



TI-86 图象计算器 参考手册

版权 © 1997 Texas Instruments Incorporated

IBM 是 International Business Machines Corporation 的注册商标

Texas Instruments Incorporated 的™ 商标

Macintosh 是 Apple Computer Inc. 的注册商标



重要信息

Texas Instruments 不对任何程序或书面资料作明确或默许的担保，包括（但不限于）对于特殊目的的销售和适用的默许担保。这些资料不含任何担保条款。

在任何情况下，**Texas Instruments** 都不对因购买或使用这些资料所造成的任何特殊的、间接的、意外的、连带的损失承担责任。不管 **Texas Instruments** 采取何种赔偿措施，它所负的唯一责任不应超过本产品的购买价格。此外，**Texas Instruments** 不对其他各方因使用这些资料而提出的任何形式的投诉承担责任。

关于射频干扰的 US FCC 信息

本设备已经过测试并证实符合依照 FCC 规则第 15 节的规定对 B 类数字设备的要求。制定这些要求是为了尽可能避免数字设备在住宅安装时产生有害的干扰。本设备能产生、使用并辐射射频能量。如果不遵照说明书安装和使用，可能对无线电通讯造成有害的干扰。但是，这不能保证在特殊的安装中也不会产生干扰。

本设备是否干扰广播或电视的接收，可通过关闭和打开该设备来确定。可尝试以下一项或多项措施来消除干扰：

- ◆ 调整接收天线的方向或位置。
- ◆ 使本设备远离接收设备。
- ◆ 将本设备与接收设备的供电回路分开，即不要将两者连接到同一电路的插座中。
- ◆ 咨询销售商或有经验的无线电/电视技术人员以寻求帮助。

目录表

TI-86 快速入门	1	在图象屏幕上绘制函数	9
准备使用新的 TI-86	2	在方程编辑器中显示并输入函数	9
安装 AAA 电池	2	改变函数的图形样式	10
开启和关闭 TI-86	2	在图形屏幕上绘制函数	11
调节对比度	2	跟踪函数	11
复位所有内存和缺省设置	3	计算给定 x 值时的 y 值（跟踪期间）	12
在主屏幕上计算	3	改变窗口变量的值	12
计算一个数的正弦值	3	取消选定函数	13
将上次答案存储到变量	3	放大图形屏幕一部分	14
表达式中使用变量	4		
编辑表达式	4	第 1 章：运行 TI-86	15
显示复数结果	5	安装或更换电池	16
在函数中使用数组	5	更换电池时间	16
显示数组中实数的整数部分	6	开启和关闭 TI-86	17
移走（退出）菜单	6	调节显示对比度	17
求平方根	7	主屏幕	18
计算导数	7	显示输入和结果	18
获得、编辑并重新计算前一个输入	8	输入数字	19
将华氏温度换算为摄氏温度	8	输入负数	19
将未计算的表达式存储到方程变量	9	使用科学或工程记数法	20
		输入复数	20

输入其他字符	21	在函数前使用 Ans	30
第二功能键	21	保存结果到变量	30
ALPHA 键	21	使用 TI-86 菜单	31
大写字母锁和小写字母锁	22	显示菜单	31
普通光标	22	菜单键	32
光标方向键	23	选择菜单项	32
插入、删除和清除字符	23	退出（删除）菜单	33
输入表达式和指令	24	查看并改变方式	34
输入表达式	24	改变方式设置	34
在表达式中使用函数	25	第 2 章： CATALOG、变量和字符	37
使用指令	25	CATALOG	38
输入函数、指令和操作符	25	存储数据到变量	39
输入连续输入项	26	创建变量名	39
忙指示器	26	存储数值到变量名	40
中断计算或图象	26	存储未计算表达式	40
诊断错误	27	存储结果	41
改正错误	27	复制变量值	41
重用以前输入及上次结果	28	显示变量值	41
取出上次输入	28	调用变量值	42
取出并编辑上次输入	28	变量按数据类型分类	42
取出以前输入	28	CATLG-VARS（CATALOG — 变量）菜单	43
取出多个输入	29	选择变量名	44
清除 ENTRY 存储区域	29	CUSTOM 菜单	44
取出上次结果	29	输入 CUSTOM 菜单项	44

清除 CUSTOM 菜单项	45
从内存中删除变量	45
CHAR (字符) 菜单	45
CHAR MISC (其他) 菜单	46
CHAR GREEK 菜单	46
CHAR INTL (国际) 菜单	46
添加修饰符到元音	46

第 3 章: 数学、微积分和测试操作 **47**

键盘数学函数	48
MATH 菜单	49
MATH NUM (数值) 菜单	49
MATH PROB (概率) 菜单	50
MATH ANGLE 菜单	51
MATH HYP (双曲) 菜单	51
MATH MISC (其他) 菜单	52
内插/外推编辑器	53
CALC (微积分) 菜单	54
TEST (关系) 菜单	55
在表达式和指令中使用测试	56

第 4 章: 常量、换算、数制和复数 **57**

使用内置和用户创建常量	58
CONS (常量) 菜单	58
CONS BLTIN (内置常量) 菜单	58

创建或重新定义用户创建常量	60
CONS EDIT (常量编辑器) 菜单	60
在表达式中输入常量名	61
换算度量单位	61
换算度量单位	61
CONV (换算) 菜单	62
CONV LENGTH (长度) 菜单	63
CONV AREA (面积) 菜单	63
CONV VOL (体积) 菜单	63
CONV TIME (时间) 菜单	63
CONV TEMP (温度) 菜单	63
CONV MASS (质量) 菜单	64
CONV FORCE (力) 菜单	64
CONV PRESS (压强) 菜单	64
CONV ENRGY (能量) 菜单	64
CONV POWER (功率) 菜单	64
CONV SPEED (速度) 菜单	64
换算以比率表示的值	65
数制	65
数制范围	66
反码和补码	66
(数值) BASE 菜单	66
BASE A-F (十六进制字符) 菜单	67
输入十六进制数字	67

BASE TYPE (类型) 菜单	67
BASE CONV (换算) 菜单	68
换算数制	68
BASE BOOL (布尔) 菜单	68
布尔运算结果	69
BASE BIT (位) 菜单	69
使用复数	70
复数结果	70
表达式中使用复数	71
CPLX (复数) 菜单	71

第 5 章: 函数图象 **73**

定义图象	74
设置图象方式	74
GRAPH 菜单	75
使用方程编辑器	76
方程编辑器 (GRAPH $y(x)=$) 菜单	76
在方程编辑器中定义函数	77
关于定义函数方程的说明	78
选择图象样式	79
在方程编辑器中设置图象样式	80
使用阴影图案区分函数	80
查看和改变统计图的开/关状态	81
设置窗口变量	81
显示窗口编辑器	82

改变窗口变量值	82
用 Δx 和 Δy 设置图象精度	83
设置图象格式	83
显示图象	85
暂停或停止绘制中的图象	85
修改已绘制的图象	85
绘制曲线族	86
记忆绘图	86

第 6 章: 图象工具 **87**

TI-86 图象工具	88
GRAPH 菜单	88
使用自由移动光标	89
图象精度	89
跟踪图象	90
停止和重新开始跟踪	91
用 ZOOM 操作调整图象屏幕的尺寸	91
GRAPH ZOOM 菜单	91
定义定制放大	93
设置缩放因子	93
放大和缩小图象	93
存储和重新调用缩放窗口变量值	95
使用交互数学函数	95
GRAPH MATH 菜单	95
影响 GRAPH MATH 操作的设置	96

使用 ROOT、FMIN、FMAX 或 INFLC.....	97
使用 Jf(x)、DIST 或 ARC.....	98
使用 dy/dx 或 TANLN.....	99
使用 ISECT.....	100
使用 YICPT.....	100
对指定 x 值计算函数值.....	101
在图象上绘制.....	101
在图象上绘制之前.....	102
保存和重新调用已绘制的图形.....	102
清除已绘制的图形.....	103
GRAPH DRAW 菜单.....	103
阴影图象区域.....	104
绘制线段.....	105
绘制垂直线或水平线.....	106
绘制圆.....	106
绘制函数、正切线或反函数.....	107
徒手绘制点、线和曲线.....	107
在图象上放置文本.....	108
打开或关闭点.....	108
第 7 章：运算表	109
显示运算表.....	110
TABLE（运算表）菜单.....	110
运算表.....	110
运算表的自变量和因变量.....	111

浏览运算表.....	111
运算表菜单.....	112
设置运算表.....	113
查看和编辑因变量方程.....	114
清除运算表.....	114

第 8 章：极坐标图象 **115**

预览：极坐标图象.....	116
定义极坐标图象.....	117
设置极坐标图象方式.....	117
GRAPH（图象）菜单.....	117
显示极坐标方程编辑器.....	118
设置图象屏幕窗口变量.....	118
设置图象格式.....	119
显示图象.....	119
在 Pol（极坐标）图象方式下使用图象工具.....	119
自由移动光标.....	119
跟踪极坐标方程.....	120
将跟踪光标移到 θ 值.....	121
使用缩放操作.....	121
GRAPH MATH（图象运算）菜单.....	122
对给定的 θ 求解方程.....	122
在极坐标图象上绘图.....	122

第 9 章: 参数图象 123

预览: 参数图象	124
定义参数图象	125
设置参数图象方式	126
GRAPH (图象) 菜单	126
显示参数方程编辑器	126
选择和取消选定参数方程	127
删除参数方程	127
设置图象屏幕窗口变量	127
设置图象格式	128
显示图象	128
在参数图象方式下使用图象工具	128
自由移动光标	128
跟踪参数函数	128
将跟踪光标移到 t 值处	129
使用缩放操作	129
GRAPH MATH (图象运算) 菜单	130
对给定 t 求解方程	130
在参数图象上绘图	130

第 10 章: 微分方程图象 131

定义微分方程图象	132
设置微分方程图象方式	132
GRAPH 菜单	133

设置图象格式	133
显示微分方程编辑器	134
设置图象屏幕窗口变量	135
设置初始条件	136
设置坐标轴	137
微分方程绘制提示	137
内置变量 fldPic	138
显示图象	138
输入并求解微分方程	139
在 SlpFld 格式下绘制	139
将方程转换为一阶方程组	140
在 DirFld 格式下绘制	141
在 FldOff 格式下绘制方程组	142
对指定值求解微分方程	144
在 DifEq (微分方程) 图象方式下使用图象工具	144
自由移动光标	144
跟踪微分方程	144
将跟踪光标移到 t 值处	145
在微分方程的图象上绘制	145
绘制方程图象并将解保存到数组中	145
使用缩放操作	147
用 EXPLR 迭代绘制解	148
对指定的 t 计算微分方程	150

第 11 章: 数组 151

TI-86 中的数组	152
LIST 菜单	152
LIST NAMES 菜单	153
创建、存储和显示数组	153
在表达式中直接输入数组	153
通过存储数组创建数组名	154
显示存储到数组名中的数组元素	154
显示或使用单个数组元素	155
将新值存储到数组元素	155
复数数组元素	156
数组编辑器	156
List Editor (数组编辑器) 菜单	156
在未命名列中创建数组名	157
在数组编辑器中插入数组名	157
显示和编辑数组元素	158
从数组中删除元素	158
从数组编辑器中删除数组	158
使用数组运算	159
LIST OPS (数组运算) 菜单	159
对数组进行数学运算	161
连接公式到数组名	162
比较普通数组和相连数组	163
用数组编辑器连接公式	163

使用数组编辑器处理与公式相连接的数组	164
执行和显示所连接公式	164
处理与连接公式相关的错误	165
公式与数组名分离	166
编辑与公式相连数组的元素	166

第 12 章: 向量 167

TI-86 的向量	168
创建、存储和显示向量	169
VECTR (向量) 菜单	169
VECTR NAMES (向量名) 菜单	169
在向量编辑器中创建向量	169
Vector Editor (向量编辑器) 菜单	170
在主屏幕中创建向量	170
创建复数向量	171
显示向量	171
在表达式中使用向量	172
编辑向量维数和元素	172
VECTR MATH (向量数学运算) 菜单	173
VECTR OPS (向量运算) 菜单	173
VECTR CPLX (复数向量) 菜单	175
向量的数学运算	176

第 13 章: 矩阵 177

TI-86 的矩阵	178
-----------------	-----

创建、存储和显示矩阵	178
MATRX (矩阵) 菜单	178
MATRX NAMES (矩阵名) 菜单	178
在矩阵编辑器中创建矩阵	178
Matrix Editor (矩阵编辑器) 菜单	179
在主屏幕中创建矩阵	180
创建复数矩阵	180
显示矩阵元素、行和子矩阵	181
在表达式中使用矩阵	181
在矩阵编辑器中编辑矩阵	182
在主屏幕编辑矩阵	182
MATRX MATH (矩阵数学运算) 菜单	183
MATRX OPS (矩阵运算) 菜单	184
MATRX CPLX (复数矩阵) 菜单	185
矩阵的数学运算	185

第 14 章: 统计 **187**

TI-86 的统计分析	188
建立统计分析	188
STAT (统计) 菜单	188
输入统计数据	189
LIST NAMES (数组名) 菜单	189
STAT CALC (统计计算) 菜单	189
自动回归方程存储	191
统计分析的结果	192

STAT VARS (统计变量) 菜单	192
绘制统计数据	194
STAT PLOT 状态屏幕	194
STAT PLOT (统计图) 菜单	195
建立统计图	195
打开和关闭统计图	195
PLOT TYPE (统计图类型) 菜单 (选择统计图类型) ...	196
统计图类型特征	196
STAT DRAW (统计图) 菜单	199
预测统计数据值	199

第 15 章: 方程求解 **201**

预览: 方程求解器	202
在方程输入编辑器中输入方程	203
建立交互式求解器编辑器	204
输入变量值	204
用上下限和估计值控制求解	204
编辑方程	205
SOLVER (方程求解器) 菜单	206
求解未知变量	206
绘制结果	207
求解器图象工具	207
求解器 ZOOM (缩放) 菜单	208
联立方程求解器	208
输入方程来同步求解	208

将方程系数和 结果存储到变量	210
多项式求根器	211
输入并求解多项式	211
将多项式的系数或根存储到变量	212

第 16 章：程序设计 **213**

在 TI-86 上编写程序	214
PRGM（程序）菜单	214
在程序编辑器中创建程序	214
程序编辑器菜单	215
PRGM I/O（输入/输出）菜单	215
TI-86 键码图	217
PRGM CTL（程序控制）菜单	218
输入命令行	220
程序编辑器中的菜单和屏幕	220
运行程序	221
中断程序	222
使用程序	223
管理内存和删除程序	223
编辑程序	223
一个程序调用另一个程序	224
拷贝一个程序到另一个程序	225
在单个程序中使用或删除变量	225
运行汇编语言程序	225
输入并存储字符串	226

STRNG（字符串）菜单	227
创建字符串	227

第 17 章：内存管理 **229**

检查可用内存	230
MEM（内存）菜单	230
检查内存使用情况	230
从内存中删除项目	231
MEM DELET（删除）菜单	231
复位 TI-86	232
MEM RESET（复位）菜单	232
ClrEnt（清除输入）	232

第 18 章：TI-86 通讯链接 **233**

TI-86 链接选项	234
链接两台 TI-86	234
链接 TI-86 和 TI-85	234
链接 TI-86 和 CBL 或 CBR 系统	234
链接 TI-86 和 PC 或 Macintosh	235
从 Internet 上下载程序	235
连接 TI-86 到其他设备	235
LINK（连接）菜单	236
选择要发送的数据	236
LINK SEND 菜单	236
初始化内存备份	237

选择要发送的变量	238
SEND WIND（窗口变量）屏幕	238
发送变量到 TI-85	239
LINK SND85（发送数据到 TI-85）菜单	239
准备接收设备	240
传送数据	240
接收传送的数据	241
重复向几个设备传送数据	242
错误情形	242
接收单元内存不足	242

第 19 章：应用 243

矩阵的数学运算.....	244
求曲线间面积.....	245
微积分基本原理.....	246
电路.....	248
程序: 泰勒级数.....	250
特征多项式和特征值.....	252
幂级数的收敛性.....	254
蓄水池问题.....	256
食肉动物 — 猎物模型.....	258
程序: Sierpinski 三角形.....	260

第 20 章: A 到 Z 函数和指令参考 261

快速查找定位器.....262

运算符按字母顺序排列 266

附录 379

TI-86 菜单图	380
疑难解答.....	392
错误情形.....	393
Equation Operating System (EOS™)	397
蕴涵乘法	397
圆括号	397
TOL（误差编辑器）	398
计算精度.....	399
有关 TI 产品、服务和保障信息	400
TI 产品和服务信息	400
服务和保障信息	400

索引



TEXAS INSTRUMENTS

TI-86

M1

F1

M2

F2

M3

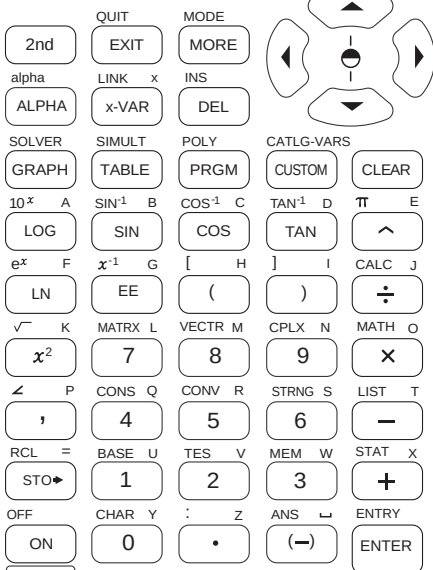
F3

M4

F4

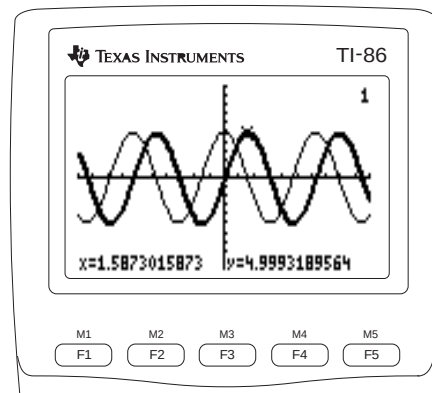
M5

F5



TI-86 快速入门

准备使用您的新 TI-86	2
在主屏幕上计算	3
在图象屏幕上绘制函数	9



准备使用您的新 TI-86

TI-86 快速入门中的主要例子展示了一些 TI-86 的常用功能。开始之前，必须先安装电池、打开计算器、调整对比度并复位内存和缺省设置。第一章对这些主题有更详细的描述。

安装 AAA 电池

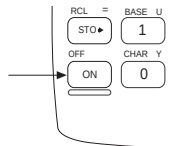
在 TI-86 的零售包里有四节 AAA 电池。取出包里的电池并将它们安装在计算器背后的电池盒里。根据电池盒里的极性（+ 和 -）图示来排放电池。

打开和关闭 TI-86

经过大约四分钟的休止期，
TI-86 自动关闭。

要打开 TI-86，按下位于键盘左下角的 **[ON]** 键。会在屏幕的左上角看到输入光标 (■) 在闪烁。如果没看到光标，请调整对比度（请参阅下面部分）。

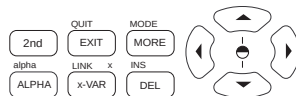
要关闭计算器，按下 **[2nd]**，然后按下 **OFF** 下面的键 **[ON]**。本指南用括号（[和]）来表示 **[2nd]** 和 **[ALPHA]** 组合击键。例如，要关闭 TI-86，按下 **[2nd] [OFF]**。



调整对比度

如果你在调整对比度时松开了 **[▲]** 或 **[▼]**，就必须再次按下 **[2nd]** 来继续调整。

- ❶ 按下并释放黄色的 **[2nd]** 键。
- ❷ 按住 **[▲]** 或 **[▼]**（在半阴影圆的上面或下面）。
 - ◆ 要变暗屏幕对比度，按住 **[▲]**。
 - ◆ 要加亮屏幕对比度，按住 **[▼]**。



复位所有的内存和缺省设置

要复位所有的内存和缺省设置，按下 **2nd** **[MEM]** **F3** **F1** **F4**。在主屏幕上将显示 **Mem cleared** 和 **Defaults set** 消息，表明确认已复位了所有的内存和缺省设置。在复位内存和缺省设置之后，可能需要调整对比度。

在主屏幕上计算

为了表示 **2nd** 和 **[ALPHA]** 的组合击键，本指南在所需按键上面的字周围放置了括号（**[和]**）。

TI-86 屏幕上的除号是一个正斜杠（/），就如分数中的一样。

计算之后，输入光标自动移到下一行，等待新的输入。

当 TI-86 对表达式求值时，它自动将答案存储到内置变量 **Ans** 中，同时替换掉以前的值。

计算一个数的正弦值

- 1 输入正弦函数。

([CLEAR]) **[SIN]**

| sin |

- 2 输入一个值。也可以输入表达式，按下 **[ENTER]** 后对它进行计算。

[(] **2nd** **[π]** **[÷]** **4** **[)]**

| sin (π/4) |

- 3 对问题求值。正弦表达式 ($\pi/4$) 的求值如图示。

[ENTER]

| sin (π/4)
| .707106781187

将上次答案存储到变量

- 1 将存储符号（**→**）粘贴到屏幕上。因为在之前必须有一个值，而又没有输入值，那么 TI-86 自动在 **→** 之前粘贴 **Ans**。（续）

([CLEAR])

[STO→]

| Ans→ |

当字母锁 (ALPHA) 起作用时按下某个键，则印在键上面的蓝色字母粘贴到屏幕上。在示例中，按下 **2** 来输入 **V**。

- ❶ 输入用来保存上次答案的变量名。确定字母锁打开。
- ❷ 将上次答案存储到变量中。存储的值显示在下一行。

[V]

Ans→V

[ENTER]

Ans→V
.787106781187

表达式中使用变量

- ❶ 输入变量，然后求它的平方。
- ❷ 求值。这将对存储在变量 **V** 中的值求平方，并显示结果。

[CLEAR]

[ALPHA] [V] [x²]V²

[ENTER]

V²
.5

编辑表达式

- ❶ 输入表达式 **(25+14)(4-3.2)**。
- ❷ 将 **3.2** 变为 **2.3**。
- ❸ 将光标移到表达式的开始处并插入一个值。插入光标在 **3** 和 **25** 之间闪烁。
- ❹ 求值。显示结果。

[CLEAR]

[2] [5] [+]

[4] [-] [3] [.] [2]

[←] [←] [←] [←] [2] [.] [3]

[2nd] [←] [2nd] [INS] [3]

[ENTER]

(25+14)(4-3.2)

(25+14)(4-2.3

3_25+14)(4-2.3)

3(25+14)(4-2.3)
198.9

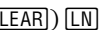
你不必为了对表达式求值而将光标移到行尾。

-  对数值取反，如 -2 。
-  表示减法，如 $5-2=3$ 。

省略号 (...) 表示显示结果超出了屏幕。

显示复数结果

- 1 输入自然对数函数。
- 2 输入一个负数。
- 3 求值。结果显示为一个复数。

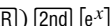
  2 

 (按下  以显示更多信息)

```
ln
ln (-2)
ln (-2)
{.69314718056, 3.1415...
```

在函数中使用数组

- 1 输入指数函数。
- 2 显示 LIST 菜单，然后从 LIST 菜单选择左括号 ({)。
- 在 TI-86 上，{ 表示一个数组的开始。
- 3 输入数组元素。用逗号分隔相邻的元素。
- 4 从 LIST 菜单选择右括号 (})，表明数组结束。
- 5 求值。常量 e 的 5 次方、10 次方和 15 次方作为数组元素显示出来。



LIST 菜单

5  10  15



 (按下  以显示更多信息)

```
e^
e^{
< > NAMES EDIT OPS
e^{5,10,15
e^{5,10,15}
{148.413159103 22026...
```

显示数组中实数的整数部分

- 1 显示 MATH 菜单。（MATH 菜单自动替换上次操作的 LIST 菜单。）
- 2 选择 **NUM** 显示 MATH NUM 菜单。MATH 菜单移到上面。
- 3 从 MATH NUM 菜单中选择 **iPart**（整数部分）函数。**iPart** 粘贴到屏幕。（以前的输入留在屏幕上，以显示 **iPart** 对以前答案的效果。）
- 4 将 **Ans** 粘贴到光标位置。（上次操作的结果数组存储在 **Ans** 中。）
- 5 显示前一个操作的结果数组元素的整数部分。

[2nd] [MATH]

MATH 菜单 →

[F1]

MATH NUM 菜单 →

[F2]

[2nd] [ANS]

[ENTER]



移走（退出）菜单

- 1 在前一个例子中，显示了 MATH 菜单和 MATH NUM 菜单 ([2nd] [MATH] [F1])。
- 2 从屏幕移走 MATH NUM 菜单。
- 3 从屏幕移走 MATH 菜单。

[EXIT]

[EXIT]



求平方根

- 1 将平方根函数粘贴到屏幕。
- 2 输入要对之求平方根的数值。
- 3 对表达式求值。显示 **144** 的平方根。

(CLEAR) 2nd [√]

√

144

√144

ENTER

√144 12

计算导数

- 1 显示 CALC 菜单，然后选择 **der1**。
- 2 输入表达式 (x^2)，其中变量 (**x**) 值为 (**8**)。
- 3 求值。显示 x^2 在 **x** 为 **8** 时的一阶导数。

(CLEAR)

2nd [CALC]

F3

CALC 菜单 →

der1

avgf nDer der1 der2 fnInt

x-VAR x^2 , x-VAR

, 8)

der1(x^2 ,x,8)

ENTER

der1(x^2 ,x,8) 16

当按下 **[ENTER]** 时, TI-86 将输入的指令或表达式存储到被称为 ENTRY 的内置内存存储区。

获得、编辑并重新计算前一个输入

- 1 取得前一个例子的上次输入。(上次输入没有被清除。)
- 2 编辑取得的输入。
- 3 求值。显示 x^2 在 x 为 3 时的一阶导数。

[2nd] **[ENTRY]**

[◀] **[◀]** **3**

[ENTER]

```
der1(x^2,x,8)      16
der1(x^2,x,8)■
der1(x^2,x,8)      16
der1(x^2,x,3■
der1(x^2,x,8)      16
der1(x^2,x,3)      6
■
```

将华氏温度换算为摄氏温度

- 1 显示 CONV 菜单。
- 2 显示 CONV TEMP 菜单。CONV 菜单上移且 TEMP 被亮显。
- 3 输入已知的度量值。如果该度量值是负的, 请使用括号。在本例中, 如果省略了括号, TI-86 将 4°F 换算为约 -15.5°C, 然后对它取反(改变符号), 返回正的 15.5°C。
- 4 选择 °F 来将华氏温度定为已知的度量单位。°F 和换算符号 (►) 被显示在度量值后面。

[CLEAR] **[2nd]** **[CONV]**

[F5]

[⌈] **[(-)]** **4** **[⌋]**

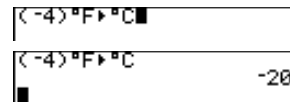
[F2]

```
LNGLTH AREA VOL TIME TEMP►
LNGLTH AREA VOL TIME TEMP
 °C °F °K °R
(-4)■
LNGLTH AREA VOL TIME TEMP
 °C °F °K °R
(-4)°F►■
LNGLTH AREA VOL TIME TEMP
 °C °F °K °R
```

当计算换算的度量时, 不必手工输入单位符号。例如, 不必输入 ° 来表示度。

(续)

- ⑤ 选择 °C 来将摄氏温度换算成的所需单位。 F1
- ⑥ 进行换算。显示 -4°F 的等效 °C 值。 ENTER

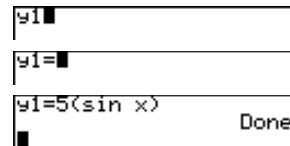


TI-86 计算器屏幕显示：第一行显示 $(-4)^{\circ}\text{F} \rightarrow ^{\circ}\text{C}$ ，第二行显示 $(-4)^{\circ}\text{F} \rightarrow ^{\circ}\text{C}$ 和结果 -20 。

将未计算的表达式存储到方程变量

当用 = 来存储到方程变量时，要先输入方程变量，然后是 =，随后再输入未计算的表达式。这与存储到 TI-86 上大多数变量的顺序相反。

- ① 输入内置的方程变量 **y1**。 (CLEAR) 2nd [alpha] [Y] 1
- ② 输入等号 (=)。 ALPHA [=]
- ③ 输入关于 **x** 的表达式。 5 [] SIN [x-VAR] []
- ④ 存储该表达式。 ENTER



TI-86 计算器屏幕显示：第一行显示 $y1$ ，第二行显示 $y1=$ ，第三行显示 $y1=5(\sin x)$ 和 Done。

下一节显示如何图形化函数 $y1=5(\sin x)$ 和 $y2=5(\cos x)$ 。

在图象屏幕上绘制函数

TI-86 在图象屏幕上绘制四类函数。要绘制图形，必须将一个未计算表达式存储到一个内置的方程变量。

本节的每个操作都是以前一操作为基础。必须从这里开始，按出现的顺序执行操作。本节的第一个操作假定为是前一节最后一个操作的延续。

在方程编辑器中显示并输入函数

- ① 显示 GRAPH 菜单。 (续) GRAPH



TI-86 计算器屏幕显示：GRAPH 菜单，选项包括 $y(x)=$, F1ND, 2NDM, TRACE, GRAPH。

在方程编辑器中，必须以自变量 x 的形式来表示每个方程（只在第五章的 **Func** 图形模式下）。

- ② 从 **GRAPH** 菜单中选择 **y(x)=** 来显示方程编辑器。5(sin x) 是在上次操作中存储到 **y1** 的未计算表达式。方程编辑器菜单以底部菜单形式显示。
- ③ 下移光标。显示 **y2=** 提示。
- ④ 在 **y2=** 提示处输入表达式 5(cosx)。注意到 **y2** 的等号(=)在输入 5 后被亮显。而且，**y1** 的等号也被亮显。这表示两个方程都要被图形化（第五章）。

[F1]

方程编辑器菜单



▼

5 [] [COS] [x-VAR] []

改变函数的图形样式

在方程编辑器中，在每个方程左边的图标表示，当方程绘制在图形屏幕上时，该方程的图形显示样式。

- ① 将光标移到 **y1**。
- ② 显示方程编辑器菜单的下一个菜单组。（在菜单组末尾的 ► 表示此菜单有更多的项。）
- ③ 从方程编辑器菜单选择 **STYLE** 为 **y1** 设置 (粗) 图形样式。

▲

[MORE]

[F3]

图形风格图标



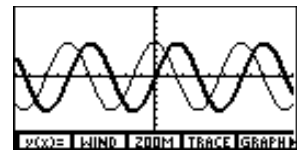
共有 7 种显示图示模式，重复按 [F3] 可浏览不同的图示显示样式。

在图形屏幕上绘制函数

- ❶

从 GRAPH 菜单中选择 **GRAPH** 在图形屏幕上绘图。显示 x 坐标轴、 y 坐标轴和 GRAPH 菜单。然后按在方程编辑器中列出顺序对每个选定的图进行绘制。

[2nd] [M5]



- ❷

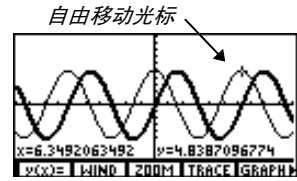
图形绘制后，可以在图形屏幕上自由移动移动光标 (+)。在图形底部显示光标坐标。

▶

▼

◀

▲

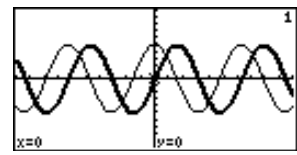


跟踪函数

- ❶

从 GRAPH 菜单中选择 **TRACE** 激活跟踪光标，借助此光标，可以跟踪任何选定函数的图形。当前函数的号码（**y1** 中的 **1**）显示在右上角。

[F4]

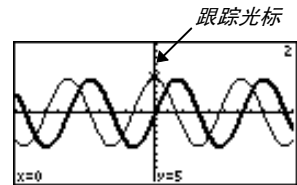


- ❷

从函数 **y1** 移动跟踪光标到函数 **y2**。右上角的 **1** 变成 **2**；**y** 值变为 **y2** 在 $x=0$ 处的值。

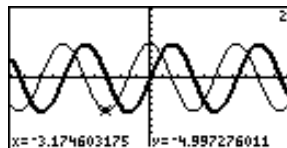
▲

(续)



- ③ 跟踪函数 y_2 。随着跟踪，显示的 y 值是 $5(\cos x)$ 在当前 x 值处的结果，此外 x 值也显示在屏幕上。

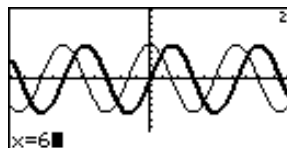
▶ 和 ◀



计算给定 x 值时的 y 值（跟踪期间）

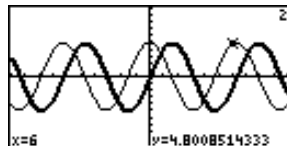
- ① 输入一个在当前图形屏幕的尺寸范围内的实数（或计算结果为实数的表达式）。当输入第一个字符时，显示 $x=$ 提示。

6



- ② 计算在 $x=6$ 时 y_2 的值。跟踪光标直接移到结果处。 y 值，或方程在 x 处的结果，显示在屏幕上。

ENTER



改变窗口变量的值

- ① 显示 GRAPH 菜单。
- ② 从 GRAPH 菜单中选择 **WIND** 以显示窗口编辑器。

GRAPH

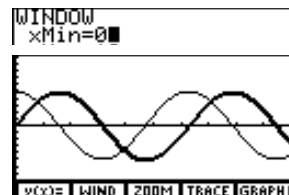
F2



窗口变量的值决定图形屏幕的尺寸。

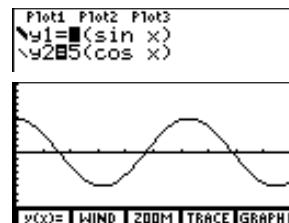
（续）

- ③ 将存储在窗口变量 **xMin** 中的值变为 **0**。 0
- ④ 在重新定义的屏幕上绘图。因为 **xMin=0**， [F5]
- 所以只显示了图形平面的第一和第四象限。



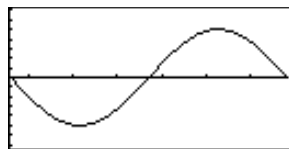
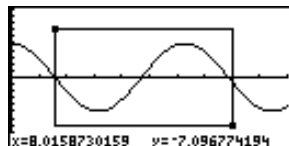
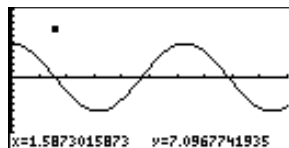
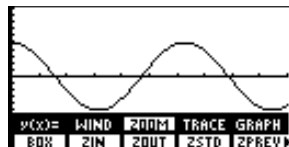
取消选定函数

- ① 从 GRAPH 菜单中选择 **y(x)=** 显示方程编辑器 [F1]
- 和方程编辑器菜单。GRAPH 菜单上移且 **y(x)=** 亮显。
- ② 从方程编辑器菜单中选择 **SELCT** 取消选定 [F5]
- 函数 **y1=**。等号不再加亮。
- ③ 在图形屏幕上绘图。因为取消选定 **y1**， [2nd] [M5]
- TI-86 只绘制 **y2**。要在方程编辑器中选择函数，请重复这些步骤。（**SELCT** 可以选择和取消选定函数。）



放大图形屏幕一部分

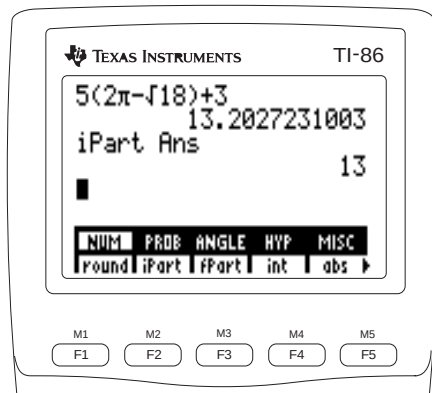
- ① 选择 **ZOOM** 以显示 GRAPH ZOOM 菜单。[F3]
GRAPH 菜单上移且 **ZOOM** 亮显。
- ② 从 GRAPH ZOOM 菜单中选择 **BOX** 以激活缩放框光标。[F1]
- ③ 将缩放框光标移到重新定义的图形屏幕的一角上，然后用一个小正方形来标记它。
 [▶] [▼] [◀] [▲]
[ENTER]
- ④ 将光标从小正方形移到重新定义的图形屏幕的对角上。在移动光标时，就在图上画了一个长方形。
 [▶] [▼] [◀] [▲]
- ⑤ 放大图形。窗口变量自动变为放缩框的指定数值。[ENTER]
- ⑥ 从图形屏幕上清除菜单。[CLEAR]



1

运行 TI-86

安装或更换电池.....	16
开启和关闭 TI-86.....	17
调节显示对比度.....	17
主屏幕	18
输入数字	19
输入其他字符.....	21
输入表达式和指令.....	24
诊断错误	27
重用以前输入及上次结果	28
使用 TI-86 菜单.....	31
查看并改变方式.....	34



安装或更换电池

新的 TI-86 使用四节 AAA 碱性电池。在开启计算器前，必须先安装这些电池。一节锂备用电池已经安装在计算器内。

为表达 [2nd] 和 [ALPHA] 击键组合，本指南对键上方的字用 ([和]) 括起来。

不要取下锂备用电池，除非安装了四节新的 AAA 电池。
适当地处置旧电池。

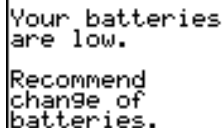
- ❶ 如果计算器已经开启，则（按 [2nd] [OFF] 键）关闭计算器，以避免保存在内存中的信息丢失。
- ❷ 滑动键盘上的保护盖。
- ❸ 竖握计算器，向下推电池盖插销，然后取下电池盖。
- ❹ 取出所有四节旧电池。
- ❺ 安装四节新的 AAA 碱性电池，安放时请参照电池盒里的极性图示（+ 和 - ）。
- ❻ 合上电池盖时，首先把两个插销插到电池盒底部的两个槽内，然后推电池盖直到插销掀钮关上。

更换电池时间

如果不是频繁使用 TI-86，第一次出现电量不足信息后，电池可以继续使用超过两个星期。

当 AAA 电池电量不足时，开启计算器会显示电池电量不足信息。通常，计算器一般在第一次显示电量不足后可继续运行一到两个星期。最终，TI-86 将自动关闭，直到更换电池前都不会运行。

锂备用电池在电池盒中，位于 AAA 电池的上方。它在 AAA 电池电量不足或被更换时保持内存信息。为防止数据丢失，不要取出锂电池，除非安装的四节电池是新的。每隔 3 或 4 年更换锂备用电池。



Your batteries
are low.
Recommend
change of
batteries.

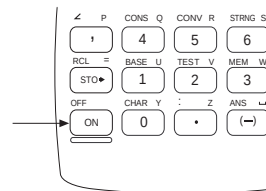
适当地处置旧电池。

要更换锂备用电池，则打开电池盖，拧下固定 BACK UP BATTERY 盖的小螺钉。按照盖上的极性图示（+ 和 -）安装新的 CR1616 或 CR1620 电池。合上盖子，并拧紧螺钉。

开启和关闭 TI-86

要开启 TI-86，则按 **[ON]**。

- ◆ 如果计算器上次是按 **[2nd] [OFF]** 关闭的，TI-86 则清除所有错误，并显示上次使用时最后显示的主屏幕。
- ◆ 如果计算器上次是由 **Automatic Power Down™** (APD™) 关闭的，TI-86 将回到关闭前的状态，包括显示内容、光标和任何错误。



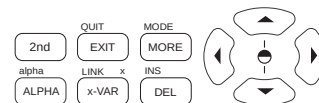
要手工关闭 TI-86，则按 **[2nd] [OFF]**。所有设置和内存内容由 **Constant Memory™** 特性保持。清除任何错误情况。

APD 在大约四分钟内无任何操作时自动关闭 TI-86 以延长电池寿命。

调节显示对比度

如果在调节对比度时松开 **[▲]** 或 **[▼]** 键，则必须再按 **[2nd]** 继续调节。

- ① 按下并松开黄色的 **[2nd]** 键。
- ② 按住 **[▲]** 或 **[▼]**（在半阴影圆圈的上方或下方）。
 - ◆ 要加暗屏幕对比度，则按住 **[▲]**。
 - ◆ 要加亮屏幕对比度，则按住 **[▼]**。



TI-86 有 40 个对比度设置, 因此每个 0 到 9 之间的数字代表四种设置。

可以随时调节显示对比度以适合不同的视角和光照条件。在调节对比度时, 右上角的 **0** (最亮) 到 **9** (最暗) 之间的数字表示当前对比度设置。如果对比度太亮或太暗则看不到此数字。

当电池电量随时间减弱时, 每个数字对应的实际对比度会发生变化。例如, 在新电池时设置对比度为 **3**, 随着电量减弱, 需要设置对比度为 **4**、然后 **5**、**6** 等等才可达到同样效果。不过, 直到出现电量不足的消息前都无须更换电池。

主屏幕

第一次开启 TI-86 时, 显示主屏幕。起初, 主屏幕是空白的, 除了左上角的输入光标 (■)。如果看不见光标, 则按 **[2nd]**, 然后按住 **[v]** 或 **[^]** 来调节对比度 (第 17 页)。

在主屏幕上, 可以输入和计算表达式, 并查看结果。也可以执行指令、保存和调出变量值、设置图象和编辑器。

要从其他任何屏幕返回主屏幕, 则按 **[2nd]** **[QUIT]**。

显示输入和结果

主屏幕最多可以显示 8 行, 每行最多 21 个字符。如果表达式或指令序列的长度超过 21 个字符, 则自动换行。

无须清除主屏幕即可开始新的输入。

方式设置控制 TI-86 解释表达式和显示结果的方式 (第 34 页)。

如果显示行超过 8 行, 文本向上滚动以便于显示。可以按 $\uparrow$ 向上滚动主屏幕, 仅能到当前输入的第一个字符。若要取出、编辑或重新执行以前的输入, 则按 $\boxed{2nd}$ $\boxed{[ENTRY]}$ (第 28 页)。

当在主屏幕上执行输入时, 结果显示在下一行的右侧。当执行指令时, 通常 **Done** 显示在下一行的右侧。

输入 $\rightarrow$ $\log 2$
结果 $\rightarrow$.301029995664

如果结果太长在屏幕上显示不下时, 则起初在右侧显示省略号 (...). 若要查看更多结果, 则按 $\rightarrow$ 。这样做后, 在左侧出现省略号。若要滚回, 则按 $\leftarrow$ 。

输入 $\rightarrow$ $2 \text{ seq}(x, x, 1, 20)$
结果 $\rightarrow$ $\leftarrow$ 2 4 6 8 10 12 14 16...

输入数字

TI-86 屏幕上的分隔符是一个向前的斜线 (/), 如同分数线一样。

每个按键主要功能的符号或缩写以白色印在键上。例如, 当按下 $\boxed{+}$ 时, 加号粘贴到光标处。本指南描述数字输入击键为 **1**、**2**、**3** 等, 而不是 $\boxed{1}$ $\boxed{2}$ $\boxed{3}$ 。

输入负数

要输入负数, 按 $\boxed{-}$ (负值键), 然后按相应数字键。例如, 要输入 -5, 则按 $\boxed{-}$ **5**。不要使用 $\boxed{-}$ (减法键) 表示负数。 $\boxed{-}$ 和 $\boxed{-}$ 是两个功能不同的键。

使用转换指令时，常用括号区分负数（第 4 章）。

仅在科学记数法中，小数点前有一位数字。

在工程记数法中，小数点前有一位、两位或三位数字，10 的指数是 3 的倍数。

TI-86 计算表达式中负数和其他函数的顺序是由 Equation Operating System™（附录）来决定的。例如， -4^2 的结果是 **-16**，而 $(-4)^2$ 的结果是 **16**。如果不能确定计算顺序，可以用 $\square$ 和 $\square$ 区分负号的用途。

使用科学或工程记数法

- 1 输入尾数（指数之前的数字部分）。该值可以是表达式。

$\square$ 19
 $\div$ 2 $\square$

(19/2)■

- 2 把 E 粘贴到光标处。

EE

(19/2)E■

- 3 若指数为负，粘贴 - 到光标处。然后输入指数，可以是一位、两位或三位数字。

$\square$ 2

(19/2)E-2■

- 4 计算表达式的值。

ENTER

(19/2)E-2 .095

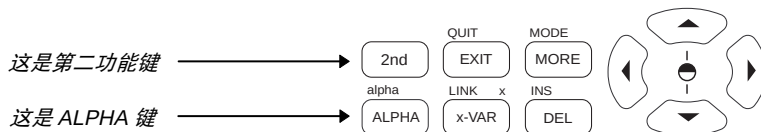
表达式中使用以科学记数法或工程记数法表示的数字时，TI-86 未必以科学记数法或工程记数法显示结果。方式设置（第 34 页）和数字的大小决定显示结果的表示方式。

输入复数

在 TI-86 中，复数 $a+bi$ 的输入有两种方式：直角坐标复数方式 (a,b) 和极坐标复数方式 $(r\angle\theta)$ 。有关复数的更多信息请参阅第 4 章。

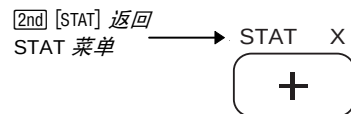
(1,2)+(-3,1) (-2,3)
 (1,2)*3
 (-1.24844050964,2.72...
 ■

输入其他字符



第二功能键

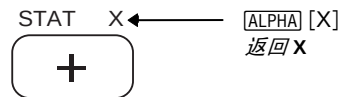
2nd 键是黄色的。按 **2nd** 后，光标变成 **■**（第二功能光标）。当按下下一个键时，印在键上方的黄色字符、缩写或字激活，而不是该键的主要功能。



ALPHA 键

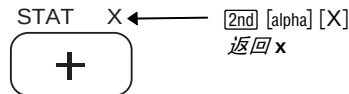
若要在文本中输入一个空格，按 **[ALPHA] []**。空格在变量名中是无效的。

[ALPHA] 键是蓝色的。按 **[ALPHA]** 后，光标变成 **■**（大写字母 (ALPHA) 光标）。按下下一个键后，印在键上方的蓝色大写字母粘贴到光标处。



为方便起见，可以按 **[x-VAR]** 代替 **2nd [alpha] [x]** 来输入常用的 **x** 变量。

按 **2nd [alpha]** 后，光标变成 **■**（小写字母 (alpha) 光标）。按下下一个键后，印在键上方的蓝色小写字符粘贴到光标处。



Name= 提示和存储符号(➔)
自动设置大写字母锁。

大写字母锁和小写字母锁

若要连续输入多于一个大写或小写字母，则设置大写字母锁（用于大写字母）和小写字母锁（用于小写字母）。

要在输入光标显示时设置大写字母锁，则按 **[ALPHA] [ALPHA]**。

- ◆ 若要取消大写字母锁，则按 **[ALPHA]**。
- ◆ 若要切换大写字母锁到小写字母锁，则按 **[2nd] [alpha]**。

要在输入光标显示时设置小写字母锁，则按 **[2nd] [alpha] [ALPHA]**。

- ◆ 若要取消小写字母锁，则按 **[ALPHA] [ALPHA]**。
- ◆ 若要切换小写字母锁到大写字母锁，则按 **[ALPHA]**。

当大写字母锁或小写字母锁开启时可以使用 **[2nd]**。此外，当按其上方没有蓝色字符的键时，例如 **[GRAPH]**、**[DEL]** 或 **[↩]** 键，这些键的主要功能仍有效。

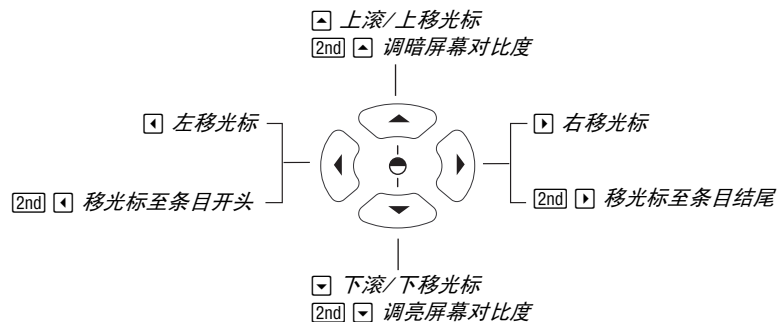
普通光标

多数情况下，光标的形状表示按下一个键时将会发生的动作。

图象和编辑器有时使用其他光标，这将在其他章节里描述。

输入	■	在当前光标处输入字符，替换已有任何字符
插入	—	在当前光标处插入字符，右移原有的字符
第二功能	Ⓜ	输入第二功能字符或执行第二功能运算（键盘上的黄色运算符）
大写字母	Ⓐ	输入大写字母（键盘上的蓝色字符）
小写字母	ⓐ	输入大写字母（键盘上的蓝色字符）的小写形式
充满	■	不能输入，在提示处已输入最大字符数或内存已满
◆		如果按 [2nd] [INS] 后按 [ALPHA] ，光标变成一个带下划线的 A (A) 。
◆		如果按 [2nd] [INS] 后按 [2nd] [ALPHA] ，光标变成一个带下划线的 a (a) 。
◆		如果按 [2nd] [INS] 后按 [2nd] ，插入光标变成一个带下划线的 ↑ (↑) 。

光标方向键



按住 $\rightarrow$ 、 $\downarrow$ 、 $\leftarrow$ 或 $\uparrow$ 时，光标会持续移动。

插入、删除和清除字符

输入光标 (■) 覆盖当前字符。

2nd [INS] 改变光标为插入光标 (■)；在插入光标处插入字符，右移已有的字符；若要取消插入，则按 2nd [INS] 或按 $\rightarrow$ 、 $\downarrow$ 、 $\leftarrow$ 或 $\uparrow$

[DEL] 删除光标处的字符；若要连续删除右侧字符，则按住 [DEL]

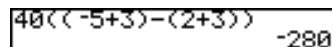
[CLEAR] 清除主屏幕上的当前输入； [CLEAR] [CLEAR] 清除整个主屏幕

输入表达式和指令

输入表达式

表达式是指一组数字和作为一个或多个函数参数的变量的组合。在 TI-86 里, 通常按书写顺序输入表达式。例如: πr^2 、 $5 \tan xStat$ 和 $40((-5+3)-(2+3))$ 都是一个表达式。

可在主屏幕里使用表达式计算结果。



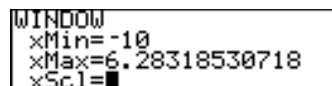
```
40((-5+3)-(2+3))
-280
```

在大部分需要数值的位置, 都可以用表达式输入数值。



```
WINDOW
xMin=-10
xMax=2π
```

例如, 可以输入一个表达式作为窗口变量 (第 5 章)。当按 ∇ 、 $\blacktriangle$ 、 $\boxed{\text{ENTER}}$ 或 $\boxed{\text{EXIT}}$ 时, TI-86 计算这个表达式并用结果替换窗口变量。

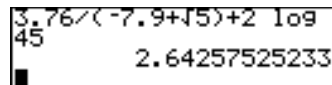


```
WINDOW
xMin=-10
xMax=6.28318530718
xScl=
```

要输入表达式, 则从键盘和菜单输入数字、变量和函数 (第 31 页)。按 $\boxed{\text{ENTER}}$ 后, 根据 EOS 计算顺序规则 (参阅附录) 计算表达式 (与光标位置无关), 然后显示结果。

要输入表达式 $3.76 \div (-7.9 + \sqrt{5}) + 2 \log 45$ 然后计算结果, 应按下面的键:

$3 \boxed{.} \boxed{7} \boxed{6} \boxed{\div} \boxed{(} \boxed{-} \boxed{7} \boxed{.} \boxed{9} \boxed{+} \boxed{2nd} \boxed{\sqrt{}} \boxed{5} \boxed{)} \boxed{+} \boxed{2} \boxed{\text{LOG}} \boxed{4} \boxed{5} \boxed{\text{ENTER}}$



```
3.76/(-7.9+√5)+2 log
45
2.64257525233
```

在表达式中使用函数

函数返回一个值，函数的例子有： $\div$ 、 $-$ 、 $+$ 、 $\sqrt{\quad}$ 和 **log**。若要使用函数，通常必须输入一个或多个有效参数。

在本指南里，可选参数用 ([和]) 括起来。当输入参数时，请勿输入这些括弧。

本指南描述函数或指令的语法时，参数用斜体表示。例如：**sin** *angle*。按 **[SIN]** 输入 **sin**，然后输入有效的角度值 *angle*（或结果为角度 *angle* 的表达式）。对于带多个参数的函数或指令，必须用逗号分隔每个参数。

一些函数要求参数在括号中。当不能确定计算顺序时，可以使用括号明确函数在表达式中的位置。

使用指令

“A 到 Z 参考”描述所有 TI-86 的函数和指令，包括必需和可选的参数。

所有指令都产生操作。例如，**CIDrw** 是一条指令，执行时清除图象中所有绘制的内容。指令不能用在表达式里。一般地，在 TI-86 中每个指令名的第一个字母大写。一些指令要求一个以上的参数，用指令名后的左括号 (() 表示。例如：**Circl**(要三个参数，**Circl**(*x,y,radius*)。

输入函数、指令和操作符

在 CATALOG 菜单里，可以按字母直接移动光标到以该字母开头的第一项（例如示例中的 [L]）。

可以用下面三种方式之一输入函数、指令或操作符（例如，**log 45**）。

- ◆ 从键盘或菜单将之粘贴到光标处 (**[LOG]** **45**)。
- ◆ 从 CATALOG 将之粘贴到光标处 (**[2nd]** **[CATLG-VARS]** **[F1]** **[L]** **[F1]** **[F1]** **[ENTER]** **45**)。
- ◆ 逐个输入字符 (**[2nd]** **[alpha]** **[ALPHA]** **[L]** **[O]** **[G]** **[=]** **[ALPHA]** **[ALPHA]** **45**)。

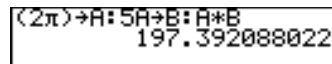
正如示例中所示，使用内置函数或指令通常更容易些。

当选择一个函数、指令或操作符时，由一个或多个字符组成的符号粘贴到光标处。一旦符号粘贴到光标处，就可以编辑单个字符。

例如，假定已按 **[2nd]** **[CATLG-VARS]** **[MORE]** **[MORE]** **[F5]** **[F1]** **[F1]** **[ENTER]** 将 **yMin** 作为表达式的一部分粘贴到光标处。然后发现需要的是 **xMin**，这时不需要按下 9 个键来选择 **xMin**，按 **[←]** **[←]** **[←]** **[←]** **[x-VAR]** 即可。

输入连续输入项

要连续输入两个或多个表达式或指令，请用冒号 (**[2nd]** **[:]**) 将它们分开。按 **[ENTER]** 后，TI-86 从左至右执行每个输入，并且显示最后一个表达式或指令的结果。整个输入组保存在上次输入中（第 28 页）。



(2π)+A:5A+B:A*B
197.392088022

在示例中 → 符号表示符号前的值保存到符号后的变量中（第 2 章）。按 **[STO→]**，可将 → 粘贴到屏幕上。

忙指示器

当 TI-86 正在计算或绘图时，一个移动的垂直线作为忙显示器显示在屏幕右上角。当暂停图象或程序时，忙指示器由暂停指示器代替，即一条移动的垂直点画线。

中断计算或图象

要中断正在进行的计算或图象，则按 **[ON]**。当中断一个计算时，显示 **ERROR 06 BREAK** 信息和菜单。

- ◆ 要返回主屏幕，则选择 **QUIT**（按 **[F5]**）。
- ◆ 要回到表达式开始处，则选择 **GOTO**（按 **[F1]**）。按 **[ENTER]** 重新计算表达式。

第 5 章：函数图象介绍如何绘制图象。

如果执行程序时，所保存的方程中出现语法错误，则选择 **GOTO** 返回到方程编辑器，而不是返回程序（第 5 章）。

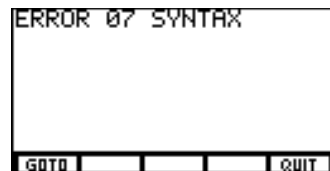
中断图象时，显示部分图象和 **GRAPH** 菜单。

- ◆ 要返回主屏幕，则按 **CLEAR** **CLEAR** 或任意非图象键。
- ◆ 要重新绘制图象，则选择一条显示图象的指令。

诊断错误

TI-86 检测到错误时，会返回错误信息，例如 **ERROR 04 DOMAIN** 或 **ERROR 07 SYNTAX**。附录中对每个错误类型和可能的错误原因进行了说明。

- ◆ 如果选择 **QUIT**（或按 **2nd** **[QUIT]**、**CLEAR** **CLEAR**），则显示主屏幕。
- ◆ 如果选择 **GOTO**，显示前一个屏幕，光标位于或接近错误处。



改正错误

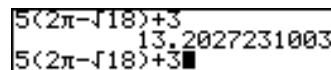
- ① 注意错误类型 (**ERROR ## errorType**)。
- ② 如果可能，选择 **GOTO**。显示前一个屏幕，光标位于或接近错误处。
- ③ 确定错误原因。如果不能，可在附录中查看可能的原因。
- ④ 改正错误后继续。

重用以前输入及上次结果

取出上次输入

当在主屏幕中按 **[ENTER]** 计算表达式或执行指令时, 整个表达式或指令放在一个名为 ENTRY (上次输入) 的存储区域中。当关闭 TI-86 时, ENTRY 保留在内存中。

要取出上次输入, 则按 **[2nd] [ENTRY]**。当前行被清除, 上次输入粘贴到当前行。



取出并编辑上次输入

❶ 在主屏幕上, 取出前一个输入。

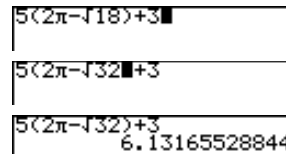
[2nd] [ENTRY]

❷ 编辑取出的输入。

[←] [←] [←] [←] [←] 32

❸ 重新执行编辑过的输入。

[ENTER]



取出以前输入

TI-86 尽可能多地在 ENTRY 中保留以前输入, 最多可到 128 字节。要滚动保存在 ENTRY 中最新输入到较早的输入, 重复按 **[2nd] [ENTRY]**。如果在显示了最早的输入后按 **[2nd] [ENTRY]**, 又会显示最新的输入; 不断按 **[2nd] [ENTRY]** 重复此次序。

用冒号分隔的连续输入项保存为一个输入 (第 26 页)。

求圆面积公式: $A=\pi r^2$ 。

方程求解器 (第 15 章) 是执行这个任务的另一个工具。

取出多个输入

要将两个或多个表达式或指令一起保存在 ENTRY 中, 则将它们输入在一行中, 相互之间用冒号分开, 然后按 **ENTER**。执行时, 整个组保存在 ENTRY 中。有许多操作这种特性以避免乏味的人工输入的方法, 下面的示例给出了其中之一。

- 1 使用试错法求面积为 200 平方厘米的圆半径。将 **8** 作为第一个估计值保存到 **r** 中, 然后执行 πr^2 。

8 **STO→** **[2nd]** **[alpha]** **[R]**
[2nd] **[:]** **[2nd]** **[π]** **[R]** **[ALPHA]**
[ALPHA] **[x²]** **ENTER**

```
8→r:πr²
201.06192983
```

- 2 取出 **8→r:πr²** 并将 **7.958** 作为一个新的估计值插入。继续估计直到结果逼近结果 **200**。

[2nd] **[ENTRY]**
[2nd] **[←]** **7** **[2nd]** **[INS]** **[□]** **958**
ENTER

```
8→r:πr²      201.06192983
7.958→r:πr²  198.956321336
```

清除 ENTRY 存储区域

要清除 ENTRY 存储区域中的所有数据, 从主屏幕的空行开始, 从 MEM 菜单中选择 **ClrEnt** (按 **[2nd]** **[MEM]** **[F5]**), 然后按 **ENTER**。

取出上次结果

当在主屏幕上或程序中成功地计算表达式后, TI-86 将结果保存到一个名为 **Ans** (上次结果) 的内置变量中。 **Ans** 可以是实数或复数、数组、向量、矩阵或字符串。当关闭 TI-86 后, **Ans** 中的值保存在内存中。

要将变量名 **Ans** 拷贝到光标处, 则按 $\boxed{2\text{nd}} \boxed{[\text{ANS}]}$ 。可以在任何地方使用变量 **Ans**, 只要其中保存的值是有效的。当计算表达式时, TI-86 用保存在 **Ans** 中的值计算结果。

- ① 计算花园的面积, 宽为 **1.7** 米, 长为 **4.2** 米。

$1 \boxed{.} 7 \boxed{\times} 4 \boxed{.} 2$
 $\boxed{\text{ENTER}}$

- ② 如果要种植 **147** 棵西红柿, 计算每平方米种几棵。

$147 \boxed{\div} \boxed{2\text{nd}} \boxed{[\text{ANS}]}$
 $\boxed{\text{ENTER}}$

1.7*4.2	7.14
147/Ans	20.5882352941

在函数前使用 Ans

上次结果保存在 **Ans** 中。如果输入表达式时首先输入的是一个需要前置参数的函数, TI-86 自动将 **Ans** 作为函数的参数输入。

- ① 输入并执行表达式。

$5 \boxed{\div} 2 \boxed{\text{ENTER}}$

- ② 输入一个没有参数的函数。 **Ans** 被粘贴到屏幕上, 后面跟着函数。

$\boxed{\times} 9 \boxed{.} 9$
 $\boxed{\text{ENTER}}$

5/2	2.5
Ans*9.9	24.75

保存结果到变量

- ① 计算半径为 **5** 米的圆面积。

$\boxed{2\text{nd}} \boxed{[\pi]} 5 \boxed{\times} 2$
 $\boxed{\text{ENTER}}$

- ② 计算高为 **3.3** 米、半径为 **5** 米的圆柱体积。

$\boxed{\times} 3 \boxed{.} 3$
 $\boxed{\text{ENTER}}$

- ③ 保存结果到变量 **V** 中。

$\boxed{\text{STO}} \boxed{\rightarrow} \boxed{V}$
 $\boxed{\text{ENTER}}$

$\pi 5^2$	78.5398163397
Ans*3.3	259.181393921
Ans→V	259.181393921

使用 TI-86 菜单

许多 TI-86 功能的符号可以在菜单中找到，而不是在 TI-86 键盘上。

显示菜单

显示特定菜单的方式取决于菜单在 TI-86 上的位置。

显示菜单的方法	示例
按上面有菜单名的键	GRAPH 显示 GRAPH 菜单
按 2nd ，然后按第二功能键菜单名	2nd [MATH] 显示 MATH 菜单
从其他菜单中选择菜单名	2nd [MATH] [F1] 显示 MATH NUM 菜单
选择编辑器或选择屏幕	2nd [LIST] [F4] 显示数组编辑器菜单，同时显示数组编辑器
偶然出现错误	1 [STO▶] [ENTER] 显示错误菜单

某些 TI-86 菜单有 25 个项。

当显示菜单时，由 1 到 5 个项组成的菜单组显示在屏幕底部。如果在菜单组的第 5 个项后显示更多符号 (▶)，菜单至少还有一个菜单组。要查看下一个菜单组，则按 **[MORE]**。最后一个菜单组不会显示 ▶ 符号。

例如，按 **2nd** **[MATH]** 显示 MATH 菜单。

当此处看到 ▶ ...



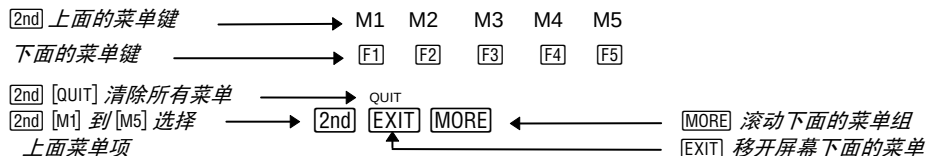
...按 **[MORE]** 显示
下一个菜单组。



在最后一个菜单组中按 **[MORE]** 后，又回到第一个菜单组。

▶、◀、◂ 和 ▸ 对菜单
无效。

菜单键

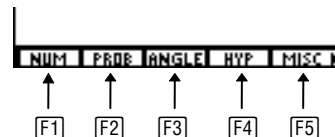


选择菜单项

附录中的菜单图显示了 TI-86 中的每个菜单。

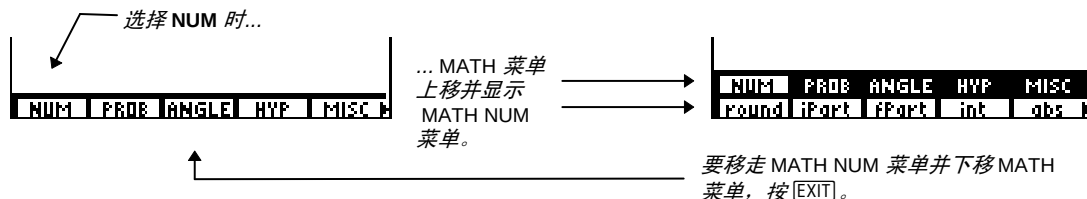
通常, TI-86 菜单项的长度不多于 5 个字符。

显示菜单时, 显示一到五个菜单项。要选择一个菜单项, 直接按菜单项下方的选择键。例如, 在右侧的 MATH 菜单中按 [F1] 选择 **NUM**, 按 [F2] 选择 **PROB**, 等等。



选择显示其他菜单的菜单项时, 第一个菜单移到屏幕的上一行, 给新的菜单留出位置。在原菜单上的所有菜单项按相反颜色显示, 除了所选菜单项。

[MORE] 只滚动下面的菜单, 它不能滚动上面的菜单。



要从上面的菜单选择菜单项，则先按 [2nd]，然后按对应键 [M1] - [M5]。

要从上面的菜单选择 **PROB**，则按 [2nd] [M2]。
要从下面的菜单选择 **iPart**，则按 [F2]。



当一个编辑器菜单显示为上面的菜单时，如果在下面的菜单中选择将显示另一个菜单的项，那么编辑器菜单仍为上面的菜单。

从下面的菜单中选择 **NUM** 时...



...方程编辑器
菜单保留并
显示 MATH NUM
菜单。

MATH 菜单消失。



上面: 方程编辑器菜单
下面: MATH 菜单

上面: 方程编辑器菜单
下面: MATH NUM 菜单

退出 (删除) 菜单

要从图像屏幕的底部删除菜单，则绘制完图像后按 [CLEAR] (第5章)。

要从屏幕删除下面的菜单，则按 [EXIT]。

当按 [EXIT] 时...



上面: MATH 菜单
下面: MATH NUM 菜单

...MATH NUM 菜单消失，MATH 菜单下移。



下面: MATH 菜单

再按 [EXIT]，
MATH 菜单消失。



无菜单

在右侧的屏幕上，默认方式设置以亮显方式显示在屏幕左侧。

本示例将十进制方式设为 2，如同美元和美分。

在 Normal 记数法中，如果结果超过了 12 位或者结果的绝对值 < .001，则以科学记数法显示。

记数法方式不影响数值的输入。

查看并改变方式

要显示方式设置，则按 **[2nd] [MODE]**。当前设置亮显。方式设置控制 TI-86 如何显示并解释数字和图象。当 TI-86 关闭时，Constant Memory 特性保存当前方式设置。所有数值，包括矩阵和数组的元素根据方式设置显示。

改变方式设置

- ❶

将光标移到想要改变的设置所在的行（在示例中为十进制设置）。

▾
- ❷

将光标移到想要改变的设置上（2 位小数）。

▶▶▶
- ❸

执行改变。

ENTER

```
Normal Sci Eng
Float 012345678901
Radian Degree
RectC PolarC
Func Pol Param DfEq
Dec Bin Oct Hex
RectV CylV SphereV
dxDer1 dxNDer
```

```
Normal Sci Eng
Float 012345678901
Radian Degree
RectC PolarC
Func Pol Param DfEq
Dec Bin Oct Hex
RectV CylV SphereV
dxDer1 dxNDer
```

记数法方式

- Normal

以数位在小数点的左侧和右侧的形式显示结果（例如 **123456.7**）
- Sci

（科学记数法）将结果显示为两部分：基数（一位小数）显示在 **E** 的左边，相应 10 的幂显示在 **E** 的右边（例如 **1.234567E5**）
- Eng

（工程记数法）将结果显示为两部分：基数（一位、两位或三位小数）显示在 **E** 的左边，相应 10 的幂（总是 3 的倍数）显示在 **E** 的右边（例如 **123.4567E3**）

十进制方式

Float

(浮点) 结果最多显示为 12 位, 包括符号和小数点

(定点)

(**012345678901** ; 每个数字为一种设置) 以指定的小数位数显示结果 (结果四舍五入到指定的数位); 第二个 **0** 设置 10 位小数; 第二个 **1** 设置 11 位小数

角度方式

Radian

将角度解释为弧度, 按弧度显示结果

Degree

将角度解释为度, 按度显示结果

复数方式

RectC

(直角坐标复数方式) 将复数结果显示为 *(real,imaginary)*

PolarC

(极坐标复数方式) 将复数结果显示为 *(magnitude∠angle)*

图象方式

Func

(函数图象) 绘制函数图象, 其中 **y** 是 **x** 的函数

Pol

(极坐标图象) 绘制函数图象, 其中 **r** 是 **θ** 的函数

Param

(参数图象) 绘制关系图象, 其中 **x** 和 **y** 是 **t** 的函数

DifEq

(微分方程图象) 根据 **t** 绘制微分方程图象

数制方式

Dec

(十进制制) 将数解释和显示为十进制数

Bin

(二进制制) 将数解释为二进制数; 结果带后缀 **b**

Oct

(八进制制) 将数解释为八进制数; 结果带后缀 **o**

Hex

(十六进制制) 将数解释为十六进制数; 结果带后缀 **h**

非十进制方式仅在主屏幕或程序编辑器中有效。

向量方式不影响向量输入的方式。

向量坐标方式

- RectV** (直角向量坐标) 结果显示为: 二维向量 $[x\ y]$, 三维向量 $[x\ y\ z]$
- CylV** (柱面向量坐标) 结果显示为: 二维向量 $[r\ \angle\theta]$, 三维向量 $[r\ \angle\theta\ z]$
- SphereV** (球面向量坐标) 结果显示为: 二维向量 $[r\ \angle\ \theta]$, 三维向量 $[r\ \angle\theta\ \angle\phi]$

微分方式

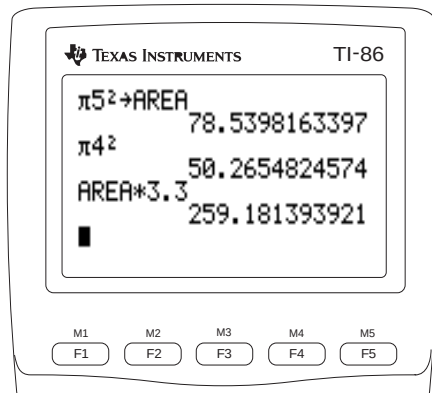
保存到 δ 中的值影响 **dxNDer** (参阅附录)。

- dxDer1** (精确微分) 用 **der1** (第 3 章) 来进行精确微分, 并计算表达式中每个函数的值 (**dxDer1** 比 **dxNDer** 精确, 但是它对表达式中函数的有效性有限制)。
- dxNDer** (数值微分) 用 **nDer** 来进行数值微分, 并计算表达式的值 (**dxNDer** 不如 **dxDer1** 精确, 但是它对表达式中函数的有效性限制不严)。

2

CATALOG、 变量和字符

CATALOG	38
存储数据到变量	39
变量按数据类型分类	42
CUSTOM 菜单	44
CHAR（字符）菜单	45

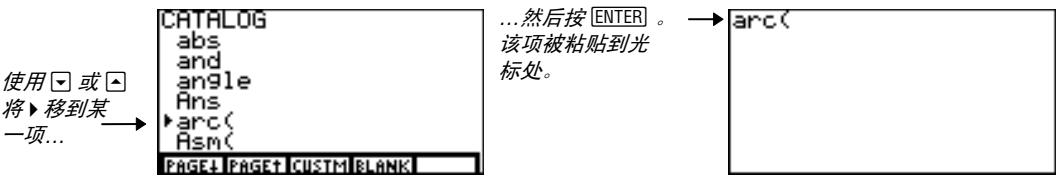


CATALOG 是 CATLG-VARS 菜单中的第一项。

CATALOG [2nd] [CATLG-VARS] [F1]

CATALOG 按字母顺序显示 TI-86 的所有函数和指令。不以字母开头的项（例如 + 或 Bin）显示在 CATALOG 末尾。

选择光标 (▸) 指示当前项。要从 CATALOG 选择某一项，则将选择光标移到该项处并按 [ENTER] 。 CATALOG 消失，所选项的名字粘贴到光标原位置。



要跳转...	操作:
到以特定字母开头的第一项	按该字母；大写字母锁开启
到 CATALOG 末尾的特殊字符	从 CATALOG 第一项按 [down]
到下一屏幕	从 CATALOG 菜单中选择 PAGE↓ ([F1])
到上一屏幕	从 CATALOG 菜单中选择 PAGE↑ ([F2])

菜单项 **CUSTM** 和 **BLANK** 在 CATALOG 菜单和每个 VARS 屏幕菜单中。借助它们，用户可以创建自己的 CUSTOM 菜单，其中可多达 15 个 CATALOG 项和变量，包括程序名。有关 CUSTOM 菜单的详细信息，请参阅第 44 页。

存储数据到变量

在 TI-86 上,可以使用下述几种方法将数据存储到变量。可以:

- ◆ 用 **[STO➡]** 存储数值到变量。
- ◆ 用 **=** 存储未计算表达式到方程变量。
- ◆ 用编辑器的 **Name=** 提示存储几种类型的数据到变量。
- ◆ 改变 TI-86 设置,或复位默认设置和内存为出厂设置。
- ◆ 执行能引起 TI-86 自动存储数据到内置变量的函数。

TI-86 有用于特殊目的的内置变量名,如方程变量、数组名、统计结果变量、窗口变量以及 **Ans**。可以存储数值到某些变量中。本指南的相应章节介绍这些变量。

创建变量名

当用 **[STO➡]**、**=**、或 **Name=** 提示来存储数据时,用户可以创建自己的变量名。当创建用户创建变量名时,请遵循以下规则。

- ◆ 用户创建变量名长度为 1 到 8 个字符。
- ◆ 第一个字符必须是字母,包括所有的 CHAR GREEK 菜单项,以及 CHAR MISC 菜单中的 Ñ、ñ、Ç 和 ç。
- ◆ 用户创建变量名不能与 TI-86 功能符号或内置变量名相同。例如不能创建 **abs**, 因为 **abs** 是绝对值函数符号。也不能创建 **Ans**, 因为它已经是内置变量名。
- ◆ TI-86 区分变量名中的大写和小写字符。例如, **ANS**、**Ans** 和 **ans** 是三个不同的变量名。因此,只有 **Ans** 是内置变量名; **ANS** 和 **ans** 可以作为用户创建变量名。

本章描述这里列举的数据存储方法中的前两个。其他方法将在相应章节里描述。

存储数值到变量名

- | | | |
|--|--|--|
| ① 输入一个数值，可以是表达式。 | $\boxed{2\text{nd}} \boxed{[\pi]} \boxed{5} \boxed{[x^2]}$ | $\pi 5^2$ |
| ② 在数值后面输入 $\rightarrow$ （存储符号）。 | $\boxed{\text{STO}\rightarrow}$ | $\pi 5^2 \rightarrow$ |
| ③ 创建一个以字母开头、长度为 1 到 8 个字符的变量名。大写字母锁开启。 | $\boxed{[A]} \boxed{[R]} \boxed{[E]} \boxed{[A]}$ | $\pi 5^2 \rightarrow \text{AREA}$ |
| ④ 存储数值到变量。存储到变量的数值作为结果显示。 | $\boxed{\text{ENTER}}$ | $\pi 5^2 \rightarrow \text{AREA}$
78.5398163397 |

存储未计算表达式

当用 $\boxed{\text{STO}\rightarrow}$ （和 $\rightarrow$ 号一起使用）将表达式存储到内存时，先计算表达式的值，然后将结果存储到一个变量中。

当用 $\boxed{[\text{ALPHA}]} \boxed{[-]}$ 、方程编辑器（第 5 章）或方程求解器（第 15 章）存储未计算表达式时，未计算表达式存储到一个方程变量。

当使用 $=$ 时，先输入变量，然后是 $=$ ，最后是表达式。

相反，当使用 $\rightarrow$ 时，先输入数值，然后是 $\rightarrow$ ，最后是变量。

在主屏幕上或在程序中存储未计算表达式的语法为：

$variable = expression$

其中变量 $variable$ 总在等号之前，而表达式 $expression$ 总在等号之后。

可以用 $=$ 存储数学表达式到一个方程变量。例如， $F = M * A$ 。

存储结果

要在计算另一个表达式之前存储结果到变量，则用 **STO►** 和 **Ans**。

在示例中，TI-86 将存储在 **AREA** 中的值乘以 3.3。

要将 **AREA** 粘贴到光标处，则按 **2nd** **[CATLG-VARS]** **F3**，将选择光标 (**►**) 移到 **AREA**，然后按 **ENTER**。

- ① 输入和计算表达式。

[ALPHA] **[ALPHA]**
[A] **[R]** **[E]** **[A]** **[ALPHA]**
[X] **3** **.** **3** **ENTER**

```
AREA*3.3
259.181393921
```

- ② 存储结果到用户创建变量或有效的内置变量。存储到变量中的数值作为结果显示。

STO► **[V]** **[O]** **[L]** **ENTER**

```
AREA*3.3
259.181393921
Ans→VOL
259.181393921
```

复制变量值

将变量 *variableA* 的内容复制到变量 *variableB* 的语法为：
variableA►*variableB*

例如，**RegEq**►**y1** 将回归方程（第 14 章）存储到变量 **y1**。

要粘贴►到光标处，则按 **STO►**。

显示变量值

- ① 使光标位于主屏幕上的空行，粘贴变量名到如上所描述的光标处。

2nd **[CATLG-VARS]**
F3 **▼**（位置可能变化）**ENTER**

```
VOL
259.181393921
```

- ② 显示变量内容。

ENTER

也可以通过将包含数据类型的变量显示于相应的编辑器（如数组编辑器或窗口变量编辑器）或图象中来显示它们。这些方法在本指南的后续章节详细描述。

要粘贴一个变量名，则从 **VARS** 菜单（第 42 页）中选择它。

调用变量值

1

移动光标到要插入待调用变量值的位置。

100

✕

2

在屏幕底部显示 **Rcl** 提示。大写字母锁开启。

2nd

[RCL]

3

输入要调用的变量名。

[V]

[O]

[L]

4

变量内容调用到光标处。 **Rcl** 提示消失，输入光标出现。

ENTER

100*

Rcl

Rcl VOL

100*259.181393921

要取消 RCL，则按 **CLEAR**。

编辑一个被调用的数值不会改变存储在变量中的数值。

当在编辑器中存储数据时，TI-86 根据编辑器识别数据类型。例如，使用向量编辑器时只存储向量。

变量按数据类型分类

TI-86 根据数据类型对变量分类并将每个变量放在一种数据类型选择屏幕上。可以通过从 CATLG-VARS 菜单中选择相应数据类型来显示每个屏幕（如第 43 页所描述）。下面是一些示例。

如果数据...	TI-86 将数据类型分为...	例如：
以 { 开头且以 } 结尾	数组（ VARS LIST 屏幕）	{1,2,3}
以 [开头且以] 结尾	向量（ VARS VECTR 屏幕）	[1,2,3]
以 [[开头且以]] 结尾	矩阵（ VARS MATRX 屏幕）	[[1,2,3][4,5,6][7,8,9]]

要显示其他菜单组，则按
[MORE]。

CATLG-VARS (CATALOG — 变量) 菜单 [2nd] [CATLG-VARS]

CATLG	ALL	REAL	CPLX	LIST	▶	VECTR	MATRX	STRNG	EQU	CONS	
						▶	PRGM	GDB	PIC	STAT	WIND

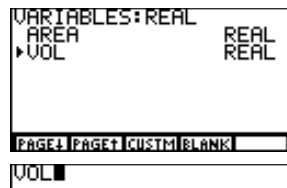
数组名 fStat、xStat 和 yStat
是 VARS STAT 屏幕中的统计
结果变量。

CATLG	显示 CATALOG 菜单
ALL	显示选择屏幕，其中显示所有变量和所有数据类型名
REAL	显示选择屏幕，其中显示所有实数变量
CPLX	显示选择屏幕，其中显示所有复数变量
LIST	显示选择屏幕，其中显示所有数组名
VECTR	显示选择屏幕，其中显示所有向量名
MATRX	显示选择屏幕，其中显示所有矩阵名
STRNG	显示选择屏幕，其中显示所有字符串变量
EQU	显示选择屏幕，其中显示所有方程变量
CONS	显示选择屏幕，其中显示所有用户定义常量
PRGM	显示选择屏幕，其中显示所有程序名
GDB	显示选择屏幕，其中显示所有图象数据库名
PIC	显示选择屏幕，其中显示所有图形名
STAT	显示选择屏幕，其中显示所有统计结果变量
WIND	显示选择屏幕，其中显示所有窗口变量

本例假定第 41 页示例中的实数变量 AREA 和 VOL 尚未从内存删除。

选择变量名

- 1 从 CATLG-VARS 菜单中选择相应数据类型
的选择屏幕。 [2nd] [CATLG-VARS] [F3]
- 2 将光标移到要选择的变量。 ▼
- 3 选择所需变量。 [ENTER]



CUSTOM 菜单 [2nd] [CATLG-VARS] [F1] [F3]

为了创建自己的 CUSTOM 菜单，最多可以从 CATALOG 和 VARS 屏幕中选择 15 项——程序名、函数、指令和其他项等。要显示 CUSTOM 菜单，则按 [CUSTOM]。使用 [F1] 到 [F5] 和 [MORE] 选择其他菜单的项。

输入 CUSTOM 菜单项

- 1 从 CATALOG 中选择 **CUSTOM**。CUSTOM 菜单显示。大写字母锁开启。 [2nd] [CATLG-VARS]
[F1] [F3]
- 2 将选择光标 (▶) 移到要拷贝到 CUSTOM 菜单中的项。 [C] ▼ ▼ ▼
- 3 将该项拷贝到选定的 CUSTOM 菜单单元，替换原有的项。 [F3]
- 4 要输入更多的项，用不同的项和单元来重复步骤 2 和 3。
- 5 显示 CUSTOM 菜单。 [2nd] [QUIT] [CUSTOM]



当向 CUSTOM 菜单中拷贝项时，可以跳过菜单单元和菜单组。

要从第二个或第三个菜单组清除一项，则按 **[MORE]** 直到该项被显示，然后选择该项。

清除 **CUSTOM** 菜单项

- ❶ 从 CATALOG 菜单选择 **BLANK**。
CUSTOM BLANK 菜单显示。
- ❷ 清除菜单项。
- ❸ 要清除更多项，重复步骤 2 和 3。

[2nd] **[CATLG-VARS]**
[F1] **[F4]**

[F3]



不能删除 TI-86 的内置变量。

使用 **DelVar**(不能删除程序变量)。

从内存中删除变量

在主屏幕或程序中，从内存中删除用户创建变量名（程序名除外）及其内容的语法为：

DelVar(variable)

要删除用户创建的变量名及其内容（包括程序名），则显示 **MEM DELET** 菜单 (**[2nd]** **[MEM]** **[F2]**)，选择数据类型，选择变量，然后按 **[ENTER]**（第 16 章）。不从 **CUSTOM** 菜单中删除变量（第 44 页）。

CHAR（字符）菜单 **[2nd]** **[CHAR]**



Ñ、ñ、Ç 和 ç 作为变量名的字符是有效的，包括第一个字符。

%, '和! 可以是函数。

所有的CHAR GREEK 菜单项都是有效的变量名字符，包括第一个字母。 π (2nd [π]) 不能作为有效的字符； π 是 TI-86 的常量。

CHAR MISC（其他）菜单 2nd [CHAR] F1

MISC	GREEK	INTL		
?	#	&	%	'

▶

!	@	\$	~	
---	---	----	---	--

▶

¿	Ñ	ñ	Ç	ç
---	---	---	---	---

CHAR GREEK 菜单 2nd [CHAR] F2

MISC	GREEK	INTL		
α	β	γ	Δ	δ

▶

ϵ	θ	λ	μ	ρ
------------	----------	-----------	-------	--------

▶

Σ	σ	τ	ϕ	Ω
----------	----------	--------	--------	----------

CHAR INTL（国际）菜单 2nd [CHAR] F3

MISC	GREEK	INTL		
´	`	^	¨	

可以通过合并 CHAR INTL 菜单中的修饰符与大写或小写元音来创建在一些语言中使用的元音。
可以在变量名和文本中使用这些元音。

添加修饰符到元音

- 1

从 CHAR INTL 菜单中选择修饰符。大写字母锁开启。必要时切换到小写字母锁。

2nd [CHAR] F3 F4
2nd [alpha]
- 2

输入要添加修饰符的大写或小写元音。

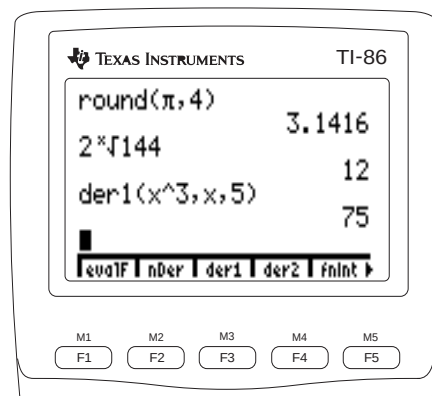
[O]



3

数学、微积分 和测试操作

键盘数学函数.....	48
MATH 菜单.....	49
CALC（微积分）菜单.....	54
TEST（关系）菜单.....	55



“A 到 Z 参考”详细描述了对于每个函数哪些数据类型是有效的参数。

最常用的数学函数位于 TI-86 键盘上。关于这些函数的语法、详细信息、和示例，请参阅“A 到 Z 参考”。

x^{-1} （乘法的逆元）等价于倒数 $1/x$ 。

键盘数学函数

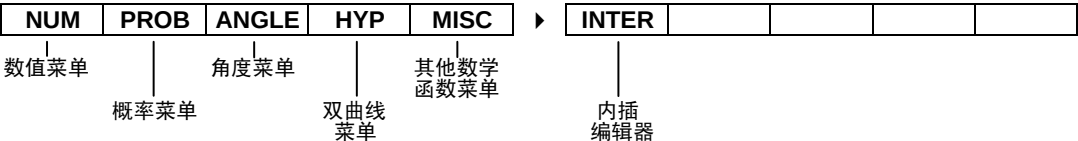
可以在实数或复数值表达式中使用数学函数。可以在一些函数中使用数组、向量、矩阵或字符串。

当使用数组、向量或矩阵时，有效的函数返回通过逐个元素计算得到的结果数组。如果在同一个表达式中使用两个数组、向量或矩阵，它们的维数必须相同。

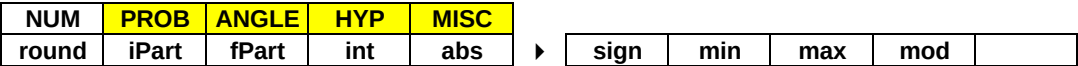
键	函数
[+]	+ （加）
[-]	- （减）
[×]	* （乘）
[÷]	/ （除）
[(-)]	- （取负）
[x²]	² （平方）
[2nd][√]	√ （平方根）
[2nd][x⁻¹]	⁻¹ （求逆）
[^]	^ （幂）
[2nd][10ˣ]	10^ （10 的幂）
[EE]	E （指数）

键	函数
[SIN]	sin （正弦）
[COS]	cos （余弦）
[TAN]	tan （正切）
[2nd][SIN⁻¹]	sin⁻¹ （反正弦；正弦的反函数）
[2nd][COS⁻¹]	cos⁻¹ （反余弦；余弦的反函数）
[2nd][TAN⁻¹]	tan⁻¹ （反正切；正切的反函数）
[LOG]	log （对数）
[LN]	ln （自然对数）
[2nd][eˣ]	eˣ （常数 e 的幂）
[2nd][π]	π （常数 pi； 3.1415926535898）

MATH 菜单 [2nd] [MATH]



MATH NUM (数值) 菜单 [2nd] [MATH] [F1]



数值可以是表达式、数组、向量或矩阵。有关具体语法选项和示例的详细信息，请参阅“A到Z参考”。

- round***(value[,#ofDecimals])*

将 *value* 四舍五入到 12 位小数或 *#ofDecimals* 位小数
- iPart** *value*

返回 *value* 的整数部分
- fPart** *value*

返回 *value* 的小数部分
- int** *value*

返回小于或等于 *value* 的最大整数
- abs** *value*

返回 *value* 的绝对值或模
- sign** *value*

如果 *value* 为正，返回 **1**；如果 *value* 为 **0**，返回 **0**；如果 *value* 为负，返回 **-1**
- min***(valueA,valueB)*

返回 *valueA* 和 *valueB* 中较小的一个
- min***(list)*

返回数组 *list* 中的最小元素
- max***(valueA,valueB)*

返回 *valueA* 和 *valueB* 中较大的一个
- max***(list)*

返回数组 *list* 中的最大元素
- mod***(numberA,numberB)*

返回 *numberA* 对 *numberB* 的余数

MATH PROB (概率) 菜单 [2nd] [MATH] [F2]

NUM	PROB	ANGLE	HYP	MISC					
!	nPr	nCr	rand	randIn	▶	randN	randBi		

! (阶乘) 对非整数有效。

value!

返回实数 *value* 的阶乘

items **nPr** *number*

返回从 *items* (**n**) 取 *number* (**r**) 的排列数

items **nCr** *number*

返回从 *items* (**n**) 取 *number* (**r**) 的组合数

rand

返回一个 > 0 且 < 1 的随机数；要控制随机数序列，先将一个整数起始值存储到 **rand**（例如 **0>rand**）

randInt、**randNorm** 和 **randBin** 在 MATH PROB 菜单中被缩写。

randInt(*lower*,*upper*
[,*#ofTrials*])

（随机整数）返回一个随机整数 *integer*，满足：
 $lower \leq integer \leq upper$ ；要返回一个随机整数数组，则为 *#ofTrials* 指定一个 > 1 的整数

randNorm(*mean*,
stdDeviation
[,*#ofTrials*])

（随机正态）返回一个随机实数，它服从由 *mean* 和 *stdDeviation* 指定的正态分布。要返回随机数数组，则为 *#ofTrials* 指定一个 > 1 的整数

randBin(*#ofTrials*,
probabilityOfSuccess
[,*#ofSimulations*])

（随机二项式）返回一个随机实数，它服从二项式分布，其中 $\#ofTrials \geq 1$ ， $0 \leq probabilityOfSuccess \leq 1$ ；要返回随机数数组，则为 *#ofSimulations* 指定一个 > 1 的整数

MATH ANGLE 菜单 [2nd] [MATH] [F3]

NUM	PROB	ANGLE	HYP	MISC
0	r	'	►DMS	

角度可以是°、' 和►DMS 的数组。

在计算中，输入项 *degrees'minutes'seconds'* 的结果只在 **Degree** 角度方式认为是度，在 **Radian** 角度方式认为是弧度。

<i>angle</i> °	覆盖当前角度方式设置用度表示 <i>angle</i>
<i>angle</i> ^r	覆盖当前角度方式设置用弧度表示 <i>angle</i>
<i>degrees'minutes'seconds'</i>	用度 <i>degrees</i> 、分 <i>minutes</i> 和秒 <i>seconds</i> 来指定角度
<i>angle</i> ►DMS	以度°分'秒"格式显示角度 <i>angle</i> ，即使用 <i>degrees'minutes'seconds'</i> 输入 DMS 角度

MATH HYP（双曲）菜单 [2nd] [MATH] [F4]

NUM	PROB	ANGLE	HYP	MISC					
sinh	cosh	tanh	sinh ⁻¹	cosh ⁻¹	►	tanh ⁻¹			

数值可以是表达式、数组、向量或矩阵。有关具体的语法选项和示例的详细信息，请参阅“**A 到 Z 参考**”。

<i>sinh value</i>	返回 <i>value</i> 的双曲正弦值
<i>cosh value</i>	返回 <i>value</i> 的双曲余弦值
<i>tanh value</i>	返回 <i>value</i> 的双曲正切值
<i>sinh⁻¹ value</i>	返回 <i>value</i> 的双曲反正弦值
<i>cosh⁻¹ value</i>	返回 <i>value</i> 的双曲反余弦值
<i>tanh⁻¹ value</i>	返回 <i>value</i> 的双曲反正切值

数值可以是表达式、数组、向量或矩阵。有关具体的语法选项和示例的详细信息，请参阅“ A 到 Z 参考”。

MATH MISC（其他）菜单 [2nd] [MATH] [F5]

NUM	PROB	ANGLE	HYP	MISC						
sum	prod	seq	lcm	gcd	►	►Frac	%	pEval	$x\sqrt{}$	eval

sum <i>list</i>	返回数组 <i>list</i> 元素的和
prod <i>list</i>	返回数组 <i>list</i> 元素的乘积
seq (<i>expression,variable,begin,end[,step]</i>)	返回一个数组，其中每个元素为表达式 <i>expression</i> 对变量 <i>variable</i> 的求值，从起始值 <i>begin</i> 到终值 <i>end</i> ，步长为 <i>step</i>
lcm (<i>valueA,valueB</i>)	返回 <i>valueA</i> 和 <i>valueB</i> 的最小公倍数
gcd (<i>valueA,valueB</i>)	返回 <i>valueA</i> 和 <i>valueB</i> 的最大公分母
<i>value</i> ► Frac	将 <i>value</i> 显示为一个分数
<i>value</i> %	返回 <i>value</i> 除以 100（乘以 0.01）的结果
<i>percent</i> % <i>number</i>	返回数 <i>number</i> 的百分比 <i>percent</i>
pEval (<i>coefficientList,xValue</i>)	返回多项式（系数在 <i>coefficientList</i> 中给出）在 <i>xValue</i> 处的值
<i>xthroot</i> $x\sqrt{}$ <i>value</i>	返回数值 <i>value</i> 的 <i>x</i> 次根 <i>xthroot</i>
eval <i>value</i>	返回一个数组，元素为当前图形方式下所有选定函数在自变量为实数 <i>value</i> 时的值

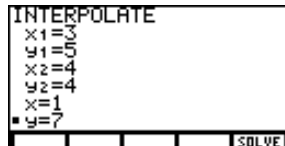
内插/外推编辑器 $\boxed{2\text{nd}} \boxed{\text{MATH}} \boxed{\text{MORE}} \boxed{\text{F1}}$

使用内插/外推编辑器，在给定两个已知数据对和未知数据对的 x 值或 y 值时，可以线性内插或外推获得未知数据对的另一个值。

要在主屏幕内插 y ，则从 CATALOG 中选择 **inter**，然后输入 **inter**($x1,y1,x2,y2,x$)。

要在主屏幕内插 x ，则输入 **inter**($y1,x1,y2,x2,y$)。

- ① 显示内插/外推编辑器。 $\boxed{2\text{nd}} \boxed{\text{MATH}} \boxed{\text{MORE}} \boxed{\text{F1}}$
- ② 输入第一个已知数据对 ($x1,y1$) 的实数值。 3 $\boxed{\text{ENTER}}$ 5 $\boxed{\text{ENTER}}$
值可以是表达式。
- ③ 输入第二个已知数据对 ($x2,y2$) 的值。 4 $\boxed{\text{ENTER}}$ 4 $\boxed{\text{ENTER}}$
- ④ 输入未知数据对的 x 值或 y 值。 1 $\boxed{\text{ENTER}}$
- ⑤ 必要时将光标移到要求解的值 (x 或 y) 处。 $\boxed{\uparrow}$ 或 $\boxed{\downarrow}$
- ⑥ 选择 **SOLVE**。 $\boxed{\text{F5}}$



可以用 $\boxed{\text{STO}} \blacktriangleright$ 键存储个别值 (第2章)。

内插或外推结果并显示；变量 x 和 y 不变。第1列中的实方块表示内插或外推值。

当求解一个值之后，可以继续使用内插/外推编辑器。

要使用微积分函数，必须设置 Dec 方式。

对于 evalF、nDer、der1 和 der2，变量 variable 可以是实数或复数或数组。
在表达式中可以使用 der1 和 der2。只能在表达式中使用 nDer。

对于 fnInt、fMin 和 fMax，必须保证 lower < upper。

CALC（微积分）菜单

2nd [CALC]

evalF	nDer	der1	der2	fnInt	▶	fMin	fMax	arc		
-------	------	------	------	-------	---	------	------	-----	--	--

微积分函数返回关于任一用户创建变量、内置变量 eqn 和 exp 以及图象变量（例如 x、t 和 θ）的值。

evalF(expression,variable,value)	返回表达式 <i>expression</i> 在变量 <i>variable</i> 为 <i>value</i> 时的值
nDer(expression,variable [,value])	返回表达式 <i>expression</i> 在变量 <i>variable</i> 为当前值或指定 <i>value</i> 值时的近似数值导数
der1(expression,variable[,value])	返回表达式 <i>expression</i> 在变量 <i>variable</i> 为当前值或指定值 <i>value</i> 时的一阶导数
der2(expression,variable[,value])	返回表达式 <i>expression</i> 在变量 <i>variable</i> 为当前值或指定值 <i>value</i> 时的二阶导数
fnInt(expression,variable, lower,upper)	返回表达式 <i>expression</i> 在变量 <i>variable</i> 的上下界为 <i>lower</i> 和 <i>upper</i> 时的数值积分
fMin(expression,variable, lower,upper)	返回表达式 <i>expression</i> 在变量 <i>variable</i> 的上下界为 <i>lower</i> 和 <i>upper</i> 时的最小值
fMax(expression,variable, lower,upper)	返回表达式 <i>expression</i> 在变量 <i>variable</i> 的上下界为 <i>lower</i> 和 <i>upper</i> 时的最大值
arc(expression,variable, start,end)	返回表达式 <i>expression</i> 在变量 <i>variable</i> 处于 <i>start</i> 和 <i>end</i> 之间时所定义的曲线长度

内置变量 δ 定义计算 `nDer`(只在 `dxNDer` 微分方式下) 和 `arc`(时的步长。内置变量 `tol` 定义计算 `fnInt`(、`fMin`(、`fMax`(和 `arc`(时的误差。它们必须 >0 。这些因素影响计算精度。 δ 越小, 近似一般越精确。例如, 如果 $\delta=0.01$, `nDer(A^3,A,5)` 返回 **75.0001** ; 而如果 $\delta=0.0001$, 则返回 **75** (参阅附录) 。

函数积分误差值保存到变量 `fnIntErr` (参阅附录) 。

在 `dxDer1` 方式时, 对于 `arc`(和 `fnInt`(, 以下函数在表达式 *expression* 中无效:
`evalF`(、`der1`(、`der2`(、`fMin`(、`fMax`(、`nDer`(、`seq`(和任何方程变量 (如 `y1`) 。

可以用下面的公式近似求当前值处的四阶导数:
`nDer(nDer(der2(x^4,x),x),x)`。

TEST (关系) 菜单 2nd [TEST]

<code>==</code>	<code><</code>	<code>></code>	<code>≤</code>	<code>≥</code>	▶	<code>≠</code>				
-----------------	-------------------	-------------------	----------------	----------------	---	----------------	--	--	--	--

关系函数对两个长度相同的数组是有效的。当 *valueA* 和 *valueB* 为数组时, 就返回一个数组, 其元素值是逐个元素计算得到的。

- `valueA==valueB`

(等于) 如果 *valueA* 等于 *valueB* , 返回 **1** ; 否则返回 **0** ; *valueA* 和 *valueB* 可以是实数或复数、数组、向量、矩阵或字符串
- `valueA<valueB`

(小于) 如果 *valueA* 小于 *valueB* , 返回 **1** ; 否则返回 **0** ; *valueA* 和 *valueB* 必须是实数或数组
- `valueA>valueB`

(大于) 如果 *valueA* 大于 *valueB* , 返回 **1** ; 否则返回 **0** ; *valueA* 和 *valueB* 必须是实数或数组

$valueA \leq valueB$ （小于或等于）如果 $valueA$ 小于或等于 $valueB$ ，返回 **1**；否则返回 **0**； $valueA$ 和 $valueB$ 必须是实数或数组

$valueA \geq valueB$ （大于或等于）如果 $valueA$ 大于或等于 $valueB$ ，返回 **1**；否则返回 **0**； $valueA$ 和 $valueB$ 必须是实数或数组

$valueA \neq valueB$ （不等于）如果 $valueA$ 不等于 $valueB$ ，返回 **1**；否则返回 **0**； $valueA$ 和 $valueB$ 可以是实数或复数、数组、向量、矩阵或字符串

在表达式和指令中使用测试

TI-86 Evaluation Operating System（参阅附录）在执行关系函数之前执行除布尔操作符以外的所有操作。例如：

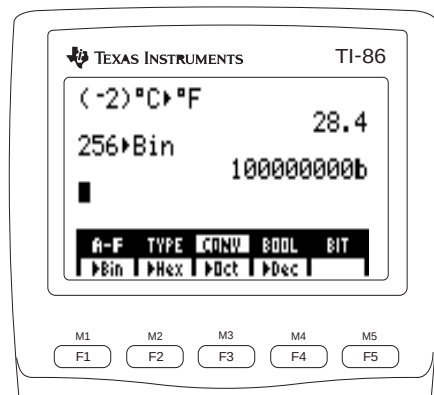
- ◆ 表达式 $2+2==2+3$ 计算为 **0**。TI-86 先执行加法，然后比较 4 和 5。
- ◆ 表达式 $2+(2==2)+3$ 计算为 **6**。TI-86 先执行括号中的测试，然后执行 2 加 1 加 3。

可以使用关系函数来控制程序流向（第 16 章）。

4

常量、换算、 数制和复数

使用内置和用户创建常量	58
换算度量单位	61
数制	65
使用复数	70

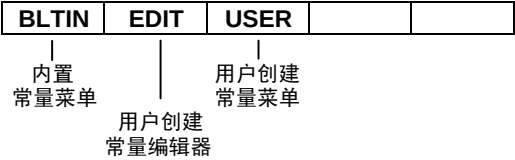


使用内置和用户创建常量

常量是存储了特定值的变量。CONS BLTIN 菜单项是 TI-86 内置的公用常量。不能编辑内置常量的值。

为便于读取，可以创建用户常量并将它们增加到用户创建常量菜单以便于访问。要输入用户创建常量，必须使用用户创建常量编辑器（第 60 页）；不能用 **STO▶** 或 **=** 来创建常量。

CONS（常量）菜单 **2nd** [CONS]



可以从 CONS BLTIN 菜单选择内置常量，或使用键盘和 CHAR GREEK 菜单输入它们。

CONS BLTIN（内置常量）菜单 **2nd** [CONS] **F1**

BLTIN	EDIT	USER			▶	Gc	g	Me	Mp	Mn
Na	k	Cc	ec	Rc	▶	μ0	ε0	h	c	u

内置常量	常量名	常量值
Na	阿伏伽德罗常数	6.0221367E23 摩尔 ⁻¹
k	Boltzman 常量	1.380658E-23 J/K
Cc	库仑常量	8.9875517873682E9 N m ² /C ²
ec	电子电荷	1.60217733E-19 C
Rc	气体常量	8.31451 J/摩尔 K
Gc	重力常量	6.67259E-11 N m ² /kg ²
g	地球重力加速度	9.80665 m/s ²
Me	电子质量	9.1093897E-31 kg
Mp	质子质量	1.6726231E-27 kg
Mn	中子质量	1.6749286E-27 kg
μ0	真空渗透性	1.2566370614359E-6 N/A ²
ε0	真空介质常数	8.8541878176204E-12 F/m
h	普朗克常数	6.6260755E-34 J s
c	真空光速	299,792,458 m/s
u	原子质量单位	1.6605402E-27 kg
π	Pi	3.1415926535898
e	自然对数底	2.718281828459

要使用 π，按 [2nd] [π] 或从 CATALOG 选择。

要使用 e^，按 [2nd] [e^x]。

要使用 e，按 [2nd] [alpha] [E]。

CONS USER 菜单项是按字母顺序排列全部已经存储的用户创建常量名。

196.9665 是金 (Au) 的原子重量。

可以稍后输入一个数值。

如果在第一个常量名被显示时选择 **PREV**，或在最后一个常量名显示时选择 **NEXT**，CONS USER 菜单将替换 CONS EDIT 菜单。

也可以从 MEM DELET CONS 屏幕删除常量。

创建或重新定义用户创建常量

- 1 显示 CONS 菜单。 [2nd] [CONS]
- 2 显示常量编辑器。Name= 提示、Value= 提示和 CONS USER 菜单显示出来。大写字母锁开启。 [F2]
- 3 输入常量名。可以输入一个以字母开头、长度为 1 到 8 个字符的新名字或从 CONS USER 菜单中选择一个名字。光标移到 **Value=** 提示处并显示 CONS EDIT 菜单（请参见下面）。 [A] [2nd] [alpha]
[U] [ENTER]
- 4 输入实数或复数常量值，可以是表达式。输入该值后，它存储到该常量。该用户创建的常量成为一个 CONS USER 菜单项。 196 [.] 9665

RTN	EDIT	USER		
-----	------	------	--	--

CONSTANT
Name=
Value=

PREV	NEXT	DELET		
------	------	-------	--	--

CONS EDIT（常量编辑器）菜单 [2nd] [CONS] [F2] 常量名 [ENTER] 或 ▼

PREV	NEXT	DELET		
------	------	-------	--	--

PREV 显示 CONS USER 菜单中前一个常量名和值（如果有）

NEXT 显示 CONS USER 菜单中下一个常量名和值（如果有）

DELET 删除常量编辑器中当前显示的常量名和值

在表达式中输入常量名

可以用下面三种方法之一在表达式中输入常量。

- ◆ 从 CONS BLTIN 菜单或 CONS USER 菜单选择常量名。
- ◆ 从 VARS CONS 屏幕中选择用户创建常量名。
- ◆ 用大写字母键、小写字母键和其他字符键来输入常量名。

换算度量单位

可以在任何表达式有效的地方
输入换算表达式。

借助 TI-86，可以将用一种度量单位表示的值换算为另一种度量单位表示的等价值。例如，可以将英寸换算为码，将夸脱换算为公升，或将华氏温度换算为摄氏温度。

要相互换算的度量单位必须兼容。例如，不能将英寸换算成华氏温度，或将码换算成卡路里。CONV 菜单上的每个菜单项（第 62 页）代表一组度量单位，比如长度 (LNGTH)、体积 (VOL)、和压强 (PRESS)。在每个菜单项内，所有单位都兼容。

换算度量单位

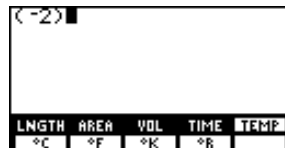
使用换算指令的语法为：

(value)currentUnit▶newUnit

在示例中，-2 摄氏度被换算成
华氏度。当数值为负时，一般
使用括号。

- ① 输入要换算的实数值。
- ② 显示 CONV 菜单。
- ③ 选择 TEMP 换算组。

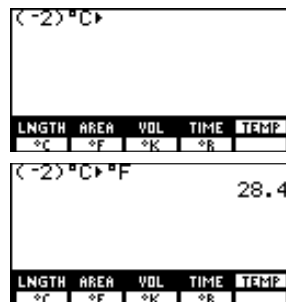
[1] [(-)] 2 [)]
[2nd] [CONV]
[F5]



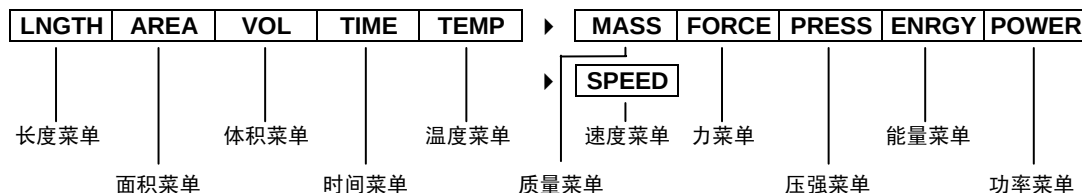
- ④ 从换算组菜单选择当前的度量单位 (°C)。单位缩写和换算符 (►) 粘贴到光标位置。 [F1]

- ⑤ 从换算组菜单选择新度量单位 (°F)。单位缩写粘贴到光标位置。 [F2]

- ⑥ 换算度量单位。 [ENTER]



CONV (换算) 菜单 [2nd] [CONV]



强调：当换算负值时，必须用括号将值和负号括起来，比如 **(-4)**。否则，TI-86 计算顺序会先执行换算，然后对换算结果执行求负运算。

如果输入... | ...TI-86 换算结果...

(-4)°C►°F	24.8 华氏度 (-4° 摄氏度被换算成华氏度)
-4°C►°F	-39.2 华氏度 (4° 摄氏度被换算成华氏度，然后取负)

CONV LENGTH（长度）菜单 [2nd] [CONV] [F1]

mm 毫米
cm 厘米
m 米
in 英寸
ft 英尺

yd 码
km 千米
mile 英里
nmile 海里
lt-yr 光年

mil 毫寸
Ang 埃
fermi 费尔米
rod 杆
fath 英寻

CONV AREA（面积）菜单 [2nd] [CONV] [F2]

ft² 平方英尺
m² 平方米
mi² 平方英里

km² 平方千米
acre 英亩
in² 平方英寸

cm² 平方厘米
yd² 平方码
ha 公顷

CONV VOL（体积）菜单 [2nd] [CONV] [F3]

liter 公升
gal 加仑
qt 夸脱
pt 品脱
oz 盎司

cm³ 立方厘米
in³ 立方英寸
ft³ 立方英尺
m³ 立方米
cup 杯

tsp 茶匙
tblsp 大汤匙
ml 毫升
galUK 加仑（英）
ozUK 盎司（英）

CONV TIME（时间）菜单 [2nd] [CONV] [F4]

sec 秒
mn 分
hr 小时

day 天
yr 年
week 周

ms 毫秒
μs 微秒
ns 纳秒

CONV TEMP（温度）菜单 [2nd] [CONV] [F5]

°C 摄氏度

°F 华氏度

°K 绝对温度

°R Rankin 度

CONV MASS (质量) 菜单 [2nd] [CONV] [MORE] [F1]**gm** 克**amu** 原子量单位**ton** 吨**kg** 千克**slug** 片**mton** 公吨**lb** 磅**CONV FORCE** (力) 菜单 [2nd] [CONV] [MORE] [F2]**N** 牛顿**tonf** 吨力**lbf** 磅力**dyne** 达因**kgf** 千克力**CONV PRESS** (压强) 菜单 [2nd] [CONV] [MORE] [F3]**atm** 大气压**lb/in²** 磅/平方英寸**inHg** 英寸汞柱**bar** 巴**mmHg** 毫米汞柱**inH₂O** 英寸水柱**N/m²** 牛顿/平方米**mmH₂** 毫米水柱**CONV ENRGY** (能量) 菜单 [2nd] [CONV] [MORE] [F4]**J** 焦耳**ft-lb** 尺磅**erg** 尔格**cal** 卡路里**kw-hr** 千瓦小时**l-atm** 公升大气压**Btu** 英制热量单位**eV** 电子伏特**CONV POWER** (功率) 菜单 [2nd] [CONV] [MORE] [F5]**hp** 马力**ftlb/s** 尺磅/秒**Btu/m** 英制热量单位/分**W** 瓦特**cal/s** 卡路里/秒**CONV SPEED** (速度) 菜单 [2nd] [CONV] [MORE] [MORE] [F1]**ft/s** 英尺/秒**mi/hr** 英里/小时**knot** 哩**m/s** 米/秒**km/hr** 公里/小时

要输入斜杠 (/)，可以使用
⏏ 键或从 CATALOG 粘贴。

换算以比率表示的值

要在主屏幕上换算以比率表示的值，可以使用括号和除法操作符 (/)。例如，如果汽车 4 小时走了 325 英里，想知道以公里/小时表示的速度，则输入表达式：

(325/4)mi/hr→km/hr 该表达式返回 **131** 公里/小时（四舍五入后）。

也可以只用一个斜杠返回此结果，比如：**325mile→km/4hr→hr**

数制

数制方式设置（第 1 章）控制 TI-86 如何解释输入的数值和如何在主屏幕上显示结果。然而，可以用数制标志符（**b**、**o**、**d** 和 **h**）输入以任意数制表示的数值。然后可以用数制转换在主屏幕上显示以任意数制表示的结果。

所有数值在内部都是以十进制形式存储的。如果在 **Dec** 以外的方式设置下执行操作，TI-86 通过将每次计算和表达式求解结果截断为整数来执行整数运算。

例如，**Hex** 方式下，**1/3+7** 返回 **7h**（1 除以 3，结果截断为 0，然后加 7）。

数制范围

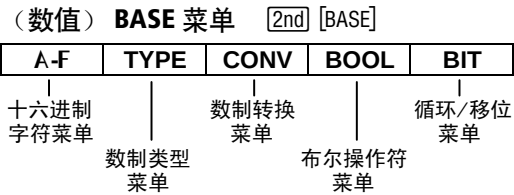
TI-86 的二进制、八进制和十六进制数的范围如下。

类型	最低值/最高值	等价十进制数
二进制	1000 0000 0000 0001 b	-32,767
	0111 1111 1111 1111 b	32,767
八进制	5120 6357 4134 0001 o	-99,999,999,999,999
	2657 1420 3643 7777 o	99,999,999,999,999
十六进制	FFFF A50C EF85 C001 h	-99,999,999,999,999
	0000 5AF3 107A 3FFF h	99,999,999,999,999

反码和补码

要得到一个二进制数的反码，在二进制数前输入 **not** 函数（第 68 页）。例如，在 **Bin** 方式下 **not 111100001111** 返回 **1111000011110000b**。

要得到二进制数的补码，在所输入数前按 。例如，在 **Bin** 方式下 **-111100001111** 返回 **1111000011110001b**。



BASE A-F 菜单项和 BASE TYPE 菜单项和一般的字母字符不同。

在示例中，上面菜单是数组编辑器菜单（**2nd** **[LIST]** 在 Dec 数制表示方式下）。

如果 Hex 数制方式未被设置，必须输入 h 标志符，即使该数包含一个十六进制特殊字符。

BASE A-F（十六进制字符）菜单 **2nd** **[BASE]** **[F1]**
这是显示在主屏幕上的 BASE A-F 菜单。要使用 A，按 **2nd** **[M]**。

A	TYPE	CONV	BOOL	BIT
B	C	D	E	F

当编辑器菜单成为上面菜单时，A 和 B 组合在一个单元中。如果按 **[F1]** 或 **[MORE]**...

{	}	NAMES	"	OPS
A-B	C	D	E	F

...A 和 B 移到两个分开的单元，E 和 F 一样组合。要切换回来，按 **[F5]** 或 **[MORE]**。

{	}	NAMES	"	OPS
A	B	C	D	E-F

输入十六进制数字

要输入十六进制数，就如十进制数那样使用数字键。按需要从菜单中选择从 A 到 F 的十六进制字符。

BASE TYPE（类型）菜单 **2nd** **[BASE]** **[F2]**

A-F	TYPE	CONV	BOOL	BIT
b	h	o	d	

在表达式中，可以用任意数制表示数，而不必考虑方式。输入数之后，从 BASE TYPE 菜单中选择对应的数制符号。数制符号粘贴到光标处。下面为几个例子。

在 Dec 方式下（默认）：

10b+10 **[ENTER]**

12

10h+10 **[ENTER]**

26

在 Bin 方式下：

10h+10 **[ENTER]**

10010b

10d+10 **[ENTER]**

1100b

在 Oct 方式下：

10b+10 **[ENTER]**

12o

10d+10 **[ENTER]**

22o

在 Hex 方式下：

10b+10 **[ENTER]**

12h

10d+10 **[ENTER]**

1Ah

BASE CONV（换算）菜单 [2nd] [BASE] [F3]

A-F	TYPE	CONV	BOOL	BIT
►Bin	►Hex	►Oct	►Dec	

数值可以是表达式、数组、向量或矩阵。关于详细的语法描述，请参阅“A 到 Z 参考”。

value►Bin

以二进制显示数值

value►Hex

以十六进制显示数值

value►Oct

以八进制显示数值

value►Dec

以十进制显示数值

换算数制

- ❶ 在 Dec 方式下，求解 10b + Fh + 10o + 10。

❷ 结果加 1 并换算成 Bin 数制显示。

❸ 结果加 1 并换算成 Hex 数制显示。

❹ 结果加 1 并换算成 Oct 数制显示。

❺ 结果加 1 并换算成 Dec 数制显示。

10b+Fh+10o+10 [ENTER] 35

Ans+1►Bin [ENTER] 100100b

Ans+1►Hex [ENTER] 25h

Ans+1►Oct [ENTER] 46o

Ans+1 [ENTER] 39
- BASE BOOL（布尔）菜单 [2nd] [BASE] [F4]
- | A-F | TYPE | CONV | BOOL | BIT |
|-----|------|------|------|-----|
| and | or | xor | not | |
- valueA* and *valueB*

valueA or *valueB*

valueA xor *valueB*

not *value*

参数和结果必须在定义的数值范围内（第 66 页）。

布尔运算结果

对布尔表达式求值时，先将参数转换为十六进制整数，然后比较两个参数的对应位，如表格所示。

如果 <i>valueA</i> 等于...	... <i>valueB</i> 等于...	结果			
		and	or	xor	not (<i>valueA</i>)
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1

结果根据当前方式设置显示。例如：

- ◆ 在 Bin 方式下， 101 and 110 返回 100b。
- ◆ 在 Hex 方式下， 5 and 6 返回 4h。

循环和移位操作使用以 16 为底的数字。为使溢出错误减少到最少，请用二进制形式输入参数。

BASE BIT （位）菜单 [2nd] [BASE] [F5]

A-F	TYPE	CONV	BOOL	BIT
rotR	rotL	shftR	shftL	

- rotR *value* 循环右移数值 *value*
- rotL *value* 循环左移数值 *value*
- shftR *value* 右移数值 *value*
- shftL *value* 左移数值 *value*

使用复数

存储复数的变量名被列举在 VARS CPLX 屏幕上（第2章）。

复数可以是数组、矩阵和向量的元素。

复数有两个分量：实数分量 (a) 和虚数分量 ($+bi$)。在 TI-86 上，可以这样输入复数 $a+bi$ ：

- ◆ **(real,imaginary)** 直角坐标格式
- ◆ **(magnitude $\angle$ angle)** 极坐标格式

可以以直角坐标或极坐标格式输入复数，而不必考虑当前的复数方式设置。分隔符 (, 或 $\angle$) 决定其格式。

- ◆ 要输入直角坐标格式，用逗号 ($\square$) 分隔实数 *real* 和虚数 *imaginary* 分量。
- ◆ 要输入极坐标格式，用角度符号 (2^{nd} $\square$) 分隔模 *magnitude* 和角度 *angle* 分量。

每个分量（实数、虚数、模或角度）可以是实数或结果为实数的表达式；当按 $\square$ 时对表达式求值。

当 **RectC** 复数方式被设置时，以直角坐标格式显示复数，而不考虑其输入时格式（如右侧所示）。

```
(6,1)          (6,1)
(6 $\angle$ 1)
(3.24181383521,5.048...
```

当 **PolarC** 复数方式被设置时，以极坐标系格式显示复数，而不考虑其输入时格式（如右侧所示）。

```
(6,1)          (6,1)
(6.0827625303 $\angle$ .16514...
(6 $\angle$ 1)          (6 $\angle$ 1)
```

复数结果

结果中的复数，包括数组、矩阵和向量元素，以方式设置（第1章）或显示转换指令（第61页）指定的格式（直角坐标或极坐标）显示。

- ◆ 当 **Radian** 角度方式设置时，结果显示为 **(magnitude $\angle$ angle)**。
- ◆ 当 **Degree** 角度方式设置时，结果显示为 **(real,imaginary)**。

图象格式设置 **RectGC** 和 **PolarGC**（第5章）确定图象屏幕坐标的复数格式。

例如，当 **PolarC** 和 **Degree** 方式设置时，**(2,1)-(1∠45)** 返回 **(1.32565429614∠12.7643896828)**。

表达式中使用复数

- ◆ 直接输入复数。
- ◆ 使用大写字母键、小写字母键和其他字符键输入复数。
- ◆ 从 VARS CPLX 屏幕选择复数。

CPLX（复数）菜单 2nd [CPLX]

conj	real	imag	abs	angle	▶	▶Rec	▶Pol			
conj (real,imaginary)		返回复数、数组、向量或矩阵的共轭复数；结果为 (real,-imaginary)								
conj (magnitude∠angle)		返回 (magnitude∠-angle)								
real (real,imaginary)		返回复数、数组、向量或矩阵的实部；结果为实数 real								
real (magnitude∠angle)		返回模与角度余弦的积 (magnitude*cosine(angle))								
imag (real,imaginary)		返回复数、数组、向量或矩阵的虚部；结果为 imaginary								
imag (magnitude∠angle)		返回模与角度正弦的积 (magnitude*sine(angle))								
abs (real,imaginary)		(绝对值) 返回复数、数组、向量或矩阵的模；结果为 $\sqrt{(real^2+imaginary^2)}$								
abs (magnitude∠angle)		返回模 magnitude								

可以输入复数名或者数组、向量或矩阵作为任何CPLX 菜单项的参数。

angle (real,imaginary)	返回 $\tan^{-1}(\text{imaginary}/\text{real})$ （在第二象限和第三象限分别调整 π 和 $-\pi$ ）计算的复数、数组、向量或矩阵的极坐标角度；结果为 $\tan^{-1}(\text{imaginary}/\text{real})$
angle (magnitude∠angle)	返回角度 <i>angle</i> （其中 $-\pi < \text{angle} \leq \pi$ ）
complexValue►Rec	以直角坐标格式 (<i>real,imaginary</i>) 显示 <i>complexValue</i> ，而不考虑复数方式设置；它只在命令的尾部且只当 <i>complexValue</i> 为复数时有效
complexValue►Pol	以极坐标格式 (<i>magnitude∠angle</i>) 显示 <i>complexValue</i> ，而不考虑复数方式设置；它只在命令的尾部且只当 <i>complexValue</i> 为复数时有效

从 LIST 菜单中选择 { 和 }。

必须输入逗号来分隔数组元素。

可以直接输入复数数组、向量或矩阵。下面是输入数组的语法。要输入复数向量或矩阵，用大括号替换下面的括号并使用正确的数据类型格式（第 12 章和第 13 章）。

在直角坐标格式中，对复数数组使用 **conj**、**real**、**imag**、**abs** 和 **angle** 的语法为：

conj{(*realA,imaginaryA*),(*realB,imaginaryB*),(*realC,imaginaryC*),...}

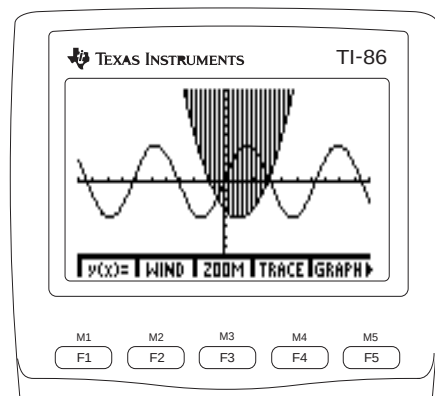
在极坐标格式中，对复数数组使用 **conj**、**real**、**imag**、**abs** 和 **angle** 的语法为：

real{(*magnitudeA∠angleA*),(*magnitudeB∠angleB*),(*magnitudeC∠angleC*),...}

使用数组时，TI-86 逐个元素计算结果，然后返回数组，根据复数方式设置表示数组元素。

5 函数图象

定义图象	74
设置图象方式	74
GRAPH 菜单	75
使用方程编辑器	76
设置窗口变量	81
设置图象格式	83
显示图象	85



定义图象

本章描述在 **Func** 图象方式下绘制函数的过程, 而该过程对 TI-86 每种图象方式都是类似的。第 8 章、第 9 章和第 10 章描述极坐标方程、参数方程和微分方程的各自独特的方面。第 6 章描述各种图象工具, 许多工具在所有图象方式都能使用。

在每次定义图象时下面的步骤中有一些是不必要的。

- ❶ 设置图象方式 (第 74 页)。
- ❷ 在方程编辑器中定义、编辑或选择一个或多个函数 (第 76 页和第 77 页)。
- ❸ 选择每个函数的图象样式 (第 79 页)。
- ❹ 必要时取消选择统计图 (第 81 页)。
- ❺ 设置观察窗口变量 (第 81 页)。
- ❻ 选择图象格式设置 (第 83 页)。

设置图象方式

要显示方式屏幕, 按 **2nd** [MODE]。所有的默认方式设置, 包括 **Func** 图象方式, 亮显在右面的图形中。图象方式在第 5 行。

- ◆ **Func** (函数图象)
- ◆ **Pol** (极坐标图象; 第 8 章)
- ◆ **Param** (参数图象; 第 9 章)
- ◆ **DifEq** (微分方程图象; 第 10 章)

```
Normal Sci Eng
Float 012345678901
Radian Degree
RectC PolarC
Func Pol Param DifEq
Dec Bin Oct Hex
RectV CylV SphereV
dxDer1 dxNDer
```

每种图象方式都有唯一的方程编辑器。在输入函数之前必须选择图象方式和 **Dec** 数制方式。**TI-86** 在内存中保留所有存储到 **Func**、**Pol**、**Param** 和 **DifEq** 方程编辑器中的方程。每种方式还有唯一的图象格式设置和窗口变量。

统计图开/关状态、缩放因子、方式设置和误差应用于所有的图象方式；对图象方式的改变不影响它们。

这些方式设置影响绘制结果。

- ◆ **Radian** 或 **Degree** 角度方式影响对一些函数的解释。
- ◆ **dxDer1** 或 **dxNDer** 微分方式影响选定函数的绘制。

GRAPH 菜单 GRAPH

y(x)=	WIND	ZOOM	TRACE	GRAPH	▶	MATH	DRAW	FORMT	STGDB	RCGDB
						▶	EVAL	STPIC	RCPIC	

- y(x)=

显示方程编辑器；用此屏幕输入要绘制的函数
- WIND

显示窗口编辑器；用此编辑器改变图象屏幕的尺寸
- ZOOM

显示 GRAPH ZOOM 菜单；用这些项改变图象屏幕的尺寸
- TRACE

激活跟踪光标；用该光标跟踪指定函数的图象
- GRAPH

显示图象屏幕；绘制所有选定的函数和已打开的统计图
- MATH

显示 GRAPH MATH 菜单；用该菜单从数学角度浏览图象
- DRAW

显示 GRAPH DRAW 菜单；用该菜单在图象上绘图或测试象素

第 1 章详细描述所有的方式设置。

第 6 章描述下面这些 GRAPH 菜单项：
ZOOM、TRACE、MATH、DRAW、STGDB、RCGDB、EVAL、STPIC 和 RCPIC。

- FORMT** 显示图象格式屏幕；用此屏幕选择图象格式设置。
- STGDB** 显示 **Name=** 提示和 STGDB 菜单；用提示输入 **GDB** 变量
- RCGDB** 显示 **Name=** 提示和 RCGDB 菜单；用该菜单调用图象数据库
- EVAL** 显示 **Eval x=** 提示；为要求解的当前函数输入一个 **x** 值
- STPIC** 显示 **Name=** 提示和 STPIC 菜单；用该提示输入 **PIC** 变量
- RCPIC** 显示 **Name=** 提示和 RCPIC 菜单；用此菜单调用图形

使用方程编辑器

要在 **Func** 图象方式下显示方程编辑器，从 **GRAPH** 菜单 (**GRAPH** **F1**) 选择 **y(x)=**，**GRAPH** 菜单上移，方程编辑器菜单显示为下面菜单。如果可用内存足够，可在方程编辑器中存储多达 99 个函数。



如果函数被选择，它的等号 (=) 在方程编辑器中亮显；如果它被取消选择，则它的等号不亮显。当 TI-86 绘制图象时只绘制所选择的函数。

方程编辑器 (**GRAPH y(x)=**) 菜单 **GRAPH** **F1**

y(x)=	WIND	ZOOM	TRACE	GRAPH					
x	y	INSf	DELf	SELCT	▶	ALL+	ALL-	STYLE	

x	粘贴变量 x 到当前光标处（与 $\boxed{\text{x-VAR}}$ 或 $\boxed{2\text{nd}} \boxed{[\alpha]} \boxed{[X]}$ 作用相同）
y	粘贴变量 y 到当前光标处（与 $\boxed{2\text{nd}} \boxed{[\alpha]} \boxed{[Y]}$ 作用相同）
INSf	在当前光标处的上方插入已删除的方程变量（函数）名（只插入变量名）
DELf	删除光标所在的函数
SELECT	改变光标所在函数的选择状态（选择或取消选择）
ALL+	选择方程编辑器中所有已定义的函数
ALL-	取消选择方程编辑器中所有已定义的函数
STYLE	将 7 个可用图象样式的下一个赋给光标所在的函数

在方程编辑器中定义函数

要从方程编辑器中的第 1 个函数移到最后一个，则按 $\boxed{\Delta}$ 。

要移到方程的开始处或结尾处，则按 $\boxed{2\text{nd}} \boxed{\leftarrow}$ 或 $\boxed{2\text{nd}} \boxed{\rightarrow}$ 。

省略号表示方程超出了屏幕。

- ① 显示方程编辑器。
 $\boxed{\text{GRAPH}} \boxed{\text{F1}}$
- ② 如果函数存储在方程编辑器中，则下移光标直到显示空白函数。
 $\boxed{\downarrow}$ 或 $\boxed{\text{ENTER}}$
- ③ 输入关于 **x** 的方程以定义函数。当输入第 1 个字符时，自动选择该函数。（函数的等号亮显。）
 $4 \boxed{\text{SIN}} \boxed{\text{x-VAR}}$
- ④ 移动光标到下一个函数。
 $\boxed{\text{ENTER}}$ 或 $\boxed{\downarrow}$



关于定义函数方程的说明

- ◆ 方程中可以包含函数、变量、常量、矩阵、矩阵元素、向量、向量元素、数组、数组元素、复数值或其他方程。
- ◆ 如果方程中包含矩阵、向量或负数，则方程在每点的计算结果必须是实数。
- ◆ 方程中可包含其他已定义的函数。例如给定 $y1=\sin x$ 和 $y2=4+y1$ ，函数 $y2$ 则等于 4 加上 x 的正弦。
- ◆ 要输入函数名，则从方程编辑器菜单中选择 **y**，然后输入相应的数字。
- ◆ 要插入方程变量的内容，则使用 RCL（第 1 章）。要在 Rcl 提示处输入方程变量，则使用大写字母键、小写字母键和其他字符键。
- ◆ 要从主屏幕或程序编辑器中选择所有的函数，则从 CATALOG 菜单选择 **FnOn**（或输入每个字符）并按 **ENTER**。
- ◆ 要从主屏幕或程序编辑器中选择指定函数，则从 CATALOG 菜单选择 **FnOn**（或输入每个字符），输入每个函数的编号，然后按 **ENTER**。例如，要选择 $y1$ 、 $y3$ 和 $y5$ ，输入 **FnOn 1、3、5**。
- ◆ 要从主屏幕或程序编辑器中取消选择函数，则使用 **FnOff**，方式与使用 **FnOn** 选择函数相同。
- ◆ 当函数的结果为非实数时，在图象上将不绘制该值；不返回错误。

可以编辑用 Rcl 插入的
表达式。

TI-86 在同一个图象屏幕上绘制所有选定函数。

选择图象样式

取决与所设置的图象方式，TI-86 能最多提供 7 个不同图象样式。可以将这些样式分配给指定的函数，从视觉上将它们区分开。

例如，可以将 **y1** 显示为一条连接线（\y1= 在方程编辑器中）而将 **y2** 显示为一条点画线（\y2= ），阴影 **y3**（\y3= ）上面的区域。

此外，可以控制样式图形化地显示实际现象，比如在空中飞行的球（用 $\odot$ ）或转轮上椅子的圆周运动（用 $\odot$ ）。

∇ （上阴影）和 $\blacktriangledown$ （下阴影）只在 Func 图象方式下可用。

$\cdots$ （点画线）在除 DifEq 图象方式之外的所有图象方式下可用。

图标 样式 绘制函数的特征

$\backslash$	实线	一条实线连结每个图形点；是 Connected 方式的默认选项
$\equiv$	粗线	一条粗实线连结每个图形点
∇	上阴影	阴影函数上方的区域
$\blacktriangledown$	下阴影	阴影函数下方的区域
$\odot$	轨迹	圆形光标跟踪图象的前沿并绘制出轨迹
$\circ$	动画	圆形光标跟踪图象的前沿但不绘制出轨迹
$\cdots$	点画线	用小点代表绘图点；在 Dot 方式下这是默认值

要在程序中设置图象样式，则从 CATALOG （A 到 Z 参考）中选择 **GrStl(**。

在示例中，y2 选择 （上阴影）。所有窗口变量设置为默认值（第82页）。

在方程编辑器中设置图象样式

- ❶ 显示方程编辑器。
- ❷ 将光标移到要设置图象样式的一个或多个函数处。
- ❸ 显示方程编辑器菜单项 **STYLE**。
- ❹ 重复选择 **STYLE** 将图象样式图标滚到方程名的左边。
- ❺ 查看新图象样式的图象。
- ❻ 清除 GRAPH 菜单只查看图象。

GRAPH F1



MORE

F3 F3

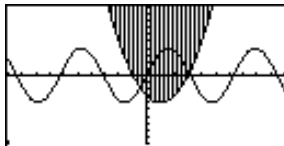
2nd F5

CLEAR

```
Plot1 Plot2 Plot3
~y1Bsin x
~y2Bx^2-2x-3
```

```
Plot1 Plot2 Plot3
~y1B4sin x
~y2Bx^2-2x-3
```

WIND ZOOM TRACE GRAPH
ALL+ ALL- STYLE



使用阴影图案区分函数

当为多个函数选择 （上阴影）或 （下阴影）时，TI-86 轮转四种阴影图案

- ◆ 第一个被阴影的函数：垂直线
- ◆ 第二个被阴影的函数：水平线
- ◆ 第三个被阴影的函数：反斜线
- ◆ 第四个被阴影的函数：正斜线

对于第五个被阴影的函数，循环又返回到垂直线，并重复此顺序。

如果指定  或  给绘制一族曲线的函数（第86页），则该族曲线的所有成员应用同一个旋转图案。

查看和改变统计图的开/关状态

在方程编辑器顶行的 **Plot1 Plot2 Plot3** 显示每个统计图的开/关状态（第 14 章）。当图形名在此行中亮显时，该图形处于打开状态。

要从方程编辑器中改变统计图的开/关状态，按 $\square$ 、 $\square$ 和 $\square$ 将光标放在 **Plot1**、**Plot2** 或 **Plot3** 上，然后按 **ENTER**。

设置窗口变量

图象屏幕窗口表示显示在图象屏幕上的坐标平面部分。通过设置窗口变量，可以定义图象屏幕窗口边界及其他属性。

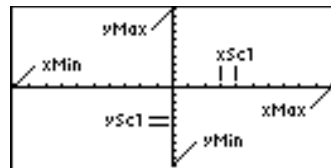
xMin、**xMax**、**yMin** 和 **yMax** 是图象屏幕的边界。

xScl（**x** 尺度）是在 **x** 坐标轴上两个相邻标记之间的距离所表示的单位数。

yScl（**y** 尺度）是在 **y** 坐标轴上两个相邻标记之间的距离所表示的单位数。

xRes 只设置函数图象的像素分辨率，使用 **1** 到 **8** 的整数。

- ◆ 在 **xRes=1**（默认）时，函数在 **x** 坐标轴的每个像素处计算并绘制。
- ◆ 在 **xRes=8** 时，函数在 **x** 坐标轴上每 8 个像素处计算并绘制。

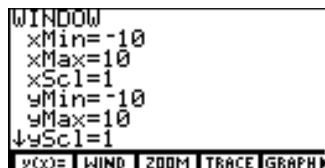


要删除两个坐标轴的标记符号，设置 **xScl=0** 和 **yScl=0**。

小的 **xRes** 值能提高图形分辨率但可能导致 TI-86 绘图更慢。

显示窗口编辑器

要显示窗口编辑器, 从 GRAPH 菜单 (**GRAPH** **F2**) 选择 **WIND** 。
 每种图象方式都有唯一的窗口编辑器。右边所示的窗口编辑器显示了在 **Func** 图象方式下的默认值。↓表示 **xRes=1** (**x** 分辨率) 在窗口编辑器中的 **yScl** 下面。



改变窗口变量值

xMin<xMax 和 **yMin<yMax**

必须都为真才能成功绘制

在示例中, **yMin** 改为 0 。

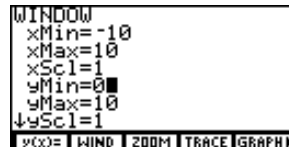
- ❶ 显示窗口编辑器。
- ❷ 将光标移到要修改的窗口变量处。
- ❸ 编辑该值, 可以是表达式。
- ❹ 计算所有表达式并存储结果。

GRAPH **F2**

↓ ↓ ↓

0

ENTER 或 ↓



要从主屏幕或程序编辑器中改变窗口变量值, 则输入数值, 然后按 **STO▶** 。可以从 **VARS WIND** 屏幕 (**2nd** **[CATLG-VARS]** **[MORE]** **[MORE]** **WIND**) 中选择窗口变量或输入单个字符。按 **ENTER** 。

用 Δx 和 Δy 设置图象精度

窗口变量 Δx 和 Δy 定义两个相邻像素中心之间的距离。当显示图象时， Δx 和 Δy 的值是用下面的公式由 $xMin$ 、 $xMax$ 、 $yMin$ 和 $yMax$ 计算得到：

$$\Delta x = (xMin + xMax) / 126$$

$$\Delta y = (yMin + yMax) / 62$$

Δx 和 Δy 不在窗口编辑器中。要修改它们，必须按上述步骤从主屏幕或程序编辑器中改变窗口变量值。当改变存储在 Δx 和 Δy 中的值时，TI-86 自动从 Δx 、 $xMin$ 、 Δy 和 $yMin$ 重新计算 $xMax$ 和 $yMax$ ，并存储新值。

设置图象格式

TI-86 为每种图象方式保存单独的格式设置。

在 DfEq 图象方式下，图象格式屏幕的键序列为 **GRAPH** **MORE** **F1**（第 10 章）。

要显示图象格式屏幕，从 GRAPH 菜单 (**GRAPH** **MORE** **F3**) 选择 **FORMT**。图象格式设置定义所显示图象的各种特征。当前的设置亮显。

要改变设置，将光标移到新的设置，然后按 **ENTER**，这与在方式屏幕上相同。

```
RectGC PolarGC
CoordOn CoordOff
DrawLine DrawDot
SeqG SimulG
GridOff GridOn
AxesOn AxesOff
LabelOff LabelOn
y(x)= LIND ZOOM TRACE GRAPH
```

DifEq 图象方式有唯一的图象格式设置（第10章）。

网格点按行和列覆盖图象屏幕，它们分别对应于 x 和 y 轴上的标记。

RectGC	用直角坐标的图象坐标 x 和 y 显示光标位置；当 RectGC 被设置时，绘制图象，移动自由移动光标，跟踪 x 和 y 的更新；如果 CoordOn 格式也被选择，则显示 x 和 y
PolarGC	用极坐标的图象坐标显示 R 和 θ ；当 PolarGC 被设置时，绘制图象，移动自由移动光标，跟踪 x 、 y 、 R 和 θ 的更新；如果 CoordOn 格式也被选择，则显示 R 和 θ
CoordOn	在图象底部显示光标的坐标值
CoordOff	不在图象底部显示光标的坐标值
DrawLine	绘制由方程编辑器中函数计算出的点，并用直线连接
DrawDot	只绘制由方程编辑器中函数计算出的点
SeqG	（顺序绘制）计算和绘制一个函数的工作全部完成后再计算和绘制下一个函数
SimulG	（同时绘制）对单个 x 值计算和绘制所有选定的函数，然后对下一个 x 值计算和绘制所有选定的函数
GridOff	去掉显示中的网格点
GridOn	显示网格点
AxesOn	显示坐标轴
AxesOff	去掉显示中的坐标轴； AxesOff 覆盖 LabelOff/LabelOn 格式设置
LabelOff	去掉显示中的坐标轴标记
LabelOn	标记坐标轴，如果 AxesOn 也被选定； x 和 y 用于 Func 、 Pol 和 Param 方式；在 DifEq 方式下有不同的标记

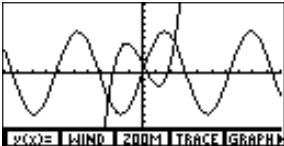
在右面的示例图象中，所有与绘制有关的设置都为默认值。

要查看图象，而 GRAPH 菜单不在底行显示，则在绘制图象后按 **CLEAR**。

当暂停时，右上角的忙指示器变为点化线。

显示图象

要显示图象，则从 GRAPH 菜单选择 **GRAPH**。显示图象屏幕。如果该图象是新定义的，则在 TI-86 绘制图象时在右上角显示忙指示器。



- ◆ 在 **SeqG** 格式下，TI-86 按函数名顺序逐个绘制选定的函数（例如，先绘制 **y1**，再绘制 **y2**，依次类推）。
- ◆ 在 **SimulG** 格式下，TI-86 同时绘制所有选定的函数。

可以从程序中显示和浏览图象（第 16 章）。要在主屏幕上使用图象命令，则从 CATALOG 中选择它们或输入单个字符。

暂停或停止绘制中的图象

- ◆ 要暂停图象绘制，按 **ENTER**。要继续绘制，再按 **ENTER**。
- ◆ 要停止图象绘制，按 **ON**。要重新绘制，从 GRAPH 菜单选择 **GRAPH**。

修改已绘制的图象

要从图象屏幕删除这些项：	按（或选择）：
光标、坐标值或菜单（要恢复菜单，按 EXIT 或 GRAPH ）	CLEAR
自由移动光标和坐标值，但不是菜单	ENTER
光标和坐标值，但不是菜单	GRAPH 或 GRAPH

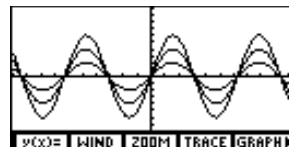
绘制曲线族

如果输入数组作为方程的参数，TI-86 则对数组中的每个值绘制此函数，从而绘制了一族曲线。在 **SimulG** 顺序绘制方式下，TI-86 对每个数组中的第一个元素顺序绘制函数，然后对第二个元素，依次类推。

当在表达式中使用两个以上的数组时，所有数组的长度必须相同。

例如， $\{2,4,6\} \sin x$ 绘制三个函数：

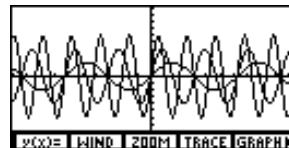
$2 \sin x$ 、 $4 \sin x$ 和 $6 \sin x$ 。



方程 $\{2,4,6\} \sin (\{1,2,3\} x)$

也绘制三个函数：

$2 \sin x$ ， $4 \sin (2x)$ 和 $6 \sin (3x)$ 。



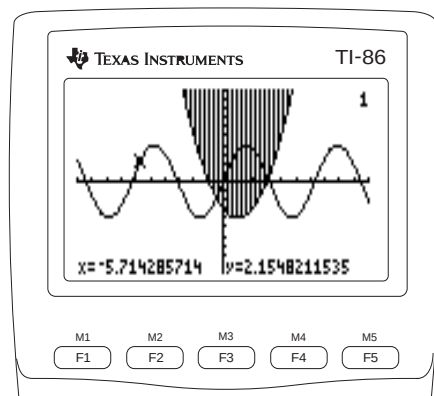
记忆绘图

记忆绘图在按 **GRAPH** 时显示前一次显示的图象，只要所有导致该图象重绘的因素自该图象上次显示以来没有改变。如果自该图象上次显示以后执行下面的任一操作，记忆绘图会重绘图象。

- ◆ 改变了影响图象的方式设置
- ◆ 改变了绘制在上个图象屏幕上的函数或统计图
- ◆ 选择或取消选择了函数或统计图
- ◆ 改变了选定函数中变量的值
- ◆ 改变了窗口变量设置值
- ◆ 改变图象格式设置

6 图象工具

TI-86 图象工具.....	88
跟踪图象	90
用 ZOOM 操作调整图象屏幕的尺寸	91
使用交互数学函数.....	95
对指定 x 值计算函数值.....	101
在图象上绘制.....	101



TI-86 图象工具

第 5 章描述了如何在 **Func** 图象方式下使用 **GRAPH** 菜单项 **y(x)=**、**WIND**、**GRAPH** 和 **FORMT** 定义和显示函数图象。本章描述如何使用其他 **GRAPH** 菜单项调整预置的图象屏幕尺寸、浏览图象并跟踪指定函数、执行数学分析、在图象上绘制以及存储和重新调用图象和图片。您可以在所有四个图象方式下使用大多数的图象工具。

这是在 **Func** 图象方式下的 **GRAPH** 菜单。**GRAPH** 菜单在不同图象方式之间有稍许不同。

GRAPH 菜单										
GRAPH										
y(x)=	WIND	ZOOM	TRACE	GRAPH	▶	MATH	DRAW	FORMT	STGDB	RCGDB
					▶	EVAL	STPIC	RCPIC		

- ZOOM

显示 **GRAPH ZOOM** 菜单；用这些项调整预置的图象屏幕尺寸
- TRACE

激活跟踪光标；用此光标跟踪指定的函数图象
- MATH

显示 **GRAPH MATH** 菜单；用此菜单从数学角度浏览图象
- DRAW

显示 **GRAPH DRAW** 菜单；用此菜单在图象上绘制
- STGDB

显示 **Name=** 提示和 **GDB** 菜单；用此提示输入 **GDB** 变量
- RCGDB

显示 **Name=** 提示和 **GDB** 菜单；用此菜单重新调用 **GDB** 变量
- EVAL

显示 **Eval x=** 提示；用此提示输入 **x** 的值，要在输入值处求解当前函数
- STPIC

显示 **Name=** 提示和 **PIC** 菜单；用此提示输入 **PIC** 变量
- RCPIC

显示 **Name=** 提示和 **PIC** 菜单；用此菜单来重新调用 **PIC** 变量

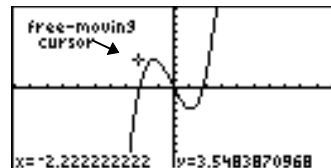
在示例中，绘制了函数 $y(x)=x^3+3x^2-4x$ 的图象。

数值显示方式设置不影响坐标显示。

使用自由移动光标

从 GRAPH 菜单中选择 GRAPH 时，显示图象屏幕，自由移动光标显示在屏幕中央。

光标显示为一个加号，其中心象素闪烁。要移动光标，按 $\rightarrow$ 、 $\leftarrow$ 、 $\uparrow$ 或 $\downarrow$ ；它按所按光标键的方向移动。



- ◆ 在 RectGC 格式下，每次光标移动更新变量 x 和 y 值。在 PolarGC 格式下，每次光标移动更新变量 x 、 y 、 R 和 θ 值。
- ◆ 在 CoordOn 格式下，当移动光标时，光标坐标 x 和 y 显示在图象屏幕的底部。

图象精度

当移动光标时所显示的坐标值与实际的数学坐标近似，精确范围在象素的宽度、高度之间。随着 $xMin$ 与 $xMax$ 之间的差值和 $yMin$ 与 $yMax$ 之间的差值越来越小（例如，放大图象时），图象更精确且坐标值更接近实际的数学坐标。

自由移动光标的坐标表示光标在图象屏幕上的位置。在函数的绘制点之间精确地移动自由移动光标是很困难的。要想容易地沿着函数移动，使用跟踪光标（第 90 页）。

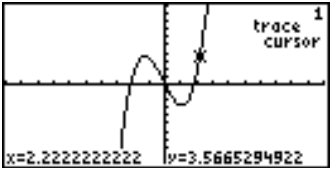
跟踪图象

要显示图象并开始跟踪，从 GRAPH 菜单中选择 **TRACE** 。

在示例中，绘制了函数
 $y(x)=x^3+3x^2-4x$ 的图象。

跟踪光标的外形为一个小方块，它的每个角各有一条闪烁的对角线。跟踪光标起初出现在第一个选定的函数上，位于离屏幕中间最近的 x 值处。

如果 **CoordOn** 格式选中，光标坐标显示在屏幕的底部。



当输入自变量值的第一个字符时，**x=** 提示显示（或是 **θ=** 或是 **t=**）。输入值可以是表达式。

如果函数在 x 值处没定义，则 y 值为空。

要移动跟踪光标...

到函数中次大或次小的绘制点

到当前方程中任何有效的自变量值（ x 、 θ 、或 t ）处

在 x 值处从一个函数到另一个函数，按方程编辑器中所选函数的顺序或相反顺序

从曲线族中的一个成员到另一个成员（第 5 章）

按这些键

▶ 或 **◀**

value **ENTER**

▼ 或 **▲**

◀ 或 **▶**

沿函数移动跟踪光标时， y 值根据 x 值计算。即 $y=y_n(x)$ 。当跟踪超出图象屏幕的顶部或底部时，显示在屏幕上的坐标继续变化，好象光标仍在屏幕上。

平移：要查看当前图象屏幕左侧或右侧的函数坐标，在跟踪时按住 **◀** 或 **▶**。如果在跟踪时平移超出了屏幕的左侧或右侧时，TI-86 将自动改变 **xMin** 和 **xMax** 的值。

快速缩放：跟踪时，可以按 **[ENTER]** 调整图象屏幕，以便跟踪光标的位置成为新图象屏幕的中心，即使已经将光标移出了显示区的顶部或底部。实际上，这是垂直平移。

停止和重新开始跟踪

要停止跟踪并恢复自由移动光标，按 **[CLEAR]** 或 **[GRAPH]** 。

要重新开始跟踪，从 **GRAPH** 菜单选择 **TRACE** 。如果记忆绘图尚未重新绘制图象（第 5 章），跟踪光标位于停止跟踪点处。

用 ZOOM 操作调整图象屏幕的尺寸

要查看当前的窗口变量值，从 **GRAPH** 菜单中选择 **WIND** 。

标准的 TI-86 图象屏幕显示由存储在窗口变量中的值所定义 **xy** 平面的一部分。使用 **GRAPH ZOOM** 菜单项，可以改变一些或所有的窗口变量值并重新显示图象，这通常是通过一次击键完成。结果显示 **xy** 平面的一个更小或更大的部分。

GRAPH ZOOM 菜单 **[GRAPH]** **[F3]**

y(x)=	WIND	ZOOM	TRACE	GRAPH						
BOX	ZIN	ZOUT	ZSTD	ZPREV	▶	ZFIT	ZSQR	ZTRIG	ZDECM	ZDATA
					▶	ZRCL	ZFACT	ZOOMX	ZOOMY	ZINT
					▶	ZSTO				

要取消任何 ZOOM 菜单项的效果并返回到默认的窗口变量值，选择 ZSTD。

如果绘制一个圆，但看起来象椭圆，可以用 ZSQR 来复位窗口变量值，以便使圆的图象看起来为圆形。

BOX	绘制一个框来定义图象屏幕
ZIN	（放大）用因子 xFact 和 yFact 放大光标周围的图象
ZOUT	（缩小）用因子 xFact 和 yFact 显示光标周围更多图象
ZSTD	以标准尺寸显示图象；复位默认的窗口变量值
ZPREV	逆转上一个放缩操作；窗口变量回复到以前的值
ZFIT	重新计算 yMin 和 yMax 以包括选定函数在当前 xMin 和 xMax 之间的 y 的最大值和最小值
ZSQR	在 x 坐标轴和 y 坐标轴上设置相等大小的象素；在一个方向调整窗口变量值，结果 $\Delta x = \Delta y$ ，而 xScl 和 yScl 保持不变；当前图象的中点（不是坐标原点）成为新图象的中点
ZTRIG	设置适用于 Radian 方式下的三角函数的内置窗口变量： xMin=-8.24668071567 xScl=1.5707963267949 ($\pi/2$) yMax=4 xMax=8.24668071567 yMin=-4 yScl=1
ZDECM	设置 $\Delta x=.1$ 、 $\Delta y=.1$ 、 xMin=-6.3 、 xMax=6.3 、 xScl=1 、 yMin=-3.1 、 yMax=3.1 和 yScl=1
ZDATA	设置窗口变量值以显示所有统计数据点；只调整 xMin 和 xMax ，只适用于直方图、散射图、和统计图（第 14 章）
ZRCL	使用存储在用户定义的缩放 — 窗口变量 (ZSTO) 中的窗口变量值
ZFACT	显示 ZOOM FACTORS 屏幕
ZOOMX	只按 xFact 因子缩小；而忽略 yFact （第 93 页）
ZOOMY	只按 yFact 因子来缩小；而忽略 xFact
ZINT	在坐标轴上设置整数值；设置 $\Delta x=1$ 、 $\Delta y=1$ 、 xScl=10 和 yScl=10 ；当前光标在按 [ENTER] 后成为新图象屏幕的中心
ZSTO	将当前窗口变量值存储到用户定义的缩放 — 窗口变量 (ZRCL)

定义定制放大

使用 **BOX**，可以放大当前图象屏幕内的任何矩形区域。

在开始下面的步骤之前，在方程编辑器中输入一个函数。

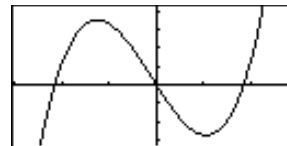
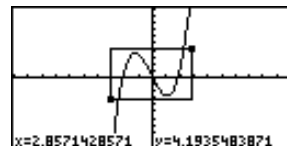
在示例中，绘制了函数

$y(x)=x^3+3x^2-4x$ 的图象。

要取消 **BOX** 而不重新定义图象屏幕，按 **CLEAR**。

当重新绘制图象时，TI-86 会更新窗口变量值。

- ① 从 **GRAPH ZOOM** 菜单中选择 **BOX**。缩放光标显示在屏幕中央。
`[GRAPH] [F3]`
`[F1]`
- ② 将光标移到任何要定义为缩放框顶点的位置；用小方块标记该顶点。
`[▶] [▼] [◀] [▲]`
`[ENTER]`
- ③ 从第一个顶点移动光标，将创建一个可调整的缩放框，它的对角顶点为小方块和光标。
`[▶] [▼] [◀] [▲]`
- ④ 在定义缩放框后，在新图象屏幕中重新绘制所有选定的函数。
`[ENTER]`
- ⑤ 从屏幕中清除菜单。
`[CLEAR]`



设置缩放因子

要从主屏幕或在程序编辑器中存储 **xFact** 或 **yFact**，可以从 **VARS ALL** 屏幕中选择这两个变量或用大写字母键和小写字母键输入。

缩放因子定义放大或缩小因子，**ZIN**、**ZOUT**、**ZOOMX** 和 **ZOOMY** 用这些因子放大或缩小某点周围的图象。要显示缩放因子编辑器，从 **GRAPH ZOOM** 菜单中选择 **ZFACT**（按 `[GRAPH] [F3] [MORE] [MORE] [F2]`）。**xFact** 和 **yFact** 必须 ≥ 1 。两个因子在所有图象方式下的默认值都是 4。

放大和缩小图象

ZIN 放大光标位置周围的那部分图象。**ZOUT** 以光标位置为中心显示图象的更大一部分。**xFact** 和 **yFact** 确定缩放的程度。下面的步骤描述如何使用 **ZIN**。要使用 **ZOUT**，在步骤 2 中用它代替 **ZIN**。

在示例中，绘制了函数
 $y(x)=x^3+3x^2-4x$ 的图象。

当选择 ZOOM 功能时，记忆
 绘图显示当前图象。

要在缩放完成之前取消它，
 按 **CLEAR**。

- 1 检查 **xFact** 和 **yFact**；如有必要，则改变它们的值。

GRAPH
F3 MORE
MORE F2

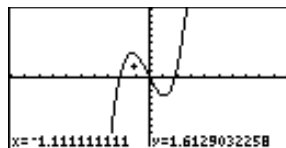
ZOOM FACTORS
 xFact=4
 yFact=4

- 2 从 GRAPH ZOOM 菜单选择 **ZIN** 显示放缩光标。

F3 F2

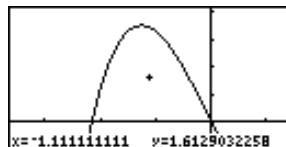
- 3 将放缩光标移到图象屏幕新的中心点。

▶ ▼ ◀ ▲



- 4 放大。TI-86 按 **xFact** 和 **yFact** 值调整图象屏幕，更新窗口变量值，并以光标位置为中心重新绘制选定函数。

ENTER



可以在当前图象上连续放大（或缩小），除非按 **ENTER**、**▶**、**▼**、**◀** 或 **▲** 以外的键。

- ◆ 要在同一点再次放大（或缩小），按 **ENTER**。
- ◆ 要在新中心点放大（或缩小），移动光标并按 **ENTER**。

要用 **xFact** 因子只在水平坐标轴上缩小图象，在上面的步骤 2 中用 **ZOOMX** 代替 **ZIN**。**ZOOMX** 以光标位置为中心绘制选定函数并更新一些窗口变量值；**yMin** 和 **yMax** 值不变。

要用 **yFact** 因子只在垂直坐标轴上缩小图象，在上面的步骤 2 中用 **ZOOMY** 代替 **ZIN**。**ZOOMY** 以光标位置为中心绘制选定函数并更新一些窗口变量值；**xMin** 和 **xMax** 值不变。

可以在任何图象方式下从 VARS WIND 屏幕中选择所有的缩放窗口变量。

也可以单独输入变量字符。

缩放窗口变量在复位默认设置时重新使用其标准默认值。

存储和重新调用缩放 — 窗口变量值

- ◆ 要将所有当前缩放窗口变量值同时存储为用户定义的定制放缩功能，从 GRAPH ZOOM 菜单选择 **ZSTO** 。
- ◆ 要执行用户定义的定制放缩，它将图象屏幕复位到存储的缩放窗口变量值，从 GRAPH ZOOM 菜单选择 **ZRCL** 。

在以下图象方式下使用 ZSTO :	存储到以下缩放窗口变量:
Func 、 Pol 、 Param 和 DifEq 图象方式	zxMin 、 zxMax 、 zxScl 、 zyMin 、 zyMax 和 zyScl
仅 Pol 图象方式	zθMin 、 zθMax 和 zθStep
仅 Param 图象方式	ztMin 、 ztMax 和 ztStep
仅 DifEq 图象方式	ztMin 、 ztMax 、 ztStep 和 ztPlot

使用交互数学函数

当选择 GRAPH MATH 操作时，记忆绘图用跟踪光标显示当前图象。要执行 GRAPH MATH 操作，按  和  移到该函数。

当 GRAPH MATH 菜单操作提示指定左边界、右边界和估计值时，指定值的精确度会影响 TI-86 计算结果所花费的时间；估计值越准确，计算时间越短。

GRAPH MATH 菜单

MATH	DRAW	FORMT	STGDB	RCGDB						
ROOT	dy/dx	f(x)	FMIN	FMAX	▶	INFLC	YICPT	ISECT	DIST	ARC
					▶	TANLN				

GRAPH MATH 菜单对于 **Pol** 和 **Param** 图象方式有稍许不同（第8章和第9章）。

DifEq 图象方式没有 GRAPH MATH 菜单。

ROOT	用指定的左边界、右边界和估计值求函数的根
dy/dx	求在跟踪光标位置处函数的数值导数（斜率）
f(x)	用指定的左边界和右边界求函数的数值积分
FMIN	用指定的左边界、右边界和估计值求函数的最小值
FMAX	用指定的左边界、右边界和估计值求函数的最大值
INFLC	用指定的左边界、右边界和估计值求函数的拐点
YICPT	求函数与 y 轴交点（ y 在 x=0 时的值）
ISECT	用指定的左边界、右边界和估计值求两个函数的交点
DIST	求指定左边界和右边界之间的直线距离
ARC	求函数在两个指定点之间的距离
TANLN	在指定点处作切线

影响 GRAPH MATH 操作的设置

- ◆ 误差变量 **tol**（参阅附录）影响 **f(x)**、**FMIN**、**FMAX** 和 **ARC** 的精度。精度随着误差值的减小而提高。
- ◆ 步长变量 **δ**（参阅附录）影响 **dy/dx**、**dxNDer** 微分方式下的 **INFLC**（第1章）、**ARC** 和 **TANLN** 的精度。精度随着步长值的减小而提高。
- ◆ 微分方式设置影响 **dy/dx**、**INFLC**、**ARC** 和 **TANLN**；**dxDer1**（精确）方式比 **dxNDer**（数值）方式（第1章）更精确。

使用 ROOT、FMIN、FMAX 或 INFLC

除了步骤 1 中的菜单选择外, 使用 **ROOT**、**FMIN**、**FMAX** 和 **INFLC** 的步骤是相同。

在示例中, 函数 $y(x)=x^3+3x^2-4x$ 被选中。这里的步骤 2 是不必要的, 因为只有一个函数被选中。

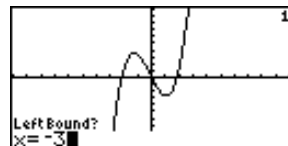
直接输入左边界、右边界或估计值时, $x=$ 提示显示在图象屏幕底部。

- 从 GRAPH MATH 菜单选择 **ROOT**。Left Bound? 提示显示。

[GRAPH] [MORE] [F1]
[F1]

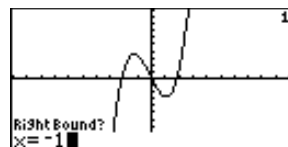
- 将光标移到要求根的函数上。

▼ ▲



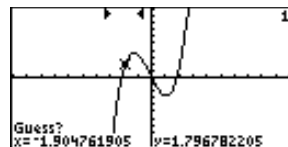
- 指定 x 的左边界。或者将跟踪光标移到左边界, 或者直接输入值。Right Bound? 显示。

[←] 3 [ENTER]
(或 [←] [→]
[ENTER])



- 和步骤 3 一样指定 x 的右边界。Guess? 显示。

[←] 1 [ENTER]
(或 [←] [→]
[ENTER])

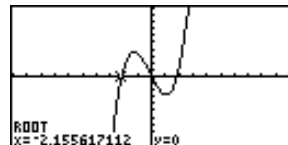


- 在左边界和右边界之间估计一个接近于根的 x 值。或者移动光标, 或者输入值。

[←] [→] (或 [←] 2)

- 求解 x 值。结果光标显示在结果点上, 光标坐标值显示, 且 x 值存储到 **Ans**。

[ENTER]



使用 $\int f(x)$ 、DIST 或 ARC

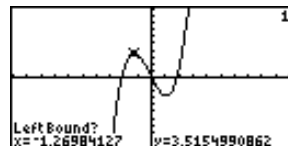
除了步骤 1 中的菜单选择外, 使用 $\int f(x)$ 、DIST 和 ARC 的步骤相同。

在示例中, 函数 $y(x)=x^3+3x^2-4x$ 被选中。这里的步骤 2 和 4 是不必要的, 因为只有一个函数被选中。

对于 DIST, 当正指定右边界时, 将绘制一条从左边界到右边界的直线。

- 1 从 GRAPH MATH 菜单选择 **DIST**。当前图象和 **Left Bound?** 提示一起显示。

[GRAPH] [MORE]
[F1] [MORE] [F4]



- 2 将光标移到左边界为一点的函数上。

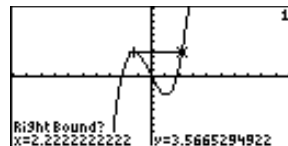
▼ ▲

- 3 选择 **x** 的左边界。或者将光标移到左边界, 或者输入 **x** 值。 **Right Bound?** 显示。

◀ ▶ [ENTER] 或
value [ENTER]

- 4 (只对 **DIST**) 如果想使右边界是另一个函数上的点, 将光标移到另一个函数上。

▼ ▲



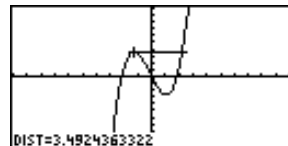
- 5 选择右边界。或者将光标移到右边界, 或者输入它的 **x** 值。

◀ ▶ 或
value

- 6 求解。

[ENTER]

- ◆ 对于 **DIST**, 结果 **DIST=** 显示并存储到 **Ans**。
- ◆ 对于 **ARC**, 结果 **ARC=** 显示并存储到 **Ans**。
- ◆ 对于 $\int f(x)$, 结果 $\int f(x)=$ 显示, 被阴影化, 并存储到 **Ans**。函数积分误差值存储到变量 **fnIntErr** (参阅附录)。要删除阴影, 从 GRAPH DRAW 菜单选择 **CLDRW** (第 103 页)。



使用 dy/dx 或 TANLN

除了步骤 1 中的菜单选择外, 使用 dy/dx 和 TANLN 的步骤相同。

在示例中, 函数
 $y(x)=x^3+3x^2-4x$ 被选中。

TANLN (GRAPH MATH 菜单) 和 TanLn (GRAPH DRAW 菜单) 都在图象上作切线; 但只有 TANLN 显示求解结果, dy/dx 。

- 1 从 GRAPH MATH 菜单中选择 dy/dx 。当前图象显示。

GRAPH MORE
F1 F2

- 2 将光标移到要求某点的导数、或斜率的函数上。

▼ ▲

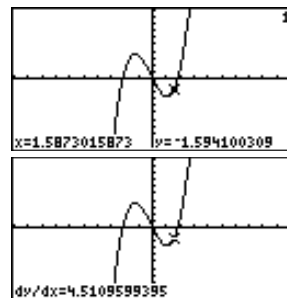
- 3 将光标移到该点 (或输入 x 值)。

◀ ▶

- 4 求解。

ENTER

- ◆ 对于 dy/dx , 结果 $dy/dx=$ 显示并存储到 Ans。
- ◆ 对于 TANLN, 也显示切线。要删除切线和 $dy/dx=$ 提示, 从 GRAPH DRAW 菜单选择 CLDRW。



在示例中, 函数
 $y(x)=x^3+3x^2-4x$ 和
 $y(x)=x^2+3x-3$ 被选中。

使用 ISECT

- 1 从 GRAPH MATH 菜单中选择 **ISECT**。当前图象显示, **First Curve?** 显示在图象屏幕底部。
- 2 选择第一个函数 (曲线)。光标移到下一个函数且 **Second Curve?** 显示。
- 3 选择第二个函数 (曲线)。**Guess?** 显示。
- 4 估计交点。或者将光标移到距交点很近的点, 或者输入 **x** 值。
- 5 求解。结果光标显示在交点处, 光标坐标就是结果, **x** 值存储到 **Ans**。

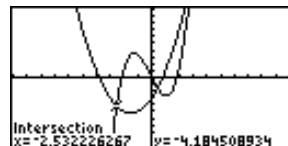
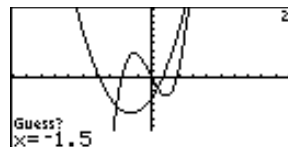
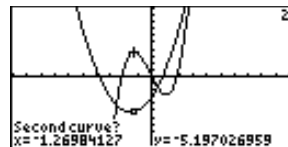
[GRAPH] [MORE]
 [F1] [MORE] [F3]

▼ ▲ [ENTER]

▼ ▲ [ENTER]

← 1 . 5
 (或 ← →)

[ENTER]



使用 YICPT

要使用 **YICPT**, 从 GRAPH MATH 菜单 ([GRAPH] [MORE] [F1] [MORE] [F2]) 选择 **YICPT**。按 ▼ 和 ▲ 选择函数, 然后按 [ENTER]。结果光标显示在函数与 **y** 轴截点处, 光标坐标值显示, **y** 值存储到 **Ans**。

要从 Eval x= 提示中清除输入的数值，按 **CLEAR**。

要取消 EVAL，在清除 Eval x= 提示之后按 **CLEAR**。

表达式对 x 值有效。

可以连续输入有效 x 值来计算选定函数值。

对指定 x 值计算函数值

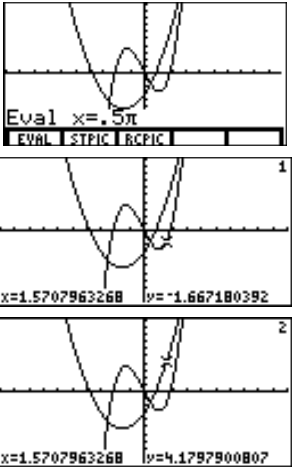
- 1 从 GRAPH 菜单中选择 **EVAL**。图象显示，**Eval x=** 显示在左下角。
- 2 输入窗口变量 **xMin** 和 **xMax** 之间的实数 **x** 值。
- 3 求解。结果光标显示在第一个选定的函数上，位于输入的 **x** 值处。坐标值显示，右上角的编号表示对哪个函数求解。
- 4 将结果光标移到下一个或前一个选定函数。结果光标在下一个或前一个函数上，位于输入的 **x** 值处，坐标值显示，函数编号改变。

GRAPH **MORE**
MORE **F1**

5 **2nd** **[π]**

ENTER

↑ **↓**



在图象上绘制

可以在任意图象方式下使用图象工具（不包括 **DrInV**）在当前图象上绘制点、线、圆、阴影区域和文本。图象工具使用所显示的 **x** 和 **y** 坐标值。

在图象上绘制之前

所有的图形都是临时的；它们没有存储在图象数据库中。任何导致记忆绘图重新绘制图象的操作都会删除所有的图形。因此，在使用任何图形工具前，要考虑是否要先执行这些绘制操作。

- ◆ 改变影响图象的方式设置
- ◆ 选择、取消选定或编辑当前函数或统计图
- ◆ 改变选定函数使用的变量值
- ◆ 改变窗口变量值
- ◆ 改变图象格式设置或图象样式
- ◆ 用 **CLDRW** 清除当前图形

保存和重新调用已绘制的图形

要将定义当前图象的元素存储到图象数据库 (**GDB**) 变量，从 **GRAPH** 菜单选择 **STGDB**。这些信息类型存储到 **GDB** 变量：

- ◆ 方程编辑器函数
- ◆ 窗口变量值
- ◆ 图象样式设置
- ◆ 格式设置

要在以后重新调用存储的 **GDB**，从 **GRAPH** 菜单选择 **RCGDB**，然后从 **GRAPH RCGDB** 菜单选择 **GDB** 变量。当重新调用 **GDB** 时，存储在 **GDB** 中的信息取代这些类型的当前信息。

要将当前显示图象（包括图形）存储到图形 (**PIC**) 变量，从 **GRAPH** 菜单选择 **STPIC**，只有图象图形存储到指定 **PIC** 变量。

要在以后将一个或更多的存储图形添加到图象中，从 **GRAPH** 菜单选择 **RPCIC**，然后从 **GRAPH RPCIC** 菜单选择 **PIC** 变量。

图象数据库 (**GDB**) 和图形 (**PIC**) 变量名长度可以为 1 到 8 个字符。第一个字符必须是字母。

下一节描述如何在图象上绘制线、点、曲线和文本；然后可以将这些图形存储到 **PIC** 变量。

清除已绘制的图形

当图象显示时要清除已绘制的图形，从 GRAPH DRAW 菜单选择 **CLDRW**。图象被重新绘制并显示，上面不再有绘制的元素。

要从主屏幕清除已绘制的图形，从 CATALOG 中选择 **CIDrw**。**CIDrw** 粘贴到光标处。按 **[ENTER]**。**Done** 显示；当再次显示该图象时，没有图形显示。

GRAPH DRAW 菜单

[GRAPH] [MORE] [F2]

MATH	DRAW	FORMT	STGDB	RCGDB						
Shade	LINE	VERT	HORIZ	CIRCL	▶	DrawF	PEN	PTON	PTOFF	PTCHG
					▶	CLDRW	PxOn	PxOff	PxChg	PxTest
					▶	TEXT	TanLn	DrInv		

只能在主屏幕或在编辑器程序中使用下面的 GRAPH DRAW 菜单项。

Shade(	阴影图象的指定区域（请参阅 104 页）
DrawF <i>expression</i>	将表达式 <i>expression</i> 作为函数绘制
PxOn(<i>row,column</i>)	打开 (<i>row,column</i>) 处的象素
PxOff(<i>row,column</i>)	关闭 (<i>row,column</i>) 处的象素
PxChg(<i>row,column</i>)	改变 (<i>row,column</i>) 处象素的打开/关闭状态
PxTest(<i>row,column</i>)	如果 (<i>row,column</i>) 处的象素为打开状态，则返回 1 ；否则返回 0
TanLn(<i>expression,x</i>)	将表达式 <i>expression</i> 作为函数绘制，且绘制表达式在 <i>x</i> 处的切线
DrInv <i>expression</i>	绘制表达式 <i>expression</i> 的倒数

DrInv 在 Pol、Param 或 DifEq 图象方式下不可用。

对于 PxOn、PxOff、PxChg 和 PxTest，行和列是整数，其中 $0 \leq \text{row} \leq 62$ 而 $0 \leq \text{column} \leq 126$ 。

对于 DrawF、TanLn 和 DrInv，表达式 *expression* 是关于 *x* 的函数。也不能用在表达式 *expression* 中包括数组的方法来绘制一族曲线。

要在没有其他图象的情况下复制此示例，在输入所示的指令之前关闭所有的方程和统计图。

阴影图象区域

阴影图象区域的语法为：

Shade(*lowerFunc*,*upperFunc*[,*xLeft*,*xRight*,*pattern*,*patternRes*])

pattern 指定了四种阴影图案之一。

- 1 垂直线（默认）
- 2 水平线
- 3 反斜线 (45°)
- 4 正斜线 (45°)

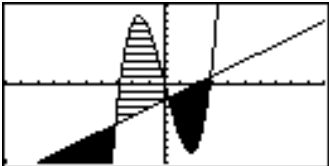
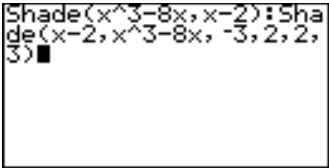
patternRes 指定了八种阴影分辨率之一。

- 1 每个像素（默认）
- 2 每两个像素
- 3 每三个像素
- 4 每四个像素
- 5 每五个像素
- 6 每六个像素
- 7 每七个像素
- 8 每八个像素

- ◆ 阴影由 *lowerFunc* 和 *upperFunc* 指定的区域。
- ◆ *xLeft* > **xMin** 和 *xRight* < **xMax** 必须为真。
- ◆ *xLeft* 和 *xRight* 指定了阴影的左边界和右边界（**xMin** 和 **xMax** 是默认值）。

这些 GRAPH DRAW 菜单项是交互式的。而且，可以在主屏幕或在程序中使用除 **PEN** 外的所有项（参阅“**A 到 Z 参考**”）。

- LINE** 由光标指定的一点到另一点绘制线段
- VERT** 绘制垂直线，可以将它移到任意显示的 **x** 值处



HORIZ 绘制水平线, 可以将它移到任意显示的 y 值处

CIRCL 用光标指定的中心点和半径绘制圆

PEN 绘制光标在图象屏幕上移动的轨迹

PTON 打开光标所在点

PTOFF 关闭光标所在点

PTCHG 改变光标所在点的打开/关闭状态

CLDRW 从图象屏幕清除所有图形; 重新绘制图象

TEXT 在图象的光标位置绘制字符

绘制线段

在示例中, 函数
 $y(x)=x^3+3x^2-4x$ 和
 $y(x)=x^2+3x-3$ 被选中。

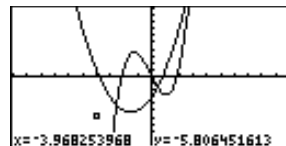
① 从 GRAPH DRAW 菜单中选择 **LINE**。图象显示。

GRAPH MORE

F2 F2

▶ ▼ ◀ ▲

ENTER



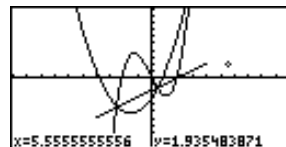
② 用光标定义线段的一个端点。

③ 定义线段的另一个端点。当移动光标时, 一条线从第一个定义的端点延伸到光标。

▶ ▼ ◀ ▲

④ 绘制线。

ENTER



要绘制多条线段, 重复步骤 2 和 3; 要取消 **LINE**, 按 **CLEAR**。

在示例中, 函数 $y(x)=x^3+3x^2-4x$ 被选中。
而且, ZIN 被执行一次,
此时缩放光标在 (0,0),
 $xFact=2$, $yFact=2$ 。

绘制垂直线或水平线

- 1 从 GRAPH DRAW 菜单中选择 **VERT** (或 **HORIZ**)。图象显示, 且一条垂直线或水平线在光标处绘制出来。
- 2 将直线移到要它通过的 x 值处 (如果是水平线, 则为 y 值处)。
- 3 在图象上绘制线。

GRAPH MORE
F2 F3
(或 F4)
← →
(或 ↑ ↓)
ENTER



要绘制多条线, 重复步骤 2 和 3; 要取消 **VERT** 或 **HORIZ**, 按 CLEAR。

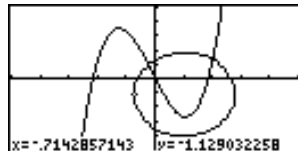
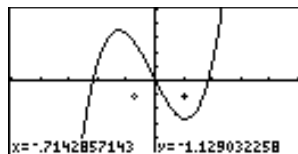
绘制圆

在示例中, 函数 $y(x)=x^3+3x^2-4x$ 被选中。
而且, ZIN 执行一次,
此时缩放光标在 (0,0),
 $xFact=2$, $yFact=2$ 。

这时圆看起来是圆, 而不管窗口变量的值。当从 CATALOG 中使用来绘制圆时, 当前窗口变量值可能会破坏其形状。

- 1 从 GRAPH DRAW 菜单选择 **CIRCL**。图象显示。
- 2 用光标定义圆的中心点。
- 3 将光标移到环境的任意点。
- 4 绘制圆。

GRAPH MORE F2
F5
→ ↓ ← ↑
ENTER
→ ↓ ← ↑
ENTER



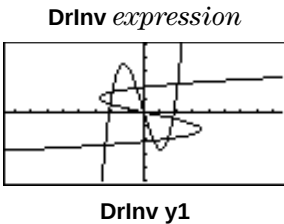
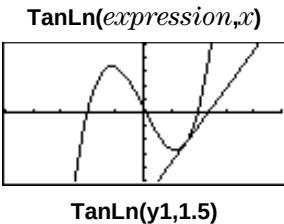
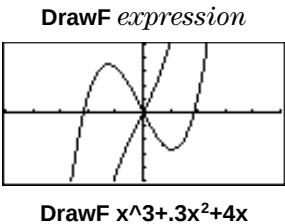
要绘制多个圆, 重复步骤 2 和 5; 要取消 **CIRCL**, 按 CLEAR。

对于 DrawF、TanLn 和 DrInv，可以将任何存储有效表达式的变量用作表达式（包括已取消选定的方程变量）。

在示例中， $y1=x^3+3x^2-4x$ 被选中。

绘制函数、正切线或反函数

对于 DrawF、TanLn 和 DrInv，表达式 *expression* 是关于 *x* 的函数。当从 GRAPH DRAW 菜单中选择 DrawF、TanLn 或 DrInv 时，该项粘贴到主屏幕或程序编辑器。在执行时，返回图形。DrInv 通过在 *y* 坐标轴上绘制 *x* 值和在 *x* 坐标轴上绘制 *y* 值来绘制 *expression* 的反函数。DrInv 只可用于 Func 图象方式下。



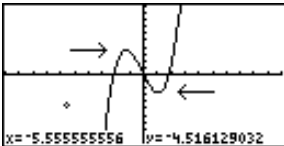
在示例中，函数 $y(x)=x^3+3x^2-4x$ 被选中。而且，ZSTD 已被执行。

要绘制对角线或曲线，请打开笔，按 **ENTER** **ENTER**，按 **↓** **↑**（或 **↓** **↑**，依次类推），并重复。

徒手绘制点、线和曲线

- ❶ 从 GRAPH DRAW 菜单选择 PEN。
- ❷ 将光标移到开始绘图处。
- ❸ 打开笔。
- ❹ 绘制想绘制的任何东西。
- ❺ 关闭笔。

GRAPH **MORE** **F2**
MORE **F2**
↓ **↓** **↓** **↓**
ENTER
↓ **↓** **↓** **↓**
ENTER



要绘制多个点、线、或曲线，重复步骤 2 和 s；要取消，按 **CLEAR**。

此示例对 **PEN** 的示例图象增加。在开始之前，可能要将箭头存储到图形变量（第 102 页）。

要在使用 **TEXT** 时擦除字符，将 **TEXT** 光标移到字符上面然后按 **[ALPHA] [↵]** 或 **[2nd] [alpha] [↵]** 来覆写它。

在示例中，函数 $y(x)=x^3+3x^2-4x$ 被选中。而且，**ZSTD** 已执行。在 $(-5,5)$ 、 $(5,5)$ 、 $(5,-5)$ 和 $(-5,-5)$ 处的点被打开。

在图象上放置文本

- 1 从 **GRAPH DRAW** 菜单选择 **TEXT**。文本光标显示。
- 2 将光标移到要输入文本处。在文本光标下面输入文本。
- 3 设置小写字母锁，输入 **min**。（小写光标（**[a]**）显示在右上角）
- 4 将光标移到另一个位置。
- 5 输入 **max**（小写字母锁保持打开）。

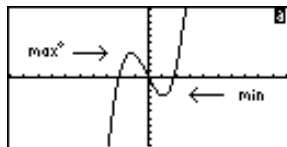
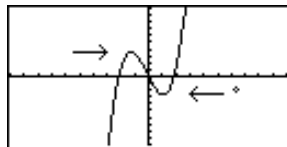
[GRAPH] [MORE] [F2]
[MORE] [MORE] [MORE]
[F1]

[▶] [▼] [◀] [▲]

[2nd] [alpha] [ALPHA]
[M] [I] [N]

[▶] [▼] [◀] [▲]

[M] [A] [X]



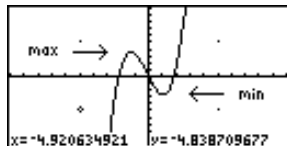
打开或关闭点

- 1 从 **GRAPH DRAW** 菜单选择 **PTON**（或 **PTOFF**）。
- 2 将光标移到要绘制（或擦除）点的地方。
- 3 打开（或关闭）点。

[GRAPH] [MORE] [F2]
[MORE] [F3]

[▶] [▼] [◀] [▲]

[ENTER]

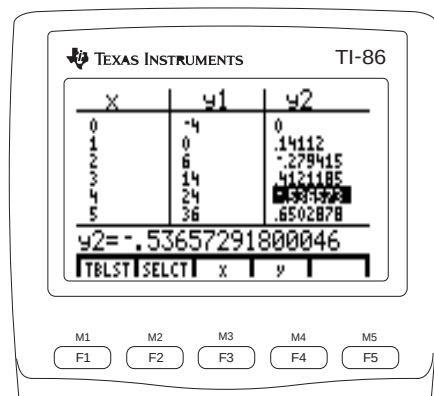


要继续绘制点，重复步骤 2 和 3。要取消 **PTON**，按 **[CLEAR]**。

7

运算表

显示运算表	110
设置运算表	113
清除运算表	114



显示运算表

要显示方程编辑器，按 **GRAPH** **F1**（第5章）。

运算表在方程编辑器中显示选定函数的自变量值和因变量值，选定函数可最多 99 个。运算表中的每个因变量代表一个方程编辑器中存储的选定函数，该函数可用于当前图象方式下。

TABLE（运算表）菜单



运算表 **TABLE** **F1**

在示例中，选择了 $y_1=x^2+3x-4$ 和 $y_2=\sin(3x)$ 并设置所有默认值。

如果必要，运算表在列中缩写数值。

编辑行（函数名和当前显示单元的精确值）



要编辑方程，在方程运算表列中按 **↑**，直到光标在顶端行亮显方程变量，然后按 **ENTER**。存储到当前方程变量的表达式显示在编辑行。

运算表的自变量和因变量

图象方式	自变量	因（方程）变量
Func （函数）	x	y1 到 y99
Pol （极坐标）	θ	r1 到 r99
Param （参数）	t	xt1/yt1 到 xt99/yt99
DifEq （微分方程）	t	Q1 到 Q9

在 **DifEq** 方式下，如果方程有初始条件数组，运算表用第一个数组元素求解方程（第 10 章）。

浏览运算表

要...	这样做：
显示运算表中更多因变量	按 ↓ 或 ↑
显示任何列中数值的较大值	按 ↓ （只在 Indpnt: Auto 被设置时；第 112 页）
设置 TblStart 为较小值	在自变量列按 ↑ ，直到光标移过当前 TblStart （第 112 页）
在编辑行显示方程，可以编辑或取消选定它	按 ↓ 或 ↑ 将光标移到方程变量列，按住 ↑ ，直到光标亮显方程名，然后按 ENTER ；方程显示在编辑行

运算表菜单 TABLE F1

运算表对每种图象方式有唯一的菜单，如下所示。

在函数图象方式下

TBLST	SELCT	x	y	
-------	-------	---	---	--

在极坐标图象方式下

TBLST	SELCT	θ	r	
-------	-------	----------	---	--

在参数图象方式下

TBLST	SELCT	t	xt	yt
-------	-------	---	----	----

在微分方程图象方式下

TBLST	SELCT	t	Q	
-------	-------	---	---	--

TBLST 显示运算表设置编辑器

SELCT 在编辑行上，取消选定方程

x 和 **y**； **θ** 和 **r**；**t**、**xt** 和 **yt** 或 **t** 和 **Q** 在编辑行上，将变量粘贴到光标处；变量根据图象方式而变

- ◆ 要将方程添加到运算表，在方程编辑器中选择方程（第 5 章）。**SELCT** 只从运算表中删除方程。
- ◆ 要从运算表列中删除方程，从运算表菜单中选择 **SELCT**。被删除方程后面的其他方程左移一列。
- ◆ 要用 **SELCT** 取消选定方程，方程和光标必须显示在编辑行。如果方程在编辑行而光标不在，按 ENTER。
- ◆ 要比较两个在方程编辑器中非连续定义的因变量，使用运算表屏幕菜单的 **SELCT** 来取消选定两者之间的因变量。

设置运算表

要使用当前的运算表设置来显示运算表，从 **TABLE** 菜单选择 **TABLE**。

要显示运算表设置编辑器，从 **TABLE** 菜单选择 **TBLST**。右边的屏幕显示了运算表设置的默认值。

TblStart 指定在运算表（只当 **Indpnt: Auto** 选中时）中的第一个自变量值（**x**，**θ** 或 **t**）。



TblStart 和 ΔTbl 必须是实数；可以输入表达式。

ΔTbl （运算表步长）指定运算表中相邻的两个自变量值之间的增加或减小步长。

- ◆ 如果 ΔTbl 为正，则当向下滚动运算表时 **x**、**θ** 或 **t** 的值增大。
- ◆ 如果 ΔTbl 为负，则当向下滚动运算表时 **x**、**θ** 或 **t** 的值减小。

Indpnt: Auto 自动从 **TblStart** 开始显示运算表第一列的自变量值。

Indpnt: Ask 显示一个空运算表。当在 **x=** 提示 (**x=value** **ENTER**) 下输入 **x** 值时，每个值填加到自变量列且对应的因变量值计算并显示出来。当 **Ask** 设置时，不能滚动超出当前显示在运算表中的六个自变量。

在 **DifEq** 图象方式下，最好设置 **TblStart** = **tMin** 和 ΔTbl = **tStep**。

在示例中，选择了 $y1=x^2+3x-4$ 和 $y2=\sin(3x)$ 并设置所有默认值。

当在编辑行显示方程时，该列方程名亮显。

查看和编辑因变量方程

- 1

显示运算表。
- 2

将光标移到要编辑因变量所在的列，然后沿该列上移，直到因变量名亮显。
- 3

在编辑行显示方程。
- 4

编辑方程。
- 5

输入已编辑的方程。
 - ◆ 因变量值重新计算。
 - ◆ 光标回到已编辑的因变量第一个值。
 - ◆ 方程编辑器更新。

TABLE F1

⬇ ⬆

ENTER

⬆ ⬆ ⬆ 5 ⬆

+ 1

ENTER

x	y1	y2
0	-4	0
1	0	.14112
2	6	-.279415
3	14	.4121185
4	24	.538573
5	36	.6502878
y1=x^2+3x-4		
TBLST	SELECT	x y

x	y1	y2
0	1	0
1	7	.14112
2	15	-.279415
3	25	.4121185
4	37	.538573
5	51	.6502878
y1=1		
TBLST	SELECT	x y

清除运算表

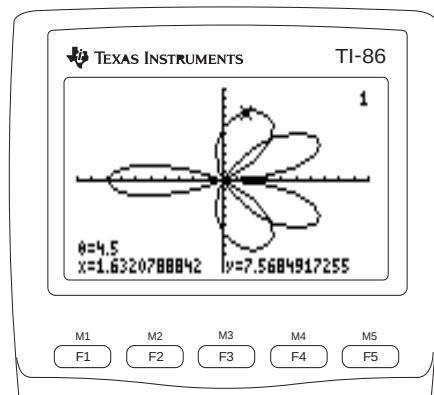
当在程序中使用 CItbl 时，运算表在程序执行时被清除(第16章)。

要在 Indpnt: Ask 设置时清除运算表，从 CATALOG 选择 CItbl，然后再按 **ENTER**。所有自变量和因变量列被清除。CItbl 在 Indpnt: Auto 设置时什么也不做。

8

极坐标图象

预览：极坐标图象.....	116
定义极坐标图象.....	117
在 Pol（极坐标）图象方式下使用 图象工具.....	119



预览：极坐标图象

极坐标方程 $A \sin(B\theta)$ 的图象构成了花瓣的形状。下面绘制方程在 $A=8$ 和 $B=2.5$ 时的图象，然后研究在 A 和 B 为其他值时花瓣的形状。

- 1 从方式屏幕选择 **Pol** 方式。
- 2 显示方程编辑器和极坐标方程编辑器菜单。
- 3 （如果有的话，取消选定或删除所有方程。）存储 **r1(θ)=8sin(2.5θ)**。
- 4 从 **GRAPH ZOOM** 菜单选择 **ZSTD**。
r1 被绘制在图象屏幕上。
- 5 显示窗口编辑器，然后再将 **θMax** 修改为 **4π**。
- 6 从 **GRAPH ZOOM** 菜单选择 **ZSQR**。
xMin 和 **xMax** 改变以便按正确的比例显示图象。
- 7 改变 A 和 B 的值并重新显示图象。

要从图象屏幕删除 **GRAPH** 菜单，如图所示，按 **CLEAR**。

要重新显示 **GRAPH** 菜单，按 **GRAPH**。

[2nd] [MODE] [] [] []
[] [] [ENTER]

[GRAPH] [F1]

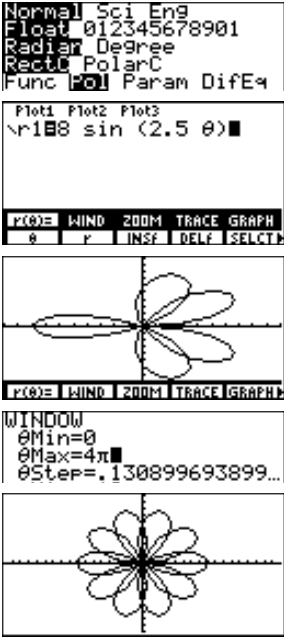
([MORE] [F2] [MORE])
8 [SIN] [] 2 [] 5 [F1] []

[2nd] [M3] [F4]

[F2]
[] 4 [2nd] [π]

[F3] [MORE] [F2]

[F1] (输入其他的 A 和 B 值)



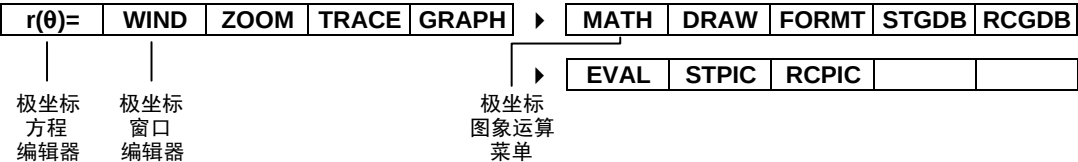
定义极坐标图象

定义极坐标图象的步骤与定义函数图象的步骤相似。本章假定您已熟悉了第 5 章：函数图象和第 6 章：图象工具。第 8 章详细描述极坐标图象与函数图象的差异。

设置极坐标图象方式

要显示方式屏幕，按 **2nd** **[MODE]**。要绘制极坐标方程，必须在输入方程、设置格式或编辑窗口变量值之前选择 **Pol** 图象方式。TI-86 为每种图象方式分别保留方程、格式和窗口数据。

GRAPH（图象）菜单 **[GRAPH]**



第 5 章描述了以下 GRAPH 菜单项：
GRAPH 和 FORMT。

第 6 章描述了以下 GRAPH 菜单项：
ZOOM、TRACE、DRAW、STGDB、RCGDB、EVAL、STPIC 和 RCPIC。

显示极坐标方程编辑器

要显示极坐标方程编辑器，在 **Pol** 图象方式 (**GRAPH** **F1**) 下从 **GRAPH** 菜单选择 **r(θ)=**。除了用 **θ** 和 **r** 替换 **x** 和 **y** 外，显示在底行的极坐标方程编辑器菜单与 **Func** 方式方程编辑器菜单一样。

在此编辑器中，如果内存足够，可以最多输入和显示 99 个极坐标方程，即 **r1** 到 **r99**。方程的自变量定义为 **θ**。

在 **Pol** 图象方式下默认的图象样式是  (实线)。 (上阴影)和  (下阴影) 图象样式不能用于 **Pol** 图象方式。

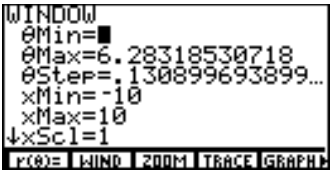


设置图象屏幕窗口变量

要显示极坐标窗口编辑器，从 **GRAPH** 菜单 (**GRAPH** **F2**) 选择 **WIND**。**Pol** 图象方式与 **Func** 图象方式有相同的窗口变量，但除了以下变量：

- ◆ **xRes** 不能用于 **Pol** 图象方式。
- ◆ **θMin**、**θMax** 和 **θStep** 可用于 **Pol** 图象方式。

显示在右侧图形中的数值是 **Radian** 方式下的默认值。↓表示 **yMin=-10**、**yMax=10** 和 **yScl=1** 超出了屏幕。



θMin=0	指定要在图象屏幕中计算的第一个 θ 值
θMax 的默认值为 2π 。	
θMax=6.28318530718	指定要在图象屏幕中计算的最后一个 θ 值
θStep 的默认值为 $\pi/24$ 。	
θStep=.13089969389957	指定从一个 θ 值到下一个 θ 值的增量

DrawLine 图象格式一般比
DrawDot 图象格式显示更有
 意义的极坐标图象。

设置图象格式

要在 **Pol** 图象方式下显示格式屏幕，从 **GRAPH** 菜单 (**GRAPH** **MORE** **F3**) 选择 **FORMT**。第 5 章描述了格式设置。尽管同样的设置可用于 **Func**、**Pol** 和 **Param** 图象方式，但 **TI-86** 在内存中分别为每种方式保存格式设置。在 **Pol** 图象方式下，**PolarGC** 以 **r** 和 **θ** 的形式显示光标坐标，**r** 和 **θ** 是定义方程的变量。

显示图象

要绘制所选定的极坐标方程，可以从 **GRAPH** 菜单选择 **GRAPH**、**TRACE**、**EVAL**、**RCGDB** 或 **ZOOM**、**MATH**、**DRAW** 或 **RCPIC** 操作。**TI-86** 对每个 **θ** 值计算 **r**（从 **θ Min** 到 **θ Max**，间隔为 **θ Step**）然后绘制每个点。在图象绘制过程中变量 **θ** 、**r**、**x** 和 **y** 的值不断更新。

在 **Pol**（极坐标）图象方式下使用图象工具

自由移动光标

自由移动光标在 **Pol** 图象中的工作方式与在 **Func** 图象中的相同。

- ◆ 在 **RectGC** 格式下，移动光标更新 **x** 和 **y** 的值；如果选定了 **CoordOn** 格式，则显示 **x** 和 **y**。
- ◆ 在 **PolarGC** 格式下，移动光标更新 **x**、**y**、**r** 和 **θ** 的值；如果选定了 **CoordOn** 格式，则显示 **r** 和 **θ** 。

跟踪极坐标方程

要开始跟踪，从 GRAPH 菜单（按 **GRAPH** **F4**）选择 **TRACE**。跟踪光标出现在第一个被选中方程的 **θMin** 处。

- ◆ 在 **RectGC** 格式下，移动跟踪光标更新 **θ**、**x** 和 **y** 值；如果选定了 **CoordOn** 格式，则显示 **θ**、**x** 和 **y**。
- ◆ 在 **PolarGC** 格式下，移动跟踪光标更新 **x**、**y**、**r** 和 **θ**；如果选定了 **CoordOn** 格式，则显示 **r** 和 **θ**。

要移动跟踪光标...	按:
以 θStep 为增量或减量沿着方程图象	 或 
由一个方程到另一个方程	 或 

快速缩放可用于 **Pol** 图象下；
平移则不可用（第 6 章）。

如果将跟踪光标移出了图象屏幕的顶部或底部，则屏幕底部的坐标值继续相应地改变。

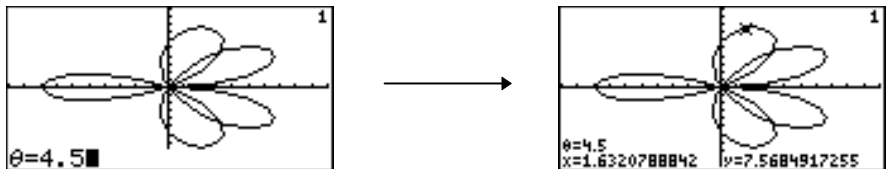
如果已经绘制了一族曲线，则在移到下一个极坐标方程之前可用  和  遍历每条曲线。

将跟踪光标移到 θ 值

要将跟踪光标移到当前方程上任意有效的 θ 值，输入该数值。当输入第一个数字时，在左下角显示 $\theta=$ 提示。输入的值对当前图象屏幕必须有效。当完成输入后按 **ENTER** 重新激活跟踪光标。

在示例中将绘制 $r1=8\sin(2.5\theta)$ 的图象。

在右边的图象上显示了 θ 、 x 和 y 的值，这是因为选择的是 RectGC 图象格式。



使用缩放操作

GRAPH ZOOM 的菜单项，除 ZFIT 之外，在 Pol 图象中的工作方式与在 Func 图象中的相同。在 Pol 图象方式下，ZFIT 在 x 和 y 两个方向调整图象屏幕。

缩放操作只影响 x 窗口变量 ($xMin$ 、 $xMax$ 和 $Xscl$) 和 y 窗口变量 ($yMin$ 、 $yMax$ 和 $yScl$)，但 ZSTO 和 ZRCL 除外，它们也影响 θ 窗口变量 (θMin 、 θMax 和 $\theta Step$)。

GRAPH MATH（图象运算）菜单

GRAPH MORE F1

MATH	DRAW	FORMT	STGDB	RCGDB
DIST	dy/dx	dr/dθ	ARC	TANLN

其他 GRAPH MATH 菜单项
和第 6 章中所描述的一样。

dr/dθ 求函数在一点处的数值导数（斜率）

由 DIST 和 ARC 计算的距离是直角坐标平面上的距离。**dy/dx** 和 **dr/dθ** 与 RectGC 或 PolarGC 格式无关。

在导数未被定义的点处，**TANLN** 将绘制一条直线，但没有结果显示或存储到 **Ans**。

对给定的 θ 求解方程

当跟踪光标未激活时，GRAPH 的菜单项 **EVAL** 对给定的 **θ** 值在图象上直接求解选定的极坐标方程。程序或主屏幕中的 **eval** 返回 **r** 值的数组。

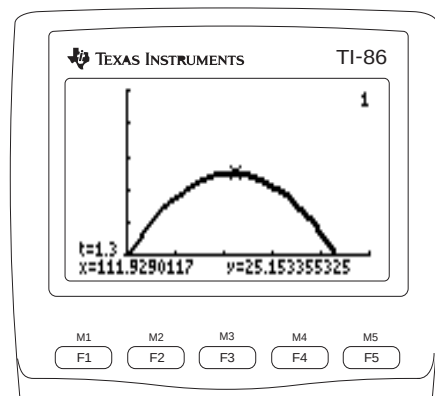
在极坐标图象上绘图

GRAPH DRAW 菜单项在 **Pol** 图象中的工作方式与在 **Func** 图象中的相同。在 **Pol** 图象方式下的 DRAW 指令坐标是指图象屏幕的 **x** 和 **y** 坐标。**DrInv** 不能用于 **Pol** 图象方式。

9

参数 图象

预览：参数图象.....	124
定义参数图象.....	125
在参数图象方式下使用图象工具	128



预览：参数图象

下面绘制一个参数方程的图象，该方程描述一个抛掷初速度为 30 米/秒、初始角度与水平面（地平面）呈 25 度小球的飞行轨迹。问小球将飞行多远？何时触地？飞行多高？

- ❶ 从方式屏幕中选择 **Param** 方式。

$\boxed{2\text{nd}} \boxed{\text{MODE}} \boxed{\downarrow} \boxed{\downarrow} \boxed{\downarrow}$
 $\boxed{\downarrow} \boxed{\rightarrow} \boxed{\rightarrow} \boxed{\text{ENTER}}$

```
Normal Sci Eng
Float 012345678901
Radian Degree
RectO PolarC
Func Pol Param DfE=
```

- ❷ 显示方程编辑器和参数方程编辑器菜单。取消选定所有方程和统计图（如果已定义）。

$\boxed{\text{GRAPH}} \boxed{\text{F1}}$
 $\boxed{\text{(MORE)}} \boxed{\text{F2}} \boxed{\text{(MORE)}}$

```
Plot1 Plot2 Plot3
\xt1=30t cos (25°)
yt1=30t sin (25°)-9...
\xt2=
yt2=
```

- ❸ 定义小球的轨迹为时间 **t** 的函数 **xt1** 和 **yt1** 。

水平： $xt1 = tv_0 \cos(\theta)$

垂直： $yt1 = tv_0 \sin(\theta) - 1/2(gt^2)$

重力常数： $g = 9.8 \text{ 米/秒}^2$

$30 \boxed{\text{F1}} \boxed{\text{COS}} \boxed{\boxed{\text{)}}} \boxed{25}$
 $\boxed{2\text{nd}} \boxed{\text{MATH}} \boxed{\text{F3}} \boxed{\text{F1}}$
 $\boxed{\boxed{\text{)}}} \boxed{\downarrow} \boxed{30} \boxed{2\text{nd}} \boxed{\text{M1}}$
 $\boxed{\text{SIN}} \boxed{\boxed{\text{)}}} \boxed{25} \boxed{\text{F1}} \boxed{\boxed{\text{)}}} \boxed{-}$
 $9 \boxed{-} \boxed{8} \boxed{\div} \boxed{2} \boxed{2\text{nd}} \boxed{\text{M1}}$
 $\boxed{x^2} \boxed{\downarrow}$

t	xt	yt	DELF	SELC
0	r	i	►DMS	

- ❹ 将垂直分量向量定义为 **xt2** 和 **yt2** 以及将水平分量向量定义为 **xt3** 和 **yt3** 。

$0 \boxed{\downarrow} \boxed{2\text{nd}} \boxed{\text{M3}} \boxed{1} \boxed{\downarrow}$
 $\boxed{2\text{nd}} \boxed{\text{M2}} \boxed{1} \boxed{\downarrow} \boxed{0}$

```
Plot1 Plot2 Plot3
yt1=30t sin (25°)-9...
\xt2=0
yt2=yt1
\xt3=xt1
yt3=0
```

t	xt	yt	DELF	SELC
0	r	i	►DMS	

- ❺ 将 **xt3/yt3** 的图象样式改变为 --- （粗线）。将 **xt2/yt2** 和 **xt1/yt1** 的图象样式改变为 --- （轨迹）。

$\boxed{\text{EXIT}} \boxed{\text{MORE}} \boxed{\text{F4}} \boxed{\uparrow}$
 $\boxed{\uparrow} \boxed{\text{F4}} \boxed{\text{F4}} \boxed{\uparrow} \boxed{\uparrow}$
 $\boxed{\uparrow} \boxed{\text{F4}} \boxed{\text{F4}}$

```
Plot1 Plot2 Plot3
\xt1=30t cos (25°)
yt1=30t sin (25°)-9...
\xt2=0
yt2=yt1
\xt3=xt1
```

EQN=	WIND	ZOOM	TRACE	GRAPH
INSE	ALL+	ALL-	STWIR	

在示例中忽略了除重力之外的所有力。对于初始速度 v_0 和角度 θ ，小球的位置是时间的函数，有水平和垂直两个分量。

- ⑥ 输入下面的窗口变量值。

tMin=0 xMin=-20 yMin=-5

tMax=5 xMax=100 yMax=15

tStep=.1 xScl=50 yScl=10

[2nd] [M2] 0 ▾ 5 ▾

▾ 1 ▾ (-) 20 ▾

100 ▾ 50 ▾ (-) 5

▾ 15 ▾ 10

```
WINDOW
↑tMin=-20
xMax=100
xScl=50
yMin=-5
yMax=15
yScl=10
F(t)= [WIND] [ZOOM] [TRACE] [GRAPH]
```

- ⑦ 设置 **SimulG** 和 **AxesOff** 图象格式，这样小球的轨迹和向量将同时绘制在一个已清空图象屏幕上。

[MORE] [F3] ▾ ▾ ▾

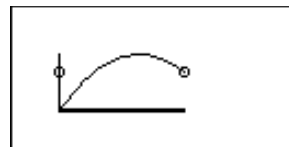
▸ [ENTER] ▾ ▾ ▾

[ENTER]

```
RectGC PolarGC
CoordOn CoordOff
DrawLine DrawDot
SeqG SimulG
GridOff GridOn
AxesOn AxesOff
LabelOff LabelOn
F(t)= [WIND] [ZOOM] [TRACE] [GRAPH]
```

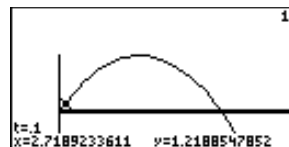
- ⑧ 绘制图象。绘制操作同时显示飞行中的小球和运动的垂直和水平分量向量。

[F5]



- ⑨ 跟踪图象来获得数值结果。跟踪从 **tMin** 处开始，按时间跟踪球的轨迹。显示的 **x** 值表示距离；**y** 值表示高度；**t** 值表示时间。

[F4] ▸



定义参数图象

定义参数图象的步骤与定义函数图象的步骤类似。本章假定您已经熟悉了第5章：函数图象和第6章：图象工具。本章详述参数图象与函数图象的不同之处。

要模拟飞行中的小球，将 **xt1/yt1** 的图象样式改变为  (动画)。

第 5 章描述了下面的 GRAPH 菜单项：
GRAPH 和 FORMT。

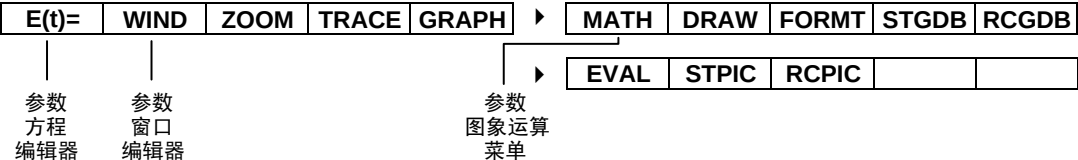
第 6 章描述了下面的 GRAPH 菜单项：
ZOOM、TRACE、DRAW、STGDB、RCGDB、EVAL、STPIC 和 RCPIC。

参数图象常用的应用是绘制
随时间而变化方程的图象。

设置参数图象方式

要显示方式屏幕，按 [2nd] [MODE]。要绘制参数方程，必须在输入方程、设置格式或编辑窗口变量值之前选择 **Param** 图象方式。TI-86 在内存中为每种图象方式分别保留方程、格式和窗口数据。

GRAPH（图象）菜单 [GRAPH]



显示参数方程编辑器

要显示参数方程编辑器，在 **Param** 图象方式 ([GRAPH] [F1]) 下从 GRAPH 菜单选择 **E(t)=**。除了用 **t** 和 **xt** 替换 **x** 和 **y**、**yt** 替换 **INSf** 外，显示在底行的方程编辑器菜单与 **Func** 方式方程编辑器菜单相同。

在此编辑器中，如果内存足够，可以输入并显示最多 99 个参数方程的 **x** 和 **y** 分量，即 **xt1** 和 **yt1** 到 **xt99** 和 **yt99**。每个分量的自变量都定义为 **t**。

x 和 **y** 两个分量定义一个参数方程。必须为每个方程都定义 **xt** 和 **yt**。**Param** 方式下默认的图象样式是 （实线）。图象样式 （上阴影）和 （下阴影）不能用于 **Param** 方式。



选择和取消选定参数方程

当参数方程被选定时, **xt** 和 **yt** 的等号 (=) 亮显。要改变参数方程的选择状态, 将光标移到 **xt** 或 **yt** 上, 然后从方程编辑器菜单选择 **SELECT**。**xt** 和 **yt** 的状态改变。

删除参数方程

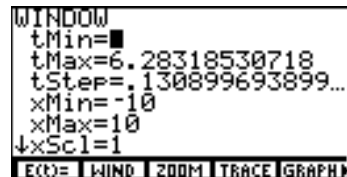
要用 **DELf** 删除参数程, 将光标移到 **xt** 或 **yt** 上, 然后从方程编辑器菜单选择 **DELf**。两个分量都被删除。

要用 **MEM DELET** 菜单 (第 17 章) 删除参数方程, 必须选择 **xt** 分量。如果选择 **yt** 分量, 方程仍保留在内存中。

设置图象屏幕窗口变量

要显示参数窗口编辑器, 从 **GRAPH** 菜单 (**GRAPH** **F2**) 选择 **WIND**。**Param** 图象方式与 **Func** 图象方式窗口变量基本相同, 但以下变量除外:

- ◆ **xRes** 不能用于 **Param** 方式。
- ◆ **tMin**、**tMax** 和 **tStep** 可用于 **Param** 方式。



在右侧图形中显示的数值是 **Radian** 方式的默认值。 $\downarrow$ 表示 **yMin=-10**、**yMax=10** 和 **yScl=1** 超出了屏幕。

tMin=0	指定起始 t 值
tMax 的默认值为 2π 。	tMax=6.28318530718 指定结束 t 值
tStep 的默认值为 $\pi/24$ 。	tStep=.13089969389957 指定 t 从一个值到下一个值的增量

DrawLine 图象格式一般比
DrawDot 图象格式显示更有
意义的参数图象。

设置图象格式

要在 **Param** 图象方式下显示格式屏幕，从 **GRAPH** 菜单 (**GRAPH** **MORE** **F3**) 选择 **FORMAT**。第 5 章描述了格式设置。**TI-86** 在内存中分别为 **Func**、**Pol**、**Param** 和 **DifEq** 图象方式保留了格式设置。

显示图象

要绘制定选定的参数方程，可以选择 **GRAPH**、**TRACE**、**EVAL**、**RCGDB** 或 **ZOOM**、**MATH**、**DRAW** 或 **RPIC** 操作。**TI-86** 为每个 **t** 值计算 **x** 和 **y**（从 **tMin** 到 **tMax**，间隔为 **tStep**）的值，然后绘制每个由 **x** 和 **y** 定义的点。在绘制图象的过程中更新变量 **x**、**y** 和 **t** 的值。

在参数图象方式下使用图象工具

自由移动光标

自由移动光标在 **Param** 图象下的工作方式与在 **Func** 图象下相同。

- ◆ 在 **RectGC** 格式下，移动光标时更新 **x** 和 **y** 的值。如果选择了 **CoordOn** 格式，则显示 **x** 和 **y** 的值。
- ◆ 在 **PolarGC** 格式下，移动光标时更新 **x**、**y**、**r** 和 θ 的值；如果选择了 **CoordOn** 格式，则显示 **r** 和 θ 的值。

跟踪参数函数

要开始跟踪，从 **GRAPH** 菜单 (**GRAPH** **F4**) 选择 **TRACE**。跟踪开始时，跟踪光标位于第一个选定函数的 **tMin** 处。

- ◆ 在 **RectGC** 格式下，移动跟踪光标更新 **x**、**y** 和 **t** 的值；如果选择了 **CoordOn** 格式，则显示 **t**、**x** 和 **y** 的值。

◆ 在 **PolarGC** 格式下，移动跟踪光标更新 **x**、**y**、**r**、**θ** 和 **t** 的值；如果选择了 **CoordOn** 格式，则显示 **r**、**θ** 和 **t** 的值。**x** 和 **y**（或 **r** 和 **θ**）的值由 **t** 计算得到。

要移动跟踪光标...

按：

以 **tStep** 增量或减量沿方程图象
由一个方程到另一个方程

▸ 或 ▹
▾ 或 ▴

快速缩放可用于 **Param** 图象；
平移则不能（第 6 章）。

如果将跟踪光标移出了图象屏幕的顶部或底部，屏幕底部的坐标值继续相应地变化。如果已经绘制了一族曲线，则在移到下一个参数函数之前可用 ▾ 和 ▴ 遍历每条曲线。

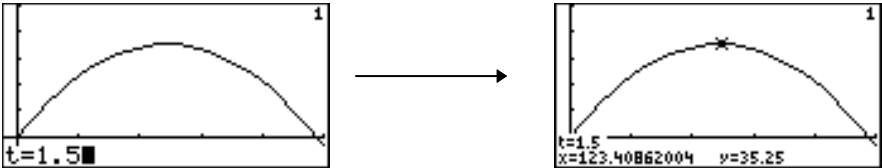
将跟踪光标移到 **t** 值处

可以在 **t=** 提示处输入表达式。

要将跟踪光标移到当前方程上任何有效的 **t** 值处，输入该数值。当输入第一个数字时，在左下角显示 **t=** 提示。输入的值必须对当前图象屏幕有效。当完成输入后按 **[ENTER]** 重新激活跟踪光标。

在示例中，参数方程为：
 $xt1=95t \cos 30^\circ$
 $yt1=95t \sin 30^\circ - 16t^2$
而且，设置 **AxesOn** 图象格式。

（第 124 页中的示例与此示例相似。）



使用缩放操作

除 **ZFIT** 外，**GRAPH ZOOM** 菜单项在 **Param** 图象中的工作方式与在 **Func** 图象中相同。在 **Param** 方式下，**ZFIT** 在 **x** 和 **y** 两个方向调整图象屏幕。

GRAPH ZOOM 菜单项只影响 **x** 窗口变量（**xMin**、**xMax** 和 **xScl**）和 **y** 窗口变量（**yMin**、**yMax** 和 **yScl**），但 **ZSTO** 和 **ZRCL** 除外，后两者还影响 **t** 窗口变量（**tMin**、**tMax** 和 **tStep**）。

GRAPH MATH（图象运算）菜单 **GRAPH** **MORE** **F1**

MATH	DRAW	FORMT	STGDB	RCGDB					
DIST	dy/dx	dy/dt	dx/dt	ARC	▶	TANLN			

其他 **GRAPH MATH** 菜单项
与第 5 章中描述的相同。

- dy/dx** 返回 **yt** 导数除以 **xt** 导数
- dy/dt** 返回 **yt** 方程关于 **t** 的导数
- dx/dt** 返回 **xt** 方程关于 **t** 的导数

由 **DIST** 和 **ARC** 计算出的距离是直角坐标平面上的距离。
在导数未被定义点处，**TANLN** 将绘制一条直线，但没有结果显示或存储到 **Ans**。

对给定 **t** 求解方程

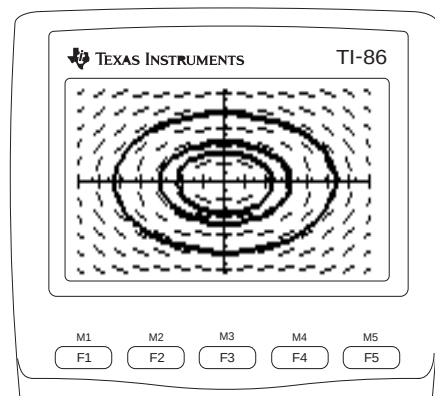
当跟踪光标未激活时，**GRAPH** 的菜单项 **EVAL** 对给定的 **t** 值求解图象中选定的极坐标方程。程序或主屏幕中的 **eval** 返回关于 **x** 和 **y** 数值的数组，格式如下：**{xt1(t) yt1(t) xt2(t) yt2(t) ...}**。

在参数图象上绘图

DRAW 菜单项在 **Param** 图象下的工作方式与在 **Func** 图象下相同。**Param** 图象下的 **DRAW** 指令坐标是指图象屏幕的 **x** 和 **y** 坐标值。

10 微分 方程图象

定义微分方程图象.....	132
输入并求解微分方程	139
在 DifEq（微分方程）图象方式下 使用图象工具.....	144



第 8 章和第 9 章开始于一个示例；而第 10 章中将有几个微分方程示例。

定义微分方程图象

定义微分方程图象的步骤与定义函数图象的步骤相似。本章假定您已经熟悉了第 5 章：函数图象和第 6 章：图象工具。本章详细介绍微分方程图象与函数图象的不同之处。

通常，**DifEq** 图象方式与其他图象方式的不同之处在于：

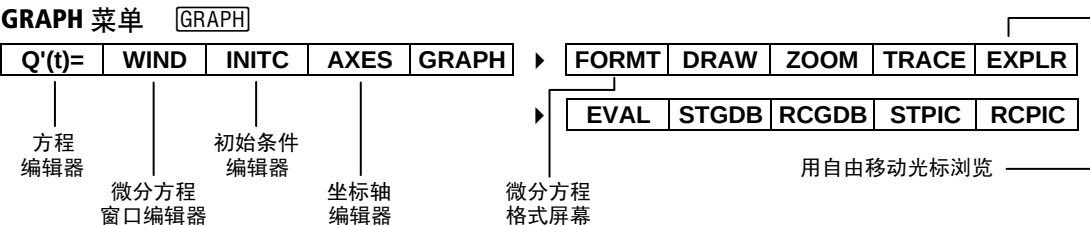
- ◆ 必须在定义方程之前选择域格式或接受默认值（第 133 页）。
- ◆ 如果方程次数高于一阶，必须将它转换为等价的一阶方程组，然后保存在方程编辑器中（第 140 页和第 142 页）。
- ◆ 当选择了 **FldOff** 域格式后，必须设置方程组中每个方程的初始条件（第 136 页）。
- ◆ 选择完域格式设置后，必须从 **GRAPH** 菜单中选择 **AXES**，并输入坐标轴信息或接受默认值（第 137 页）。

设置微分方程图象方式

要显示方式屏幕，按 **[2nd] [MODE]**。要绘制微分方程的图象，必须在设置格式、输入方程或编辑窗口变量值之前选择 **DifEq** 图象方式。TI-86 在内存中保存每种图象方式的格式、方程和窗口数据。

第 5 章描述 GRAPH 菜单项 GRAPH。

第 6 章描述以下 GRAPH 菜单项：
DRAW、ZOOM、TRACE、EVAL、STGDB、RCGDB、STPIC 和 RCPIC。



设置图象格式

要在 DifEq 图象方式下显示格式屏幕，则从 GRAPH 菜单选择 FORMT (GRAPH MORE F1)。

- ◆ RK Euler 和 SlpFld DirFld FldOff 格式设置仅在 DifEq 图象方式下有效。
- ◆ RectGC PolarGC、DrawLine DrawDot 和 SeqG SimulG 格式设置在 DifEq 图象方式下无效。
- ◆ 所有其他的格式设置与第 5 章介绍的相同。



TI-86 为每种图象方式保存各自的格式设置。

求解方法格式

- RK** 用 Runge-Kutta 法解微分方程比用 Euler 求解方法格式更精确，但是速度不如它快
- Euler** 用 Euler 法解微分方程；需要指定在 tStep 值之间的迭代次数，在窗口编辑器中 EStep= 提示代替 difTol= 提示

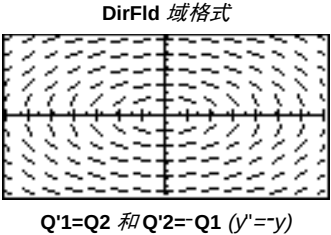
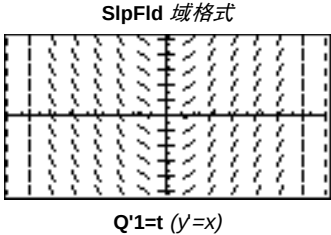
域格式

- SlpFld** (斜率域) 仅将斜率域添加到一阶方程的图象中, 图象的 **x** 轴为 **t**, **y** 轴为指定的 **Q_n** 方程
- DirFld** (方向域) 仅将方向域添加到二阶方程的图象中, 图象的 **x** 轴为 **Q_{x#}**, **y** 轴为 **Q_{y#}**
- FldOff** (关闭域) 绘制所有选定微分方程的图象, 图象的 **x** 轴为 **t** 或 **Q₁**, **y** 轴为 **Q₁** 或 **Q₂**, 并且无域; 必须为所有方程定义初始条件 (第 136 页)

下面的示例显示了基本的斜率和方向域; 所有未指定的设置和数值都为默认值。要复制这些示例, 复位默认值, 在 **DifEq** 图象方式下输入指定的信息, 然后按 **GRAPH** **F5**。

坐标信息保存到变量 **GDB** 和 **PIC**。

要将菜单从图象中删除, 如示例所示, 按 **CLEAR**。



显示微分方程编辑器

要显示微分方程编辑器, 在 **DifEq** 图象方式下 (**GRAPH** **F1**) 从 **GRAPH** 菜单选择 **Q'(t)=**。底行中的 **DifEq** 方程编辑器菜单与 **Func** 方式的方程编辑器菜单基本相同, 除了用 **t** 和 **Q** 代替 **x** 和 **y**。

在此编辑器中, 如果有足够的内存, 可输入并显示最多 9 个一阶微分方程, 即从 $Q'1$ 到 $Q'9$ 。方程按照自变量 t 和/或 Q' 来定义。

在 **DifEq** 方程中可以引用另一个微分方程的变量, 例如 $Q'2=Q1$ 。但是不能在 **DifEq** 方程中输入数组。



当 TI-86 计算一个微分方程组时, 它从 $Q'1$ 开始引用方程编辑器中的所有方程, 而不管其选择状态如何。必须从 $Q'1$ 开始连续定义 $Q'n$ 方程变量。例如, 如果 $Q'1$ 和 $Q'2$ 未定义, 但试图去求解 $Q'3$ 中定义的方程, 计算器将返回错误。

TI-86 允许单独分析每个方程。例如, 可以输入 $Q'1=t$ 和 $Q'2=t^2$ 并单独分析每个方程。

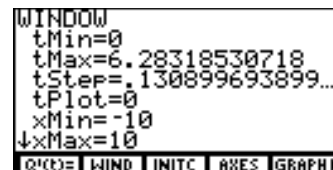
TI-86 只绘制那些选定的并适合于指定坐标轴的方程图象。

- ◆ **DifEq** 方式下默认的图象样式是 thick (粗线)。
- ◆ thin (上阴影)、 dashed (下阴影) 和 dots (点画线) 在 **DifEq** 图象方式下无效。

设置图象屏幕窗口变量

要显示微分方程窗口编辑器, 从 **GRAPH** 菜单 (**GRAPH** **F2**) 选择 **WIND**。**DifEq** 与 **Func** 图象方式窗口变量基本相同, 除了:

- ◆ **xRes** 在 **DifEq** 方式下无效。
- ◆ **tMin**、**tMax**、**tStep** 和 **tPlot** 在 **DifEq** 方式下有效。
- ◆ **difTol** (RK) 和 **EStep** (Euler) 在 **DifEq** 方式下有效。



第 135 页图形中所显示的值是弧度 **Radian** 方式下的默认值。**x** 和 **y** 的设置与坐标轴变量（第 137 页）一致。 $\downarrow$ 表示 **xScl=1**、**yMin=-10**、**yMax=10**、**yScl=1** 和 **difTol=.001**（在 **RK** 格式中）或 **EStep=1**（在 **Euler** 格式中）超出了屏幕。

tMax 默认值为 2π 。

tStep 默认值为 $\pi/24$

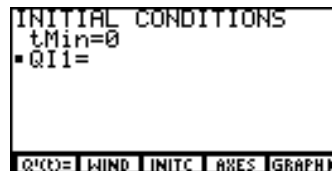
tMin=0	指定在图象屏幕内开始计算的 t 值
tMax=6.28318530718	指定在图象屏幕内最后计算的 t 值
tStep=.1308969389958	指定从一个 t 值到下一个 t 值的增量
tPlot=0	指定开始绘制的点（当 t 是坐标轴时被忽略）
difTol=.001 （在 RK 格式中）	指定误差以帮助选择求解步长大小，必须 $\geq 1E-12$
EStep=1 （在 Euler 格式中）	指定在 tStep 值之间的 Euler 迭代次数，必须是一个 >0 且 ≤ 25 的整数

设置初始条件

初始条件信息保存到变量 **GDB** 和 **PIC** 中。

要显示初始条件编辑器，从 **GRAPH** 菜单 (**GRAPH** **F3**) 选择 **INITC**。在这个编辑器中，可以给方程编辑器中的每个一阶方程设置 **t=tMin** 时的初值。

tMin 是要计算的第一个 **t** 值。**QI1** 是 **Qn** 的初值。一小方块显示在初始条件变量旁边表示所定义的微分方程需要输入一个值。



可以给初始条件 **tMin** 和 **QIn** 输入表达式、数组或数组名。当输入数组名时，数组元素在按 **ENTER**、 $\downarrow$ 或 $\uparrow$ 之后显示。

- ◆ 如果设置了 **SlpFld** 或 **DirFld** 格式，则不必指定初始条件。**TI-86** 返回相应不包含特定解的域。
- ◆ 如果设置了 **FldOff** 格式，则必须指定初始条件。

设置坐标轴

要显示坐标轴编辑器, 在 **DifEq** 方式 (**GRAPH** **F4**) 下从 **GRAPH** 菜单选择 **AXES**。

x= 将变量赋给 **x** 轴

dTime= 指定一个时间点 (实数)

y= 将变量赋给 **y** 轴

fldRes= (分辨率) 设置行数 (1 到 25)

在 **x=** 和 **y=** 提示处, 可以输入自变量 **t**, 以及 **Q**、**Q'**、**Q_n** 或 **Q'_n**, 其中 **n** 是 ≥ 1 且 ≤ 9 的整数。如果将 **t** 赋给一个坐标轴, 并将 **Q_n** 或 **Q'_n** 赋给另一个坐标轴, 则只绘制保存在 **Q_n** 或 **Q'_n** 中的方程; 不绘制方程编辑器中的其他微分方程; 这些方程的选择状态被忽略。**dTime** 只对这样的二阶方程有效, 变量 **t** 出现在对应一阶方程组的任意一个方程中。

坐标轴编辑器和每个域格式的默认值显示在下面。当设置 **SlpFld** 域格式时, **x** 轴总是 **t**, 所以 **AXES: SlpFld** 编辑器不显示 **x=t**。

坐标轴信息保存到变量
GDB 和 **PIC** 中。

设置 **SlpFld** 格式时:

```
AXES: SlpFld
y=Q1
fldRes=15

Q'(t)= WIND INITC AXES GRAPH
Q
```

设置 **DirFld** 格式时:

```
AXES: DirFld
x=Q1
y=Q2
dTime=0
fldRes=15

Q'(t)= WIND INITC AXES GRAPH
Q
```

设置 **FldOff** 格式时:

```
AXES: FldOff
x=t
y=Q

Q'(t)= WIND INITC AXES GRAPH
Q t Q'
```

微分方程绘制提示

- ◆ 由于 **TI-86** 绘制方程前先绘制斜率和方向域, 可以按 **[ENTER]** 暂停作图, 查看没有绘制解的域。
- ◆ 如果不为指定到坐标轴的方程确定初始条件, **TI-86** 将简单地绘制域后就停止。这允许同时访问域格式选项和交互初始条件。

统计图和屏幕绘图不保存到变量 **fldPic** 中。

内置变量 **fldPic**

当 TI-86 绘制域时, 它将域和所有显示标记、坐标轴或光标位置等信息保存到内置变量 **fldPic** 中。

以下操作不更新 **fldPic** :

- ◆ 将求解方法格式从 **RK** 切换到 **Euler** 或从 **Euler** 切换到 **RK**
- ◆ 输入或编辑任意初始条件变量的值 (从 **QI1** 到 **QI9**)
- ◆ 编辑 **difTol**、**EStep**、**tMin**、**tMax**、**tStep** 或 **tPlot** 的值
- ◆ 改变图象样式

以下操作更新 **fldPic** :

- ◆ 编辑方程编辑器中的方程
- ◆ 给坐标轴重新赋值, 编辑 **dTime** 值或编辑 **fldRes** 值
- ◆ 使用 **GRAPH ZOOM** 菜单项
- ◆ 改变格式设置而不是求解方法格式
- ◆ 编辑 **xMin**、**xMax**、**xScl**、**yMin**、**yMax** 或 **yScl** 的值

显示图象

要绘制微分方程, 可以从 **GRAPH** 菜单中选择 **GRAPH**、**TRACE**、**EVAL** 或 **STGDB**, 或 **DRAW**、**ZOOM** 或 **STPIC** 操作。TI-86 从 **tMin** 到 **tMax** 求解每个方程。如果 **t** 不是坐标轴, 则从 **tPlot** 开始绘制每个点; 否则从 **tMin** 开始绘制。在绘制图象的过程中变量 **x**、**y**、**t** 和 **Qn** 被更新。

tStep 影响跟踪的分辨率和图象外观, 但不影响跟踪值的精度。 **tStep** 不决定求解的步长大小; 使用 **RK** 算法 (Runge-Kutta 2-3) 决定步长大小。如果 **x** 轴是 **t**, 设置 **tStep** < (**tMax** - **tMin**)/126 将增加绘制时间, 而不增加精度。

输入并求解微分方程

在 **Func** 图象方式下，**x** 是自变量，**y** 是方程变量。为避免 TI-86 上的 **Func** 方程和 **DifEq** 方程之间的冲突，在 **DifEq** 图象方式下 **t** 是自变量，**Q'n** 是方程变量。因此，在微分方程编辑器中输入方程时，必须将它表达为 **t** 和 **Q'n** 的形式。

例如，要表示一阶微分方程 $y'=x^2$ ，则要用 t^2 代替 x^2 ，用 **Q'1** 代替 y' ，然后在方程输入编辑器中输入 **Q'1=t²**。

在 SlpFld 格式下绘制

❶ 显示方式屏幕并设置 **DifEq** 图象方式。

[2nd] [MODE] ▾ ▾ ▾
▾ ▶ ▶ ▶ ▶ [ENTER]

❷ 显示格式屏幕并设置 **SlpFld** 域格式。

[GRAPH] [MORE] [F1] ▾
▾ ▾ ▾ ▾ [ENTER]

❸ 显示方程编辑器并保存微分方程 $y'=x^2$ ，用 **Q'1** 代替 y' ，用 **t** 代替 x 。清除所有其他方程。

[F1] [F1] x²

❹ 显示初始条件编辑器并输入初始条件。
小方块表示这个初始条件是需要

[2nd] [M3] 3



在示例中，开始已设置默认窗口变量值。

在SlpFld 域格式中， $x=t$ 总为真； $y=Q1$ 和 $fldRes=15$ 是默认的坐标轴设置。

- 5

显示坐标轴编辑器并输入要求解的方程变量。（不要设置 $y=Q$ ）

[F4] [F1] 1
- 6

接受或改变 **fldRes** （分辨率）。
- 7

显示图象。使用默认的窗口变量值时，这个图象中的斜率域并不能说明什么。

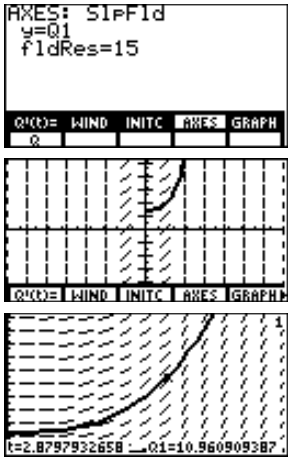
[2nd] [M5]
- 8

改变窗口变量 **xMin** 、 **xMax** 、 **yMin** 和 **yMax** 。

[F2] [v] [v] [v] [v] 0 [v]
5 [v] [v] 0 [v] 20
- 9

从 GRAPH 菜单选择 **TRACE** 重新绘制图象并激活跟踪光标。跟踪结果。显示 **t** 和 **Q1** 的跟踪光标坐标。

[MORE] [F4]
[▶] 和 [◀]



将方程转换为一阶方程组

在 TI-86 上，要输入二阶或更高阶（最高为 9 阶）的微分方程，则必须将它转换为一阶方程组。例如，要输入二阶微分方程 $y'' = -y$ ，则必须将它转换为两个一阶方程，如下表所示：

微分...	将变量定义为...	然后代入：
$Q'1=y'$	$Q1=y$	$Q'1=Q2$ （因为 $Q'1=y'=Q2$ ）
$Q'2=y''$	$Q2=y'$	$Q'2=-Q1$

在 DirFld 格式下绘制

- ① 显示方式屏幕并设置 DifEq 图象方式。

2nd [MODE] $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ ENTER

Func Pol Param DIF=2
Vec Bin Oct Hex
Rect CylV SphereV

- ② 显示格式屏幕并设置 DirFld 图象格式。

GRAPH MORE F1 $\downarrow$
 $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ ENTER

SlrFld DIF=1C FldOff
Q'(t)= WIND INITC AXES GRAPH

- ③ 显示方程编辑器并将由 $y''=-y$ 转换得到的微分方程组保存到方程编辑器中, 用 Q1 代入 y , 用 Q2 代入 y' 。

F1 F2 2 $\downarrow$ (-) F2 1

Plot1 Plot2 Plot3
Q'1=Q2
Q'2=-Q1

- ④ 显示初始条件编辑器并输入初始条件, 如果需要一个特定的解。要输入初始条件的数组, 则从 LIST 菜单中使用 { 和 }。

2nd [M3] 2nd [LIST] F1
1 2 5 F2 $\downarrow$ F1
2nd [π] 4 5 $\downarrow$
75 F2

Q'(t)= WIND INITC AXES GRAPH
t Q INSP DELF SELCT

INITIAL CONDITIONS
tMin=0
QI1={1,2,5}
QI2={π,4,5.75}

- ⑤ 显示坐标轴编辑器并输入要求解的两个方程变量。必须省略记号 (')。

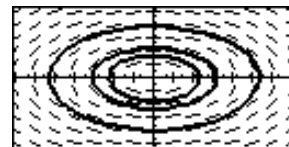
2nd [M4] F1 1 $\downarrow$
F1 2

Q'(t)= WIND INITC AXES GRAPH
< > NAMES EDIT OPS

AXES: DirFld
x=Q1
y=Q2
dTime=0
fldRes=15

- ⑥ 接受或改变 fldRes (分辨率)。

EXIT MORE F3 F4



- ⑦ 从 GRAPH ZOOM 菜单选择 ZSTD 来设置标准窗口变量并显示图象。

CLEAR

- ⑧ 从屏幕上清除 GRAPH 菜单。

在 DifEq 图象方式下, t 是自变量, Q^n 是因变量, 其中 $n \geq 1$ 且 ≤ 9 。

在示例中, 开始已设置默认窗口变量值。

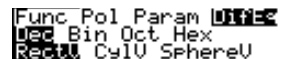
如果选择 DirFld 域格式, 则 $x=Q1$ 、 $y=Q2$ 、 $dTime=0$ 和 $fldRes=15$ 是坐标轴的默认设置。
因为 t 不是方程的一部分, $dTime$ 被忽略。

在 FldOff 格式下绘制方程组

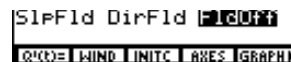
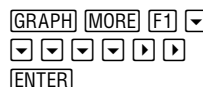
在这个示例中, 必须将 4 阶微分方程 $y^{(4)}-y=e^{-x}$ 转换等价的一阶微分方程组, 如下表所示:

微分...	将变量定义为...	然后代入:
$Q'1=y'$	$t=x$	$Q'1=Q2$ (因为 $Q'1=y'=Q2$)
$Q'2=y''$	$Q1=y$	$Q'2=Q3$
$Q'3=y'''$	$Q2=y'$	$Q'3=Q4$
$Q'4=y^{(4)}$	$Q3=y''$	$Q'4=e^{-t}+Q1$ (因为 $Q'4=y^{(4)}=e^{-x}+y=e^{-t}+Q1$)
	$Q4=y'''$	

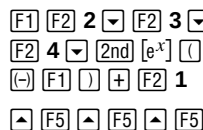
- ① 显示方式屏幕并设置 DfEq 图象方式。



- ② 显示格式屏幕并设置 FldOff 域格式。



- ③ 显示方程编辑器并保存由 $y^{(4)}=e^{-x}+y$ 转换成的微分方程组, 按表中所示的代入。



- ④ 取消选择 $Q'3$ 、 $Q'2$ 和 $Q'1$, 只绘制 $Q'4=e^{-(t)}+Q1$ 。

在 DfEq 图象方式下, t 是自变量, $Q'n$ 是因变量, 其中 $n \geq 1$ 且 ≤ 9 。

- 5 显示窗口编辑器并设置窗口变量值。

2nd [M2] 10 .
01 0
(-) 4 4

```
WINDOW
tMin=0
tMax=10
tStep=.01
tPlot=0
xMin=0
xMax=10
xScl=1
yMin=-4
yMax=4
yScl=1
difTol=.001
Q(t)=L WIND INITC AXES GRAPHM
```

- 6 显示初始条件编辑器并输入初始条件。
小方块表示这个初始条件是需要。

F3 3 (-) 5 25
7 5
(-) 5 75

```
INITIAL CONDITIONS
tMin=0
Q11=3
Q12=-5.25
Q13=7.5
Q14=-5.75
```

- 7 显示坐标轴编辑器。输入要求解的方程
变量。

F4

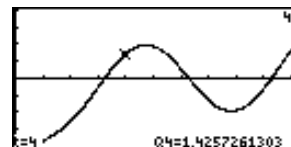
```
AXES: FldOff
x=t
y=Q
```

- 8 显示图象。用跟踪光标浏览方程。

EXIT MORE F4
和

- 9 输入 t 值, 跟踪光标移动到对应 t 值的结果处。 t 和 $Q4$ 的坐标值显示。

4 ENTER



如果设置 FldOff 域格式, 则
 $x=t$ 和 $y=Q$ 是坐标轴的默认
设置。

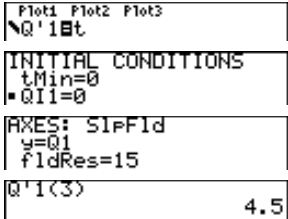
要将 **Q'1** 粘贴到主屏幕上，
可以从 **CHAR MISC** 菜单
或 **CATALOG** 中选择它。

由于 TI-86 的系统要求，
必须在计算器中将 **Q1(3)**
表示为 **Q'1(3)**。

对指定值求解微分方程

在 **DifEq** 图象方式下的主屏幕上，可以求解保存在指定自变量或
表达式中的微分方程。语法为：**Q'n(value)**。

- ◆ 方程必须保存到 **DifEq** 方程变量（从 **Q'1** 到 **Q'9**）。
- ◆ 必须定义初始条件。
- ◆ 结果有时会不同，取决于坐标轴设置。



在 **DifEq**（微分方程）图象方式下使用图象工具

自由移动光标

自由移动光标在 **DifEq** 方式下的工作方式与在 **Func** 图象方式下的相同。显示 **x** 和 **y** 的光标坐
标值，并更新变量。

跟踪微分方程

要开始跟踪，从 **GRAPH** 菜单选择 **TRACE** (**GRAPH** **MORE** **F4**)。跟踪光标显示在第一个方程的
CtPlot 或接近 **tPlot** 的地方（如果 **t** 是一个坐标轴，则是 **tMin**）。

显示在屏幕底部的跟踪坐标反映坐标轴设置。例如，如果 **x=t** 且 **y=Q1**，则显示 **t** 和 **Q1**。如
果 **t** 不是坐标轴，则显示三个跟踪值。如果 **t** 是坐标轴，仅显示 **t** 和指定为 **y** 轴的变量。

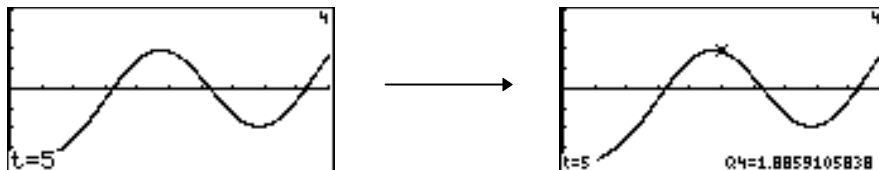
快速缩放在 **DifEq** 图象方式下
有效；平移则不行（第 6 章）。

跟踪光标按 **tStep** 步长增量或减量移动。当跟踪方程时，更新并显示坐标。如果光标移出屏
幕，显示在屏幕底部的坐标值继续相应地变化。

将跟踪光标移到 t 值处

要将跟踪光标移动到当前方程上的任意有效 t 值处, 则输入该数值。当输入第一位数时, $t=$ 提示显示在左下角。输入值必须对当前图象屏幕有效。完成输入后, 按 **ENTER** 重新激活跟踪光标。

t 和 Q 值显示在右侧的图象中, 因为选择了 $x=t$ 和 $y=Q$ 图象坐标轴。



在微分方程的图象上绘制

GRAPH DRAW 菜单项在 **DifEq** 图象方式下的作用与在 **Func** 图象方式下的作用相同。DRAW 指令的坐标是图象屏幕的 x 轴坐标和 y 轴坐标。

DrEqu 仅在 **DifEq** 方式下有效。 **DrInv** 在 **DifEq** 方式下无效。

绘制方程图象并将解保存到数组中

在当前图象屏幕上绘制解并将结果保存到指定的数组名中, 其语法为:

DrEqu($xAxisVariable, yAxisVariable[, xList, yList, tList]$)

$xAxisVariable$ 和 $yAxisVariable$ 指定用于绘图的坐标轴; 它们可能与当前图象屏幕的坐标轴设置不同。

DrEqu(不保存 x 、 y 或 t 的值)。

$xList$ 、 $yList$ 和 $tList$ 是可选的数组名, 可以将解 x 、 y 和 t 保存到其中。然后在主屏幕上或数组编辑器中显示 (第 11 章)。

使用自由移动光标选择初始条件。

不能跟踪绘图。但是在绘制方程后可以将 $xList$ 、 $yList$ 或 $tList$ 作为统计图绘制, 然后跟踪它们 (第 14 章)。另外可用统计回归模型拟合这些数组 (第 14 章)。

在示例中设置了默认窗口变量值。必要时, 从 GRAPH ZOOM 菜单中选择 ZSTD。

如果选择了 FldOff 域格式, 在使用 DrEqu(前必须输入初始条件)。

- 1 显示方式屏幕并设置 DifEq 图象方式。

2nd [MODE] $\downarrow$ $\downarrow$ $\downarrow$
 $\downarrow$ $\rightarrow$ $\rightarrow$ $\rightarrow$ ENTER

```
Func Pol Param Off
Dec Bin Oct Hex
Rectl CylV SphereV
SlpFld Off FldOff
Q'(t)=LIND INITC RES GRAPH
```

- 2 显示格式屏幕并设置 DirFld 域格式。

GRAPH MORE F1 $\downarrow$
 $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ $\rightarrow$
 ENTER

```
Plot1 Plot2 Plot3
Q'1=Q2
Q'2=-Q1
```

- 3 显示方程编辑器并保存方程 $Q'1=Q2$ 和 $Q'2=-Q1$ 。(删除所有其他方程)

F1 F2 2 $\downarrow$ (-) F2 1

```
DrEqu(
```

- 4 删除格式屏幕, 然后从 GRAPH DRAW 菜单选择 DrEqu。DrEqu(粘贴到主屏幕。

EXIT EXIT GRAPH
 MORE F2 F1

```
DrEqu(Q1,Q2,LX,LV,LT)
```

- 5 将变量赋给 x 轴和 y 轴。

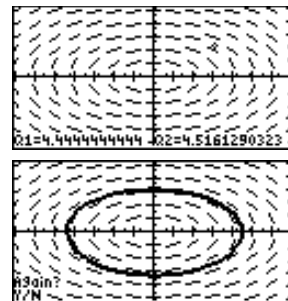
ALPHA [Q] 1 $\downarrow$
 ALPHA [Q] 2 $\downarrow$

- 6 指定用来保存 x 、 y 和 t 的解数组名。

ALPHA [L] ALPHA [X]
 $\downarrow$ ALPHA [L] ALPHA
 [Y] $\downarrow$ ALPHA [L]
 ALPHA [T] $\downarrow$

在示例中, 因为未设置初始条件, $Q'1$ 的方程未绘制。

- 7 显示图象屏幕并绘制方向域。 ENTER
- 8 将自由移动光标移到需要的初始条件坐标处。 ▶ ▼ ◀ ▲
- 9 绘制解。 x 、 y 和 t 的解数组保存在 **LX**、**LY** 和 **LT** 中。显示 **Again?** 提示, 大写字母锁仅在 **[Y]** 和 **[N]** 时开启。
 - ◆ 要将 **DrEqu** 用于新的初始条件, 按 **[Y]**、▶、▼、◀ 或 ▲。
 - ◆ 要退出 **DrEqu**(并显示 **GRAPH** 菜单, 按 **[N]** 或 EXIT。



使用缩放操作

GRAPH ZOOM 菜单项, 除 **ZFIT** 外在 **DifEq** 图象方式下的作用与在 **Func** 图象方式下的作用相同。在 **DifEq** 图象方式下, **ZFIT** 在 x 方向和 y 方向调整图象屏幕。

ZOOM 菜单项只影响 x (**xMin**、**xMax** 和 **xScl**) 和 y (**yMin**、**yMax** 和 **yScl**) 窗口变量。除非用 **ZSTD** 和 **ZRCL**, t 窗口变量 (**tMin**、**tMax**、**tStep** 和 **tPlot**) 并不受影响。可能需要编辑 t 窗口变量以确保绘制足够多的点。**ZSTD** 设置 **difTol=0.001**, 并将 t 和 Q 设置为坐标轴。

用 EXPLR 迭代绘制解

- ① 显示方式屏幕并设置 **DifEq** 图象方式。

2nd **[MODE]** **▼** **▼** **▼** **ENTER**

```
Func Pol Param DifEq
Dec Bin Oct Hex
RectV CylV SphereV
```

- ② 显示格式屏幕并设置 **FldOff** 域格式。

GRAPH **MORE** **F1** **▼**
▼ **▼** **▼** **▼** **▼** **▼**
ENTER

```
SlpFld DirFld FldOff
Q'(t)= WIND INITC AXES GRAPH
```

- ③ 显示方程编辑器并保存方程 **$Q'1=.001Q1(100-Q1)$** 。(删除所有其他的方程。)

F1 **□** **001** **F2** **1** **□**
100 **□** **F2** **1** **□**

```
Plot1 Plot2 Plot3
Q'1=.001 Q1(100-Q1)
```

- ④ 将坐标轴设为 **$x=t$** 和 **$y=Q1$** 。

2nd **[M4]** **▼** **▼** **1**

```
AXES: FldOff
x=t
y=Q1
```

- ⑤ 显示窗口编辑器并设置窗口变量值。

2nd **[M2]** **▼** **100** **▼**
□ **2** **▼** **▼** **▼**
100 **▼** **▼** **▼** **110**

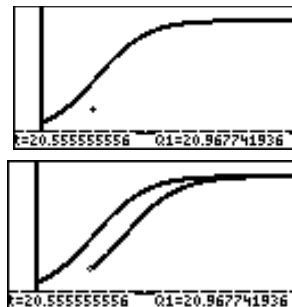
```
WINDOW
tMin=0
tMax=100
tStep=.2
tPlot=0
xMin=-10
xMax=100
xScl=1
yMin=-10
yMax=110
yScl=1
difTol=.001
Q'(t)= WIND INITC AXES GRAPH
```

- ⑥ 显示初始条件编辑器并输入初始条件。

F3 **10**

```
INITIAL CONDITIONS
tMin=0
Q11=10
```

- 7 从 GRAPH 菜单选择 **EXPLR**。 [MORE] [F5]
- 8 将自由移动光标移到要对其求解的初始条件处。 [▶] [▼] [◀] [▲]
- 9 用光标坐标 (x,y) 作为初始条件 $(t,Q'1(t))$ 绘制 **Q1** 的解。 [ENTER]



要继续绘制更多的解，移动自由移动光标，然后按 **[ENTER]**。

要停止使用 **EXPLR**，按 **[EXIT]**。

如果设置了 **SlpFld** 或 **DirFld**，坐标轴自动设置为指定解。

- ◆ 对于 **SlpFld**，设置 $x=t$ 和 $y=Q1$ 。
- ◆ 对于 **DirFld**，设置 $x=Q1$ 和 $y=Q2$ 。

如果坐标轴设置为指定解 t 、 Qn 或 $Q'n$ ，则绘制此解。

如果坐标轴未设置为指定解，同时 t 是一个变量， Q 是另一个变量，则绘制 **Q1**。

如果两个坐标轴都设置为 Q 变量，执行 **EXPLR** 会引起错误。

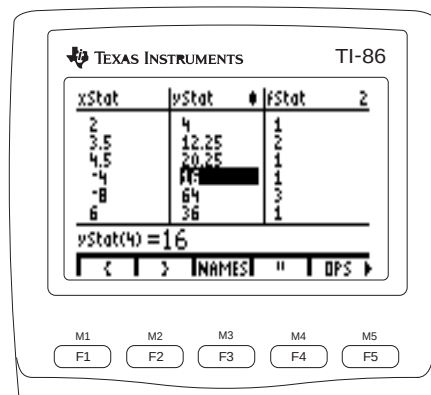
对指定的 t 计算微分方程

当跟踪光标未激活时, GRAPH 菜单项 **EVAL** 对指定的 t 值计算当前所选微分方程 Q_n , $t_{\text{Min}} \leq t \leq t_{\text{Max}}$ 。可以在图象上直接使用它。在程序中或从主屏幕上, **eval** 返回 Q 值的数组。

当设置了 **DirFld** 或 **SlpFld** 域格式, 在使用 **EVAL** 前必须指定初始条件。

11 数组

TI-86 中的数组	152
创建、存储和显示数组	153
数组编辑器	156
使用数组运算	159
对数组进行数学运算	161
连接公式到数组名	162



在 TI-86 中所能存储的数组长度和个数只受其内存容量的限制。

如果在方程或表达式中输入一个以上数组, 那么所有数组的元素个数必须相同。

TI-86 中的数组

数组是实数或复数元素的集合, 例如 {5,-20,13,9}。在 TI-86 中, 能:

- ◆ 直接在表达式中输入数组 (第 153 页)
- ◆ 输入数组并将它存到数组名 (变量) (第 154 页)
- ◆ 在数组编辑器中输入数组名 (第 156 页), 然后直接输入元素或使用所连接的公式自动产生它们 (第 161 页)
- ◆ 使用 Calculator-Based Laboratory™ (CBL) 或 Calculator-Based Ranger™ (CBR) 收集数据并将数据存储到 TI-86 上的数组名 (第 18 章)
- ◆ 使用 LIST OPS 的菜单项 **seq** 动态地创建数组 (第 159 页)

在 TI-86 中, 可用数组:

- ◆ 作为函数参数的一组值, 来返回结果数组 (第 1 章)
- ◆ 作为方程的一部分来绘制一族曲线 (第 5 章)
- ◆ 作为一组统计数据来用统计函数进行分析并在图象屏幕上绘制 (第 14 章)

LIST 菜单 [2nd] [LIST]

{	}	NAMES	EDIT	OPS
左括号	右括号	数组名 菜单	数组编辑器	数组运算 菜单

当输入数组时, { (左括号) 指定开始; } (右括号) 指定结束。若要粘贴 { 或 } 到光标处, 则从 LIST 菜单选择即可。

这里显示的 LIST NAMES 菜单中没有用户创建的数组名。

LIST NAMES 菜单 $\boxed{2nd} \boxed{[LIST]} \boxed{F3}$

{	}	NAMES	EDIT	OPS
fStat	xStat	yStat		

第 14 章描述 fStat、xStat 和 yStat。

每个用户创建的数组名都添加到 LIST NAMES 菜单和 VARS LIST 屏幕。数组名, 包括 fStat、xStat 和 yStat, 在这两个地方都以字母顺序排列。

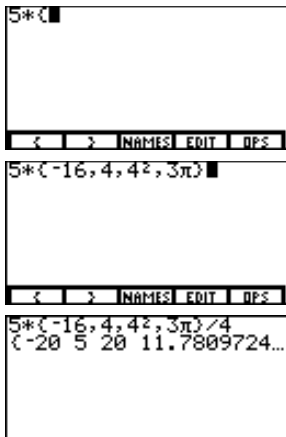
创建、存储和显示数组

在表达式中直接输入数组

直接输入数组的语法是: $\{element1, element2, ..., element n\}$

- 1 输入数组之前的表达式部分。 $5 \boxed{\times}$
- 2 从 LIST 菜单选择 { 开始输入数组。 $\boxed{2nd} \boxed{[LIST]} \boxed{F1}$
- 3 输入数组元素, 元素之间用逗号分隔。数组元素可以是表达式。 $\boxed{(-)} \boxed{16} \boxed{,} \boxed{4} \boxed{,}$
 $4 \boxed{[x^2]} \boxed{,} \boxed{3}$
 $\boxed{2nd} \boxed{[\pi]}$
- 4 从 LIST 菜单选择 } 结束数组。 $\boxed{F2}$
- 5 输入数组之后的表达式部分。 $\boxed{\div} \boxed{4}$
- 6 计算表达式。首先计算那些是表达式的元素。 $\boxed{ENTER}$

省略号 (...) 表示数组元素超出了窗口。用 $\boxed{\blacktriangleright}$ 和 $\boxed{\blacktriangleleft}$ 滚动数组。



通过存储数组创建数组名

存储数组的语法是: {*element1,element2,... ,element n*}>*listName*

用 **[STO▶]** 存储数组名时, 无需输入左括号 { }。

要从内存删除数组名, 则使用 MEM DELETE:LIST 屏幕 (第 17 章)。

- ❶

直接输入数组。(要存储以数组表示, 且当前存储在 **Ans** 中的结果, 如示例中所示, 则从步骤 2 开始。)

(步骤 2 到步骤 5 同上)
- ❷

将 **➔** 粘贴到光标处。大写字母锁开启。

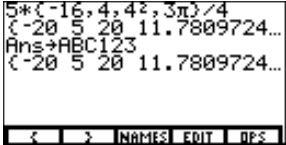
[STO▶]
- ❸

输入数组名。从 LIST NAMES 菜单选择数组名或直接输入数组名, 数组名要求以字母开头, 长度为 1 到 8 个字符。

[A][B][C]
[ALPHA] 1 2 3
- ❹

将数组保存到数组名。

[ENTER]



显示存储到数组名中的数组元素

TI-86 区分数组名中字母的大小写。例如, **ABC123**、**Abc123** 和 **abc123** 是三个不同的数组名。

- ❶

在主屏幕中输入数组名; 从 LIST NAMES 菜单选择数组名或输入字符。

[2nd][LIST][F3]
[F1]
- ❷

显示数组元素。

[ENTER]



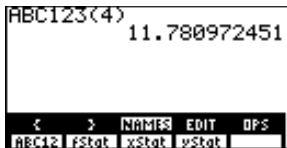
显示或使用单个数组元素

显示或使用单个数组元素的语法是: *listName(element#)*

listName(element#) 在表达式中有效。

element# ≥ 1 且 $\leq$ 数组维数。

- ❶ 输入数组名; 从 LIST NAMES 菜单选择数组名或输入字符。 [2nd] [LIST] [F3]
- ❷ 在括号中输入数组中数组元素的序号。 [] 4 []
- ❸ 显示数组元素。 [ENTER]



将新值存储到数组元素

将数值存储到当前数组元素或添加到数组末尾的语法是:

value $\rightarrow$ *listName(element#)*

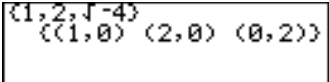
value 可以是表达式。

- ❶ 输入要保存到当前数组元素或添加到数组末尾的数值。 [2nd] [✓] 18
- ❷ 将 $\rightarrow$ 粘贴到光标处。 [STO $\rightarrow$]
- ❸ 输入数组名; 从 LIST NAMES 菜单选择数组名或输入字符。 [F1]
- ❹ 在括号内输入元素的序号。(在本例中, 5 是超出 **ABC123** 当前维数的序号。)
- ❺ 输入新值到元素序号处。(✓18 的计算结果添加为第 5 个元素。)



复数数组元素

数组元素可以是复数。只要有一个数组元素为复数，数组中所有元素都显示为复数。（ $\sqrt{-4}$ 的结果为复数。）

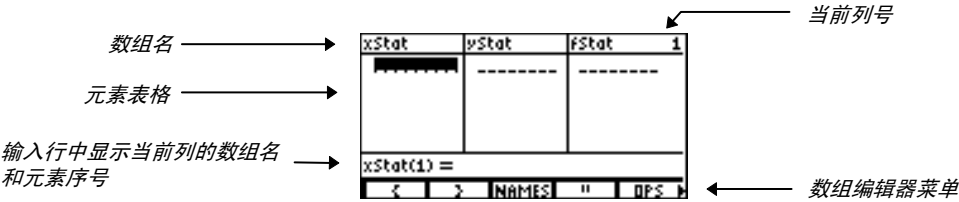


数组编辑器 [2nd] [LIST] [F4]

数组编辑器是一个表格，可在其中保存、编辑和显示内存中多达 20 个数组。还可在数组编辑器中创建数组名，并连接公式到数组。

按 [2nd] [STAT] [F2] 显示数组编辑器。

数组编辑器必要时缩写数组名和元素值。输入行显示整个数组名和元素值。



List Editor（数组编辑器）菜单 [2nd] [LIST] [F4]

{	}	NAMES	"	OPS	▶	REAL				
---	---	-------	---	-----	---	------	--	--	--	--

数组编辑器的菜单项 {、}、NAMES 和 OPS 与 LIST 的菜单项相同（第 152 页）。

- " 指定连接到数组名的公式开始与结束
- ▶REAL 转换当前数组为实数数组

要在数组编辑器中使用 LIST OPS 菜单项（或其他函数或指令），光标位置必须对结果合适。例如，如果数组名亮显，则可以使用 LIST OPS 菜单项 **sortA**；如果数组元素亮显，则不能使用该菜单项。

内存复位后, **xStat**、**yStat** 和 **fStat** 保存在 1、2 和 3 列中。复位默认值不影响数组编辑器。

要将第 1 列的数组名移动到未命名列, 则按 $\leftarrow \rightarrow$ 。

若全部 20 列中都有数组名, 则必须删除一个数组名为未命名列留出空间。

要取消插入数组名, 则按 **[CLEAR]**。

若一个公式与 **ABC123** 相连, 输入行中将显示公式, 而不是步骤 3 (第 162 页) 中所显示的数组。

在未命名列中创建数组名

- 1 显示数组编辑器。 $\leftarrow$ **[2nd]** **[LIST]** **[F4]**
- 2 将光标移到未命名列 (第 4 列)。输入行中显示提示符 **Name=**。大写字母锁开启。 $\rightarrow \rightarrow \rightarrow \rightarrow$
- 3 输入数组名。数组名显示在当前列的顶部。在输入行中, 显示数组名提示。该数组名成为 **LIST NAMES** 菜单项和 **VARS LIST** 屏幕项。 **[X]** **[Y]** **[Z]** **[ENTER]**

yStat	fStat		4
-----	-----		
Name=XYZ			
ABC12	XYZ	fStat	xStat yStat
yStat	fStat	WP	4
-----	-----	-----	
XYZ =			
← → NAMES " OPS			

在数组编辑器中插入数组名

- 1 移动光标到第 3 列。 $\leftarrow$
- 2 插入一个新的、未命名列。数组名右移, 第 3 列清除。显示提示符 **Name=** 和 **LIST NAMES** 菜单。 $\leftarrow$ **[2nd]** **[INS]**
- 3 从 **LIST NAMES** 菜单选择 **ABC12**, 将数组名 **ABC123** 插入到第 3 列。 **ABC123** 所保存的元素填充表格的第 3 列。在输入行显示 **ABC123** 所有元素的精确值。 **[F1]** **[ENTER]**

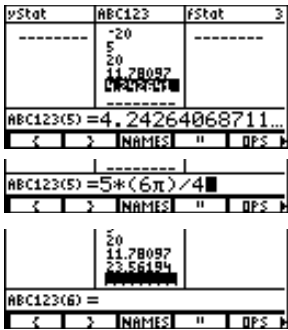
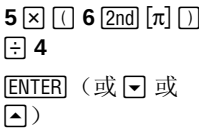
yStat		fStat	3
-----		-----	
Name=ABC123			
ABC12	XYZ	fStat	xStat yStat
yStat	ABC123	fStat	3
-----	-----	-----	
	20 11.78097 4.242641		
ABC123 = (-20, 5, 20, 11.7...			
← → NAMES " OPS			

要取消任何编辑并恢复光标处的原始元素, 按 **CLEAR** **ENTER**。

可将表达式作为元素输入。

显示和编辑数组元素

- 1 移动光标到 **ABC123** 的第 5 个元素。输入行显示数组名、括号中的元素序号和元素的精确值。
- 2 切换到编辑元素环境, 在输入行中编辑元素。
- 3 输入所编辑元素。计算表达式, 结果存为当前元素。



从数组中删除元素

要从数组中删除单个元素, 则将光标移到元素上, 然后按 **DEL**。通过下述三种方法之一清除数组的所有元素。

- ◆ 在数组编辑器中, 按 **↑** 移动光标到数组名并按 **CLEAR** **ENTER**。
- ◆ 在数组编辑器中, 移动光标到每个元素并按 **DEL**。
- ◆ 在主屏幕或在程序编辑器中, 输入 **0→dimL listName** (数组名) 设置 *listName* 的维数为 **0** (参阅 “A 到 Z 参考”)。

从数组编辑器中删除数组

要从数组编辑器中删除数组, 则将光标移动到数组名上, 然后按 **DEL**。数组并未从内存中删除; 仅是从数组编辑器中删除。

用下述两种方法之一可删除数组编辑器中的所有用户创建数组，并将数组名 **xStat**、**yStat** 和 **fStat** 恢复到列 **1**、**2** 和 **3**。

- ◆ 使用无参数的 **SetLEdit**（第 161 页）。
- ◆ 复位内存（第 17 章）。复位默认值不影响数组编辑器。

使用数组运算

LIST OPS（数组运算）菜单 **[2nd]** **[LIST]** **[F5]**

{	}	NAMES	EDIT	OPS						
dimL	sortA	sortD	min	max	▶ <table><tr><td>sum</td><td>prod</td><td>seq</td><td>li↗vc</td><td>vc↘li</td></tr></table>	sum	prod	seq	li↗vc	vc↘li
sum	prod	seq	li↗vc	vc↘li						
					▶ <table><tr><td>Fill</td><td>aug</td><td>cSum</td><td>DeltaI</td><td>Sortx</td></tr></table>	Fill	aug	cSum	DeltaI	Sortx
Fill	aug	cSum	DeltaI	Sortx						
					▶ <table><tr><td>Sorty</td><td>Select</td><td>SetLE</td><td>Form</td><td></td></tr></table>	Sorty	Select	SetLE	Form	
Sorty	Select	SetLE	Form							

除 **Fill**(外并且有时 **dimL** 也除外，对于所有其他 **LIST OPS** 菜单项，可直接输入数组 **{element1,element2,...}** 作为 *list* 参数。

SortA 和 **SortD** 基于模排列复数数组。

dimL *list*
#ofElements → **dimL** *listName*
#ofElements → **dimL** *listName*

sortA *list*
sortD *list*

返回数组 *list* 维数（或元素个数）
创建长度为 **#ofElements** 的数组 *listName*；每个元素值为 **0**
重置现有数组 *listName* 的维数；在新维数内以前输入的元素仍保留，新元素值都为 **0**；旧数组中在新维数外的元素被删除
将数组 *list* 的元素按照从小到大的顺序排序
将数组 *list* 的元素按照从大到小的顺序排序

对于复数数组, **min** 或 **max**
返回模的最小或最大值。

min(list)

max(list)

sum list

prod list

seq(expression,variable,
begin,end[,step])

li>vc list

li>vc {element1,element2,...}

vc>li vector

vc>li [element1,element2,...]

Fill(number,listName)

aug(listA,listB)

cSum(list)

从菜单选择 **Deltalst** 将
Deltalst(粘贴到光标处)。

Deltalst(list)

Sortx [ListName,ListName,
frequencyListName]

Sortx 和 **Sorty** 中的两个数组
必须有相同数目的元素。

Sorty [xListName,ListName,
frequencyListName]

返回实数或复数数组 *list* 的最小元素

返回实数或复数数组 *list* 的最大元素

返回实数或复数数组 *list* 中最后一个元素至第一个元素所有元素之和

返回实数或复数数组 *list* 的所有元素之积

返回一个数组, 它的每个元素是对 *variable* 的每个值, 即从 *begin* 到 *end*, 增量为 *step* (*step* 可以是负数) 计算 *expression* 的结果

转换实数或复数数组 *list* 为向量

转换实数或复数向量 *vector* 为数组

将数组 *listName* 的每个元素值都赋为实数或复数 *number*
(参数) 合并 *listA* 和 *listB* 的实数或复数元素

返回一个数组, 其元素是实数或复数数组 *list* 中相应元素到最后一个元素的累积和

返回一个数组, 其元素是实数或复数数组 *list* 中相邻元素之差

按 **x** 元素的升序排序数组 *xListName*, 然后按此顺序排序
xListName、*yListName* 和 *frequencyListName* 中的 **x** 和 **y**
数据对, 以及 (可选) 频数; **xStat** 和 **yStat** 为默认值

按 **y** 元素的升序排序数组 *xListName*, 然后按此顺序排序
xListName、*yListName* 和 *frequencyListName* 中的 **x** 和 **y**
数据对, 以及 (可选) 频数; **xStat** 和 **yStat** 为默认值

从菜单中选择 **SetLE** 粘贴 **SetLEdit** 到光标处。

可以创建新数组名作为 **SetLEdit** 的参数。

Select(*xListName*,
yListName)

从一个散点图或 xy 直线图（仅应用于它们）中选择一个或多个指定的数据点，然后将选定的数据点存储到两个数组，即 *xListname* 和 *yListname*（第 14 章）

SetLEdit [*column1ListName*,
column2ListName,...,
column20ListName]

设置数组编辑器；若 **SetLEdit** 有 1 到 20 个数组 *ListNames*，则按指定的顺序装入它们；若 **SetLEdit** 无参数，则删除数组编辑器中当前所有数组，并在 1、2、3 列输入默认数组 **xStat**、**yStat** 和 **fStat**

Form("formula",*listName*)

连接公式 *formula* 到数组 *listName*；公式 *formula* 结果为数组，它在数组 *listName* 中动态地存储并更新（第 162 页）

对数组进行数学运算

在许多 TI-86 函数中，可使用数组作为单个参数；结果也是数组。对于数组中的每个元素，函数应有效；然而，当绘制时，未定义点不会造成错误。

当在同一函数中使用数组作为两个或多个参数时，所有数组应有相同数目的元素（相同维数）。下面是一些数组作为单个参数的示例。

{1,2,3}+10 返回 **{11 12 13}**

{5,10,15}*{2,4,6} 返回 **{10 40 90}**

3+{1,7,(2,1)} 返回 **{{(4,0) (10,0) (5,1)}**

$\sqrt{\{4,16,36,64\}}$ 返回 **{2 4 6 8}**

$\sin \{7,5\}$ 返回 **{.656986598719 -.958924274663}**

$\{1,15,36\}<19$ 返回 **{1 1 0}**

连接公式到数组名

只有当公式与数组分离后, 才能编辑由连接公式创建的数组元素。

如果相连公式中包含一个以上数组名, 则每个数组的维数必须相同。

在主屏幕的空行执行下列步骤。

- ◆ 可将公式连接到数组名, 这样公式会生成一个数组, 它存储在该数组名中, 并动态更新。
- ◆ 当编辑被公式引用的数组的元素时, 连接此公式的数组中相应元素更新。
- ◆ 当编辑公式本身时, 连接此公式的数组中所有元素更新。

在主屏幕或程序编辑器中将公式连接到数组名的语法是:

Form("formula",listName)

若输入新数组名作为 **Form(** 的第 2 个参数, 则执行时将创建这个数组名并将之保存在 LIST NAMES 菜单和 VARS LIST 屏幕。

- 1 将元素保存到数组名。
- 2 从 LIST OPS 菜单选择 **Form**, **Form(** 粘贴到光标处。
- 3 在引号内输入公式。
- 4 输入逗号, 然后输入数组名, 公式将与之相连。
- 5 连接公式到数组名。

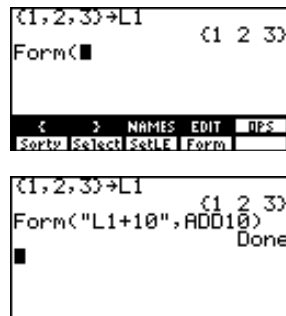
```
[2nd] [LIST] [F1] 1 [ , ] 2 [ , ]
3 [F2] [STO➡] [L] [ALPHA]
1 [ENTER]

[F5] [MORE] [MORE]
[MORE] [F4]

[2nd] [STRNG] [F1] [ALPHA]
[L] 1 [ + ] 10 [F1]

[ , ] [ALPHA] [ALPHA] [A]
[D] [D] [ALPHA] 10 [ ]

[ENTER]
```



可用数组编辑器 (第 157 页)
查看连接到数组名的公式。

比较普通数组和相连数组

按下述步骤观察普通数组与相连数组之间的差异。下面的示例以上述连接公式到数组的示例为基础。注意步骤 1 中的公式未连接到 **LX**，这是因为它并未被引号括起来。

- ❶ 通过存储表达式 **L1+10** 到数组名 **LX** 产生一个普通数组。

ALPHA [L] 1 +
10 [STO→] [L] [X]
[ENTER]

L1+10→LX	<11 12 13>
----------	------------

- ❷ 将 **L1** 中的第 2 个元素改为 **-8**，并显示编辑后的数组。

(-) 8 [STO→] [L]
ALPHA 1 [] 2 []
[2nd] [:] [ALPHA]
[L] 1 [ENTER]

L1+10→LX	<11 12 13>
-8→L1(2):L1	<1 -8 3>

- ❸ 比较普通数组 **LX** 与 **ADD10** 的元素，公式 **L1+10** 连接到 **ADD10**。注意 **LX** 的第 2 个元素无变化，同时 **ADD10** 的第 2 个元素已重新计算，因为 **L1** 第 2 个元素已编辑过。

[2nd] [LIST]
[F3] [F2]
[ENTER] [F4]
[ENTER]

ADD10	<11 2 13>
LX	<11 12 13>
< > NAMES EDIT OPS REC12 ADD10 L1 LX fStat	

如果其他数组名存储在 LIST NAMES 菜单，按 [F1] 和 [F3] 未必会将 **ADD10** 和 **LX** 粘贴到所示的主屏幕上。

用数组编辑器连接公式

- ❶ 显示数组编辑器。
- ❷ 亮显要将公式与其连接的数组名。
- ❸ 在引号中输入公式。

[2nd] [LIST] [F4]
[] []
[F4] 4 × [F3] [F2]
[2nd] [F4]

xStat	yStat	fStat
-2 9 6 1	-----	-----
yStat = "4*xStat"		
< > NAMES " OPS fStat xStat yStat		

示例中 LIST NAMES 菜单只有 **fStat**、**xStat** 和 **yStat**，同时 **xStat**={-2,9,6,1,-7}。

所连公式必须用引号括起来。

数组编辑器在有公式与之相连的数组名旁显示一个公式锁符号。

- ④ 连接公式并生成数组。
- ◆ TI-86 计算每个数组元素。
- ◆ 锁符号显示在已连接公式的数组名旁。

[ENTER]

xStat	yStat	fStat
-2	8	-----
9	36	
6	36	
1	4	
-7	-28	

yStat(1) = -8		
← → NAMES " OPS		

若要编辑所连接公式，则在步骤 3 按 [ENTER]，然后编辑这个公式。

使用数组编辑器处理与公式相连接的数组

当编辑所连公式中引用的数组的元素时，TI-86 更新公式所连接的数组元素。

xStat	yStat	fStat	1
5	10	-----	
10	20		
15	30		
20	40		
xStat(1) = -33			
← → NAMES " OPS			
fStat	xStat	yStat	

xStat	yStat	fStat	1
-33	-66	-----	
10	20		
15	30		
20	40		
xStat(2) = 10			
← → NAMES " OPS			
fStat	xStat	yStat	

如果在数组编辑器的 3 个当前列中显示有与公式相连的数组时，编辑或输入这 3 列中所显示的数组元素，则 TI-86 执行编辑和输入的时间较长。要减少这种影响，则向左或向右滚动列，将与公式相连的数组移出当前所显示的 3 列。

执行和显示所连接公式

所连接公式的执行结果必须是数组。例如: "5*xStat", "seq(x,x,1,10)" 和 "{3,5, -8,4}^2/10"。当显示公式相连接的数组时，公式执行。只要公式所引用的数组被改变时，无论在主屏幕、数组编辑器或程序中，公式也要执行。

可将一个尚未与数组连接的公式成功地连接到另一个数组。例如, 可将 **"5*xStat"** 与数组名 **BY5** 连接, 后者中没有元素存储在 **xStat** 中。然而, 如果在 **xStat** 中无元素时显示 **BY5**, 则会发生错误。

当在数组编辑器中连接这样一个公式到数组名时, 连接会成功, 但会发生错误。这是因为数组编辑器在连接公式到数组名后会立即执行公式。

若要再查看数组编辑器, 必须返回到主屏幕, 输入一些使公式与数组关联的操作, 或者使用 **LIST OPS** 菜单项 **SetLE** (第 161 页) 从数组编辑器中删除与公式连接的数组。

处理与连接公式相关的错误

连接公式所引用数组的所有元素对于连接公式必须有效。

在主屏幕中, 可将引用无元素数组 (维数为 **0**; 第 161 页) 的公式与另一个数组相连。不过, 只有在公式所引用的数组中输入至少一个元素后, 才能在数组编辑器或主屏幕显示与公式相连的数组。

提示: 如果试图在数组编辑器中显示与公式相连的数组时出现错误菜单, 可以首先选择 **GOTO**, 记下连接到数组名的公式, 然后按 **[CLEAR]** **[ENTER]** 清除公式。随后可用数组编辑器找到错误原因。做适当修改后, 可重新将公式与数组名相连。

若不想清除公式, 可选择 **QUIT**, 在主屏幕显示所引用的数组, 找到错误原因并改正。要在主屏幕编辑数组元素, 则将新值保存到 *listName(element#)* (第 155 页)。

公式与数组名分离

通过下列五种方法之一可分离公式:

- ◆ 使用 **dimL** 改变数组维数 (第 159 页)。
- ◆ 使用 **value→listName(element#)** 把 **value** 存到与公式相连的数组元素中。
- ◆ 使用 **""→listName** , 其中 **listName** 是与公式相连的数组。
- ◆ 在数组编辑器中, 把光标移到与公式相连的数组名上, 然后按 **[ENTER] [CLEAR] [ENTER]** 。所有的数组元素保留, 但公式被分离, 锁符号消失。
- ◆ 在数组编辑器中, 把光标移到与公式相连的数组元素上。按 **[ENTER]** , 编辑该元素, 然后按 **[ENTER]** 。元素改变, 公式被分离, 锁符号消失。所有其他元素不变。

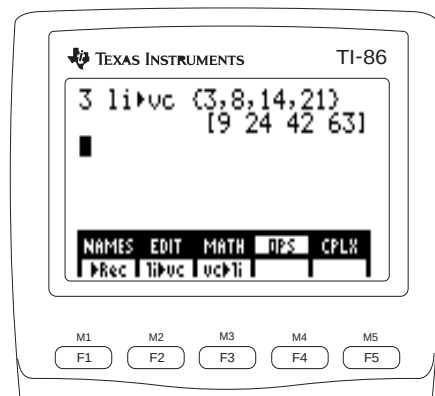
编辑与公式相连数组的元素

如上所述, 公式与数组名分离的一种方法是编辑与公式连接数组的元素。TI-86 有相应的保护措施, 以避免将光标移到数组的一个元素时不小心将公式与数组分离。

由于这种保护特性, 必须在开始编辑与公式相连数组的元素前按 **[ENTER]** 。这种保护特性可避免删除与公式相连数组中的元素。为了删除与公式相连数组的元素, 必须先使用上述方法之一分离公式。

12 向量

TI-86 的向量	168
创建、存储和显示向量	169
向量的数学运算	176



TI-86 的向量

向量是排列为一行或一列的一维数组。向量元素可以为实数，也可以是复数。在主屏幕或向量编辑器中可以创建、显示和编辑向量。当创建向量后，向量元素保存到向量名。

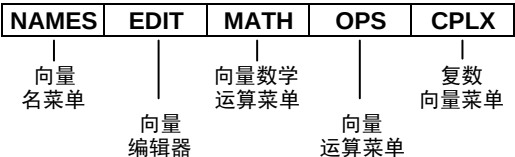
TI-86 的向量编辑器垂直显示向量。在主屏幕中，向量水平输入和显示。当在表达式中使用向量时，TI-86 会自动地以适合表达式的形式（行向量或列向量）解释向量。例如，列向量适合表达式 *matrix***vector*。

在 TI-86 中，可用直角坐标方式在向量中存储多达 255 个元素。可使用二元素或三元素向量定义二维或三维空间的距离和方向。可用不同的形式表达二元素或三元素向量，这依赖于向量类型。

表达形式	输入:	TI-86 返回:
二元素的直角坐标向量	$[x,y]$	$[x\ y]$
二元素的柱面坐标向量	$[r\angle\theta]$	$[r\angle\theta]$
二元素的球面坐标向量	$[r\angle\theta]$	$[r\angle\theta]$
三元素的直角坐标向量	$[x,y,z]$	$[x\ y\ z]$
三元素的柱面坐标向量	$[r\angle\theta,z]$	$[r\angle\theta\ z]$
三元素的球面坐标向量	$[r\angle\theta\angle\phi]$	$[r\angle\theta\angle\phi]$

创建、存储和显示向量

VECTR (向量) 菜单 [2nd] [VECTR]



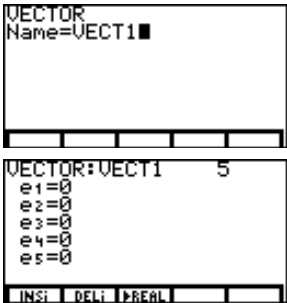
VECTR NAMES (向量名) 菜单 [2nd] [VECTR] [F1]

VECTR NAMES 菜单以字母顺序显示所有当前存储的向量名。若要在当前光标处粘贴一个向量名，只需从菜单中选择它。

在向量编辑器中创建向量 [2nd] [VECTR] [F2]

TI-86 区分向量名字符的大小写。例如，VECT1、Vect1 和 vect1 是三个不同的向量名。

- 1 显示向量 **Name=** 提示屏幕。 [2nd] [VECTR] [F2]
- 2 大写字母锁开启。VECTR NAMES 菜单显示。 [V][E][C][T]
输入一个 1 到 8 个字符长，以字母开头的名称。 [ALPHA] 1
- 3 显示向量编辑器。也显示向量编辑器菜单。 [ENTER]
- 4 接受或改变长度为 ≥ 1 且 ≤ 255 的整数的向量元素维数。显示向量；所有元素为 0。 5 [ENTER]



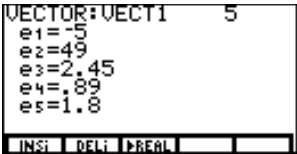
在第一列中↓或↑表示还有其他向量元素。

可以在向量元素提示处输入表达式。

- ⑤ 在每个向量元素提示处输入向量元素值。可以输入表达式。按 **ENTER** 或 **↓** 移到下一个提示处。向量元素存储到 **VECT1**，成为 **VECTR NAMES** 的菜单项。
- ⌈** 5 **↓** 49

↓ 2 **↓** 45 **↓**

↓ 89 **↓** 1 **↓** 8



Vector Editor（向量编辑器）菜单 **2nd** **[VECTR]** **F2** *向量名* **ENTER**

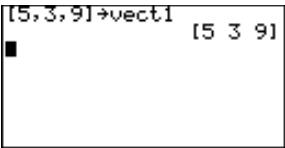
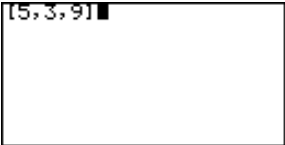
INSi	DELi	>REAL		
------	------	-------	--	--

- INSi** 在光标处插入空白元素 (**en=**) 提示；当前元素下移
- DELi** 删除光标处的元素，该元素也从向量中删除；向量元素上移
- >REAL** 转换当前显示的复数向量为实向量

在主屏幕中创建向量

- ① 用 **[]** 定义向量开始。 **2nd** **[I]**
- ② 输入向量元素，元素之间用逗号分隔。 **5** **↓** **3** **↓** **9**
- ③ 用 **]** 定义向量结束。 **2nd** **[I]**
- ④ 保存向量到向量名，向量名长度为 1 到 8 个字符，并以字母开头。向量水平显示，向量名成为 **VECTR NAMES** 的菜单项。

STO▶ **2nd** **[alpha]** **[V]**
[E] **[C]** **[T]** **[ALPHA]**
[ALPHA] **1** **ENTER**



使用 **MEM DELETE:VECTR** 屏幕（第 17 章）从内存删除向量名。

创建复数向量

如果向量中任一元素是复数，则向量的所有元素以复数显示。例如，如果输入向量 **[1,2,(3,1)]**，则 TI-86 显示 **[(1,0) (2,0) (3,1)]**。

用两个实向量创建复数向量的语法是：

realVector+(0,1)*imaginaryVector*→*complexVectorName*

realVector 包含每个元素的实部； *imaginaryVector* 包含其虚部。

显示向量

要显示向量，将向量名粘帖到主屏幕上，然后按 **[ENTER]**。

在主屏幕或程序中显示向量名 *vectorName* 中指定元素的语法是：

vectorName(*element#*)

实二元素或三元素向量结果的显示依赖于当前向量方式设置：**RectV**、**CylV** 或 **SphereV**（第 1 章）。可从 **VECTR OPS** 菜单中选择向量转换指令来取代当前方式设置（第 173 页）。

复数向量仅以直角坐标形式显示。

当表达式执行时, 结果以向量形式显示。

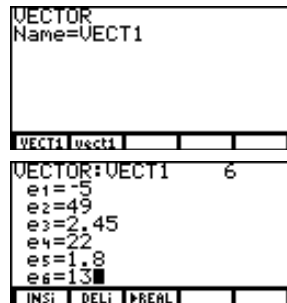
在表达式中使用向量

- ◆ 可直接输入向量 (例如, $35-[5,10,15]$)。
- ◆ 可使用 **[ALPHA]** 和 **[2nd] [alpha]** 输入以字母命名的向量名。
- ◆ 可从 VECTR NAMES 菜单选择向量名 (**[2nd] [VECTR] [F1]**)。
- ◆ 可从 VARS VECTR 菜单选择向量名 (**[2nd] [CATLG-VARS] [MORE] [F1]**)。

编辑向量维数和元素

- 1 显示向量 **Name=** 提示屏幕。 **[2nd] [VECTR] [F2]**
- 2 输入向量名; 从 VECTR NAMES 选择向量名或输入组成向量名的字符。 **[F1]**
- 3 显示向量编辑器。 **[ENTER]**
- 4 改变或接受向量维数。 **6 [ENTER]**
- 5 移动光标到任意元素并编辑它。继续移动光标到其他元素。 **[↓] [↓] [↓] 22**
[↓] [↓] 13
- 6 保存改变并退出向量编辑器。 **[EXIT]**

可用 **[CLEAR]**、**[DEL]** 和 **[2nd] [INS]** 编辑矩阵元素。也可以覆盖已存在字符。



在主屏幕使用 **[STO→]** 改变元素值的语法是:

value→*vectorName*(*element*#)

VECTR MATH (向量数学运算) 菜单 2nd [VECTR] [F3]

NAMES	EDIT	MATH	OPS	CPLX
cross	unitV	norm	dot	

cross(*vectorA*,*vectorB*) 返回向量 *vectorA* 和向量 *vectorB* 的交积，两者都是实数或复数的二元素或三元素向量；用变量表示， **cross**([a,b,c],[d,e,f]) 返回 [bf-ce cd-af ae-bd]

unitV *vector* 返回单位向量，其中实数或复数向量 *vector* 的每个元素都除以向量范数

norm *vector* 返回 Frobenius 范数 ($\sqrt{\Sigma(real^2+imaginary^2)}$), 其中求和是对实数或复数向量 *vector* 的所有元素

dot(*vectorA*,*vectorB*) 返回向量 *vectorA* 和向量 *vectorB* 的点积，两者都是实数或复数向量；用变量表示， **dot**([a,b,c],[d,e,f]) 返回 ad+be+cf

VECTR OPS (向量运算) 菜单 2nd [VECTR] [F4]

NAMES	EDIT	MATH	OPS	CPLX	
dim	Fill	►Pol	►Cyl	►Sph	►►Rec li►vc vc►li

按 [STO►] 在 #ofElements 之后输入 ► 符号。

- dim** *vector*

#ofElements►dim*vectorName*

#ofElements►dim*vectorName*

Fill(*number*,*vectorName*)
- 返回向量 *vector* 的维数（或其中的元素个数）

创建指定长度 (#ofElements) 的新向量 *vectorName* ；每个元素为 0

重新设置向量 *vectorName* 的维数为指定长度 (#ofElements)

将实数或复数 *number* 存储到向量 *vectorName* 中的每个元素

对于下面的转换函数, 柱面坐标形式 $[r \ \theta \ z]$ 的三元素向量转换方程是:

$$x = r \cos\theta \quad y = r \sin\theta \quad z = z$$

球面坐标形式 $[r \ \theta \ \phi]$ 的三元素向量转换方程是:

$$x = r \cos\theta \sin\phi \quad y = r \sin\theta \sin\phi \quad z = r \cos\phi$$

<code>vector</code> → Pol	用极坐标形式 $[r \angle \theta]$ 显示二元素向量 <i>vector</i>
<code>vector</code> → Cyl	用柱面坐标形式 $[r \angle \theta \ 0]$ 或 $[r \angle \theta \ z]$ 显示二元素或三元素向量 <i>vector</i>
<code>vector</code> → Sph	用球面坐标形式 $[r \angle \theta \ 0]$ 或 $[r \angle \theta \ \phi]$ 显示二元素或三元素向量 <i>vector</i>
<code>complexVector</code> → Rec	用直角坐标形式 $[x \ y]$ 或 $[x \ y \ z]$ 显示二元素或三元素复数向量 <i>complexVector</i>
li → vc <i>list</i>	将实数或复数数组 <i>list</i> 转换为向量
vc → li <i>vector</i>	将实数或复数向量 <i>vector</i> 转换为数组

复数元素仅对 **li**→**vc** 和 **vc**→**li** 有效。

VECTR CPLX (复数向量) 菜单 $\boxed{2\text{nd}} \boxed{[\text{VECTR}]} \boxed{[\text{F5}]}$

NAMES	EDIT	MATH	OPS	CPLX
conj	real	imag	abs	angle

- conj** *complexVector* 返回一个向量, 其每个元素是复数向量 *complexVector* 对应元素的共轭复数
- real** *complexVector* 返回一个实向量, 其每个元素是复数向量 *complexVector* 对应元素的实部
- imag** *complexVector* 返回一个实向量, 其每个元素是复数向量 *complexVector* 对应元素的虚部
- abs** *Vector* 返回一个实向量, 其每个元素或是实向量 *vector* 对应元素的绝对值, 或是复数向量 *complexvector* 对应元素的模
- angle** *complexVector* 返回一个实向量, 如果复数向量 *complexVector* 的元素是实数, 则向量元素为 0; 如果复数向量 *complexVector* 的元素是复数, 则向量元素为极角; 极角用 $\tan^{-1}(\text{complex}/\text{real})$ 计算, 在第二象限、第三象限时分别用 $+\pi$ 、 $-\pi$ 调整

向量的数学运算

向量 *vectorA* 和向量 *vectorB* 进行加减时它们的维数必须相等。

vectorA+vectorB

向量 *vectorA* 中的每个元素和向量 *vectorB* 中的对应元素相加, 返回和的向量

vectorA-vectorB

向量 *vectorA* 中的每个元素减去向量 *vectorB* 中的对应元素, 返回差的向量

两个向量不能相乘或相除。

*vector*number* or
*number*vector*

返回一个向量, 它的每个元素是实数或复数向量 *vector* 中的每个元素与实数或复数 *number* 的乘积

*matrix*vector*

返回一个向量, 它的每个元素是向量 *vector* 中的每个元素与矩阵 *matrix* 中的每个元素的乘积; 矩阵 *matrix* 的列维数与向量 *vector* 的维数必须相等

vector/number

返回一个向量, 它的每个元素是实数或复数向量 *vector* 中的每个元素被实数或复数 *number* 除的商

-vector

(求逆) 改变向量 *vector* 中每个元素的符号

== 和 **≠** 在 TEST 菜单上。

vectorA==vectorB

如果对应元素的比较都为真, 返回 **1**; 否则返回 **0**

vectorA≠vectorB

只要有一对元素的比较为假就返回 **1**

round、**iPart**、**fPart** 和 **int** 在 MATH NUM 菜单上。

round(vector[,#ofDecimals])

向量 *vector* 中的每个元素四舍五入为 12 位, 或四舍五入为指定的小数位数 *#ofDecimals*

iPart *vector*

返回实数或复数向量 *vector* 中每个元素的整数部分

fPart *vector*

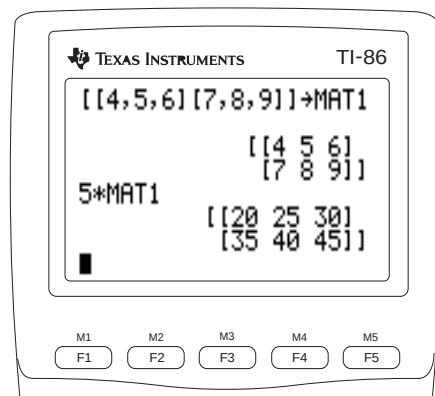
返回实数或复数向量 *vector* 中每个元素的小数部分

int *vector*

返回实数或复数向量 *vector* 中元素的最大整数

13 矩阵

TI-86 的矩阵	178
创建、存储和显示矩阵	178
矩阵的数学运算	185

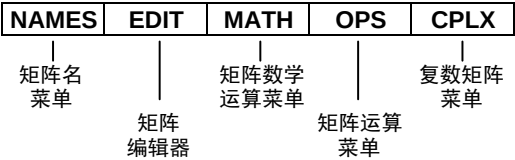


TI-86 的矩阵

矩阵是一个以行和列形式排列的二维数组。矩阵元素可以是实数，也可以是复数。在主屏幕或矩阵编辑器中可创建、显示和编辑矩阵。创建一个矩阵后，元素保存到矩阵名。

创建、存储和显示矩阵

MATRIX（矩阵）菜单 **[2nd] [MATRIX]**



MATRIX NAMES（矩阵名）菜单 **[2nd] [MATRIX] [F1]**

MATRIX NAMES 菜单包含所有当前保存的矩阵名，它们按字母顺序排列。若要在当前光标处粘贴矩阵名，只需从菜单中选择它。

在矩阵编辑器中创建矩阵 **[2nd] [MATRIX] [F2]**

- 1 显示矩阵 **Name=** 提示屏幕。
- 2 大写字母锁开启。MATRIX NAMES 菜单显示。输入一个长度为 1 到 8 个字符，且以字母开头的名称。

[2nd] [MATRIX] [F2]

[M] [A] [T]

[ALPHA] 1



TI-86 区分矩阵名字符的大小写。例如，**MAT1** 和 **mat1** 是两个不同的向量名。

矩阵行两端的省略号(...)表示屏幕之外还有其他列。

最后一列中的↓或↑表示还有其他列。

- ③ 显示矩阵编辑器和矩阵编辑器菜单。
- ④ 接受或改变屏幕右上角的矩阵维数 (行×列) ($1 \leq \text{行} \leq 255$ 且 $1 \leq \text{列} \leq 255$) ; 最大维数受可用内存限制。矩阵显示: 所有元素为 0。
- ⑤ 在元素提示处 (1,1= 表示行 1 , 列 1) 输入每一个元素值。可以输入表达式。按 **ENTER** 移到下一个元素。按 **↓** 移到下一行。

ENTER

10 **ENTER** 4 **ENTER**

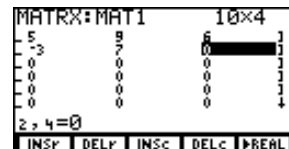
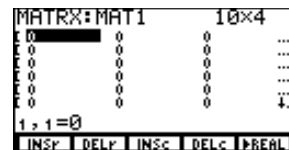
(←) 4 **ENTER** 5

ENTER 9 **ENTER** 6

ENTER 1 **ENTER**

(←) 3 **ENTER** 7

ENTER 等



Matrix Editor (矩阵编辑器) 菜单 **2nd** **[MATRIX]** **F2** 矩阵名 **ENTER**

INSp	DELr	INSc	DELc	▶REAL
-------------	-------------	-------------	-------------	--------------

- INSp** 在光标处插入一行; 其后的行依次下移
- DELr** 在光标处删除一行; 其后的行依次上移
- INSc** 在光标处插入一列; 其后的列依次右移
- DELc** 在光标处删除一列; 其后的列依次左移
- ▶REAL** 将所显示的复数矩阵转换成实数矩阵

在主屏幕中创建矩阵

- ❶ 用【】定义矩阵开始, 然后用另一个【】定义第一行的开始。输入这一行的所有元素, 元素用逗号分隔。用】定义第一行的结束。

2nd [] 2nd []
2 , 4 , 6 ,
8 2nd []

[[2,4,6,8]]

当左括弧在 **STO►** 之前时可以用。

- ❷ 用【】定义后续行的开始。输入行元素, 元素用逗号分隔。用】定义每一行的结束。然后用】定义矩阵结束。

2nd [] (←) 1 ,
(←) 3 , (←) 5 ,
(←) 7 2nd []
2nd []

[[2,4,6,8]] [-1, -3, -5, -7]]

使用 MEM DELETE:MATRX 屏幕 (第 17 章) 从内存删除矩阵名。

- ❸ 保存矩阵到矩阵名。输入一个长度为 1 到 8 个字符, 且以字母开头的名称; 或从 MATRX NAMES 菜单中选择。显示矩阵。若矩阵名是新创建的, 则成为 MATRX NAMES 菜单项

STO► 2nd [alpha]
[M][A][T]
ALPHA ALPHA 1
ENTER

[[2,4,6,8]] [-1, -3, -5, -7]] → mat1
[[2 4 6 8]
[-1 -3 -5 -7]]

创建复数矩阵

只要有一个矩阵元素是复数, 所有矩阵元素就以复数形式显示。例如, 当输入矩阵 **[[1,2][5,(3,1)]]** 时, TI-86 显示 **[[(1,0) (2,0)] [(5,0) (3,1)]]**。

用相同维数的两个实数矩阵创建一个复数矩阵的语法是:

realMatrix + (0,1) *imaginaryMatrix* → *complexMatrixName*

realMatrix 包含每个元素的实部; *imaginaryMatrix* 包含每个元素的虚部。

可用 $\leftarrow$ 、 $\rightarrow$ 、 $\uparrow$ 和 $\downarrow$ 查看当前屏幕外的元素。

显示矩阵元素、行和子矩阵

要在主屏幕上显示现有的矩阵，则输入矩阵名，或从 **MATRIX NAMES** 菜单选择矩阵名，然后按 **[ENTER]**。显示每个元素的精确值。非常大的元素值可能以指数形式显示。

显示矩阵 *matrixName* 中指定元素的语法是：

matrixName(row,column)

显示矩阵 *matrixName* 中某行的语法是：

matrixName(row)

显示矩阵 *matrixName* 子矩阵的语法是：

matrixName(beginRow,beginColumn,endRow,endColumn)

```

[[ -4  5  9  6]
 [ 1 -3  7  0]
 [ 0  0  0  0]
 [ 0  0  0  0]
 [ 0  0  0  0]
 [ 0  0  0  0]↓
MAT1(2,2)                -3
MAT1(2)                   [1 -3 7 0]
MAT1(1,2,2,3)             [[ 5  9]
                           [-3 7]]

```

在表达式中使用矩阵

- ◆ 可以直接输入矩阵（例如， $5 \cdot [[2,3][3,5]]$ ）。
- ◆ 可用 **[ALPHA]** 和 **[2nd]** [alpha] 输入矩阵名的每个字符（例如，**MAT1*3**）。
- ◆ 可从 **MATRIX NAMES** 菜单选择矩阵名 (**[2nd]** **[MATRIX]** **[F1]**)。
- ◆ 可从 **VARS MATRIX** 屏幕选择矩阵名 (**[2nd]** **[CATLG-VARS]** **[MORE]** **[F2]**)。

当表达式执行时，结果以矩阵形式显示。

在矩阵编辑器中编辑矩阵

- 1 显示矩阵 **Name=** 提示屏幕。
- 2 输入矩阵名。或从 **MATRIX NAMES** 菜单选择矩阵名, 或输入字符。
- 3 显示矩阵编辑器。
- 4 编辑或接受行维数, 然后编辑或接受列维数。
- 5 将光标移动到任意元素并编辑它。继续移动光标到其他元素。
- 6 保存改变并退出矩阵编辑器。

[2nd] [MATRIX] [F2]

[M][A][T]

[ALPHA] 1

[ENTER]

5 [DEL] [ENTER]

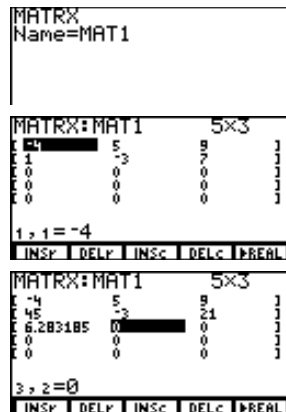
3 [ENTER]

[▼] 45 [ENTER] [▶]

21 [ENTER] 2 [2nd]

[π] [ENTER]

[EXIT]



可用 [CLEAR]、[DEL] 和 [2nd] [INS] 编辑矩阵元素。也可以改写现有的字符。

在主屏幕编辑矩阵

改变矩阵元素值的语法是:

value→*matrixName*(*row,column*)

改变某行所有元素值的语法是:

[*valueA,valueB,...,value n*]→*matrixName*(*row*)

改变某行中从指定列开始的部分元素值的语法是:

[*valueA,valueB,...,value n*]→*matrixName*(*row,beginColumn*)

改变矩阵 *matrixName* 中子矩阵元素值的语法是:

[[*valueA,...,value n*] ... [*valueA,...,value n*]]→*matrixName*(*beginRow,beginColumn*)

MATRIX MATH (矩阵数学运算) 菜单 2nd [MATRIX] [F3]

NAMES	EDIT	MATH	OPS	CPLX					
det	τ	norm	eigVl	eigVc	►	rnorm	cnorm	LU	cond

det <i>squareMatrix</i>	计算方阵 <i>squareMatrix</i> 的行列式值
<i>matrix</i> ^T	返回转置矩阵; 每个元素的坐标 (row,column) 互换
norm <i>matrix</i>	返回 Frobenius 范数 ($\sqrt{\Sigma(\text{real}^2+\text{imaginary}^2)}$), 其中求和是对实数或复数矩阵 <i>matrix</i> 中的所有元素
eigVl <i>squareMatrix</i>	返回一个数组, 其元素是实数或复数方阵 <i>squareMatrix</i> 的标准特征值
eigVc <i>squareMatrix</i>	返回一个矩阵, 包括实数或复数方阵 <i>squareMatrix</i> 的特征向量; 每列对应一个特征值
rnorm <i>matrix</i>	(行范数) 计算矩阵 <i>matrix</i> 每行中所有元素的绝对值 (复数元素为模) 和, 返回其中的最大值
cnorm <i>Matrix</i>	(列范数) 计算矩阵 <i>matrix</i> 每列中所有元素的绝对值 (复数元素为模) 和, 返回其中的最大值
LU (<i>matrix</i> , <i>lMatrixName</i> , <i>uMatrixName</i> , <i>pMatrixName</i>)	计算实数或复数矩阵 <i>matrix</i> 的 Crout LU (下三角—上三角) 分解; 保存下三角矩阵到 <i>lMatrixName</i> , 保存上三角矩阵到 <i>uMatrixName</i> , 同时保存置换矩阵 (它描述计算时的行交换) 到 <i>pMatrixName</i>
cond <i>squareMatrix</i>	Calculates cnorm (方阵) <i>squareMatrix</i> * cnorm (方阵) <i>squareMatrix</i> ⁻¹ ; 乘积越接近1, 方阵 <i>squareMatrix</i> 在矩阵函数中越稳定

MATRX OPS (矩阵运算) 菜单 [2nd] [MATRX] [F4]

NAMES	EDIT	MATH	OPS	CPLX												
dim	Fill	ident	ref	rref	▶ <table><tr><td>aug</td><td>rSwap</td><td>rAdd</td><td>multR</td><td>mRAdd</td></tr><tr><td>▶ <table><tr><td>randM</td><td></td><td></td><td></td><td></td></tr></table></td></tr></table>	aug	rSwap	rAdd	multR	mRAdd	▶ <table><tr><td>randM</td><td></td><td></td><td></td><td></td></tr></table>	randM				
aug	rSwap	rAdd	multR	mRAdd												
▶ <table><tr><td>randM</td><td></td><td></td><td></td><td></td></tr></table>	randM															
randM																

按 [STO▶] 在右括号之后输入
→ 符号。

若使用 **aug(** , 则矩阵 *matrixA*
的行数必须等于矩阵 *matrixB*
的行数, 或等于向量 *vector* 中
的元素个数。

用 **randM(** 创建的矩阵元素是
≥-9 且 ≤9 的整数。

dim <i>matrix</i>	以数组 { 行数, 列数 } 的形式返回矩阵 <i>matrix</i> 的维数
{rows,columns}→dim <i>matrixName</i>	新建指定维数的矩阵 <i>matrixName</i> ; 每个元素为 0
{rows,columns}→dim <i>matrixName</i>	重新设置矩阵 <i>matrixName</i> 到指定的维数
Fill (<i>number,matrixName</i>)	将实数或复数 <i>number</i> 存储到矩阵 <i>matrixName</i> 的每个元素
ident <i>dimension</i>	返回 <i>dimension</i> × <i>dimension</i> 的方阵
ref <i>matrix</i>	返回矩阵 <i>matrix</i> 的行梯形矩阵
rref <i>matrix</i>	返回矩阵 <i>matrix</i> 的简化行梯形矩阵
aug (<i>matrixA,matrixB</i>)	合并矩阵 <i>matrixA</i> 和矩阵 <i>matrixB</i>
aug (<i>matrix,vector</i>)	合并矩阵 <i>matrix</i> 和向量 <i>vector</i>
rSwap (<i>matrix,rowA,rowB</i>)	交换矩阵 <i>matrix</i> 的行 <i>rowA</i> 和行 <i>rowB</i> 后, 返回矩阵
rAdd (<i>matrix,rowA,rowB</i>)	矩阵 <i>matrix</i> 的行 <i>rowA</i> 与行 <i>rowB</i> 相加, 结果保存在行 <i>rowB</i> 后, 返回矩阵 <i>matrix</i>
multR (<i>number,matrix,row</i>)	矩阵的行 <i>row</i> 乘以数 <i>number</i> , 结果保存在行 <i>row</i> 后, 返回矩阵 <i>matrix</i>
mRAdd (<i>number,matrix,rowA,rowB</i>)	((<i>rowA*number</i>)+ <i>rowB</i>) 的结果保存在 <i>rowB</i> 后。返回矩阵 <i>matrix</i>
randM (<i>rows,columns</i>)	用随机产生的元素创建指定维数的矩阵

MATRIX CPLX (复数矩阵) 菜单 [2nd] [MATRIX] [F5]

NAMES	EDIT	MATH	OPS	CPLX
conj	real	imag	abs	angle

conj <i>complexMatrix</i>	返回一个矩阵，其每个元素是复数矩阵 <i>complexMatrix</i> 中对应元素的共轭复数
real <i>complexMatrix</i>	返回一个实数矩阵，其每个元素是复数矩阵 <i>complexMatrix</i> 中对应元素的实部
imag <i>complexMatrix</i>	返回一个实数矩阵，其每个元素是复数矩阵 <i>complexMatrix</i> 中对应元素的虚部
abs <i>matrix</i>	返回一个实数矩阵，其每个元素是实数矩阵 <i>matrix</i> 中对应元素的绝对值或复数矩阵 <i>matrix</i> 对应元素的模
angle <i>complexMatrix</i>	返回一个实矩阵，其每个元素当复数矩阵 <i>complexMatrix</i> 中对应元素是实数时为 0 ；当复数矩阵 <i>complexMatrix</i> 中对应元素是复数时为极角；极角用 $\tan^{-1}(\text{imaginary} / \text{real})$ 计算，在第二象限、第三象限时分别用 $+\pi$ 、 $-\pi$ 调整

矩阵的数学运算

矩阵 <i>matrixA</i> 和矩阵 <i>matrixB</i> 进行加减时，它们的维数必须相等。	<i>matrixA</i> + <i>matrixB</i>	矩阵 <i>matrixA</i> 的每个元素和矩阵 <i>matrixB</i> 的对应元素相加，返回和的矩阵
	<i>matrixA</i> - <i>matrixB</i>	矩阵 <i>matrixA</i> 的每个元素减去矩阵 <i>matrixB</i> 的对应元素，返回差的矩阵

两个矩阵进行相乘时, 矩阵 $matrixA$ 的列维数必须等于矩阵 $matrixB$ 的行维数。

按 $\boxed{2nd} \boxed{[x^{-1}]}$ 输入 $^{-1}$ 。不要使用 $\boxed{x-VAR} \boxed{\square} \boxed{(-)} \boxed{1}$ 。

$e^{\wedge}$ 、 $\sin$ 和 $\cos$ 并不返回每个矩阵元素的指数值、正弦值或余弦值。

矩阵 $matrixA$ 和矩阵 $matrixB$ 进行关系运算时它们的维数必须相同。

$==$ 和 $\neq$ 在 TEST 菜单。

round 、 iPart 、 fPart 和 int 在 MATH NUM 菜单。

$matrixA * matrixB$ or
 $matrixB * matrixA$

$matrix * number$ or
 $number * matrix$

$matrix * vector$

$\sim matrix$

$squareMatrix^{-1}$

$matrix^2$

$squareMatrix^{power}$

$e^{\wedge} squareMatrix$

$\sin squareMatrix$

$\cos squareMatrix$

$matrixA == matrixB$

$matrixA \neq matrixB$

$\text{round}(matrix[, \#ofDecimals])$

$\text{iPart } matrix$

$\text{fPart } matrix$

$\text{int } matrix$

矩阵 $matrixA$ 和矩阵 $matrixB$ 相乘, 返回积的方阵

返回一个矩阵, 其每个元素是实数或复数矩阵 $matrix$ 中对应元素与实数或复数 $number$ 的乘积

返回一个向量, 其每个元素是向量 $vector$ 的每个元素与矩阵 $matrix$ 的每个元素的乘积, 矩阵 $matrix$ 的列维数必须等于向量 $vector$ 的维数

(取反) 改变矩阵 $matrix$ 中每个元素的符号

返回方阵 $squareMatrix$ 的逆矩阵 (不是每个元素的倒数)

求方阵的平方

求方阵 $squareMatrix$ 的 $power$ 次方

返回实方阵 $squareMatrix$ 的方阵指数

返回实方阵 $squareMatrix$ 的方阵正弦

返回实方阵 $squareMatrix$ 的方阵余弦

如果每个对应元素比较为真, 返回 **1**; 否则返回 **0**

如果有一对元素比较为假, 则返回 **1**

矩阵 $matrix$ 的每个元素四舍五入为 12 位, 或四舍五入到指定的小数位数 $\#ofDecimals$

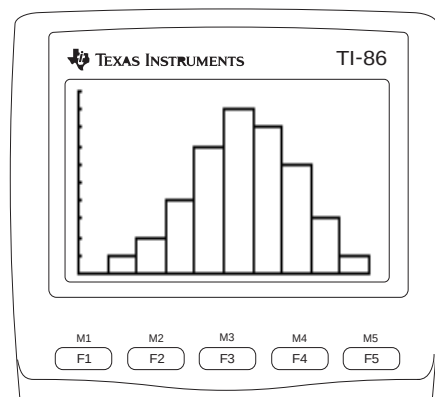
返回实数或复数矩阵 $matrix$ 中每个元素的整数部分

返回实数或复数矩阵 $matrix$ 中每个元素的小数部分

返回实数或复数矩阵 $matrix$ 中每个元素的最大整数

14 统计

TI-86 的统计分析.....	188
建立统计分析.....	188
统计分析的结果.....	192
绘制统计数据.....	194
STAT DRAW（统计图）菜单.....	199
预测统计数据值.....	199



TI-86 的统计分析

TI-86 可分析保存在数组中的一元和二元统计数据。一元数据有一个测量变量。二元数据有一对变量，即自变量和因变量。

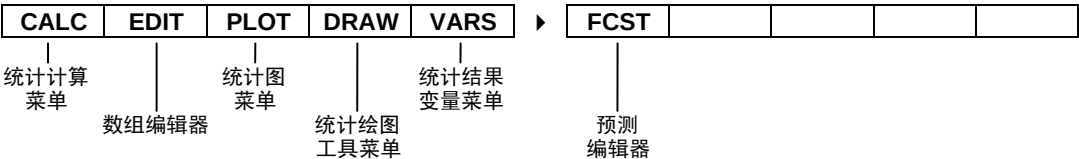
当分析任何一种数据时，可指定自变量值的出现频数。指定的频数必须是 ≥ 0 的实数。

建立统计分析

- ❶ 在一个或多个数组（第 11 章）中输入统计数据。
- ❷ 计算统计变量或用一种模型拟合这些数据。
- ❸ 绘制这些数据。
- ❹ 绘制数据的回归方程。

STAT（统计）菜单 [2nd][STAT]

按 [2nd][STAT][F2] 或 [2nd][LIST][F4]，都显示相同的数组编辑器。请参阅第 11 章中有关数组编辑器的描述。



输入统计数据

用于统计分析的数据保存在数组中，数组可在数组编辑器（第 11 章）、主屏幕（第 11 章）或程序（第 16 章）中创建和编辑。TI-86 中有三个内置的用于统计的数组名，即 **xStat**（变量 **x** 数组）、**yStat**（变量 **y** 数组）和 **fStat**（频数数组）。TI-86 的统计函数将这些数组作为默认输入。

LIST NAMES（数组名）菜单 $\boxed{2\text{nd}} \boxed{[\text{STAT}]} \boxed{[\text{F2}]} \boxed{[\text{F3}]}$

此处显示的 LIST NAMES 菜单没有用户创建的数组名。

{	}	NAMES	EDIT	OPS
fStat	xStat	yStat		

fStat 数组，自动更新为上次统计计算时所用的频数值，其所有元素的默认值均为 **1**

xStat 数组，自动更新为上次统计分析时所用 **x** 数组的数据

yStat 数组，自动更新为上次统计分析时所用 **y** 数组的数据

编辑 **xStat** 或 **yStat** 的元素
将清除统计结果变量中的所有值。

STAT CALC（统计计算）菜单 $\boxed{2\text{nd}} \boxed{[\text{STAT}]} \boxed{[\text{F1}]}$

CALC	EDIT	PLOT	DRAW	VARS						
OneVa	TwoVa	LinR	LnR	ExpR	▶ <table><tr><td>PwrR</td><td>SinR</td><td>LgstR</td><td>P2Reg</td><td>P3Reg</td></tr></table>	PwrR	SinR	LgstR	P2Reg	P3Reg
PwrR	SinR	LgstR	P2Reg	P3Reg						
					▶ <table><tr><td>P4Reg</td><td>StReg</td><td></td><td></td><td></td></tr></table>	P4Reg	StReg			
P4Reg	StReg									

STAT CALC 函数将结果保存到统计结果变量（第 193 页）。

每个 STAT CALC 菜单项的语法描述如下。

OneVa （一元）用一个测量变量分析数据

TwoVa （二元）分析成对数据

对于回归分析，用最小二乘法
计算统计结果。

LinR	(线性回归) 用模型方程 $y=a+bx$ 拟合数据; 显示 a (斜率) 和 b (y 截距) 的值
LnR	(对数回归) 使用变换值 $\ln(x)$ 和 y , 用模型方程 $y=a+b \ln x$ 拟合数据; 显示 a 和 b 的值
ExpR	(指数回归) 使用变换值 x 和 $\ln(y)$, 用模型方程 $y=ab^x$ 拟合数据, 显示 a 和 b 的值; 数组 x 和数组 y 的元素必须为整数
PwrR	(幂回归) 使用变换值 $\ln(x)$ 和 $\ln(y)$, 用模型方程 $y=ax^b$ 拟合数据, 显示 a 和 b 的值
SinR	(正弦回归) 用模型方程 $y=a*\sin(bx+c)+d$ 拟合数据; 显示 a 、 b 、 c 和 d 的值。 SinR 要求至少 4 个数据点; 它还要求每个周期至少两个点以避免频率估计值出现偏差
LgstR	(逻辑回归) 用模型方程 $y=a/(1+be^{cx})+d$ 拟合数据; 显示 a 、 b 、 c 和 d 的值
P2Reg	(二次回归) 用二次多项式 $y=ax^2+bx+c$ 拟合数据; 显示 a 、 b 和 c 的值; 对于三个数据点, 方程是多项式拟合; 对于四个或更多个点, 它是多项式回归; P2Reg 要求至少三个数据点
P3Reg	(三次回归) 用三次多项式 $y=ax^3+bx^2+cx+d$ 拟合数据; 显示 a 、 b 、 c 和 d 的值; 对于四个数据点, 方程是多项式拟合; 对于五个或更多个点, 它是多项式回归; P3Reg 要求至少四个数据点
P4Reg	(四次回归) 用四次多项式 $y=ax^4+bx^3+cx^2+dx+e$ 拟合数据; 显示 a 、 b 、 c 、 d 和 e 的值; 对于五个数据点, 方程是多项式拟合; 对于六个或更多个点, 它是多项式回归; P4Reg 要求至少五个数据点
StReg	(保存回归方程) 将 StReg (粘贴到主屏幕; 输入 变量 并按 [ENTER] ; 当前回归方程保存到该变量中

用迭代最小二乘拟合计算
SinR 和 **LgstR** 。

当选择 **OneVa** 或 **TwoVa** 时, 显示 **OneVar** 或 **TwoVar** 的缩写。

对于 **PwrR** 和 **ExpR**, $xList$ 和 $yList$ 的元素必须为 ≥ 1 的整数。

iterations 的默认值为 64。

OneVa 的语法是:

OneVar [$xList, frequencyList$]

TwoVa 的语法是:

TwoVar [$xList, yList, frequencyList$]

LinR、**LnR**、**ExpR**、**PwrR**、**P2Reg**、**P3Reg** 和 **P4Reg** 的语法是:

TwoVar [$xList, yList, frequencyList$]

SinR 的语法是:

SinR [$iterations, xList, yList, period, equationVariable$]

iterations 表示迭代次数; 迭代次数越多, 拟合效果越好, 但花费的时间更长。 *period* 是开始计算时所用的初始估计值。

LgstR 的语法是:

LgstR [$iterations, xList, yList, frequencyList, equationVariable$]

在回归计算后将 **RegEq** 的内容拷贝到变量的语法是:

StReg(*variable*)

自动回归方程存储

LinR、**LnR**、**ExpR**、**PwrR**、**SinR**、**LgstR**、**P2Reg**、**P3Reg** 和 **P4Reg** 是回归模型, 每个回归模型都有一个可选参数 *equationVariable*, 可给它指定一个方程变量, 如 **y1**。执行后回归方程自动存储到指定的方程变量中, 并且该函数被选择。

无论是否指定了 *equationVariable*, 回归方程总是存储到结果变量 **RegEq**, 该变量是 STAT VARS 菜单中的项。回归方程显示实际的结果值。

PRegC 是多项式回归所计算的唯一统计结果变量。

一元和二元统计函数使用相同的结果变量。

统计变量的计算和保存如下页中的表格所示。

可以用大写字母键、小写字母键和 CHAR GREEK 菜单输入一些结果变量。

多项式回归、正弦回归或对数回归的结果保存在 **PRegC** (多项式/回归系数) 中。 **PRegC** 是一个包含方程系数的数组。例如, 对于 **P3Reg**, 结果 **PRegC={3 5 -2 7}** 表示 $y=3x^3+5x^2-2x+7$ 。

统计分析的结果

进行统计分析时, 计算结果保存在结果变量中, 分析所用数组的数据保存在 **xStat**、**yStat** 和 **fStat** 中。若编辑数组或改变分析的类型, 所有统计变量将被清除。

STAT VARS (统计变量) 菜单 **2nd** **[STAT]** **F5**

CALC	EDIT	PLOT	DRAW	VARS						
$\bar{x}$	σx	Sx	$\bar{y}$	σy	▶	Sy	Σx	Σx^2	Σy	Σy^2
					▶	Σxy	RegEq	corr	a	b
					▶	n	minX	maxX	minY	maxY
					▶	Med	PRegC	Qrtl1	Qrtl3	tolMe

要将结果变量粘贴到光标处, 则从 **STAT VARS** 菜单, 或从 **VARS STAT** 选择屏幕选择该变量。

- ◆ 若要在表达式中使用结果变量, 则将它粘贴到适当的光标位置。
- ◆ 若要显示结果变量的值, 则将它粘贴到主屏幕, 然后按 **[ENTER]**。
- ◆ 若要在计算之后把结果保存到其他变量, 则将结果变量粘贴到主屏幕, 按 **[STO▶]**, 输入一个新变量, 然后按 **[ENTER]**。

这些词在表格中采用缩写形式:
pop = population (总体)
std dev = standard deviation (标准差)
coeff = coefficient (系数)
int = intercept (截距)
reg eq = regression equation (回归方程)
pts = points (点)
min = minimum (最小值)
max = maximum (最大值)

结果 变量	一元 统计	二元 统计	其他
x 的平均值	$\bar{x}$	$\bar{x}$	
x 的总体标准偏差	σx	σx	
x 的样本标准偏差	Sx	Sx	
y 的平均值		$\bar{y}$	
y 的总体标准偏差		σy	
y 的样本总体偏差		Sy	
x 值的总和	Σx	Σx	
x ² 值的总和	Σx^2	Σx^2	
y 值的总和		Σy	
y ² 值的总和		Σy^2	
x * y 的总和		Σxy	
回归方程			RegEq
多项式、 LgstR 和 SinR 系数			a (y 截距) b (斜率)

结果 变量	一元 统计	二元 统计	其他
相关系数			corr
回归方程的 y 截距			a
回归方程的斜率			b
回归/拟合系数			a, b
数据点个数	n	n	
x 的最小值	minX	minX	
x 的最大值	maxX	maxX	
x 的最大值		minY	
y 的最大值		maxY	
中位值	Med		
第一个四分位数			Qrtl1
第三个四分位数			Qrtl3
多项式 LgstR 和 SinR 回归系数			PRegC

第一个四位数 (**Qrtl1**) 是 **minX** 和 **Med** (中位值) 之间的中位值。第三个四位数 (**Qrtl3**) 是 **Med** 和 **maxX** 之间的中位值。

当计算逻辑回归时, 如果在计算出结果前已满足 TI-86 的内部误差, 则 **1** 保存到 **tolMet** (**tolMe**); 否则, **0** 保存到 **tolMet**。

绘制统计数据

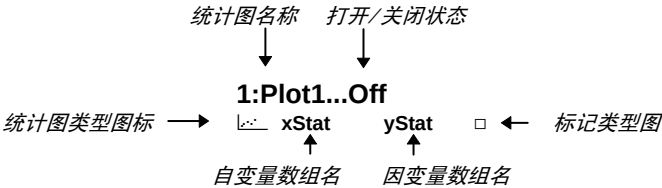
可以绘制 1、2 或 3 组统计数据。五种可用图类型是散点图、xy 直线图、直方图、改进式箱线图和标准箱线图。

- 1 保存统计数据到一个或几个数组中（第 11 章）。
- 2 适当选择或取消选择当前方程编辑器中的函数（第 5 章）。
- 3 定义统计图。
- 4 开启要显示的统计图。
- 5 定义图形屏幕的窗口变量（第 5 章）。
- 6 显示和浏览所绘制的图形（第 6 章）。

STAT PLOT 状态屏幕 [2nd] [STAT] [F3]

STAT PLOT 状态屏幕汇总了 **Plot1**、**Plot2** 和 **Plot3** 的设置。下面的图示说明 **Plot1** 的设置。这个屏幕不是交互式的。要改变设置，则从 STAT PLOT 状态屏幕菜单选择 **PLOT1**、**PLOT2** 或 **PLOT3**。

这个屏幕是默认统计图设置。若选择了其他的统计图类型，一些提示可能改变。



在显示统计图编辑器时，
STAT PLOT 菜单仍保留，
这样能容易地切换到其他
统计图。

在本指南中，语法中的括号
（ [和] ）指定参数为可选。
除非是向量和矩阵，不要输入
括号。

无须打开统计图即可改变
设置。

也可以使用 STAT PLOT 菜单
项 **PIOn** 或 **PIOff** 打开或关闭
统计图。

STAT PLOT （统计图）菜单 $\boxed{2nd}$ $\boxed{STAT}$ $\boxed{F3}$

PLOT1	PLOT2	PLOT3	PIOn	PIOff
-------	-------	-------	------	-------

PLOT1 显示 **Plot1** 的统计图编辑器

PLOT2 显示 **Plot2** 的统计图编辑器

PLOT3 显示 **Plot3** 的统计图编辑器

PIOn [1,2,3] 打开所有统计图（若未输入参数）或仅打开指定统计图

PIOff [1,2,3] 关闭所有统计图（若未输入参数）或仅关闭指定统计图

要打开或关闭所有三个统计图，则从 STAT PLOT 菜单选择 **PIOn** 或 **PIOff**。**PIOn** 或 **PIOff** 粘贴到主屏幕，按 $\boxed{ENTER}$ 。所有统计图立即打开或关闭。

建立统计图

要设置统计图，从 STAT PLOT 菜单选择 **PLOT1**、**PLOT2** 或 **PLOT3**。显示用于所选统计图的编辑器。每种统计图类型有唯一的统计图编辑器。右侧屏幕显示的是默认类型 $\boxed{L\wedge}$ （散点图）的编辑器。若选择其他类型，某些提示可能改变。



打开和关闭统计图

当显示统计图编辑器时，光标处于 **On** 上。

- ◆ 要打开统计图，则按 $\boxed{ENTER}$ 。
- ◆ 要关闭统计图，则按 $\boxed{\blacktriangleright}$ $\boxed{ENTER}$ 。

要显示 PLOT TYPE 菜单，则将光标移到 **Type=** 提示处的统计图类型图标上。

选择一种统计图类型后，统计图编辑器的外观可能改变。

PLOT TYPE（统计图类型）菜单（选择统计图类型）

PLOT1	PLOT2	PLOT3	PIOn	PIOff
SCAT	xyLINE	MBOX	HIST	BOX

在提示处...	输入信息:	默认值:	显示菜单:
Xlist Name=	自变量数据的数组名	xStat	LIST NAMES 菜单
Ylist Name=	因变量数据的数组名	yStat	LIST NAMES 菜单
Freq=	频数的数组名（或 1）	fStat（默认值： 1）	LIST NAMES 菜单
Mark=	统计图标记（□ 或 + 或 •）	□（对于 HIST 或 BOX 则没有）	PLOT MARK 菜单

- ◆ 在 Xlist Name= 提示处输入的任何数组保存到数组名 xStat。
- ◆ 在 Ylist Name= 提示处输入的任何数组保存到数组名 yStat。
- ◆ 在 Freq= 提示处输入的任何数组保存到数组名 fStat。

统计图类型特征

正如窗口变量值（第 5 章）所定义的，统计图显示在图形屏幕 (GRAPH) [FS]。某些图象工具可应用于统计图。

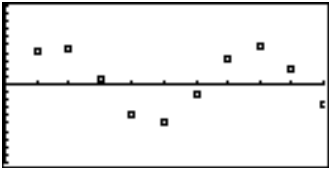
在这些统计图示例中，取消选择所有函数。菜单也用 [CLEAR] 从屏幕中清除。

SCAT（散点图）成对绘制 Xlist Name 和 Ylist Name 中的数据，每个点用框 (□)、十字 (+) 或点 (•) 表示。Xlist Name 和 Ylist Name 的长度必须相同。Xlist Name 和 Ylist Name 可以是一个数组。

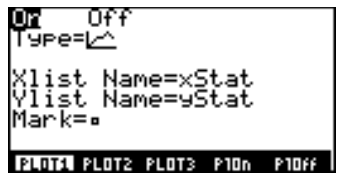


例如：
xStat={1 2 3 4 5 6 7 8 9 10}
yStat=5 sin(xStat)

窗口变量值：
xMin=0 yMin=-10
xMax=10 yMax=10

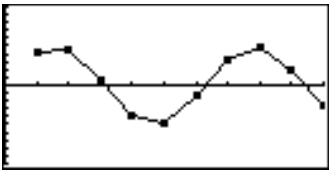


 **xyLINE** 是一种散点图，图中绘出每个数据点，并按数据点在 **Xlist Name** 和 **Ylist Name** 中出现的顺序将它们连接起来。可能要在绘制数组前使用 **LIST OPS** 菜单（第 11 章）中的 **SortA** 或 **SortD** 对它们进行排序。

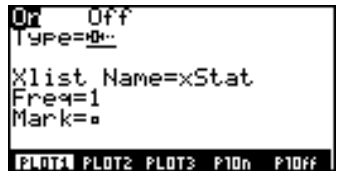


例如:
 $xStat=\{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\}$
 $yStat=5 \sin(xStat)$

窗口变量值:
 $xMin=0$ $yMin=-10$
 $xMax=10$ $yMax=10$

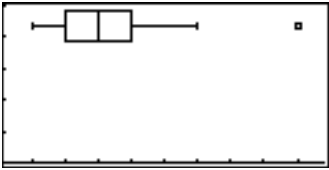


 **MBOX**（改进箱线图）如标准箱线图一样绘制一元数据，但不包括超过四分位数 $1.5 * \text{四分位数间距}$ 的点。（四分位数间距定义为第三个四分位数 Q_3 和第一个四分位数 Q_1 之差。）用所选的 **Mark**（□ 或 + 或 •）在点触线后单独绘制这些点。



例如:
 $xStat=\{1\ 2\ 2\ 2.5\ 3\ 3.3\ 4\ 4\ 2\ 6\ 9\}$

通过从 **GRAPH ZOOM** 菜单选择 **ZDATA** 设置窗口变量。

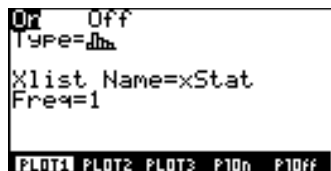


点触线是指突出箱线的线。

可以跟踪这些被称为局外点的点。如果局外点存在，则点触线的末端显示 **x=** 提示；否则显示 **xMin** 和 **xMax**。 Q_1 、**Med**（中位值）和 Q_3 定义箱线。

改进箱线图是根据 **xMin** 和 **xMax** 绘制的，但忽略了 **yMin** 和 **yMax**。当绘制了两个改进箱线图时，第一个处于屏幕的顶部；而第二个处于屏幕的中央。当绘制了三个改进箱线图时，第一个处于屏幕的顶部，第二个处于屏幕的中央，第三个处于屏幕的底部。

 **HIST** (直方图) 绘制一元数据。xScl 窗口变量的值决定每个条的宽度, 条从 xMin 处开始。ZDATA (GRAPH ZOOM 菜单) 调整 xMin、xMax、yMin 和 yMax 以包括所有值, 同时也调整 xScl。 $(xMax - xMin) / xScl \leq 47$ 必须为真。在条边缘出现的值被记入右面的条。

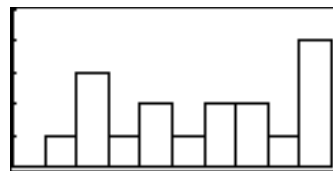


例如:

xStat={1 2 2 2 3 8 9 5 6 6 7 7 4
4 9 9 9}

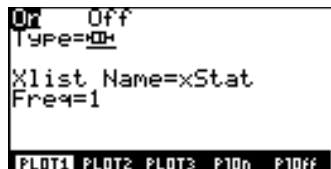
窗口变量值:

xMin=0 yMin=0
xMax=10 yMax=5



点触线是指突出箱线边线的线。

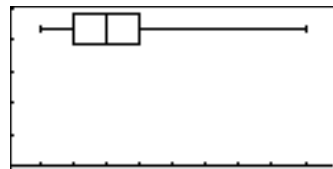
 **BOX** (标准箱线图) 绘制一元数据。图上的点触线从这组数据中的最小点 (xMin) 延伸到第一个四分位数 (Q_1), 从第三个四分位数 (Q_3) 延伸到最大点 (xMax)。箱线由 Q_1 、Med (中位值) 和 Q_3 定义。



例如:

xStat={1 2 2 2.5 3 3.3 4 4 2 6 9}

通过从 GRAPH ZOOM 菜单选择
ZDATA 设置窗口变量值。



箱线图是根据 xMin 和 xMax 绘制的, 但忽略了 yMin 和 yMax。当绘制了两个箱线图时, 第一个处于屏幕的顶部; 而第二个处于屏幕的中央。当绘制了三个箱线图时, 第一个处于屏幕的顶部, 第二个处于屏幕的中央, 第三个处于屏幕的底部。

STAT DRAW (统计图) 菜单 2nd [STAT] F4

CALC	EDIT	PLOT	DRAW	VARS						
HIST	SCAT	xyLINE	BOX	MBOX	▶	DRREG	CLDRW	DrawF	STPIC	RCPIC

如果选择前五个 STAT DRAW 菜单项中的任意一个, TI-86 将绘制存储在 **xStat** 和 **yStat** 数组中的数据。

HIST	绘制一元数据的直方图
SCAT	绘制数据点的散点图
xyLINE	绘制数据点并用直线连接相邻两个点
BOX	绘制数据点的箱线图
MBOX	绘制数据点的改进箱线图
DRREG	(绘制回归方程) 绘制当前回归方程
CLDRW	(清除图形) 显示当前图象, 其中没有图形
DrawF	<i>expression</i> (绘制函数) 将表达式 <i>expression</i> 绘制成图形
STPIC	(保存图形) 显示图形变量 Name= 提示; 输入以字母开头的有效变量名, 然后按 ENTER 保存当前图形
RCPIC	(调用图形) 显示图形变量 Name= 提示和菜单; 选择或输入有效变量名, 然后按 ENTER ; 重绘制保存的图形

预测统计数据值

使用预测编辑器, 可以预测基于当前回归方程的 **x** 值或 **y** 值。若要使用预测编辑器, 回归方程则必须保存到 **RegEq**。

- ① 在数组编辑器输入统计数据。右边屏幕显示 **fStat** 的所有元素为 **1**，无须输入它们。**1** 是 **fStat** 中所有元素的默认值。不过，如果其他元素已被存入 **fStat**，则必须清除它们。

[2nd] [STAT] [F2]
 [.] 1 [v] 1 [.] 1
 [v] 2 [v] 4 [v] 5
 [v] 1 [v] 2
 [v] 3 [v] 4 [v] 2

xStat	yStat	fStat	3
1	1	1	
1	1	1	
2	1	1	
4	1	1	
5	1	1	

fStat(6) =

< > NAMES " OPS

- ② 显示主屏幕。
- ③ 对 **xStat** 和 **yStat** 执行线性回归。统计结果显示。

[EXIT]
 [2nd] [STAT] [F1]
 [F3] [ENTER]

```
LinReg
y=a+bx
a=1.65548262
b=.305130075
corr=.54274108
n=5
```

■

CALC EDIT PLOT DRAW VARS

- ④ 删除 STAT CALC 菜单来显示所有结果，包括 **n**。

[EXIT]

- ⑤ 显示预测编辑器。当前回归模型显示在顶行。

[MORE] [F1]

FORECAST:LinReg
 x=3
 y=■

■ ■ ■ ■ SOLVE

- ⑥ 输入 **x=3**，然后移动光标到 **y=** 提示。

3 [v]

FORECAST:LinReg
 x=3
 y=2.5708728422076

■ ■ ■ ■ SOLVE

- ⑦ 从预测编辑器菜单选择 **SOLVE** 来求解 **x=3** 时 **y** 的值。小方块表示结果。可继续使用预测编辑器求解其他 **x** 或 **y** 的值。

[F5]

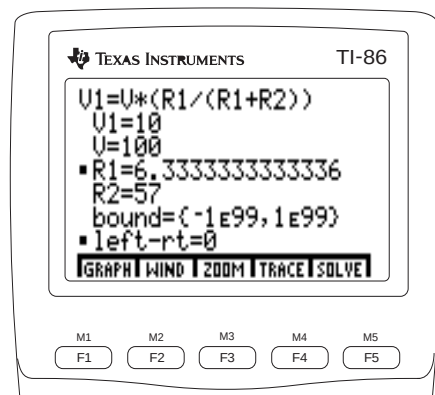
在预测编辑器提示处输入的值必须是实数，或者是计算结果为实数的表达式。

若最近的计算是多项式回归，则只能预测 **y** 的值。

当使用 **FCST** 时，不更新 **x**、**y** 和 **Ans** 的值。要保存 **x** 值或 **y** 值，将光标移动到要保存值的变量上，按 [STO➡]，在 **Sto** 提示处输入有效变量名，然后按 [ENTER]。

15 方程求解

预览：方程求解器.....	202
在方程输入编辑器中输入方程.....	203
建立交互式求解器编辑器.....	204
求解未知变量.....	206
绘制结果.....	207
求解器图象工具.....	207
联立方程求解器.....	208
多项式求根器.....	211



预览：方程求解器 [2nd] [SOLVER]

在方程求解器中，可输入一个表达式或方程，存储数值到表达式或方程的所有变量中，除一个变量外，然后求解未知变量。下述步骤介绍此求解器。请阅读本章获得详细信息。

VARs EQU 菜单是 **VARs EQU** 屏幕（第 2 章）的菜单。

示例使用了一个分压器公式。

R1 和 **R2** 代表电阻。

V 和 **V1** 代表电压。

要求解主屏幕或程序编辑器中方程的未知变量，请从 **CATALOG**（“A 到 Z 参考”）中选择 **Solver**。

- 1 显示方程输入编辑器。VARs EQU 菜单显示在屏幕底部。

- 2 输入方程。按 **[ENTER]** 后，显示交互式求解器编辑器和求解器菜单。

- 3 输入每个变量的值，除了未知变量 **R1**。某些变量中可能已存储有值。

- 4 移动光标到要求解的变量处。可输入一个估计值。

- 5 求解方程的变量。小方块标记结果变量和方程 **left-rt=0**（方程的左边减去方程的右边）。如果编辑某个数值或离开屏幕，方块消失。

[2nd] [SOLVER]

[ALPHA] [V] 1 [ALPHA]

[=] [ALPHA] [V] []

[ALPHA] [R] 1 [÷] []

[ALPHA] [R] 1 [÷] []

[ALPHA] [R] 2 [] []

[ENTER]

10 [] 100 [] [] 57

[]

[F5]

```
Eqn: U1=V(R1/(R1+R2))
U1=
U=
R1=
R2=
bound=(-1e99,1e99)
GRAPH WIND ZOOM TRACE SOLVE
```

```
U1=V(R1/(R1+R2))
U1=10
U=100
R1=
R2=57
bound=(-1e99,1e99)
GRAPH WIND ZOOM TRACE SOLVE
```

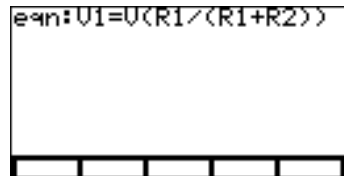
```
U1=V(R1/(R1+R2))
U1=10
U=100
R1=6.33333333333336
R2=57
bound=(-1e99,1e99)
left-rt=0
GRAPH WIND ZOOM TRACE SOLVE
```

在方程输入编辑器中输入方程

方程求解器使用两个编辑器：方程输入编辑器（在其中输入和编辑要求解的方程）和交互式求解器编辑器（在其中输入已知变量的值，选择要求解的变量，并显示结果）。

要显示方程输入编辑器，按 **[2nd] [SOLVER]**。在这个编辑器中，可以：

- ◆ 直接输入方程。
- ◆ 输入已定义方程变量的字母或从 **VARS EQU** 菜单中选择方程变量。
- ◆ 调用已定义方程变量的内容。



方程的等号左边可不止一个变量，如 $A+B=C+\sin D$ 。

在输入或编辑方程时，TI-86 自动将它保存到变量 **eqn** 中。

可在方程输入编辑器中显示其他菜单。

省略号 (...) 说明已输入的方程超出了屏幕。要直接移到方程开始处，按 **[2nd] [←]**；要直接移到末尾，按 **[2nd] [→]**。

VARS EQU 菜单是 **VARS EQU** 屏幕（第 2 章）的菜单。这些菜单项是用来保存方程的所有变量。这包括所有选定的和取消选定的方程变量，这些变量已在四种图象方式（第 5、8、9 和 10 章）的方程编辑器中定义。菜单项按字母顺序排列。

- ◆ 如果从菜单中选择一个方程变量，该变量将粘贴到光标处，按变量名长度覆写字符。
- ◆ 如果按 **[2nd] [RCL]**，则从菜单中选择一个方程变量，然后按 **[ENTER]**，变量内容将插入到光标处。

如果输入一个方程变量，TI-86 自动将它转换为方程 **exp=equationVariable**。如果直接输入一个表达式，TI-86 自动转换表达式为方程 **exp=expression**。

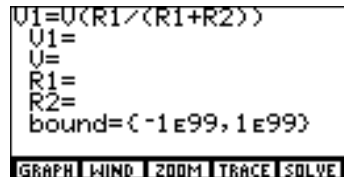
在本例中，方程 $V1=V(R1/(R1+R2))$ 已输入到方程输入编辑器中。

如果为 eqn 输入了一个方程，exp= 将是交互式求解器编辑器中第一个变量提示。

建立交互式求解器编辑器

在方程输入编辑器中存入一个方程到 eqn 后，按 **ENTER** 显示交互式求解器编辑器。

方程显示在编辑器顶部。方程中的每个变量均作为提示显示出来。变量中已存入的值被显示出；未定义的变量显示为空格。求解器菜单显示在屏幕底部（第 206 页）。



bound={-1E99,1E99} 是一个包含默认下限 (**-1E99**) 和默认上限 (**1E99**) 的数组。可编辑这些上下限（如下所示）。

输入变量值

为求解未知变量，必须定义方程中的所有其他变量。在交互式求解器编辑器中输入或编辑变量值时，新值保存到内存中的变量。对于所有变量，可输入表达式，在按 **ENTER**、**▼**、**▲** 或 **EXIT** 后计算表达式的值。在计算的每步，所有表达式的值必须为实数。

用上下限和估计值控制求解

求解器仅在指定范围内求解。无论何时显示交互式求解器编辑器，都将显示默认限值 **bound={-1E99,1E99}**。这是 TI-86 求解的最大范围。

TI-86 迭代求解方程。为控制迭代过程, 可输入接近解的下限和上限, 并在未知变量提示中输入上下限范围内的一个估计值。

用指定的上下限和估计值控制求解过程给 TI-86 提供了下述两方面的帮助。

- ◆ 更快地找到解。
- ◆ 当方程有多个解时, 更有可能找到所要求的解。

在 **bound=** 提示处设置更精确界限值的语法是:

bound={lowerBound,upperBound}

在未知变量的提示处, 可输入一个估计值, 或两个估计值的数组。如果不输入估计值, TI-86 使用 $(lowerBound+upperBound)/2$ 作为估计值。

在求解器图象上 (第 207 页), 可通过移动自由移动光标或跟踪光标到图中 *下限* 和 *上限* 之间的某一点来估计结果。要使用新的估计值求解未知变量, 则从求解器图象菜单选择 **SOLVE**。结果显示在交互式求解器编辑器上。

编辑方程

要在交互式求解器编辑器显示时编辑已存入 **eqn** 的方程, 按 $\boxed{\Delta}$ 直到光标位于该方程上。方程输入编辑器显示。在编辑时, TI-86 自动存储已编辑的方程到 **eqn** 中。

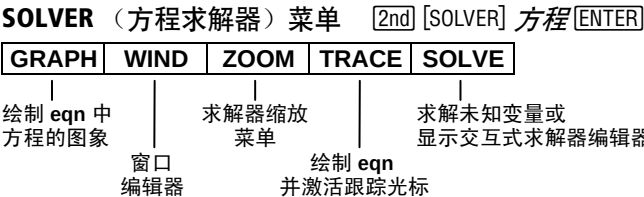
如果通过调用方程变量 (如 **y1**) 的内容将方程存储到 **eqn** 中, 然后编辑存入 **eqn** 的方程, 那么原方程 (例如在 **y1** 中) 不会改变。同样, 接下来编辑方程变量的内容 (例如 **y1**) 也不改变 **eqn**。

lowerBound < upperBound 必须为真。

如果一个有效的二元素数组已存入数组变量, 可在 bound= 提示处输入数组变量, 。

如果退出方程求解器, 当再回到方程求解器时, 将显示已存入 eqn 的方程。

可在交互式求解器编辑器中显示其他菜单



要显示窗口编辑器，从求解器菜单选择 **WIND** 。

当从求解器菜单选择 **GRAPH** 或 **WIND** 时，**EDIT** 替代已在菜单上选择的项。要从图象或窗口编辑器返回交互式求解器编辑器，选择 **EDIT** 。

求解未知变量

在存储所有已知变量的值、设置上下限和输入一个估计值（可选）后，将光标移动到未知变量的提示处。

省略号 (...) 代表变量值超出了屏幕。要滚动该值，按 ▸ 和 ▢。

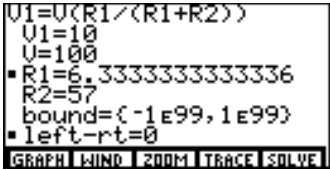
当编辑任何值时方块消失。

求解后，可编辑变量的值或方程，然后求解方程中的相同变量或其他变量。

要求解方程，则从求解器菜单 ([F5]) 选择 **SOLVE** 。

- ◆ 小方块标记所求的变量。结果的值显示。
- ◆ 小方块也标记 **left-rt=** 提示。此提示处的值是方程左边的值减去方程右边的值，以所求变量的新值计算。如果结果精确，将显示 **left-rt=0** 。

某些方程有多个解。要求解其他结果，可输入新的估计值，或者设置新的上下限，然后求解同一个变量。



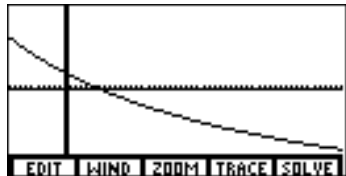
右边的图绘制了第 202 页中示例的结果。窗口变量值为:

xMin=-10 yMin=-50
xMax=50 yMax=50

绘制结果

当从求解器菜单 (**F1**) 选择 **GRAPH** 时, 显示求解器图象, 其中包含自由移动光标。

- ◆ 纵坐标代表在每个自变量值处方程左边减去方程右边的结果 (**left-right**)。
- ◆ 横坐标代表所求方程的自变量。



在图中, 方程 **left-rt=0** 的解存在, 即图象与 **x** 轴的交点。求解器图象:

- ◆ 使用当前窗口和格式设置 (第 5 章)。
- ◆ 根据当前图象方式不绘制结果。
- ◆ 总以函数图象绘制结果。
- ◆ 绘制所选择的函数或已打开的统计图时不绘制结果。

求解器图象工具

正如在其他图象上一样, 可用自由移动光标浏览结果图象。浏览时, 更新变量的坐标值 (**x** 轴) 和 **left-rt** 的值 (**y** 轴)。

要激活跟踪光标, 则从求解器菜单选择 **TRACE**。可用跟踪光标在求解器图象上平移、快速缩放和输入特定的值 (第 6 章)。

要从跟踪返回求解器菜单, 按 **[EXIT]**。

可在图象上使用自由移动光标或跟踪光标选择一个估计值。

第 6 章和“A 到 Z 参考”详细描述了这些特征。

SIMULT 系数不是变量。

可在系数输入屏幕显示其他菜单。

求解器 **ZOOM**（缩放）菜单 **2nd** [SOLVER] 方程 **ENTER** **F3**

GRAPH	WIND	ZOOM	TRACE	SOLVE
BOX	ZIN	ZOUT	ZFACT	ZSTD

- BOX**
- 绘制方框重定义观察窗口（第 6 章）
- ZIN**
- 按因子 **xFact** 和 **yFact** 放大光标周围的图象（第 6 章）
- ZOUT**
- 按因子 **xFact** 和 **yFact** 显示光标周围更多的图象（第 6 章）
- ZFACT**
- 显示 ZOOM FACTORS 屏幕（第 6 章）
- ZSTD**
- 按标准尺寸显示图象，复位 **Func** 图象方式的默认窗口变量值

联立方程求解器 **2nd** [SIMULT]

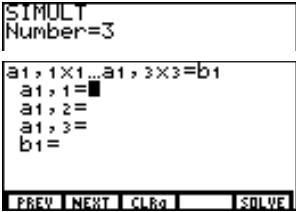
联立方程求解器求解带有 30 个未知变量、由至多 30 个线性方程构成的方程组。

输入方程来同步求解

- 1**
- 显示 SIMULT 方程个数屏幕。
- 2**
- 输入方程的个数，是一个 ≥ 2 且 ≤ 30 的整数。显示第一个方程的系数输入编辑器（对于有 n 个方程和 n 个未知数的方程组）。也显示 SIMULT ENTRY 菜单。

2nd [SIMULT]

3 **ENTER**



选择 **PREV** 或 **NEXT** 从一个方程的系数输入编辑器移到另一个方程的编辑器。

要在系数间移动, 按 $\square$ 、 $\square$ 或 **ENTER**。如果可能的话, 这些键从最后或第一个系数移到下一个或前一个系数输入屏幕。

省略号说明值超出了屏幕。按 $\square$ 和 $\square$ 滚动显示该值。

- ③ 为方程的每个系数和 b_1 (即该方程的解) 输入一个实数或复数值, 或者计算结果是实数或复数值的表达式。

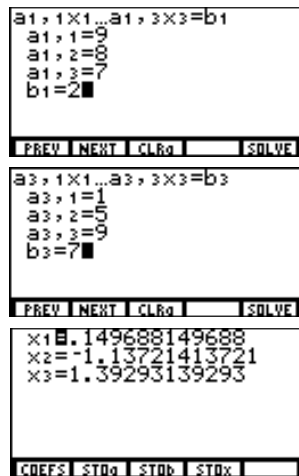
9 $\square$ 8 $\square$ 7 $\square$ 2

- ④ 显示第二个和第三个方程的系数输入屏幕, 并输入它们的值。

$\square$ (或 **ENTER** 或 **F2**) 5 $\square$ (-) 6 $\square$
 (-) 4 $\square$ 2
 $\square$ 1 $\square$ 5 $\square$ 9 $\square$ 7

- ⑤ 求解方程组。多项式的结果被计算, 并显示在结果屏幕上。结果不存储到变量中, 且不能编辑。SIMULT RESULT 菜单显示。

F5



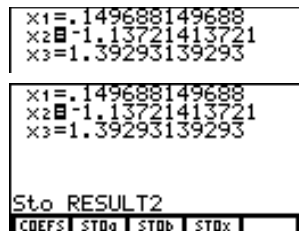
将方程系数和结果存储到变量

- ◆ 选择 **STOa**，存储系数 $a_{1,1}$; $a_{1,2}$; ...; $a_{n,n}$ 到一个 $n \times n$ 维矩阵。
- ◆ 选择 **STOb**，存储解 $b_1, b_2, \dots, b_n$ 到一个 n 维向量。
- ◆ 选择 **STOx**，存储结果 $x_1, x_2, \dots, x_n$ 到一个 n 维向量。

要存储系数输入屏幕或结果屏幕上的单个值，请执行以下步骤。

从 **SIMULT RESULT** 菜单选择 **COEFS**，切换到系数输入屏幕。

- ① 移动光标到要存储的系数或结果旁的等号 。
- ② 变量 **Name=** 提示显示。大写字母锁开启。 
- ③ 输入想保存其值的变量。
 [R][E][S][U][L]
 [T]  2
- ④ 存储值。变量名变成 **VARs REAL** 屏幕或 **VARs CPLX** 屏幕上的一项。 



要在主屏幕或程序中同时求解方程，则选择 **simult**（从 **CATALOG**）。

从 **SIMULT RESULT** 菜单选择 **COEFS**，返回在其中可编辑系数并计算新解的系数输入屏幕。

多项式求根器 $\boxed{2\text{nd}} \boxed{\text{POLY}}$

求根器求解多达 30 次实数或复数多项式。

输入并求解多项式

POLY 系数不是变量。

可在系数输入编辑器上显示其他菜单。

省略号说明值超出了屏幕。按下 $\boxed{\downarrow}$ 和 $\boxed{\uparrow}$ 滚动显示该值。

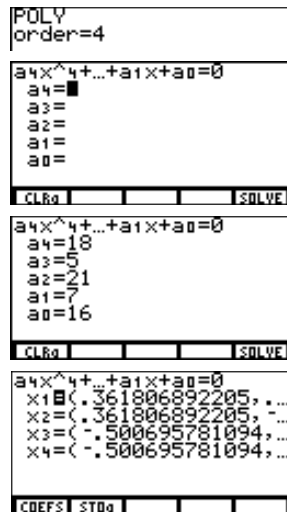
- 1 显示 POLY 命令屏幕。
- 2 输入 2 和 30 间的一个整数。系数输入编辑器的顶部显示方程，系数提示显示在左边，POLY ENTRY 菜单在底部。
- 3 为每个系数输入一个实数或复数值，或者输入计算结果是实数或复数值的表达式。
要清除所有系数，从 POLY ENTRY 菜单选择 CLRa。
- 4 求解方程。计算并显示多项式的根。结果不存入变量，且不能编辑它们。POLY RESULT 菜单也显示。结果可以是复数。

$\boxed{2\text{nd}} \boxed{\text{POLY}}$

4 $\boxed{\text{ENTER}}$

18 $\boxed{\downarrow}$ 5 $\boxed{\downarrow}$ 21
 $\boxed{\downarrow}$ 7 $\boxed{\downarrow}$ 16

$\boxed{\text{F5}}$

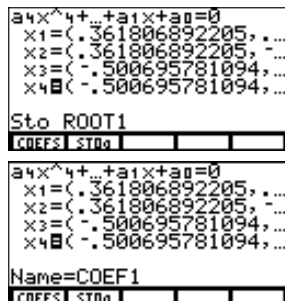


从 POLY RESULT 菜单选择
COEFS 切换到系数输入屏幕。

要在主屏幕上或程序中求根,
从 CATALOG 选择 **poly**。

将多项式的系数或根存储到变量

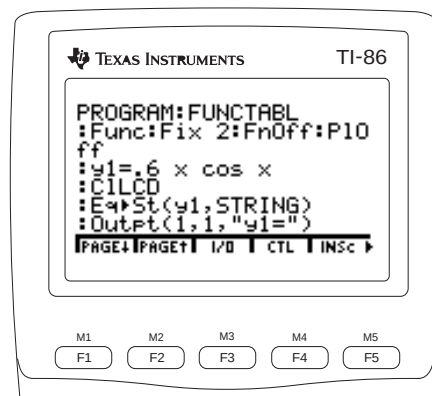
- ① 移动光标到要存储的系数或根旁边的等号 = 处。 ▼ ▼ ▼
- ② 显示 **Sto** 提示。大写字母锁开启。 [STO▶]
- ③ 输入要保存其值的变量。 [R][O][O][T]
- ④ 存储该值。 [ALPHA] 1
- ⑤ 显示系数数组名的 **Name=** 提示, 大写字母锁开启。 [ENTER]
- ⑥ 输入其中要保存系数的数组变量名。 [F2]
- ⑦ 存储多项式系数值。 [C][O][E][F]
- ⑧ 存储多项式系数值。 [ALPHA] 1
- ⑨ 存储多项式系数值。 [ENTER]



从 POLY RESULT 菜单选择 **COEFS**，返回可在其中编辑系数并计算新解的系数输入屏幕。

16 程序设计

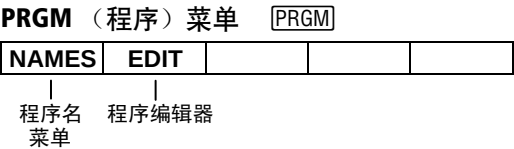
在 TI-86 上编写程序.....	214
运行程序	221
使用程序	223
运行汇编语言程序.....	225
输入并存储字符串.....	226



在 TI-86 上编写程序

程序是输入或下载的、由表达式和指令组成的集合。当运行程序时，程序中的表达式和指令执行。

可在程序中使用 TI-86 的大多数功能。程序能检索并更新存入内存的所有变量。另外，程序编辑菜单有输入/输出命令，如 **Input** 和 **Disp**，和程序控制命令，如 **If**、**Then**、**For** 和 **While**。



TI-86 区分程序名中的大小写字母。例如，**ABC**、**Abc** 和 **abc** 是三个不同的程序名。

在程序编辑器中创建程序

从 PRGM 菜单 (PRGM F2) 选择 **EDIT** 开始编写程序。程序 **Name=** 提示和 PRGM NAMES 菜单显示。大写字母锁开启。输入一个长度为 1 到 8 个字符、以字母开头的程序名。可从 PRGM NAMES 菜单选择程序名来编辑一个已存在的程序。

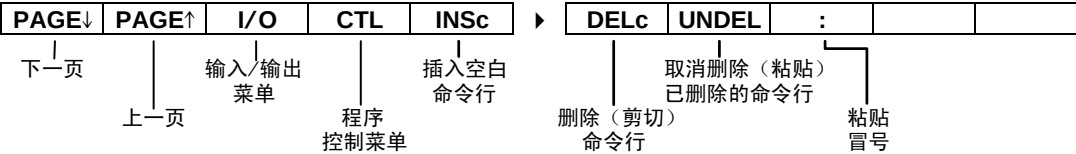


输入程序名后，按 **[ENTER]**。显示程序编辑器和程序编辑器菜单。程序名显示在屏幕顶部。光标位于第一个命令行上，以冒号开始。**TI-86** 自动在每个命令行前放置冒号。

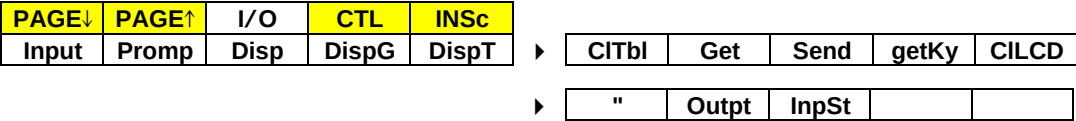


在编写程序时，命令存储到程序名中。

程序编辑器菜单 **[PRGM]** **[F2]** 程序名 **[ENTER]**



PRGM I/O（输入/输出）菜单 **[PRGM]** **[F2]** 程序名 **[ENTER]** **[F3]**



PRGM I/O 菜单项是指令。当程序运行时，它们所执行的操作发生。

要参阅如何在程序中使用 **PRGM I/O** 菜单项的示例，请参阅“**A 到 Z 参考**”。

如果在 **Input** 或 **Prompt** 提示处输入表达式作为变量的值, 则计算并存储表达式。

对于 **Input** 和 **Prompt**, 内置变量如 **y1** 和 **r1** 不可作为变量。

要在 **Disp** 或 **DispG** 后临时暂停程序并检查程序正在显示什么, 则在下一命令行输入 **Pause** (第 219 页)。

Input

Input *variable*

Input *promptString,variable*

Input "string",*variable*

Input "CBLGET",*variable*

Prompt *variableA*

[,*variableB,variableC,...*]

Disp

Disp *valueA,valueB,...*

Disp *variableA,variableB,...*

Disp "textA","textB",...

DispG

DispT

CITbl

Get(*variable*)

Send(*listName*)

getKy

CILCD

"string"

显示当前图象, 并允许使用自由移动光标

暂停程序, 显示 ? 提示, 然后存储输入值到变量 *variable* 中

暂停程序, 显示 *promptString* 或 *string* (最多 21 个字符) 作为提示, 存储输入值到变量 *variable* 中

尽管在 TI-86 上最好使用 **Get**(命令, 但可使用 **Input** 从 CBL、CBR 或 TI-86 (TI-85 兼容) 接收变量 *variable*

以 ? 显示每个变量来提示输入该变量的值

显示主屏幕

显示每个值 *value*

显示保存在每个变量中的值

在当前显示行的左边显示每个文本 *text* 字符串

显示当前图象

显示当前运算表并暂停程序

如果 **Indpnt:Ask** 设置, 则清除当前运算表 (第 7 章)

从 CBL、CBR 或其他 TI-86 获取数据并将它们存入变量 *variable* 中
发送数组名 *listName* 的内容到 CBL 或 CBR

根据键码图 (第 217 页) 返回最后按下键所对应的数; 如果没有按下任何键, 则返回 0

清除主屏幕 (LCD 代表液晶显示器)

指定字符串 *string* 的开始和结束

Output(row,column,"string")	在显示器上从指定的行 <i>row</i> 和列 <i>column</i> 开始显示字符串 <i>string</i> 、字符串名 <i>stringName</i> 或变量 <i>variable</i> 中存储的值 <i>value</i>
Output(row,column,stringName)	
Output(row,column,value)	
Output(row,column,variable)	
Output("CBLSEND",listName)	虽然在 TI-86 上最好使用 Send(, 但可使用 Output(来发送数组名 <i>listName</i> 到 CBL 或 CBR (为了与 TI-85 兼容)
InpSt promptString,variable	暂停程序, 显示 <i>promptString</i> 或 ? , 等待输入; 将输入作为一个字符串存入变量 <i>variable</i> 中; 忽略输入中的引号
InpSt variable	

TI-86 键码图

在程序遇到 **getKey** 时, 根据右边的键码图返回一个与最后按下的键相对应的数。如果没有键被按下, **getKey** 返回 **0**。在循环内部使用 **getKey** 来转移控制, 如创建视频游戏程序时。

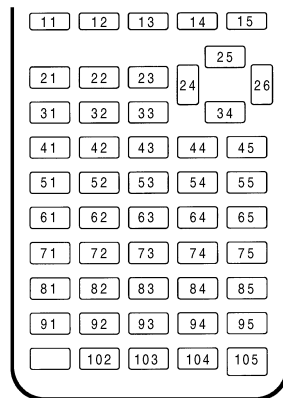
下面的程序返回所按键的键码。

```

:Float
:0→A
:Lb1 TOP
:getKey→A
:If A>0
:Disp A
:Goto TOP

```

要终止 (中断) 这个程序, 按 **[ON]**, 然后按 **[F5]**。



PRGM CTL（程序控制）菜单 PRGM F2 程序名 ENTER F4

PAGE↓	PAGE↑	I/O	CTL	INSc	
If	Then	Else	For	End	▶ While Repea Menu Lbl Goto
					▶ IS> DS< Pause Retur Stop
					▶ DelVa GrStl LCust

要参阅如何在程序中使用 PRGM CTL 菜单项的示例，请参阅“ A 到 Z 参考”。

If、While 和 Repeat 指令可以嵌套。

If <i>condition</i>	如果条件 <i>condition</i> 为假（计算结果为 0），则跳过下一个程序命令；如果条件 <i>condition</i> 为真（计算结果为非零值），程序继续执行下一条命令
Then	跟在 If 后，如果条件 <i>condition</i> 为真则执行一组命令
Else	跟在 If 和 Then 之后，如果条件 <i>condition</i> 为假则执行一组命令
For(循环可被嵌套。 For(<i>variable</i> , <i>begin</i> , <i>end</i> [, <i>step</i>])	变量从起始值 <i>begin</i> 开始，以可选的实数步长 <i>step</i> ，重复一组命令直到变量 <i>variable</i> > <i>end</i> 终值；默认步长 <i>step</i> 为 1
End	标志一组程序命令的结束；For(、While、Repeat 和 Else 组必须以 End 结束；Then 组没有相关的 Else 指令时也必须以 End 结束
While <i>condition</i>	当条件 <i>condition</i> 为真时，重复一组命令；当遇到 While 指令时，测试条件 <i>condition</i> ；通常定义条件 <i>condition</i> 的表达式是一个关系测试（第 3 章）
Repeat <i>condition</i>	重复一组指令直到条件 <i>condition</i> 为真；当遇到 End 指令时测试条件 <i>condition</i>

Menu (<i>item#</i> ," <i>title1</i> ", <i>label1</i> [], <i>item#</i> , " <i>title2</i> ", <i>label2</i> ,...)	当从菜单键 [F1] 到 [F5] 中选择该指令时, 它在程序中设置分支; 当遇到该指令时, 显示第一个菜单组, 菜单组最多 3 个 (最多 15 个标题 <i>titles</i>); 当选择一个标题 <i>title</i> 时, 程序执行分支到标题 <i>title</i> 所代表的标记 <i>label</i> ; <i>item#</i> 是一个 ≥ 1 且 ≤ 15 的整数, 指定标题 <i>title</i> 的菜单位置; 标题 <i>title</i> 是一个文本串, 长度为 1 到 8 个字符 (在菜单中可能被缩写)
Lbl <i>label</i>	为程序命令分配一个标记 <i>label</i> ; <i>label</i> 是由字母开始的 1 到 8 个字符长的串
Goto <i>label</i>	转移控制到用 <i>label</i> 标记的程序分支
IS> (<i>variable</i> , <i>value</i>)	变量 <i>variable</i> 加 1 ; 若结果 $> value$, 则跳过下一命令; 若结果 $\leq value$, 则执行下一条命令; 变量 <i>variable</i> 不能为内置变量
DS< (<i>variable</i> , <i>value</i>)	变量 <i>variable</i> 减 1 ; 若结果 $< value$, 则跳过下一条命令; 若结果 $\geq value$, 则执行下一条命令; 变量 <i>variable</i> 不能为内置变量
Pause	终止程序以便能检查结果, 包括显示的图象或运算表; 要继续程序, 则按 [ENTER]
Pause <i>value</i>	在主屏幕上显示数值 <i>value</i> 以便能滚动显示较大的值, 如数组、向量或矩阵; 要继续执行, 按 [ENTER]
Return	退出子程序 (第 224 页) 并返回调用程序, 即使在嵌套循环内部遇到; 在主程序中, 都将停止程序并返回主屏幕 (每个子程序执行完成后有一隐含的 Return , 退出子程序并返回调用程序)
Stop	停止程序运行并返回主屏幕
DelVar (<i>variable</i>)	从内存删除变量 <i>variable</i> (除程序名外) 和它的内容

GrStl (<i>function#</i> , <i>graphStyle#</i>)	指定函数的图象样式,函数用 <i>function#</i> 表示,图象样式用 <i>graphStyle#</i> 表示; <i>function#</i> 是方程变量的数字部分,如在 y5 中的 5 ; <i>graphStyle#</i> 是一个 ≥ 1 且 ≤ 7 的整数,其中 1 =  (实线)、 2 =  (粗线)、 3 =  (上阴影)、 4 =  (下阴影)、 5 =  (轨迹)、 6 =  (动画) 和 7 =  (点画线)
LCust (<i>item#</i> ," <i>title</i> " [, <i>item#</i> ," <i>title</i> ",...])	调出(定义)TI-86 定制菜单,该菜单在按 CUSTOM 时显示; <i>item#</i> 是一个 ≥ 1 且 ≤ 15 的整数; <i>title</i> 是一个 1 到 8 个字符长的字符串(在菜单中可能被缩写)

输入命令行

比屏幕宽度长的命令行自动在下一行开始处继续显示。

可在命令行上输入任何指令或表达式,只要它们能在主屏幕上执行。在程序编辑器中,每个新的命令行由冒号开始。要在单个命令行上输入多条指令或表达式,用冒号将它们分隔。

按 **ENTER**, 将光标移到下一新的命令行。按  则不能移到下一新的命令行。不过,可通过按  返回到已有的命令行去编辑它们。

程序编辑器中的菜单和屏幕

所有 CATALOG 项在程序编辑器中均有效。

TI-86 菜单和屏幕在程序编辑器中显示时是变化的。对程序无效的菜单项不出现在菜单中。对程序无效的菜单,如 **LINK** 菜单或 **MEM** 菜单,则根本不显示。

当从某个屏幕(如方式或图象格式屏幕)选择设置时,所选设置被粘贴到命令行中光标处。

通常在编辑器中将值保存到一些变量，如窗口变量中，这些变量就成为程序专用菜单上的项，如 GRAPH WIND 菜单。当选择它们时，它们被粘贴到命令行上光标所在位置。

运行程序

- ❶ 粘贴程序名到主屏幕上。或者从 PRGM NAMES 菜单 (**PRGM** **F1**) 选择它，或输入组成程序名的字符。
- ❷ 按 **ENTER**。程序开始运行。

要在暂停后继续执行程序，
则按 **ENTER**。

每个结果都更新上次结果变量 **Ans**（第 1 章）。TI-86 在程序运行时报告错误。程序中执行的命令不更新上次输入存储区域 **ENTRY**（第 1 章）。

下面的示例程序与它在 TI-86 屏幕上显示的一样。程序：

- ◆ 在图象窗口中，通过计算函数、函数的一阶导数和二阶导数在间隔处的值，创建一个表格
- ◆ 以三种不同的图象样式显示函数及其导数的图象，激活跟踪光标，并暂停程序以便能跟踪函数

```

PROGRAM:FUNCTABL
:Func:Fix 2:FnoFF:P10
ff
:y1=.6 x cos x
:CLLCD
:Eq►St(y1,STRING)
:Outpt(1,1,"y1=")
:Outpt(1,4,STRING)
:Outpt(8,1,"PRESS ENT
ER")
:Pause
:CLLCD
:y2=der1(y1,x,x)
:y3=der2(y1,x,x)
:DispT
:GrStl(1,1):GrStl(2,2
):GrStl(3,7)
:2→xRes
:ZTrig
:Trace

```

程序名

设置图象和十进制方式（方式屏幕）；关闭函数（GRAPH VARS 菜单）和统计图（STAT PLOT 菜单）

定义函数（赋值语句）

清除主屏幕（PRGM I/O 菜单）

将 **y1** 转换为字符串变量 **STRING**（STRNG 菜单）

在行 1 和列 1（PRGM I/O 菜单）处显示 **y1=**

在行 1 和列 4（PRGM I/O 菜单）处显示保存到 **STRING** 中的字符串

在行 8 和列 1（PRGM I/O 菜单）处显示 **PRESS ENTER**

暂停程序（PRGM CTL 菜单）

清除主屏幕（PRGM I/O 菜单）

用 **y1** 的一阶导数定义 **y2**（CALC 菜单）

用 **y1** 的二阶导数定义 **y3**（CALC 菜单）

显示运算表（PRGM I/O 菜单）

设置 **y1**、**y2** 和 **y3** 的图象样式（PRGM CTL 菜单）

将 2 保存到窗口变量 **xRes**（GRAPH WIND 菜单）

设置观察窗口变量（GRAPH ZOOM 菜单）

显示图象，激活跟踪光标并暂停（GRAPH 菜单）

中断程序

按 **[ON]** 中断程序。ERROR 06 BREAK 菜单显示。

- ◆ 选择 **GOTO** (**[F1]**)，显示断点所在的程序编辑器。
- ◆ 选择 **QUIT** (**[F5]**)，返回主屏幕。

使用程序

管理内存和删除程序

若要检查是否有足够的内存可用于想输入或下载的程序，则显示检查内存屏幕（[2nd] [MEM] [F1]；第 17 章）。若要增加可用内存，考虑从内存删除所选的项或数据类型（第 17 章）。

编辑程序

当编写完程序后，可在程序编辑器显示它，并编辑任何命令行。

程序编辑器不显示↓用于表示
命令行超出了屏幕。

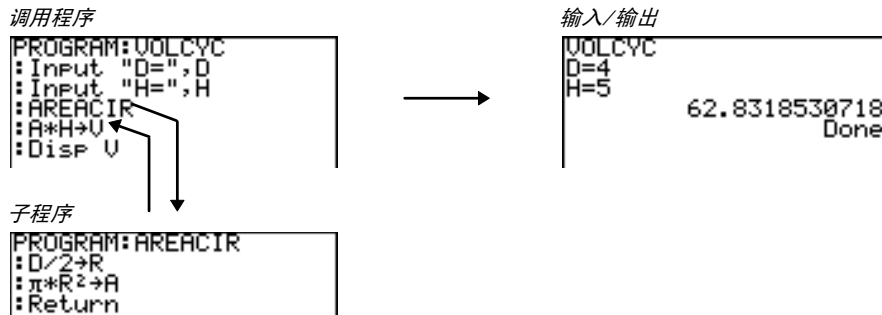
- ❶ 显示程序编辑器 ([PRGM] [F2])。同时显示 PRGM NAMES 菜单。
- ❷ 输入要编辑程序的名字。或从 PRGM NAMES 菜单中选择，或输入组成程序名的字符。
- ❸ 编辑程序命令行。
 - ◆ 移动光标到合适位置，然后删除、改写或插入字符。
 - ◆ 按 [CLEAR] 清除整个命令行，前导冒号除外，然后输入新的程序命令。
 - ◆ 选择程序编辑器菜单项 INSc ([F5]) 和 DELc ([MORE] [F1]) 插入或删除命令行。

一个程序调用另一个程序

在 TI-86 中, 任何保存的程序都能被其他程序当作子程序调用。在程序编辑器中, 在命令行中输入子程序名。

- ◆ 按 **[PRGM]** 显示 PRGM NAMES 菜单, 然后选择程序名。
- ◆ 使用大写字母键和小写字母键输入组成程序名的字符。

如果调用程序在运行时遇到程序名, 则执行的下一条命令是子程序中第一条命令。当在子程序的末尾遇到 **Return** (或隐含 **Return**) 时, 它返回到调用程序的下一条命令。



用于 **Goto** 和 **Lbl** 指令的标记 *label* 只在其所在的程序中有效。在一个程序中的标记 *label* 不能被另一个程序所识别。不能用 **Goto** 跳到其他程序的标记 *label*。

拷贝一个程序到另一个程序

- ❶ 在程序编辑器中显示新的或已存在的程序。
- ❷ 将光标移到要拷贝程序的命令行上。
- ❸ 显示 **Rcl** 提示 (**[2nd]** **[RCL]**)。
- ❹ 输入要拷贝的程序名。或从 **PRGM NAMES** 菜单选择, 或输入组成程序名的字符。
- ❺ 按 **[ENTER]**。被拷贝程序的内容插入到该程序的光标处。

在单个程序中使用或删除变量

若要在程序内使用变量, 而在程序运行结束后不再需要它, 则可在程序中使用 **DelVar** 从内存中删除这个变量。

右边的程序段使用变量 **A** 和 **B** 作为计数器, 然后从内存中删除它们。

```
:3→B
:For (A,1,100,1)
:B+A→B
:End
:Disp A
:Disp B
:DelVar(A)
:DelVar(B)
```

运行汇编语言程序

汇编语言程序是一种程序, 同本章中介绍的一般程序相比, 它的运行速度快, 控制计算器的能力强。可下载并运行 **TI** 创建的汇编语言程序以增加非 **TI-86** 中内置的功能。例如, 可在 **TI-86** 中下载 **TI-83** 的金融或推理统计功能。

TI 的汇编语言程序和其他程序可从 TI 的 World Wide Web 站点:
<http://www.ti.com/calc> 得到。

当下载汇编语言程序时, 它作为 PRGM NAMES 菜单项与其他程序一起保存。可以:

- ◆ 使用 TI-86 通讯链接 (第 18 章) 传送它。
- ◆ 用 MEM DELETE:PRGM 屏幕删除它 (第 17 章)。
- ◆ 从其他程序中当作子程序调用它 (第 224 页)。

运行汇编程序名 *assemblyProgramName* 的语法是: **Asm**(*assemblyProgramName*)

若编写汇编语言程序, 从 CATALOG 使用以下两条指令。

AsmComp(*AsciiAssemblyPrgmName*, 编译用 ASCII 码编写的汇编语言程序并存储为十六进制版本
 HexAssemblyPrgmName)

AsmPrgm 标识汇编语言程序; 必须作为汇编语言程序的第一行输入

输入并存储字符串

字符串是用引号括起的字符序列。

- ◆ 字符串定义在程序中显示的字符。
- ◆ 字符串在程序中接受从键盘的输入。

直接输入字符串的语法为:

"string"

使用  合并 (连接) 两个或更多个字符串的语法为:

"stringA"+"stringB"+"stringC"+...

不要使用引号输入字符串名。
 在合并时, 能用任何字符串名
stringName 代替 "string"。

STRNG（字符串）菜单 2nd [STRNG]

"	sub	lngh	EqSt	StEq
---	-----	------	------	------

" 也标记与数组相连公式的开始和结束；它也是数组编辑菜单里的一项。（第 11 章）

"string"

sub("string",begin,length)
sub(stringName,begin,length)

lngh "string" or lngh stringName

EqSt(equationVariable,stringName)

StEq(stringName,equationVariable)

标记字符串 *string* 的开始和结束

返回 "string" 或字符串名 *stringName* 的子集，从 *begin* 字符所在位置开始，长度为 *length* 个字符

返回 "string" 或字符串名 *stringName* 中的字符个数

将方程变量 *equationVariable* 的内容转换为字符串名 *stringName*

将字符串名 *stringName* 转换为方程变量 *equationVariable*

创建字符串

在主屏幕或程序编辑器中的空白行开始下面的步骤。

- 1

显示 STRNG 菜单。
- 2

依次输入左引号、字符串 **SOLVE & GRAPH** 和右引号。
- 3

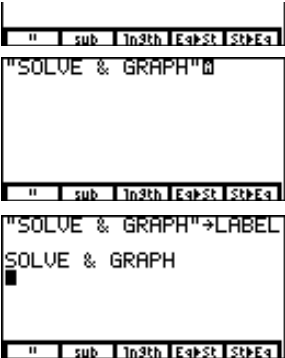
保存字符串到字符串变量名 **LABEL**。

要计算字符串的内容，必须用 **StEq**(把它转换成一个方程。

2nd [STRNG]

[F1] [ALPHA] [ALPHA]
[S][O][L][V][E][L]
2nd [CHAR] [F1] [F3] [L]
[G][R][A][P][H]
2nd [STRNG] [F1]

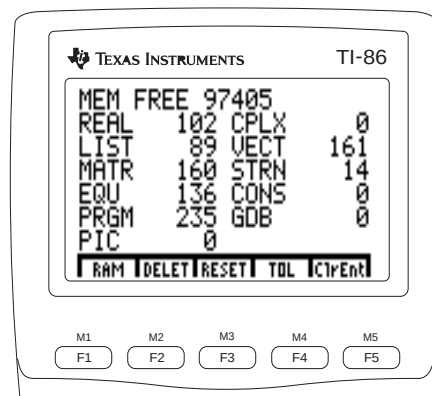
[ALPHA] [STO➡]
[L][A][B][E][L]
[ENTER]



17

内存管理

检查可用内存.....	230
从内存中删除项目.....	231
复位 TI-86	232



检查可用内存

MEM (内存) 菜单 [2nd] [MEM]

请参考附录查找有关 TOL
(误差编辑器) 的信息。

RAM	DELET	RESET	TOL	ClrEnt
检查内存 屏幕	内存删除 菜单	内存/默认值 复位菜单	误差 编辑器	清除 ENTRY 存储区域

检查内存使用情况 [2nd] [MEM] [F1]

当所有内存被清除, 且所有默认值被设置时, 标准 TI-86 的可用随机存取内存 (RAM) 的大小为 **98,224** 字节。当向 RAM 中保存信息时, 可在检查内存屏幕中监视内存分配。

MEM FREE 报告 RAM 中的可用字节总数。相反, 屏幕中所有其他数据则报告每种数据类型当前占有的字节数。例如, 如果在内存中存入一个 50 字节的矩阵, **MATR** 总数将增加 50 字节, 而 **MEM FREE** 总数将减少 50, 变为 **98174** 个字节。

MEM FREE	98224	
REAL	19	CPLX 0
LIST	39	VECT 0
MATR	0	STRN 0
EQU	0	CONS 0
PRGM	18	GDB 0
PIC	0	
RAM DELET RESET TOL ClrEnt		

要显示某特定变量占用的字节数, 则显示这种数据类型的 DELETED 屏幕 (第 231 页)。必要时可滚动屏幕。

从内存中删除项目

MEM DELET (删除) 菜单 [2nd] [MEM] [F2]

ALL	REAL	CPLX	LIST	VECTR	▶	MATRX	STRNG	EQU	CONS	PRGM
					▶	GDB	PIC			

删除 **xt** 部分, 即可删除参数方程。

在本例中, 方程 $y5=x^3-x^2+4x-1$ 被删除。

输入字母可直接移动到以这个字母开头的第一项; 大写字母锁开启。

每个 MEM DELET 菜单项显示对应数据类型的删除屏幕。例如, 当选择 **LIST** 时, MEM DELETE:LIST 屏幕显示。请使用 DELETE 屏幕删除任意用户创建变量以及存储于其中的信息。

- 从 MEM 菜单中选择 **DELET** 显示 MEM DELET 菜单。 [2nd] [MEM] [F2]
- 选择要删除项的数据类型。用 **PAGE↓** 或 **PAGE↑** 每次向下或向上滚动六项。 [MORE] [F3]
- 移动选择光标 (▶) 到要删除的项 (**y5**)。前面是大写字母项, 后面是小写字母项, 它们都以字母顺序排列。 ▼ ▼ ▼
- 删除该项。要删除屏幕上的其他项, 请重复步骤 3 和 4。 [ENTER]

DELETE:EQU		
▶y1	14	EQU
y2	14	EQU
y3	14	EQU
y4	14	EQU
y5	33	EQU
PAGE↓ PAGE↑		

DELETE:EQU		
y1	14	EQU
y2	14	EQU
y3	14	EQU
▶y4	14	EQU
PAGE↓ PAGE↑		

复位 TI-86

在复位所有内存之前, 请考虑删除所选信息以增加内存容量 (第 231 页)。

在选择和确认 ALL 或 DFLTS 后, 对比度被设置为默认值; 可用 [2nd] [Δ] 或 [2nd] [▽] (第 1 章) 进行调节。

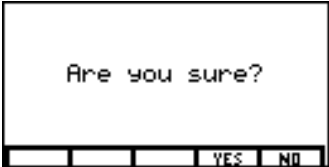
MEM RESET (复位) 菜单 [2nd] [MEM] [F3]

RAM	DELET	RESET	TOL	ClrEnt
ALL	MEM	DFLTS		

- ALL 确认后, 所有数据被清除, 内存复位; 显示这两条信息
- MEM 确认后, 从内存清除所有存储的数据; 显示 **Mem Cleared**
- DFLTS 确认后, 复位所有默认值; 显示 **Defaults Set**

当选择 ALL、MEM 或 DFLTS 时, 显示一个确认菜单。

- ◆ 要确认选定的复位, 则选择 **YES** (按 [F4])。
- ◆ 要取消选定的复位, 则选择 **NO** (按 [F5])。



ClrEnt (清除输入) [2nd] [MEM] [F5]

TI-86 在 ENTRY 中保留尽可能多的以前输入, 多达 128 字节容量。

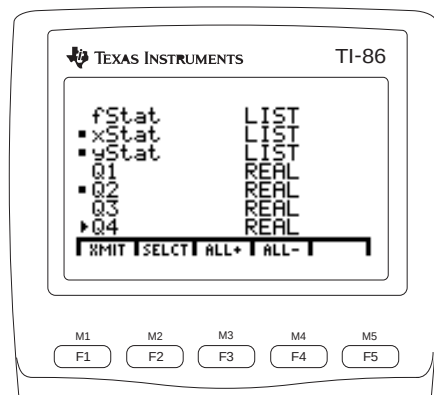
要清除所有输入的 ENTRY 存储区域, 在主屏幕 ([2nd] [MEM] [F5] [ENTER]) 中的空行上执行 **ClrEnt**。



18

TI-86 通讯链接

TI-86 链接选项	234
连接 TI-86 到其他设备	235
选择要发送的数据	236
准备接收设备	240
传送数据	240
接收传送的数据	240



TI-86 链接选项

使用 TI-86 附带的端对端连接电缆，可在 TI-86 和若干其他设备间传送数据。

链接两台 TI-86

可链接两台 TI-86 设备并选择要传送的数据类型，包括程序。可备份一台 TI-86 的整个内存到另一台 TI-86 上。

链接 TI-86 和 TI-85

可选择数据类型（包括程序）从 TI-85 传送到 TI-86 上。使用 **SND85**（第 239 页）可将大多数变量和程序从 TI-86 传送到 TI-85 上，数组、向量或矩阵除外，它们超出了 TI-85 的能力。

在 TI-86 上运行 TI-85 的程序时，TI-85 的 **PrtScrn** 程序指令是无效的。TI-86 的 **EOS** 蕴涵乘法也与 TI-85 的不同（参阅附录）。例如，TI-85 解释 **sin 2x** 为 **sin (2x)**，而 TI-86 解释 **sin 2x** 为 **(sin 2)x**。

链接 TI-86 和 CBL 或 CBR 系统

Calculator-Based Laboratory™ (CBL™) 和 Calculator-Based Ranger™ (CBR™) 系统是可选 TI 附件，可从实际事件，如科学实验中搜集数据。CBL 和 CBR 存储数据到数组，可把数组传送到 TI-86，继而进行分析。可将数组名从 TI-86 传送到 CBL 或 CBR。

链接 TI-86 和 PC 或 Macintosh

TI-86 TI-GRAPH LINK™ 是一个可选附件, 它链接 TI-86 到 IBM® — 兼容或 Macintosh® 计算机。

从 Internet 上下载程序

如果有 TI-GRAPH LINK 和 internet 服务, 则可从 TI 的 World Wide Web 站点下载程序:

<http://www.ti.com/calc>

可从 TI 的 web 站点下载各种程序, 包括增加诸如 TI-83 金融和推理统计等功能的汇编语言程序。该站点还链接到许多其他 TI-86 的 web 站点, 它们由用户组、高等学校、大学和个人维护。

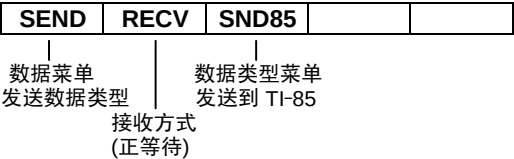
连接 TI-86 到其他设备

在开始向或从 TI-86 传送数据前, 将它连到其他设备。

- ❶ 将端到端连接电缆的一端插入计算器底部边缘的端口上。
- ❷ 将电缆的另一端牢牢插入其他设备 (或 PC 适配器)。

链接菜单在程序编辑器中不可用。

LINK（连接）菜单 [2nd] [LINK]

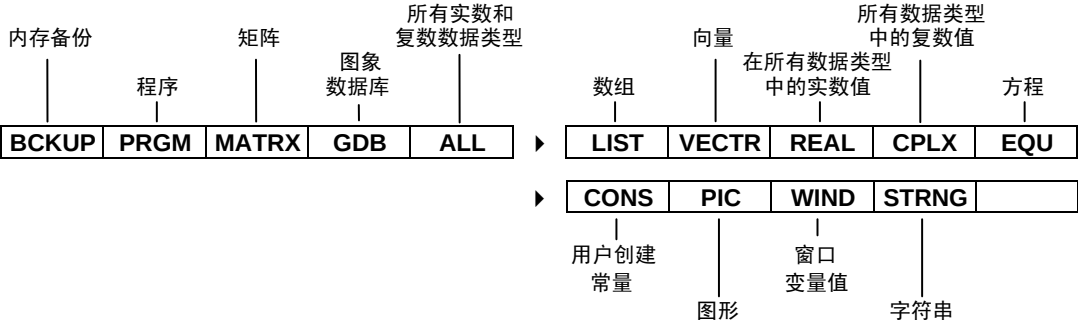


CBL、CBR 和 TI-86 TI-GRAPH LINK 具有内置的无声链接，这使得无须设置设备发送或接收。

选择要发送的数据

要在选择屏幕上列出指定数据类型的变量，则从 LINK SEND 菜单选择这种数据类型。当选择 BCKUP 时，显示 Memory Backup 信息。

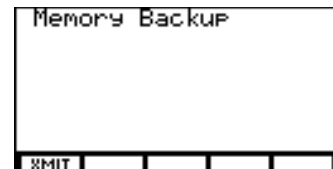
LINK SEND 菜单 [2nd] [LINK] [F1]



初始化内存备份

要初始化内存备份, 则从 LINK SEND 菜单 ([2nd] [LINK] [F1] [F1]) 选择 **BCKUP**。右侧的屏幕显示。

要完成内存备份, 则使其他单元准备接收数据 (第 239 页), 然后从内存备份菜单选择 **XMIT** ([F1])。



警告: 当传送 **BCKUP** 时, 所传送的内存将覆盖接收单元中的所有内存; 接收单元内存中的所有信息丢失。要取消内存备份初始化, 则按 [EXIT]。

作为防止内存意外丢失的安全检查, 当接收计算器接到将进行备份传送的通知时, 它显示警告信息和确认菜单, 如右侧屏幕所示。

- ◆ 要继续备份传送, 则选择 **CONT**。备份传送继续进行, 备份数据替换接收计算器的所有内存。
- ◆ 要取消备份并保留接收计算器的所有内存, 则选择 **EXIT**。



若备份期间出现传送错误,
接收计算器的内存将复位。

若内存中没有存储所选数据类型的数据, 则显示下列信息:
NO VARS OF THIS TYPE。

选择要发送的变量

当选择任何 LINK SEND 菜单项, 除 **BCKUP** 或 **WIND** 外, 所选数据类型的全部变量在选择屏幕上以字母顺序排列。右侧的屏幕是 SEND ALL 屏幕 (**2nd** [**LINK**] **F1** **F5**)。

- ◆ 指定每个变量的数据类型。
- ◆ 小方块表示 **xStat**、**yStat** 和 **Q2** 被选择发送。
- ◆ 选择光标在 **Q4** 旁。

fStat	LIST
▪ xStat	LIST
▪ yStat	LIST
Q1	REAL
▪ Q2	REAL
Q3	REAL
▶ Q4	REAL
XMIT SELECT ALL+ ALL-	

要选择某个变量发送, 则用  和  移动选择光标到该变量旁边, 然后从选择屏幕菜单选择 **SELECT** (**F2**)。

- ◆ 要选择这种类型的所有变量, 则从选择屏幕菜单 (**F3**) 选择 **ALL+**。
- ◆ 要取消选择这种类型的所有变量, 则从选择屏幕菜单 (**F4**) 选择 **ALL-**。

要完成所选变量的传送, 则使其他单元准备接收数据传送 (第 239 页), 然后从选择屏幕菜单 (**F1**) 选择 **XMIT**。

SEND WIND (窗口变量) 屏幕

当从 LINK SEND 菜单 (**2nd** [**LINK**] **F1** **MORE** **MORE** **F3**) 选择 **WIND**, SEND WIND 屏幕显示。SEND WIND 屏幕中的项表示窗口变量、格式设置和 TI-86 图象方式及 **ZRCL** (用户创建缩放) 的所有图象屏幕数据。右侧的屏幕表示已选择 **Func** 和 **DifEq** 图象方式的图象屏幕数据。

Func	WIND
Pol	WIND
Param	WIND
• DifEq	WIND
ZRCL	WIND
XMIT SELECT ALL+ ALL-	

Func 选择发送 **Func** 图象方式的窗口变量值和格式设置

- Pol** 选择发送 **Pol** 图象方式的窗口变量值和格式设置
- Param** 选择发送 **Param** 图象方式的窗口变量值和格式设置
- DifEq** 选择发送 **DifEq** 图象方式的窗口变量值、**difTol**、坐标轴设置和格式设置
- ZRCL** 选择发送用户创建缩放的窗口变量值和任意方式的格式设置

要完成选择变量的传送，则使其他单元接收数据传送（如下所示），然后从内存备份菜单 (**F1**) 选择 **XMIT**。

发送变量到 TI-85

选择变量发送到 **TI-85** 的步骤与选择变量发送到 **TI-86** 的步骤相同。不过， **LINK SND85** 菜单的项比 **LINK SEND** 菜单的少。

TI-86 中数组、向量和矩阵的容量多于 **TI-85**。若发送到 **TI-85** 的数组、向量或矩阵中的元素超过了 **TI-85** 的容量，则超过 **TI-85** 容量的元素被截断。

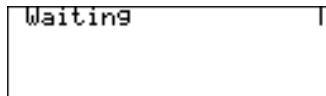
LINK SND85（发送数据到 TI-85）菜单 [2nd] [LINK] [F3]

MATRX	LIST	VECTR	REAL	CPLX	►	CONS	PIC	STRNG		
-------	------	-------	------	------	---	------	-----	-------	--	--

要让 PC 接收数据, 请参考
TI-GGRAPH LINK 指南。

准备接收设备

要让 TI-86 或 TI-85 接收数据传送, 则从 LINK 菜单 (2nd [LINK] F2) 选择 **RECV**。 **Waiting** 信息和忙指示器显示。计算器已准备接收传送项。



要取消接收方式, 而不接收任何项, 则按 [ON]。当显示 **LINK TRANSMISSION ERROR** 信息时, 从菜单 (F1) 选择 **EXIT**。LINK 菜单显示。

传送数据

在发送单元上选择好数据类型, 且接收单元已准备接收数据后, 可以开始数据传送。

要开始传送, 则从发送计算器的选择屏幕菜单 (F1) 中选择 **XMIT**。

要中断传送, 则在任意一台计算器中按 [ON]。当 **LINK TRANSMISSION ERROR** 信息显示时, 从菜单 (F1) 中选择 **EXIT**。LINK 菜单显示。

接收传送的数据

当 TI-86 接收传送的数据时, 逐行显示每个变量的名称和数据类型。若所有选择项已被成功传送, 则 **Done** 信息显示。要滚动传送变量, 则按 ▾ 和 ▴。

在传送期间，如果接收计算器的内存中已有所传送的变量名，则传送过程中断。显示重复的变量名、其数据类型和 DUPLICATE NAME 菜单，如右侧屏幕所示。

要继续或取消传送过程，必须从 DUPLICATE NAME 菜单中选择相应项。



RENAM 显示 **Name=** 提示；输入唯一的变量名；按 **ENTER** 继续传送

OVERW 用被发送变量的数据（覆盖）替换存储在接收单元变量中的数据

SKIP 不覆盖接收单元的数据；尝试发送下一个被选择的变量

EXIT 取消数据传送

重复向几台设备传送数据

在传送过程完成后，LINK 菜单显示，所有选择保持不变。可向不同的 TI-86 传送相同的选择，而无需重选数据。

要向另一台设备传送相同的选择，则需进行如下操作：从接收设备取下端到端连接电缆；把它连到其他设备；使设备准备接收数据；然后依次选择 **SEND**、**ALL** 和 **XMIT**。

错误情形

如果有错误情形发生，一般会很快出现：

- ◆ 电缆没有与发送计算器的端口连接。
- ◆ 电缆没有与接收计算器的端口连接。
- ◆ 接收单元未设置为接收传送。
- ◆ 试图在 TI-86 和 TI-85 间做备份。

接收单元内存不足

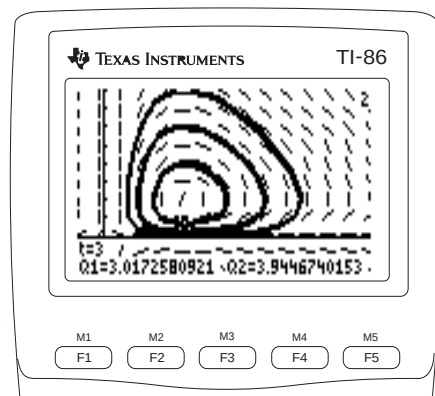
如果接收单元没有足够的内存来接收一个数据项，接收单元显示 **LINK MEMORY FULL**、变量名和数据类型。

- ◆ 要跳过该变量，则选择 **SKIP**。继续传送下一项。
- ◆ 要取消传送，则选择 **EXIT**。

若连接了电缆，但出现了传送错误，将电缆更紧地插入两台计算器并再试一次。

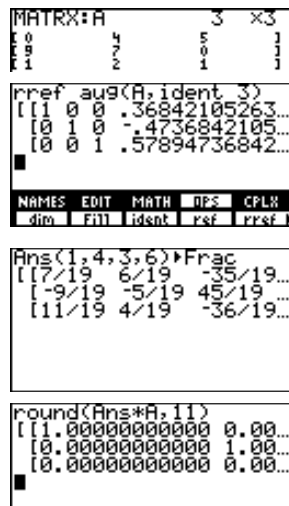
19 应用

矩阵的数学运算.....	244
求曲线间面积.....	245
微积分基本原理.....	246
电路.....	248
程序：泰勒级数.....	250
特征多项式和特征值.....	252
幂级数的收敛性.....	254
蓄水池问题.....	256
食肉动物 — 猎物模型.....	258
程序：Sierpinski 三角形.....	260



矩阵的数学运算

- ① 在矩阵编辑器，输入矩阵 **A** 如图所示。
- ② 在主屏幕上，从 **MATRX OPS** 菜单选择 **rref**。
- ③ 要追加一个 3×3 的单位矩阵到矩阵 **A**，从 **MATRX OPS** 菜单选择 **aug**，输入 **A**，从 **MATRX OPS** 菜单选择 **ident**，然后输入 **3**。执行表达式。
- ④ 输入 **Ans**（步骤 3 得到的矩阵存储在其中）。定义一个包含结果部分的子矩阵。子矩阵从元素 (1,4) 开始，在元素 (3,6) 处结束。
- ⑤ 从 **MATH MISC** 菜单选择 **►Frac**，显示子矩阵的等价分数矩阵。
- ⑥ 检查结果。将小数方式设置为 11（最后的 **1**），为子矩阵的等价分数矩阵与 **A** 的乘积从 **MATH NUM** 菜单中选择 **round**。



显示结果矩阵元素到 11 位小数是为了说明精度。

求曲线间面积

求由右侧函数所围成区域的面积。

$$f(x)=300x/(x^2+625)$$

$$g(x)=3\cos(.1x)$$

$$x=75$$

必要时, 从方程编辑器菜单
选择 **ALL-** 取消选择所有函数。
同时, 关闭所有统计图。

- 1 在 **Func** 图象方式中, 从 **GRAPH** 菜单选择 **y(x)=**, 显示方程编辑器, 输入所示的方程。

$$y1=300x/(x^2+625)$$

$$y2=3\cos(.1x)$$

- 2 从 **GRAPH** 菜单选择 **WIND** 并设置所示的窗口变量。

$$xMin=0$$

$$xMax=100$$

$$xScl=10$$

$$yMin=-5$$

$$yMax=10$$

$$yScl=1$$

$$xRes=1$$

- 3 从 **GRAPH** 菜单选择 **GRAPH** 显示图象屏幕。

- 4 从 **GRAPH MATH** 菜单选择 **ISECT**。移动跟踪光标到函数的交点。按 **[ENTER]** 选择 **y1**。光标移到 **y2**。按 **[ENTER]**。再按 **[ENTER]** 以初始估计值设置当前光标位置。结果使用求解器。交点的 **x** 值, 即积分的下限, 存入 **Ans** 和 **x**。

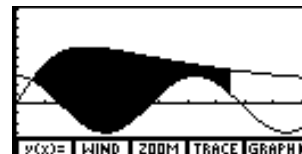
- 5 积分区域在 **y1** 和 **y2** 之间, 从 **x=5.5689088189** 到 **x=75**。要看到图中的区域, 则返回主屏幕, 从 **GRAPH DRAW** 菜单选择 **Shade**, 并执行表达式:

$$\text{Shade}(y2,y1,Ans,75)$$

- 6 从 **MEM** 菜单选择 **TOL** 并设置 **tol=1E-5**。

- 7 在主屏幕, 用 **fnInt** (**CALC** 菜单) 计算积分。面积为 325.839961998。

$$\text{fnInt}(y1-y2,x,Ans,75)$$



必要时, 从方程编辑器菜单
选择 **ALL-** 取消选择所有函数。
同时, 关闭所有统计图。

在本例中, $\text{nDer}(y2,x)$ 近似
等于 $y3$; 不能用 $\text{der1}(y2,x)$
定义 $y3$ 。

微积分基本原理

考虑以下三个函数:

$$F(x)_1 = (\sin x)/x$$

$$F(x)_2 = \int_0^x (\sin t)/t$$

$$F(x)_3 = \frac{d}{dx} \int_0^x (\sin t)/t dt$$

- 在 **Func** 图象方式, 从 **GRAPH** 菜单选择 **y(x)=**, 然后输入函数并在方程编辑器中设置所示的图象样式。(**fnInt** 和 **nDer** 是 **CALC** 菜单项。)

$$y1=(\sin x)/x$$

$$y2=\text{fnInt}(y1(t),t,0,x)$$

$$y3=\text{nDer}(y2,x)$$

- 从 **MEM** 菜单选择 **TOL** 显示误差编辑器。若要提高计算速度, 则设置 **tol=0.1** 和 **δ=0.001**。

- 从 **GRAPH** 菜单选择 **WIND**, 设置所示的窗口变量值:

$$xMin=-10$$

$$xMax=10$$

$$xScl=1$$

$$yMin=-2.5$$

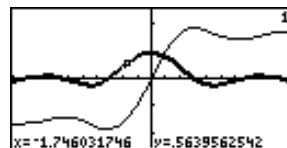
$$yMax=2.5$$

$$yScl=1$$

$$xRes=4$$

- 从 **GRAPH** 菜单选择 **TRACE** 显示图象和跟踪光标。

- 跟踪 **y1** 和 **y3** 以验证 **y1** 的图象和 **y3** 的图象是否从视觉上就可以区分开。



不能从视觉上区分 **y1** 和 **y3** 的图象, 以图象方式证明以下等式:

$$\frac{d}{dx} \int_0^x (\sin t)/t dt = (\sin x)/x$$

- ⑥ 在方程编辑器中取消选择 **y2**。
- ⑦ 从 TABLE 菜单选择 **TBLST**。设置 **TblStart=1**、**ΔTbl=1** 和 **Indpnt:Auto**。
- ⑧ 从 TABLE 菜单选择 **TABLE** 显示试算表。比较 **y1** 的结果和 **y3** 的结果来以数值形式证明上面的公式。

X	Y1	Y3
1	.841471	.8414709
2	.4546487	.4546487
3	.04704	.04704
4	-.189201	-.189201
5	-.191785	-.191785
6	-.046569	-.046569

X=1

TBLST	SELECT	X	Y
-------	--------	---	---

电路

测量设备测量了未知电路中的直流电流 (C) (单位: 毫安) 和电压 (V) (单位: 伏特)。由这些测量值, 可以用方程 $CV=P$ 计算以毫瓦为单位的功率 (P)。被测功率的平均值为多少?

借助于 TI-86, 可以用跟踪光标、内插/外推编辑器和回归预测来估算电流为 125 毫安时的毫瓦功率。

- 1 在数组编辑器的相邻两列, 将下面的电流测量值和电压测量值分别存储在数组名 **CURR**、**VOLT** 中。

{10,20,40,60,80,100,120,140,160}→CURR

{2,4.2,10,18,32.8,56,73.2,98,136}→VOLT

CURR	VOLT	POWER
10	2	20
20	4.2	84
40	10	400
60	18	1080
80	32.8	2624
100	56	5600

- 2 在数组编辑器下一列, 输入数组名 **POWER**。

- 3 在数组编辑器的输入行中输入 **POWER** 的公式 **CURR * VOLT**。
按 **[ENTER]** 计算功率值并将结果存储到数组名 **POWER** 中。

CURR	VOLT	POWER
10	2	20
20	4.2	84
40	10	400
60	18	1080
80	32.8	2624
100	56	5600

- 4 从 GRAPH 菜单选择 **WIND** 并按设置所示的窗口变量值:

xMin=0 xMax=max(POWER) xScl=1000 yMin=0 yMax=max(CURR) yScl=10 xRes=4

- 5 在主屏幕中, 从 CATALOG 选择 **FnOff** 并按 **[ENTER]** 取消选择方程编辑器中的所有函数。从 CATALOG 选择 **Plot1** 并设置统计图, **POWER** 在 x 坐标轴上, **CURR** 在 y 坐标轴上。

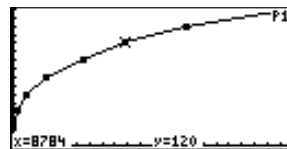
```
FnOff Done
Plot1(2,POWER,CURR,1)
Done
```

圆括号中的 7 和 8 指定
POWER 和 CURR 的第 7 和
第 8 个元素。

要在 LinR 后输入每个回归,
则按 2nd [ENTRY] 并按需要进行编辑。

- ⑥ 从 GRAPH 菜单选择 **TRACE** 来在图象屏幕上显示统计图和跟踪光标。
- ⑦ 跟踪统计图来近似 **POWER** 在 **CURR=125** 处的值。借助此统计数据, 可以跟踪到与 **CURR=125** 最接近的地方是 **CURR=120** (y 坐标轴上)。
- ⑧ 从 MATH 菜单选择 **INTER** 显示内插/外推编辑器。要在 **CURR=125** 处内插 **POWER**, 则输入最接近的数对:

$x1=POWER(7)$ $y1=CURR(7)$
 $x2=POWER(8)$ $y2=CURR(8)$
- ⑨ 输入 $y=125$ 并求解 x 。
- ⑩ 在主屏幕, 从 STAT CALC 菜单选择 **LinR**, 用线性回归模型方程拟合存储在 **POWER** 和 **CURR** 中的数据。记下结果变量 **corr** 的值。
- ⑪ 用对数 (**LnR**)、指数 (**ExpR**) 和幂 (**PwrR**) 回归模型分别拟合这些数据, 并记下每个回归方程的 **corr** 值。比较每个回归方程的 **corr** 值来确定哪个模型拟合数据最精确 (**corr** 值最接近 1)。
- ⑫ 再次执行最精确的回归方程, 然后从 STAT 菜单选择 **FCST**。要预测 **POWER** 在 **CURR=125** 处的值, 则输入 $y=125$ 并求解 x 。
比较此预测结果与步骤 9 中的结果。



```

INTERPOLATE
x1=8784
y1=120
x2=13720
y2=140
x=10018
y=125
SOLVE
  
```

```

LinR POWER,CURR
  
```

```

FORECAST:PwrReg
x=9393.6276510757
y=125
SOLVE
  
```

程序: 泰勒级数

在运行此程序时, 输入一个函数并指定其阶次和中心点。然后程序计算函数的泰勒级数近似值, 并绘制所输入函数的图象。此例说明如何从一个程序中以子程序的形式调用另一个程序。

- 1 在输入程序 **TAYLOR** 之前, 从 **PRGM** 菜单选择 **EDIT**, 在 **Name=** 提示处输入 **MOBIUS**, 然后输入下面的简短程序来存储 **Mobius** 级数。程序 **TAYLOR** 调用此程序并以子程序的形式运行它。

```
PROGRAM:MOBIUS
:{1,-1,-1,0,-1,1,-1,0,0,1,-1,0,-1,1,1,0,-1,0,-1,0}→MSERIES
:Return
```

- 2 从 **PRGM** 菜单选择 **EDIT**, 在 **Name=** 提示处输入 **TAYLOR**, 然后输入该程序计算泰勒级数。

此程序所需的较高次导数值是
基于 “Numerical Differentia-
tion of Analytic Functions,”
J. N. Lyness and C. B. Moler,
SIAM Journal of Numerical
Analysis 4 (1967): 202-210
中的方法计算出来的。

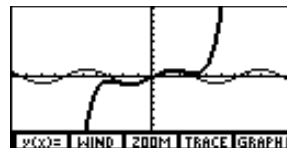
ε 位于 CHAR GREEK 菜单	——	PROGRAM:TAYLOR
		:Func:FnoFF
		:y14=pEval(TPOLY,x-center)
		:GrSt1(14,2)
		:1E-9→ε:.1→rr
		:ClLCD
用户输入方程函数	——	:InpSt "FUNCTION: ",EQ
		:StEq(EQ,y13)
用户输入阶次	——	:Input "ORDER: ",order
		:order+1→dimL TPOLY
		:Fill(0,TPOLY)
用户输入中心点	——	:Input "CENTER: ",center
		:evalF(y13,x,center)→f0
		:f0→TPOLY(order+1)

```

:If order≥1
:der1(y13,x,center)→TPOLY(order)
:If order≥2
:der2(y13,x,center)/2→TPOLY(order-1)
:If order≥3
开始 Then 组 ——— :Then
调用子程序 ——— :MOBIUS
开始 For 组 ——— :For(N,3,order,1)
:abs f0→gmax:gmax→bmi
:1→m:0→ssum
开始 While 组 ——— :While abs bmi≥ε*gmax
创建嵌套 While 组 ——— :While MSERIES(m)==0
: m+1→m
: End
: 0→bsum
创建嵌套 For 组 ——— :For(J,1,m*N,1)
: rr*e^(2π(J/(m*N))*(0,1))+(center,0)→x
: real y13→gval
: bsum+gval→bsum
: max(abs gval,gmax)→gmax
: End
: bsum/(m*N)-f0→bmi
: ssum+MSERIES(m)*bmi→ssum
: m+1→m
结束 While 组 ——— :End
: ssum/(rr^N)→TPOLY(order+1-N)
结束 For 组 ——— :End
结束 Then 组 ——— :End
:ZStd

```

- ③ 在主屏幕中, 从 PRGM NAMES 菜单选择 **TAYLOR**, 然后按 **[ENTER]** 运行该程序。
- ④ 提示时, 请输入: **FUNCTION: sin x**
ORDER: 5
CENTER: 0



特征多项式和特征值

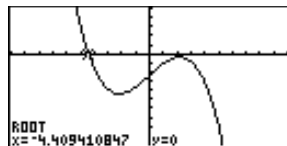
- ① 在矩阵编辑器或主屏幕中, 输入下面的矩阵 **A**。
$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow A$$
- ② 在主屏幕中, 从 MATRX MATH 菜单选择 **eigVl** 求矩阵 **A** 的复数特征值并将结果存储到数组名 **EV**。
- ③ 在不知道 $C_p(x)$ 的解析式是基于公式 $C_p(x) = \det(A - xI)$ 的情况下绘制矩阵 **A** 的特征多项式 $C_p(x)$ 。在 **Func** 图象方式下, 从 **GRAPH** 菜单选择 **y(x)=** 并在方程编辑器中输入下面的函数。
 $y_1 = \det(A - x \cdot \text{ident } 3)$

```
eigVl A→EV
{(-4.40941084667,0) ...}
```

- ④ 从 **GRAPH** 菜单选择 **WIND** 并设置所示的窗口变量值。
xMin=-10 xMax=10 xScl=1 yMin=-100 yMax=50
- ⑤ 从 **GRAPH MATH** 菜单选择 **ROOT** 并用它来交互地显示实数特征值。使用 **Left Bound=-5**、**Right Bound=4** 和 **Guess=4.5**。

比较被交互显示的根 (x 的值) 和步骤 2 中结果数组的第一个元素。

yScl=10 xRes=4



第一个特征值为实数, 因为虚部为 0。

必要时, 从方程编辑器菜单选择 **ALL-** 取消选择所有函数。同时, 关闭所有统计图。

下一步, 使用数组编辑器和三次多项式回归方程来为特征多项式 $y1=\det(A-x*\text{ident } 3)$ 求解关于 x 的解析式。请创建两个能用来求解解析式的数组。

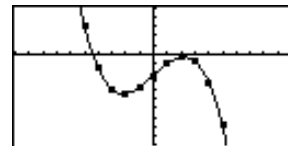
- ⑥ 在数组编辑器, 通过在 **xStat** 输入行输入表达式 **seq(N,N,-10,21)** 创建 **xStat** 的元素。 **seq** 位于 MATH MISC 菜单。

xStat	yStat	fStat
-----	-----	-----
xStat =seq(N,N,-10,21)■		

- ⑦ 通过将公式 "**y1(xStat)**" 连接到输入行中 **yStat** 来创建 **yStat** 的元素。当按 **[ENTER]** 或退出数组编辑器时, 计算该表达式的值。

xStat	yStat	fStat	2
-10 -9 -8 -7	-----	-----	
yStat ="y1(xStat)"■			
←	→	NAME	OPS
EQ	fStat	xStat	yStat

- ⑧ 在主屏幕, 从 CATALOG 选择 **Plot1** 并执行 **Plot1(2,xStat,yStat,1)**, 将 **Plot1** 打开为使用数组 **xStat** 和 **yStat** 的 xy 直线统计图。



- ⑨ 从 GRAPH 菜单选择 **GRAPH** 在图象屏幕上显示 **Plot1** 和 **y1**。

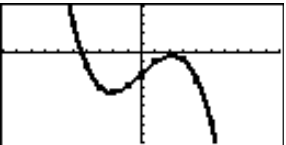
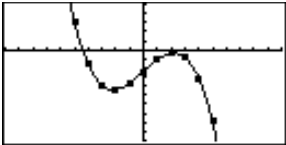
P3Reg xStat,yStat,y2■

- ⑩ 在主屏幕, 从 STAT CALC 菜单选择 **P3Reg**。执行 **P3Reg xStat,yStat,y2** 求关于 x 的显式特征多项式, 并将它存储到 **y2**。存储在结果数组 **PRÉG** 中的三次回归系数建议为: $a=-1$ 、 $b=0$ 、 $c=14$ 和 $d=-24$ 。于是该特征多项式为 $C_p(x)=-x^3+14x-24$ 。

CubicReg
 $y=ax^3+bx^2+cx+d$
 $n=32$
 $PRÉG=$
 $\{-1 -1E-12 14 -23.99...$

要从图象屏幕清除菜单,
则按 **[CLEAR]**。

- ⑪ 通过将 **y1**、**y2** (**Cp(x)** 存储在其中) 和 **Plot1** 绘制在一起来支持此推测。
- ⑫ 在方程编辑器, 输入矩阵 **A** 的显式特征多项式并选择所示的  (粗线) 图象样式。
- y3=-x^3+14x-24**
- ⑬ 绘制 **y1**、**y2**、**y3** 和 **Plot1** 。



- ⑭ 在方程编辑器中取消选择 **y2** 。
- ⑮ 从 **TABLE** 菜单选择 **TABLE** 来在运算表中显示 **y1** 和 **y3** 。
比较特征多项式的值。

x	y1	y3
-4	-32	-32
-2	-78	-78
0	-156	-156
2	-288	-288
4	-424	-424
6	-562	-562

y3 $-x^3+14x-24$

TABLE **SELECT** x y

幂级数的收敛性

$(\sin x)/x$ 的闭合形式的解析反导数不存在。不过, 用 **t** 替换 **x** 后, 可以通过如下步骤来求其无穷级数形式的解析解: 采用 $\sin t$ 的级数定义, 用 **t** 除级数的每一项, 然后逐项合并来生成:

$$\sum_{n=1}^{\infty} -1^{n+1}t^{2n-1}/((2n-1)(2n-1)!)$$

在 TI-86 上用 **sum** 和 **seq** 绘制此幂级数解的有穷近似值。

必要时, 从方程编辑器菜单选择 **ALL-** 取消选择所有函数。同时, 关闭所有统计图。

此示例在 **Param** 方式下设置, 该方式允许用 **tStep** 控制求解并提高绘制速度。

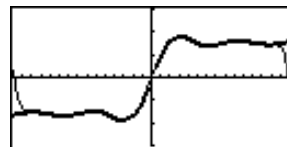
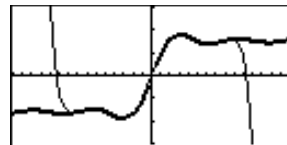
要从图象屏幕清除菜单, 按 **[CLEAR]**。

- ① 从 **MEM** 菜单选择 **TOL** 并设置 **tol=1**。
- ② 在方式屏幕上, 设置 **Radian** 角度方式和 **Param** 图象方式。
- ③ 在方程编辑器, 输入所示的幂级数近似值的参数方程。从 **LIST OPS** 菜单选择 **sum** 和 **seq**。从 **MATH PROB** 菜单选择 **!**。

$$\text{\texttt{xt1=t}} \qquad \texttt{yt1=sum seq((-1)^(j+1)t^(2j-1)/((2j-1)(2j-1)!),j,1,10,1)}$$
- ④ 在方程编辑器, 输入所示的参数方程来绘制 $(\sin x)/x$ 的反导数的图象, 并将之与幂级数近似值的图象比较。(从 **CALC** 菜单选择 **fnInt**。)

$$\text{\texttt{xt2=t}} \qquad \texttt{yt2=fnInt((sin w)/w,w,0,t)}$$
- ⑤ 从 **GRAPH** 菜单选择 **WIND** 并设置所示的窗口变量值。

tMin=-15	xMin=-15	yMin=-3
tMax=15	xMax=15	yMax=3
tStep=0.5	xScl=1	yScl=1
- ⑥ 从 **GRAPH** 菜单选择 **FORMT** 并设置 **SimulG** 格式。
- ⑦ 从 **GRAPH** 菜单选择 **GRAPH** 来在图象屏幕上绘制参数方程的图象。
- ⑧ 在方程编辑器, 通过将 **yt1** 的值从 **10** 改为 **16** 来计算此幂级数的前 16 项。再次绘制此方程。
 在此例中, 窗口变量 **tStep** 控制绘制速度。从 **GRAPH** 菜单选择 **WIND**, 设置 **tStep=1**, 观察在绘制速度和曲线光滑度方面的差别。



蓄水池问题

在 TI-86 上, 可以用参数图象动画来求解问题。

考虑一个高度为 2 米的蓄水池。必须在蓄水池的边上安装一个小阀, 以便从打开的阀喷出的水落在距蓄水池尽可能远的地面。应该将阀安装在什么高度, 才能使阀完全打开时水流的长度最大?

假定在 $\text{time}=0$ 时水箱满, x 方向没有加速度, 且 y 方向没有初速度。而且, 忽略阀大小和阀类型等因素。两次积分 x 方向和 y 方向的加速度定义, 则得到方程 $x=v_0t$ 和 $y=h_0-(gt^2)/2$ 。求 v_0 的泊努利方程并将结果代入 v_0t 中, 则得到下面两个参数方程:

$$xt=t\sqrt{2g(2-h_0)} \quad yt=h_0-(gt^2)/2$$

t = 时间 (秒)

h_0 = 阀高度 (米)

g = 重力常数的固定加速度

在 TI-86 上绘制这些方程的图象时, y 轴 ($x=0$) 是要安装阀的蓄水池侧面, x 轴 ($y=0$) 为地面。每个被绘制的参数方程表示阀在几个高度之一时的水流。

必要时, 从方程编辑器菜单
选择 **ALL-** 取消选择所有函数。
同时, 关闭所有统计图。

- 1 在 **Param** 图象方式下, 从 **GRAPH** 菜单选择 **E(t)=** 并在方程编辑器中输入所示的方程。下面这对方程绘制阀安装高度为 0.5 时水流的轨迹。

$$x1=t\sqrt{2g(2-0.5)}$$

$$y1=0.5-(g*t^2)/2$$

- 2 将光标移到 **xt2=**。按 **[2nd] [RCL] [F2] [ALPHA] 1**, 然后按 **[ENTER]** 以将 **xt1** 的内容重新调入到 **xt2**。对于 **xt2**, 将阀高度 (为 0.5) 改变为 0.75 米。对 **yt1** 和 **yt2** 执行相同的操作。
- 3 重复步骤 3 再创建三对方程。对于 **xt3** 和 **yt3**, 阀高度改为 1.0 米; **xt4** 和 **yt4**, 阀高度改为 1.5 米; **xt5** 和 **yt5**, 阀高度改为 1.75 米。
- 4 从 **GRAPH** 菜单选择 **WIND** 并设置所示的窗口变量值。

$$tMin=0$$

$$xMin=0$$

$$yMin=0$$

$$tMax=\sqrt{4/g}$$

$$xMax=2$$

$$yMax=2$$

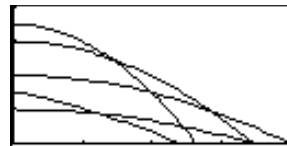
$$tStep=0.01$$

$$xScl=0.5$$

$$yScl=0.5$$

- 5 从 **GRAPH** 菜单选择 **FORMT** 并设置 **SimulG** 图象格式。
- 6 从 **GRAPH** 菜单选择 **GRAPH** 绘制水从五个给定高度喷射时的轨道。

哪个高度水流飞得最远?



要从图象屏幕清除菜单,
按 **[CLEAR]**。

食肉动物 — 猎物模型

食肉动物和猎物（比如狐狸和兔子）总数的增长率依赖于两个物种的总数。初始值问题是食肉动物 — 猎物模型的一种形式。

$$F' = -F + 0.1F * R \quad R' = 3R - F * R$$

Q1 = 狐狸总数 (F)

Q2 = 兔子总数 (R)

QI1 = 狐狸初始总数 (2)

QI2 = 兔子初始总数 (5)

求 3 个月 ($t=3$) 后狐狸和兔子的总数。

- 在 **DifEq** 图象方式下，从 **GRAPH** 菜单选择 **Q't=**，输入函数，并在方程编辑器中设置所示的图象样式。

$$Q'1 = -Q1 + 0.1Q1 * Q2 \quad Q'2 = 3Q2 - Q1 * Q2$$

- 从 **GRAPH** 菜单选择 **FORMT**，设置 **FldOff** 域格式。
- 从 **GRAPH** 菜单选择 **WIND**，设置所示的窗口变量值。

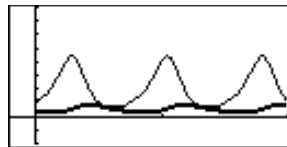
tMin=0	xMin=-1	yMin=-10
tMax=10	xMax=10	yMax=40
tStep=$\pi/24$	xScl=5	yScl=5
tPlot=0		difTol=.001

- 从 **GRAPH** 菜单选择 **INITC**，设置所示的初始条件。

tMin=0	QI1=2	QI2=5
---------------	--------------	--------------

- ⑤ 从 GRAPH 菜单选择 **GRAPH** 绘制两种总数关于时间的图象。
- ⑥ 要看到相位平面解的方向域, 则从 GRAPH 屏幕选择 **FORMT**, 然后设置 **DirFld** 域格式。
- ⑦ 从 GRAPH 菜单选择 **INITC**, 删除 **QI1** 和 **QI2** 的值。
- ⑧ 从 GRAPH 菜单选择 **GRAPH** 来显示相位平面解的方向域。
- ⑨ 要看到方向域顶部的一族特定相位平面解, 则从 GRAPH 菜单选择 **INITC**, 然后为 **QI1** 和 **QI2** 输入所示的数组。
 $QI1=\{2,6,7\}$ $QI2=\{6,12,18\}$
- ⑩ 从 GRAPH 菜单选择 **TRACE** 显示跟踪光标所在的图象。
- ⑪ 按 **3** 看在 $t=3$ 时有多少狐狸和兔子存活。(**Q1** (狐狸) 和 **Q2** (兔子) 的值舍入为整数。) 在 $t=6$ 和 $t=12$ 时有多少狐狸和兔子存活呢?

相位平面的轨道收敛时 **Q1** 和 **Q2** 的值是多少呢? 该值的意义如何?



程序: Sierpinski 三角形

此程序创建著名不规则碎片图, 即 Sierpinski 三角形, 并将之存储到图形变量 **TRI**。

- ❶ 从 PRGM 菜单选择 **EDIT**, 在 **Name=** 提示处输入 **SIERP**, 然后输入此程序。

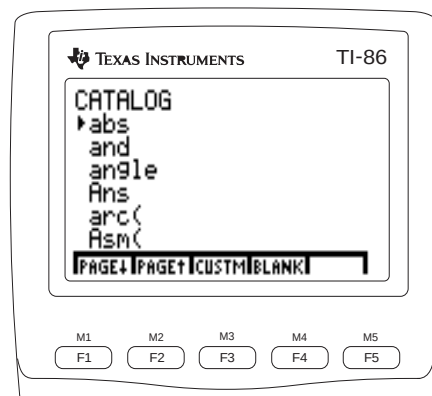
	PROGRAM:SIERP	
	:FnOff :ClDrw	
	:PlOff	
	:AxesOff	
设置观察窗口	┌:0→xMin:1→xMax	
	└:0→yMin:1→yMax	
	:rand→X:rand→Y	
开始	——:For(K,1,3000)	
For 组	:rand→N	
	┌:If N≤(1/3)	
If/Then 组	└:Then	
	└: .5X→X	
	└: .5Y→Y	
	└:End	
	└:If N>(1/3) and N≤(2/3)	If/Then 组
	└: Then	
	└: .5(.5+X)→X	
	└: .5(1+Y)→Y	
	└: End	
	└:If N>(2/3)	If/Then 组
	└: Then	
	└: .5(1+X)→X	
	└: .5Y→Y	
	└: End	
	——:PtOn(X,Y)	绘制点
	——:End	For 结束
	——:StPic TRI	存储图形

- ❷ 在主屏幕, 从 PRGM NAMES 菜单选择 **SIERP**, 按 **[ENTER]** 运行该程序, 它可能需要运行几分钟。
- ❸ 运行程序之后, 可以通过执行 **RcPic TRI** 来调用并显示该图形。



20 A 到 Z 函数 和指令参考

快速查找定位器	262
运算符按字母顺序排列	266



快速查找定位器

本节以功能组形式列出了 TI-86 函数和指令以及它们在本章中的描述所在的页号。

图形

Axes(..... 271	Drlnv 287	Line(..... 314	RectGC 344	ZFit 373
AxesOff 271	dxDer1 288	Param 333	SeqG 351	ZIn 373
AxesOn 271	dxNDer 288	Pol 336	Shade(..... 352	ZInt 374
Circl(..... 273	FldOff 295	PolarGC 336	SimulG 354	ZOut 375
CIDrw 273	FnOff 296	PtChg(..... 338	SlpFld 358	ZPrev 375
CoordOff 275	FnOn 297	PtOff(..... 338	StGDB 361	ZRcl 376
CoordOn 275	Func 299	PtOn(..... 338	StPic 362	ZSqr 376
DifEq 281	GridOff 301	PxChg(..... 340	TanLn(..... 366	ZStd 377
DirFld 282	GridOn 302	PxOff(..... 340	Text(..... 366	ZTrig 378
DrawDot 285	GrStl(..... 302	PxOn(..... 340	Trace 367	
DrawF 286	Horiz 304	PxTest(..... 340	Vert 369	
DrawLine 286	LabelOff 310	RcGDB 343	ZData 371	
DrEq(..... 287	LabelOn 310	RcPic 343	ZDecm 372	

数组

aug(..... 270	→dimL 282	li→vc 316	SetLEdit 351	Sorty 359
cSum(..... 278	Fill(..... 295	prod 338	sortA 359	sum 364
Deltalst(..... 279	Form(..... 298	Select(..... 350	sortD 359	vc→li 369
dimL 282	数组输入: { } 316	seq(..... 351	Sortx 359	

数学、代数和微积分

abs 267	除: / 284	大于或等于: $\geq$... 301	nPr 326	shftL 353
加: + 267	DMS 输入: ' 285	h 302	o 326	shftR 353
and 268	►DMS 285	Hex 302	Oct 327	sign 354
angle 269	dxDer1 288	►Hex 303	►Oct 327	simult(..... 354
Ans 269	dxNDer 288	imag 306	or 328	sin 355
arc(..... 269	e^ 288	int 308	百分数: % 334	sin ⁻¹ 355
赋值: = 270	Eng 290	inter(..... 309	pEval(..... 334	sinh 356
b 271	Eq►St(..... 290	求逆: ⁻¹ 309	►Pol 336	sinh ⁻¹ 356
Bin 272	相等: = 290	iPart 309	PolarC 336	Solver(..... 358
►Bin 272	等于: == 291	lcm(..... 311	极坐标复数: $\angle$... 336	平方: ² 360
ClrEnt 273	Euler 291	小于: < 312	poly 337	平方根: $\sqrt{\quad}$ 360
CITbl 273	eval 291	小于或等于: $\leq$... 312	幂: ^ 337	St►Eq(..... 361
conj 275	evalF(..... 292	ln 316	10 的幂: 10^ 337	存储到
cos 276	指数: E 292	log 318	Radian 341	变量 ► 362
cos ⁻¹ 276	阶乘: ! 294	max(..... 319	弧度输入 $^{\circ}$ 341	减: - 363
cosh 277	Fix 295	min(..... 320	real 343	tan 364
cosh ⁻¹ 277	Float 295	mod(..... 320	►Rec 343	tan ⁻¹ 365
d 278	fMax(..... 296	乘: * 321	RectC 344	tanh 365
Dec 278	fMin(..... 296	nCr 322	RK 345	tanh ⁻¹ 365
►Dec 279	fnInt(..... 296	nDer(..... 323	根: $\sqrt[x]{\quad}$ 346	xor 370
Degree 279	fPart 298	负号: - 323	rotL 347	
度输入: $^{\circ}$ 279	►Frac 298	Normal 324	rotR 347	
der1(..... 280	gcd(..... 299	not 325	round(..... 348	
der2(..... 280	大于: > 300	不等于: $\neq$ 326	Sci 349	

 矩阵

aug(.....270	→dim281	LU(.....318	rAdd(.....340	rSwap(.....348
cnorm.....273	eigVc.....289	矩阵输入: [].....319	randM(.....342	转置: T367
cond.....274	eigVl.....289	mRAdd(.....321	ref.....344	
det.....281	Fill(.....295	multR(.....322	rnorm.....346	
dim.....281	ident.....304	norm.....323	rref.....348	

 编程

Asm(.....269	DispT284	Get(.....299	Input.....307	Prompt338
AsmComp(.....270	DS<(.....288	getKy300	IS>(.....310	Repeat.....345
AsmPrgm.....270	Else.....290	Goto300	Lbl.....311	Return345
CILCD.....273	End290	lAsk304	LCust(.....311	Send(.....350
DelVar(.....280	相等: =290	lAuto304	Menu(.....320	Stop362
Disp.....283	等于: ==291	If305	Outpt(.....329	Then.....366
DispG283	For(.....297	lnpSt.....307	Pause333	While369

 统计

Box272	LnR317	PIOn334	randInt(.....342	SinR357
ExpR293	MBox.....319	Plot1(.....335	randM(.....342	Sortx359
fcstx294	OneVar327	Plot2(.....335	randNorm(.....342	Sorty359
fcsty294	P2Reg330	Plot3(.....335	Scatter349	StReg(.....362
Hist303	P3Reg331	PwrR.....339	Select(.....350	TwoVar368
LgstR313	P4Reg332	rand341	SetLEdit.....351	xyline370
LinR315	PIOff.....334	randBin(.....341	ShwSt.....354	

字符串

合并: +274	lngh 316	StEq(..... 361	字符串输入: ".....363	sub(.....363
EqSt(.....290				

向量

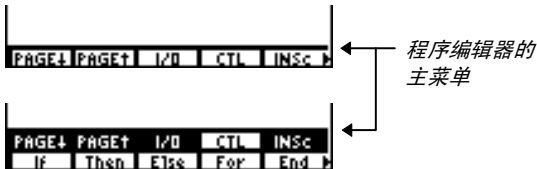
cnorm.....273	dim 281	livc 316	Sph360	向量输入: [].....369
cross(.....277	→dim 281	norm 323	SphereV360	
►Cyl278	dot(..... 285	RectV 344	unitV368	
CylV278	Fill(..... 295	rnorm..... 346	vc►li369	

运算符按字母顺序排列

本节中的所有运算符都包含在 CATALOG 中。非字母的运算符（例如 +, ! 和 >）列在 CATALOG 的末尾。不过，在 A 到 Z 参考中，这些运算符是根据它们的等价字母顺序进行排列的（例如加法，阶乘和大于）。

通常可以使用 CATALOG 选择一个运算符，将它粘贴到主屏幕上或程序编辑器的命令行中。也可使用本节列出的特定击键、菜单或屏幕。

- † 表示只有在程序编辑器中才能粘贴运算符名的菜单和屏幕。在大多数情况下，可以从主屏幕使用这些菜单和屏幕来执行交互操作，而不需要粘贴名字。
- ‡ 表示只有在程序编辑器的主菜单中才有效的菜单和屏幕。在主屏幕上无法使用这些菜单和屏幕选择运算符。



某些运算符的语法使用括号 [] 表示可选的参数。如果使用可选的参数，则不要输入方括号。

abs

MATH NUM 菜单
CPLX 菜单
MATRX CPLX 菜单
VECTR CPLX 菜单

abs <i>realNumber</i> 或 abs (<i>realExpression</i>)	abs -256.4 [ENTER]	256.4
返回实数 <i>realNumber</i> 或实数表达式 <i>realExpression</i> 的绝对值。	abs -4*3+13 [ENTER]	25
	abs (-4*3+13) [ENTER]	1
abs (<i>complexNumber</i>)	abs (3,4) [ENTER]	5
返回复数 <i>complexNumber</i> 的模。	abs (3∠4) [ENTER]	3
abs (<i>real,imaginary</i>) 返回 $\sqrt{(real^2+imaginary^2)}$ 。		
abs (<i>magnitude∠angle</i>) 返回 <i>magnitude</i> 。		
abs <i>list</i>	abs {1.25,-5.67} [ENTER]	{1.25 5.67}
abs <i>matrix</i>		
abs <i>vector</i>	abs [(3,4),(3∠4)] [ENTER]	[5 3]
返回一个数组、矩阵或向量，其中每个元素是参数中相应实数或复数的绝对值或模。		

加: +



numberA + numberB	在 RectC 复数方式下:	
返回两个实数或复数的和。	(2,5)+(5,9) [ENTER]	(7,14)
number+list	4+{1,2,3} [ENTER]	{5 6 7}
返回一个数组，其中每个元素值是实数或复数 <i>number</i> 与数组 <i>list</i> 中的每个实数或复数元素相加之和。	3+{1,7,(2,1)} [ENTER]	{(4,0) (10,0) (5,1)}

listA+listB
matrixA+matrixB
vectorA+vectorB

返回一个数组、矩阵或向量，其值为参数中相应的实数或复数元素之和。两个参数的维数必须相同。

有关两个字符串相加的信息，请参阅 274 页的 **Concatenation**。

```
{1,2,3}+{4,5,6} [ENTER] {5 7 9}

[[1,2,3][4,5,6]]+[[4,5,6][7,8,9]]
[ENTER] [[5 7 9 ]
[11 13 15]]

[1,2,3]+[4,5,6] [ENTER] [5 7 9]
```

and

BASE BOOL 菜单

integerA and *integerB*

按位比较两个实整数。两个整数在内部转换为二进制。当相应的位进行比较时，每位均为 1 时，结果为 1；否则结果为 0。返回值为位比较结果的和。

例如, $78 \text{ and } 23 = 6$

$$78 = 1001110\mathbf{b}$$
$$23 = 0010111\mathbf{b}$$
$$\underline{\underline{0000110b}} = 6$$

可以输入实数而不是整数，但是在比较前已自动被截断。

在 **Dec** 数制方式下:

78 and 23 6

在 **Bin** 数制方式下:

1001110 and 10111

Ans►Dec

110b
6d

Ans►Dec ENTER

angle

CPLX 菜单

MATRX CPLX 菜单

VECTR CPLX 菜单

angle (*complexNumber*)

返回复数 *complexNumber* 的极坐标角度, 在第二象限和第三象限分别调整 $+\pi$ 、 $-\pi$ 。实数的极坐标角度总是 0。

angle (*real*,*imaginary*) 返回 $\tan^{-1}(\text{imaginary}/\text{real})$ 。

angle (*magnitude* $\angle$ *angle*) 返回 *angle*, $-\pi < \text{angle} \leq \pi$ 。

angle *complexList*

angle *complexMatrix*

angle *complexVector*

返回一个数组、矩阵或向量, 其中每个元素是参数中相应元素的极坐标角度。

如果复数向量 *complexVector* 只有两个实数元素, 返回值为一个实数, 而不是向量。

在 **Radian** 角度方式下和 **PolarC** 复数方式下:

angle (3,4) .927295218002

angle (3 $\angle$ 2) 2

(6 $\angle$ π /3) $\rightarrow$ A (6 $\angle$ 1.0471975512)

angle A 1.0471975512

angle {(3,4),(3 $\angle$ 2)}
{.927295218002 2}

Ans

Ans

返回上次计算结果。

1.7 $\times$ 4.2 7.14

147/Ans 20.5882352941

arc(

CALC 菜单

arc (*expression*,*variable*,*start*,*end*)

返回表达式 *expression* 关于变量 *variable* 的长度, 范围为 *variable* = *start* 到 *variable* = *end*。

arc(x²,x,0,1)

1.47894285752

arc(cos x,x,0, π)

3.82019778904

Asm(

CATALOG

Asm(*assemblyProgramName*)

执行汇编语言程序。详细信息请参阅第 16 章。

AsmComp(

CATALOG

AsmComp(*AsciiAssemblyPrgmName*,*HexAssemblyPrgmName*)

编译用 ASCII 格式编写的汇编语言程序，并保存十六进制版本。编译后的十六进制版本，只使用 ASCII 版本的一半存储空间，但不能进行编辑。

当执行 ASCII 版本程序时，TI-86 每次都进行编译。为加快执行速度，使用 **AsmComp**(对 ASCII 版本程序进行一次编译，然后每次运行程序时执行十六进制版本的程序。

AsmPrgm

CATALOG

AsmPrgm

必须用作汇编程序语言的第一行。

赋值: =

ALPHA [=]

equationVariable = *expression*

将表达式 *expression* 存储到方程变量 *equationVariable*，而不计算表达式 *expression*。（如果使用 STO▸ 将表达式存储到一个变量中，则先计算表达式，然后将结果保存到变量中。）

y1=2 x²+6 x-5 ENTER Done

用于图象的内置方程变量是区分大小写的。使用 **y1**，而不是 **Y1**。

aug(

LIST OPS 菜单

MATRX OPS 菜单

aug(*listA*,*listB*)

返回一个数组，该数组是将数组 *listB* 附加到（合并）数组 *listA* 的尾部后得到的。数组可以是实数数组，也可以是复数数组。

aug({1,-3,2},{5,4}) ENTER
{1 -3 2 5 4}

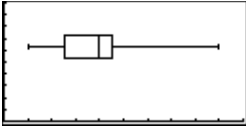
aug(<i>matrixA</i>,<i>matrixB</i>)	<code>[[1,2,3][4,5,6]]>MATA</code> <code>[ENTER]</code>	<code>[[1 2 3]</code> <code>[4 5 6]]</code>
返回一个矩阵，该矩阵是将矩阵 <i>matrixB</i> 作为新列附加到（合并）矩阵 <i>matrixA</i> 的尾部后得到的。矩阵可以是实数矩阵，也可以是复数矩阵。两个矩阵的行数必须相同。	<code>[[7,8][9,10]]>MATB</code> <code>[ENTER]</code>	<code>[[7 8]</code> <code>[9 10]]</code>
aug(<i>matrix</i>,<i>vector</i>)	<code>aug(MATA,MATB)</code> <code>[ENTER]</code>	<code>[[1 2 3 7 8]</code> <code>[4 5 6 9 10]]</code>
返回一个矩阵，该矩阵是将向量 <i>vector</i> 作为新列附加到（合并）矩阵 <i>matrix</i> 的尾部后得到的。参数可以是实数或复数。矩阵 <i>matrix</i> 中的行数必须与向量 <i>vector</i> 中的元素个数相同。		

Axes(	Axes(<i>xAxisVariable</i>,<i>yAxisVariable</i>)	<code>Axes(Q1,Q2)</code> <code>[ENTER]</code>	Done
† GRAPH VARS 菜单	指定在 DifEq 图象方式下坐标轴所对应的变量。 <i>xAxisVariable</i> 或 <i>yAxisVariable</i> 可以为 t 、 Q1 到 Q9 或 Q'1 到 Q'9 。		

AxesOff	AxesOff	
† 图象格式屏幕	关闭图象坐标轴。	

AxesOn	AxesOn	
† 图象格式屏幕	打开图象坐标轴。	

b	<i>integerb</i>	在 Dec 数制方式下：	
BASE TYPE 菜单	指定实整数 <i>integer</i> 为二进制数，不管是何种数制设置方式。	<code>10b</code> <code>[ENTER]</code> <code>10b+10</code> <code>[ENTER]</code>	2 12

Bin † 方式屏幕	Bin 设置二进制数制方式。结果显示带有 b 后缀。无论在何种数制方式下，都可以从 BASE TYPE 菜单用 b 、 d 、 h 或 o 指示符将相应的数值指定为二进制、十进制、十六进制或八进制。	在 Bin 数制方式下： 10+ Fh +10 o +10 d [ENTER] 100011 b
►Bin BASE CONV 菜单	<i>number</i> ► Bin <i>list</i> ► Bin <i>matrix</i> ► Bin <i>vector</i> ► Bin 返回实数或复数参数的二进制等价值。	在 Dec 数制方式下： 2*8 [ENTER] 16 Ans► Bin [ENTER] 10000 b {1,2,3,4}► Bin [ENTER] {1 b 10 b 11 b 100 b }
Box † STAT DRAW 菜单	Box <i>xList</i> , <i>frequencyList</i> 使用 <i>xList</i> 中的实数数据和 <i>frequencyList</i> 中的频数在当前图象上绘制一个箱线图。 Box <i>xList</i> 使用频数 1。 Box 使用内置变量 xStat 和 fStat 中的数据。这些变量必须包含相同维数的有效数据，否则会产生错误。	从 ZStd 图象屏幕开始： {1,2,3,4,5,9}► XL [ENTER] {1 2 3 4 5 9} {1,1,1,4,1,1}► FL [ENTER] {1 1 1 4 1 1} 0► xMin :0► yMin [ENTER] 0 Box XL , FL [ENTER] 

Circl(

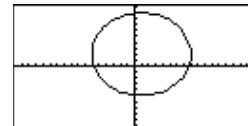
† GRAPH DRAW 菜单

Circl($x,y,radius$)

以为 (x,y) 圆心, $radius$ 为半径, 在当前图象上绘制圆。

从 **ZStd** 图象屏幕开始:

ZSqr:Circl(1,2,7) **[ENTER]**



CIDrw

† GRAPH DRAW 菜单

† STAT DRAW 菜单

CIDrw

从当前图象上删除所有绘制的元素。

CILCD

‡ 程序编辑器
I/O 菜单

CILCD

清除主屏幕 (LCD)。

ClrEnt

MEM 菜单

ClrEnt

清除上次输入存储区域的内容。

CITbl

‡ 程序编辑器
I/O 菜单

CITbl

如果设置了 **Indpnt: Ask** (**IAsk**, 第 304 页), 从当前运算表中清除所有的数值。

cnorm

MATRX MATH 菜单

cnorm *matrix*

返回实数或复数矩阵 *matrix* 的列范数。对于每一列, **cnorm** 将该列所有元素的绝对值 (复数元素的模) 相加, 然后返回这些列和的最大值。

[[1,-2,3][4,5,-6]]>MAT **[ENTER]**

[[1 -2 3]
[4 5 -6]]
9

cnorm MAT **[ENTER]**

合并： + ⊕	cnorm <i>vector</i> 返回向量 <i>vector</i> 中实数或复数元素的绝对值的和。	<code>[-1,2,-3]→VEC</code> <code>[ENTER]</code> <code>cnorm VEC</code> <code>[ENTER]</code>	<code>[-1 2 -3]</code> <code>6</code>
	stringA+stringB 返回一个字符串，该字符串是将字符串 <i>stringB</i> 附加（合并）到字符串 <i>stringA</i> 的尾部后得到的。	<code>"your name:"→STR</code> <code>[ENTER]</code> <code>"Enter "+STR</code> <code>[ENTER]</code>	<code>your name:</code> <code>Enter your name:</code>
cond MATRX MATH 菜单	cond <i>squareMatrix</i> 返回一个实数或复数方阵 <i>squareMatrix</i> 的条件数，条件数的计算如下： cnorm <i>squareMatrix</i> * cnorm <i>squareMatrix</i> ⁻¹ 条件数表示方阵 <i>squareMatrix</i> 对于某些矩阵函数，特别是求逆，其性态如何。对于良性矩阵，条件数接近 1。 log(cond <i>squareMatrix</i>) 表示在计算逆矩阵时因舍入误差而可能丢失的位数。 对于逆不存在的矩阵， cond 返回一个错误。	<code>[[1,0,0][0,1,0][0,0,1]]→MAT1</code> <code>[ENTER]</code> <code>cond MAT1</code> <code>[ENTER]</code> <code>log (Ans)</code> <code>[ENTER]</code> <code>[[1,2,3][4,5,6][7,8,9]]→MAT2</code> <code>[ENTER]</code> <code>cond MAT2</code> <code>[ENTER]</code> <code>log (Ans)</code> <code>[ENTER]</code>	<code>[[1 0 0]</code> <code>[0 1 0]</code> <code>[0 0 1]]</code> <code>1</code> <code>0</code> <code>[[1 2 3]</code> <code>[4 5 6]</code> <code>[7 8 9]]</code> <code>1.8E14</code> <code>14.2552725051</code>

conj

- CPLX 菜单
- MATRIX CPLX 菜单
- VECTR CPLX 菜单

conj (<i>complexNumber</i>)	在 RectC 复数方式下:
返回复数 <i>complexNumber</i> 的共轭复数。	conj (3,4) ENTER (3,-4)
在 RectC 方式下, conj (<i>real,imaginary</i>) 返回 (<i>real,-imaginary</i>)。	conj (3∠2) ENTER (-1.24844050964,-2.7...
在 PolarC 方式下, conj (<i>magnitude∠angle</i>) 返回 (<i>magnitude∠-angle</i>), $-\pi < angle \leq \pi$ 。	在 PolarC 复数方式下:
	conj (3∠2) ENTER (3∠-2)
conj <i>complexList</i>	conj (3,4) ENTER (5∠-.927295218002)
conj <i>complexMatrix</i>	conj {√-2,(3,4)} ENTER
conj <i>complexVector</i>	{(1.41421356237∠-1.5...
返回复数数组、矩阵或向量, 其中每个元素是原数据的共轭复数。	

CoordOff

- ↑ 图象格式屏幕

CoordOff
关闭图象底部显示的光标坐标。

CoordOn

- ↑ 图象格式屏幕

CoordOn
在图象底部显示光标坐标。

COS [COS]	cos <i>angle</i> 或 cos (<i>expression</i>) 返回角度 <i>angle</i> 或表达式 <i>expression</i> 的余弦, 其中 <i>angle</i> 或 <i>expression</i> 可以是实数, 也可以是复数。 根据当前的角度方式, 角度可以解释为度或弧度。在任何模式下, 从 MATH ANGLE 菜单, 可以分别用 ° 或 <i>r</i> 指示符指定将角度表示为度或弧度。 cos <i>list</i> 返回一个数组, 其中每个元素是数组 <i>list</i> 中相应元素的余弦值。 cos <i>squareMatrix</i> 返回一个方阵, 它是方阵 <i>squareMatrix</i> 的矩阵余弦。矩阵余弦对应于用幂级数或 Cayley-Hamilton Theorem 计算得到的结果。这并非简单的计算各元素的余弦值。	在 Radian 角度方式下: cos $\pi/2$ [ENTER] -1.5 cos ($\pi/2$) [ENTER] 0 cos 45° [ENTER] .707106781187 在 Degree 角度方式下: cos 45 [ENTER] .707106781187 cos ($\pi/2$) <i>r</i> [ENTER] 0 在 Radian 角度方式下: cos {0, $\pi/2$, π } [ENTER] {1 0 -1} 在 Degree 角度方式下: cos {0,60,90} [ENTER] {1 .5 0}
COS⁻¹ [2nd] [COS ⁻¹]	cos⁻¹ <i>number</i> 或 cos⁻¹ (<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的反余弦, 其中 <i>number</i> 或 <i>expression</i> 可以是实数, 也可以是复数。 cos⁻¹ <i>list</i> 返回一个数组, 其中每个元素是数组 <i>list</i> 中相应的元素的反余弦值。	在 Radian 角度方式下: cos ⁻¹ .5 [ENTER] 1.0471975512 在 Degree 角度方式下: cos ⁻¹ 1 [ENTER] 0 在 Radian 角度方式下: cos ⁻¹ {0,.5} [ENTER] {1.57079632679,1.047...

cosh MATH HYP 菜单	cosh <i>number</i> 或 cosh (<i>expression</i>)	<code>cosh 1.2</code> [ENTER]	1.81065556732
	返回 <i>number</i> 或 <i>expression</i> 的双曲余弦，其中 <i>number</i> 或 <i>expression</i> 可以是实数，也可以是复数。		
	cosh <i>list</i>	<code>cosh {0,1.2}</code> [ENTER]	{1 1.81065556732}
	返回一个数组，表达每个元素是数组 <i>list</i> 中相应元素的双曲余弦值。		
cosh⁻¹ MATH HYP 菜单	cosh⁻¹ <i>number</i> 或 cos⁻¹ (<i>expression</i>)	<code>cosh⁻¹ 1</code> [ENTER]	0
	返回 <i>number</i> 或 <i>expression</i> 的反双曲余弦，其中 <i>number</i> 或 <i>expression</i> 可以是实数，也可以是复数。		
	cosh⁻¹ <i>list</i>	<code>cosh⁻¹ {1,2.1,3}</code> [ENTER]	{0 1.37285914424 1.7...
	返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的反双曲余弦值。		
cross(VECTR MATH 菜单	cross (<i>vectorA</i> , <i>vectorB</i>)	<code>cross([1,2,3],[4,5,6])</code> [ENTER]	[-3 6 -3]
	返回两个实数或复数向量的向量积，例如： cross([a,b,c],[d,e,f]) = [bf-ce cd-af ae-bd]	<code>cross([1,2],[3,4])</code> [ENTER]	[0 0 -2]
	两个向量必须有相同的维数（或者 2 个元素，或者 3 个元素）。二维的向量被当作三维向量，第三个元素是 0。		

cSum(LIST OPS 菜单	cSum(<i>list</i>) 返回数组 <i>list</i> 中从第一个元素开始的各实数或复数元素的累积和。	cSum({1,2,3,4}) [ENTER] {1 3 6 10} {10,20,30}➤L1 [ENTER] cSum(L1) [ENTER] {10 20 30} {10 30 60}
►Cyl VECTR OPS 菜单	vector►Cyl 以柱面形式， [r∠θ z] 显示一个 2 或 3 元素实向量 <i>vector</i> 的结果，即使显示方式并未设置为柱面 (CylV)。	[-2,0]►Cyl [ENTER] [2∠3.14159265359 0] [-2,0,1]►Cyl [ENTER] [2∠3.14159265359 1]
CylV ↑ 方式屏幕	CylV 设置柱面向量坐标方式 ([r∠θ z])。	在 CylV 向量坐标方式和 Radian 角度方式下： [3,4,5] [ENTER] [5∠.927295218002 5]
d BASE TYPE 菜单	numberd 无论在那种数制方式设置下，都将实数 <i>number</i> 表示为十进制数。	在 Bin 数制方式下： 10d [ENTER] 1010b 10d+10 [ENTER] 1100b
Dec ↑ 方式屏幕	Dec 设置十进制数制方式。无论在何种数制方式下，从 BASE TYPE 菜单都可以分别用 b 、 d 、 h 或 o 指示符将相应的数值表示为二进制、十进制、十六进制或八进制数。	在 Dec 数制方式下： 10+10b+Fh+10o [ENTER] 35

►Dec

BASE CONV 菜单

number►**Dec**

list►Dec

matrix ▶ **Dec**

vector►Dec

返回实数或复数参数的十进制等价值。

在 **Hex** 数制方式下:

2*F

1Eh

Ans►Dec ENTER

30d

{A,B,C,D,E}►Dec ENTER

$$\{10d \ 11d \ 12d \ 13d \ 14d\}$$

Degree

† 方式屏幕

Degree

设置角度的度表示方式。

在 **Degree** 角度方式下:

sin 90 ENTER

1

sin ($\pi/2$) [ENTER]

.027412133592

度输入: °

MATH ANGLE 菜单

number^o 或 (*expression*)^o

无论在那种角度方式设置下，都将实数 *number* 或表达式 *expression* 表示为度。

在 **Radian** 角度方式下:

cos 90 ENTER

-.448073616129

cos 90° ENTER

0

 $list^{\circ}$

将数组 *list* 中的每个元素用度来表示。

$$\cos \{45, 90, 180\}^\circ \quad \boxed{\text{ENTER}}$$
$$\{.707106781187 \quad 0 \quad -1\}$$

Deltalst(

LIST OPS 菜单

(Delta 显示在菜单上)

DeltaIst(*list*)

返回一个数组，该数组包含数组 *list* 中相邻实数或复数元素之差。即数组第一个元素是 *list* 中的第一个元素减去 *list* 的第二个元素，第二个元素是 *list* 中的第二个元素减去 *list* 的第三个元素如此类推。得到的结果数组总是比 *list* 少一个元素。

Delta1st({20,30,45,70})
$$\{10 \quad 15 \quad 25\}$$

DelVar(‡ 程序编辑器 CTL 菜单 (DelVa 显示 在菜单上)	DelVar(<i>variable</i>) 从内存中删除指定的用户创建变量 <i>variable</i> 。 不能使用 DelVar(来删除程序变量或内置变量。	2→A ENTER 2 (A+2)² ENTER 16 DelVar(A) ENTER Done (A+2)² ENTER ERROR 14 UNDEFINED
der1(CALC 菜单	der1(<i>expression,variable,value</i>) 返回表达式 <i>expression</i> 在变量 <i>variable</i> 为实数或复数 <i>value</i> 时的一阶导数。 der1(<i>expression,variable</i>) 使用变量 <i>variable</i> 的当前值。 der1(<i>expression,variable,list</i>) 返回一个数组，该数组包含在由 <i>list</i> 中元素所指定值处的一阶导数。	der1(x^3,x,5) ENTER 75 3→x ENTER 3 der1(x^3,x) ENTER 27 der1(x^3,x,{5,3}) ENTER {75 27}
der2(CALC 菜单	der2(<i>expression,variable,value</i>) 返回表达式 <i>expression</i> 在变量 <i>variable</i> 为实数或复数 <i>value</i> 时的二阶导数。 der2(<i>expression,variable</i>) 使用变量 <i>variable</i> 的当前值。 der2(<i>expression,variable,list</i>) 返回一个数组，该数组包含在由 <i>list</i> 中元素所指定值处的二阶导数。	der2(x^3,x,5) ENTER 30 3→x ENTER 3 der2(x^3,x) ENTER 18 der2(x^3,x,{5,3}) ENTER {30 18}

det MATRIX MATH 菜单	det <i>squareMatrix</i> 返回方阵 <i>squareMatrix</i> 的行列式。实数矩阵返回实数值，复数矩阵返回复数值。	[[1,2][3,4]]>MAT ENTER det MAT ENTER	[[1 2] [3 4]] -2
DifEq † 方式屏幕	DifEq 设置微分方程图象方式。		
dim MATRIX OPS 菜单 VECTR OPS 菜单	dim <i>matrix</i> 返回一个数组，该数组包含实数或复数矩阵 <i>matrix</i> 的维数（行数和列数）。 dim <i>vector</i> 返回实数或复数向量 <i>vector</i> 的长度（元素个数）。	[[2,7,1][-8,0,1]]>MAT ENTER dim MAT ENTER dim [-8,0,1] ENTER	[[2 7 1] [-8 0 1]] {2 3} 3
>dim STO►,然后是 MATRIX OPS 菜单 STO►,然后是 VECTR OPS 菜单	{<i>rows,columns</i>}>dim <i>matrixName</i> 如果矩阵名 <i>matrixName</i> 不存在，用指定的维数创建一个新的矩阵，全部元素为 0。 如果矩阵名 <i>matrixName</i> 存在，用指定的维数重新构造矩阵，在新维数之内已存在的元素保持不变，之外的元素则被删除。如果创建其他元素，则值为 0。	[[2,7][-8,0]]>MAT ENTER {3,3}>dim MAT ENTER MAT ENTER	[[2 7] [-8 0]] {3 3} [[2 7 0] [-8 0 0] [0 0 0]]

	#ofElements→dim vectorName 如果向量名 <i>vectorName</i> 不存在,用指定个数 #ofElements 创建一个新向量,全部元素为 0。 如果向量名 <i>vectorName</i> 存在,用指定个数 #ofElements 重新构造向量。在新维数之内已存在的元素保持不变,之外的元素则被删除。如果创建其他元素,则值为 0。	DelVar(VEC) ENTER Done 4→dim VEC ENTER 4 VEC ENTER [0 0 0 0] [1,2,3,4]→VEC ENTER [1 2 3 4] 2→dim VEC ENTER 2 VEC ENTER [1 2] 3→dim VEC ENTER 3 VEC ENTER [1 2 0]
dimL LIST OPS 菜单	dimL list 返回实数或复数数组 <i>list</i> 的长度(元素个数)。	dimL {2,7,-8,0} ENTER 4 1/dimL {2,7,-8,0} ENTER .25
→dimL STO▶, 然后是 LIST OPS 菜单	#ofElements→dimL listName 如果数组名 <i>listName</i> 不存在,用指定个数 #ofElements 创建一个新的数组,全部元素为 0。 如果数组名 <i>listName</i> 存在,用指定个数 #ofElements 重新构造数组。在新维数内已存在的元素保持不变,之外的元素则被删除。如果创建其他元素,则值为 0。	3→dimL NEWLIST ENTER 3 NEWLIST ENTER {0 0 0} {2,7,-8,1}→L1 ENTER {2 7 -8 1} 5→dimL L1 ENTER 5 L1 ENTER {2 7 -8 1 0} 2→dimL L1 ENTER 2 L1 ENTER {2 7}
DirFld † 图象格式屏幕 (滚动到第二屏)	DirFld 在 DirEq 图象方式下,打开方向域。要关闭方向和斜率域,则用 FldOff。	

Disp

‡ 程序编辑器
I/O 菜单

Disp *valueA,valueB,valueC, ...*

显示每个值。值可以包含字符串和变量名。

Disp

显示主屏幕。

```
10→x [ENTER] 10
Disp x^3+3 x-6 [ENTER] 1024
Done
"Hello"→STR [ENTER] Hello
Disp STR+", Jan" [ENTER] Hello, Jan
Done
```

DispG

† GRAPH 菜单
‡ 程序编辑器
I/O 菜单

DispG

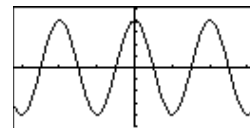
显示当前图象。

函数名大小写相关。
用 **y1**，而不要用 **Y1**。

要从窗口变量名列表中选择，按
[2nd] [CATLG-VARS] [MORE] [MORE] [F5]。

在 **Func** 图象方式下的程序段：

```
⋮
:y1=4cos x
:-10→xMin:10→xMax
:-5→yMin:5→yMax
:DispG
⋮
```



DispT

‡ 程序编辑器
I/O 菜单

除： /
⊞

DispT

显示运算表。

函数名大小写相关。
用 **y1**，而不要用 **Y1**。

程序段在 **Func** 图象方式下：

⋮
:y1=4cos x
:DispT
⋮

X	y1	
0	4	
1	2.161209	
2	1.68459	
3	-.85897	
4	-2.61457	
5	1.124649	
X=0		

numberA/*numberB* 或 (*expressionA*)/(*expressionB*)

返回一个参数被另一参数除的结果。参数可以是实数，也可以是复数。

-98/4 **ENTER** -24.5
-98/(4*3) **ENTER** -8.16666666667

number/*list* 或 (*expression*)/*list*

返回一个数组，其中每个元素是数 *number* 或表达式 *expression* 被 *list* 中相应元素除的结果。

100/{10,25,2} **ENTER** {10 4 50}

list/*number* 或 *list*/(*expression*)
vector/*number* 或 *vector*/(*expression*)

返回一个数组或向量，其中每个元素为数组 *list* 或向量 *vector* 中相应元素被 *number* 或 *expression* 除的结果。

{120,92,8}/4 **ENTER** {30 23 2}

在 **RectC** 复数方式下：
[8,1,(5,2)]/2 **ENTER**
[(4,0) (.5,0) (2.5,1...

listA/*listB*

返回一个数组，其中每个元素是 *listA* 元素被 *listB* 中相应元素除的结果。两个数组必须有相同的维数。

{1,2,3}/{4,5,6} **ENTER** { .25 .4 .5}

DMS 输入: ' MATH ANGLE 菜单 在三角计算中, DMS 输入的结果只在 Degree 角度方式下被当成度, 在 Radian 角度方式下被当成弧度。	degrees'minutes'seconds' 指明输入的角度是 DMS 格式。Degrees (≤ 999, 999)、minutes (< 60) 和 seconds (< 60, 可能有小数位) 必须输入为实数, 不能是变量名或表达式。 不能使用符号 ° 和 " 指定 degrees 和 seconds。例如, 根据当前的角度方式设置, 5°59' 被解释为蕴涵乘法 5° * 59'。	54'32'30' [ENTER] 54.5416666667 在 Degree 角度方式下: cos 54'32'30' [ENTER] .580110760699 在 Radian 角度方式下: cos 54'32'30' [ENTER] -.422502666138 在 Degree 角度方式下, 不要使用以下符号: 5°59' [ENTER] 295
►DMS MATH ANGLE 菜单	angle►DMS 以 DMS 格式显示 angle。即使使用 degrees'minutes'seconds' 来输入 DMS 角度, 其结果仍按格式 degrees°minutes'seconds" 来显示。	在 Degree 角度方式下: 45.371►DMS [ENTER] 45°22'15.6" 54'32'30'*2 [ENTER] 109.083333333 Ans►DMS [ENTER] 109°5'0"
dot(VECTR MATH 菜单	dot(vectorA,vectorB) 返回两个实数或复数向量的点积。 dot([a,b,c],[d,e,f]) 返回 a*d+b*e+c*f。	dot([1,2,3],[4,5,6]) [ENTER] 32
DrawDot † 图象格式屏幕	DrawDot 设置点图象格式。	

DrawF

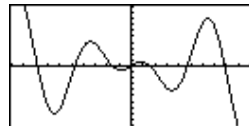
GRAPH DRAW 菜单

DrawF *expression*

在当前图象上（根据 x ）绘制 *expression*。

在 **Func** 图象方式下:

ZStd:DrawF 1.25 x cos x ENTER



DrawLine

† 图象格式屏幕

DrawLine

设置连线图象格式。

DrEqu(

† GRAPH 菜单

要输入变量 Q' 的字符', 则
使用 CHAR MISC 菜单

DrEqu(*xAxisVariable*,*yAxisVariable*,*xList*,*yList*,*tList*)

在 **DifEq** 图象方式下, 对存储在由 *xAxisVariable* 和 *yAxisVariable* 指定的 Q' 变量中的一组微分方程的解进行。如果方向域是关闭的 (选择了 **FldOff**) , 初始值也必须保存起来。

结果绘制完后, **DrEqu**(等待将光标移到一个新的初始值处, 并按 **[ENTER]** 来绘制新结果。

然后会提醒按 **Y** (指定另一个初始值) 或者 **N** (停止)。

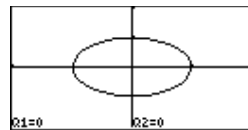
对于最后绘制的解, **x**、**y** 和 **t** 的值 (从它们的初始值开始) 都分别保存在 *xList*、*yList* 和 *tList* 中。

DrEqu(*xAxisVariable*,*yAxisVariable*)

不存储解的 **x**、**y** 和 **t** 的值。

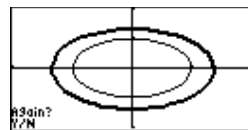
在 **DifEq** 图象方式下, 从 **ZStd** 图象屏幕开始:

```
Q'1=Q2:Q'2=-Q1 [ENTER]           Done
0>tMin:1>QI1:0>QI2 [ENTER]         0
DrEqu(Q1,Q2,XL,YL,TL) [ENTER]
```



将光标移到新的初始值处。

[ENTER]



按下 **N** 停止绘制。然后检查 **XL**、**YL** 和 **TL**。

DrInv

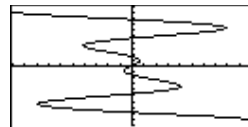
GRAPH DRAW 菜单

DrInv *expression*

通过在 **y** 轴绘制 **x** 的值, 在 **x** 轴绘制 **y** 的值来绘制 *expression* 的逆元。

在 **Func** 图象方式下:

```
ZStd:DrInv 1.25 x cos x [ENTER]
```



DS<(‡ 程序编辑器 CTL 菜单	:DS<(variable,value) :command-if-variable≥value :commands 变量 <i>variable</i> 减 1。如果结果 < <i>value</i>，跳过 <i>command-if-variable≥value</i>。 如果结果 ≥ <i>value</i>，则执行 <i>command-if-variable≥value</i>。 <i>variable</i> 不能是内置变量。	程序段: : :9>A :Lb1 Start :Disp A :DS<(A,5) :Goto Start :Disp "A is now <5" :
dxDer1 † 方式屏幕	dxDer1 将 der1 设置为当前的微分类型。der1 进行准确微分，并计算表达式中的每个函数的值。它比 dxNDer 更精确，但是它的限制比较严，只有某些函数在表达式中有效。	当前微分类型被 arc(和 TanLn(函数使用，也被交互式图象操作 dy/dx、dr/dθ、dy/dt、dx/dt、ARC、TanLn 和 INFLC 使用。
dxNDer † 方式屏幕	dxNDer 将 nDer 设置为当前的微分类型。nDer 进行数值微分，并计算表达式的值。不如 dxDer1 精确，但对表达式中函数有效性的限制不严。	当前微分类型被 arc(和 TanLn(函数使用，也被交互式图象操作 dy/dx、dr/dθ、dy/dt、dx/dt、ARC、TanLn 和 INFLC 使用。
e^ [2nd] [e^x]	e^power 或 e^(expression) 返回以 e 为底的 <i>power</i> 或 <i>expression</i> 次幂。参数可以是实数或复数。	e^0 [ENTER] 1

e^{list}

返回一个数组, 其中每个元素是数组 *list* 中相应元素为指数以 **e** 为底的幂。

e^{1,0,.5} **[ENTER]**
{2.71828182846 1 1.6...}

e^{squareMatrix}

返回一个方阵, 它是 *squareMatrix* 的矩阵指数。矩阵指数对应幂级数或 Cayley-Hamilton Theorem 技术计算的结果。这并非简单的计算每个元素的指数。

方阵 *squareMatrix* 不能有重复的特征值。

eigVc

MATRIX MATH 菜单

方阵 *squareMatrix* 不能有重复的特征值。

eigVc squareMatrix

返回一个包含实数或复数 *squareMatrix* 特征向量的矩阵, 其中结果的每列相当于一个特征值。实数矩阵的特征向量可能是复数。注意特征向量是不唯一的; 它可能会乘以一个常数因子。TI-86 的特征向量是标准特向量。

在 **RectC** 复数方式下:

[[[-1,2,5][3,-6,9][2,-5,7]]>MAT
[ENTER]
[[[-1 2 5]
[3 -6 9]
[2 -5 7]]

eigVc MAT **[ENTER]**
[[(.800906446592,0) ...
[(-.484028886343,0) ...
[(-.352512270699,0) ...

eigVl

MATRIX MATH 菜单

eigVl squareMatrix

返回一个包含实数或复数 *squareMatrix* 特征值的数组, 实数矩阵的特征值可能是复数。

在 **RectC** 复数方式下:

[[[-1,2,5][3,-6,9][2,-5,7]]>MAT
[ENTER]
[[[-1 2 5]
[3 -6 9]
[2 -5 7]]

eigVl MAT **[ENTER]**
{(-4.40941084667,0) ...

Else	请参阅 305 页 If 的语法信息。请参阅 If:Then:Else:End 的语法。		
‡ 程序编辑器 CTL 菜单			
End	End		
‡ 程序编辑器 CTL 菜单	标识 While 、 For 、 Repeat 或 If-Then-Else 循环的结束。		
Eng	Eng		
† 方式屏幕	设置工程记数法方式，其中 10 的指数是 3 的倍数。	在 Eng 记数法方式下: 123456789 ENTER 123.456789E6 在 Normal 记数法方式下: 123456789 ENTER 123456789	
Eq→St(	Eq→St(<i>equationVariable</i> , <i>stringVariable</i>)		
STRNG 菜单	将方程变量 <i>equationVariable</i> 的内容转换为一个字符串，并存储在字符串变量 <i>stringVariable</i> 中。确定指定的是方程变量，而不是方程。 要创建一个方程变量，则使用等号 (=) 定义变量。例如，输入 A=B*C ，不是 B*C→A 。	A=B*C ENTER Done 5→B ENTER 5 2→C ENTER 2 A ENTER 10 Eq→St(A,STR) ENTER Done STR ENTER B*C	
等于: =	请参阅第 270 页的 Assignment 语法信息。		
ALPHA [=]	如果在表达式中使用了 =，而该表达式在行开始处的第一个参数不是变量名，则 = 会被处理为 -(。	= 被处理为 -(的例子，其中 4=6+1 计算为 4-(6+1): 4=6+1 ENTER -3 对于真/假的比较，则用 == : 4==6+1 ENTER 0	

相等: ==

TEST 菜单

操作符 == 用来比较参数,
而 = 用来将值或表达式赋给
变量。

$numberA == numberB$	$2+2==2+2$ ENTER	1
$matrixA == matrixB$	$2+(2==2)+2$ ENTER	5
$vectorA == vectorB$	$[1,2]==[3-2,-1+3]$ ENTER	1
$stringA == stringB$	"A"=="a" ENTER	0
测试条件 $argumentA == argumentB$ 是真还是假。数值、矩阵和方式可以是实数,也可以是复数。如果是复数,则比较每个元素的模。字符串是区分大小写的。		
• 如果为真 ($argumentA = argumentB$), 则返回 1。 • 如果为假 ($argumentA \neq argumentB$), 则返回 0。		
$listA == listB$	$\{1,5,9\}==\{1,-6,9\}$ ENTER	$\{1\ 0\ 1\}$
返回一个 1 和/或 0 的数组来指明在 $listA$ 中的每个元素是否 = $listB$ 中的相应元素。		

Euler

† 图象格式屏幕
(向下滚动到
第二屏)

Euler

在 **DifEq** 图象方式下,使用基于 **Euler** 方法的算法来解微分方程。**Euler** 方法一般不如 **RK** 方法精确,但是求解速度比较快。

eval

MATH MISC 菜单

eval $xValue$	记住内置方程变量 y1 和 y2 是区分大小写的。	
返回一个数组,该数组元素为所有已定义和选择函数在实数 $xValue$ 值处的 y 值。	$y1=x^3+x+5$ ENTER	Done
	$y2=2\ x$ ENTER	Done
	eval 5 ENTER	{135 10}

evalF(CALC 菜单	evalF (<i>expression,variable,value</i>)	evalF(x^3+x+5,x,5) ENTER	135
	返回表达式 <i>expression</i> 在变量 <i>variable</i> 为实数或复数 <i>value</i> 时的值。		
	evalF (<i>expression,variable,list</i>)	evalF(x^3+x+5,x,{3,5}) ENTER	{35 135}
	返回一个数组，其中每个元素是表达式 <i>expression</i> 在变量 <i>variable</i> 为数组 <i>list</i> 中相应元素时的值。		
指数： E EE	<i>number</i> E <i>power</i> 或 (<i>expression</i>) E (<i>expression</i>)	12.3456789E5 ENTER (1.78/2.34)E2 ENTER	1234567.89 76.0683760684
	返回实数或复数 <i>number</i> 乘以 10 的 <i>power</i> 次方的结果， <i>power</i> 是一个整数，范围 -999 < <i>power</i> < 999。任何 <i>expressions</i> 必须计算得到相应的值。		
	<i>list</i> E <i>power</i> 或 <i>list</i> E (<i>expression</i>)	{6.34,854.6}E3 ENTER	{6340 854600}
	返回一个数组，其中每个元素是 <i>list</i> 中相应元素乘以 10 的 <i>power</i> 次方。		

ExpR

STAT CALC 菜单

内置方程变量如 **y1**、**r1** 和 **xt1** 是区分大小写的。不要使用 **Y1**、**R1** 和 **XT1**。

ExpR *xList,yList,frequencyList,equationVariable*

用指数回归模型 ($y=ab^x$) 拟合 *xList* 和 *yList* (*y* 值必须 > 0) 中实数对, 频数为 *frequencyList*。回归方程保存在方程变量 *equationVariable* 中, 它必须是一个内置方程变量, 例如 **y1**、**r1** 和 **xt1**。

用于 *xList*, *yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

ExpR *xList,yList,equationVariable*

频数为 1。

ExpR *xList,yList,frequencyList*

只将回归方程保存在 **RegEq** 中。

ExpR *xList,yList*

频数为 1, 同时只将回归方程保存在 **RegEq** 中。

ExpR *equationVariable*

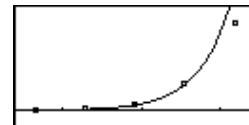
xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat** 的值。这些内置变量必须包含有相同维数的有效数据; 否则会产生错误。回归方程保存在方程变量 *equationVariable* 和 **RegEq** 中。

在 **Func** 图象方式下:

```
{1,2,3,4,5}→L1 [ENTER]
{1,20,55,230,742}→L2 [ENTER]
ExpR L1,L2,y1 [ENTER]
```

```
ExpReg
y=a*b^x
a=.411389488
b=4.78796057
corr=.97681282
n=5
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER] Done
```



		ExpR	
		使用 xStat 、 yStat 和 fStat 的值，同时只将回归方程保存在 RegEq 中。	
阶乘: ! MATH PROB 菜单	<i>number!</i> 或 (<i>expression</i>)!	6! ENTER	720
	返回一个整数或非整数的阶乘，其中整数范围为 0 ≤ 到 ≤ 449，非整数范围为 0 ≤ 到 ≤ 449.9。对于一个非整数，用 Gamma 函数来求阶乘。 <i>expression</i> 必须能够计算出合适的解。	12.5! ENTER	1710542068.32
	<i>list!</i> 返回一个数组。其中每个元素是数组 <i>list</i> 中相应元素的阶乘。	{6,7,8}! ENTER	{720 5040 40320}
fcstx † STAT 菜单		fcstx <i>yValue</i> 基于当前回归方程 (ReqEq)，根据实数值 <i>yValue</i> 返回预测的 x 值。	
fcsty † STAT 菜单		fcsty <i>xValue</i> 基于当前回归方程 (ReqEq)，根据实数值 <i>xValue</i> 返回预测的 y 值。	

Fill(
LIST OPS 菜单
MATRX OPS 菜单
VECTR OPS 菜单

Fill (<i>number</i> , <i>listName</i>)	{3,4,5}→L1 ENTER	{3 4 5}
Fill (<i>number</i> , <i>matrixName</i>)	Fill(8,L1) ENTER	Done
Fill (<i>number</i> , <i>vectorName</i>)	L1 ENTER	{8 8 8}
用一个实数或复数 <i>number</i> 替换已有数组名 <i>listName</i> , 矩阵名 <i>matrixName</i> 或向量名 <i>vectorName</i> 中的每个 元素。	Fill((3,4),L1) ENTER L1 ENTER	Done {(3,4) (3,4) (3,4)}

Fix
† 方式屏幕

Fix <i>integer</i> 或 Fix (<i>expression</i>)	Fix 3 ENTER	Done
设置固定小数方式为整数 <i>integer</i> 个小数位，其中 $0 \leq integer \leq 11$ 。表达式 <i>expression</i> 必须能够计算出 一个合适的整数。	$\pi/2$ ENTER Float ENTER $\pi/2$ ENTER	1.571 Done 1.57079632679

FldOff
† 图象格式屏幕
(向下滚动到
第二屏)

FldOff
在 **DifEq** 图象方式下，关闭斜率和方向域。用 **SlpFld**
打开斜率域；用 **DirFld** 打开方向域。

Float
† 方式屏幕

Float	在 Radian 角度方式下：	
设置浮点小数方式。	Fix 11 ENTER	Done
	sin ($\pi/6$) ENTER	.50000000000
	Float ENTER	Done
	sin ($\pi/6$) ENTER	.5

fMax(CALC 菜单	fMax(<i>expression,variable,lower,upper</i>) 返回表达式 <i>expression</i> 的极大值,表达式的变量 <i>variable</i> 取值范围在实数 <i>lower</i> 和 <i>upper</i> 之间。 误差通过内置变量 tol 来控制,其默认值为 1E-5 。要查看或设置 tol , 则按 [2nd] [MEM] [F4] 显示误差编辑器。	fMax(sin x,x,-π,π) [ENTER] 1.57079632598
fMin(CALC 菜单	fMin(<i>expression,variable,lower,upper</i>) 返回表达式 <i>expression</i> 的极小值,表达式的变量 <i>variable</i> 取值范围在实数 <i>lower</i> 和 <i>upper</i> 之间。 误差通过内置变量 tol 来控制,其默认值为 1E-5 。要查看或设置 tol , 则按 [2nd] [MEM] [F4] 显示误差编辑器。	fMin(sin x,x,-π,π) [ENTER] -1.57079632691
fnInt(CALC 菜单	fnInt(<i>expression,variable,lower,upper</i>) 返回表达式 <i>expression</i> 关于变量 <i>variable</i> 的数值函数积分, 变量 <i>variable</i> 取值范围在实数 <i>lower</i> 和 <i>upper</i> 之间。 误差通过内置变量 tol 来控制,默认值为 1E-5 。要查看或设置 tol , 按 [2nd] [MEM] [F4] 显示误差编辑器。	fnInt(x²,x,0,1) [ENTER] .333333333333
FnOff † GRAPH VARS 菜单	FnOff <i>function#</i>,<i>function#</i>, ... 取消选择指定方程函数的序号。	FnOff 1,3 [ENTER] Done

FnOff	取消选择所有方程函数的序号。	FnOff ENTER	Done
FnOn <i>function#</i> , <i>function#</i> , ...	选择指定方程函数的序号，除了已选择的。	FnOn 1,3 ENTER	Done
FnOn	选择所有方程函数的序号。	FnOn ENTER	Done
:For (<i>variable</i> , <i>begin</i> , <i>end</i> , <i>step</i>) 或 :loop :End :commands	:For (<i>variable</i> , <i>begin</i> , <i>end</i>) :loop :End :commands 迭代执行循环 <i>loop</i> 中的命令，其中迭代次数由变量 <i>variable</i> 控制。第一次进入循环时， <i>variable</i> = <i>begin</i> 。循环结束处，变量 <i>variable</i> 递增步长 <i>step</i> 。循环重复执行直到 <i>variable</i> > <i>end</i> 。如果未指定步长 <i>step</i> ， 其默认值为 1 。 可以指定值使得 <i>begin</i> > <i>end</i> 。如果这样，确认指定一个负数步长 <i>step</i> 。	程序段： : : For(A,0,8,2) Disp A ² End :<	

Form(LIST OPS 菜单	Form("formula",listName) 基于相连的公式 <i>formula</i> , 自动生成数组名 <i>listName</i> 的内容。如果用数组表达公式 <i>formula</i> , 则可以在另一个数组内容基础上生成一个新数组。 当编辑公式 <i>formula</i> 或编辑公式引用的数组时, <i>listName</i> 的内容会自动更新。	{1,2,3,4}➔L1 ENTER {1 2 3 4} Form("10*L1",L2) ENTER Done L2 ENTER {10 20 30 40} {5,10,15,20}➔L1 ENTER L2 ENTER {5 10 15 20} {50 100 150 200} Form("L1/5",L2) ENTER Done L2 ENTER {1 2 3 4}
fPart MATH NUM 菜单	fPart <i>number</i> 或 fPart (<i>expression</i>) 返回一个实数或复数 <i>number</i> 或表达式 <i>expression</i> 的小数部分。	fPart 23.45 ENTER .45 fPart (-17.26*8) ENTER -.08
	fPart <i>list</i> fPart <i>matrix</i> fPart <i>vector</i> 返回一个数组、矩阵或向量, 其每个元素是指定参数中相应元素的小数部分。	[[1,-23.45][-99.5,47.15]]➔MAT ENTER [[1 -23.45] [-99.5 47.15]] fPart MAT ENTER [[0 -.45] [-.5 .15]]
Frac MATH MISC 菜单	<i>number</i> ➔Frac 将一个实数或复数 <i>number</i> 显示为它的等价有理数, 分数部分简化为最简单项。 如果 <i>number</i> 不能被简化, 或者分母超过 4 位数, 则返回等价的小数。	1/3+2/7 ENTER .619047619048 Ans➔Frac ENTER 13/21

<i>list</i> ►Frac	{1/2+1/3,1/6-3/8}►L1 ENTER
<i>matrix</i> ►Frac	{.8333333333333333 - .208...
<i>vector</i> ►Frac	Ans►Frac ENTER {5/6 -5/24}
返回一个数组、矩阵和方式，其中每个元素是参数中相应元素的等价有理数。	

Func

† 式向屏幕

gcd(

MATH MISC 菜单

Func

设置函数图象方式。

gcd (<i>integerA</i> , <i>integerB</i>)	gcd(18,33) ENTER	3
返回两个非负整数的最大公约数。		
gcd (<i>listA</i> , <i>listB</i>)	gcd({12,14,16},{9,7,5}) ENTER	{3 7 1}
返回一个数组，其中每个元素是数组 <i>listA</i> 和数组 <i>listB</i> 中两个相应元素的最大公约数。		

Get(

‡ 程序编辑器
I/O 菜单

Get(variable)

从 CBL 或 CBR 系统或另一台 TI-86 中获取数据，并将它保存在变量 *variable* 中。

getKy ‡ 程序编辑器 I/O 菜单	getKy 返回上次按下的键代码。如果没有按任何键，getKy 返回 0。请参阅 16 章的 TI-86 键码图。	程序: PROGRAM:CODES :Lbl TOP :getKy>KEY :While KEY==0 : getKy>KEY :End :Disp KEY :Goto TOP 终止程序按 [ON]，然后按 [F5]。
Goto ‡ 程序编辑器 CTL 菜单	Goto label 将程序控制跳转（分支）到指定的标记 label 处，标记由已存在的 Lbl 指令指定。	程序段: : :0>TEMP:1>J :Lbl TOP :TEMP+J>TEMP :If J<10 :Then : J+1>J : Goto TOP :End :Disp TEMP :
大于: > TEST 菜单	numberA > numberB 或 (expressionA) > (expressionB) 测试条件是真还是假。参数必须是实数。 - 如果为真 (numberA > numberB)，返回 1。 - 如果为假 (numberA ≤ numberB)，返回 0。	2>0 [ENTER]1 88>123 [ENTER]0 -5>-5 [ENTER]0 (20*5/2)>(18*2) [ENTER]1

<i>number>list</i>	<code>1>{1,-6,10} [ENTER]</code>	<code>{0 1 0}</code>
返回一个包含 1 和/或 0 的数组，其中每个元素表示 <i>number</i> 是否 > 数组 <i>list</i> 中相应的元素。		
<i>listA>listB</i>	<code>{1,5,9}>{1,-6,10} [ENTER]</code>	<code>{0 1 0}</code>
返回一个包含 1 和/或 0 的数组，其中每个元素表示数组 <i>listA</i> 中的每个元素是否 > 数组 <i>listB</i> 中相应的元素。		
<i>numberA ≥ numberB</i> 或 (expressionA) ≥ (expressionB)	<code>2≥0 [ENTER]</code>	<code>1</code>
测试条件是真还是假。参数必须是实数。		
• 如果为真 (<i>numberA ≥ numberB</i>)，返回 1 。		
• 如果为假 (<i>numberA < numberB</i>)，返回 0 。		
<i>number ≥ list</i>	<code>1≥{1,-6,10} [ENTER]</code>	<code>{1 1 0}</code>
返回一个包含 1 和/或 0 的数组，其中每个元素表示 <i>number</i> 是否 ≥ 数组 <i>list</i> 中相应元素。		
<i>listA ≥ listB</i>	<code>{1,5,9}≥{1,-6,10} [ENTER]</code>	<code>{1 1 0}</code>
返回一个包含 1 和/或 0 的数组，其中每个元素表示数组 <i>listA</i> 中的每个元素是否 ≥ 数组 <i>listB</i> 中相应元素。		

GridOff

† 图象格式屏幕

GridOff
关闭网格格式，不显示网格点。

GridOn

† 图象格式屏幕

GridOn

打开网格格式，按行和列显示网格点，行和列对应于坐标轴上的标记。

GrStl(

CATALOG

GrStl(function#,graphStyle#)

设置 *function#* 图象样式。对 *graphStyle#* 指定一个从 1 到 7 之间的整数：

- 1 = （实线）
- 4 = （下阴影）
- 7 = （点画线）
- 2 = （粗线）
- 5 = （路径）
- 3 = （上阴影）
- 6 = （动画）

取决于图象方式，一些图象样式可能无效。

在 **Func** 图象方式下：

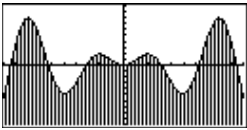
y1=x sin x

GrStl(1,4)

ZStd

Done

Done



h

BASE TYPE 菜单

integerh

将一个整数 *integer* 表示为十六进制，无论设置了何种数制方式。

在 **Dec** 数制方式下：

10h

10h+10

16

26

Hex

† 方式屏幕

Hex

设置十六进制数制方式。结果显示有 **h** 后缀。无论在何种数制方式下，从 **BASE TYPE** 菜单都可以分别用 **b**、**d**、**h** 或 **o** 指示符将相应的数值指定为二进制、十进制、十六进制或八进制数。

要输入十六进制数 **A** 到 **F**，则使用 **BASE A-F** 菜单，不要使用 来键入字母。

在 **Hex** 数制方式下：

F+10b+10o+10d

23h

►Hex

BASE CONV 菜单

number►Hex

list►Hex

matrix►Hex

vector►Hex

返回实数或复数参数的等价十六进制数。

在 Bin 数制方式下:

1010*1110 [ENTER] 10001100b

Ans►Hex [ENTER] 8Ch

{100,101,110}►Hex [ENTER] {4h 5h 6h}

Hist

† STAT DRAW 菜单

Hist *xList*,*frequencyList*

用 *xList* 中的实数数据和 *frequencyList* 中的频数, 在当前图象上绘出直方图。

Hist *xList*

频数为 1。

Hist

使用内置变量 **xStat** 和 **fStat** 中的数据。这些变量必须有相同维数的有效数据; 否则会产生错误。

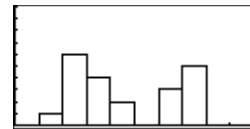
从一个 **ZStd** 图象屏幕开始:

{1,2,3,4,6,7}→XL [ENTER] {1 2 3 4 6 7}

{1,6,4,2,3,5}→FL [ENTER] {1 6 4 2 3 5}

0→xMin:0→yMin [ENTER] 0

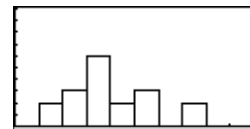
Hist XL,FL [ENTER]



{1,1,2,2,2,3,3,3,3,3,3,4,4,5,5,5,7,7}→XL [ENTER]

{1 1 2 2 2 3 3 3 3 3 3 ...

C1Drw:Hist XL [ENTER]



Horiz

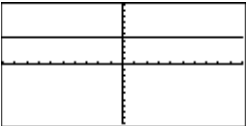
↑ GRAPH DRAW 菜单

Horiz *yValue*

在当前图象上在 *yValue* 值处绘制一条水平线。

在一个 **ZStd** 图象屏幕中：

Horiz 4.5 **ENTER**



IAsk

CATALOG

IAsk

设置运算表允许用户输入自变量值。

IAuto

CATALOG

IAuto

设置运算表使 TI-86 基于 **TblStart** 和 **ΔTbl** 的输入值自动生成自变量值。

ident

MATRX OPS 菜单

ident *dimension*

返回 *dimension* 行 × *dimension* 列的单位矩阵。

ident 4 **ENTER**

```
[ [1 0 0 0]
[ 0 1 0 0]
[ 0 0 1 0]
[ 0 0 0 1]]
```

If

‡ 程序编辑器
CTL 菜单

```
:If condition
:command-if-true
:commands
```

如果条件 *condition* 为真, 执行 *command-if-true*。否则跳过 *command-if-true*。若计算得到任何非零值, 则条件 *condition* 为真; 若得到零值, 则为假。

如果条件 *condition* 为真时需要执行多条命令, 则使用 **If:Then:End** 来代替。

```
:If condition
:Then
:commands-if-true
:End
:commands
```

如果条件 *condition* 为真 (非零), 执行从 **Then** 到 **End** 之间的命令 *commands-if-true*。否则跳过 *commands-if-true*, 继续执行在 **End** 之后的下一条命令。

程序段:

```
⋮
:If x<0
:Disp "x is negative"
⋮
```

程序段:

```
⋮
:If x<0
:Then
:Disp "x is negative"
:abs(x)→x
:End
⋮
```

```
:If condition
:Then
:commands-if-true
:Else
:commands-if-false
:End
:commands
```

如果条件 *condition* 为真（非零），执行从 **Then** 到 **Else** 之间的命令 *commands-if-true*。然后继续执行在 **End** 之后的下一条命令。

如果条件 *condition* 为假（零），执行从 **Else** 到 **End** 之间的命令 *commands-if-false*。然后继续执行在 **End** 之后的下一条命令。

程序段:

```
:
:If x<0
:Then
: Disp "x is negative"
:Else
: Disp "x is positive or zero"
:End
:
```

imag

CPLX 菜单

imag (complexNumber)	imag (3,4) ENTER	4
返回复数 <i>complexNumber</i> 的虚部（非实数）。实数的虚部总为 0。	imag (3∠4) ENTER	-2.27040748592
imag (real,imaginary) 返回 <i>imaginary</i> 。		
imag (magnitudeAngle) 返回 <i>magnitude sin angle</i> 。		
imag complexList	imag {-2,(3,4),(3∠4)} ENTER	
imag complexMatrix		{0 4 -2.27040748592}
imag complexVector		
返回一个数组、矩阵或向量，其中每个元素是原参数的虚部。		

InpSt

‡ 程序编辑器
I/O 菜单

InpSt <i>promptString,variable</i>	程序段:
暂停程序，显示提示信息 <i>promptString</i> ，并等待用户输入响应。该响应以字符串方式保存在变量 <i>variable</i> 中。 当输入响应时，用户不需要输入引号。	:InpSt "Enter your name:",STR
要提示输入数字或表达式，而不是字符串，则用 Input 。	:
InpSt <i>variable</i>	:
将 ? 显示为提示符。	:

Input

‡ 程序编辑器
I/O 菜单

Input <i>promptString,variable</i>	程序段:
暂停程序，显示提示信息 <i>promptString</i> ，并等待用户输入响应。该响应以用户输入的方式保存在变量 <i>variable</i> 中。	:Input "Enter test score:",SCR
• 数字或表达式保存为数字或表达式。	:
• 数组、向量或矩阵保存为数组、向量或矩阵。	:
• 带有 " 的输入保存为字符串。	:
Input <i>variable</i>	
将 ? 显示为提示符。	

	Input 暂停程序，显示图象屏幕，允许用户通过移动自由移动光标更新 x 和 y（或 PolarGC 图象格式中的 r 和 θ）。按 [ENTER] 继续执行程序。	在 RectGC 图象格式中的程序段：	
		<pre>⋮ :Input :Disp x,y ⋮</pre> Input "CBLGET",L1 [ENTER] Done	
	Input "CBLGET",<i>variable</i> 接收从 CBL 或 CBR 系统发送来的数组数据，并将它存储在 TI-86 的变量 <i>variable</i> 中。对 CBL 和 CBR 都可使用 "CBLGET" 语法。 也可通过使用 Get[来接收数据，请参阅第 299 页的解释。		
int MATH NUM 菜单	int <i>number</i> 或 int (<i>expression</i>) 返回小于或等于 $\leq$ <i>number</i> 或 <i>expression</i> 的最大整数。参数可以是实数，也可以是复数。 对于一个非整数的负数，int 返回的整数小于该数的整数部分。要返回确切的整数部分，则使用 iPart。	int 23.45 [ENTER] 23 int -23.45 [ENTER] -24	
	int <i>list</i> int <i>matrix</i> int <i>vector</i> 返回一个数组、矩阵或向量，其中每个元素是小于或等于指定参数中相应元素的最大整数。	[[1.25,-23.45][-99,47.15]]→MAT [ENTER] [[1.25 -23.45] [-99 47.15]] int MAT [ENTER] [[1 -24] [-99 47]]	

inter(
† MATH 菜单

inter (<i>x1,y1,x2,y2,xValue</i>)	使用点 (3,5) 和 (4,4), 求当 $x=1$ 时的 y 的值: inter (3,5,4,4,1) ENTER	7
inter (<i>y1,x1,y2,x2,yValue</i>)	使用点 (-4,-7) 和 (2,6), 求当 $y=10$ 时 x 的值: inter (-7,-4,6,2,10) ENTER	3.84615384615

求逆: $^{-1}$
2nd [x^{-1}]

<i>number</i> $^{-1}$ 或 (<i>expression</i>) $^{-1}$	5^{-1} ENTER	.2
返回实数或复数 <i>number</i> 的倒数, 其中 <i>number</i> $\neq 0$ 。	(10*6) $^{-1}$ ENTER	.016666666667
<i>list</i> $^{-1}$	{-.5,10,2/8} $^{-1}$ ENTER	{-2 .1 4}
返回一个数组, 其中每个元素为数组 <i>list</i> 中相应元素的倒数。		
<i>squareMatrix</i> $^{-1}$	[[1,2][3,4]] $^{-1}$ ENTER	[[-2 1] [1.5 - .5]]
返回方阵 <i>squareMatrix</i> 的逆矩阵, 其中矩阵行列式 $\neq 0$ 。		

iPart
MATH NUM 菜单

iPart <i>number</i> 或 iPart (<i>expression</i>)	iPart 23.45 ENTER	23
返回数 <i>number</i> 或表达式 <i>expression</i> 的整数部分。参数可以是实数, 也可以是复数。	iPart -23.45 ENTER	-23

	iPart list iPart matrix iPart vector 返回一个数组、矩阵或向量，其中每个元素是指定参数中相应元素的整数部分。	[[1.25,-23.45][-99.5,47.15]]>MAT ENTER [[1.25 -23.45] [-99.5 47.15]] iPart MAT ENTER [[1 -23] [-99 47]]
IS>(† 程序编辑器 CTL 菜单	:IS>(variable,value) :command-if-variable≤value :commands 变量 <i>variable</i> 增 1。如果结果 > <i>value</i>，跳过 <i>command-if-variable≤value</i>。 如果结果 ≤ <i>value</i>，则执行 <i>command-if-variable≤value</i>。 <i>variable</i> 不能是内置变量。	程序段: : :0>A :Lbl Start :Disp A :IS>(A,5) :Goto Start :Disp "A is now >5" :
LabelOff † 图象格式屏幕	LabelOff 关闭坐标轴标记。	
LabelOn † 图象格式屏幕	LabelOn 打开坐标轴标记。	

Lbl

‡ 程序编辑器
CTL 菜单

Lbl *label*

创建一个不超过 8 个字符的标记 *label*。程序可以使用 **Goto** 指令转移控制（分支）到指定的标记处。

InpSt 将输入存储为字符串。这
样可确保将字符串存储到
password 变量中。

程序段，假定正确的口令已保存在变量 **password** 中：

```

:
:Lbl Start
:InpSt "Enter password:",PSW
:If PSW≠password
:Goto Start
:Disp "Welcome"
:

```

lcm(

MATH MISC 菜单

lcm(*integerA*,*integerB*)

返回两个非负整数的最小公倍数。

lcm(5,2)	ENTER	10
lcm(6,9)	ENTER	18
lcm(18,33)	ENTER	198

LCust(

‡ 程序编辑器
CTL 菜单

LCust(*item#*,"*title*"[,*item#*,"*title*", ...])

加载（定义） TI-86 的定制菜单，当用户按 后显示，菜单最多有 15 项，按每组五项，共三组进行显示。
对于每对 *item#*/*title*：

- *item#* — 是一个 1 到 15 的整数，标识项在菜单中的位置。项数字必须按顺序指定，但可跳过数字。
- "*title*" — 是一个至多 8 个字符的字符串（不记引号），在项被选中时，它将粘贴到当前光标处。可以是变量名、表达式、函数名、程序名或任何文本串。

程序段：

```

:
:LCust(1,"t",2,"Q'1",3,"Q'2",4,"R
K",5,"Euler",6,"QI1",7,"QI2",8,"t
Min")
:

```

执行后，当用户按 ：

t	Q'1	Q'2	RK	Euler
---	-----	-----	----	-------

小于：< TEST 菜单	$numberA < numberB$ 或 $(expressionA) < (expressionB)$	$2 < 0$ ENTER	0
	测试条件是真还是假。参数必须是实数。	$88 < 123$ ENTER	1
	• 如果是真 ($numberA < numberB$)，返回 1 。	$-5 < -5$ ENTER	0
	• 如果是假 ($numberA \geq numberB$)，返回 0 。	$(20 * 5 / 2) < (18 * 3)$ ENTER	1
	$number < list$	$1 < \{1, -6, 10\}$ ENTER	{0 0 1}
	返回一个包含 1 和/或 0 的数组，其中每个元素表示 $number$ 是否 < 数组 $list$ 中相应的元素。		
	$listA < listB$	$\{1, 5, 9\} < \{1, -6, 10\}$ ENTER	{0 0 1}
	返回一个包含 1 和/或 0 的数组，其中每个元素表示数组 $listA$ 中的每个元素是否 < 数组 $listB$ 中相应的元素。		
	$numberA \leq numberB$ 或 $(expressionA) \leq (expressionB)$	$2 \leq 0$ ENTER	0
	测试条件是真还是假。参数必须是实数。	$88 \leq 123$ ENTER	1
小于或等于：≤ TEST 菜单	• 如果是真 ($numberA \leq numberB$)，返回 1 。	$-5 \leq -5$ ENTER	1
	• 如果是假 ($numberA > numberB$)，返回 0 。	$(20 * 5 / 2) \leq (18 * 3)$ ENTER	1
	$number \leq list$	$1 \leq \{1, -6, 10\}$ ENTER	{1 0 1}
	返回一个包含 1 和/或 0 数组，其中每个元素表示 $number$ 是否 ≤ 数组 $list$ 中相应的元素。		
	$listA \leq listB$	$\{1, 5, 9\} \leq \{1, -6, 10\}$ ENTER	{1 0 1}
	返回一个包含 1 和/或 0 数组，其中每个元素表示数组 $listA$ 中每个元素是否 ≤ 数组 $listB$ 中相应元素。		

LgstR

STAT CALC 菜单

内置方程变量如 **y1**、**r1** 和 **xt1** 是区分大小写的。不要使用 **Y1**、**R1** 和 **XT1**。

LgstR 返回 **tolMet** 值, 它表示结果是否满足 TI-86 的内部误差。

- 如果 **tolMet=1**, 则结果在内部误差范围之内。
- 如果 **tolmet=0**, 则误差超出了内部误差, 尽管它在通常情况下可能是有用的。

LgstR [*iterations*],*xList*,*yList*,*frequencyList*,*equationVariable*

用逻辑回归模型 ($y=a/(1+be^{cx})+d$) 拟合在 *xList* 和 *yList* 中实数对, 频数为 *frequencyList*。回归方程保存在方程变量 *equationVariable*, 必须是内置方程变量, 如 **y1**、**r1** 和 **xt1**。方程系数总是以数组的形式保存在内置变量 **PRegC** 中。

迭代次数 *iterations* 是可选的。如果忽略, 默认值为 64。*iterations* 越大, 结果越精确, 但是需要更多的计算时间。迭代次数较小, 结果精度则不高, 但计算时间较短。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

LgstR [*iterations*],*xList*,*yList*,*equationVariable*

频数为 1。

LgstR [*iterations*],*xList*,*yList*,*frequencyList*

只将回归方程保存在 **RegEq** 中。

LgstR [*iterations*],*xList*,*yList*

频数为 1, 同时将回归方程保存在 **RegEq** 中。

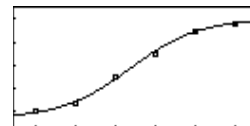
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER]
{1 2 3 4 5 6}
{1,1.3,2.5,3.5,4.5,4.8}→L2 [ENTER]
{1 1.3 2.5 3.5 4.5 4...}
LgstR L1,L2,y1 [ENTER]
```

```
LogisticReg
y=a/(1+be^(cx))+d
n=6
tolMet=1
PRegC=
{4.31285605279 51.75...
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



LgstR [*iterations*,]*equationVariable*

xList, *yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。回归方程保存在方程变量 *equationVariable* 和 **RegEq** 中。

LgstR [*iterations*]

使用 **xStat**、**yStat** 和 **fStat**，同时将回归方程只保存在 **RegEq** 中。

Line(

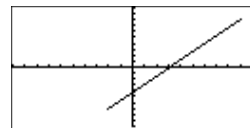
† GRAPH DRAW 菜单

Line(*x1,y1,x2,y2*)

从点 (*x1,y1*) 到点 (*x2,y2*) 绘制一条直线。

在 **Func** 图象方式下和一个 **ZStd** 图象屏幕上:

Line(-2,-7,9,8) **ENTER**



LinR

STAT CALC 菜单

内置方程变量如 **y1**、**r1** 和 **xt1** 是区分大小写的。不要使用 **Y1**、**R1** 和 **XT1**。

LinR *xList*, *yList*, *frequencyList*, *equationVariable*

用线性回归模型 ($y=a+bx$) 拟合在 *xList* 和 *yList* (*y* 值必须 > 0) 中实数对, 频数为 *frequencyList*。回归方程保存在方程变量 *equationVariable* 中, 必须是一个内置方程变量, 如 **y1**、**r1** 和 **xt1**。

xList, *yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

LinR *xList*, *yList*, *equationVariable*

频数为 1。

LinR *xList*, *yList*, *frequencyList*

只将回归方程保存在 **RegEq** 中。

LinR *xList*, *yList*

频数为 1, 只将回归方程保存在 **RegEq** 中。

LinR *equationVariable*

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。回归方程保存在方程变量 *equationVariable* 和 **RegEq** 中。

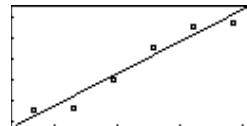
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER]
                                {1 2 3 4 5 6}
{4.5,4.6,6,7.5,8.5,8.7}→L2 [ENTER]
                                {4.5 4.6 6 7.5 8.5 8.7}
LinR L1,L2,y1 [ENTER]
```

```
LinReg
y=a+bx
a=3.213333333
b=.977142857
corr=.97454752
n=6
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



	LinR 使用 xStat、yStat 和 fStat，只将回归方程保存在 RegEq 中。	
数组输入: {} LIST 菜单	{element1,element2, ...} 定义一个数组，其中每个元素可以是实数或复数或变量。	{1,2,3}→L1 [ENTER] {1 2 3} 在 RectC 复数方式下: {3,(2,4),8*2}→L2 [ENTER] {{(3,0) (2,4) (16,0)}
li▼vc LIST OPS 菜单 VECTR OPS 菜单	li▼vc list 返回一个向量，是由实数或复数数组 <i>list</i> 转换得到的。	li▼vc {2,7,-8,0} [ENTER] [2 7 -8 0]
ln [LN]	ln number 或 ln (expression) 返回一个实数或复数 <i>number</i> 或表达式 <i>expression</i> 的自然对数。 ln list 返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的自然对数。	ln 2 [ENTER] .69314718056 ln (36.4/3) [ENTER] 2.49595648597 在 RectC 复数方式下: ln -3 [ENTER] (1.09861228867,3.141... ln {2,3} [ENTER] {.69314718056 1.0986...
lngh STRNG 菜单	lngh string 返回字符串 <i>string</i> 的长度（字符个数）。字符数包括空格但不包括引号。	lngh "The answer is:" [ENTER] 14 "The answer is:"→STR [ENTER] The answer is: lngh STR [ENTER] 14

LnR

STAT CALC 菜单

内置方程变量如 **y1**、**r1** 和 **xt1** 是区分大小写的。不要使用 **Y1**、**R1** 和 **XT1**。

LnR *xList*, *yList*, *frequencyList*, *equationVariable*

用对数回归模型 ($y=a+b \ln x$) 拟合在 *xList* 和 *yList* (*x* 值必须 > 0) 中的实数对, 频数为 *frequencyList*。回归方程保存在 *equationVariable* 中, 必须是一个内置方程变量, 例如, **y1**、**r1** 和 **xt1**。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

LnR *xList*, *yList*, *equationVariable*

频数为 1。

LnR *xList*, *yList*, *frequencyList*

只将回归方程保存在 **RegEq** 中。

LnR *xList*, *yList*

频数为 1, 只将回归方程保存在 **RegEq** 中。

LnR *equationVariable*

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

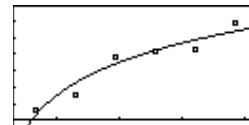
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER]
                        {1 2 3 4 5 6}
{.6,1.5,3.8,4.2,4.3,5.9}→L2 [ENTER]
                        {.6 1.5 3.8 4.2 4.3 5.9}
LnR L1,L2,y1 [ENTER]
```

```
LnReg
y=a+blnx
a=.252233501
b=2.85543117
corr=.962862433
n=6
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



LnR

使用 **xStat**、**yStat** 和 **fStat**，只将回归方程保存在 **RegEq** 中。

log

LOG

log *number* 或 **log** (*expression*)

返回一个实数或复数 *number* 或表达式 *expression* 的对数，其中：

$10^{\text{logarithm}} = \text{number}$

log 2 **ENTER** .301029995664

log (36.4/3) **ENTER** 1.08398012893

在 **RectC** 复数方式下：

log (3,4) **ENTER**
(.698970004336,.4027...

log *list*

返回一个数组，其中每个元素是数组 *list* 中相应元素的对数。

在 **RectC** 复数方式下：

log {-3,2} **ENTER**
{(.47712125472,1.364...

LU(

MATRIX MATH 菜单

LU(*matrix*,*lMatrixName*,*uMatrixName*,*pMatrixName*)

计算一个实数或复数矩阵 *matrix* 的 Crout LU（下三角—上三角）分解。下三角矩阵保存在 *lMatrixName* 中，上三角矩阵保存在 *uMatrixName* 中，置换矩阵（描述计算时所进行的行交换）保存在 *pMatrixName* 中。

lMatrixName * *uMatrixName* = *pMatrixName* * *matrix*

[[6,12,18][5,14,31][3,8,18]]
➔MAT **ENTER** [[6 12 18]

[5 14 31]

[3 8 18]]

LU(MAT,L,U,P) **ENTER** Done

L **ENTER** [[6 0 0]

[5 4 0]

[3 2 1]]

U **ENTER** [[1 2 3]

[0 1 4]

[0 0 1]]

P **ENTER** [[1 0 0]

[0 1 0]

[0 0 1]]

矩阵输入: []

[2nd] [I] 和 [2nd] [I]

[[row1][row2] ...]

定义一个矩阵, 按行输入。其中每个元素是实数或复数或变量。

输入每行 [row] 的格式为: [element,element, ...]。

[[1,2,3][4,5,6]]>MAT [ENTER]

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

max(

MATH NUM 菜单

max(numberA,numberB)

返回两个实数或复数之间较大的数。

max(2.3,1.4) [ENTER]

2.3

max(list)

返回数组 *list* 中的最大元素。

max({1,9, $\pi/2$,e^2}) [ENTER]

9

max(listA,listB)

返回一个数组, 其中每个元素是数组 *listA* 和数组 *listB* 中相应两个元素中较大的一个。

max({1,10},{2,9}) [ENTER]

{2 10}

MBox

† STAT DRAW 菜单

MBox *xList*,*frequencyList*

用 *xList* 中的实数和 *frequencyList* 中的频数, 在当前的图象上绘制改进箱线图。

从一个 **ZStd** 图象屏幕开始:

{1,2,3,4,5,9}>XL [ENTER]

{1 2 3 4 5 9}

{1,1,1,4,1,1}>FL [ENTER]

{1 1 1 4 1 1}

0>xMin:0>yMin [ENTER]

0

MBox XL,FL [ENTER]

MBox *xList*

频数为 1。

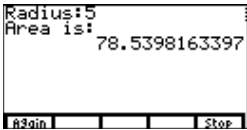
MBox

使用内置变量 **xStat** 和 **fStat** 中的数据。这些变量必须包含相同维数的有效数据, 否则会产生错误。



Menu(

‡ 程序编辑器
CTL 菜单

Menu <i>(item#,"title1",label1[,... ,item#,"title15",label15])</i>	程序段
执行程序时生成一个至多有 15 个项的菜单。菜单按照每组五项，共三组进行显示。对于每个项：	<pre>: :Lb1 A :Input "Radius:",RADIUS :Disp "Area is:",π*RADIUS² :Menu(1,"Again",A,5,"Stop",B) :Lb1 B :Disp "The End"</pre>
• <i>item#</i> — 是一个 1 到 15 之间的整数，表示项在菜单中的位置。 • <i>"title"</i> — 在菜单上显示的项的文本串。一般使用 1 到 5 个字符，多余的字符在菜单上看不见。 • <i>label</i> — 用户选择该项后，程序执行时跳转到该有效的标记。	执行后的示例： 

min(

MATH NUM 菜单

min <i>(numberA,numberB)</i>	<code>min(3,-5)</code> <code>[ENTER]</code> -5
返回两个实数或复数之间较小的一个。	<code>min(-5.2,-5.3)</code> <code>[ENTER]</code> -5.3
	<code>min(5,2+2)</code> <code>[ENTER]</code> 4
min <i>(list)</i>	<code>min({1,3,-5})</code> <code>[ENTER]</code> -5
返回数组 <i>list</i> 中的最小元素。	
min <i>(listA,listB)</i>	<code>min({1,2,3},{3,2,1})</code> <code>[ENTER]</code> {1 2 1}
返回一个数组，其中每个元素是数组 <i>listA</i> 和数组 <i>listB</i> 中相应两个元素中较小的一个。	

mod(

MATH NUM 菜单

mod <i>(numberA,numberB)</i>	<code>mod(7,0)</code> <code>[ENTER]</code> 7
返回 <i>numberA</i> 对 <i>numberB</i> 模数。参数必须是实数。	<code>mod(7,3)</code> <code>[ENTER]</code> 1
	<code>mod(-7,3)</code> <code>[ENTER]</code> 2
	<code>mod(7,-3)</code> <code>[ENTER]</code> -2
	<code>mod(-7,-3)</code> <code>[ENTER]</code> -1

mRAdd(

MATRIX OPS 菜单

mRAdd (<i>number</i> , <i>matrix</i> , <i>rowA</i> , <i>rowB</i>)	<code>[[5,3,1][2,0,4][3,-1,2]]>MAT</code> <code>[ENTER]</code>	<code>[[5 3 1] [2 0 4] [3 -1 2]]</code>
返回矩阵操作“乘然后加行”的结果，其中：		
a. 实数或复数矩阵 <i>matrix</i> 的行 <i>rowA</i> 乘以实数或复数 <i>number</i> 。	<code>mRAdd(5,MAT,2,3)</code> <code>[ENTER]</code>	<code>[[5 3 1] [2 0 4] [13 -1 22]]</code>
b. 结果加到行 <i>rowB</i> （并保存在行 <i>rowB</i> ）。		

Multiplication: *



<i>numberA</i> * <i>numberB</i>	<code>2*5</code> <code>[ENTER]</code>	<code>10</code>
返回两个实数或复数的乘积。		
<i>number</i> * <i>list</i> 或 <i>list</i> * <i>number</i> <i>number</i> * <i>matrix</i> 或 <i>matrix</i> * <i>number</i> <i>number</i> * <i>vector</i> 或 <i>vector</i> * <i>number</i>	<code>4*{10,9,8}</code> <code>[ENTER]</code> 在 RectC 复数方式下： <code>[8,1,(5,2)]*3</code> <code>[ENTER]</code>	<code>{40 36 32}</code> <code>[(24,0) (3,0) (15,6)]</code>
返回一个数组、矩阵或向量，其中每个元素是数 <i>number</i> 与数组 <i>list</i> 、矩阵 <i>matrix</i> 或向量 <i>vector</i> 中相应元素的乘积。		
<i>listA</i> * <i>listB</i>	<code>{1,2,3}*{4,5,6}</code> <code>[ENTER]</code>	<code>{4 10 18}</code>
返回一个数组，其中每个元素是数组 <i>listA</i> 的每个元素与数组 <i>listB</i> 相应元素的乘积。两个数组必须有相同的维数。		
<i>matrix</i> * <i>vector</i>	<code>[[1,2,3][4,5,6]]>MAT</code> <code>[ENTER]</code> <code>MAT*[7,8,9]</code> <code>[ENTER]</code>	<code>[1 2 3] [4 5 6] [50 122]</code>
返回一个向量，其中矩阵 <i>matrix</i> 乘以向量 <i>vector</i> 。矩阵 <i>matrix</i> 的列数必须等于向量 <i>vector</i> 中元素个数。		

$matrixA * matrixB$		[[2,2][3,4]]→MAT A ENTER	[[2 2] [3 4]]
返回一个矩阵，其中矩阵 $matrixA$ 乘以矩阵 $matrixB$ 。 矩阵 $matrixA$ 的列数必须等于矩阵 $matrixB$ 中的行数。		[[1,2,3][4,5,6]]→MAT B ENTER	[[1 2 3] [4 5 6]]
		MAT A*MAT B ENTER	[[10 14 18] [19 26 33]]
multR(			
MATRX OPS 菜单		multR(number,matrix,row)	[[5,3,1][2,0,4][3,-1,2]]→MAT ENTER
返回矩阵操作“行乘”的结果，其中：			[[5 3 1] [2 0 4] [3 -1 2]]
a. 实数或复数矩阵 $matrix$ 的指定行 row 乘以实数或复数 $number$ 。		multR(5,MAT,2) ENTER	[[5 3 1] [10 0 20] [3 -1 2]]
b. 结果保存在同一行 row 中。			
nCr			
MATH PROB 菜单		$items$ nCr $number$	5 nCr 2 ENTER 10
返回每次从 $items(n)$ 中取出 $number(r)$ 的组合数。参数必须都是非负整数。			

nDer(CALC 菜单 要查看或设置 δ 的值，则按 [2nd] [MEM] [F4] 显示误差屏幕。	nDer(<i>expression,variable,value</i>) 返回表达式 <i>expression</i> 在变量 <i>variable</i> 的值为实数或复数 <i>value</i> 时的近似数值导数。近似数值导数是通过下述点的割线的斜率： $(value-\delta,f(value-\delta))$ 和 $(value+\delta,f(value+\delta))$ 步长值 δ 越小，近似值越精确。 nDer(<i>expression,variable</i>) 使用变量 <i>variable</i> 的当前值。	对于 $\delta=.001$: nDer($x^3,x,5$) [ENTER] 75.000001 对于 $\delta=1E-4$: nDer($x^3,x,5$) [ENTER] 75 5$\rightarrow$x [ENTER] 5 nDer(x^3,x) [ENTER] 75
取负: - [+/-]	- <i>number</i> 或 -(<i>expression</i>) - <i>list</i> - <i>matrix</i> - <i>vector</i> 返回实数或复数参数的负数。	-2+5 [ENTER] 3 -(2+5) [ENTER] -7 -(0,-5,5) [ENTER] {0 5 -5}
norm MATRIX MATH 菜单 VECTR MATH 菜单	norm <i>matrix</i> 返回实数或复数矩阵 <i>matrix</i> 的 Frobenius 范数，计算公式如下： $\sqrt{\Sigma(real^2+imaginary^2)}$ 其中求和是对所有元素。	[[1,-2][-3,4]]$\rightarrow$MAT [ENTER] [[1 -2] [-3 4]] norm MAT [ENTER] 5.47722557505

norm <i>vector</i>	norm [3,4,5] ENTER	7.07106781187
返回实数或复数向量 <i>vector</i> 的长度，其中： norm [a,b,c] 返回 $\sqrt{a^2+b^2+c^2}$ 。		
norm <i>number</i> 或 norm (<i>expression</i>)	norm -25 ENTER	25
norm <i>list</i>	在 Radian 角度方式下： norm {-25,cos -(π /3)} ENTER	{25 .5}
Normal	在 Eng 记数法方式下：	
† 方式屏幕	123456789 ENTER	123.456789E6
	在 Sci 记数法方式下：	
	123456789 ENTER	1.23456789E8
	在 Normal 记数法方式下：	
	123456789 ENTER	123456789

not

BASE BOOL 菜单

not integer

返回一个实整数 *integer* 的反码。在计算器内部，整数 *integer* 用一个十六位的二进制数表示。求反码时翻转每位的值（0 变为 1，反之亦然）。

例如: **not 78**:

78 = 0000000001001110b
 111111110110001b （反码）

└ 符号位: 1 表示一个负数

要求负的二进制数的模，则确定它的补码（先计算反码，然后加 1）。例如：

111111110110001b = 78 的反码
 0000000001001110b （反码）
 + 0000000000000001b
 0000000001001111b = 79 （补码）

因此， **not 78** = -79

可以输入实数来代替整数，但是在比较前进行自动截断。

在 **Dec** 记数法方式下：

not 78 **ENTER** -79

在 **Bin** 记数法方式下：

not 1001110 **ENTER** 111111110110001b

Ans▶Dec **ENTER** -79d

不等于: $\neq$ TEST 菜单	$numberA \neq numberB$	$2+2\neq3+2$ ENTER	1
	$matrixA \neq matrixB$	$2+(2\neq3)+2$ ENTER	5
	$vectorA \neq vectorB$	$[1,2]\neq[3-2,-1+3]$ ENTER	0
	$stringA \neq stringB$	$"A" \neq "a"$ ENTER	1
测试条件 $argumentA \neq argumentB$ 是真还是假。数字、矩阵和向量可以是实数,也可以是复数。如果是复数,则比较各元素的模。字符串是区分大小写的。			
• 如果为真 ($argumentA \neq argumentB$), 返回 1。 • 如果为假 ($argumentA = argumentB$), 返回 0。			
	$listA \neq listB$	$\{1,5,9\}\neq\{1,-6,9\}$ ENTER	{ 0 1 0 }
返回一个包含 1 和/或 0 数组, 其中每个元素表示数组 $listA$ 中的每个元素是否 $\neq$ 数组 $listB$ 中相应的元素。			
nPr MATH PROB 菜单	$items\ nPr\ number$	$5\ nPr\ 2$ ENTER	20
	返回每次从 $items$ (n) 中取出 $number$ (r) 的排列数。 两个参数必须都是非负整数。		
O BASE TYPE 菜单	$integer\ o$	在 Dec 数制方式下:	
	无论为何种数制方式设置, 都将实数 $integer$ 表示为八进制数。	$10o$ ENTER	8
		$10o+10$ ENTER	18

Oct

† 方式屏幕

Oct

设置八进制数制方式。结果显示有 **b** 后缀。在任何数制方式下，从 **BASE TYPE** 菜单都可以分别用 **b**、**d**、**h** 或 **o** 指示符将相应的数值表示为二进制、十进制、十六进制或八进制数。

在 **Oct** 数制方式下：

10+10b+Fh+10d ENTER 43o

►Oct

BASE CONV 菜单

number►Oct

list►Oct

matrix►Oct

vector►Oct

返回实数或复数参数的等价八进制数。

在 **Dec** 数制方式下：

2*8 ENTER 16

Ans►Oct ENTER 20o

{7,8,9,10}►Oct ENTER {7o 10o 11o 12o}

OneVar

STAT CALC 菜单
(OneVa 显示在菜单上)

OneVar *xList*,*frequencyList*

执行一元统计分析，使用 *xList* 中的实数点和 *frequencyList* 中的频数。

xList 和 *frequencyList* 的值分别自动保存在内置变量 **xStat** 和 **fStat** 中。

OneVar *xList*

频数为 1。

{0,1,2,3,4,5,6}►XL ENTER {0 1 2 3 4 5 6}

OneVar XL ENTER

```
1-Var Stats
x=3
Σx=21
Σx²=91
Sx=2.1602469
σx=2
↓n=7
```

向下滚动可查看更多结果。

OneVar

xList 和 *frequencyList* 分别使用 **xStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据；否则会产生错误。

or
BASE BOOL 菜单

integerA or integerB

将两个整数按位比较。在内部，两个整数转换为二进制数。相应的两位比较时，如果两位中有任意一位为 1，则结果为 1；如果两位均为 0 时，则结果为 0。返回值为位比较结果的和。

例如：78 **or** 23 = 95

$$\begin{array}{r} 78 = 1001110b \\ 23 = 0010111b \\ \hline 1011111b = 95 \end{array}$$

可以输入实数来代替整数，但是在比较前它们被自动截断。

在 **Dec** 数制方式下：
78 or 23 ENTER 95

在 **Bin** 数制方式下：
1001110 or 10111 ENTER 1011111b
Ans▶Dec ENTER 95d

Output(

‡ 程序编辑器
I/O 菜单

Output(*row,column,string*)

在行 *row* 和列 *column* 处显示字符串, 其中 $1 \leq row \leq 8$
and $1 \leq column \leq 21$ 。

Output(*row,column,value*)

在指定的行 *row* 和列 *column* 处显示 *value* 。

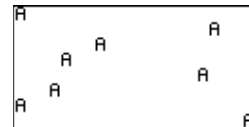
Output("CBLSEND",*listName*)

将 *listName* 中的内容发送到 CBL 或 CBR 系统。
也可以使用第 350 页介绍的 **Send**(来发送数据。

程序段:

```
:  
:CLLCD  
:For(i,1,8)  
:  Output(i,randInt(1,21),"A")  
:End  
:
```

执行后结果:



P2Reg

STAT CALC 菜单

内置方程的变量, 例如 **y1**, **r1** 和 **xt1** 是区分大小写的。不能使用 **Y1**、**R1** 和 **XT1**。

P2Reg *xList,yList,frequencyList,equationVariable*

使用 *xList* 和 *yList* 中的实数对和 *frequencyList* 中的频数进行二次多项式回归。回归方程保存在 *equationVariable* 中, 必须是一个内置方程变量, 例如, **y1**、**r1** 和 **xt1**。方程的系数总是以数组的形式保存在内置变量 **PRegC** 中。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

P2Reg *xList,yList,equationVariable*

频数为 1。

P2Reg *xList,yList,frequencyList*

只将回归方程保存在 **RegEq** 中。

P2Reg *xList,yList*

频数为 1, 只将回归方程保存在 **RegEq** 中。

P2Reg *equationVariable*

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

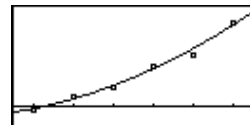
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER]
{1 2 3 4 5 6}
{-2,6,11,23,29,47}→L2 [ENTER]
{-2 6 11 23 29 47}
P2Reg L1,L2,y1 [ENTER]
```

```
QuadraticReg
y=aX^2+bX+c
n=6
PRegC=
{.964285714286 2.564...
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



P2Reg

使用 **xStat**、**yStat** 和 **fStat**，只将回归方程保存在 **RegEq** 中。

P3Reg

STAT CALC 菜单

内置方程变量,例如 **y1**，**r1** 和 **xt1** 是区分大小写的。不能使用 **Y1**，**R1** 和 **XT1**。

P3Reg *xList,yList,frequencyList,equationVariable*

使用 *xList* 和 *yList* 中的实数对和 *frequencyList* 中的频数进行三次多项式回归。回归方程保存在 *equationVariable* 中，必须是一个内置方程变量，例如，**y1**、**r1** 和 **xt1**。方程的系数总是以数组的形式保存在内置变量 **PRegC** 中。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

P3Reg *xList,yList,equationVariable*

频数为 1。

P3Reg *xList,yList,frequencyList*

只将回归方程保存在 **RegEq** 中。

P3Reg *xList,yList*

频数为 1，只将回归方程保存在 **RegEq** 中。

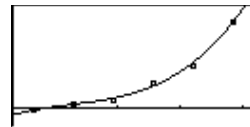
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER] {1 2 3 4 5 6}
{-6,15,27,88,145,294}→L2 [ENTER] {-6 15 27 88 145 294}
P3Reg L1,L2,y1 [ENTER]
```

```
CubicReg
y=ax³+bx²+cx+d
n=6
PRegC=
{3.2037037037 -18.99...
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



P3Reg *equationVariable*

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据;否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

P3Reg

使用 **xStat**、**yStat** 和 **fStat**，只将回归方程保存在 **RegEq** 中。

P4Reg

STAT CALC 菜单

内置方程变量, 例如, **y1**,
r1 和 **xt1** 是区分大小写的。
不能使用 **Y1**, **R1** 和 **XT1**。

P4Reg *xList,yList,frequencyList,equationVariable*

使用 *xList* 和 *yList* 中的实数对和 *frequencyList* 中的频数进行四次多项式回归。回归方程保存在 *equationVariable* 中, 必须是一个内置方程变量, 例如, **y1**、**r1** 和 **xt1**。方程的系数总是以数组的形式保存在内置变量 **PRegC** 中。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

P4Reg *xList,yList,equationVariable*

频数为 1。

P4Reg *xList,yList,frequencyList*

只将回归方程保存在 **RegEq** 中。

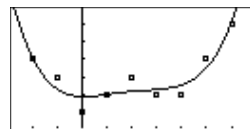
在 **Func** 图象方式下:

```
{-2,-1,0,1,2,3,4,5,6}→L1 [ENTER]
{-2 -1 0 1 2 3 4 5 6}
{4,3,1,2,3,2,2,4,6}→L2 [ENTER]
{4 3 1 2 3 2 2 4 6}
P4Reg L1,L2,y1 [ENTER]
```

```
QuarticReg
y=ax^4+bx^3+cx^2+dx+e
n=9
PRegC=
{.014568764569 -.109...
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



P4Reg *xList,yList*

频数为 1，同时只将回归方程保存在 **RegEq** 中。

P4Reg *equationVariable*

xList、*yList* 和 *frequencyList* 使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据；否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

P4Reg

使用 **xStat**、**yStat** 和 **fStat**，只将回归方程保存在 **RegEq** 中。

Param

† 方式屏幕

Param

设置参数图象方式。

Pause

‡ 程序编辑器
CTL 菜单

Pause *string*

Pause *value*

Pause *list*

Pause *matrix*

Pause *vector*

显示指定的参数，然后挂起程序直到用户按 **ENTER**。

程序段：

```
⋮  
:Input "Enter x:",x  
:y1=x2-6  
:Disp "y1 is:",y1  
:Pause "Press ENTER to graph"  
:ZStd  
⋮
```

Pause 将程序挂起，直到用户按 ENTER 。			
百分数：% MATH MISC 菜单	number% 或 (expression)%	5% ENTER	.05
	返回实数 <i>number</i> 或表达式 <i>expression</i> 被 100 除的结果。	5%*200 ENTER	10
		(10+5)%*200 ENTER	30
pEval MATH MISC 菜单	pEval(coefficientList,xValue)	计算多项式 $y=2x^2+2x+3$ 在 $x=5$ 的值:	
	返回多项式（系数在 <i>coefficientList</i> 中给出）在变量值为 <i>xValue</i> 时的值。	pEval({{2,2,3}},5) ENTER	63
PIOff STAT PLOT 菜单	PIOff [1,2,3]	P1Off 1,3 ENTER	Done
	取消选择指定的统计图序号。		
	PIOff	P1Off ENTER	Done
	取消选择所有统计图序号。		
PIOn STAT PLOT 菜单	PIOn [1,2,3]	P1On 2,3 ENTER	Done
	除了已选的统计图序号，选择指定的统计图序号。		
	PIOn	P1On ENTER	Done
	选择所有的统计图序号。		

Plot1(Plot2(Plot3(

† STAT PLOT 菜单

右侧关于 **Plot1(** 的语法和描述同样适用于 **Plot2(** 和 **Plot3(**。

散点图 

Plot1(1,xListName,yListName,mark)

Plot1(1,xListName,yListName)

用 *xListName* 和 *yListName* 中的实数对定义和选择统计图。

可选参数 *mark* 指定用来绘制点的字符, 如果忽略 *mark*, 则使用方框。

mark: 1 = 方框 (□) 2 = 十字 (+) 3 = 点 (•)

xy 直线图 

Plot1(2,xListName,yListName,mark)

Plot1(2,xListName,yListName)

改进箱线图 

Plot1(3,xListName,1 或 frequencyListName,mark)

Plot1(3,xListName,1 或 frequencyListName)

Plot1(3,xListName)

用 *xListName* 中的实数点和指定的频数定义并选择箱线图。如果省略 1 或者 *frequencyListName*, 则频数为 1。

直方图 

Plot1(4,xListName,1 或 frequencyListName)

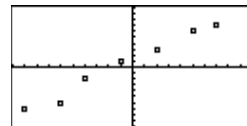
Plot1(4,xListName)

箱线图 

Plot1(5,xListName,1 或 frequencyListName)

Plot1(5,xListName)

```
{-9,-6,-4,-1,2,5,7,10}>L1 [ENTER]
      {-9 -6 -4 -1 2 5 7 1...
{-7,-6,-2,1,3,6,7,9}>L2 [ENTER]
      {-7 -6 -2 1 3 6 7 9}
Plot1(1,L1,L2) [ENTER]
ZStd [ENTER] Done
```



Pol † 方式屏幕	Pol 设置极坐标图象方式。
►Pol CPLX 菜单	<i>complexNumber</i>►Pol 无论是何种复数方式，都用极坐标形式 (<i>magnitude</i>∠<i>angle</i>) 显示复数 <i>complexNumber</i>。 <i>list</i>►Pol <i>matrix</i>►Pol <i>vector</i>►Pol 返回一个数组、矩阵或向量，其中参数的每个元素用极坐标形式显示。 在 RectC 复数方式下: √-2 ENTER (0,1.41421356237) Ans►Pol ENTER (1.41421356237∠1.570... {1,√-2} ENTER {(1,0) (0,1.141421356... Ans►Pol ENTER {(1∠0) (1.4142135623...
PolarC † 方式屏幕	PolarC 设置极坐标复数方式 (<i>magnitude</i> ∠ <i>angle</i>)。
极坐标复数: ∠ 2nd ∠	<i>magnitude</i>∠<i>angle</i> 用来以极坐标方式输入复数。根据当前的角度方式解释角度 <i>angle</i>。 在 Radian 角度方式和 PolarC 复数方式下: (1,2)+(3∠π/4) ENTER (5.16990542093∠.9226...
PolarGC † 图象格式屏幕	PolarGC 用极坐标形式显示图象坐标。

	10^{list} 返回一个数组，其中每个元素是以 10 为底，数组 <i>list</i> 中相应元素为指数的乘方。	10^{1.5,-2} ENTER {31.6227766017 .01}
prod LIST OPS 菜单 MATH MISC 菜单	prod <i>list</i> 返回数组 <i>list</i> 中所有元素的乘积。	prod {1,2,4,8} ENTER 64 prod {2,7,-8} ENTER -112
Prompt ‡ 程序编辑器 I/O 菜单 (Prompt 显示在菜单上)	Prompt <i>variableA</i>[,<i>variableB</i>,...] 提示用户给变量 <i>variableA</i> 输入值，然后给变量 <i>variableB</i> 输入值，等等。	程序段: ⋮ :Prompt A,B,C ⋮
PtChg(† GRAPH DRAW 菜单	PtChg(<i>x,y</i>) 反转图象坐标 (<i>x,y</i>) 处的点。	PtChg(-6,2)
PtOff(† GRAPH DRAW 菜单	PtOff(<i>x,y</i>) 擦除图象坐标 (<i>x,y</i>) 处的点。	PtOff(3,5)
PtOn(† GRAPH DRAW 菜单	PtOn(<i>x,y</i>) 在图象坐标 (<i>x,y</i>) 处绘制点。	PtOn(3,5)

PwrR

STAT CALC 菜单

内置方程变量, 例如, **y1**, **r1** 和 **xt1** 是区分大小写的。不能使用 **Y1**, **R1** 和 **XT1**。

PwrR *xList,yList,frequencyList,equationVariable*

用幂回归模型 ($y=ax^b$) 拟合 *xList* 和 *yList* 中的正实数对, 频数为 *frequencyList*。回归方程保存在 *equationVariable* 中, 必须是一个内置方程变量, 例如, **y1**、**r1** 和 **xt1**。

xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。回归方程也保存在内置方程的变量 **RegEq** 中。

PwrR *xList,yList,equationVariable*

频数为 1。

PwrR *xList,yList,frequencyList*

只将回归方程保存在 **RegEq** 中。

PwrR *xList,yList*

频数为 1, 只将回归方程保存在 **RegEq** 中。

PwrR *equationVariable*

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

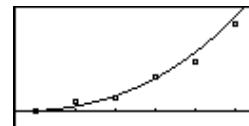
在 **Func** 图象方式下:

```
{1,2,3,4,5,6}→L1 [ENTER]
                                {1 2 3 4 5 6}
{1,17,21,52,75,133}→L2 [ENTER]
                                {1 17 21 52 75 133}
PwrR L1,L2,y1 [ENTER]
```

```
PwrReg
y=a*x^b
a=1.43992723
b=2.56896944
corr=.977662979
n=6
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



PwrR

使用 **xStat**、**yStat** 和 **fStat**，只将回归方程保存在 **RegEq** 中。

PxChg(

GRAPH DRAW 菜单

PxChg(*row,column*)

反转 (*row,column*) 处的象素，其中 $0 \leq row \leq 62$ 且 $0 \leq column \leq 126$ 。

PxChg(10,95)

PxOff(

GRAPH DRAW 菜单

PxOff(*row,column*)

擦除 (*row,column*) 处的象素，其中 $0 \leq row \leq 62$ 且 $0 \leq column \leq 126$ 。

PxOff(10,95)

PxOn(

GRAPH DRAW 菜单

PxOn(*row,column*)

在 (*row,column*) 处绘制点，其中 $0 \leq row \leq 62$ 且 $0 \leq column \leq 126$ 。

PxOn(10,95)

PxTest(

GRAPH DRAW 菜单

PxTest(*row,column*)

如果 (*row,column*) 处的象素处于开状态，则返回 **1**；否则返回 **0**。其中 $0 \leq row \leq 62$ 且 $0 \leq column \leq 126$ 。

假定在 (10,95) 的象素已处于开状态:

PxTest(10,95) **ENTER**

1

rAdd(

MATRX OPS 菜单

rAdd(*matrix,rowA,rowB*)

返回一个矩阵，该矩阵是将实数或复数矩阵 *matrix* 的行 *rowA* 与行 *rowB* 相加，然后将结果保存在行 *rowB* 中得到的。

[[5,3,1]][2,0,4][3,-1,2]]**ENTER**→MAT
[[5 3 1]
[2 0 4]
[3 -1 2]]

rAdd(MAT,2,3) **ENTER** [[5 3 1]
[2 0 4]
[5 -1 6]]

Radian

† [2nd] [MODE]

弧度输入: r

MATH ANGLE 菜单

Radian

设置角度方式为弧度。

 $number^r$ 或 $(expression)^r$ 无论为何种角度方式设置, 都将实数 $number$ 或 $expression$ 表示为弧度。 $list^r$ 将实数数组 $list$ 中的元素表示为弧度。在 **Radian** 角度方式下:
 $\sin(\pi/2)$ [ENTER] 1
 $\sin 90$ [ENTER] .893996663601
在 **Degree** 角度方式下:
 $\cos(\pi/2)$ [ENTER] .999624216859
 $\cos(\pi/2)^r$ [ENTER] 0
 $\cos\{\pi/2, \pi\}^r$ [ENTER] {0 -1}

rand

MATH PROB 菜单

rand

返回一个 0 到 1 之间的随机数。

要控制随机数序列, 首先将一个整数起始值保存在 **rand** 中 (例如 $0 \rightarrow \text{rand}$)。

前两个示例可能有不同的结果:

 rand [ENTER] .943597402492
 rand [ENTER] .146687829222

 $0 \rightarrow \text{rand}; \text{rand}$ [ENTER] .943597402492
 $0 \rightarrow \text{rand}; \text{rand}$ [ENTER] .943597402492

randBin(

MATH PROB 菜单
(randBi 显示在菜单上)**randBin(#ofTrials, probabilityOfSuccess, #ofSimulations)**

返回一个元素为随机整数的数组, 每个元素服从二项分布, 其中 $\#ofTrials \geq 1$ 且 $0 \leq \text{probabilityOfSuccess} \leq 1$ 。
 $\#ofSimulations$ 是一个 ≥ 1 的整数, 它指定数组中返回的整数个数。

保存在 **rand** 中的起始值也影响 **randBin(**。**randBin(#ofTrials, probabilityOfSuccess)**

返回单个随机整数。

 $1 \rightarrow \text{rand}; \text{randBin}(5, .2, 3)$ [ENTER] {0 3 2}
 $0 \rightarrow \text{rand}; \text{randBin}(5, .2)$ [ENTER] 1

randInt() MATH PROB 菜单 (randIn 显示在菜单上)	randInt(lower,upper,#ofTrials) 返回一个元素为随机整数的数组，其中每个元素满足： $lower \leq \text{整数} \leq upper$ 。 <i>#ofTrials</i> 是一个 ≥ 1 的整数，它指定数组中返回的整数个数。 保存在 rand 中的起始值也影响 randInt() 。	<code>1→rand:randInt(1,10,3)</code> ENTER {8 9 3}
	randInt(lower,upper) 返回单个随机整数。	<code>0→rand:randInt(1,10)</code> ENTER 10
randM() MATRX OPS 菜单	randM(rows,columns) 返回一个 $rows \times columns$ 的矩阵，其元素为一位数的随机整数（从 -9 到 9）。	<code>0→rand:randM(2,3)</code> ENTER [[4 -2 0] [-7 8 8]]
randNorm() MATH PROB 菜单 (randN 显示在菜单上)	randNorm(mean,stdDeviation,#ofTrials) 返回一个元素为随机数的数组，其元素服从由 <i>mean</i> 和 <i>stdDeviation</i> 指定的正态分布。 <i>#ofTrials</i> 是一个 ≥ 1 的整数，它指定数组中返回的元素个数。每个返回值可以是任意实数，但是大多数要落在区间： $[mean-3(stdDeviation),mean+3(stdDeviation)]$ 。 保存在 rand 中的起始值也影响 randNorm() 。	<code>1→rand:randNorm(0,1,3)</code> ENTER {-.660585055265 -1.0...
	randNorm(mean,stdDeviation) 返回单个随机数。	<code>0→rand:randNorm(0,1)</code> ENTER -1.58570962271

RcGDB

† GRAPH 菜单

RcGDB *graphDataBaseName*
恢复保存在 *graphDataBaseName* 中的所有设置。有关的设置列表请参阅第 361 页的 **StGDB**。

RcPic

† GRAPH 菜单

RcPic *pictureName*
显示当前图象，并将图形保存在图形名 *pictureName* 中。

real

CPLX 菜单

real (*complexNumber*)
返回复数 *complexNumber* 的实部。
real (*real,imaginary*) 返回 *real*。
real (*magnitude∠angle*) 返回 *magnitude*cos (angle)*。
real *complexList*
real *complexMatrix*
real *complexVector*
返回一个数组、矩阵或向量，其中每个元素是参数中相应元素的实部。

在 **Radian** 角度方式下：
real (3,4) 3
real (3∠4) -1.96093086259

在 **Radian** 角度方式下：
real {-2,(3,4),(3∠4)}
{-2 3 -1.96093086259}

►Rec

CPLX 菜单

complexNumber►**Rec**
无论是何种复数方式，都用直角坐标形式 (*real,imaginary*) 表示复数 *complexNumber*。

在 **PolarC** 复数方式下：
√-2 (1.41421356237∠1.570...
Ans►**Rec** (0,1.41421356237)


```
complexList ▶ Rec
complexMatrix ▶ Rec
complexVector ▶ Rec
```

返回一个数组、矩阵或向量，其中每个元素是参数中相应元素的直角坐标形式。

在 **PolarC** 复数方式下:

$[(3\pi/6), \sqrt{-2}]$ **[ENTER]**
 $[(3\angle .523598775598) (...]$
Ans►Rec **[ENTER]**
 $[(2.59807621135, 1.5)...$

RectC

十 方式屏幕

RectC

设置直角坐标复数方式 (*real,imaginary*)。

在 **RectC** 复数方式下:

 $\sqrt{-2}$ **ENTER** (0,1.41421356237)

RectGC

十 图象格式屏幕

RectGC

用直角坐标形式显示图象坐标。

RectV

十 式向屏幕

RectV

设置直角坐标的向量坐标方式 **[x y z]**。

在 **RectV** 向量坐标方式下:

3*[4<5] [ENTER] [3.40394622556 -11.5...

ref

MATRIX OPS 菜单

ref *matrix*

返回实数或复数矩阵 *matrix* 的行梯形矩阵。矩阵的列数必须大于或等于矩阵的行数。

```
[[4,5,6][7,8,9]]>MAT [ENTER]
[4 5 6]
[7 8 9]]
ref MAT [ENTER]
[[1 1.14285714286 1...
[0 1 2 ...
```

Repeat

- ‡ 程序编辑器
- CTL 菜单
- (Repea 显示
- 在菜单上)

:Repeat *condition*
:commands-to-repeat
:End
:commands

执行 *commands-to-repeat* 直到条件 *condition* 为真。

程序段:

```

:
:6>N
:1>Fact
:Repeat N<1
: Fact*N>Fact
: N-1>N
:End
:Disp "6!=",Fact
:
```

Return

- ‡ 程序编辑器
- CTL 菜单
- (Retur 显示
- 在菜单上)

Return

在子程序中, 退出子程序并返回到调用程序。在主程序中, 停止执行程序并返回到主屏幕。

调用程序的程序段:

```

:
:Input "Diameter:",DIAM
:Input "Height:",HT
:AREACIRC
:VOL=AREA*HT
:Disp "Volume =",VOL
:
```

AREACIRC 子程序:

```

PROGRAM:AREACIRC
:RADIUS=DIAM/2
:AREA= $\pi$ *RADIUS2
:Return
```

RK

- † 图象格式屏幕
- (向下滚动到
- 第二屏)

RK

在 **DifEq** 图象方式下, 用基于 Runge-Kutta 方法的算法求解微分方程。RK 一般比 Euler 精确, 但求解时间较长。

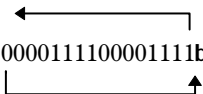
rnorm	MATRIX MATH 菜单	rnorm <i>matrix</i>	[[[-5,6,-7][3,3,9][9,-9,-7]] →MAT ENTER [[[-5 6 -7] [3 3 9] [9 -9 -7]] rnorm MAT ENTER 25 rnorm [15,-18,7] ENTER 18	
		返回实数或复数 <i>矩阵</i> 的行范数, 对于每一行, rnorm 求这行中所有元素绝对值 (复数元素的模) 的和。返回值为最大的和。		
根: $\sqrt[x]{}$	MATH MISC 菜单	rnorm <i>vector</i>		
		返回实数或复数向量 <i>vector</i> 中各元素绝对值 (或模) 的最大值。		
		$x^{th}root^x\sqrt{number}$ 或 $x^{th}root^x\sqrt{(expression)}$	$5^x\sqrt{32}$ ENTER	2
		返回 <i>number</i> 或 <i>expression</i> 的 <i>x</i> 次方根 $x^{th}root$ 。参数可以是实数或复数。		
		$x^{th}root^x\sqrt{list}$	$5^x\sqrt{\{32,243\}}$ ENTER	{ 2 3 }
		返回一个数组, 其中每个元素是数组 <i>list</i> 中相应元素的 <i>x</i> 次方根 $x^{th}root$ 。		
		$x^{th}rootList^x\sqrt{list}$	$\{5,2\}^x\sqrt{\{32,25\}}$ ENTER	{ 2 5 }
		返回一个数组, 其中每个元素是由数组 $x^{th}rootList$ 和 <i>list</i> 中相应元素指定的根。		

rotL

BASE BIT 菜单

rotL *integer*

返回实整数 *integer* 循环左移一位的结果。在内部, 整数 *integer* 用一个十六位的二进制数表示。当循环左移位时, 最左边的一位移到最右边。



rotL 0000111100001111b = 0001111000011110b

rotL 在 **Dec** 数制方式下无效。用 **BASE A-F** 菜单输入十六进制数 **A** 到 **F**。不要用 **[ALPHA]** 键入字母。

在 **Bin** 数制方式下:

rotL 0000111100001111 **[ENTER]**
1111000011110b

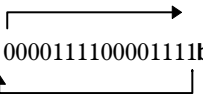
开头的 0 未显示。

rotR

BASE BIT 菜单

rotR *integer*

返回实整数 *integer* 循环右移一位的结果。在内部, 整数 *integer* 用一个十六位的二进制数表示。当循环右移位时, 最右边的一位移到最左边。



rotR 0000111100001111b = 1000011110000111b

rotR 在 **Dec** 数制方式下无效。用 **BASE A-F** 菜单输入十六进制数 **A** 到 **F**。不要用 **[ALPHA]** 键入字母。

在 **Bin** 数制方式下:

rotR 0000111100001111 **[ENTER]**
1000011110000111b

round(MATH NUM 菜单	round(<i>number</i>,#ofDecimals) round(<i>number</i>) 返回一个已四舍五入到指定位数 <i>#ofDecimals</i> （从 0 到 11）的实数或复数 <i>number</i>。如果忽略 <i>#ofDecimals</i>，数 <i>number</i> 四舍五入到小数点后第 12 位。	round(π,4) ENTER round(π/4,4) ENTER round(π/4) ENTER	3.1416 .7854 .785398163397
	round(<i>list</i>,#ofDecimals) round(<i>matrix</i>,#ofDecimals) round(<i>vector</i>,#ofDecimals) 返回一个数组、矩阵或向量，其中每个元素是参数中相应元素的四舍五入值。<i>#ofDecimals</i> 是可选的。	round($\{\pi,\sqrt{2},\ln 2\}$,3) ENTER round([[ln 5,ln 3]][π,e¹],2) ENTER	{3.142 1.414 .693} [[1.61 1.1] [3.14 2.72]]
rref MATRX OPS 菜单	rref <i>matrix</i> 返回一个实数或复数矩阵 <i>matrix</i> 的简化行梯形矩阵。矩阵的列数必须大于或等于行数。	[[4,5,6][7,8,9]]$\rightarrow$MAT ENTER rref MAT ENTER	[[4 5 6] [7 8 9]] [[1 0 -.999999999999... [0 1 2 ...
	rSwap(<i>matrix</i>,rowA,rowB) 返回一个矩阵，该矩阵是将实数或复数矩阵 <i>matrix</i> 中的行 <i>rowA</i> 与行 <i>rowB</i> 交换后得到的。	[[5,3,1][2,0,4][3,-1,2]]$\rightarrow$MAT ENTER rSwap(MAT,2,3) ENTER	[[5 3 1] [2 0 4] [3 -1 2]] [[5 3 1] [3 -1 2] [2 0 4]]

Scatter

† STAT DRAW 菜单
(Scatte 在菜单
上显示)

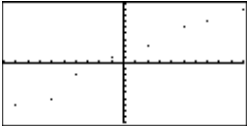
Scatter *xList,yList*

用数组 *xList* 和 *yList* 中的实数对在当前图象上绘制散
射图。

Scatter

使用内置变量 **xStat** 和 **yStat** 中数据。这些变量必须包
含相同维数的有效数据，否则会产生错误。

```
{-9,-6,-4,-1,2,5,7,10}→XL [ENTER]  
{-9 -6 -4 -1 2 5 7 1...  
{-7,-6,-2,1,3,6,7,9}→YL [ENTER]  
{-7 -6 -2 1 3 6 7 9}  
ZStd:Scatter XL,YL [ENTER]
```



Sci

† 方式屏幕

Sci

设置科学记数法显示方式。

```
在 Sci 记数法方式下：  
123456789 [ENTER] 1.23456789E8  
  
在 Normal 记数方式下：  
123456789 [ENTER] 123456789
```

Select(

LIST OPS 菜单

Select(*xListName*,*yListName*)

如果散点图和 xy 直线图是当前选择，且绘制在图象屏幕上，则可以选择那些数据点的子集（范围）。所选数据点保存在 *xListName* 和 *yListName* 中。

Select(*xListName*,*yListName*) 显示当前图象屏幕，并启动交互式会话，此时可以选择数据点的范围。

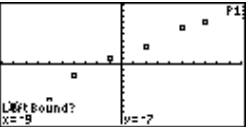
- a. 将光标移到选择范围的最左边（左界限）的点，然后按 **ENTER**。
- b. 将光标移到选择范围的最右边（右界限）的点，然后按 **ENTER**。

数组 *xListName* 和 *yListName* 的新统计图代替了原统计图，前者的点是从后者中选出来的。

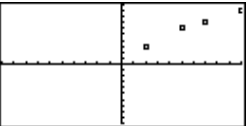
```
{-9,-6,-4,-1,2,5,7,10}→L1 ENTER  
{-9 -6 -4 -1 2 5 7 1...  
{-7,-6,-2,1,3,6,7,9}→L2 ENTER  
{-7 -6 -2 1 3 6 7 9}  
Plot1(1,L1,L2):ZStd ENTER
```

图象显示后:

```
Select(L10,L20) ENTER
```



将光标移到点 (2,3) 并按 **ENTER**。然后移到点 (10,9) 按 **ENTER**。



```
L10 ENTER {2 5 7 10}  
L20 ENTER {3 6 7 9}
```

Send(

‡ 程序编辑器
I/O 菜单

Send(*listName*)

将 *listName* 的内容发送给 CBL 或 CBR 系统。

```
{1,2,3,4,5}→L1:Send(L1) ENTER  
Done
```


Shade(

GRAPH DRAW 菜单

Shade(lowerFunc,upperFunc,xLeft,xRight,pattern,patternRes)

在当前图象上绘制 *lowerFunc* 和 *upperFunc* 关于 **x** 的图象,并阴影由 *lowerFunc*、*upperFunc*、*xLeft* 和 *xRight* 界定的区域。阴影样式由图案 *pattern* (从 **1** 到 **4**) 和图案分辨率 *patternRes* (从 **1** 到 **8**) 决定。

pattern:

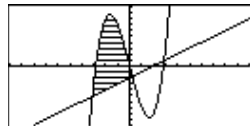
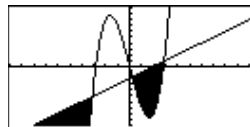
- | | |
|---------------------|----------------|
| 1 = 垂直线 (默认) | 3 = 反斜线 |
| 2 = 水平线 | 4 = 正斜线 |

patternRes (分辨率)

- | | |
|-------------------------|--------------------|
| 1 = 每 1 个像素 (默认) | 5 = 每 5 个像素 |
| 2 = 每 2 个像素 | 6 = 每 6 个像素 |
| 3 = 每 3 个像素 | 7 = 每 7 个像素 |
| 4 = 每 4 个像素 | 8 = 每 8 个像素 |

Shade(lowerFunc,upperFunc)

将 *xLeft* 和 *xRight* 分别设置到 **xMin** 和 **xMax** 中,使用 *pattern* 和 *patternRes* 的默认值。

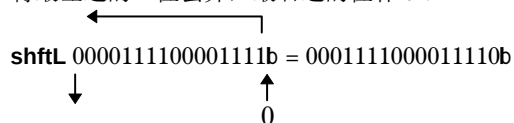
在 **Func** 图象方式下:Shade(x-2,x^3-8 x,-5,1,2,3) ENTERC1Drw:Shade(x^3-8 x,x-2) ENTER

shftL

BASE BIT 菜单

shftL *integer*

返回整数 *integer* 左移一位的结果。在内部, 整数 *integer* 表示为一个十六位的二进制数。当向左移位时, 将最左边的一位丢弃, 最右边的位补 0。



shftL 0000111100001111b = 0001111000011110b

shftL 在 Dec 数制方式下无效。用 BASE A-F 菜单输入十六进制数 A 到 F。不要用 [ALPHA] 键入字母。

在 Bin 数制方式下:

shftL 0000111100001111 [ENTER]
1111000011110b

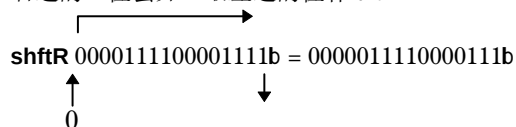
先导 0 不显示。

shftR

BASE BIT 菜单

shftR *integer*

返回整数 *integer* 右移一位的结果。整数 *integer* 在内部表示为一个十六位的二进制数。当向右移位时, 将最右边的一位丢弃, 最左边的位补 0。



shftR 0000111100001111b = 0000011110000111b

shftR 在 Dec 数制方式下无效。用 BASE A-F 菜单输入十六进制数 A 到 F。不要用 [ALPHA] 键入字母。

在 Bin 数制方式下:

shftR 0000111100001111 [ENTER]
11110000111b

先导 0 不显示。

ShwSt

CATALOG

sign

MATH NUM 菜单

ShwSt

显示最近的统计计算结果。

sign number 或 sign (expression)

sign -3.2 ENTER -1

如果参数 < 0, 返回 -1; 如果参数 > 0, 返回 1; 如果参数 = 0, 返回 0。参数必须是实数。

sign (6+2-8) ENTER 0

sign list

sign {-3.2,16.8,6+2-8} ENTER {-1 1 0}

返回一个数组, 其元素为 -1、1 或 0, 分别表示数组 list 中相应元素的符号。

SimulG

† 图象格式屏幕

SimulG

设置同时绘制格式, 同时绘制全部所选择的函数。

simult(squareMatrix,vector)

求下列方程中 x 和 y:

3x - 4y = 7

x + 6y = 6

[[3,-4][1,6]]>MAT ENTER [[3 -4]

[1 6]]

[7,6]>VEC ENTER [7 6]

simult(MAT,VEC) ENTER [3 .5]

解是 x=3 和 y=.5。

$$a_{1,1}X_1 + a_{1,2}X_2 + a_{1,3}X_3 + \dots = b_1$$
$$a_{2,1}X_1 + a_{2,2}X_2 + a_{2,3}X_3 + \dots = b_2$$
$$a_{3,1}X_1 + a_{3,2}X_2 + a_{3,3}X_3 + \dots = b_3$$

矩阵 *squareMatrix* 的行包含方程的系数 **a**, 向量包含常量 **b**。

sin

[SIN]

sin <i>angle</i> 或 sin (<i>expression</i>)	在 Radian 角度方式下：
返回 <i>angle</i> 或 <i>expression</i> 的正弦值，参数可以是实数或复数。	sin $\pi/2$ [ENTER] 0
	sin ($\pi/2$) [ENTER] 1
	sin 45° [ENTER] .707106781187
角度根据当前角度方式解释为度或弧度。在任何角度方式下，从 MATH ANGLE 菜单可以用表示符 ° 或 π 将角度分别表示为度或弧度。	在 Degree 角度方式下：
	sin 45 [ENTER] .707106781187
	sin ($\pi/2$) π [ENTER] 1
sin <i>list</i>	在 Radian 角度方式下：
返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的正弦值。	sin {0, $\pi/2,\pi$ } [ENTER] {0 1 0}
	在 Degree 角度方式下：
sin squareMatrix	sin {0,30,90} [ENTER] {0 .5 1}
返回一个方阵，该矩阵是 <i>squareMatrix</i> 的矩阵正弦值。矩阵正弦对应于用幂级数或 Cayley-Hamilton Theorem 技术的计算结果。这并非简单地计算矩阵各元素的正弦值。	

方阵不能有重复的特征值。

sin⁻¹

[2nd] [SIN⁻¹]

sin⁻¹ <i>number</i> 或 sin⁻¹ (<i>expression</i>)	在 Radian 角度方式下：
返回 <i>number</i> 或 <i>expression</i> 的反正弦值，参数可以是实数或复数。	sin ⁻¹ .5 [ENTER] .523598775598
	sin ⁻¹ {0,.5} [ENTER] {0 .523598775598}
sin⁻¹ <i>list</i>	在 Degree 角度方式下：
返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的反正弦值。	sin ⁻¹ 1 [ENTER] 90

sinh MATH HYP 菜单	sinh <i>number</i> 或 sinh (<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的双曲正弦值，参数可以是实数或复数。	sinh 1.2 ENTER	1.50946135541
	sinh <i>list</i> 返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的双曲正弦值。	sinh {0,1.2} ENTER	{0 1.50946135541}
sinh⁻¹ MATH HYP 菜单	sinh⁻¹ <i>number</i> 或 sinh⁻¹(<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的反双曲正弦值，参数可以是实数或复数。	sinh ⁻¹ 1 ENTER	.88137358702
	sinh⁻¹ <i>list</i> 返回一个数组，其中每个元素是数组 <i>list</i> 中相应元素的反双曲正弦值。	sinh ⁻¹ {1,2.1,3} ENTER	{.88137358702 1.4874...

SinR

STAT CALC 菜单

内置方程变量, 例如, **y1**, **r1** 和 **xt1** 是区分大小写的。不能用 **Y1**, **R1** 和 **XT1**。

如果指定了一个间隔, TI-86 求解的速度可能会加快, 否则在不该求得解时求得解。

SinR [*iterations*],*xList*,*yList* [,*period*],*equationVariable*

用正弦回归模型 ($y=a \sin(bx+c)+d$) 拟合 *xList* 和 *yList* 中的实数对, 使用可选的估计 *period*。回归方程保存在 *equationVariable* 中, 必须是一个内置方程变量, 例如, **y1**、**r1** 和 **xt1**。方程的系数总是以数组的形式保存在内置变量 **PRegC** 中。

iterations 是可选的; 它指定 TI-86 试图求解的最大次数 (从 1 到 16)。如果省略它, 则使用默认值 8。一般地, 此值越大, 精度越高, 但执行时间越长; 反之亦然。

如果省略可选的 *period*, 在 *xList* 中的数值是等差数列。如果指定了 *period*, 则 *x* 的值之间的差可以不相等。

xList 和 *yList* 的值分别自动保存在内置变量 **xStat** 和 **yStat** 中。回归方程也保存在内置方程变量 **RegEq** 中。

无论是何种角度方式设置, **SinR** 的输出总是为弧度。

SinR [*iterations*],*xList*,*yList* [,*period*]

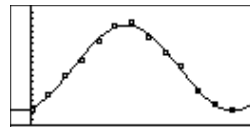
只将回归方程保存在 **RegEq** 中。

```
seq(x,x,1,361,30)→L1 [ENTER]
{1 31 61 91 121 151 ...
{5.5,8,11,13.5,16.5,19,19.5,17,
14.5,12.5,8.5,6.5,5.5}→L2 [ENTER]
{5.5 8 11 13.5 16.5...
SinR L1,L2,y1 [ENTER]
```

```
SinReg
y=a*sin(bx+c)+d
PRegC=
{6.77022677941 .0162...
```

```
Plot1(1,L1,L2) [ENTER]
ZData [ENTER]
```

Done



SlpFld

† 图象格式屏幕
（向下滚动到
第二屏）

Solver(

† **2nd** [SOLVER]

SinR [*iterations*,]*equationVariable*

xListName 和 *yListName* 分别使用 **xStat** 和 **yStat** 。这些内置变量
必须包含相同维数的有效数据；否则会产生错误。回归方程保存在 *equationVariable* 和 **RegEq** 中。

SinR [*iterations*]

使用 **xStat** 和 **yStat** ， 只将回归方程保存在 **RegEq** 中。

SlpFld

在 **DifEq** 图象方式下， 打开斜率域。用 **FldOff** 关闭方向和斜率域。

Solver(*equation,variable,guess,{lower,upper}*)

对变量 *variable* 求解方程 *equation* ， 已经给出初始估计值 *guess* 和求解界限 *lower* 和 *upper* 。方程 *equation* 可以是表达式， 并假定它等于 0 。

Solver(*equation,variable,guess*)

分别使用 **-1E99** 和 **1E99** 作为 *upper* 和 *lower* 的值。

Solver(*equation,variable,{guessLower,guessUpper}*)

使用 *guessLower* 和 *guessUpper* 之间的割线开始搜索。
Solver(也将搜索这个范围之外的解。

若 $y=5$ ， 求 $x^3+y^2=125$ 中 x 的解。估计解的近似值为 4：

```
5→y ENTER 5
Solver( $x^3+y^2=125,x,4$ ) ENTER Done
x ENTER 4.64158883361
```

sortA

LIST OPS 菜单

SortA *list*

返回一个数组，其中数组 *list* 中的实数或复数元素按升序排列。

```
{5,8,-4,0,-6}→L1 [ENTER]
SortA L1 [ENTER]
{5 8 -4 0 -6}
{-6 -4 0 5 8}
```

sortD

LIST OPS 菜单

SortD *list*

返回一个数组，其中数组 *list* 中的实数或复数元素按降序排列。

```
{5,8,-4,0,-6}→L1 [ENTER]
SortD L1 [ENTER]
{5 8 -4 0 -6}
{8 5 0 -4 -6}
```

Sortx

LIST OPS 菜单

Sortx *xListName,yListName,frequencyListName***Sortx** *xListName,yListName*

按 **x** 元素的升序，排序 *xListName*、*yListName* 和 *frequencyListName* 中的实数或复数数据对 **x** 和 **y**，以及它们的频数（可选）。更新数组的内容以反映这些变化。

```
{3,1,2}→XL [ENTER]
{0,8,-4}→YL [ENTER]
Sortx XL,YL [ENTER]
XL [ENTER]
YL [ENTER]
{3 1 2}
{0 8 -4}
Done
{1 2 3}
{8 -4 0}
```

Sortx

xList 和 *yList* 分别使用内置变量 **xStat** 和 **yStat**。这些内置变量必须包含相同维数的有效数据；否则会产生错误。

Sorty

LIST OPS 菜单

Sorty *xListName,yListName,frequencyListName***Sorty** *xListName,yListName*

按 **y** 元素的升序，排序 *xListName*、*yListName* 和 *frequencyListName* 中的实数或复数数据对 **x** 和 **y**，以及它们的频数（可选）。更新数组的内容以反映这些变化。

```
{3,1,2}→XL [ENTER]
{0,8,-4}→YL [ENTER]
Sorty XL,YL [ENTER]
YL [ENTER]
XL [ENTER]
{3 1 2}
{0 8 -4}
Done
{-4 0 8}
{2 3 1}
```


Sorty

xList 和 *yList* 分别使用内置变量 **xStat** 和 **yStat** 。这些内置变量必须包含相同维数的有效数据；否则会产生错误。

►Sph

VECTR OPS 菜单

vector►Sph

以球面坐标形式 [*r* ∠ *θ* ∠ *φ*] 或 [*r* ∠ *θ* ∠ *φ*] 显示 2 或 3 元素向量 *vector* ，即使显示方式并未设置为球面坐标 (**SphereV**)。

在 **RectV** 向量坐标方式下：

[0,-1]►Sph ENTER
[1∠-1.57079632679∠1...
[0,0,-1]►Sph ENTER
[1∠0∠3.14159265359]

SphereV

† 2nd [MODE]

SphereV

设置球面坐标的向量坐标方式 [*r* ∠ *θ* ∠ *φ*]。

在 **SphereV** 向量坐标方式下：

[1,2] ENTER
[2.2360679775∠1.1071...

平方： ²

x²

*number*² 或 (*expression*)²
*list*²
*squareMatrix*²

返回实数或复数参数与自身的乘积。要求负数的平方，则用括号将之括起来。
方阵 *squareMatrix* 与自身的乘积不是简单地计算每个元素的平方。

25² ENTER 625
(16+9)² ENTER 625
-2² ENTER -4
(-2)² ENTER 4
{-2,4,25}² ENTER {4 16 625}
[[2,3][4,5]]² ENTER [[16 21]
[28 37]]

平方根： √

2nd [√]

√*number* 或 √(*expression*)

返回 *number* 或 *expression* 的平方根，参数可以是实数，也可以是复数。

√25 ENTER 5
√(25+11) ENTER 6

StEq(

STRNG 菜单

$\sqrt{\text{list}}$

返回一个数组，其元素是数组 *list* 中相应元素的平方根。

StEq(*stringVariable*,*equationVariable*)

将 *stringVariable* 转换为一个数、表达式或方程，并将它保存在变量 *equationVariable* 中。

要转换字符串并保留相同的变量名，可以设置方程变量 *equationVariable* 等于字符串变量 *stringVariable*。

如果在此处用 **Input** 代替 **InpSt**，
则计算当前 *x* 值处的输入表达式的
值，并保存结果（不是表达式）。

在 **RectC** 复数方式下：

$\sqrt{\{-2,25\}}$ [ENTER]
{(0,1.41421356237)} (...)

"5"→x:6 x [ENTER]
ERROR 10 DATA TYPE
"5"→x:StEq(x,x):6 x [ENTER] 30

程序段：

```
⋮  
:InpSt "Enter y1(x):",STR  
:StEq(STR,y1)  
:Input "Enter x:",x  
:Disp "Result is:",y1(x)  
⋮
```

不能直接将字符串保存在嵌入方程的变量中。

StGDB

† GRAPH 菜单

StGDB *graphDataBaseName*

创建图象数据库 (GDB) 变量，包含当前的：

- 图象方式、图象格式设置和范围变量。
- 在方程编辑器中的函数，不管是否选择了它们，以及它们的图象样式。

用 **RcGDB** 保存数据库并重建图象（第 343 页）。

Stop
‡ 程序编辑器
CTL 菜单

保存到变量: →
[STO→]

StPic
† GRAPH 菜单

StReg(
STAT CALC 菜单

Stop	程序段: : :Input N :If N==999 :Stop : 使用 N==999 , 不使用 N=999 。
<i>number</i> → <i>variable</i> 或 (<i>expression</i>)→ <i>variable</i> <i>string</i> → <i>variable</i> <i>list</i> → <i>variable</i> <i>vector</i> → <i>variable</i> <i>matrix</i> → <i>variable</i> 将指定参数保存到变量 <i>variable</i> 中。	10→A:4*A [ENTER] 40 "Hello"→STR [ENTER] Hello {1,2,3}→L1 [ENTER] {1 2 3} [1,2,3]→VEC [ENTER] [1 2 3] [[1,2,3][4,5,6]]→MAT [ENTER] [[1 2 3] [4 5 6]]
StPic <i>pictureName</i> 将当前图象屏幕的图形保存到 <i>pictureName</i> 中。	
StReg (<i>variable</i>) 将最近计算的回归方程保存到变量 <i>variable</i> 中。这样通 过将回归方程保存到任意变量 (内置方程变量除外) 中 来保留回归方程。 按 [2nd] [RCL] EQ [ENTER] 调用方程。 然后按 [ENTER] 计算当前 x 处的值。	{1,2,3,4,5}→L1 [ENTER] {1 2 3 4 5} {1,20,55,230,742}→L2 [ENTER] {1 20 55 230 742} ExpR L1,L2:StReg(EQ) [ENTER] Done 8→x [ENTER] 8 Rcl EQ [ENTER] .41138948780597*4.7879605684671^x [ENTER] 113620.765451

字符串输入: "

STRNG 菜单

‡ 程序编辑器
I/O 菜单

"string"

定义一个字符串。当显示该串时，在屏幕上左对齐。
字符串被解释为文本字符，而不是数字。例如，不能用象 "4" 或 "A*8" 这样的字符串进行计算。要相互转换字符串变量与方程变量，则分别用 **Eq→St**(和 **St→Eq**((如第 290 页和第 361 页中所述)。

```
"Hello"→STR [ENTER]
Disp STR+", Jan" [ENTER]
Hello, Jan
Done
```

sub(

STRNG 菜单

sub(string,begin,length)

返回一个新的字符串，它是字符串 *string* 的子集，从字符 *begin* 处开始，长度为指定的 *length*。

```
"The answer is:"→STR [ENTER]
sub(STR,5,6) [ENTER]
The answer is:
answer
```

减法: -

□

numberA - numberB

返回 *numberA* 减去 *numberB* 的差。参数可以是实数或复数。

```
6-2 [ENTER] 4
10--4.5 [ENTER] 14.5
```

list - number

返回一个数组，其元素是数组 *list* 中相应元素减去数 *number* 的差。参数可以是实数或复数。

```
{10,9,8}-4 [ENTER] {6 5 4}
在 RectC 复数方式下:
{8,1,(5,2)}-3 [ENTER]
{(5,0) (-2,0) (2,2)}
```

	$listA - listB$ $matrixA - matrixB$ $vectorA - vectorB$ 返回一个数组、矩阵或向量，其元素是第一个参数中相应元素减去第二个参数中相应元素的差。两个实数或复数参数的维数必须相同。	$\{5,7,9\}-\{4,5,6\}$ [ENTER] $\{1\ 2\ 3\}$ $[[5,7,9][11,13,15]]-[[4,5,6][7,8,9]]$ [ENTER] $[[1\ 2\ 3][4\ 5\ 6]]$ $\{5,7,9\}-[1,2,3]$ [ENTER] $[4\ 5\ 6]$
sum MATH MISC 菜单 LIST OPS 菜单	$sum\ list$ 返回数组 $list$ 中所有实数或复数元素的和。	$sum\ \{1,2,4,8\}$ [ENTER] 15 $sum\ \{2,7,-8,0\}$ [ENTER] 1
tan [TAN]	$tan\ angle$ 或 $tan\ (expression)$ 返回 $angle$ 或 $expression$ 的正切值，参数可以是实数，也可以是复数。 根据当前角度方式将角度解释为度或弧度。在任何角度方式下，从 MATH ANGLE 菜单可以用表示符 $^{\circ}$ 或 r 分别将角度表示为度或弧度。	在 Radian 角度方式下： $tan\ \pi/4$ [ENTER] 0 $tan\ (\pi/4)$ [ENTER] 1 $tan\ 45^{\circ}$ [ENTER] 1 在 Degree 角度方式下： $tan\ 45$ [ENTER] 1 $tan\ (\pi/4)^r$ [ENTER] 1
	$tan\ list$ 返回一个数组，其元素是数组 $list$ 中相应元素的正切值。	在 Degree 角度方式下： $tan\ \{0,45,60\}$ [ENTER] $\{0\ 1\ 1.73205080757\}$

tan⁻¹ 2nd [TAN⁻¹]	tan⁻¹ <i>number</i> 或 tan⁻¹ (<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的反正切值，参数可以是实数也可以是复数。	在 Radian 角度方式下： tan⁻¹ .5 ENTER .463647609001 在 Degree 角度方式下： tan⁻¹ 1 ENTER 45 在 Radian 角度方式下： tan⁻¹ {0,.2,.5} ENTER {0 .19739555985 .463...
	tan⁻¹ <i>list</i> 返回一个数组，其元素是数组 <i>list</i> 中相应元素的反正切值。	
tanh MATH HYP 菜单	tanh <i>number</i> 或 tanh (<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的双曲正切值，参数可以是实数，也可以是复数。	tanh 1.2 ENTER .833654607012
	tanh <i>list</i> 返回一个数组，其元素是数组 <i>list</i> 中相应元素的双曲正切函数值。	tanh {0,1.2} ENTER {0 .833654607012}
tanh⁻¹ MATH HYP 菜单	tanh⁻¹ <i>number</i> 或 tanh⁻¹ (<i>expression</i>) 返回 <i>number</i> 或 <i>expression</i> 的反双曲正切值，参数可以是实数，也可以是复数。	tanh⁻¹ 0 ENTER 0
	tanh⁻¹ <i>list</i> 返回一个数组，其元素是数组 <i>list</i> 中相应元素的反双曲正切值。	在 RectC 复数方式下： tanh⁻¹ {0,2.1} ENTER {(0,0) (.51804596584...

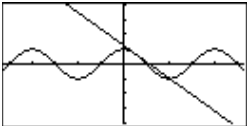
TanLn(

GRAPH DRAW 菜单

TanLn(*expression*,*xValue*)

在当前图象上绘制表达式 *expression* , 然后在 *xValue* 处绘制切线。

在 **Func** 图象方式和 **Radian** 角度方式下:
ZTrig:TanLn(cos x, $\pi/4$) **[ENTER]**



Text(

† GRAPH DRAW 菜单

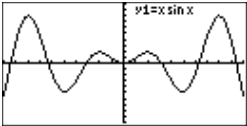
Text(*row*,*column*,*string*)

从当前图象的象素 (*row*,*column*) 处开始写文本 *string* , 其中 $0 \leq row \leq 57$ 且 $0 \leq column \leq 123$ 。
图象底部的文本可能被显示的菜单遮住。按 **[CLEAR]** 可移走菜单。

在 **Func** 图象方式和 **ZStd** 图象屏幕中的程序段:

```
⋮  
:y1=x sin x  
:Text(0,70,"y1=x sin x")  
⋮
```

执行后:



Then

‡ 程序编辑器
CTL 菜单

请参阅 **If** 的语法信息, 从第 305 页开始。请参阅 **If:Then:End** 和 **If:Then:Else:End** 的语法。

Trace

↑ GRAPH 菜单

转置： T

MATRIX MATH 菜单

Trace

显示当前图象并允许用户跟踪函数。在程序中按 **ENTER** 终止跟踪，并继续执行程序。

$matrix^T$	$[[1,2][3,4]]\rightarrow MATA$ ENTER	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
返回一个转置实数矩阵或转置复数矩阵，其中元素 row ， $column$ 是矩阵 $matrix$ 的元素 $column$ ， row 的交换。例如：	$MATA^T$ ENTER	$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T$ 返回 $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$	$[[1,2,3][4,5,6][7,8,9]]\rightarrow MATB$ ENTER	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
对于复数矩阵，取每个元素的复共轭。	$MATB^T$ ENTER	$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$
	在 RectC 复数方式下：	
	$[[(1,2), (1,1)] [(3,2), (4,3)]]$	
	$\rightarrow MATC$ ENTER	$\begin{bmatrix} (1,2) & (1,1) \\ (3,2) & (4,3) \end{bmatrix}$
	$MATC^T$ ENTER	$\begin{bmatrix} (1,-2) & (3,-2) \\ (1,-1) & (4,-3) \end{bmatrix}$

TwoVar

STAT CALC 菜单
(TwoVa 显示在菜单上)

TwoVar *xList*,*yList*,*frequencyList*

对 *xList* 和 *yList* 中的实数对执行二元统计分析, 使用 *frequencyList* 中的频数。
xList、*yList* 和 *frequencyList* 的值分别自动保存在内置变量 **xStat**、**yStat** 和 **fStat** 中。

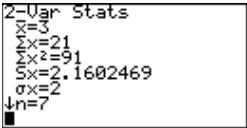
TwoVar *xList*,*yList*

频数为 1。

TwoVar

xList、*yList* 和 *frequencyList* 分别使用 **xStat**、**yStat** 和 **fStat**。这些内置变量必须包含相同维数的有效数据; 否则会产生错误。

{0,1,2,3,4,5,6}→L1
{0 1 2 3 4 5 6}
{0,1,2,3,4,5,6}→L2
{0 1 2 3 4 5 6}
TwoVar L1,L2



向下滚动可以看到更多的结果。

unitV

VECTR MATH 菜单

unitV *vector*

返回实数或复数向量 *vector* 的单位向量, 其中:

unitV [a,b,c] 返回 $\left[\frac{a}{\text{norm}} \frac{b}{\text{norm}} \frac{c}{\text{norm}} \right]$

和

norm 是 $\sqrt{a^2+b^2+c^2}$ 。

在 **RectV** 式向坐标量向下:
unitV [1,2,1]
[.408248290464 .8164...

vc►li
LIST OPS 菜单
VECTR OPS 菜单

向量输入: []
[2nd] [1] 和 [2nd] [1]

Vert
† GRAPH DRAW 菜单

While
‡ 程序编辑器
CTL 菜单

vc►li <i>vector</i> 将一个实数或复数向量 <i>vector</i> 转换为数组。	vc►li [2,7,-8,0] [ENTER] {2 7 -8 0} (vc►li [2,7,-8,0])² [ENTER] {4 49 64 0} {2 7 -8 0}
[<i>element1,element2,...</i>] 定义一个向量，其中每个元素是实数或复数或变量。	[4,5,6]►VEC [ENTER] [4 5 6] 在 PolarC 复数方式下: [5,(2∠π/4)]►VEC [ENTER] [(5∠0) (2∠.785398163...
Vert <i>xValue</i> 在当前图象的 <i>xValue</i> 处画一条垂直线。	在 ZStd 图象屏幕上: Vert -4.5 [ENTER] 
:While <i>condition</i> :commands-while-true :End :command 只要条件 <i>condition</i> 为真，执行 <i>commands-while-true</i> 。	程序段: : :1►J :0►TEMP :While J≤20 : TEMP+1/J►TEMP : J+1►J :End :Disp "Reciprocal sums to 20",TEMP : :

xor

BASE BOOL 菜单

integerA xor integerB

将两个实数按位比较。两个实数在内部被转换为二进制数。相应的两位比较时，如果其中一位为 1，则结果为 1；如果两位均为 0 或均为 1 时，结果为 0。返回值为位比较结果之和。

例如： 78 xor 23 = 89

78 = 1001110b
23 = 0010111b
1011001b = 89

可以输入实数来代替整数，但是在比较前它们被自动截断。

在 **Dec** 数制方式下：

78 xor 23 89

在 **Bin** 数制方式下：

1001110 xor 10111 1011001b

Ans $\blacktriangleright$ Dec 89d

xyline

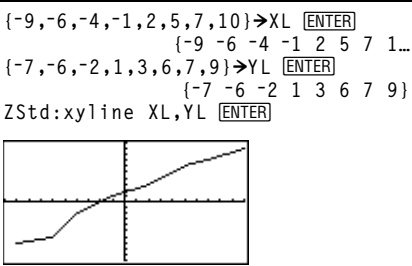
† STAT DRAW 菜单

xyline *xList,yList*

用 *xList* 和 *yList* 中的实数对在当前图象上绘制直线图。

xyline

使用内置变量 **xStat** 和 **yStat** 中的数据。这些变量必须包含相同维数的有效数据；否则会产生错误。



ZData

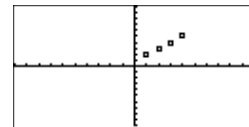
† GRAPH ZOOM 菜单

ZData

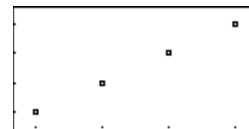
基于当前定义的统计图调整窗口变量值,使所有统计数据点都可以绘出来,然后更新图象屏幕。

在 **Func** 图象方式下:

```
{1,2,3,4}→XL  {1 2 3 4}  
{2,3,4,5}→YL  {2 3 4 5}  
Plot1(1,XL,YL)  Done  
ZStd 
```



ZData



ZDecm

† GRAPH ZOOM 菜单

ZDecm

设置窗口变量, 例如 $\Delta x = \Delta y = .1$, 然后以屏幕的原点为中心更新图象屏幕。

xMin=-6.3 yMin=-3.1

xMax=6.3 yMax=3.1

xScl=1 yScl=1

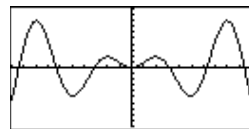
ZDecm 的一个优点是可以按 .1 的步长跟踪图象。

在 **Func** 图象方式下:

y1=x sin x **[ENTER]**

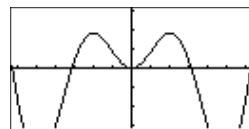
ZStd **[ENTER]**

Done



如果绘制图象上半部, x 的值从 0 开始, 并且按 .1587301587 增加。

ZDecm **[ENTER]**



如果绘制这张图象, x 的值按 .1 增加。

ZFit

† GRAPH ZOOM 菜单

ZFit

重新计算 **yMin** 和 **yMax** 以便包含所选函数在当前的 **xMin** 和 **xMax** 之间的最大值 **y** 和最小值 **y**。然后更新图象屏幕。

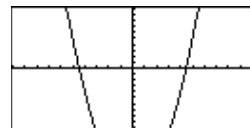
这并不影响 **xMin** 和 **xMax**。

在 **Func** 图象方式下:

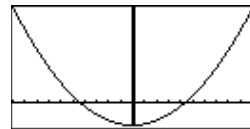
$$y1=x^2-20 \quad \text{ENTER}$$

ZStd ENTER

Done



ZFit ENTER



ZIn

† GRAPH ZOOM 菜单

ZIn

将当前光标位置周围的部分图象放大。

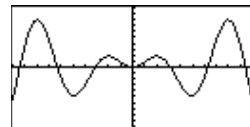
缩放因子由内置变量 **xFact** 和 **yFact** 的值来设置，它们的默认值均为 4。

在 **Func** 图象方式下:

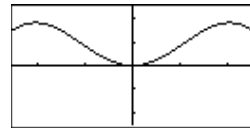
$$y1=x \sin x \quad \text{ENTER}$$

ZStd ENTER

Done



ZIn ENTER



ZInt

† GRAPH ZOOM 菜单

ZInt

设置窗口变量的值, 这样每个象素在所有方向 ($\Delta x = \Delta y = 1$) 上均为整数, 设置 **xScl=yScl=10**, 然后更新图象屏幕。

当前的光标位置成为新图象的中心。

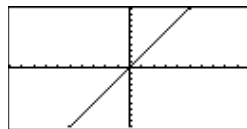
ZInt 的一个优点是可以整数步长跟踪图象。

在 **Func** 图象方式下:

y1=der1(x^2-20 ,x)

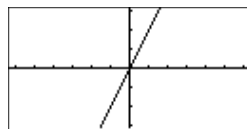
ZStd

Done



如果跟踪上方的图象, **x** 的值从 0 开始, 并且增量为 .1587301587。

ZInt



如果跟踪该图象, **x** 的值按 1 增加。

ZOut

† GRAPH ZOOM 菜单

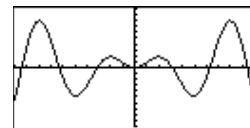
ZOut

将当前光标位置周围的部分图象缩小以显示更多的图象。
缩放因子由内置变量 **xFact** 和 **yFact** 的值来设置，它们的默认值均为 4。

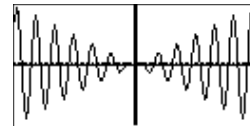
在 **Func** 图象方式下:

y1=x sin x [ENTER]
ZStd [ENTER]

Done



ZOut [ENTER]



ZPrev

† GRAPH ZOOM 菜单

ZPrev

使用在执行上一条 **ZOOM** 指令前所显示图象的窗口变量重新绘制图象。

ZRcl

† GRAPH ZOOM 菜单

ZRcl

将窗口变量的值设置为以前用户定义的缩放窗口变量中保存的值，然后更新图象屏幕。

要设置用户定义的缩放窗口变量，可用下述方法之一：

- 按 **[GRAPH]** **[F3]** **[MORE]** **[MORE]** **[MORE]** **[F1]** (**ZSTO**) 保存当前图象的窗口变量。
– 或 –
- 将应用值保存到缩放窗口变量值，变量名以 **z** 开头，后面是常通的窗口变量名。例如，将 **xMin** 的值保存到 **zxMin** 中，将 **yMin** 的值保存到 **zyMin** 中，等等。

ZSqr

† GRAPH ZOOM 菜单

ZSqr

设置窗口变量来产生“正方形”象素，其中 $\Delta x = \Delta y$ ，然后更新图象屏幕。

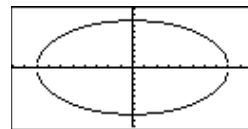
当前图象的中心（不必是坐标轴的交点）成为新图象的中心。

在其他类型的缩放中，正方形可能看起来象矩形，圆看起来象椭圆。用 **ZSqr** 可以得到更精确的形状。

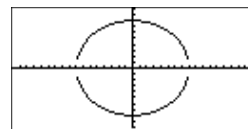
在 **Func** 图象方式下：

$y1 = \sqrt{(8^2 - x^2)}$; $y2 = -y1$ **[ENTER]**
ZStd **[ENTER]**

Done



ZSqr **[ENTER]**



ZStd

† GRAPH ZOOM 菜单

ZStd

将窗口变量设置为标准默认值, 然后更新图象屏幕。

Func 图象方式下:

xMin=-10 **yMin=-10**
xMax=10 **yMax=10**
xScl=1 **yScl=1**

Pol 图象方式下:

θMin=0 **xMin=-10** **yMin=-10**
θMax=6.28318530718 (2π) **xMax=10** **yMax=10**
θStep=.130899693899... (π/24) **xScl=1** **yScl=1**

Param 图象方式下:

tMin=0 **xMin=-10** **yMin=-10**
tMax=6.28318530718 (2π) **xMax=10** **yMax=10**
tStep=.130899693899... (π/24) **xScl=1** **yScl=1**

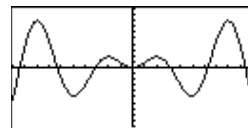
DifEq 图象方式下:

tMin=0 **xMin=-10** **yMin=-10**
tMax=6.28318530718 (2π) **xMax=10** **yMax=10**
tStep=.130899693899... (π/24) **xScl=1** **yScl=1**
tPlot=0 **diffTol=.001**

在 **Func** 图象方式下:

$y1=x \sin x$ [ENTER]
ZStd [ENTER]

Done



ZTrig

† GRAPH ZOOM 菜单

ZTrig

设置窗口变量为预置值，这些值适合在 **Radian** 角度方式 ($\Delta x=\pi/24$) 下绘制三角函数，然后更新图象屏幕。

xMin=-8.24668071567

xMax=8.24668071567

xScl=1.5707963267949 ($\pi/2$)

yMin=-4

yMax=4

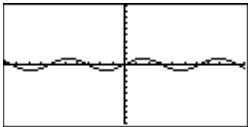
yScl=1

在 **Func** 图象方式下:

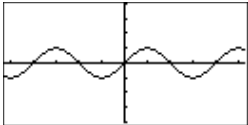
y1=sin x

ZStd

Done

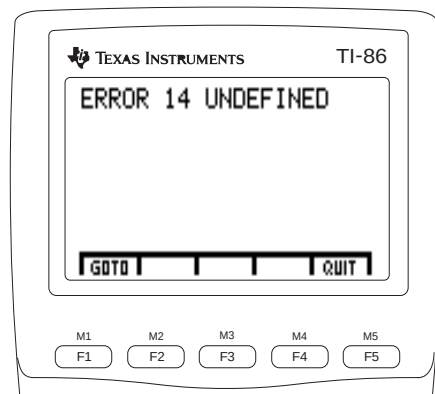


ZTrig



A 附录

TI-86 菜单图	380
疑难解答	392
错误情形	393
Equation Operating System (EOS™)	397
TOL (误差编辑器)	398
计算精度	399
有关 TI 产品、服务和保障信息	400



TI-86 菜单图

本节描述出现在 TI-86 键盘上的 TI-86 菜单，从顶部开始。如果某个菜单中包含显示其他菜单的项，其他菜单直接显示在主菜单下。在程序编辑器中，某些菜单的外观稍有改变。菜单图省去用户创建名称菜单，例如菜单 LIST NAMES 和 CONS USER。

LINK（链接）菜单 $\boxed{2\text{nd}} \boxed{\text{LINK}}$

SEND	RECV	SND85		
------	------	-------	--	--

链接菜单在程序编辑器中无效。

LINK SEND（链接发送）菜单 $\boxed{2\text{nd}} \boxed{\text{LINK}} \boxed{\text{F1}}$

BCKUP	PRGM	MATRX	GDB	ALL	▶	LIST	VECTR	REAL	CPLX	EQU	▶	CONS	PIC	WIND	STRNG	
-------	------	-------	-----	-----	---	------	-------	------	------	-----	---	------	-----	------	-------	--

SEND BCKUP（发送备份）菜单 $\boxed{2\text{nd}} \boxed{\text{LINK}} \boxed{\text{F1}} \boxed{\text{F1}}$

XMIT				
------	--	--	--	--

LINK SEND（链接发送）选择屏幕菜单 $\boxed{2\text{nd}} \boxed{\text{LINK}} \boxed{\text{F1}}$ 数据类型

XMIT	SELECT	ALL+	ALL-	
------	--------	------	------	--

LINK SND85（链接发送到 TI-85）菜单 $\boxed{2\text{nd}} \boxed{\text{LINK}} \boxed{\text{F3}}$

MATRX	LIST	VECTR	REAL	CPLX	▶	CONS	PIC	STRNG		
-------	------	-------	------	------	---	------	-----	-------	--	--

在程序编辑器中，DrEqu 是 GRAPH 菜单中的项。

GRAPH（图象）菜单 $\boxed{\text{GRAPH}}$ 在 Func 图象方式下

y(x)=	WIND	ZOOM	TRACE	GRAPH	▶	MATH	DRAW	FORMT	STGDB	RCGDB	▶	EVAL	STPIC	RCPIC		
-------	------	------	-------	-------	---	------	------	-------	-------	-------	---	------	-------	-------	--	--

GRAPH (图象) 菜单 **GRAPH** 在 **Pol** 图象方式下

$r(\theta)=$	WIND	ZOOM	TRACE	GRAPH	▸	MATH	DRAW	FORMT	STGDB	RCGDB	▸	EVAL	STPIC	RCPIC		
--------------	------	------	-------	-------	---	------	------	-------	-------	-------	---	------	-------	-------	--	--

GRAPH (图象) 菜单 **GRAPH** 在 **Param** 图象方式下

$E(t)=$	WIND	ZOOM	TRACE	GRAPH	▸	MATH	DRAW	FORMT	STGDB	RCGDB	▸	EVAL	STPIC	RCPIC		
---------	------	------	-------	-------	---	------	------	-------	-------	-------	---	------	-------	-------	--	--

GRAPH (图象) 菜单 **GRAPH** 在 **DifEq** 图象方式下

$Q'(t)=$	WIND	INITC	AXES	GRAPH	▸	FORMT	DRAW	ZOOM	TRACE	EXPLR	▸	EVAL	STGDB	RCGDB	STPIC	RCPIC
----------	------	-------	------	-------	---	-------	------	------	-------	-------	---	------	-------	-------	-------	-------

方程编辑器菜单 **GRAPH** **F1** 在 **Func** 图象方式下

$y(x)=$	WIND	ZOOM	TRACE	GRAPH												
x	y	INSf	DELf	SELECT	▸	ALL+	ALL-	STYLE								

方程编辑器菜单 **GRAPH** **F1** 在 **Pol** 图象方式下

$r(\theta)=$	WIND	ZOOM	TRACE	GRAPH												
θ	r	INSf	DELf	SELECT	▸	ALL+	ALL-	STYLE								

方程编辑器菜单 **GRAPH** **F1** 在 **Param** 图象方式下

$E(t)=$	WIND	ZOOM	TRACE	GRAPH												
t	xt	yt	DELf	SELECT	▸	INSf	ALL+	ALL-	STYLE							

方程编辑器菜单 **GRAPH** **F1** 在 **DifEq** 图象方式下

$Q'(t)=$	WIND	INITC	AXES	GRAPH												
t	Q	INSf	DELf	SELECT	▸	ALL+	ALL-	STYLE								

GRAPH VARS (图象变量) 菜单 **GRAPH** **F1** 仅用于程序编辑器

y(x)=	WIND	ZOOM	TRACE	GRAPH												
y	x	xt	yt	t	▶	r	θ	Q1	Q'1	t	▶	FnOn	FnOff	Axes	Q1	dTime
											▶	fIdRes				

GRAPH WIND (窗口变量) 菜单 **GRAPH** **F2** 仅用于程序编辑器

y(x)=	WIND	ZOOM	TRACE	GRAPH															
xMin	xMax	xScl	yMin	yMax	▶	yScl	tMin	tMax	tStep	θMin	▶	θMax	θStep	tPlot	difTol	xRes			
											▶	EStep							

GRAPH ZOOM (图象缩放) 菜单 **GRAPH** **F3**

y(x)=	WIND	ZOOM	TRACE	GRAPH												
BOX	ZIN	ZOUT	ZSTD	ZPREV	▶	ZFIT	ZSQR	ZTRIG	ZDECM	ZDATA	▶	ZRCL	ZFACT	ZOOMX	ZOOMY	ZINT
											▶	ZSTO				

要在**DifEq**方式下显示**GRAPH ZOOM**菜单, 则按**GRAPH** **MORE** **F3**。

GRAPH MATH (图象运算) 菜单 **GRAPH** **MORE** **F1** 在**Func**图象方式下

MATH	DRAW	FORMT	STGDB	RCGDB											
ROOT	dy/dx	f(x)	FMIN	FMAX	▶	INFLC	YICPT	ISECT	DIST	ARC	▶	TANLN			

在**DifEq**图象方式下无**GRAPH MATH**菜单。

GRAPH MATH (图象运算) 菜单 **GRAPH** **MORE** **F1** 在**Pol**图象方式下

MATH	DRAW	FORMT	STGDB	RCGDB					
DIST	dy/dx	dr/dθ	ARC	TANLN					

GRAPH MATH (图象运算) 菜单 **GRAPH** **MORE** **F1** 在 **Param** 图象方式下

MATH	DRAW	FORMT	STGDB	RCGDB				
DIST	dy/dx	dy/dt	dx/dt	ARC	▶	TANLN		

GRAPH DRAW (图象绘制) 菜单 **GRAPH** **MORE** **F2**

MATH	DRAW	FORMT	STGDB	RCGDB				
Shade	LINE	VERT	HORIZ	CIRCL	▶	DrawF	PEN	PTON
						PTOFF	PTCHG	▶
						CLDRW	PxOn	PxOff
						PxChg	PxTest	
						▶	TEXT	TanLn
						DrInv		

DrInv 仅在 **Func** 图象方式下有效。

DrEqu 仅在 **DifEq** 图象方式下有效。

SOLVER 菜单 **2nd** **[SOLVER]** 方程 **ENTER** **SOLVER ZOOM** 菜单 **2nd** **[SOLVER]** 方程 **ENTER** **F3**

GRAPH	WIND	ZOMM	TRACE	SOLVE				
					BOX	ZINT	ZOUT	ZFACT
					ZSTD			

TABLE 菜单 **TABLE**

TABLE	TBLST			
-------	-------	--	--	--

TABLE SETUP 菜单 **TABLE** **F2**

TABLE				
-------	--	--	--	--

运算表屏幕菜单 **TABLE** **F1**

在 **Func** 图象方式下

TBLST	SELCT	x	y	
-------	-------	---	---	--

在 **Pol** 图象方式下

TBLST	SELCT	θ	r	
-------	-------	---	---	--

在 **Param** 图象方式下

TBLST	SELCT	t	xt	yt
-------	-------	---	----	----

在 **DifEq** 图象方式下

TBLST	SELCT	t	Q	
-------	-------	---	---	--

SIMULT ENTRY* 菜单 **2nd** **[SIMULT]** (整数 ≥ 2 且 ≤ 30) **ENTER**

PREV	NEXT	CLRq		SOLVE
------	------	------	--	-------

SIMULT RESULT 菜单 **F5**

COEFS	STOa	STOb	STOx	
-------	------	------	------	--

PRGM (编程) 菜单 [PRGM]

NAMES	EDIT			
-------	------	--	--	--

程序编辑器菜单 [PRGM] [F2] 程序名 [ENTER]

PAGE↓	PAGE↑	I/O	CTL	INSc	▶	DELc	UNDEL	:		
-------	-------	-----	-----	------	---	------	-------	---	--	--

PRGM I/O (输入/输出) 菜单 [PRGM] [F2] 程序名 [ENTER] [F3]

PAGE↓	PAGE↑	I/O	CTL	INSc												
Input	Prompt	Disp	DispG	DispT	▶	CITbl	Get	Send	getKy	CILCD	▶	"	Outpt	InpSt		

PRGM CTL (控制) 菜单 [PRGM] [F2] 程序名 [ENTER] [F4]

PAGE↓	PAGE↑	I/O	CTL	INSc												
If	Then	Else	For	End	▶	While	Repea	Menu	Lbl	Goto	▶	IS>	DS<	Pause	Retur	Stop
▶ DelVa GrStl LCust																

POLY ENTRY 菜单 [2nd] [POLY] (整数 ≥ 2 且 ≤ 30) [ENTER]

CLRq				SOLVE
------	--	--	--	-------

POLY RESULT 菜单 [F5]

COEFS	STOa			
-------	------	--	--	--

CUSTOM 菜单 [CUSTOM]

					▶						▶					
--	--	--	--	--	---	--	--	--	--	--	---	--	--	--	--	--

使用CUSTOM 菜单创建用户自己的菜单 (第2章)。

CATLG-VARS (目录 — 变量) 菜单 [2nd] [CATLG-VARS]

CATLG	ALL	REAL	CPLX	LIST	▶	VECTR	MATRX	STRNG	EQU	CONS	▶	PRGM	GDB	PIC	STAT	WIND
-------	-----	------	------	------	---	-------	-------	-------	-----	------	---	------	-----	-----	------	------

CATLG-VARS (目录 — 变量) 选择菜单 [2nd] [CATLG-VARS] [F1] 或选择数据类型

PAGE↓	PAGE↑	CUSTM	BLANK	
-------	-------	-------	-------	--

CALC 菜单 [2nd] [CALC]

evalF	nDer	der1	der2	fnInt	▶	fMin	fMax	arc		
-------	------	------	------	-------	---	------	------	-----	--	--

MATRIX (矩阵) 菜单 [2nd] [MATRIX] 矩阵编辑器菜单 [2nd] [MATRIX] [F2] 矩阵名 [ENTER]

NAMES	EDIT	MATH	OPS	CPLX		INSr	DELr	INSc	DELc	▶REAL
-------	------	------	-----	------	--	------	------	------	------	-------

MATRIX MATH (矩阵数学运算) 菜单 [2nd] [MATRIX] [F3]

NAMES	EDIT	MATH	OPS	CPLX						
det	τ	norm	eigVl	eigVc	▶	rnorm	cnorm	LU	cond	

MATRIX OPS (矩阵运算) 菜单 [2nd] [MATRIX] [F4]

NAMES	EDIT	MATH	OPS	CPLX															
dim	Fill	ident	ref	rref	▶	aug	rSwap	rAdd	multR	mRAdd	▶	randM							

MATRIX CPLX (矩阵复数) 菜单 [2nd] [MATRIX] [F5]

NAMES	EDIT	MATH	OPS	CPLX
conj	real	imag	abs	angle

VECTR (向量) 菜单 [2nd] [VECTR] 向量编辑器菜单 [2nd] [VECTR] [F2] 向量名 [ENTER]

NAMES	EDIT	MATH	OPS	CPLX		INSi	DELi	▶REAL		
-------	------	------	-----	------	--	------	------	-------	--	--

VECTR MATH (向量数学运算) 菜单 [2nd] [VECTR] [F3]

NAMES	EDIT	MATH	OPS	CPLX
cross	unitV	norm	dot	

VECTR OPS (向量运算) 菜单 $\boxed{2nd} \boxed{[VECTR]} \boxed{F4}$

NAMES	EDIT	MATH	OPS	CPLX					
dim	Fill	►Pol	►Cyl	►Sph	►	►Rec	li►vc	vc►li	

VECTR CPLX (向量复数) 菜单 $\boxed{2nd} \boxed{[VECTR]} \boxed{F5}$

NAMES	EDIT	MATH	OPS	CPLX
conj	real	imag	abs	angle

CPLX (复数) 菜单 $\boxed{2nd} \boxed{[CPLX]}$

conj	real	imag	abs	angle	►	►Rec	►Pol			
------	------	------	-----	-------	---	------	------	--	--	--

MATH (数学运算) 菜单 $\boxed{2nd} \boxed{[MATH]}$

NUM	PROB	ANGLE	HYP	MISC	►	INTER				
-----	------	-------	-----	------	---	-------	--	--	--	--

MATH NUM (数值) 菜单 $\boxed{2nd} \boxed{[MATH]} \boxed{F1}$

NUM	PROB	ANGLE	HYP	MISC					
round	iPart	fPart	int	abs	►	sign	min	max	mod

MATH PROB (概率) 菜单 $\boxed{2nd} \boxed{[MATH]} \boxed{F2}$

NUM	PROB	ANGLE	HYP	MISC					
!	nPr	nCr	rand	randIn	►	randN	randBi		

MATH ANGLE (角度) 菜单 $\boxed{2nd} \boxed{[MATH]} \boxed{F3}$

NUM	PROB	ANGLE	HYP	MISC
◊	r	r	►DMS	

MATH HYP (双曲线) 菜单 [2nd] [MATH] [F4]

NUM	PROB	ANGLE	HYP	MISC				
sinh	cosh	tanh	$\sinh^{-1}$	$\cosh^{-1}$	$\tanh^{-1}$			

MATH MISC (其他) 菜单 [2nd] [MATH] [F5]

NUM	PROB	ANGLE	HYP	MISC				
sum	prod	seq	lcm	gcd	$\rightarrow$ Frac	%	pEval	$x\sqrt{\quad}$ eval

CONS (常量) 菜单 [2nd] [CONS]

BLTIN	EDIT	USER		
-------	------	------	--	--

CONS BLTIN (内置常量) 菜单 [2nd] [CONS] [F1]

BLTIN	EDIT	USER												
Na	k	Cc	ec	Rc										
					Gc	g	Me	Mp	Mn					
										μ0	ε0	h	c	u

CONV (换算) 菜单 [2nd] [CONV]

LNGTH	AREA	VOL	TIME	TEMP				
MASS	FORCE	PRESS	ENRGY	POWER	SPEED			

CONV LNGTH (长度) 菜单 [2nd] [CONV] [F1]

LNGTH	AREA	VOL	TIME	TEMP										
mm	cm	m	in	ft	yd	km	mile	nmile	lt-yr	mil	Ang	fermi	rod	fath

CONV AREA* (面积) 菜单 [2nd] [CONV] [F2]

LNGTH	AREA	VOL	TIME	TEMP				
ft ²	m ²	mi ²	km ²	acre	in ²	cm ²	yd ²	ha

2nd [CONV] F3

LNGTH	AREA	VOL	TIME	TEMP
liter	gal	qt	pt	oz

cm ³	in ³	ft ³	m ³	cup
tsp	tbsp	mi	galUK	ozUK

[2nd] [CONV] [F4]

LNGTH	AREA	VOL	TIME	TEMP
sec	mn	hr	day	yr

▶ week ms μs ns

[2nd] [CONV] [F5]

LNGTH	AREA	VOL	TIME	TEMP
°C	°F	°K	°R	

2nd [CONV] MORE F1

MASS	FORCE	PRESS	ENRGY	POWER
gm	kg	lb	amu	slug

2nd [CONV] MORE F2

MASS	FORCE	PRESS	ENRGY	POWER
N	dyne	tonf	kgf	lbf

[2nd] [CONV] [MORE] [F3]

MASS	FORCE	PRESS	ENRGY	POWER					
atm	bar	N/m ²	lb/in ²	mmHg	▶	mmH2	inHg	inH20	

2nd [CONV] MORE F4

MASS	FORCE	PRESS	ENRGY	POWER
------	-------	-------	-------	-------

J	cal	Btu	ft-lb	kw-hr	▶	eV	erg	I-atm		
---	-----	-----	-------	-------	---	----	-----	-------	--	--

CONV POWER (功率) 菜单

[2nd] [CONV] [MORE] [F5]

CONV SPEED 菜单

[2nd] [CONV] [MORE] [MORE] [F1]

MASS	FORCE	PRESS	ENRGY	POWER
hp	W	ftlb/s	cal/s	Btu/m

SPEED				
ft/s	m/s	mi/hr	km/hr	knot

STRNG (字符串) 菜单 [2nd] [STRNG]

"	sub	lngh	EqSt	StEq
---	-----	------	------	------

LIST (数组) 菜单 [2nd] [LIST]

{	}	NAMES	EDIT	OPS
---	---	-------	------	-----

LIST NAMES 菜单 [2nd] [LIST] [F3]

{	}	NAMES	EDIT	OPS
fStat	xStat	yStat		

数组编辑器菜单 [2nd] [LIST] [F4]

{	}	NAMES	"	OPS	▶	REAL				
---	---	-------	---	-----	---	------	--	--	--	--

LIST OPS (运算) 菜单 [2nd] [LIST] [F5]

{	}	NAMES	EDIT	OPS
dimL	sortA	sortD	min	max

sum	prod	seq	livc	vcHi	▶	Fill	aug	cSum	Delta	Sortx
-----	------	-----	------	------	---	------	-----	------	-------	-------

Sorty	Select	SetLE	Form	
-------	--------	-------	------	--

数值 **BASE** 菜单 [2nd] [BASE]

A-F	TYPE	CONV	BOOL	BIT
-----	------	------	------	-----

BASE A-F (十六进制) 菜单 [2nd] [BASE] [F1]

A	TYPE	CONV	BOOL	BIT
B	C	D	E	F

BASE TYPE 菜单 [2nd] [BASE] [F2]

A-F	TYPE	CONV	BOOL	BIT
b	h	o	d	

BASE BOOL (布尔) 菜单 [2nd] [BASE] [F4]

A-F	TYPE	CONV	BOOL	BIT
and	or	xor	not	

TEST (关系) 菜单 [2nd] [TEST]

==	<	>	≤	≥	▶	≠				
----	---	---	---	---	---	---	--	--	--	--

MEM (内存) 菜单 [2nd] [MEM]

RAM	DELET	RESET	TOL	ClrEnt
-----	-------	-------	-----	--------

MEM DELET (删除) 菜单 [2nd] [MEM] [F2]

ALL	REAL	CPLX	LIST	VECTR	▶	MATRX	STRNG	EQU	CONS	PRGM	▶	GDB	PIC				
-----	------	------	------	-------	---	-------	-------	-----	------	------	---	-----	-----	--	--	--	--

MEM RESET 菜单 [2nd] [MEM] [F3]

RAM	DELET	RESET	TOL	ClrEnt
ALL	MEM	DFLTS		

BASE CONV (换算) 菜单 [2nd] [BASE] [F3]

A-F	TYPE	CONV	BOOL	BIT
►Bin	►Hex	►Oct	►Dec	

BASE BIT 菜单 [2nd] [BASE] [F5]

A-F	TYPE	CONV	BOOL	BIT
rotR	rotL	shftR	shftL	

MEM RESET Are You Sure? 菜单

			YES	NO
--	--	--	-----	----

STAT (统计) 菜单 [2nd] [STAT]

CALC	EDIT	PLOT	DRAW	VARs	▶	FCST				
------	------	------	------	------	---	------	--	--	--	--

STAT CALC (计算) 菜单 [2nd] [STAT] [F1]

CALC	EDIT	PLOT	DRAW	VARs													
OneVa	TwoVa	LinR	LnR	ExpR	▶	PwrR	SinR	LgstR	P2Reg	P3Reg	▶	P4Reg	StReg				

按 [2nd] [STAT] [F2] 后, 显示数组
编辑器和数组菜单。

STAT PLOT 菜单 [2nd] [STAT] [F3]

PLOT1	PLOT2	PLOT3	PIOn	PIOff
-------	-------	-------	------	-------

Plot Type 菜单 [2nd] [STAT] [F3] ([F1]、[F2] 或 [F3]) ▾

PLOT1	PLOT2	PLOT3	PIOn	PIOff
SCAT	xyLINE	MBOX	HIST	BOX

统计图标记菜单 [2nd] [STAT] [F3] ([F1]、[F2] 或 [F3]) ▾ ([F1]、[F2] 或 [F3]) ▾ ▾ ▾

PLOT1	PLOT2	PLOT3	PIOn	PIOff
□	+	•		

STAT DRAW 菜单 [2nd] [STAT] [F4]

CALC	EDIT	PLOT	DRAW	VARS
HIST	SCAT	xyLINE	BOX	MBOX

▶ DRREG CLDRW DrawF STPIC RCPIC
STAT VARS (统计结果变量) 菜单 [2nd] [STAT] [F5]

CALC	EDIT	PLOT	DRAW	VARS
$\bar{x}$	σx	Sx	$\bar{y}$	σy

▶ Sy Σx Σx^2 Σy Σy^2 ▶ Σxy RegEq corr a b
▶ n minX maxX minY maxY ▶ Med PRegC Qrt11 Qrt13 toMe
CHAR (字符) 菜单 [2nd] [CHAR]

MISC	GREEK	INTL		
------	-------	------	--	--

CHAR MISC (其他) 菜单 [2nd] [CHAR] [F1]

MISC	GREEK	INTL		
?	#	&	%	'

▶ ! @ \$ ~ | ▶ ´ Ñ ñ Ç ç

$\tilde{N}$ 、 $\tilde{n}$ 、 Ç 和 ç 可以作为变量名的第一个字母。

%、' 和 ! 可以是函数。

所有 CHAR GREEK 菜单项是有效的变量名字符，包括第一个字母。 π (2nd $[\pi]$) 作为字符是无效的； π 是 TI-86 中的一个常量。

希腊字符菜单 2nd $[\text{CHAR}]$ $[\text{F2}]$

MISC	GREEK	INTL							
α	β	γ	Δ	δ	▶				
					ϵ	θ	λ	μ	ρ
					▶				
					Σ	σ	τ	ϕ	Ω

CHAR INTL (国际字母符号) 菜单 2nd $[\text{CHAR}]$ $[\text{F3}]$

MISC	GREEK	INTL		
✓	✓	^	..	

疑难解答

- 如果在屏幕上看不见任何东西，则可能需要调节对比度（第 1 章）。
 - 欲使屏幕变暗，按下并释放 2nd 后再按住 $\blacktriangle$ 。
 - 欲使屏幕变亮，按下并释放 2nd 后再按住 $\blacktriangledown$ 。
- 如果错误菜单显示，请按照第 1 章描述的步骤处理。必要时可参阅附录中“错误情形”一节（第 393 页）中有关错误的详细介绍。
- 如果显示棋盘状光标 ($\blacksquare$)，要么在提示处输入了最大数目的字符，要么内存已满。如果内存已满，按 2nd $[\text{MEM}]$ $[\text{F2}]$ 选择数据类型，然后从内存中删除一些项（第 17 章）。
- 如果忙指示器（点画线）显示在右上角，图象或程序暂停；TI-86 正等待输入。按 $[\text{ENTER}]$ 继续或按 $[\text{ON}]$ 中断。
- 如果计算器似乎根本不工作，请确保电池是新的，并已安装正确。请参阅第 1 章中有关电池的信息。

错误情形

当 TI-86 发现错误时，它将显示错误消息 **ERROR # type** 和错误菜单。第 1 章描述了如何改正错误。本节描述引起错误的可能原因和示例。要查找函数或指令的正确参数，以及参数的限制，请参阅第 20 章：A 到 Z 函数和指令参考。

绘制图象时不会产生错误 1 到错误 5。TI-86 允许在图象中使用未定义的值。

01 OVERFLOW	◆ 输入数字超出计算器范围。 ◆ 表达式的计算结果超出计算器范围。
02 DIV BY ZERO	◆ 除零。 ◆ 用垂直线进行线性回归。
03 SINGULAR MAT	◆ 用奇异矩阵（行列式值 = 0）作为 x^{-1}、Simult 或 LU 的参数。 ◆ 用于回归的数组至少有一个不合适。 ◆ 作为 exp、cos 或 sin 参数的矩阵有重复的特征值。
04 DOMAIN	◆ 函数或指令的参数值越界。 ◆ 用 $-x$ 进行逻辑回归或幂回归，或者用 $-y$ 进行指数回归。
05 INCREMENT	seq 中的增量是 0，或者是符号错误；循环增量是 0。
06 BREAK	按 ON 中断了程序、 DRAW 指令或表达式计算。
07 SYNTAX	输入了一个数值；检查放错位置的函数、参数、圆括号或逗号；请参阅 A 到 Z 参考中的语法描述。

- | | |
|------------------------|---|
| 08 NUMBER BASE | <ul style="list-style-type: none">◆ 在某种数制时输入了无效的位，例如 7b。◆ 执行了在 Bin、Oct 或 Hex 数制下不允许的运算。 |
| 09 MODE | 试图保存到非当前图象方式的窗口变量；或者使用了仅在非当前图象方式下有效的指令；例如，在 Pol 、 Param 或 DifEq 图象方式下使用 DrInV 。 |
| 10 DATA TYPE | <ul style="list-style-type: none">◆ 输入的数据或变量与数据类型不符。◆ 输入的参数与函数或指令参数的数据类型不符，例如将程序名作为 sortA 的参数。◆ 在编辑器中，输入了不允许的数据类型；请查看相应的章节。◆ 试图将数据保存到一个被保护的数据类型中，例如常量、程序、图形或图象数据库中。◆ 试图将不合适的数据保存到被限制使用的内置变量中，例如数组名 xStat，yStat 和 fStat。 |
| 11 ARGUMENT | 执行的函数或指令缺少所有参数。 |
| 12 DIM MISMATCH | 用两个或多个数组、矩阵、向量作参数，但是各参数的维数不相等，例如 {1,2}+{1,2,3} 。 |
| 13 DIMENSION | <ul style="list-style-type: none">◆ 输入参数的维数错误。◆ 输入的矩阵或向量的维数 < 1 或 > 255 或是非整数。◆ 非方阵求逆。 |
| 14 UNDEFINED | 引用的变量未定义。 |
| 15 MEMORY | 内存不足，无法执行指定的命令；执行命令前必须从内存中删除一些项（第 17 章）。 |
| 16 RESERVED | 非法使用内置变量。 |
| 17 INVALID | 引用的变量或使用的函数无效。 |

- 18 ILLEGAL NEST** 试图将无效的函数作为 **seq(** 或 **CALC** 函数的参数；例如，**der1(der1(x^3,x),x)**。
- 19 BOUND** 定义的上限小于指定的下限，或定义的下限大于指定的上限。
- 20 GRAPH WINDOW** ♦ 定义图象屏幕时，一个或多个窗口变量不合适；例如定义的 **xMax < xMin** 。
- ♦ 窗口变量太大或太小，无法正确绘制；例如将图象缩小超出了计算器的范围。
- 21 ZOOM** **ZOOM** 操作引起错误；试图用直线定义了 **ZBOX** 。
- 22 LABEL** 在编程中，**Goto** 指令的标记未用 **Lbl** 指令定义。
- 23 STAT** ♦ 进行统计计算时，使用的数组至少有一个不合适，例如数组中的数据点少于两个。
- ♦ 频数数组中至少有一个元素 < 0 。
- ♦ 绘制直方图时， $(xMax - xMin)/xScl \leq 63$ 必须为真
- 24 CONVERSION** 度量单位换算时，度量单位不对应，例如将伏特转换为公升。
- 25 SOLVER** ♦ 在求解编辑器中，方程中无变量。
- ♦ 试图在光标位于边界时绘制。
- 26 SINGULARITY** 在求解编辑器中，方程包含一个奇异点，函数在此处没有定义。
- 27 NO SIGN CHNG** 求解器编辑器未检测到符号的变化。
- 28 ITERATIONS** 求解器超过了允许的最大迭代次数。
- 29 BAD GUESS** ♦ 初始估计值在指定的界限外。
- ♦ 初始估计值和其周围的一些点未定义。

错误 26 到 29 发生在求解过程中。在 **SOLVER** 中检查函数图象或对于 **left-rt** 变量的图象。如果方程有解，改变界限和/或初始估计值。

30 DIF EQ SETUP	在 DifEq 图象方式下，在方程编辑器中的方程必须是从 Q'1 到 Q'9 ，且每个方程必须有从 QI1 到 QI9 相关的初始条件。
31 DIF EQ MATH	用于拟合算法的步长太小；检查方程和初始值；给窗口变量 difTol 设置一个较大值；试着改变 tMin 或 tMax 来检查不同区域的结果。
32 POLY	所有系数均为 0 。
33 TOL NOT MET	运算返回的结果无法达到所要求的精度。
34 STAT PLOT	显示的统计图引用了未定义的数组。
35 AXES	绘制 DifEq 图象时，设置的坐标轴不合适。
36 FLD/ORDER	◆ 试图用 SlpFld 域格式设置来绘制二阶或更高阶的微分方程。修改域格式或阶次。 ◆ 试图用 DirFld 域格式设置来绘制三阶或更高阶的微分方程。修改域格式或阶次。
37 LINK MEMORY FULL	传送项时，接收单元的内存不足；跳过项或取消传送。
38 LINK TRANSMISSION ERROR	◆ 无法传送项目；检查连接电缆是否插牢，接收单元是否已准备好接收数据（第 18 章）。 ◆ 已按 [ON] 中断传送。
39 LINK DUPLICATE NAME	传送的项目名与接收单元上已存在的项目同名。

Equation Operating System (EOS™)

Equation Operating System (EOS) 控制 TI-86 的运算次序。EOS 首先进行括号中的计算，然后按以下次序计算表达式中的函数：

在同一优先级中，EOS 从左到右计算函数。

多参数函数，例如 **nDeriv** (**A2,A,6**)，按照遇到顺序计算。

TI-86 蕴涵乘法规则不同于 TI-85 的规则。例如，TI-86 将 $1/2x$ 计算为 $(1/2)*x$ ，而 TI-85 计算为 $1/(2*x)$ 。

- 1st 在参数后输入的函数，例如， 2 、 $^{-1}$ 、 $!$ 、 $^\circ$ 、 r 和转换
- 2nd 幂和根，例如 2^5 或 $5\sqrt{32}$
- 3rd 单参数函数，函数名在参数前。例如 $\sqrt{}$ 、**sin**(或 **log**(
- 4th 排列 (**nPr**) 和组合 (**nCr**)
- 5th 乘法，蕴涵乘法和除法
- 6th 加法和减法
- 7th 关系函数，例如 $>$ 或 $\leq$
- 8th 逻辑运算 **and**
- 9th 逻辑运算 **or** 和 **xor**

蕴涵乘法

TI-86 认可蕴涵乘法，因此在所有情况下不必按 $\square$ 来表示乘法。例如，TI-86 将 2π 、 $4\sin(46)$ 、 $5(1+2)$ 和 $(2*5)7$ 解释为蕴涵乘法。

圆括号

首先完成圆括号中的计算。例如，在表达式 $4(1+2)$ 中，EOS 先计算圆括号中的 $1+2$ ，然后用 4 乘以 6 。

$4*1+2$	
$4(1+2)$	6
	12

可以省略表达式最后的右括号 ()。所有左括号在表达式最后自动给出右括号。这一规则在保存或显示转换指令前的左括号也正确。

数组名、矩阵名或方程函数名后的左括号并不解释为蕴涵乘法。左括号后的参数是指定的数组元素、矩阵元素或方程函数求解的变量值。

TOL (误差编辑器) 2nd [MEM] F4

在 TI-86 中, 一些函数的计算精度由变量 **tol** 和 δ 来控制。这些变量值可能会影响 TI-86 计算或绘制的速度。

```
TOLERANCE
tol=1E-5
 $\delta$ =.001
```

变量 **tol** 定义在计算函数 **fnInt()**、**fMin()**、**fMax()** 和 **arc()**, 以及 GRAPH MATH 运算 $\Sigma f(x)$ 、**FMIN**、**FMAX** 和 **ARC** (第 6 章) 的误差。**tol** 必须是一个 $\geq 1E-12$ 正数

保存在 δ 中的数值必须是一个正实数。 δ 定义步长的大小, TI-86 用此步长计算 **dxNDer** 方式下函数 **arc**; 计算 **nDer**; 计算运算 **dy/dx**、**dr/d θ** 、**dy/dt**、**dx/dt**、**INFLC**、**TANLN**、**ARC** 和 **dxNDer** 方式下的所有函数 (第 6 章)。

在主屏幕或程序中用 STO► 将数值保存到 **tol** 或 δ 中。可以从 CATALOG 中选择 **tol** 和 δ 。同样可以直接输入 **tol**, 从 CHAR GREEK 菜单中选择 δ 。

计算精度

为了达到最大的精度，TI-86 在内部保留的位数多于显示位数。保存在内存中的值用 14 位数字和三位指数来表示。

- ◆ 可将长度多达 12 位的数值保存到大多数窗口变量中；可将长度多达 14 位的数值保存到变量 **xScl**、**yScl**、**tStep** 和 **θStep** 中。
- ◆ 当显示一个数值时，显示的数值根据方式设置（第 1 章）进行四舍五入，最多有 12 位数字和三位指数。
- ◆ 第 4 章描述了十六进制、八进制和二进制数的计算。

有关 TI 产品、服务和保障信息

TI 产品和服务信息

有关 TI 产品和服务的详细信息，请通过 e-mail 与 TI 联系或访问 TI 计算器的 WWW 主页。

e-mail 地址：

ti-cares@ti.com

Internet 地址：

education.ti.com

服务和保障信息

有关保障的时效和范围或者产品服务的信息，请参阅产品中的保障声明，或与当地的 Texas Instruments 零售商/代理商联系。

索引

" (字符串), 216, 227

" (数组编辑器菜单), 156

! (阶乘), 294

➤, 362

≥ (大于等于), 56, 301

≤ (小于等于), 55, 312

≠ (不等于), 56, 326

π (圆周率), 48

√ (平方根), 360

[√] (平方根) 键, 48

$\bar{x}$ (统计结果变量), 193

$\bar{y}$ (统计结果变量), 193

⁻¹ (反), 48, 309

➔dim, 184, 281

➔dimL, 282

f(x) (函数数值积分), 96, 98

ΔTbl (运算表步长), 113

σx (统计结果变量), 193

Σx² (统计结果变量), 193

σy (统计结果变量), 193

% (百分比), 52, 334

< (小于), 55, 312

= (赋值), 270

= (相等), 290

== (关系相等), 55, 291

> (大于), 55, 300

[], 319, 369

^ (指数), 48

{ }, 316

10ⁿ (10 的 *n* 次方), 48, 337

b (二进制), 271

h (十六进制), 302

d (小数), 278

o, 326

►Bin (转换为二进制), 68, 272

►Cyl (转换为柱面坐标),
174, 278

►Dec (转换为十进制), 279

►DMS (转换成度/分/秒), 51,
285

►Frac (转换为分数), 52, 298

►Hex (转换为十六进制), 68,
303

►Oct (转换为八进制), 327

►Pol (转换为极坐标), 71, 174,
336

►REAL (转换为实数), 156,
170, 179

►Rec (转换为直角坐标), 71,
174, 343

►Sph (转换为球坐标), 174,
360

[ENTRY] 键, 19

10 的幂 (10ⁿ), 20, 34, 337

abs (绝对值), 49, 71, 175, 185,
267

ALL-, 77

ALL, 43, 232

ALL+, 77

Ans (上次结果), 29, 30, 41,
269

APD, 参见 Automatic Power
Down

arc(), 54, 269

ARC, 96, 98

Asm (汇编语言程序), 269

AsmComp (编译汇编语言程
序), 226, 270

AsmPrgm (汇编语言程序),
226, 270

aug(), 160, 184, 270

Automatic Power Down, 17

Axes(), 271

BASE BIT 菜单, 69

BASE BOOL (布尔) 菜单, 68

BASE CONV (换算) 菜单, 68

BASE TYPE 菜单, 67

BASEA-F (十六进制) 菜单, 67

BASE 菜单, 66

BCKUP (内存备份), 237

Bin (二进制), 35, 272

BOX (GRAPH ZOOM 菜单),
14, 92, 93

BOX (ZOOM 菜单), 208

BREAK 菜单, 26

CALC (微积分) 菜单, 54

CATALOG, 25, 38

快速查找定位器, 262

CATLG (CATALOG), 43

CATLG-VARS (CATALOG 变
量) 菜单, 43

CHAR GREEK 菜单, 46

CHAR INTL (国际) 菜单, 46

CHAR MISC (其他) 菜单, 46

CHAR (字符) 菜单, 45

- Circl(, 273
 CIRCL (圆), 105, 106
 CLDRW (清除图形), 103, 105, 273
 CILCD (清除 LCD), 216, 273
 ClrEnt (清除输入), 232, 273
 CITbl (清除运算表), 114, 216, 273
 cnorm (列范数), 183, 273
 cond (条件编号), 183, 274
 conj (共轭复数), 71, 175, 185, 275
 CONS BLTIN (内置常量)菜单, 58
 CONS EDIT 菜单, 60
 CONS (常量), 43
 CONS (常量)菜单, 58
 CONV AREA 菜单, 63
 CONV ENRGY (能量)菜单, 64
 CONV FORCE 菜单, 64
 CONV LGTH (长度)菜单, 63
 CONV MASS 菜单, 64
 CONV POWER 菜单, 64
 CONV PRESS (压强)菜单, 64
 CONV SPEED 菜单, 64
 CONV TEMP (温度)菜单, 8, 63
 CONV TIME 菜单, 63
 CONV VOL (体积)菜单, 63
 CONV (换算)菜单, 62
 corr (相关系数), 193
 $\cos^{-1}$ (反余弦), 48, 276
 cos (余弦), 48, 186, 276
 $\cosh^{-1}$ (反双曲余弦), 51, 277
 cosh (双曲余弦), 51, 277
 CPLX (复数变量), 43, 71
 cross(, 173, 277
 cSum((累加和), 160, 278
 CUSTOM 菜单, 44
 拷贝项, 44
 清除项, 45
 CyIV (柱面向量坐标), 36, 278
 Dec (十进制), 35, 65
 Dec (十进制计数法), 278
 DELc (删除列), 179
 DELETE, 60
 DELf (删除函数), 77
 DELi (删除元素), 170
 DELr (删除行), 179
 Deltalst((删除数组), 160, 279
 DelVar((删除变量), 219, 280
 der1((一阶导数), 54, 280
 der2((二阶导数), 54, 280
 det (行列式), 183, 281
 DFLT5 (默认值), 232
 DifEq (微分方程方式), 35, 74, 239, 281
 difTol (误差), 136
 dim (维数), 173, 184, 281
 dimL (数组的维数), 159, 282
 DirFld (方向域), 134, 282
 Disp (显示), 216, 283
 DispG (显示图象), 283
 DispT (显示运算表), 284
 DIST (距离), 96, 98
 dot(, 173, 285
 dr/d θ , 122
 DRAW, 75, 88
 DrawDot, 84, 285
 DrawF (绘制函数), 103, 107, 286
 DrawLine, 84, 286
 DrEqu((绘制方程), 145, 287
 DrInv (绘制反函数), 103, 107, 287
 DS<((减 1 和跳过), 219, 288
 DUPLICATE NAME 菜单, 241
 dx/dt, 130
 dxDer1 (精确微分), 36, 75, 288
 dxNDer (数值微分), 36, 75, 288
 dy/dt, 130
 dy/dx, 96, 99, 130
 E (指数), 48, 292
 e^ (以 e 为底的幂), 288
 eigVc (特征向量), 183, 289
 eigVl (特征值), 183, 289
 Else, 218, 306
 e-mail 地址 (TI 用户支持), 392
 End, 218, 290, 297, 306
 Eng (工程记数法), 34, 20, 290
 ENTRY 存储区域, 28, 29
 EOS. 参见 Equation Operating System
 EqSt((将方程转换为字符串), 227, 290
 eqn (方程)变量, 54, 203, 205

- EQU (方程变量), 43
 EStep, 136
 Euler 方法, 133, 291
 eval, 52, 76, 88, 101, 122, 130, 150, 291
 evalF, 54, 292
 e^x (以常数 e 为底的幂), 48
 EXIT (取消数据传送), 241
 EXPLR (浏览), 148
 ExpR (指数回归), 190, 293
 exp 变量, 54, 203
 fcstx (预测 x), 294
 fcsty (预测 y), 294
 Fill, 160, 173, 295
 Fill, 184
 Fix, 295
 FldOff (斜率和方向域关闭), 134, 295
 fldPic (域) 变量, 138
 fMax((函数最大值), 54, 296
 FMAX (函数最大值), 96, 97
 fMin((函数最小值), 54, 296
 FMIN (函数最小值), 96, 97
 fnInt((函数积分), 54, 296
 FnOff (函数关闭), 296
 FnOn (函数打开), 297
 For, 218, 297
 Form, 161, 298
 FORMT (图象格式), 76
 fPart (分数部分), 49, 176, 186, 298
 fStat (频数数组), 189
 Func (函数方式), 35, 74, 239, 299
 gcd((最大公分母), 52, 299
 GDB (图象数据库), 43
 GDB 变量, 102
 Get, 299
 getKy (获取键码), 216, 300
 键码图, 217
 GOTO, 26, 27, 300
 GRAPH DRAW 菜单, 75, 103, 122, 145
 GRAPH LINK, 235
 GRAPH MATH 菜单, 75, 95, 122, 130
 GRAPH MATH 算子
 其他设置的影响, 96
 使用 $\int f(x)$ 、DIST 或 ARC, 98
 使用 dy/dx 或 TANLN, 99
 使用 ISECT, 100
 使用 ROOT、FMIN、FMAX 或 INFLC, 97
 使用 YICPT, 100
 GRAPH ZOOM 菜单, 75, 91, 147
 GRAPH (求解器菜单), 206
 GRAPH 菜单, 27, 31, 75, 88, 117, 126, 133
 GrStl((图象样式), 220, 302
 Hex (十六进制), 35, 302
 Hist (直方图), 303
 HORIZ (水平线), 105, 106
 Horiz, 304
 IAsk, 304
 IAuto, 304
 ident (单位方阵), 184, 304
 If, 218, 305, 306
 imag (虚数), 71, 175, 185, 306
 INFLC (拐点), 96, 97
 INITC (初始条件), 136
 InpSt, 217, 307
 Input (PRGM I/O 菜单), 216, 307
 INSc (插入列), 179
 INsf (插入函数), 77
 INSi (插入元素), 170
 INSr (插入行), 179
 int (整数), 49, 176, 186, 308
 inter((插入), 309
 Internet
 e-mail 地址 (TI 用户支持), 392
 下载程序, 235
 IPart (整数部分), 6, 49, 176, 186, 309
 IS>((增 1 和跳过), 219, 310
 ISECT (交点), 96, 100
 Lbl (标记), 219, 224, 311
 lcm((最小公倍数), 52, 311
 LCust((加载自定义菜单), 220, 311
 left-rt, 202
 LgstR (逻辑回归), 190, 193, 313
 list>v (数组转换为向量), 160, 174, 316
 Line, 314
 LINE, 104, 105
 LINK SEND85 菜单, 239
 LINK SEND 菜单, 236

- LINK 菜单, 236
- LinR (线性回归), 190, 315
- LIST NAMES 菜单, 153, 189
- LIST OPS 菜单, 159
- LIST 菜单, 152
- ln (自然对数), 48, 316
- length (字符串长度), 227, 316
- LnR (对数回归), 190, 317
- log, 48, 318
- LU(下三角-上三角), 183, 318
- Macintosh
 链接到, 235
- MATH ANGLE 菜单, 51
- MATH HYP (双曲线) 菜单, 51
- MATH MISC (其他) 菜单, 52
- MATH NUM (数字) 菜单, 31, 49
- MATH PROB (概率) 菜单, 50
- MATH (图象菜单), 88
- MATH, 75
- MATH 菜单, 31, 49
- MATRIX CPLX (复数) 菜单, 185
- MATRIX MATH 菜单, 183
- MATRIX NAMES 菜单, 178
- MATRIX OPS (运算) 菜单, 184
- MATRIX (矩阵) 菜单, 178
- MATRIX (矩阵名), 43
- max(, 49, 160, 319
- maxX, 193
- maxY, 193
- MBox, 319
- Med (中位值), 193
- MEM DELET (删除) 菜单, 231
- MEM FREE (可用内存), 230
- MEM RESET 菜单, 232
- MEM (内存) 菜单, 29, 230
- MEM (清除内存), 232
- Menu(, 219, 320
- min(, 49, 160, 320
- minX, 193
- minY, 193
- mod(, 49, 320
- mRAdd(, 321
- mRAdd, 184
- multR((乘行), 184, 322
- n (统计结果变量), 193
- nCr (组合数), 50, 322
- nDer((数值导数), 54, 323
- nPr (排列数), 50, 326
- Oct (八进制), 35, 327
- OneVa (一元), 189, 191, 327
- Outpt(, 217, 329
- OVERW (覆写), 241
- P2Reg (二次回归), 190, 330
- P3Reg (三次回归), 190, 331
- P4Reg (四次回归), 190, 332
- Par, 74
- Param (参数方式), 35, 239, 333
- PC
 链接到, 235
- PEN, 105
- pEval(, 52, 334
- pi, 59
- PIC (图形名), 43
- PIC 变量
 保存图象, 102
 输入, 76
- PIOff (统计图关闭), 195, 334
- PIOn (统计图打开), 195, 334
- Plot1(, 335
- PLOT1, 195
- Plot2(, 335
- PLOT2, 195
- Plot3(, 335
- PLOT3, 195
- Pol (极坐标方式), 35, 74, 239, 336
- PolarC (极坐标复数方式), 35, 336
- PolarGC (极坐标图象坐标系), 84, 336
- poly, 337
- PRRegC, 193
- PRGM CTL 菜单, 218
- PRGM I/O (输入/输出) 菜单, 215
- PRGM (程序名), 43
- PRGM 菜单, 214
- prod (乘积), 52, 160, 338
- Prompt (PRGM I/O 菜单), 216, 338
- PtChg(, 338
- PTCHG, 105
- PtOff(, 338
- PTOFF, 105, 108
- PtOn(, 338
- PTON, 105, 108
- PwrR (幂回归), 190, 339

- PxChg(, 103, 340
 PxOff(, 103, 340
 PxOn(, 103, 340
 PxTest(, 103, 340
 Q'n 方程变量, 135
 QrtI1, 193
 QrtI3, 193
 r (弧度输入), 341
 rAdd(, 340
 rAdd, 184
 Radian (角度方式), 35
 rand Bin((随机二项式), 50, 341
 rand Int((随机整数), 50, 342
 rand (随机数), 50, 341
 randM((随机矩阵), 184, 342
 randNorm((随机正态), 50, 342
 RCGDB (调用图象数据库), 76, 88, 343
 RcPic (调用图形), 76, 102, 343
 RCPIC 菜单, 76
 REAL, 43, 175, 185, 343
 RectC (直角坐标复数), 35, 344
 RectGC (直角图象坐标系), 84, 344
 RectV (直角向量坐标方式), 36, 344
 RECV (LINK SND85 菜单), 240
 RECV (LINK 菜单), 236
 ref (行梯阵), 184, 344
 RENAM (重命名), 241
 Repeat (PRGM CTL 菜单), 218, 345
 Return (PRGM CTL 菜单), 219, 345
 RK (Runge-Kutta) 法, 133, 345
 rnorm (行范数), 183, 346
 ROOT, 96, 97
 $x\sqrt{}$, 346
 RotL (循环左移), 69, 347
 RotR (循环右移), 69, 347
 round(, 49, 176, 348
 rref (简化的行梯矩阵), 184, 348
 rSwap((行置换), 184, 348
 Scatter (统计图的类型), 349
 Sci (科学记数法), 20, 34, 349
 SELCT, 112
 Select(, 161, 350
 SELECT, 77
 SEND WIND 屏幕, 238
 Send(, 216, 350
 SEND (LINK 菜单), 236
 seq((序列), 52, 160, 351
 SeqG (序列图象), 84, 351
 SetLE, 159
 SetLEdit, 161, 351
 Shade(, 103, 104, 352
 ShiftL (左移), 69, 353
 ShiftR (右移), 69, 353
 ShwSt (显示字符串), 354
 sign, 49, 354
 SimulG (同时绘制), 84, 354
 SIMULT ENTRY 菜单, 208
 SIMULT RESULT 菜单, 209
 simult(, 210, 354
 SIMULT 次序屏幕, 208
 $\sin^{-1}$ (反正弦), 48, 355
 $\sin$ (正弦), 48, 186, 355
 $\sinh^{-1}$ (反双曲正弦), 51, 356
 $\sinh$ (双曲正弦), 51, 356
 SinR (正弦回归), 190, 193, 357
 SKIP, 241
 SlpFld (斜率域), 134, 358
 SND85 (LINK 菜单), 236
 SOLVE, 205
 Solver(, 358
 sortA, 159, 359
 sortD, 159, 359
 Sortx, 160, 359
 Sorty, 160, 359
 SphereV (球向量坐标方式), 36, 360
 StEq((字符串转换为方程), 227, 361
 STAT PLOT 菜单, 195
 STAT PLOT 状态屏幕, 194
 STAT VARS (统计变量) 菜单, 192
 STAT (统计结果变量), 43
 STATCALC (计算) 菜单, 189
 STAT 菜单, 188
 STGDB (保存图象数据库), 76, 88, 361
 STOb, 210
 STOb, 210
 Stop, 219, 362

- STOx, 210
 STPIC (保存图形), 76, 88, 362
 STPIC 菜单, 76
 StReg (保存回归方程), 190, 362
 STRNG (字符串) 菜单, 227
 STRNG (字符串变量), 43
 STYLE, 77
 sub((字符串的子集), 227, 363
 Sx (统计结果变量), 193
 T (转置), 367
 TABLE 菜单, 110
 $\tan^{-1}$ (反正切), 48, 365
 tan (正切), 48, 364
 $\tanh^{-1}$ (反双曲正切), 51, 365
 tanh (双曲正切), 51, 365
 TanLn(, 103, 107, 366
 TANLN (正切线), 96, 99
 TBLST (运算表设置编辑器), 112, 113
 TEST 菜单, 55
 Text(, 366
 TEXT, 105
 Then, 218, 305, 306
 TI-GRAPH LINK, 235
 tMax, 127, 136
 tMin, 127, 136
 TOL (误差编辑器), 398
 tPlot, 136
 TRACE (光标), 75
 TRACE (求解器菜单), 207
 Trace (图象菜单), 367
 TRACE, 88
 tStep, 127, 136, 138
 TwoVa (二元), 189, 368
 unitV (单位向量), 173, 368
 VARS CPLX (复数变量) 屏幕, 71
 VARS EQU 菜单, 203
 vclli (将向量转换为数组), 160, 174, 369
 VECTR CPLX (复数) 菜单, 175
 VECTR MATH 菜单, 173
 VECTR NAMES 菜单, 169
 VECTR OPS (运算) 菜单, 173
 VECTR (向量名), 43
 VECTR 菜单, 169
 VERT (垂直线), 104, 106, 369
 While, 218, 369
 WIND (窗口变量), 43, 35, 75, 238
 WIND (求解器菜单), 206
 XMIT (传送), 237, 240
 xRes (分辨率), 81
 xScl (尺度), 81
 xStat (x 变量数组), 189
 xyline, 370
 x 变量, 77
 y(x)=, 75
 YICPT (y 交点), 96, 100
 yScl (尺度), 81
 yStat (y 变量数组), 189
 y 变量, 77
 ZDATA (GRAPH ZOOM 菜单), 92
 ZData, 371
 ZDECM (GRAPH ZOOM 菜单), 92
 ZDecm, 372
 ZFACT (ZOOM FACTOR), 92, 208
 ZFIT (GRAPH ZOOM 菜单), 92
 ZFit, 129, 373
 ZIn (放大), 373
 ZIN (放大), 92, 208
 ZINT (GRAPH ZOOM 菜单), 92
 ZInt, 374
 ZOOM, 14, 75, 88
 参数图象, 129
 定制, 93
 极坐标图象, 121
 ZOOMX (GRAPH ZOOM 菜单), 92
 ZOOMEY (GRAPH ZOOM 菜单), 92
 ZOOM 操作, 147
 ZOUT (缩小), 92, 208, 375
 ZPREV (缩放前一个窗口), 92, 375
 ZRCL (GRAPH ZOOM 菜单), 92, 95
 用户创建的缩放变量, 239
 ZRcl (重新调用缩放), 376
 ZSQR (GRAPH ZOOM 菜单), 92
 ZSqr, 376
 ZSTD (GRAPH ZOOM 菜单), 92
 ZSTD (标准默认值), 208, 377

ZSTO (GRAPH ZOOM 菜单),
92, 95

ZTRIG (GRAPH ZOOM 菜单),
92

ZTrig, 378

A

安装电池, 16

B

八进制数, 35, 66

八进制整数, 326

百分比 (%), 334

保存, 18

保存到变量中 (➔), 362

保存方程结果, 210

保存方程系数, 210

保存符号, 22

保存数据, 39

保存图象显示, 102

备用电池, 16

编程

编辑程序, 223

创建程序, 214

调用程序, 224

复制程序, 225

汇编语言, 225

入门, 214

删除程序, 223

使用变量, 225

输入命令行, 220

下载汇编程序, 225

已定义, 214

运行程序, 221

中断程序, 222

编辑方程, 205

编辑器菜单, 33

变量, 21

x 变量, 77

y 变量, 77

创建, 39

大写和小写名称, 39

复制, 41

根据数据类型分类, 42

将结果保存到, 3, 30

将数据保存到, 39

名称, 44

删除, 45

显示, 41

运算表中的变量方程组, 114

在表达式中, 4

在运算表屏幕中, 111

重新调用, 42

标记打开, 84, 310

标记关闭, 84, 310

表达式 (续)

计算, 29, 30

使用复数, 71

使用矩阵, 181

使用向量, 172

输入数组, 153

表达式, 18, 20, 24, 25, 26, 30,
48

编辑, 4

输入, 24

补码 (二进制数), 66

不等于 ($\neq$), 326

布尔算子, 68, 268, 325, 328,
370

C

菜单

键, 32

删除, 6, 33

上部, 32

退出, 6

下部, 33

显示, 31

选择项, 32

在编辑器中, 33

菜单的菜单项, 31

菜单图, 380

参数, 25

参数方程

删除, 127

图象, 126

选择和取消选择, 127

参数图象, 74

窗口变量, 127

定义, 125

方程编辑器, 126

方式, 35, 126

跟踪, 128

绘图, 130

默认图象样式, 126

缩放, 129

图象格式, 128

图象工具, 128

显示, 128

自由移动光标, 128

插入光标, 22, 23

取消, 23

常量, 59

定义, 58

名称, 61

内置, 58

用户自定义, 58, 60
常量内存特征, 17, 34
乘 (*), 321
乘法输入
 检索, 29
程序编辑器, 214
 菜单和屏幕, 215, 220
程序流, 56
重定义用户创建的常量, 60
除法 (/), 284
除法符号, 3
传送数据, 234, 240
 重复到几台设备上, 242
 窗口变量, 239
 错误情形, 242
 内存不足, 242
 选择变量, 238
窗口编辑器, 75
 极坐标, 118
窗口变量, 82
 Δx 和 Δy , 83
 改变, 12, 82
 图象屏幕, 81
 微分方程, 135
错误, 17, 27
 改正, 27

 来自相连公式, 165
 诊断, 27
错误菜单, 31
错误类型, 27
错误情形, 393
错误消息, 27
D
大写字母光标, 22
大写字母键, 21
大写字母字符, 22
大于 (>), 300
大于等于 ($\geq$), 301
当前输入, 19
 清除, 23
当前项, 38
导数
 计算, 7
等于 (=), 290
点
 打开和关闭, 108
 绘制, 108
电池, 2, 16-18
电池盒, 16
电话 (TI 用户支持), 392

电量不足消息, 16, 18
调用变量值, 18, 42
度 $^{\circ}$, 51
度/分/秒格式, 51
度的复数方式, 70
度量单位
 转换, 61
度输入 ($^{\circ}$), 279
端到端电缆, 234, 235
对比度
 调整, 2, 18
多项式的根
 保存到变量中, 212
多项式根查找器, 211
多项式系数
 保存到变量中, 212
多项式值, 52

E
二进制数制, 35, 66
二进制整数, 271

F
反函数
 绘图, 107
范数, 173, 183, 323

方程
 编辑, 205
 计算, 122, 130
 求解, 206
 输入, 203
方程编辑器, 74, 75, 76, 80
 参数, 126
 极坐标, 118
 输入函数, 77
 图象样式, 77
方程编辑器菜单, 76
方程变量, 40, 43, 78
方程操作系统, 397
方程存储
 自动回归, 191
方程结果
 保存到变量中, 210
方程求解器, 40, 202
 作图工具, 207
方程输入编辑器, 203
方程系数
 保存到变量中, 210
方式设置, 19, 20, 70
 改变, 34
 数制, 65
 显示, 34

非 (布尔), 66, 69, 325

分隔符, 70

分数, 3, 19

浮点数, 35, 295

负数

输入, 19

负数符号 (-), 20

复数, 29, 70

分隔符, 70

输入, 20

应用在表达式中, 71

在结果中, 70

作为结果显示, 5

作为数组元素, 156

复数变量, 43

复数菜单, 71

复数的极角表示, 72

复数的实部, 71

复数的虚部, 71

复数方式, 35

复数矩阵, 180

复数值, 48

复位内存, 232

赋值, 270

G

根查找器, 211

跟踪光标, 75, 90, 144, 205

快速缩放, 91

平移, 90

停止跟踪并继续运行程序,
91

移动, 90, 121, 129

在参数图象中, 128

在极坐标图象中, 120

跟踪函数, 11

更改 TI-86 设置, 39

更换电池, 16

公式

拆离, 166

连接, 163

连接到数组名, 162

估计值, 204

在交互式求解编辑器中, 205

关闭 TI-86, 2, 17

关系函数, 55, 56

光标, 17, 22

插入, 22

大写字母, 22

方向键, 23

改变, 23

跟踪, 90

满, 22

输入, 22

位置, 19, 20, 21, 25

小写字母, 22

选择, 38

移动, 23

自由移动, 128, 144, 205

滚动, 19

国际字母, 46

H

函数, 25, 38

跟踪, 11

绘图, 107

绘制, 11

计算, 101

键盘, 48

取消选择, 13

删除, 77

输入, 25

与数组一起使用, 5, 161

在方程编辑器中输入, 76,
77, 78

函数的语法, 25

函数图象, 73, 74

方式, 35

行

矩阵的, 181

合并 (+), 274

和, 52, 160, 364

弧度复数方式, 70

弧度角度方式, 75, 341

弧度输入 (°), 341

华氏温度

转换为摄氏温度, 8

换算

►Bin, 272

►Dec, 279

►DMS, 51, 285

►Frac, 52, 298

►Hex, 303

►Oct, 327

►Pol, 336

►REAL, 156

►Rec, 343

►Sph, 360

Eq►St, 227

li►vc, 160

St►Eq(, 227, 361

vc►li, 160

换算度量单位, 61

换算以速度表示的数值, 65

回归模型, 191

汇编语言程序, 225

绘图

参数图象, 130

点, 108

函数, 切线, 反函数, 107

极坐标图象, 122

徒手绘制点, 线, 曲线, 107

微分方程图象, 145

线段, 105

圆, 106

直线, 105, 106

绘制函数, 9, 11

绘制统计数据, 194

或 (布尔), 69, 328

J

极坐标方程

跟踪, 120

极坐标复数 ($\angle$), 336

极坐标复数表示形式, 20, 70

极坐标复数方式, 35, 336

极坐标图象, 74, 84

窗口编辑器, 118

定义, 117

方程编辑器, 118

方式, 35

跟踪, 120

跟踪光标, 120, 121

绘制, 122

默认图象样式, 118

缩放, 121

图象格式, 119

图象工具, 119

显示, 119

自由移动光标, 119

计算

中断, 26

计算导数, 7

计算方程, 122, 130

记数法, 34

工程记数法, 34

科学记数法, 34

普通记数法, 34

记忆绘图, 86

绘图工具, 102

在 GRAPH MATH 中, 95

在图象缩放中, 94

加 (+), 267

减法 (-), 363

检查 RAM 屏幕, 230

键, 48

大写字母, 21

第二功能, 21

主要功能, 19, 21, 22

键码图, 217

将华氏温度转换为摄氏温度, 8

交互式求解编辑器, 204

上下限, 204

角度, 71, 175, 185, 269

(用度表示), 51

角度方式, 35, 75, 279

角度值, 35

阶乘 (!), 50, 294

接收传送的数据, 241

结果

保存到变量, 41

显示, 19

绘制, 148

结果, 20, 24

结果方法格式, 133

精确微分

矩阵, 29

创建, 178, 180

从内存中删除, 180

方括号 [], 180, 319

名称, 43

使用 $\boxed{\text{STO}}$ 编辑, 182

使用数学函数, 185

显示元素, 行, 子矩阵, 181

已定义, 178

在表达式中使用, 181

矩阵编辑器菜单, 179

矩阵输入 [], 319

K

开启 TI-86, 2, 17

拷贝变量值, 41

快速查找定位器 (A到Z参考),
262

快速缩放, 91

在参数图象中, 129

在极坐标图象中, 120

L

连接指令, 235

连续输入, 26

链接选项, 234

链接指令, 235

联立方程求解器, 208

M

满光标, 22

忙指示器, 26, 85

命令行, 220

模, 49

N

内插/外推编辑器, 53

内存, 16, 17, 22, 28, 29, 223

复位, 3, 232

可用的, 230

删除项, 231

内存备份

初始化, 237

覆盖警告, 237

内置变量, 39, 45, 138

内置常量, 58

逆, 309

P

平方($\square$), 360

平方根($\sqrt{}$), 7, 360

平移, 90

Q

起始值, 50

切线

绘制, 107

清除 CUSTOM 菜单项, 45

清除 ENTRY 存储区域, 29

求解器 ZOOM 菜单, 208

求解器菜单, 206

求解器图像, 207

求微分方程, 139

求未知变量, 206

曲线

绘图, 107

曲线段的长度, 54

曲线族

图像, 86

在参数图像中, 129

在极坐标图像中, 120

S

上部菜单, 32

选择一个菜单项, 33

上次结果, 28, 29

保存到变量中, 3

上次输入, 26, 28

上次输入, 8

检索, 28

重新使用, 28

重新执行, 19

上下限 $\{[-1E99, 1E99]\}$, 204

上下限, 204

上一个表达式的结果, 26

设置图像格式, 83

设置图像样式, 80

省略号

在矩阵的行中, 179

在行尾, 19

十进制制, 35

十进制, 20

十进制方式, 34, 35, 65

定点 (012345678901), 35

浮点, 35

十进制数, 278

十六进制制, 35, 66

十六进制字符菜单, 67

实数, 29

实数变量, 43

实数的整数部分

显示, 6

授权信息, 400

输入

保存, 29

执行, 19

输入 CBLGET, 216

输入光标, 18, 22, 23

数据类型选择屏幕, 42

数学函数, 48

与矩阵一起使用, 185

与数组一起使用, 161

数值导数, 54

数值微分, 36

数制, 65

标志符, 65

范围, 66

方式, 35

类型, 标志, 67

数制类型符号, 67

数字

输入, 19

数组, 29, 43, 52

保存, 154

比较, 163

编辑元素, 166

插入, 157

拆离公式, 166

创建, 157

从内存中删除, 154

从数组编辑器中删除, 158

括号 {}, 316

连接公式, 162, 166

删除元素, 158

使用, 152

- 显示数组元素, 154
- 相连的公式, 165
- 与函数一起使用, 5
- 在表达式中输入, 153
- 作为参数, 161
- 数组编辑器, 31, 67, 156, 188
 - 连接公式, 163, 164
 - 删除数组, 158
- 数组编辑器菜单, 156
- 数组名, 43
- 数组输入[], 316
- 数组元素
 - 复数, 156
 - 删除, 158
- 数组元素 (续)
 - 保存变量 155
 - 编辑, 158
 - 显示, 155, 158
- 数组元素总和, 52
- 双曲函数, 51
- 算子
 - 输入, 25
- 随机数, 50
- 缩放窗口变量
 - 保存和调用, 95

T

- 特征符号, 39
- 提示, 22
 - Evalx=, 76
 - Name=, 22, 39, 76
 - Rcl, 42
 - Sto, 212
- 统计分析, 188
 - 结果, 192
- 统计数据
 - 绘制, 194, 195
 - 输入, 189
- 统计图
 - 打开和关闭, 195
 - 改变开/关状态, 81
 - 设置, 195
- 图象, 75
 - 定义, 74
 - 曲线族, 86
 - 停止, 85
 - 显示, 85
 - 修改, 85
 - 阴影, 104
 - 暂停, 85
 - 中断, 26
- 图象窗口的大小, 75

- 图象方式, 35
 - 参数, 126
 - 函数,
 - 极坐标, 35, 117
 - 设置, 74
 - 微分方程, 144
- 图象格式
 - 参数图象, 128
 - 极坐标图象, 119
 - 屏幕, 76
 - 设置, 83
 - 微分方程, 133, 137
- 图象工具
 - 在参数图象中, 128
 - 在方程求解器中, 207
 - 在极坐标图象中, 119
 - 在微分方程图象中, 144
- 图象精确度, 89
- 图象屏幕, 75
 - 设置窗口变量, 81
- 图象数据库 (GDB), 102
 - 调用, 76
- 图象缩放
 - 定义屏幕, 92
 - 定制定义, 93
 - 放大, 92, 93

- 记忆绘图, 94
- 设置缩放因子, 93
- 缩小, 92, 93
- 图象样式, 79
 - GrStl(), 302
 - 改变, 10
 - 设置, 79
- 图形
 - 保存, 102
 - 调用, 102
 - 清除, 103
- 退出菜单, 6, 33

W

- 网格点, 84
- 网格点打开, 84, 302
- 网格点关闭, 84, 301
- 微分方程
 - DrEqu(), 287
 - EXPLR, 148
 - Q'n 方程变量, 135
 - 编辑器, 134
 - 变为一阶微分方程组, 142
 - 初始条件编辑器, 136
 - 窗口变量, 135
 - 定义图象, 132

方式, 144
 跟踪, 144
 绘制结果, 148
 求解, 139
 设置图象方式, 132
 设置图象格式, 133
 设置坐标轴, 137
 使用 EVAL, 150
 图象, 132, 137, 139, 141, 142

微分方程编辑器, 134

微分方程图象, 74

方式, 35
 绘图, 145
 显示, 138

微分方式, 36

微积分函数, 54

未计算的表达式
 保存, 9, 40

未知变量
 求解, 206

X

下部菜单, 32

显示, 17

显示菜单, 31

显示对比度

调整, 17, 18

显示结果的记数法, 20

相等(==), 291

相连公式:

求解错误, 165

执行, 164

相连公式数组

比较, 163

编辑元素, 166

创建, 162

箱线图(统计图), 272

向量, 29

编辑维数和元素, 172

创建, 170

从内存中删除, 170

方括号[], 369

复数, 171, 180

显示, 171

形式, 168

已定义的, 168

与数学函数一起使用, 176

运算, 173

在表达式中使用, 172

向量编辑器, 168

向量编辑器菜单, 170

向量输入[], 369

向量坐标方式, 36

项的排列, 50

像素分辨率

对于函数图象, 81

小数点, 35

小写字母光标, 22

小于(<), 312

小于等于($\leq$), 312

选择光标, 38

Y

以 x 为变量求函数值, 101

异或(布尔), 69, 370

阴影

分辨率, 104

图案, 104

阴影图案, 80

用户创建的常量, 43, 58, 60

用户创建的缩放变量, 239

用户支持, 392

与(布尔), 69, 268

语法错误, 27

域格式, 134

元素

矩阵, 181

圆

绘制, 106

圆括号, 20, 25, 56, 61, 397

运算

第二, 2

运算表, 110

定位, 111

清除, 114

设置, 113

设置编辑器, 113

显示, 110

运算表菜单, 112

运算表设置编辑器, 113

运算次序, 56

运算次序规则, 20, 62

运行程序, 221

蕴涵乘法, 397

Z

暂停(PRG CTL 菜单), 219

暂停, 26, 333

暂停指示器, 26

整数部分, 49

正态, 34, 324

正弦

计算, 3
直角向量坐标系, 36
直角坐标的复数形式, 20
直角坐标复数, 70
直角坐标复数方式, 35
直角坐标图象, 84
直线
 绘图, 107
值, 24, 25, 29
指令, 25
 输入, 25
 执行, 19
指令的语法, 25
指令序列
 显示, 18
指数 (Σ), 292
中断 (程序), 222
中断程序, 222
中断计算, 26
中断图象, 26, 27
主屏幕, 17, 18, 23, 24, 26, 27
 显示输入和结果, 18
转至 (PRGM CTL 菜单), 219, 224
转置 (T), 367
子程序, 224
子矩阵
 显示, 181
字符, 19
 第二, 22
 大小写, 22
 黄色, 21
 蓝色, 21, 22
 删除, 23
 输入, 21
 字母, 22
字符串, 29
 保存, 226, 227
 创建, 226
 合并, 226
 已定义, 226
字符串输入, 363
字母锁, 22, 44
 取消, 22
 设置, 22
自动回归方程的存储, 191
自然对数, 48
自由移动光标, 84, 144
 参数图象, 128
 极坐标图象, 119
最大字符数, 22

作图工具, 101
坐标关闭, 84, 275
坐标开启, 84, 275
坐标轴, 137
 域格式, 137
坐标轴编辑器, 137
坐标轴关闭, 84, 271
坐标轴开启, 84, 271