Important

Texas Instruments makes no warranty, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis.

In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the purchase price of this equipment. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

US FCC Information Concerning Radio Frequency Interference

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference with radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, you can try to correct the interference by one or more of the following measures:

♦ Reorient or relocate the receiving antenna.
♦ Increase the separation between the equipment and receiver.
♦ Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
♦ Consult the dealer or an experienced radio/television technician for help.
# Table of Contents

**TI-86 Quick Start**

- Preparing to Use Your New TI-86 ..................................................... 1
- Installing the AAA Batteries ......................................................... 2
- Turning On and Turning Off the TI-86 ............................................. 2
- Adjusting the Contrast .................................................................. 2
- Resetting All Memory and Defaults .............................................. 3
- Calculating on the Home Screen ................................................. 3
- Calculating the Sine of a Number .................................................. 3
- Storing the Last Answer to a Variable ......................................... 3
- Using a Variable in an Expression ................................................ 4
- Editing an Expression .................................................................. 4
- Displaying a Complex Number as a Result .................................... 5
- Using a List with a Function .......................................................... 5
- Displaying the Integer Part of Real Numbers in a List .................. 6
- Removing (Exiting) a Menu ............................................................ 6
- Finding the Square Root ................................................................. 7
- Calculating Derivatives ................................................................ 7
- Retrieving, Editing, and Re-evaluating the Previous Entry ........... 8
- Converting Degrees Fahrenheit to Degrees Celsius ..................... 8
- Storing an Unevaluated Expression to an Equation Variable ....... 9
- Plotting Functions on the Graph Screen ...................................... 9

**Displaying and Entering Functions in the Equation Editor** ........ 9
**Changing the Graph Style of a Function** .................................. 10
**Plotting a Function on the Graph Screen** ................................ 11
**Tracing a Function** .................................................................... 11
**Evaluating y for a Specific x Value (During a Trace)** ............... 12
**Changing a Window Variable Value** .......................................... 12
**Deselecting a Function** ............................................................... 13
**Zooming In on a Portion of the Graph Screen** ......................... 14

**Chapter 1: Operating the TI-86**

- Installing or Replacing Batteries ................................................. 15
- When to Replace Batteries ............................................................. 16
- Turning On and Turning Off the TI-86 .......................................... 17
- Adjusting the Display Contrast ...................................................... 17
- The Home Screen ....................................................................... 18
- Displaying Entries and Answers .................................................. 18
- Entering Numbers ...................................................................... 19
- Entering Negative Numbers ........................................................ 19
- Using Scientific or Engineering Notation .................................... 20
- Entering Complex Numbers ......................................................... 20
- Entering Other Characters ............................................................ 21
- The 2nd Key .............................................................................. 21
The ALPHA Key................................................................. 21
ALPHA-lock and alpha-lock................................................ 22
Common Cursors............................................................ 22
Cursor Direction Keys....................................................... 23
Inserting, Deleting, and Clearing Characters......................... 23
Entering Expressions and Instructions.................................... 24
Entering an Expression....................................................... 24
Using Functions in Expressions.......................................... 25
Using an Instruction.......................................................... 25
Entering Functions, Instructions, and Operators..................... 25
Entering Consecutive Entries.............................................. 26
The Busy Indicator............................................................ 26
Interrupting a Calculation or Graph.................................... 26
Diagnosing an Error.......................................................... 27
Correcting an Error........................................................... 27
Reusing Previous Entries and the Last Answer....................... 28
Retrieving the Last Entry................................................... 28
Retrieving and Editing the Last Entry.................................... 28
Retrieving Previous Entries............................................... 28
Retrieving Multiple Entries............................................... 29
Clearing the ENTRY Storage Area....................................... 29
Retrieving the Last Answer................................................ 29
Using Ans Preceding a Function.......................................... 30
Storing Results to a Variable............................................. 30
Using TI-86 Menus.......................................................... 31
Displaying a Menu.......................................................... 31
The Menu Keys............................................................... 32
Selecting a Menu Item...................................................... 32
Exiting (Removing) a Menu................................................ 33
Viewing and Changing Modes............................................ 34
Changing a Mode Setting.................................................. 34

Chapter 2:
The CATALOG, Variables, and Characters 37
The CATALOG................................................................. 38
Storing Data to Variables.................................................. 39
Creating a Variable Name.................................................. 39
Storing a Value to a Variable Name..................................... 40
Storing an Unevaluated Expression.................................... 40
Storing an Answer............................................................ 41
Copying a Variable Value................................................... 41
Displaying a Variable Value................................................ 41
Recalling a Variable Value.................................................. 42
Classifying Variables as Data Types..................................... 42
The CATLG-VARS (CATALOG-Variables) Menu.................... 43
Selecting a Variable Name.................................................. 44
The CUSTOM Menu.......................................................... 44
Entering CUSTOM Menu Items.......................................... 44
Clearing CUSTOM Menu Items.......................................... 45
Deleting a Variable from Memory....................................... 45
Chapter 3:
Math, Calculus, and Test Operations

Keyboard Mathematical Functions

The MATH Menu

The MATH NUM (Number) Menu

The MATH PROB (Probability) Menu

The MATH ANGLE Menu

The MATH HYP (Hyperbolic) Menu

The MATH MISC (Miscellaneous) Menu

The Interpolate/Extrapolate Editor

The CALC (Calculus) Menu

The TEST (Relational) Menu

Using Tests in Expressions and Instructions

Chapter 4:
Constants, Conversions, Bases, and Complex Numbers

Using Built-In and User-Created Constants

The CONS (Constants) Menu

The CONS BLTIN (Built-In Constants) Menu

Creating or Redefining a User-Created Constant

The Constant Editor Menu

Entering a Constant Name in an Expression

Converting Units of Measure

Converting a Unit of Measure

The CONV (Conversions) Menu

The CONV LNGTH (Length) Menu

The CONV AREA Menu

The CONV VOL (Volume) Menu

The CONV TIME Menu

The CONV TEMP (Temperature) Menu

The CONV MASS Menu

The CONV FORCE Menu

The CONV PRESS (Pressure) Menu

The CONV ENRGY (Energy) Menu

The CONV POWER Menu

The CONV SPEED Menu

Converting a Value Expressed as a Rate

Number Bases

Number Base Ranges

One’s and Two’s Complements

The (Number) BASE Menu

The BASE A-F (Hexadecimal Characters) Menu

Entering Hexadecimal Digits

The BASE TYPE Menu
The BASE CONV (Conversion) Menu ..........................................68
Converting Number Bases..........................................................68
The BASE BOOL (Boolean) Menu ..............................................68
Results of Boolean Operations...................................................69
The BASE BIT Menu..................................................................69
Using Complex Numbers.........................................................70
Complex Results........................................................................70
Using a Complex Number in an Expression..............................71
The CPLX (Complex Number) Menu.........................................71

Chapter 5: Function Graphing  .............................................73
Defining a Graph...........................................................................74
Setting the Graph Mode.............................................................74
The GRAPH Menu......................................................................75
Using the Equation Editor..........................................................76
The Equation Editor (GRAPH y(x)=) Menu............................76
Defining a Function in the Equation Editor..............................77
Notes about Defining Function Equations...............................78
Selecting Graph Styles...............................................................79
Setting the Graph Style in the Equation Editor.........................80
Using Shading Patterns to Differentiate Functions.....................80
Viewing and Changing On/Off Status of Stat Plots.....................81
Setting the Window Variables....................................................81
Displaying the Window Editor...................................................82
Changing a Window Variable Value..........................................82
Setting Graphing Accuracy with \( \Delta x \) and \( \Delta y \) .................83
Setting the Graph Format...........................................................83
Displaying a Graph.................................................................85
Pausing or Stopping a Graph in Progress.................................85
Modifying a Drawn Graph........................................................85
Graphing a Family of Curves.....................................................86
Smart Graph..............................................................................86

Chapter 6: Graph Tools .........................................................87
Graph Tools on the TI-86..........................................................88
The GRAPH Menu.................................................................88
Using the Free-Moving Cursor..................................................89
Graphing Accuracy...................................................................89
Tracing a Graph ........................................................................90
Stopping and Resuming a Trace..............................................91
Resizing the Graph Screen with ZOOM Operations....................91
The GRAPH ZOOM Menu........................................................91
Defining a Custom Zoom In.....................................................93
Setting Zoom ... Out on a Graph..............................................93
Storing and Recalling Zoom Window Variable Values................95
Using Interactive Math Functions..............................................95
The GRAPH MATH Menu.........................................................95
Settings That Affect GRAPH MATH Operations ......................96
Using ROOT, FMIN, FMAX, or INFLC.................................97
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using ( f(x) ), DIST, or ARC</td>
<td>98</td>
</tr>
<tr>
<td>Using ( dy/dx ) or TANLN</td>
<td>99</td>
</tr>
<tr>
<td>Using ISECT</td>
<td>100</td>
</tr>
<tr>
<td>Using YICPT</td>
<td>100</td>
</tr>
<tr>
<td>Evaluating a Function for a Specified ( x )</td>
<td>101</td>
</tr>
<tr>
<td>Drawing on a Graph</td>
<td>101</td>
</tr>
<tr>
<td>Before Drawing on a Graph</td>
<td>102</td>
</tr>
<tr>
<td>Saving and Recalling Drawn Pictures</td>
<td>102</td>
</tr>
<tr>
<td>Clearing Drawn Pictures</td>
<td>103</td>
</tr>
<tr>
<td>The GRAPH DRAW Menu</td>
<td>103</td>
</tr>
<tr>
<td>Shading Areas of a Graph</td>
<td>104</td>
</tr>
<tr>
<td>Drawing a Line Segment</td>
<td>105</td>
</tr>
<tr>
<td>Drawing a Vertical or Horizontal Line</td>
<td>106</td>
</tr>
<tr>
<td>Drawing a Circle</td>
<td>106</td>
</tr>
<tr>
<td>Drawing a Function, Tangent Line, or Inverse Function</td>
<td>107</td>
</tr>
<tr>
<td>Drawing Freehand Points, Lines, and Curves</td>
<td>107</td>
</tr>
<tr>
<td>Placing Text on a Graph</td>
<td>108</td>
</tr>
<tr>
<td>Turning On or Turning Off Points</td>
<td>108</td>
</tr>
<tr>
<td><strong>Chapter 7: Tables</strong></td>
<td>109</td>
</tr>
<tr>
<td>Displaying the Table</td>
<td>110</td>
</tr>
<tr>
<td>TABLE Menu</td>
<td>110</td>
</tr>
<tr>
<td>The Table</td>
<td>110</td>
</tr>
<tr>
<td>Independent and Dependent Variables in the Table</td>
<td>111</td>
</tr>
<tr>
<td>Navigating the Table</td>
<td>111</td>
</tr>
<tr>
<td>The Table Menus</td>
<td>112</td>
</tr>
<tr>
<td>Setting Up the Table</td>
<td>113</td>
</tr>
<tr>
<td>Viewing and Editing Dependent Variable Equations</td>
<td>114</td>
</tr>
<tr>
<td>Clearing the Table</td>
<td>114</td>
</tr>
<tr>
<td><strong>Chapter 8: Polar Graphing</strong></td>
<td>115</td>
</tr>
<tr>
<td>Preview: Polar Graphing</td>
<td>116</td>
</tr>
<tr>
<td>Defining a Polar Graph</td>
<td>117</td>
</tr>
<tr>
<td>Setting Polar Graphing Mode</td>
<td>117</td>
</tr>
<tr>
<td>The GRAPH Menu</td>
<td>117</td>
</tr>
<tr>
<td>Displaying the Polar Equation Editor</td>
<td>118</td>
</tr>
<tr>
<td>Setting the Graph Screen Window Variables</td>
<td>118</td>
</tr>
<tr>
<td>Setting the Graph Format</td>
<td>119</td>
</tr>
<tr>
<td>Displaying the Graph</td>
<td>119</td>
</tr>
<tr>
<td>Using Graph Tools in Pol Graphing Mode</td>
<td>119</td>
</tr>
<tr>
<td>The Free-Moving Cursor</td>
<td>119</td>
</tr>
<tr>
<td>Tracing a Polar Equation</td>
<td>120</td>
</tr>
<tr>
<td>Moving the Trace Cursor to a ( \theta ) Value</td>
<td>121</td>
</tr>
<tr>
<td>Using Zoom Operations</td>
<td>121</td>
</tr>
<tr>
<td>The GRAPH MATH Menu</td>
<td>122</td>
</tr>
<tr>
<td>Evaluating an Equation for a Specified ( \theta )</td>
<td>122</td>
</tr>
<tr>
<td>Drawing on a Polar Graph</td>
<td>122</td>
</tr>
</tbody>
</table>
### Chapter 9: Parametric Graphing 123
- Preview: Parametric Graphing .................................................. 124
- Defining a Parametric Graph..................................................... 125
  - Setting Parametric Graphing Mode ........................................... 126
  - The GRAPH Menu.................................................................. 126
  - Displaying the Parametric Equation Editor ........................... 126
  - Selecting and Deselecting a Parametric Equation .................. 127
  - Deleting a Parametric Equation ............................................. 127
  - Setting the Graph Screen Window Variables ......................... 127
  - Setting the Graph Format .................................................. 128
  - Displaying the Graph.......................................................... 128
- Using Graph Tools in Param Graphing Mode ......................... 128
  - The Free-Moving Cursor .................................................... 128
  - Tracing a Parametric Function ............................................. 128
  - Moving the Trace Cursor to a t Value.................................. 129
  - Using Zoom Operations .................................................... 129
  - The GRAPH MATH Menu...................................................... 130
  - Evaluating an Equation for a Specified t ............................. 130
  - Drawing on a Parametric Graph .......................................... 130

### Chapter 10: Differential Equation Graphing 131
- Defining a Differential Equation Graph .................................. 132
- Setting Differential Equation Graphing Mode .......................... 132
- The GRAPH Menu.................................................................. 133
- Setting the Graph Format..................................................... 133
- Displaying the Differential Equation Editor ............................ 134
- Setting the Graph Screen Window Variables ............................ 134
- Setting the Initial Conditions .............................................. 135
- Setting the Axes .................................................................. 135
- Differential Equation Graphing Tips ...................................... 136
  - The Built-In Variable fIdPic .............................................. 136
  - Displaying the Graph.......................................................... 138
- Entering and Solving Differential Equations ............................ 138
  - Graphing in SlpFld Format ................................................. 139
  - Transforming an Equation into a First-Order System .......... 140
  - Graphing in DirFld Format .................................................. 141
  - Graphing a System of Equations in FldOff Format.............. 142
  - Solving a Differential Equation for a Specified Value .......... 144
- Using Graph Tools in DifEq Graphing Mode ......................... 144
  - The Free-Moving Cursor .................................................... 144
  - Tracing a Differential Equation ........................................... 144
  - Moving the Trace Cursor to a t Value.................................. 145
  - Drawing on a Differential Equation Graph .......................... 145
  - Drawing an Equation and Storing Solutions to Lists ............ 145
  - Using ZOOM Operations ................................................... 147
  - Drawing Solutions Interactively with EXPLR ....................... 148
  - Evaluating Differential Equations for a Specified t .......... 150
Chapter 11: Lists  151
Lists on the TI-86 ................................................................. 152
The LIST Menu ..................................................................... 152
The LIST NAMES Menu ...................................................... 153
Creating, Storing, and Displaying Lists ............................. 153
Entering a List Directly in an Expression ......................... 153
Creating a List Name by Storing a List ............................. 154
Displaying List Elements Stored to a List Name ............... 154
Displaying or Using a Single List Element ....................... 155
Storing a New Value to a List Element ............................. 155
Complex List Elements ...................................................... 156
The List Editor .................................................................... 156
The List Editor Menu .......................................................... 156
Creating a List Name in the Unnamed Column ................. 157
Inserting a List Name into the List Editor ....................... 157
Displaying and Editing a List Element ......................... 158
Deleting Elements from a List ........................................ 158
Removing a List from the List Editor ............................. 158
Using List Operations ....................................................... 159
The LIST OPS (Operations) Menu ................................. 159
Using Mathematical Functions with Lists ....................... 161
Attaching a Formula to a List Name .............................. 162
Comparing an Attached List with a Regular List .............. 163
Using the List Editor to Attach a Formula ....................... 163
Using the List Editor With Attached-Formula Lists .......... 164
Executing and Displaying Attached Formulas ................. 164
Handling Errors Related to Attached Formulas .......... 164
Detaching a Formula from a List Name ......................... 165
Editing an Element of a Attached Formula List ............. 166

Chapter 12: Vectors  167
Vectors on the TI-86 ............................................................. 168
Creating, Storing, and Displaying Vectors ....................... 169
The VECTR (Vector) Menu .............................................. 169
The VECTR NAMES Menu ............................................... 169
Creating a Vector in the Vector Editor ......................... 169
The Vector Editor Menu .................................................. 170
Creating a Vector on the Home Screen ......................... 170
Creating a Complex Vector ........................................... 171
Displaying a Vector ......................................................... 171
Using a Vector in an Expression ....................................... 172
Editing Vector Dimension and Elements ....................... 172
The VECTR MATH Menu ............................................... 173
The VECTR OPS (Operations) Menu ......................... 173
The VECTR CPLEX (Complex) Menu ......................... 175
Using Mathematical Functions with Vectors ................. 176

Chapter 13: Matrices  177
Matrices on the TI-86 ............................................................. 178
Creating, Storing, and Displaying Matrices ..................... 178
# Ti-86 Table of Contents

<table>
<thead>
<tr>
<th>Chapter 14: Statistics</th>
<th>187</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Analysis on the TI-86</td>
<td>188</td>
</tr>
<tr>
<td>Setting Up a Statistical Analysis</td>
<td>188</td>
</tr>
<tr>
<td>The STAT (Statistics) Menu</td>
<td>189</td>
</tr>
<tr>
<td>Entering Statistical Data</td>
<td>189</td>
</tr>
<tr>
<td>The LIST NAMES Menu</td>
<td>189</td>
</tr>
<tr>
<td>The STAT CALC (Calculations) Menu</td>
<td>189</td>
</tr>
<tr>
<td>Automatic Regression Equation Storage</td>
<td>191</td>
</tr>
<tr>
<td>Results of a Statistical Analysis</td>
<td>192</td>
</tr>
<tr>
<td>The STAT VARS (Statistical Variables) Menu</td>
<td>192</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15: Equation Solving</th>
<th>201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview: The Equation Solver</td>
<td>202</td>
</tr>
<tr>
<td>Entering an Equation in the Equation-Entry Editor</td>
<td>203</td>
</tr>
<tr>
<td>Setting Up the Interactive-Solver Editor</td>
<td>204</td>
</tr>
<tr>
<td>Entering Variable Values</td>
<td>204</td>
</tr>
<tr>
<td>Controlling the Solution with Bounds and a Guess</td>
<td>204</td>
</tr>
<tr>
<td>Editing the Equation</td>
<td>205</td>
</tr>
<tr>
<td>The Solver Menu</td>
<td>206</td>
</tr>
<tr>
<td>Solving for the Unknown Variable</td>
<td>206</td>
</tr>
<tr>
<td>Graphing the Solution</td>
<td>207</td>
</tr>
<tr>
<td>Solver Graph Tools</td>
<td>207</td>
</tr>
<tr>
<td>The Solver ZOOM Menu</td>
<td>208</td>
</tr>
<tr>
<td>The Simultaneous Equation Solver</td>
<td>208</td>
</tr>
<tr>
<td>Entering Equations to Solve Simultaneously</td>
<td>208</td>
</tr>
<tr>
<td>Storing Equation Coefficients and Results to Variables</td>
<td>210</td>
</tr>
<tr>
<td>The Polynomial Root-Finder</td>
<td>211</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The MATRX (Matrix) Menu</th>
<th>178</th>
</tr>
</thead>
<tbody>
<tr>
<td>The MATRX NAMES Menu</td>
<td>178</td>
</tr>
<tr>
<td>Creating a Matrix in the Matrix Editor</td>
<td>178</td>
</tr>
<tr>
<td>The Matrix Editor Menu</td>
<td>179</td>
</tr>
<tr>
<td>Creating a Matrix on the Home Screen</td>
<td>180</td>
</tr>
<tr>
<td>Creating a Complex Matrix</td>
<td>180</td>
</tr>
<tr>
<td>Displaying Matrix Elements, Rows, and Submatrices</td>
<td>181</td>
</tr>
<tr>
<td>Using a Matrix in an Expression</td>
<td>181</td>
</tr>
<tr>
<td>Editing Matrices in the Matrix Editor</td>
<td>182</td>
</tr>
<tr>
<td>Editing Matrices on the Home Screen</td>
<td>182</td>
</tr>
<tr>
<td>The MATRX MATH Menu</td>
<td>183</td>
</tr>
<tr>
<td>The MATRX OPS (Operations) Menu</td>
<td>184</td>
</tr>
<tr>
<td>The MATRX CPLX (Complex ) Menu</td>
<td>185</td>
</tr>
<tr>
<td>Using Mathematical Functions with Matrices</td>
<td>185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 14: Statistics</th>
<th>187</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Analysis on the TI-86</td>
<td>188</td>
</tr>
<tr>
<td>Setting Up a Statistical Analysis</td>
<td>188</td>
</tr>
<tr>
<td>The STAT (Statistics) Menu</td>
<td>189</td>
</tr>
<tr>
<td>Entering Statistical Data</td>
<td>189</td>
</tr>
<tr>
<td>The LIST NAMES Menu</td>
<td>189</td>
</tr>
<tr>
<td>The STAT CALC (Calculations) Menu</td>
<td>189</td>
</tr>
<tr>
<td>Automatic Regression Equation Storage</td>
<td>191</td>
</tr>
<tr>
<td>Results of a Statistical Analysis</td>
<td>192</td>
</tr>
<tr>
<td>The STAT VARS (Statistical Variables) Menu</td>
<td>192</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15: Equation Solving</th>
<th>201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview: The Equation Solver</td>
<td>202</td>
</tr>
<tr>
<td>Entering an Equation in the Equation-Entry Editor</td>
<td>203</td>
</tr>
<tr>
<td>Setting Up the Interactive-Solver Editor</td>
<td>204</td>
</tr>
<tr>
<td>Entering Variable Values</td>
<td>204</td>
</tr>
<tr>
<td>Controlling the Solution with Bounds and a Guess</td>
<td>204</td>
</tr>
<tr>
<td>Editing the Equation</td>
<td>205</td>
</tr>
<tr>
<td>The Solver Menu</td>
<td>206</td>
</tr>
<tr>
<td>Solving for the Unknown Variable</td>
<td>206</td>
</tr>
<tr>
<td>Graphing the Solution</td>
<td>207</td>
</tr>
<tr>
<td>Solver Graph Tools</td>
<td>207</td>
</tr>
<tr>
<td>The Solver ZOOM Menu</td>
<td>208</td>
</tr>
<tr>
<td>The Simultaneous Equation Solver</td>
<td>208</td>
</tr>
<tr>
<td>Entering Equations to Solve Simultaneously</td>
<td>208</td>
</tr>
<tr>
<td>Storing Equation Coefficients and Results to Variables</td>
<td>210</td>
</tr>
<tr>
<td>The Polynomial Root-Finder</td>
<td>211</td>
</tr>
</tbody>
</table>
Entering and Solving a Polynomial ........................................... 211
Storing a Polynomial Coefficient or Root to a Variable ............ 212

Chapter 16: Programming ........................................................ 213
Writing a Program on the TI-86 ................................................. 214
The PRGM Menu .................................................................... 214
Creating a Program in the Program Editor ............................... 214
The Program Editor Menu ........................................................ 214
The PRGM I/O (Input/Output) Menu ....................................... 215
The TI-86 Key Code Diagram ................................................... 217
The PRGM CTL Menu ............................................................. 218
Entering a Command Line ......................................................... 220
Menus and Screens in the Program Editor ............................... 220
Running a Program ................................................................... 221
Breaking (Interrupting) a Program ........................................... 222
Working with Programs ............................................................ 223
Managing Memory and Deleting a Program ............................ 223
Editing a Program ................................................................... 223
Calling a Program from Another Program .............................. 224
Copying a Program to Another Program Name ....................... 225
Using and Deleting Variables within a Single Program .......... 225
Running an Assembly Language Program .............................. 225
Entering and Storing a String .................................................... 226
The STRNG (String) Menu ...................................................... 227
Creating a String ................................................................... 227

Chapter 17: Memory Management ......................................... 229
Checking Available Memory ...................................................... 230
The MEM (Memory) Menu ...................................................... 230
Deleting Items from Memory .................................................... 231
The MEM DELET (Delete) Menu ............................................. 231
Resetting the TI-86 ............................................................... 232
The MEM RESET (Reset) Menu .............................................. 232
ClrEnt (Clear Entry) ............................................................... 232

Chapter 18: The TI-86 Communication Link ........................... 233
TI-86 Linking Options ............................................................ 234
Linking Two TI-86s ............................................................... 234
Linking a TI-86 and a TI-85 ..................................................... 234
Linking a TI-86 and a CBL 2/CBL or CBR System .................. 234
Linking a TI-86 and a PC or Macintosh ................................. 235
Downloading Programs from the Internet .............................. 235
Connecting the TI-86 to Another Device ................................. 235
The LINK Menu ................................................................. 236
Selecting Data to Send ......................................................... 236
The LINK SEND Menu .......................................................... 236
Initiating a Memory Backup .................................................... 237
Selecting Variables to Send ..................................................... 238
The SEND WIND (Window Variables) Screen ....................... 238
Sending Variables to a TI-85 .................................................. 239
The LINK SND85 (Send Data to TI-85) Menu ........................................ 239
Preparing the Receiving Device........................................................ 240
Transmitting Data ...................................................................... 240
Receiving Transmitted Data ...................................................... 241
  Repeating Transmission to Several Devices .......................... 242
  Error Conditions ................................................................. 242
  Insufficient Memory in Receiving Unit................................ 242

Chapter 19: Applications ................................................................. 243
Using Math Operations with Matrices ......................................... 244
Finding the Area between Curves .............................................. 245
The Fundamental Theorem of Calculus ..................................... 246
Electrical Circuits...................................................................... 248
Program: Taylor Series ............................................................ 250
Characteristic Polynomial and Eigenvalues .............................. 252
Convergence of the Power Series ............................................. 254
Reservoir Problem .................................................................. 256
Predator-Prey Model ................................................................. 258
Program: Sierpinski Triangle .................................................... 260

Chapter 20:
A to Z Function and Instruction Reference ...................................... 261
Quick-Find Locator .................................................................. 262
Alphabetical Listing of Operations ............................................ 266

Appendix ..................................................................................... 379
TI-86 Menu Map ........................................................................ 380
Handling a Difficulty................................................................. 392
Error Conditions ..................................................................... 393
Equation Operating System (EOS™) ........................................ 397
  Implied Multiplication .......................................................... 397
  Parentheses ......................................................................... 397
TOL (The Tolerance Editor) ...................................................... 398
Computational Accuracy ......................................................... 399
Support and Service Information ............................................ 400
  Product Support ................................................................ 400
  Product Service ................................................................ 401
Other TI Products and Services ............................................. 401
Warranty Information ............................................................... 402
  Customers in the U.S. and Canada Only ............................. 402
  Australia & New Zealand Customers Only .......................... 403
  All Customers outside the U.S. and Canada ....................... 404

Index
TI-86 Quick Start

Preparing to Use Your New TI-86 ........................................... 2
Calculating on the Home Screen ........................................... 3
Plotting Functions on the Graph Screen ............................... 9
Preparing to Use Your New TI-86

The brief examples in the TI-86 Quick Start demonstrate some common TI-86 features. Before you begin, you must install the batteries, turn on the calculator, adjust the contrast, and reset the memory and the defaults. Chapter 1 has more details on these topics.

Installing the AAA Batteries

Four AAA batteries are included in the TI-86 retail package. Remove the batteries from the package and install them in the battery compartment on the back of the calculator. Arrange the batteries according to the polarity (+ and −) diagram in the battery compartment.

Turning On and Turning Off the TI-86

To turn on the TI-86, press ▲, which is in the bottom-left corner of the keyboard. You should see the entry cursor ( ¯ ) blinking in the top-left corner of the screen. If you do not see it, adjust the contrast (see below).

To turn off the calculator, press 2nd, and then the key under OFF, which is ON. This guidebook uses brackets ( [ ] ) to express 2nd and ALPHA keystroke combinations. For example, to turn off the TI-86, press 2nd OFF.

Adjusting the Contrast

1. Press and release the yellow 2nd key.
2. Press and hold ▲ or ▼ (above or below the half-shaded circle).
   ♦ To darken the screen contrast, press and hold ▲.
   ♦ To lighten the screen contrast, press and hold ▼.

After about four minutes of inactivity, the TI-86 turns off automatically.

If you release ▲ or ▼ while adjusting the contrast, you must press 2nd again to continue the adjustment.
Resetting All Memory and Defaults
To reset all memory and defaults, press 2nd [MEM] [F3] [F4]. The messages Mem cleared and
Defaults set are displayed on the home screen, confirming that all memory and defaults are
reset. You may need to adjust the contrast after memory and default reset.

Calculating on the Home Screen
To replicate the screens shown in the Quick Start activities, reset all memory and defaults
once before you begin. Before doing an activity, press CLEAR to clear the screen (except
before the entry retrieval and integer-part examples). Otherwise, the screens your TI-86
shows may differ from the screens pictured next to the activities.

Calculating the Sine of a Number
1. Enter the sine function. (CLEAR) SIN
2. Enter a value. You can enter an expression,
   which is evaluated when you press ENTER.
3. Evaluate the problem. The evaluation of the
   expression sine(\(\pi/4\)) is displayed.

Storing the Last Answer to a Variable
1. Paste the store symbol ( \(\rightarrow\) ) to the screen.
   Since a value must precede \(\rightarrow\), but you did
   not enter a value, the TI-86 automatically
   pasted Ans before \(\rightarrow\).
Enter the variable name to which you want to store the last answer. ALPHA-lock is on.

Store the last answer to the variable. The stored value is displayed on the next line.

Using a Variable in an Expression

1. Enter the variable, and then square it.
   - \( V \)
   - \( V \) is squared and displayed.
   - \( V \)

2. Evaluate. The value stored to the variable \( V \) is squared and displayed.
   - \( V \)

Editing an Expression

1. Enter the expression \((25+14)(4-3.2)\).
   - \((25+14)(4-3.2)\)
   - \((25+14)(4-3.2)\)

2. Change 3.2 to 2.3.
   - \((25+14)(4-3.2)\)
   - \((25+14)(4-3.2)\)

3. Move the cursor to the beginning of the expression and insert a value. The insert cursor blinks between 3 and 25.
   - \((25+14)(4-2.3)\)
   - \((25+14)(4-2.3)\)

4. Evaluate. The result is displayed.
   - \((25+14)(4-2.3)\)
   - \((25+14)(4-2.3)\)

When ALPHA-lock is on and you press a key, the letters printed in blue above the keys are pasted to the screen. In the example, press \( \text{Z} \) to enter a \( V \).

You need not move the cursor to the end of the line to evaluate the expression.
Displaying a Complex Number as a Result

1. Enter the natural log function.
2. Enter a negative number.
3. Evaluate. The result is displayed as a complex number.

Using a List with a Function

1. Enter the exponential function.
2. Display the LIST menu, and then select the open brace ( { ) from the LIST menu.
   On the TI-86, { specifies the beginning of a list.
3. Enter the list elements. Separate each element from the next with a comma.
4. Select the close brace ( } ) from the LIST menu to specify the end of the list.
5. Evaluate. The results of the constant $e$ raised to the 5th, 10th, and 15th powers are displayed as list elements.
Displaying the Integer Part of Real Numbers in a List

1. Display the MATH menu. (The MATH menu automatically replaces the LIST menu from the last activity.)

2. Select NUM to display the MATH NUM menu. The MATH menu shifts up.

3. Select the iPart (integer part) function from the MATH NUM menu. iPart is pasted to the screen. (The previous entry was left on the screen to illustrate the effect of iPart on the previous answer.)

4. Paste Ans to the cursor location. (The result list from the previous activity is stored to Ans.)

5. Display the integer part of the result list elements from the previous activity.

Removing (Exiting) a Menu

1. In the previous example, the MATH menu and the MATH NUM menu are displayed (2nd [MATH] [F1]).

2. Remove the MATH NUM menu from the screen.

3. Remove the MATH menu from the screen.
Finding the Square Root

1. Paste the square root function to the screen. 
   \((\text{CLEAR}) \ 2\text{nd} \ \sqrt{}\)
2. Enter a value for which you want to find the square root. 
   \(144\)
3. Evaluate the expression. The square root of \(144\) is displayed.

Calculating Derivatives

1. Display the CALC menu, and then select \(\text{der1}\). 
   \((\text{CLEAR}) \ 2\text{nd} \ \text{CALC} \ F3\)
2. Enter an expression \((x^2)\) with respect to a variable \((x)\) at a given point \((8)\). 
   \(\text{xVAR} \ x^2 \ \text{xVAR} \ 8 \ \text{I}\)
3. Evaluate. The first derivative of \(x^2\) with respect to \(x\) at \(8\) is displayed.
Retrieving, Editing, and Re-evaluating the Previous Entry

1. Retrieve the last entry from the previous example. (The last activity was not cleared.)
2. Edit the retrieved entry.
3. Evaluate. The first derivative of $x^2$ with respect to $x$ at 3 is displayed.

Converting Degrees Fahrenheit to Degrees Celsius

1. Display the CONV menu.
2. Display the CONV TEMP menu. The CONV menu shifts up and TEMP is highlighted.
3. Enter the known measurement. If the measurement is negative, use parentheses. In this example, if you omit parentheses, the TI-86 converts 4°F to about -15.5°C, which it then negates (changes the sign of), returning a positive 15.5°C.
4. Select °F to designate Fahrenheit as the known measurement unit. °F and the conversion symbol (°) are displayed after the measurement. (continued)
Select °C to designate Celsius as the unit to which you want to convert.

Convert. The °C equivalent of -4°F is displayed.

When storing to an equation variable using =, enter the equation variable first, then =, and then the unevaluated expression. This is the opposite from the order for storing to most other variables on the TI-86.

**Storing an Unevaluated Expression to an Equation Variable**

1. Enter the built-in equation variable y1.
2. Enter the equals sign (=).
3. Enter an expression in terms of x.
4. Store the expression.

The next section shows how to graph the functions y1=5(sin x) and y2=5(cos x).

**Plotting Functions on the Graph Screen**

The TI-86 plots four types of functions on the graph screen. To plot a graph, you must store an unevaluated expression to a built-in equation variable.

Each activity in this section builds upon the activity that precedes it. You must start here and perform the activities in the sequence in which they are presented. The first activity in this section assumes you are continuing from the last activity in the previous section.

**Displaying and Entering Functions in the Equation Editor**

1. Display the GRAPH menu.
In the equation editor, you must express each equation in terms of the independent variable \( x \) (in Func graphing mode only; Chapter 5).

2. Select \( y(x) = \) from the GRAPH menu to display the equation editor. \( 5(\sin x) \) is the unevaluated expression stored to \( y1 \) in the previous activity. The equation editor menu is displayed as the lower menu.

3. Move the cursor down. The \( y2= \) prompt is displayed.

4. Enter the expression \( 5(\cos x) \) at the \( y2= \) prompt. Notice that the equals sign (\( = \)) of \( y2 \) is highlighted after you enter 5. Also, the equals sign of \( y1 \) is highlighted. This indicates that both equations are selected to be graphed (Chapter 5).

Changing the Graph Style of a Function

In the equation editor, the icon to the left of each equation specifies the style in which the graph of that equation appears when you plot it on the graph screen.

1. Move the cursor to \( y1 \).

2. Display the next menu group of the equation editor menu. (\( \uparrow \) at the end of a menu group indicates that the menu has more items.)

3. Select \( \text{STYLE} \) from the equation editor menu to set \( \nabla \) (thick) graph style for \( y1 \).
Plotting a Function on the Graph Screen

1. Select GRAPH from the GRAPH menu to plot the graph on the graph screen. The x- and y-axes and GRAPH menu are displayed. Then each selected graph is plotted in the order in which it is listed in the equation editor.

2. When the graph is plotted, you can move the free-moving cursor ( + ) around the graph screen. The cursor coordinates are displayed at the bottom of the graph.

Tracing a Function

1. Select TRACE from the GRAPH menu to activate the trace cursor, with which you can trace along the graph of any selected function. The number of the current function (the 1 in y1) is displayed in the top-right corner.

2. Move the trace cursor from the function y1 to the function y2. The 1 in the top-right corner changes to 2; the y value changes to the value of y2 at x=0. (continued)
3. Trace the function \( y^2 \). As you trace, the displayed \( y \) value is the solution for \( 5(\cos x) \) at the current \( x \) value, which also is displayed on the screen.

**Evaluating \( y \) for a Specific \( x \) Value (During a Trace)**

1. Enter a real number (or an expression that resolves to a real number) that is within the dimensions of the current graph screen. When you enter the first character, the \( x= \) prompt is displayed.

2. Evaluate \( y^2 \) at \( x=6 \). The trace cursor moves directly to the solution. The \( y \) value, or solution of the equation at \( x \), is displayed on the screen.

**Changing a Window Variable Value**

1. Display the GRAPH menu.

2. Select WIND from the GRAPH menu to display the window editor.

*The window variables values determine the dimensions of the graph screen.*

(continued)
3. Change the value stored in the \texttt{xMin} window variable to 0.

4. Plot the graph on the redefined graph screen. Since \textit{xMin}=0, only the first and fourth quadrants of the graph plane are displayed.

**Deselecting a Function**

1. Select \texttt{y(x)=} from the \texttt{GRAPH} menu to display the equation editor and equation editor menu. The \texttt{GRAPH} menu shifts up and \texttt{y(x)=} is highlighted.

2. Select \texttt{SELCT} from the equation editor menu to deselect the function \texttt{y1=}. The equals sign is no longer highlighted.

3. Plot the graph on the graph screen. Since you deselected \texttt{y1}, the TI-86 only plots \texttt{y2}. To select a function in the equation editor, repeat these steps. (\texttt{SELCT} both selects and deselects functions.)
Zooming In on a Portion of the Graph Screen

1. Select **ZOOM** to display the GRAPH ZOOM menu. The GRAPH menu shifts up and **ZOOM** is highlighted.

2. Select **BOX** from the GRAPH ZOOM menu to activate the zoom-box cursor.

3. Move the zoom-box cursor to a point that is to be a corner of the redefined graph screen, and then mark the point with a small square.

4. Move the cursor away from the small square to a point that is to be the opposite corner of the redefined graph screen. As you move the cursor, a rectangle is drawn on the graph.

5. Zoom in on the graph. The window variables change automatically to the specifications of the zoom box.

6. Clear the menus from the graph screen.
1 Operating the TI-86

Installing or Replacing Batteries ........................................ 16
Turning On and Turning Off the TI-86 ............................... 17
Adjusting the Display Contrast .......................................... 17
The Home Screen .......................................................... 18
Entering Numbers ....................................................... 19
Entering Other Characters ........................................... 20
Entering Expressions and Instructions .............................. 24
Diagnosing an Error ......................................................... 27
Reusing Previous Entries and the Last Answer ............... 28
Using TI-86 Menus .......................................................... 31
Viewing and Changing Modes ........................................... 34
Installing or Replacing Batteries

Your new TI-86 includes four AAA alkaline batteries. You must install them before you can turn on the calculator. A lithium backup battery is installed in the calculator already.

1. If the calculator is on, turn it off (press \[2nd\] [OFF]) to avoid loss of information stored in memory.
2. Slide the protective cover over the keyboard.
3. Holding the calculator upright, push down on the battery cover latch, and then remove the cover.
4. Remove all four old batteries.
5. Install four new AAA alkaline batteries, arranged according to the polarity (+ and –) diagram inside the battery compartment.
6. Replace the battery cover by inserting the two prongs into the two slots at the bottom of the battery compartment, and then push the cover until the latch snaps closed.

When to Replace Batteries

When the AAA batteries are low, a low-battery message is displayed as you turn on the calculator. Generally, the calculator will continue to operate for one or two weeks after the low-battery message is first displayed. Eventually, the TI-86 will turn off automatically and will not operate until you replace the AAA batteries.

The lithium backup battery is inside the battery compartment, above the AAA batteries. It retains all memory when the AAA batteries are low or have been removed. To avoid loss of data, do not remove the lithium battery unless four fresh AAA batteries are installed. Replace the lithium backup battery about every three or four years.
To replace the lithium backup battery, remove the battery cover and unscrew the tiny screw holding the BACK UP BATTERY cover in place. Install a new CR1616 or CR1620 battery according to the polarity (+ and -) diagram on the cover. Replace the cover and screw.

**Turning On and Turning Off the TI-86**

To turn on the TI-86, press \[^\].

- If you previously had turned off the calculator by pressing \[2nd\][OFF], the TI-86 clears any errors and displays the home screen as it was last displayed.
- If Automatic Power Down™ (APD™) previously had turned off the calculator, the TI-86 will return as you left it, including the display, cursor, and any error.

To turn off the TI-86 manually, press \[2nd\][OFF]. All settings and memory contents are retained by the Constant Memory™ feature. Any error condition is cleared.

APD turns off the TI-86 automatically after about four minutes of non-use to extend battery life.

**Adjusting the Display Contrast**

1. Press and release the yellow \[2nd\] key.
2. Press and hold \[\(\uparrow\)\] or \[\(\downarrow\)\] (above or below the half-shaded circle).
   - To darken the screen contrast, press and hold \[\(\uparrow\)\].
   - To lighten the screen contrast, press and hold \[\(\downarrow\)\].

Properly dispose of the old battery.
You can adjust the display contrast anytime to suit your viewing angle and lighting conditions. As you adjust, a number from 0 (lightest) to 9 (darkest) in the top-right corner indicates the current contrast setting. The number is not visible when the contrast is extremely light or dark.

As the batteries weaken over time, the actual contrast level of each number shifts. For example, say you set the contrast to 3 with fresh batteries. As the batteries weaken, you will need to set the contrast to 4, then 5, then 6, and so on, to retain the original contrast level. However, you need not replace the batteries until the low-battery message is displayed.

The Home Screen

When you first turn on your TI-86, the home screen is displayed. Initially, the home screen is a blank screen, except for the entry cursor (■) in the top-left corner. If you do not see the cursor, press 2nd, and then press and hold ▼ or ▲ to adjust the contrast (page 17).

On the home screen, you can enter and evaluate expressions, and view the results. You also can execute instructions, store and recall variable values, and set up graphs and editors.

To return to the home screen from any other screen, press 2nd [QUIT].

Displaying Entries and Answers

The home screen displays up to eight lines with a maximum of 21 characters per line. If an expression or series of instructions exceeds 21 characters and spaces, it automatically continues on the next line.

After all eight lines are full, text scrolls off the top of the display. You can press ▲ to scroll up the home screen, only as far as the first character in the current entry. To retrieve, edit, and re-execute previous entries, use 2nd [ENTRY] (page 28).
When an entry is executed on the home screen, the answer is displayed on the right side of the next line. When you execute an instruction, Done is typically displayed on the right side of the next line.

If an answer is too long to display on the screen, an ellipsis (...) is displayed, initially to the right. To view more of the answer, press \( \text{[\text{Up]} \text{Arrow]} \). When you do, an ellipsis is displayed to the left. To scroll back, press \( \text{[\text{Down]} \text{Arrow]} \).

**Entering Numbers**

A symbol or abbreviation of each key’s primary function is printed in white on the key. For example, when you press \( \text{[+] \text{Symbol]} \), a plus sign is pasted to the cursor location. This guidebook describes number-entry keystrokes as \( 1, 2, 3 \), and so on, instead of \( \text{Y} \text{Z} \text{[\text{Symbol}]} \).

**Entering Negative Numbers**

To enter a negative number, press \( \text{[\text{Negate]} \text{Key]} \), and then press the appropriate number keys. For example, to enter \(-5\), press \( \text{[Negate]} 5 \). Do not attempt to express a negative number using \( \text{[\text{Subtract]} \text{Key]} \), \( \text{[Negate]} \) and \( \text{[Subtract]} \) are two different keys with different uses.

The order in which the TI-86 evaluates negation and other functions within an expression is governed by the Equation Operating System™ (Appendix). For example, the result of \(-4^2\) is \(-16\), while the result of \((-4)^2\) is \(16\). If you are unsure about the order of evaluation, use \( \text{[\text{Negate]} \text{Symbol]} \) and \( \text{[\text{Subtract]} \text{Symbol]} \) to clarify the intended use of the negation symbol.
Chapter 1: Operating the TI-86

Using Scientific or Engineering Notation

1. Enter the mantissa (part of the number that precedes the exponent). This value can be an expression.

2. Paste \( \times 10^n \) to the cursor location.

3. If the exponent is negative, paste \( \times -10^n \) to the cursor location. Then enter a one-, two-, or three-digit exponent.

4. Evaluate the expression.

When you include scientific- or engineering-notation numbers in an expression, the TI-86 does not necessarily display answers in scientific or engineering notation. The mode settings (page 34) and the size of the number determine the notation of displayed answers.

Entering Complex Numbers

On the TI-86, the complex number \( a+bi \) is entered as \((a,b)\) in rectangular complex-number form or as \((r \angle \theta)\) in polar complex-number form. For more information about complex numbers, read Chapter 4.

Entering Other Characters

This is the 2nd key

This is the ALPHA key
Chapter 1: Operating the TI-86

The 2nd Key

The 2nd key is yellow. When you press 2nd, the cursor becomes \( \text{˘} \) (the 2nd cursor). When you press the next key, the yellow character, abbreviation, or word printed above that key is activated, instead of the key’s primary function.

\[ \text{2nd} \ [\text{STAT}] \text{ returns the STAT menu} \]

\[ \begin{array}{c}
\text{STAT} \\
\text{X}
\end{array} \]

The ALPHA Key

The [ALPHA] key is blue. When you press [ALPHA], the cursor becomes \( \text{‡} \) (the uppercase ALPHA cursor). When you press the next key, the blue uppercase character printed above that key is pasted to the cursor location.

\[ \begin{array}{c}
\text{STAT} \\
\text{X}
\end{array} \]

\[ \text{ALPHA} [\text{X}] \text{ returns an } x \]

When you press 2nd [alpha], the cursor becomes \( \text{˘} \) (the lowercase alpha cursor). When you press the next key, the lowercase version of the blue character is pasted to the cursor location.

ALPHA-lock and alpha-lock

To enter more than one uppercase or lowercase alpha character consecutively, set ALPHA-lock (for uppercase letters) or alpha-lock (for lowercase letters).

To set ALPHA-lock when the entry cursor is displayed, press [ALPHA] [ALPHA].

♦ To cancel ALPHA-lock, press [ALPHA].
♦ To switch from ALPHA-lock to alpha-lock, press 2nd [alpha].

To set alpha-lock when the entry cursor is displayed, press 2nd [alpha] [ALPHA].

♦ To cancel alpha-lock, press [ALPHA] [ALPHA].
To switch from alpha-lock to ALPHA-lock, press [ALPHA].

You can use [2nd] when ALPHA-lock or alpha-lock is on. Also, if you press a key that has no blue character above it, such as [GRAPH], [DEL], or [†], the key’s primary function still applies.

**Common Cursors**

- **Entry**    |  Enters a character at the cursor, overwriting any existing character
- **Insert**   |  Inserts a character at the cursor location and shifts remaining characters right
- **Second**   |  Enters a 2nd character or executes a 2nd operation (yellow on the keyboard)
- **ALPHA**    |  Enters an uppercase ALPHA character (blue on the keyboard)
- **alpha**    |  Enters the lowercase version of an ALPHA character (blue on the keyboard)
- **Full**     |  Accepts no data; maximum characters are entered at a prompt or memory is full

- If you press [ALPHA] after [2nd] [INS], the cursor becomes an underlined A (A).
- If you press [2nd] [ALPHA] after [2nd] [INS], the cursor becomes an underlined a (a).
- If you press [2nd] after [2nd] [INS], the insert cursor becomes an underlined ↑ (↑).
Cursor Direction Keys

- moves cursor left
- moves cursor to beginning of entry
- scrolls/moves cursor down
- lightens screen contrast
- moves cursor right
- moves cursor to end of entry
- scrolls/moves cursor up
- darkens screen contrast

If you hold down [!, #, !], or $, the cursor continues to move.

Inserting, Deleting, and Clearing Characters

- Changes the cursor to the insert cursor (____); inserts characters at the insert cursor and shifts remaining characters right; to cancel insert, press 2nd [INS] or press [!, #, !], or $
- Deletes a character at the cursor; to continue deleting to the right, hold down DEL
- Clears the current entry on the home screen; CLEAR CLEAR clears the entire home screen
Entering Expressions and Instructions

Entering an Expression

An expression is any combination of numbers and variables that serve as arguments for one or more functions. On the TI-86, you typically enter an expression in the same order as you would write it on paper. For example, $\pi^2$, $5 \tan \text{Stat}$, and $40(5+3)-(2+3)$ are expressions.

You can use an expression on the home screen to calculate an answer.

In most places where a value is required, you can use an expression to enter the value.

For example, enter an expression as a window variable value (Chapter 5). When you press #, $, b$, or , the TI-86 evaluates the expression and replaces it with the result.

To enter an expression, you enter numbers, variables, and functions from the keyboard and menus (page 31). When you press ENTER, the expression is evaluated (regardless of the cursor location) according to EOS order-of-evaluation rules (Appendix), and the answer is displayed.

To enter the expression $3.76 \div (7.9 + \sqrt{5}) + 2 \log 45$ and then evaluate it, you would press these keys:

\[
\begin{align*}
3 & \ 76 \ \div \ 7 \ 9 \ \ 2nd \ \sqrt \ 5 \ \ 2 \ \ LOG \ 45 \ \ ENTER
\end{align*}
\]
Using Functions in Expressions
A function returns a value. Some examples of functions are +, -, +, √, and log. To use functions, you usually must enter one or more valid arguments.

When this guidebook describes the syntax of a function or instruction, each argument is in italics. For example: \( \text{sin} \) angle. Press \( \text{SIN} \) to enter \( \text{sin} \), and then enter a valid angle measurement (or an expression that resolves to angle). For functions or instructions with more than one argument, you must separate each argument from the other with a comma.

Some functions require the arguments to be in parentheses. When you are unsure of the evaluation order, use parentheses to clarify a function’s place within an expression.

Using an Instruction
An instruction initiates an action. For example, ClDrw is an instruction that, when executed, clears all drawn elements from a graph. You cannot use an instruction in an expression.

Generally, the first letter of each instruction name is uppercase on the TI-86. Some instructions take more than one argument, as indicated by an open parenthesis ( ) at the end of the name. For example, \( \text{Circl} \) requires three arguments, \( \text{Circl}(x,y,\text{radius}) \).

Entering Functions, Instructions, and Operators
You can enter a function, instruction, or operator in any of three ways (log 45, for example).

1. Paste it to the cursor location from the keyboard or a menu (log 45).
2. Paste it to the cursor location from the CATALOG (2nd [CATLG-VARS] F1 [L] F1 F1 ENTER 45).
3. Enter it letter by letter ( 2nd [alpha] ALPHA [L] [O] [G] [L] [A] [P] [H] [A] ALPHA [A] [L] [P] [H] [A] [A] 45).

As you can see in the example, using the built-in function or instruction typically is easier.
When you select a function, instruction, or operator, a symbol comprising one or more characters is pasted to the cursor location. Once the symbol is pasted to the cursor location, you can edit individual characters.

For example, assume that you pressed \textbf{\texttt{-/\&\&\&\&b}} to paste \text{yMin} to the cursor location as part of an expression. Then you realized you wanted \text{xMin}. Instead of pressing nine keys to select \text{xMin}, you can simply press \texttt{!2}.

**Entering Consecutive Entries**

To enter two or more expressions or instructions consecutively, separate each from the next with a colon (\texttt{-:}). When you press \texttt{b}, the TI-86 executes each entry from left to right and displays the result of the last expression or instruction. The entire group entry is stored in last entry (page 28).

**The Busy Indicator**

When the TI-86 is calculating or graphing, a moving vertical line is displayed as the busy indicator in the top-right corner of the screen. When you pause a graph or a program, the busy indicator is replaced by the pause indicator, a moving vertical dotted line.

**Interrupting a Calculation or Graph**

To interrupt a calculation or graph in progress, press \texttt{ON}. When you interrupt a calculation, the \text{ERROR 06 BREAK} message and menu are displayed.

- To return to the home screen, select \texttt{QUIT} (press \texttt{F5}).
- To go to the beginning of the expression, select \texttt{GOTO} (press \texttt{F1}). Press \texttt{ENTER} to recalculate the expression.
When you interrupt a graph, a partial graph and the GRAPH menu are displayed.

- To return to the home screen, press [CLEAR CLEAR] or any non-graphing key.
- To restart graphing, select an instruction that displays the graph.

**Diagnosing an Error**

When the TI-86 detects an error, it returns an error message, such as **ERROR 04 DOMAIN** or **ERROR 07 SYNTAX**. The Appendix describes each error type and possible reasons for the error.

- If you select QUIT (or press 2nd [QUIT] or [CLEAR CLEAR]), the home screen is displayed.
- If you select GOTO, the previous screen is displayed with the cursor on or near the error.

**Correcting an Error**

1. Note the error type (**ERROR # errorType**).
2. Select GOTO, if available. The previous screen is displayed with the cursor on or near the error.
3. Determine the cause for the error. If you cannot, refer to the Appendix for possible causes.
4. Correct the error and continue.
Chapter 1: Operating the TI-86

Reusing Previous Entries and the Last Answer

Retrieving the Last Entry
When you press [ENTER] on the home screen to evaluate an expression or to execute an instruction, the entire expression or instruction is placed in a storage area called ENTRY (last entry). When you turn off the TI-86, ENTRY is retained in memory.

To retrieve the last entry, press 2nd [ENTRY]. The current line is cleared and the entry is pasted to the line.

Retrieving and Editing the Last Entry

1. On the home screen, retrieve the previous entry.
2. Edit the retrieved entry.
3. Re-execute the edited entry.

Retrieving Previous Entries
The TI-86 retains as many previous entries as possible in ENTRY, up to a capacity of 128 bytes. To scroll from the newest to the older previous entries stored to ENTRY, repeat 2nd [ENTRY]. If you press 2nd [ENTRY] after displaying the oldest stored entry, the newest stored entry is displayed again; continuing to press 2nd [ENTRY] repeats the order.
Retrieving Multiple Entries

To store two or more expressions or instructions together to ENTRY, enter them on one line, separating each from the other with a colon, and then press [ENTER]. Upon execution, the entire group is stored in ENTRY. The example below shows one of many ways you can manipulate this feature to avoid tedious manual re-entry.

1. Use trial and error to find the radius of a circle with an area of 200 square centimeters. Store \(8\) as your first guess, then execute \(\pi r^2\).

2. Retrieve \(8\pi r^2\) and insert \(7.958\) as a new guess. Continue guessing to approach the answer of \(200\).

Clearing the ENTRY Storage Area

To clear all data from the ENTRY storage area, begin on a blank line on the home screen, select ClrEnt from the MEM menu (press [2nd] [MEM] [F5]), and then press [ENTER].

Retrieving the Last Answer

When an expression is evaluated successfully on the home screen or in a program, the TI-86 stores the answer to a built-in variable called Ans (last answer). Ans may be a real or complex number, list, vector, matrix, or string. When you turn off the TI-86, the value in Ans is retained in memory.
To copy the variable name Ans to the cursor location, press [2nd] [ANS]. You can use the variable Ans anywhere that the value stored to it is valid. When the expression is evaluated, the TI-86 calculates the result using the value stored in Ans.

1. Calculate the area of a garden plot 1.7 meters by 4.2 meters.
2. Calculate the yield per square meter if the plot produces a total of 147 tomatoes.

Using Ans Preceding a Function
Previous answers are stored to Ans. If you begin an expression by entering a function that requires a preceding argument, the TI-86 automatically enters Ans as the argument.

1. Enter and execute an expression.
2. Enter a function without an argument. Ans is pasted to the screen, followed by the function.

Storing Results to a Variable
1. Calculate the area of a circle with radius 5 meters.
2. Calculate the volume of a cylinder of radius 5 meters and height 3.3 meters.
3. Store the result to the variable V.
Using TI-86 Menus

The symbols for many TI-86 features are found in menus instead of on the TI-86 keyboard.

Displaying a Menu

The way to display a particular menu depends on the menu’s location on the TI-86.

Menu-Displaying Method  Example

Press a key that has a menu name on it  [GRAPH] displays the GRAPH menu
Press 2nd and then a 2nd-key menu name  2nd [MATH] displays the MATH menu
Select a menu name from another menu  2nd [MATH] F1 displays the MATH NUM menu
Select an editor or selection screen  2nd [LIST] F2 displays the list editor menu with the list editor
Accidentally commit an error  1 [STO►] ENTER displays the error menu

When you display a menu, a menu group of one to five items is displayed on the bottom of the screen. If the more symbol (►) is displayed after the fifth item in a menu group, the menu continues for at least one more menu group. To view the next menu group, press MORE. The last menu group of one to five items does not have a ► symbol.

For example, press 2nd [MATH] to display the MATH menu.

Some TI-86 menus have as many as 25 items.

1, 2, 3, and 4 do not work on menus.
The Menu Keys

**Upper Menu Keys**
- **M1**
- **M2**
- **M3**
- **M4**
- **M5**

**Lower Menu Keys**
- **1**
- **2**
- **3**
- **4**
- **5**

- **[ Quit]** clears all menus
- **[2nd]** through **[5]** selects upper menu items

Selecting a Menu Item

When you display a menu, one to five items are displayed. To select a menu item, press the menu selection key directly below the item. For example, in the MATH menu to the right, press **&** to select **NUM**, press **'** to select **PROB**, and so on.

When you select a menu item that displays another menu, the first menu moves up one line on the screen to make room for the new menu. All items on the original menu are displayed in reverse type, except the item you selected.

When you select **NUM**...

...the MATH menu moves up and the MATH NUM menu is displayed.

To remove the MATH NUM menu and move the MATH menu down, press **EXIT**.
To select an item from the upper menu, press \[2\text{nd}\] and the appropriate key [M1] through [M5].

To select \texttt{PROB} from the upper menu, press \[2\text{nd}\][M2].
To select \texttt{IPart} from the lower menu, press \[2\text{nd}\][F].

When an editor menu is displayed as the upper menu, and you select an item from the lower menu that displays yet another menu, the editor menu remains as the upper menu.

When you select \texttt{NUM} from the lower menu...
...the equation editor menu remains and the MATH NUM menu is displayed.

**Exiting (Removing) a Menu**

To remove the lower menu from the screen, press \[\text{EXIT}\].

When you press \[\text{EXIT}\]...
...the MATH NUM menu disappears and the MATH menu moves down.
Press \[\text{EXIT}\] again, and the MATH menu disappears.

To remove a menu from the bottom of a graph screen, press \[\text{CLEAR}\] after plotting the graph (Chapter 5).
Viewing and Changing Modes

To display the mode settings, press 2nd [MODE]. The current settings are highlighted. Mode settings control how the TI-86 displays and interprets numbers and graphs. The Constant Memory feature retains current mode settings when the TI-86 is turned off. All numbers, including elements of matrices and lists, are displayed according to the mode settings.

Changing a Mode Setting

1. Move the cursor to the line of the setting that you want to change (decimal setting in the example).
2. Move the cursor to the setting you want (2 decimal places).
3. Execute the change.

Notation Modes

| Normal        | Displays results with digits to the left and right of the decimal (as in 123456.7) |
| Sci (scientific) | Displays results in two parts: significant digits (with one digit to the left of the decimal) are displayed to the left of E and the appropriate power of 10 is displayed to the right of E (as in 1.234567E5) |
| Eng (engineering) | Displays results in two parts: significant digits (with one, two, or three digits to the left of the decimal) are displayed to the left of E and the appropriate power of 10 (which is always a multiple of 3) is displayed to the right of E (as in 123.4567E3) |
Decimal Modes

**Float** (floating) Displays results up to 12 digits, plus any sign and the floating decimal point

**(fixed)** (012345678901; each number is a setting) Displays results with the specified number of
digits to the right of the decimal point (rounds answers to the specified decimal place);
the second 0 sets 10; the second 1 sets 11

Angle Modes

**Radian** Interprets angle values as radians; displays answers in radians

**Degree** Interprets angle values as degrees; displays answers in degrees

Complex Number Modes

**RectC** (rectangular complex mode) Displays complex-number results as *(real,imaginary)*

**PolarC** (polar complex mode) Displays complex-number results as *(magnitude,angle)*

Graphing Modes

**Func** (function graphing) Plots functions where *y* is a function of *x*

**Pol** (polar graphing) Plots functions where *r* is a function of *θ*

**Param** (parametric graphing) Plots relations where *x* and *y* are functions of *t*

**DifEq** (differential equation graphing) Plots differential equations in terms of *t*

Number Base Modes

**Dec** (decimal number base) Interprets and displays numbers as decimal (base 10)

**Bin** (binary number base) Interprets numbers as binary (base 2); displays *b* suffix with answers

**Oct** (octal number base) Interprets numbers as octal (base 8); displays *o* suffix with answers

**Hex** (hexadecimal number base) Interprets numbers as hexadecimal (base 16); displays *h*

suffix with answers

Non-decimal modes are valid only on the home screen or in the program editor.
Vector modes do not affect how you enter vectors.

Vector Coordinate Modes

- **RectV** (rectangular vector coordinates) Displays answers in the form \([x \ y]\) for two-element vectors and \([x \ y \ z]\) for three-element vectors.
- **CylV** (cylindrical vector coordinates) Displays results in the form \([r \ \theta]\) for two-element vectors and \([r \ \theta \ z]\) for three-element vectors.
- **SphereV** (spherical vector coordinates) Displays results in the form \([r \ \theta]\) for two-element vectors and \([r \ \theta \ \phi]\) for three-element vectors.

Differentiation Modes

- **dxDer1** (exact differentiation) Uses `der1` (Chapter 3) to differentiate exactly and calculate the value for each function in an expression (dxDer1 is more accurate than dxNDer, but it restricts the kinds of functions that are valid in the expression).
- **dxNDer** (numeric differentiation) Uses `nDer` to differentiate numerically and calculate the value for an expression (dxNDer is less accurate than dxDer1, but more kinds of functions are valid in the expression).

The value stored to \(\delta\) affects dxNDer (Appendix).
The CATALOG, Variables, and Characters

The CATALOG ................................................................. 38
Storing Data to Variables ............................................. 39
Classifying Variables as Data Types .............................. 42
The CUSTOM Menu ....................................................... 44
The CHAR (Character) Menu ....................................... 45
The CATALOG is the first item on the CATLG-VARS menu.

The CATALOG displays all TI-86 functions and instructions in alphabetical order. Items that do not begin with a letter (such as + or Bin) are at the end of the CATALOG.

The selection cursor (►) indicates the current item. To select an item from the CATALOG, move the selection cursor to the item and press [ENTER]. The CATALOG disappears and the name is pasted to the previous cursor location.

Use ▼ or ▲ to move ► to an item...

To jump...

To the first item beginning with a particular letter Press the letter; ALPHA-lock is on
To special characters at the end of the CATALOG Press ◄ from the first CATALOG item
Down one whole screen Select PAGE↓ from the CATALOG menu (F1)
Up one whole screen Select PAGE↑ from the CATALOG menu (F2)

The menu items CUSTM and BLANK are on the CATALOG menu and each VARS screen menu. With them, you can create and edit your own CUSTOM menu of up to 15 CATALOG items and variables, including program names. For details about the CUSTOM menu, read page 44.
Storing Data to Variables

On the TI-86, data can be stored to variables in several ways. You can:

- Use \texttt{STO\rightarrow} to store a value to a variable.
- Use \texttt{=} to store an unevaluated expression to an equation variable.
- Use an editor's \texttt{Name=} prompt to store several types of data to a variable.
- Change TI-86 settings or reset defaults and memory to the factory settings.
- Execute functions that cause the TI-86 to store data automatically to built-in variables.

The TI-86 has built-in variable names with specific purposes, such as equation variables, list names, statistical result variables, window variables, and \texttt{Ans}. You can store values to some of them. They are introduced in the appropriate chapters of this guidebook.

Creating a Variable Name

You can create your own variable name when you use \texttt{STO\rightarrow}, \texttt{=}, or a \texttt{Name=} prompt to store data. When you create a user-created variable name, follow these guidelines.

- The user-created variable name can be from one to eight characters long.
- The first character must be a letter, which includes all CHAR GREEK menu items, as well as Ñ, ñ, Ç, and ç from the CHAR MISC menu.
- A user-created variable name cannot replicate a TI-86 feature symbol or built-in variable. For example, you cannot create \texttt{abs}, because \texttt{abs} is the absolute value function symbol. You cannot create \texttt{Ans}, because it is already a built-in variable name.
- The TI-86 distinguishes between uppercase and lowercase characters in variable names. For example, \texttt{ANS}, \texttt{Ans}, and \texttt{ans} are three different variable names. Therefore, only \texttt{Ans} is a built-in variable name; \texttt{ANS} and \texttt{ans} can be user-created variable names.
Storing a Value to a Variable Name

1. Enter a value, which can be an expression.
2. Enter \( \Rightarrow \) (the store symbol) next to the value.
3. Create a variable name one to eight characters long, starting with a letter. ALPHA-lock is on.
4. Store the value to the variable. The value stored to the variable is displayed as a result.

Storing an Unevaluated Expression

When you store an expression to memory using \( \Rightarrow \) (with the \( \Rightarrow \) sign), the expression is evaluated and the result is stored to a variable.

When you store an unevaluated expression using [ALPHA] \( \Rightarrow = \), or the equation editor (Chapter 5), or the equation solver (Chapter 15), the unevaluated expression is stored to an equation variable.

To store an unevaluated expression on the home screen or in a program, the syntax is:

\[
\text{variable} = \text{expression}
\]

where \text{variable} always precedes the equals sign and \text{expression} always follows the equals sign.

You can use \( = \) to store a mathematical expression to an equation variable. For example, \( \text{F} = \text{M} \times \text{A} \).
Storing an Answer

To store an answer to a variable before you evaluate another expression, use \texttt{STO\textsuperscript{+}} and \texttt{Ans}.

1. Enter and evaluate an expression.

\begin{verbatim}
\texttt{AREA} \times 3.3
\end{verbatim}

2. Store the answer to a user-created variable or to a valid built-in variable. The value stored to the variable is displayed as a result.

\begin{verbatim}
\texttt{STO\textsuperscript{+} [V] [O] [L] ENTER}
\end{verbatim}

Copying a Variable Value

To copy the contents of \texttt{variableA} into \texttt{variableB}, the syntax is: \texttt{variableA \Rightarrow variableB}

For example, \texttt{RegEq \Rightarrow y1} stores the regression equation (Chapter 14) to the variable \texttt{y1}.

Displaying a Variable Value

1. With the cursor on a blank line on the home screen, paste the variable name to the cursor location, as described above.

\begin{verbatim}
\texttt{\textit{(location may vary) ENTER}}
\end{verbatim}

2. Display the contents of the variable.

You also can display variables containing some data types by displaying them in the appropriate editor (such as the list editor or window variable editor) or graph. These methods are detailed in subsequent chapters of this guidebook.
Recalling a Variable Value

1. Move the cursor to where you want to insert the recalled variable value.
2. Display the Rcl prompt at the bottom of the screen. ALPHA-lock is on.
3. Enter the variable name you want to recall.
4. Recall the variable contents to the cursor location. The Rcl prompt disappears and the entry cursor returns.

Classifying Variables as Data Types

The TI-86 classifies variables according to data type and places each variable on a data-type selection screen. You can display each screen by selecting the appropriate data type from the CATLG-VARS menu, as described on page 43. Here are some examples.

<table>
<thead>
<tr>
<th>If data...</th>
<th>The TI-86 classifies the data type as...</th>
<th>For example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begins with { and ends with }</td>
<td>A list (VARS LIST screen)</td>
<td>{1,2,3}</td>
</tr>
<tr>
<td>Begins with [ and ends with ]</td>
<td>A vector (VARS VECTR screen)</td>
<td>[1,2,3]</td>
</tr>
<tr>
<td>Begins with [[] and ends with ]]</td>
<td>A matrix (VARS MATRX screen)</td>
<td>[[1,2,3][4,5,6][7,8,9]]</td>
</tr>
</tbody>
</table>
The CATLG-VARS (CATALOG-Variables) Menu

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

To display additional menu groups, press [MORE].

The list names fStat, xStat, and yStat are statistical result variables on the VARS STAT screen.

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

Displays the CATALOG
Displays a selection screen with all variables and names of all data types
Displays a selection screen with all real number variables
Displays a selection screen with all complex number variables
Displays a selection screen with all list names
Displays a selection screen with all vector names
Displays a selection screen with all matrix names
Displays a selection screen with all string variables
Displays a selection screen with all equation variables
Displays a selection screen with all user-defined constants
Displays a selection screen with all program names
Displays a selection screen with all graph database names
Displays a selection screen with all picture names
Displays a selection screen with all statistical result variables
Displays a selection screen with all window variables
Selecting a Variable Name

1. Select the appropriate data-type selection screen from the CATLG-VARS menu.
2. Move the cursor to the variable you want to select.
3. Select the variable you want.

The CUSTOM Menu

You can select up to 15 items from the CATLOG and VARS screens—program names, functions, instructions, and other items—to create your own CUSTOM menu. To display your CUSTOM menu, press CUSTOM. Use F1 through F3 and MORE to select items like any other menu.

Entering CUSTOM Menu Items

1. Select CUSTM from the CATLOG. The CUSTOM menu is displayed. ALPHA-lock is on.
2. Move the selection cursor (↑) to the item you want to copy to the CUSTOM menu.
3. Copy the item to the CUSTOM menu cell you select, replacing any previous item.
4. To enter more items, repeat steps 2 and 3 using different items and cells.
5. Display the CUSTOM menu.
Clearing CUSTOM Menu Items

1. Select BLANK from the CATALOG menu. The CUSTOM BLANK menu is displayed.
2. Clear the menu item.
3. To clear more items, repeat steps 2 and 3.

Deleting a Variable from Memory

From the home screen or in a program, to delete from memory one user-created variable name (except a program name) and its contents, the syntax is:

```
DelVar(variable)
```

To delete user-created variable names and their contents (including program names), display the MEM DELET menu (2nd MEM F2), select the data type, select the variable, and then press ENTER (Chapter 16). Deleting a variable does not remove it from the CUSTOM menu (page 44).

The CHAR (Character) Menu

```
2nd [CHAR]
```

- MISC
- GREEK
- INTL

miscellaneous characters menu
international characters menu
Greek characters menu
Chapter 2: The CATALOG, Variables, and Characters

The CHAR MISC (Miscellaneous) Menu

You can combine modifiers on the CHAR INTL menu with uppercase or lowercase vowels to create vowels used in some languages. You can use these vowels in variable names and text.

Adding a Modifier to a Vowel

1. Select the modifier from the CHAR INTL menu. ALPHA-lock is on. If necessary, switch to alpha-lock.
2. Enter the uppercase or lowercase vowel over which you want the modifier.
Math, Calculus, and Test Operations

Keyboard Mathematical Functions ........................................ 48
The MATH Menu ................................................................. 49
The CALC (Calculus) Menu .................................................. 54
The TEST (Relational) Menu ............................................... 55
Keyboard Mathematical Functions

You can use these mathematical functions in expressions with real or complex values. You can use some of them with lists, vectors, matrices, or strings.

When you use lists, vectors, or matrices, the valid functions return a list of results calculated on an element-by-element basis. If you use two lists, vectors, or matrices in the same expression, they must be equal in dimension.

The most common mathematical functions are on the TI-86 keyboard. For syntax, details, and examples of these functions, refer to the A to Z Reference.

\[ x^3 \] (the multiplicative inverse) is equivalent to the reciprocal, \( 1/x \).

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+ (add)</td>
</tr>
<tr>
<td>-</td>
<td>- (subtract)</td>
</tr>
<tr>
<td>*</td>
<td>* (multiply)</td>
</tr>
<tr>
<td>/</td>
<td>/ (divide)</td>
</tr>
<tr>
<td>^</td>
<td>^ (raise to a specified power)</td>
</tr>
<tr>
<td>(2\text{nd}) [\sqrt{}]</td>
<td>(\sqrt{}) (square root)</td>
</tr>
<tr>
<td>(2\text{nd}) [x^{-2}]</td>
<td>(x^{-2}) (inverse)</td>
</tr>
<tr>
<td>(2\text{nd}) [10^x]</td>
<td>10(^x) (10 to a specified power)</td>
</tr>
<tr>
<td>(2\text{nd}) [\pi]</td>
<td>(\pi) (constant pi; 3.1415926535898)</td>
</tr>
<tr>
<td>[SIN]</td>
<td>sin (sine)</td>
</tr>
<tr>
<td>[COS]</td>
<td>cos (cosine)</td>
</tr>
<tr>
<td>[TAN]</td>
<td>tan (tangent)</td>
</tr>
<tr>
<td>[2nd] [\sin^{-1}]</td>
<td>(\sin^{-1}) (arcsine; inverse of sine)</td>
</tr>
<tr>
<td>[2nd] [\cos^{-1}]</td>
<td>(\cos^{-1}) (arccosine; inverse of cosine)</td>
</tr>
<tr>
<td>[2nd] [\tan^{-1}]</td>
<td>(\tan^{-1}) (arctangent; inverse of tangent)</td>
</tr>
<tr>
<td>[LOG]</td>
<td>log (logarithm)</td>
</tr>
<tr>
<td>[LN]</td>
<td>ln (natural log)</td>
</tr>
<tr>
<td>2\text{nd} [e^x]</td>
<td>(e^x) (constant e raised to a power)</td>
</tr>
<tr>
<td>2\text{nd} [\pi]</td>
<td>(\pi) (constant pi; 3.1415926535898)</td>
</tr>
</tbody>
</table>
# The MATH Menu

- **NUM**
- **PROB**
- **ANGLE**
- **HYP**
- **MISC**

- **INTER**

The MATH NUM (Number) Menu

- **NUM**
- **PROB**
- **ANGLE**
- **HYP**
- **MISC**

- **round**
- **iPart**
- **fPart**
- **int**
- **abs**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>round</strong>(value [, #ofDecimals])</td>
<td>Rounds value to 12 decimal places or to #ofDecimals</td>
</tr>
<tr>
<td><strong>iPart</strong>(value)</td>
<td>Returns the integer part or parts of value</td>
</tr>
<tr>
<td><strong>fPart</strong>(value)</td>
<td>Returns the fractional part or parts of value</td>
</tr>
<tr>
<td><strong>int</strong>(value)</td>
<td>Returns the largest integer less than or equal to value</td>
</tr>
<tr>
<td><strong>abs</strong>(value)</td>
<td>Returns the absolute value or magnitude of value</td>
</tr>
<tr>
<td><strong>sign</strong>(value)</td>
<td>Returns 1 if value is positive; 0 if value is 0; -1 if value is negative</td>
</tr>
<tr>
<td><strong>min</strong>(valueA, valueB)</td>
<td>Returns the smaller of valueA and valueB</td>
</tr>
<tr>
<td><strong>min</strong>(list)</td>
<td>Returns the smallest element of list</td>
</tr>
<tr>
<td><strong>max</strong>(valueA, valueB)</td>
<td>Returns the larger of valueA and valueB</td>
</tr>
<tr>
<td><strong>max</strong>(list)</td>
<td>Returns the largest element of list</td>
</tr>
<tr>
<td><strong>mod</strong>(numberA, numberB)</td>
<td>Returns numberA modulo numberB</td>
</tr>
</tbody>
</table>

*value can sometimes be an expression, list, vector, or matrix. For details about specific syntax options and examples, refer to the A to Z Reference.*
### The MATH PROB (Probability) Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>nPr</td>
<td>nCr</td>
<td>rand</td>
<td>randn</td>
</tr>
</tbody>
</table>

- **1 (factorial)** is valid for non-integers.

- **value!** Returns the factorial of a real value.
- **items nPr number** Returns the number of permutations of **items** (n) taken **number** (r) at a time.
- **items nCr number** Returns the number of combinations of **items** (n) taken **number** (r) at a time.
- **rand** Returns a random number > 0 and < 1; to control a random number sequence, first store an integer seed value to **rand** (such as **0→rand**).
- **randInt(lower,upper [#ofTrials])** (random integer) Returns a random integer bound by the specified integers, **lower** ≤ integer ≤ **upper**; to return a list of random integers, specify an integer > 1 for **#ofTrials**.
- **randNorm(mean, stdDeviation [#ofTrials])** (random normal) Returns a random real number from a normal distribution specified by **mean** and **stdDeviation**; to return a list of random numbers, specify an integer > 1 for **#ofTrials**.
- **randBin(#ofTrials, probabilityOfSuccess [#ofSimulations])** (random binomial) Returns a random real number from a binomial distribution, where **#ofTrials** ≥ 1 and 0 ≤ **probabilityOfSuccess** ≤ 1; to return a list of random numbers, specify an integer > 1 for **#ofSimulations**.

*randInt, randNorm, and randBin are abbreviated in the MATH PROB menu.*
The MATH ANGLE Menu

- The MATH ANGLE Menu can be accessed by pressing 2nd [MATH] [F3].

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>r</td>
<td>'</td>
<td>➤DMS</td>
<td></td>
</tr>
</tbody>
</table>

- angle can be a list for °, ′, ″, and ➤DMS.

- In a calculation, the result of a degrees'minutes'seconds' entry is treated as degrees in Degree angle mode only. It is treated as radians in Radian angle mode.

The MATH HYP (Hyperbolic) Menu

- The MATH HYP (Hyperbolic) Menu can be accessed by pressing 2nd [MATH] [F4].

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>sinh</td>
<td>cosh</td>
<td>tanh</td>
<td>sinh⁻¹</td>
<td>cosh⁻¹</td>
</tr>
</tbody>
</table>

- value can sometimes be an expression, list, vector, or matrix. For details about specific syntax options and examples, refer to the A to Z Reference.

- sinh value
- cosh value
- tanh value
- sinh⁻¹ value
- cosh⁻¹ value
- tanh⁻¹ value

- Overrides current angle mode setting to express angle in degrees
- Overrides current angle mode setting to express angle in radians
- Designates an angle as degrees, minutes, and seconds
- Displays angle in degrees’minutes’seconds’ format, even though you use degrees’minutes’seconds’ to enter a DMS angle

- Returns the hyperbolic sine of value
- Returns the hyperbolic cosine of value
- Returns the hyperbolic tangent of value
- Returns the hyperbolic arcsine of value
- Returns the hyperbolic arccosine of value
- Returns the hyperbolic arctangent of value
## The MATH MISC (Miscellaneous) Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum</td>
<td>prod</td>
<td>seq</td>
<td>lcm</td>
<td>gcd</td>
</tr>
</tbody>
</table>

- **num** \( \text{list} \) \( \text{prod} \) \( \text{seq} \) \( \text{lcm} \) \( \text{gcd} \)
  - **sum** \( \text{list} \)
  - **prod** \( \text{list} \)
  - **seq** \( \text{expression,variable,} \) \( \begin{array}{c} \text{begin,end[step]} \end{array} \)
  - **lcm** \( \text{valueA,valueB} \)
  - **gcd** \( \text{valueA,valueB} \)
  - **value** \( \rightarrow \text{Frac} \)
  - **value** \% \( \rightarrow \text{percent} \) \% \( \rightarrow \text{number} \)
  - **pEval** \( \text{coefficientList,xValue} \)
  - \( x^n \) \( \sqrt[n]{\text{value}} \)
  - **eval** \( \text{value} \)

*value* can sometimes be an expression, list, vector, or matrix. For details about specific syntax options, refer to the A to Z Reference.

- Returns the sum of the elements of *list*
- Returns the product of the elements of *list*
- Returns a list in which each element is the value of *expression* evaluated for *variable* from *begin* to *end* by *step*
- Returns the least common multiple of *valueA* and *valueB*
- Returns the greatest common divisor of *valueA* and *valueB*
- Displays *value* as a fraction
- Returns *value* divided by 100 (multiplied by .01)
- Returns percent of *number*
- Returns the value of a polynomial (whose coefficients are given in *coefficientList*) at *xValue*
- Returns the \( x^n \) root of *value*
- Returns a list of the values of all selected functions in the current graphing mode for the real *value* of the independent variable
The Interpolate/Extrapolate Editor

Using the interpolate/extrapolate editor, you can interpolate or extrapolate a value linearly, given two known pairs and the x-value or y-value of the unknown pair.

1. Display the interpolate/extrapolate editor.
2. Enter real values for the first known pair \((x_1, y_1)\). The values can be expressions.
3. Enter values for the second known pair \((x_2, y_2)\).
4. Enter a value for either the \(x\) value or the \(y\) value of the unknown pair.
5. If necessary, move the cursor to the value for which you want to solve \((x \text{ or } y)\).
6. Select SOLVE.

The result is interpolated or extrapolated and displayed; the variables \(x\) and \(y\) are not changed. A solid square in the first column indicates the interpolated or extrapolated value. After solving for a value, you can continue to use the interpolate/extrapolate editor.
The CALC (Calculus) Menu

You must set Dec mode to use the calculus functions.

For `evalF`, `nDer`, `der1`, and `der2`, `variable` can be a real or complex number or list.

You can use `der1` and `der2` in `expression`. You can use `nDer` once in `expression`.

For `fnInt`, `fMin`, and `fMax`, `lower < upper` must be true.

The calculus functions return values with respect to any user-created variable, to built-in variables `eqn` and `exp`, and to graphing variables such as `x`, `t`, and `θ`.

- `evalF(expression, variable, value)` Returns the value of `expression` with respect to `variable` for a given variable `value`
- `nDer(expression, variable[, value])` Returns an approximate numerical derivative of `expression` with respect to `variable` for the current variable value or specified variable `value`
- `der1(expression, variable[, value])` Returns the value of the first derivative of `expression` with respect to `variable` for the current variable value or specified variable `value`
- `der2(expression, variable[, value])` Returns the value of the second derivative of `expression` with respect to `variable` for the current variable value or specified variable `value`
- `fnInt(expression, variable, lower, upper)` Returns the numerical integral of `expression` with respect to `variable` between `lower` and `upper` boundaries
- `fMin(expression, variable, lower, upper)` Returns the minimum value of `expression` with respect to `variable` between `lower` and `upper` boundaries
- `fMax(expression, variable, lower, upper)` Returns the maximum value of `expression` with respect to `variable` between `lower` and `upper` boundaries
- `arc(expression, variable, start, end)` Returns the length of a segment of a curve defined by `expression` with respect to `variable` between `start` and `end`
The built-in variable $d$ defines the step size in calculating $\text{nDer}$ (in dxNDer differentiation mode only) and $\text{arc}$. The built-in variable $\text{tol}$ defines the tolerance in calculating $\text{fnInt}$, $\text{fMin}$, $\text{fMax}$, and $\text{arc}$. The value of each must be >0. These factors affect the accuracy of the calculations. As $d$ becomes smaller, the approximation typically is more accurate. For example, $\text{nDer}(A^3,A,5)$ returns $75.0001$ if $d = .01$, but returns $75$ if $d = .0001$ (Appendix).

The function integral error value is stored to the variable $\text{fnIntErr}$ (Appendix).

For $\text{arc}$ and $\text{fnInt}$ while dxDer1 mode is set, these functions are not valid in expression: evalF, der1, der2, fMin, fMax, nDer, seq, and any equation variable, such as $y1$.

You can approximate the fourth derivative at the current value of $x$ with this formula: $\text{nDer}(\text{nDer}(\text{der2}(x^4,x),x),x)$.

### The TEST (Relational) Menu

![TEST Menu](image)

Relational functions are valid for two lists of the same length. When $\text{valueA}$ and $\text{valueB}$ are lists, a list of results calculated element by element is returned.

- $\text{valueA} = \text{valueB}$ (equal to) Returns 1 if $\text{valueA}$ is equal to $\text{valueB}$; returns 0 if not equal; $\text{valueA}$ and $\text{valueB}$ can be real or complex numbers, lists, vectors, matrices, or strings
- $\text{valueA} < \text{valueB}$ (less than) Returns 1 if $\text{valueA}$ is less than $\text{valueB}$; returns 0 if $\text{valueA}$ is not less than $\text{valueB}$; $\text{valueA}$ and $\text{valueB}$ must be real numbers or lists
- $\text{valueA} > \text{valueB}$ (greater than) Returns 1 if $\text{valueA}$ is greater than $\text{valueB}$; returns 0 if $\text{valueA}$ is not greater than $\text{valueB}$; $\text{valueA}$ and $\text{valueB}$ must be real numbers or lists
- $\text{valueA} \leq \text{valueB}$ (less than or equal to) Returns 1 if $\text{valueA}$ is less than or equal to $\text{valueB}$; returns 0 if $\text{valueA}$ is not less than or equal to $\text{valueB}$; $\text{valueA}$ and $\text{valueB}$ must be real numbers or lists
- $\text{valueA} \geq \text{valueB}$ (greater than or equal to) Returns 1 if $\text{valueA}$ is greater than or equal to $\text{valueB}$; returns 0 if $\text{valueA}$ is not greater than or equal to $\text{valueB}$; $\text{valueA}$ and $\text{valueB}$ must be real numbers or lists

- $\text{valueA} \neq \text{valueB}$ (not equal to) Returns 0 if $\text{valueA}$ is equal to $\text{valueB}$; returns 1 if not equal; $\text{valueA}$ and $\text{valueB}$ can be real or complex numbers, lists, vectors, matrices, or strings
Chapter 3: Math, Calculus, and Test Operations

`valueA≥valueB` (greater than or equal to) Returns 1 if `valueA` is greater than or equal to `valueB`; returns 0 if `valueA` is not greater than or equal to `valueB`; `valueA` and `valueB` must be real numbers or lists.

`valueA≠valueB` (not equal to) Returns 1 if `valueA` is not equal to `valueB`; returns 0 if `valueA` is equal to `valueB`; `valueA` and `valueB` can be real or complex numbers, lists, vectors, matrices, or strings.

Using Tests in Expressions and Instructions

The TI-86 Evaluation Operating System (Appendix) performs all operations except Boolean operators before it performs relational functions. For example:

- The expression `2+2==2+3` evaluates to 0. The TI-86 performs the addition first, and then compares 4 to 5.
- The expression `2+(2==2)+3` evaluates to 6. The TI-86 performs the test in parentheses first, and then adds 2, 1, and 3.

You can use relational functions to control program flow (Chapter 16).
4 Constants, Conversions, Bases, and Complex Numbers

Using Built-In and User-Created Constants ....................... 58
Converting Units of Measure ............................................. 61
Number Bases .............................................................. 65
Using Complex Numbers ................................................... 70
Using Built-In and User-Created Constants

A constant is a variable with a specific value stored to it. The CONS BLTIN menu items are common constants built into the TI-86. You cannot edit the value of a built-in constant.

You can create your own constants and add them to the user-created constant menu for easy access. To enter a user-created constant, you must use the user-created constant editor (page 60); you cannot use \texttt{STO} or \texttt{=} to create a constant.

The CONS (Constants) Menu

<table>
<thead>
<tr>
<th>BLTIN</th>
<th>EDIT</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The CONS BLTIN (Built-In Constants) Menu

<table>
<thead>
<tr>
<th>BLTIN</th>
<th>EDIT</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>k</td>
<td>Cc</td>
</tr>
<tr>
<td>ec</td>
<td>Rc</td>
<td></td>
</tr>
</tbody>
</table>

- You can select built-in constants from the CONS BLTIN menu or enter them using the keyboard and the CHAR GREEK menu.

- \( \mu_0 \), \( \varepsilon_0 \), \( h \), \( c \), \( u \)
### Chapter 4: Constants, Conversions, Bases, and Complex Numbers

<table>
<thead>
<tr>
<th>Built-In Constant</th>
<th>Constant Name</th>
<th>Constant Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>Avogadro's number</td>
<td>6.0221367E23 mole⁻¹</td>
</tr>
<tr>
<td>k</td>
<td>Boltzman's constant</td>
<td>1.380658E-23 J/K</td>
</tr>
<tr>
<td>Cc</td>
<td>Coulomb constant</td>
<td>8.9875517873682E9 N m²/C²</td>
</tr>
<tr>
<td>ec</td>
<td>Electron charge</td>
<td>1.60217733E-19 C</td>
</tr>
<tr>
<td>Rc</td>
<td>Gas constant</td>
<td>8.31451 J/mole K</td>
</tr>
<tr>
<td>Gc</td>
<td>Gravitational constant</td>
<td>6.67259E-11 N m²/kg²</td>
</tr>
<tr>
<td>g</td>
<td>Earth acceleration due to gravity</td>
<td>9.80665 m/sec²</td>
</tr>
<tr>
<td>Me</td>
<td>Mass of an electron</td>
<td>9.1093897E-31 kg</td>
</tr>
<tr>
<td>Mp</td>
<td>Mass of a proton</td>
<td>1.6726231E-27 kg</td>
</tr>
<tr>
<td>Mn</td>
<td>Mass of a neutron</td>
<td>1.6749286E-27 kg</td>
</tr>
<tr>
<td>μ0</td>
<td>Permeability of a vacuum</td>
<td>1.2566370614359E-6 N/A²</td>
</tr>
<tr>
<td>ε0</td>
<td>Permittivity of a vacuum</td>
<td>8.8541878176204E-12 F/m</td>
</tr>
<tr>
<td>h</td>
<td>Planck's constant</td>
<td>6.6260755E-34 J sec</td>
</tr>
<tr>
<td>c</td>
<td>Speed of light in a vacuum</td>
<td>299,792,458 m/sec</td>
</tr>
<tr>
<td>u</td>
<td>Atomic mass unit</td>
<td>1.6605402E-27 kg</td>
</tr>
<tr>
<td>π</td>
<td>Pi</td>
<td>3.1415926535898</td>
</tr>
<tr>
<td>e</td>
<td>Base of natural log</td>
<td>2.718281828459</td>
</tr>
</tbody>
</table>

To use π, press [2nd] [x] or select it from the CATALOG.
To use e^a, press [2nd] [e^x].
To use e, press [2nd] [alpha] [E].
Creating or Redefining a User-Created Constant

1. Display the CONS menu.

2. Display the constant editor. The Name= prompt, Value= prompt, and CONS USER menu are displayed. ALPHA-lock is on.

3. Enter a constant name. Either enter a new name one to eight characters long, starting with a letter, or select a name from the CONS USER menu. The cursor moves to the Value= prompt and the CONS EDIT menu is displayed (see below).

4. Enter the real or complex constant value, which can be an expression. The value is stored to the constant as you enter it. The user-created constant becomes a CONS USER menu item.

196.9665 is the atomic weight of gold (Au).

You can enter a value later.

The Constant Editor Menu

<table>
<thead>
<tr>
<th>PreV</th>
<th>NeXT</th>
<th>DeLET</th>
</tr>
</thead>
</table>

PREV Displays the name and value (if any) of the previous constant on the CONS USER menu.

NEXT Displays the name and value (if any) of the next constant on the CONS USER menu.

DELETE Deletes the name and value of the constant currently displayed in the constant editor.
Chapter 4: Constants, Conversions, Bases, and Complex Numbers

Entering a Constant Name in an Expression
You can enter a constant in an expression in any of three ways.
- Select the constant name from the CONS BLTIN menu or the CONS USER menu.
- Select a user-created constant name from the VARS CONS screen.
- Use the ALPHA keys, alpha keys, and other character keys to enter a constant name.

Converting Units of Measure
With the TI-86, you can convert a value measured in one unit into its equivalent value in another unit of measure. For example, you can convert inches to yards, quarts to liters, or degrees Fahrenheit to degrees Celsius.

The units of measure from which and to which you convert must be compatible. For example, you cannot convert inches to degrees Fahrenheit, or yards to calories. Each menu item on the CONV menu (page 62) represents a unit-of-measure group, such as length (LNGTH), volume (VOL), and pressure (PRESS). Within each menu, all units are compatible.

Converting a Unit of Measure
To use any conversion instruction, the syntax is:
(value)(currentUnit)(newUnit)

1. Enter the real value to be converted.
2. Display the CONV menu.
3. Select the TEMP conversion group.

You can enter a conversion expression anywhere that an expression is valid.

In the example, 2 degrees Celsius is converted to degrees Fahrenheit. Always use parentheses when value is negative.
4. Select the current unit of measure (°C) from the conversion group menu. The unit abbreviation and conversion symbol (°) are pasted to the cursor location.

5. Select the new unit of measure (°F) from the conversion group menu. The unit abbreviation is pasted to the cursor location.

6. Convert the measurement.

The CONV (Conversions) Menu

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>AREA</th>
<th>VOL</th>
<th>TIME</th>
<th>TEMP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Important: When you convert a negative value, you must enclose in parentheses the value and its negation sign, as in (-4). Otherwise, the TI-86 order of evaluation will perform the conversion first, and then apply the negation to the converted value.

If you enter... | ...The TI-86 converts it to...
--- | ---
(-4)°C+F | 24.8 degrees Fahrenheit (-4° Celsius converted to degrees Fahrenheit)
-4°C+F | -39.2 degrees Fahrenheit (4° Celsius converted to degrees Fahrenheit, then negated)
### The CONV LNGTH (Length) Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>millimeters</td>
</tr>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>m</td>
<td>meters</td>
</tr>
<tr>
<td>in</td>
<td>inches</td>
</tr>
<tr>
<td>ft</td>
<td>feet</td>
</tr>
<tr>
<td>yd</td>
<td>yards</td>
</tr>
<tr>
<td>km</td>
<td>kilometers</td>
</tr>
<tr>
<td>mil</td>
<td>mils</td>
</tr>
<tr>
<td>Ang</td>
<td>Angstroms</td>
</tr>
<tr>
<td>mile</td>
<td>miles</td>
</tr>
<tr>
<td>nmile</td>
<td>nautical miles</td>
</tr>
<tr>
<td>rod</td>
<td>rods</td>
</tr>
<tr>
<td>lit-yr</td>
<td>light-years</td>
</tr>
<tr>
<td>fath</td>
<td>fathoms</td>
</tr>
</tbody>
</table>

### The CONV AREA Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft²</td>
<td>square feet</td>
</tr>
<tr>
<td>m²</td>
<td>square meters</td>
</tr>
<tr>
<td>mi²</td>
<td>square miles</td>
</tr>
<tr>
<td>km²</td>
<td>square kilometers</td>
</tr>
<tr>
<td>cm²</td>
<td>square centimeters</td>
</tr>
<tr>
<td>acre</td>
<td>acres</td>
</tr>
<tr>
<td>in²</td>
<td>square inches</td>
</tr>
<tr>
<td>ha</td>
<td>hectares</td>
</tr>
</tbody>
</table>

### The CONV VOL (Volume) Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>liter</td>
<td>liters</td>
</tr>
<tr>
<td>gal</td>
<td>gallons</td>
</tr>
<tr>
<td>qt</td>
<td>quarts</td>
</tr>
<tr>
<td>pt</td>
<td>pints</td>
</tr>
<tr>
<td>oz</td>
<td>ounces</td>
</tr>
<tr>
<td>cm³</td>
<td>cubic centimeters</td>
</tr>
<tr>
<td>in³</td>
<td>cubic inches</td>
</tr>
<tr>
<td>ft³</td>
<td>cubic feet</td>
</tr>
<tr>
<td>m³</td>
<td>cubic meters</td>
</tr>
<tr>
<td>galUK</td>
<td>UK gallons</td>
</tr>
<tr>
<td>tsp</td>
<td>teaspoons</td>
</tr>
<tr>
<td>tbsp</td>
<td>tablespoons</td>
</tr>
<tr>
<td>ml</td>
<td>milliliters</td>
</tr>
<tr>
<td>cup</td>
<td>cups</td>
</tr>
<tr>
<td>ozUK</td>
<td>UK ounces</td>
</tr>
</tbody>
</table>

### The CONV TIME Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec</td>
<td>seconds</td>
</tr>
<tr>
<td>mn</td>
<td>minutes</td>
</tr>
<tr>
<td>hr</td>
<td>hours</td>
</tr>
<tr>
<td>day</td>
<td>days</td>
</tr>
<tr>
<td>yr</td>
<td>years</td>
</tr>
<tr>
<td>week</td>
<td>weeks</td>
</tr>
<tr>
<td>ms</td>
<td>milliseconds</td>
</tr>
<tr>
<td>µs</td>
<td>microseconds</td>
</tr>
<tr>
<td>ns</td>
<td>nanoseconds</td>
</tr>
</tbody>
</table>

### The CONV TEMP (Temperature) Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>°C</td>
<td>degrees Celsius</td>
</tr>
<tr>
<td>°F</td>
<td>degrees Fahrenheit</td>
</tr>
<tr>
<td>°K</td>
<td>degrees Kelvin</td>
</tr>
<tr>
<td>°R</td>
<td>degrees Rankin</td>
</tr>
<tr>
<td>°</td>
<td></td>
</tr>
</tbody>
</table>
### The CONV MASS Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gm</td>
<td>grams</td>
<td>atomic mass units</td>
</tr>
<tr>
<td>kg</td>
<td>kilograms</td>
<td>slugs</td>
</tr>
<tr>
<td>lb</td>
<td>pounds</td>
<td>metric tons</td>
</tr>
</tbody>
</table>

### The CONV FORCE Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Newtons</td>
<td>ton force</td>
</tr>
<tr>
<td>dyne</td>
<td>dynes</td>
<td>kilogram force</td>
</tr>
</tbody>
</table>

### The CONV PRESS (Pressure) Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm</td>
<td>atmospheres</td>
<td>lb/in² pounds per square inch</td>
</tr>
<tr>
<td>bar</td>
<td>bars</td>
<td>mmHg millimeters of mercury</td>
</tr>
<tr>
<td>N/m²</td>
<td>Newtons per square meter</td>
<td>mmH₂O millimeters of water</td>
</tr>
</tbody>
</table>

### The CONV ENRGY (Energy) Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>Joules</td>
<td>ft-lb foot-pounds</td>
</tr>
<tr>
<td>cal</td>
<td>calories</td>
<td>kw-hr kilowatt hours</td>
</tr>
<tr>
<td>Btu</td>
<td>British thermal units</td>
<td>eV electron Volts</td>
</tr>
</tbody>
</table>

### The CONV POWER Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hp</td>
<td>horsepower</td>
<td>ft-lb/s foot-pounds per second</td>
</tr>
<tr>
<td>W</td>
<td>Watts</td>
<td>cal/s calories per second</td>
</tr>
<tr>
<td>Btu/m</td>
<td>British thermal units</td>
<td></td>
</tr>
</tbody>
</table>

### The CONV SPEED Menu

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft/s</td>
<td>feet per second</td>
<td>mi/hr miles per hour</td>
</tr>
<tr>
<td>m/s</td>
<td>meters per second</td>
<td>km/hr kilometers per hour</td>
</tr>
</tbody>
</table>
Converting a Value Expressed as a Rate

To convert a value expressed as a rate on the home screen, you can use parentheses and the division operator ( / ). For example, if a car travels 325 miles in 4 hours, and you want to know the rate of speed in kilometers per hour, enter this expression:

\[(325 \div 4) \text{mi/hr} \div 4 \text{km/hr}\]

This expression returns 131 km/hr (rounded up).

You also can return this result using only a forward slash, as in: \( \frac{325 \text{mi}}{4 \text{hr}} \div 4 \text{hr} \div 4 \text{km} \)

Number Bases

The number base mode setting (Chapter 1) controls how the TI-86 interprets an entered number and displays results on the home screen. However, you can enter numbers in any number base using number base designators b, o, d, and h. Then you can display the result on the home screen in any number base using number base conversions.

All numbers are stored internally as decimal. If you perform an operation in a mode setting other than Dec, the TI-86 performs integer mathematics, truncating to an integer after every calculation and expression.

For example, in Hex mode, \( 1/3+7 \) returns 7h (1 divided by 3, truncated to 0, and then added to 7).
Chapter 4: Constants, Conversions, Bases, and Complex Numbers

Number Base Ranges
Binary, octal, and hexadecimal numbers on the TI-86 are defined in these ranges.

<table>
<thead>
<tr>
<th>Type</th>
<th>Low Value/High Value</th>
<th>Decimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>1000 0000 0000 0001b</td>
<td>-32,767</td>
</tr>
<tr>
<td></td>
<td>0111 1111 1111 1111b</td>
<td>32,767</td>
</tr>
<tr>
<td>Octal</td>
<td>5120 6357 4194 0001o</td>
<td>-99,999,999,999,999</td>
</tr>
<tr>
<td></td>
<td>2657 1420 3643 7777o</td>
<td>99,999,999,999,999</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>FFFF A50C EF85 C001h</td>
<td>-99,999,999,999,999</td>
</tr>
<tr>
<td></td>
<td>0000 5AF3 107A 3FFFh</td>
<td>99,999,999,999,999</td>
</tr>
</tbody>
</table>

One’s and Two’s Complements
To obtain the one’s complement of a binary number, enter the not function (page 68) before the number. For example, not 111100001111 in Bin mode returns 1111000011110000.

To obtain the two’s complement of a binary number, press  before entering the number. For example, -111100001111 in Bin mode returns 1111000011110001.

The (Number) BASE Menu

<table>
<thead>
<tr>
<th>A-F</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- hexadecimal characters menu
- base conversion menu
- rotate/shift menu

- base type menu
- Boolean operator menu
The BASE A-F (Hexadecimal Characters) Menu

This is the BASE A-F menu displayed on the home screen. To use $A$, press 2nd [M].

<table>
<thead>
<tr>
<th>A</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

In the example, the upper menu is the list editor menu (2nd [LIST] in Dec number base mode). If Hex number base mode is not set, you must enter the $h$ designator, even if the number contains a special hexadecimal character.

Entering Hexadecimal Digits

To enter a hexadecimal number, use the number keys as you would for a decimal number. Select the hexadecimal characters $A$ through $F$ from the menu as needed.

The BASE TYPE Menu

<table>
<thead>
<tr>
<th>A-F</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>h</td>
<td>o</td>
<td>d</td>
<td></td>
</tr>
</tbody>
</table>

In an expression, you can designate a number in any number base, regardless of mode. After you enter the number, select the appropriate base type symbol from the BASE TYPE menu. The base type symbol is pasted to the cursor location. Here are some examples.

In Dec mode (default): 10b+10 ENTER 12 10h+10 ENTER 26
In Oct mode: 10b+10 ENTER 12o 10d+10 ENTER 22o
In Bin mode: 10h+10 ENTER 10010b 101d+10 ENTER 1100b
In Hex mode: 10b+10 ENTER 12h 10d+10 ENTER 1Ah
The BASE CONV (Conversion) Menu

<table>
<thead>
<tr>
<th>A-F</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bin</td>
<td>Hex</td>
<td>Oct</td>
<td>Dec</td>
<td></td>
</tr>
</tbody>
</table>

value ➔ Bin Displays value as binary
value ➔ Hex Displays value as hexadecimal
value ➔ Dec Displays value as decimal

Converting Number Bases

1. In Dec mode, solve \(10b + Fh + 10c + 10\).

2. Add 1 to the result and convert it to Bin number base display.

3. Add 1 to the result and convert it to Hex number base display.

4. Add 1 to the result and convert it to Oct number base display.

5. Add 1 to the result and convert it to Dec number base display.

The BASE BOOL (Boolean) Menu

<table>
<thead>
<tr>
<th>A-F</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>and</td>
<td>or</td>
<td>xor</td>
<td>not</td>
<td></td>
</tr>
</tbody>
</table>

valueA and valueB  valueA or valueB  valueA xor valueB  not value
Results of Boolean Operations

When a Boolean expression is evaluated, the arguments are converted to hexadecimal integers and the corresponding bits of the arguments are compared, as this table shows.

<table>
<thead>
<tr>
<th>valueA is...</th>
<th>...and valueB is...</th>
<th>and</th>
<th>or</th>
<th>xor</th>
<th>not (valueA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The result is displayed according to the current mode setting. For example:

- In **Bin** mode, 101 and 110 returns **100**.
- In **Hex** mode, 5 and 6 returns **4**.

The **BASE BIT Menu**

Both the argument and the result must be within defined number ranges (page 66).

<table>
<thead>
<tr>
<th>A-F</th>
<th>TYPE</th>
<th>CONV</th>
<th>BOOL</th>
<th>BIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>rotR</td>
<td>roTL</td>
<td>shftR</td>
<td>shflL</td>
<td></td>
</tr>
</tbody>
</table>

- rotR value  Rotates value right
- roTL value  Rotates value left
- shftR value  Shifts value right
- shftL value  Shifts value left

Rotate and shift operate on 16 base digits. To minimize an overflow error, enter the argument in binary form.
Using Complex Numbers

A complex number has two components: real (a) and imaginary (+bi). On the TI-86, you enter the complex number a+bi as:

- **(real, imaginary)** in rectangular form
- **(magnitude \(\angle\) angle)** in polar form

You can enter a complex number in rectangular or polar form, regardless of the current complex number mode setting. The separator (, or \(\angle\)) determines the form.

- To enter rectangular form, separate **real** and **imaginary** with a comma (,).
- To enter polar form, separate **magnitude** and **angle** with an angle symbol (\(\angle\)).

Each component (**real**, **imaginary**, **magnitude**, or **angle**) can be a real number or an expression that evaluates to a real number; expressions are evaluated when you press **ENTER**.

When **RectC** complex number mode is set, complex numbers are displayed in rectangular form, regardless of the form in which you enter them (as shown to the right).

When **PolarC** complex number mode is set, complex numbers are displayed in polar form, regardless of the form in which you enter them (as shown to the right).

### Complex Results

Complex numbers in results, including list, matrix, and vector elements, are displayed in the form (rectangular or polar) specified by the mode setting (Chapter 1) or by a display conversion instruction (page 61).

- When **Radian** angle mode is set, results are displayed as **(magnitude \(\angle\) angle)**.
- When **Degree** angle mode is set, results are displayed as **(real, imaginary)**.
For example, when PolarC and Degree modes are set, (2,1)−(1∠45) returns (1.32565429614−12.7643896828).

**Using a Complex Number in an Expression**
- Enter the complex number directly.
- Use the ALPHA keys, alpha keys, and other character keys to enter a complex variable.
- Select a complex variable from the VARS CPLX screen.

**The CPLX (Complex Number) Menu**

<table>
<thead>
<tr>
<th>conj (real,imaginary)</th>
<th>Returns the complex conjugate of a complex value, list, vector or matrix; the result is (real,imaginary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>conj (magnitude∠angle)</td>
<td>Returns (magnitude∠angle)</td>
</tr>
<tr>
<td>real (real,imaginary)</td>
<td>Returns the real portion of a complex number, list, vector, or matrix as a real number; the result is real</td>
</tr>
<tr>
<td>real (magnitude∠angle)</td>
<td>Returns magnitude*cosine(angle)</td>
</tr>
<tr>
<td>imag (real,imaginary)</td>
<td>Returns the imaginary (non-real) portion of a complex number, list, vector, or matrix as a real number; the result is imaginary</td>
</tr>
<tr>
<td>imag (magnitude∠angle)</td>
<td>Returns magnitude*sin(angle)</td>
</tr>
<tr>
<td>abs (real,imaginary)</td>
<td>(Absolute value) Returns the magnitude (modulus) of a complex number, list, vector, or matrix of complex numbers; the result is (\sqrt{(\text{real}^2 + \text{imaginary}^2)})</td>
</tr>
<tr>
<td>abs (magnitude∠angle)</td>
<td>Returns magnitude</td>
</tr>
</tbody>
</table>
Chapter 4: Constants, Conversions, Bases, and Complex Numbers

angle \( (real, imaginary) \)  
Returns the polar angle of a complex number, list, vector, or matrix calculated as \( \tan^{-1} \frac{imaginary}{real} \) (adjusted by \( \pi \) in the second quadrant or \( -\pi \) in the third quadrant); the result is \( \tan^{-1} \frac{imaginary}{real} \)

angle \( (magnitude \angle angle) \)  
Returns \( angle \) (where \( -\pi < angle < \pi \))

complexValue \( \rightarrow \) Rec  
Displays \( complexValue \) in rectangular format \((real, imaginary)\), regardless of complex mode setting; valid only at the end of a command and only when \( complexValue \) is indeed complex

complexValue \( \rightarrow \) Pol  
Displays \( complexValue \) in polar format \((magnitude \angle angle)\), regardless of complex mode setting; valid only at the end of a command and only when \( complexValue \) is indeed complex

You can enter a complex list, vector, or matrix directly. The syntax below is for lists. To enter a complex vector or matrix, substitute brackets for braces below and use the correct form for either data type (Chapters 12 and 13).

In rectangular form, to use lists of complex numbers with \( \text{conj, real, imag, abs, and angle} \), the syntax is:
\[
\text{conj}\{ (realA, imaginaryA), (realB, imaginaryB), (realC, imaginaryC), ... \}
\]

In polar form, to use lists of complex numbers with \( \text{conj, real, imag, abs, and angle} \), the syntax is:
\[
\text{real}\{ (magnitudeA \angle angleA), (magnitudeB \angle angleB), (magnitudeC \angle angleC), ... \}
\]

When you use a list the TI-86 calculates the result element by element and returns a list, in which each element is expressed according to the complex mode setting.
5 Function Graphing

- Defining a Graph ............................................................ 74
- Setting the Graph Mode ............................................... 74
- The GRAPH Menu .......................................................... 75
- Using the Equation Editor ............................................ 76
- Setting the Window Variables ....................................... 81
- Setting the Graph Format ............................................. 83
- Displaying a Graph ....................................................... 85
Defining a Graph

This chapter describes the process for graphing functions in **Func** graphing mode, but the process is similar for each TI-86 graphing mode. Chapters 8, 9, and 10 describe the unique aspects of polar, parametric, and differential equation graphing modes. Chapter 6 describes various graphing tools, many of which you can use in all graphing modes.

1. Set the graphing mode (page 74).
2. Define, edit, or select one or more functions in the equation editor (pages 76 and 77).
3. Select the graph style for each function (page 79).
4. Deselect stat plots, if necessary (page 81).
5. Set the viewing window variables (page 81).
6. Select the graph format settings (page 83).

Setting the Graph Mode

To display the mode screen, press `[2nd] [MODE]`. All default mode settings, including **Func** graphing mode, are highlighted in the picture to the right. The graphing modes are on the fifth line.

- **Func** (function graphing)
- **Pol** (polar graphing; Chapter 8)
- **Param** (parametric graphing; Chapter 9)
- **DifEq** (differential equation graphing; Chapter 10)
Each graphing mode has a unique equation editor. You must select the graphing mode and **Dec** number base mode before you enter the functions. The TI-86 retains in memory all equations stored to the **Func**, **Pol**, **Param**, and **DifEq** equation editors. Each mode also has unique graph format settings and window variables.

Stat plot on/off status, zoom factors, mode settings, and tolerance apply to all graphing modes; changing the graphing mode does not affect them.

These mode settings affect graphing results.

- **Radian** or **Degree** angle mode affects the interpretation of some functions.
- **dxDer1** or **dxNDER** differentiation mode affects plotting of selected functions.

**The GRAPH Menu**

![GRAPH Menu](graph_menu.png)

- **y(x)=** Displays the equation editor; use this screen to enter functions to be graphed
- **WIND** Displays the window editor; use this editor to change graph screen dimensions
- **ZOOM** Displays the GRAPH ZOOM menu; use these items to change the graph screen dimensions
- **TRACE** Activates the trace cursor; use this cursor to trace along the graph of a specific function
- **GRAPH** Displays the graph screen; graphs all selected functions and turned on stat plots
- **MATH** Displays the GRAPH MATH menu; use this menu to explore graphs mathematically
- **DRAW** Displays the GRAPH DRAW menu; use this menu to draw on graphs or test pixels

*Chapter 6 describes these GRAPH menu items: ZOOM, TRACE, MATH, DRAW, STGDB, RCGDB, EVAL, STPIC, and RCPIC.*
Chapter 5: Function Graphing

FORMT  Displays the graph format screen; use this screen to select graph format settings
STGDB  Displays the Name= prompt and STGDB menu; use this prompt to enter a GDB variable
RCGDB  Displays the Name= prompt and RCGDB menu; use this menu to recall a graph database
EVAL   Displays the Eval x= prompt; enter an x for which you want to solve the current function
STPIC  Displays the Name= prompt and STPIC menu; use this prompt to enter a PIC variable
RCPIC  Displays the Name= prompt and RCPIC menu; use this menu to recall a picture

Using the Equation Editor

To display the equation editor in Func graphing mode, select \( y(x) = \) from the GRAPH menu (GRAPH \[ F1 \]). The GRAPH menu shifts up and the equation editor menu is displayed as the lower menu. You can store up to 99 functions in the equation editor, if sufficient memory is available.

If a function is selected, its equals sign (=) is highlighted in the equation editor. If the function is deselected, its equals sign is not highlighted. Only selected functions are plotted when the TI-86 plots a graph.

The Equation Editor (GRAPH y(x)=) Menu

<table>
<thead>
<tr>
<th>y(x)=</th>
<th>WIND</th>
<th>ZOOM</th>
<th>TRACE</th>
<th>GRAPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>INSf</td>
<td>DELf</td>
<td>SELCT</td>
</tr>
</tbody>
</table>

\[ \text{Graph} \ [ F1 ] \]
Chapter 5: Function Graphing

Defining a Function in the Equation Editor

1. Display the equation editor.

2. If functions are stored in the equation editor, move the cursor down until a blank function is displayed.

3. Enter an equation in terms of x to define the function. When you enter the first character, the function is selected automatically. (The function's equals sign is highlighted.)

4. Move the cursor to the next function.
Notes about Defining Function Equations

- You can include functions, variables, constants, matrices, matrix elements, vectors, vector elements, lists, list elements, complex values, or other equations in the equation.
- If you include matrices, vectors, or complex values, the equation must evaluate to a real number at each point.
- You can include another defined function in an equation. For example, given $y_1 = \sin x$ and $y_2 = 4 + y_1$, the function $y_2$ would equal 4 plus the sine of $x$.
- To enter a function name, select $y$ from the equation editor menu, and then enter the appropriate number.
- To insert the contents of an equation variable, use RCL (Chapter 1). To enter the equation variable at the Rcl prompt, use the ALPHA keys, alpha keys, and other character keys.
- To select all functions from the home screen or in the program editor, select FnOn from the CATALOG (or enter the individual characters) and press ENTER.
- To select specific functions from the home screen or in the program editor, select FnOn from the CATALOG (or enter the individual characters), enter the number of each function, and press ENTER. For example, to select $y_1$, $y_3$, and $y_5$, enter FnOn 1,3,5.
- To deselect functions from the home screen or in the program editor, use FnOff the same way you use FnOn to select functions.
- When a function evaluates to a non-real number, the value is not plotted on the graph; no error is returned.
Selecting Graph Styles

Depending on which graphing mode is set, the TI-86 offers up to seven distinct graph styles. You can assign these styles to specific functions to visually differentiate each from the others.

For example, you can show $y_1$ as a connected line ($\text{\textbackslash } y_1=$ in the equation editor) and $y_2$ as a dotted line ($\text{\textbackslash } y_2=$), and shade the area above $y_3$ ($\text{\textbackslash } y_3=$).

Also, you can manipulate the styles to illustrate actual phenomena graphically, such as a ball flying through the air (using $\text{\textbackslash }$) or the circular movement of a chair on a Ferris wheel (using $\text{\textbackslash } y$).

<table>
<thead>
<tr>
<th>Icon Style</th>
<th>Characteristics of the Plotted Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{\textbackslash }$ Line</td>
<td>A solid line connects each plotted point; this is the default in Connected mode</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Thick</td>
<td>A thick solid line connects each plotted point</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Above</td>
<td>Shades the area above the function</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Below</td>
<td>Shades the area below the function</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Path</td>
<td>A circle cursor traces the leading edge of the function and draws a path as it plots</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Animate</td>
<td>A circle cursor traces the leading edge of the function as it plots; does not draw a path</td>
</tr>
<tr>
<td>$\text{\textbackslash }$ Dot</td>
<td>A small dot represents each plotted point; this is the default in Dot mode</td>
</tr>
</tbody>
</table>

To set the graph style from a program, select GrStl( from the CATALOG (A to Z Reference).
Chapter 5: Function Graphing

Setting the Graph Style in the Equation Editor

1. Display the equation editor.
2. Move the cursor to the function or functions for which you want to set the graph style.
3. Display the equation editor menu item STYLE.
4. Select STYLE repeatedly to scroll the graph style icons to the left of the equation name.
5. View the graph with the new graph style.
6. Clear the GRAPH menu to view the graph only.

Using Shading Patterns to Differentiate Functions

When you select \(\uparrow\) (shade above) or \(\downarrow\) (shade below) for more than one function, the TI-86 rotates through a series of four shading patterns.

- First shaded function: vertical lines
- Second shaded function: horizontal lines
- Third shaded function: negatively sloping diagonal lines
- Fourth shaded function: positively sloping diagonal lines

The rotation returns to vertical lines for the fifth shaded function and repeats the order.
Chapter 5: Function Graphing

Viewing and Changing On/Off Status of Stat Plots

Plot1, Plot2, Plot3 on the top line of the equation editor displays the on/off status of each stat plot (Chapter 14). When a plot name is highlighted on this line, the plot is on.

To change the on/off status of a stat plot from the equation editor, press $\downarrow$, $\rightarrow$, and $\leftarrow$ to place the cursor on Plot1, Plot2, or Plot3, and then press [ENTER].

Setting the Window Variables

The graph screen window represents the portion of the coordinate plane displayed on the graph screen. By setting window variables, you can define the graph screen window boundaries and other attributes.

- $x_{\text{Min}}$, $x_{\text{Max}}$, $y_{\text{Min}}$, and $y_{\text{Max}}$ are the graph screen boundaries.

- $x_{\text{Scl}}$ (x scale) is the number of units represented by the distance from one tick mark to the next tick mark on the x-axis.

- $y_{\text{Scl}}$ (y scale) is the number of units represented by the distance from one tick mark to the next tick mark on the y-axis.

- $x_{\text{Res}}$ sets pixel resolution for function graphs only, using integers 1 through 8.

- At $x_{\text{Res}}=1$ (the default), functions are evaluated and graphed at each pixel on the x-axis.

- At $x_{\text{Res}}=8$, functions are evaluated and graphed at every eighth pixel along the x-axis.

To remove tick marks from both axes, set $x_{\text{Scl}}=0$ and $y_{\text{Scl}}=0$.

Small $x_{\text{Res}}$ values improve graph resolution but may cause the TI-86 to plot graphs more slowly.
Displaying the Window Editor

To display the window editor, select WIND from the GRAPH menu (GRAPH [F2]). Each graphing mode has a unique window editor. The window editor to the right shows the default values in Func graphing mode. ↓ indicates that xRes=1 (x resolution) is below yScl on the window editor.

Changing a Window Variable Value

1. Display the window editor.
2. Move the cursor to the window variable you want to change.
3. Edit the value, which can be an expression.
4. Evaluate any expressions and store the value.

To change a window variable value from the home screen or in the program editor, enter the value, and then press [STO•]. Either select the window variable from the VARS WIND screen (2nd [CATLG-VARS] MORE [MORE] WIND) or enter individual characters. Press [ENTER].
Setting Graphing Accuracy with $\Delta x$ and $\Delta y$

The window variables $\Delta x$ and $\Delta y$ define the distance from the center of one pixel to the center of any adjacent pixel. When you display a graph, the values of $\Delta x$ and $\Delta y$ are calculated from $x_{\text{Min}}, x_{\text{Max}}, y_{\text{Min}},$ and $y_{\text{Max}}$ using these formulas:

$$\Delta x = \frac{(x_{\text{Min}} + x_{\text{Max}})}{126}$$
$$\Delta y = \frac{(y_{\text{Min}} + y_{\text{Max}})}{62}$$

$\Delta x$ and $\Delta y$ are not on the window editor. To change them, you must follow the steps above for changing a window variable value from the home screen or in the program editor. When you change the values stored to $\Delta x$ and $\Delta y$, the TI-86 automatically recalculates $x_{\text{Max}}$ and $y_{\text{Max}}$ from $\Delta x, x_{\text{Min}}, \Delta y,$ and $y_{\text{Min}},$ and the new values are stored.

Setting the Graph Format

To display the graph format screen, select FORMT from the GRAPH menu (GRAPH MORE [F3]). The graph format settings define various characteristics of the displayed graph. The current settings are highlighted.

To change a setting, move the cursor onto the new setting, and then press ENTER, the same as on the mode screen.
## Chapter 5: Function Graphing

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RectGC</td>
<td>Displays the cursor location as rectangular graph coordinates ( x ) and ( y ); when RectGC is set, plotting the graph, moving the free-moving cursor, and tracing update ( x ) and ( y ); if CoordOn format also is selected, ( x ) and ( y ) are displayed</td>
</tr>
<tr>
<td>PolarGC</td>
<td>Displays the cursor location as polar graph coordinates ( R ) and ( \theta ); when PolarGC is set, plotting the graph, moving the free-moving cursor, and tracing update ( x ), ( y ), ( R ) and ( \theta ); if CoordOn format also is selected, ( R ) and ( \theta ) are displayed</td>
</tr>
<tr>
<td>CoordOn</td>
<td>Displays the cursor coordinates at the bottom of the graph</td>
</tr>
<tr>
<td>CoordOff</td>
<td>Does not display the cursor coordinates at the bottom of the graph</td>
</tr>
<tr>
<td>DrawLine</td>
<td>Draws a line between the points calculated for the functions in the equation editor</td>
</tr>
<tr>
<td>DrawDot</td>
<td>Plots only the calculated points for the functions in the equation editor</td>
</tr>
<tr>
<td>SeqG</td>
<td>(sequential graphing) Evaluates and plots one function completely before evaluating and plotting the next function</td>
</tr>
<tr>
<td>SimulG</td>
<td>(simultaneous graphing) Evaluates and plots all selected functions for a single value of ( x ) and then evaluates and plots them for the next value of ( x )</td>
</tr>
<tr>
<td>GridOff</td>
<td>Omits the grid points from the display</td>
</tr>
<tr>
<td>GridOn</td>
<td>Displays grid points</td>
</tr>
<tr>
<td>AxesOn</td>
<td>Displays the axes</td>
</tr>
<tr>
<td>AxesOff</td>
<td>Omits the axes from the display; AxesOff overrides the CoordOff format setting</td>
</tr>
<tr>
<td>LabelOff</td>
<td>Omits the axis labels from the display</td>
</tr>
<tr>
<td>LabelOn</td>
<td>Labels the axes, if AxesOn is also selected; ( x ) and ( y ) for Func, Pol, and Param modes; various labels in DifEq mode</td>
</tr>
</tbody>
</table>

---

DifEq graphing mode has a unique set of graph format settings (Chapter 10). Grid points cover the graph screen in rows that correspond to the tick marks on each axis.
Displaying a Graph

To display a graph, select `GRAPH` from the `GRAPH` menu. The graph screen is displayed. If the graph is newly defined, the busy indicator is displayed at the top-right corner as the TI-86 draws the graph.

- In `SeqG` format, the TI-86 draws each selected function one by one, in function-name order (for example, `y1` is graphed first, `y2` is graphed second, and so on).
- In `SimulG` format, the TI-86 draws all selected graphs simultaneously.

You can display and explore a graph from a program (Chapter 16). To use graphing commands on the home screen, select them from the CATALOG or entering the individual characters.

Pausing or Stopping a Graph in Progress

- To pause graph plotting, press `b`. To resume plotting, press `b` again.
- To stop graph plotting, press `^`. To replot, select `GRAPH` from the `GRAPH` menu.

Modifying a Drawn Graph

To remove these items from the graph screen:

<table>
<thead>
<tr>
<th>Press (or select):</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cursor, coordinate values, or menus (To restore menus, press <code>EXIT</code> or <code>GRAPH</code>)</td>
<td><code>CLEAR</code></td>
</tr>
<tr>
<td>Free-moving cursor and coordinate values but not the menus</td>
<td><code>ENTER</code></td>
</tr>
<tr>
<td>Cursor and coordinate values but not the menus</td>
<td><code>GRAPH</code> or <code>GRAPH</code></td>
</tr>
</tbody>
</table>
Graphing a Family of Curves

If you enter a list as an element in an equation, the TI-86 plots the function for each value in the list, graphing a family of curves. In SimulG graphing order mode, the TI-86 graphs all functions sequentially for the first element in each list, then for the second element, and so on.

For example, \( \{2,4,6\} \sin x \) graphs three functions: \( 2 \sin x \), \( 4 \sin x \), and \( 6 \sin x \).

The equation \( \{2,4,6\} \sin (\{1,2,3\} x) \) also graphs three functions: \( 2 \sin x \), \( 4 \sin (2x) \), and \( 6 \sin (3x) \).

Smart Graph

Smart Graph displays the previously displayed graph when you press [GRAPH], as long as all factors that would cause reploting are unchanged since the graph was last displayed. Smart Graph replots if you performed any of these actions since the graph was last displayed.

- Changed a mode setting that affects graphs
- Changed a function or stat plot that was plotted on the last graph screen
- Selected or deselected a function or stat plot
- Changed the value of a variable in a selected function
- Changed the value of a window variable setting
- Changed a graph format setting

When you use more than one list in an expression, all lists must have the same dimension.
Graph Tools

Graph Tools on the TI-86 ....................................................... 88
Tracing a Graph ........................................................................ 90
Resizing the Graph Screen with ZOOM Operations ............... 91
Using Interactive Math Functions ........................................... 95
Evaluating a Function for a Specified x................................. 101
Drawing on a Graph .............................................................. 101
Graph Tools on the TI-86

Chapter 5 describes how to use the GRAPH menu items \( y(x) = \), WIND, GRAPH, and FORMT to define and display the graph of a function in Func graphing mode. This chapter describes how to use the other GRAPH menu items to apply preset graph screen dimensions, explore the graph and trace specific functions, perform mathematical analyses, draw on graphs, and store and recall graphs and drawings. You can use most graph tools in all four graphing modes.

The GRAPH Menu

This is the GRAPH menu in Func graphing mode. The GRAPH menu differs slightly from graphing mode to graphing mode.

- **ZOOM** Displays the GRAPH ZOOM menu; use these items to apply preset graph screen dimensions
- **TRACE** Activates the trace cursor; use this cursor to trace along graphs of specific functions
- **MATH** Displays the GRAPH MATH menu; use this menu to explore graphs mathematically
- **DRAW** Displays the GRAPH DRAW menu; use this menu to draw on graphs
- **STGDB** Displays the Name= prompt and GDB menu; use this prompt to enter a GDB variable
- **RCGDB** Displays the Name= prompt and GDB menu; use this menu to recall a GDB variable
- **EVAL** Displays the Eval x= prompt; use this prompt to enter an x value for which you want to solve the current function
- **STPIC** Displays the Name= prompt and PIC menu; use this prompt to enter a PIC variable
- **RCPIC** Displays the Name= prompt and PIC menu; use this menu to recall PIC variable
Using the Free-Moving Cursor

When you select `GRAPH` from the `GRAPH` menu, the graph screen is displayed with the free-moving cursor at the center of the screen.

The cursor appears as a plus sign with a flashing center pixel. To move the cursor, press `[`, `,`, `]`, or `|`; it moves in the direction of the cursor key you press.

- In `RectGC` format, each cursor movement updates the variables `x` and `y`. In `PolarGC` format, each cursor movement updates `x`, `y`, `R`, and `θ`.
- In `CoordOn` format, the `x` and `y` cursor coordinates are displayed at the bottom of the graph screen as you move the cursor.

Graphing Accuracy

The coordinate values displayed as you move the cursor approximate actual mathematical coordinates, accurate to within the width and height of the pixel. As the difference between `xmin` and `xmax` and between `ymin` and `ymax` becomes smaller (for example, when you zoom in on a graph), graphing is more accurate and coordinate values approximate the actual mathematical coordinates more closely.

The free-moving cursor coordinates represent the cursor location on the graph screen. Moving the free-moving cursor precisely from one plotted point to the next along a function is very difficult. To move along a function easily, use the trace cursor (page 90).

In the example, the function \( y(x) = x^3 + 0.3x^2 - 4x \) is graphed.
Tracing a Graph

To display the graph and begin a trace, select TRACE from the GRAPH menu.

The trace cursor appears as a small square with a flashing diagonal line at each corner. Initially, the trace cursor appears on the first selected function, at the x value closest to the middle of the screen.

If CoordOn format is selected, the cursor coordinates are displayed at the bottom of the screen.

To move the trace cursor...

<table>
<thead>
<tr>
<th>Press these keys:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press these keys:</td>
</tr>
<tr>
<td>Press these keys:</td>
</tr>
<tr>
<td>Press these keys:</td>
</tr>
</tbody>
</table>

| Press these keys: |
| Press these keys: |
| Press these keys: |
| Press these keys: |

As you move the trace cursor along a function, the y value is calculated from the x value. That is, y=yn(x). When you trace beyond the top or bottom of the graph screen, the coordinates displayed on the screen continue to change as if the cursor were still on the screen.

Panning: To view function coordinates to the left or right of the current graph screen, press and hold [2nd] or [3] while tracing. When you pan beyond the left or right side of the screen during a trace, the TI-86 automatically changes the values of xMin and xMax.
Quick Zoom: While tracing, you can press [ENTER] to adjust the graph screen so that the trace cursor location becomes the center of a new graph screen, even if you have moved the cursor beyond the top or bottom of the display. In effect, this is vertical panning.

Stopping and Resuming a Trace
To stop tracing and restore the free-moving cursor, press [CLEAR] or [GRAPH].

To resume tracing, select TRACE from the GRAPH menu. If Smart Graph has not replotted the graph (Chapter 5), the trace cursor is at the point where you stopped tracing.

Resizing the Graph Screen with ZOOM Operations
The standard TI-86 graph screen displays the portion of the xy plane defined by the values stored to the window variables. With the GRAPH ZOOM menu items, you can change some or all of the window variable values and redisplay the graph, usually with one keystroke. As a result, a smaller or larger portion of the xy plane is displayed.

The GRAPH ZOOM Menu

<table>
<thead>
<tr>
<th>y(x)=</th>
<th>WIND</th>
<th>ZOOM</th>
<th>TRACE</th>
<th>GRAPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOX</td>
<td>ZIN</td>
<td>ZOUT</td>
<td>ZSTD</td>
<td>ZPREV</td>
</tr>
</tbody>
</table>

- ZFIT
- ZSQR
- ZTRIG
- ZDECM
- ZDATA
- ZRCL
- ZFACT
- ZOOMX
- ZOOMY
- ZINT
- ZSTO
## Chapter 6: Graph Tools

To cancel the effect of any ZOOM menu item and return to the default window variable values, select **ZSTD**.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BOX</strong></td>
<td>Draws a box to define the graph screen</td>
</tr>
<tr>
<td><strong>ZIN</strong></td>
<td>(zoom in) Magnifies the graph around the cursor by factors of ( x_{\text{Fact}} ) and ( y_{\text{Fact}} )</td>
</tr>
<tr>
<td><strong>ZOUT</strong></td>
<td>(zoom out) Displays more of the graph around the cursor by factors of ( x_{\text{Fact}} ) and ( y_{\text{Fact}} )</td>
</tr>
<tr>
<td><strong>ZSTD</strong></td>
<td>Displays the graph in standard dimensions; resets the default window variable values</td>
</tr>
<tr>
<td><strong>ZPREV</strong></td>
<td>Reverses the last zoom operation; window variables revert to previous values</td>
</tr>
<tr>
<td><strong>ZFIT</strong></td>
<td>Recalculates ( y_{\text{Min}} ) and ( y_{\text{Max}} ) to include the minimum and maximum ( y ) values of the selected functions between the current ( x_{\text{Min}} ) and ( x_{\text{Max}} )</td>
</tr>
<tr>
<td><strong>ZSQR</strong></td>
<td>Sets equal-size pixels on the ( x )-axis and ( y )-axis; adjusts window variable values in one direction so that ( \Delta x = \Delta y ), while ( x_{\text{Scl}} ) and ( y_{\text{Scl}} ) remain unchanged; the midpoint of the current graph (not the axes intersection) becomes the midpoint of the new graph</td>
</tr>
<tr>
<td><strong>ZTRIG</strong></td>
<td>Sets built-in window variables appropriate for trigonometric functions in <strong>Radian</strong> mode: ( x_{\text{Min}} = 8.24668071567 ), ( x_{\text{Max}} = 8.24668071567 ), ( y_{\text{Min}} = -4 ), ( y_{\text{Max}} = 4 ), ( x_{\text{Scl}} = 1.5707963267949 ) (( \pi / 2 ))</td>
</tr>
<tr>
<td><strong>ZDECM</strong></td>
<td>Sets ( \Delta x = 1 ), ( \Delta y = 1 ), ( x_{\text{Min}} = 6.3 ), ( x_{\text{Max}} = 6.3 ), ( x_{\text{Scl}} = 1 ), ( y_{\text{Min}} = -3.1 ), ( y_{\text{Max}} = 3.1 ), and ( y_{\text{Scl}} = 1 )</td>
</tr>
<tr>
<td><strong>ZDATA</strong></td>
<td>Sets window variable values to display all statistical data points; adjusts ( x_{\text{Min}} ) and ( x_{\text{Max}} ) only; applies to histograms, scatter plots, and stat plots only (Chapter 14)</td>
</tr>
<tr>
<td><strong>ZRCL</strong></td>
<td>Uses window variable values stored in the user-defined zoom-window variables (<strong>ZSTO</strong>)</td>
</tr>
<tr>
<td><strong>ZFACT</strong></td>
<td>Displays the ZOOM FACTORS screen</td>
</tr>
<tr>
<td><strong>ZOOMX</strong></td>
<td>Zooms out by a factor of ( x_{\text{Fact}} ) only; ignores ( y_{\text{Fact}} ) (page 93)</td>
</tr>
<tr>
<td><strong>ZOOMY</strong></td>
<td>Zooms out by a factor of ( y_{\text{Fact}} ) only; ignores ( x_{\text{Fact}} )</td>
</tr>
<tr>
<td><strong>ZINT</strong></td>
<td>Sets integer values on the axes; sets ( \Delta x = 1 ), ( \Delta y = 1 ), ( x_{\text{Scl}} = 10 ), and ( y_{\text{Scl}} = 10 ); the current cursor becomes the center of the new graph screen after you press ( \text{ENTER} )</td>
</tr>
<tr>
<td><strong>ZSTO</strong></td>
<td>Stores current window variable values to user-defined zoom-window variables (<strong>ZRCL</strong>)</td>
</tr>
</tbody>
</table>

If you graph a circle but it appears elliptical, you can use **ZSQR** to reset the window variable values so that the circle graph appears circular.
Defining a Custom Zoom In

Using BOX, you can zoom in on any rectangular area within the current graph screen.

1. Select BOX from the GRAPH ZOOM menu. The zoom cursor is displayed at center screen.

2. Move the cursor to any spot you want to define as a corner of the zoom box; mark the corner with a small square.

3. Move the cursor away from the first corner, creating an adjustable box whose diagonal corners are the small square and the cursor.

4. When you have defined the box, replot all selected functions in the new graph screen.

5. Clear the menus from the screen.

Setting Zoom Factors

Zoom factors define the magnification or reduction factor by which ZIN, ZOUT, ZOOMX, and ZOOMY zoom in or zoom out around a point. To display the zoom factors editor, select ZFACT from the GRAPH ZOOM menu (press GRAPH F3 MORE MORE F2). xFact and yFact must be \( \geq 1 \). The default value for both factors is 4 in all graphing modes.

Zooming In and Zooming Out on a Graph

ZIN magnifies the part of the graph surrounding the cursor location. ZOUT displays a greater portion of the graph, centered on the cursor location. xFact and yFact determine the extent. The steps below describe how to use ZIN. To use ZOUT, select it instead of ZIN in step 2.
In the example, the function $y(x) = x^3 + 0.3x^2 - 4x$ is graphed.

1. Check $x_{\text{Fact}}$ and $y_{\text{Fact}}$; change as needed.

2. Select ZIN from the GRAPH ZOOM menu to display the zoom cursor.

3. Move the zoom cursor to the intended new center point of the graph screen.

4. Zoom in. The TI-86 adjusts the graph screen by $x_{\text{Fact}}$ and $y_{\text{Fact}}$, updates window variable values, and replots the selected functions centered on the cursor location.

You can continue to zoom in (or zoom out) on the current graph, unless you press a key other than $\text{ENTRY}$, $\text{OK}$, $\text{DEL}$, or $\text{CLR}$.

- To zoom in (or zoom out) again at the same point, press $\text{ENTRY}$.
- To zoom in (or zoom out) at a new center point, move the cursor and press $\text{ENTRY}$.

To zoom out only on the horizontal axis by a factor of $x_{\text{Fact}}$, select ZOOMX instead of ZIN in step 2 above. ZOOMX plots the selected functions centered on the cursor location and updates some window variable values; $y_{\text{Min}}$ and $y_{\text{Max}}$ are unchanged.

To zoom out only on the vertical axis by a factor of $y_{\text{Fact}}$, select ZOOMY instead of ZIN in step 2 above. ZOOMY plots the selected functions centered on the cursor location and updates some window variable values; $x_{\text{Min}}$ and $x_{\text{Max}}$ are unchanged.
Storing and Recalling Zoom-Window Variable Values

- To store all current zoom-window variable values simultaneously as a user-defined custom zoom feature, select ZSTO from the GRAPH ZOOM menu.
- To execute a user-defined custom zoom, which resets the graph screen to the stored zoom-window variables, select ZRCL from the GRAPH ZOOM menu.

Using ZSTO in these graphing modes: Stores to these zoom-window variables:

<table>
<thead>
<tr>
<th>Graphing Mode</th>
<th>Zoom-Window Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Func, Pol, Param, and DifEq</td>
<td>zxMin, zxMax, zxScl, zyMin, zyMax, and zyScl</td>
</tr>
<tr>
<td>Pol</td>
<td>zMin, zMax, and zStep</td>
</tr>
<tr>
<td>Param</td>
<td>ztMin, ztMax, and ztStep</td>
</tr>
<tr>
<td>DifEq</td>
<td>ztMin, ztMax, ztStep, and ztPlot</td>
</tr>
</tbody>
</table>

Using Interactive Math Functions

When you select a GRAPH MATH operation, Smart Graph displays the current graph with the trace cursor. To perform the GRAPH MATH operation, press [ and ] to move to the function.

When a GRAPH MATH menu operation prompts you to specify left bound, right bound, and guess, the accuracy of the values you specify will affect the length of time the TI-86 spends calculating the answer; the better the guess, the shorter the calculation time.
Chapter 6: Graph Tools

The GRAPH MATH menu differs slightly for Pol and Param graphing modes (Chapters 8 and 9).

DiffEq graphing mode has no GRAPH MATH menu.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>Finds the root of a function using a specified left bound, right bound, and guess</td>
</tr>
<tr>
<td>dy/dx</td>
<td>Finds a numeric derivative (slope) of a function at the trace cursor location</td>
</tr>
<tr>
<td>( f(x) )</td>
<td>Finds a function's numerical integral using a specified left bounds and right bound</td>
</tr>
<tr>
<td>FMIN</td>
<td>Finds a function's minimum using a specified left bound, right bound, and guess</td>
</tr>
<tr>
<td>FMAX</td>
<td>Finds a function's maximum using a specified left bound, right bound, and guess</td>
</tr>
<tr>
<td>INFLC</td>
<td>Finds a function's inflection point using a specified left bound, right bound, and guess</td>
</tr>
<tr>
<td>YICPT</td>
<td>Finds a function's y-intercept (( y ) at ( x=0 ))</td>
</tr>
<tr>
<td>ISECT</td>
<td>Finds the intersection of two functions using a specified left bound, right bound, and guess</td>
</tr>
<tr>
<td>DIST</td>
<td>Finds the straight-line distance between a specified left bound and right bound</td>
</tr>
<tr>
<td>ARC</td>
<td>Finds the distance along a function between two specified points on the function</td>
</tr>
<tr>
<td>TANLN</td>
<td>Draws the tangent line at a specified point</td>
</tr>
</tbody>
</table>

Settings That Affect GRAPH MATH Operations

- The tolerance variable \( \text{tol} \) (Appendix) affects the accuracy of \( f(x) \), FMIN, FMAX, and ARC. Accuracy increases as the tolerance value becomes smaller.
- The step-size variable \( \delta \) (Appendix) affects the accuracy of \( \frac{dy}{dx} \), INFLC in \( dxNDer \) differentiation mode (Chapter 1), ARC, and TANLN. Accuracy increases as the step-size value becomes smaller.
- The differentiation mode setting affects \( \frac{dy}{dx} \), INFLC, ARC, and TANLN; \( dxDer1 \) (exact) mode is more accurate than \( dxNDer \) (numeric) mode (Chapter 1).
Using ROOT, FMIN, FMAX, or INFLC

The steps for ROOT, FMIN, FMAX, and INFLC are the same, except for the menu selection in step 1.

1. Select ROOT from the GRAPH MATH menu. A Left Bound? prompt is displayed.
2. Move the cursor onto the function for which you want to find a root.
3. Specify the left bound for x. Either move the trace cursor to the left bound or enter a value directly. Right Bound? is displayed.
4. Specify the right bound for x as in step 3. Guess? is displayed.
5. Guess an x value near the root between the left bound and the right bound. Either move the cursor or enter a value.
6. Solve for x. The result cursor is displayed at the solution point, the cursor coordinate values are displayed, and the x value is stored in Ans.
Using f(x), DIST, or ARC

The steps for using f(x), DIST, and ARC are the same, except for the menu selection in step 1.

1. Select DIST from the GRAPH MATH menu. The current graph is displayed with a Left Bound? prompt.
2. Move the cursor onto the function on which the left bound is a point.
3. Select the left bound for x. Either move the cursor to the left bound or enter the x value. Right Bound? is displayed.
4. (DIST only) If you want the right bound to be a point on another function, move the cursor to the other function.
5. Select the right bound. Either move the cursor to the right bound or enter its x value.
   - For DIST, the solution DIST= is displayed and stored in Ans.
   - For ARC, the solution ARC= is displayed and stored in Ans.
   - For f(x), the solution f(x)= is displayed, shaded, and stored in Ans. The function integral error value is stored to the variable fnIntErr (Appendix). To remove the shading, select CLDRW from the GRAPH DRAW menu (page 103).
Using \( \frac{dy}{dx} \) or TANLN

The steps for using \( \frac{dy}{dx} \) and TANLN are the same, except for the menu selection in step 1.

1. Select \( \frac{dy}{dx} \) from the GRAPH MATH menu. The current graph is displayed.
2. Move the cursor to the function with the point for which you want to find the derivative, or slope.
3. Move the cursor to the point (or enter the \( x \) value).
4. Solve.
   - For \( \frac{dy}{dx} \), the solution \( \frac{dy}{dx} = \) is displayed and stored in Ans.
   - For TANLN, a tangent line also is displayed.
     To remove the tangent line and \( \frac{dy}{dx} = \) prompt, select CLDRW from the GRAPH DRAW menu.

\( y(x) = x^3 + 3x^2 - 4x \) is selected.

TANLN (GRAPH MATH menu) and TanLn (GRAPH DRAW menu) both draw a tangent line on the graph; only TANLN displays the solution, \( \frac{dy}{dx} \).
Using ISECT

In the example, the functions $y(x) = x^3 + 0.3x^2 - 4x$ and $y(x) = x^2 + 3x - 3$ are selected.

1. Select ISECT from the GRAPH MATH menu. The current graph is displayed with First Curve? at the bottom of the graph screen.

2. Select the first function (curve). The cursor moves to the next function and Second Curve? is displayed.

3. Select the second function (curve). Guess? is displayed.

4. Guess the intersection. Either move the cursor to a point near an intersection or enter an $x$ value.

5. Solve. The result cursor is displayed at the intersection, the cursor coordinates are the result, and the $x$ value is stored to Ans.

Using YICPT

To use YICPT, select YICPT from the GRAPH MATH menu (GRAPH MORE [F1] MORE [F2]). Press ▲ and ▼ to select a function, and then press ENTER. The result cursor is displayed at the $y$-intercept, the cursor coordinate values are displayed, and $y$ is stored in Ans.
### Evaluating a Function for a Specified x

1. **Select EVAL** from the GRAPH menu. The graph is displayed with the **Eval x=** prompt in the bottom-left corner.

2. **Enter a real x value between window variables xMin and xMax.**

3. **Evaluate.** The result cursor is on the first selected function at the entered x value. The coordinate values are displayed. The number in the top-right corner indicates which function is evaluated.

4. **Move the result cursor to the next or previous selected function.** The result cursor is on the next or previous function at entered x value, the coordinate values are displayed, and the function number changes.

### Drawing on a Graph

You can use the drawing tools (except DrInv) to draw points, lines, circles, shaded areas, and text on the current graph in any graphing mode. The drawing tools use the display’s x- and y-coordinate values.
Before Drawing on a Graph

All drawings are temporary; they are not stored in a graph database. Any action that causes Smart Graph to replot the graph erases all drawings. Therefore, before you use any drawing tool, consider whether you want to perform any of these graphing activities first.

- Change a mode setting that affects graphs
- Select, deselect, or edit a current function or stat plot
- Change the value of a variable used in a selected function
- Change a window variable value
- Change a graph format setting or graph style
- Clear current drawings with CLDRW

Saving and Recalling Drawn Pictures

To store the elements that define the current graph to a graph database (GDB) variable, select STGDB from the GRAPH menu. These information types are stored to a GDB variable:

- Equation editor functions
- Graph style settings
- Window variable values
- Format settings

To recall the stored GDB later, select RCGDB from the GRAPH menu, and then select the GDB variable from the GRAPH RCGDB menu. When you recall a GDB, the information stored in the GDB replaces any current information of these types.

To store the current graph display, including drawings, to a picture (PIC) variable, select STPIC from the GRAPH menu. Only the graph picture is stored to the specified PIC variable.

To superimpose one or more stored graph pictures onto a graph later, select RCPIC from the GRAPH menu, and then select the PIC variable from the GRAPH RCPIC menu.

Graph database (GDB) and picture (PIC) variable names can be from one to eight characters long. The first character must be a letter.

The next section describes how to draw lines, points, curves, and text onto a graph; you then can store the drawings to a PIC variable.
Clearing Drawn Pictures

To clear drawn pictures while the graph is displayed, select **CLDRW** from the GRAPH DRAW menu. The graph is replotted and displayed with no drawn elements.

To clear drawn pictures from the home screen, select **ClDrw** from the CATALOG. **ClDrw** is pasted to the cursor location. Press **ENTER**. **Done** is displayed; when you display the graph again, no drawings are displayed.

The GRAPH DRAW Menu

![Graph Draw Menu]

You can use these GRAPH DRAW menu items only on the home screen or in the program editor.

- **Shade**
  - **Shade(expression)**: Shades a specified area of a graph (See page 104)

- **DrawF**
  - **DrawF(expression)**: Draws expression as a function

- **PxOn**
  - **PxOn(row,column)**: Turns on the pixel at (row,column)

- **PxOff**
  - **PxOff(row,column)**: Turns off the pixel at (row,column)

- **PxChg**
  - **PxChg(row,column)**: Changes the on/off status of the pixel at (row,column)

- **PxTest**
  - **PxTest(row,column)**: Returns 1 if the pixel at (row,column) is on, or 0 if the pixel is off

- **TanLn**
  - **TanLn(expression,x)**: Draws expression as a function and a tangent line of expression at x

- **DrInv**
  - **DrInv(expression)**: Draws the inverse of expression

DrInv is not available in Pol, Param, or DifEq graphing modes.

For **PxOn**, **PxOff**, **PxChg**, and **PxTest**, **row** and **column** are integers, where 0 ≤ **row** ≤ 62 and 0 ≤ **column** ≤ 126.

For **DrawF**, **TanLn**, and **DrInv**, **expression** is in terms of **x**. Also, you cannot include a list in **expression** to draw a family of curves.
Shading Areas of a Graph

To shade an area of a graph, the syntax is:

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

- **pattern** specifies one of four shading patterns.
  1. vertical (default)
  2. horizontal
  3. negative slope (45°)
  4. positive slope (45°)

- **patternRes** specifies one of eight shading resolutions.
  1. every pixel (default)
  2. every second pixel
  3. every third pixel
  4. every fourth pixel
  5. every fifth pixel
  6. every sixth pixel
  7. every seventh pixel
  8. every eighth pixel

- The area that is specifically above `lowerFunc` and below `upperFunc` is shaded.
- `xLeft > xMin` and `xRight < xMax` must be true.
- `xLeft` and `xRight` specify left and right bounds for shading. (`xMin` and `xMax` are defaults.)

These `GRAPH DRAW` menu items are interactive. Also, you can use all of them, except `PEN`, on the home screen or in a program (A to Z Reference).

- **LINE** Draws a line segment from one point to another point you specify with the cursor
- **VERT** Draws a vertical line, which you can move to any displayed x value
HORIZ  Draws a horizontal line, which you can move to any displayed y value
CIRCL  Draws a circle with a center point and radius you specify with the cursor
PEN    Draws the path of the cursor as you move it on the graph screen
PTON   Turns on the point at the cursor location
PTOFF  Turns off the point at the cursor location
PTCHG  Changes the on/off status of a point at the cursor location
CLDRW  Clears all drawings from the graph screen; replots the graph
TEXT   Draws characters on the graph at the cursor location

**Drawing a Line Segment**

1. Select **LINE** from the GRAPH DRAW menu. The graph is displayed.
2. Define one segment endpoint with the cursor.
3. Define the other endpoint of the segment. As you move the cursor, a line anchored at the first defined endpoint extends to the cursor.
4. Draw the line.

To draw more line segments, repeat steps 2 and 3; to cancel **LINE**, press **CLEAR**.

*In the example, the functions \( y(x) = x^3 + 0.3x^2 - 4x \) and \( y(x) = x^2 + 3x \) are selected.*
Drawing a Vertical or Horizontal Line

1. Select **VERT** (or **HORIZ**) from the GRAPH DRAW menu. The graph is displayed and a vertical or horizontal line is drawn at the cursor.

2. Move the line to the \(x\) value (or to the \(y\) value, if horizontal) through which you want the line to pass.

3. Draw the line on the graph.

To draw more lines, repeat steps 2 and 3; to cancel **VERT** or **HORIZ**, press **CLEAR**.

Drawing a Circle

1. Select **CIRCL** from the GRAPH DRAW menu. The graph is displayed.

2. Define the center point of the circle with the cursor.

3. Move the cursor to any point on the intended circumference.

4. Draw the circle.

To draw more circles, repeat steps 2 through 4; to cancel **CIRCL**, press **CLEAR**.

In the example, the function \(y(x)=x^3+3x^2-4x\) is selected. Also, **ZIN** was executed once with the zoom cursor at (0,0), \(x\text{Fact}=2\), and \(y\text{Fact}=2\).
Drawing a Function, Tangent Line, or Inverse Function

For DrawF, TanLn, and DrInv, you can use as expression any variable to which a valid expression is stored (including deselected equation variables).

In the illustrations, \( y_1 = x^3 + 3x^2 - 4x \) is selected.

For DrawF, TanLn, and DrInv, \( \text{expression} \) is in terms of \( x \). When you select DrawF, TanLn, or DrInv from the GRAPH DRAW menu, it is pasted to the home screen or program editor. Upon execution, the drawing is returned. DrInv draws the inverse of \( \text{expression} \) by plotting its \( x \) values on the y-axis and its \( y \) values on the x-axis. DrInv is available only in Func graphing mode.

```
DrawF \( x^3 + 3x^2 - 4x \)
TanLn(y1, 1.5)
DrInv y1
```

Drawing Freehand Points, Lines, and Curves

1. Select PEN from the GRAPH DRAW menu.
2. Move the cursor to where you want to begin drawing.
3. Turn on the pen.
4. Draw whatever you want.
5. Turn off the pen.

To draw a diagonal line or curve, turn on the pen, press \( \begin{bmatrix} \text{ENTER} & \text{ENTER} & \text{ENTER} \end{bmatrix} \), press \( \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \), and so on, and repeat.

To draw more points, lines, or curves, repeat steps 2 through 5. To cancel, press CLEAR.
Placing Text on a Graph

1. Select TEXT from the GRAPH DRAW menu. The text cursor is displayed.
2. Move the cursor to where you want to enter text. Text is entered below the text cursor.
3. Set alpha-lock and enter min. (The alpha cursor ( eradicate ) is displayed in the top-right corner.
4. Move the cursor to another location.
5. Enter max (alpha-lock remains on).

Turning On or Turning Off Points

1. Select PTON (or PTOFF) from the GRAPH DRAW menu.
2. Move the cursor to where you want to draw (or erase) a point.
3. Turn on (or turn off) the point.

In the example, the function \( y(x) = x^3 + 3x^2 - 4x \) is selected. Also, ZSTD was executed. Points are turned on at \((-5,5), (5,5), (5,-5), \) and \((-5,-5)\).
7 Tables

Displaying the Table .................................................... 110
Setting Up the Table ..................................................... 113
Clearing the Table ....................................................... 114
Displaying the Table

The table displays the independent values and corresponding dependent values for up to 99 selected functions in the equation editor. Each dependent variable in the table represents a selected function stored in the equation editor for the current graphing mode.

To display the equation editor, press `GRAPH` (Chapter 5).

In the example, \( y_1 = x^2 + 3x - 4 \) and \( y_2 = \sin(3x) \) are selected and all defaults set. The table abbreviates values in the columns, if necessary.

To edit an equation, press `^` in the equation’s table column until the cursor highlights the equation variable on the top line, and then press `ENTER`. The expression stored to the current equation variable is displayed in the edit line.
### Independent and Dependent Variables in the Table

<table>
<thead>
<tr>
<th>Graphing Mode</th>
<th>Independent Variable</th>
<th>Dependent (Equation) Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Func</strong> (function)</td>
<td>x</td>
<td>y1 through y99</td>
</tr>
<tr>
<td><strong>Pol</strong> (polar)</td>
<td>θ</td>
<td>r1 through r99</td>
</tr>
<tr>
<td><strong>Param</strong> (parametric)</td>
<td>t</td>
<td>xt1/yt1 through xt99/yt99</td>
</tr>
<tr>
<td><strong>DifEq</strong> (differential equation)</td>
<td>t</td>
<td>Q1 through Q9</td>
</tr>
</tbody>
</table>

#### Navigating the Table

To... Do this:
- Display more dependent variables in the table  Press [X] or [Y]
- Display greater values in any column  Press [X] (only when *Indpnt: Auto* is set; page 112)
- Set *TblStart* to a lower value  Press [X] in the independent variable column until the cursor moves past the current *TblStart* (page 112)
- Display the equation in the edit line, where you can edit or deselect it  Press [X] or [Y] to move the cursor to an equation variable column, hold [X] until the cursor highlights the equation name, and then press [ENTER]; the equation is displayed in the edit line

*In DifEq mode, if an equation has an initial conditions list, the table uses the first list element to evaluate the equation (Chapter 10).*
The Table Menus  

The table has a unique menu for each graphing mode, as shown below.

**In Function Graphing Mode**

<table>
<thead>
<tr>
<th>TBLST</th>
<th>SELCT</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
</table>

**In Parametric Graphing Mode**

<table>
<thead>
<tr>
<th>TBLST</th>
<th>SELCT</th>
<th>t</th>
<th>xt</th>
<th>yt</th>
</tr>
</thead>
</table>

**In Polar Graphing Mode**

<table>
<thead>
<tr>
<th>TBLST</th>
<th>SELCT</th>
<th>θ</th>
<th>r</th>
</tr>
</thead>
</table>

**In Differential Equation Graphing Mode**

<table>
<thead>
<tr>
<th>TBLST</th>
<th>SELCT</th>
<th>t</th>
<th>Q</th>
</tr>
</thead>
</table>

**TBLST**  
Displays the table setup editor

**SELCT**  
On the edit line, deselects or cancels deselection of the equation

- x and y; θ and r; t, xt, and yt; or t and Q  
On the edit line, pastes the variable to the cursor location; the variables change according to graphing mode

- To add an equation to the table, select it in the equation editor (Chapter 5). **SELCT** only removes equations from the table.

- To remove an equation from a column in the table, select **SELCT** from the table menu. Remaining equations that follow the removed equation shift left one column.

- To deselect an equation with **SELCT**, the equation and cursor must be displayed in the edit line. If the equation is in the edit line but the cursor is not, press [ENTER].

- To compare two dependent variables not defined consecutively in the equation editor, use **SELCT** from the table screen menu to deselect the dependent variables in between.
Setting Up the Table

To display the table using the current table setup settings, select TABLE from the TABLE menu. The screen to the right shows the default table setup settings.

TblStart specifies the first independent variable value (x, θ, or t) in the table (only when Indpnt: Auto is selected).

@Tbl (table step) specifies the increment or decrement from one independent variable value to the next independent variable value in the table.

<table>
<thead>
<tr>
<th>TABLE SETUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TblStart = 0</td>
</tr>
<tr>
<td>@Tbl = 1</td>
</tr>
<tr>
<td>Indpnt: Auto Ask</td>
</tr>
</tbody>
</table>

Indpnt: Auto displays independent variable values automatically in the first column of the table, starting at TblStart.

Indpnt: Ask displays an empty table. As you enter x values in the x= prompt (x=value ENTER), each value is added to the independent variable column and the corresponding dependent variable values are calculated and displayed. When Ask is set, you cannot scroll beyond the six independent variable values that are currently displayed in the table.

TblStart and @Tbl must be real numbers; you can enter an expression.

In DfEq graphing mode, it is a good practice to set TblStart = tMin and @Tbl = tStep.
Viewing and Editing Dependent Variable Equations

1. Display the table.

2. Move the cursor into the column of the dependent variable you want to edit, and then move up the column until the name is highlighted.

3. Display the equation in the edit line.

4. Edit the equation.

5. Enter the edited equation.
   - The dependent variable values are recalculated.
   - The cursor returns to the edited dependent variable’s first value.
   - The equation editor is updated.

Clearing the Table

When you use `ClTbl` in a program, the table is cleared upon program execution (Chapter 16).
Preview: Polar Graphing

The graph of the polar equation $A \sin (B \theta)$ forms the shape of a flower. Graph the flower for $A=8$ and $B=2.5$. Then explore the appearance of the flower for other values of $A$ and $B$.

1. Select Pol mode from the mode screen.
2. Display the equation editor and polar equation editor menu.
3. (Deselect or delete all equations if any.)
   Store $r1(\theta) = 8 \sin (2.5 \theta)$.
4. Select ZSTD from the GRAPH ZOOM menu. $r1$ is plotted on the graph screen.
5. Display the window editor, and then change $\theta_{\text{Max}}$ to $4\pi$.
6. Select ZSQR from the GRAPH ZOOM menu. $x_{\text{Min}}$ and $x_{\text{Max}}$ are changed to display the graph in correct proportion.
7. Change the values of $A$ and $B$ and redisplay the graph.

To remove the GRAPH menu from the graph screen, as shown, press [CLEAR].

To redisplay the GRAPH menu, press [GRAPH].
Defining a Polar Graph

The steps for defining a polar graph are similar to the steps for defining a function graph. This chapter assumes that you are familiar with Chapter 5: Function Graphing and Chapter 6: Graph Tools. Chapter 8 details aspects of polar graphing that differ from function graphing.

Setting Polar Graphing Mode

To display the mode screen, press [2nd] [MODE]. To graph polar equations, you must select Pol graphing mode before you enter equations, set the format, or edit window variable values. The TI-86 retains separate equation, format, and window data for each graphing mode.

The GRAPH Menu

Chapter 5 describes these GRAPH menu items: GRAPH and FORMT.

Chapter 6 describes these GRAPH menu items: ZOOM, TRACE, DRAW, STGDB, RCGDB, EVAL, STPIC, and RCPIC.
Displaying the Polar Equation Editor

To display the polar equation editor, select \( r(\theta) \) from the GRAPH menu in Pol graphing mode (GRAPH [F1]). The polar equation editor menu displayed on the bottom line is the same as the Func mode equation editor menu, except that \( \theta \) and \( r \) replace \( x \) and \( y \).

In this editor, you can enter and display up to 99 polar equations, \( r_1 \) through \( r_{99} \), if sufficient memory is available. Equations are defined in terms of the independent variable \( \theta \).

The default graph style is \( \backslash \) (line) in Pol graphing mode. \( \backslash \) (shade above) and \( \backslash \) (shade below) graph styles are not available in Pol graphing mode.

Setting the Graph Screen Window Variables

To display the polar window editor, select WIND from the GRAPH menu (GRAPH [F2]). Pol graphing mode has the same window variables as Func graphing mode, except:

- \( x\text{Res} \) is not available in Pol graphing mode.
- \( \theta\text{Min}, \theta\text{Max}, \) and \( \theta\text{Step} \) are available in Pol graphing mode.

The values shown in the picture to the right are the defaults in Radian mode. \( \downarrow \) indicates that \( y\text{Min}=10, y\text{Max}=10, \) and \( y\text{Scl}=1 \) are beyond the screen.

\( \theta\text{Min}=0 \) Specifies the first \( \theta \) value to evaluate within the graph screen
\( \theta\text{Max}=6.28318530718 \) Specifies the last \( \theta \) value to evaluate within the graph screen
\( \theta\text{Step}=1.3089969389957 \) Specifies the increment from one \( \theta \) value to the next \( \theta \) value

\( \theta\text{Max} \) default is \( 2\pi \).
\( \theta\text{Step} \) default is \( \pi/24 \).
Setting the Graph Format

To display the format screen in Pol graphing mode, select FORMT from the GRAPH menu (GRAPH MORE [F3]). Chapter 5 describes the format settings. Although the same settings are available for Func, Pol, and Param graphing modes, the TI-86 retains in memory separate format settings for each mode. In Pol graphing mode, PolarGC shows the cursor coordinates in terms of r and θ, the variables that define the equations.

Displaying the Graph

To plot the selected polar equations, you can select GRAPH, TRACE, EVAL, RCGDB, or a ZOOM, MATH, DRAW, or RCPIC operation, from the GRAPH menu. The TI-86 evaluates r for each value of θ (from θMin to θMax in intervals of θStep) and then plots each point. As the graph is plotted, the variables θ, r, x, and y are updated.

Using Graph Tools in Pol Graphing Mode

The Free-Moving Cursor

The free-moving cursor in Pol graphing works the same as in Func graphing.

° In RectGC format, moving the cursor updates the values of x and y; if CoordOn format is selected, x and y are displayed.

° In PolarGC format, moving the cursor updates x, y, r, and θ; if CoordOn format is selected, r and θ are displayed.
Tracing a Polar Equation

To begin a trace, select TRACE from the GRAPH menu (press \texttt{GRAPH F4}). The trace cursor appears on the first selected equation at $\theta_{\text{Min}}$.

- In RectGC format, moving the trace cursor updates the values of $\theta$, $x$, and $y$; if CoordOn format is selected, $\theta$, $x$, and $y$ are displayed.
- In PolarGC format, moving the trace cursor updates $x$, $y$, $r$, and $\theta$; if CoordOn format is selected, $r$ and $\theta$ are displayed.

To move the trace cursor...

<table>
<thead>
<tr>
<th>Press:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{Along the graph of the equation by increments or decrements of $\theta_{\text{Step}}$}</td>
<td>$\Delta$ or $\nabla$</td>
</tr>
<tr>
<td>\texttt{From one equation to another}</td>
<td>$\Box$ or $\Box$</td>
</tr>
</tbody>
</table>

QuickZoom is available in Polar graphing; panning is not (Chapter 6).
Moving the Trace Cursor to a \( \theta \) Value

To move the trace cursor to any valid \( \theta \) value on the current equation, enter the number. When you enter the first digit, a \( \theta = \) prompt is displayed in the bottom-left corner. The value you enter must be valid for the current graph screen. When you have completed the entry, press [ENTER] to reactivate the trace cursor.

In the example, \( r_1 = 8 \sin(2.5 \theta) \) is graphed.

Values for \( \theta \), \( x \), and \( y \) are displayed on the graph to the right because RectGC graph format is selected.

Using Zoom Operations

The GRAPH ZOOM menu items, except ZFIT, work the same in Pol graphing as in Func graphing. In Pol graphing mode, ZFIT adjusts the graph screen in both the \( x \) and \( y \) directions.

The zoom operations affect only the \( x \) window variables (\( x\text{Min}, x\text{Max}, \) and \( X\text{scl} \)) and the \( y \) window variables (\( y\text{Min}, y\text{Max}, \) and \( Y\text{scl} \)), except ZSTO and ZRCL, which also affect the \( \theta \) window variables (\( \theta\text{Min}, \theta\text{Max}, \) and \( \theta\text{Step} \)).
The GRAPH MATH Menu

<table>
<thead>
<tr>
<th>MATH</th>
<th>DRAW</th>
<th>FORMT</th>
<th>STGDB</th>
<th>RCGDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIST</td>
<td>dy/dx</td>
<td>dr/d\theta</td>
<td>ARC</td>
<td>TANLN</td>
</tr>
</tbody>
</table>

The other GRAPH MATH menu items are the same as described in Chapter 6.

- \( \frac{dr}{d\theta} \): Finds the numerical derivative (slope) of a function at a point.
- The distances calculated by \( \text{DIST} \) and \( \text{ARC} \) are distances in the rectangular coordinate plane. \( \frac{dy}{dx} \) and \( \frac{dr}{d\theta} \) are independent of the RectGC or PolarGC format.
- At a point where the derivative is undefined, \( \text{TANLN} \) will draw the line, but no result is displayed or stored in \( \text{Ans} \).

**Evaluating an Equation for a Specified \( \theta \)**

When the trace cursor is not active, the GRAPH menu item \( \text{EVAL} \) evaluates selected polar equations directly on the graph for a given value of \( \theta \). \( \text{eval} \) in a program or from the home screen returns a list of \( r \) values.

**Drawing on a Polar Graph**

The GRAPH DRAW menu items work the same in Pol graphing as in Func graphing. DRAW instruction coordinates in Pol graphing mode are the x- and y-coordinates of the graph screen. \( \text{DrInv} \) is not available in Pol graphing mode.
Preview: Parametric Graphing

Graph the parametric equation that describes the path of a ball kicked at an initial speed of 30 meters per second, at an initial angle of 25 degrees with the horizontal (from ground level). How far does the ball travel? When does it hit the ground? How high does it go?

1. Select Param mode from the mode screen.

2. Display the equation editor and parametric equation editor menu. Deselect all equations and plots (if any are defined).

3. Define the path of the ball as $x_1$ and $y_1$ in terms of $t$.
   - Horizontal: $x_1=vt_0\cos(\theta)$
   - Vertical: $y_1=vt_0\sin(\theta)-\frac{1}{2}(gt^2)$
   - Gravity constant: $g=9.8 \text{ m/sec}^2$

4. Define the vertical component vector as $x_2$ and $y_2$ and define the horizontal component vector as $x_3$ and $y_3$.

5. Change the graph style of $x_3/y_3$ to  \( \blacktriangledown \) (thick). Change the graph style of $x_2/y_2$ and $x_1/y_1$ to  \( \oslash \) (path).

In the example, ignore all forces except gravity. For initial velocity $v_0$ and angle $\theta$, the position of the ball as a function of time has horizontal and vertical components.
Enter these window variable values.

<table>
<thead>
<tr>
<th>tMin</th>
<th>xMin</th>
<th>yMin</th>
<th>tMax</th>
<th>xMax</th>
<th>yMax</th>
<th>tStep</th>
<th>xScl</th>
<th>yScl</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>15</td>
<td>0.1</td>
<td>50</td>
<td>10</td>
</tr>
</tbody>
</table>

Set SimulG and AxesOff graphing formats, so the path of the ball and the vectors will be plotted simultaneously on a clear graph screen.

Plot the graph. The plotting action simultaneously shows the ball in flight and the vertical and horizontal component vectors of the motion.

Trace the graph to obtain numerical results. Tracing begins at tMin and traces the path of the ball over time. The value displayed for x is distance; y is height; t is time.

**Defining a Parametric Graph**

The steps for defining a parametric graph are similar to the steps for defining a function graph. This chapter assumes that you are familiar with Chapter 5: Function Graphing and Chapter 6: Graph Tools. This chapter details those aspects of parametric graphing that differ from function graphing.
Setting Parametric Graphing Mode

To display the mode screen, press [2nd] [MODE]. To graph parametric equations, you must select **Param** graphing mode before you enter equations, set the format, or edit window variable values. The TI-86 retains in memory separate equation, format, and window data for each graphing mode.

**The GRAPH Menu**

```
E(t)= WIND ZOOM TRACE GRAPH MATH DRAW FORMT STGDB RCGDB
```

```
GRAPH menu items: E(t)= WIND ZOOM TRACE GRAPH
```

```
GRAPH menu items: parametric parametric parametric
equation window graph math
```

**Displaying the Parametric Equation Editor**

To display the parametric equation editor, select **E(t)=** from the GRAPH menu in **Param** graphing mode (GRAPH [F1]). The equation editor menu displayed on the bottom line is the same as the **Func**-mode equation editor menu, except that **t** and **xt** replace **x** and **y**, and **yt** displaces **INSf**.

In this editor, you can enter and display both the **x** and **y** components of up to 99 parametric equations, **xt1** and **yt1** through **xt99** and **yt99**, if sufficient memory is available. Each is defined in terms of the independent variable **t**.

Two components, **x** and **y**, define a single parametric equation. You must define both **xt** and **yt** for each equation. The default graph style is ▼ (line) in **Param** mode. △ (shade above) and △ (shade below) graph styles are not available in **Param** mode.
Selecting and Deselecting a Parametric Equation

When a parametric equation is selected, the equals signs (=) of both $xt$ and $yt$ are highlighted. To change the selection status of a parametric equation, move the cursor onto either $xt$ or $yt$, and then select SELCT from the equation editor menu. The status is changed for $xt$ and $yt$.

Deleting a Parametric Equation

To delete a parametric equation using DELf, move the cursor to either $xt$ or $yt$, and then select DELf from the equation editor menu. Both components are deleted.

To delete a parametric equation using the MEM DELETE menu (Chapter 17), you must select the $xt$ component. If you select the $yt$ component, the equation is retained in memory.

Setting the Graph Screen Window Variables

To display the parametric window editor, select WIND from the GRAPH menu ([GRAPH 2]). Param graphing mode has the same window variables as Func graphing mode, except:
- $xRes$ is not available in Param mode.
- $tMin$, $tMax$, and $tStep$ are available in Param mode.

The values shown in the picture to the right are the defaults in Radian mode. $tMin=0$, $tMax=6.28318530718$, and $yMax=10$ and $yScl=1$ are beyond the screen.

- $tMin=0$ specifies the starting $t$ value.
- $tMax=6.28318530718$ specifies the ending $t$ value.
- $tStep=.1308996389957$ specifies the increment from one $t$ value to the next.
Chapter 9: Parametric Graphing

Setting the Graph Format
To display the format screen in Param graphing mode, select FORMT from the GRAPH menu \( [\text{GRAPH} \quad \text{MORE} \quad F3] \). Chapter 5 describes the format settings. The TI-86 retains in memory separate format settings for Func, Pol, Param, and DifEq graphing modes.

Displaying the Graph
To plot the selected parametric equations, you can select GRAPH, TRACE, EVAL, RCGDB, or a ZOOM, MATH, DRAW, or RCPIC operation. The TI-86 evaluates \( x \) and \( y \) for each value of \( t \) (from \( \text{tMin} \) to \( \text{tMax} \) in intervals of \( \text{tStep} \)) and then plots each point defined by \( x \) and \( y \). As the graph is plotted, the variables \( x \), \( y \), and \( t \) are updated.

Using Graph Tools in Param Graphing Mode

The Free-Moving Cursor
The free-moving cursor in Param graphing works the same as in Func graphing.

\[ \diamond \quad \text{In RectGC format, moving the cursor updates the values of } x \text{ and } y \text{; if CoordOn format is selected, } x \text{ and } y \text{ are displayed.} \]

\[ \diamond \quad \text{In PolarGC format, moving the cursor updates } x, y, r, \text{ and } \theta; \text{ if CoordOn format is selected, } r \text{ and } \theta \text{ are displayed.} \]

Tracing a Parametric Function
To begin a trace, select TRACE from the GRAPH menu \( [\text{GRAPH} \quad F4] \). When you begin a trace, the trace cursor is on the first selected function at \( \text{tMin} \).

\[ \diamond \quad \text{In RectGC format, moving the trace cursor updates the values of } x, y, \text{ and } t; \text{ if CoordOn format is selected, } t, x, \text{ and } y \text{ and are displayed.} \]
In PolarGC format, moving the trace cursor updates \( x \), \( y \), \( r \), \( \theta \), and \( t \); if CoordOn format is selected, \( r \), \( \theta \), and \( t \) are displayed. The \( x \) and \( y \) (or \( r \) and \( \theta \)) values are calculated from \( t \).

To move the trace cursor...

<table>
<thead>
<tr>
<th>Press:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Along the graph of the equation by increments or decrements of ( t )Step</td>
</tr>
<tr>
<td>From one equation to another</td>
</tr>
</tbody>
</table>

If you move the trace cursor beyond the top or bottom of the graph screen, the coordinate values at the bottom of the screen continue to change appropriately. If you have graphed a family of curves, \( \leftarrow \) and \( \Rightarrow \) move through each curve before moving to the next parametric function.

### Moving the Trace Cursor to a \( t \) Value

To move the trace cursor to any valid \( t \) value on the current equation, enter the number. When you enter the first digit, a \( t= \) prompt is displayed in the bottom-left corner. The value you enter must be valid for the current graph screen. When you have completed the entry, press \[\text{ENTER}\] to reactivate the trace cursor.

#### Using Zoom Operations

The GRAPH ZOOM menu items, except ZFIT, work the same in Param graphing as in Func graphing. In Param mode, ZFIT adjusts the graph screen in both the \( x \) and \( y \) directions.
The GRAPH ZOOM menu items affect only the x window variables (xMin, xMax, and xScl) and the y window variables (yMin, yMax, and yScl), except ZSTO and ZRCL, which also affect the t window variables (tMin, tMax, and tStep).

The GRAPH MATH Menu

<table>
<thead>
<tr>
<th>MATH</th>
<th>DRAW</th>
<th>FORMT</th>
<th>STGDB</th>
<th>RCGDB</th>
<th></th>
<th>TANLN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIST</td>
<td>dy/dx</td>
<td>dy/dt</td>
<td>dx/dt</td>
<td>ARC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The other GRAPH MATH menu items are the same as described in Chapter 5.

dy/dx Returns the derivative of yt divided by the derivative of xt

dy/dt Returns the derivative of the yt equation at a point with respect to t

dx/dt Returns the derivative of the xt equation at a point with respect to t

The distances calculated by DIST and ARC are distances in the rectangular coordinate plane.

At a point where the derivative is undefined, TANLN will draw the line, but no result is displayed or stored in Ans.

Evaluating an Equation for a Specified t

When the trace cursor is not active, the GRAPH menu item EVAL evaluates selected polar equations directly on the graph for a given value of t. eval in a program or from the home screen returns a list of x and y values in this form: {xt1(t) yt1(t) xt2(t) xt2(t) ...}.

Drawing on a Parametric Graph

The DRAW menu items work in Param graphing the same as in Func graphing. DRAW instruction coordinates in Param graphing are the x- and y-coordinate values of the graph screen.
10 Differential Equation Graphing

Defining a Differential Equation Graph ......................... 132
Entering and Solving Differential Equations .................... 139
Using Graph Tools in DifEq Graphing Mode ................. 144
Defining a Differential Equation Graph

Most steps for defining a differential equation graph are similar to the steps for defining a function graph. This chapter assumes that you are familiar with Chapter 5: Function Graphing and Chapter 6: Graph Tools. This chapter details aspects of differential equation graphing that differ from function graphing.

Generally, DifEq graphing mode differs from other graphing modes in these ways.

- You must select the field format or accept the default before defining the equations (page 133).
- If an equation is higher than first order, you must convert it to an equivalent system of first-order differential equations, and then store the system in the equation editor (page 140 and page 142).
- When FldOff field format is selected, you must set initial conditions for each equation in the system (page 136).
- After you have selected the field format setting, you must select AXES from the GRAPH menu and enter axes information or accept the defaults (page 137).

Setting Differential Equation Graphing Mode

To display the mode screen, press [2nd] [MODE]. To graph differential equations, you must select DifEq graphing mode before you set the format, enter equations, or edit window variable values. The TI-86 retains in memory separate format, equation, and window data for each graphing mode.
The GRAPH Menu

<table>
<thead>
<tr>
<th>Q'(t)</th>
<th>WIND</th>
<th>INITC</th>
<th>AXES</th>
<th>GRAPH</th>
<th>FORMT</th>
<th>DRAW</th>
<th>ZOOM</th>
<th>TRACE</th>
<th>EXPLR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- equation editor
- initial conditions editor
- differential equation window editor
- axes editor
- differential equation format screen
- explore with the free-moving cursor

Setting the Graph Format

To display the format screen in DifEq graphing mode, select FORMT from the GRAPH menu (GRAPH MORE F1).

- The RK Euler and SlpFld DirFld FldOff format settings are available only in DifEq mode.
- The RectGC PolarGC, DrawLine DrawDot, and SeqG SimulG format settings are not available in DifEq graphing mode.
- All other format settings are the same as described in Chapter 5.

Solution Method Format

- RK Uses the Runge-Kutta method to solve differential equations more accurately than the Euler solution method format, but not as fast.
- Euler Uses the Euler method to solve differential equations; requires a number of iterations between tStep values, so EStep= prompt replaces difTol= prompt on the window editor.
Field Format

**SlpFld**
(slope field) Adds the slope field to the graph of only one first-order equation with \( t \) on the x-axis and a specified \( Qn \) equation on the y-axis

**DirFld**
(direction field) Adds the direction field to the graph of only one second-order equation with \( Qx\# \) on the x-axis and \( Qy\# \) on the y-axis

**FldOff**
(field off) Graphs all selected differential equations with \( t \) or \( Q1 \) on the x-axis, \( Q1 \) or \( Q2 \) on the y-axis, and no field; initial conditions must be defined for all equations (page 136)

The examples below show the basic slope and direction fields; all unspecified settings and values are defaults. To replicate these examples, reset defaults, enter the specified information in \textit{DifEq} graphing mode, and then press 6*

\[ Q'1 = t \quad (y' = x) \]

\[ Q'1 = Q2 \quad \text{and} \quad Q'2 = Q1 \quad (y' = -y) \]

**Axes information is stored to GDB and PIC variables.**

To remove menus from a graph, as shown in the examples, press \texttt{CLEAR}.

**Displaying the Differential Equation Editor**

To display the differential equation editor, select \( Q'(t) \) from the \texttt{GRAPH} menu in \textit{DifEq} graphing mode (\texttt{GRAPH F1}). The \textit{DifEq} equation editor menu on the bottom line is the same as the \texttt{Func} mode equation editor menu, except that \( t \) and \( Q \) replace \( x \) and \( y \).
In this editor, you can enter and display a system of up to nine first-order differential equations, \( Q'1 \) through \( Q'9 \), if sufficient memory is available. Equations are defined in terms of the independent variable \( t \) and/or \( Q' \).

You can refer to another differential equation variable in a DifEq equation, as in \( Q'2=Q1 \). However, you cannot enter a list in a DifEq equation.

When the TI-86 calculates a differential equation system, it references all equations in the equation editor, regardless of selection status, starting at \( Q'1 \). You must define \( Q'n \) equation variables consecutively, starting at \( Q'1 \). For example, if \( Q'1 \) and \( Q'2 \) are not defined, but you attempt to solve an equation defined in \( Q'3 \), the calculator returns an error.

The TI-86 allows you to analyze each equation independently. For example, you can enter \( Q'1=t \) and \( Q'2=t^2 \) and analyze each equation independently.

The TI-86 graphs only those selected equations that are appropriate for the specified axes.

- The default graph style is \( \backslash \) (thick) in DifEq mode.
- \( \backslash \) (shade above), \( \backslash \) (shade below), and \( \cdot \) (dot) are not available in DifEq graphing mode.

**Setting the Graph Screen Window Variables**

To display the differential equation window editor, select WIND from the GRAPH menu (GRAPH [F2]). DifEq has the same window variables as Func graphing mode, except:

- \( xRes \) is not available in DifEq mode.
- \( tMin, tMax, tStep, \) and \( tPlot \) are available in DifEq mode.
- \( difTol \) (RK) and \( EStep \) (Euler) are available in DifEq mode.
The values shown in the picture on page 135 are defaults in Radian mode. x and y settings correspond to the axes variables (page 137). ↓ indicates that xScl=1, yMin=-10, yMax=10, yScl=1, and difTol=.001 (in RK format) or EStep=1 (in Euler format) are beyond the screen.

\[
\begin{align*}
t_{\text{Min}} &= 0 & \text{Specifies the } t \text{ value at which to begin evaluating within a graph screen} \\
t_{\text{Max}} &= 6.28318530718 & \text{Specifies the last } t \text{ value to evaluate within a graph screen} \\
t_{\text{Step}} &= 0.1308969389958 & \text{Specifies the increment from one } t \text{ value to the next } t \text{ value} \\
t_{\text{Plot}} &= 0 & \text{Specifies the point at which plotting begins (ignored when } t \text{ is an axis)} \\
d_{\text{ifTol}} &= .001 \text{ (in RK format)} & \text{Specifies tolerance to help select step size for solving; must be } \geq 1E-12 \\
\text{EStep} &= 1 \text{ (in Euler format)} & \text{Specifies Euler iterations between } t_{\text{Step}} \text{ values; must be an integer } >0 \text{ and } \leq 25
\end{align*}
\]

### Setting the Initial Conditions

To display the initial conditions editor, select INITC from the GRAPH menu (\textit{GRAPH F3}). On this editor, you can set the initial value at \( t=t_{\text{Min}} \) for each first-order equation in the equation editor.

- \( t_{\text{Min}} \) is the first \( t \) value to evaluate. \( Q_{\text{In}} \) is the initial value of \( Q_n \). A small square next to an initial condition variable indicates that a value is required for a defined differential equation.

You can enter an expression, list, or list name for initial conditions \( t_{\text{Min}} \) and \( Q_{\text{In}} \). When you enter a list name, the elements are displayed when you press \text{ENTER}, \text{↓} or \text{↑}.

- If \text{SipFld} or \text{DirFld} format is set, you need not specify initial conditions. The TI-86 returns the appropriate field with no specific solutions.
- If \text{FldOff} format is set, you must specify initial conditions.
Setting the Axes

To display the axes editor, select **AXES** from the **GRAPH** menu in **DifEq** mode (**GRAPH F4**).

- **x=** assigns a variable to the x-axis
- **y=** assigns a variable to the y-axis
- **dTime=** specifies a point in time (real number)
- **fldRes=** (resolution) sets number of rows (1 through 25)

At the **x=** and **y=** prompts, you can enter the independent variable **t**, as well as **Q, Q', Qn, or Q'n**, where **n** is an integer ≥ 1 and ≤ 9. If you assign **t** to one axis and **Qn** or **Q'n** to the other axis, only the equation stored to **Qn** or **Q'n** is plotted; other differential equations in the equation editor are not plotted; their selection status is ignored. **dTime** is only valid for second-order equations with **t** in either equation.

The axes editor and defaults for each field format are shown below. When **SlpFld** field format is set, the x-axis is always **t**, so the **AXES: SlpFld** editor does not display **x=t**.

Axes information is stored to **GDB** and **PIC** variables.

### Differential Equation Graphing Tips

- Since the TI-86 plots slope fields and direction fields before it plots equations, you can press **ENTER** to pause the graph and view the fields with no solutions plotted.
- If you do not specify initial conditions for the equations assigned to the axes, the TI-86 simply draws the field and stops. This gives you access to both the field format options and the interactive initial conditions simultaneously.
The Built-In Variable fldPic

As the TI-86 plots a field, it stores the field and any displayed label, axes, or cursor coordinate information to the built-in variable fldPic.

These actions do not update fldPic.

- Switching the solving method format from RK to Euler or from Euler to RK
- Entering or editing any initial condition variable value (Q1 through Q9)
- Editing a value for difTol, EStep, tMin, tMax, tStep, or tPlot
- Changing a graph style

These actions update fldPic.

- Editing an equation in the equation editor
- Re-assigning an axis, editing a dTime value, or editing a fldRes value
- Using a GRAPH ZOOM menu item
- Changing a format setting other than solving method format
- Editing a value for xMin, xMax, xScl, yMin, yMax, or yScl

Displaying the Graph

To plot the differential equations, you can select GRAPH, TRACE, EVAL, or STGDB, or a DRAW, ZOOM, or STPIC operation, from the GRAPH menu. The TI-86 solves each equation from tMin to tMax. If t is not an axis, it plots each point beginning at tPlot; otherwise, it begins at tMin. As a graph is plotted, the variables x, y, t, and Qn are updated.

tStep affects trace resolution and graph appearance, but not the accuracy of the trace values. tStep does not determine the step size for solving; using the RK algorithm (Runge-Kutta 2-3) determines the step size. If the x-axis is t, setting tStep = (tMax – tMin) / 126 increases plotting time without increasing accuracy.
Entering and Solving Differential Equations

In **Func** graphing mode, x is the independent variable and y is the equation variable. To avoid conflict between **Func** equations and **DifEq** equations on the TI-86, t is the independent variable and Q'n is the equation variable in **DifEq** graphing mode. Therefore, when you enter an equation in the differential equation editor, you must express it in terms of t and Q'n.

For example, to express the first-order differential equation y' = x^2, you would substitute t^2 for x^2 and Q'1 for y', and then enter Q'1 = t^2 in the equation editor.

**Graphing in SlpFld Format**

1. Display the mode screen and set **DifEq** graphing mode.
2. Display the format screen and set **SlpFld** field format.
3. Display the equation editor and store the differential equation y' = x^2, substituting Q'1 for y' and t for x. Clear any other equations.
4. Display the initial conditions editor and enter the initial conditions. A small square indicates that an initial condition is required.

In the example, the default window variable values are set initially.
In SlpFld field format, x=t is always true; y=Q1 and fldRes=15 are the default axes settings.

5. Display the axes editor and enter the equation variable for which you want to solve. (Do not set y=Q.)

6. Accept or change fldRes (resolution).

7. Display the graph. With the default window variable values set, the slope fields for this graph are not very illustrative.

8. Change the window variables xMin, xMax, yMin, and yMax.

9. Select TRACE from the GRAPH menu to replot the graph and activate the trace cursor. Trace the solution. The trace cursor coordinates for t and Q1 are displayed.

Transforming an Equation into a First-Order System

On the TI-86, to enter a second-order or higher (up to ninth-order) differential equation, you must transform it to a system of first-order differential equations. For example, to enter the second-order differential equation \( y'' = -y \), you must transform it to two first-order differential equations, as shown in the chart below.

<table>
<thead>
<tr>
<th>Differentiate...</th>
<th>Define the variables as...</th>
<th>And then substitute:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q'1 = y' )</td>
<td>( Q1 = y )</td>
<td>( Q'1 = Q2 ) (since ( Q'1 = y' = Q2 ))</td>
</tr>
<tr>
<td>( Q'2 = y'' )</td>
<td>( Q2 = y' )</td>
<td>( Q'2 = -Q1 )</td>
</tr>
</tbody>
</table>
Graphing in DirFld Format

1. Display the mode screen and set DifEq graphing mode.
2. Display the format screen and set DirFld graphing format.
3. Display the equation editor and store the transformed system of differential equations for \( y'' = y \) to the equation editor, substituting \( Q_1 \) for \( y \) and \( Q_2 \) for \( y' \).
4. Display the initial conditions editor and enter the initial conditions if you want a specific solution. To enter a list of initial conditions, use \{ \) and \} from the LIST menu.
5. Display the axes editor and enter the two equation variables for which you want to solve. You must omit the prime mark (').
6. Accept or change fldRes (resolution).
7. Select ZSTD from the GRAPH ZOOM menu to set the standard window variable values and display the graph.
8. Clear the GRAPH menu from the screen.

In DifEq graphing mode, \( t \) is the independent variable and \( Q^n \) is the dependent variable, where \( n \geq 1 \) and \( n \leq 9 \).

In the example, the default window variable values are set initially.

When DirFld field format is selected, \( x = Q_1 \), \( y = Q_2 \), \( dTime = 0 \), and \( fldRes = 15 \) are the default axes settings. Since \( t \) is not part of the equation, \( dTime \) is ignored.
Graphing a System of Equations in FldOff Format

For this example, you must transform the fourth-order differential equation $y^{(4)} - y = e^x$ into an equivalent system of first-order differential equations, as shown in the chart below.

<table>
<thead>
<tr>
<th>Differentiate...</th>
<th>Define the variables as...</th>
<th>And then substitute:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q'1 = y'$</td>
<td>$t = x$</td>
<td>$Q'1 = Q2$ (since $Q'1 = y' = Q2$)</td>
</tr>
<tr>
<td>$Q2 = y''$</td>
<td>$Q1 = y$</td>
<td>$Q2 = Q3$</td>
</tr>
<tr>
<td>$Q3 = y'''$</td>
<td>$Q2 = y'$</td>
<td>$Q3 = Q4$</td>
</tr>
<tr>
<td>$Q4 = y^{(4)}$</td>
<td>$Q3 = y''$</td>
<td>$Q4 = e^t + Q1$ (since $Q'4 = y^{(4)} = e^x + y = e^t + Q1$)</td>
</tr>
</tbody>
</table>

1. Display the mode screen and set DifEq graphing mode.
2. Display the format screen and set FldOff field format.
3. Display the equation editor and store the transformed system of differential equations for $y^{(4)} = e^x + y$, substituting as shown in the chart.
4. Deselect $Q'3$, $Q'2$, and $Q'1$ to plot $Q'4 = e^t(t) + Q1$ only.

In DifEq graphing mode, $t$ is the independent variable and $Q_n$ is the equation variable, where $n \geq 1$ and $\leq 9$. 
5. Display the window editor and set the window variable values.

\[ f(x) = 0 \]

6. Display the initial conditions editor and enter the initial conditions. A small square indicates that an initial condition is required.

\[ (3, 5) \]

7. Display the axes editor. Enter the equation variables for which you want to solve.

\[ y = Q \]

8. Display the graph. Explore the equation with the trace cursor.

9. Enter a \( t \) value to move the trace cursor to the solution for that \( t \) value. The \( t \) and \( Q \) coordinates are displayed.

When FldOff field format is selected, \( x=t \) and \( y=Q \) are the default axes settings.
Solving a Differential Equation for a Specified Value

On the home screen in DifEq graphing mode, you can solve a differential equation stored to a specified independent variable value or expression. The syntax is: \( Q'(value) \).

- The equation must be stored to a DifEq equation variable (\( Q'1 \) through \( Q'9 \)).
- The initial conditions must be defined.
- The result sometimes varies, depending on the axes settings.

Using Graph Tools in DifEq Graphing Mode

The Free-Moving Cursor

The free-moving cursor works in DifEq mode as it does in Func graphing. The cursor coordinate values for \( x \) and \( y \) are displayed, and the variables are updated.

Tracing a Differential Equation

To begin a trace, select TRACE from the GRAPH menu (\( 6 \)). The trace cursor appears on the first equation at or near \( tPlot \) (or \( tMin \), if \( t \) is an axis).

The trace coordinates displayed at the bottom of the screen reflect the axes settings. For example, if \( x=t \) and \( y=Q1 \), then \( t \) and \( Q1 \) are displayed. If \( t \) is not an axis, three trace values are displayed. If \( t \) is an axis, only \( t \) and the variable designated as the y-axis are displayed.

The trace cursor moves in increments or decrements of \( tStep \). As you trace an equation, the coordinates are updated and displayed. If the cursor moves off the screen, the coordinate values displayed at the bottom of the screen continue to change appropriately.
Moving the Trace Cursor to a \( t \) Value

To move the trace cursor to any valid \( t \) value on the current equation, enter the number. When you enter the first digit, a \( t= \) prompt is displayed in the bottom-left corner. The value you enter must be valid for the current graph screen. When you have completed the entry, press [ENTER] to reactivate the trace cursor.

Drawing on a Differential Equation Graph

The GRAPH DRAW menu items work the same in DifEq graphing mode as in Func graphing. DRAW instruction coordinates are the \( x \)- and \( y \)-coordinates of the graph screen.

\textbf{DrEqu} is available only in DifEq mode. \textbf{DrInv} is not available in DifEq graphing mode.

\textbf{Drawing an Equation and Storing Solutions to Lists}

To draw a solution on the current graph screen and store the results to specified list names, the syntax is:

\begin{verbatim}
DrEqu(xAxisVariable,yAxisVariable,xList,yList,tList)
\end{verbatim}

\textit{xAxisVariable} and \textit{yAxisVariable} specify the axes on which the drawing is based; they may differ from the current graph screen's axes settings.
Chapter 10: Differential Equation Graphing

DrEqu( does not store values to x, y, or t.

\[ xList, yList, \text{ and } tList \text{ are optional list names to which you can store the solutions } x, y, \text{ and } t. \] You then can display the lists on the home screen or in the list editor (Chapter 11).

Use the free-moving cursor to select initial conditions.

You cannot trace the drawing. However, you can plot \( xList, yList, \text{ or } tList \) as a stat plot after you draw the equation, and then trace them (Chapter 14). Also, you can fit statistical regression models to the lists (Chapter 14).

1. Display the mode screen and set DifEq graphing mode.
2. Display the format screen and set DirFld field format.
3. Display the equation editor and store the equations \( Q'1=Q2 \) and \( Q'2=Q1. \) (Delete all other equations.)
4. Remove the format screen, and then select DrEq from the GRAPH DRAW menu. DrEq( is pasted to the home screen.
5. Assign variables to the x- and y-axes.
6. Specify list names to which to store the solution lists for x, y, and t.

In the example, the default window variable values are set. If necessary, select ZSTD from the GRAPH ZOOM menu.

If you select FldOff field format, you must enter initial conditions before you use DrEq. DrEq( does not store values to x, y, or t.

In the example, the default window variable values are set. If necessary, select ZSTD from the GRAPH ZOOM menu.
In the example, since no initial conditions were set, the equation in Q1 is not plotted.

1. Display the graph screen and plot the direction field.

2. Move the free-moving cursor to the initial condition coordinates you want.

3. Draw the solution. The solution lists for x, y, and t are stored to LX, LY, and LT. The Again? prompt is displayed and ALPHA-lock is on for [Y] and [N] only.
   * To use DrEqu( again with new initial conditions, press [Y], [X], [C], or [D].
   * To leave DrEqu( and display the GRAPH menu, press [N] or EXIT.

**Using ZOOM Operations**

The GRAPH ZOOM menu items, except ZFIT, work the same in DifEq graphing mode as in Func graphing mode. In DifEq graphing mode, ZFIT adjusts the graph screen in both the x direction and y direction.

The ZOOM menu items affect only the x (xMin, xMax, and xScl) and y (yMin, yMax, and yScl) window variables. The t window variables (tMin, tMax, tStep, and tPlot) are not affected, except with ZSTD and ZRCL. You may want to edit the t window variables to ensure that sufficient points are plotted. ZSTD sets difTol=.001 and t and Q as the axes.
Drawing Solutions Interactively with EXPLR

1. Display the mode screen and set **DifEq** graphing mode.

2. Display the format screen and set **FldOff** field format.

3. Display the equation editor and store the equation \( Q' = 0.001Q(100 - Q) \). (Delete all other equations.)

4. Set the axes to \( x = t \) and \( y = Q \).

5. Display the window editor and set the window variable values.

6. Display the initial conditions editor and enter the initial condition.
Select EXPLR from the GRAPH menu.

Move the free-moving cursor to the initial condition for which you want to solve.

Draw the solution to Q1, using the cursor coordinates (x,y) as initial condition (t,Q'(t)).

To continue drawing more solutions, move the free-moving cursor and then press [ENTER].

To stop using EXPLR, press [EXIT].

If SlpFld or DirFld is set, the axes are set to specific solutions automatically.
- For SlpFld, x=t and y=Q1 are set.
- For DirFld, x=Q1 and y=Q2 are set.

If the axes are set to a specific solution t, Qn, or Q'n, that solution is drawn.

If the axes are not set to a specific solution and t is one variable and Q is the other, Q1 is drawn.

If both axes are set to a Q variable, executing EXPLR results in an error.
Evaluating Differential Equations for a Specified t

When the trace cursor is not active, the GRAPH menu item `EVAL` evaluates currently selected differential equations $Q_n$ for a specified value of $t$, $t_{\text{Min}} \leq t \leq t_{\text{Max}}$. You can use it directly on the graph. In a program or from the home screen, `eval` returns a list of $Q$ values.

When `DirFld` or `SlpFld` field format is set, you must specify initial conditions before using `EVAL`. 
11 Lists

Lists on the TI-86 .................................................. 152
Creating, Storing, and Displaying Lists ..................... 153
The List Editor ....................................................... 156
Using List Operations ........................................... 159
Using Mathematical Functions with Lists .................. 161
Attaching a Formula to a List Name ........................... 162
Lists on the TI-86

A list is a set of real or complex elements, as in \(\{5, 20, 13, 9\}\). On the TI-86, you can:

- Enter a list directly in an expression (page 153)
- Enter a list and store it to a list name (variable) (page 154)
- Enter a list name in the list editor (page 156), and then enter elements directly or use an attached formula to generate them automatically (page 161)
- Collect data with the Calculator-Based Laboratory™ (CBL 2™/CBL™) or Calculator-Based Ranger™ (CBR) and store it to a list name on the TI-86 (Chapter 18)
- Create lists dynamically using the LIST OPS menu item \texttt{seq} (page 159)

On the TI-86, you can use a list:

- As a set of values for an argument in a function to return a list of answers (Chapter 1)
- As part of an equation to graph a family of curves (Chapter 5)
- As a set of statistical data to analyze with statistical functions and plot on the graph screen (Chapter 14)

The LIST Menu

When you enter a list, \{ (open brace) specifies the beginning and \} (close brace) specifies the end. To paste \{ or \} to the cursor location, select either from the LIST menu.
The LIST NAMES menu shown here has no user-created list names.

Chapter 14 describes fStat, xStat, and yStat.

Each user-created list name is added to the LIST NAMES menu and VARS LIST screen. List names, including fStat, xStat, and yStat, are sorted in alphanumeric order in both places.

Creating, Storing, and Displaying Lists

Entering a List Directly in an Expression

To enter a list directly, the syntax is: \( \{element_1, element_2, ..., element_n\} \)

1. Enter any part of the expression that precedes the list.
2. Select \{ from the LIST menu to begin the list.
3. Enter each list element, separating each from the other with a comma. Each list element can be an expression.
4. Select } from the LIST menu to end the list.
5. Enter any part of the expression that follows the list.
6. Evaluate the expression. Any elements that are expressions are evaluated first.

An ellipsis (...) indicates that a list continues beyond the screen. Use \[ and \] to scroll the list.
Chapter 11: Lists

Creating a List Name by Storing a List
To store a list, the syntax is: \{element1,element2, ... ,element n\} \rightarrow \text{listName}

1. Enter a list directly. (To store a result expressed as a list and currently stored in \text{Ans}, as shown in the example, begin these steps at step 2.)
2. Paste \rightarrow to the cursor location. ALPHA-lock is on.
3. Enter the list name. Either select a name from the \text{LIST NAMES} menu or directly enter a name one to eight characters long, starting with a letter.
4. Store the list to the list name.

Displaying List Elements Stored to a List Name
1. Enter the list name on the home screen; either select it from the \text{LIST NAMES} menu or enter the characters.
2. Display the list elements.

You need not enter the close brace (\)} when you use \text{STO} \rightarrow to store a list name.

To delete a list name from memory, use the MEM \text{DELETE:LIST} screen (Chapter 17).

The TI-86 distinguishes between uppercase and lowercase letters in list names. For example, ABC123, ABC123, and abc123 are three different list names.
Displaying or Using a Single List Element

To display or use a single list element, the syntax is: \texttt{listName(element#)}

1. Enter the list name; either select it from the LIST NAMES menu or enter the characters.
2. In parentheses, enter the element's place number in the list.
3. Display the list element.

Storing a New Value to a List Element

To store a value to a current element or one element beyond the end of a list, the syntax is: \texttt{value\textgreater listName(element#)}

1. Enter the value to be stored in a current list element or one element beyond the end.
2. Paste \texttt{\textgreater} to the cursor location.
3. Enter the list name; either select it from the LIST NAMES menu or enter the characters.
4. Enter the element's place number in parentheses. (In the example, 5 is one beyond the current dimension of \texttt{ABC123}).
5. Enter the new value to the element number. (\sqrt{18} is evaluated and added as the fifth element.)
Complex List Elements
A complex number can be a list element. If at least one list element is a complex number, all elements in the list are displayed as complex. (√-4 results in a complex number.)

The List Editor  2nd [LIST] F4
The list editor is a table where you can store, edit, and view up to 20 lists that are in memory. Also, you can create list names and attach formulas to lists in the list editor.

The List Editor Menu  2nd [LIST] F4

- Designates the beginning and end of a formula to be attached to a list name
- REAL  Converts the current list to a list of real numbers

To use LIST OPS menu items (or any other functions or instructions) in the list editor, the cursor location must be appropriate for the result. For example, you can use the LIST OPS menu item sortA when a list name is highlighted but not when an element is highlighted.
Creating a List Name in the Unnamed Column

1. Display the list editor.

2. Move the cursor to the unnamed column (column 4). The Name= prompt is displayed in the entry line. ALPHA-lock is on.

3. Enter the list name. The list name is displayed at the top of the current column. In the entry line, a list name prompt is displayed. The name becomes a LIST NAMES menu item and a VARS LIST screen item.

Inserting a List Name into the List Editor

1. Move the cursor to column 3.

2. Insert a new, unnamed column. List names shift right, clearing column 3. The Name= prompt and LIST NAMES menu are displayed.

3. Select ABC12 from the LIST NAMES menu to insert the list name ABC123 into column 3. Elements stored to ABC123 fill the column 3 table of elements. The full value of all ABC123 elements is displayed in the entry line.

After memory is reset, xStat, yStat, and fStat are stored to columns 1, 2, and 3. Resetting defaults does not affect the list editor.

To move from the list name in column 1 to the unnamed column, press 2nd.] [F4].

If all 20 columns have list names, you must remove a list name to make room for the unnamed column.

To cancel the list name insertion, press CLEAR.

If a formula were attached to ABC123, the formula would be displayed in the entry line instead of the list shown in step 3 (page 162.)
Displaying and Editing a List Element

1. Move the cursor onto the fifth element of ABC123. In the entry line, the list name, the element number in parentheses, and the element’s full value are displayed.

2. Switch to edit-element context and edit the element in the entry line.

3. Enter the edited element. Any expression is evaluated and the value is stored to the current element.

Deleting Elements from a List

To delete a single element from a list, move the cursor onto the element and press DEL. The element is deleted. You can clear all elements from a list in any of three ways.

- In the list editor, press [CLEAR ENTER].
- In the list editor, move the cursor onto each element, and then press DEL one by one.
- On the home screen or in the program editor, enter 0→dimL listName to set the dimension of listName to 0 (A to Z Reference).

Removing a List from the List Editor

To remove a list from the list editor, move the cursor onto the list name and then press DEL. The list is not deleted from memory; it is only removed from the list editor.
You can remove all user-created lists from the list editor and restore list names \texttt{xStat}, \texttt{yStat}, and \texttt{fStat} to columns 1, 2, and 3 in either of two ways.

\begin{itemize}
  \item Use \texttt{SetLEdit} with no arguments (page 161).
  \item Reset all memory (Chapter 17). Resetting defaults does not affect the list editor.
\end{itemize}

\section*{Using List Operations}

\textbf{The LIST OPS (Operations) Menu} \[2\text{nd} \text{ LIST} \ F5\]

\begin{tabular}{|c|c|c|c|c|}
\hline
\texttt{dimL} & \texttt{sortA} & \texttt{sortD} & \texttt{min} & \texttt{max} \\
\hline
\end{tabular}

\[\text{OPS}\]

\begin{tabular}{|c|c|c|c|c|c|}
\hline
& \texttt{sum} & \texttt{prod} & \texttt{seq} & \texttt{li\texttt{vc}} & \texttt{vc\texttt{li}} \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|c|}
\hline
\texttt{Fill} & \texttt{aug} & \texttt{cSum} & \texttt{Deltal} & \texttt{Sortx} \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|}
\hline
\texttt{Sorty} & \texttt{Select} & \texttt{SetLE} & \texttt{Form} \\
\hline
\end{tabular}

\textit{For all LIST OPS menu items except Fill( and sometimes dimL, a directly entered list \{$\text{element1,element2,\ldots}$\} is valid for the list argument.}

\texttt{dimL\ list} \hspace{1cm} \texttt{Returns the dimension of (or number of elements in) list}

\texttt{#ofElements \rightarrow dimL\ listName} \hspace{1cm} \texttt{Creates listName as a list that is #ofElements in length; each element is a 0}

\texttt{#ofElements \rightarrow dimL\ listName} \hspace{1cm} \texttt{Redimensions an existing listName; previously entered elements within the new dimension remain; each new list element is a 0; each element in the old list that is outside the new dimension is deleted}

\texttt{sortA\ list} \hspace{1cm} \texttt{Sorts list elements in ascending order, from low to high values}

\texttt{sortD\ list} \hspace{1cm} \texttt{Sorts list elements in descending order, from high to low values}
Chapter 11: Lists

For a complex list, min or max returns the smallest or largest magnitude (modulus).

- **min(list)**: Returns the smallest element of a real or complex list.
- **max(list)**: Returns the largest element of a real or complex list.
- **sum(list)**: Returns the sum of all the elements of a real or complex list, adding from the last element to the first.
- **prod(list)**: Returns the product of all the elements of a real or complex list.
- **seq(expression, variable, begin, end[, step])**: Returns a list in which each element is the result of the evaluation of expression with regard to variable for the values ranging from begin to end in intervals of step (step can be negative).
- **li2v list**
- **li2v [element1, element2, ...]**: Converts a real or complex list to a vector.
- **v2li vector**
- **v2li [element1, element2, ...]**: Converts a real or complex vector to a list.
- **Fill(number, listName)**: Stores a real or complex number to every element of listName.
- **aug(listA, listB)**: (augment) Concatenates the real or complex elements of listA and listB.
- **cSum(list)**: Returns a list of the cumulative sums of real or complex list elements, starting with the first element and proceeding to the last.
- **Delta1st(list)**: Returns a list containing the differences between consecutive elements for all elements in a real or complex list.
- **Sortx [ListName, ListName, frequencyListName]**: In ascending order of x elements, sorts xListName, sorts x and y data pairs, and optionally, their frequencies, in xListName, yListName, and frequencyListName; xStat and yStat are defaults.
- **Sorty [xListName, ListName, frequencyListName]**: In ascending order of y elements, sorts xListName, sorts x and y data pairs, and optionally, their frequencies, in xListName, yListName, and frequencyListName; xStat and yStat are defaults.

Selecting Delta1 from the menu pastes Delta1st( to the cursor location.

For Sortx and Sorty, both lists must have the same number of elements.
Using Mathematical Functions with Lists

You can use a list as a single argument for many TI-86 functions; the result is a list. The function must be valid for every element in the list; however, when graphing, undefined points do not result in an error.

When you use lists for two or more arguments in the same function, all lists must have the same number of elements (equal dimension). Here are some examples of a list as a single argument.

\[
\begin{align*}
\{1,2,3\}+10 & \text{ returns } \{11,12,13\} & \sqrt{\{4,16,36,64\}} & \text{ returns } \{2,4,6,8\} \\
\{5,10,15\} \times (2,4,6) & \text{ returns } \{10,40,90\} & \sin (7,5) & \text{ returns } \{.656986598719, .958924274663\} \\
3\times\{1,7,(2,1)\} & \text{ returns } \{(4,0) (10,0) (5,1)\} & \{1,15,36\} \cdot <19 & \text{ returns } \{1\ 1\ 0\} \\
\end{align*}
\]
Attaching a Formula to a List Name

You can attach a formula to a list name so that the formula generates a list that is stored and dynamically updated in the list name.

- When you edit an element of a list that is referenced in the formula, the corresponding element in the list to which the formula is attached is updated.
- When you edit the formula itself, all elements in the list to which the formula is attached are updated.

To attach a formula to a list name on the home screen or in the program editor, the syntax is:

```
Form("formula",listName)
```

When you enter a new list name as the second argument for `Form`, the list name is created and stored in the LIST NAMES menu and VARS LIST screen upon execution.

1. Store elements to a list name.
2. Select `Form` from the LIST OPS menu; `Form` is pasted to the cursor location.
3. Enter a formula in quotation marks.
4. Enter a comma and then the list name to which you want to attach the formula.
5. Attach the formula to the list name.
Comparing an Attached List with a Regular List

To see the differences between an attached list and a regular list, follow these steps. The example below builds on the example above for attaching a formula to a list. Notice that the formula in step 1 below is not attached to LX because it is not set off by quotation marks.

1. Generate a regular list by storing the expression \( L1+10 \) to the list name LX.

2. Change the second element in \( L1 \) to -8 and display the edited list.

3. Compare the elements of the regular list LX with ADD10, to which the formula \( L1+10 \) is attached. Notice that element 2 of LX is unchanged. Meanwhile, element 2 of ADD10 has been recalculated, since element 2 of \( L1 \) has been edited.

Using the List Editor to Attach a Formula

1. Display the list editor.
2. Highlight the list name to which you want to attach the formula.
3. Enter the formula in quotation marks.
Attach the formula and generate the list.

- The TI-86 calculates each list element.
- A lock symbol is displayed next to the list name to which the formula is attached.

To edit an attached formula, press [ENTER] in step 3, and then edit the formula.

**Using the List Editor With Attached-Formula Lists**

When you edit an element of a list referenced in an attached formula, the TI-86 updates the corresponding element in the list to which the formula is attached.

When you edit or enter elements of a displayed list in any of the three current list editor columns while an attached-formula list also is displayed, the TI-86 takes slightly longer to execute the edit or entry. To reduce this effect, move lists with formulas off the current three-column display, either by scrolling columns to the left or right or by rearranging the list editor.

**Executing and Displaying Attached Formulas**

An attached formula must resolve to a list upon execution. Some examples of formulas that resolve to a list are "5×xStat", "seq(x,x,1,10)", and "[(3,5,-8,4)^2/10". Execution of the formula occurs when you attempt to display the list to which the formula is attached. Also, the formula is executed whenever a list referenced by the formula is modified — whether on the home screen, in the list editor, or in a program.
You can successfully attach to a list a formula that does not yet resolve to a list. For example, you can attach "5*xStat" to the list name BY5 with no elements stored to xStat. However, if you attempt to display BY5 when xStat has no elements, an error occurs.

When you attach such a formula to a list name in the list editor, the formula is successfully attached, but an error occurs. This is because the list editor attempts to execute the formula immediately after attaching it to the list name.

To view the list editor again, you must return to the home screen and either enter something to cause the formula to resolve to a list or remove the attached-formula list from the list editor using the LIST OPS menu item SetLE (page 161).

Handling Errors Related to Attached Formulas

On the home screen, you can attach to a list a formula that references another list that has no elements (dimension is 0; page 161). However, you cannot display the attached-formula list in the list editor or on the home screen until you enter at least one element to the list that the formula references.

**Tip:** If an error menu is returned when you attempt to display an attached-formula list in the list editor, you can select GOTO, write down the formula that is attached to the list name, and then press CLEAR ENTER to detach (clear) the formula. Then you can use the list editor to find the source of the error. After making the appropriate changes, you can re-attach the formula to the list name.

If you do not want to clear the formula, you can select QUIT, display the referenced list on the home screen, and find and edit the source of the error. To edit an element of a list on the home screen, store the new value to listName(element#) (page 155).
Detaching a Formula from a List Name

You can detach a formula in any of five ways.

- Use `dimL` to change the dimension of the list (page 159).
- Use `value\rightarrow listName(element#)` to store `value` to an attached-formula list element.
- Use `"\rightarrow listName`, where `listName` is the attached-formula list.
- In the list editor, move the cursor onto the name of the attached-formula list, and then press `ENTER CLEAR ENTER`. All list elements remain, but the formula is detached and the lock symbol disappears.
- In the list editor, move the cursor onto an element of the attached-formula list. Press `ENTER`, edit the element, and then press `ENTER`. The element changes, the formula is detached, and the lock symbol disappears. All other list elements remain.

Editing an Element of a Attached-Formula List

As described above, one way to detach a formula from a list name is to edit an element of the attached-formula list. The TI-86 protects against inadvertently detaching the formula from the list name when you move the cursor onto one of the elements.

Because of the protection feature, you must press `ENTER` before you can edit an element of an attached-formula list. The protection feature prevents you from deleting an element of an attached-formula list. To delete an element of a attached-formula list, you must first detach the formula in any of the ways described above.
12 Vectors

Vectors on the TI-86 ......................................................... 168
Creating, Storing, and Displaying Vectors .................. 169
Using Mathematical Functions with Vectors .............. 176
Vectors on the TI-86

A vector is a one-dimensional array, arranged in either one row or one column. The vector elements can be real or complex. You can create, display, and edit vectors on the home screen or in the vector editor. When you create a vector, the elements are stored to the vector name.

The TI-86 vector editor displays a vector vertically. On the home screen, a vector is entered and displayed horizontally. When you use a vector in an expression, the TI-86 automatically interprets the vector in the form (row vector or column vector) that is appropriate for the expression. For example, a column vector is appropriate for the expression $matrix\cdot vector$.

On the TI-86, you can store up to 255 elements to a vector in rectangular form. You can use two- or three-element vectors to define magnitude and direction in a two- or three-dimensional space. You can express two- or three-element vectors in different forms, depending on the type of vector.

<table>
<thead>
<tr>
<th>To express a...</th>
<th>You enter:</th>
<th>And the TI-86 returns:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-element rectangular vector</td>
<td>$[x,y]$</td>
<td>$[x\ y]$</td>
</tr>
<tr>
<td>Two-element cylindrical vector</td>
<td>$[r\ θ]$</td>
<td>$[r\ θ]$</td>
</tr>
<tr>
<td>Two-element spherical vector</td>
<td>$[r\ θ]$</td>
<td>$[r\ θ]$</td>
</tr>
<tr>
<td>Three-element rectangular vector</td>
<td>$[x\ y\ z]$</td>
<td>$[x\ y\ z]$</td>
</tr>
<tr>
<td>Three-element cylindrical vector</td>
<td>$[r\ θ\ z]$</td>
<td>$[r\ θ\ z]$</td>
</tr>
<tr>
<td>Three-element spherical vector</td>
<td>$[r\ θ\ φ]$</td>
<td>$[r\ θ\ φ]$</td>
</tr>
</tbody>
</table>
Creating, Storing, and Displaying Vectors

The VECTR (Vector) Menu

The VECTR NAMES Menu

The TI-86 distinguishes between uppercase and lowercase letters in vector names. For example, VECT1, Vect1, and vect1 are three different vector names.

Creating a Vector in the Vector Editor
Enter each vector element value at each vector element prompt. You can enter expressions. To move to the next prompt, press ENTER or #. The vector elements are stored to VECT1, which becomes a VECTR NAMES menu item.

The Vector Editor Menu

<table>
<thead>
<tr>
<th>INSi</th>
<th>DELi</th>
<th>→REAL</th>
</tr>
</thead>
</table>

INSi  Inserts a blank element (enter) prompt at the cursor location; shifts current elements down
DELi  Deletes the element from the cursor location and from the vector; shifts elements up
→REAL Converts the displayed complex number vector to a real number vector

Creating a Vector on the Home Screen

1. Define the beginning of the vector with [.]  5 3 9
2. Enter each vector element, separating each from the next with a comma.
3. Define the end of the vector with [].  5 3 9
4. Store the vector to a vector name from one to eight characters long, starting with a letter. The vector is displayed horizontally and the vector name becomes a VECTR NAMES menu item.
Creating a Complex Vector
If any element of a vector is complex, all elements of the vector are displayed as complex. For example, when you enter the vector \([1, 2, (3, 1)]\), the TI-86 displays \([(1, 0), (2, 0), (3, 1)]\).

To create a complex vector from two real vectors, the syntax is:
`realVector + (0, 1) \times imaginaryVector \rightarrow complexVectorName`

`realVector` contains the real part of each element and `imaginaryVector` contains the imaginary part.

Displaying a Vector
To display a vector, paste the vector name to the home screen, and then press [ENTER]. To display a specific element of `vectorName` on the home screen or in a program, the syntax is:
`vectorName(element#)`

Real two- and three-element vector results are displayed according to the current vector mode setting: `RectV`, `CylV`, or `SphereV` (Chapter 1). You can select a vector conversion instruction from the VECTR OPS menu to override the mode setting (page 173).

Complex vectors are displayed in rectangular form only.
Using a Vector in an Expression

- You can enter the vector directly (for example, 35→[5,10,15]).
- You can use ALPHA and [2nd] [alpha] to enter a vector name’s individual characters.
- You can select the vector name from the VECTR NAMES menu ([2nd] [VECTR] [F1]).
- You can select the vector name from the VARS VECTR screen ([2nd] [CATLG-VARS] MORE [F1]).

Editing Vector Dimension and Elements

1. Display the vector Name= prompt screen.
2. Enter the vector name. Either select it from the VECTR NAMES menu or enter the characters.
3. Display the vector editor.
4. Change or accept the vector dimension.
5. Move the cursor to any element and edit it. Continue moving the cursor to other elements.
6. Save the changes and exit the vector editor.

To use STO to change an element value on the home screen, the syntax is:

\text{value}\to\text{vectorName(element#)}
The VECTR MATH Menu

- NAMES EDIT MATH OPS CPLX
- cross unitV norm dot

**cross(vectorA, vectorB)** Returns the cross product of vectorA and vectorB, both of which are real or complex two-element or three-element vectors; expressed with variables, \(\text{cross}([a,b,c],[d,e,f])\) returns \([bf-ce cd-af ae-bd]\).

**unitV vector** Returns a unit vector where each element of a real or complex vector is divided by the vector norm.

**norm vector** Returns the Frobenius norm (\(\sqrt{\sum(\text{real}^2+\text{imaginary}^2)}\)) where the sum is over all elements of a real or complex vector.

**dot(vectorA, vectorB)** Returns the dot product of vectorA and vectorB, both of which are real or complex vectors; expressed with variables, \(\text{dot}([a,b,c],[d,e,f])\) returns \(ad+be+cf\).

The VECTR OPS (Operations) Menu

- NAMES EDIT MATH OPS CPLX
- dim Fill \(\rightarrow\) Pol \(\rightarrow\) Cyl \(\rightarrow\) Sph

**dim vector** Returns the dimension of (or number of elements in) vector.

**#ofElements\(\rightarrow\)dim vectorName** Creates a new vectorName of the specified length (#ofElements); each element is 0.

**#ofElements\(\rightarrow\)dim vectorName** Redimensions vectorName to the specified length (#ofElements).

**Fill(number, vectorName)** Stores a real or complex number to every element in vectorName.

Press \(\text{STO}\) to enter the \(\rightarrow\) symbol after #ofElements.
For the conversion functions below, the three-element vector conversion equations for cylindrical form \([r \theta z]\) are:

\[
\begin{align*}
x &= r \cos \theta \\
y &= r \sin \theta \\
z &= z
\end{align*}
\]

The three-element vector conversion equations for spherical form \([r \theta \phi]\) are:

\[
\begin{align*}
x &= r \cos \theta \sin \phi \\
y &= r \sin \theta \sin \phi \\
z &= r \cos \phi
\end{align*}
\]

- \texttt{vector\rightarrowPol} Displays a 2-element \texttt{vector} in polar form \([r \theta]\)
- \texttt{vector\rightarrowCyl} Displays a 2- or 3-element \texttt{vector} as a cylindrical vector \([r \theta \theta 0]\) or \([r \theta \theta z]\)
- \texttt{vector\rightarrowSph} Displays a 2- or 3-element \texttt{vector} as a spherical vector \([r \theta \theta 0]\) or \([r \theta \theta \phi]\)
- \texttt{complexVector\rightarrowRec} Displays a 2- or 3-element \texttt{complexVector} in rectangular form \([x y]\) or \([x y z]\)
- \texttt{li\rightarrowvc} Converts a real or complex \texttt{list} into a \texttt{vector}
- \texttt{vc\rightarrowli} Converts a real or complex \texttt{vector} into a \texttt{list}

Complex elements are valid only for \texttt{li\rightarrowvc} and \texttt{vc\rightarrowli}.
The VECTR CPLX (Complex) Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>conj</td>
<td>real</td>
<td>imag</td>
<td>abs</td>
<td>angle</td>
</tr>
</tbody>
</table>

**conj complexVector**
Returns a vector in which each element is the complex conjugate of the corresponding element of a `complexVector`.

**real complexVector**
Returns a real vector in which each element is the real portion of the corresponding element of a `complexVector`.

**imag complexVector**
Returns a real vector in which each element is the imaginary portion of the corresponding element of a `complexVector`.

**abs Vector**
Returns a real vector in which each element is either the absolute value of the corresponding element of a real `vector` or the magnitude (modulus) of the corresponding element of a `complexVector`.

**angle complexVector**
Returns a real vector in which each element is either `0` if the element of `complexVector` is real or the polar angle if the element of `complexVector` is complex; polar angles are calculated as $\tan^{-1}(\text{complex}/\text{real})$ adjusted by $+\pi$ in the second quadrant and by $-\pi$ in the third quadrant.
Using Mathematical Functions with Vectors

To add or subtract two vectors, the dimension of \( \text{vectorA} \) must equal the dimension of \( \text{vectorB} \).

You cannot multiply two vectors or divide one vector by another vector.

\( \text{vectorA} + \text{vectorB} \) Adds each \( \text{vectorA} \) element to the corresponding \( \text{vectorB} \) element; returns a vector of the sums

\( \text{vectorA} - \text{vectorB} \) Subtracts each \( \text{vectorB} \) element from the corresponding \( \text{vectorA} \) element; returns a vector of the differences

\( \text{vector} \times \text{number} \) or \( \text{number} \times \text{vector} \) Returns a vector that is the product of a real or complex \( \text{number} \) times each element in a real or complex \( \text{vector} \)

\( \text{matrix} \times \text{vector} \) Returns a vector that is the product of each \( \text{vector} \) element times each \( \text{matrix} \) element; \( \text{matrix} \) column dimension and \( \text{vector} \) dimension must be equal

\( \text{vector} / \text{number} \) Returns a vector that is the quotient of each real or complex \( \text{vector} \) element divided by a real or complex \( \text{number} \)

\( \text{~vector} \) (negation) Changes the sign of each \( \text{vector} \) element

\( \text{vectorA} \approx \text{vectorB} \) Returns 1 if every corresponding element comparison is true; returns 0 if any is false

\( \text{vectorA} 
eq \text{vectorB} \) Returns 1 if at least one corresponding element comparison is false

\( \text{round}(\text{vector}, \#\text{ofDecimals}) \) Rounds each \( \text{vector} \) element to 12 digits, or rounds to specified \#ofDecimals

\( \text{iPart} \text{vector} \) Returns the integer part of each real or complex \( \text{vector} \) element

\( \text{fPart} \text{vector} \) Returns the fractional part of each real or complex \( \text{vector} \) element

\( \text{int} \text{vector} \) Returns the greatest integer of each real or complex \( \text{vector} \) element

\( \text{==} \) and \( \neq \) are on the TEST menu.

\( \text{round}, \text{iPart}, \text{fPart}, \text{and int} \) are on the MATH NUM menu.
13 Matrices

Matrices on the TI-86 ...................................................... 178
Creating, Storing, and Displaying Matrices ....................... 178
Using Mathematical Functions with Matrices ................. 185
Matrices on the TI-86

A matrix is a two-dimensional array, arranged in rows and columns. The matrix elements can be real or complex. You can create, display, and edit matrices on the home screen or in the matrix editor. When you create a matrix, the elements are stored to the matrix name.

Creating, Storing, and Displaying Matrices

The MATRX (Matrix) Menu

The MATRX NAMES Menu

The TI-86 distinguishes between uppercase and lowercase letters in matrix names. For example, MAT1 and mat1 are two different vector names.

Creating a Matrix in the Matrix Editor

1. Display the matrix Name= prompt screen.
2. ALPHA-lock is on. The MATRX NAMES menu is displayed. Enter a name from one to eight characters long, starting with a letter.
3. Display the matrix editor and the matrix editor menu.

4. Accept or change the matrix dimensions (row × column) in the top-right corner of the screen, (1 ≤ row ≤ 255 and 1 ≤ column ≤ 255); maximum combination is subject to memory availability. The matrix is displayed; all elements are 0.

5. Enter each matrix element value at the element prompt (1,1 = for row 1, column 1). You can enter expressions. To move to the next element, press ENTER. To move to the next row, press #.

The Matrix Editor Menu

<table>
<thead>
<tr>
<th>INSr</th>
<th>DELr</th>
<th>INSc</th>
<th>DELc</th>
<th>➪REAL</th>
</tr>
</thead>
</table>

INSr  Inserts a row at the cursor location; shifts subsequent rows down
DELr  Deletes row at the cursor location; shifts subsequent rows up
INSc  Inserts a column at the cursor location; shifts subsequent columns to the right
DELc  Deletes the column at the cursor location; shifts subsequent columns to the left
➤REAL Converts the displayed complex number matrix to a real number matrix

An ellipsis (…) at either end of matrix rows indicates additional columns.

↓ or ↑ in the last column indicates additional rows.
Creating a Matrix on the Home Screen

1. Define the start of the matrix with [ and then define the start of the first row with another [. Enter each element for the row, separating them with commas. Define the end of the first row with ].

2. Define the start of each subsequent row with [ . Enter the row elements, separating each from the next with a comma. Define the end of each row with ]. Then define the end of the matrix with ].

3. Store the matrix to a matrix name. Either enter a name from one to eight characters long, starting with a letter, or select a name from the MATRIX NAMES menu. The matrix is displayed. If newly created, the matrix name becomes a MATRIX NAMES menu item.

Creating a Complex Matrix

If any matrix element is complex, all elements of the matrix are displayed as complex. For example, when you enter the matrix $\begin{bmatrix} 1 & 2 \\ 5 & (3,1) \end{bmatrix}$, the TI-86 displays $\begin{bmatrix} (1,0) & (2,0) \\ (5,0) & (3,1) \end{bmatrix}$.

To create a complex matrix from two real matrices with the same dimensions, the syntax is: `realMatrix+(0,1)imaginaryMatrix>complexMatrixName`

`realMatrix` contains the real part of each element and `imaginaryMatrix` contains the imaginary part of each element.
Displaying Matrix Elements, Rows, and Submatrices

To display an existing matrix on the home screen, enter the matrix name's individual characters or select it from the MATRIX NAMES menu, and then press [ENTER]. The full value of each element is displayed. Elements with very large values may be expressed exponentially.

To display specific elements of $matrixName$, the syntax is: $matrixName(row, column)$

To display a row of $matrixName$, the syntax is: $matrixName(row)$

To display a submatrix of $matrixName$, the syntax is: $matrixName(beginRow, beginColumn, endRow, endColumn)$

Using a Matrix in an Expression

- You can enter the matrix directly (for example, $5*[[2,3][3,5]]$).
- You can use [ALPHA] and [2nd] [alpha] to enter a matrix name’s individual characters (for example, MAT1*3).
- You can select the matrix name from the MATRIX NAMES menu ([2nd] [MATRIX] [F1]).
- You can select the matrix name from the VARS MATRIX screen ([2nd] [CATLG-VARS] [MORE] [F2]).
Editing Matrices in the Matrix Editor

1. Display the matrix Name= prompt screen.
2. Enter the matrix name. Either select it from the MATRX NAMES menu or enter the characters.
3. Display the matrix editor.
4. Edit or accept the row dimension, and then edit or accept the column dimension.
5. Move the cursor to any element and edit it. Continue moving the cursor to other elements.
6. Save the changes and leave the matrix editor.

Editing Matrices on the Home Screen

To change a matrix element value, the syntax is:
value\rightarrow matrixName(row,column)

To change the values of an entire row of elements, the syntax is:
[valueA,valueB,...,value n]\rightarrow matrixName(row)

To change the values of part of a row, beginning at a specified column, the syntax is:
[valueA,valueB,...,value n]\rightarrow matrixName(row,bEGINColumn)

To change the values of a submatrix within matrixName, the syntax is:
[[valueA,...,value n]...[valueA,...,value n]\rightarrow matrixName(beginRow,bEGINColumn)
The MATRIX MATH Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>det</td>
<td>T</td>
<td>norm</td>
<td>eigVl</td>
<td>eigVc</td>
</tr>
</tbody>
</table>

- **det squareMatrix**: Returns the determinant of *squareMatrix*
- **matrixᵀ**: Returns a transposed matrix; each element’s (row,column) coordinates switch
- **norm matrix**: Returns the Frobenius norm (√(real²+imaginary²)) where the sum is over all elements of a real or complex *matrix*
- **eigVl squareMatrix**: Returns a list of the normalized eigenvalues of a real or complex *squareMatrix*
- **eigVc squareMatrix**: Returns a matrix containing the eigenvectors for a real or complex *squareMatrix*; each column corresponds to an eigenvalue
- **rnorm matrix**: (row norm) Returns the largest of the sums of the absolute values of the elements (magnitudes of complex elements) in each row of *matrix*
- **cnorm Matrix**: (column norm) Returns the largest of the sums of the absolute values of the elements (magnitudes of complex elements) in each column of *matrix*
- **LU(matrix, lMatrixName, uMatrixName, pMatrixName)**: Calculates the Crout LU (lower-upper) decomposition of a real or complex *matrix*; stores the lower triangular matrix to *lMatrixName*, the upper triangular matrix to *uMatrixName*, and the permutation matrix (which describes the row swaps done during calculation) in *pMatrixName*
- **cond squareMatrix**: Calculates \( \text{cnorm} \cdot \text{cnorm} \cdot \text{matrix}^{-1} \); the closer the product is to 1, the more stable *squareMatrix* can be expected to be in matrix functions
### The MATRX OPS (Operations) Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim</td>
<td>Fill</td>
<td>ident</td>
<td>ref</td>
<td>rref</td>
</tr>
</tbody>
</table>

- **dim** *matrix*
  - Returns the dimensions of *matrix* as a list (rows columns)

- **{rows,columns}dim** *matrixName*
  - Creates a new *matrixName* of the specified dimensions; each element is 0

- **{rows,columns}dim** *matrixName*
  - Redimensions *matrixName* to the specified dimensions

- **Fill(number,*matrixName*)**
  - Stores a real or complex *number* to each *matrixName* element

- **ident** *dimension*
  - Returns the square identity matrix of dimension × dimension

- **ref** *matrix*
  - Returns the row-echelon form of *matrix*

- **rref** *matrix*
  - Returns the reduced row-echelon form of *matrix*

- **aug(*matrixA*,*matrixB*)**
  - Concatenates *matrixA* and *matrixB*

- **aug(*matrix*,vector)**
  - Concatenates *matrix* and vector

- **rSwap(*matrix*,rowA,rowB)**
  - Returns a matrix after swapping rowA and rowB of *matrix*

- **rAdd(*matrix*,rowA,rowB)**
  - Returns *matrix* with (rowA+rowB) of *matrix* stored in rowB

- **multR(number,*matrix*,row)**
  - Returns *matrix* with (row*number) stored in row

- **mRAdd(number,*matrix*,rowA,rowB)**
  - Returns *matrix* with ((rowA*number)+rowB) stored in rowB

- **randM(rows,columns)**
  - Creates a matrix of specified dimensions with random elements

---

Press **STO→** to enter the ➞ symbol after the close brace.

When you use **aug**, the number of rows in *matrixA* must equal the number of rows in *matrixB* or the number of elements in vector.

Elements of matrices created with **randM** are integers ≥9 and ≤9.
The MATRX CPLX (Complex) Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>conj</td>
<td>real</td>
<td>imag</td>
<td>abs</td>
<td>angle</td>
</tr>
</tbody>
</table>

**conj complexMatrix**
Returns a matrix in which each element is the complex conjugate of the corresponding element of a complexMatrix.

**real complexMatrix**
Returns a real matrix in which each element is the real portion of the corresponding element of a complexMatrix.

**imag complexMatrix**
Returns a real matrix in which each element is the imaginary portion of the corresponding element of a complexMatrix.

**abs matrix**
Returns a real matrix in which each element is either the absolute value of the corresponding element of a real matrix or the magnitude (modulus) of the corresponding element of a complex matrix.

**angle complexMatrix**
Returns a real matrix in which each element is either 0 if the element of complexMatrix is real or the polar angle if the element of complexMatrix is complex; the polar angles are calculated as \(\tan^{-1}(\text{imaginary/real})\) adjusted by \(+\pi\) in the second quadrant and by \(-\pi\) in the third quadrant.

---

**Using Mathematical Functions with Matrices**

To add or subtract two matrices, the dimensions of matrixA must equal the dimensions of matrixB.

- **matrixA+matrixB**
  Adds each matrixA element to the corresponding matrixB element; returns a matrix of the sums.

- **matrixA−matrixB**
  Subtracts each matrixB element from the corresponding matrixA element; returns a matrix of the differences.
Chapter 13: Matrices

To multiply two matrices, the column dimension of matrixA must equal the row dimension of matrixB.

matrixA·matrixB or matrixB·matrixA

matrix·number or number·matrix

matrix·vector

Multiplies matrixA and matrixB; returns a square matrix of the products

Returns a matrix that is the product of a real or complex number times each element in a real or complex matrix

Returns a vector that is the product of each vector element times each matrix element; the matrix column dimension and vector dimension must be equal

(matrix) Changes the sign of each element in matrix

Returns a matrix that is the product of a real or complex number times each element in a real or complex matrix

Squares a square matrix

(e^ squareMatrix)

Returns the square matrix exponential of a real squareMatrix

(sin squareMatrix)

Returns the square matrix sine of a real squareMatrix

(cos squareMatrix)

Returns the square matrix cosine of a real squareMatrix

(matrixA==matrixB)

Returns 1 if every corresponding element comparison is true; returns 0 if any is false

(matrixA≠matrixB)

Returns 1 if at least one corresponding element comparison is false

round(matrix, #ofDecimals)

 Rounds each matrix element to 12 digits or to specified #of Decimals

iPart matrix

Returns the integer part of each element of a real or complex matrix

fPart matrix

Returns the fractional part of each element of a real or complex matrix

int matrix

Returns the greatest integer of each element of a real or complex matrix

To enter -1, press (-) [-]. Do not use [VAR] 1.

e^, sin, and cos do not return the exponential, sine, or cosine of each matrix element.

To make relational comparisons, matrixA and matrixB must have equal dimensions.

== and ≠ are on the TEST menu.

round, iPart, fPart, and int are on the MATH NUM menu.
14 Statistics

Statistical Analysis on the TI-86 .................................................. 188
Setting Up a Statistical Analysis.............................................. 188
Results of a Statistical Analysis........................................... 192
Plotting Statistical Data ....................................................... 194
The STAT DRAW Menu ....................................................... 199
Forecasting a Statistical Data Value ................................. 199
Statistical Analysis on the TI-86

With the TI-86, you can analyze one-variable and two-variable statistical data, which are stored in lists. One-variable data has one measured variable. Two-variable data has pairs comprising an independent variable and a dependent variable.

When analyzing either kind of data, you can specify a frequency of occurrence for the independent variable values. These specified frequencies must be real numbers ≥ 0.

Setting Up a Statistical Analysis

1. Enter the statistical data into one or more lists (Chapter 11).
2. Calculate the statistical variables or fit a model to the data.
3. Plot the data.
4. Graph the regression equation for the plotted data.

The STAT (Statistics) Menu

The same list editor is displayed, whether you press 2nd [STAT] or 2nd [LIST]. For a description of the list editor, see Chapter 11.
Entering Statistical Data

Data for statistical analysis is stored in lists, which you can create and edit in the list editor (Chapter 11), on the home screen (Chapter 11), or in a program (Chapter 16). The TI-86 has three built-in list names for statistics, xStat (x-variable list), yStat (y-variable list), and fStat (frequency list). TI-86 statistical functions use these lists as defaults.

The LIST NAMES Menu

The list names shown here have no user-created list names. Editing an element of xStat or yStat clears any values stored to statistical result variables.

<table>
<thead>
<tr>
<th>LIST NAMES Commands</th>
<th>2nd [STAT] [2] [3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>fStat</td>
<td>xStat</td>
</tr>
<tr>
<td>yStat</td>
<td></td>
</tr>
</tbody>
</table>

fStat: An automatically updated list of the frequency values used in the last statistical computation requiring a frequency; default is a list where each element is 1

xStat: An automatically updated list of the data from the x-list used in the last statistical analysis

yStat: An automatically updated list of the data from the y-list used in the last statistical analysis

The STAT CALC (Calculations) Menu

The STAT CALC functions store the results to statistical result variables (page 193).

The syntax description for each STAT CALC menu item follows this section.

<table>
<thead>
<tr>
<th>STAT CALC Commands</th>
<th>2nd [STAT] F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC</td>
<td></td>
</tr>
<tr>
<td>OneVa</td>
<td>TwoVa</td>
</tr>
<tr>
<td>LinR</td>
<td>LnR</td>
</tr>
<tr>
<td>ExpR</td>
<td>PwrR</td>
</tr>
<tr>
<td>SinR</td>
<td>LgstR</td>
</tr>
<tr>
<td>P2Reg</td>
<td>P3Reg</td>
</tr>
<tr>
<td>P4Reg</td>
<td>StReg</td>
</tr>
</tbody>
</table>

OneVa: (one variable) Analyzes data with one measured variable

TwoVa: (two variable) Analyzes paired data
For regression analysis, the statistical results are calculated using a least-squares fit.

LinR (linear regression) Fits the model equation \( y = a + bx \) to the data; displays values for \( a \) (slope) and \( b \) (y-intercept)

LnR (logarithmic regression) Fits the model equation \( y = a + b \ln x \) to the data using transformed values \( \ln(x) \) and \( y \); displays values for \( a \) and \( b \)

ExpR (exponential regression) Fits the model equation \( y = ab^x \) to the data using transformed values \( x \) and \( \ln(y) \); displays values for \( a \) and \( b \); elements in the x-list and y-list elements must be integers

PwrR (power regression) Fits the model equation \( y = ax^b \) to the data using transformed values \( \ln(x) \) and \( \ln(y) \); displays values for \( a \) and \( b \)

SinR (sinusoidal regression) Fits the model equation \( y = a \sin(bx + c) + d \) to the data; displays values for \( a \), \( b \), \( c \), and \( d \); SinR requires at least four data points; it also requires at least two data points per cycle to avoid aliased frequency estimates

LgstR (logistic regression) Fits the model equation \( y = a/(1 + be^{cx}) + d \) to the data; displays \( a \), \( b \), \( c \), and \( d \)

P2Reg (quadratic regression) Fits the second-degree polynomial \( y = ax^2 + bx + c \) to the data; displays values for \( a \), \( b \), and \( c \); for three data points, the equation is a polynomial fit; for four or more, it is a polynomial regression; P2Reg requires at least three data points

P3Reg (cubic regression) Fits the third-degree polynomial \( y = ax^3 + bx^2 + cx + d \) to the data; displays values for \( a \), \( b \), \( c \), and \( d \); for four points, the equation is a polynomial fit; for five or more, it is a polynomial regression; P3Reg requires at least four data points

P4Reg (quartic regression) Fits the fourth-degree polynomial \( y = ax^4 + bx^3 + cx^2 + dx + e \) to the data; displays values for \( a \), \( b \), \( c \), \( d \), and \( e \); for five points, the equation is a polynomial fit; for six or more, it is a polynomial regression; P4Reg requires at least five data points

StReg (store regression equation) Pastes StReg( to the home screen; enter a variable and press ENTER; the current regression equation is stored to variable
When you select OneVa or TwoVa, the abbreviation OneVar or TwoVar is displayed.

For OneVa, the syntax is:
OneVar [xList,frequencyList]

For TwoVa, the syntax is:
TwoVar [xList,yList,frequencyList]

For LinR, LnR, ExpR, PwrR, P2Reg, P3Reg, and P4Reg, the syntax is:
TwoVar [xList,yList,frequencyList]

For SinR, the syntax is:
SinR [iterations,xList,yList,period,equationVariable]

itearations is the number of iterations to go through; higher values for iterations produce a better fit, but take longer to calculate. period is an initial guess at which to begin calculation.

For LgstR, the syntax is:
LgstR [iterations,xList,yList,frequencyList,equationVariable]

To copy the contents RegEq to any variable after calculating the regression, the syntax is:
StReg(variable)

**Automatic Regression Equation Storage**

LinR, LnR, ExpR, PwrR, SinR, LgstR, P2Reg, P3Reg, and P4Reg are regression models. Each regression model has an optional argument, equationVariable, for which you can specify an equation variable, such as y1. Upon execution, the regression equation is stored automatically to the specified equation variable, and the function is selected.

Regardless of whether you specify equationVariable, the regression equation always is stored to the result variable RegEq, which is an item on the STAT VARS menu. The regression equation displays the actual result values.
The result for a polynomial regression, sinusoidal regression, or logistic regression is stored in \texttt{PRegC} (polynomial/regression coefficients). \texttt{PRegC} is a list containing the coefficients for an equation. For example, for \texttt{P3Reg}, the result \texttt{PRegC={3 5 -2 7)} would represent \(y=3x^3+5x^2-2x+7\).

\section*{Results of a Statistical Analysis}

When you perform a statistical analysis, the calculated results are stored in the result variables and the data from the lists used in the analysis are stored to \texttt{xStat}, \texttt{yStat}, and \texttt{fStat}. If you edit a list or change the type of analysis, all statistical variables are cleared.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{CALC} & \textbf{EDIT} & \textbf{PLOT} & \textbf{DRAW} & \textbf{VARS} \\
\hline
\hline
\hline
\end{tabular}
\end{table}

\texttt{PRegC} is the only statistical result variable calculated for a polynomial regression.

One- and two-variable statistical functions share the result variables.

The statistical variables are calculated and stored as shown in the table on the next page.

You can use ALPHA keys, alpha keys, and the CHAR GREEK menu to enter some result variables.

To paste a result variable to the cursor location, either select the variable from the \texttt{STAT VARS} menu or select the variable from the \texttt{VARS STAT} selection screen.

\begin{itemize}
\item To use a result variable in an expression, paste it to the appropriate cursor location.
\item To display the value of a result variable, paste it to the home screen and press \texttt{ENTER}.
\item To store results to another variable after a calculation, paste the result variable to the home screen, press \texttt{STO\hbox{\char95}}, enter a new variable, and then press \texttt{ENTER}.
\end{itemize}
### Result Variables

<table>
<thead>
<tr>
<th></th>
<th>1-Var Stats</th>
<th>2-Var Stats</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean of x values</td>
<td>$\bar{x}$</td>
<td>$\bar{x}$</td>
<td></td>
</tr>
<tr>
<td>pop std dev of x</td>
<td>$\sigma_x$</td>
<td>$\sigma_x$</td>
<td></td>
</tr>
<tr>
<td>sample std dev of x</td>
<td>$S_x$</td>
<td>$S_x$</td>
<td></td>
</tr>
<tr>
<td>mean of y values</td>
<td>$\bar{y}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pop std dev of y</td>
<td>$\sigma_y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sample std dev of y</td>
<td>$S_y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum of x values</td>
<td>$\sum x$</td>
<td>$\sum x$</td>
<td></td>
</tr>
<tr>
<td>sum of $x^2$ values</td>
<td>$\sum x^2$</td>
<td>$\sum x^2$</td>
<td></td>
</tr>
<tr>
<td>sum of y values</td>
<td>$\sum y$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum of $y^2$ values</td>
<td>$\sum y^2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum of $x \times y$</td>
<td>$\sum xy$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regression equation</td>
<td>RegEq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>polynomial, $LgstR$,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and $\text{SinR}$ coeff's</td>
<td>a (y-int)</td>
<td>b (slope)</td>
<td></td>
</tr>
</tbody>
</table>

### Result Variables

<table>
<thead>
<tr>
<th></th>
<th>1-Var Stats</th>
<th>2-Var Stats</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>correlation coeff</td>
<td>$\text{corr}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y-intercept of reg eq</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slope of reg eq</td>
<td>b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regression/fit coeff</td>
<td>a, b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of data pts</td>
<td>$n$</td>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>min of x values</td>
<td>$\text{minX}$</td>
<td>$\text{minX}$</td>
<td></td>
</tr>
<tr>
<td>max of x values</td>
<td>$\text{maxX}$</td>
<td>$\text{maxX}$</td>
<td></td>
</tr>
<tr>
<td>min of y values</td>
<td>$\text{minY}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max of y values</td>
<td>$\text{maxY}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>median</td>
<td>$\text{Med}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st quartile</td>
<td>$\text{Qrtl1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd quartile</td>
<td>$\text{Qrtl3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>polynomial $LgstR$,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and $\text{SinR}$ reg coeff's</td>
<td>PRegC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first quartile ($\text{Qrtl1}$) is the median of the points between $\text{minX}$ and $\text{Med}$ (median). The third quartile ($\text{Qrtl3}$) is the median of the points between $\text{Med}$ and $\text{maxX}$.

When you calculate a logistic regression, 1 is stored to $\text{tolMet}$ ($\text{tolMe}$) if the TI-86 internal tolerance was met before the calculator arrived at a result; if not met, 0 is stored to $\text{tolMet}$.  

---

**These words are abbreviated in the table:**
- pop = population
- std dev = standard deviation
- coeff = coefficient
- int = intercept
- reg eq = regression equation
- pts = points
- min = minimum
- max = maximum
Plotting Statistical Data

You can plot one, two, or three sets of statistical list data. The five available plot types are scatter plot, xyLine, histogram, modified box plot, and regular box plot.

1. Store the statistical data in one or more lists (Chapter 11).
2. Select or deselect functions in the current equation editor as appropriate (Chapter 5).
3. Define the statistical plot.
4. Turn on the plots you want to display.
5. Define the window variables for the graph screen (Chapter 5).
6. Display and explore the plotted graph (Chapter 6).

The STAT PLOT Status Screen

The STAT PLOT status screen summarizes the settings for Plot1, Plot2, and Plot3. The illustration below identifies the settings for Plot1. This screen is not interactive. To change a setting, select PLOT1, PLOT2, or PLOT3 from the STAT PLOT status screen menu.

- Stat plot name
- On/Off status
- Plot-type icon
- xStat
- yStat
- Mark-type icon
- Independent list name
- Dependent list name
Chapter 14: Statistics

The STAT PLOT Menu

When you display a stat plot editor, the STAT PLOT menu remains so that you can easily switch to another stat plot.

In this guidebook, brackets ([ and ]) with syntax specify arguments as optional. Do not enter brackets, except with vectors and matrices.

The STAT PLOT Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>[STAT]</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT1</td>
<td>PLOT2</td>
<td>PLOT3</td>
</tr>
</tbody>
</table>

PLOT1
Displays the stat plot editor for Plot1

PLOT2
Displays the stat plot editor for Plot2

PLOT3
Displays the stat plot editor for Plot3

PlOn [1,2,3]
Turns on all plots (if you enter no arguments) or turns on specified plots only

PlOff [1,2,3]
Turns off all plots (if you enter no arguments) or turns off specified plots only

To turn on or turn off all three stat plots, select PlOn or PlOff from the STAT PLOT menu. PlOn or PlOff is pasted to the home screen. Press ENTER. All stat plots are now on or off.

Setting Up a Stat Plot

To set up a stat plot, select PLOT1, PLOT2, or PLOT3 from the STAT PLOT menu. The stat plot editor for the selected stat plot is displayed. Each stat plot type has a unique stat plot editor. The screen to the right shows the stat plot editor for the default (scatter plot). If you select another plot type, some prompts may change.

Turning On and Turning Off a Stat Plot

When you display a stat plot editor, the cursor is on the On option.

♦ To turn on the stat plot, press ENTER.
♦ To turn off the stat plot, press \ \ ENTER.
Chapter 14: Statistics

The PLOT TYPE Menu (Selecting a Plot Type)

To display the PLOT TYPE menu, move the cursor onto the plot type icon at the Type= prompt.

When you select a plot type, the appearance of the stat plot editor may change.

Stat plots are displayed on the graph screen (GRAPH Ps), as defined by the window variable values (Chapter 5). Some graph tools apply to stat plots.

In these stat plot examples, all functions are deselected. Also, menus are cleared from the screen with CLEAR.

Plot Type Characteristics

SCAT (scatter plot) plots the data points from Xlist Name and Ylist Name as coordinate pairs, representing each point with a box ( ), cross ( + ), or dot ( ) mark type. Xlist Name and Ylist Name must be the same length. Xlist Name and Ylist Name can be the same list.

For the example:

\[
\begin{align*}
\text{xMin} &= 0 \\
\text{yMin} &= -10 \\
\text{xMax} &= 10 \\
\text{yMax} &= 10 \\
\end{align*}
\]

\[
\begin{align*}
xStat &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\
yStat &= 5 \sin(xStat)
\end{align*}
\]

Window variable values:

\[
\begin{align*}
\text{xMin} &= 0 \\
\text{yMin} &= -10 \\
\text{xMax} &= 10 \\
\text{yMax} &= 10 \\
\end{align*}
\]
xyLINE is a scatter plot in which the data points are plotted and connected in order of appearance in Xlist Name and Ylist Name. You may want to use SortA or SortD from the LIST OPS menu (Chapter 11) to sort the lists before you plot them.

For the example:
\[ xStat = \{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10\} \]
\[ yStat = 5 \sin(xStat) \]

Window variable values:
\[ xMin = 0 \quad yMin = 10 \]
\[ xMax = 10 \quad yMax = 10 \]

MBOX (modified box plot) plots one-variable data, like the regular box plot, except that the points are 1.5 * Interquartile Range beyond the quartiles. (The Interquartile Range is defined as the difference between the third quartile \( Q_3 \) and the first quartile \( Q_1 \).) These points are plotted individually beyond the whisker, using the Mark ( \( \circ \) or \( + \) or \( \bullet \) ) you select.

For the example:
\[ xStat = \{1 \ 2 \ 2.5 \ 3 \ 3.3 \ 4 \ 4 \ 2 \ 6 \ 9\} \]

Window variable values are set by selecting ZDATA from the GRAPH ZOOM menu.

You can trace these points, which are called outliers. When outliers exist, the end of each whisker will display an \( x= \) prompt. When no outliers exist, \( xMin \) and \( xMax \) are the prompts for the end of each whisker. \( Q_1 \), Med (median), and \( Q_3 \) define the box.

Modified box plots are plotted with respect to \( xMin \) and \( xMax \), but ignore \( yMin \) and \( yMax \).

When two modified box plots are plotted, the first one plots at the top of the screen and the
second plots in the middle. When three are plotted, the first one plots at the top, the second in the middle, and the third at the bottom.

HIST (histogram) plots one-variable data. The \( xScl \) window variable value determines the width of each bar, beginning at \( xMin \). ZDATA (GRAPH ZOOM menu) adjusts \( xMin \), \( xMax \), \( yMin \), and \( yMax \) to include all values, and also adjusts \( xScl \). \( \frac{xMax - xMin}{xScl} \leq 47 \) must be true. A value that occurs on the edge of a bar is counted in the bar to the right.

For the example:
\[
xStat = \{1 \ 2 \ 2 \ 3 \ 8 \ 9 \ 5 \ 6 \ 6 \ 7 \ 7 \ 4 \ 4 \ 9 \ 9 \ 9\}
\]

Window variable values:
\[
xMin = 0 \quad yMin = 0 \quad xMax = 10 \quad yMax = 5
\]

Whiskers are the lines protruding from the sides of the box.

BOX (regular box plot) plots one-variable data. The whiskers on the plot extend from the minimum data point in the set \( (xMin) \) to the first quartile (\( Q_1 \)) and from the third quartile (\( Q_3 \)) to the maximum point (\( xMax \)). The box is defined by \( Q_1 \), Med (median), and \( Q_3 \).

For the example:
\[
xStat = \{1 \ 2 \ 2.5 \ 3 \ 3.3 \ 4 \ 4 \ 2 \ 6 \ 9\}
\]

Window variable values are set by selecting ZDATA from the GRAPH ZOOM menu.

Box plots are plotted with respect to \( xMin \) and \( xMax \), but ignore \( yMin \) and \( yMax \). When two box plots are plotted, the first one plots at the top of the screen and the second plots in the
middle. When three are plotted, the first one plots at the top, the second in the middle, and the third at the bottom.

**The STAT DRAW Menu**

<table>
<thead>
<tr>
<th>CALC</th>
<th>EDIT</th>
<th>PLOT</th>
<th>DRAW</th>
<th>VARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIST</td>
<td>SCAT</td>
<td>xyLINE</td>
<td>BOX</td>
<td>MBOX</td>
</tr>
</tbody>
</table>

When you select any of the first five STAT DRAW menu items, the TI-86 plots the data stored in the lists xStat and yStat.

Forecasting a Statistical Data Value

Using the forecast editor, you can forecast an x-value or y-value based on the current regression equation. To use the forecast editor, a regression equation must be stored to RegEq.
Enter stat data in the list editor. The screen to the right shows all fStat elements as 1, but you need not enter them. 1 is the default for all fStat elements. However, if other elements are stored to fStat, you must clear them.

Display the home screen.

Execute a linear regression for xStat and yStat. The statistical results are displayed.

Remove the STAT CALC menu to display all results, including n.

Display the forecast editor. The current regression model is displayed on the top line.

Enter x=3, and then move the cursor to the y= prompt.

Select SOLVE from the forecast editor menu to solve for y at x=3. A small square indicates the solution. You can continue to use the forecast editor with other values for x or y.

Values entered at forecast editor prompts must be real numbers or expressions that evaluate to real numbers.

When you use FCST, the values of x, y, and Ans are not updated. To store the x value or y value, move the cursor onto the variable to be stored, press Stored, enter a valid variable name at the Sto prompt, and then press ENTER.

If the most recent calculation was a polynomial regression, you can only forecast the y value.
15
Equation Solving

Preview: The Equation Solver ................................. 202
Entering an Equation in the Equation-Entry Editor .... 203
Setting Up the Interactive-Solver Editor ................... 204
Solving for the Unknown Variable ............................. 206
Graphing the Solution ............................................ 207
Solver Graph Tools .................................................. 207
The Simultaneous Equation Solver ............................. 208
The Polynomial Root-Finder ..................................... 211
Preview: The Equation Solver  

With the equation solver, you can enter an expression or equation, store values to all but one variable in the expression or equation, and then solve for the unknown variable. These steps introduce the solver. For details, read this chapter.

1. Display the equation-entry editor. The VARS EQU menu is displayed on the bottom of the screen.

2. Enter an equation. When you press [enter], the interactive-solver editor and solver menu are displayed.

3. Enter values for each variable, except the unknown variable R1. Some variables may have values stored to them already.

4. Move the cursor to the variable for which you want to solve. You may enter a guess.

5. Solve the equation for the variable. Small squares mark both the solution variable and the equation left=rt=0 (the left side of the equation minus the right side of the equation). If you edit a value or leave the screen, the squares disappear.

The VARS EQU menu is a menu version of the VARS EQU screen (Chapter 2).

The example uses a formula for a voltage divider.

R1 and R2 represent resistors.

V and V1 represent voltage.

To solve for the unknown variable in an equation on the home screen or in the program editor, select Solver from the CATALOG (A to Z Reference).
The equation can have more than one variable to the left of the equal sign, as in $A+B=C+\sin D$.

You can display other menus in the equation-entry editor.

An ellipsis (…) indicates that an entered equation continues beyond the screen. To move directly to the start of the equation, press [2nd] [1]. To move directly to the end, press [2nd] [4].

Chapter 15: Equation Solving

15EQSOLV.DOC   TI-86, Chap 15, US English   Bob Fedorisko   Revised: 02/13/01 2:34 PM   Printed: 02/13/01 3:04 PM   Page 203 of 12

Entering an Equation in the Equation-Entry Editor

The equation solver uses two editors: the equation-entry editor, where you enter and edit the equation you want to solve, and the interactive-solver editor, where you enter known variable values, select the variable for which you want to solve, and display the solution.

To display the equation-entry editor, press [2nd] [SOLVER]. In this editor, you can:

- Enter an equation directly.
- Enter a defined equation variable's individual characters or select it from the VARS EQU menu.
- Recall the contents of a defined equation variable.

As you enter or edit the equation, the TI-86 automatically stores it to the variable `eqn`.

The VARS EQU menu is a menu version of the VARS EQU screen (Chapter 2). The items are all variables to which an equation is stored. This includes all selected and deselected equation variables defined in the equation editors of all four graphing modes (Chapters 5, 8, 9, and 10). The menu items are in alphanumeric order.

- If you select an equation variable from the menu, the variable is pasted to the cursor location, overwriting characters for the length of the variable name.
- If you press [2nd] [RCL], select an equation variable from the menu, and then press [ENTER], the variable contents are inserted at the cursor location.

If you enter an equation variable, the TI-86 automatically converts it to the equation `exp=equationVariable`. If you enter an expression directly, the TI-86 automatically converts the expression to the equation `exp=expression`. 
Setting Up the Interactive-Solver Editor

After you have stored an equation to `eqn` in the equation-entry editor, press `ENTER` to display the interactive-solver editor.

The equation is displayed across the top of the editor. Each variable in the equation is displayed as a prompt. Values already stored to variables are displayed; undefined variables are blank. The solver menu is displayed on the bottom of the editor (page 206).

`bound={-1E99,1E99}` is a list containing the default lower bound (`-1E99`) and the default upper bound (`1E99`). You can edit the bounds (below).

Entering Variable Values

To solve for an unknown variable, you must define every other variable in the equation. When you enter or edit a variable value in the interactive-solver editor, the new value is stored to the variable in memory. For any variable, you may enter an expression, which is evaluated when you press `ENTER`, `↑`, `→`, or `EXIT`. Expressions must resolve to real numbers at each step of the calculation.

Controlling the Solution with Bounds and a Guess

The solver seeks a solution only within the specified bounds. Whenever you display the interactive-solver editor, the default `bound={-1E99,1E99}` is displayed. These are the maximum bounds for the TI-86.
The TI-86 solves equations through an iterative process. To control that process, you can enter lower bounds and upper bounds that are close to the solution, and enter a guess within those bounds in the prompt for the unknown variable.

Controlling the process with specific bounds and a guess helps the TI-86 in two ways.

♦ It finds a solution more quickly.
♦ It is more likely to find the solution you want when an equation has multiple solutions.

To set more precise bounds at the bound= prompt, the syntax is:

```
bound={lowerBound,upperBound}
```

At the prompt for the unknown variable, you may enter a guess or a list of two guesses. If you do not enter a guess, the TI-86 uses \((\text{lowerBound} + \text{upperBound})/2\) as a guess.

On the solver graph (page 207), you can guess a solution by moving the free-moving cursor or trace cursor to a point on the graph between \(\text{lowerBound}\) and \(\text{upperBound}\). To solve for the unknown variable using the new guess, select SOLVE from the solver graph menu. The solution is displayed on the interactive-solver editor.

**Editing the Equation**

To edit the equation stored to \(\text{eqn}\) when the interactive-solver editor is displayed, press \(\text{â„œ}\) until the cursor is on the equation. The equation-entry editor is displayed. The TI-86 automatically stores the edited equation to \(\text{eqn}\) as you edit.

If you store an equation to \(\text{eqn}\) by recalling the contents of an equation variable, such as \(y1\), and then edit the equation stored to \(\text{eqn}\), the original equation (in \(y1\), for example) is not changed. Likewise, subsequently editing the contents of the equation variable (\(y1\), for example) does not change \(\text{eqn}\).
The Solver Menu

You can display other menus in the interactive-solver editor.

You can display other menus in the interactive-solver editor.

<table>
<thead>
<tr>
<th>GRAPH</th>
<th>WIND</th>
<th>ZOOM</th>
<th>TRACE</th>
<th>SOLVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>graphs the equation in eqn</td>
<td>solver zoom menu</td>
<td>solves for the unknown variable or displays the interactive-solver editor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To display the window editor, select WIND from the solver menu.

When you select GRAPH or WIND from the solver menu, EDIT replaces the item you selected on the menu. To return to the interactive-solver editor from the graph or window editor, select EDIT.

Solving for the Unknown Variable

After you have stored all known variable values, set the bounds, and entered a guess (optional), move the cursor to the prompt for the unknown variable.

To solve, select SOLVE from the solver menu (F5).

- A small square marks the variable for which you solved. The solution value is displayed.
- A small square also marks the left-rt prompt. The value at this prompt is the value of the left side of the equation minus the value of the right side of the equation evaluated at the new value of the variable for which you solved. If the solution is precise, left-rt=0 is displayed.

Some equations have more than one solution. To look for additional solutions, you can enter a new guess or set new bounds, and then solve for the same variable.
Graphing the Solution

When you select **GRAPH** from the solver menu ([F1]), the solver graph is displayed with the free-moving cursor.

- The vertical axis represents the result of the left side of the equation minus the right side of the equation (left–right) at each independent variable value.
- The horizontal axis represents the independent variable for which you solved the equation.

On the graph, solutions exist for the equation where \( \text{left} - \text{rt} = 0 \), which is where the graph intersects the x-axis. The solver graph:

- Uses the current window and format settings (Chapter 5).
- Does not graph the solution according to the current graphing mode.
- Always graphs a solution as a function graph.
- Does not graph selected functions or turned on stat plots along with the solution.

**Solver Graph Tools**

You can explore the graph of a solution with the free-moving cursor, as you would on any other graph. When you do, the coordinate values for the variable (the x-axis) and the value \( \text{left} - \text{rt} \) (the y-axis) are updated.

To activate the trace cursor, select **TRACE** from the solver menu. Panning, QuickZoom, and entering a specific value (Chapter 6) are available with the trace cursor on the solver graph. To return to the solver menu from a trace, press **EXIT**.
Chapter 15: Equation Solving

The Solver ZOOM Menu

2nd [SOLVER] equation ENTER F3

<table>
<thead>
<tr>
<th>GRAPH</th>
<th>WIND</th>
<th>ZOOM</th>
<th>TRACE</th>
<th>SOLVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOX</td>
<td>ZIN</td>
<td>ZOUT</td>
<td>ZFACT</td>
<td>ZSTD</td>
</tr>
</tbody>
</table>

Chapter 6 and the A to Z Reference describe these features in detail.

- BOX: Draws a box to redefine the viewing window (Chapter 6)
- ZIN: Magnifies the graph around the cursor by factors of xFact and yFact (Chapter 6)
- ZOUT: Displays more of the graph around the cursor by factors of xFact and yFact (Chapter 6)
- ZFACT: Displays the ZOOM FACTORS screen (Chapter 6)
- ZSTD: Displays the graph in standard dimensions; resets the default window variable values for Func graphing mode

The Simultaneous Equation Solver

2nd [SIMULT]

The simultaneous equation solver solves systems of up to 30 linear equations with 30 unknowns.

Entering Equations to Solve Simultaneously

1. Display the SIMULT number screen.
2. Enter an integer ≥ 2 and ≤ 30 for the number of equations. The coefficients-entry editor for the first equation (for a system of n equations and n unknowns) is displayed. The SIMULT ENTRY menu also is displayed.
3. [ENTER]

The SIMULT coefficients are not variables.

You can display other menus in the coefficients-entry screen.
Enter a real or complex value (or an expression that resolves to one) for each coefficient in the equation and for $b_1$, which is the solution to that equation.

Display the coefficients-entry screen for the second and third equation, and enter values for them.

Solve the equations. The results of the polynomial are calculated and displayed on the result screen. Results are not stored to variables and cannot be edited. The SIMULT RESULT menu is displayed.
Storing Equation Coefficients and Results to Variables

- To store coefficients \( a_{1,1}, a_{1,2}, \ldots, a_{n,n} \) to an \( n \times n \) matrix, select \texttt{STOa}.
- To store solutions \( b_1, b_2, \ldots, b_n \) to a vector of dimension \( n \), select \texttt{STOb}.
- To store the results \( x_1, x_2, \ldots, x_n \) to a vector of dimension \( n \), select \texttt{STOx}.

To store a single value on the coefficients-entry screen or result screen, follow these steps.

1. Move the cursor to the = sign next to the coefficient or result you want to store.
2. Display the variable \texttt{Name=} prompt. \texttt{ALPHA}-lock is on.
3. Enter the variable to which you want to store the value.
4. Store the value. The variable name becomes an item on the \texttt{VARS REAL} screen or \texttt{VARS CPLX} screen.

To return to the coefficients-entry screen, where you can edit coefficients and calculate new solutions, select \texttt{COEFS} from the \texttt{SIMULT RESULT} menu.
Chapter 15: Equation Solving

The Polynomial Root-Finder

The root finder solves up to 30th-order real or complex polynomials.

Entering and Solving a Polynomial

1. Display the POLY order screen.

   2. Enter an integer between 2 and 30. The coefficients-entry editor is displayed with the equation across the top, the coefficient prompts along the left side, and the POLY ENTRY menu on the bottom.

3. Enter a real or complex value (or an expression that resolves to one) for each coefficient.

   To clear all coefficients, select CLRa from the POLY ENTRY menu.

4. Solve the equation. The roots of the polynomial are calculated and displayed. Results are not stored to variables and you cannot edit them. Also, the POLY RESULT menu is displayed. Results can be complex numbers.

The POLY coefficients are not variables.

You can display other menus in the coefficients-entry editor.

Ellipses indicate that a value continues beyond the screen. Press  and  to scroll the value.
Storing a Polynomial Coefficient or Root to a Variable

1. Move the cursor to the = sign next to the coefficient or root value you want to store.
2. Display the **Sto** prompt. ALPHA-lock is on.
3. Enter the variable to which you want to store the value.
4. Store the value.
5. Display the **Name=** prompt for the coefficients list name. ALPHA-lock is on.
6. Enter the list variable name to which you want to store the coefficients.
7. Store the polynomial coefficient values.

To return to the coefficients-entry screen, where you can edit coefficients and calculate new solutions, select **COEFS** from the POLY RESULT menu.

To switch to the coefficients-entry screen, select **COEFS** from the POLY RESULT menu.

To find roots on the homescreen or in a program, select **poly** from the CATALOG.
16 Programming

Writing a Program on the TI-86 ....................................... 214
Running a Program ...................................................... 221
Working with Programs ............................................... 223
Running an Assembly Language Program ..................... 225
Entering and Storing a String .................................... 226
Writing a Program on the TI-86

A program is a set of expressions, instructions, or both, which you enter or download. Expressions and instructions in the program are executed when you run the program.

You can use most TI-86 features in a program. Programs can retrieve and update all variables stored to memory. Also, the program editor menu has input/output commands, such as Input and Disp, and program control commands, such as If, Then, For, and While.

The PRGM Menu

The TI-86 distinguishes between uppercase and lowercase letters in program names. For example, ABC, Abc, and abc would be three different program names.

Creating a Program in the Program Editor

To begin writing a program, select EDIT from the PRGM menu (PRGM [F2]). The program Name= prompt and PRGM NAMES menu are displayed. ALPHA-lock is on. Enter a program name from one to eight characters long, beginning with a letter. To edit an existing program, you can select the name from the PRGM NAMES menu.
After you enter a program name, press ENTER. The program editor and program editor menu are displayed. The program name is displayed at the top of the screen. The cursor is on the first command line, which begins with a colon. The TI-86 automatically places a colon at the beginning of each command line.

As you write the program, the commands are stored to the program name.

**The Program Editor Menu**

<table>
<thead>
<tr>
<th>PRGM</th>
<th>F2</th>
<th><strong>programName</strong></th>
<th>ENTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE↓</td>
<td>PAGE↑</td>
<td>I/O</td>
<td>CTL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>page down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>input/output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>insert a blank</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>command line</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>delete (cut) a</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>command line</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>paste a colon</td>
<td></td>
</tr>
</tbody>
</table>

**The PRGM I/O (Input/Output) Menu**

<table>
<thead>
<tr>
<th>PRGM</th>
<th>F2</th>
<th><strong>programName</strong></th>
<th>ENTER</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE↓</td>
<td>PAGE↑</td>
<td>I/O</td>
<td>CTL</td>
<td>INSc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input</td>
<td>Promp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disp</td>
<td>DispG</td>
<td>DispT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getKy</td>
<td>CILCD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Outpt</td>
<td>InpSt</td>
<td></td>
</tr>
</tbody>
</table>

The PRGM I/O menu items are instructions. The actions they perform occur as the program runs.

To see examples that show how to use PRGM I/O menu items in programs, refer to the A to Z Reference.
If you enter an expression for variable at an Input or Prompt prompt, it is evaluated and stored.

For Input and Prompt, built-in variables such as y1 and r1 are not valid as variable.

To halt the program temporarily after Disp or DispG and examine what the program is displaying, enter Pause on the next command line (page 219).

Input
Input variable

Input
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
ClTbl
Get(variable)
Send(listName)
getKy
CILCD

Displays the current graph and lets you use the free-moving cursor
Pauses a program, displays ? as a prompt, and then stores your response to variable
Pauses a program, displays promptString or string (up to 21 characters) as a prompt, and then stores your response to variable
Although using Get( is preferred on the TI-86, you can use Input to receive variable from a CBL 2/CBL, CBR, or TI-86 (TI-85 compatible)
Displays each variable with ? to prompt you to enter a value for that variable
Displays the home screen
Displays each value
Displays the value stored to each variable
Displays each text string on the left side of the current display line
Displays the current graph
Displays the current table and temporarily halts the program
Clears the current table if Indpt: Ask is set (Chapter 7)
Get(variable) Gets data from a CBL 2/CBL, CBR, or another TI-86 and stores it to variable
Send(listName) Sends the contents of listName to a CBL 2/CBL or CBR
getKy Returns a number corresponding to the last key pressed, according to the key code diagram (page 217); if no key was pressed, returns 0
CILCD Clears the home screen (LCD stands for liquid crystal display)
"string"

Specifies the beginning and end of a string

Output(row,column,"string")

Displays string,<stringName>,<value>, or a value stored to <variable>

beginning at the specified <row> and <column> on the display

Output(row,column,<stringName>)

Output(row,column,<value>)

Output(row,column,<variable>)

Output("CBLSEND",<listName>)

Although using Send() is preferred on the TI-86, you can use Output() to send <listName> to a CBL 2/CBL or CBR (for TI-85 compatibility)

InpSt promptString,<variable>

Pauses a program, displays promptString or ?, and waits for a response; stores the response to <variable> always as a string; omit quotation marks from your response

The TI-86 Key Code Diagram

When getKy is encountered in a program, it returns a number corresponding to the last key pressed, according to the key code diagram to the right. If no key has been pressed, getKy returns 0.

Use getKy inside loops to transfer control, such as when you create a video game.

This program returns the key code of each key you press.

:Float
:0⇒A
:Lbl TOP
:getKy⇒A
:If A>0
:Disp A
:Goto TOP

To break (interrupt) the program, press [09] and then press [F5].
The PRGM CTL Menu

The PRGM CTL Menu

If  Then  Else  For  End
While  Repea  Menu  Lbl  Goto
IS>  DS<  Pause  Retur  Stop
DelVa  GrStl  LCust

To see examples that show how to use PRGM CTL menu items in programs, refer to the A to Z Reference.

If, While, and Repeat instructions can be nested.

If condition

If condition is false (evaluates to 0), the next program command is skipped; if condition is true (evaluates to a nonzero value), the program continues on to the next command.

Then

Following If, executes a group of commands if condition is true.

Else

Following If and Then, executes a group of commands if condition is false.

For( loops can be nested.

For( variable, begin, end [step] )

Starting at begin, repeats a group of commands by an optional real step until variable > end; default step is 1.

End

Identifies the end of a group of program commands; For(, While, Repeat, and Else groups must end with End; Then groups without an associated Else instruction also must end with End.

While condition

Repeats a group of commands while condition is true; condition is tested when the While instruction is encountered; typically, the expression that defines condition is a relational test (Chapter 3).

Repeat condition

Repeats a group of commands until condition is true; condition is
Menu\( (\text{item}\#, \text{"title1"}, \text{"label1"}, \text{item}\#, \text{"title2"}, \text{label2}...) \) tested when the End instruction is encountered

Sets up branching within a program as selected from menu keys \([F1]\) through \([F5]\); when encountered, displays the first of up to 3 menu groups (up to 15 titles); when you select a title, the program branches to the label that the title represents; item\# is an integer \(\geq 1\) and \(\leq 15\) that specifies title's menu placement; title is a text string from one to eight characters long (may be abbreviated in the menu).

Lbl \(\text{label}\)

Assigns a label to a program command; label can be one to eight characters long, starting with a letter.

Goto \(\text{label}\)

Transfers control to the program branch labeled with label.

IS\(>(\text{variable}, \text{value})\)

Adds 1 to variable; if the answer is \(>\) value, the next command is skipped; if the answer is \(\leq\) value, the next command is executed; variable cannot be a built-in variable.

DS\(<(\text{variable}, \text{value})\)

Subtracts 1 from variable; if the answer is \(<\) value, the next command is skipped; if the answer is \(\geq\) value, the next command is executed; variable cannot be a built-in variable.

Pause

Halt the program so that you can examine results, including displayed graphs and tables; to resume the program, press [ENTER].

Pause \(\text{value}\)

Displays value on the home screen so that you can scroll large values, such as lists, vectors, or matrices; to resume, press [ENTER].

Return

Exits a subroutine (page 224) and returns to the calling program, even if encountered within nested loops; within the main program, stops the program and returns to the home screen (an implied Return exits each subroutine upon completion and returns to the calling program).

Stop

Stops a program and returns to the home screen.
Chapter 16: Programming

DelVar(variable)

Deletes from memory variable (except program names) and its contents.

GrStl(function#,graphStyle#)

Specifies the graph style represented by graphStyle# for the function represented by function#; function# is the number part of an equation variable, such as the 5 in y5; graphStyle# is an integer ≥ 1 and ≤ 7, where 1 = \ (line), 2 = \ (thick), 3 = \ (shade above), 4 = \ (shade below), 5 = \ (path), 6 = \ (animate), and 7 = \ (dotted).

*LCust(item#,"title"

Loads (defines) the TI-86 custom menu, which is displayed when you press CUSTOM; item# is an integer ≥ 1 and ≤ 15; title is a string with one to eight characters (may be abbreviated in the menu).

Entering a Command Line

You can enter on a command line any instruction or expression that you could execute on the home screen. In the program editor, each new command line begins with a colon. To enter more than one instruction or expression on a single command line, separate each with a colon.

To move the cursor down to the next new command line, press \. You cannot move to the next new command line by pressing #. However, you can return to existing command lines to edit them by pressing #.

Menus and Screens in the Program Editor

TI-86 menus and screens may be altered when displayed in the program editor. Menu items that are invalid for a program are omitted from menus. Menus that are not valid in a program, such as the LINK menu or MEM menu, are not displayed at all.

When you select a setting from a screen such as the mode screen or graph format screen, the setting you select is pasted to the cursor location on the command line.
Variables to which you typically store values from an editor, such as the window variables, become items on program-only menus, such as the GRAPH WIND menu. When you select them, they are pasted to the cursor location on the command line.

**Running a Program**

1. Paste the program name to the home screen. Either select it from the PRGM NAMES menu (PRGM F1) or enter individual characters.
2. Press [ENTER]. The program begins to run.

Each result updates the last-answer variable Ans (Chapter 1). The TI-86 reports errors as the program runs. Commands executed during a program do not update the previous-entry storage area ENTRY (Chapter 1).

The example program below is shown as it would appear on a TI-86 screen. The program:

- Creates a table by evaluating a function, its first derivative, and its second derivative at intervals in the graphing window
- Displays the graph of the function and its derivatives in three different graph styles, activates the trace cursor, and pauses to allow you to trace the function
PROGRAM:FUNCTABL
:Func:Fix 2:FnOff:PlOff
:y1=.6 x cos x
:ClLCD
:Eq Stmt(y1,STRING)
:Outpt(1,1,″y1=″)
:Outpt(1,4,STRING)
:Outpt(8,1,″PRESS ENTER″)
:Pause
:ClLCD
:y2=der1(y1,x,x)
:y3=der2(y1,x,x)
:DispT
:GrSt1(1,1):GrSt1(2,2)
:GrSt1(3,7)
:2→xRes
:ZTrig
:Trace

The name of the program
Set graphing and decimal modes (mode screen); turn off
functions (GRAPH VARS menu) and plots (STAT PLOT menu)
Define the function (assignment statement)
Clear the home screen (PRGM I/O menu)
Convert y1 into the string variable STRING (STRING menu)
Display y1 as at row 1, column 1 (PRGM I/O menu)
Display value stored to STRING at row 1, col. 4 (PRGM I/O menu)
Display PRESS ENTER at line 8, column 4 (PRGM I/O menu)
Pause the program (PRGM CTL menu)
Clear the home screen (PRGM I/O menu)
Define y2 as the first derivative of y1 (CALC menu)
Define y3 as the second derivative of y1 (CALC menu)
Display the table (PRGM I/O menu)
Set graph styles for y1, y2, and y3 (PRGM CTL menu)
Store 2 to the window variable xRes (GRAPH WIND menu)
Set the viewing window variables (GRAPH ZOOM menu)
Display the graph, activate trace cursor, and pause (GRAPH menu)

Breaking (Interrupting) a Program

To break (interrupt) the program, press [ON]. The ERROR 06 BREAK menu is displayed.

♦ To display the program editor where the interruption occurred, select GOTO ([F1]).
♦ To return to the home screen, select QUIT ([F5]).
Working with Programs

Managing Memory and Deleting a Program
To check whether adequate memory is available for a program you want to enter or download, display the Check RAM screen (2nd MEM [F1]; Chapter 17). To increase available memory, consider deleting selected items or data types from memory (Chapter 17).

Editing a Program
After you write a program, you can display it in the program editor and edit any command line.

1. Display the program editor ([PRGM [F2]). The PRGM NAMES menu also is displayed.
2. Enter the name of the program you want to edit. Either select the name from the PRGM NAMES menu or enter the individual characters.
3. Edit the program command lines.
   - Move the cursor to the appropriate location, and then delete, overwrite, or insert characters.
   - Press [CLEAR] to clear the entire command line, except for the leading colon, and then enter a new program command.
   - Select program editor menu items [INSc (F5) and [DELC (MORE [F1]) to insert and delete command lines.

The program editor does not display a \( \downarrow \) to indicate that command lines continue beyond the screen.
Calling a Program from Another Program

On the TI-86, any stored program can be called from another program as a subroutine. In the program editor, enter the subroutine program name on a command line by itself.

- Press PRGM to display the PRGM NAMES menu, and then select the program name.
- Use ALPHA keys and alpha keys to enter the program name’s individual characters.

When the program name is encountered as the calling program runs, the next command executed is the first command in the subroutine. It returns to the next command in the calling program when it encounters Return (or implied Return) at the end of a subroutine.

-label used with Goto andLbl is local to the program where it is located. label in one program is not recognized by another program. You cannot use Goto to branch to a label in another program.
Copying a Program to Another Program Name

1. Display a new or existing program in the program editor.
2. Move the cursor to the command line on which you want to copy a program.
3. Display the Rcl prompt (2nd [RCL]).
4. Enter the name of the program you want to copy. Either select the name from the PRGM NAMES menu or enter individual characters.
5. Press [ENTER]. The contents of the recalled program name are inserted into the other program at the cursor location.

Using and Deleting Variables within a Single Program

If you want to use variables within a program but do not need them after the program is run, you can use DelVar( within the program to delete the variables from memory.

The program segment to the right uses the variables A and B as counters and then deletes them from memory.

Running an Assembly Language Program

An assembly language program is a program that runs much faster and has greater control of the calculator than the regular programs described in this chapter. You can download and run TI-created assembly language programs to add features to your TI-86 that are not built in. For example, you can download the TI-83 finance or inferential statistics features to use on your TI-86.
TI assembly language programs and other programs are available on TI's World Wide Web site:
http://www.ti.com/calc

When you download an assembly language program, it is stored among the other programs as a PRGM NAMES menu item. You can:

♦ Transmit it using the TI-86 communication link (Chapter 18).
♦ Delete it using the MEM DELETE:PRGM screen (Chapter 17).
♦ Call it from another program as a subroutine (page 224).

To run an assemblyProgramName, the syntax is: Asm(assemblyProgramName)

If you write an assembly language program, use the two instructions below from the CATALOG.

AsmComp(AsciiAssemblyPrgmName,
HexAssemblyPrgmName) Compiles an assembly language program written in ASCII and stores the hex version
AsmPrgm Identifies an assembly language program; must be entered as the first line of an assembly language program

**Entering and Storing a String**

A string is a sequence of characters that you enclose within quotation marks.

♦ A string defines characters to be displayed in a program.
♦ A string accepts input from the keyboard in a program.

To enter a string directly, the syntax is:
"string"

To concatenate (join together) two or more strings, use +. The syntax is:
"stringA"+"stringB"+"stringC"+...
The STRNG (String) Menu

2nd [STRING]

“” sub lngth Eq

Marks the start and end of string

sub("string",begin,length)

Returns a subset of "string" or stringName, starting at
begin character place and length characters long

sub(stringName,begin,length)

string

begin

length

Eq

Returns the number of characters in "string" or stringName

lngth "string" or lngth stringName

Converts equationVariable contents to stringName

EqSt(equationVariable,stringName)

Converts stringName to equationVariable

SbEq(stringName,equationVariable)

Creating a String

1. Display the STRNG menu.
2. Enter the open quotation mark, then
   the string SOLVE & GRAPH, and then
   the close quotation mark.
3. Store the string to the string variable
   name LABEL.

To evaluate the contents of a
string, you must use SbEq( to convert it to an equation.
Checking Available Memory ............................................ 230
Deleting Items from Memory ........................................... 231
Resetting the TI-86 .......................................................... 232
Checking Available Memory

The MEM (Memory) Menu

<table>
<thead>
<tr>
<th>RAM</th>
<th>DELET</th>
<th>RESET</th>
<th>TOL</th>
<th>ClrEnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>check-RAM screen</td>
<td>memory/default reset menu</td>
<td>clears ENTRY storage area</td>
<td>memory delete menu</td>
<td>tolerance editor</td>
</tr>
</tbody>
</table>

Checking Memory Usage

When all memory is cleared and all defaults are set, the standard TI-86 has 98,224 bytes of available random-access memory (RAM). As you store information to RAM, you can monitor memory allocation on the Check RAM screen.

MEM FREE reports the total number of bytes available in RAM. Conversely, all other numbers on the screen report the number of bytes that each data type currently occupies. For example, if you were to store a 50-byte matrix in memory, the MATR total would increase to 50 bytes, while the MEM FREE total would decrease by 50 to 98174 bytes.

To display the number of bytes that a specific variable occupies, display the DELETE screen for that data type (page 231). Scroll the screen, if necessary.

For information on TOL (the tolerance editor), refer to the Appendix.
Deleting Items from Memory

The MEM DELET (Delete) Menu

1. Select DELET from the MEM menu to display the MEM DELET menu.
2. Select the data type of the item you want to delete. To scroll down to the next six items or up to the previous six items, select PAGE $ or PAGE #.
3. Move the selection cursor (▼) to the item you want to delete (y5). The uppercase items are in alphanumeric order, followed by the lowercase items in alphanumeric order.
4. Delete the item. To delete other items on the screen, repeat steps 3 and 4.

Each MEM DELET menu item displays the deletion screen for that data type. For example, when you select LIST, the MEM DELETE:LIST screen is displayed. Use the DELETE screens to delete any user-created variable and the information stored to it.

To delete a parametric equation, delete the xt component.

In the example, the equation y = x^3 - x^2 - 4x - 1 is deleted.

To move directly to the first item beginning with any letter, enter that letter; ALPHA-lock is on.
Resetting the TI-86

The MEM RESET (Reset) Menu

<table>
<thead>
<tr>
<th>RAM</th>
<th>DELET</th>
<th>RESET</th>
<th>TOL</th>
<th>ClrEnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>MEM</td>
<td>DFLTS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ALL When confirmed, all data is cleared and memory is reset; both messages are displayed
MEM When confirmed, clears all stored data from memory; Mem Cleared is displayed
DFLTS When confirmed, resets all defaults; Defaults Set is displayed

When you select ALL, MEM, or DFLTS, a confirmation menu is displayed.

- To confirm the selected reset, select YES (press [F4]).
- To cancel the selected reset, select NO (press [F5]).

ClrEnt (Clear Entry)  

The TI-86 retains as many previous entries as possible in ENTRY, up to a capacity of 128 bytes.

To clear the ENTRY storage area of all entries, execute ClrEnt on a blank line on the home screen (2nd MEM [F5] ENTER).
The TI-86 Communication Link

TI-86 Linking Options ...................................................... 234
Connecting the TI-86 to Another Device ............................. 235
Selecting Data to Send ..................................................... 236
Preparing the Receiving Device ....................................... 240
Transmitting Data ......................................................... 240
Receiving Transmitted Data ............................................ 240
**TI-86 Linking Options**

Using the unit-to-unit cable included with the TI-86, you can transmit data between the TI-86 and several other devices.

**Linking Two TI-86s**
You can link two TI-86 units and select the data types to be transmitted, including programs. You can back up the entire memory of a TI-86 onto another TI-86.

**Linking a TI-86 and a TI-85**
You can select the data types, including programs, to transfer from a TI-85 to a TI-86. You can send most variables and programs from a TI-86 to a TI-85 using **SND85** (page 239), except lists, vectors, or matrices that exceed TI-85 capacity.

When you run a TI-85 program on a TI-86, the TI-85 **PrtScrn** program instruction is not valid. Also, the EOS implied multiplication on the TI-86 differs from the TI-85 (Appendix). For example, the TI-85 interprets \( \sin 2x \) as \( \sin (2x) \); the TI-86 interprets \( \sin 2x \) as \( (\sin 2)x \).

**Linking a TI-86 and a CBL 2/CBL or CBR System**
The Calculator-Based Laboratory™ (CBL 2™/CBL™) and Calculator-Based Ranger™ (CBR™) systems are optional TI accessories that collect data from physical occurrences, such as science experiments. The CBL 2/CBL and CBR store data to lists, which you can transmit to a TI-86 and analyze. You can transmit list names to a CBL 2/CBL or CBR from a TI-86.
Linking a TI-86 and a PC or Macintosh

TI-86 TI-GRAPH LINK™ is an optional system that links a TI-86 with an IBM®-compatible or Macintosh® computer.

Downloading Programs from the Internet

If you have TI-GRAPH LINK and internet services, you can download programs from TI's World Wide Web site at:

http://www.ti.com/calc

You can download various programs from TI's web site, including assembly language programs that add features such as TI-83 finance and inferential statistics. The site also links to many other TI-86 web sites maintained by user groups, high schools, universities, and individuals.

Connecting the TI-86 to Another Device

Before you begin to transmit data to or from the TI-86, connect it to the other device.

1. Firmly insert one end of the unit-to-unit cable into the port on the bottom edge of the calculator.
2. Firmly insert the other end of the cable into the other device (or PC adapter).
The LINK Menu

| SEND | RECV | SND85 |

The link menus are not available in the program editor.

Selecting Data to Send

To list the variables for a specific data type on a selection screen, select the data type from the LINK SEND menu. When you select BCKUP, the message Memory Backup is displayed.

The LINK SEND Menu

<table>
<thead>
<tr>
<th>BCKUP</th>
<th>PRGM</th>
<th>MATRX</th>
<th>GDB</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>VECTR</td>
<td>REAL</td>
<td>CPLX</td>
<td>EQU</td>
</tr>
<tr>
<td>CONS</td>
<td>PIC</td>
<td>WIND</td>
<td>STRNG</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 18: The TI-86 Communication Link

Initiating a Memory Backup

To initiate a memory backup, select BCKUP from the LINK SEND menu (2nd [LINK] F1 F1). The screen to the right is displayed.

To complete memory backup, prepare the other unit to receive data transmission (page 239), and then select XMIT from the memory backup menu (F1).

**Warning:** When you transmit BCKUP, the transmitted memory overwrites all memory in the receiving unit; all information in the memory of the receiving unit is lost. To cancel initiation of a memory backup, press EXIT.

As a safety check to prevent accidental loss of memory, when the receiving calculator is notified of an incoming backup transmission, it displays the warning message and confirmation menu, as shown in the screen to the right.

- To continue the backup transmission, select CONT. The backup transmission continues, replacing all receiving-calculator memory with the backup data.
- To cancel backup and retain all receiving-calculator memory, select EXIT.

Selecting Variables to Send

If a transmission error occurs during a backup, the receiving-calculator memory is reset.

If no data of the type you select is stored in memory, the message is displayed: NO VARS OF THIS TYPE.
When you select any LINK SEND menu item, except BCKUP or WIND, each variable of the selected data type is listed in alphanumeric order on a selection screen. The screen to the right is the SEND ALL screen (2nd [LINK] F1 [F6]).

♦ The data type of each variable is specified.
♦ Small squares indicate that xStat, yStat, and Q2 are selected to be sent.
♦ The selection cursor is next to Q4.

To select a specific variable to be sent, use [ and ] to move the selection cursor next to the variable, and then select SELCT (F2) from the selection screen menu.
♦ To select all variables of this type, select ALL+ from the selection screen menu (F3).
♦ To deselect all variables of this type, select ALL- from the selection screen menu (F4).

To complete transmission of the selected variables, prepare the other unit to receive data transmission (page 239), and then select XMIT from the selection screen menu (F1).

**The SEND WIND (Window Variables) Screen**

When you select WIND from the LINK SEND menu (2nd [LINK] F1 MORE MORE F3), the SEND WIND screen is displayed.

Each SEND WIND screen item represents the window variables, format settings, and any other graph-screen data for that TI-86 graphing mode and for ZRCL (user-created zoom). The screen to the right shows that the graph screen data for Func and DifEq graphing modes are selected.

**Func** Select to send Func graphing mode window variable values and format settings
Pol Select to send Pol graphing mode window variable values and format settings

Param Select to send Param graphing mode window variable values and format settings

DifEq Select to send DifEq graphing mode window variable values, difTol, axes settings, and format settings

ZRCL Select to send user-created zoom window variables, and format settings in any mode

To complete transmission of the selected variables, prepare the other unit to receive data transmission (below), and then select XMIT from the memory backup menu (F1).

Sending Variables to a TI-85

The steps for selecting variables to send to a TI-85 are the same as those for selecting variables to send to a TI-86. However, the LINK SND85 menu has fewer items than the LINK SEND menu.

The TI-86 has more capacity for lists, vectors, and matrices than the TI-85. If you send to the TI-85 a list, vector, or matrix that has more elements than the TI-85 allows, the elements that exceed TI-85 capacity are truncated.

The LINK SND85 (Send Data to TI-85) Menu

<table>
<thead>
<tr>
<th>MATRX</th>
<th>LIST</th>
<th>VECTR</th>
<th>REAL</th>
<th>CPLX</th>
<th>CONS</th>
<th>PIC</th>
<th>STRNG</th>
</tr>
</thead>
</table>

2nd [LINK] F3
Preparing the Receiving Device

To prepare a TI-86 or TI-85 to receive data transmission, select RECV from the LINK menu (2nd [LINK] F2). The message Waiting and the busy indicator are displayed. The calculator is ready to receive transmitted items.

To cancel receive mode without receiving items, press [ON]. When the LINK TRANSMISSION ERROR message is displayed, select EXIT from the menu (F1). The LINK menu is displayed.

Transmitting Data

After you select data types on the sending unit and prepare the receiving unit to receive data, you can begin transmitting.

To begin transmitting, select XMIT on the selection screen menu of the sending calculator (F1).

To interrupt transmission, press [ON] on either calculator. When the LINK TRANSMISSION ERROR message is displayed, select EXIT from the menu (F1). The LINK menu is displayed.

Receiving Transmitted Data

As the TI-86 receives transmitted data, each variable name and data type is displayed line by line. If all selected items are transmitted successfully, the message Done is displayed. To scroll the transmitted variables, press [ ▲ ] and [ ▼ ].
During transmission, if a transmitted variable name is stored already in the memory of the receiving calculator, transmission is interrupted. The duplicated variable name, its data type, and the DUPLICATE NAME menu are displayed, as shown in the screen to the right.

To resume or cancel transmission, you must select an item from the DUPLICATE NAME menu.

**RENAME** Displays the **Name** prompt; enter a unique variable name; press **ENTER** to continue transmission

**OVERW** (overwrite) Replaces data stored to the receiving unit’s variable with sent variable data

**SKIP** Does not overwrite the receiving unit’s data; attempts to send the next selected variable

**EXIT** Cancels the data transmission
Repeating Transmission to Several Devices

After transmission is complete, the LINK menu is displayed and all selections remain. You can transmit the same selections to a different TI-86 without having to re-select data.

To repeat a transmission with another device, disconnect the unit-to-unit cable from the receiving unit; connect it to another device; prepare the device to receive data; and then select SEND, then ALL, and then XMIT.

Error Conditions

A transmission error occurs after a few seconds if:

- The cable is not connected to the port of the sending calculator.
- The cable is not connected to the port of the receiving calculator.
- The receiving unit is not set to receive transmission.
- You attempt a backup between a TI-86 and a TI-85.

Insufficient Memory in Receiving Unit

If the receiving unit does not have sufficient memory to receive an item, the receiving unit displays LINK MEMORY FULL and the variable name and data type.

- To skip the variable, select SKIP. Transmission resumes with the next item.
- To cancel transmission altogether, select EXIT.

If the cable is connected but a transmission error occurs, push the cable in more firmly to both calculators and try again.
19 Applications

Using Math Operations with Matrices ......................... 244
Finding the Area between Curves .............................. 245
The Fundamental Theorem of Calculus ...................... 246
Electrical Circuits ............................................... 248
Program: Taylor Series ........................................... 250
Characteristic Polynomial and Eigenvalues .................. 252
Convergence of the Power Series .............................. 254
Reservoir Problem ............................................... 256
Predator-Prey Model ............................................. 258
Program: Sierpinski Triangle ................................. 260
Using Math Operations with Matrices

1. In the matrix editor, enter matrix \( A \) as shown.

2. On the home screen, select \( \text{rref} \) from the \( \text{MATRX} \) \( \text{OPS} \) menu.

3. To append a 3\( \times \)3 identity matrix to matrix \( A \), select \( \text{aug} \) from the \( \text{MATRX} \) \( \text{OPS} \) menu, enter \( A \), select \( \text{ident} \) from the \( \text{MATRX} \) \( \text{OPS} \) menu, and then enter 3. Execute the expression.

4. Enter \( \text{Ans} \) (to which the matrix from step 3 is stored). Define a submatrix that contains the solution portion of the result. The submatrix begins at element (1,4) and ends at element (3,6).

5. Select \( \text{Frac} \) from the \( \text{MATH} \) \( \text{MISC} \) menu and display the fractional equivalent of the submatrix.

6. Check the result. Set the decimal mode to 11 (the last 1) Select \( \text{round} \) from the \( \text{MATH} \) \( \text{NUM} \) menu for the product of the fractional equivalent of the submatrix times \( A \).

Displaying the result matrix elements to 11 decimal places illustrates accuracy.
Finding the Area between Curves

Find the area of the region bounded by: 
\[ f(x) = \frac{x}{(x^2 + 625)} \]
\[ g(x) = 3 \cos(0.1x) \]
\[ x = 75 \]

1. In Func graphing mode, select \( y(x) = \) from the GRAPH menu to display the equation editor and enter the equations as shown.
\[ y_1 = 300 \frac{x}{(x^2 + 625)} \]
\[ y_2 = 3 \cos(0.1x) \]

2. Select WIND from the GRAPH menu and set the window variables as shown.
\[ x_{\text{Min}} = 0 \quad x_{\text{Max}} = 100 \quad x_{\text{Scl}} = 10 \quad y_{\text{Min}} = -5 \quad y_{\text{Max}} = 10 \quad y_{\text{Scl}} = 1 \quad x_{\text{Res}} = 1 \]

3. Select GRAPH from the GRAPH menu to display the graph screen.

4. Select ISECT from the GRAPH MATH menu. Move the trace cursor to the intersection of the functions. Press ENTER to select \( y_1 \). The cursor moves to \( y_2 \). Press ENTER again to set the current cursor location as the initial guess. The solution uses the solver. The value of \( x \) at the intersection, which is the lower limit of the integral, is stored to \( \text{Ans} \) and \( x \).

5. The area to integrate is between \( y_1 \) and \( y_2 \), from \( x = 5.5689088189 \) to \( x = 75 \). To see the area on a graph, return to the home screen, select Shade from the GRAPH DRAW menu, and execute this expression:
\[ \text{Shade}(y_2, y_1, \text{Ans}, 75) \]

6. Select TOL from the MEM menu and set \( \text{tol} = 1 \times 10^{-5} \).

7. On the home screen, compute the integral with \( \text{fnInt} \) (CALC menu). The area is 325.839961998.
\[ \text{fnInt}(y_1 - y_2, x, \text{Ans}, 75) \]
The Fundamental Theorem of Calculus

Consider these three functions:

\[ F(x)_1 = \frac{(\sin x)}{x} \quad F(x)_2 = \int_0^x \frac{(\sin t)}{t} \, dt \quad F(x)_3 = \frac{d}{dx} \int_0^x \frac{(\sin t)}{t} \, dt \]

1. In Func graphing mode, select \( y(x) \) from the GRAPH menu, and then enter the functions and set graph styles in the equation editor as shown. (\( \text{fnInt} \) and \( \text{nDer} \) are CALC menu items.)

\[ y_1 = (\sin x) \div x \quad y_2 = \text{fnInt}(y_1(x), t, 0, x) \quad y_3 = \text{nDer}(y_2, x) \]

2. Select TOL from the MEM menu to display the tolerance editor. To improve the rate of the calculations, set \( \text{tol}=0.1 \) and \( \delta=0.001 \).

3. Select WIND from the GRAPH menu and set the window variable values as shown.

\[ x_{\text{Min}}=10 \quad x_{\text{Max}}=10 \quad x_{\text{Scl}}=1 \quad y_{\text{Min}}=-2.5 \quad y_{\text{Max}}=2.5 \quad y_{\text{Scl}}=1 \quad \text{xRes}=4 \]

4. Select TRACE from the GRAPH menu to display the graph and the trace cursor.

5. Trace \( y_1 \) and \( y_3 \) to verify that the graph of \( y_1 \) and the graph of \( y_3 \) are visually indistinguishable.

The inability to visually distinguish between the graphs of \( y_1 \) and \( y_3 \) graphically supports the fact that:

\[ \frac{d}{dx} \int_0^x \frac{(\sin t)}{t} \, dt = \frac{(\sin x)}{x} \]
6. Deselect y2 in the equation editor.
7. Select TBLST from the TABLE menu. Set TblStart=1, ΔTbl=1, and Indpnt: Auto.
8. Select TABLE from the TABLE menu to display the table. Compare the solution of y1 with the solution of y3 to numerically support the formula above.
Electrical Circuits

A measurement device has measured the DC current (C) in milliamperes and voltage (V) in volts on an unknown circuit. From these measurements, you can calculate power (P) in milliwatts using the equation $CV = P$. What is the average of the measured power?

With the TI-86, you can estimate the power in milliwatts at a current of 125 milliamperes using the trace cursor, the interpolate/extrapolate editor, and a regression forecast.

1. In two consecutive columns of the list editor, store the current measurements shown below to the list name CURR and the voltage measurements shown below to the list name VOLT.
   
   \[
   \{10, 20, 40, 60, 80, 100, 120, 140, 160\} \rightarrow \text{CURR}
   \]
   
   \[
   \{2, 4.2, 10, 18, 32.8, 56, 73.2, 98, 136\} \rightarrow \text{VOLT}
   \]

2. In the next column of the list editor, enter the list name POWER.

3. Enter the formula CURR * VOLT in the list editor entry line for POWER. Press \[ \text{ ENTER } \] to calculate the values for power and store the answers to the list name POWER.

4. Select WIND from the GRAPH menu and set the window variable values as shown.
   
   \[
   \text{xMin}=0 \quad \text{xMax}=\text{max(POWER)} \quad \text{xScl}=1000 \quad \text{yMin}=0 \quad \text{yMax}=\text{max(CURR)} \quad \text{yScl}=10 \quad \text{xRes}=4
   \]

5. From the home screen, select FnOff from the CATALOG and press \[ \text{ ENTER } \] to deselect all functions in the equation editor.

   Select \[ \text{Plot1(} \text{POWER, CURR, 1)} \]

   From the home screen, select \[ \text{Plot1(} \text{CURR, POWER, 1)} \]

   Press \[ \text{ ENTER } \] to set up a stat plot with POWER on the x-axis and CURR on the y-axis.
Select TRACE from the GRAPH menu to display the stat plot and trace cursor on the graph screen.

Trace the stat plot to approximate the value of POWER at CURR=125. With this statistical data, the closest to CURR=125 that you can trace to is CURR=120 (on the y-axis).

Select INTER from the MATH menu to display the interpolate/extrapolate editor. To interpolate POWER at CURR=125, enter the nearest pairs:

\[
\begin{align*}
\text{x1} &= \text{POWER}(7) \\
\text{y1} &= \text{CURR}(7) \\
\text{x2} &= \text{POWER}(8) \\
\text{y2} &= \text{CURR}(8)
\end{align*}
\]

Enter y=125 and solve for x.

On the home screen, select LinR from the STAT CALC menu to fit the linear regression model equation to the data stored to POWER and CURR. Write down the value of the result variable corr.

Fit the logarithmic (LnR), exponential (ExpR), and power (PwrR) regressions to the data, writing down the value of corr for each regression. Compare the corr values of each regression to determine which model fits the data most accurately (the corr value closest to 1).

Execute the most accurate regression again, and then select FCST from the STAT menu. To forecast POWER at CURR=125, enter y=125 and solve for x.

Compare this answer with the answer returned in step 9.
Program: Taylor Series

When you run this program, you can enter a function and specify the order and center point. Then the program calculates the Taylor Series approximation for the function and plots the function you entered. This example shows how to call a program from another program as a subroutine.

1. Before you enter the program TAYLOR, select EDIT from the PRGM menu, enter MOBIUS at the Name= prompt, and then enter this brief program to store the Mobius Series. The program TAYLOR calls this program and runs it as a subroutine.

```basic
PROGRAM: MOBIUS
: {1,1,0,1,1,0,1,1,0,1,1,0,0,0,0,0,1,0,1,0,1,0}® MSERIES
: Return
```

2. Select EDIT from the PRGM menu, enter TAYLOR at the Name= prompt, and then enter this program to calculate the Taylor Series.

```basic
PROGRAM: TAYLOR
: Func: FnOff
: y14® pEval(TPOLY,x=center)
: GrSt1(14,2)
: 1® E
: 9® : 1® r1
: ClLCD

User enters equation function ———> InpSt "FUNCTION: ", EQ
: St® Eq(EQ,y13)

User enters order ———> Input "ORDER: ", order
: order® 1® dimL TPOLY
: Fill(0,TPOLY)

User enters center ———> Input "CENTER: ", center
: evalF(y13,x,center)® f0
: f0® TPOLY(order+1)
```

\textbf{Begin Then group} \( \text{If order} \geq 1 \)
\( \text{der} 1(y_{13}, x, \text{center}) \mapsto \text{TPOLY(order)} \)
\( \text{If order} \geq 2 \)
\( \text{der} 2(y_{13}, x, \text{center}) / 2 \mapsto \text{TPOLY(order - 1)} \)
\( \text{If order} \geq 3 \)
\begin{itemize}
  \item\textit{Begins Then group} \( \text{der} 3(y_{13}, x, \text{center}) \mapsto \text{TPOLY(order)} \)
  \item\textit{Begins subroutine} \( \text{MOBIUS} \)
  \item\textit{Begins For group} \( \text{For} \text{N, 3, order, 1} \)
  \item\text{abs f0} \( \mapsto \text{gmax} \)
  \item\text{gmax} \( \mapsto \text{gmax} \)
  \item\text{m} \( \mapsto 0 \)
  \item\text{ssum} \( \mapsto \text{ssum} \)
  \item\textit{Begins While group} \( \text{While abs bmi} \geq \text{gmax} \)
  \item\text{While MSERIES(m)} = 0 \( \text{m} \mapsto 1 \)
  \item\text{End} \( \text{bsum} \mapsto 0 \)
  \item\text{For} \text{J, 1, m} \( \mapsto \text{N, 1} \)
  \item\text{rr} \( \mapsto e^{2 \pi (J / (m \times N)) \times (0, 1)} \) \( + \) (center, 0) \( \mapsto \text{x} \)
  \item\text{real y13} \( \mapsto \text{gval} \)
  \item\text{bsum} \( \mapsto \text{bsum} \)
  \item\text{max(abs gval, gmax)} \( \mapsto \text{gmax} \)
  \item\text{End} \( \text{bsum} \) \( / (m \times N) \) \( - \) \( \text{f0} \mapsto \text{bmi} \)
  \item\text{ssum} \( + \) \( \text{MSERIES(m)} \) \( \times \text{bmi} \) \( \mapsto \text{ssum} \)
  \item\text{m} \( \mapsto 1 \)
\end{itemize}
\begin{itemize}
  \item\textit{Ends While group} \( \text{End} \)
  \item\text{Create nested For group} \( \text{End} \)
  \item\text{Create nested While group} \( \text{End} \)
  \item\textit{Ends For group} \( \text{End} \)
  \item\textit{Ends Then group} \( \text{ZStd} \)
On the home screen, select **TAYLOR** from the PRGM NAMES menu, and then press **ENTER** to run the program.

When prompted, enter:

<table>
<thead>
<tr>
<th>FUNCTION: sin x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER: 5</td>
</tr>
<tr>
<td>CENTER: 0</td>
</tr>
</tbody>
</table>

### Characteristic Polynomial and Eigenvalues

1. In the matrix editor or on the home screen, enter matrix **A** as shown.

$$\begin{bmatrix}-1 & 2 & 5 \\ 3 & 6 & 9 \\ 2 & 5 & 7 \end{bmatrix}$$

2. On the home screen, select **eigVl** from the MATRX MATH menu to find the complex eigenvalues for the matrix **A** and store them to the list name **EV**.

3. Graph the characteristic polynomial **$C_p(x)$** of matrix **A** without knowing the analytic form of **$C_p(x)$** based on the formula **$C_p(x)=\det(A-x\cdot I)$**. In **Func** graphing mode, select **$y(x)$** from the **GRAPH** menu and enter the function in the equation editor as shown.

   $$\\\\text{\texttt{y1=det(A-x*ident 3)}}$$

4. Select **WIND** from the **GRAPH** menu and set the window variable values as shown.

   \[
   x\text{Min}=10 \quad x\text{Max}=10 \quad x\text{Scl}=1 \quad y\text{Min}=-100 \quad y\text{Max}=50 \quad y\text{Scl}=10 \quad x\text{Res}=4
   \]

5. Select **ROOT** from the **GRAPH** MATH menu and use it to display the real eigenvalue interactively. Use **Left Bound** = 5, **Right Bound** = 4, and **Guess** = 4.5.

   Compare the root (**x** value) you displayed interactively with the first element of the result list in step 2.

The first eigenvalue is real, since the imaginary part is 0.

If necessary, select **ALL-** from the equation editor menu to deselect all functions. Also, turn off all stat plots.
Next, use the list editor and a degree-three polynomial regression to find an analytic formula in terms of $x$ for the characteristic polynomial $y_1 = \text{det}(A - x \cdot \text{ident} 3)$. Create two lists that you can use to find the analytic formula.

6 In the list editor, create elements for $x_{\text{Stat}}$ by entering the expression $\text{seq}(N, N, -10, 21)$ in the $x_{\text{Stat}}$ entry line. $\text{seq}$ is on the MATH MISC menu.

7 Create elements for $y_{\text{Stat}}$ by attaching the formula "$y_1(x_{\text{Stat}})$" to $y_{\text{Stat}}$ in the entry line. The expression is evaluated when you press ENTER or exit the list editor.

8 On the home screen, select Plot1 from the CATALOG and execute Plot1(2, $x_{\text{Stat}}, y_{\text{Stat}}, 1$) to turn on Plot1 as an xyLine plot using the lists $x_{\text{Stat}}$ and $y_{\text{Stat}}$.

9 Select GRAPH from the GRAPH menu to display Plot1 and $y_1$ on the graph screen.

10 On the home screen, select P3Reg from the STAT CALC menu. Execute P3Reg $x_{\text{Stat}}, y_{\text{Stat}}, y_2$ to find the explicit characteristic polynomial in terms of $x$ and store it to $y_2$.

The cubic regression coefficients stored in the result list PRegC suggest that $a = -1$, $b = 0$, $c = 14$, and $d = -24$. So the characteristic polynomial seems to be $C_p(x) = -x^3 + 14x - 24$. 

To clear the menus from the graph screen, press CLEAR.
1. Support this conjecture by graphing $y_1$, $y_2$ (to which $Cp(x)$ is stored), and Plot1 together.

2. In the equation editor, enter the apparent characteristic polynomial of matrix $A$ and select (thick) graph style as shown.

\[ y_3 = x^3 + 14x - 24 \]

3. Graph $y_1$, $y_2$, $y_3$, and Plot1.

4. Deselect $y_2$ in the equation editor.

5. Select TABLE from the TABLE menu to display $y_1$ and $y_3$ in the table.

   Compare the values for the characteristic polynomial.

**Convergence of the Power Series**

A closed-form analytic antiderivative of $(\sin x)/x$ does not exist. However, substituting $t$ for $x$, you can find an infinite series analytic solution by taking the series definition of $\sin t$, dividing each term of the series by $t$, and then integrating term by term to yield:

\[
\sum_{n=1}^{\infty} -1^{n+1} \frac{t^{2n-1}}{(2n-1)(2n-1)!}
\]

Plot finite approximations of this power series solution on the TI-86 with `sum` and `seq`.
1. Select TOL from the MEM menu and set tol=1.
2. On the mode screen, set Radian angle mode and Param graphing mode.
3. In the equation editor, enter the parametric equations for the power series approximation as shown. Select sum and seq from the LIST OPS menu. Select ! from the MATH PROB menu.
   \[ xt1=t \]
   \[ yt1=sum(seq((L1)^(j+1)t^(2jN1)\frac{(2j-1)!}{((2j-1)(2j-1))!}j,1,10,1) \]
4. In the equation editor, enter the parametric equations as shown to plot the antiderivative of \((\sin x)/x\) and compare it with the plot of the power series approximation. (Select fnint from the CALC menu.)
   \[ xt2=t \]
   \[ yt2=fnint((\sin w)/w,w,0,t) \]
5. Select WIND from the GRAPH menu and set the window variable values as shown.
   \[ tMin=15 \]
   \[ xMin=15 \]
   \[ yMin=3 \]
   \[ tMax=15 \]
   \[ xMax=15 \]
   \[ yMax=3 \]
   \[ tStep=0.5 \]
   \[ xScl=1 \]
   \[ yScl=1 \]
6. Select FORMT from the GRAPH menu and set SimulG format.
7. Select GRAPH from the GRAPH menu to plot the parametric equations on the graph screen.
8. In the equation editor, modify yt1 to compute the first 16 terms of the power series by changing 10 to 16. Plot the equations again.
   In this example, the window variable tStep controls the plotting speed. Select WIND from the GRAPH menu and set tStep=1 and observe the difference in plotting speed and curve smoothness.

If necessary, select ALL from the equation editor menu to deselect all functions. Also, turn off all stat plots.

This example is set up in Param mode, which allows you to control the solution with tStep and increase plotting speed.

To clear the menus from the graph screen, press [CLEAR].
Reservoir Problem

On the TI-86, you can use parametric graphing animation to solve a problem.

Consider a water reservoir with a height of 2 meters. You must install a small valve on the side of the reservoir such that water spraying from the open valve hits the ground as far away from the reservoir as possible. At what height should you install the valve to maximize the length of the water stream when the valve is wide open?

Assume a full tank at time=0, no acceleration in the x direction, and no initial velocity in the y direction. Also, ignore valve-size and valve-type factors. Integrating the definition of acceleration in both the x and y directions twice yields the equations x=v_0t and y=h_0-(gt^2)/2. Solving Bernoulli’s equation for v_0 and substituting into v_0t results in this pair of parametric equations:

\[x(t) = t\sqrt{2g(2-h_0)}\]
\[y(t) = h_0 - \frac{(gt^2)}{2}\]

- t = time in seconds
- h_0 = height of the valve in meters
- g = the built-in acceleration of gravity constant

When you graph these equations on the TI-86, the y-axis (x=0) is the side of the reservoir where the valve is to be installed. The x-axis (y=0) is the ground. Each plotted parametric equation represents the water stream when the valve is at each of several heights.
In **Param** graphing mode, select $E(t)=$ from the GRAPH menu and enter the equations in the equation editor as shown. This pair of equations plots the path of the water stream when the valve is installed at a height of 0.5 meters.

\[
x_{t1} = t \sqrt{(2g(2 - 0.5))} \quad y_{t1} = 0.5 - \frac{(g t^2)}{2}
\]

1. Move the cursor to $x_{t2}$. Press [2nd] [RCL] [F2] ALPHA 1, and press ENTER to recall the contents of $x_{t1}$ into $x_{t2}$. For $x_{t2}$, change the valve height (which is 0.5) to 0.75 meters. Do the same with $y_{t1}$ and $y_{t2}$.

2. Repeat step 3 to create three more pairs of equations. Change the valve height to 1.0 meters for $x_{t3}$ and $y_{t3}$, 1.5 meters for $x_{t4}$ and $y_{t4}$, and 1.75 meters for $x_{t5}$ and $y_{t5}$.

3. Select **WIND** from the GRAPH menu and set the window variable values as shown.

   \[
   \begin{align*}
   t_{\text{Min}} &= 0 \\
   t_{\text{Max}} &= \sqrt{(4/g)} \\
   t_{\text{Step}} &= 0.01 \\
   x_{\text{Min}} &= 0 \\
   x_{\text{Max}} &= 2 \\
   x_{\text{Scl}} &= 0.5 \\
   y_{\text{Min}} &= 0 \\
   y_{\text{Max}} &= 2 \\
   y_{\text{Scl}} &= 0.5
   \end{align*}
   \]

4. Select **FORMT** from the GRAPH menu and set **SimulG** graph format.

5. Select **GRAPH** from the GRAPH menu to plot the trajectory of the water jets from the five specified heights.

Which height seems to create the longest water stream?
**Predator-Prey Model**

The growth rates of predator and prey populations, such as foxes and rabbits, depend upon the populations of both species. This initial-value problem is a form of the predator-prey model.

\[
\begin{align*}
F' &= F + 0.1F \cdot R \\
R' &= 3R - F \cdot R
\end{align*}
\]

- \( Q_1 \) = population of foxes (F)
- \( Q_2 \) = population of rabbits (R)
- \( Q_{11} \) = initial population of foxes (2)
- \( Q_{12} \) = initial population of rabbits (5)

Find the population of foxes and rabbits after 3 months (\( t=3 \)).

1. In DifEq graphing mode, select \( Q'1 = \) from the GRAPH menu and enter the functions and set graph styles in the equation editor as shown.

\[
\begin{align*}
\backslash Q'1 &= Q1 + 0.1Q1 \cdot Q2 \\
\backslash Q'2 &= 3Q2 - Q1 \cdot Q2
\end{align*}
\]

2. Select FORMT from the GRAPH menu and set FldOff field format.

3. Select WIND from the GRAPH menu and set the window variable values as shown.

   - \( tMin = 0 \)
   - \( xMin = -1 \)
   - \( yMin = 10 \)
   - \( tMax = 10 \)
   - \( xMax = 10 \)
   - \( yMax = 40 \)
   - \( tStep = \pi / 24 \)
   - \( xScl = 5 \)
   - \( yScl = 5 \)
   - \( \text{difTol} = .001 \)

4. Select INITC from the GRAPH menu and set the initial conditions as shown.

   - \( tMin = 0 \)
   - \( Q_{11} = 2 \)
   - \( Q_{12} = 5 \)
5. Select **GRAPH** from the GRAPH menu to plot the graph of the two populations over time.

6. To see the direction field of the phase-plane solution, select **FORMT** from the GRAPH screen, and then set **DirFld** field format.

7. Select **INITC** from the GRAPH menu and delete the values for \( Q_1 \) and \( Q_2 \).

8. Select **GRAPH** from the GRAPH menu to display the direction field of the phase-plane solution.

9. To see a family of specific phase-plane solutions on top of the direction field, select **INITC** from the GRAPH menu, and then enter lists for \( Q_1 \) and \( Q_2 \) as shown.

   \[
   Q_1 = \{2, 6, 7\} \quad Q_2 = \{6, 12, 18\}
   \]

10. Select **TRACE** from the GRAPH menu to display the graph with the trace cursor.

11. Press 3 to see how many foxes and how many rabbits are alive at \( t=3 \). (Round the values of \( Q_1 \) (foxes) and \( Q_2 \) (rabbits) to whole numbers.) How many foxes and rabbits are alive at \( t=6 \) at \( t=12 \)? On what value of \( Q_1 \) and \( Q_2 \) do the phase-plane orbits seem to converge? What is the significance of this value?
Chapter 19: Applications

Program: Sierpinski Triangle

This program creates a drawing of a widely known fractal, the Sierpinski Triangle, and stores the drawing to the picture variable TRI.

1. Select EDIT from the PRGM menu, enter SIERP at the Name= prompt, and then enter this program.

```
PROGRAM:SIERP
:FnOff :ClDrw:PlOff:AxesOff
Sets viewing window
:Begins
For group
:If/Then
:If N≤(1/3) Then
:.5X→X
:.5Y→Y
:End
 If/Then group
If/Then group
If/Then group
:If N>(1/3) and N≤(2/3) Then
:.5(1+X)→X
:.5(1+Y)→Y
:End
:If N>(2/3) Then
:.5(1+X)→X
:.5Y→Y
:End
End of For
End of For
Stores picture
:PtOn(X,Y)
:StPic TRI
```

2. On the home screen, select SIERP from the PRGM NAMES menu and press [ENTER] to run the program, which may run for several minutes before completion.

3. After you run the program, you can recall and display the picture by executing RcPic TRI.
A to Z Function and Instruction Reference

Quick-Find Locator........................................................... 262
Alphabetical Listing of Operations..................................... 266
Quick-Find Locator

This section lists the TI-86 functions and instructions in functional groups along with the page numbers where they are described in this chapter.

### Graphing

<table>
<thead>
<tr>
<th>Function</th>
<th>Page Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axes</td>
<td>271</td>
</tr>
<tr>
<td>AxesOff</td>
<td>271</td>
</tr>
<tr>
<td>AxesOn</td>
<td>271</td>
</tr>
<tr>
<td>Circ1</td>
<td>273</td>
</tr>
<tr>
<td>ClDrw</td>
<td>273</td>
</tr>
<tr>
<td>CoordOff</td>
<td>275</td>
</tr>
<tr>
<td>CoordOn</td>
<td>275</td>
</tr>
<tr>
<td>DiffEq</td>
<td>281</td>
</tr>
<tr>
<td>DirFid</td>
<td>282</td>
</tr>
<tr>
<td>DrawDot</td>
<td>285</td>
</tr>
<tr>
<td>DrawF</td>
<td>286</td>
</tr>
<tr>
<td>DrawLine</td>
<td>286</td>
</tr>
<tr>
<td>DrEqu</td>
<td>287</td>
</tr>
<tr>
<td>DrInv</td>
<td>287</td>
</tr>
<tr>
<td>Line</td>
<td>314</td>
</tr>
<tr>
<td>RectGC</td>
<td>344</td>
</tr>
<tr>
<td>ZFit</td>
<td>373</td>
</tr>
<tr>
<td>ZIn</td>
<td>373</td>
</tr>
<tr>
<td>ZInt</td>
<td>374</td>
</tr>
<tr>
<td>ZOut</td>
<td>375</td>
</tr>
<tr>
<td>ZPrev</td>
<td>375</td>
</tr>
<tr>
<td>ZSqr</td>
<td>376</td>
</tr>
<tr>
<td>ZStd</td>
<td>377</td>
</tr>
<tr>
<td>ZTrig</td>
<td>378</td>
</tr>
</tbody>
</table>

### Lists

<table>
<thead>
<tr>
<th>Function</th>
<th>Page Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>aug</td>
<td>270</td>
</tr>
<tr>
<td>cSum</td>
<td>278</td>
</tr>
<tr>
<td>Deltalst</td>
<td>279</td>
</tr>
<tr>
<td>dimL</td>
<td>282</td>
</tr>
<tr>
<td>dimL</td>
<td>282</td>
</tr>
<tr>
<td>dimL</td>
<td>282</td>
</tr>
<tr>
<td>dimL</td>
<td>282</td>
</tr>
<tr>
<td>dimL</td>
<td>282</td>
</tr>
<tr>
<td>List entry: { }</td>
<td>316</td>
</tr>
<tr>
<td>SetLEdit</td>
<td>351</td>
</tr>
<tr>
<td>Sorty</td>
<td>359</td>
</tr>
<tr>
<td>Sortx</td>
<td>359</td>
</tr>
<tr>
<td>Sortx</td>
<td>359</td>
</tr>
<tr>
<td>Mathematics, Algebra, and Calculus</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>abs</td>
<td>Division: /</td>
</tr>
<tr>
<td>Addition: +</td>
<td>DMS entry: °</td>
</tr>
<tr>
<td>and</td>
<td>&gt;DMS</td>
</tr>
<tr>
<td>angle</td>
<td>dxDer1</td>
</tr>
<tr>
<td>Ans</td>
<td>dxNDer</td>
</tr>
<tr>
<td>arc</td>
<td>e^</td>
</tr>
<tr>
<td>Assignment: =</td>
<td>Eng</td>
</tr>
<tr>
<td>b</td>
<td>EcpS(</td>
</tr>
<tr>
<td>Bin</td>
<td>Equal: =</td>
</tr>
<tr>
<td>Bin</td>
<td>Equal to: ==</td>
</tr>
<tr>
<td>ClrEnt</td>
<td>Euler</td>
</tr>
<tr>
<td>CItbl</td>
<td>eval</td>
</tr>
<tr>
<td>conj</td>
<td>evalF(</td>
</tr>
<tr>
<td>cos</td>
<td>Exponent: E</td>
</tr>
<tr>
<td>cos^{-1}</td>
<td>Factorial: !</td>
</tr>
<tr>
<td>cosh</td>
<td>Fix</td>
</tr>
<tr>
<td>cosh^{-1}</td>
<td>Float</td>
</tr>
<tr>
<td>d</td>
<td>fMax(</td>
</tr>
<tr>
<td>Dec</td>
<td>fMin(</td>
</tr>
<tr>
<td>Dec</td>
<td>fnInt(</td>
</tr>
<tr>
<td>Degree</td>
<td>fPart</td>
</tr>
<tr>
<td>Degree entry: °</td>
<td>Frac</td>
</tr>
<tr>
<td>der1</td>
<td>gcd</td>
</tr>
<tr>
<td>der2</td>
<td>Greater than: &gt;</td>
</tr>
<tr>
<td>Greater than or equal to: ≥</td>
<td>301</td>
</tr>
<tr>
<td>h</td>
<td>Integrals: ∫</td>
</tr>
<tr>
<td>Hex</td>
<td>int</td>
</tr>
<tr>
<td>inter</td>
<td>Inverse: ^{-1}</td>
</tr>
<tr>
<td>lcm</td>
<td>Less than: &lt;</td>
</tr>
<tr>
<td>Less than or equal to: ≤</td>
<td>312</td>
</tr>
<tr>
<td>ln</td>
<td>log</td>
</tr>
<tr>
<td>max</td>
<td>log</td>
</tr>
<tr>
<td>min</td>
<td>log</td>
</tr>
<tr>
<td>mod</td>
<td>log</td>
</tr>
<tr>
<td>Multiplication: *</td>
<td>log</td>
</tr>
<tr>
<td>nCr</td>
<td>nDer</td>
</tr>
<tr>
<td>Negation: ~</td>
<td>323</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal</td>
</tr>
<tr>
<td>not</td>
<td>Normal</td>
</tr>
<tr>
<td>not equal to: ≠</td>
<td>326</td>
</tr>
<tr>
<td>nPr</td>
<td>nPr</td>
</tr>
<tr>
<td>or</td>
<td>nPr</td>
</tr>
<tr>
<td>Oct</td>
<td>Oct</td>
</tr>
<tr>
<td>Oct</td>
<td>Oct</td>
</tr>
<tr>
<td>polar</td>
<td>Oct</td>
</tr>
<tr>
<td>or</td>
<td>Oct</td>
</tr>
<tr>
<td>Percent: %</td>
<td>334</td>
</tr>
<tr>
<td>Percent: %</td>
<td>334</td>
</tr>
<tr>
<td>pEval</td>
<td>Pol</td>
</tr>
<tr>
<td>Polar complex: ϕ</td>
<td>Pol</td>
</tr>
<tr>
<td>Polar complex: ϕ</td>
<td>Pol</td>
</tr>
<tr>
<td>poly</td>
<td>Power: ^</td>
</tr>
<tr>
<td>Power: ^</td>
<td>337</td>
</tr>
<tr>
<td>Power of 10: 10^x</td>
<td>337</td>
</tr>
<tr>
<td>Power of 10: 10^x</td>
<td>337</td>
</tr>
<tr>
<td>Power of 10: 10^x</td>
<td>337</td>
</tr>
<tr>
<td>Rad</td>
<td>Radian</td>
</tr>
<tr>
<td>Radian</td>
<td>Radian</td>
</tr>
<tr>
<td>Radian</td>
<td>Radian</td>
</tr>
<tr>
<td>real</td>
<td>Radian</td>
</tr>
<tr>
<td>Radian entry: r</td>
<td>real</td>
</tr>
<tr>
<td>real</td>
<td>Rec</td>
</tr>
<tr>
<td>real</td>
<td>Rec</td>
</tr>
<tr>
<td>real</td>
<td>Rec</td>
</tr>
<tr>
<td>RK</td>
<td>real</td>
</tr>
<tr>
<td>Root: √</td>
<td>346</td>
</tr>
<tr>
<td>Root: √</td>
<td>346</td>
</tr>
<tr>
<td>rotL</td>
<td>rotL</td>
</tr>
<tr>
<td>rotL</td>
<td>rotL</td>
</tr>
<tr>
<td>rotL</td>
<td>rotL</td>
</tr>
<tr>
<td>rotR</td>
<td>rotR</td>
</tr>
<tr>
<td>rotR</td>
<td>rotR</td>
</tr>
<tr>
<td>rotR</td>
<td>rotR</td>
</tr>
</tbody>
</table>
### Matrices

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aug</td>
<td>270</td>
</tr>
<tr>
<td>cnorm</td>
<td>273</td>
</tr>
<tr>
<td>cond</td>
<td>274</td>
</tr>
<tr>
<td>det</td>
<td>281</td>
</tr>
<tr>
<td>dim</td>
<td>281</td>
</tr>
<tr>
<td>➤dim</td>
<td>281</td>
</tr>
<tr>
<td>LU</td>
<td>318</td>
</tr>
<tr>
<td>rAdd</td>
<td>340</td>
</tr>
<tr>
<td>rSwap</td>
<td>348</td>
</tr>
<tr>
<td>Transpose</td>
<td>367</td>
</tr>
</tbody>
</table>

### Programming

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asm</td>
<td>269</td>
</tr>
<tr>
<td>AsmComp</td>
<td>270</td>
</tr>
<tr>
<td>AsmPrgm</td>
<td>270</td>
</tr>
<tr>
<td>CILCD</td>
<td>273</td>
</tr>
<tr>
<td>DelVar</td>
<td>280</td>
</tr>
<tr>
<td>Disp</td>
<td>283</td>
</tr>
<tr>
<td>DispG</td>
<td>283</td>
</tr>
<tr>
<td>DispT</td>
<td>284</td>
</tr>
<tr>
<td>DispG</td>
<td>284</td>
</tr>
<tr>
<td>DS&lt;</td>
<td>288</td>
</tr>
<tr>
<td>Else</td>
<td>290</td>
</tr>
<tr>
<td>End</td>
<td>290</td>
</tr>
<tr>
<td>Equal;</td>
<td>290</td>
</tr>
<tr>
<td>Equal to</td>
<td>291</td>
</tr>
<tr>
<td>For</td>
<td>297</td>
</tr>
<tr>
<td>Get</td>
<td>299</td>
</tr>
<tr>
<td>Input</td>
<td>307</td>
</tr>
<tr>
<td>Prompt</td>
<td>338</td>
</tr>
<tr>
<td>Repeat</td>
<td>345</td>
</tr>
<tr>
<td>Return</td>
<td>345</td>
</tr>
<tr>
<td>Send</td>
<td>350</td>
</tr>
<tr>
<td>Stop</td>
<td>362</td>
</tr>
<tr>
<td>Then</td>
<td>366</td>
</tr>
<tr>
<td>While</td>
<td>369</td>
</tr>
</tbody>
</table>

### Statistics

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box</td>
<td>272</td>
</tr>
<tr>
<td>ExpR</td>
<td>293</td>
</tr>
<tr>
<td>fcstx</td>
<td>294</td>
</tr>
<tr>
<td>fcsty</td>
<td>294</td>
</tr>
<tr>
<td>Hist</td>
<td>303</td>
</tr>
<tr>
<td>LgstR</td>
<td>313</td>
</tr>
<tr>
<td>LinR</td>
<td>315</td>
</tr>
<tr>
<td>LnR</td>
<td>317</td>
</tr>
<tr>
<td>MBox</td>
<td>319</td>
</tr>
<tr>
<td>OneVar</td>
<td>327</td>
</tr>
<tr>
<td>P2Reg</td>
<td>330</td>
</tr>
<tr>
<td>P3Reg</td>
<td>331</td>
</tr>
<tr>
<td>P4Reg</td>
<td>332</td>
</tr>
<tr>
<td>Plot1</td>
<td>335</td>
</tr>
<tr>
<td>Plot2</td>
<td>335</td>
</tr>
<tr>
<td>Plot3</td>
<td>335</td>
</tr>
<tr>
<td>PlOn</td>
<td>334</td>
</tr>
<tr>
<td>rand</td>
<td>341</td>
</tr>
<tr>
<td>randBin</td>
<td>341</td>
</tr>
<tr>
<td>randInt</td>
<td>342</td>
</tr>
<tr>
<td>rAdd</td>
<td>340</td>
</tr>
<tr>
<td>rNorm</td>
<td>342</td>
</tr>
<tr>
<td>rSwap</td>
<td>348</td>
</tr>
<tr>
<td>SinR</td>
<td>357</td>
</tr>
<tr>
<td>Sortx</td>
<td>359</td>
</tr>
<tr>
<td>Sorty</td>
<td>359</td>
</tr>
<tr>
<td>StReg</td>
<td>362</td>
</tr>
<tr>
<td>TwoVar</td>
<td>368</td>
</tr>
<tr>
<td>xyline</td>
<td>370</td>
</tr>
</tbody>
</table>

---

Chapter 20: A to Z Function and Instruction Reference
Strings

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concatenation: <code>+</code></td>
<td>274</td>
</tr>
<tr>
<td>Length</td>
<td>316</td>
</tr>
<tr>
<td>String</td>
<td>361</td>
</tr>
<tr>
<td>String entry: <code>&quot;</code></td>
<td>363</td>
</tr>
<tr>
<td>Substring</td>
<td>363</td>
</tr>
</tbody>
</table>

Vectors

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNorm</td>
<td>273</td>
</tr>
<tr>
<td>Dim</td>
<td>281</td>
</tr>
<tr>
<td>Dimension</td>
<td>281</td>
</tr>
<tr>
<td>Norm</td>
<td>323</td>
</tr>
<tr>
<td>SphereV</td>
<td>360</td>
</tr>
<tr>
<td>RectV</td>
<td>344</td>
</tr>
<tr>
<td>UnitV</td>
<td>368</td>
</tr>
<tr>
<td>Vector</td>
<td>369</td>
</tr>
<tr>
<td>Vector entry: <code>[ ]</code></td>
<td>369</td>
</tr>
</tbody>
</table>
Alphabetical Listing of Operations

All the operations in this section are included in the CATALOG. Non-alphabetic operations (such as `+`, `1`, and `>` are listed at the end of the CATALOG. In this A to Z Reference, however, these operations are listed under their alphabetic equivalent (such as addition, factorial, and greater than).

You always can use the CATALOG to select an operation and paste it to the home screen or to a command line in the program editor. You also can use the specific keystrokes, menus, or screens listed in this section.

† Indicates menus or screens that paste the operation’s name only if you are in the program editor. In most cases, you can use these menus or screens from the home screen to perform the operation interactively, without pasting the name.

‡ Indicates menus or screens that are valid only from the program editor’s main menu. From the home screen, you cannot use these menus or screens to select an operation.

The syntax for some operations uses brackets `[ ]` to indicate optional arguments. If you use an optional argument, do not enter the brackets.
**abs**

**MATH NUM menu**

**CPLX menu**

**MATRX CPLX menu**

**VECTR CPLX menu**

**abs**

- **realNumber** or **abs(realExpression)**
  - Returns the absolute value of realNumber or realExpression.
  
  ```text
  abs \( \text{realNumber} \) or \( \text{abs}(\text{realExpression}) \)
  ```

- **abs(complexNumber)**
  - Returns the magnitude (modulus) of complexNumber.
    - **abs(real, imaginary)** returns \( \sqrt{\text{real}^2 + \text{imaginary}^2} \).
    - **abs(magnitude, angle)** returns magnitude.
  
  ```text
  abs (complexNumber)
  ```

**Abs**

- **list**
- **matrix**
- **vector**
  - Returns a list, matrix, or vector in which each element is the absolute value of the corresponding real or complex element in the argument.
  
  ```text
  abs \{1.25, -5.67\} \rightarrow \{1.25, 5.67\}
  abs [(3, 4), (3, 4)] \rightarrow [(5, 3)]
  ```

**Addition:**

- **numberA + numberB**
  - Returns the sum of two real or complex numbers.
  
  ```text
  numberA + numberB
  ```

- **number + list**
  - Returns a list in which a real or complex number is added to each element of a real or complex list.
  
  ```text
  number + list
  ```
listA + listB
matrixA + matrixB
vectorA + vectorB

Returns a list, matrix, or vector that is the sum of the corresponding real or complex elements in the arguments. The two arguments must have the same dimension.

For information about adding two strings, refer to Concatenation on page 274.

and

BASE BOOL menu

integerA and integerB

Compares two real integers bit by bit. Internally, both integers are converted to binary. When corresponding bits are compared, the result is 1 if both bits are 1; otherwise, the result is 0. The returned value is the sum of the bit results.

For example, 78 and 23 = 6.

78 = 1001110b
23 = 0010111b
00001110b = 6

You can enter real numbers instead of integers, but they are truncated automatically before the comparison.
### Chapter 20: A to Z Function and Instruction Reference

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **angle** | CPLX menu, MATRX CPLX menu, VECTR CPLX menu  
**angle (complexNumber)**  
Returns the polar angle of complexNumber, adjusted by +\( \pi \) in the 2nd quadrant or -\( \pi \) in the 3rd quadrant. The polar angle of a real number is always 0.  
**angle (real, imaginary)** returns \( \tan^{-1}(\text{imaginary/real}) \).  
**angle (magnitude, angle)** returns angle, \( -\pi < \text{angle} \leq \pi \). |
| **angle complexList** |  
Returns a list, matrix, or vector in which each element is the polar angle of the corresponding element in the argument.  
If complexVector has only two real elements, the returned value is a real number, not a vector. |
| **angle complexMatrix** |
| **angle complexVector** |
| **Ans** |  
\( \text{Ans} \)  
Returns the last answer. |
| **arc** | CALC menu  
**arc (expression, variable, start, end)**  
Returns the length along expression with respect to variable, from variable = start to variable = end. |
| **Asm** | CATALOG  
**Asm (assemblyProgramName)**  
Executes an assembly language program. For more information, refer to Chapter 16. |

**Example**

In Radian angle mode and PolarC complex number mode:

- \( \text{angle (3,4)} \) \( \text{ENTER} \) \( 0.927295218002 \)
- \( \text{angle (3,2)} \) \( \text{ENTER} \) \( 2 \)
- \( 6\pi/3 \) \( \text{A ENTER} \) \( 6\pi/1.0471975512 \)
- \( \text{angle A ENTER} \) \( 1.0471975512 \)
- \( \text{angle {{3,4},{3,2}}} \) \( \text{ENTER} \) \( \{0.927295218002 \, 2\} \)

\[ \text{Ans} \]

\[ \text{Ans} \]

\[ \text{Ans} \]

\[ \text{Ans} \]

\[ \text{Ans} \]
AsmComp

**Catalog**

Compiles an assembly language program written in ASCII and stores the hex version. The compiled hex version, which uses about half the storage space of the ASCII version, cannot be edited.

When you execute the ASCII version, the TI-86 compiles it each time. To speed up execution, use `AsmComp` to compile the ASCII version once and then execute the hex version each time you want to run the program.

AsmPrgm

**Catalog**

Must be used as the first line of an assembly language program.

**Assignment:**

Stores expression to `equationVariable`, without evaluating `expression`. (If you use `{STO} variables to store an expression to a variable, the expression is evaluated and then the result is stored.)

aug

**List Ops Menu**

Returns a list consisting of `listB` appended (concatenated) to the end of `listA`. The lists can be real or complex.

```plaintext
aug({1, L3, 2}, {5, 4})
```

```
{1, -3, 2, 5, 4}
```
aug(matrixA, matrixB)

Returns a matrix consisting of matrixB appended as new columns to the end of matrixA. The matrices can be real or complex. Both must have the same number of rows.

```
[[1,2,3][4,5,6]] → MATA ENTER
[[1 2 3]
[4 5 6]]
[[7,8][9,10]] → MATB ENTER
[[7 8 ]
[9 10]]
```

```
aug(MATA, MATB) ENTER
[1 2 3 7 8 ]
[4 5 6 9 10]
```

Aug(matrix, vector)

Returns a matrix consisting of vector appended as a new column to the end of matrix. The arguments can be real or complex. The number of rows in matrix must equal the number of elements in vector.

```
[[1,2,3]
[4,5,6]]
```

```
MATA b
[[1 2 3]
[4 5 6]]
```

```
MATB b
[[7 8 ]
[9 10]]
```

```
aug(MATA, MATB) ENTER
[1 2 3 7 8 ]
[4 5 6 9 10]
```

Axes(† GRAPH VARS menu

Axes(xAxisVariable, yAxisVariable)

Specifies the variables plotted for the axes in DifEq graphing mode. The xAxisVariable or yAxisVariable can be t, Q1 through Q9, or Q'1 through Q'9.

```
Axes(Q1, Q2) ENTER
Done
```

AxesOff(† graph format screen

Turns off the graph axes.

AxesOn(† graph format screen

Turns on the graph axes.

b

BASE TYPE menu

integer b

Designates a real integer as binary, regardless of the number base mode setting.

```
in Dec number base mode:

10b ENTER 2
10b+10 ENTER 12
```
Chapter 20: A to Z Function and Instruction Reference

Bin
† mode screen
Sets binary number base mode. Results are displayed with the b suffix. In any number base mode, you can designate an appropriate value as binary, decimal, hexadecimal, or octal by using the b, d, h, or o designator, respectively, from the BASE TYPE menu.

Bin
BASE CONV menu

number→Bin
list→Bin
matrix→Bin
vector→Bin
Returns the binary equivalent of the real or complex argument.

In Bin number base mode:
10+fh+10e+10d ENTER 100011b

In Dec number base mode:
2→B ENTER 16
Ans→Bin ENTER 10000b

{1,2,3,4}→Bin ENTER {1b 10b 11b 100b}

Box
† STAT DRAW menu

Box xList,frequencyList
Draws a box plot on the current graph, using the real data in xList and the frequencies in frequencyList.

Box xList
Uses frequencies of 1.

Box
Uses the data in built-in variables xStat and fStat. These variables must contain valid data of the same dimension; otherwise, an error occurs.

Starting with a ZStd graph screen:
{1,2,3,4,5,9}→XL ENTER
{1,1,1,4,1,1}→FL ENTER
0→xMin:0→yMin ENTER
0
Box XL,FL ENTER
**Circl(**

**† GRAPH DRAW menu**

Draws a circle with center \((x, y)\) and \(radius\) on the current graph.

**Starting with a ZStd graph screen:**

\[ZSqr: \text{Circl}(1,2,7) \quad \text{ENTER}\]

---

**ClDrw**

**† GRAPH DRAW menu**

Clears all drawn elements from the current graph.

---

**CILCD**

**‡ program editor**

Clears the home screen (LCD).

---

**ClrEnt**

**MEM menu**

Clears the contents of the Last Entry storage area.

---

**ClTbl**

**‡ program editor**

Clears all values from the current table if \(\text{Indpt: Ask (IAsk, page 304)}\) is set.

---

**cnorm**

**MATRX MATH menu**

Returns the column norm of a real or complex \(matrix\).

For each column, \(\text{cnorm}\) sums the absolute values (magnitudes of complex elements) of the elements in that column and returns the largest of those column sums.

\[
\begin{bmatrix}1 & 2 & 3 \end{bmatrix} \begin{bmatrix}4 & 5 & 6 \end{bmatrix} \rightarrow \text{MAT (ENTER)}
\]

\[
\begin{bmatrix}1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{cnorm (ENTER)}
\]

\(9\)
Chapter 20: A to Z Function and Instruction Reference

**cnorm vector**

Returns the sum of the absolute values of the real or complex elements in `vector`.

```
[-1,2,-3]→VEC ENTER  
[-1 2 -3]
```

**Concatenation:**

```
stringA + stringB
```

Returns a string consisting of `stringB` appended (concatenated) to the end of `stringA`.

```
"your name:" → STR ENTER
your name:
"Enter " + STR ENTER
Enter your name:
```

**cond**

MATRX MATH menu

Returns the condition number of a real or complex `squareMatrix`, which is calculated as:

```
cnorm squareMatrix * cnorm squareMatrix⁻¹
```

The condition number indicates how well-behaved `squareMatrix` is expected to be for certain matrix functions, particularly inverse. For a well-behaved matrix, the condition number is close to 1.

```
log(cond squareMatrix) indicates the number of digits that may be lost due to round-off errors in computing the inverse.
```

For a matrix with no inverse, `cond` returns an error.

```
[[1,0,0][0,1,0][0,0,1]]→MAT1
[[1 0 0] 
[0 1 0] 
[0 0 1]]
```

```
cond MAT1 ENTER  
1
```

```
[[1,2,3][4,5,6][7,8,9]]→MAT2
[[1 2 3] 
[4 5 6] 
[7 8 9]]
```

```
cond MAT2 ENTER  
1.8214
```

```
log (Ans) ENTER  
14.2552725051
```
conj
CPLX menu
MATRX CPLX menu
VECTR CPLX menu
conj (complexNumber)
Returns the complex conjugate of complexNumber.
In RectC mode, conj (real,imaginary) returns 
(real;-imaginary).
In PolarC mode, conj (magnitude;angle) returns 
magnitude;angle, -π < angle ≤ π.
conj complexList
conj complexMatrix
conj complexVector
Returns a complex list, matrix, or vector in which each 
element is the complex conjugate of the original.

CoordOff
† graph format screen
CoordOff
Turns off cursor coordinates so they are not displayed 
at the bottom of a graph.

CoordOn
† graph format screen
CoordOn
Displays cursor coordinates at the bottom of a graph.
The squareMatrix cannot have repeated eigenvalues.

\[ \text{cos} \angle \text{ or } \text{cos} (\text{expression}) \]

Returns the cosine of \( \angle \) or \( \text{expression} \), which can be real or complex.

An angle is interpreted as degrees or radians according to the current angle mode. In any angle mode, you can designate an angle as degrees or radians by using the \(^{\circ}\) or \( \pi \) designator, respectively, from the MATH ANGLE menu.

\[ \text{cos} \text{ list} \]

Returns a list in which each element is the cosine of the corresponding element in \( \text{list} \).

\[ \text{cos} \text{ squareMatrix} \]

Returns a square matrix that is the matrix cosine of \( \text{squareMatrix} \). The matrix cosine corresponds to the result calculated using power series or Cayley-Hamilton Theorem techniques. This is not the same as simply calculating the cosine of each element.

\[ \text{cos}^{-1} \text{ number or } \text{cos}^{-1} (\text{expression}) \]

Returns the arccosine of \( \text{number} \) or \( \text{expression} \), which can be real or complex.

\[ \text{cos}^{-1} \text{ list} \]

Returns a list in which each element is the arccosine of the corresponding element in \( \text{list} \).
### Chapter 20: A to Z Function and Instruction Reference

#### cosh

**MATH HYP menu**

\[ \text{cosh} \]  

- **cosh** \( \text{number} \) or **cosh** (\( \text{expression} \))
  - Returns the hyperbolic cosine of \( \text{number} \) or \( \text{expression} \), which can be real or complex.
  - **Example:**  
    
    \[
    \text{cosh 1.2} \quad \text{(ENTER)} \quad 1.8106556732
    \]

- **cosh** \( \text{list} \)
  - Returns a list in which each element is the hyperbolic cosine of the corresponding element in \( \text{list} \).
  - **Example:**  
    
    \[
    \text{cosh \{0,1.2\}} \quad \text{(ENTER)} \quad \{1 1.8106556732\}
    \]

#### cosh\(^{-1}\)

**MATH HYP menu**

\[ \text{cosh}^{-1} \]  

- **cosh\(^{-1}\)** \( \text{number} \) or **cos**\(^{-1}\) (\( \text{expression} \))
  - Returns the inverse hyperbolic cosine of \( \text{number} \) or \( \text{expression} \), which can be real or complex.
  - **Example:**  
    
    \[
    \text{cosh\(^{-1}\) 1} \quad \text{(ENTER)} \quad 0
    \]

- **cosh\(^{-1}\)** \( \text{list} \)
  - Returns a list in which each element is the inverse hyperbolic cosine of the corresponding element in \( \text{list} \).
  - **Example:**  
    
    \[
    \text{cosh\(^{-1}\) \{1,2,1,3\}} \quad \text{(ENTER)} \quad \{0 1.37285914424 1.7\}
    \]

#### cross()

**VECTR MATH menu**

\[ \text{cross} \]  

- **cross**(\( \text{vectorA}, \text{vectorB} \))
  - Returns the cross product of two real or complex vectors, where:
    
    \[ \text{cross([a,b,c],[d,e,f]) = [bf-ce, cd-af, ae-bd]} \]
  - Both vectors must have the same dimension (either 2 or 3 elements). A 2-D vector is treated as a 3-D vector with 0 as the third element.
  - **Example:**  
    
    \[
    \text{cross([1,2,3],[4,5,6])} \quad \text{(ENTER)} \quad [-3 6 -3]
    \]

- **Example:**  
  
  \[
  \text{cross([1,2],[3,4])} \quad \text{(ENTER)} \quad \{0 0 -2\}
  \]
### cSum

**LIST OPS menu**

**cSum(list)**

Returns a list of the cumulative sums of the real or complex elements in `list`, starting with the first element.

- `cSum({1,2,3,4})`<br>  `b` `{1,3,6,10}`<br>  `cSum({10,20,30})`<br>  `L1` `{10,20,30}`<br>  `cSum(L1)`<br>  `{10,30,60}`

### Cyl

**VECTR OPS menu**

**Cyl(vector)**

Displays a 2- or 3-element real vector result in cylindrical form, `[r<0 z]`, even if the display mode is not set for cylindrical (`CylV`).

- `[-2,0]`<br>  `Cyl`<br>  `{2.14159265359,0}`<br>  `[-2,0,1]`<br>  `Cyl`<br>  `{2.14159265359,1}`

### CylV

**† mode screen**

Sets cylindrical vector coordinate mode ([r<0 z]).

- `In CylV vector coordinate mode and Radian angle mode: [3,4,5]`<br>  `{3.92729521802,5}`

### d

**BASE TYPE menu**

Designates a real number as decimal, regardless of the number base mode setting.

- `In Bin number base mode: 10d`<br>  ` ENTER `<br>  `1010b`<br>  `10d+10`<br>  ` ENTER `<br>  `1100b`

### Dec

**† mode screen**

Sets decimal number base mode. In any number base mode, you can designate an appropriate value as binary, decimal, hexadecimal, or octal by using the `b`, `d`, `h`, or `c` designator, respectively, from the BASE TYPE menu.

- `In Dec number base mode: 10+10b+Fh+10e`<br>  ` ENTER `<br>  `35`
Chapter 20: A to Z Function and Instruction Reference

Dec
BASE CONV menu

number→Dec
list→Dec
matrix→Dec
vector→Dec

Returns the decimal equivalent of the real or complex argument.

Degree
† mode screen

Degree
Sets degree angle mode.

In Hex number base mode:
2→F [ENTER] 13h
Ans→Dec [ENTER] 30d
[A,B,C,D,E]→Dec [ENTER] [10d 11d 12d 13d 14d]

In Degree angle mode:
sin 90 [ENTER] 1
sin (π/2) [ENTER] .027412133592

cos 90 [ENTER] -.44807366129

cos 90° [ENTER] 0

cos {45,90,180}° [ENTER] {.707106781187 0 -1}

Degree entry: °
MATH ANGLE menu

number° or (expression)°

Designates a real number or expression as degrees, regardless of the angle mode setting.

list°

Designates each element in list as degrees.

Deltalst(
LIST OPS menu
(Deltal shows on menu)

Deltalst(list)

Returns a list containing the differences between consecutive real or complex elements in list. This subtracts the first element in list from the second element, the second from the third, and so on. The resulting list is always one element shorter than list.
### DelVar( variable )

Deletion of the specified user-created variable from memory. You cannot use DelVar( to delete a program variable or built-in variable.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DelVar(A)</td>
<td>2⇒A (ENTER)</td>
</tr>
<tr>
<td>(A+2)² (ENTER)</td>
<td>2</td>
</tr>
<tr>
<td>DelVar(A)</td>
<td>16</td>
</tr>
<tr>
<td>(A+2)² (ENTER)</td>
<td>Done</td>
</tr>
</tbody>
</table>

### der1( expression, variable, value )

Returns the first derivative of expression with respect to variable at the real or complex value.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der1(x^3,x,5)</td>
<td>75</td>
</tr>
</tbody>
</table>

### der1( expression, variable )

Uses the current value of variable.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der1(x^3,x)</td>
<td>27</td>
</tr>
</tbody>
</table>

### der1( expression, variable, list )

Returns a list containing the first derivatives at the values specified by the elements in list.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der1(x^3,x,[5,3])</td>
<td>{75 27}</td>
</tr>
</tbody>
</table>

### der2( expression, variable, value )

Returns the second derivative of expression with respect to variable at the real or complex value.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der2(x^3,x,5)</td>
<td>30</td>
</tr>
</tbody>
</table>

### der2( expression, variable )

Uses the current value of variable.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der2(x^3,x)</td>
<td>18</td>
</tr>
</tbody>
</table>

### der2( expression, variable, list )

Returns a list containing the second derivatives at the values specified by the elements in list.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>der2(x^3,x,[5,3])</td>
<td>{30 18}</td>
</tr>
</tbody>
</table>
**det**

MATRX MATH menu

**det squareMatrix**

Returns the determinant of *squareMatrix*. The result is real for a real matrix, complex for a complex matrix.

\[
\begin{bmatrix}
1 & 2 \\
3 & 4 \\
\end{bmatrix}
\]

\[ \text{det MAT} \begin{bmatrix}
1 & 2 \\
3 & 4 \\
\end{bmatrix} \]

-2

**DifEq**

† mode screen

Sets differential equation graphing mode.

**dim**

MATRX OPS menu

**dim vector**

Returns the length (number of elements) of a real or complex vector.

\[
\begin{bmatrix}
2 & 7 & 0 \\
8 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[ \text{dim MAT} \begin{bmatrix}
2 & 7 & 0 \\
8 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix} \{3,3\} \]

\[ \text{MAT} \begin{bmatrix}
2 & 7 & 0 \\
8 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix} \]

**dim**

VECTR OPS menu

\[
\begin{bmatrix}
L8,0,1 \\
\end{bmatrix}
\]

\[ \text{dim} \begin{bmatrix}
L8,0,1 \\
\end{bmatrix} \]

3

\[ \text{dim} \begin{bmatrix}
2,7,1 \\
-8,0,1 \\
\end{bmatrix} \]

\[ \text{dim} \begin{bmatrix}
2,7,1 \\
-8,0,1 \\
\end{bmatrix} \]

\[ \text{dim} \begin{bmatrix}
2,7,1 \\
-8,0,1 \\
\end{bmatrix} \]

3

\[ \text{dim} \begin{bmatrix}
2,7 \\
8,0 \\
\end{bmatrix} \]

\[ \text{dim} \begin{bmatrix}
2,7 \\
8,0 \\
\end{bmatrix} \]

\[ \text{dim} \begin{bmatrix}
2,7 \\
8,0 \\
\end{bmatrix} \]

\[ \text{dim} \begin{bmatrix}
2,7 \\
8,0 \\
\end{bmatrix} \]

3

**⇒dim**

STOP, then MATRX OPS menu

STOP, then VECTR OPS menu

If *matrixName* does not exist, creates a new matrix with the specified dimensions and fills it with zeros.

If *matrixName* exists, redimensions that matrix to the specified dimensions. Existing elements within the new dimensions are not changed; elements outside the new dimensions are deleted. If additional elements are created, they are filled with zeros.
#ofElements>dim vectorName

If vectorName does not exist, creates a new vector with the specified #ofElements and fills it with zeros.

If vectorName exists, redimensions that vector to the specified #ofElements. Existing elements within the new dimension are not changed; elements outside the new dimension are deleted. If additional elements are created, they are filled with zeros.

```
#ofElements>dim vectorName
```

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 menu

```
#ofElements>dim vectorName
```

```
dimL list
Returns the length (number of elements) of a real or complex list.
```

dimL ENTER 4
1/dimL ENTER .25

```
#ofElements>dim listName
```

```
#ofElements>dim listName
```

```
dimL list
```

```
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

† graph format screen (scroll down to second screen)

In DiffEq graphing mode, turns on direction fields. To turn off direction and slope fields, use FldOff.
### Disp

| **Disp** | **Disp** valueA,valueB,valueC, ... | 10>[X] ENTER  
Disp x^3+3 x-6 ENTER | 10 | Done |

Displays each value. The values can include strings and variable names.

Disp

Displays the home screen.

#### DispG

<table>
<thead>
<tr>
<th><strong>DispG</strong></th>
<th><strong>DispG</strong></th>
<th>Program segment in <strong>Func</strong> graphing mode:</th>
</tr>
</thead>
<tbody>
<tr>
<td>dispG</td>
<td>dispG</td>
<td>...:y1=4cos x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:-10&gt;xMin:10&gt;xMax</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:-5&gt;yMin:5&gt;yMax</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DispG</td>
</tr>
</tbody>
</table>

Function names are case-sensitive. Use y1, not Y1.

To select from a list of window variable names, press [2nd] [CATLG-VARS] [MATH] [MORE] [F5].

---

![Graph](image-url)
### DispT

**DispT**

Displays the table.

Function names are case-sensitive. Use \( y_1 \), not \( Y_1 \).

<table>
<thead>
<tr>
<th>Program segment in <strong>Func</strong> graphing mode:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ y_1 = 4 \cos x ]</td>
</tr>
<tr>
<td>[ \text{DispT} ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( x = 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.14</td>
</tr>
<tr>
<td>2.09</td>
</tr>
<tr>
<td>1.25</td>
</tr>
<tr>
<td>-0.25</td>
</tr>
<tr>
<td>-1.25</td>
</tr>
<tr>
<td>-2.09</td>
</tr>
<tr>
<td>-3.14</td>
</tr>
</tbody>
</table>

### Division: \(/

\[ \text{numberA/numberB or (expressionA)/(expressionB)} \]

Returns one argument divided by another. The arguments can be real or complex.

<table>
<thead>
<tr>
<th>( \frac{98}{4} )</th>
<th>( \frac{98}{4 \times 1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ENTER} )</td>
<td>( \text{ENTER} )</td>
</tr>
<tr>
<td>-24.5</td>
<td>-8.1666666667</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \text{number/list or (expression)/list} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list in which each element is ( \text{number} ) or ( \text{expression} ) divided by the corresponding element in ( \text{list} ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \frac{100}{(10,25,2)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ENTER} )</td>
</tr>
</tbody>
</table>
| \( \{10 \ 4 \ 50\} \)

<table>
<thead>
<tr>
<th>( \text{list/number or list/(expression)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list or vector in which each element of ( \text{list} ) or ( \text{vector} ) is divided by ( \text{number} ) or ( \text{expression} ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \frac{{120,92,8}}{4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ENTER} )</td>
</tr>
</tbody>
</table>
| \( \{30 \ 23 \ 2\} \)

<table>
<thead>
<tr>
<th>( \text{vector/number or vector/(expression)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list or vector in which each element of ( \text{list} ) or ( \text{vector} ) is divided by ( \text{number} ) or ( \text{expression} ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \frac{[8,1,(5,2)]}{2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ENTER} )</td>
</tr>
</tbody>
</table>
| \( [(4,0),(5,0),(2,5),\ldots] \)

<table>
<thead>
<tr>
<th>( \text{listA/listB} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list in which each element of ( \text{listA} ) is divided by the corresponding element of ( \text{listB} ). The lists must have the same dimension.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \frac{{1,2,3}}{{4,5,6}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ENTER} )</td>
</tr>
</tbody>
</table>
| \( \{0.25 \ 0.4 \ 0.5\} \)
DMS entry: 

MATH ANGLE menu

In a trig calculation, the result of a DMS entry is treated as degrees in the Degree angle mode only. It is treated as radians in Radian angle mode.

Designates the entered angle is in DMS format. degrees (≤ 999,999), minutes (< 60), and seconds (< 60, may have decimal places) must be entered as real numbers, not as variable names or expressions.

Do not use ° and " symbols to specify degrees and seconds. For example, 5°59' is interpreted as implied multiplication of 5° * 59' according to the current angle mode setting.

\[
\begin{array}{ll}
\text{54}'32''30''} & \text{54.5416666667} \\
\text{In Degree angle mode:} & \\
\cos 54'32''30'' & \cos 5.80110760699 \\
\text{In Radian angle mode:} & \\
\cos 54'32''30'' & \cos -0.42250266138 \\
\text{Do not use the following notation; in Degree angle mode:} & \\
5'59'' & \text{295}
\end{array}
\]

\[
\begin{array}{ll}
\text{\textbf{angle}→DMS} & \\
\text{Displays angle in DMS format. The result is shown in degrees'minutes'seconds' format, even though you use degrees'minutes'seconds' to enter a DMS angle.} & \\
45.371\text{→DMS} & \text{45°22'}15.6''} \\
54'32''30'' & \text{109.083333333} \\
\text{Ans→DMS} & \text{109°5'}0''}
\end{array}
\]

\[
\begin{array}{ll}
\text{\textbf{dot}((vectorA, vectorB))} & \\
\text{Returns the dot product of two real or complex vectors.} & \\
\text{dot([a,b,c],[d,e,f])} & \text{returns } a\cdot d + b\cdot e + c\cdot f.
\end{array}
\]

\[
\begin{array}{ll}
\text{\textbf{DrawDot}} & \\
\text{† graph format screen} & \\
\text{Sets dot graphing format.}
\end{array}
\]
### DrawF

**GRAPH DRAW menu**

**DrawF** \( expression \)

Draws \( expression \) (in terms of \( x \)) on the current graph.

In **Func** graphing mode:

ZStd: DrawF 1.25 \( x \) cos \( x \) [ENTER]

---

### DrawLine

**† graph format screen**

**DrawLine**

Sets connected line graphing format.
DrEqu

To enter the ' character for
the Q' variables, use the
CHAR MISC menu.

DrEqu(xAxisVariable,yAxisVariable,xList,yList,tList)

In DiffEq graphing mode, draws the solution to a set of
differential equations stored in the Q' variables
specified by xAxisVariable and yAxisVariable. If
direction fields are off (FldOff is selected), the initial
values must be stored also.

After the solution is drawn, DrEqu waits for you to
move the cursor to a new initial value and press [ENTER]
to draw the new solution.

You then are prompted to press Y (to specify another
initial value) or N (to stop).

For the last-drawn solution, the x, y, and t values
(begining at their initial values) are stored to xList,
yList, and tList, respectively.

DrEqu(xAxisVariable,yAxisVariable)

Does not store x, y, and t values for the solution.

DrInv

Graph DRAW menu

DrInv expression

Draws the inverse of expression by plotting x values on
the y-axis and y values on the x-axis.

In DiffEq graphing mode, starting with a ZStd
graph screen:

- Q'1=Q2:Q2=Q1 [ENTER] Done
- o:QMin=1:Q11=0:Q12 [ENTER] 0
- DrEqu(Q1,Q2,XL,YL,TL) [ENTER]

Move the cursor to a new initial value.

Press N to stop graphing. You can then examine XL, YL, and TL.
<table>
<thead>
<tr>
<th><strong>DS&lt;</strong></th>
<th>Program editor&lt;br&gt;CTL menu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:DS&lt;</strong>(variable,value)&lt;br&gt;command-if-variable=value&lt;br&gt;commands</td>
<td><strong>Program segment:</strong>&lt;br&gt;...&lt;br&gt;:9^A&lt;br&gt;:Lbl Start&lt;br&gt;:Disp A&lt;br&gt;:DS&lt;(A,5)&lt;br&gt;:Goto Start&lt;br&gt;:Disp &quot;A is now &lt;5&quot;&lt;br&gt;:...&lt;br&gt;Decrements variable by 1. If the result is &lt; value, skips command-if-variable=value.&lt;br&gt;If the result is ≥ value, then command-if-variable=value is executed.&lt;br&gt;variable cannot be a built-in variable.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>dxDer1</strong></th>
<th>mode screen&lt;br&gt;dxDer1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sets der1 as the current differentiation type. der1 differentiates exactly and calculates the value for each function in an expression. It is more accurate than dxNDer, but more restrictive in that only certain functions are valid in the expression.</strong></td>
<td>The current differentiation type is used by the arc( and TanLn( functions, as well as interactive graphing operations dy/dx, dr/dθ, dy/dt, dx/dt, ARC, TanLn, and INFLC.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>dxNDer</strong></th>
<th>mode screen&lt;br&gt;dxNDer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sets nDer as the current differentiation type. nDer differentiates numerically and calculates the value for an expression. It is less accurate than dxDer1, but less restrictive in the functions that are valid in the expression.</strong></td>
<td>The current differentiation type is used by the arc( and TanLn( functions, as well as interactive graphing operations dy/dx, dr/dθ, dy/dt, dx/dt, ARC, TanLn, and INFLC.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>e^</strong></th>
<th>2nd [^e^]&lt;br&gt;e^power or e^(expression)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Returns e raised to power or expression. The argument can be real or complex.</strong></td>
<td>e^0 [ENTER] 1</td>
</tr>
</tbody>
</table>
\( e^\text{list} \)

Returns a list in which each element is \( e \) raised to the power specified by the corresponding element in \( \text{list} \).

\( e^{\text{squareMatrix}} \)

Returns a square matrix that is the matrix exponential of \( \text{squareMatrix} \). The matrix exponential corresponds to the result calculated using power series or Cayley-Hamilton Theorem techniques. This is not the same as simply calculating the exponential of each element.

\( \text{eigVc} \)

\( \text{MATRX MATH menu} \)

The \( \text{squareMatrix} \) cannot have repeated eigenvalues.

Returns a matrix containing the eigenvectors for a real or complex \( \text{squareMatrix} \), where each column in the result corresponds to an eigenvalue. The eigenvectors of a real matrix may be complex. Note that an eigenvector is not unique; it may be scaled by any constant factor. TI-86 eigenvectors are normalized.

\( \text{eigVl} \)

\( \text{MATRX MATH menu} \)

Returns a list of the eigenvalues of a real or complex \( \text{squareMatrix} \). The eigenvalues of a real matrix may be complex.

In RectC complex number mode:

\[
\begin{bmatrix}
1 & 2 & 5 \\
3 & -6 & 9 \\
2 & -5 & 7
\end{bmatrix}
\]

\( \text{eigVc} \) \( \text{MAT} \) \( \text{ENTER} \)

\[
\begin{bmatrix}
0.800906446592 & \ldots \\
-0.484028886343 & \ldots \\
-0.352512270699 & \ldots
\end{bmatrix}
\]

In RectC complex number mode:

\[
\begin{bmatrix}
1 & 2 & 5 \\
3 & -6 & 9 \\
2 & -5 & 7
\end{bmatrix}
\]

\( \text{eigVl} \) \( \text{MAT} \) \( \text{ENTER} \)

\[
\begin{bmatrix}
-4.0941084667 & \ldots
\end{bmatrix}
\]
Else
‡ program editor
CTL menu

Refer to syntax information for If, beginning on page 305. See the If:Then:Else:End syntax.

End
‡ program editor
CTL menu

Identifies the end of a While, For, Repeat, or If-Then-Else loop.

Eng
† mode screen

Sets engineering notation mode, in which the power-of-10 exponent is a multiple of 3.

In Eng notation mode:

\[123456789 \text{ ENTER} \quad 123.456789\times10^6\]

In Normal notation mode:

\[123456789 \text{ ENTER} \quad 123456789\]

Eq\#St( equationVariable,stringVariable )
STRNG menu

Converts the contents of equationVariable to a string and stores it to stringVariable. Be sure to specify an equation variable, not an equation.

To create an equation variable, use an equal sign (=) to define the variable. For example, enter \(\text{eq} = \text{B} + \text{C}\), not \(\text{B} = \text{C} \to \text{eq}\).

Example of \(=\) treated as \(-\):( where \(4 = 6 + 1\) is evaluated as \(-6+1\):

\[4 = 6 + 1 \text{ ENTER} \quad -3\]

For true/false comparison, use \(==\) instead:

\[4 == 6 + 1 \text{ ENTER} \quad 0\]
Equal to: ==

TEST menu

The == operator is used to compare arguments, while = is used to assign a value or expression to a variable.

Tests whether the condition argumentA == argumentB is true or false. Numbers, matrices, and vectors can be real or complex. If complex, the magnitude (modulus) of each element is compared. Strings are case-sensitive.

- If true (argumentA = argumentB), returns 1.
- If false (argumentA ≠ argumentB), returns 0.

```
2+2==2+2  ENTER  1
2+(2==2)+2  ENTER  5
[1,2]==[3-2,-1+3]  ENTER  1
"A"=="a"  ENTER  0
```

```
listA == listB
{1,5,9}==(1,-6,9)  ENTER  {1 0 1}
```

Euler

† graph format screen (scroll down to second screen)

In DifEq graphing mode, uses an algorithm based on the Euler method to solve differential equations. Typically, Euler is less accurate than RK but finds the solutions much quicker.

```
eval xValue
```

Returns a list containing the y values of all defined and selected functions evaluated at a real xValue.

```
y1=x^3+x+5  ENTER  Done
y2=2x  ENTER  Done
eval 5  ENTER  (135 10)
```

Remember that built-in equation variables y1 and y2 are case-sensitive:

```
y1=x^3+x+5
y2=2x
```
Chapter 20: A to Z Function and Instruction Reference

**evalF(expression,variable,value)**

CALC menu

Returns the value of expression evaluated with respect to variable at a real or complex value.

**evalF(expression,variable,list)**

Returns a list containing the values of expression evaluated with respect to variable at each element in list.

**Exponent: \( E \)**

\( number \ E \) power or \( (expressionA) \ E (expressionB) \)

Returns a real or complex number raised to the power of 10, where power is a real integer such that \(-999 < power < 999\). Any expressions must evaluate to appropriate values.

\( list \ E \) power or \( list \ E (expression) \)

Returns a list in which each element is the corresponding element in list raised to the power of 10.
ExpR

STAT CALC menu

Built-in equation variables such as y1, r1, and xt1 are case-sensitive. Do not use Y1, R1, and XT1.

ExpR xList, yList, frequencyList, equationVariable

Fits an exponential regression model \((y=ab^x)\) to real data pairs in xList and yList (y values must be > 0) and frequencies in frequencyList. The regression equation is stored to equationVariable, which must be a built-in equation variable such as y1, r1, and xt1.

Values used for xList, yList, and frequencyList are stored automatically to built-in variables xStat, yStat, and fStat, respectively. The regression equation is stored also to built-in equation variable RegEq.

ExpR xList, yList, equationVariable

Uses frequencies of 1.

ExpR xList, yList, frequencyList

Stores the regression equation to RegEq only.

ExpR xList, yList

Uses frequencies of 1, and stores the regression equation to RegEq only.

ExpR equationVariable

Uses xStat, yStat, and fStat for xList, yList, and frequencyList, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to equationVariable and RegEq.
ExpR
Uses \texttt{xStat}, \texttt{yStat}, and \texttt{fStat}, and stores the regression equation to \texttt{RegEq} only.

Factorial: \(!\)

**MATH PROB menu**

\(n!\) or \((expression)!\)

- Returns the factorial of a real integer or non-integer, where \(0 \leq \text{integer} \leq 449\) and \(0 \leq \text{non-integer} \leq 449.9\). For a non-integer, the Gamma function is used to find the factorial. An \textit{expression} must evaluate to an appropriate value.

\(l!\)

- Returns a list in which each element is the factorial of the corresponding element in \textit{list}.

\textbf{fcstx}

**\(\uparrow\) STAT menu**

\texttt{fcstx }\(y\text{Value}\)

- Based on the current regression equation (\texttt{RegEq}), returns the forecasted \(x\) at a real \(y\text{Value}\).

\textbf{fcsty}

**\(\uparrow\) STAT menu**

\texttt{fcsty }\(x\text{Value}\)

- Based on the current regression equation (\texttt{RegEq}), returns the forecasted \(y\) at a real \(x\text{Value}\).
Chapter 20: A to Z Function and Instruction Reference

**Fill(**

- **LIST OPS menu**
- **MATRX OPS menu**
- **VECTR OPS menu**

**Fill(number,LISTName)**

**Fill(number,MATRXName)**

**Fill(number,VECTORName)**

Replaces each element in an existing LISTName, MATRXName, or VECTORName with a real or complex number.

```
{3,4,5} \rightarrow L1
```

```
\text{Fill}(8,L1)
```

Done

```
L1 \rightarrow \{8 8 8\}
```

```
\text{Fill}((3,4),L1)
```

Done

```
L1 \rightarrow \{(3,4) (3,4) (3,4)\}
```

**Fix**

† mode screen

- **Fix**
- **Fix(expression)**

Sets fixed decimal mode for integer number of decimal places, where $0 \leq \text{integer} \leq 11$. An expression must evaluate to an appropriate integer.

```
\text{Fix 3}
```

Done

```
\pi/2 \rightarrow 1.571
```

```
\text{Float}
```

```
\pi/2 \rightarrow 1.57079632679
```

**FldOff**

† graph format screen (scroll down to second screen)

- **FldOff**

In **DifEq** graphing mode, turns off the slope and direction fields. To turn on slope fields, use **SlpFld**. To turn on direction fields, use **DirFld**.

**Float**

† mode screen

- **Float**

Sets floating decimal mode.

```
\text{Fix 11}
```

Done

```
\sin(\pi/6) \rightarrow .50000000000
```

```
\text{Float}
```

```
\sin(\pi/6) \rightarrow .5
```

In **Radian** angle mode:
Chapter 20: A to Z Function and Instruction Reference

\textbf{fMax(} \textbf{CALC menu} \textbf{fMax(} \textbf{expression,variable,lower,upper)} \textbf{Returns the value at which a local maximum of} \textbf{expression} \textbf{with respect to} \textbf{variable} \textbf{occurs, between real} \textbf{lower} \textbf{and} \textbf{upper} \textbf{values for} \textbf{variable}. \textbf{The tolerance is controlled by the built-in variable} \textbf{tol}, \textbf{whose default is} \textbf{1E}-5. \textbf{To view or set} \textbf{tol}, \textbf{press} \textbf{2nd [MEM]} \textbf{F4} \textbf{to display the tolerance editor.} \textbf{fMax(sin x,x,\pi,\pi)} \textbf{ENTER} \textbf{1.57079632598}

\textbf{fMin(} \textbf{CALC menu} \textbf{fMin(} \textbf{expression,variable,lower,upper)} \textbf{Returns the value at which a local minimum of} \textbf{expression} \textbf{with respect to} \textbf{variable} \textbf{occurs, between real} \textbf{lower} \textbf{and} \textbf{upper} \textbf{bounds for} \textbf{variable}. \textbf{The tolerance is controlled by the built-in variable} \textbf{tol}, \textbf{whose default is} \textbf{1E}-5. \textbf{To view or set} \textbf{tol}, \textbf{press} \textbf{2nd [MEM]} \textbf{F4} \textbf{to display the tolerance editor.} \textbf{fMin(sin x,x,\pi,\pi)} \textbf{ENTER} \textbf{-1.57079632691}

\textbf{fnInt(} \textbf{CALC menu} \textbf{fnInt(} \textbf{expression,variable,lower,upper)} \textbf{Returns the numerical function integral of} \textbf{expression} \textbf{with respect to} \textbf{variable}, \textbf{between real} \textbf{lower} \textbf{and} \textbf{upper} \textbf{bounds for} \textbf{variable}. \textbf{The tolerance is controlled by the built-in variable} \textbf{tol}, \textbf{whose default is} \textbf{1E}-5. \textbf{To view or set} \textbf{tol}, \textbf{press} \textbf{2nd [MEM]} \textbf{F4} \textbf{to display the tolerance editor.} \textbf{fnInt(x^2,x,0,1)} \textbf{ENTER} \textbf{.333333333333}

\textbf{FnOff} \textbf{† GRAPH VARS menu} \textbf{FnOff} \textbf{function#,function#, ...} \textbf{Deselects the specified equation function numbers.} \textbf{FnOff 1,3} \textbf{ENTER} \textbf{Done}
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FnOff</td>
<td>Deselects all equation function numbers.</td>
</tr>
<tr>
<td>FnOn</td>
<td>Selects all equation function numbers.</td>
</tr>
<tr>
<td>FnOn†</td>
<td>Selects the specified equation function numbers, in addition to any others</td>
</tr>
<tr>
<td></td>
<td>already selected.</td>
</tr>
</tbody>
</table>

**FnOn**

- **FnOn** `function#, function#`, ...
  - Selects the specified equation function numbers, in addition to any others already selected.

**FnOn**

- **FnOn** `function#`, ...
  - Selects the specified equation function numbers, in addition to any others already selected.

**For(**

- **For** `(variable, begin, end, step)` or **For** `(variable, begin, end)`
  - Executes the commands in `loop` iteratively, where the number of repetitions is controlled by `variable`. The first time through the loop, `variable = begin`. At the **End** of the loop, `variable` is incremented by `step`. The loop is repeated until `variable > end`. If you do not specify `step`, the default is 1.
  - You can specify values such that `begin > end`. If so, be sure to specify a negative `step`.

**Program segment:**

```
For(A, 0, 8, 2)  \Disp A^2  End
     \Disp 0, 4, 16, 36, and 64.

For(A, 0, 8)    \Disp A^2  End
     \Disp 0, 1, 4, 9, 16, 25, 36, 49, and 64.
```
Form("formula",listName)
Generates the contents of listName automatically, based on the attached formula. If you express formula in terms of a list, you can generate one list based on the contents of another.
The contents of listName are updated automatically if you edit formula or edit a list referenced in formula.

fPart
Returns the fractional part of a real or complex number or expression.

fPart list
Returns a list, matrix, or vector in which each element is the fractional part of the corresponding element in the specified argument.

Frac
Displays a real or complex number as its rational equivalent, a fraction reduced to its simplest terms.
If number cannot be simplified or if the denominator is more than four digits, the decimal equivalent is returned.
list\text{→}\text{Frac}
matrix\text{→}\text{Frac}
vector\text{→}\text{Frac}

Returns a list, matrix, or vector in which each element is the rational equivalent of the corresponding element in the argument.

\begin{align*}
\{1/2+1/3,1/6-3/8\} & \text{L1 Enter} \\
\{.833333333333, -.208\} & \text{Ans\text{→}Frac Enter} \\
\{5/6,5/24\} & \text{Ans Enter}
\end{align*}

**Func**

Sets function graphing mode.

**gcd(**

\text{gcd}(\text{integerA, integerB})

Returns the greatest common divisor of two nonnegative integers.

**gcd( listA, listB)**

Returns a list in which each element is the gcd of the two corresponding elements in listA and listB.

\begin{align*}
\text{gcd}(18,33) & \text{ Enter} \\
3 & \\
\text{gcd}\{12,14,16\},\{9,7,5\} & \text{ Enter} \\
\{3,7,1\} &
\end{align*}

**Get(**

\text{Get}(\text{variable})

Gets data from a CBL or CBR System or another TI-86 and stores it to variable.
**getKy**

Returns the key code for the last key pressed. If no key has been pressed, `getKy` returns 0. Refer to the TI-86 key code diagram in Chapter 16.

**Goto**

Transfers (branches) program control to the `label` specified by an existing `Lbl` instruction.

**Greater than:** >

Tests whether the condition is true or false. The arguments must be real numbers.
- If true `(numberA > numberB)`, returns 1.
- If false `(numberA ≤ numberB)`, returns 0.
<table>
<thead>
<tr>
<th><strong>number &gt; list</strong></th>
<th>Returns a list of 1s and/or 0s to indicate if <code>number</code> is &gt; the corresponding element in <code>list</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{1, 6, 10}</code></td>
<td><code>{0, 1, 0}</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>listA &gt; listB</strong></th>
<th>Returns a list of 1s and/or 0s to indicate if each element in <code>listA</code> is &gt; the corresponding element in <code>listB</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{1, 9, 5}</code></td>
<td><code>{0, 1, 0}</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Greater than or equal to: ≥</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TEST menu</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>numberA ≥ numberB</strong> or <code>{expressionA} ≥ {expressionB}</code></th>
<th>Tests whether the condition is true or false. The arguments must be real numbers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{2} ≥ 0</code></td>
<td><code>{1, 6, 10}</code></td>
</tr>
<tr>
<td><code>88 ≥ 123</code></td>
<td><code>{0}</code></td>
</tr>
<tr>
<td><code>-5 ≥ -5</code></td>
<td><code>{1}</code></td>
</tr>
<tr>
<td><code>{20 * 5 / 2} ≥ {10 * 2}</code></td>
<td><code>{1}</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>number ≥ list</strong></th>
<th>Returns a list of 1s and/or 0s to indicate if <code>number</code> is ≥ the corresponding element in <code>list</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{1, 6, 10}</code></td>
<td><code>{1, 1, 0}</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>listA ≥ listB</strong></th>
<th>Returns a list of 1s and/or 0s to indicate if each element in <code>listA</code> is ≥ the corresponding element in <code>listB</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{1, 9, 5}</code></td>
<td><code>{1, 1, 0}</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>GridOff</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>† graph format screen</td>
</tr>
</tbody>
</table>
GridOn
† graph format screen
   Turns on grid format so that grid points are displayed in
   rows and columns corresponding to the tick marks on
   each axis.

GrStl( function#, graphStyle#)
   Sets the graph style for function#. For graphStyle#,
specify an integer from 1 through 7:
   1 = \ (line)  4 = h (below)  7 = ', (dot)
   2 = \ (thick)  5 = o (path)
   3 = \ (above)  6 = @ (animate)
   Depending on the graphing mode, some graph styles may
   not be available.

h
   BASE TYPE menu
   Designates a real integer as hexadecimal, regardless of
   the number base mode setting.

Hex
† mode screen
   Sets hexadecimal number base mode. Results are
   displayed with the h suffix. In any number base mode,
you can designate an appropriate value as binary,
decimal, hexadecimal, or octal by using the b, d, h, or o
   designator, respectively, from the BASE TYPE menu.

   To enter hexadecimal numbers A through F, use the
   BASE A-F menu. Do not use [ALPHA] to type a letter.
Chapter 20: A to Z Function and Instruction Reference

**Hex**

BASE CONV menu

*number* → Hex

*list* → Hex

*matrix* → Hex

*vector* → Hex

Returns the hexadecimal equivalent of the real or complex argument.

<table>
<thead>
<tr>
<th>In Bin number base mode:</th>
<th>1010 → 1110 ENTER</th>
<th>8Ch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100,101,110 ENTER</td>
<td>4h 5h 6h</td>
</tr>
</tbody>
</table>

**Hist**

† STAT DRAW menu

Hist *xList*, frequencyList

Draws a histogram on the current graph, using the real data in *xList* and the frequencies in frequencyList.

Hist *xList*

Uses frequencies of 1.

Hist

Uses the data in built-in variables xStat and fStat. These variables must contain valid data of the same dimension; otherwise, an error occurs.

Starting with a ZStd graph screen:

[1,2,3,4,6,7] → XL ENTER

[1,6,4,2,3,5] → FL ENTER

0 → yMin ENTER

Hist XL, FL ENTER

[1,1,2,2,3,3,3,3,3,3,4,4,5,5,5,7,7] → XL ENTER

[1 1 2 2 3 3 3 3 3 ... ] → CDrw: Hist XL ENTER

Horiz† GRAPH DRAW menu

Horiz \( y\)Value
Draws a horizontal line on the current graph at \(y\)Value.

In a \(Z\)Std graph screen:
Horiz 4.5 \(\text{ENTER}\)

IAAsk CATLOG

IAAsk
Sets the table so that the user can enter individual values for the independent variable.

IAuto CATLOG

IAuto
Sets the table so that the TI-86 generates the independent-variable values automatically, based on values entered for TblStart and \(\Delta\)Tbl.

Ident MATRX OPS menu

Ident \(\text{dimension}\)
Returns the identity matrix of \(\text{dimension}\) rows \(\times\) \(\text{dimension}\) columns.

Ident 4 \(\text{ENTER}\)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
If

program editor

CTL menu

:If condition
:command-if-true
:commands

If condition is true, executes command-if-true. Otherwise, skips command-if-true. The condition is true if it evaluates to any nonzero number, or false if it evaluates to zero.

To execute multiple commands if condition is true, use If:Then:End instead.

:If condition
:Then
:commands-if-true
:End
:commands

If condition is true (nonzero), executes commands-if-true from Then to End. Otherwise, skips commands-if-true and continues with the next command following End.

Program segment:

:If x<0
:Disp "x is negative"
:End

Program segment:

:If x<0
:Then
:Disp "x is negative"
:abs(x)→x
:End

:If condition
:Then
commands-if-true
:Else
commands-if-false
:End
commands

If condition is true (nonzero), executes commands-if-true from Then to Else and then continues with the next command following End.

If condition is false (zero), executes commands-if-false from Else to End and then continues with the next command following End.

imag

Returns the imaginary (nonreal) part of complexNumber. The imaginary part of a real number is always 0.

imag(real,imaginary) returns imaginary.
imag(magnitude,angle) returns magnitude \sin angle.

imag complexList
imag complexMatrix
imag complexVector

Returns a list, matrix, or vector in which each element is the imaginary part of the original argument.
InpSt

‡ program editor
I/O menu

InpSt promptString, variable
Pauses a program, displays promptString, and waits for the user to enter a response. The response is stored to variable always as a string. When entering the response, the user should not enter quotation marks.

To prompt for a number or expression instead of a string, use Input.

InpSt variable
Displays ? as the prompt.

Program segment:
```
:InpSt "Enter your name:", STR
```

Input

‡ program editor
I/O menu

Input promptString, variable
Pauses a program, displays promptString, and waits for the user to enter a response. The response is stored to variable in the form in which the user enters it.

- A number or expression is stored as a number or expression.
- A list, vector, or matrix is stored as a list, vector, or matrix.
- An entry enclosed in " marks is stored as a string.

Input variable
Displays ? as the prompt.

Program segment:
```
:Input "Enter test score:", SCR
```
**Input**

Pauses a program, displays the graph screen, and lets
the user update \( x \) and \( y \) (or \( r \) and \( \theta \) in PolarGC graph
format) by moving the free-moving cursor. To resume
the program, press \( \text{ENTER} \).

**Input "CBLGET",variable**

Receives list data sent from a CBL or CBR System and
stores it to variable on the TI-86. Use this "CBLGET"
syntax for both CBL and CBR.

You can receive data also by using \text{Get} \( \) as described on
page 299.

\[
\text{int number or int (expression)}
\]

Returns the largest integer ≤ \( \text{number or expression} \). The
argument can be real or complex.

For a negative non-integer, \text{int} returns the integer that is
one less than the integer part of the number. To return
the exact integer part, use \text{iPart} instead.

\[
\begin{align*}
\text{int list} & \quad \text{int matrix} \\
& \quad \text{int vector}
\end{align*}
\]

Returns a list, matrix, or vector in which each element
is the largest integer less than or equal to the

**Program segment in RectGC graph format:**

\[
\begin{align*}
: & \text{Input} \\
& \text{Disp} \ x,y
\end{align*}
\]

\[
\text{Input "CBLGET",L1 \text{ ENTER} \ Done}
\]

\[
\begin{align*}
\text{int} & \quad 23.45 \ \text{ENTER} \quad 23 \\
& \quad -23.45 \ \text{ENTER} \quad -24 \\
\text{int MAT} & \quad \text{[[1.25,-23.45],[-99,47.15]]} \rightarrow \text{MAT} \quad \text{[[1.25,-23.45]}
\ &= \quad \text{[-99,47.15]]} \\
\end{align*}
\]
inter(† MATH menu
inter(x1,y1,x2,y2,xValue)
Calculates the line through points (x1,y1) and (x2,y2) and then interpolates or extrapolates a y value for the specified xValue.
inter(y1,x1,y2,x2,yValue)
Interpolates or extrapolates an x value for the specified yValue. Notice that points (x1,y1) and (x2,y2) must be entered as (y1,x1) and (y2,x2).

Using points (3,5) and (4,4), find the y value at x=1:
inter(3,5,4,4,1) ENTER 7

Using points (-4,-7) and (2,6), find the x value at y=10:
inter(-7,-4,6,2,10) ENTER 3.84615384615

Inverse: \(^{-1}\)

5 \(^{-1}\) ENTER .2

(10+6) \(^{-1}\) ENTER .01666666667

\([-\cdot.5,10,2/8]\) \(^{-1}\) ENTER \([-\cdot2\ .1\ .4]\)

list \(^{-1}\)
Returns a list in which each element is 1 divided by the corresponding element in list.
squareMatrix \(^{-1}\)
Returns an inverted squareMatrix, where det \(\neq 0\).

iPart

iPart number or iPart(expression)
Returns the integer part of number or expression. The argument can be real or complex.

iPart 23.45 ENTER 23
iPart -23.45 ENTER -23
Chapter 20: A to Z Function and Instruction Reference

iPart list
iPart matrix
iPart vector

Returns a list, matrix, or vector in which each element is the integer part of the corresponding element in the specified argument.

\[
\begin{bmatrix}
1.25, -23.45
\end{bmatrix}
\begin{bmatrix}
-99.5, 47.15
\end{bmatrix}
\]

\[
\text{iPart MAT} \ \text{ENTER} \\
\begin{bmatrix}
1, -23
\end{bmatrix}
\begin{bmatrix}
-99, 47
\end{bmatrix}
\]

IS>(

‡ program editor
CTL menu

:IS>(variable, value)

\text{command-if-variables}value
\text{commands}

Increments \text{variable} by 1. If the result is > \text{value}, skips \text{command-if-variables}value.

If the result is ≤ \text{value}, then \text{command-if-variables}value is executed.

\text{variable} cannot be a built-in variable.

LabelOff

† graph format screen

Turns off axes labels.

LabelOn

† graph format screen

Turns on axes labels.
**Lbl**

**Label**

A label is a string of up to eight characters used to specify a location in a program. A program can use a Goto instruction to transfer control (branch) to a specified label.

Program segment, assuming a correct password has already been stored to the **password** variable:

```plaintext
"Lbl Start
:InpSt "Enter password:" , PSW
:If PSW ≠ password
:Goto Start
:Disp "Welcome"
```

**lcm(***integer***A,***integer***B)**

Returns the least common multiple of two nonnegative integers.

<table>
<thead>
<tr>
<th>lcm(5,2)</th>
<th>lcm(6,9)</th>
<th>lcm(10,33)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER</td>
<td>ENTER</td>
<td>ENTER</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>198</td>
</tr>
</tbody>
</table>

**LCust(***item***#,"***title***" [,***item***#,"***title***", ...])**

Loads a custom menu displayed when the user presses [CUSTOM]. The menu can have up to 15 items, shown in three groups of five items. For each **item#/title** pair:

- **item** — integer from 1 through 15 that identifies the item's position in the menu. The item numbers must be specified in order, but you can skip numbers.
- **title** — string with up to 8 characters (not counting the quotes) that will be pasted to the current cursor location when the item is selected. This can be a variable name, expression, function name, program name, or any text string.

Program segment:

```plaintext
```

After executed and when the user presses [CUSTOM]:

1 2 3 4 5 6 7 8 9 0
<table>
<thead>
<tr>
<th>Less than: &lt;</th>
<th>\text{TEST menu}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{numberA} &lt; \text{numberB} \text{ or } (\text{expressionA}) &lt; (\text{expressionB})</td>
<td>\begin{align*} &amp;\text{2} &lt; 0 \ \text{(ENTER)} \quad 0 \ &amp;\text{88} &lt; 123 \ \text{(ENTER)} \quad 1 \ &amp;\text{-5} &lt; -5 \ \text{(ENTER)} \quad 0 \ &amp;\text{(20 \cdot 5/2)} &lt; (18 \cdot 3) \ \text{(ENTER)} \quad 1 \end{align*}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>\text{number} &lt; \text{list}</th>
<th>Returns a list of 1s and/or 0s to indicate if \text{number} is &lt; the corresponding element in \text{list}.</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{listA} &lt; \text{listB}</td>
<td>Returns a list of 1s and/or 0s to indicate if each element in \text{listA} is &lt; the corresponding element in \text{listB}.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Less than or equal to: ≤</th>
<th>\text{TEST menu}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{numberA} ≤ \text{numberB} \text{ or } (\text{expressionA}) ≤ (\text{expressionB})</td>
<td>\begin{align*} &amp;\text{2} ≤ 0 \ \text{(ENTER)} \quad 0 \ &amp;\text{88} ≤ 123 \ \text{(ENTER)} \quad 1 \ &amp;\text{-5} ≤ -5 \ \text{(ENTER)} \quad 1 \ &amp;\text{(20 \cdot 5/2)} ≤ (18 \cdot 3) \ \text{(ENTER)} \quad 1 \end{align*}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>\text{number} ≤ \text{list}</th>
<th>Returns a list of 1s and/or 0s to indicate if \text{number} is ≤ the corresponding element in \text{list}.</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{listA} ≤ \text{listB}</td>
<td>Returns a list of 1s and/or 0s to indicate if each element in \text{listA} is ≤ the corresponding element in \text{listB}.</td>
</tr>
</tbody>
</table>

| \begin{align*} &\text{1} ≤ \{1, -6, 10\} \ \text{(ENTER)} \quad \{0 \ 0 \ 1\} \\ &\text{1,5,9} ≤ \{1, -6, 10\} \ \text{(ENTER)} \quad \{0 \ 0 \ 1\} \\ &\text{1} ≤ \{1, -6, 10\} \ \text{(ENTER)} \quad \{1 \ 0 \ 1\} \\ &\text{1,5,9} ≤ \{1, -6, 10\} \ \text{(ENTER)} \quad \{1 \ 0 \ 1\} \end{align*} |
LgstR

STAT CALC menu

Built-in equation variables such as \( y_1, r_1, \) and \( xT_1 \) are case-sensitive. Do not use \( Y_1, R_1, \) and \( XT_1 \).

\( \text{LgstR} \) returns a tolMet value that indicates if the result meets the TI-86’s internal tolerance.

- If tolMet=1, the result is within the internal tolerance.
- If tolMet=0, the result is outside the internal tolerance, although it may be useful for general purposes.

\( \text{LgstR} \) \( [\) \( \text{iterations} \), \( xList \), \( yList \), \( frequencyList \), \( equationVariable \) \( ] \)

Fits a logistic regression model \( (y=a/(1+be^{cx})+d) \) to real data pairs in \( xList \) and \( yList \) and frequencies in \( frequencyList \). The regression equation is stored to \( equationVariable \), which must be a built-in equation variable such as \( y_1, r_1, \) and \( xT_1 \). The equation’s coefficients always are stored as a list to built-in variable \( \text{PRegC} \).

The number of \( \text{iterations} \) is optional. If omitted, 64 is the default. A large number of \( \text{iterations} \) may produce more accurate results but may require longer calculation times. A smaller number may produce less accurate results but with shorter calculation times.

Values used for \( xList \), \( yList \), and \( frequencyList \) are stored automatically to built-in variables \( xStat \), \( yStat \), and \( fStat \), respectively. The regression equation is stored also to built-in equation variable \( \text{RegEq} \).

In Func graphing mode:

\[
\begin{align*}
\text{LgstR} \, \text{L1}, \text{L2}, \text{y1} & \quad \text{L1} \, \text{L1} \, \text{L2} \\
\text{Plot1}(1, \text{L1}, \text{L2}) & \quad \text{Done}
\end{align*}
\]

\( \text{LgstR} \) \( [\) \( \text{iterations} \), \( xList \), \( yList \), \( frequencyList \) \( ] \)

Uses frequencies of 1.

\( \text{LgstR} \) \( [\) \( \text{iterations} \), \( xList \), \( yList \), \( frequencyList \) \( ] \)

Stores the regression equation to \( \text{RegEq} \) only.

\( \text{LgstR} \) \( [\) \( \text{iterations} \), \( xList \), \( yList \) \( ] \)

Uses frequencies of 1, and stores the regression equation to \( \text{RegEq} \) only.
LgstR [iterations,equationVariable]

Uses xStat, yStat, and fStat for xList, yList, and frequencyList, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to equationVariable and RegEq.

LgstR [iterations]

Uses xStat, yStat, and fStat, and stores the regression equation to RegEq only.

Line(†)

Draws a line from point \((x_1,y_1)\) to \((x_2,y_2)\).

In Func graphing mode and a ZStd graph screen:

\[\text{Line(-2,-7,9,8)}\]
LinR

STAT CALC menu

Built-in equation variables such as $y_1$, $r_1$, and $xt_1$ are case-sensitive. Do not use $Y1$, $R1$, and $XT1$.

LinR $xList,yList,frequencyList,equationVariable$

Fits a linear regression model ($y=a+bx$) to real data pairs in $xList$ and $yList$ and frequencies in $frequencyList$. The regression equation is stored to $equationVariable$, which must be a built-in equation variable such as $y_1$, $r_1$, and $xt_1$.

Values used for $xList$, $yList$, and $frequencyList$ are stored automatically to built-in variables $xStat$, $yStat$, and $fStat$, respectively. The regression equation is stored also to built-in equation variable $RegEq$.

LinR $xList,yList,equationVariable$

Uses frequencies of 1.

LinR $xList,yList,frequencyList$

Stores the regression equation to $RegEq$ only.

LinR $xList,yList$

Uses frequencies of 1, and stores the regression equation to $RegEq$ only.

LinR $equationVariable$

Uses $xStat$, $yStat$, and $fStat$ for $xList$, $yList$, and $frequencyList$, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to $equationVariable$ and $RegEq$.

In Func graphing mode:

```
**{1,2,3,4,5,6}**⇒L1 ENTER
**{1 2 3 4 5 6}**⇒L2 ENTER
**{4.5,4.6,6.7,5,8,5.8}**⇒L2 ENTER
LinR L1,L2,y1 ENTER
```

*ZData*
LinR

Uses \texttt{xStat}, \texttt{yStat}, and \texttt{fStat}, and stores the regression equation to \texttt{RegEq} only.

\textbf{List entry: \{ \}}

\textbf{LIST menu}

\textit{L1 \{1 2 3\} ENTER}

defines a list in which each element is a real or complex number or variable.

\textbf{li\textasciitilde vc list}

\textbf{VECTR OPS menu}

\textbf{li\textasciitilde vc \{2,7,-8,0\} ENTER}

\textit{[2 7 -8 0]}

\textbf{In number or In (expression)}

\textbf{In (36.4/3) ENTER}

\textit{2.4959648597}

returns the natural logarithm of a real or complex number or expression.

\textbf{In list}

\textbf{In \{2,3\} ENTER}

\textit{[.69314718056 1.0986]}...

returns a list in which each element is the natural logarithm of the corresponding element in list.

\textbf{lngth string}

\textbf{lngth \"The answer is: \" ENTER}

\textit{14}

returns the length (number of characters) of string. The character count includes spaces but not quotation marks.

\textbf{lngth STR ENTER}

\textit{14}
LnR

STAT CALC menu

Built-in equation variables such as \( y_1 \), \( r_1 \), and \( xT_1 \) are case-sensitive. Do not use \( Y1 \), \( R1 \), and \( XT1 \).

LnR \( xList, yList, frequencyList, equationVariable \)

Fits a logarithmic regression model \( (y=a+b \ln x) \) to the real data pairs in \( xList \) and \( yList \) (\( x \) values must be > 0) and frequencies in \( frequencyList \). The regression equation is stored to \( equationVariable \), which must be a built-in equation variable such as \( y1 \), \( r1 \), and \( xT1 \).

Values used for \( xList \), \( yList \), and \( frequencyList \) are stored automatically to built-in variables \( xStat \), \( yStat \), and \( fStat \), respectively. The regression equation is stored also to built-in equation variable \( RegEq \).

LnR \( xList, yList, equationVariable \)

Uses frequencies of 1.

LnR \( xList, yList, frequencyList \)

Stores the regression equation to \( RegEq \) only.

LnR \( xList, yList \)

Uses frequencies of 1, and stores the regression equation to \( RegEq \) only.

LnR \( equationVariable \)

Uses \( xStat \), \( yStat \), and \( fStat \) for \( xList \), \( yList \), and \( frequencyList \), respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to \( equationVariable \) and \( RegEq \).
LnR

Uses xStat, yStat, and fStat, and stores the regression equation to RegEq only.

log

log number or log (expression)

Returns the logarithm of a real or complex number or expression, where:

\[10^{\text{logarithm}} = \text{number}\]

log list

Returns a list in which each element is the logarithm of the corresponding element in list.

LU(

MATRX MATH menu

Calculates the Crout LU (lower-upper) decomposition of a real or complex matrix. The lower triangular matrix is stored in lMatrixName, the upper triangular matrix in uMatrixName, and the permutation matrix (which describes the row swaps done during the calculation) in pMatrixName.

\[lMatrixName \times uMatrixName = pMatrixName \times \text{matrix}\]
Matrix entry: [ ]

Matrix entry: 

\[
\begin{bmatrix}
\text{row1} \\
\text{row2} \\
... \\
\end{bmatrix}
\]

Defines a matrix entered row-by-row in which each element is a real or complex number or variable. Enter each row as \[\text{element, element, ...}\].

**max(**

**MATH NUM menu**

\[
\text{max}(\text{numberA, numberB})
\]

Returns the larger of two real or complex numbers.

\[
\text{max}(\text{list})
\]

Returns the largest element in list.

\[
\text{max}(\text{listA, listB})
\]

Returns a list in which each element is the larger of the corresponding elements in listA and listB.

**MBox**

**STAT DRAW menu**

\[
\text{MBox xList, frequencyList}
\]

Draws a modified box plot on the current graph, using the real data in xList and the frequencies in frequencyList.

\[
\text{MBox xList}
\]

Uses frequencies of 1.

\[
\text{MBox}
\]

Uses the data in built-in variables xStat and fStat. These variables must contain valid data of the same dimension; otherwise, an error occurs.
Menu( ‡ program editor
CTL menu

Menu(item#,"title1",label1[, ... ,item#,"title15",label15])
Generates a menu of up to 15 items during program
execution. Menus are displayed as three groups of five
items. For each item:
• item# — integer from 1 through 15 that identifies this
item's position in the menu.
• "title" — text string that will be displayed for this
item on the menu. Typically, use from 1 through 5
characters; additional characters may not be seen on
the menu.
• label — valid label to which program execution will
branch when the user selects this item.

Program segment:
:…
:Lbl A
:Input "Radius:", RADIUS
:Disp "Area is:", π*RADIUS^2
:Menu(1,"Again",A,5,"Stop",B)
:Lbl B
:Disp "The End"

Example when executed:
Radius:
Area is: 78.5398163397

min(numberA,numberB)
Returns the smaller of two real or complex numbers.

min(list)
Returns the smallest element in list.

min(listA,listB)
Returns a list in which each element is the smaller of
the corresponding elements in listA and listB.

mod(numberA,numberB)
Returns numberA modulo numberB. The arguments
must be real.
**mRAdd**

**MATRX OPS menu**

Returns the result of a “multiply and add” matrix operation, where:

a. rowA of a real or complex matrix is multiplied by a real or complex number.

b. The results are added to (and then stored in) rowB.

```
\[
\begin{pmatrix}
5 & 3 & 1 \\
2 & 0 & 4 \\
3 & L & 2
\end{pmatrix}
mRAdd(5,MAT,2,3) \quad \begin{pmatrix}
5 & 3 & 1 \\
2 & 0 & 4 \\
13 & 1 & 22
\end{pmatrix}
```

**Multiplication: **

```
\text{numberA} \bullet \text{numberB}
```

Returns the product of two real or complex numbers.

```
2 \bullet 5 \quad \text{ENTER} \quad 10
```

```
4 \bullet (10,9,8) \quad \text{ENTER} \quad (40,36,32)
```

**In RectC complex number mode:**

```
\begin{pmatrix} 8,1,(5,2) \end{pmatrix} \bullet 3 \quad \text{ENTER} \quad \begin{pmatrix}
24,0 & (3,0) & (15,6)
\end{pmatrix}
```

```
\text{listA} \bullet \text{listB}
```

Returns a list in which each element of listA is multiplied by the corresponding element of listB. The lists must have the same dimension.

```
\{1,2,3\} \bullet \{4,5,6\} \quad \text{ENTER} \quad \{4,10,18\}
```

```
\text{matrix} \bullet \text{vector}
```

Returns a vector in which matrix is multiplied by vector. The number of columns in matrix must equal the number of elements in vector.

```
\begin{pmatrix} 1,2,3 & 4,5,6 \end{pmatrix} \bullet \begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix} \quad \text{ENTER} \quad \begin{pmatrix}
50 & 122
\end{pmatrix}
```
Chapter 20: A to Z Function and Instruction Reference

matrixA * matrixB

Returns a matrix in which matrixA is multiplied by matrixB. The number of columns in matrixA must equal the number of rows in matrixB.

```
matrixA = [[2,2]
           [3,4]]
matrixB = [[1,2,3]
           [4,5,6]]
matrixA * matrixB = [[10, 14, 18]
                     [19, 26, 33]]
```

```
multR(5, MAT, 2)
```

```
multR(number, matrix, row)
```

Returns the result of a “row multiplication” matrix operation, where:

a. The specified row of a real or complex matrix is multiplied by a real or complex number.

b. The results are stored in the same row.

```
multR(5, MAT, 2) = [[5  3  1]
                    [10 0  20]
                    [3 -1  2]]
```

nCr

```
items nCr number
```

Returns the number of combinations of items (n) taken number (r) at a time. Both arguments must be real nonnegative integers.

```
5 nCr 2 = 10
```
nDer(expression, variable, value)

Returns an approximate numerical derivative of expression with respect to variable evaluated at a real or complex value. The approximate numerical derivative is the slope of the secant line through the points: 

\[(value, f(value + \delta)) \text{ and } (value - \delta, f(value - \delta))\]

As the step value \(\delta\) gets smaller, the approximation usually gets more accurate.

For \(\delta = 0.001:\)

\[
nDer(x^3, x, 5) \rightarrow \text{75.000001}
\]

For \(\delta = 1 \times 10^{-4}:\)

\[
nDer(x^3, x, 5) \rightarrow \text{75}
\]

nDer(expression, variable)

Uses the current value of variable.

\[
5 \times x \rightarrow \text{5}
\]

\[
nDer(x^3, x) \rightarrow \text{75}
\]

Negation: ~

- number or ~ (expression)
- list
- matrix
- vector

Returns the negative of the real or complex argument.

\[
-2+5 \rightarrow \text{3}
\]

\[
-(2+5) \rightarrow \text{-7}
\]

\[
-(0,5,5) \rightarrow \{0,5,-5\}
\]

norm matrix

MATRX MATH menu

Returns the Frobenius norm of a real or complex matrix, calculated as:

\[\sqrt{\sum (real^2 + imaginary^2)}\]

where the sum is over all elements.

\[
\|\begin{bmatrix}1 & -2 \\ -3 & 4 \end{bmatrix}\| \rightarrow \text{5.4772557505}
\]
norm vector

Returns the length of a real or complex vector, where:

\[ \text{norm} \ [a, b, c] \text{ returns } \sqrt{a^2 + b^2 + c^2}. \]

\[
\text{norm} \ [3, 4, 5] \text{ ENTER } 7.07106781187
\]

norm number or norm (expression)

Returns the absolute value of a real or complex number or expression, or of each element in list.

\[
\text{norm} \ -25 \text{ ENTER } 25
\]

In Radian angle mode:

\[
\text{norm} \ \{-25, \cos \left(\frac{\pi}{3}\right)\} \text{ ENTER } \{25, 5\}
\]

Normal

† mode screen

Normal

Sets normal notation mode.

In Eng notation mode:

\[
123456789 \text{ ENTER } 123.456789\times 6
\]

In Sci notation mode:

\[
123456789 \text{ ENTER } 1.23456789\times 8
\]

In Normal notation mode:

\[
123456789 \text{ ENTER } 123456789
\]
not integer

Returns the one's complement of a real integer. Internally, integer is represented as a 16-bit binary number. The value of each bit is flipped (0 becomes 1, and vice versa) for the one's complement.

For example, not 78:

\[
78 = 0000000001001110_b \\
1111111101100001_b \quad \text{(one's complement)}
\]

Sign bit; 1 indicates a negative number

To find the magnitude of a negative binary number, determine its two's complement (take the one's complement and then add 1). For example:

\[
111111110110001_b = \text{one's complement of } 78 \\
000000001011110_b \quad \text{(one's complement)} \\
+ 000000000000001_b \\
000000001011111_b = 79 \quad \text{(two's complement)}
\]

Therefore, not 78 = -79.

You can enter real numbers instead of integers, but they are truncated automatically before the comparison.
### Not equal to: ≠

**TEST menu**

<table>
<thead>
<tr>
<th>expression</th>
<th>description</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>numberA ≠ numberB</code></td>
<td>Tests whether the condition <code>argumentA ≠ argumentB</code> is true or false. Numbers, matrices, and vectors can be real or complex. If complex, the magnitude (modulus) of each element is compared. Strings are case-sensitive.</td>
<td><code>2+2 ≠ 3+2</code>&lt;br&gt;<code>[1,2] ≠ [3-2,1+3]</code>&lt;br&gt;<code>“A” ≠ “a”</code></td>
</tr>
<tr>
<td><code>matrixA ≠ matrixB</code></td>
<td></td>
<td><code>b&lt;sub&gt;1&lt;/sub&gt;</code></td>
</tr>
<tr>
<td><code>vectorA ≠ vectorB</code></td>
<td></td>
<td><code>b&lt;sub&gt;5&lt;/sub&gt;</code></td>
</tr>
<tr>
<td><code>stringA ≠ stringB</code></td>
<td></td>
<td><code>b&lt;sub&gt;0&lt;/sub&gt;</code></td>
</tr>
</tbody>
</table>

- If true (`argumentA ≠ argumentB`), returns 1.
- If false (`argumentA = argumentB`), returns 0.

| listA ≠ listB | Returns a list of 1s and/or 0s to indicate if each element in `listA` is ≠ the corresponding element in `listB`. | `{1,5,9} ≠ {1,-6,9}`<br>`{0 1 0}` |

### nPr

**MATH PROB menu**

| items nPr number | Returns the number of permutations of `items` (`n`) taken `number` (`r`) at a time. Both arguments must be real nonnegative integers. | `5 nPr 2`<br>`20` |

### O

**BASE TYPE menu**

| integer o | Designates a real integer as octal, regardless of the number base mode setting. | In **Dec** number base mode: | `10o`<br>`18o`<br>`8`<br>`18` |
Oct

† mode screen

Sets octal number base mode. Results are displayed with the œ suffix. In any number base mode, you can designate an appropriate value as binary, decimal, hexadecimal, or octal by using the b, d, h, or œ designator, respectively, from the BASE TYPE menu.

Oct

Oct

In Oct number base mode:

\[ 10 + 10b + 10h + 10d \]  

43o

Oct

BASE CONV menu

\( \text{number} \rightarrow \text{Oct} \)

\( \text{list} \rightarrow \text{Oct} \)

\( \text{matrix} \rightarrow \text{Oct} \)

\( \text{vector} \rightarrow \text{Oct} \)

Returns the octal equivalent of the real or complex argument.

Oct

Oct

In Dec number base mode:

\[ 2 \rightarrow b \]  

\[ \text{Ans} \rightarrow \text{Oct} \]  

16

\[ \{7, 8, 9, 10\} \rightarrow \text{Oct} \]  

\{7œ 10œ 11œ 12œ\}

Oct

Oct

OneVar

STAT CALC menu

(OneVa shows on menu)

\( \text{OneVar} \ x\text{List}, f\text{requencyList} \)

Performs one-variable statistical analysis using real data points in \( x\text{List} \) and frequencies in \( f\text{requencyList} \).

The values used for \( x\text{List} \) and \( f\text{requencyList} \) are stored automatically to built-in variables \( x\text{Stat} \) and \( f\text{Stat} \), respectively.

\( \text{OneVar} \ x\text{List} \)

Uses frequencies of 1.

OneVar

\( x\text{List} \)

\( {\{0, 1, 2, 3, 4, 5, 6\}} \rightarrow XL \)  

\{0 1 2 3 4 5 6\}

OneVar XL

\( 1-\text{Var} \rightarrow \text{Stats} \)

\( \{0\} \rightarrow 2 \)

\( \{0\} \rightarrow 2 \)

\( \{1.682469\} \rightarrow 2 \)

\( \{\} \rightarrow 2 \)

Scroll down to see more results.
OneVar
Uses xStat and fStat for xList and frequencyList. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs.

or
BASE BOOL menu

integerA or integerB
Compares two real integers bit by bit. Internally, both integers are converted to binary. When corresponding bits are compared, the result is 1 if either bit is 1; the result is 0 only if both bits are 0. The returned value is the sum of the bit results.

For example, 78 or 23 = 95.

\[
\begin{align*}
78 &= 1001110_b \\
23 &= 0010111_b \\
78 \text{ or } 23 &= 1011111_b = 95
\end{align*}
\]

You can enter real numbers instead of integers, but they are truncated automatically before the comparison.
Outpt(  
† program editor  
I/O menu  

Outpt(row,column,string)  
Displays string beginning at row and column, where  
1 ≤ row ≤ 8 and 1 ≤ column ≤ 21.  

Outpt(row,column,value)  
Displays value beginning at the specified row and  
column.  

Outpt("CBLSEND",listName)  
Sends the contents of listName to the CBL or CBR  
System.  

You can send data also by using Send( as described on  
page 350.  

Program segment:  
...  
:ClrLCD  
:For(i,1,8)  
: Outpt(i,randInt(1,21),"A")  
:End  
...  

Example result after execution:
**P2Reg**

**STAT CALC menu**

Built-in equation variables such as \(y_1\), \(r_1\), and \(xt_1\) are case-sensitive. Do not use \(Y1\), \(R1\), and \(XT1\).

**P2Reg** \(xList,yList,frequencyList\), \(equationVariable\)

Performs a second order polynomial regression using real data pairs in \(xList\) and \(yList\) and frequencies in \(frequencyList\). The regression equation is stored to \(equationVariable\), which must be a built-in equation variable such as \(y1\), \(r1\), and \(xt1\). The equation’s coefficients always are stored as a list to built-in variable \(PRegC\).

Values used for \(xList\), \(yList\), and \(frequencyList\) are stored automatically to built-in variables \(xStat\), \(yStat\), and \(fStat\), respectively. The regression equation is stored also to built-in equation variable \(RegEq\).

**P2Reg** \(xList\), \(yList\), \(equationVariable\)

Uses frequencies of 1.

**P2Reg** \(xList\), \(yList\), \(frequencyList\)

Stores the regression equation to \(RegEq\) only.

**P2Reg** \(xList\), \(yList\)

Uses frequencies of 1, and stores the regression equation to \(RegEq\) only.

**P2Reg** \(equationVariable\)

Uses \(xStat\), \(yStat\), and \(fStat\) for \(xList\), \(yList\), and \(frequencyList\), respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to \(equationVariable\) and \(RegEq\).
Chapter 20: A to Z Function and Instruction Reference

**P2Reg**

Uses \(xStat\), \(yStat\), and \(fStat\), and stores the regression equation to \(\text{RegEq}\) only.

**P3Reg**

**STAT CALC menu**

**Built-in equation variables** such as \(y1\), \(r1\), and \(xt1\) are case-sensitive. Do not use \(Y1\), \(R1\), and \(XT1\).

**P3Reg \(xList, yList, frequencyList, equationVariable\)**

Performs a third order polynomial regression using real data pairs in \(xList\) and \(yList\) and frequencies in \(frequencyList\). The regression equation is stored to \(equationVariable\), which must be a built-in equation variable such as \(y1\), \(r1\), and \(xt1\). The equation’s coefficients always are stored as a list to built-in variable \(PRegC\).

Values used for \(xList\), \(yList\), and \(frequencyList\) are stored automatically to built-in variables \(xStat\), \(yStat\), and \(fStat\), respectively. The regression equation is stored also to built-in equation variable \(\text{RegEq}\).

**P3Reg \(xList, yList, equationVariable\)**

Uses frequencies of 1.

**P3Reg \(xList, yList, frequencyList\)**

Stores the regression equation to \(\text{RegEq}\) only.

**P3Reg \(xList, yList\)**

Uses frequencies of 1, and stores the regression equation to \(\text{RegEq}\) only.

---

In **Func** graphing mode:

\[
\{1, 2, 3, 4, 5, 6\} \rightarrow L1 \quad \{1, 2, 3, 4, 5, 6\} \\
\{1, 2, 3, 4, 5, 6\} \rightarrow L2 \quad \{1, 2, 3, 4, 5, 6\} \\
P3Reg \ L1, L2, y1 \quad \text{ENTER} \\
\]

**Plot1(L1, L2, y1) ENTER Done**

**ZData** (ENTER)
Chapter 20: A to Z Function and Instruction Reference

**P3Reg** equationVariable
Uses xStat, yStat, and fStat for xList, yList, and frequencyList, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to equationVariable and RegEq.

**P3Reg**
Uses xStat, yStat, and fStat, and stores the regression equation to RegEq only.

**P4Reg** xList,yList,frequencyList,equationVariable
Performs a fourth order polynomial regression using real data pairs in xList and yList and frequencies in frequencyList. The regression equation is stored to equationVariable, which must be a built-in equation variable such as y1, r1, and xt1. The equation's coefficients always are stored as a list to built-in variable PRegC.

Values used for xList, yList, and frequencyList are stored automatically to built-in variables xStat, yStat, and fStat, respectively. The regression equation is stored also to built-in equation variable RegEq.

**P4Reg** xList,yList,equationVariable
Uses frequencies of 1.

**P4Reg** xList,yList,frequencyList
Stores the regression equation to RegEq only.

In Func graphing mode:

In Func graphing mode:

```
(L1,L2,y1 \( \leq \) L1

\{3,1,2,3,2,2,6\} \( \leq \) L2

\{4,3,1,2,3,2,4\} \( \leq \) L4

P4Reg L1,L2,y1 \( \leq \) L1

\{0.145675, \ldots \}

ZData \( \leq \) L1

P4Reg L1,L2 \( \leq \) L1
```

Plot1(1,L1,L2) \( \leq \) L1

Done
P4Reg \(xList, yList\)

Uses frequencies of 1, and stores the regression equation to \(\text{RegEq}\) only.

P4Reg \(\text{equationVariable}\)

Uses \(xStat\), \(yStat\), and \(fStat\) for \(xList\), \(yList\), and \(\text{frequencyList}\), respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to \(\text{equationVariable}\) and \(\text{RegEq}\).

P4Reg

Uses \(xStat\), \(yStat\), and \(fStat\), and stores the regression equation to \(\text{RegEq}\) only.

\[\text{Param} \quad \text{† mode screen}\]

\(\text{Param}\)

Sets parametric graphing mode.

\[\text{Pause} \quad \text{‡ program editor \ CTL menu}\]

\(\text{Pause}\)

Displays the specified argument and then suspends program execution until the user presses \(\text{ENTER}\).

Program segment:

\[
\begin{align*}
\text{Program segment:} \\
&:\text{Input} "\text{Enter } x":,x \\
&:\text{value} = x^2 - 6 \\
&:\text{Disp} "y_1 = \text{is}:" , y_1 \\
&:\text{Pause} "\text{Press ENTER to graph}" \\
&:\text{ZStd} \\
&:\end{align*}
\]

\]
Pause
Suspend program execution until the user presses \( \text{ENTER} \).

Percent: %  
MATH MISC menu

\[
\begin{array}{l|l|l}
\text{number} % \text{ or (expression)} % & 5% \text{ ENTER} & .05 \\
& 5%*200 \text{ ENTER} & 10 \\
& (10+5)*200 \text{ ENTER} & 30 \\
\end{array}
\]

pEval(  
MATH MISC menu

\[
\text{pEval(coefficientList},xValue)\]
Returns the value of a polynomial (whose coefficients are given in \text{coefficientList}) at \text{xValue}.

Evaluate \( y=2x^2+2x+3 \) at \( x=5 \):
\[
p\text{Eval}((2,2,3),5) \text{ ENTER} \quad 63
\]

PlOff  
STAT PLOT menu

\[
\text{PlOff [1,2,3]} \quad \text{PlOff 1,3 ENTER} \quad \text{Done}
\]
Deselects the specified stat plot numbers.

\[
\text{PlOff} \quad \text{PlOff ENTER} \quad \text{Done}
\]
Deselects all stat plot numbers.

PlOn  
STAT PLOT menu

\[
\text{PlOn [1,2,3]} \quad \text{PlOn 2,3 ENTER} \quad \text{Done}
\]
Selects the specified stat plot numbers, in addition to any plot numbers that are already selected.

\[
\text{PlOn} \quad \text{PlOn ENTER} \quad \text{Done}
\]
Selects all stat plot numbers.
Plot1(1, xListName, yListName, mark)
Plot1(1, xListName, yListName)

Defines and selects the plot using real data pairs in xListName and yListName.

The optional mark specifies the character used to plot the points. If you omit mark, a box is used.

mark: 1 = box (□)  2 = cross (+)  3 = dot (*)

xyLine plot
Plot1(2, xListName, yListName, mark)
Plot1(2, xListName, yListName)

Modified box plot
Plot1(3, xListName, 1 or frequencyListName, mark)
Plot1(3, xListName, 1 or frequencyListName)
Plot1(3, xListName)

Defines and selects the plot using real data points in xListName with the specified frequencies. If you omit 1 or frequencyListName, frequencies of 1 are used.

Histogram
Plot1(4, xListName, 1 or frequencyListName)
Plot1(4, xListName)

Box plot
Plot1(5, xListName, 1 or frequencyListName)
Plot1(5, xListName)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pol</strong></td>
<td>Sets polar graphing mode.</td>
</tr>
<tr>
<td>† mode screen</td>
<td></td>
</tr>
<tr>
<td>( \text{Pol} )</td>
<td>Displays <code>complexNumber</code> in polar form ((\text{magnitude}, \text{angle})), regardless of the complex number mode.</td>
</tr>
<tr>
<td>CPLX menu</td>
<td></td>
</tr>
<tr>
<td>( \text{Pol} )</td>
<td>Displays <code>complexNumber</code> in polar form ((\text{magnitude}, \text{angle})), regardless of the complex number mode.</td>
</tr>
<tr>
<td>( \text{Pol} )</td>
<td>Displays <code>complexNumber</code> in polar form ((\text{magnitude}, \text{angle})), regardless of the complex number mode.</td>
</tr>
<tr>
<td>( \text{Pol} )</td>
<td>Displays <code>complexNumber</code> in polar form ((\text{magnitude}, \text{angle})), regardless of the complex number mode.</td>
</tr>
<tr>
<td>list[Pol]</td>
<td>Returns a list, matrix, or vector in which each element of the argument is displayed in polar form.</td>
</tr>
<tr>
<td>matrix[Pol]</td>
<td></td>
</tr>
<tr>
<td>vector[Pol]</td>
<td>Returns a list, matrix, or vector in which each element of the argument is displayed in polar form.</td>
</tr>
<tr>
<td><strong>PolarC</strong></td>
<td>Sets polar complex number mode ((\text{magnitude}, \text{angle})).</td>
</tr>
<tr>
<td>† mode screen</td>
<td></td>
</tr>
<tr>
<td><strong>PolarC</strong></td>
<td>Sets polar complex number mode ((\text{magnitude}, \text{angle})).</td>
</tr>
<tr>
<td>Polar complex: ( \angle )</td>
<td>Used to enter complex numbers in polar form. The <code>angle</code> is interpreted according to the current angle mode.</td>
</tr>
<tr>
<td>( \text{PolarC} )</td>
<td></td>
</tr>
<tr>
<td><strong>PolarGC</strong></td>
<td>Displays graph coordinates in polar form.</td>
</tr>
</tbody>
</table>
### poly

**poly coefficientList**

Returns a list containing the real and complex roots of a polynomial whose coefficients are given in `coefficientList`.

\[ a_nx^n + ... + a_2x^2 + a_1x + a_0 = 0 \]

Find the roots of \(2x^3 - 8x^2 - 14x + 20 = 0\):

\[ \text{poly} \{2,-8,-14,20\} \]

\[ \{5 \ -2 \ 1\} \]

### Power: ^

**number^power or (expression)^expression**

Returns `number` raised to `power`. The arguments can be real or complex.

\[ 4^2 \]

\[ 16 \]

\[ 2^{\frac{1}{2}} \]

\[ .03125 \]

**listA^listB**

Returns a list in which each element of `listA` is raised to the power specified by the corresponding element in `listB`.

\[ \{2,3,4\}^{3,4,5} \]

\[ \{8,81,1024\} \]

**squareMatrix^power**

Returns a matrix equivalent to `squareMatrix` multiplied by itself `power` number of times, where \(0 \leq \text{power} \leq 255\). This is not the same as simply raising each element to `power`.

\[ \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}^3 \]

\[ \begin{bmatrix} 116 & 153 \\ 204 & 269 \end{bmatrix} \]

### Power of 10: 10^*

**10^power or 10^(expression)**

Returns 10 raised to `power` or `expression`, which can be real or complex.

\[ 10^{1.5} \]

\[ 31.6227766017 \]

\[ 10^{-2} \]

\[ .01 \]
10^list
Returns a list in which each element is 10 raised to the
power specified by the corresponding element in list.

\[ 10^{1.5,L2} \]
\[ b \{31.6227766017,0.1} \]

prod
LIST OPS menu
MATH MISC menu

prod list
Returns the product of all real or complex elements in
list.

\[ \text{prod} \{1,2,4,8\} \]
\[ \text{prod} (1,2,4,8) \text{ ENTER} \]
\[ 64 \]

\[ \text{prod} \{2,7,-8\} \]
\[ \text{prod} (2,7,-8) \text{ ENTER} \]
\[ -112 \]

Prompt
I/O menu
(Prompt shows on menu)

Prompts the user to enter a value for variableA, then variableB, and so on.

\[ \text{Program segment:} \]
\[ :\text{Prompt } A,B,C \]
\[ :\]

PtChg(x,y)
Reverses the point at graph coordinates (x,y).

\[ \text{PtChg(-6,2)} \]

PtOff(x,y)
Erases the point at graph coordinates (x,y).

\[ \text{PtOff(3,5)} \]

PtOn(x,y)
Draws the point at graph coordinates (x,y).

\[ \text{PtOn(3,5)} \]
PwrR

STAT CALC menu

Built-in equation variables such as y1, r1, and xt1 are case-sensitive. Do not use Y1, R1, and XT1.

PwrR xList, yList, frequencyList, equationVariable
Fits a power regression model (y=ax^b) to positive real data pairs in xList and yList, using frequencies in frequencyList. The regression equation is stored to equationVariable, which must be a built-in equation variable such as y1, r1, and xt1.

Values used for xList, yList, and frequencyList are stored automatically to built-in variables xStat, yStat, and fStat, respectively. The regression equation is stored also to built-in equation variable RegEq.

PwrR xList, yList, equationVariable
Uses frequencies of 1.

PwrR xList, yList, frequencyList
Stores the regression equation to RegEq only.

PwrR xList, yList
Uses frequencies of 1, and stores the regression equation to RegEq only.

PwrR equationVariable
Uses xStat, yStat, and fStat for xList, yList, and frequencyList, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to equationVariable and RegEq.

In Func graphing mode:

{1,2,3,4,5,6}→L1 ENTER
{1,17,21,52,75,133}→L2 ENTER
PwrR L1, L2, y1 ENTER

\[ y = 2.0392723 \times x^{0.2698944} \]
\[ r^2 = 0.7652979 \]

\[ n = 6 \]

Plot(1, L1, L2) ENTER Done
ZData ENTER

---

Chapter 20: A to Z Function and Instruction Reference 339
PwrR

Uses xStat, yStat, and fStat, and stores the regression equation to RegEq only.

PxChg(row, column)

Reverses the pixel at (row, column), where 0 ≤ row ≤ 62 and 0 ≤ column ≤ 126.

PxOff(row, column)

Erases the pixel at (row, column), where 0 ≤ row ≤ 62 and 0 ≤ column ≤ 126.

PxOn(row, column)

Draws the pixel at (row, column), where 0 ≤ row ≤ 62 and 0 ≤ column ≤ 126.

PxTest(row, column)

Returns 1 if the pixel at (row, column) is on, 0 if it is off; 0 ≤ row ≤ 62 and 0 ≤ column ≤ 126.

Assuming the pixel at (10, 95) is already on:

PxTest(10, 95) ENTER 1

rAdd(matrix, rowA, rowB)

Returns a matrix in which rowA of a real or complex matrix is added to (and stored in) rowB.

rAdd(MAT, 2, 3)

[[5 3 1] [2 0 4] [3 -1 2]] 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

<table>
<thead>
<tr>
<th><strong>Radian</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>† 2nd [MODE]</strong></td>
</tr>
<tr>
<td><strong>Radian entry:</strong></td>
</tr>
<tr>
<td><strong>MATH ANGLE menu</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>( \text{number} = ) or ((\text{expression}) = )</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>( \text{list} = )</td>
</tr>
</tbody>
</table>

**Designates a real number or expression as radians, regardless of the angle mode setting.**

**Designates each element in a real list as radians.**

### rand

<table>
<thead>
<tr>
<th><strong>rand</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MATH PROB menu</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### randBin(

| **randBin(#ofTrials,probabilityOfSuccess,#ofSimulations)** | Returns a list of random integers from a binomial distribution, where \#ofTrials \( \geq 1 \) and \( 0 \leq \text{probabilityOfSuccess} \leq 1 \). The #ofSimulations is an integer \( \geq 1 \) that specifies the number of integers returned in the list. |
| --- | 
|  | A seed value stored to **rand** also affects **randBin**. |
|  | \[ 1\|rand:randBin(5,.2,3) \text{ ENTER } \] \{0 3 2\} |

### randBin(#ofTrials,probabilityOfSuccess)

| **randBin(#ofTrials,probabilityOfSuccess)** | Returns a single random integer. |
| --- | 
|  | \[ 0\|rand:randBin(5,.2) \text{ ENTER } \] 1 |
**randInt(**

MATH PROB menu (randln shows on menu)

**randInt(###)**

- **Returns a list of random integers bound by the specified integers, lower ≤ integer ≤ upper.** The **#ofTrials** is an integer ≥ 1 that specifies the number of integers returned in the list.
- A seed value stored to rand also affects randInt().

**randInt(###)**

- **Returns a single random integer.**

**randM(###)**

MATRX OPS menu

**randM(###)**

- **Returns a rows × columns matrix filled with random one-digit integers (-9 to 9).**

**randNorm(###)**

MATH PROB menu (randN shows on menu)

**randNorm(###)**

- **Returns a list of random numbers from a normal distribution specified by mean and stdDeviation.** The **#ofTrials** is an integer ≥ 1 that specifies how many numbers are returned. Each returned number could be any real number, but most will be within the interval: [mean−3(stdDeviation), mean+3(stdDeviation)].
- A seed value stored to rand also affects randNorm().

**randNorm(###)**

- **Returns a single random number.**
### RcGDB

**RcGDB graphDataBaseName**

Restores all settings stored in `graphDataBaseName`. For a list of settings, refer to `StGDB` on page 361.

### RcPic

**RcPic pictureName**

Displays the current graph and adds the picture stored in `pictureName`.

### real

**real (complexNumber)**

Returns the real part of `complexNumber`.

- `real (real, imaginary)` returns `real`.
- `real (magnitude, angle)` returns `magnitude * cos (angle)`.

### Rec

**complexNumber Rec**

Displays `complexNumber` in rectangular form `(real, imaginary)` regardless of the complex number mode.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radian</td>
<td><code>real (3,4)</code> ENTER</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><code>real (3\angle4)</code> ENTER</td>
<td>-1.96093086259</td>
</tr>
<tr>
<td>Radian</td>
<td><code>real [-2,(3,4), (3\angle4)]</code> ENTER</td>
<td><code>-2 3 -1.96093086259</code></td>
</tr>
<tr>
<td>PolarC</td>
<td><code>√²</code> ENTER</td>
<td><code>(1.41421356237 \angle 1.570...</code></td>
</tr>
<tr>
<td></td>
<td><code>Ans\#Rec</code> ENTER</td>
<td><code>(0,1.41421356237)</code></td>
</tr>
</tbody>
</table>
**complexList→Rec**  
**complexMatrix→Rec**  
**complexVector→Rec**  

Returns a list, matrix, or vector in which each element of the argument is displayed in rectangular form.

**RectC**  
† mode screen  
Sets rectangular complex number mode (real, imaginary).

**RectGC**  
† graph format screen  
Displays graph coordinates in rectangular form.

**RectV**  
† mode screen  
Sets rectangular vector coordinate mode \([x \ y \ z]\).

**ref**  
MATRX OPS menu  

Returns the row-echelon form of a real or complex matrix. The number of columns must be greater than or equal to the number of rows.

In **PolarC** complex number mode:

\[
[(3\angle \pi/6), \sqrt{-2}] \rightarrow \text{ENTER} \\
[(3\angle 5.23598775598) \ (\ldots) \\
\text{Ans→Rec} \rightarrow \text{ENTER} \\
[(2.59807621135, 1.5)\ldots

In **RectC** complex number mode:

\[
\sqrt{2} \rightarrow \text{ENTER} \\
(0, 1.41421356237)

In **RectV** vector coordinate mode:

\[
3\times[4 \leq 5] \rightarrow \text{ENTER} \\
[3.40394622556 -11.5\ldots

**ref matrix**

\[
[[4, 5, 6][7, 8, 9]] \rightarrow \text{MAT} \rightarrow \text{ENTER} \\
[[4 \ 5 \ 6] \\
[7 \ 8 \ 9]] \rightarrow \text{ref} \rightarrow \text{MAT} \rightarrow \text{ENTER} \\
[[1 \ 1.14285714286 \ 1\ldots \\
[0 \ 1 \ 2\ldots

**ref**  
MATRX OPS menu
Repeat

‡ program editor
CTL menu
(Repea shows on menu)

:Repeat condition
:\commands-to-repeat
:End
:\commands

Executes \textit{commands-to-repeat} until \textit{condition} is true.

Return

‡ program editor
CTL menu
(Retur shows on menu)

\textbf{Return}

In a subroutine, exits the subroutine and returns to the calling program. In the main program, stops execution and returns to the home screen.

RK

‡ graph format screen
(scroll down to second screen)

\textbf{RK}

In \textit{DifEq} graphing mode, uses an algorithm based on the Runge-Kutta method to solve differential equations. Typically, RK is more accurate than Euler but takes longer to find the solutions.
**rnorm**

MATRX MATH menu

Returns the row norm of a real or complex matrix. For each row, rnorm sums the absolute values (magnitudes of complex elements) of all elements on that row. The returned value is the largest of the sums.

**rnorm vector**

Returns the largest absolute value (or magnitude) in a real or complex vector.

---

**Root:** \( \sqrt{x} \)

MATH MISC menu

\( x^{\text{throot}}(\text{number or expression}) \)

Returns the \( x^{\text{throot}} \) of \( \text{number or expression} \). The arguments can be real or complex.

\( x^{\text{throot}}(\text{list}) \)

Returns a list in which each element is the \( x^{\text{throot}} \) of the corresponding element in \( \text{list} \).

\( x^{\text{throotList}}(\text{list}) \)

Returns a list in which each element is the root specified by the corresponding elements in \( x^{\text{throotList}} \) and \( \text{list} \).
rotL

BASE BIT menu

rotL integer

Returns a real integer with bits rotated one to the left. Internally, integer is represented as a 16-bit binary number. When the bits are rotated left, the leftmost bit rotates to the rightmost bit.

\[ \text{rotL} \quad 0000111100001111_b = 0001111000011110_b \]

rotL is not valid in Dec number base mode. To enter hexadecimal numbers A through F, use the BASE A-F menu. Do not use \[\text{ALPHA}\] to type a letter.

In Bin number base mode:

\[ \text{rotL} \quad 00001111100001111 \quad \text{ENTER} \]

1111000011110b

Leading zeros are not displayed.

rotR

BASE BIT menu

rotR integer

Returns a real integer with bits rotated one to the right. Internally, integer is represented as a 16-bit binary number. When the bits are rotated right, the rightmost bit rotates to the leftmost bit.

\[ \text{rotR} \quad 0000111100001111_b = 1000011110000111_b \]

rotR is not valid in Dec number base mode. To enter hexadecimal numbers A through F, use the BASE A-F menu. Do not use \[\text{ALPHA}\] to type a letter.

In Bin number base mode:

\[ \text{rotR} \quad 00001111100001111 \quad \text{ENTER} \]

1000011110000111b
### round(

**MATH NUM menu**

**round**(number)

Returns a real or complex number rounded to the specified #ofDecimals (0 to 11). If #ofDecimals is omitted, number is rounded to 12 decimal places.

**round**(number, #ofDecimals)

Returns a real or complex number rounded to the specified #ofDecimals (0 to 11). If #ofDecimals is omitted, number is rounded to 12 decimal places.

**round**(list, #ofDecimals)

Returns a list, matrix, or vector in which each element is the rounded value of the corresponding element in the argument. #ofDecimals is optional.

**round**(matrix, #ofDecimals)

Returns a list, matrix, or vector in which each element is the rounded value of the corresponding element in the argument. #ofDecimals is optional.

**round**(vector, #ofDecimals)

Returns a list, matrix, or vector in which each element is the rounded value of the corresponding element in the argument. #ofDecimals is optional.

**Example:**

```
round(π, 4) enter 3.1416
round(π/4, 4) enter .7854
round(π/4) enter .785398163397
```

**Example:**

```
round([ln 5, ln 3] [π, e^1]), 2
```

### rref

**MATRX OPS menu**

**rref** matrix

Returns the reduced row-echelon form of a real or complex matrix. The number of columns must be greater than or equal to the number of rows.

**Example:**

```
[[4, 5, 6] [7, 8, 9]] \rightarrow MAT enter
```

```
[[4 5 6] [7 8 9]]
```

### rSwap

**MATRX OPS menu**

**rSwap**(matrix, rowA, rowB)

Returns a matrix with rowA of a real or complex matrix swapped with rowB.

**Example:**

```
rSwap(MAT, 2, 3)
```

```
[[5 3 1] [2 0 4] [3 -1 2]]
```

```
[[5 3 1] [3 -1 2] [2 0 4]]
```
### Scatter

† STAT DRAW menu  
(Scatte shows on menu)

**Scatter** \(xList, yList\)

Draws a scatter plot on the current graph, using the real data pairs in \(xList\) and \(yList\).

\[
\begin{align*}
\{ -9, -6, -4, -1, 2, 5, 7, 10 \} & \rightarrow XL \quad \text{ENTER} \\
\{ -9, -6, -4, -1, 2, 5, 7, 1 \ldots \} & \rightarrow XL \quad \text{ENTER} \\
\{ -7, -6, -2, 1, 3, 6, 7, 9 \} & \rightarrow YL \quad \text{ENTER} \\
\{ -7, -6, -2, 1, 3, 6, 7, 9 \} & \rightarrow XL, YL \quad \text{ENTER}
\end{align*}
\]

\(ZStd: \text{Scatter XL, YL} \quad \text{ENTER}\)

**Sci**

† mode screen

**Sci**

Sets scientific notation display mode.

In **Sci** notation mode:

\[
123456789 \quad \text{ENTER} \quad 1.23456789 \times 8
\]

In **Normal** notation mode:

\[
123456789 \quad \text{ENTER} \quad 123456789
\]
Select(\(x_{\text{ListName}}, y_{\text{ListName}}\))

Select(\(x_{\text{ListName}}, y_{\text{ListName}}\)) displays the current graph screen and starts an interactive session during which you select a range of data points.

- Move the cursor to the leftmost (left bound) point of the range you want to select and press \(\text{ENTER} \).
- Then move the cursor to the rightmost (right bound) point of the range you want to select and press \(\text{ENTER} \).

A new stat plot of \(x_{\text{ListName}}\) and \(y_{\text{ListName}}\) replaces the plot from which you selected the points.

Send(listName)

Sends the contents of listName to the CBL or CBR System.

\{(1,2,3,4,5)\} \rightarrow \text{L1} \quad \text{ENTER}
\{-9,6,-4,-1,2,5,7,10\} \rightarrow \text{L1} \quad \text{ENTER}
\{-7,-6,-2,1,3,6,7,9\} \rightarrow \text{L2} \quad \text{ENTER}
\text{Plot1(1,1),L2):ZStd} \quad \text{ENTER}

After the graph is displayed:

Select(L10,L20) \quad \text{ENTER}

Move the cursor to point (2,3) and press \(\text{ENTER} \). Then move to (10,9) and press \(\text{ENTER} \).
**seq**

**MATH MISC menu**

seq(expression, variable, begin, end, step)

Returns a list containing a sequence of numbers created by evaluating expression from variable = begin to variable = end in increments of step.

seq(expression, variable, begin, end)

Uses a step of 1.

**SeqG**

† graph format screen

Sets sequential graphing format, in which selected functions are plotted one at a time.

**SetLEdit**

LIST OPS menu

(SetLE shows on menu)

SetLEdit column1ListName, ..., column20ListName

Removes all lists from the list editor and then stores one or more ListNames in the specified order, starting with column 1.

SetLEdit

Removes all lists from the list editor and stores built-in lists xStat, yStat, and fStat in columns 1 through 3, respectively.
### Shade(\(lowerFunc, upperFunc, x\text{Left}, x\text{Right}, pattern, patternRes\))

<table>
<thead>
<tr>
<th>(Shade(lowerFunc, upperFunc, x\text{Left}, x\text{Right}, pattern, patternRes))</th>
<th>In Func graphing mode: (Shade(x-2, x^3-8, x, -5, 1, 2, 3)) [ENTER]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draws (lowerFunc) and (upperFunc) in terms of (x) on the current graph and shades the area bounded by (lowerFunc, upperFunc, x\text{Left},) and (x\text{Right}). The shading style is determined by (pattern) (1 through 4) and (patternRes) (1 through 8).</td>
<td><img src="Shade.png" alt="" /></td>
</tr>
</tbody>
</table>

**pattern:**
- \(1\) = vertical (default)
- \(2\) = horizontal
- \(3\) = negative-slope 45°
- \(4\) = positive-slope 45°

**patternRes** (resolution):
- \(1\) = every pixel (default)
- \(2\) = every 2nd pixel
- \(3\) = every 3rd pixel
- \(4\) = every 4th pixel
- \(5\) = every 5th pixel
- \(6\) = every 6th pixel
- \(7\) = every 7th pixel
- \(8\) = every 8th pixel

### Shade(\(lowerFunc, upperFunc\))

Sets \(x\text{Left}\) and \(x\text{Right}\) to \(x\text{Min}\) and \(x\text{Max}\), respectively, and uses the defaults for \(pattern\) and \(patternRes\).
**shftL**

**BASE BIT menu**

**shftL** integer

Returns a real integer with bits shifted one to the left. Internally, integer is represented as a 16-bit binary number. When the bits are shifted left, the leftmost bit is dropped and 0 is used as the rightmost bit.

\[
\text{shftL } 0000111100001111 = 0001111000011110
\]

**shftL** is not valid in **Dec** number base mode. To enter hexadecimal numbers A through F, use the BASE A-F menu. Do not use **ALPHA** to type a letter.

**In Bin** number base mode:

\[
\text{shftL } 0000111100001111 \text{ ENTER} \\
111000011111b
\]

Leading zeros are not displayed.

**shftR**

**BASE BIT menu**

**shftR** integer

Returns a real integer with bits shifted one to the right. Internally, integer is represented as a 16-bit binary number. When the bits are shifted right, the rightmost bit is dropped and 0 is used as the leftmost bit.

\[
\text{shftR } 0000111100001111 = 0000111100001110
\]

**shftR** is not valid in **Dec** number base mode. To enter hexadecimal numbers A through F, use the BASE A-F menu. Do not use **ALPHA** to type a letter.

**In Bin** number base mode:

\[
\text{shftR } 0000111100001111 \text{ ENTER} \\
111100000111b
\]

Leading zeros are not displayed.
**Chapter 20: A to Z Function and Instruction Reference**

**ShwSt**

Displays the results of the most recent stat calculation.

**sign**

* MATH NUM menu

**sign number** or **sign (expression)**

Returns `-1` if the argument is `< 0`, `1` if it is `> 0`, or `0` if it is `= 0`. The argument must be real.

**sign list**

Returns a list in which each element is `-1`, `1`, or `0` to indicate the sign of the corresponding element in `list`.

**SimulG**

† graph format screen

Sets simultaneous graphing format, in which all selected functions are plotted at the same time.

**simult(squareMatrix,vector)**

Returns a vector containing the solutions to a system of simultaneous linear equations that have the form:

\[
\begin{align*}
\text{a11}x_1 + \text{a12}x_2 + \ldots &= \text{b1} \\
\text{a21}x_1 + \text{a22}x_2 + \ldots &= \text{b2} \\
\text{a31}x_1 + \text{a32}x_2 + \ldots &= \text{b3}
\end{align*}
\]

Each row in `squareMatrix` contains the `a` coefficients of an equation, and `vector` contains the `b` constants.

Solve the following for `x` and `y`:

\[
\begin{align*}
3x - 4y &= 7 \\
x + 6y &= 6
\end{align*}
\]

The solution is `x=3` and `y=.5`.
\textbf{sin}\textsuperscript{-1} \textsuperscript{number or \textit{sin}^{-1} (expression)}

Returns the arcsine of \textit{number} or \textit{expression}, which can be real or complex.

\textbf{sin \textit{list}}

Returns a list in which each element is the arcsine of the corresponding element in \textit{list}.

\textbf{sin} \textsuperscript{-1} \textit{list}

Returns a list in which each element is the arcsine of the corresponding element in \textit{list}. 

The squareMatrix cannot have repeated eigenvalues.
### sinh

**MATH HYP menu**

<table>
<thead>
<tr>
<th>sinh number or sinh (expression)</th>
<th>sinh 1.2 ENTER</th>
<th>1.50946135541</th>
</tr>
</thead>
</table>

Returns the hyperbolic sine of `number` or `expression`, which can be real or complex.

<table>
<thead>
<tr>
<th>sinh list</th>
<th>sinh (0,1.2) ENTER</th>
<th>{0 1.50946135541 }</th>
</tr>
</thead>
</table>

Returns a list in which each element is the hyperbolic sine of the corresponding element in `list`.

### sinh⁻¹

**MATH HYP menu**

<table>
<thead>
<tr>
<th>sinh⁻¹ number or sinh⁻¹(expression)</th>
<th>sinh⁻¹ 1 ENTER</th>
<th>.88137358702</th>
</tr>
</thead>
</table>

Returns the inverse hyperbolic sine of `number` or `expression`, which can be real or complex.

<table>
<thead>
<tr>
<th>sinh⁻¹ list</th>
<th>sinh⁻¹ (1,2,1,3) ENTER</th>
<th>{.88137358702 1.4874...}</th>
</tr>
</thead>
</table>

Returns a list in which each element is the inverse hyperbolic sine of the corresponding element in `list`.
SinR

STAT CALC menu

Built-in equation variables such as y1, r1, and xt1 are case-sensitive. Do not use Y1, R1, and XT1.

Attempts to fit a sinusoidal regression model (y = a sin(bx + c) + d) to real data pairs in xList and yList, using an optional estimated period. The regression equation is stored to equationVariable, which must be a built-in equation variable such as y1, r1, and xt1. The equation’s coefficients always are stored as a list to built-in variable PRegC.

iterations is optional; it specifies the maximum number of times (1 through 16) the TI-86 will attempt to find a solution. If omitted, 8 is used. Typically, larger values result in better accuracy but longer execution times, and vice versa.

If you omit the optional period, the difference between values in xList should be equal and in sequential order. If you specify period, the differences between x values can be unequal.

Values used for xList and yList are stored automatically to built-in variables xStat and yStat, respectively. The regression equation is stored also to built-in equation variable RegEq.

The output of SinR is always in radians, regardless of the angle mode setting.

Stores the regression equation to RegEq only.
SinR \[\text{iterations,} \text{equationVariable}\]

Uses \text{xStat} and \text{yStat} for \text{xList} and \text{yList}, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs. The regression equation is stored to \text{equationVariable} and \text{RegEq}.

SinR \[\text{iterations}\]

Uses \text{xStat} and \text{yStat}, and stores the regression equation to \text{RegEq} only.

**SlpFld**

† graph format screen

(Scroll down to second screen)

**Solver**(equation,variable,guess,\{lower,upper\})

Solves equation for variable, given an initial guess and lower and upper bounds within which the solution is sought. equation can be an expression, which is assumed to equal 0.

If \( y = 5 \), solve \( x^3 + y^2 = 125 \) for \( x \). You guess the solution is approximately 4:

\[
5 \rightarrow \text{y [ENTER]}
\]

\[
\text{Solver}(x^3+y^2=125,x,4) \quad \text{[ENTER]} \quad \text{Done}
\]

\[
x \quad \text{[ENTER]} \quad 4.64158883361
\]

**Solver**(equation,variable,guess)

Uses \(-1e99\) and \(1e99\) for upper and lower, respectively.

**Solver**(equation,variable,\{guessLower,guessUpper\})

Uses the secant line between \text{guessLower} and \text{guessUpper} to start the search. \text{Solver} will still search for a solution outside of this range.
sortA  
LIST OPS menu

SortA list

Returns a list in which the real or complex elements of
list are sorted in ascending order.

{5,8,-4,0,-6}⇒L1 ENTER
{5 8 -4 0 -6}
SortA L1 ENTER
{-6 -4 0 5 8}

sortD  
LIST OPS menu

SortD list

Returns a list in which the real or complex elements of
list are sorted in descending order.

{5,8,-4,0,-6}⇒L1 ENTER
{5 8 -4 0 -6}
SortD L1 ENTER
{8 5 0 -4 -6}

Sortx xListName,yListName,frequencyListName

In ascending order of x elements, sorts real or complex
x and y data pairs and, optionally, their frequencies in
xListName, yListName, and frequencyListName. The
lists' contents are updated to reflect the changes.

Sortx xListName,yListName

Uses built-in variables xStat and yStat for xListName
and yListName, respectively. These built-in variables
must contain valid data of the same dimension;
otherwise, an error occurs.

Sorty xListName,yListName,frequencyListName

In ascending order of y elements, sorts real or complex
x and y data pairs and, optionally, their frequencies in
xListName, yListName, and frequencyListName. The
lists' contents are updated to reflect the changes.
Sorty

Uses built-in variables $x_{\text{Stat}}$ and $y_{\text{Stat}}$ for $x_{\text{ListName}}$ and $y_{\text{ListName}}$, respectively. These built-in variables must contain valid data of the same dimension; otherwise, an error occurs.

$\text{Sph}$

VECTR OPS menu

Displays a 2- or 3-element vector as spherical coordinates in $[r \leq \theta \leq 0]$ or $[r \leq \theta \leq \phi]$ form, respectively, even if the display mode is not set for spherical ($\text{SphereV}$).

$\text{SphereV}$

Sets spherical vector coordinate mode $[r \leq \theta \leq \phi]$.

Square: $^2$

Returns a real or complex argument multiplied by itself. To square a negative number, enclose it in parentheses. A $\text{squareMatrix}$ multiplied by itself is not the same as simply squaring each element.

Square root: $\sqrt{}$

Returns the square root of $\text{number}$ or $\text{expression}$, which can be real or complex.
\( \sqrt{\text{list}} \)

Returns a list in which element is the square root of the corresponding element in \( \text{list} \).

In Rect\( \text{C} \) complex number mode:
\[
\sqrt{\{-2, 2.5\}} \quad \text{ENTER} \quad (0, 1.41421356237) \ldots
\]

\( \text{St} \)\text{Eq} ( \text{stringVariable}, \text{equationVariable} )

| STRING menu |

\( \text{St} \)\text{Eq} ( \text{stringVariable}, \text{equationVariable} )

Converts \( \text{stringVariable} \) to a number, expression, or equation, and stores it in \( \text{equationVariable} \).

To convert the string and retain the same variable name, you can set \( \text{equationVariable} \) equal to \( \text{stringVariable} \).

If you use \text{Input} instead of \text{InpSt} here, the entered expression is evaluated at the current value of \( x \) and the result (not the expression) is stored.

\( \text{StGDB} \)

| GRAPH menu |

\( \text{StGDB} \) \( \text{graphDataBaseName} \)

Creates a graph database (GDB) variable that contains the current:
- Graphing mode, graph format settings, and range variables.
- Functions in the equation editor, whether they are selected, and their graph styles.

To restore the database and recreate the graph, use \( \text{RcGDB} \) (page 343).
Stop

‡ program editor
CTL menu

Ends program execution and returns to the home screen.

Program segment:

Use N=999, not N=999.

Store to variable: →

number→variable or (expression)→variable
string→variable
list→variable
vector→variable
matrix→variable

Stores the specified argument to variable.

StPic

† GRAPH menu

Stores a picture of the current graph screen to pictureName.

StReg(

STAT CALC menu

Stores the most recently calculated regression equation to variable. This lets you save a regression equation by storing it to any variable as opposed to a built-in equation variable.

Use N=999, not N=999.

Example:

{1,2,3,4,5}→L1
{1,2,3,4,5}→L2
ExpR L1,L2:StReg(EQ)

Rcl EQ
.41138948780597\times 4.7879605684671\times ^ x

Done

Rcc EQ
.41138948780597\times 4.7879605684671\times ^ x

Done
String entry: "

STNG menu
‡ program editor
I/O menu

"string"

Defines a string. When you display a string, it is left-
justified on the screen.

Strings are interpreted as text characters, not numbers.
For example, you cannot perform a calculation with
strings such as "4" or "A+8". To convert between string
variables and equation variables, use EqSt and StEq
as described on pages 290 and 361, respectively.

sub(

STNG menu

sub(string,begin,length)

Returns a new string that is a subset of string, starting
at character number begin and continuing for the
specified length.

Subtraction: -

numberA - numberB

Returns the value of numberB subtracted from
numberA. The arguments can be real or complex.

list - number

Returns a list in which number is subtracted from each
element of list. The arguments can be real or complex.
listA - listB
matrixA - matrixB
vectorA - vectorB

Returns a list, matrix, or vector that is the result of each element in the second argument subtracted from the corresponding element in the first argument. The two real or complex arguments must have the same dimension.

\[
\begin{bmatrix}
5,7,9 \\
11,13,15
\end{bmatrix} - \begin{bmatrix}
4,5,6 \\
7,8,9
\end{bmatrix} = \begin{bmatrix}
1,2,3 \\
4,5,6
\end{bmatrix}
\]

sum

MATH MISC menu
LIST OPS menu

\textbf{sum list}

Returns the sum of all real or complex elements in \textit{list}.

\[
\text{sum } \{1,2,4,8\} \quad \text{ENTER} \quad 15
\]

\[
\text{sum } \{2,7,-8,0\} \quad \text{ENTER} \quad 1
\]

tan

MATH ANGLE menu

\textbf{tan angle or tan (expression)}

Returns the tangent of \textit{angle} or \textit{expression}, which can be real or complex.

An angle is interpreted as degrees or radians according to the current angle mode. In any angle mode, you can designate an angle as degrees or radians by using the \(^\circ\) or \(\text{r}\) designator, respectively, from the MATH ANGLE menu.

\textbf{tan list}

Returns a list in which each element is the tangent of the corresponding element in \textit{list}.

\[
\tan \left( \frac{\pi}{4} \right) \quad \text{ENTER} \quad 1
\]

\[
\tan \left( \frac{\pi}{4} \right) \quad \text{ENTER} \quad 0
\]

In Radian angle mode:

\[
\tan 45^\circ \quad \text{ENTER} \quad 1
\]

\[
\tan 45 \quad \text{ENTER} \quad 0
\]

In Degree angle mode:

\[
\tan \left( \frac{\pi}{4} \times \text{r} \right) \quad \text{ENTER} \quad 1
\]

\[
\tan \left( \frac{\pi}{4} \times \text{r} \right) \quad \text{ENTER} \quad 1
\]

In Degree angle mode:

\[
\tan \{0,45,60\} \quad \text{ENTER} \quad 0 \ 1 \ 1.73205080757
\]
\[ \text{tan}^{-1} \]

\[ \text{tan}^{-1} \text{ number or } \text{tan}^{-1} \text{ (expression)} \]

Returns the arctangent of \text{number} or \text{expression}, which can be real or complex.

\[ \text{In Radian angle mode:} \]
\[ \text{tan}^{-1} \text{ .5} \text{ ENTER} \]
\[ .463647609001 \]

\[ \text{In Degree angle mode:} \]
\[ \text{tan}^{-1} \text{ 1} \text{ ENTER} \]
\[ 45 \]

\[ \text{In Radian angle mode:} \]
\[ \text{tan}^{-1} \{0,.2,.5\} \text{ ENTER} \]
\[ \{0 .1973955985 .463647609001\} \]

\[ \text{tan}^{-1} \text{ list} \]

Returns a list in which each element is the arctangent of the corresponding element in \text{list}.

\[ \text{tanh} \]

\[ \text{tanh \ number or } \text{tanh \ (expression)} \]

Returns the hyperbolic tangent of \text{number} or \text{expression}, which can be real or complex.

\[ \text{tanh \} 1.2 \text{ ENTER} \]
\[ .833654607012 \]

\[ \text{tanh \ list} \]

Returns a list in which each element is the hyperbolic tangent of the corresponding element in \text{list}.

\[ \text{tanh}^{-1} \]

\[ \text{tanh}^{-1} \text{ number or } \text{tanh}^{-1} \text{ (expression)} \]

Returns the inverse hyperbolic tangent of \text{number} or \text{expression}, which can be real or complex.

\[ \text{tanh}^{-1} \text{ 0} \text{ ENTER} \]
\[ 0 \]

\[ \text{tanh}^{-1} \text{ list} \]

Returns a list in which each element is the inverse hyperbolic tangent of the corresponding element in \text{list}.

\[ \text{In RectC complex number mode:} \]
\[ \text{tanh}^{-1} \text{ (0,2.1) ENTER} \]
\[ \{(0,0) (.518045965845,1.10714871779)\} \]
Chapter 20: A to Z Function and Instruction Reference

**TanLn(**

**GRAPH DRAW menu**

`TanLn(expression,xValue)`

Draws `expression` on the current graph and then draws a tangent line at `xValue`.

In **Func** graphing mode and **Radian** angle mode:

```
ZTrig:TanLn(cos x,π/4) ENTER
```

**Text(**

† **GRAPH DRAW menu**

`Text(row,column,string)`

Writes a text `string` on the current graph beginning at pixel `(row,column)`, where `0 ≤ row ≤ 57` and `0 ≤ column ≤ 123`.

Text at the bottom of the graph may be covered by a displayed menu. To remove the menu, press [CLEAR].

Program segment in **Func** graphing mode and a **ZStd** graph screen:

```
:yl=x sin x
:Text(0,70,"yl=x sin x")
```

When executed:

Refer to syntax information for **If**, beginning on page 305. See the **If:Then:End** and **If:Then:Else:End** syntax.
Trace

† GRAPH menu
Displays the current graph and lets the user trace a function. From a program, press [ENTER] to stop tracing and continue with the program.

Transpose: ↑
MATRX MATH menu
Returns a transposed real or complex matrix in which element row,column is swapped with element column,row of matrix. For example:

\[
\begin{bmatrix}
  a & b \\
  c & d
\end{bmatrix}^T \rightarrow \begin{bmatrix}
  a & c \\
  b & d
\end{bmatrix}
\]

For complex matrices, the complex conjugate of each element is taken.

\[
\begin{bmatrix}
  [1,2][3,4]\rightarrow MATA \rightarrow [1,2] \\
  [3,4]\rightarrow 1,3 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  [1,2,3][4,5,6][7,8,9]\rightarrow MATB \rightarrow [1,2,3] \\
  [4,5,6] \rightarrow [1,4,7] \\
  [7,8,9] \rightarrow [2,5,8] \\
\end{bmatrix}
\]

In RectC complex number mode:

\[
\begin{bmatrix}
  [(1,2),(1,1)][(3,2),(4,3)]\rightarrow MATC \rightarrow [(1,2),(1,1)] \\
  [(3,2),(4,3)] \rightarrow [(1,-2),(3,-2)] \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  [(1,-2),(3,-2)] \rightarrow [(1,-1),(4,-3)]
\end{bmatrix}
\]
TwoVar

STAT CALC menu
(TwoVa shows on menu)

TwoVar \( xList, yList, frequencyList \)

Performs two-variable statistical analysis on the real data pairs in \( xList \) and \( yList \), using the frequencies in \( frequencyList \).

Values used for \( xList \), \( yList \), and \( frequencyList \) are stored automatically to the built-in variables \( xStat \), \( yStat \), and \( fStat \), respectively.

TwoVar \( xList, yList \)

Uses frequencies of 1.

TwoVar

Uses \( xStat \), \( yStat \), and \( fStat \) for \( xList \), \( yList \), and \( frequencyList \). These built-in variables must contain valid data of the same dimension; otherwise, an error occurs.

unitV

VECTR MATH menu

unitV \( \text{vector} \)

Returns a unit vector of a real or complex \( \text{vector} \), where:

\[
\text{unitV} \; [a, b, c] \; \text{returns} \; \left[ \frac{a}{\text{norm}}, \frac{b}{\text{norm}}, \frac{c}{\text{norm}} \right]
\]

and

\( \text{norm} = \sqrt{a^2 + b^2 + c^2} \).
### vc®li

**LIST OPS menu**

**VECTR OPS menu**

**vc®li vector**

Returns a real or complex `vector` converted to a list.

\[
\text{vc®li [2,7,-8,0] ENTER (2 7 -8 0)}
\]

\[
(\text{vc®li [2,7,-8,0]])^2 \text{ ENTER} (2 7 -8 0)
\]

- **Example:**
  \[
  (\text{vc®li [2,7,-8,0]})^2 \text{ ENTER} \]

### Vector entry:

- **[ ]**
  - **[2nd] [ ]** and **[2nd] [ ]**

\[
\text{Vert xValue}
\]

Draws a vertical line on the current graph at `xValue`.

- **Example:**
  \[
  \text{Vert -4.5 ENTER}
  \]

### Vert

† **GRAPH DRAW menu**

- **[ ]**
  - **[2nd] [ ]** and **[2nd] [ ]**

\[
\text{While condition commands-while-true :End command}
\]

Executes `commands-while-true` as long as `condition` is true.

- **Program segment:**
  
  ```plaintext
  :While J ≥ 20
  :TEMP+1/J→TEMP
  :J+1→J
  :End
  :Disp "Reciprocal sums to 20",TEMP
  :;
  ```
**xor**

**BASE BOOL menu**

Compares two real integers bit by bit. Internally, both integers are converted to binary. When corresponding bits are compared, the result is 1 if either bit (but not both) is 1; the result is 0 if both bits are 0 or both bits are 1. The returned value is the sum of the bit results.

For example, \(78 \text{ xor } 23 = 89\).

\[
\begin{align*}
78 & = 1001110_b \\
23 & = 0010111_b \\
1011001_b & = 89
\end{align*}
\]

You can enter real numbers instead of integers, but they are truncated automatically before the comparison.

**xyline**

**† STAT DRAW menu**

Draws a line plot on the current graph, using the real data pairs in \(xList\) and \(yList\).

\[
\begin{align*}
\text{xyline} & \quad \text{xList,yList} \\
\text{xyline} & \quad \text{Uses the data in built-in variables xStat and yStat. These variables must contain valid data of the same dimension; otherwise, an error occurs.}
\end{align*}
\]
ZData
† GRAPH ZOOM menu

ZData

Adjusts the window variable values based on the currently defined statistical plots so that all stat data points will be plotted, and then updates the graph screen.

In Func graphing mode:

\begin{align*}
\{1,2,3,4\} & \rightarrow XL \quad \text{ENTER} \\
\{2,3,4,5\} & \rightarrow YL \quad \text{ENTER} \\
\text{Plot1}(1,XL,YL) & \quad \text{ENTER} \quad \text{Done} \\
\end{align*}
### ZDecm

† GRAPH ZOOM menu

<table>
<thead>
<tr>
<th>ZDecm</th>
<th>ZDecm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the window variable values such that ( \Delta x = \Delta y = 1 ), and then updates the graph screen with the origin centered on the screen.</td>
<td></td>
</tr>
<tr>
<td>( x_{\text{Min}} = 6.3 ) \quad ( y_{\text{Min}} = 3.1 ) \quad ( x_{\text{Max}} = 6.3 ) \quad ( y_{\text{Max}} = 3.1 ) \quad ( x_{\text{Scl}} = 1 ) \quad ( y_{\text{Scl}} = 1 )</td>
<td></td>
</tr>
</tbody>
</table>

One of the benefits of **ZDecm** is that you can trace in .1 increments.

In **Func** graphing mode:

\[ y = x \sin x \]

If you trace the graph above, \( x \) values start at 0 and increment by .1587301587.

\[ Z_{\text{Decm}} \]

If you trace this graph, the \( x \) values increment by .1.
**ZFit**

† GRAPH ZOOM menu

Recalculates yMin and yMax to include the minimum and maximum y values of the selected functions between the current xMin and xMax, and then updates the graph screen.

This does not affect xMin and xMax.

**ZIn**

† GRAPH ZOOM menu

Zooms in on the part of the graph centered around the current cursor location.

Zoom factors are set by the values of built-in variables xFact and yFact; the default is 4 for both factors.
ZInt

† GRAPH ZOOM menu

Sets the window variable values so that each pixel is an integer in all directions ($\Delta x=\Delta y=1$), sets $x$Scl=yScl=10, and then updates the graph screen.

The current cursor location becomes the center of the new graph.

One of the benefits of ZInt is that you can trace in whole number increments.

In Func graphing mode:

\[ y_1=\text{der1}(x^{1/20},x) \]

ZInt

Done

If you trace the graph above, $x$ values start at 0 and increment by $0.1587301587$.

ZInt

If you trace this graph, $x$ values increment by 1.
ZOut
† GRAPH ZOOM menu

ZOut

Zooms out to display more of the graph, centered around the current cursor location.

Zoom factors are set by the values of built-in variables \( xFact \) and \( yFact \); the default is 4 for both factors.

In Func graphing mode:

\[
y_1 = x \sin x \quad \text{ENTER} \\
\text{ZStd ENTER} \\
\text{ZOut ENTER}
\]

ZPrev
† GRAPH ZOOM menu

ZPrev

Replots the graph using the window variable values of the graph that was displayed before you executed the previous ZOOM instruction.
Chapter 20: A to Z Function and Instruction Reference

**ZRcl**
- **ZRcl**
  - Sets the window variables to values stored previously in the user-defined zoom-window variables, and then updates the graph screen.
  - To set user-defined zoom-window variables, either:
    - Press `ZSTO` to store the current graph’s window variables.
    - **ZSqr**
  - Sets the window variable values to produce “square” pixels where $\Delta x = \Delta y$, and then updates the graph screen.
  - The center of the current graph (not necessarily the axes intersection) becomes the center of the new graph.
  - In other types of zooms, squares may look like rectangles and circles may look like ovals. Use **ZSqr** for a more accurate shape.
ZStd

Sets the window variables to the standard default values, and then updates the graph screen.

**Func** graphing mode:
- \( x_{\text{Min}} = -10 \)
- \( y_{\text{Min}} = -10 \)
- \( x_{\text{Max}} = 10 \)
- \( y_{\text{Max}} = 10 \)
- \( x_{\text{Scl}} = 1 \)
- \( y_{\text{Scl}} = 1 \)

**Pol** graphing mode:
- \( \theta_{\text{Min}} = 0 \)
- \( \theta_{\text{Max}} = 6.28318530718 \) (\( 2\pi \))
- \( x_{\text{Min}} = -10 \)
- \( x_{\text{Max}} = 10 \)
- \( y_{\text{Min}} = -10 \)
- \( y_{\text{Max}} = 10 \)
- \( x_{\text{Scl}} = 1 \)
- \( y_{\text{Scl}} = 1 \)

**Param** graphing mode:
- \( t_{\text{Min}} = 0 \)
- \( t_{\text{Max}} = 6.28318530718 \) (\( 2\pi \))
- \( x_{\text{Min}} = -10 \)
- \( x_{\text{Max}} = 10 \)
- \( y_{\text{Min}} = -10 \)
- \( y_{\text{Max}} = 10 \)
- \( x_{\text{Scl}} = 1 \)
- \( y_{\text{Scl}} = 1 \)

**DifEq** graphing mode:
- \( t_{\text{Min}} = 0 \)
- \( t_{\text{Max}} = 6.28318530718 \) (\( 2\pi \))
- \( x_{\text{Min}} = -10 \)
- \( x_{\text{Max}} = 10 \)
- \( y_{\text{Min}} = -10 \)
- \( y_{\text{Max}} = 10 \)
- \( x_{\text{Scl}} = 1 \)
- \( y_{\text{Scl}} = 1 \)
- \( t_{\text{Plot}} = 0 \)
- \( \text{difTol} = .001 \)
ZTrig
† GRAPH ZOOM menu
Sets the window variables to preset values appropriate for plotting trig functions in Radian angle mode (Δx=π/24), and then updates the graph screen.

- xMin= 8.24668071567
- xMax= 8.24668071567
- xScl= 1.5707963267949 (π/2)
- yMin= -4
- yMax= 4
- yScl= 1

In Func graphing mode:
y1= sin x

Done

ZStd [ENTER]

ZTrig [ENTER]
TI-86 Menu Map.............................................................. 380
Handling a Difficulty ........................................................ 392
Error Conditions............................................................... 393
Equation Operating System (EOS™) ................................ 397
TOL (The Tolerance Editor) ................................................. 398
Computational Accuracy.................................................. 399
Support and Service Information ..................................... 400
Warranty Information ...................................................... 402
TI-86 Menu Map

This section presents the TI-86 menus as they appear on the TI-86 keyboard, starting at the top. If a menu has items that display other menus, the other menus follow directly below the main menu. In the program editor, the appearance of some menus changes slightly. The menu map omits user-created-name menus, such as the LIST NAMES and CONS USER menus.

LINK Menu 2nd [LINK]
SEND | RECV | SND85

LINK SEND Menu 2nd [LINK] [F1]
BCKUP | PRGM | MATRX | GDB | ALL | LIST | VECTR | REAL | CPLX | EQU | CONS | PIC | WIND | STRNG

SEND BCKUP Menu 2nd [LINK] [F1] [F1]
XMIT

LINK SEND Selection Screen Menu 2nd [LINK] [F1] data type
XMIT | SELCT | ALL+ | ALL-

LINK SND85 Menu 2nd [LINK] [F3]
MATRX | LIST | VECTR | REAL | CPLX | CONS | PIC | STRNG

GRAPH Menu [GRAPH] in Func graphing mode
y(x)= | WIND | ZOOM | TRACE | GRAPH | MATH | DRAW | FORMAT | STGDB | RCGB | EVAL | STPIC | RCPIC

The link menus are not available in the program editor.

In the program editor, DrEqu is available as a GRAPH menu item.
GRAPH Menu  (GRAPH) in Pol graphing mode

\[ r(q) = \text{WIND ZOOM TRACE GRAPH} \]

GRAPH Menu  (GRAPH) in Param graphing mode

\[ E(t) = \text{WIND ZOOM TRACE GRAPH} \]

GRAPH Menu  (GRAPH) in DifEq graphing mode

\[ Q'(t) = \text{WIND INITC AXES GRAPH} \]

Equation Editor Menu  (GRAPH) F1 in Func graphing mode

\[ y(x) = \text{WIND ZOOM TRACE GRAPH} \]

Equation Editor Menu  (GRAPH) F1 in Pol graphing mode

\[ r(q) = \text{WIND ZOOM TRACE GRAPH} \]

Equation Editor Menu  (GRAPH) F1 in Param graphing mode

\[ E(t) = \text{WIND ZOOM TRACE GRAPH} \]

Equation Editor Menu  (GRAPH) F1 in DifEq graphing mode

\[ Q'(t) = \text{WIND INITC AXES GRAPH} \]
GRAPH VARS (Graph Variables) Menu  \[\text{GRAPH} \ F1\text{ in the program editor only}\]

\[
\begin{array}{c}
\text{y}(x)=\text{WIND} \ \text{ZOOM} \ \text{TRACE} \ \text{GRAPH} \\
y \ x \ xt \ yt \ t \\
r \ 0 \ Q1 \ Q1' \ t
\end{array}
\]

\[
\begin{array}{c}
\text{FnOn} \ \text{FnOff} \ \text{Axes} \ Q1 \ \text{dTime} \\
\text{FidRes}
\end{array}
\]

GRAPH WIND (Window Variables) Menu  \[\text{GRAPH} \ F2\text{ in the program editor only}\]

\[
\begin{array}{c}
xMin \ xMax \ xScl \ yMin \ yMax \\
yMin \ tMin \ tMax \ tStep \ tMin'
\end{array}
\]

\[
\begin{array}{c}
\text{tStep} \ \text{iStep} \ \text{tPlot} \ \text{difTol} \ \text{xRes}
\end{array}
\]

GRAPH ZOOM Menu  \[\text{GRAPH} \ F3\]

\[
\begin{array}{c}
\text{BOX} \ \text{ZIN} \ \text{ZOUT} \ \text{ZSTD} \ \text{ZPREV} \\
\text{ZFIT} \ \text{ZSQR} \ \text{ZTRIG} \ \text{ZDECM} \ \text{ZDATA} \\
\text{ZRCL} \ \text{ZFACT} \ \text{ZOOMX} \ \text{ZOOMY} \ \text{ZINT} \\
\text{ZSTO}
\end{array}
\]

To display the GRAPH ZOOM menu in DifEq mode, press \[\text{GRAPH} \ \text{MORE} \ F3\].

DifEq graphing mode has no GRAPH MATH menu.

GRAPH MATH Menu  \[\text{GRAPH} \ \text{MORE} \ F1\text{ in Func graphing mode}\]

\[
\begin{array}{c}
\text{MATH} \ \text{DRAW} \ \text{FORMT} \ \text{STGDB} \ \text{RCGDB} \\
\text{ROOT} \ \text{dy} / dx \ [x] \ \text{FMIN} \ \text{FMAX} \\
\text{INFLC} \ \text{YICPT} \ \text{ISECT} \ \text{DIST} \ \text{ARC} \ \text{TANLN}
\end{array}
\]

GRAPH MATH Menu  \[\text{GRAPH} \ \text{MORE} \ F1\text{ in Pol graphing mode}\]

\[
\begin{array}{c}
\text{MATH} \ \text{DRAW} \ \text{FORMT} \ \text{STGDB} \ \text{RCGDB} \\
\text{DIST} \ \text{dy} / dx \ \text{dr} / df \ \text{ARC} \ \text{TANLN}
\end{array}
\]
Drinv is available only in Func graphing mode.
DrEqu is available only in DifEq graphing mode.
<table>
<thead>
<tr>
<th>Menu</th>
<th>Description</th>
</tr>
</thead>
</table>
| PRGM Menu | **PRGM**  
**NAMES** EDIT |
| Program Editor Menu | **PRGM** F2 *program name* ENTER  
PAGE: PAGE I/O CTL INSc DELc UNDEL |
| PRGM I/O (Input/Output) Menu | **PRGM** F2 *program name* ENTER F3  
PAGE: PAGE I/O CTL INSc Input Promp Disp DispG DispT Ctbl Get Send getKy CLLCD " Outpt InptSt |
| PRGM CTL (Control) Menu | **PRGM** F2 *programName* 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 Menu | **2nd** [POLY] *(integer ≥ 2 & ≤ 30)* ENTER  
CLRq SOLVE |
| POLY RESULT Menu | **F5**  
CLRF SOLVE |
| CUSTOM Menu | **CUSTOM**  
| CATLG-VARS Menu | **2nd** [CATLG-VARS]  
CATLG ALL REAL CPLX LIST VECTR MATRIX STRING EQU CONS PRGM GDB PIC STAT WIND |
| CATLG-VARS Selection Menu | **2nd** [CATLG-VARS] F1 or select a data type  
| PAGE: PAGE CUSTOM|BLANK |
CALC Menu  
2nd [CALC]  
evalF nDer der1 der2 fnInt  > fMin fMax arc  

MATRX Menu  
2nd [MATRX]  
Matrix Editor Menu  
2nd [MATRX] F2 matrixName  ENTER  

MATRX MATH Menu  
2nd [MATRX] F3  
MATRX OPS (Operations) Menu  
2nd [MATRX] F4  

MATRX CPLX Menu  
2nd [MATRX] F5  

VECTR Menu  
2nd [VECTR]  
Vector Editor Menu  
2nd [VECTR] F2 vectorName  ENTER  

VECTR MATH Menu  
2nd [VECTR] F3  

NAMES EDIT MATH OPS CPLX  
INSr DELr INSr DELr RREAL  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  

REAL  
REAL  
REAL  
REAL  
REAL  

rnorm cnorm LU cond  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  

randM  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  

rSwap rAdd mRAdd  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  
NAMES EDIT MATH OPS CPLX  

randM  
cross unitV norm dot
VECTR OPS (Operations) Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim</td>
<td>Fill</td>
<td>→Pol</td>
<td>→Cyl</td>
<td>→Sph</td>
</tr>
</tbody>
</table>

VECTR CPLX Menu

<table>
<thead>
<tr>
<th>NAMES</th>
<th>EDIT</th>
<th>MATH</th>
<th>OPS</th>
<th>CPLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>conj</td>
<td>real</td>
<td>imag</td>
<td>abs</td>
<td>angle</td>
</tr>
</tbody>
</table>

CPLX (Complex Number) Menu

| conj  | real | imag | abs | angle |

MATH Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>round</td>
<td>iPart</td>
<td>fPart</td>
<td>int</td>
<td>abs</td>
</tr>
</tbody>
</table>

MATH NUM (Number) Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>nPr</td>
<td>nCr</td>
<td>rand</td>
<td>randin</td>
</tr>
</tbody>
</table>

MATH PROB (Probability) Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>randN</td>
<td>randB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MATH ANGLE Menu

<table>
<thead>
<tr>
<th>NUM</th>
<th>PROB</th>
<th>ANGLE</th>
<th>HYP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>•</td>
<td>r</td>
<td>°</td>
<td>+DMS</td>
<td></td>
</tr>
</tbody>
</table>
MATH HYP (Hyperbolic) Menu

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

MATH MISC (Miscellaneous) Menu

NUM PROB ANGLE HYP MISC
sum prod seq lcm gcd → »Frac % pEval x⁻¹ eval

CONS (Constants) Menu

CONS BLTIN (Built-In Constants) Menu

CONS BLTIN (Built-In Constants) Menu
BLTIN EDIT USER
Na k Cc ec Rc → Gc g Me Mp Mn → μ0 ε0 h c u

CONV (Conversions) Menu

CONV LENGTH (Length) Menu

CONV LENGTH (Length) Menu
LENGTH AREA VOL TIME TEMP
LENGTH AREA VOL TIME TEMP
mm cm m in ft → yd km mile nmile ft-yr → mil Ang fermi rod fath

CONV AREA Menu

CONV AREA Menu
LENGTH AREA VOL TIME TEMP
ft² m² mi² km² acre → in² cm² yd² ha
CONV VOL (Volume) Menu  2nd [CONV] [F3]

CONV TIME Menu  2nd [CONV] [F4]

CONV TEMP (Temperature) Menu  2nd [CONV] [F5]

CONV MASS Menu  2nd [CONV] MORE [F1]

CONV FORCE Menu  2nd [CONV] MORE [F2]

CONV PRESS (Pressure) Menu  2nd [CONV] MORE [F3]

CONV ENRGY (Energy) Menu  2nd [CONV] MORE [F4]
### Appendix 389

#### CONV POWER Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>CONV</th>
<th>MORE</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASS</td>
<td>FORCE</td>
<td>PRESS</td>
<td>ENERGY</td>
</tr>
<tr>
<td>hp</td>
<td>W</td>
<td>ft/lb/s</td>
<td>cal/s</td>
</tr>
</tbody>
</table>

#### CONV SPEED Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>CONV</th>
<th>MORE</th>
<th>MORE</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPEED</td>
<td>ft/s</td>
<td>m/s</td>
<td>mi/hr</td>
<td>km/hr</td>
</tr>
</tbody>
</table>

#### STRNG Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>STRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>sub</td>
</tr>
</tbody>
</table>

#### LIST Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>LIST</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>)</td>
<td>NAMES</td>
</tr>
</tbody>
</table>

#### LIST NAMES Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>LIST</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>)</td>
<td>NAMES</td>
</tr>
<tr>
<td>fStat</td>
<td>xStat</td>
<td>yStat</td>
</tr>
</tbody>
</table>

#### List Editor Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>LIST</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>)</td>
<td>NAMES</td>
</tr>
</tbody>
</table>

#### LIST OPS (Operations) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>LIST</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>)</td>
<td>NAMES</td>
</tr>
<tr>
<td>dimL</td>
<td>sortA</td>
<td>sortD</td>
</tr>
<tr>
<td>sum</td>
<td>prod</td>
<td>seq</td>
</tr>
<tr>
<td>Fill</td>
<td>aug</td>
<td>cSum</td>
</tr>
<tr>
<td>Sorty</td>
<td>Select</td>
<td>SetLE</td>
</tr>
</tbody>
</table>

#### The (Number) BASE Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
</tr>
</tbody>
</table>

#### BASE A-F (Hexadecimal) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
<td>CONV</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

#### BASE TYPE Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
<th>F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
<td>CONV</td>
</tr>
<tr>
<td>b</td>
<td>h</td>
<td>c</td>
</tr>
</tbody>
</table>

#### BASE CONV (Conversions) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
<td>CONV</td>
</tr>
<tr>
<td>bBin</td>
<td>hHex</td>
<td>cOct</td>
</tr>
</tbody>
</table>
### BASE BOOL (Boolean) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
<th>F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
<td>CONV</td>
</tr>
<tr>
<td>BOOL</td>
<td>BIT</td>
<td></td>
</tr>
<tr>
<td>and</td>
<td>or</td>
<td>xor</td>
</tr>
<tr>
<td>not</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### BASE BIT Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>BASE</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-F</td>
<td>TYPE</td>
<td>CONV</td>
</tr>
<tr>
<td>BOOL</td>
<td>BIT</td>
<td></td>
</tr>
<tr>
<td>rotR</td>
<td>rotL</td>
<td>shftR</td>
</tr>
</tbody>
</table>

### TEST (Relational) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>[TEST]</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>&lt;</td>
</tr>
</tbody>
</table>

### MEM (Memory) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>[MEM]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>DELET</td>
</tr>
</tbody>
</table>

### MEM DELET (Delete) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>[MEM]</th>
<th>F2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>DELET</td>
<td>RESET</td>
</tr>
<tr>
<td>ALL</td>
<td>REAL</td>
<td>CPLX</td>
</tr>
<tr>
<td>MATRX</td>
<td>STRNG</td>
<td>EQU</td>
</tr>
<tr>
<td>GDB</td>
<td>PIC</td>
<td></td>
</tr>
</tbody>
</table>

### MEM RESET Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>[MEM]</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>DELET</td>
<td>RESET</td>
</tr>
<tr>
<td>ALL</td>
<td>MEM</td>
<td>DFLTS</td>
</tr>
</tbody>
</table>

### STAT (Statistics) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC</td>
<td>EDIT</td>
</tr>
<tr>
<td>FCST</td>
<td></td>
</tr>
</tbody>
</table>

### STAT CALC (Calculations) Menu

<table>
<thead>
<tr>
<th>2nd</th>
<th>STAT</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC</td>
<td>EDIT</td>
<td>PLOT</td>
</tr>
<tr>
<td>OneVa</td>
<td>TwoVa</td>
<td>LinR</td>
</tr>
<tr>
<td>PwrR</td>
<td>SinR</td>
<td>LgstR</td>
</tr>
<tr>
<td>P4Reg</td>
<td>StReg</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 391

STAT PLOT Menu

PLOT1 | PLOT2 | PLOT3 | PLOT4

Plot Type Menu

PLOT1 | PLOT2 | PLOT3 | PLOT4

Plot Mark Menu

PLOT1 | PLOT2 | PLOT3 | PLOT4

STAT DRAW Menu

HIST | SCAT | xyLINE | MBOX

STAT VARS (Statistical Result Variables) Menu

CALC | EDIT | PLOT | DRAW | VARS

CHAR (Character) Menu

MISC | GREEK | INTL

CHAR MISC (Miscellaneous) Menu

MISC | GREEK | INTL

\(\bar{x}, \bar{y}, \bar{z}\), and \(\bar{c}\) are valid as the first letter of a variable name.

\%, *, and ! can be functions.
Handling a Difficulty

1. If you cannot see anything on the screen, you may need to adjust the contrast (Chapter 1).
   - To darken the screen, press and release 2nd, and then press and hold 4.
   - To lighten the screen, press and release 2nd, and then press and hold 5.

2. If an error menu is displayed, follow the steps in Chapter 1. Refer to the Error Conditions section of the Appendix (page 393) for details about specific errors, if necessary.

3. If a checkerboard cursor (˜) is displayed, then either you have entered the maximum number of characters in a prompt or memory is full. If memory is full, press 2nd [MEM] F2, select a data type, and then delete some items from memory (Chapter 17).

4. If the busy indicator (dotted line) is displayed in the top-right corner, a graph or program has paused; the TI-86 is waiting for input. Press ENTER to continue or press ON to break.

5. If the calculator does not seem to work at all, be sure the batteries are fresh and that they are installed properly. Refer to battery information in Chapter 1.
**Error Conditions**

When the TI-86 detects an error, it displays an error message **ERROR # type** and the error menu. Chapter 1 describes how to correct an error. This section describes possible causes for the errors and examples. To find the proper arguments for a function or instruction, as well as restrictions on those arguments, refer to Chapter 20: A to Z Function and Instruction Reference.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 OVERFLOW</td>
<td>♦ You attempted to enter a number that is beyond the calculator’s range.</td>
</tr>
<tr>
<td></td>
<td>♦ You attempted to execute an expression with a result that is beyond the</td>
</tr>
<tr>
<td></td>
<td>calculator’s range.</td>
</tr>
<tr>
<td>02 DIV BY ZERO</td>
<td>♦ You attempted to divide by zero.</td>
</tr>
<tr>
<td></td>
<td>♦ You attempted a linear regression with a vertical line.</td>
</tr>
<tr>
<td>03 SINGULAR MAT</td>
<td>♦ You attempted to use a singular matrix (determinate = 0) as the argument</td>
</tr>
<tr>
<td></td>
<td>for ( L_1 ), Simult, or LU.</td>
</tr>
<tr>
<td></td>
<td>♦ You attempted a regression with at least one inappropriate list.</td>
</tr>
<tr>
<td></td>
<td>♦ You attempted to use a matrix with repeated eigenvalues as the argument</td>
</tr>
<tr>
<td></td>
<td>for ( \exp, \cos, ) or ( \sin ).</td>
</tr>
<tr>
<td>04 DOMAIN</td>
<td>♦ You attempted to use an argument that is out of the range of valid values</td>
</tr>
<tr>
<td></td>
<td>for the function or instruction.</td>
</tr>
<tr>
<td></td>
<td>♦ You attempted a logarithmic or power regression with a ( -x ) or an</td>
</tr>
<tr>
<td></td>
<td>exponential regression with a ( -y ).</td>
</tr>
<tr>
<td>05 INCREMENT</td>
<td>The increment in ( \text{seq} ) is 0 or has the wrong sign; the increment</td>
</tr>
<tr>
<td></td>
<td>for a loop is 0.</td>
</tr>
<tr>
<td>06 BREAK</td>
<td>You pressed [( ^2 )] to break a program, DRAW instruction, or expression</td>
</tr>
<tr>
<td></td>
<td>evaluation.</td>
</tr>
<tr>
<td>07 SYNTAX</td>
<td>You entered a value; look for misplaced functions, arguments, parentheses,</td>
</tr>
<tr>
<td></td>
<td>or commas; check the syntax description in the A to Z Reference.</td>
</tr>
</tbody>
</table>

Errors 1 through 5 do not occur during graphing. The TI-86 allows for undefined values on a graph.
08 NUMBER BASE
◆ You entered an invalid digit in a number base, such as 7b.
◆ You attempted an operation that is not allowed in Bin, Oct, or Hex base mode.

09 MODE
You attempted to store to a window variable of a noncurrent graphing mode,
or to use an instruction valid only in noncurrent graphing modes; for example,
using DrInv in Pol, Param, or DifEq graphing mode.

10 DATA TYPE
◆ You entered a value or variable that is an inappropriate data type.
◆ You entered an argument that is an inappropriate data type for a function
  or an instruction, such as a program name for sortA.
◆ In an editor, you entered a data type that is not allowed; check the
  appropriate chapter.
◆ You attempted to store data to a protected data type, such as a constant,
  program, picture, or graph database.
◆ You attempted to store inappropriate data to a restricted built-in variable,
  such as the list names xStat, yStat, and fStat.

11 ARGUMENT
You attempted to execute a function or instruction without all the arguments.

12 DIM MISMATCH
You attempted to use two or more lists, matrices, or vectors as arguments, but
the dimensions of all arguments are not equal, such as {1,2}+{1,2,3}.

13 DIMENSION
◆ You entered an argument with an inappropriate dimension.
◆ You entered a matrix or vector dimension < 1 or > 255 or a noninteger.
◆ You attempted to invert a matrix that is not a square matrix.

14 UNDEFINED
You are referencing a variable that currently is not defined.

15 MEMORY
Memory is insufficient to perform the desired command; you must delete
items from memory (Chapter 17) before executing this command.

16 RESERVED
You attempted to use a built-in variable inappropriately.

17 INVALID
You attempted to reference a variable or use a function where it is not valid.
### Appendix 395

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 ILLEGAL NEST</td>
<td>You attempted to use an invalid function in an argument for <code>seq</code> or a <code>CALC</code> function; for example, <code>der1(der1(x^3,x),x))</code>.</td>
</tr>
<tr>
<td>19 BOUND</td>
<td>You defined an upper bound that is less than the specified lower bound or a lower bound that is greater than the specified upper bound.</td>
</tr>
<tr>
<td>20 GRAPH WINDOW</td>
<td>One or more window variable values is incompatible with the others for defining the graph screen; for example, you defined <code>xMax &lt; xMin</code>. Window variables are too small or too large to graph correctly; for example, you attempted to zoom out beyond the calculator’s range.</td>
</tr>
<tr>
<td>21 ZOOM</td>
<td>A ZOOM operation resulted in an error; you attempted to define <code>ZBOX</code> with a line.</td>
</tr>
<tr>
<td>22 LABEL</td>
<td>In programming, the <code>Goto</code> instruction label is not defined with a <code>Lbl</code> instruction.</td>
</tr>
<tr>
<td>23 STAT</td>
<td>You attempted a stat calculation with at least one inappropriate list, such as a list with less than two data points. At least one element of a frequency list is &lt; 0. <code>(xMax - xMin)/xScl ≤ 63</code> must be true when plotting a histogram.</td>
</tr>
<tr>
<td>24 CONVERSION</td>
<td>When converting measurements, the units are incompatible, as in volts to liters.</td>
</tr>
<tr>
<td>25 SOLVER</td>
<td>In the solver editor, the equation does not contain a variable. You attempted to graph with the cursor positioned on bound.</td>
</tr>
<tr>
<td>26 SINGULARITY</td>
<td>In the solver editor, the equation contains a singularity, which is a point at which the function is not defined.</td>
</tr>
<tr>
<td>27 NO SIGN CHNG</td>
<td>The solver did not detect a sign change.</td>
</tr>
<tr>
<td>28 ITERATIONS</td>
<td>The solver has exceeded the maximum permitted number of iterations.</td>
</tr>
<tr>
<td>29 BAD GUESS</td>
<td>The initial guess was outside the specified bounds. The initial guess and several points around the guess are undefined.</td>
</tr>
</tbody>
</table>

*Errors 26 through 29 occur during the solving process. Examine a graph of the function or a graph of the variable vs. left-right in the SOLVER. If the equation has a solution, change bounds and/or the initial guess.*
30 DIF EQ SETUP  In DifEq graphing mode, equations in the equation editor must be from \( Q'1 \) to \( Q'9 \) and each must have an associated initial condition from \( Q1 \) to \( Q9 \).

31 DIF EQ MATH  The step size used by the fitting algorithm has become too small; check the equations and initial values; try a larger value for the window variable \( \text{difTol} \); try changing \( \text{tMin} \) or \( \text{tMax} \) to examine a different region of the solution.

32 POLY  All coefficients are 0.

33 TOL NOT MET  The algorithm cannot return a result accurate to the requested tolerance.

34 STAT PLOT  You attempted to display a stat plot that references an undefined list.

35 AXES  You attempted to plot a DifEq graph with improper axes set.

36 FLD/ORDER  ♦ You attempted to plot a 2nd-order or higher differential equation with \( \text{SlpFld} \) field format set; change field format or modify the order.
   ♦ You attempted to plot a 3rd-order or higher differential equation with \( \text{DirFld} \) field format set; change field format or modify the order.

37 LINK MEMORY FULL  You attempted to transmit an item with insufficient available memory in the receiving unit; skip the item or cancel the transmission.

38 LINK TRANSMISSION ERROR  ♦ Unable to transmit item; check to see that the cable is firmly connected to both units and the receiving unit is ready to receive data (Chapter 18).
   ♦ You pressed \( \text{EXIT} \) to break during transmission.

39 LINK DUPLICATE NAME  You attempted to transmit an item when an item with the same name already exists in the receiving unit.
Equation Operating System (EOS™)

The Equation Operating System (EOS) governs the order of evaluation on the TI-86. Calculations within parentheses are evaluated first, and then EOS evaluates functions within an expression in this order:

1st Functions that are entered after the argument, such as $^2$, $^{-1}$, $'$, $!$, and conversions

2nd Powers and roots, such as $2^5$ or $5\sqrt[3]{2}$

3rd Single-argument functions that precede the argument, such as $\sqrt{()}$, $\sin()$, or $\log()$

4th Permutations ($nPr$) and combinations ($nCr$)

5th Multiplication, implied multiplication, and division

6th Addition and subtraction

7th Relational functions, such as $>$ or $\leq$

8th Logic operator and

9th Logic operators or and xor

**Implied Multiplication**

The TI-86 recognizes implied multiplication, so you need not press $\times$ to express multiplication in all cases. For example, the TI-86 interprets $1\div 2x$, $4\sin(46)$, $5(1+2)$, and $(2\times 5)7$ as implied multiplication.

**Parentheses**

All calculations inside a pair of parentheses are completed first. For example, in the expression $4(1+2)$, EOS evaluates $1+2$ inside the parentheses first, and then multiplies 3 by 4.

\[
\begin{align*}
4 \times 1+2 & = 6 \\
4 \times (1+2) & = 12
\end{align*}
\]
You can omit the close parenthesis ( ) at the end of an expression. All open parenthetical elements are closed automatically at the end of an expression. This is also true for open parenthetical elements that precede the store or display-conversion instructions.

Open parentheses after list names, matrix names, or equation function names are not interpreted as implied multiplication. Arguments that follow these open parentheses are specified list elements, matrix elements, or values for which to solve the equation function.

**TOL (The Tolerance Editor)**

On the TI-86, the computational accuracy of some functions is controlled by the variables tol and $\delta$. The values stored to these variables may affect the speed at which the TI-86 calculates or plots.

The variable tol defines the tolerance in calculating the functions $\text{fnInt}(\text{fMin}(\text{fMax}(\text{arc}(\text{f}(x))$, FMIN, FMAX, and ARC (Chapter 6). tol must be a positive value $\geq 1E^{-12}$.

The value stored to $\delta$ must be a positive real number. $\delta$ defines the step size the TI-86 uses to calculate the functions $\text{arc}$ in dxNDer mode; $\text{nDer}$; and the operations $\text{dy/dx}$, $\text{dr/dt}$, $\text{dy/dt}$, $\text{dx/dt}$, INFLC, TANLN, and ARC, all in dxNDer mode (Chapter 6).

To store a value to tol or $\delta$ on the home screen or in a program, use [STO]. You can select tol and $\delta$ from the CATALOG. Also, you can enter tol directly and select $\delta$ from the CHAR GREEK menu.
Computational Accuracy

To maximize accuracy, the TI-86 carries more digits internally than it displays. Values are stored in memory using up to 14 digits with a 3-digit exponent.

◆ You can store values up to 12 digits long to most window variables. To xScl, yScl, tStep, and qStep, you can store values up to 14 digits long.

◆ When a value is displayed, the displayed value is rounded as specified by the mode setting (Chapter 1), with a maximum of 12 digits and a 3-digit exponent.

◆ Chapter 4 describes calculations in hexadecimal, octal, and binary number bases.
Support and Service Information

Product Support

Customers in the U.S., Canada, Puerto Rico, and the Virgin Islands
For general questions, contact Texas Instruments Customer Support:

    phone: 1-800-TI-CARES (1-800-842-2737)
    e-mail: ti-cares@ti.com

For technical questions, call the Programming Assistance Group of Customer Support:

    phone: 1-972-917-8324

Customers outside the U.S., Canada, Puerto Rico, and the Virgin Islands
Contact TI by e-mail or visit the TI Calculator home page on the World Wide Web.

    e-mail: ti-cares@ti.com
    Internet: education.ti.com
Product Service

Customers in the U.S. and Canada Only
Always contact Texas Instruments Customer Support before returning a product for service.

Customers outside the U.S. and Canada
Refer to the leaflet enclosed with this product or contact your local Texas Instruments retailer/distributor.

Other TI Products and Services
education.ti.com
Warranty Information

Customers in the U.S. and Canada Only

One-Year Limited Warranty for Commercial Electronic Product

This Texas Instruments electronic product warranty extends only to the original purchaser and user of the product.

Warranty Duration. This Texas Instruments electronic product is warranted to the original purchaser for a period of one (1) year from the original purchase date.

Warranty Coverage. This Texas Instruments electronic product is warranted against defective materials and construction. THIS WARRANTY IS VOID IF THE PRODUCT HAS BEEN DAMAGED BY ACCIDENT OR UNREASONABLE USE, NEGLECT, IMPROPER SERVICE, OR OTHER CAUSES NOT ARISING OUT OF DEFECTS IN MATERIALS OR CONSTRUCTION.

WARRANTY DISCLAIMERS. ANY IMPLIED WARRANTIES ARISING OUT OF THIS SALE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO THE ABOVE ONE-YEAR PERIOD. TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR LOSS OF USE OF THE PRODUCT OR OTHER INCIDENTAL OR CONSEQUENTIAL COSTS, EXPENSES, OR DAMAGES INCURRED BY THE CONSUMER OR ANY OTHER USER.

Some states/provinces do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you.

Legal Remedies. This warranty gives you specific legal rights, and you may also have other rights that vary from state to state or province to province.

Warranty Performance. During the above one (1) year warranty period, your defective product will be either repaired or replaced with a reconditioned model of an equivalent quality, (at TI's option) when the product is returned, postage prepaid, to Texas Instruments Service Facility. The warranty for the repaired or replacement unit will continue for the warranty of the original unit or six (6) months, whichever is longer. Other than the postage requirement, no charge will be made for such repair and/or replacement. TI strongly recommends that you insure the product for value prior to mailing.

Software. Software is licensed, not sold. TI and its licensors do not warrant that the software will be free from errors or meet your specific requirements. All software is provided "AS IS."

Copyright. The software and any documentation supplied with this product are protected by copyright.
Australia & New Zealand Customers only

One-Year Limited Warranty for Commercial Electronic Product

This Texas Instruments electronic product warranty extends only to the original purchaser and user of the product.

Warranty Duration. This Texas Instruments electronic product is warranted to the original purchaser for a period of one (1) year from the original purchase date.

Warranty Coverage. This Texas Instruments electronic product is warranted against defective materials and construction. This warranty is void if the product has been damaged by accident or unreasonable use, neglect, improper service, or other causes not arising out of defects in materials or construction.

Warranty Disclaimers. Any implied warranties arising out of this sale, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, are limited in duration to the above one-year period. Texas Instruments shall not be liable for loss of use of the product or other incidental or consequential costs, expenses, or damages incurred by the consumer or any other user.

Some jurisdictions do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you.

Legal Remedies. This warranty gives you specific legal rights, and you may also have other rights that vary from jurisdiction to jurisdiction.

Warranty Performance. During the above one (1) year warranty period, your defective product will be either repaired or replaced with a new or reconditioned model of an equivalent quality (at TI’s option) when the product is returned to the original point of purchase. The repaired or replacement unit will continue for the warranty of the original unit or six (6) months, whichever is longer. Other than your cost to return the product, no charge will be made for such repair and/or replacement. TI strongly recommends that you insure the product for value if you mail it.

Software. Software is licensed, not sold. TI and its licensors do not warrant that the software will be free from errors or meet your specific requirements. All software is provided “AS IS.”

Copyright. The software and any documentation supplied with this product are protected by copyright.
All Customers outside the U.S. and Canada

For information about the length and terms of the warranty, refer to your package and/or to the warranty statement enclosed with this product, or contact your local Texas Instruments retailer/distributor.
Index

" (string), 216, 227
" (List Editor menu), 156
! (factorial), 294
→, 362
≥ (greater than or equal to), 56, 301
≤ (less than or equal to), 55, 312
# (not equal to), 56, 326
π (pi), 48
√ (square root), 360
[^] (square root) key, 48
ó (statistical result variable), 193
ó (statistical result variable), 193
⁻¹ (inverse), 48, 309
→dim, 184, 281
→dimL, 282
[f(x) (function numerical integral), 96, 98
ΔTbl (table step), 113
σ (statistical result variable), 193
Σx² (statistical result variable), 193
σy (statistical result variable), 193
% (percent), 52, 334
< (less than), 55, 312
= (assign to), 270
= (equals), 290
== (relational equals), 55, 291
> (greater than), 55, 300
Ans (last answer), 29, 30, 41, 269
answer
   displaying, 19
   storing to a variable, 41
APD. See Automatic Power Down
ARC, 96, 98
arc, 54, 269
argument, 25
Asm (assembly language program), 269
AsmComp (compile assembly language program), 226, 270
AsmPrgm (assembly language program), 226, 270
assembly language programs, 225
assignment, 270
attached formulas
   executing, 164
   resolving errors, 165
attached-formula list
   comparing, 163
   creating, 162
   editing elements, 166
Automatic Power Down, 17
automatic regression equation storage, 191
AXES, 137
Axes editor, 137
   field formats, 137
AxesOff, 84, 271
AxesOn, 84, 271
B
b (binary), 271
backup battery, 16

alpha cursor, 22
ALPHA key, 21
ALPHA-lock, 22, 44
   canceling, 22
   setting, 22
and (Boolean), 69, 268
angle, 71, 175, 185, 269
   expressed in degrees, 51
angle modes, 35
angle values, 35
A
abs (absolute value), 49, 71,
   175, 185, 267
addition (+), 267
ALL, 43, 232
ALL−, 77
ALL+, 77
ALPHA character, 22
ALPHA cursor, 22

99INDEX.DOC   TI-86, Index, US English   Bob Fedorisko   Revised: 02/13/01 2:51 PM   Printed: 02/27/01 1:29 PM   Page 405 of 15
BASE A-F (Hexadecimal) menu, 67
BASE BIT menu, 69
BASE BOOL (Boolean) menu, 68
BASE CONV (Conversion) menu, 68
BASE menu, 66
BASE TYPE menu, 67
BASE Type symbol, 67
batteries, 2, 16-18
battery compartment, 16
BCKUP (memory backup), 237
Bin (binary), 35, 272
4 Bin (to binary), 68, 272
Bin (binary), 35, 272
Bin (to binary), 68, 272
binary integer, 271
binary number base, 35, 66
Binary operators, 68, 268, 325, 328, 370
binompdf (binomial), 156
bound=[-1E99,1E99], 204
bounds, 204
Box (stat plot), 272
BOX (GRAPH ZOOM menu), 14, 92, 93
Box (ZOOM menu), 208
break (program), 222
BREAK menu, 26
built-in constants, 58
built-in variables, 39, 45, 138
busy indicator, 26, 85
C
CALC (Calculus) menu, 54
calculating derivatives, 7
calculation
interrupting, 26
calculus functions, 54
CATALOG, 25, 38
Quick-Find Locator, 262
CATLG (CATALOG), 43
CATLG-VARS (CATALOG Variables) menu, 43
changing TI-86 settings, 39
CHAR (Character) menu, 45
CHAR GREEK menu, 46
CHAR INTL (International) menu, 46
CHAR MISC (Miscellaneous) menu, 46
characters, 19
alpha, 22
blue, 21, 22
case, 22
characters (continued)
deleting, 23
entering, 21
second, 22
yellow, 21
check RAM screen, 230
CIRCL (circle), 105, 106
Circl(), 273
circles
drawing, 106
CLDRW (clear drawing), 103, 105, 273
clearing CUSTOM menu items, 45
clearing ENTRY storage area, 29
CILCD (clear LCD), 216, 273
CleEnt (clear entry), 232, 273
CITbl (clear table), 114, 216, 273
cnorm (column norm), 183, 273
command line, 220
complements (binary numbers), 66
complex matrix, 180
Complex Number menu, 71
complex number modes, 35
complex number variables, 43
complex numbers, 29, 70
as list elements, 156
displaying as result, 5
entering, 20
in results, 70
separator, 70
using in expressions, 71
complex values, 48
concatenation (+), 274
cond (condition number), 183, 274
cnj (complex conjugate), 71, 175, 185, 275
connecting instructions, 235
CONS (constants), 43
CONS (Constants) menu, 58
CONS BLTIN (Built-In Constants) menu, 58
CONS EDIT menu, 60
consecutive entries, 26
Constant Memory feature, 17, 34
constants, 59
built-in, 58
defined, 58
name, 61
user-created, 58, 60
contrast
  adjusting, 2, 18
CONV (Conversions) menu, 62
CONV AREA menu, 63
CONV ENRGY (Energy) menu, 64
CONV FORCE menu, 64
CONV LNGTH (Length) menu, 63
CONV MASS menu, 64
CONV POWER menu, 64
CONV PRESS (Pressure) menu, 64
CONV SPEED menu, 64
CONV TEMP (Temperature) menu, 8, 63
CONV TIME menu, 63
CONV VOL (Volume) menu, 63
conversions
  ►Bin, 272
  ►Dec, 279
  ►DMS, 51, 285
  ►Frac, 52, 298
  ►Hex, 303
  ►Oct, 327
  ►Pol, 336
  ►REAL, 156
conversions (continued)
  ►Rec, 343
  ►Sph, 360
  ►St, 227
  ►tst, 160
  ►v, 160
  ►Eq(, 227, 361
  cSum( (cumulative sum), 160, 278
  current entry, 19
  clearing, 23
  current item, 38
  cursor, 17, 22
  ALPHA, 22
  alpha, 22
  changing, 23
  direction keys, 23
  entry, 22
  free-moving, 128, 144, 205
  full, 22
  insert, 22
  location, 19, 20, 21, 25
  moving, 23
  second, 22
  selection, 38
  trace, 90
  curves
    drawing, 107
  CUSTOM menu, 44
  clearing items, 45
  copying items, 44
  Customer Support, 392
  ►Cyl (to cylindrical), 174, 278
  CyIV (cylindrical vector coordinate mode), 36, 278
D
  d (decimal), 278
  data type selection screen, 42
  Dec (decimal number base mode), 278
  Dec (decimal), 35, 65
  ►Dec (to decimal), 279
  decimal, 20
  decimal mode, 34, 35, 65
  fixed (012345678901), 35
  floating, 35
  decimal number, 278
  decimal number base, 35
  decimal point, 35
  degree angle mode, 35, 75, 279
  degree complex-number mode, 70
  degree entry (°), 279
  degrees°, 51
  degrees/minutes/seconds form, 51
  DELc (delete column), 179
  DELET, 60
Index

DELF (delete function), 77
DELI (delete element), 170
DELR (delete row), 179
Deltalst( (delta list), 160, 279
DelVar( (delete variable), 219, 280
der1( (first derivative), 54, 280
der2( (second derivative), 54, 280
derivatives calculating, 7
det (determinant), 183, 281
DFLTS (defaults), 232
DifEq (differential equation mode), 35, 74, 239, 281
differential equation editor, 134
differential equation graphs, 74
displaying, 138
drawing, 145
differential equations changing to first order, 142
defining graph, 132
drawing solutions, 148
DrEqU(, 287
editor, 134
EXPLR, 148
differential equations (continued)
graphing, 132, 137, 139, 141, 142
initial conditions editor, 136
Q'n equation variables, 135
setting axes, 137
setting graph format, 133
setting graphing mode, 132
solving, 139
tracing, 144
using EVAL, 150
window variables, 135
differentiation modes, 36
diffTol (tolerance), 136
dim (dimension), 173, 184, 281
dimL (dimension of list), 159, 282
DirFld (direction field), 134, 282
Disp (display), 216, 283
DispG (display graph), 283
display, 17
display contrast adjusting, 17, 18
displaying a menu, 31
DispT (display table), 284
DIST (distance), 96, 98
division (÷), 284
division symbol, 3
DMS (to degrees/minutes/seconds), 51, 285
dot(, 173, 285
dr/dθ, 122
DRAW, 75, 88
DrawDot, 84, 285
DrawF (draw function), 103, 107, 286
drawing
circles, 106
differential equation graphs, 145
freehand points, lines, curves, 107
function, tangent line, inverse function, 107
line segments, 105
lines, 105, 106
parametric graphs, 130
points, 108
polar graphs, 122
drawing tools, 101
drawings
clearing, 103
drawings (continued)
recalling, 102
saving, 102
DrawLine, 84, 286
DrEqU( (draw equation), 145, 287
DrInv (draw inverse), 103, 107, 287
DS<( (decrement and skip), 219, 288
DUPLICATE NAME menu, 241
dx/dt, 130
dxDer1 (exact differentiation), 36, 75, 288
dxNDer (numeric differentiation), 36, 75, 288
dy/dt, 130
dy/dx, 96, 99, 130
E
E (exponent), 48, 292
e^x (e raised to power), 288
editing equations, 205
editor menu, 33
eigVc (eigenvector), 183, 289
eigVl (eigenvalue), 183, 289
element
matrix, 181
ellipsis
at end of line, 19
in matrix row, 179
Else, 218, 306
e-mail address (TI Customer Support), 392
End, 218, 290, 297, 306
Eng (engineering notation), 34, 20, 290
entry
executing, 19
storing to, 29
entry cursor, 18, 22, 23
[ENTRY] key, 19
ENTRY Storage Area, 28, 29
EOS. See Equation Operating System
ExpSt( (equation to string), 227, 290
exp variable, 54, 203, 205
EQV (equation variables), 43
equal (=), 290
equal to (==), 291
equation
entering, 203
evaluating, 122, 130
equation coefficients
storing to a variable, 210
equation editor, 74, 75, 76, 80
entering a function, 77
graph styles, 77
parametric, 126
polar, 118
Equation Editor menu, 76
Equation Operating System, 397
equation results
storing to a variable, 210
equation solver, 40, 202
graph tools, 207
equation storage
automatic regression, 191
equation variables, 40, 43, 78
equation-entry editor, 203
equations
editing, 205
solving, 206
error conditions, 393
error message, 27
error type, 27
errors, 17, 27
correcting, 27
diagnosing, 27
from attached formulas, 165
EStep, 136
Euler method, 133, 291
eval, 52, 76, 88, 101, 122, 130,
150, 291
evalF(, 54, 292
evaluating a function for x, 101
evaluating equations, 122, 130
e^x (constant e raised to a power), 48
exact differentiation, 36
EXIT (cancel data transmission), 241
exiting a menu, 6, 33
exp variable, 54, 203
EXPLR (explore), 148
exponent (E), 292
ExpR (exponential regression),
190, 293
expression, 18, 20, 24, 25, 26,
30, 48
editing, 4
everting, 24
expression (continued)
everting a list, 153
everting, 29, 30
using a complex number, 71
using a vector, 172
using matrix, 181
F
factorial (!), 50, 294
Fahrenheit
converting to Celsius, 8
family of curves
graphing, 86
in parametric graphs, 129
in polar graphs, 120
fcstx (forecast x), 294
fcsty (forecast y), 294
feature symbol, 39
field formats, 134
Fill, 184
Fill(, 160, 173, 295
Fix, 295
FldOff (slope and direction fields off), 134, 295
fldPic (field) variable, 138
Float, 35, 295
FMAX (function maximum), 96, 97
fMax( (function maximum), 54, 296
FMIN (function minimum), 96, 97
fMin( (function minimum), 54, 296
fnInt( (function integral), 54, 296
FnOff (functions off), 296
FnOn (functions on), 297
For(, 218, 297
Form(, 161, 298
FORMT (graph format), 76
formulas
attaching, 163
attaching to list name, 162
detaching, 166
fPart (fractional part), 49, 176, 186, 298
▷Frac (to fractions), 52, 298
fraction, 3, 19
free-moving cursor, 84, 144
parametric graphs, 128
polar graphs, 119
fStat (frequency list), 189
full cursor, 22
Func (function mode), 35, 74, 239, 299
function graphs, 73, 74
mode, 35
functions, 25, 38
deleting, 77
deselecting, 13
drawing, 107
entering, 25
entering in the equation editor, 76, 77, 78
evaluating, 101
keyboard, 48
plotting, 11
tracing, 11
using with lists, 5, 161
G
gcd( (greatest common denominator), 52, 298
GDB (graph database), 43
GDB variable, 102
Get(, 299
getKy (get key), 216, 300
key code diagram, 217
GOTO, 26, 27, 300
Goto (PRGM CTL menu), 219, 224
graph, 75
defining, 74
displaying, 85
family of curves, 86
interrupting, 26
modifying, 85
pausing, 85
shading, 104
stopping, 85
GRAPH (Solver menu), 206
graph database (GDB), 102
recalling, 76
GRAPH DRAW menu, 75, 103, 122, 145
graph format
differential equations, 133, 137
parametric graphs, 128
polar graphs, 119
screen, 76
setting, 83
GRAPH LINK, 235
GRAPH MATH menu, 75, 95, 122, 130
GRAPH MATH operations
effect of other settings, 96
using f(x), DIST, or ARC, 98
using dy/dx or TANLN, 99
using ISECT, 100
using ROOT, FMIN, FMAX, or INFLC, 97
using YICPT, 100
GRAPH menu, 27, 31, 75, 88, 117, 126, 133
graph modes, 35
setting, 74
differential equations, 144
function, parametric, 126
polar, 35, 117
graph screen, 75
setting window variables, 81
graph screen dimensions, 75
graph styles, 79
changing, 10
GrStl(, 302
setting, 79
graph tools
in differential equation graphs, 144
in equation solver, 207
graph tools (continued)  
in parametric graphs, 128  
in polar graphs, 119  

graph zoom  
defining custom, 93  
defining screen, 92  
setting zoom factors, 93  
Smart Graph, 94  
zooming in, 92, 93  
zooming out, 92, 93  
GRAPH ZOOM menu, 75, 91, 147  

graphing accuracy, 89  
greater than (>), 300  
greater than or equal to (≥), 301  
grid points, 84  
GridOff, 84, 301  
GridOn, 84, 302  
GrStl( (graph style), 220, 302  
Guess, 204  
in interactive solver editor, 205  

H  
H (hexadecimal), 302  
Hex (hexadecimal), 35, 302  
Hex (hexadecimal), 68, 303  
hexadecimal characters menu, 67  
hexadecimal number base, 35, 66  
Hist (histogram), 303  
home screen, 17, 18, 23, 24, 26, 27  
displaying entries and answers, 18  
Horiz, 304  
HORIZ (horizontal line), 105, 106  
hypercyclic functions, 51  
I  
IAsk, 304  
IAuto, 304  
ident (identity), 184, 304  
If, 218, 305, 306  
imag (imaginary), 71, 175, 185, 306  
imaginary portion of complex number, 71  
implied multiplication, 397  
INFLC (inflection point), 96, 97  
INTC (initial conditions), 136  
InpSt, 217, 307  
Input (PRGM I/O menu), 216, 307  
Input CBLGET, 216  
INSc (insert column), 179  
insert cursor, 22, 23  
canceling, 23  
INSf (insert function), 77  
INSi (insert element), 170  
INSt (insert row), 179  
installing batteries, 16  
instructions, 25  
entering, 25  
execting, 19  
int (integer), 49, 176, 186, 308  
integer part, 49  
integer part of real numbers displaying, 6  
inter( (interpolate), 309  
interactive-solver editor, 204  
boundaries, 204  
international letters, 46  
Internet  
downloading programs, 235  
e-mail address (TI Customer Support), 392  
interpolate/extrapolate editor, 53  
interrupting a calculation, 26  
interrupting a graph, 26, 27  
interrupting a program, 222  
inverse, 309  
inverse function  
drawing, 107  
IPart (integer part), 6, 49, 176, 186, 309  
IS>( (increment and skip), 219, 310  
ISECT (intersection), 96, 100  
items on menus, 31  
K  
keys, 48  
2nd, 21  
ALPHA, 21  
primary function, 19, 21, 22  
key code diagram, 217  
L  
LabelOff, 84, 310  
LabelOn, 84, 310
last answer, 28, 29
    storing to variable, 3
last entry, 26, 28
Lbl (label), 219, 224, 311
lcm( (least common multiple), 52, 311
LCust( (load custom menu), 220, 311
left−rt, 202
length of segment of curve, 54
    less than (<), 312
    less than or equal to (≤), 312
LgstR (logistic regression), 190, 193, 313
li4vc (list to vector), 160, 174, 316
LINE, 104, 105
Line, 314
Lines
    drawing, 107
LINK menu, 236
LINK SEND menu, 236
LINK SEND85 menu, 239
linking instructions, 235
linking options, 234
LinR (linear regression), 190, 315
list, 29, 43, 52
    as an argument, 161
    attaching formulas, 165
    attaching formula, 162, 166
    braces [ ], 316
    comparing, 163
    creating, 157
    deleting an element, 158
    deleting from memory, 154
    detaching formulas, 166
    displaying list elements, 154
    editing elements, 166
    entering in an expression, 153
    inserting, 157
    removing from list editor, 158
    storing, 154
    uses, 152
    using with function, 5
list editor, 31, 67, 156, 188
    attaching formulas, 163, 164
    removing a list, 158
List Editor menu, 156
list element
    complex, 156
    deleting, 158
list element (continued)
    displaying, 155, 158
    editing, 158
    storing a value to, 155
list entry { }, 316
LIST menu, 152
list names, 43
LIST NAMES menu, 153, 189
LIST OPS menu, 159
ln (natural log), 48, 316
lngth (length of string), 227, 316
LnR (logarithmic regression), 190, 317
log, 48, 318
low-battery message, 16, 18
lower menu, 32
LU( (lower-upper), 183, 318

M
Macintosh
    linking to, 235
MATH, 75
MATH (Graph menu), 88
MATH ANGLE menu, 51
MATH HYP (Hyperbolic) menu, 51
MATH menu, 31, 49
MATH MISC (Miscellaneous) menu, 52
MATH NUM (Number) menu, 31, 49
MATH PROB (Probability) menu, 50
mathematical functions, 48
    using with lists, 161
    with a matrix, 185
matrix, 29
    brackets [], 180, 319
    creating, 178, 180
    defined, 178
    deleting from memory, 180
    displaying elements, rows, submatrices, 181
    editing using [STO*], 182
    names, 43
    using in expression, 181
    using math functions, 185
Matrix Editor menu, 179
matrix entry [], 319
MATRX (matrix names), 43
MATRX (Matrix) menu, 178
MATRX CPLX (Complex) menu, 185
MATRX MATH menu, 183
MATRX NAMES menu, 178
MATRX OPS (Operations) menu, 184
max(, 49, 160, 319
maximum characters, 22
maxX, 193
maxY, 193
MBox, 319
Med (median), 193
MEM (clear memory), 232
MEM (Memory) menu, 29, 230
MEM RESET menu, 232
MEM DELET (Delete) menu, 231
MEM FREE (available memory), 230
memory, 16, 17, 22, 28, 29, 223
available, 230
deleting items, 231
resetting, 3, 232
memory backup
initiating, 237
overwrite warning, 237
menus
displaying, 31
exiting, 6
menus (continued)
in editors, 32
keys, 32
lower, 32
removing, 6, 33
selecting items, 32
upper, 33
menu map, 380
Menu(, 219, 320
min(, 49, 160, 320
minX, 193
minY, 193
mod(, 49, 320
mode settings, 19, 20, 70
changing, 34
displaying, 34
number base, 65
modulo, 49
mRAdd, 184
mRAdd(, 321
multiple entries
retrieving, 29
multiplication (*), 321
multR( (multiply row), 184, 322
N
n (statistical results variable), 193
natural log, 48
nCr (number of combinations), 50, 322
nDer( (numerical derivative), 54, 323
negation symbol (¬), 20
negative numbers
entering, 19
norm, 173, 183, 323
Normal, 34, 324
not (Boolean), 66, 69, 325
not equal to (≠), 326
notation modes,
engineering, 34
normal, 34
scientific, 34
notation of displayed answers,
20
nPr (number of permutations), 50, 326
number base, 65
designators, 65
ranges, 66
type, designating, 67
modes, 35
numbers
entering, 19
umeric differentiation, 36
numerical derivative, 54
O
0, 326
Oct (octal), 35, 327
Oct (to octal), 327
octal integer, 326
octal number base, 35, 66
OneVa (OneVar), 189, 191, 327
operation
second, 22
operator
entering, 25
or (Boolean), 69, 328
order of operations, 56
order-of-evaluation rules, 20, 62
Outpt(, 217, 329
OVERW (overwrite), 241
Index

P
P2Reg (quadratic regression), 190, 330
P3Reg (cubic regression), 190, 331
P4Reg (quartic regression), 190, 332
panning, 90
Par, 74
Param (parametric mode), 35, 239, 333
parametric equation
deleting, 127
graphing, 126
selecting and deselecting, 127
parametric graphs, 74
default graph style, 126
defining, 125
displaying, 128
drawing, 130
equation editor, 126
free-moving cursor, 128
graph format, 128
graph tools, 128
mode, 35, 126
tracing, 128
window variables, 127
Zoom, 129
parentheses, 20, 25, 56, 61, 397
pause, 26, 333
Pause (PRGM CTL menu), 219
pause indicator, 26
PC
linking to, 235
PEN, 105
percent (%), 334
permutations of items, 50
pEval(), 52, 334
phone (TI Customer Support), 392
pi, 59
PIC (picture names), 43
PIC variable
entering, 76
storing graph, 102
pictures
recalling, 102
saving, 102
pixel resolution
for function graphs, 81
PIOff (plot off), 195, 334
PIOn (plot on), 195, 334
PLOT1, 195
Plot1(), 335
PLOT2, 195
Plot2(), 335
PLOT3, 195
Plot3(), 335
plotting functions, 9, 11
plotting statistical data, 194
points
drawing, 108
turning on and off, 108
Pol (polar mode), 35, 74, 239, 336
Pol (to polar), 71, 174, 336
polar angle of complex number, 72
polar complex (), 336
polar complex mode, 35, 336
polar complex number form, 20, 70
polar equation
tracing, 120
polar graphs, 74, 84
default graph style, 118
defining, 117
displaying, 119
drawing, 122
equation editor, 118
free-moving cursor, 119
graph format, 119
graph tools, 119
polar graphs (continued)
mode, 35
trace cursor, 120, 121
tracing, 120
window editor, 118
Zoom, 121
PolarC (polar complex mode), 35, 336
PolarGC (polar graph coordinates), 84, 336
poly, 337
polynomial coefficient
storing to a variable, 212
polynomial root
storing to a variable, 212
polynomial root-finder, 211
polynomial value, 52
power of 10 (10^), 20, 34, 337
PRegC, 193
previous entries, 8
re-executing, 19
retrieving, 28
Index

reusing, 28
PRGM (program names), 43
PRGM CTL menu, 218
PRGM I/O (Input/Output) menu, 215
PRGM menu, 214
prod (product), 52, 160, 338
program editor, 214
menus and screens, 215, 220
program flow, 56
programming
assembly language, 225
calling a program, 224
copying a program, 225
creating programs, 214
defined, 214
deleting a program, 223
downloading assembly programs, 225
editing a program, 223
entering a command line, 220
getting started, 214
interrupting program, 222
running program, 221
using variables, 225
Prompt (PRGM I/O menu), 216, 338
prompts, 22
Eval x, 76
Name, 22, 39, 76
Rcl, 42
Sto, 212
PTCHG, 105
PToff, 105, 108
PtOn, 103, 340
PwR (power regression), 190, 339
PxChg, 103, 340
PxOff, 103, 340
PxOn, 103, 340
PxTest, 103, 340
Q
Q'n equation variables, 135
Qr11, 193
Qr13, 193
Quick Zoom, 91
in parametric graphing, 129
in polar graphing, 120
Quick-Find Locator (A to Z Reference), 262
R
r (radian entry), 341
rAdd, 184
rAddI, 340
Radian (angle mode), 35
radian angle mode, 75, 341
radian complex-number mode, 70
radian entry (?), 341
rand (random), 50, 341
randBin (random binomial), 50, 341
randInt (random integer), 50, 342
randM (random matrix), 184, 342
randNorm (random normal), 50, 342
random number, 50
RCGDB (recall graph database), 76, 88, 343
RecPic (recall picture), 76, 102, 343
RCPIC menu, 76
REAL, 43, 175, 185, 343
REAL (to real number), 156, 170, 179
real number variables, 43
real numbers, 29
real portion of complex number, 71
Rec (to rectangular), 71, 174, 343
recalling variable values, 18, 42
receiving transmitted data, 241
rectangular complex mode, 35
rectangular complex numbers, 70
rectangular complex-number form, 20
rectangular graph, 84
rectangular vector coordinates, 36
RectC (rectangular complex), 35, 344
RectGC (rectangular graph coordinates), 84, 344
RectV (rectangular vector coordinate mode), 36, 344
RECV (LINK menu), 236
RECV (LINK SND85 menu), 240
redefining user-created
constants, 60
ref (row echelon form), 184, 344
regression models, 191
relational functions, 55, 56
RENAME (rename), 241
Repeat (PRGM CTL menu), 218, 345
replacing batteries, 16
resetting memory, 232
result, 20, 24
result of last expression, 26
Return (PRGM CTL menu), 219, 345
RK (Runge-Kutta) method, 133, 345
row
of matrix, 181
rref (reduced row echelon), 184, 348
rSwap( (row swap), 184, 348
running a program, 221
S
Scatter (stat plot type), 349
Sci (scientific notation), 20, 34, 349
scrolling, 19
seed value, 50
SECT, 112
SELECT, 77
Select(, 161, 350
selection cursor, 38
SEND (LINK menu), 236
SEND WIND screen, 238
Send(, 216, 350
separator, 70
seq (sequence), 52, 160, 351
SegG (sequential graphing), 84, 351
series of instructions
displaying, 18
SetLE, 159
SetLEdit, 161, 351
setting graph format, 83
setting graph style, 80
Shade(, 103, 104, 352
shading
pattern, 104
resolution, 104
shading patterns, 80
ShftL (shift left), 69, 353
ShftR (shift right), 69, 353
ShwSt (show string), 354
sign, 49, 354
SimulG (simultaneous
graphing), 84, 354
SIMULT ENTRY menu, 208
SIMULT order screen, 208
SIMULT RESULT menu, 209
simult(, 210, 354
simultaneous equation solver, 208
sin (sine), 48, 186, 355
sin⁻¹ (arcsine), 48, 355
sine
calculating, 3
sinh (hyperbolic sine), 51, 356
sinh⁻¹ (inverse hyperbolic sine), 51, 356
SinR (sinusoidal regression), 190, 193, 357
SKIP, 241
SlopFld (slope field), 134, 358
Smart Graph, 86
drawing tools, 102
in GRAPH MATH, 95
in Graph Zoom, 94
SND85 (LINK menu), 236
solution method formats, 133
solutions
drawing, 148
SOLVE, 205
solver graph, 207
Solver menu, 206
Solver ZOOM menu, 208
Solve(, 358
solving differential equations, 139
solving for unknown variable, 206
sortA, 159, 359
sortD, 159, 359
Sortx, 160, 359
Sorty, 160, 359
Sph (to spherical), 174, 360
SphereV (spherical vector coordinate mode), 36, 360
square (\(\times\)), 360
square root (\(\sqrt{\cdot}\)), 7, 360
S\(\rightarrow\)Eq (string to equation), 227, 361
STAT (statistical result variables), 43
STAT CALC (Calculations) menu, 189
STAT menu, 188
Stat Plot
  changing on/off status, 81
  setting up, 195
  turning on and off, 195
STAT PLOT menu, 195
STAT PLOT status screen, 194
STAT VARS (Statistical Variables) menu, 192
statistical analysis, 188
results, 192
statistical data
  entering, 189
  plotting, 194, 195
STGDB (store graph database), 76, 88, 361
STOa, 210
STOb, 210
Stop, 219, 362
Store, 18
store symbol, 22
store to variable (\(\rightarrow\)), 362
storing a graph display, 102
storing data, 39
storing equation coefficients, 210
storing equation results, 210
STOX, 210
STPIC (store picture), 76, 88, 362
STPIC menu, 76
StReg (store regression equation), 190, 362
string, 29
  concatenating, 226
  creating, 226
  defined, 226
  storing, 226, 227
string entry, 363
STRING (string variables), 43
STRING (String) menu, 227
STYLE, 77
sub (subset of string), 227, 363
submatrix
displaying, 181
subroutines, 224
subtraction (\(-\)), 363
sum, 52, 160, 364
sum of elements of list, 52
Sx (statistical result variable), 193
syntax error, 27
syntax of function, 25
syntax of instruction, 25
T
T (transpose), 367
table, 110
  clearing, 114
  displaying, 110
  navigating, 111
  setting up, 113
  setup editor, 113
TABLE menu, 110
Table menus, 112
table setup editor, 113
tan (tangent), 48, 364
tan\(^{-1}\) (arctangent), 48, 365
tangent line
drawing, 107
tanh (hyperbolic tangent), 51, 365
tanh\(^{-1}\) (inverse hyperbolic tangent), 51, 365
TanLa, 103, 107, 366
TBLST (table setup editor), 112, 113
test menu, 55
text, 105
Text, 366
Then, 218, 305, 306
ti-GRAPH LINK, 235
tMax, 127, 136
tMin, 127, 136
TOL (Tolerance Editor), 398
tPlot, 136
trace, 88
TRACE (cursor), 75
Trace (Graph menu), 367
TRACE (Solver menu), 207
trace cursor, 75, 90, 144, 205
  in parametric graphing, 128
  in polar graphing, 120
  moving, 90, 121, 129
  panning, 90
  Quick Zoom, 91
stopping and resuming, 91
transmitting data, 234, 240
error conditions, 242
insufficient memory, 242
transmitting data (continued)
repeating to several devices, 242
selecting variables, 238
window variables, 239
transpose (T), 367
tStep, 127, 136, 138
turning off TI-86, 2, 17
turning on TI-86, 2, 17
TwoVa (TwoVar), 189, 368

U
unevaluated expression
storing, 9, 40
units of measure
converting, 61
unit-to-unit cable, 234, 235
unitV (unit vector), 173, 368
unknown variable
solving for, 206
upper menu, 32
selecting an item, 33
user-created constants, 43, 58, 60
user-created zoom variables, 239
V
value, 24, 25, 29
variable, 21
classifying as data types, 42
creating, 41
deleting, 45
displaying, 41
in expressions, 4
in table screen, 111
names, 44
recalling, 42
storing data to, 39
storing results to, 3, 30
uppercase and lowercase names, 39
x variable, 77
y variable, 77
variable equations
in a table, 114
VARS CPLX (complex variables) screen, 71
VARS EQU menu, 203
voli (vector to list), 160, 174, 369
vector, 29
brackets [ ], 369
complex, 171, 180
creating, 170
defined, 168
deleting from memory, 170
displaying, 171
editing dimension and elements, 172
forms, 168
operations, 173
using in an expression, 172
with math functions, 176
vector coordinate modes, 36
vector editor, 168
Vector Editor menu, 170
vector entry [ ], 369
VECTR (vector names), 43
VECTR CPLX (Complex) menu, 175
VECTR MATH menu, 173
VECTR menu, 169
VECTR NAMES menu, 169
VECTR OPS (Operations) menu, 173
VERT (vertical line), 104, 106, 369
W
warranty information, 400, 402
While, 218, 369
WIND (Solver menu), 206
WIND (window variables), 43, 35, 75, 238
window editor, 75
polar, 118
window variables, 82
Δx and Δy, 83
changing, 12, 82
differential equations, 135
graph screen, 81
X
x variable, 77
XMIT (transmit), 237, 240
Xor (Boolean), 69, 370
xRes (resolution), 81
xScl (scale), 81
xStat (x-variable list), 189
<table>
<thead>
<tr>
<th>xyline, 370</th>
</tr>
</thead>
</table>
| **Y**
| y variable, 77 |
| y(x)=, 75 |
| YICPT (y-intercept), 96, 100 |
| yScl (scale), 81 |
| yStat (y-variable list), 189 |
| **Z**
| ZData, 371 |
| ZDATA (GRAPH ZOOM menu), 92 |
| ZDecm, 372 |
| ZDECM (GRAPH ZOOM menu), 92 |
| ZFACT (ZOOM FACTOR), 92, 208 |
| ZFit, 129, 373 |
| ZFTT (GRAPH ZOOM menu), 92 |
| ZIn (zoom in), 373 |
| ZIN (zoom in), 92, 208 |
| ZInt, 374 |
| ZINT (GRAPH ZOOM menu), 92 |
| ZOOM, 14, 75, 88 |
| custom, 93 |
| parametric graphs, 129 |
| polar graphs, 121 |
| ZOOM operations, 147 |
| ZOOM window variables storing and recalling, 95 |
| ZOOMX (GRAPH ZOOM menu), 92 |
| ZOOMY (GRAPH ZOOM menu), 92 |
| ZOUT (zoom out), 92, 208, 375 |
| ZPREV (zoom previous), 92, 375 |
| ZRCL (GRAPH ZOOM menu), 92, 95 |
| user-created zoom variables, 239 |
| ZRcl (zoom recall), 376 |
| ZSqr, 376 |
| ZSQR (GRAPH ZOOM menu), 92 |
| ZSTD (GRAPH ZOOM menu), 92 |
| ZSTD (standard defaults), 208, 377 |
| ZSTO (GRAPH ZOOM menu), 92, 95 |
| ZTrig, 378 |
| ZTRIG (GRAPH ZOOM menu), 92 |