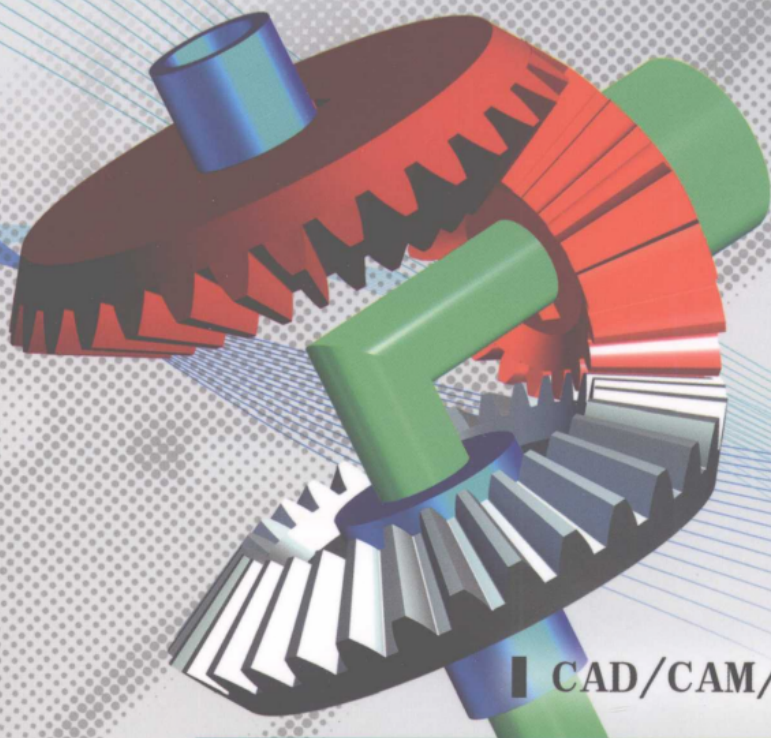




■ CAD/CAM/CAE工程应用丛书 ■

图表详解 UG NX 二次开发



■ 莫 蓉 常智勇 刘红军 杜 微 编著



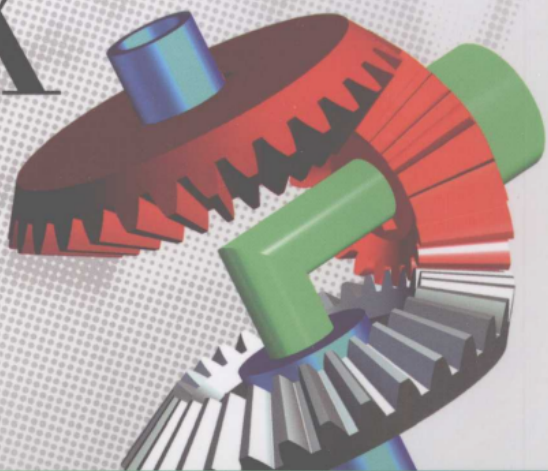
电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

■ CAD/CAM/CAE工程应用丛书 ■

图表详解

UG NX

二次开发



© 分社网址: jdjt.phei.com.cn



责任编辑: 范子瑜
封面设计: 徐海燕



本书贴有激光防伪标志, 凡没有防伪标志者, 属盗版图书。

ISBN 978-7-121-06672-6



9 787121 066726 >

定价: 36.00 元
(含光盘 1 张)

CAD/CAM/CAE 工程应用丛书

图表详解 UG NX 二次开发

莫蓉 常智勇 刘红军 杜微 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

Unigraphics NX (简称UG) 系统是应用最为广泛的大型 CAD/CAE/CAM 集成化软件之一, 为适应广大用户的特殊需要, 系统的二次开发是必不可少的。

本书从 CAD 应用开发的角度, 结合 UG (CAD 部分) 的开发文档, 论述了 UG 软件二次开发的基础和主要功能, 并给出了实例。主要内容包括特征建模、草图、自由形状特征、工程制图、装配、KF 等。

本书是一本 UG 的二次开发实用指导书, 示例丰富、实用, 语言通俗易懂, 对在校学生及 UG 系统二次开发人员都是一本难得的实用参考手册和应用指导书。本书适用于具有一定 C 语言基础和 UG 应用基础的人员。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

图表详解 UG NX 二次开发/莫蓉等编著. —北京: 电子工业出版社, 2008. 6

(CAD/CAM/CAE 工程应用丛书)

ISBN 978-7-121-06672-6

I. 图… II. 莫… III. 计算机辅助设计-应用软件, UG NX-程序设计 IV. TP391.72

中国版本图书馆 CIP 数据核字(2008)第 068383 号

责任编辑: 范子瑜

印 刷: 北京市海淀区四季青印刷厂

装 订: 北京鼎盛东极装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 18 字数: 438 千字

印 次: 2008 年 6 月第 1 次印刷

印 数: 4000 册 定价: 36.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

Unigraphics NX (简称 UG) 软件是目前国内外应用最为广泛的大型 CAD/CAE/CAM 集成化软件之一, 功能强大, 内容丰富, 涵盖了从设计、分析、加工、管理等各个领域。它的领先技术始终位于先进制造领域的前沿, 几乎每年推出的新版本, 都反映了当前该领域发展的最新成果。

几十年来, UG 软件在客户需求的驱动下, 发展迅速, 今天的 UG 软件汇集了航空、航天、汽车工业的专业经验, 并经历了飞机、汽车多个大型工程项目的实际考验, 被公认为世界一流的企业级集成的 CAD/CAE/CAM 软件。

UG 系统是一个通用的 CAD/CAE/CAM 集成系统, 在我国航空、航天、汽车、机械、家电行业得到了广泛应用。但是由于客户的特殊需求和面向专业化开发的要求, 仍然需要在其基础上进行一定的二次开发工作, 基于 UG 系统的二次开发已经成为拓展用户应用范围、将用户的设计知识融入设计软件、满足客户化特殊需求的一个非常方便的开发平台。虽然 UG 提供了丰富的 API 开发帮助文档, 但是开发人员还是迫切希望有一本实用的介绍 UG 二次开发的中文入门指导书。

此书的编写, 正是为了满足这些人员的需求, 主要面向 CAD 工程技术人员和 UG 系统的二次开发人员, 目的是使人们在学习本书以后, 能够较快地掌握 UG (CAD 部分) 的二次开发方法, 从而提高开发效率。

本书仅涉及了 UG (CAD) 常用部分的开发, 并没有包罗 CAD 的全部内容。用户在学习基本内容后, 应当多上机练习, 加深对 API 函数的理解和使用。希望读者在掌握基本内容基础上能够举一反三, 利用 UG 的开发文档, 与工程实际相结合, 取得更大收获。

本书共由 12 章组成, 其中第 1 章介绍了参数化特征造型的基础; 第 2 章介绍了 VC++ 的调试环境; 第 3 章介绍了 UG API 的开发框架; 第 4 章介绍了特征造型和自定义特征等内容; 第 5 章介绍了曲线构建和曲面造型方法; 第 6 章介绍了 UG 提供的菜单制作和界面设计方法以及基于微软 MFC 的界面设计方法; 第 7 章介绍了 UG 的装配以及 WAVE 几何链接器的开发; 第 8 章介绍了文件的各种操作功能的开发; 第 9 章介绍了属性的开发方法; 第 10 章介绍了表达式的开发方法; 第 11 章介绍了工程制图、视图、标注的开发内容; 第 12 章介绍了知识融接技术 (KF - Knowledge Fusion) 的开发方法。

本书的第 1 章由莫蓉编写, 第 2、3 章由常智勇编写, 第 4 章由李永毅、万能和赵岩编写, 第 5 章由罗明编写, 第 6、12 章由杜微编写, 第 7 章由陈泽峰和张岩编写, 第 8 章由赵岩编写, 第 9 章由郝李楠编写, 第 10 章由王红霞编写, 第 11 章由刘红军编写。光盘为各章的实例程序。

UG 系统在不断发展, 软件的版本更新速度也比较快, 但是 API 函数的变化不大, 读者在使用时可以参照 UG 提供的 API 帮助文件使用新的函数。一般来说, 最基本的操作功能的 API 在版本的升级中变化不大, 所以本书也适合于以后版本的 API 开发。

由于作者水平有限, 书中可能存在错误, 敬请广大读者批评指正。

作 者

目 录

第 1 章 参数化特征造型基础	1
1.1 绪论	1
1.2 参数化特征造型的基本概念	1
1.2.1 几何模型类型	1
1.2.2 参数化特征模型	3
1.3 曲面造型的基本概念	4
1.3.1 自由曲线	4
1.3.2 自由曲线的常用术语	4
1.3.3 自由曲面	6
1.3.4 自由曲面特征中常用术语	6
第 2 章 VC 编程环境简介	8
2.1 Visual C++ 6.0 的开发环境	8
2.1.1 VC++6.0 简介	8
2.1.2 VC6 项目的建立	9
2.1.3 VC6 项目环境的设置	9
2.1.4 VC6 程序的编译	10
2.1.5 VC6 程序的运行	11
2.1.6 VC6 程序的调试	11
2.2 VC6 编程举例	12
2.2.1 建立程序框架	12
2.2.2 编写代码	13
2.2.3 调试和排错	14
2.2.4 发布	16
第 3 章 UG/Open API 编程框架	17
3.1 UG API 基础	17
3.1.1 UG API 基本概念	17
3.1.2 API 帮助文件	17
3.1.3 开发语言	19
3.2 内部和外部 UF 的使用方法	19
3.2.1 User Exit 的概念	20
3.2.2 ufusr 入口	20
3.2.3 ufsta 入口	23
3.2.4 外部模式的功能	28
3.3 UF 中常见的数据类型	29
3.3.1 UF 函数命名规则	29

3.3.2	UF 中数据结构标识	30
3.3.3	错误号	30
3.4	UG 系统中对象的基本概念及区别	31
3.4.1	对象	31
3.4.2	Object、Body 和 Feature 的区别	31
3.4.3	UF 的 License	32
3.4.4	坐标转换	32
第 4 章	特征设计	33
4.1	UG 特征的分类	33
4.1.1	UG 特征的分类	33
4.1.2	特征的常用操作	34
4.2	体素特征的创建	34
4.2.1	创建块	35
4.2.2	块特征举例	35
4.3	扫描特征的创建	35
4.3.1	拉伸特征的创建	36
4.3.2	回转特征的创建	38
4.4	成型特征的创建	40
4.4.1	孔特征的创建	41
4.4.2	创建孔特征举例	41
4.5	引用特征的创建	43
4.5.1	线性阵列特征的创建	43
4.5.2	镜像特征的创建	45
4.6	草图特征及其操作	46
4.6.1	创建草图	47
4.6.2	初始化草图	47
4.6.3	添加尺寸	47
4.6.4	添加尺寸约束	48
4.6.5	添加几何约束	49
4.6.6	添加对象	49
4.6.7	查询表达式	50
4.6.8	查询几何约束	50
4.6.9	更新草图	50
4.6.10	退出草图	51
4.6.11	程序实例	51
4.7	基准特征及其操作	53
4.7.1	创建固定基准面	53
4.7.2	创建固定基准轴	54
4.7.3	创建关联基准面	54

4.7.4	创建关联基准轴	55
4.7.5	程序实例	56
4.8	操作特征的创建	58
4.8.1	倒圆特征	58
4.8.2	倒角特征	58
4.9	自定义特征	60
4.9.1	自定义特征基本概念	60
4.9.2	用程序实例化自定义特征	60
4.9.3	查询自定义特征	67
4.9.4	修改自定义特征	68
4.9.5	综合举例	69
4.10	特征的访问	71
4.10.1	特征的查询	71
4.10.2	特征的遍历	72
4.11	特征参数的修改	73
4.11.1	函数介绍	73
4.11.2	特征参数修改举例	74
4.12	特征的删除	75
4.12.1	删除特征函数	75
4.12.2	删除特征举例	75
4.13	特征操作的综合实例程序	78
第5章	自由形状特征	82
5.1	自由形状特征简介	82
5.2	曲线的创建	82
5.2.1	常用曲线的结构说明	82
5.2.2	解析曲线的创建	83
5.2.3	样条曲线的创建	86
5.2.4	其他样条曲线的构建	89
5.3	自由曲线的访问	93
5.3.1	点、直线、圆弧数据的访问	93
5.3.2	样条数据的访问	95
5.4	曲线的修改	101
5.5	自由曲面创建	103
5.5.1	基于点的曲面创建	104
5.5.2	基于线的曲面创建	106
5.5.3	基于面的曲面创建	110
5.6	曲面的访问	110
5.6.1	UF_MODL_ask_face_data	110
5.6.2	UF_MODL_ask_face_parm	111

5.6.3	UF_MODL_ask_face_props	112
5.6.4	UF_MODL_ask_face_uv_minmax	112
5.6.5	UF_MODL_ask_face_smoothness	113
5.7	自由曲面的修改	114
5.7.1	UF_MODL_edit_bsurf	114
5.7.2	UF_MODL_trim_sheet	114
5.7.3	UF_MODL_update	115
5.8	自由曲线曲面的实例程序	115
5.8.1	实例一:创建圆弧、直线及过渡圆角	115
5.8.2	实例二:通过曲线创建曲面	118
5.8.3	实例三:曲线、曲面数据访问	121
第6章	UG/Open API 菜单和界面制作	125
6.1	MenuScript 菜单定制	125
6.1.1	脚本语言介绍	125
6.1.2	菜单脚本文件	125
6.2	UIStyler 用户界面制作	127
6.2.1	UIStyler 可视化界面	127
6.2.2	程序框架介绍	138
6.3	MFC 界面制作举例	140
6.3.1	MFC 程序框架	140
6.3.2	MFC 界面与程序交互	147
6.4	常用界面函数	148
6.4.1	消息对话框	149
6.4.2	文件操作对话框	149
6.4.3	对象选择对话框	150
第7章	装配操作	152
7.1	装配的基本概念	152
7.1.1	装配中常用术语	152
7.1.2	装配中常用数据结构	153
7.2	装配树的操作	157
7.2.1	装配树的结构和对象访问函数	157
7.2.2	遍历装配结构树	157
7.2.3	修改装配结构树	161
7.3	装配坐标系的操作	163
7.3.1	坐标系的分类和矩阵的结构	163
7.3.2	坐标系操作的常用函数	165
7.3.3	矩阵操作的常用函数	168
7.4	配合关系的操作	172
7.4.1	装配配合关系的概念	172

7.4.2	配合关系的操作	173
7.4.3	部件的阵列操作	176
7.5	其他常用装配操作	178
7.5.1	引用集的操作	178
7.5.2	爆炸视图的操作	181
7.6	WAVE 几何链接	184
7.6.1	WAVE 几何链接简介	184
7.6.2	典型函数介绍	185
7.6.3	应用实例	186
7.7	对象操作的实例程序	188
7.7.1	装配结构树遍历实例	188
7.7.2	装配条件遍历实例	189
7.7.3	部件阵列遍历实例	191
第8章	文件操作	194
8.1	基本数据类型	194
8.1.1	结构类型	194
8.1.2	枚举类型	196
8.2	文件的基本操作	197
8.2.1	文件的基本操作	197
8.2.2	导入/导出操作	198
8.2.3	遍历查询操作	200
8.2.4	状态操作	201
8.3	零件族的操作	202
8.3.1	查询操作	202
8.3.2	设置操作	205
8.4	文件的存储历史	206
8.4.1	查询操作	206
8.4.2	设置操作	207
8.5	文件操作的实例程序	208
8.5.1	基本操作实例	208
8.5.2	零件族操作实例	213
8.5.3	存储历史操作实例	214
第9章	属性操作	216
9.1	属性的基本概念	216
9.1.1	属性	216
9.1.2	常用的属性操作及函数	216
9.1.3	属性类型	217
9.2	属性操作的典型数据结构	218
9.2.1	UF_ATTR_value_s	218

9.2.2	UF_ATTR_value_u	218
9.2.3	UF_ATTR_part_attr_s	218
9.3	属性操作	219
9.3.1	部件属性对象标识的查询	219
9.3.2	属性的创建	219
9.3.3	属性的查找	220
9.3.4	属性的读取	221
9.3.5	属性的删除	223
9.4	程序实例	224
第10章	表达式	228
10.1	表达式的基本概念	228
10.1.1	表达式的组成	228
10.1.2	数学表达式	228
10.1.3	条件表达式	228
10.1.4	几何表达式	229
10.1.5	运算符	229
10.2	表达式的创建、访问和修改	230
10.2.1	表达式的创建	230
10.2.2	表达式的访问	231
10.2.3	表达式的修改	233
10.3	表达式操作的实例程序	234
第11章	二维工程图	236
11.1	二维工程图 API 的基本参数	236
11.1.1	用于制图元素分类的整型数组 MPI 参数	236
11.1.2	用于制图元素定义数据的实型参数值数组 MPR	240
11.1.3	用于显示的整型参数数组 DPI	241
11.1.4	用于字符串倾斜角度的实型数组	241
11.2	二维工程图的典型数据结构	242
11.2.1	UF_DRAW_info_s	242
11.2.2	UF_DRAW_view_info_s	242
11.2.3	UF_DRAW_arrow_parms_s	243
11.3	二维工程图纸	243
11.3.1	二维工程图纸的创建	243
11.3.2	二维工程图纸信息的查询与图纸修改	244
11.3.3	二维工程图纸的删除	246
11.4	工程视图的操作	247
11.4.1	二维工程图视图的创建	247
11.4.2	二维工程图视图的移动	249
11.4.3	二维工程图视图的比例查询与设置	250

11.4.4	二维工程图视图的更新	251
11.5	二维工程图制图元素的操作	253
11.5.1	基本制图参数的查询与设置	253
11.5.2	尺寸参数信息的查询	256
11.5.3	尺寸标注	257
11.5.4	其他标注对象的创建	259
第12章	KF 开发	262
12.1	知识融合基本概念	262
12.1.1	UG 的知识融合 (KF) 概念	262
12.1.2	KF 编程方式	262
12.2	知识融合的语法	263
12.2.1	DFA 文件基本结构	263
12.2.2	KF 属性	264
12.2.3	类	265
12.2.4	表达式	266
12.2.5	条件语句	267
12.2.6	循环语句	267
12.3	可视化编程	269
12.3.1	KF 工具条	270
12.3.2	知识融合导航器 (KFN)	270
12.4	手工编程	272
12.4.1	KF 应用程序基本结构	272
12.4.2	KF 应用程序部署	272
12.4.3	KF 应用程序运行	273
12.5	KF 的实例程序	274
12.5.1	KF 程序与 UIStyler 对话框关联	274
12.5.2	KF 中的对象选择	275
参考文献		277

第1章 参数化特征造型基础

1.1 绪论

UG NX 软件是一个集成化的 CAD/CAE/CAM 系统软件,它为工程设计人员提供了非常强大的应用工具,可以对产品进行设计、工程分析、工程图绘制、数控加工程序编制等,同时还提供了很多面向专业的应用工具(例如注塑模、钣金设计等)。

虽然 UG NX 在发展过程中不断推出新版本,功能也在不断增加和强大,但是由于企业的行业特点和专业不同,新的需求范围广,需要一些专业化、个性化的功能和工具。UG NX 作为一个面向机械行业的通用 CAD/CAE/CAM 软件,需要在此基础上进行二次开发以满足实际的需要。从二次开发的角度讲,新版本的 API 函数的变化不大,所以本书提供的开发方法仍然可以应用于以后的发展版本。

从企业应用的角度讲,应该遵从如下原则:应尽可能直接使用 UG NX 的已有功能,如不能满足要求,应首先选择定制开发的方式,只有在上述情况都不能满足的情况下才应进行二次开发,以减少企业专用软件的开发成本和以后的维护成本。

本书作为一个 UG 二次开发基础的指导书,将介绍基于 UG NX 关于 CAD 方面二次开发的基本方法。本章首先介绍参数化特征造型基础和曲面造型基础。

1.2 参数化特征造型的基本概念

目前各种三维 CAD 软件系统都具有参数化特征造型的功能,而三维参数化实体造型的理论基础是实体造型原理。在进行 UG 二次开发中,经常会遇到一些和实体造型有关的术语,在 API 函数中也经常碰到各种表示实体几何造型中的数据结构,因此本节主要介绍实体模型的表示方法。

1.2.1 几何模型类型

几何建模指的是在计算机上描述和构造几何对象的方法,其模型的表达类型分为:线框模型、表面模型、实体模型。

(1) 线框模型 在三维模型中按照一定的拓扑关系将点和棱边有序连接的模型,在计算机内描述一个三维线框模型必须给出两类信息:顶点(Vertex)表——存储模型中各项



点的三维坐标；边（Edge）表——存储模型中的各棱边，由指针指向各棱边的顶点，实际上表达了简单的拓扑连接关系。

(2) 表面模型 数据结构是以“面—棱边—点”三层信息表示的，表面（Face）由封闭的棱边围成，棱边由点构成，它们形成了一种拓扑关系。表面模型的曲面可以是简单的解析曲面，也可以是自由曲面。构造自由曲面的方法有很多，UG NX 使用的是非均匀有理 B 样条（NURBS: Non - Uniform Rational B - Spline）方法。

表面模型避免了线框模型的二义性。由于定义了面，可以根据不同的观察方向消除隐藏线和隐藏面；可以对面着色，显示逼真的色调图形；还可以利用面的信息进行数控加工刀具轨迹的计算。在数控加工中刀具轨迹的计算和物体的表面特性有很大关系，直接影响到刀具轨迹的生成，因而表面建模主要描述物体的表面特性，如曲率连续性、法矢、光顺性等。在 UG NX 中对于独立存在的曲面称为片体（Sheet Body）。而作为实体表面的曲面则是 Face。

(3) 实体模型 从理论上讲，以“体—面—环—棱边—点”的 5 层拓扑结构信息表示模型即可表达实体，但是由于实际数据结构处理的方便和需要，一般的实体模型的表达都对此进行了扩充，例如 ACIS（几何造型核心软件）采用了 7 层结构，如图 1-1（a）所示。UG NX 采用了 Parasolid 几何造型核心软件，如图 1-1（b）所示。

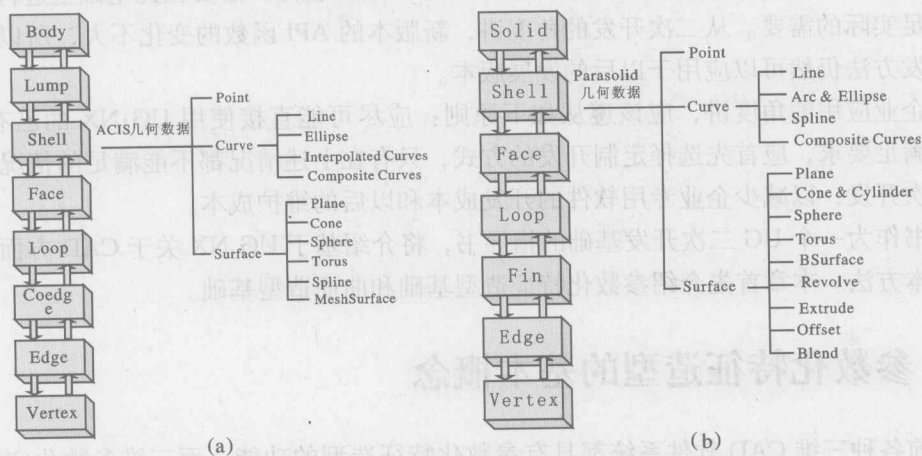


图 1-1 ACIS 和 Parasolid 的几何造型核心层次结构

以 Parasolid 为例，体（Solid）是由表面围成的封闭空间，壳（Shell）是构成体的表面集合，表面（Face）是由棱边围成的封闭区域，环（Loop）是封闭的棱边集合，包括外环与内环，例如一个孔在一个表面上形成了一个内环，表面本身的棱边序列是外环。共边（Fin）是表达翼边数据结构的公共边，边（Edge）是由顶点构成，顶点（Vertex）是最底层的拓扑结构。在图 1-1（b）的左侧是几何模型的拓扑层次关系，表达了模型中各几何元素之间的连接关系，右侧是几何模型的几何类型信息，表示了点（point）、线（Curve）、面（Surface）的内容，几何信息是可以几何数据、方程、公式、表达式表达的一类信息，例如点可以用 (x_i, y_i, z_i) 表示，直线可以用直线方程表示，BSurface 可以用 B - Spline 表示等。

实体建模最常用的是边界描述法 (Boundary Representation, B-Rep) 和构造性实体几何法 (Computed Structure Geometry, CSG), 在 UG NX 中采用的是 B-Rep 方法。

实体模型的表达信息丰富, 除了能实现表面模型的功能外, 还能够满足物理性能计算, 例如质量与质心计算、重量及工程分析的需求。在产品设计中, 实体建模技术更符合人们对真实产品的理解和习惯。

1.2.2 参数化特征模型

1. 特征模型

实体建模方法在表示物体形状和几何特性方面是完整有效的, 能够满足对物体的描述和工程的需要, 但是从工程应用和系统集成的角度看, 还存在一些问题: 实体建模中的操作是面向几何的 (点、线、面), 而非工程描述 (例如槽、孔、凸台的构造特征), 不方便工程技术人员的使用。在 UG 中主要利用形状特征造型, 例如孔、槽、凸台、拉伸等。形状特征的基础还是实体模型, 但是进行了工程语义的抽象, 即语义 + 形状特征。

UG NX 中, 除了形状特征外, 在几何形位公差方面也提供了特征功能。几何形位公差的传统表示方法一般是在二维视图中通过标注功能反映这些非几何信息, 以便为工艺设计和数控加工提供公差信息。

一个特征造型系统是很复杂的, 不同的 CAD 系统, 即使底层几何核心平台是相同的, 开发的 CAD 系统还是有很大的不同。例如, 基于 Parasolid 几何核心的 UG NX 和 Solidworks 系统, 它们的功能、特点、风格各不相同。开发一个 CAD 系统, 除了实体造型的核心外, 还需要融合先进的特征设计方法、参数化技术、图形学技术、交互技术等。

2. 参数化

在产品设计中, 设计实质上是由给定的功能、结构、材料及制造等方面的约束进行描述, 经过反复迭代、不断修改从而求得满足设计要求的解的过程。除此之外, 设计人员经常碰到如下情况: 许多零件的形状具有相似性, 区别仅是尺寸的不同; 在原有零件的基础上做一些小的改动来产生新零件; 设计一分析仿真的迭代过程经常需要进行修改。这些需求采用传统的造型方法是难以满足的, 一般只能重新建模, 而参数化方法提供了设计修改的可能性。目前的三维 CAD 系统都包含参数化功能, 而且大部分参数化功能和特征设计结合到一起, 使特征模型成为参数的载体。

参数化设计一般是指设计对象的结构形状基本不变, 而用一组参数来约定尺寸关系。参数与设计对象的控制尺寸有显式对应关系, 设计结果受尺寸驱动。

参数化设计的主要特点是:

- ✱ 基于特征: 将某些具有代表性的几何形状定义为形状特征, 并将其所有尺寸设定为可修改的参数, 形成实体, 以此为基础来进行更为复杂的几何形体创建。
- ✱ 全尺寸约束: 将形状和尺寸联合起来考虑, 通过尺寸约束来实现对几何形状的控制。在 UG NX 中, 存在三维特征的参数化和二维草图的参数化。
- ✱ 参数化设计可以在不同级别实现。在 UG NX 中, 零件级参数化, 通过草图实现



截面形状的修改,通过三维特征的参数修改实现实体模型的修改。装配级参数化,通过零件之间的装配约束实现零件之间的相对方位修改。系统级参数化通过 WAVE 实现装配树不同层次节点上的几何关联设计。

1.3 曲面造型的基本概念

1.3.1 自由曲线

在飞机、汽车、船舶等工程设计中,物体的表面用解析表达式是不能满足要求的,因此需要一种能够表达非解析形式的曲线(曲面)来表达曲线(曲面)的形状,这类曲线(曲面)就称做自由曲线(自由曲面)。自由曲线(曲面)是一种由逼近、插值、拟合方法等得到的曲线(曲面)。

UG NX 自由形状特征(Free Form Feature)中的许多曲面生成是利用已有曲线经过不同的操作实现的,如过曲线(Through Curves)等,因此自由曲线是自由曲面的基础,而自由曲线指的是样条曲线,它有一些特殊的几何性质和一些重要的基本概念。

曲线的表示分为几类:

- ✱ 规则曲线与基本曲线:直接由数学方程定义的解析曲线,如直线、二次曲线(圆弧、椭圆、双曲线等),在 UG NX 中属于基本曲线类。
- ✱ 计算类曲线:由计算生成的曲线,如相交曲线、投影曲线、偏置曲线等,一般是在已有曲线的基础上通过计算得到的曲线。
- ✱ 构造类曲线:由点生成的曲线,用户必须输入一系列点(UG 中分别称为极点、定义点),通过插值、逼近、拟合构造曲线,一般指自由曲线。
- ✱ 规律曲线:预先定义参数方程,通过给定的参数定义曲线,用户必须输入参数生成曲线,如螺旋线。

1.3.2 自由曲线的常用术语

(1) 非均匀有理 B 样条 在 UG 中样条曲线都是用非均匀有理 B 样条 NURBS (Non-Uniform Rational B-Spline) 表示的,它能够将二次圆锥曲线和自由曲线统一起来,并且给用户提供了更多的控制调整曲线形状的能力。

(2) 曲线的次数和多项式表示 每一条样条曲线都是由一个多项式表达的,多项式中的最大指数称为多项式的次数(Degree),它是构造样条的一个基本输入参数,例如下面的多项式具有不同的次数:

$$7x^2 + 5x - 3 = 35 \quad \text{次数为 } 2$$

$$2t^3 - 3t^2 + t = 6 \quad \text{次数为 } 3$$

(3) 曲线的段数、次数与点数的关系 在 UG 中一条曲线可以是单段曲线,也可以由多段曲线段组成。单段样条的次数由输入点的个数决定,即次数 = 点的个数 - 1,如图 1-2 (a)

的曲线次数为18次。由于次数严格受点数的控制,所以单段样条应用并不多,在UG NX中单段样条的最高次数为24。多段样条是指一条曲线由2段以上的曲线段组成,在曲线段的拼接处自动保持曲率连续,用户可指定次数,但次数必须小于输入点数,多段样条的输入点数无限制。例如图1-2(b)所示有7个点,可以指定为3次样条,由此得到的曲线为多段曲线段拼接而成。3次样条曲线是应用最为广泛的一类曲线。建议尽可能使用3次样条。

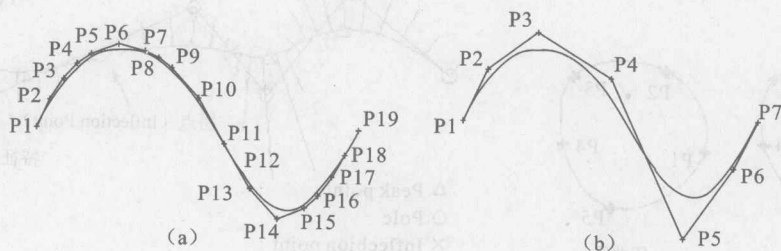
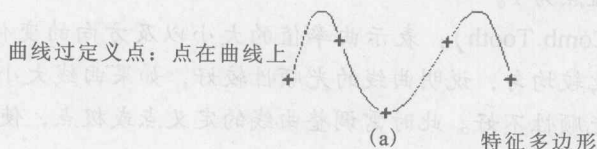


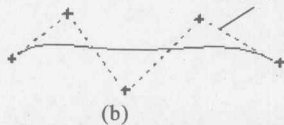
图1-2 次数与段数

(4) 定义点、极点与节点 这里我们均以UG NX的术语进行说明。

✱ 定义点 (Defined Point): 定义点定义了曲线的形状,数学上采用插值算法,保证这些点落在构造的样条曲线上,如图1-3(a)所示。工程上常称这些点为型值点。例如,飞机机翼的截面线及叶片叶身的截面线就是以定义点定义的。



由极点产生曲线: 特征多边形的首末点在曲线上



拟合曲线: 曲线仅通过端点

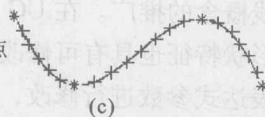


图1-3 点与曲线的关系

✱ 极点 (Poles): 又称为控制多边形顶点,数学上采用逼近算法,它大致描绘出曲线的变化方向,曲线只通过控制多边形的首末点,控制多边形的形状将影响曲线的形状。第一、第二点控制了曲线的首端斜率,最后两点控制了曲线的末端斜率,如图1-3(b)所示。

✱ 节点 (Knot Points): 是多段样条的拼接点,即每小段样条曲线段的端点,单段样条只有两个节点。多段样条有多个节点。

(5) 开 (Open) 曲线与闭 (Closed) 曲线 开曲线是指样条曲线的首末端点不重合,而闭曲线是指曲线的首末端点重合,如图1-4所示。