 Δ 域取二者相应**交集中的最大可能域**，δ域是相应域的**并集中的最大可能域**

几何元素：取为ϵ域交集中的一个同类元素；

ϵ域和 Δ 域，取为其**交集**中的最大可能域；δ域，取为其**并集**中的最大可能域。

c.包含关系

同维元素的重合关系判断条件：01 和 02 在它们的ϵ域交集内存在一个它们交集范围的逼近表示，则 01 和 02 重合

判别条件：如果在 01 的ϵ域中存在 01 的逼近表示 M1，在 02 的ϵ域中存在 02 的逼近表示 M2，且 M1 包含于 M2，则 01 包含于 02.

调整：几何元素：01 取为ϵ域交集中的一个同类元素。

ϵ域和 Δ 域：取为其**交集**中的最大可能域；

δ域：延伸到 O2 的δ域为止；相应域的**并集中的最大可能域**

O2 不变!!!!

调整的时候可能会出现δ和 Δ 大小不符合定义

d. 相交关系：判别条件：两者之间的ϵ域交集非空，且不是重合和包含关系。

调整：几何元素取为相交所得的几何元素。

ϵ域取为其交集中的最大可能域。

（3）不一致性关系：如果出现ϵ域为空，或者 Δ 域包含于δ域。

ACIS 几何造型引擎

目标：开发具有灵活、可扩度、开放结构的三维几何造型引擎，有效支持 CAX 系统开发

一、背景

BUILD----->ROMULUS------->UGNx------>ACIS----->软件包

先进性：a. 采用面向对象技术

b. 具有极好的可扩展性

c. 支持非流形体表示与处理

ACIS 类体系

ATTRIB------用于支持用户在分流数据结构中定义所需属性的类。

Application Interfaces

(1)、C++ Interface------C++ classes、API functions、DI

(2)、Scheme Interface

(3)、MFC Interface　　ACIS MFC 提供了与 MFC 的接口。

参数化建模

1. 传统实体造型问题

a. 精确建模不方便、不高效

b. 不能有效支持系列化产品设计

原因：传统实体造型不表示与处理**形状设计意图**。

2. 几何约束：施加在几何元素之上，对其形状和位置进行限定的条件，如平行、垂直、重合、各类距离。

3. 参数化集合模型：由一组**形状参数**决定的几何模型

几何约束：施加在几何元素之上，对其形状与位置进行限定的条件

结构约束：垂直/平行

尺寸约束：距离

代数约束：参数之间满足的代数关系

一种参数化几何模型：几何模型（B-rep）+**几何约束**

4. 参数化建模过程

（1）交互设计草图：定义拓扑结构，生成包括拓扑结构的初始形状

（2）定义几何约束：生成参数化几何模型

（3）转变为一组参数值，自动生成几何模型所需的精确几何模型

（4）**修改相关的参数值，生成变动设计**

5. 关键技术问题

a. 几何约束模型的有效性

几何元素的**自由度**：几何元素在空间自由运动的自由程度，如点在二维空间自由度为 2，直线为 2，线段为 4；总自由度=总约束度+3

几何元素的**约束度**

有效性：每个独立几何元素的自由度与其约束度相等。

无效：过约束（无解）、欠约束（无穷解）

b. 约束求解

c. 多解问题

6. 约束求解

（1）. 整体约束求解

a. **特征点集**（边界顶点、圆心、自由曲线的控制顶点）的确定

b. 约束方程组的求解：数值求解

Jacobi 矩阵

（2）.局部约束求解

a.几何约束图：用顶点表示几何体，用边表示几何约束（以几何元素为结点，以它们之间的约束关系为弧的图）

方法 1：几何推理法

基于几何推理将无向几何约束图转化为有向几何约束图

寻找强连通分枝

对强连通分枝用整体方程组求解，对其他元素局部求解

即从某个节点依次出发，把其他能计算出来（自由度约束度相等）的节点给计算出来

b.基于图分解的局部化

方法二：Owen 图分解算法：（三角分解法）

1.将几何体和几何约束用图的形式表示出来；

2.将图分裂成一些**双连通**子图；（如果任意两个顶点之间存在一条路径，则称这个图是连通的）（我们将图在某一个顶点分开《将这个顶点分别复制到两个分开的部分中去》，如果原图被分成两个不连通的子图，我们称这个顶点式一个***关节点***。）（如果一个图中含有两个以上的顶点，但是不含有关节点，我们称这个图是**双连通**的）（如果一个图含有三个以上的顶点，但是不含有关节对，我们称这个图是**三连通**的）

2.将双连通分支在**关节点对**处分裂为一系列子分支，如果关节点对之间不存在边（即没有约束），那么在它们之间增加一条虚边，递归处理，生成所有可能的**子刚体**；

3.求解每一子刚体，然后再将子刚体拼接成整个刚体。

算法终止于所有子图不能继续分解为止，如果最后得到的都是三角形，那么原约束问题可以在二次的时间复杂度里构造；如果最后还有三连通子图存在，就不能在二次时间复杂度里构造。

参数化特征建模

问题：传统的基本体素不符合设计人员的要求；难以有效支持 CAM 等下游应用

一、特征概念

定义：特征是零件上具有**特定工程意义**的几何形状。包含丰富的工程语义,他是在更高层次上对几何形体上的凹腔、孔、槽等的集成描述.

特性：是零件/三维模型的一个特性构成部分，必须在模型上有所体现；

对应于特定的形状，采用参数化表示；具有可预测的特性，支持语义推理等智能行为；

与工程领域相关；特征数目没有限定：设计特性；制造特性；装配特性；分析特性。

二、特性表示

1.属性：几何表示（B-rep）

定形、定体约束

二维草图（定形和定位约束）

继成参数

非几何信息、公差、功能、材料等。

2.方法：构造几何表示的方法

实例化方法，参数继承发方法，语义维护方法，避免特征被破坏，特征识别方法

继承规则、有效性规则、识别方法

三、特征模型

解决重复表示 feature 的问题，如 100 个 hole

可以通过一个基本特征和一个由分布规则定义的具有相同形状而仅位置不同的一组特征

基于特征表示产品的模型，常见的是**特征图**：以特征为节点，以特征之间的关系为弧。

四、特征建模

1.自动特征识别

从 B-rep 中自动识别出特征，建立特征模型。

几何造型系统（用户）----->几何模型------>特征识别（特定特征）----->特征模型

难题一：用户自定义特征；难题二：三维参数法方法，用户给了草图和几何约束，怎么有效求解出参数化方法；基于过程式语言定义特征的参数化表示以及实例化方法

拓扑元素永久命名：怎么给拓扑元素命名，维持一一对应关系

实际使用中，特征可能会增加或者消失，比如圆角可能会因为边出现了槽，从一整个长的圆角边变成了两个圆角短的圆角边

2.特征设计：基于特征进行产品设计，建立产品的特征模型和几何模型。

图略。

用户自定义特征的定义与实例化：

（1）类定义（如扫成----拉伸或旋转）

基本特性：孔、槽等

用户自定义：基于特征的组合

（2）基于脚本语言定义：几何模型、几何约束及其求解方法

（3）陈述式方法：用户交互定义库特征的三维几何约束

模型：所有库特征中通用的三维约束求解方法进行实例化。

语义模型

陈述式： 语义模型是在关系模型基础上增加全新的数据构造器和数据处理原语，用来表达复杂的结构和丰富的语义的一类新的数据模型。

（1）几何约束模型的有效性

种类多

（2）方程组、局部方程组

特征模式（Feature Pattern）

可通过一个基本的特征元素和一个复制规则定义的具有相同结构和尺寸，仅位置不同的一组特征

定义简单：一个基本特征

一个位置分布规则：如图分布、阵列分布

基于图的特征识别算法（Joshi & Chang）

1. **属性**(弧的凸凹性)面邻接图（AFAG）

以面为结点，以面之间的相邻关系为弧，以边的凹凸性作为弧的属性的图

凸边：相邻两个面之间面向体内的夹角小于 180

凹边：相邻两个面之间面向体内的夹角大于 180

盲槽/通槽/盲台阶：根据凹边来识别

2. 方法步骤 （这里的弧其实就是邻接面之间的边）

a. 构建每类特征的 AFAG

b. 构建待识别零件的 AFAG

c. 对零件的 AFAG 进行子图分解：删除凸面与相关弧（删除所有凸面（周围所有边都是凸边））

d. 将分解出的子图与 AFAG 库特征的 AFAG 进行匹配，识别出加工特征。

问题

相交特征的识别：难识别，不唯一性

有的特征带凸面，因此通过删除凸面的子图匹配有时不适用

发展方向

长期目标：更快（运行速度快）、 更强（功能、可靠性）、更方便（用户友好、智能）

更强：能够有效表示、构建、处理高复杂度的几何模型；集合操作的健壮性；支持真三维参数的建模，现在主要还是通过二维参数约束三维；支持异质实体建模

更快：三维 CAD 模型检索与重用；基于设计历史的快速修改；分布式计算的几何建模算法

更方便；基于单位的三维 CAD 建模；三维交互；保证有效性的直接建模

更智能：AI+CAD；面向应用需求的智能形状分析与处理；智能重建；特征模型的语义维护

具体值得研究的问题：

1. 高度复杂几何模型的**快速（分布式计算如 GPU，云计算平台；非 B-rep 核心表示）**构建与处理；

2. 直接建模（独立于设计历史的编辑、修改）；

3. 三维 CAD 模型的**重用**（检索）；

4. 三维 CAD 模型的**重建**（视图重建、扫描重建、图像重建）；

5. 建模的可靠性（符号计算）；

6. 真三维参数化建模（自由形状特征的参数化、参数有效域的确定）；

7. 鲁棒的 CAD 模型交换（B-rep 交换的修复、参数化特征模型的交换）；

8. 异质实体建模；

9. 基于草绘、VR 的建模