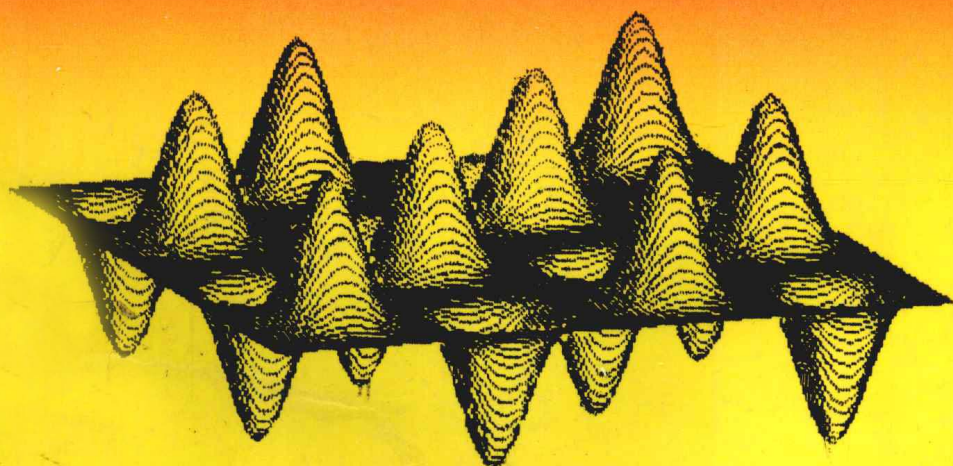


计算物理导引

赵金保 吕克璞 余庚荪 编著



兰州大学出版社

*AN INTRODUCTION TO
COMPUTATIONAL PHYSICS*

计 算 物 理 导 引

主编 赵金保

副主编 吕克璞 余庚荪

兰州大学出版社

(甘) 新登字第 08 号

计算物理导引

赵金保 主编

兰州大学出版社出版

(兰州大学校内)

甘肃省激光照排中心排版

兰州人民印刷厂印刷

甘肃省新华书店发行

开本: 850×1168 毫米 $1/32$ 印张: 4.5

1993 年 9 月 第 1 版

1993 年 9 月 第 1 次印刷

字数: 94 千字

印数: 1—1000 册

ISBN7-311-00633-3/O·84 定价: 4.65 元

· 引 言

计算物理是门新兴的学科。本世纪 40 年代第一台电子计算机问世后，大大促进了科学技术的发展。特别是最近二、三十年来，高速、大规模及微型计算机的出现，几乎影响了人类所涉及的一切生活、生产及科学技术领域，创造了许多奇迹。就科技领域而言，许多以计算机为主要工具的边缘学科如计算数学、计算力学、计算物理学、计算化学、计算经济学等如雨后春笋，得到了长足的发展，计算物理学正是在计算数学、物理学、计算机图形处理技术等基础上建立起来的。它的出现使得现今的物理学扩展为理论物理、实验物理及计算物理三个相互关联，相互促进而又具有一定相对独立性的三个部分。

由于高速大规模计算机的应用，使得用传统解析方法难以求解的许多大规模物理方程及一些非线性方程可以得到适当的解答。而且，计算机并不只是一种高级“算盘”，它不仅提高计算速度，更重要的是延伸和扩展了人类的思维和智能，人们有可能按新的构思，在计算机上模拟一些物理过程，从中发现新的现象和规律，使得计算物理的研究上升到一个新的台阶。但是，计算物理的出现毕竟是最近几十年的事情，许多方面很不成熟。目前，国内外已有一些计算物理方面的教材和专著出版，有的是按问题建立体系（如量子力学中的数值计算方法，流体力学中的数值计算方法，分子动力学中的数值计算方法等），有的则是按数学方法建立体系（如有限元法，有限差分法，蒙特卡洛方法等），各有千秋，但多数比较深，是供研究生用的，适合大学生阅读的为数不多。

西北师范大学物理系计算物理研究室的同仁在给本科生开设算法语言和计算方法的基础上，1987年以来又为本科生开设了计算物理选修课，受到了同学们的欢迎。我们是在同学们已掌握高等数学和计算方法的基础上讲授这门课的。其目的不在于全面而系统地介绍计算物理的各个方面，而是想通过一些典型问题的讨论引起同学们的兴趣，为将来进一步学习打下一定基础。因此，它是一门“入门”课程。教材的系统，既不是按计算方法排列，也没有完全按学科部门排列。所涉及的内容都是同学们可以接受，可以在计算机上较短时间内能独立完成的。一些内容（如样条函数拟合、弹簧摆运动的研究等）曾是我室同仁的教学研究成果。

需要说明的是 Fortran 语言是科学计算中常用的算法语言，但在计算量不大的情况下，为了在微机上绘图方便，我们使用了 Basic 语言。

在本教材编写过程中，研究室的董晨钟，段文山，孙建安，薛具奎，张万顺等同志参加了讨论，并得到物理系领导和同学们的鼓励和支持。在一些章节中还吸收了杜宽仪，任小康，李幕唐等同志的研究成果。杨鸿武同志协助文字和图形的打印。在此，我们表示衷心的感谢。

编 者

1993 年 3 月 .

目 录

第一章	计算机图形处理基础	(1)
第二章	样条函数拟合方法在物理实验中的应用	(38)
第三章	弹簧摆的运动	(67)
第四章	行波的数值模拟实验	(79)
第五章	点电荷电场的描绘	(88)
第六章	静电场拉普拉斯方程的有限差分法	(99)
第七章	薛定谔方程的数值解法	(107)
第八章	氢原子电子云的绘制	(121)

第一章 计算机图形处理基础

人们与外界的信息交换有 70% 是以图象图形的形式通过眼睛进行的。图象图形表示的东西比文字的起源更早。它不需要人们花很长时间的學習就能看懂，而且信息量大，重点突出。现代计算机技术的迅速发展，在诸如各种光波的转换（为可见光），参数定量化，工业设计（辅助设计 *CAD*, 辅助制造 *CAM*），图信息管理等实用化技术领域中，人们对图象图形的要求和依赖性大大地增加了。图象图形处理技术已经是开拓人们视界的一个有力武器。计算物理是随着计算机的发展而建立的一门交叉学科。它涉及计算数学和力学、光学、热学、电磁学等基本理论和方法。特别，通过计算机数值模拟和图形显示，使那些既抽象又难懂的数理现象变得直观、形象、浅显易懂。所以，计算机图形处理已经成为现代科技和教育中用来表示计算和实验结果的一种重要手段。

§1. *BASIC* 图形处理语句 及其应用举例

本节对 *BASIC* 语言中有关图形语句作一简介，并给出一些应用程序实例供读者参考。不熟悉 *BASIC* 语言的读者，可参考有关扩展 *BASIC* 使用手册。

§1.1 圆语句— *CIRCLE*

【功能】 在显示屏幕上画圆、椭圆和扇形。

【语法】 *CIRCLE* (x,y), R , CL , α , β , ε

【解释】 其中，(x,y)表示圆心坐标； R 表示圆半径； CL

表示颜色代号; α , β , ε 分别表示开始角度、终止角度(画扇形)和圆的扁平率(画椭圆)。如果只画圆时, α , β , ε 可以省略。

【实例】 *CIRCLE* (80, 80), 50, 4

§1.2 线段语句— *LINE*

【功能】 在显示屏幕上画线段和矩形。

【语法】 *LINE* (x_1, y_1)-(x_2, y_2), *CL*, *B*, *BF*, 形式

【解释】 其中, (x_1, y_1)和 (x_2, y_2)分别是线段的起点和终点; *CL* 表示颜色代号; *B* 表示画以 (x_1, y_1) 和 (x_2, y_2) 为对角的矩形; *BF* 表示把矩形内填涂; “形式”是指用什么线(虚线、点线等)画线段, 一般用十六进制数表示, 如 *&HF99F* 等。其表达形式如下:

&HF99F = 11110011001111

— — — — —

如果只画线段, 可省略 “*B*”、“*BF*”。

【实例】 *LINE* (100, 150)-(150, 200), 4,,, *&HF99F*

§1.3 点语句— *PSET*

【功能】 在显示屏幕上显示一个坐标为 (x, y) 的点。

【语法】 *PSET* (x, y), *CL* ①

PSET STEP (x, y), *CL* ②

【解释】 其中, (x, y)表示点的坐标(式①指绝对坐标, 而式②指相对坐标), *CL* 表示颜色代号。此语句也常用来画线。

【实例】 *PSET* (100, 100), 4

§1.4 图形填涂语句— *PAINT*

【功能】 用指定的着色属性和边界属性给所画图形涂色。

【语法】 *PAINT* (x, y), 着色属性, 边界属性, 背景属

性

【解释】 (x, y) 为着色中心坐标 (图形内)。着色属性指涂色代号, 边界属性指图形边界色代号, 背景属性指屏幕底色 (与初期设定有关) 详情参考有关 *BASIC* 手册。

§1.5 图形信息传送语句— PUT

【功能】 将以数据方式储存的图形传送到显示屏幕指定坐标位置。常和图形数据传送语句 *GET* 配合使用, 使动画和高速物体移动成为可能。

【语法】 *PUT* (x, y) , 数组名, 工作动词

【解释】 (x, y) 是被传送图形所在位置的左上角点坐标。数组名指储存图形数据的数组。工作动词可以是 *PSET*、*PRESET*、*OR*、*AND*、*XOR*, 详情参考有关 *BASIC* 手册。

§1.6 图形信息传送语句— GET

【功能】 把屏幕上显示的图形以数据方式保存到文件数组中。需要显示时, 利用 *PUT* 语句。

【语法】 *GET* $(x_1, y_1)-(x_2, y_2)$, 数组名

【解释】 其中, (x_1, y_1) 和 (x_2, y_2) 是包含图形的最小矩形的对角点坐标。数组用来储存图形数据, 可以是除字符串型外的任何类型但其容量必须大到能装下图形的所有数据。详细计算方法参考有关 *BASIC* 手册。

这里必须说明的是, *BASIC* 图形语句与机型有关。请注意对照移植。为了帮助读者理解掌握以上语句的用法, 给出以下简单的应用程序实例, 以供学习参考。

例 1. 球 (图 1-1, 源程序 1)

球面坐标计算公式:

$$x = R \sin \alpha \cdot \cos \beta$$

$$y = R \sin \alpha \cdot \sin \beta$$

$$z = R \cos \alpha$$

源程序中 $\alpha = TA$, $\beta = TB$, $X = FX$, $Y = FY$, $Z = FZ$.410, 420, 430 行。

坐标旋转公式:

$$xa = \cos(t_x) \cos(t_z)$$

$$ya = \sin(t_x) \sin(t_z) \cos(t_y) - \cos(t_x) \sin(t_y)$$

$$za = \cos(t_x) \sin(t_z) \cos(t_y) + \sin(t_x) \sin(t_y)$$

$$xb = \cos(t_x) \sin(t_z)$$

$$yb = \sin(t_x) \sin(t_z) \sin(t_y) + \cos(t_x) \cos(t_y)$$

$$zb = \cos(t_x) \sin(t_z) \sin(t_y) - \sin(t_x) \cos(t_y)$$

$$xc = -\sin(t_x)$$

$$yc = \sin(t_x) \cos(t_y)$$

$$zc = \cos(t_x) \cos(t_y)$$

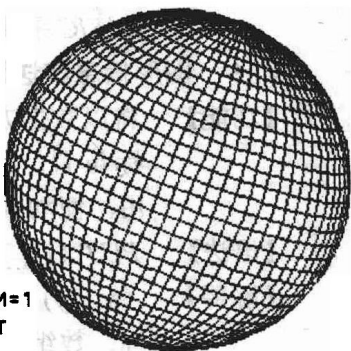


图 1-1 球

t_x , t_y , t_z 是 x , y , z 轴的旋转角度。源程序中 310 ~ 390 行。

图形坐标:

$$x_2 = xa \cdot FX + ya \cdot FY + xa \cdot FZ + x_0$$

$$y_2 = -(xb \cdot FX + yb \cdot FY + zb \cdot FZ) + y_0$$

源程序中 470, 480 行。

例 2. 有趣的三角函数 (图 1-2, 源程序 2. 图 1-3、源程序 3)。改变源程序中的各种参数, 可得到各种有趣的图形。

10 PEM --- 源程序 1 - - - - -

20 CLS

30 SCREEN 2

40 PI=3.14159

50 T=PI/180

60 TX=-80*T : TY=30*T : TZ=-.3

```

70  NN=36
80  DT=PT/NN
90  DIST=-1
100 X0=340 : Y0=160
110 R=100
120 '
130 '
140 GOSUB 300
150 FOR TB=0 TO PI STEP
    DT : M=1
160 FOR TA=0 TO 2.1*PI STEP DT
170 GOSUB 410
180 NEXT TA , TB
190 FOR TA=0 TO PI STEP DT : M=1
200 FOR TB=0 TO 2.1*PI STEP DT
210 GOSUB 410
220 NEXT TB , TA
230 END
300 '
310 XA=COS(TY)*COS(TZ)
320 YA=SIN(TX)*SIN(TY)*COS(TZ)-COS(TX)*SIN(TZ)
330 ZA=COS(TX)*SIN(TY)*COS(TZ)+SIN(TX)*SIN(TZ)
340 XB=COS(TY)*SIN(TZ)
350 YB=SIN(TX)*SIN(TY)*SIN(TZ)+COS(TX)*COS(TZ)
360 ZB=COS(TX)*SIN(TY)*SIN(TZ)-SIN(TX)*COS(TZ)
370 XC=-SIN(TY)
380 YC=SIN(TX)*COS(TY)
390 ZC=COS(TX)*COS(TY)

```

```

400 RETURN
410 FX=R*SIN(TA)*COS(TB)
420 FY=R*SIN(TA)*SIN(TB)
430 FZ=R*COS(TA)
440 F=FX*XC+FY*YC+FZ*YZ
450 IF F=<0 THEN M=1 : GOTO 520
470 X2=XA*FX+YA*FY+ZA*FZ +X0
480 Y2=(XB*FX+YB*FY+ZB*FZ)*DIST+Y0
490 IF M=1 THEN : M=2 : GOTO 510
500 LINE (X1 , Y1)-(X2 , Y2) , 6
510 X1=X2 : Y1=Y2
520 RETURN

```

```

100 REM -- 源程序 2--
110 WIDTH 80 , 25 : CONSOLE 0 , 25 , 0
120 PI=3.14159 : D=60
130 FOR A=0 TO 2*PI STEP 2*PI/720
140 E=D*(1+SIN(4*A))
150 X1=320+E*COS(A)
160 X2=320+E*COS(A+PI/5)
170 Y1=200+E*SIN(A)
180 Y2=200+E*SIN(A+PI/5)
190 LINE (X1 , Y1)-(X2 , Y2)
200 NEXT A

```

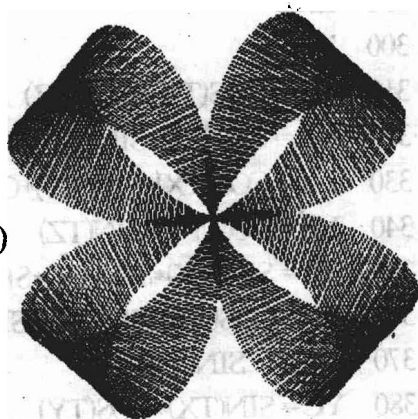


图 1-2 花结

```

100 REM-- 源程序 3-- 波峰和谷-----
110 WIDTH 80 , 25 : CONSOLE 0 , 25 , 0
120 PI=3.14159 : D=60
130 FOR PY=60 TO 300 STEP 3
140 AL=2*PI*(PY-20)/360
150 L=PI*SIN(AL)
160 FOR A=0 TO 6*PI STEP PI/8
170 X=440/((6*PI)*A+100
180 Y=18*SIN(A+L)+PY
185 IF A=0 THEN POINT(X , Y)
190 LINE -(X , Y)
195 NEXT A
200 NEXT PY
210 END

```

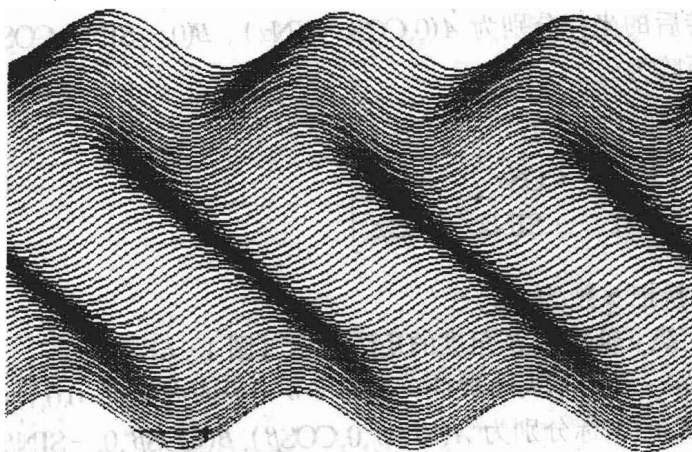


图 1-3 波峰和波谷

参考文献

GW-BASIC User's Reference 470 , 00561 Microsoft

§2. 曲面的计算和立体图形的二维显示方法

三维形体的二维显示技术，是计算机图形处理中的基础。在科学计算中常用等高线法 (§3) 和投影法。投影就是把空间图形投射到投影面上而得到的平面图形。本节主要介绍透视投影方法。这里，涉及数学中坐标变换的有关概念。

§2.1 空间坐标变换

任何空间坐标变换都可以由平行移动和若干二维旋转变换复合而成。以下，设在空间右手坐标系中，分别以三个坐标轴为旋转轴，相应的坐标平面均为右手螺旋方向，即沿轴向原点看去是逆时针方向 (图 1-4)。

1. 绕 x 轴旋转 α 角度的矩阵表示 (图 1-5)

固定 x 轴。不失一般性，在 $y o z$ 平面内的点 $(0,1,0)$ 和 $(0,0,1)$ 经旋转后的坐标分别为 $A(0, \cos\alpha, \sin\alpha)$, $B(0, -\sin\alpha, \cos\alpha)$ 。其旋转矩阵为：

$$T_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \quad (2-1)$$

其逆变换把 α 换为 $-\alpha$ 。

2. 绕 y 轴旋转 β 角度的矩阵表示 (图 1-6)

固定 y 轴。不失一般性，设在 $z o x$ 平面内的点 $(0,0,1)$, $(1,0,0)$ 经旋转后的坐标分别为 $A(\sin\beta, 0, \cos\beta)$, $B(\cos\beta, 0, -\sin\beta)$ 。其旋转矩阵为：

$$T_y = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \quad (2-2)$$

其逆变换把 β 换为 $-\beta$.

3. 绕 z 轴旋转 γ 角度的矩阵表示 (图 1-7)

固定 z 轴。不失一般性, 设在 $x o y$ 平面内的点 $(1,0,0), (0,1,0)$, 经旋转后的坐标分别为 $A(\cos\gamma, \sin\gamma, 0), B(-\sin\gamma, \cos\gamma, 0)$. 其旋转矩阵为:

$$T_z = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-3)$$

其逆变换把 γ 换为 $-\gamma$.

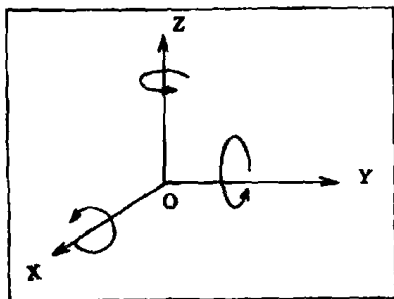


图 1-4 旋转方向

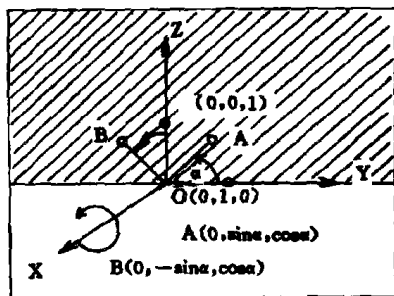


图 1-5 绕 x 轴旋转

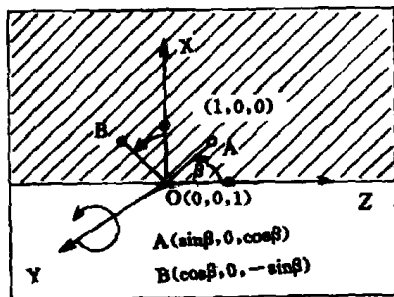


图 1-6 绕 y 轴旋转

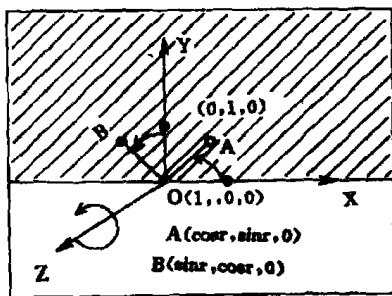


图 1-7 绕 z 轴旋转

4. 顺次绕 x 、 y 、 z 轴旋转 α 、 β 、 γ 角度的连续变换

以上旋转变换的连续复合变换应是 T_x 、 T_y 、 T_z 的乘积, 即

$$T = T_x \cdot T_y \cdot T_z = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \quad (2-4)$$

其中

$$\left\{ \begin{array}{l} T_{11} = \cos\alpha \cdot \cos\gamma \\ T_{21} = \sin\alpha \cdot \sin\beta \cdot \cos\gamma - \cos\alpha \cdot \sin\gamma \\ T_{31} = \cos\alpha \cdot \sin\beta \cdot \cos\gamma + \sin\alpha \cdot \sin\gamma \\ T_{12} = \cos\beta \cdot \sin\gamma \\ T_{22} = \sin\alpha \cdot \sin\beta \cdot \sin\gamma + \cos\alpha \cdot \cos\gamma \\ T_{32} = \cos\alpha \cdot \sin\beta \cdot \sin\gamma - \sin\alpha \cdot \cos\gamma \\ T_{13} = -\sin\beta \\ T_{23} = \sin\alpha \cdot \cos\beta \\ T_{33} = \cos\alpha \cdot \cos\beta \end{array} \right. \quad (2-5)$$

那么, 对于空间任一点 $p(x, y, z)$, 经变换 T 后的新坐标 XX , YY, ZZ 是

$$[XX, YY, ZZ] = [x, y, z] \cdot T$$

即

$$\begin{aligned} XX &= X \cdot T_{11} + Y \cdot T_{21} + Z \cdot T_{31} \\ YY &= X \cdot T_{12} + Y \cdot T_{22} + Z \cdot T_{32} \\ ZZ &= X \cdot T_{13} + Y \cdot T_{23} + Z \cdot T_{33} \end{aligned} \quad (2-6)$$

§2.2 中心透视投影基本概念

所谓中心透视投影就是将投影面 (屏幕画面) 置于投影中心 (视点) 和观察物体之间 (图 1-8)。利用图形对有关术语说明如下:

- ① 视点 p 叫投影中心，一般指眼睛的位置。
- ② 投影面：指屏幕显示画面。
- ③ 透视点：设 Q 为观察物体上一点，则连线 PQ 与投影面的交点 S 称透视点。
- ④ 用户坐标系：指被观察物体所在的坐标系 (x, y, z) 图 1-8 ①。
- ⑤ 观察坐标系：也称视点坐标系。坐标原点为视点 p ，且为左手系。一般可经用户坐标系的平移和旋转而建立。图 1-8 ③。
- ⑥ 屏幕坐标系：指显示屏幕上的二维坐标系，常取屏幕中心 O_c 为原点。图 1-8 ②。

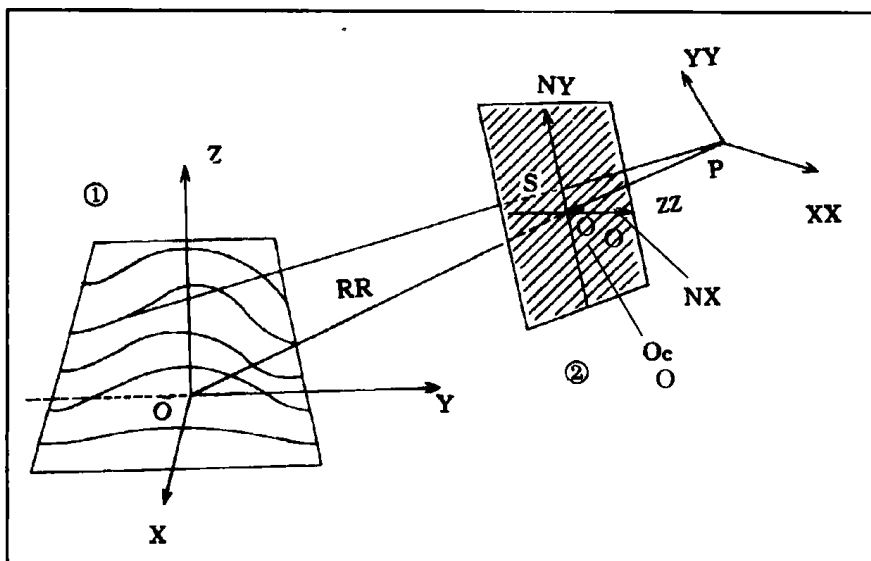


图 1-8 透视投影示意

§2.3 透视投影过程和图形坐标计算

当视点在空间任意位置时，用一个坐标系来描述投影过程是相当复杂的。如果引入观察坐标系作为过渡，就可以使问题简