**3.1 为什么要参数化多项式曲线？** 2020年9月10日12点14分

为了能够设计,操纵和渲染曲线和曲面,我们必须选择一些数学模型来定义曲线和曲面.我们将采用的模型类型包括根据某种方程组指定曲线或曲面S,该曲线或曲面用于确定哪些点位于S上,哪些不在S上.鉴于我们采用了这种模型,我们面临以下列出的许多问题.

**一块还是几块连接在一起**？我们可以通过一个方程组对整个形状S进行建模.这通常在数学上更简单,并且非常适合非常简单的形状.但是对于复杂的形状(可能是复合形状),通常会发生组合爆炸,这使得该方法不切实际.此外,这种方法不是模块化的,并且不适用于解决可能需要修改某些部分而使其他部分保持不变的问题.通常通过将曲线或曲面分解为更简单的零件,并指定这些零件如何以某种程度的平滑度连接在一起来解决此类设计问题.在CAGD行话中,我们用样条曲线对复合形状进行建模.但是,由于将复合形状分解为更简单的部分,因此了解如何有效处理这些更简单的构建块非常重要,我们现在将集中精力为单个曲线或曲面建模.稍后,我们将回到样条曲线.

**参数模型还是隐式模型**？在数学上,当通过方程组对形状S进行建模时，我们将这些方程视为定义了某个函数F,并且将形状S视为该函数的范围或该函数的零轨迹（即，在F映射下为“零”的点集）.稍微简化一下,形状S存在于某个维数为的对象空间(通常是仿射平面或3D仿射空间)中，并且形状的维数为(对于曲线,s = 1;对于平面,s = 2）.刚才提到的两个模型用函数来定义S,其中是另一个称为参数空间的空间(对于曲线,通常,其中表示仿射线,对于,其中A2表示仿射平面）或函数，在这种情况下,P没有标准名称.在的第一种情况下,我们说我们有一个参数模型,在的第二种情况下,我们说我们有一个隐式模型.让我们检查每个模型.

在**参数模型**中,将由函数指定的维度为s的形状S定义为函数F的范围,其中的维度为.因此,参数空间P同样具有维度.对于某个参数值(可能有多个参数值),形状S上的某个点可以表示为.例如,函数定义为

表示仿射平面中的直线.函数定义为

表示仿射平面上的抛物线.

对于空间曲线的一个更好的例子,函数定义为

表示在3D仿射空间中称为扭曲立方的曲线.对于曲面的示例,函数定义为

代表所谓的双曲线抛物面.粗略地说,它看起来像一个马鞍(一个无限的马鞍!).

函数定义为

代表所谓的猴子鞍.

在隐式模型中,将由函数指定的维度为s的形状S定义为函数F的零轨迹,其中E的维度为,作为函数F的零集:

在这种情况下,空间是向量空间,并且具有某个维度.当然,是可能的.例如,如果是定义为

由于方程没有实数解,因此F定义了空曲线.为了避免这种情况,我们可以假设我们的空间是在复数域(或更笼统地说,是代数封闭域)上定义的.这将具有一定的数学优势,但对形状的可视化没有太大帮助,因为我们主要对形状的实际部分(曲线和曲面)感兴趣.因此,我们将做出一个工作假设,即我们仅考虑定义非空形状的函数.

这种假设存在一些严重的问题.对于可以确定任意代数方程式是否具有规则的检验,目前还不清楚.这些复杂性是更多关注参数模型的动机之一.但是,隐式模型对于处理某些类别的问题更为自然,因为直接可以使用形状的隐式方程.

作为隐式定义的曲线的简单示例,函数定义为

在所有成立,定义了一条直线(实际上,与上面定义的同一直线).函数定义为

定义了与上面相同抛物线,其中.

**还有一些示例这里简单地跳过**.

如果P的维数,则不必说S是函数F的零轨迹：E→P，因为函数F对应于d个标量值函数（F1，...，Fd），我们 通常说S由方程组定义

或S是上述方程组的零轨迹.对于另一个熟悉的示例,单位圆由等式隐式定义

**函数F的类别是什么**？尽管三角函数非常好,但是出于计算（算法）的原因，我们可能希望使用更简单的函数类.当然,我们应该只考虑足够可微的连续函数,以产生合理平滑的形状.可通过多项式定义的函数类别表现得很好,并且对于大多数用途而言已足够.当多项式不足时,我们可以转到由有理函数(多项式的分数)定义的函数类别,这对于大多数计算应用程序来说已经足够了. 事实上,处理有理数分数实际上已简化为处理多项式.从实践的角度来看,使用多项式很少会受到限制(合理域上的连续函数可以通过多项式函数来近似).因此,我们将处理多项式函数(和有理函数).

在隐式模型中,研究由多项式方程组定义的形状S本质上是代数几何的主要问题.这是一个令人着迷,令人尊敬并且非常困难的主题,远远超出了本课程的范围.强烈建议大胆的读者咨询富尔顿[37]或哈里斯[41].在参数模型中,由于多项式可以任意阶数微分,因此在某种意义上研究由参数多项式系统(或有理分数)的系统定义的形状S可以归因于另一个著名的微分几何.我们将非常适度地使用微分几何的基本概念,并更适度地使用代数几何的基本概念.实际上,我们主要使用的是多线性代数的某些元素.决定使用多项式(或有理数)函数后,还有一个问题.

**F的度数是多少**？ 在涉及许多将小块连接在一起(样条线)的大多数实际应用中,或的度对于小块就足够了.但是,对于单个零件,度可能会很高.通常,度的选择取决于所涉及的部分以及每部分的复杂程度.

由于对以多项式方程的零位点建模的形状的研究非常困难,因此我们将主要侧重于参数化建模的形状的研究.处理参数模型还有其他优点.例如,通过简单地考虑参数空间的子集,更容易“切出”形状的各个部分.同样,函数F：P→E可立即访问形状S上的点:每个参数值都会产生一个位于形状S上的点F（a）。对于隐式模型而言，情况并非如此F：E→P不能立即访问S中的点。给定点b∈E，确定b∈S通常是否需要求解代数方程组，这可能非常困难。不过，给定形状S的参数定义F：P→E，找到隐式定义G：E→P通常对于S是有用的。在F由有理函数给出的情况下，可以证明它总是有可能找到S的代数隐式定义G(实际上，正如我们将很快看到的那样，关于我们将曲线或曲面视为已定义的区域存在一些微妙之处。)。这称为隐式化,是使用结果的代数概念完成的.

应当指出,并非总是可以从隐式代数G：E→P表示S到参数代数定义F：P→E表示S。多项式函数的m = 2表示不等式Braic定义， 对于有理函数，当m = 3时。

还应注意，如果我们考虑参数曲线函数比多项式或有理函数F更一般,例如，A→E由仅连续的函数定义，则一些意外对象将显示为曲线轨迹 。 例如，Peano和Hilbert（还有其他人）表明存在空间填充曲线，即连续曲线F：[0，1]→A2，其迹线是整个单位正方形（包括其内部）！ 看到问题.这是假设函数F充分可微的良好动机.多项式肯定合格！

回顾平面曲线:直线,中的平面,抛物线,椭圆,双曲线

**定义3.1.1** 给定有限维的任何仿射空间和的任何仿射参考系,最大度为的(参数化)多项式曲线为映射,使得每一个,

其中,每个是中度的多项式.给定的的任何仿射参考系,最大度m的(参数化)多项式曲线段是多项式曲线(最大度m)的约束. 中的点的集合称为多项式曲线F的迹线,类似地,中的点的集合称为多项式曲线段的迹线.

为了简化表示,我们将视为实数,并写代替.直观地,通过使用多项式映射弯曲和扭曲仿射线来获得多项式曲线.应当注意,如果是定义了最大度m的多项式F的多项式的最大次数,则是可能的,仅要求.这个决定看似不寻常,实际上,稍后对于CAGD应用程序将很方便.例如,我们将需要连接可能不同度的曲线段,并且方便地将其中一些曲线段的度“提高”到一个共同的度.同样,在大多数情况下,我们对曲线的轨迹比对实际的参数表示更感兴趣.实际上,许多不同的参数表示形式可能会产生相同的轨迹,有时也称为与相关的几何曲线.

**3.2 度数为1和2的多项式曲线** 2020年9月10日13点11分

本节内容提供了一阶和二阶参数曲线转隐式形式的方法。

对于一阶曲线,即直线,其参数形式如下所示:

如果,则F的迹线收缩到单点.如果或,我们可以消除和之间的,得到隐式方程

这是一条直线方程.

现在让我们考虑m=2,即二次曲线.其形式为

因为我们已经考虑了的情况,所以我们假设或.我们首先证明通过改变坐标(相当于旋转),我们始终可以假设(或).如果且,令并考虑下面给出的矩阵R:

在坐标变换下

我们得到

如果,那么我们可以消除和之间的t,从而得到隐式表达.

其余内容略过.

**3.3 初见极形式** 2020年9月10日14点20分

本节内容一开始介绍了应用仿射变换的一阶曲线(直线)的极形式.

任意二次曲线

的极形式(对称,双仿射变换)为

使得.

一切都很好,但是我们获得了什么呢？ 我们得到的是,利用极坐标形式是对称且为仿射的事实,我们可以证明每条二次曲线完全由三个点(称为控制点)确定,此外,还有一种很好的算法可以确定任意点 这些控制点的曲线,只需使用三个线性插值步骤即可.因此,为简单起见,假设我们有一个二阶曲线F：A→A3,由三个二次多项式F1，F2，F3给出.正如我们刚才解释的那样,我们可以计算它们的极坐标形式f1，f2，f3，然后得到对称的仿射图f：A2→A3，使得对于所有X∈A，F（X）= f（X，X）.

让我们为选择一个仿射参考系,即两个不同的点(如果愿意,我们可以假设,但这不是必须的).就像我们已经说过的那样,每个可以唯一地表示为和的重心组合,例如,其中.

让我们计算

由于是对称且为仿射,我们得到

和的系数显然是对称的仿射函数,并且它们的总和为1,这很容易通过扩展乘积来验证

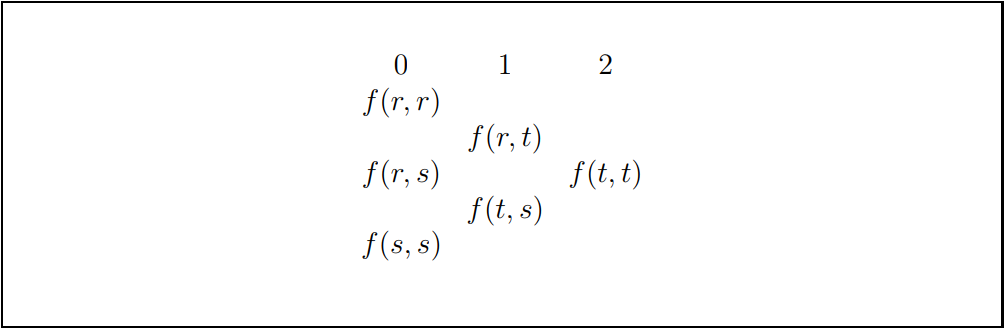
因为,我们得到

点和被称为控制点或B´ezier控制点,正如我们将看到的,它们在de Casteljau中起主要作用 算法及其扩展.如果我们令r = 0且s = 1，则t1 =λ1且t2 =λ2，因此，通过让t1 = t2 = t获得与f（t1，t2）对应的多项式函数,我们得到

多项式

被称为2度的Bernstein多项式.因此,F（t）也由控制点f（0，0），f（0，1）和f（1，1）和Bernstein多项式确定.顺便提及,这还表明二次曲线必须包含在一个平面中,该平面由控制点b0，b1，b2确定.

**3.4 初见de Casteljau算法** 2020年9月10日14点56分



该算法包括两个阶段.在第一阶段,我们通过线性插值计算两个点

且

其中从两个控制点和计算，从和计算,内插比为

由于对称,,所以在第二阶段,我们计算最终点

即从在第一阶段计算出的点和，内插比也为

因此,通过三个线性插值步骤,我们获得了曲线上的点.请注意,有两个控制点和在曲线上,但不在曲线上.

**剩下的内容是更详细的说明和示例**.

3.5 三阶多项式曲线 2020年9月11日10点02分

本节内容继承自3.2节，与本书的主题无关，因此略过

3.6 三阶多项式的分类 2020年9月11日

本节内容继承自3.5节，与本书的主题无关，因此略过

**3.7 再遇极形式** 2020年9月11日10点24分

首先,我们需要定义阶数为3的多项式的极形式(或开花).定阶数为3的任何多项式,

F的极性形式是对称三族函数,即对于的所有排列都取相同值的函数,即使得

其中每一个均为仿射映射,并使得

我们很容易地验证必须由

如果给定一个多项式三次曲线,由度的三个多项式确定,我们可以确定它们的极坐标形式,并获得对称的三重仿射映射,对于所有，使得.再次,让我们在中选择一个仿射基,其中,然后计算

由于是对称的且是三重仿射,我们得到

和的系数显然是对称的三重仿射,它们加起来等于1,因为可以通过展开乘积轻松验证

因为,我们得到

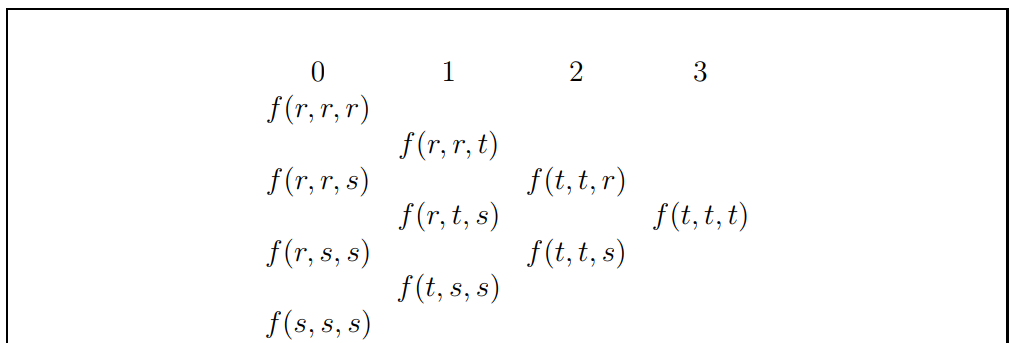
点和被称为控制点或B´ezier控制点.它们在de Casteljau算法及其扩展中起着重要作用.请注意,由定义的多项式曲线通过两个点和,但不通过其他控制点.如果我们令且,则,则通过令获得与相关的多项式函数.我们得到

多项式

是次数为3的伯恩斯坦多项式。它们构成次数为的多项式的向量空间的基础。因此，曲线上的点F（t）可以用控制点f（r，r，r表示） ），f（r，r，s），f（r，s，s）和f（s，s，s）以及伯恩斯坦多项式.但是，扩展de Casteljau算法更有用.

3.8 再遇de Casteljau算法 2020年9月11日13点51分

让我们假设三次曲线由控制点和(其中).给定任何.的计算可以安排在一个三角形数组中,如下所示,该三角形数组由三个阶段组成:



上面的计算通常是针对进行的,但是对于任意，即使在之外,它也同样有效.当在之外时,我们通常说是通过外推法计算的.

让我们经历计算的各个阶段.在第一阶段,我们计算三个点

插值比为

在第二阶段,由于对称,，我们计算出两个点

插值比同样为

在第三阶段,我们计算点

插值比仍然为

本章剩余内容均为例题和说明.